SMART E-TAX INVOICE GENERATOR

By

Atita Jityanan

Jiratchaya Fuengthanakul

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN ROBOTICS AND AI
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2022
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG PROJECT CERTIFICATE

Project Title	Smart E-Tax Invoice Generator
Student Name	Atita Jityanan Jiratchaya Feungthanakul
Degree	Bachelor of Engineering in Robotics and AI
Project Advisor	Signed: _____ (Prof.Dr. Pitikhate Sooraksa)

Project	Smart E-Tax Invoice Generator
Student	Atita Jityanan
	Jiratchaya Fuengthanakul
Degree	Bachelor of Engineering in Robotics and AI
Project Advisor	Prof.Dr. Pitikhate Sooraksa
Academic Years	2022

Abstract

This thesis presents Smart e-Tax Invoice Generator, which is designed to streamline the process of generating tax invoices electronically. This web-based solution offers a range of features and functionalities to simplify and automate invoice generation, ensuring compliance with tax regulations. Key features of the e-tax invoice generator include accurately generated tax invoice matched with customer's order as requested, automated tax calculations, certified generated tax invoice with digital signature and secure storage and retrieval of generated invoices for further tax operation for the company. These features collectively enable business, Bringhome.theBacon our company participants who are under process of Company Limited registration, to save time and effort in generating accurate tax invoices and further operation after completing the registration process adhere with tax guidance. The project was tested using actual customer's order transactions and accurately generated tax invoice for individual customers.

Keywords: Tax invoice, Web application, Digital Signature

Acknowledgment

We would like to express our gratitude to Prof.Dr. Pitikhate Sooraksa, project advisor who guided us throughout the project from the beginning to the end. We are grateful for his dedication and willingness to invest his time and knowledge in my academic development.

Secondly, my sincere thanks to Nutnicha Sirorattankul, the representative of the company participating for this project and who is also our friend. Thank you for spending their precious time to give us an opportunity to explore and initiate a real use case for their company, without their support this project wouldn't be this complete. Their expertise , and insightful feedback have played a crucial role in shaping the direction and quality of this thesis.

I, Atita Jitayanan, would like to take this opportunity to express my gratitude to my project college, Jiratchaya. The project would not have been completed without her cooperation and hardworking guidance. Not all people will contribute themselves to the work and as much as her, I really appreciate and am grateful to work with her on our last project as a college student.

For the past three months, while working on this project, I, Jiratchaya Fuengthanakul, would like to express my sincere gratitude for the time we have spent together overcoming various obstacles. She is an incredibly meticulous individual who consistently identifies any issues within the project and is adept at finding suitable solutions. Furthermore, our collaboration has resulted in a remarkable boost in productivity.

Lastly, we would also like to say thank you to our lecturers at King Mongkut's Institute of Technology Ladkrabang, who supported us and offered deep insight into this project.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background and significant

With the rapid advancement of technology and the increasing accessibility of the internet, global e-commerce has become one of the fastest-growing industries in the world. The shift towards online shopping has been driven by factors such as convenience, greater product variety, and competitive pricing. According to Lertad Supadhiloke, Director of e-Commerce at LINE Thailand, speaking at the LINE Shopping Social Commerce 2022 event, "the social commerce market of LINE SHOPPING is having Gross Merchandise Value around 252% in 2022" ('ก้าวกระโดดอย่างมั่นใจในยุค Social Commerce ด้วย LINE SHOPPING' session on main stage, August 22th, 2022). The predicted statistic isn't temporary growth but customer behavior on digital transformation was consistent. E-commerce has revolutionized the way businesses operate, enabling them to reach a wider customer base and sell their products and services online.

However, with the rise of e-commerce, many businesses have registered as verified company and are capable of compliance with various business regulations including Value Added Tax (VAT) registration as states from Revenue Department of Thailand, "Business operators who have income from online product sales are required to submit Form PP30 to pay monthly value-added tax (VAT), regardless of whether they generate income from operating a business or not" (para. 9). The complexity of managing and generating tax invoices has also increased. Generating tax invoices manually can be a time-consuming and error-prone process, especially for businesses that deal with many transactions according to an interview with Nutnicha Siroratthanakul, a selected-company representative, "On February 2nd campaign (2.2) sales on one of the e-commerce platform, they've got more than 100 transactions on that day" (Interview, February 7th, 2023). This has led to the need for an automated solution that can generate tax invoices quickly and accurately, while also ensuring compliance with tax laws and regulations.

To address this need, the e-tax invoice generator project was initiated. The project aims to provide an automated solution that simplifies the invoicing process for businesses operating in the e-commerce sector. The e-tax invoice generator will be integrated into e-commerce platforms, enabling businesses to generate tax invoices automatically as soon as a transaction is completed.

In summary, the e-tax invoice generator project is a response to the challenges faced by businesses in the e-commerce sector in generating tax invoices. The project aims to

provide an automated solution that simplifies the invoicing process, reduces errors, and ensures compliance with tax regulations.

1.2 Research Objectives

The main objective is to develop a web application that automates the process of generating tax invoices for businesses. The web application will be designed to generate electronic invoices with digitally signed signatures that are compliant with the tax laws and regulations of the jurisdiction and capable of verification.

1.3 Scope of Research

1.3.1 Interview and gather requirements related to the tax invoice operating process from one company.

1.3.2 Design and develop the web application for generating e-tax invoices that are compliant with the Revenue Department.

1.3.3 Testing the web application, including its effectiveness, accuracy, and usability on one business and discussing the development and improvement of the web application.

1.4 Methods of Conducting Research

1.4.1 Analyze the key features and functionality of the web application, including how it automates the process of generating tax invoices, digitally signed invoices, and ensures compliance with tax laws and regulations using design thinking process and reviewing relevant literature and research related to tax invoice generation, tax invoice structure and digital signature.

1.4.2 Design and develop the web application with the key features analyzed from design thinking process and research.

1.4.3 Present the results of testing the web application, including its effectiveness, accuracy, and usability on one business.

1.4.4 Discuss the limitations of the research with the company, receiving suggested future directions for the development and improvement of the web application.

1.5 Expected benefits

1.5.1 Aim to help businesses save time and resources by automating the invoicing process and ensuring accuracy and consistency in the generation of invoices compliant with the Revenue Department of Thailand.

1.5.2 Improve the overall efficiency of the invoicing process by reducing errors and streamlining the documentation and record-keeping processes.

Chapter 2
Review of Literature

2.1 Related theory

2.1.1 Value-Added-Tax

Value Added Tax (VAT) has been implemented in Thailand since 1992 replacing Business Tax (BT). VAT is an indirect tax imposed on the value added of each stage of production and distribution. Value Added Tax (VAT) has a general rate of 7%. Certain activities such as export of goods, services rendered in Thailand and utilized outside Thailand, international transportation by aircraft or sea-vessels, supply of goods and services to government agencies or state-owned enterprises under foreign-aid program, supply of goods and services to the United Nations and its agencies, as well as supply of goods and services between bonded warehouses or between enterprises located in EPZs, are liable to VAT at a zero percent rate.

For the scope of the research, the selected-company is an e-commerce business, selling products mainly online, thus their selling operation is within the scope of commercial or professional activity and is therefore subject to value-added tax (VAT) in accordance with section 77/2(1) of the Revenue Code. The income received must be calculated as the taxable base for VAT purposes when delivering goods to customers, unless payment is received before the delivery of goods as the following condition:

2.1.1.1 Any person or entity who regularly supplies goods or provides services in Thailand and has an annual turnover exceeding 1.8 million baht is subject to VAT in Thailand must register to be VAT registered person or entity (Form VAT 01) before the operation of business or within 30 days after its income reaches the threshold. The registration application must be submitted to Area Revenue Offices if the business is situated in Bangkok or to the Area Revenue Branch Offices if it is situated elsewhere. Should the taxpayer have several branches, a registration application must be submitted to the Revenue Office where the headquarter is situated.

2.1.1.2 Any person or entity who regularly supplies goods or provides services in Thailand and has an annual turnover not exceeding 1.8 million baht is exempt from VAT in accordance with section 81/1 of the Revenue Code and does not have to register for VAT or pay VAT. In case of any entity purpose to register for VAT, they must notify the Director General of the Revenue Department and must complete the registration and payment within 30 days of the notification.

In conclusion, a registered entrepreneur who earns income from online sales is required to submit Form P.P.30 to pay monthly value-added tax regardless of whether they have income from business operations or not.

2.1.2 e-Tax Invoice and e-Receipt

2.1.2.1 Definition of e-Tax Invoice and e-Receipt

e-Tax Invoice and e-Receipt are the acronym for the process of preparing tax invoices, credit notes, debit notes, and receipts in electronic formats, which includes a digital signature. These documents must be submitted to the Revenue Department through designated channels or by using a method that prepares tax invoices, credit notes, and debit notes with a time stamp. The documents should be sent via email to the purchaser or recipient of the goods or services. Business operators must choose one system to use.

2.1.2.2 Type of e-Tax Invoice and e-Receipt operation

The Department of Revenue in Thailand provides two ways for business operators to prepare electronic tax invoices, which are the e-Tax Invoice & e-Receipt system and the e-Tax Invoice by Email system.

Business operators can choose to prepare electronic tax invoices in 2 ways:

(1) e-Tax Invoice & e-Receipt system: Registered Value Added Tax (VAT) operators or those responsible for issuing receipts (with no income limit) can prepare digital signature-required electronic tax invoices and deliver them to buyers or service recipients. They also have the responsibility to send the data to the revenue department.

(2) e-Tax Invoice by Email system: Registered VAT operators with an income not exceeding 30 million baht per year can prepare electronic tax invoices and send them via email to buyers or service recipients, along with a CC copy to the central system for time-stamping. The system automatically sends the data to the revenue department.

1) e-Tax Invoice and e-Receipt

Business operators must prepare electronic tax invoices or electronic receipts with a digital signature and deliver them to buyers or service recipients via electronic means or as agreed upon. They also have the responsibility to send the data to the revenue department through the prescribed channels by the 15th day of the following tax month. Buyers or service recipients who receive electronic tax invoices or electronic receipts must check the accuracy of the documents and keep the electronic data as required by law.

Figure 1 Electronic tax invoices and electronic receipts operation to deliver to buyers or service recipients.

Steps for creating and submitting electronic tax invoices and electronic receipts

(1) Obtain an electronic certification from a reliable electronic service provider (Certification Authority: CA) under the certification of the Thailand National Root Certification Authority (Thailand NRCA), which is managed by the Electronic Transactions Development Agency (Public Organization) or ETDA.
(2) Register through the registration program and verify the digital signature.
(3) Create an electronic tax invoice or electronic receipt with a digital signature and send it to the buyer or recipient via electronic means, as stipulated by the Electronic Transactions Act.
(4) Prepare the electronic tax invoice or electronic receipt in XML file format (ETDA 3-2560) with a digital signature, and submit the data to the Revenue Department via upload or host-to-host, or through an electronic data service provider (host-to-host and electronic data service provider submissions must meet the criteria and conditions set by the Revenue Department).
(5) Track and verify the results of data submission via the tracking system.
(6) Safely and accurately store the electronic data as required by law.

2) e-Tax Invoice by Email System

Businesses registered for value-added tax (VAT) with an annual revenue of not more than 30 million baht and not yet ready to enter the electronic tax invoice system or electronic receipt system with digital signature can create e-tax invoices using the e-Tax Invoice by Email system. The system generates the tax invoice file in PDF/A-3 format and sends the draft tax invoice via email to the buyer along with a CC copy to the central system email address: cse@etax.teda.th. The central system will then stamp the time on the tax invoice and send the certified e-tax invoice (with time stamp) to both the buyer and the seller as evidence.

2.1.3 Tax Invoice

2.1.3.1 Definition of Tax Invoice

A tax invoice is an important document that registered value-added tax (VAT) operators must prepare and issue to buyers or service recipients every time they sell goods or provide services. The purpose of this document is to show the value of the goods or services and the amount of VAT that the registered operator is required to collect or charge from the buyer or service recipient each time.

2.1.3.2 Tax Invoice Operating Process

1) In the case of selling goods, registered value-added tax (VAT) operators are required to issue a tax invoice and deliver it to the buyer immediately upon delivery of the goods or when the right to the goods is transferred to the buyer before delivery, or when payment for the goods is received before delivery, depending on the circumstances.
2) In the case of providing services, registered VAT operators are required to issue a tax invoice and deliver it to the service recipient immediately upon receiving payment for the services or when the services are used, whether by themselves or by others, depending on the circumstances.

In addition, registered operators must prepare and keep a copy of the tax invoice at the place of business or other locations specified by the Director-General for a period of not less than 5 years.

2.1.2.3 Tax Invoice Issuer

1) A registered business operator who is liable for value-added tax, calculated from sales tax deducted by purchase tax.
2) A non-governmental market vendor who sells the property of a registered business operator, and issues a tax invoice in the name of the registered business operator under section 86/3 and the notification of the Revenue Department No. 87/2542.
3) An agent in the Kingdom of the registered business operator in the Kingdom, with the establishment of agents for sales and delivery of goods to agents. This is only for agreements appointing agents for the type of product under section 86 paragraph four.
4) An agent in the Kingdom of a registered business operator located outside the Kingdom who has submitted a request for approval to the Director-General of the Revenue Department under section 86/2. The agent must issue a tax invoice in the name of the registered business operator located

outside the Kingdom according to the criteria, methods, and conditions specified by the Director-General's Order.

5) A business operator located outside the Kingdom who comes to conduct sales of goods or services in the Kingdom on a temporary basis under section 85/3 and section 86 paragraph two.
6) A registered business operator who has been removed from the value-added tax register due to ceasing operations or the Director-General ordered the withdrawal of the value-added tax registration. The Director-General of the Revenue Department shall allow the business operator who has been removed from the value-added tax register to issue a tax invoice, additional invoice, or credit note temporarily until the business is stopped under section 86/11.
7) A business operator who conducts business in the Kingdom and has not registered for value-added tax under section 83 or 84, but issues a tax invoice for the sale of goods or services in the name of the purchaser.

2.1.2.4 Type of tax invoice

1) Full Tax Invoice (Section 86/4)
2) Simplified Tax Invoice (Section 86/6)
3) Debit Note (Section 86/9)
4) Credit Note (Section 86/10)
5) Receipt issued by government agencies for marketing or other purposes as per Section
6) Receipt issued by the Revenue Department for Value Added Tax payment according to Section 83/6 or Section 83/7 (Section 86/14)
7) Receipt issued by the Customs Department or the Excise Department for the collection of Value Added Tax for the Revenue Department (Section 86/14)
8) In this section, we will only discuss Full Tax Invoices and Simplified Tax Invoices, which will be further explained in the next topic.

2.1.2.5 Complete Tax-Invoice structure

1) The words "Tax Invoice" must be prominently displayed.
2) The phrase "Tax Invoice" is required by law to be included in the document to indicate that it is a tax invoice. Additionally, if the registered business owner wishes to prepare a full tax invoice along with other commercial documents, such as receipts, delivery notes, or invoices, which are all contained in the same set, the tax invoice is not the first document in the set. In this case, the following should be done:

(a) The words "Document Issued as a Set" must be included in both the tax invoice and the duplicate of the tax invoice in the set.

(b) The phrase "Copy of Tax Invoice" must be included in the duplicate of the tax invoice. The phrases "Document Issued as a Set" and "Copy of - - - Tax Invoice" must be printed or created using a computer system. In the case of a tax invoice prepared using a computer system, it must be stamped with a rubber stamp, written with ink, typed, or made to appear in the same way using other methods.

3) The list of terms "name, address, and taxpayer identification number of the operator registered for value-added tax and the registration number for tax invoice issuance."

(a) The name of the tax invoice issuer refers to the name of the operator registered for value-added tax, or the name of the business establishment registered for value-added tax, or the trade name of the business establishment registered for value-added tax. The name of the tax invoice issuer cannot be abbreviated.

In the case where the tax invoice issuer or the recipient of the tax invoice is a juristic person, the following abbreviations may be used to represent the status:
(1) Company Limited - Co., Ltd.
(2) Public Company Limited - PCL.
(3) Limited Partnership - LP.
(4) Ordinary Partnership - OP.

(b) The address of the tax invoice issuer refers to the location of the business establishment registered for value-added tax as specified in the VAT registration (Form VAT 20).

The Revenue Department has issued an announcement by the Director-General of the Revenue Department regarding value-added tax (Version 199) dated December 26, 2013, stipulating that the registration applicant who has completed the full form of tax invoice issuance must specify the details of the business establishment, which is the place of sale of goods or services, on the tax invoice.

(1) In the case where the registered address of the seller or service provider in the Value Added Tax registration form (Form VAT 20) is a head office, the message "Head Office" or an abbreviation such as "HO" or "HQ" should be specified to indicate that it is a head office, or a five-digit zero-filled number (00000) should be specified to indicate that the number 00000 is the code for the head office, which should be stated on the invoice.

(2) In the case where the registered address of the seller or service provider in the Value Added Tax registration form (Form VAT 20) is a branch, the message "Branch No. ..." should be specified, with the branch number being the same as the one registered in the Value Added Tax registration form. For example, "Branch No. ..." "br.no ..." or a five-digit number should be specified to indicate the code for "Branch No. ...", such as 00001, and stated on the invoice.

Observation note:
The text in (1) and (2) can be printed or created using a computer system, stamped with a rubber stamp, or written with ink or typewriter. It should be noted that for invoices made since January 1, 2015, a 13-digit taxpayer identification number is required for all taxpayers, whether they are individuals, partnerships, or corporations, instead of the previously used 10-digit identification number when submitting tax return forms.

Note: Due to the fact that some vendors or service providers who are registered for value added tax (VAT) and some purchasers or recipients of goods or services still have misunderstandings in practice regarding the identification of the purchaser's or recipient's taxpayer identification number in a full tax invoice according to the announcement of the Director-General of the Revenue Department regarding value added tax (199th edition) dated December 26, 2013, the Revenue Department has issued the following clarification:

(1) Vendors or service providers who are registered for VAT must specify the purchaser's or recipient's taxpayer identification number in a full tax invoice only if the purchaser or recipient is a registered VAT operator. If the purchaser or recipient is not a registered VAT operator, the vendor or service provider is not required to specify the purchaser's or recipient's taxpayer identification number in the tax invoice. In cases where vendors

or service providers who are registered as VAT operators have notified customers who are purchasers or recipients of goods or services about their taxpayer identification number by any means such as posting a notice or notifying customers in general or individually or by any other means, if the purchaser or recipient who is a registered VAT operator does not inform the vendor or service provider of their taxpayer identification number or informs the vendor or service provider that they do not have a taxpayer identification number, the vendor or service provider may not specify the taxpayer identification number of the purchaser or recipient in the tax invoice.

(a) If the buyer or service recipient does not provide the tax identification number and the details regarding the aforementioned establishment and receives a full tax invoice that does not include these details, the input tax according to the tax invoice cannot be used in the calculation of value-added tax, in accordance with Section 82/5 (2) of the Revenue Code.

(b) If such omission is intended to evade taxes, it is considered a violation of the law and subject to penalties.

(c) If the buyer or service recipient is not a registered value-added tax person, they do not need to provide their own tax identification number to the seller or service provider.

(d) In the case of a "serial number of the tax invoice and the number of the book, if any," a tax invoice without a serial number cannot be used in the calculation of input tax, and in this case, the issuer of the tax invoice is considered to have committed an offense by failing to include important details in the invoice, and is subject to a fine of no more than 2,000 baht.

4) The list "Name, Type, Category, Quantity, and Value of Goods or Services" The name, type, and category of goods or services should be specified in the list, except in cases where the name, type, and category of goods or services that are not subject to value-added tax need to be specified in the tax invoice. This can be done by arranging a symbol or a separate list item clearly indicating that it is a tax-exempt goods or services.

5) The list "Amount of Value-Added Tax Calculated from the Value of Goods or Services, Clearly Separated from the Value of Goods or Services" The amount of value-added tax calculated from the value of goods or services should be clearly separated and indicated in the list item.

6) The list "Date of Issuance of Tax Invoice"

The date of issuance of tax invoice is an important item that the Revenue Department requires to be included in the tax invoice. It is an indication of the date when the tax liability for value-added tax arises, which is the date when goods are delivered or ownership of goods is transferred to the buyer, or when services are used by the taxpayer or other persons, or when payment for goods or services is received. The date of issuance of tax invoice can be indicated using numeric values to represent the month, and can be expressed in either Buddhist Era (B.E.) or Common Era (A.D.).

In the case where a registered operator wishes to prepare a full tax invoice together with other commercial documents such as receipts, delivery notes, and invoices, which are multiple copies in the same set, and the tax invoice is not the first document in the set, the following must be done:

(a) The words "documents are issued as a set" must be included in both the tax invoice and its duplicate copy.
(b) The words "copy of tax invoice" must be included in the duplicate copy of the tax invoice.
(c) The words "documents are issued as a set" and "copy of tax invoice" must be printed or prepared by computer. If the tax invoice is prepared by computer, the rubber stamp, ink, typewriter, or other method must be used to imprint these words in the same manner.

If a registered operator has multiple places of business, and tax invoices of the places of business that are not the head office are sent to the buyers or service recipients every time goods are sold or services are provided, the words "branch issuing tax invoice is..." must be included in such tax invoices. This message must also be printed or prepared by computer and stamped with a rubber stamp, written in ink, typed, or displayed by other methods in the same manner.

2.1.2.6 Guidelines of tax invoice preparing procedure

The following guidelines outline the proper procedure for preparing a tax invoice:

1) The tax invoice must be written in Thai language. In the event that it is also prepared in English, it should be included in the same document. If it is necessary to use a foreign language, permission from the Director-General of the Revenue Department must be obtained.

2) The currency unit used in the tax invoice must be Thai Baht and the numbers should be presented in Thai or Arabic numerals. If a foreign currency is to be used, permission from the Director-General of the Revenue Department must be obtained.

3) It is permissible to issue a single tax invoice for multiple sales or services provided.

4) The tax invoice must include a complete and comprehensive list of items sold or services provided.

5) Any alterations or changes, including erasing, deleting, or adding to the invoice, are strictly prohibited. Any modifications or changes to the invoice will render it invalid for tax purposes, and it will be considered a violation of tax regulations.

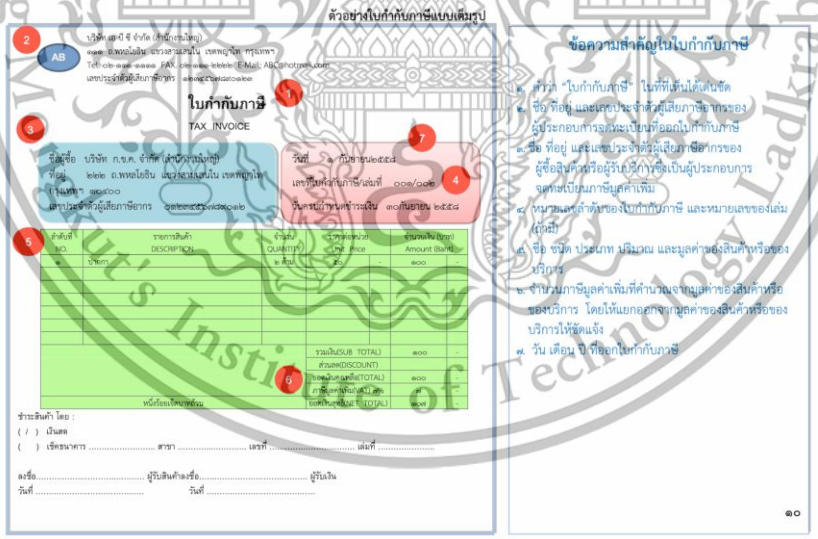

Figure 2 Example of full tax invoice guideline

2.1.4 Digital Signature

2.1.4.1 Definition of Digital Signature

The act of electronically signing using an electronic certificate that identifies a natural person or legal entity as the owner of the signature is intended to demonstrate that such person or entity has accepted the contents of the electronic data. The use of electronic signatures via PKI technology and electronic certificates, including e-Tax Invoice and e-Receipt, is a means of providing assurance to those involved in electronic transactions that the signer using PKI technology is the actual person who completed the transaction or accepted the contents of the electronic data. Such electronic signatures cannot be repudiated, and it can also be verified whether the electronic data has been altered after the signature was affixed. This is in accordance with Sections 9 and 26 of the Electronic Transactions Act B.E. 2544.

2.1.4.2 Type of Digital Signature

Type of digital signature	Example of digital signature	Example of security insurance	Example of necessary details of digital signature
Type 1 General Digital Signature	-Including signer name at the end of email	- Verify compliance with email security policies - Record email decision-making information in an appropriate document management system.	Signature details - Name that is printed at the end of the email content - Date and time of the signature or email submission - Identity of the signer - Method used to verify identity
	- Scanning an image of a handwritten signature and attaching it to a document. - Using a stylus to write a signature by hand on a screen and saving it.	- Matched the methods used by individuals to express their intentions with the important contextual information about the signature, such as the signature format, the signature process, and the date and time of the signature. - Matched the methods used by individuals to	Signature details - Name that is printed at the end of the email content - Date and time of the signature or email submission - Identity of the signer - Method used to verify identity

		express their intentions with the clear and explicit signature text. - Regularly conduct signature verification processes to demonstrate consistency in the use of signatures.	
	Automatic system with user authentication, combined with the format of an electronic signature of type 1.	- Perform regular checks on the system and its functions - Automatically record necessary context information if possible. - Perform appropriate processes to verify and authenticate individuals and control access to information. - Configure or control documents and content to prevent modification after signing. - Keep transaction records and send them to relevant parties as evidence of the transaction.	Signature details - Recording logs of user authentication events - Recording logs of intent expression during signature (such as button clicks or markings in the acceptance box) - Identity of the signer - Method used to verify identity
Type 2 Authorized Digital Signature	- Public Key Infrastructure (PKI) Digital signature	Managing digital certificates and key data in a Public Key Infrastructure (PKI) structure appropriately to ensure that the digital signature can confirm the identity of the signatory and detect changes to the message and electronic	Signature information - Name that is printed at the end of the email content - Date and time of the signature or email submission - Recording logs of user authentication events - Recording logs of intent expression during signature (such as button clicks or

		signature.	markings in the acceptance box)
Type 3 Authorized Digital Signature with Certificate from Certificate Authorities	Public Key Infrastructure (PKI) Digital signature with Certificate from Certificate Authorities	Managing digital certificates and key data in a Public Key Infrastructure (PKI) structure appropriately to ensure that the digital signature can confirm the identity of the signatory and detect changes to the message and electronic signature.	Signature information - Name that is printed at the end of the email content - Date and time of the signature or email submission - Recording logs of user authentication events - Recording logs of intent expression during signature (such as button clicks or markings in the acceptance box)

Table 1 Type of Digital Signature

2.1.5 Company Introduction

This section will provide an overview of company participants, mainly focusing on online business.

2.1.5.1 Company Background

Bringhome.theBacon is a brand's name of an online small business focusing on offering stationary and office supplies with fun and unique designs such as notebook, memo pad, sticker, and container. The brand was established in February 2019 and still goes on as founders saw a gap in the market for gift items that combined their love for cute stuff. With Charot, their mascot, as their inspiration, they created a brand that offers products that are both functional and aesthetically pleasing. While Bringhome.thebacon offers both online and offline storefronts to sell their products, the majority of their sales occur through online platforms.

Bringhome.theBacon is now currently in the process of registering as a Company Limited (Co. Ltd.) and aims to complete the registration by 2023. As a part of the registration process, the business needs to be in compliance with various business regulations including Value Added Tax (VAT) registration.

Figure 3 Bringhome.theBacon's logo

2.1.5.2 Business structure

1) Products

Bringhome.theBacon's product category includes a range of stationary and office supplies, such as folders, notepads, notebooks, and organizing containers with distinctively in competitive position from other competitors by their unique design and aesthetic. The list of some of the products from their category is illustrated as follows.

- Folder

Figure 4 Bringhome plastic folder

- Postcard/Poster

Figure 5 Candy shop Mini Poster

- Griptok

Figure 6 Bringhome Griptok

- Memo Pad

Figure 7 Bringhome A5 memopad

- Stickers

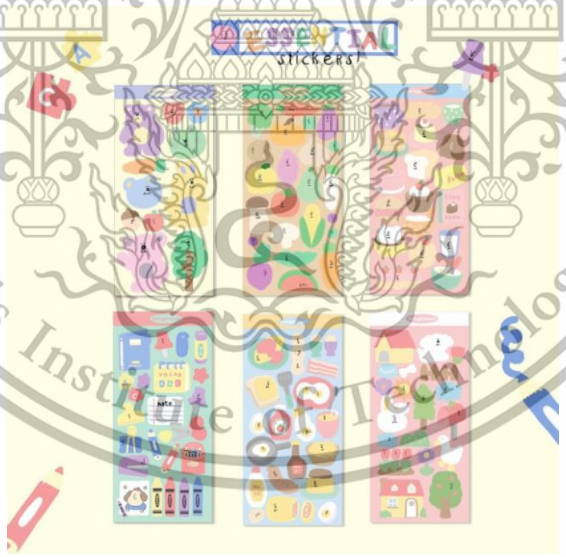

Figure 8 Bringhome Essential sticker

- Boxes and storage

Figure 9 Bringhome essential can

- Miscellaneous

Figure 10 Mickey&Milly's American Diner Cap

Figure 11 Bringhome AirPods Case

2) Sales and Marketing Strategy

- Contact Channel

Bringhome.theBacon has three main social media platforms both for communicating and selling purposes as declared below.

- Instagram

Bringhome.thebacon's primary platform for marketing their products is Instagram. The brand keeps its followers informed by regularly posting about new product releases as well as allowing customers to publicly review their products by tagging them.

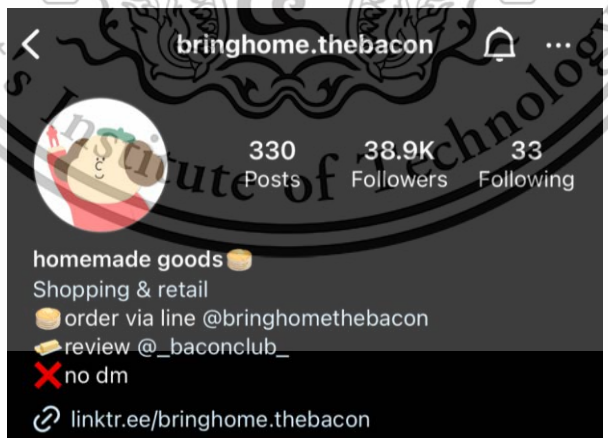

Figure 12 @bringhome.thebacon account on Instagram

- LINE Official Account

Bringhome.thebacon has established a LINE Official account to serve as a customer service point, enabling customers to make inquiries about their products. The use of Instagram's direct message channel (DM) for customer inquiries is prohibited, as messages sent through this channel are prone to getting lost, whereas the LINE Official account messages are more structured and easier to manage. Consequently, Bringhome.thebacon has chosen to prioritize the utilization of their LINE Official account as their primary customer service channel.

Figure 13 @bringhome.thebacon LINE Official Account

- Website

Bringhome.thebacon (https://www.bringhome-thebacon.com) also serves its customers through its website. The website is designed to be user-friendly, with clear categories and a simple layout that makes it easy to browse for products. The main purposes of the website are to provide information about the products and promote them based on their categories, as well as to provide contact information, an "About" section for Bringhome.thebacon, frequently asked questions (FAQs), and a suggestion form.

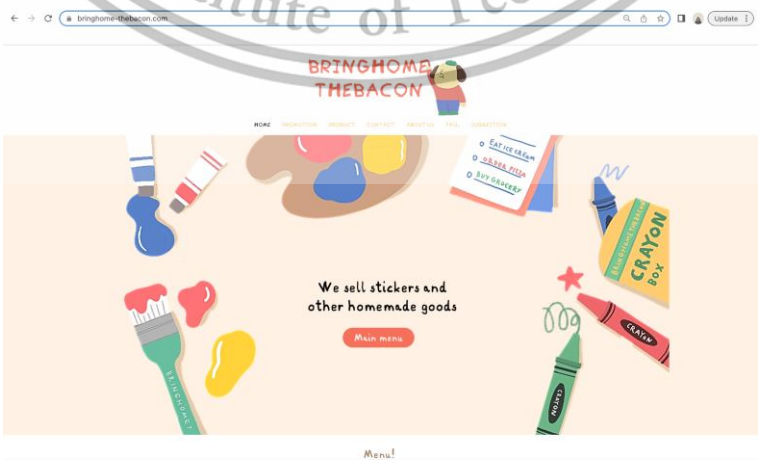

Figure 14 https://www.bringhome-thebacon.com UI example

- Selling Platform

Bringhome.theBacon's selling platform comprises both offline and online storefronts, each of which will be examined in more detail below.

- Offline storefront
  - Multi-brand store

In addition to offering their products online, Bringhome.thebacon has also partnered with several multi-brand stores to expand their customer reach. Some of the stores that carry Bringhome.thebacon products include Daddy and the Muscle Academy, Medium and More, Round Carrots, I Found Something Good, and LKN.Multibrand store. By collaborating with the mentioned stores, Bringhome.thebacon is able to showcase their products alongside other complementary brands, providing customers with a wider range of options.

  - Seasonal events

Bringhome.thebacon also participates in seasonal events, such as craft and creative markets. For instance, the brand has been featured in the GOOD HOOD Market initiated by GOOD HOOD AND FRIENDS CO., LTD., which is known for showcasing innovative and unique products from up-and-coming brands. These events provide Bringhome.thebacon with an opportunity to connect with potential customers face-to-face, allowing them to showcase their products and receive direct feedback. By participating in these events, Bringhome.thebacon is able to expand its customer base and build brand recognition beyond their online presence.

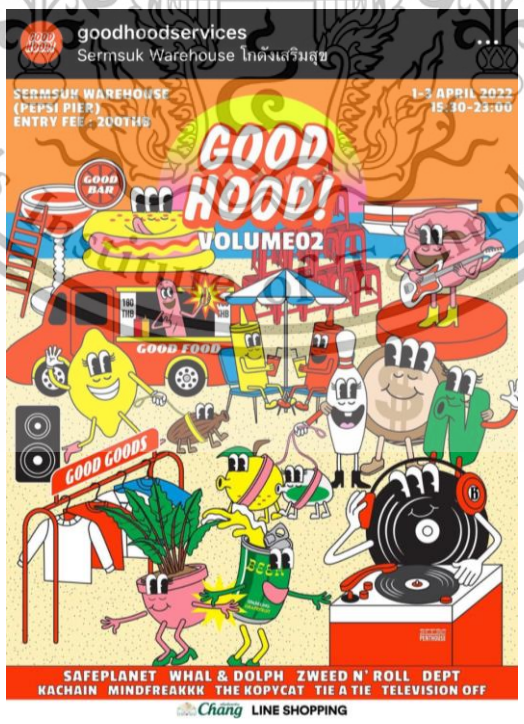

Figure 15 GOOD HOOD Services event

- Online storefront

As Bringhome.theBacon's primary selling approach is online, hence the main channel for their selling activities is through e-commerce platforms. To facilitate their e-commerce activities, Bringhome.theBacon relies on the platforms of Shopee and LINE Shopping.

- Shopee

Shopee is a leading e-commerce platform in Southeast Asia and Taiwan that allows merchants to create online storefronts and sell products directly to customers. Bringhome.theBacon has chosen Shopee as one of its main e-commerce platforms due to its wide customer reach and user-friendly interface. Shopee offers several advantages for Bringhome.theBacon, including access to a large and diverse customer base, secure payment and transaction processing, and convenient shipping and delivery options. Additionally, Shopee's features such as flash sales, in-app marketing, and promotional campaigns provide opportunities for Bringhome.theBacon to increase brand visibility and drive sales. With Shopee, Bringhome.theBacon is able to effectively showcase and sell its stationary and office supplies to a broad online audience.

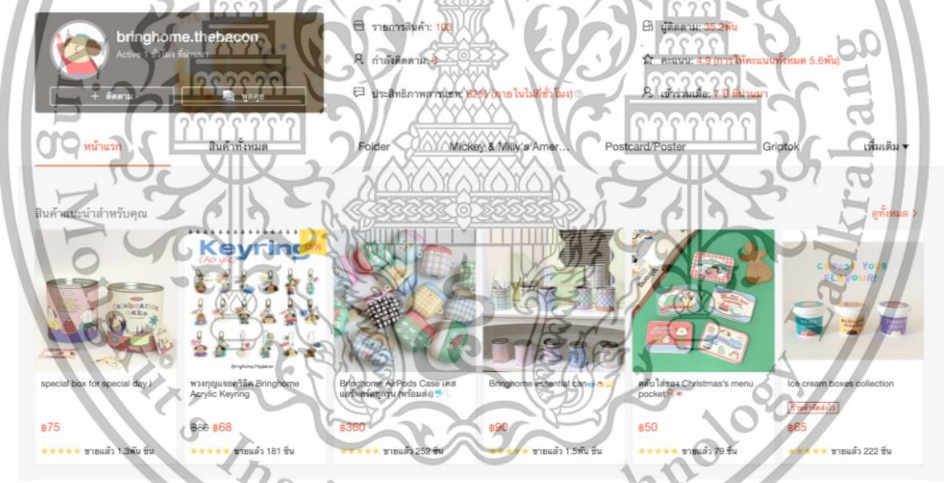

Figure 16 Bringhome.theBacon on Shopee

- LINE shopping

LINE Shopping is a popular e-commerce platform in Thailand that allows businesses to sell their products online. As one of the major e-commerce platforms used by Bringhome.theBacon, LINE Shopping provides various benefits such as a large user base (LINE users), convenient payment options, and a built-in chat feature through LINE Official Account that allows Bringhome.theBacon to communicate, promotes their business with customers easily and provide better customer service.

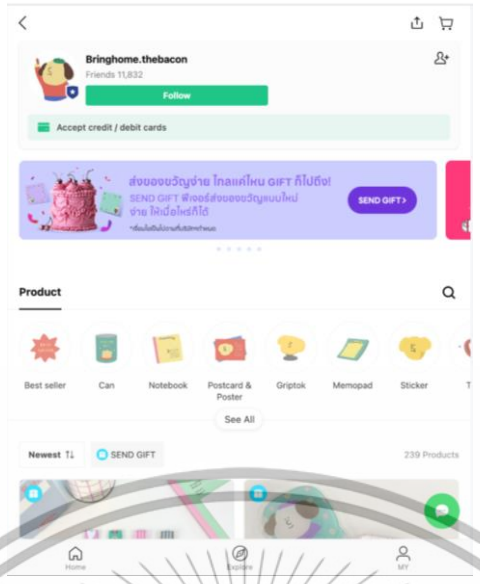

Figure 17 Bringhome.theBacon on LINE Shopping

2.2 Related technology

2.2.1 Web application architecture

Web applications have become an integral part of our daily lives. From social media platforms to e-commerce websites, web applications are the backbone of our online experiences. The architecture of a web application plays a crucial role in its performance, scalability, and user experience. The web application provides the various components of web application architecture, including the frontend, backend, database, and hosting.

2.2.1.1 Frontend

1) Html

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript). "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web. An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

2) CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now CSS without a version number.

3) Javascript

JavaScript is a popular scripting language used for web development, created by Brendan Eich in just 10 days in May 1995. Over the years, it has evolved to become a versatile language that is used in a wide range of applications, from web development to mobile app development and even server-side programming. One of the earliest works on JavaScript is "JavaScript: The Definitive Guide" by David Flanagan, first published in 1996. This book provides a comprehensive overview of the language and its features, including syntax, data types, functions, and objects. It has been updated several times to reflect changes to the language, with the most recent edition covering ECMAScript 6, the latest version of the JavaScript standard. In recent years, there has been a growing interest in using JavaScript for server-side programming, with the rise of Node.js. "Node.js in Action" by Mike Cantelon, Marc Harter, T.J. Holowaychuk, and Nathan Rajlich, published in 2014, provides a comprehensive guide to building scalable and high-performance applications with Node.js. It covers topics such as asynchronous programming, database integration, and testing. In conclusion, JavaScript is a versatile and widely-used language that has evolved significantly since its creation in 1995. There are many resources available for learning and mastering JavaScript, from books and online resources to structured learning programs. As the web continues to evolve, JavaScript will remain a crucial tool for building interactive and engaging web applications.

2.2.1.2 Backend

1) Nodejs

Node.js is a popular server-side JavaScript runtime, created by Ryan Dahl in 2009. It provides an event-driven, non-blocking I/O model that makes it well-suited for building scalable and high-performance applications. Over the years, Node.js has gained popularity among developers due to its ease of use, speed, and versatility. One of the earliest works on Node.js is "Node.js in Action" by Mike Cantelon, Marc Harter, T.J.

Holowaychuk, and Nathan Rajlich, published in 2014. This book provides a comprehensive guide to building scalable and high-performance applications with Node.js. It covers topics such as asynchronous programming, database integration, testing, and deployment. There are also many online resources available for learning Node.js, such as the official Node.js documentation, which provides a comprehensive guide to the language and its features, as well as interactive tutorials and examples. Additionally, there are numerous online courses, coding bootcamps, and MOOCs (Massive Open Online Courses) that provide structured learning paths for mastering Node.js. In conclusion, Node.js is a versatile and widely-used server-side JavaScript runtime that has gained popularity among developers due to its ease of use, speed, and versatility. There are many resources available for learning and mastering Node.js, from books and online resources to structured learning programs. As microservices architecture gains popularity, Node.js will continue to be a crucial tool for building scalable and high-performance applications.

1.1) Node.js Libraries

- Express.js

Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application. It's a layer built on the top of the Node js that helps manage servers and routes.

2.2.1.3 Database

1) Definition of Database

A database is an organized collection of data, stored and accessed electronically. Databases are used to store and manage large amounts of structured and unstructured data, and they can be used to support a wide range of activities, including data storage, data analysis, and data management. There are many different types of databases, including relational databases, object-oriented databases, and NoSQL databases, and they can be used in a variety of settings, including business, scientific, and government organizations.

2) Database Usage

Databases are used to store and manage large amounts of structured and unstructured data, and they can be used to support a wide range of activities, including data storage, data analysis, and data management. They are used in a variety of settings, including business, scientific, and government organizations. Some examples of how databases are used include storing customer information in a customer relationship management (CRM) system, storing financial transactions in an accounting system, storing inventory and orders in an e-commerce system, storing patient records in a

healthcare system, and storing student records in an educational institution. In each of these cases, the database is used to store and organize data in a structured manner, allowing multiple users to access and update the data simultaneously and ensuring the integrity and security of the data. The database also provides tools for data analysis and decision-making and allows for the creation of reports and other outputs based on the data.

3) Types of Databases

- Relational databases

A relational database organizes data into rows and columns, which collectively form a table. Data is typically structured across multiple tables, which can be joined together via a primary key or a foreign key. These unique identifiers demonstrate the different relationships which exist between tables, and these relationships are usually illustrated through different types of data models. Analysts use SQL queries to combine different data points and summarize business performance, allowing organizations to gain insights, optimize workflows, and identify new opportunities. Structured Query Language (SQL) is the standard programming language for interacting with relational database management systems, allowing database administrator to add, update, or delete rows of data easily. Originally known as SEQUEL, it was simplified to SQL due to a trademark issue. SQL queries also allow users to retrieve data from databases using only a few lines of code. Given this relationship, it's easy to see why relational databases are also referred to as "SQL databases" at times.

Benefits of relational databases

The primary benefit of the relational database approach is the ability to create meaningful information by joining the tables. Joining tables allows you to understand the relations between the data, or how the tables connect. SQL includes the ability to count, add, group, and also combine queries. SQL can perform basic math and subtotal functions and logical transformations. Analysts can order the results by date, name, or any column.

- Object-oriented databases

Object-oriented databases (also called object databases) are a type of database management system (DBMS). These databases store data in the form of objects, which are self-contained units of data and functionality. Object-oriented databases are designed to support the storage and management of complex, interrelated data.

- Non-Relational databases - NoSQL

NoSQL, also referred to as "not only SQL", "non-SQL", is an approach to database design that enables the storage and querying of data outside the traditional structures found in relational databases. While it can still store data found within relational database management systems (RDBMS), it just stores it differently compared to an RDBMS. The decision to use a relational database versus a non-relational database is largely contextual, and it varies depending on the use case. Instead of the typical tabular structure of a relational database, NoSQL databases house data within one data structure, such as JSON document. Since this non-relational database design does not require a schema, it offers rapid scalability to manage large and typically unstructured data sets. NoSQL is also type of distributed database, which means that information is copied and stored on various servers, which can be remote or local. This ensures availability and reliability of data. If some of the data goes offline, the rest of the database can continue to run.

NoSQL use cases

(1) Managing data relationships: Managing the complex aggregation of data and the relationships between these points is typically handled with a graph-based NoSQL database. This includes recommendation engines, knowledge graphs, fraud detection applications, and social networks, where connections are made between people using various data types.

(2) Low-latency performance: Gaming, home fitness applications, and ad technology all require high throughput for real-time data management. This infrastructure provides the greatest value to the consumer, whether that's market bidding updates or returning the most relevant ads. Web applications require in-memory NoSQL databases to provide rapid response time and manage spikes in usage without the lag that can comes with disk storage.

Benefits of Non-relational databases

Each type of NoSQL database has strengths that make it better for specific use cases. However, they all share the following advantages for developers and create the framework to provide better service customers, including:

(1) Cost-effectiveness: It is expensive to maintain high-end, commercial RDBMS. They require the purchase of licenses, trained database managers, and powerful hardware to scale vertically. NoSQL databases allow you to quickly scale horizontally, better allocating resources to minimize costs.

(2) Flexibility: Horizontal scaling and a flexible data model also mean NoSQL databases can address large volumes of rapidly changing data, making them great for agile development, quick iterations, and frequent code pushes.

(3) Replication: NoSQL replication functionality copies and stores data across multiple servers. This replication provides data reliability, ensuring access during down time and protecting against data loss if servers go offline.

(4) Speed: NoSQL enables faster, more agile storage and processing for all users, from developers to sales teams to customers. Speed also makes NoSQL databases generally a better fit for modern, complex web applications, e-commerce sites, or mobile applications.

4) Selected Database

Figure 18 MongoDB Logo

In this research, authors selected MongoDB, which MongoDB (link resides outside IBM) is an open source, non relational database management system (DBMS) that uses flexible documents instead of tables and rows to process and store various forms of data. As a NoSQL database solution, MongoDB does not require a relational database management system (RDBMS), so it provides an elastic data storage model that enables users to store and query multivariate data types with ease. This not only simplifies database management for developers but also creates a highly scalable environment for cross-platform applications and services.

4.1) Collections

MongoDB documents or collections of documents are the basic units of data. Formatted as Binary JSON (Java Script Object Notation), these documents can store various types of data and be distributed across multiple systems. Since MongoDB employs a dynamic schema design, users have unparalleled flexibility when creating data records, querying document collections through MongoDB aggregation and analyzing large amounts of information.

4.2) Schema

A schema is a JSON object that defines the structure and contents of your data. Schemas represent types of data rather than specific values. It supports many built-in schema types. These include primitives, like strings and numbers, as well as structural types, like objects and arrays, which it can combine to create schemas that represent custom object types.

5) MongoDB vs. MySQL

MySQL uses a structured query language to access stored data. In this format, schemas are used to create database structures, utilizing tables as a way to standardize data types so that values are searchable and can be queried properly. A mature solution, MySQL is useful for a variety of situations including website databases, applications and commercial product management. Because of its rigid nature, MySQL is preferable to MongoDB when data integrity and isolation are essential, such as when managing transactional data. But MongoDB's less-restrictive format and higher performance make it a better choice, particularly when availability and speed are primary concerns.

Benefits of MongoDB

(1) MongoDB stores data as JSON based documents that do not enforce the schema. It allows us to store hierarchical data in a document. This makes it easy to store and retrieve data in an efficient manner.

(2) It is easy to scale up or down as per the requirement since it is a document based database. MongoDB also allows us to split data across multiple servers.

(3) MongoDB provides rich features like indexing, aggregation, file store, etc.

(4) MongoDB performs fast with huge data.

(5) MongoDB provides drivers to store and fetch data from different applications developed in different technologies such as C#, Java, Python, Node.js, etc.

2.2.1.4 Hosting website (with DDNS)

1) Definition of DDNS

DDNS, most commonly known as Dynamic DNS, is an automatic method of refreshing a name server. It can dynamically update DNS records without the need for human interaction. It is extremely useful for updating IPv4 and IPv6 records when the host has changed its IP address. Dynamic DNS is a method of updating and changing the DNS automatically in real time. Many web properties, such as APIs or websites, run on internet connections that have their IP addresses changed frequently; this creates a problem if the operators of those properties want to give a hosted resource a specific domain name, which must then store an IP address in Domain Name System (DNS) records. Dynamic DNS (DDNS) is a service that keeps the DNS updated with a web property's correct IP address, even if that IP address is constantly being updated. For example, if a web administrator is operating a small website with a domain name of www.example.com and an IP address of 192.0.2.0, anytime another user enters www.example.com into their browser, the DNS will direct them to the server at 192.0.2.0. If the admin's ISP dynamically changes the IP to 192.0.2.1, a dynamic DNS service can automatically update the admin's DNS records so that other users trying to visit www.example.com will now go to the correct IP address.

2) DDNS Architecture

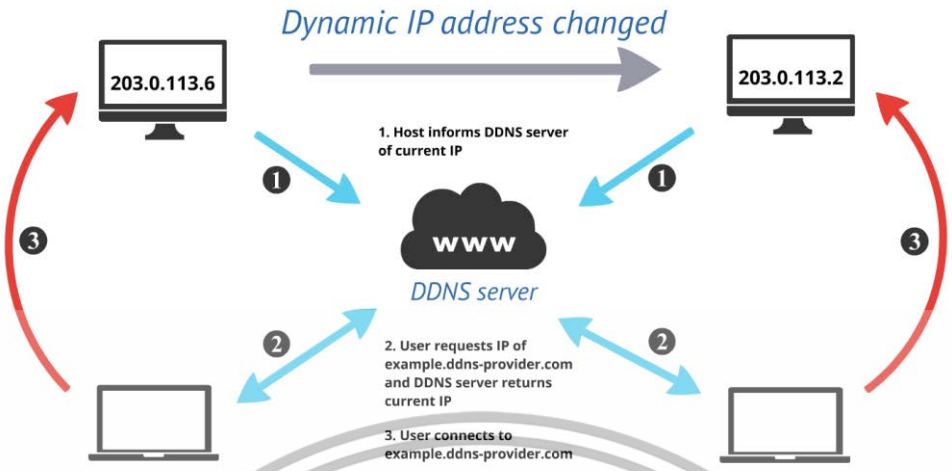

Figure 19 DDNS Architecture

There are a number of companies who offer dynamic DNS services with varying features and technologies. One very common method of enabling dynamic DNS is by providing users with software which runs on their computer or router. This software communicates with the dynamic DNS service provider anytime the IP addresses provided by the ISP are updated, and the dynamic DNS provider in turn updates the DNS with those changes, providing almost instant updates. In this section, author will describe the companies offer DNS services for this research which is True Corporation

3) TRUE DDNS

True Corporation offers a DDNS service that allows users to access their network devices or services using a domain name that stays the same even if their IP address changes. This service is commonly used for remote access to home networks, hosting websites or servers at home, and accessing networked cameras or other devices. The True DDNS service works by associating a domain name with a user's current IP address, which can change periodically as a result of changes to the user's internet connection. Users can sign up for the service and configure their router or device to send updates to the True DDNS service with their current IP address. The True DDNS service then maps the user's domain name to their current IP address, allowing users to access their devices or services using a human-readable domain name rather than an IP address. Overall, True DDNS is a useful service for users who have a dynamic IP address that changes frequently and need to maintain a consistent domain name for accessing their network devices or services. However, it's important to note that DDNS services may not always be reliable, and changes to DNS records can take time to propagate, so it may not be suitable for high-availability or mission-critical applications.

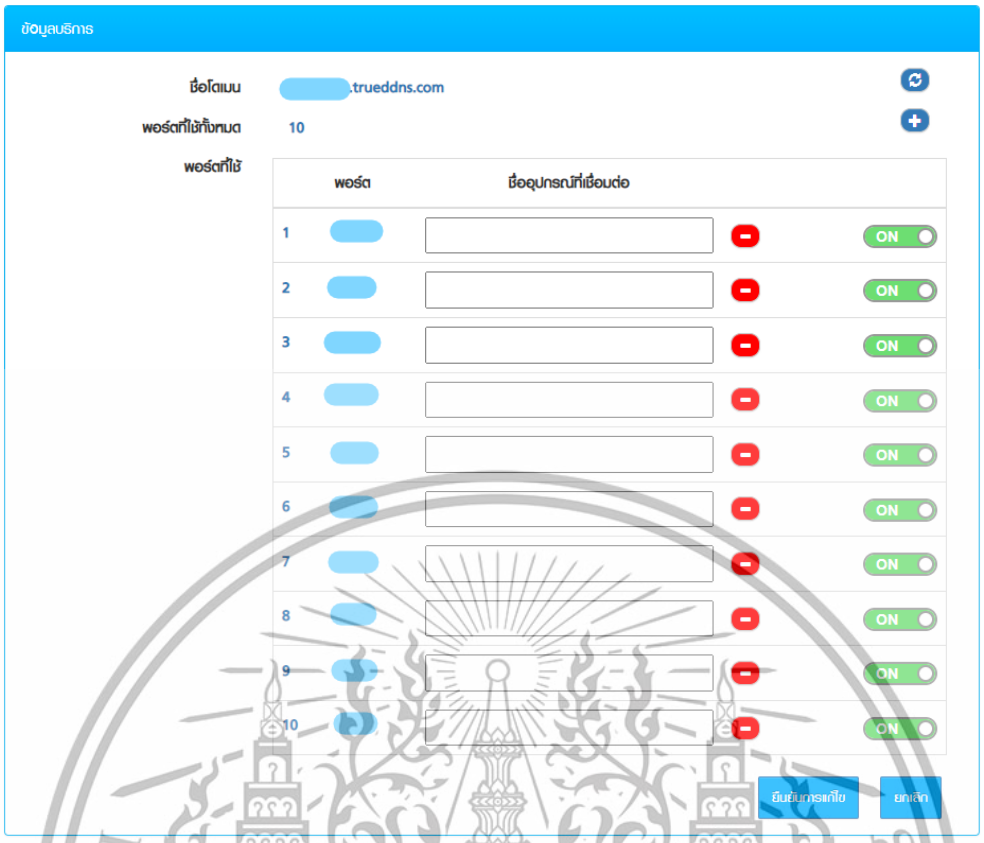

Figure 20 True DDNS

3.1) Domain name

The domain name used by True DDNS is "trueddns.com". This is the domain name that users can use to access their network devices or services using a human-readable domain name rather than an IP address. Users can sign up for a True DDNS account and create a subdomain under the trueddns.com domain name, which will be associated with their current IP address. When the user's IP address changes, the subdomain will be updated to point to the new IP address, allowing users to continue to access their network devices or services using the same domain name.

3.2) Port

It can use up to 10 ports with True DDNS. The number of ports you need to use depends on the specific device or service you want to access. If you have multiple devices or services that use different ports, you can use multiple ports with True DDNS. However, to access a device or service through a specific port, you need to forward the appropriate port from the router or device to the port of the device or service you want to access. It's important to note that opening too many ports can increase the security risk to your system, so it is recommended to only forward necessary ports and choose ports that are not already in use by other devices or services on your network to increase security.

3.3) IPv4 Port Mapping

Port mapping is to map a port of the IP address of an extranet host to a machine on the Intranet to provide corresponding services. When a user accesses this port of the IP, the server automatically maps the request to a machine within the corresponding LAN. Port mapping can be dynamic or static. Port mapping is a common operation during router configuration. The purpose is to change ports for different services to achieve more flexible applications.

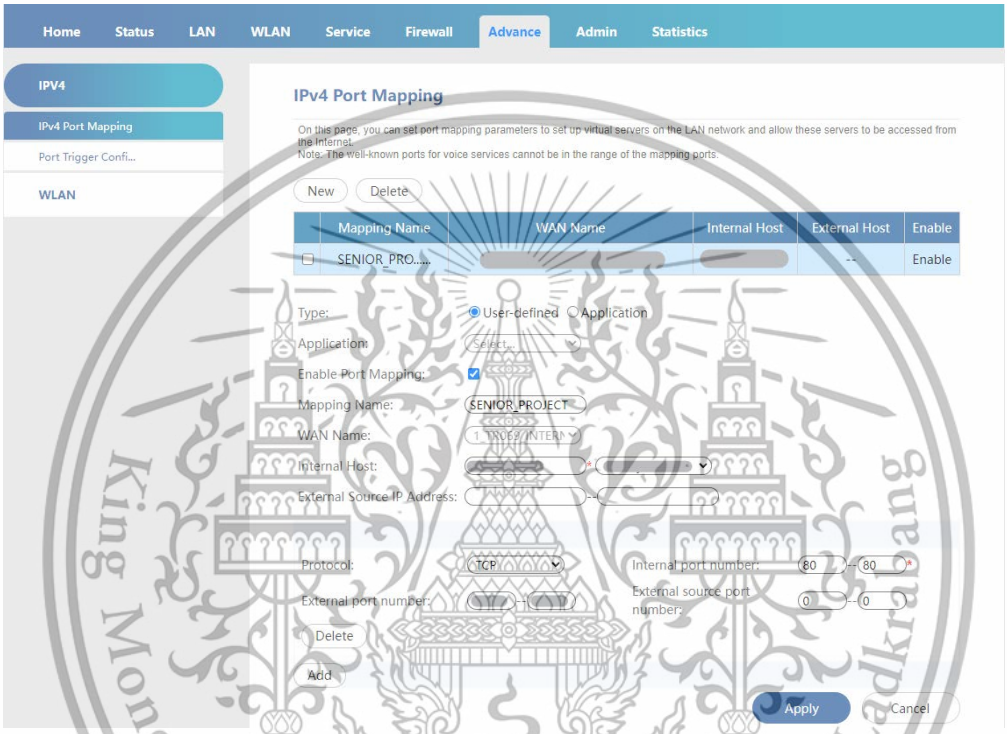

Figure 21 IPv4 Mapping

3.4) Internal Host

Internal Host DDNS refers to the use of a dynamic DNS service within your local network to assign a domain name to your internal devices or servers. This allows you to easily access your internal network resources from outside your network without having to remember the IP addresses of each device. With an Internal Host DDNS, you can create a domain name that points to the internal IP address of your device, and then access that device by typing in the domain name rather than the IP address.

3.5) Internal port number

A number assigned to a specific application or service running on a device within your internal network. These port numbers allow incoming data to be routed to the correct application or service running on the device. For example, HTTP traffic typically uses port 80, while HTTPS traffic typically uses port 443. When a user accesses a website

hosted on a server within your internal network, the router will use the port number associated with the HTTP or HTTPS service running on the server to direct incoming traffic to the correct application or service. Configuring port numbers correctly is important to ensure that incoming traffic is properly directed to the appropriate application or service, and to avoid conflicts between different services or applications running on the same device.

3.6) External port number

The external port number used by True DDNS will vary based on the port number that has been set up in your router's settings to forward incoming traffic to the specific device or service within your internal network. To set up the external port number, you need to create port forwarding rules in your router's settings that direct incoming traffic to the specific device or service on your network. This helps ensure that incoming traffic is correctly routed to the intended device or service on your internal network.

2.2.2 Public Key Infrastructure (PKI)

2.2.2.1 Definition of Public Key Infrastructure (PKI)

A Public Key Infrastructure (PKI) is a combination of policies, procedures and technology needed to manage digital certificates in a public key cryptography scheme. A digital certificate is an electronic data structure that binds an entity, being an institution, a person, a computer program, a web address etc., to its public key. Digital certificates are used for secure communication, using public key cryptography, and digital signatures. The purpose of a PKI is to make sure that the certificate can be trusted. Public key cryptography is an application of asymmetric cryptography. In asymmetric cryptography, two different but mathematically related keys are used to accomplish encryption and decryption of data. Data encrypted with one key can only be decrypted with the other key, and vice versa. Additionally, it is not possible to deduce one key knowing the other. In public key cryptography, the "public key" is meant for public distribution while the "private key" is to be only accessible to the key pair owner. A public-private key pair has two very useful properties:

1) Public Key

The public key is used to encrypt data in a way that only the key pair owner can decrypt the data, using the private key. This is useful for secure communication.

2) Private Key

Only the key pair owner can encrypt data with the private key, ensuring all recipients of the authenticity of the sender, for only the associated public key will decrypt the data. This is used as a digital form of signature.

2.2.2.2 Digital Signature Algorithm

Digital signatures can be used to distribute a message in plaintext form when the recipients must identify and verify the message sender. Signing a message does not alter the message; it simply generates a digital signature string you can either bundle with the message or transmit separately. A digital signature is a short piece of data that is encrypted with the sender's private key. Decrypting the signature data using the sender's public key proves that the data was encrypted by the sender or by someone who had access to the sender's private key. Digital signatures are generated by using public key signature algorithms. A private key generates the signature, and the corresponding public key must be used to validate the signature. This process is shown in the following illustration. There are two steps involved in creating a digital signature from a message. The first step involves creating a hash value (also known as a message digest) from the message. This hash value is then signed, using the signer's private key. The following is an illustration of the steps involved in creating a digital signature. To verify a signature, both the message and the signature are required. First, a hash value must be created from the message in the same way the signature was created. This hash value is then verified against the signature by using the public key of the signer. If the hash value and the signature match, you can be confident that the message is indeed the one the singer originally signed and that it has not been tampered with. The following diagram illustrates the process involved in verifying a digital signature.

A hash value consists of a small amount of binary data, typically around 160 bits. This is produced by using a hashing algorithm. A number of these algorithms are listed later in this section.

All hash values share the following properties, regardless of the algorithm used:

(1) The length of the hash value is determined by the type of algorithm used, and its length does not vary with the size of the message. The most common hash value lengths are either 128 or 160 bits.

(2) Every pair of non-identical messages translates into a completely different hash value, even if the two messages differ only by a single bit. Using today's technology, it is not feasible to discover a pair of messages that translate to the same hash value without breaking the hashing algorithm.

(3) Each time a particular message is hashed using the same algorithm, the same hash value is produced.

(4) All hashing algorithms are one-way. Given a hash value, it is not possible to recover the original message. In fact, none of the properties of the original message can be determined given the hash value alone.

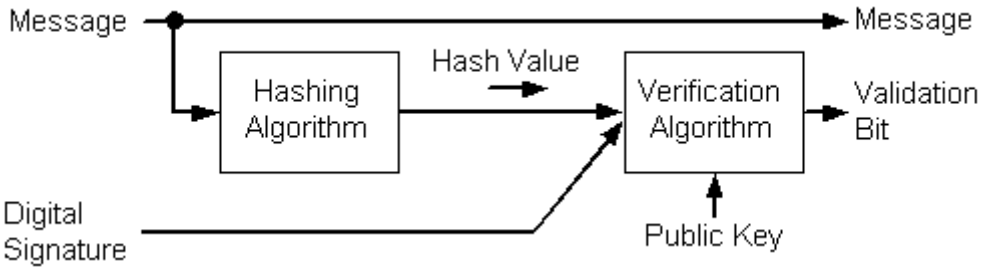

Figure 22 Digital Signature Algorithm

1) PKCS7: Cryptographic Message Syntax Standard

The Cryptographic Message Syntax (CMS) is the IETF's standard for cryptographically protected messages. It can be used to digitally sign, digest, authenticate or encrypt any form of digital data. CMS is based on the syntax of PKCS #7, which in turn is based on the Privacy-Enhanced Mail standard. The newest version of CMS (as of 2009) is specified in RFC 5652 (but see also RFC 5911 for updated ASN.1 modules conforming to ASN.1 2002). The architecture of CMS is built around certificate-based key management, such as the profile defined by the PKIX working group. CMS is used as the key cryptographic component of many other cryptographic standards, such as S/MIME, PKCS #12 and the RFC 3161 Digital timestamping protocol. OpenSSL is open source software that can encrypt, decrypt, sign and verify, compress and uncompress CMS documents.

2) Relationship between PKCS7 Standard and Digital Signatures

PKCS #7 is one the most famous and extensively used standard from the series of PKCS (Public Key Cryptography Standards) by RSA Security LLC. PKCS #7 is the specific standard used for generation and verification of digital signatures and certificates managed by a PKI (Public Key Infrastructure). This standard served as the basis for the S/MIME (Secure/Multipurpose Internet Mail Extensions) standard. PKCS #7 proposes a broad-spectrum syntax and format for creation of digital signatures which is elaborated in detail in RFC 2315. It also allows compatibility with Privacy-Enhanced Mail (PEM) format which is the default and most commonly used file format for storage and sharing of crypto keys and digital certificates. PKCS #7 allows PEM compatible signed-data to be converted into PEM messages.

2.2.3 Other Technology Tools

2.2.3.1 Code editor

1) VS code

Visual Studio Code is a free, lightweight but powerful source code editor that runs on your desktop and on the web and is available for Windows, macOS, Linux, and Raspberry Pi OS. It comes with built-in support for JavaScript, TypeScript, and Node.js

and has a rich ecosystem of extensions for other programming languages (such as C++, C#, Java, Python, PHP, and Go), runtimes (such as .NET and Unity), environments (such as Docker and Kubernetes), and clouds (such as Amazon Web Services, Microsoft Azure, and Google Cloud Platform). The code in the Visual Studio Code repository is open source under the MIT License. The Visual Studio Code product itself ships under a standard Microsoft product license, as it has a small percentage of Microsoft-specific customizations. It's free despite the commercial license.

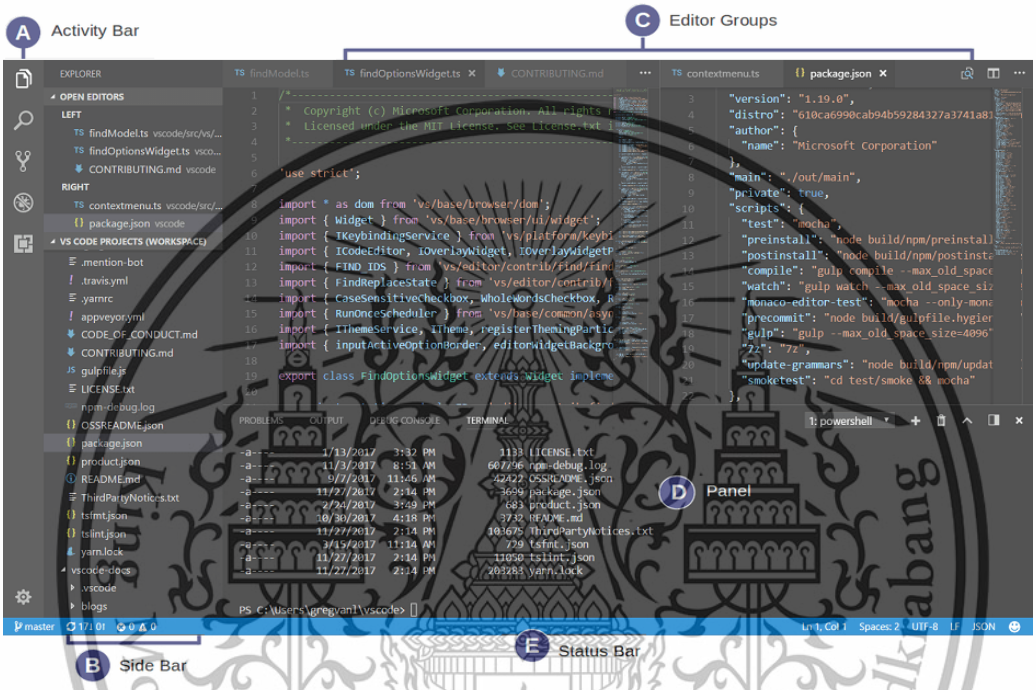

Figure 23 VS Code User Interface

VS Code comes with a simple and intuitive layout that maximizes the space provided for the editor while leaving ample room to browse and access the full context of your folder or project. The UI is divided into five main areas:

(1) Editor: The main area to edit your files. You can open as many editors as you like side by side vertically and horizontally.

(2) Primary Sidebar: Contains different views like the Explorer to assist you while working on your project.

(3) Status Bar: Information about the opened project and the files you edit.

(4) Activity Bar: Located on the far left-hand side, this lets you switch between views and gives you additional context-specific indicators, like the number of outgoing changes when Git is enabled.

(5) Panel: An additional space for views below the editor region. By default, it houses output, debug information, errors and warnings, and an integrated terminal. Panel can also be moved to the left or right for more vertical space.

2) GitHub

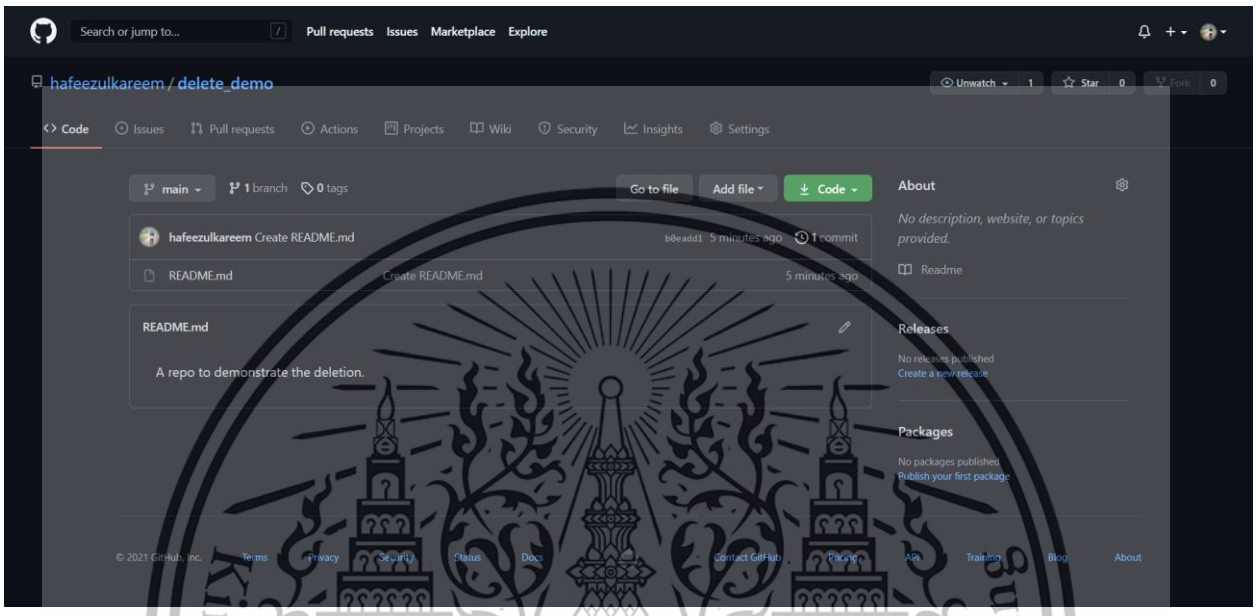

Figure 24 GitHub User Interface

GitHub is a web-based version control and collaboration platform for software developers. Microsoft, the biggest single contributor to GitHub, acquired the platform for $7.5 billion in 2018. GitHub, which is delivered through a software as a service (SaaS) business model, was started in 2008. It was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster. Git is used to store the source code for a project and track the complete history of all changes to that code. It lets developers collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers.

2.1) How does GitHub work

GitHub facilitates social coding by providing a hosting service and web interface for the Git code repository, as well as management tools for collaboration. The developer platform can be thought of as a social networking site for software developers. Members can follow each other, rate each other's work, receive updates for specific open source projects, and communicate publicly or privately.

2.2) The important terms GitHub developers use

2.2.1) Pull request: If a developer would like to share their modifications, they can send a pull request to the owner of the original repository.

2.2.2) Merge**:** If, after reviewing the modifications, the original owner would like to pull the modifications into the repository, they can accept the modifications and merge them with the original repository.

2.2.3) Push**:** This is the reverse of a pull -- a programmer sends code from a local copy to the online repository.

2.2.4) Commit**:** A commit, or code revision, is an individual change to a file or set of files. By default, commits are retained and interleaved onto the main project, or they can be combined into a simpler merge via commit squashing. A unique ID is created when each commit is saved that lets collaborators keep a record of their work. A commit can be thought of as a snapshot of a repository.

2.2.5) Clone**:** A clone is a local copy of a repository.

3) Figma

Figma is a tool primarily targeted toward people who want to design interfaces. The software was originally launched in September 2016. At the time of writing in January 2023, over four million people use Figma. Figma is a handy tool for brainstorming. Users can use text boxes to write anything they might need to refer to later, along with using different shapes and adding comments. In Figma, individuals can choose from numerous dimension sizes to accurately depict what their page would look like. The tool is incredibly helpful if one wants to create app prototypes and see how they will feature on multiple devices. When creating an app prototype with Figma, users can add text and change its fonts and sizes. They're also able to integrate several plugins, along with numerous other things. Figma is one of the best interface design tools on the market, and it has plenty of uses for designers. But even if someone is in another creative field, they can use Figma in several other ways. It's handy for the early planning stages, and it can be used alongside numerous apps to make collaborating with others easier. People can use Figma to design website pages and apps, along with landing pages and more. And if someone doesn't want to start from scratch, they'll find plenty of free templates.

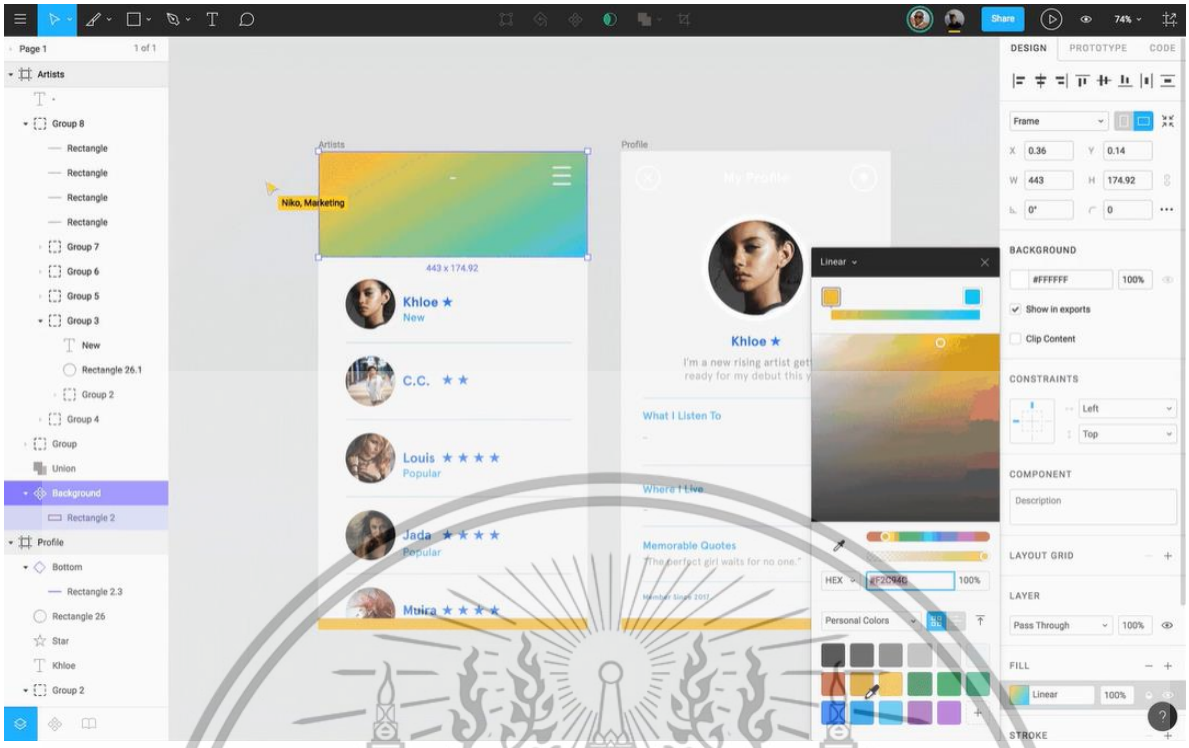

Figure 25 Figma User Interface

Chapter 3
Methodology

3.1 Project Timeline

Smart E-Tax Invoice Generator project is committed to a four-month timeline, consisting of four main phases: Requirements Gathering, Design, Development, and Testing, as illustrated in the following figures

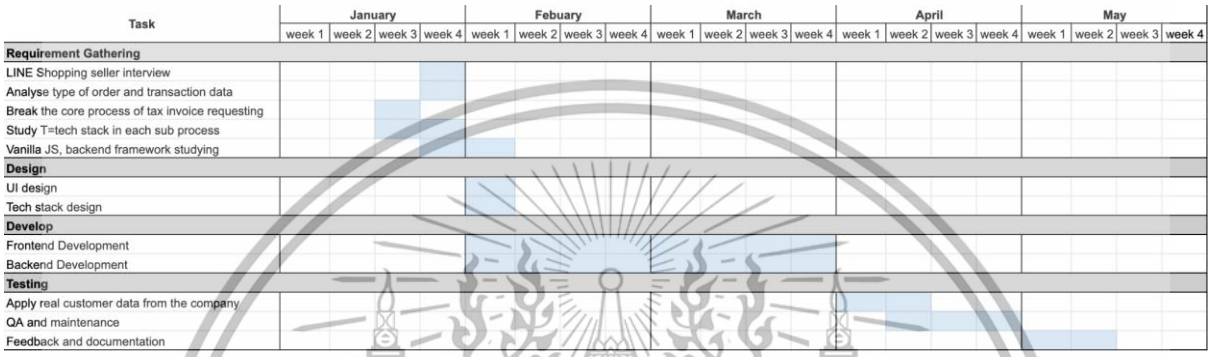

Figure 26 Smart E-Tax Invoice Generator Development Timeline

3.2 Methodology

3.2.1 Research

Research is an important part of the product development process, in which a Smart e-Tax Invoice Generator from this research. A good research helps authors guide and validate the demand or need for a product before any design and development work begins. It's crucial to assess whether there is a need to solve a problem before jumping into designing a solution. As researching is one of the product development processes, this section will be focused on how authors set up research plans which are separated into main research purposes: user research with company and tax invoice process.

3.2.1.1 User research with company

1) Research Plan

User research with a company starts with writing a user research plan guide for the researcher to ensure that the objectives of the research will be answered at the end of the project. For the research methodology list below is the outline included in the user research plan.

1.1) Research Objective

The authors emphasize the importance of comprehensively understanding the company's sales processes and associated conflicts, with a particular focus on tax

invoice generation, due to the increasing number of transactions being observed. This research aims to be as follows.

1.1.1) Gain insights into the behavior and interests of customers when it comes to making online purchases with the company.

1.1.2) Identify and analyze the scope of the sale processes and conflicts to prevent potential challenges in the company's tax and accounting processes in the future.

1.2) Research Question

1.2.1) Can you explain the process of how an order is placed on your online platform, from start to finish?

1.2.2) Can we observe MyShop's user interface? What are the necessary features related to order transaction data?

1.3) Methodology

The research methodology conducted in this research is an in-depth interview with the selected participants.

1.4) Company Participants

The company participating in this research is Bringhome.theBacon, an online business specializing in the sale of stationary and office supplies, as discussed in Chapter 2.

In conclusion, a user research plan is an essential component for conducting successful research with a company. The plan should include a clear and concise research objective, a well-defined methodology, a participant recruitment strategy, an interview script, and a timeline.These elements enable researchers to achieve their research objectives and acquire valuable insights from the user research procedure.

2) Research Summary

2.1) Executive Summary

During the interview the company representative aimed to prepare for the future operation of the company by gaining insights into customers' behavior and interests when making online purchases. The representative sought to understand customers' preferences, such as their preferred payment methods, the frequency of purchases, and the factors that influence their buying decisions. This information helps the company tailor its future operations to meet the needs and interests of its customers since their branding is uniquely designed to represent the core identity. The representative mentioned that customers are female customers between the ages of 13 and 23 who

tend to purchase stationery and desk decorative products, such as notebooks and stickers. These customers are characterized by a 'Studygram' lifestyle, which showcases their productive lifestyle and aesthetic study products via social media, as illustrated from the figure below. As a result, the company is seeking a customer-centric design solution that uses their design system and characters.

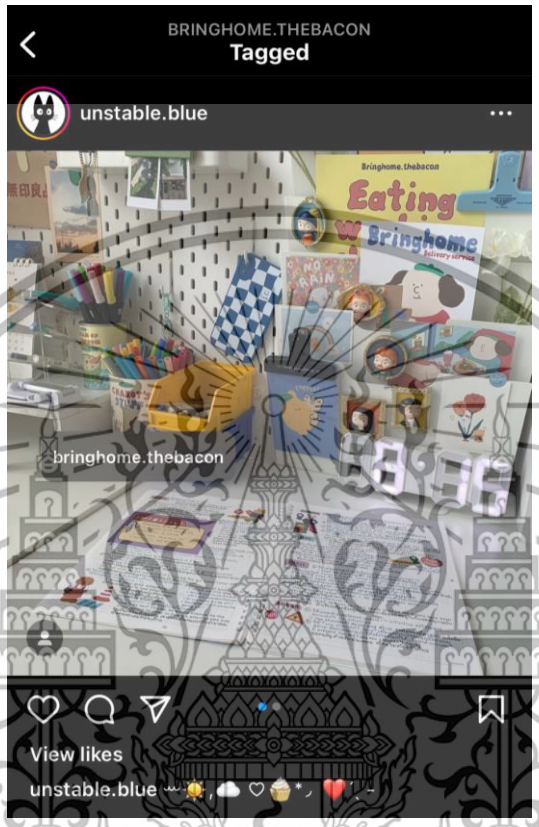

Figure 27 Influencer's review on Instagram

Additionally, the representative aimed to identify and analyze the scope of the company's sales processes to prevent potential challenges in the future. This included examining potential conflicts that could arise and cause issues with the company's tax and accounting processes due to the incoming transaction during the day. For example, the company representative mentioned that during a special campaign (2.2 Campaign) on one of their e-commerce platforms, a total of 154 orders were accommodated on February 2nd, 2023. The representative expressed concern that manually generating tax invoices for numerous order transactions is susceptible to human error, which can negatively impact customer experience and tax processes once their company is registered. Moreover, issuing property tax invoices for every sale to ensure accurate and proper accounting and tax reporting. The representative sought to gain a clear understanding of the company's sales processes to tax processes in the future.

2.2) Key Findings

2.2.1) Tailor-made solution using Bringhome.theBacon design system

Based on the company perspective, Bringhome.theBacon is mainly focused on their brand identity as it is reflected in all aspects of its products and branding, including social media channels, website, and other solutions. It is essential that all these components adhere to the company's design system, guidelines, and characteristics. Therefore, Bringhome.theBacon requires a custom-made solution that is specifically tailored to its brand identity and can effectively incorporate its design system, guidelines, and characteristics.

2.2.2) Cover all order transaction and issuing property tax invoices, ensuring the accuracy of the result tax invoice

The process of correctly retrieving and displaying customer information is crucial to ensure that the generated e-Tax Invoice matches the right order transaction. This requires a robust and reliable system that can efficiently retrieve customer information and order transaction details from a database. Additionally, security is paramount when it comes to e-Tax Invoices. The process of encrypting the tax invoice helps to ensure that the invoice is secure and tamper-proof, which is essential in preventing any unauthorized access or modifications which will be declared in the further context.

3.3.1.2 Tax Invoice Research

1) Process

The process of how a customer requests a tax invoice from a seller involves several steps to ensure accurate and compliant documentation. It begins when the customer contacts the seller and expresses their need for a tax invoice. Upon receiving the customer's request, the seller carefully collects the necessary information to generate the tax invoice. The customer is asked to provide key details such as their full name, complete address, and tax identification number. Once the customer's information is obtained, the seller proceeds to verify the request. This involves reviewing the provided information to ensure its accuracy and completeness. After verifying the request, the seller moves on to the tax invoice generation stage. Using the customer's information and the relevant details of the purchase or transaction, the seller creates a comprehensive tax invoice. This document typically includes the customer's name and address, details of the products or services purchased, any applicable taxes or fees, and the total amount payable. Once the tax invoice is generated, the seller prepares it for delivery to the customer. The mode of delivery may depend on the agreed-upon method between the parties involved. It could be a physical document, where the seller prints and hands over the tax invoice to the customer. Alternatively, the tax invoice may be

sent electronically in the form of a PDF file, which can be conveniently shared via email or any other digital communication platform. By following this detailed process, the customer can successfully request a tax invoice from the seller. This ensures compliance with tax regulations and provides the customer with accurate and reliable documentation for their purchase.

2) Required Information

To ensure compliance with tax regulations, there are specific pieces of information that must be included in the tax invoice. This list of required information was cited from the literature from chapter 2 and summarized declared below.

2.1) The words "Tax Invoice" must be prominently displayed. The phrase "Tax Invoice" is required by law to be included in the document to indicate that it is a tax invoice.

2.2) The date of tax invoice has issued

2.3) The list of terms "name, address, and taxpayer identification number of the operator registered for value-added tax and the registration number for tax invoice issuance."

2.3.1) The name of the tax invoice issuer refers to the name of the operator registered for value-added tax. In this case the tax invoice issuer or the recipient of the tax invoice is a juristic person with incoming Company Limited - Co., Ltd. status.

2.3.2) The address of the tax invoice issuer refers to the location of the business establishment registered for value-added tax

2.3.3) The taxpayer identification number of tax invoice issuer’s

2.4) The list of terms "name, address, and taxpayer identification number of the operator registered for value-added tax and the registration number for the purchaser or recipient who is a registered VAT operator."

2.4.1) The name of the purchaser or recipient who is a registered VAT operator

2.4.2) The address of the the purchaser or recipient who is a registered VAT operator

2.4.3) The purchaser's or recipient's taxpayer identification number who is a registered VAT operator

2.4.4) Other information, such as mobile number

2.5) The list "Name, Type, Category, Quantity, and Value of Goods or Services." This can be done by arranging a symbol or a separate list item clearly indicating that it is a tax-exempt goods or services.

2.6) The list "Amount of Value-Added Tax Calculated from the Value of Goods or Services, Clearly Separated from the Value of Goods or Services" should be clearly separated and indicated in the list item.

2.7) Other information required by the Director General of the Revenue Department

In conclusion, to comply with tax regulations, it is important to include specific pieces of information in the tax invoice. These include the phrase "Tax Invoice," the date of issuance, the name, address, and taxpayer identification number of the registered VAT operator, the registration number for tax invoice issuance, and the name, address, and taxpayer identification number of the registered VAT operator for the purchaser or recipient. Additionally, the list should include the name, type, category, quantity, and value of goods or services, as well as the amount of value-added tax calculated from the value of goods or services. Other information required by the Director General of the Revenue Department may also need to be included.

3) Additional restriction in issuing tax invoice

In addition to the information provided above for the requirements for issuing tax invoices, it is necessary to create credibility for the document by certifying its correctness. This can be done by signing the document by hand or by using electronic signatures, which can be digital signatures or digital handwritten signatures. In this case, if a reliable digital signature is used that meets the requirements specified in Section 26 of the Electronic Transactions Act, which is a digital signature used when there is a need to create clear evidence of the intention of the signature owner (e.g. approval, acceptance, notification, acknowledgment, witness), it will help reduce concerns about the problem of later denial, such as denying knowledge of the terms of use or denying agreement. However, in the case of issuing a full-form tax invoice under Section 86/4 of the Revenue Code, there are no requirements for the authorized person to sign and stamp the original tax invoice. Therefore, digital signatures can also be used, or the signature does not have to be signed in the original tax invoice. However, to build confidence for the applicant for the tax invoice and to prepare for the submission of tax invoices into the revenue department's e-tax invoice system for the company after registration and entering the e-tax invoice system, a digital handwritten signature in XML Advanced Electronic Signatures (XAdES) 1 format will be created individual tax invoice.

3.3.1.3 Solution initiation

The result of user research with the company and tax invoice process lead to workflow and features of Smart e-Tax Invoice Generator as stated below.

1) Customer filling form: Developing a medium for receiving customer information with request form user interface, to gather customer's information including name, address and tax identification number.
2) e-Tax Invoice generation: generating tax invoice customer's information and order transaction details from the database as PDF file.
3) Encryption: encrypting tax invoice with XAdES digital signature.

The workflow and features include a customer request form interface to gather customer information, e-Tax invoice generation from the database as a PDF file, and encryption of the invoice with XAdES digital signature.

3.2.2. Workflow design

This workflow diagram shows the structure of a smart e-tax invoice generator project which includes client, frontend backend and hosting.

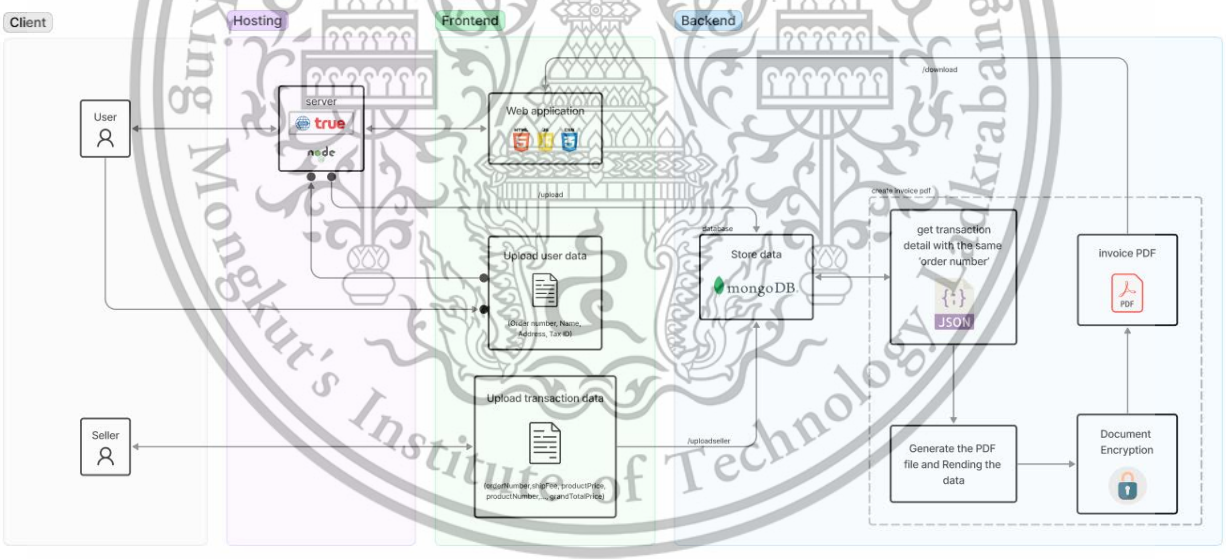

Figure 28 Workflow diagram of Web Application

3.3.2.1 Choosing techstack

1) SQL vs NoSQL

1.1) SQL

SQL stands for "Structured Query Language,"is widely used for querying relational databases, where data is stored in rows and tables that are linked in various ways. One table record may link to one other or to many others, or many table records may be related to many records in another table. These relational databases, which offer

fast data storage and recovery, can handle great amounts of data and complex SQL queries. SQL databases are valuable in handling structured data, or data that has relationships between its variables and entities. SQL database schema organizes data in relational, tabular ways, using tables with columns or attributes and rows of records. Because SQL works with such a strictly predefined schema, it requires organizing and structuring data before starting with the SQL database.

1.2) NoSQL

NoSQL is a non-relational database, meaning it allows different structures than a SQL database (not rows and columns) and more flexibility to use a format that best fits the data. NoSQL systems work with different data structures within a database. Because they allow a dynamic schema for unstructured data, there's less need to pre-plan and pre-organize data, and it's easier to make modifications. NoSQL databases allow you to add new attributes and fields, as well as use varied syntax across databases. NoSQL databases are not relational, so they don't solely store data in rows and tables. Instead, they generally fall into one of four types of structures:

(1) Column-oriented, where data is stored in cells grouped in a virtually unlimited number of columns rather than rows.

(2) Key-value stores, which use an associative array (also known as a dictionary or map) as their data model. This model represents data as a collection of key-value pairs.

(3) Document stores, which use documents to hold and encode data in standard formats, including XML, YAML, JSON (JavaScript Object Notation) and BSON. A benefit is that documents within a single database can have different data types.

(4) Graph databases, which represent data on a graph that shows how different sets of data relate to each other. Neo4j, RedisGraph (a graph module built into Redis) and OrientDB are examples of graph databases.

SQL	NoSQL
unstructured or semi-structured data	structure data
run on a single server	run on clusters of servers
many experts available to support SQL and programming relational data	NoSQL users do benefit from open-source system
applications that require complex queries and transactions	ideal for applications that require high scalability
strict rules for data consistency	data can be entered in a variety of formats and structures.

Table 2 the Different between SQL and NoSQL

2) Why NoSQL is the database choice for developing this research?

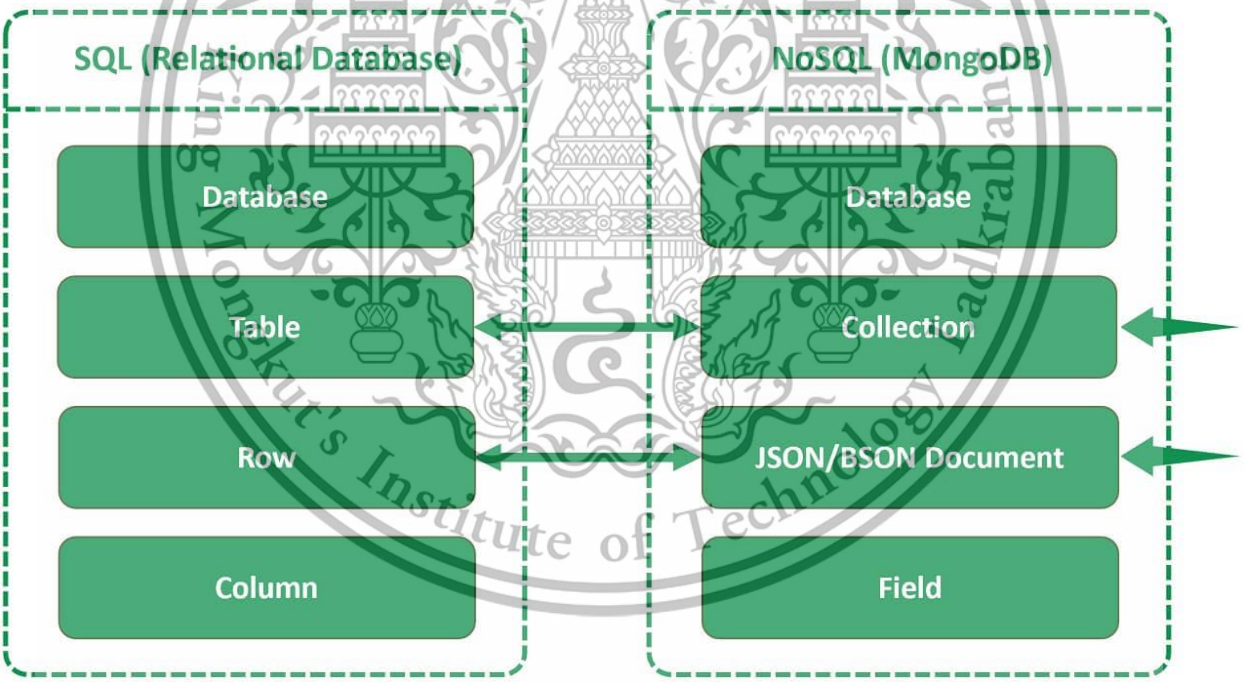

Figure 29 SQL vs NoSQL

Since the received data is separated into two sides, the store's side and the customer's side, with the store's data being divided into different order numbers, and the customer's side having data stored with different order numbers as well, these two sets of data will need to be used together by grouping them based on the same order number. This is necessary to create a PDF file from this combined data.

After evaluating the data obtained from the store and customer data, it was determined that a NoSQL database would be the better choice. This is because the data obtained from the store is in the form of an Excel file, which is a document format. Meanwhile, the data obtained from customers is stored in the form of a JSON file, which has a large number of customer data components combined together. NoSQL databases offer greater flexibility than traditional SQL databases because they do not require a fixed schema. This means that you can store data in any format you choose, without having to worry about predefined data types and structures. In addition to flexibility, NoSQL databases can be faster than SQL databases in certain situations because they are optimized for specific use cases. They also use distributed architectures and in-memory caching to provide faster query response times. Based on the data obtained from the store and customer data, a NoSQL database is the recommended solution. Its flexibility and performance advantages make it a better choice than a traditional SQL database.

Why choose Node.js as a Backend

Node.js is a universal language that uses JavaScript to build web applications for both the front-end and back-end. It's the optimal choice for developing server-side applications or backends due to its large size, adaptability to a wide range of situations and contexts, and the numerous flexibility and functionality features it offers. Node.js contains a library of packages and modules that are essential for the implementation process. Moreover, Node.js can install particular packages during development. Although it already has many packages available for use, the library is continually expanding. As a result, Node.js offers numerous necessary packages to apply to any project.

The example of package that benefit to the smart E-tax invoice generator project

(1) Express

Express is a NodeJS web application framework. It is used widely to create and publish applications. It is fast, unopinionated, and comes with a robust collection of HTTP helpers. It is also best known for its development of APIs.

(2) MongoDB

It’s an official MongoDB driver for NodeJS, that provides API for MongoDB databases. Same as MySQL is a relational database that uses structured query language for accessing MongoDB. It ensures data integrity and meets the standards for security and compliance issues.

(3) Request

The request is an HTTP library that is used to make HTTP calls. GET, PUSH, PUT, POST are the four requests made by HTTP during server-side implementation. It deals with the authentication part of the application.

In conclusion, Node.js offers numerous advantages as a backend technology for web development. Its ability to handle concurrent connections efficiently, high performance, scalability, and widely packaged ecosystem make it a compelling choice for building modern web applications. With its active community and growing support, Node.js continues to evolve and remains an excellent option for developers looking for a flexible backend solution.

3) TRUE DDNS

TRUE DDNS provide an efficient solution for individuals facing the challenge of changing their IP address frequently. By automatically updating domain name records, TRUE DDNS ensures service continuity, accessibility, and enhanced security. It's also a cost-saving service because its service is generally available at a lower cost or even free of charge in some instances. By opting for TRUE DDNS, organizations and individuals can apply the benefits of IP address management without the financial obligation associated with static IP addresses.

TRUE DDNS provides an additional layer of security by allowing users to access services behind a dynamic IP address using a domain name. This feature shields the IP address from direct disclosure, decreasing potential attacks or exploitation attempts. By keeping the IP address hidden and using a domain name instead (name.trueddns.com), TRUE DDNS offers enhanced security measures and helps safeguard sensitive information or services from malicious parties.

In summary, TRUE DDNS offers a range of additional benefits, including compatibility with dynamic DNS clients, support for custom domain names, extensive documentation and support resources, and integration possibilities with other services. These features enhance the overall user experience, streamline management processes, and contribute to the effectiveness and efficiency of TRUE DDNS as a dynamic IP address management solution.

3.2.3. Development

In the development process, the first step is to thoroughly understand and analyze the data in the documents. Due to the abundance of data in the document, it is necessary to clean the data before using it, as some columns may be unnecessary in the e-tax invoice file. Therefore, they need to be removed before proceeding. Understanding the structure of the e-tax invoice is crucial since it contains sensitive data. Researching e-tax invoice standards is also essential during development to find the best solution for users.

The front end serves as the customer-facing platform, showcasing the unique design of the store to customers. Additionally, user experience plays a vital role, including a user-friendly design and error handling that effectively guides users in case of missing or repetitive data entry.

3.2.3.1 Data Format

In this research, authors use JSON data format to exchange data between the client and the server. The reason for choosing JSON data format is that JSON is a lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate. For example, submitted form requests are considered to be user-generated data which JSON is suited to store static data format. Additionally, static JSON data can be used to display information on the page without the need for an additional request to the server. There are two main groups of data used in this research which are Customer filled form and Order transaction data as described below.

1) Customer filled form data format

Each customer filled form is represented as a JSON document that includes various fields with information about the order and the customer who placed it. Below is a list of the fields included in a customer filled form data format, along with a brief explanation of their purpose.

1.1) "_id": This field represents the unique identifier of the document and is automatically generated by the database system. In this case, it's an ObjectId that has a value of "6455541d6542d680379a8381".

1.2) "orderNumber": This field represents the unique order number associated with the document and has a value of "2023010541250503".

1.3) "firstname": This field represents the first name of the customer who placed the order and has a value of "John".

1.4) "lastname": This field represents the last name of the customer who placed the order and has a value of "Doe".

1.5) "address": This field represents the customer's address and has a value of "Bangkok, Thailand".

1.6) "taxID": This field represents the customer's tax ID and has a value of "1234567890123".

1.7) "mobileNum": This field represents the customer's mobile phone number and has a value of "0847920384".

1.8) "_v": This field represents the document's version and is automatically managed by the database system. In this case, it has a value of 0.

1.9) "invoiceDownloaded": This field represents whether the invoice associated with this order has been downloaded or not and has a value of "true".


```
{
"_id":{"$oid":"6455541d6542d680379a8381"},
"orderNumber":"2023010541250503",
"firstname":"John",
"lastname":"Doe",
"address":"Bangkok,Thailand",
"taxID":"1234567890123",
"mobileNum":"0847920384",
"__v":{"$numberInt":"0"},
"invoiceDownloaded":true
}
```


2) Order transaction data format

The raw data related to order transactions is stored in a CSV (Comma-Separated Values) file format. In the order transaction data structure, each row of the CSV file represents a single order, and each cell within that row represents a specific product associated with that order. If an order contains multiple products, each product will be represented by its own cell within the same row as the order number. For example, if an order with the number "20230515" contains three products, there will be three cells in the same row as "20230515", with each cell representing a different product. Not only the file type that is not in the right format, but the data also includes key fields in Thai and null values, it becomes crucial to manage the data properly to ensure accurate tracking and reporting. The figure below shows the raw data of order transactions.

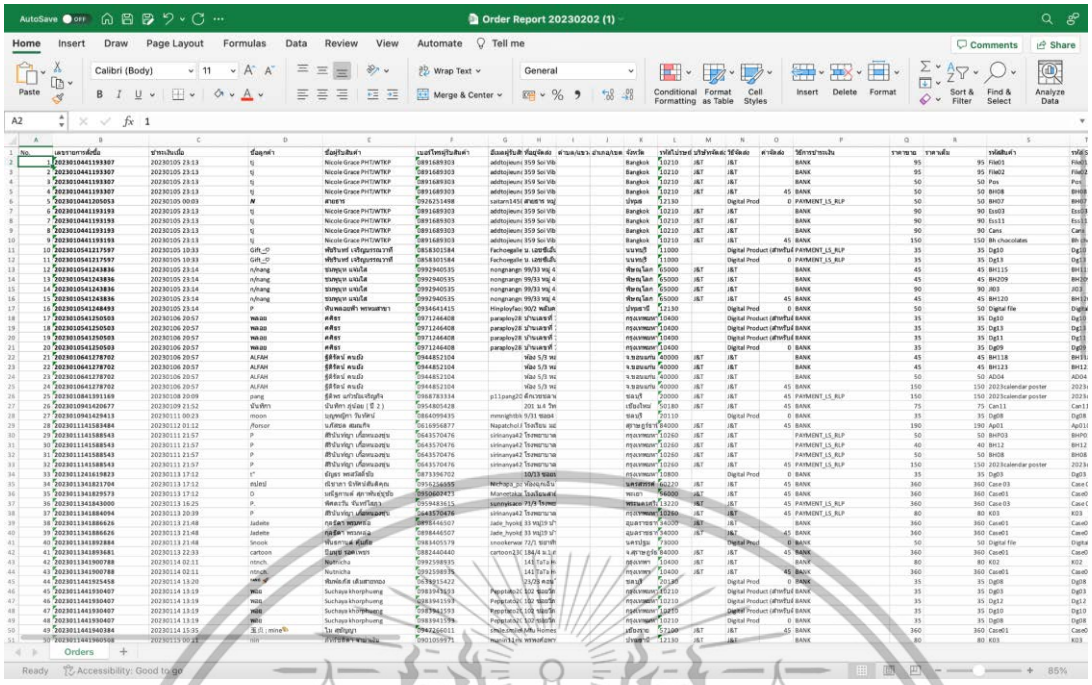

Figure 30 Unorganized Order Transaction data in .csv format

The authors manage the key fields in Thai and null values by manually changing the key fields into English and filling the null values as 0. Additionally, select only the necessary fields including order number, product number, product name, quantity, price, and discount.

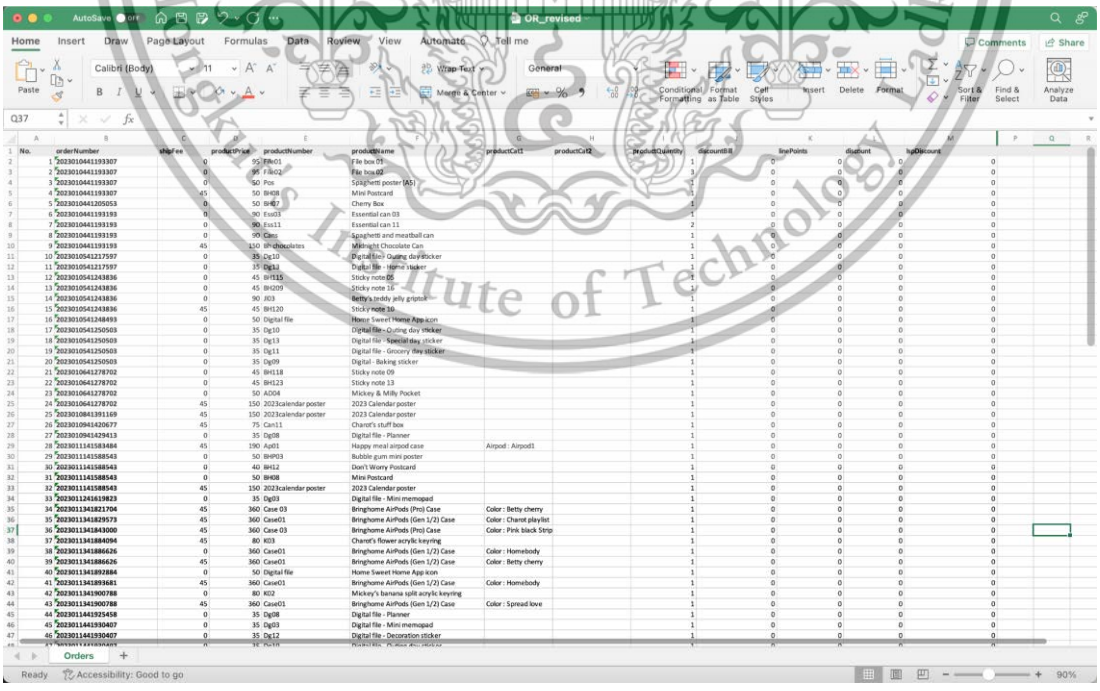

Figure 31 Organized Order Transaction data in .csv format

Once the order transaction data in CSV format has been properly managed and structured, it can be converted into JSON (JavaScript Object Notation) format using an online converter. This process involves taking the modified CSV file and inputting it into the online converter, which will then transform the data into a JSON file format and write to MongoDB as shown below.

2.1) "_id": This field represents the unique identifier of the document and is automatically generated by the database system.

2.2) "No.": This field represents the order number associated with the product.

2.3) "orderNumber": This field represents the unique order number associated with the product.

2.4) "shipFee": This field represents the shipping fee for the product.

2.5) "productPrice": This field represents the price of the product.

2.6) "productNumber": This field represents the product number or SK.

2.7) "productName": This field represents the name of the product.

2.8) "productQuantity": This field represents the quantity of the product ordered.

2.9) "discountBill": This field represents the discount applied to the product based on the customer's billing details.

2.10) "linePoints": This field represents any loyalty program points earned for the purchase.

2.11) "discount": This field represents any discount applied to the product.

2.12) "lspDiscount": This field represents any discount applied to the product based on the customer's loyalty program status.


```
{
"_id":{"$oid":"644221bf59f84e88dfc0d2dd"},
"No.":{"$numberInt":"5"},
"orderNumber":"2023010441205053",
"shipFee":{"$numberInt":"0"},
"productPrice":{"$numberInt":"50"},
"productNumber":"BH07",
"productName":"Cherry Box",
"productQuantity":{"$numberInt":"1"},
"discountBill":{"$numberInt":"0"},
"linePoints":{"$numberInt":"0"},
"discount":{"$numberInt":"0"},
```

"lspDiscount":{"$numberInt":"0"},
}

In conclusion, the selection of JSON was motivated by its lightweight nature, making it highly readable and writable for humans, while also being easily parsed and generated by machines. By utilizing JSON, the research effectively manages submitted form requests, which consist of user-generated data, in a static data format that suits the purpose. Moreover, the utilization of static JSON data enables the seamless display of information on web pages without the need for additional server requests.

3.2.3.2 E-tax Invoice Operation

In this research, authors used Vanilla JavaScript, NodeJS and Handlebars in generating dynamic PDF tax invoices with both static and dynamic data on a NodeJS app running on a web browser. PDF tax invoice mainly consist of two parts, data model and e-tax invoice template The data used are separated into static and dynamic data. The static data is the company's information including company name, company tax issuer identification number, address and so on. Conversely, customer data received from filled form and order transaction data retrieved from MongoDB are considered to be dynamic data. The template of e-Tax invoice was created using Handlebars to create dynamic PDF documents like tax invoice.

1) Data model of e-Tax Invoice

The data used are separated into static and dynamic data. The static data is the company's information including company name, company tax issuer identification number, address and so on. Conversely, customer data received from filled form and order transaction data retrieved from MongoDB are considered to be dynamic data as shown below (finalOrder object.) Once the data model is completed, it will be parsed and compiled into a pre-designed template (invoiceTemplate.hbs). Finally, the invoiceTemplate function is called with the properties passed in as context, which later generates an HTML invoice.

```
finalOrder = {
        customerData: customerData,
    orderData: orderData,
    orderNumber: customerData.orderNumber,
    subtotal: parseFloat(totalBeforeTax).toFixed(2),
    discount: parseFloat(sumDiscount).toFixed(2),
    shipFee: parseFloat(shipFee).toFixed(2),
    vat: parseFloat(vat).toFixed(2),
    gtotal: parseFloat(grandTotal).toFixed(2)
}
```

```
const htmlTemplate = fs.readFileSync('./template/invoiceTemplate.hbs', 'utf8')
const invoiceTemplate = Handlebars.compile(htmlTemplate)

invoiceData = invoiceTemplate({
   finalOrder: finalOrder,
   customerData: finalOrder.customerData,
   date: date
})
```

2) e-Tax Invoice Template

The template of tax invoice is developed using Handlebars (invoiceTemplate.hbs) which finalOrder will be parsed as a JavaScript object into a template file. To illustrate from the syntax below, finalOrder.customerData.mobileNum is parsed to represent the mobile number property of filled form from customerData schema of finalOrder object.

```
// invoiceTemplate.hbs

<!-- customer details: name, address, tax ID and telephone number -->
    <table class="table user-details">
       <tr>
        <td scope="col">
          <h3
class="name">{{finalOrder.customerData.firstName}} {{finalOrder.customerData.lastName}}</h3>
          <span>{{finalOrder.customerData.address}}</span>
          <br>
          <span>โทรศัพท์: {{finalOrder.customerData.mobileNum}}</span>
          <br>
          <span>เลขประจำตัวผู้เสียภาษี: {{finalOrder.customerData.taxID}}</span>
        </td>
        {{!-- end of first cell --}}

        <td scope="col">
         <div>
          <div>วันที่: {{date}}</div>
          <div>เลขที่ใบกำกับภาษี: 1</div>
           <div>เลขที่คำสั่งซื้อ: {{finalOrder.orderNumber}}</div>
         </div>
        </td>
        {{!-- end of second cell --}}

       </tr>
     </tbody>
    </table>
    <!-- end of customer details -->
```

```
        // ...

  <!-- end of container wrapper -->
 </div>
 <!-- end of container -->
</body>
</html>
```

Additionally to loop representing the item order. The `each` helper helps iterate over an array of objects stored in `finalOrder.orderData`. For each object in the array, the template generates an HTML table row with five columns. The first column displays the index of the current object in the array, incremented by one using the `addOne` helper. The second column displays the product ID and name of the current object, enclosed in square brackets and separated by a dash. The third column displays the quantity of the product ordered, the fourth column displays the price of the product, and the fifth column displays the total price of the product, calculated as the quantity multiplied by the price.

```
<!-- item details are represented below in invoice table -->
<div class="table-responsive">
 <table class="table table-sm" style="padding: 10px">

 <!-- table header -->
 <thead>
  <tr style="background-color: #F6CBCF">
   <th scope="col" style="text-align: left">ลำดับ</th>
   <th scope="col" style="text-align: left">รายการสินค้า</th>
   <th scope="col" style="text-align: right">จำนวน</th>
   <th scope="col" style="text-align: right">ราคาต่อหน่วย</th>
   <th scope="col" style="text-align: right">จำนวนเงิน</th>
  </tr>
 </thead>
 <!-- end of table header -->

 <!-- table main content: loop show item datas -->
 <tbody>
   {{#each finalOrder.orderData}}
   <tr style="border-bottom: 1px solid #EEEEEE;">
    <td scope="row" class="cell-size" style="text-align: left">{{addOne @index}}</td>
     <td style="text-align: left">[{{this.prodId}}] {{this.prodName}}</td>
    <td style="text-align: right">{{this.quantity}}</td>
    <td style="text-align: right">{{this.price}}</td>
```

```
          <td style="text-align: right">{{this.totalFloat}}</td>
         </tr>
        {{/each}}

                    // ...

      </tbody>
      <!-- end of table body -->

      <!-- table footer: summary of order -->
      <tfoot>
       <tr>
        <th colspan="4" style="text-align: right;">รวมเป็นเงิน</td>
        <th colspan="4" style="text-align: right; border-bottom: 1px solid
#EEEEEE;">{{finalOrder.gtotal}} บาท</td>
       </tr>

       <tr>
        <th colspan="4" style="text-align: right;">ภาษีมูลค่าเพิ่ม 7%</td>
        <th colspan="4" style="text-align: right; border-bottom: 1px solid #EEEEEE;">{{finalOrder.vat}}
บาท</td>
       </tr>

       <tr>
        <th colspan="4" style="text-align: right;">ส่วนลด</td>
        <th colspan="4" style="text-align: right; border-bottom: 1px solid
#EEEEEE;">{{finalOrder.discount}} บาท</td>
       </tr>

       <tr>
        <th colspan="4" style="text-align: right;">ค่าจัดส่ง</td>
        <th colspan="4" style="text-align: right; border-bottom: 1px solid
#EEEEEE;">{{finalOrder.shipFee}} บาท</td>
       </tr>

        <tr>
        <th colspan="4" style="text-align: right; background-color:
#F6CBCF;"><b>จำนวนเงินรวมทั้งสิ้น</b></td>
        <th colspan="4" style="text-align: right; background-color: #F6CBCF;">{{finalOrder.subtotal}}
บาท</td>
       </tr>
      </tfoot>
      <!-- end of table footer-->

    <!--end of invoice table container  -->
    </table>
```

```
</div>
<!-- end of invoice table -->
```

In conclusion, the authors have successfully demonstrated the use of Vanilla JavaScript, NodeJS, and Handlebars in generating dynamic PDF tax invoices on a web browser-based NodeJS app. The tax invoices were generated by combining static data, which contains the company's information, with dynamic data, which includes customer data from a filled form and order transaction data retrieved from MongoDB. Handlebars was utilized to create the e-Tax invoice template, enabling the creation of dynamic PDF documents such as tax invoices. The approach used in this research can be applied in various business industries that require the generation of PDF tax invoices.

3) Digital Signature

Digital signatures are widely used to ensure the integrity and authenticity of electronic documents. A digital signature provides assurance that the document has not been altered since it was signed and that the signer is who they claim to be. Type of digital signature used in this research which also meets the recommendation of the Revenue Department is the XML Advanced Electronic Signatures (XAdES) signature. The XAdES signature consists of two main components: the signature itself and the signed properties. The signature contains the cryptographic information that ensures the integrity and authenticity of the signed document, while the signed properties provide additional information about the signature, such as the signing time and the signer's identity. The tool should specify the signing algorithm and provide a certificate for signing. The signing process involves the following steps:

- The document to be signed is converted to an XML format.
- The signing tool calculates a cryptographic hash of the document and signs the hash using the private key associated with the signer's certificate.
- The signature and signed properties are added to the XML document.
- The signed XML document can then be verified by a recipient using the signer's public key.

The above process is demonstrated when creating a tax invoice with an approved type of digital signature used as evidence that every time a tax invoice is generated, it will be signed before being downloaded by customers. The company can keep this file for future tax filing. However, if the company wants the format to be correct according to what the Revenue Department requires, the company must register and consult with

an accountant to proceed correctly. In this section, authors will illustrate the process of creating digital signatures for e-tax invoice by the following topics.

3.1) XAdES signature components

XAdES signature according to the suggestion of the Revenue Department consist of four mains components as stated below.

3.1.1) SignedInfo element

- CanonicalizationMethod: Citing the "enveloped-signature" canonicalization algorithm removes the Signature element from the document and then canonicalise the remaining content for signinga according to http://www.w3.org/2000/09/xmldsig#enveloped-signature
- SignatureMethod: Specifies the algorithm used for signature generation. This element must be present in the signature.
- Reference: Specifies the data that is being signed, along with information about how the data is located and how it is represented in the signature. A signature may have multiple reference elements.

2.1.2) Signature

Signature means a digital signature of signer with Base 64 encoding.

2.1.3) KeyInfo

KeyInfo specifies the Distinguished Name of the certificate holder in the electronic signature of the receipt of the X509 SubjectName. The certificate is identified by X509 and its Distinguished Name is specified by an element in the signature. The certificate is encoded in Base64.

2.1.4) SignedProperties

SignedProperties is an element found under QualifyingProperties, consisting of various elements that represent specific information related to the signing process in the XMLDSIG format. These elements are located within the SignedProperties element and include SigningTime, SigningCertificate, SignatureProductionPlace, and SignaturePolicy.

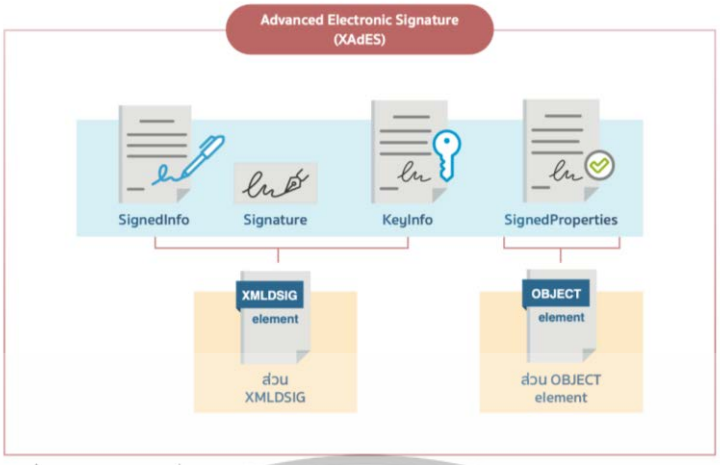

Figure 32 Components of XAdES Signature

3.2) Process of adding digital signature

The overview process of digitally signing an XML document involves generating a private key, applying a Hash function to the XML document to obtain a hash value, and encrypting the hash value with the private key to create the digital signature. In XAdES digital signature format.

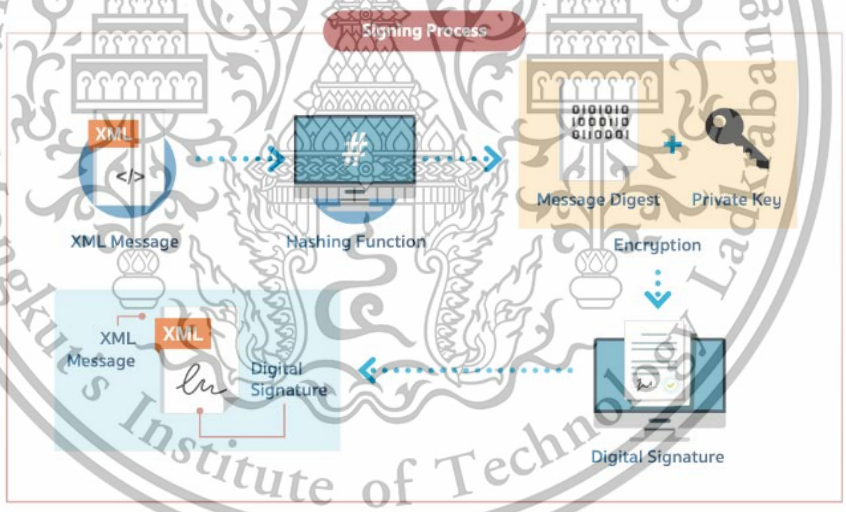

Figure 33 Signing Process

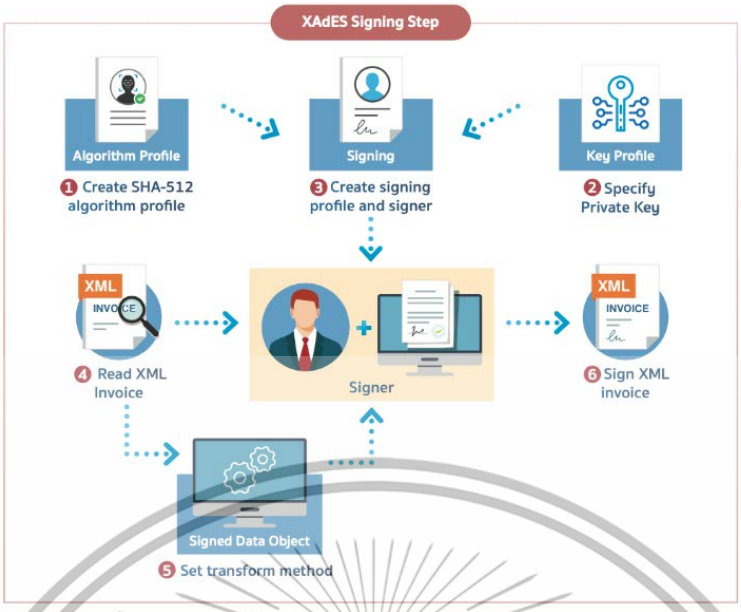

Figure 34 XAdES Signing Step

In this research, the XAdES process involves generating a private key, calculating a hash value of the XML document, creating a digital signature using the private key, including the public key certificate and signing time, and incorporating the signature and qualifying properties into the XML document. Base 64 encoding will be used for specific elements like the digest value and public key certificate for proper representation and transmission of the signature-related data. The process of creating XAdES will be explained as follows.

2.2.1) Private Key Generation

In XAdES, the private key is used to create a digital signature for a document. The private key is typically generated by the user and stored securely on their local machine or on a secure server. According to the source code, the privateKey variable was protected with the passphrase property which was encrypted during creation of the private key. The public key certificate associated with the signer's private key is included in the signature. The certificate contains the signer's public key and serves as a trusted reference for verifying the signature. It is typically encoded in Base 64 format for proper representation and transmission. The inclusion of a public key certificate in the digital signature process is an essential component for ensuring trust and verification. However, it's important to note that the public key certificate used in this research demonstration is self-authenticated and not officially certified. In the case of the company that the authors are working for, the certification process is still ongoing as the company is currently undergoing registration as a Company Limited. Therefore, the authors can only demonstrate the digital signature process using a self-authenticated

certificate, which serves as a representation for research purposes but lacks the official certification required for complete validity. To ensure full compliance and trustworthiness in real-world applications, it is necessary for the company to obtain an officially certified public key certificate from a recognized certification authority, such as EDTA. The following code block represents how the private key and the public key certificate were included in the process of preparing a digital signature.

```
//// Add XadES signature to finalOrder object
const sig = new SignedXml()

// Read the private key file and set it as the signing key
const privateKey = fs.readFileSync('./built/private_key.pem', 'utf-8')
sig.signingKey = {
    key: privateKey,
    passphrase: '// your passphrase here'
}

var certificate = fs.readFileSync('./built/certificate.pem', 'utf-8')
var encodedCertificate = Buffer.from(certificate).toString('base64');
  sig.keyInfoProvider = {
    getKeyInfo() {
     return `<X509Data><X509Certificate>${encodedCertificate}</X509Certificate></X509Data>`;
    }
}
```

2.2.2) XML Document Preparation and applied Hash Calculation

The XML document is prepared in the desired format and structure. Then, a hash function, in this case a SHA-256, is applied to the XML document. This generates a unique hash value, which serves as a fingerprint of the document's content. In this case, the XML document is an “xml” variable which is an invoice XML message prepared earlier.

```
<?xml version="1.0"?>
<TAXINV_2023010441193307 Id="_0">
 <InvoiceHeader>
  <Vendor>
   <Name>Bringhome.theBacon</Name>
   <Address>24/56 Bang Na-Trat Rd, Tambon Bang Sao Thong, Amphoe Bang Sao Thong, Chang Wat
Samut Prakan 10540</Address>
   <Phone type="telNumber">0987490837</Phone>
   <TaxId schemaID="TaxID">123456789012</TaxId>
  </Vendor>
  <Customer>
   <CustomerInfo>
```

```
      <CustomerName>
        <FirstName>Atita</FirstName>
        <LastName>Jityanan</LastName>
      </CustomerName>
      <Address>55/495 Aspire Sathorn-Taksin (Timber zone) Ratchapruk Rd., Bang ko, Bang ko, BKK,
10150</Address>
      <Phone>0909530385</Phone>
      <TaxId>1104700031805</TaxId>
    </CustomerInfo>
    <OrderHeaderInfo>
      <IssuedDateTime>5/16/2023</IssuedDateTime>
      <TaxInvoiceNumber>0.4764422354184352</TaxInvoiceNumber>
      <OrderNumber>2023010441193307</OrderNumber>
    </OrderHeaderInfo>
  </Customer>
 </InvoiceHeader>
 <OrderDetails>
  <ProductDetails id="1">
    <ProductID>File01</ProductID>
    <ProductName>File box 01</ProductName>
    <UnitPrice>95.00</UnitPrice>
    <Quantity>1</Quantity>
    <TotalPrice>95.00</TotalPrice>
  </ProductDetails>
  <ProductDetails id="2">
    <ProductID>File02</ProductID>
    <ProductName>File box 02</ProductName>
    <UnitPrice>95.00</UnitPrice>
    <Quantity>3</Quantity>
    <TotalPrice>285.00</TotalPrice>
  </ProductDetails>
  <ProductDetails id="3">
    <ProductID>Pos</ProductID>
    <ProductName>Spaghetti poster (A5)</ProductName>
    <UnitPrice>50.00</UnitPrice>
    <Quantity>1</Quantity>
    <TotalPrice>50.00</TotalPrice>
  </ProductDetails>
  <ProductDetails id="4">
    <ProductID>BH08</ProductID>
    <ProductName>Mini Postcard</ProductName>
    <UnitPrice>50.00</UnitPrice>
    <Quantity>1</Quantity>
    <TotalPrice>50.00</TotalPrice>
  </ProductDetails>
 </OrderDetails>
```

```
 <EndofBillCalculation>
  <TotalBeforeVAT>525.00</TotalBeforeVAT>
  <Discount>0.00</Discount>
  <ShipFee>45.00</ShipFee>
  <VAT>36.75</VAT>
  <GrandTotal>443.25</GrandTotal>
 </EndofBillCalculation>
</TAXINV_2023010441193307>
```

After preparing the tax invoice in XML format, the next step is to create a digital signature to be added, as per the following code snippet.

```
const xpath = `//*[local-name()="TAXINV_${orderNumber}"]`
const xml = doc.end({ pretty: true })

sig.addReference(xpath, ['http://www.w3.org/2000/09/xmldsig#enveloped-signature'])
sig.computeSignature(xml)
let signedXml = sig.getSignedXml()
```

2.2.3) Adding Object property

XAdES includes a set of qualifying properties that provide additional information about the signature which is the QualifyingProperties. The QualifyingProperties must include xades:SignedProperties, xades:SigningTime and xades:SigningCertificate.

```
const objectXml = '<Object></Object>'
const cert = forge.pki.certificateFromPem(certificate)
const issuerName = cert.issuer.attributes.map(attr => `${attr.shortName}=${attr.value}`).join(', ')
const serialNumber = cert.serialNumber.toString()

// Create the qualifying properties
const qualifyingPropertiesXml = `<xades:QualifyingProperties
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Target="#xmldsig-69b3d670-04e4-419f-8d45-
938ea867f571"><xades:SignedProperties Id="xmldsig-69b3d670-04e4-419f-8d45-938ea867f571-
signedprops"><xades:SignedSignatureProperties><xades:SigningTime>2023-05-
06T12:34:56.789Z</xades:SigningTime><xades:SigningCertificate><xades:Cert><xades:CertDigest><Di
gestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/><DigestValue>${encodedCertificate}</DigestValue></xades:CertDigest><xades:IssuerSerial><X509Is
suerName>${issuerName}</X509IssuerName><X509SerialNumber>${serialNumber}</X509SerialNumb
er></xades:IssuerSerial></xades:Cert></xades:SigningCertificate></xades:SignedSignatureProperties>
</xades:SignedProperties></xades:QualifyingProperties>`

// Add the qualifying properties to the object
const objectWithQualifyingPropertiesXml = objectXml.replace('</Object>', qualifyingPropertiesXml +
'</Object>')

// Add the object to the signed XML document
```

```
const signedXmlWithObject = signedXml.replace('</Signature>', objectWithQualifyingPropertiesXml +
'</Signature>')
```

The Signature element will be added at the end of the tax invoice as illustrated below.


```
<?xml version="1.0"?>
<TAXINV_2023010441193307 Id="_0">

        // Tax Invoice Content

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                <Reference URI="#_0">
                <Transforms><Transform
        Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/></Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <DigestValue>GUez7XH+22jGiQFAgi3Z21rvH0I=</DigestValue></Reference>
        </SignedInfo>
        <SignatureValue>vzm5pe5HX6a86OZv7+ny9mBq5DyStRIn+3QcuVyGwcpFy9cAPfdld+eNCoGqR
m3hLbIWj3TXWHWuWCQaCHVHTnSeIPHbDdLkwhMEpLCRjQVDbjF1CqJrU85QwBdJuyBpyT9j3Xu338fHt
3UwWw40tDBmL+h31A1K/p4ddEeaBPkueWSoko2/sehbi+wfF585YV04pjhvwV2izLWMjl6cUkSWyoCzA1
mo5kOAB73vHyee3UKMQaSCtLVrvV1B21j/L0squBk1VjMgNb9KtRRavwXL0U1m5MbGBZ8hFMA67gM89
9SK4LoePgba6AQIuljE2pSjn0+KOZDpADNCCLlTAw==
        </SignatureValue>
        <KeyInfo>
                <X509Data>
                        <X509Certificate>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUR2REND
                QXFRQ0NRQ3FWc3FPNDdRZUJEQU5CZ2txaGtpRzl3MEJBUXNGQURDQm56RUxNQWtH
                QTFVRUJoTUMKZEdneEVEQU9CZ05WQkFnTUIySmhibWRyYjJzeEVEQU9CZ05WQkFjTUI
                zTmhkR2h2Y200eEd6QVpCZ05WQkFvTQpFbUp5YVc1bmFHOXRaUzUwYUdWaVlXTnZia
                kVPTUF3R0ExVUVDd3dGYTIxcGRHd3hHekFaQmdOVkJBTU1FbUp5CmFXNW5hRzl0WUl
                M1MGFHVmlZV052YmpFaU1DQUdDU3FHU0liM0RRRUpBUllUWVhScGRHRXVhbWwwU
                UdkdFlXbHMMKTG1OdmJUQWVGdzB5TXpBMU1EUXdPREUyTXpKYUZ3MHlOREExTURNd
                09ERTJNekphTUlHZk1Rc3dDUVlEVlFRRwpFd0owYURFUU1BNEdBMVVFQ0F3SFltRnVa
                MnR2YXpFUU1BNEdBMVVFQnd3SGMyRjBhRzl5YmpFYk1Ca0dBMVVFCkNnd1NZbkpwY
                m1kb2IyMWxMblJvWldKaFkyOXVNUTR3REFZRFZRUUxEQVZyYldsMGJERWJNQmtHQTF
                VRUF3d1MKWW5KcGJtZG9iMjFsTG5Sb1pXSmhZMjl1TVNJd0lBWUpLb1pJaHZjTkFRa0J
                GaE5oZEdsMFlTNXFhWFFBWjIxaApV3d1WTI5dE1JSUJJakFOQmdrcWhraUc5dzBCQVF
                FRkFBT0NBUThBTUlJQkNnS0NBUUVBMFhBK1hrVytFNG5KCmdzbllWZjgvc2g5MkxxaRjlt
                SFdQOFZoOHZGQU8rR2dOay9UaHpVNHc0NFgwM3I3REFFTLzgzSjFpWE5XV3R2djcKUW
                ZySEpnUjF1R0dRUjlFK1JCVFhVNVcrVm4xZCs2OXp4Yk4yb0xlZFRSRENWSjlGUWFPVjVP
                dXgyVkxqeUppSApidGNjd0tUYWVsMDNlK0NMc2FMd1VyQW1OQVJjanJaMnFEQ2p6b1k
                1azNMYkduTlo1RTY3SFAwVzhrUlN5NS9mCjA5Z0R3djNWVjhlai9lVmZaQkdiTC92RU8xb
```

XFKR1dXdlBubHRncW5mcy9ZZFJBR3RWQ1hCMDhKNklUdng0SEYKZVRQaUpCSkVheSt
rWmorL3E4NTBUVTRUS2xnbkU3bXVLQ2xBTnQyVlQ0dm5sQmVwckFoTFFPQ0g0UTRvQ
k51dApEdWxWQzd3cmdRSURBUUFCTUEwR0NTcUdTSWIzRFFFQkN3VUFBNElCQVFDZn
BacGpkYm9kWWYrdFdsYk9QZGVmCnEyeU9OWGtIUVVsSlI3RXZQRWhETnpacjdTUzdvdl
ViRCtXeXJ2TVpKTXhkdWkwdFdEQ3U0RTF0bTl6U3krSkMKNkhzbzFOZ0ZLK3lqeGpzSVY
0Mkp1RUdFUkRJaFVnYktRcXV1VkYrR0pQWmZQeGVOS3RXL2xMMldaUzk3b3F4VApOa
3dnQlFwMXBTSVNMVzFIZG9GTGcrWGdDZjlqdkwrQjdTUnJEODVhNHlxWCt0UEVlUWpT
K2dpNElJZjVuaGFkCkVlREx3UHpCTVpYVUNGSnJqbWg1a3ZiTkNmdFlhY0kyTk5LL2g2S
Fg2VnRibVRieXh2MnV5d3hyRmd4N1k5cWcKa3FtQUUyRVpoblFZTW9DTFdJdEJUQVpG
Y3BjWUdpT1MvQmtNR0FUSno5SHBYb3RueHlJRVFudFhSbW9hNjZITQotLS0tLUVORCB
DRVJUSUZJQ0FURS0tLS0tCg==
</X509Certificate>
</X509Data>
</KeyInfo>
<Object>
<xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#"
Target="#xmldsig-69b3d670-04e4-419f-8d45-938ea867f571">
<xades:SignedProperties Id="xmldsig-69b3d670-04e4-419f-8d45-938ea867f571-
signedprops">
<xades:SignedSignatureProperties>
<xades:SigningTime>2023-05-15T18:06:14.866Z</xades:SigningTime>
<xades:SigningCertificate>
<xades:Cert>
<xades:CertDigest>
<DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/><DigestValue>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUR
2RENDQXFRQ0NRQ3FWc3FPNDdRZUJEQU5CZ2txaGtpRzl3MEJBUXNG
QURDQm56RUxNQWtHQTFVRUJoTUMKZEdneEVEQU9CZ05WQkFnTUI
ySmhibWRyYjJzeEVEQU9CZ05WQkFjTUIzTmhkR2h2Y200eEd6QVpCZ0
5WQkFvTQpFbUp5YVc1bmFHOXRaUzUwYUdWaVlXTnZiakVPTUF3R0E
xVUVDd3dGYTIxcGRHd3hHekFaQmdOVkJBTU1FbUp5CmFXNW5hRzl0
WlM1MGFHVmlZV052YmpFaU1DQUdDU3FHU0liM0RRRUpBUllUWVhS
cGRHRXVhbWwwUUdkdFlXbHMKTG1OdmJUQWVGdzB5TXpBMU1EUX
dPREUyTXpKYUZ3MHlOREExTURNd09ERTJNekphTUlHZk1Rc3dDUVlE
VlFRRwpFd0owYURFUU1BNEdBMVVFQ0F3SFltRnVaMnR2YXpFUU1BN
EdBMVVFQnd3SGMyRjBhRzl5YmpFYk1Ca0dBMVVFCkNnd1NZbkpw
1kb2IyMWxMblJvWldKaFkyOXVNUTR3REFZRFZRUUxEQVZyYldsMGJER
WJNQmtHQTFVRUF3d1MKWW5KcGJtZG9iMjFsTG5Sb1pXSmhZMjl1TV
NJd0lBWUpLb1pJaHZjTkFRa0JGaE5oZEdsMFlTNXFhWFJBWjIxaAphV3
d1WTI5dE1JSUJJakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBTUlJQ
kNnS0NBUUVBMFhBK1hrVytFNG5KCmdzbllWZjgvc2g5MkxaRjltSFdQO
FZoOHZGQU8rR2dOay9UaHpVNHc0NFgwM3I3REFTLzgzSjFpWE5XV3R
2djcKUWZySEpnUjF1R0dRUjlFK1JCVFhVNVcrVm4xZCs2OXp4Yk4yb0xl
ZFRSRENWSjlGUWFPVjVPdXgyVkxqeUppSApidGNjd0tUYWVsMDNlK0N
Mc2FMd1VyQW1OQVJjanJaMnFEQ2p6b1k1azNMYkduTlo1RTY3SFAwV

zhrUlN5NS9mCjA5Z0R3djNWVjhIai9lVmZaQkdiTC92RU8xbXFKR1dXdl
BubHRncW5mcy9ZZFJBR3RWQ1hCMDhKNklUdng0SEYKZVRQaUpCSk
VheStrWmorL3E4NTBUVTRUS2xnbkU3bXVLQ2xBTnQyVlQ0dm5sQmV
wckFoTFFPQ0g0UTRvQk51dApEdWxWQzd3cmdRSURBUUFCTUEwR0N
TcUdTSWIzRFFFQkN3VUFBNElCQVFDZnBacGpkYm9kWWYrdFdsYk9QZ
GVmCnEyeU9OWGtIUVVsSlI3RXZQRWhETnpacjdTUzdvdlViRCtXeXJ2T
VpKTXhkdWkwdFdEQ3U0RTF0bTl6U3krSkMKNkhzbzFOZ0ZLK3lqeGpz
SVY0Mkp1RUdFUkRJaFVnYktRcXV1VkYrR0pQWmZQeGVOS3RXL2xMM
ldaUzk3b3F4VApOa3dnQlFwMXBTSVNMVzFIZG9GTGcrWGdDZjlqdkwr
QjdTUnJEODVhNHlxWCt0UEVlUWpTK2dpNElJZjVuaGFkCkVlREx3UHp
CTVpYVUNGSnJqbWg1a3ZiTkNmdFlhY0kyTk5LL2g2SFg2VnRibVRieXh
2MnV5d3hyRmd4N1k5cWcKa3FtQUUyRVpoblFZTW9DTFdJdEJUQVpG
Y3BjWUdpT1MvQmtNR0FUSno5SHBYb3RueHlJRVFudFhSbW9hNjZITQ
otLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
</xades:CertDigest>
<xades:IssuerSerial>
<X509IssuerName>C=th, ST=bangkok, L=sathorn,
O=bringhome.thebacon, OU=kmitl, CN=bringhome.thebacon,
E=atita.jit@gmail.com</X509IssuerName>
<X509SerialNumber>00aa56ca8ee3b41e04</X509SerialNum
ber>
</xades:IssuerSerial>
</xades:Cert>
</xades:SigningCertificate>
</xades:SignedSignatureProperties>
</xades:SignedProperties>
</xades:QualifyingProperties>
</Object>
</Signature>
</TAXINV_2023010441193307>

Throughout the creation of the digital signature, cryptographic operations such as hashing and encryption are employed to ensure the integrity and confidentiality of the data. The resulting XAdES signature serves as a verifiable proof of the document's origin, integrity, and signer's identity.It is important to note that the use of XAdES signatures in tax invoices or any other legal documents may require adherence to specific regulations and standards imposed by tax authorities or governing bodies. Compliance with these requirements ensures the acceptance and legal validity of the digitally signed documents.

3.2.3.3 Frontend

The front-end structure of the project is separated into two important parts: main page and downloading page. To design the appearance of the webpage and animation using HTML, CSS, and JavaScript. HTML is used to structure the content of the page, and CSS is used for styling and layout. JavaScript is used to add interactivity and dynamic functionality to the webpage.

1) UX/UI design

Designing user experience (UX) and user interface (UI) is a crucial aspect of creating intuitive and engaging digital products and platforms. UX/UI design focuses on crafting seamless interactions between users and digital interfaces, ensuring that every touchpoint is optimized for user satisfaction and usability. Furthermore, incorporating elements from the company's products helps establish a seamless connection between the digital interface and the real-world offerings while also keeping the user experience in mind.

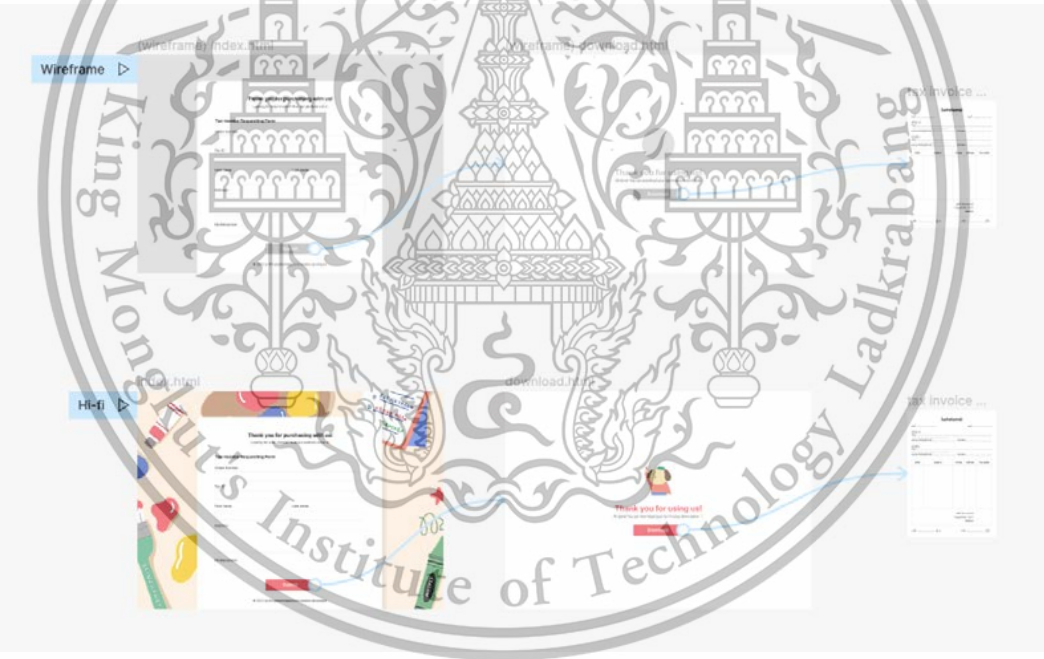

Figure 35 UX Design of Web Application

After reviewing the design with the company and incorporating their feedback, the authors are ready to proceed to the final design phase and initiate the development process. The following is a detailed list describing the UI pages and elements.

1.1) Customer filling form (index.html)

the customer filling form (index.html) located in the view folder, which is used to request an E-tax invoice and contains the necessary information required for the invoice,

including the customer's name, address, and Tax ID. The form is designed to be user-friendly and easy to navigate, with clear instructions on how to fill in each field.

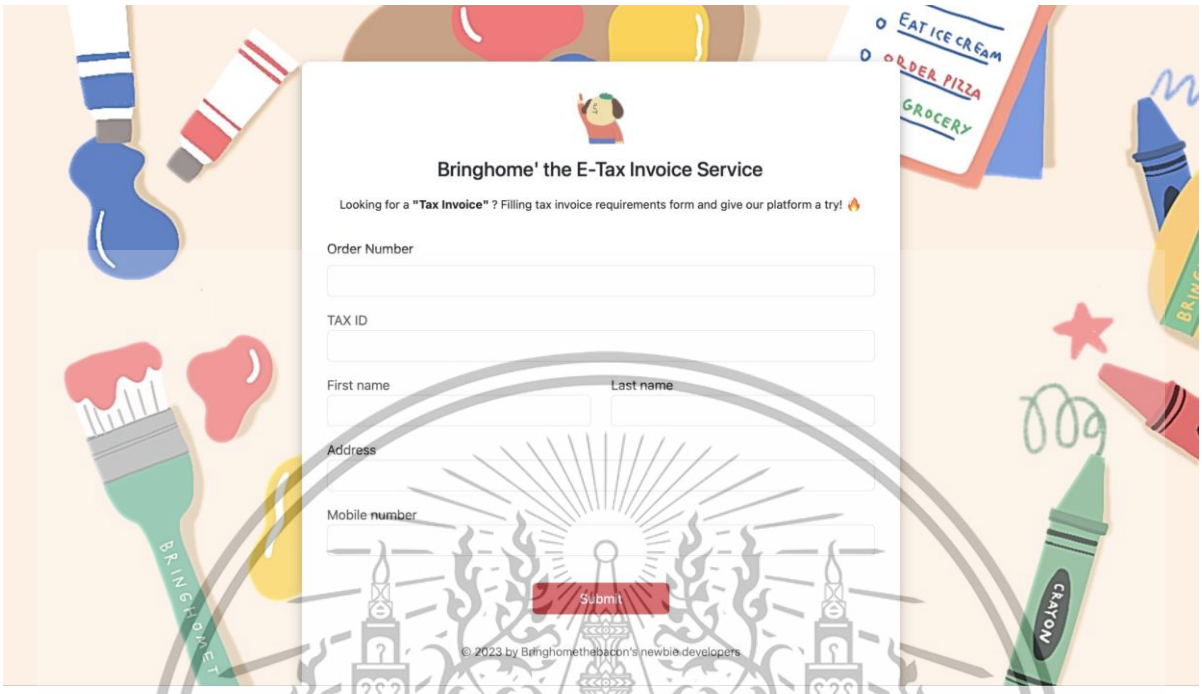

Figure 36 Customer filling form user interface

In addition, the Order Number is also required to be filled in the form, in order to retrieve the list of products with the same order number requested by the customer.

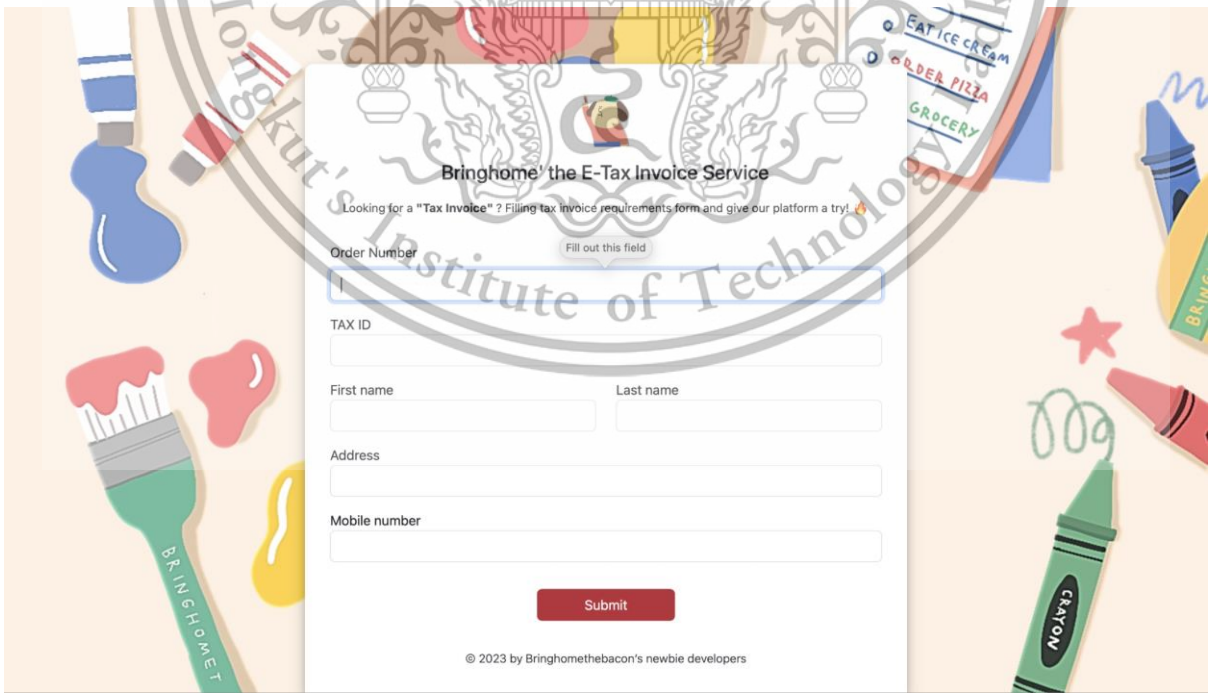

Figure 37 Required condition to fill the order number

This order number is obtained from the customer's purchase approval process, which is dependent on the LINE shopping payment system. The order number is a unique identifier that is not duplicated, meaning that every time a transaction builds, a new and unique order number is generated. The order number is a significant element of the E-tax invoice process, as it helps to accurately track each transaction and ensure that customers will receive the correct invoice.

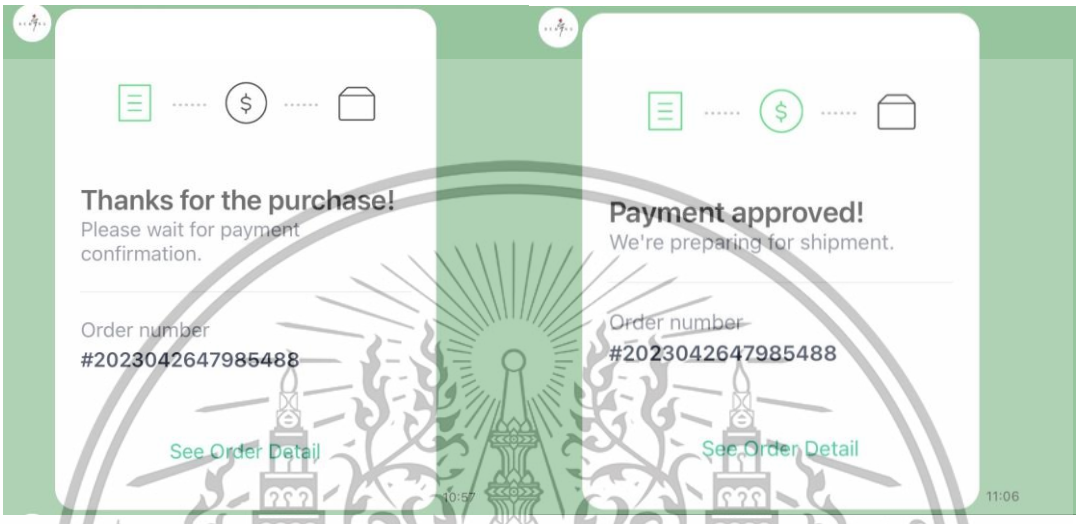

Figure 38 An order number after order is approved

Once the customer has filled in all the required information, they can submit the form by clicking the submit button. The information is sent to the back-end of the system known as the database to store the data, which processes the request and generates the E-tax invoice. After submitting the form, the process will move forward to the second part of the front-end, which involves downloading the page *(download.html)*.

1.2) Downloading page (download.html)

The second part of the front-end structure is the downloading page, which shows the status of E-tax invoice processing. After submitting the E-tax invoice form, the website leads the customer to the download page, which initially displays the status as "Generating". Once the document is completed, the status changes to display a "Download invoice" button for the customer to retrieve their invoice.

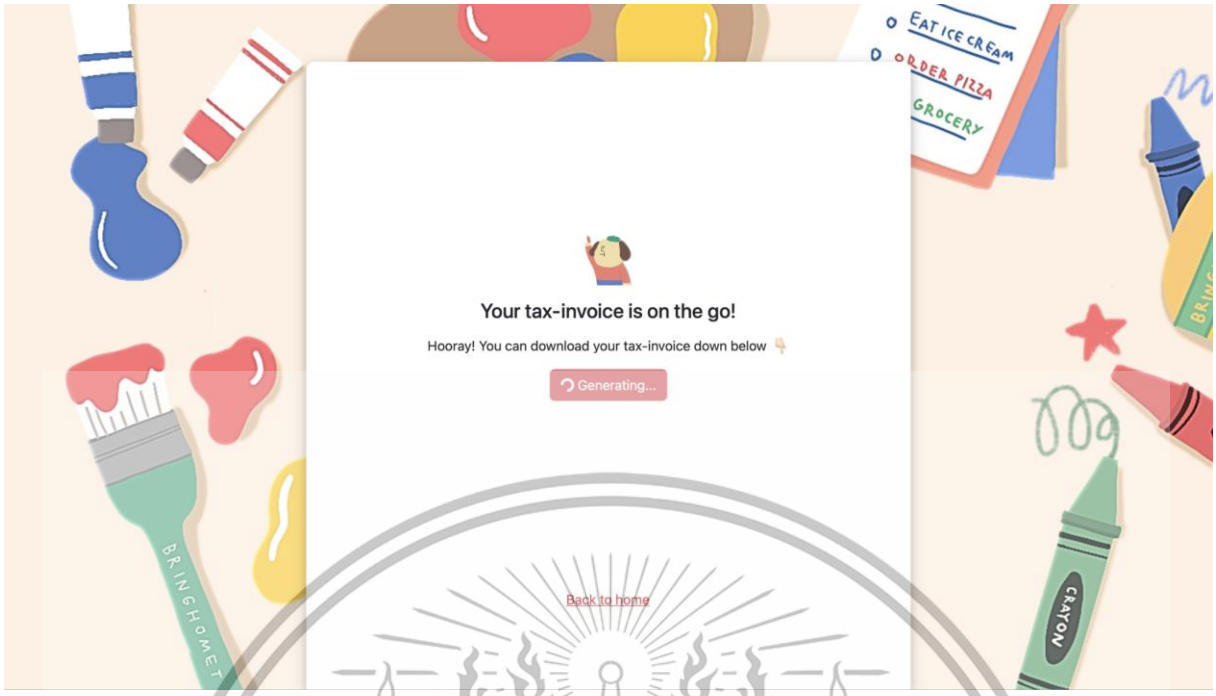

Figure 39 Downloading page shows the generating status

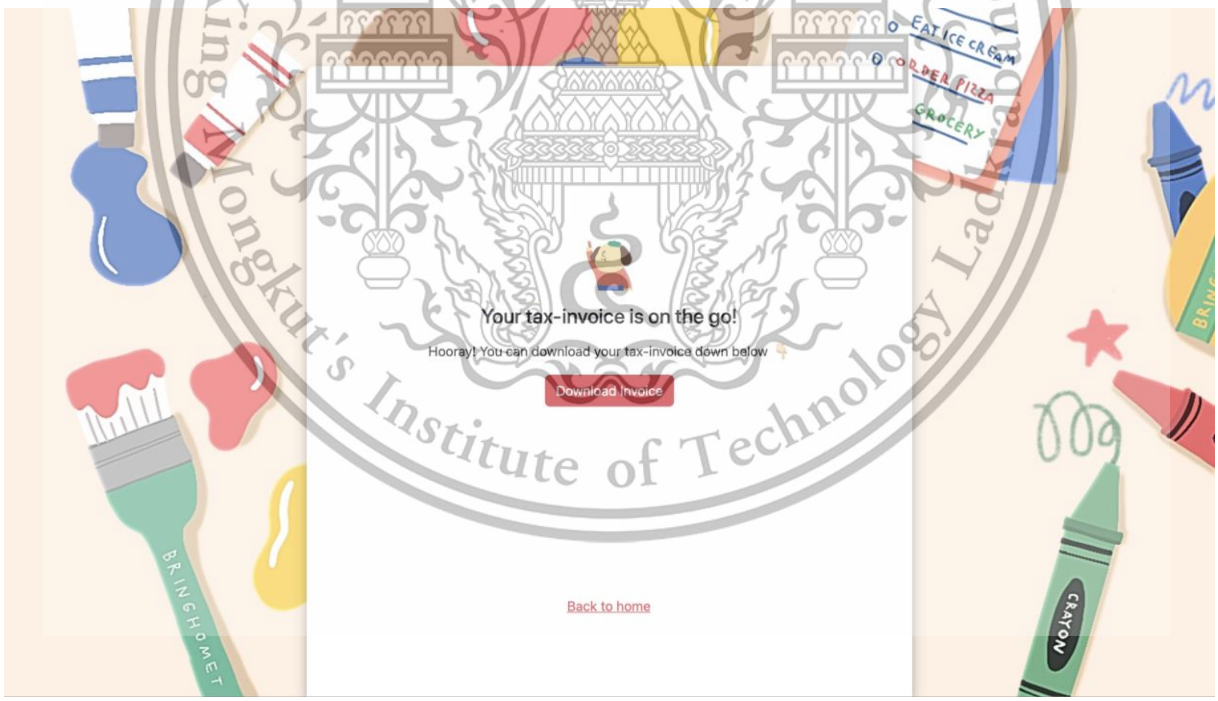

Figure 40 Change the status to download invoice when completed

3.2.3.4 Backend

1) Connect to mongoDB

Import mongoose as a library, an object data modeling for node.js and mongodb. The mongoose library allows to define the data using schema and provides an API for interacting with the MongoDB.

```
//require library mongoose
const mongoose = require('mongoose');
// handle HTTP POST
const express = require('express');
```

To get the URL endpoint to connect to a MongoDB database deployment, navigate to the 'Connect' button and select the 'Connect to Application' option. The sampling method will guide you on connecting MongoDB using the URL endpoint.

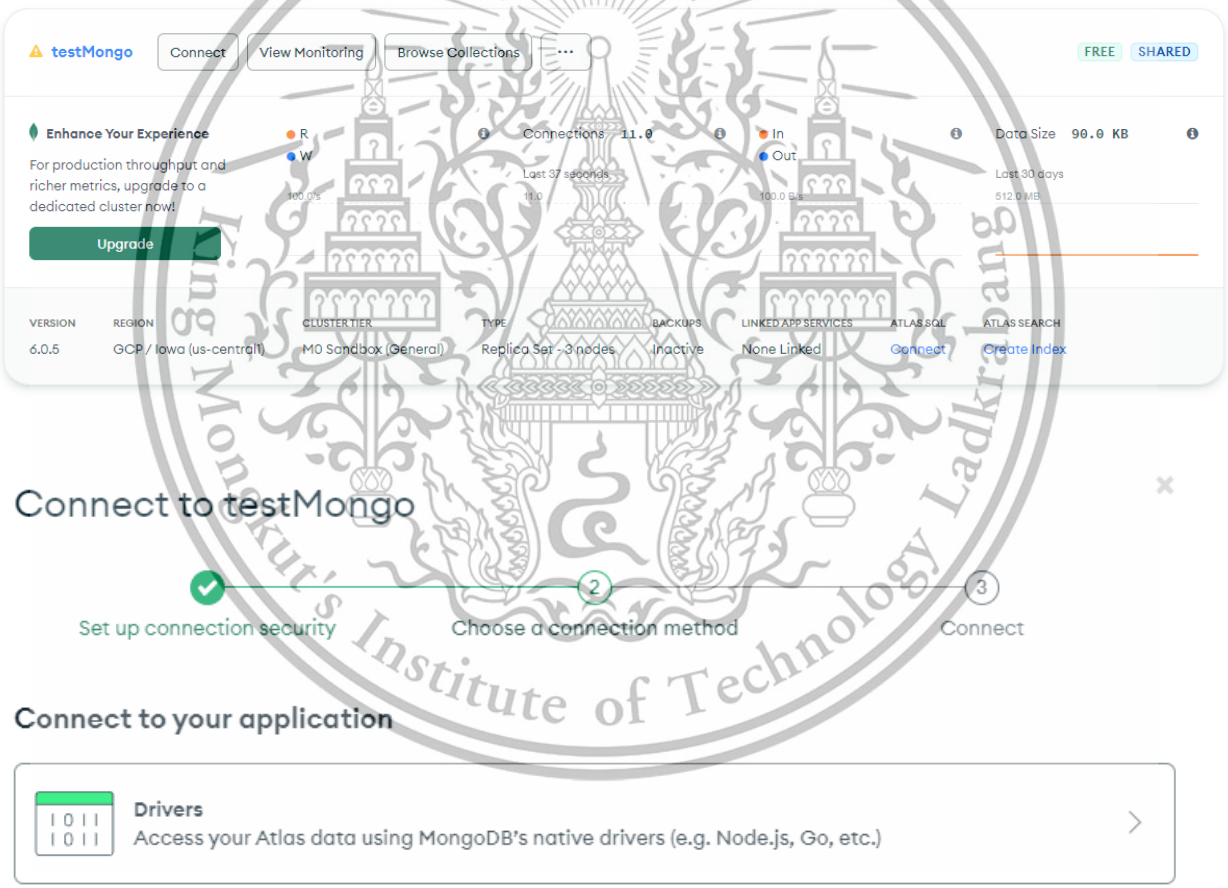

Figure 41 Step how to create MongoDB database

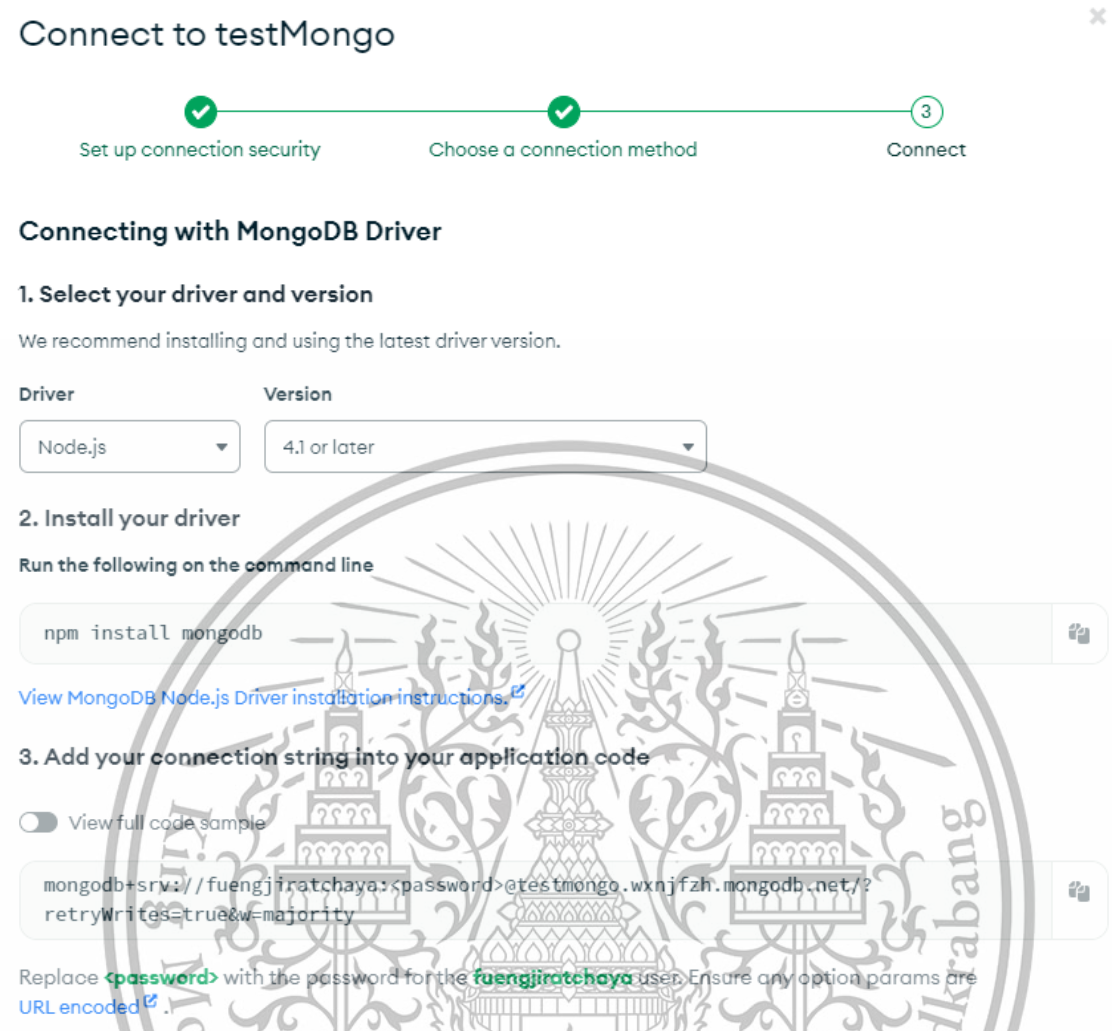

Figure 42 MongoDB connection guideline

In order to connect to MongoDB, the URL must be included in the 'mongoose.connect' method to link the MongoDB API with the application. The username and password fields must be filled in correctly to identify the URL endpoint. 'InvoiceData' is the name of the database which contains collections within it.

```
// connect to DB
const url = 'mongodb+srv://{username}:{password}@testmongo.wxnjfzh.mongodb.net/InvoiceData'
```

The next process of the MongoDB connection is checking the completeness of the result. The ‘mongoose.connect’ is called with a URL parameter that specifies the address of the database to connect to. The useNewUrlParser tells Mongoose to use the new URL parser instead of the default parser. The useUnifiedTopology enables the use of the new Server Discovery and Monitoring engine. If the connection is successful, the then() method is called with a successful result with the message "Connected to

MongoDB". If the connection fails, the catch() method is called with an error(err) result with the message "Failed to connect to MongoDB:"

```
// database connection
mongoose.connect(url, {
   useNewUrlParser: true,
   useUnifiedTopology: true
 })
   .then(result => {
     console.log("Connected to MongoDB");

   })
   .catch(err => {
     console.log("Failed to connect to MongoDB:", err);
   });
```

2) Send data to MongoDB

2.1) data submitting form structure

HTML provides a method for creating forms that allow users to submit data to a web server. The <form> element in HTML is used to create the form and is coordinated with the 'action' and 'method' attributes. The 'method' attribute uses the POST method to submit data to a URL, and the 'action' attribute is linked with the POST redirection.

The form input type is set to 'text' with specific names, including OrderNumber, ID_Number, first_name, last_name, Address, and mobileNum. These names can be accessed when requested by the body and their respective names. For example, 'orderNumber: req.body.OrderNumber' can be used to request the order number. The last part of the structure is the <Button> element, which is used to invoke the form as a POST method to the server.

```
<form method="post" action="/request-submitted">
        <!-- Order Number -->
        <div class="mb-3">
                <label class="form-label">Order Number</label>
                <input type="text" name="OrderNumber" id="OrderNumber"
                class="form-control" required>
        </div>

        <!-- TaxID -->
        <div class="mb-3">
                <label>TAX ID</label>
```

```
                <input type="text" name="ID_Number" id="ID_Number"
                class="form-control">
        </div>

        <!-- Name -->
        <div class="row mb-3">
                <!-- First name -->
                <div class="col">
                        <label>First name</label>
                        <input type="text" name="first_name" id="first_name"
                        class="form-control" aria-label="First name">
                </div>

                <!-- Last name -->
                <div class="col">
                        <label>Last name</label>
                        <input type="text" name="last_name" id="last_name"
                        class="form-control" aria-label="Last name">
                </div>
        </div>

        <!-- Address -->
        <div class="mb-3">
                <label>Address</label>
                <input type="text" name="Address" id="address"
                class="form-control">
        </div>

        <!-- Mobilenumber -->
        <div class="mb-3 ">
                <label>Mobile number</label>
                <input type="text" name="mobileNum" id="mobileNum"
                class="form-control">
        </div>

        <!-- Submit button -->
        <div class="d-grid col-3 mx-auto">
                <button class="btn btn-danger" style="margin-top:
                1rem;">Submit</button>
        </div>
</form>
```

2.2) data schema

After connecting to the database, define schemas for data and create models based on those schemas created in JSON form and mongoose.model() methods, respectively.

The CustomerDataSchema is defined in JSON format. Since the data received from the user is a string attribute, the variables in the JSON schema need to be specified as strings. If the attribute type is set as a number, but the user sends data as a string, the received data will be null.

```
const CustomerDataSchema = {
  orderNumber: String,
  firstname: String,
  lastname: String,
  address: String,
  taxID: String,
  mobileNum: String,
  invoiceDownloaded: Boolean // (pdf generator process)
}
// Create schema model to input to db
const CustomerData = mongoose.model("Customerdatas", CustomerDataSchema)
```

When a user submits a form to '/request-submitted' using the HTTP POST method, the server receives the form data in the req.body object. The 'NewCustomerData.save()' of code saves the new CustomerData instance to MongoDB using the save() method. This adds a new document to the Customerdatas collection in the database with the field values from the form data.

```
app.post('/request-submitted', (req,res) => {
//post customers data in to mongodb by request the OrderNumber of submited form
    orderNumber = req.body.OrderNumber;
    let NewCustomerData = new CustomerData({
      orderNumber: req.body.OrderNumber,
      firstname: req.body.first_name,
      lastname: req.body.last_name,
      address: req.body.Address,
      taxID: req.body.ID_Number,
      mobileNum: req.body.mobileNum,
    });
          NewCustomerData.save();
});
```

2.2.1) Transaction data schema (LINE Shopping)


```
{
 customerData: {
  orderNumber: '2023010441193307',
  firstName: 'John',
  lastName: 'Doe',
  address: 'Sathorn, Bangkok, Thailand',
  taxID: '1234567890123',
  mobileNum: '0909530385'
 },
 orderData: [
  {
   prodName: 'File box 01',
   prodId: 'File01',
   price: '95.00',
   quantity: 1,
   shipFee: 0,
   discount: 0,
   linePoints: 0,
   lspDiscount: 0,
   total: 95,
   totalFloat: '95.00'
  },
  {
   prodName: 'File box 02',
   prodId: 'File02',
   price: '95.00',
   quantity: 3,
   shipFee: 0,
   discount: 0,
   linePoints: 0,
   lspDiscount: 0,
   total: 285,
   totalFloat: '285.00'
  },
  {
   prodName: 'Spaghetti poster (A5)',
   prodId: 'Pos',
   price: '50.00',
   quantity: 1,
   shipFee: 0,
   discount: 0,
   linePoints: 0,
   lspDiscount: 0,
   total: 50,
```

```
    totalFloat: '50.00'
   },
   {
    prodName: 'Mini Postcard',
    prodId: 'BH08',
    price: '50.00',
    quantity: 1,
    shipFee: 45,
    discount: 0,
    linePoints: 0,
    lspDiscount: 0,
    total: 50,
    totalFloat: '50.00'
   }
 ],
 orderNumber: '20230104411193307',
 subtotal: '525.00',
 discount: '0.00',
 shipFee: '45.00',
 vat: '36.75',
 gtotal: '443.25'
}
```

2.2.2) Find the specific order number from customer given

Two Mongoose models, CustomerDatas and Transactions, can be defined to represent collections in a MongoDB database. The CustomerDatas model has fields for _id (a unique identifier for the document), orderNumber, first_name, last_name, Address, ID_Number, and mobileNum. The Transactions model has fields for _id, No, orderNumber, shipFee, productPrice, productNumber, productName, productQuantity, discountBill, linePoints, discount, lspDiscount, totalPrice, and grandTotalPrice.

The method CustomerDatas.find({orderNumber: orderNumber}) is used to find documents in both the CustomerDatas and Transactions collections that match the order number entered by the user in the form.

```
// Define collection customerdatas model
const CustomerDatas = mongoose.model('customerdatas', {
    _id: Number,
    orderNumber:String,
    firstname: String,
    lastname: String,
    address: String,
    taxID: String,
    mobileNum: String
})
```

CustomerDatas model

```
// Define collection transactions model
const Transactions = mongoose.model('transactions', {
    _id: Number,
    No: Number,
    orderNumber: String,
    shipFee: Number,
    productPrice: Number,
    productNumber: String,
    productName: String,
    productQuantity: Number,
    discountBill: String,
    linePoints: Number,
    discount: Number,
    lspDiscount: Number,
    totalPrice: Number,
    grandTotalPrice: Number
  });
```

Transactions model

The query that retrieves data from the condition and merges significant information required for the e-tax invoice document is obtained. The results from both queries are merged using the spread operator '...' and assigned to the variable 'dataMerged'. The resulting 'dataMerged' array contains all documents from both collections that have a matching orderNumber.

```
// find the specific order number from Ordernumber by user given
CustomerDatas.find({orderNumber: orderNumber}, (err, customerDatas) => {
    if (err) throw err;

// find the specific order number from Transactions by user given
Transactions.find({orderNumber: orderNumber}, (err, transactions) => {
    if (err) throw err;

// Merge the results and change data to JSON as "dataResult"
dataMerged = [...customerDatas, ...transactions];
```

Merge customer data and transaction

2.3) End-of-Bill calculation

The calculation process for generating the end-of-bill amount for a tax invoice including the total price before tax, value-added tax (VAT), shipping fee and grand total. Within the function, a loop iterates through the orderData array to calculate the

subtotal and shipping fees for each item in the order since some of the orders consist of several orders. The variables and function are explained as the following code snippet.

```
let subTotal = 0
let shipFee = 0
let vat = 0
let grandTotal = 0
let totalBeforeTax = 0
let discount = 0
let linePoints = 0
let lspDiscount = 0
let sumDiscount = 0

for (let j = 0; j < orderData.length; j++){
        subTotal += orderData[j].total
        shipFee += orderData[j].shipFee
}

// Total discount
sumDiscount = discount+linePoints+lspDiscount

// Total price of every product plus shipping fee
totalBeforeTax = (subTotal + shipFee) - sumDiscount

// VAT 7% that was already include in product price
vat = (totalBeforeTax * 7) / 100

// Actual total price of product after VAT
 grandTotal = subTotal - vat
 console.log('Calculate value-at-the-end')
```

2.4) Download pdf file to browser

The final step in the e-Tax Invoice generation process involves creating a PDF tax invoice and sending it as a response back to the customer. In the context of this research, the PDF tax invoice is delivered to the customer via the browser. When the customer clicks on the "Download Tax Invoice" option on the user interface, a request is sent to the server. Upon receiving the request, the server utilizes the provided data to generate the PDF tax invoice. The invoice is then dynamically streamed as a response to the customer's browser. This allows the customer to conveniently download the PDF file containing their tax invoice.

```
pdf.create(invoiceData).toStream((err, stream) => {
  if (err) {
    console.log(err);
    return res.sendStatus(500);
  }
```

```
  res.setHeader('Content-Type', 'application/pdf')
  res.setHeader('Content-Disposition', `attachment; filename=TAXINV_${finalOrder.orderNumber}.pdf`)
  stream.pipe(res);
})
console.log(`TAXINV_${finalOrder.orderNumber} is downloaded`)
```

3.2.3.5 Database

MongoDB is a popular NoSQL database that offers high scalability, availability, and performance. The Smart E-tax Invoice Generator project is designed to utilize MongoDB for storing customer data and transaction details in its database. In order to use MongoDB Atlas, users can create accounts with a diverse of authentication methods, including email/password, Google, or GitHub. Moreover, MongoDB Atlas also offers users a free tier that provides up to 512MB of storage and a cluster with three nodes. The MongoDB Atlas interface is a web-based user interface that facilitates management of the cluster, database, and collections for users. To utilize MongoDB Atlas, a cluster must be created, after which the user will have access to the cluster dashboard.

1) Process of creating database on MongoDB

1.1) Create a database cluster/project name

Before creating a collection and database, it is important to consider the database cluster. When deploying an application, selecting the appropriate cloud provider region is critical as it can impact several performance aspects. For instance, network latency, which is the distance between the database and the application, can affect the performance of data retrieval and updates. Therefore, opting for a region that is closer to users can reduce latency and enhance the user experience.

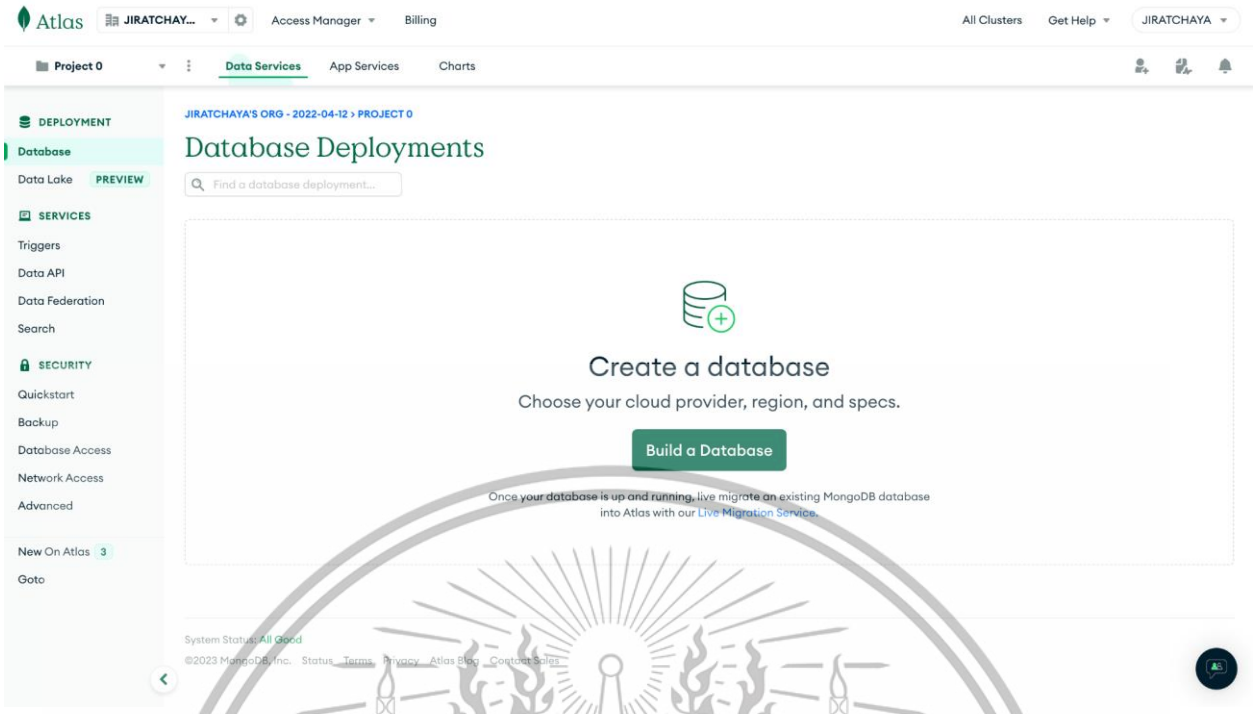

Figure 43 Build a Cluster/Database

MongoDB provides up to 512MB of free storage for a user to learn and explore the application, Additionally, MongoDB is available for various cloud providers such as AWS, Google Cloud, and AZURE that can be chosen for the best of developer needs.

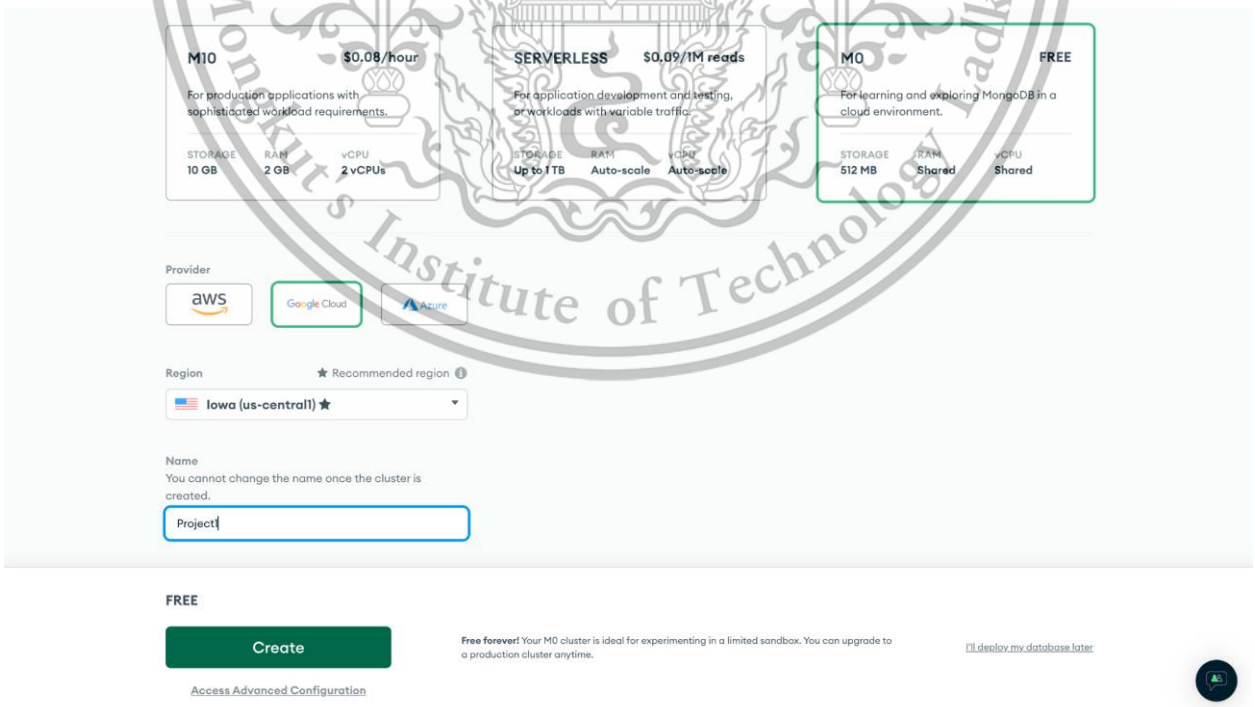

Figure 44 Select the appropriate cloud provider

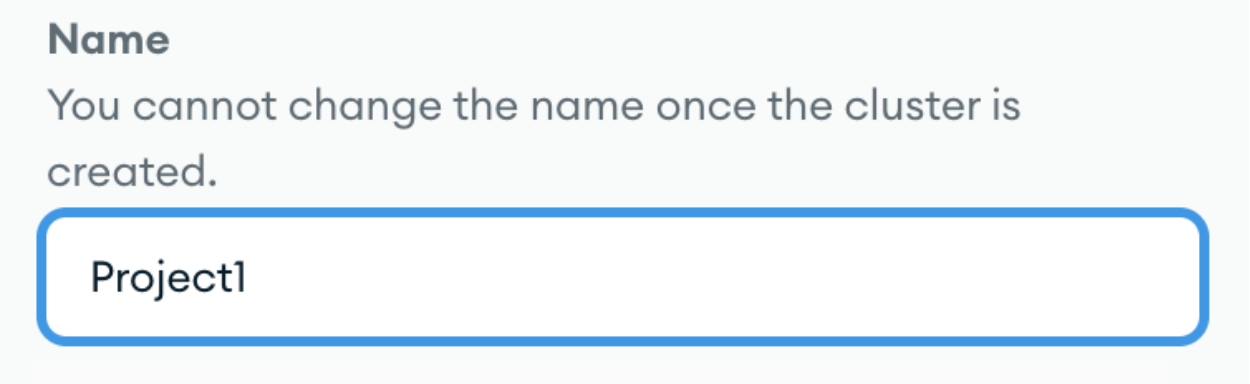

Figure 45 Set the cluster/project name

To authenticate the connection with a user who has permission to read and write any data in the project, the permission can be edited or updated as needed. Importantly, the username used in the MongoDB connection within the application must be properly named and easily readable.

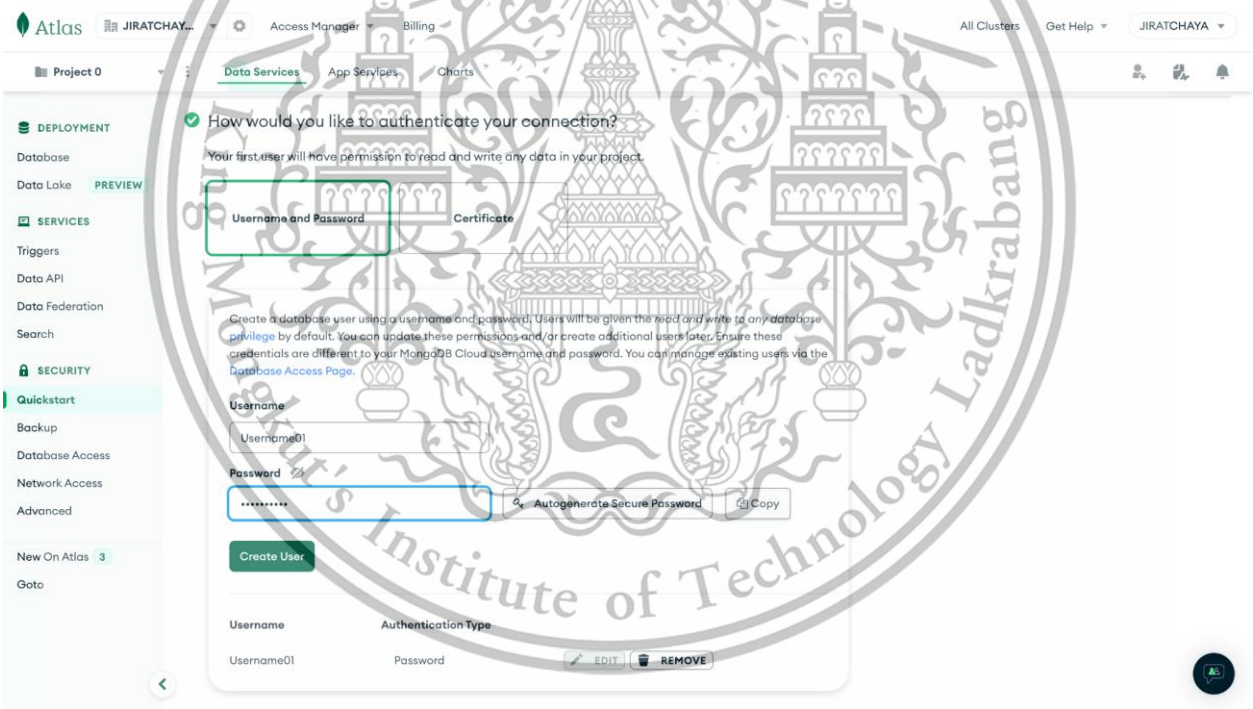

Figure 46 Set the authentication

The list of IP addresses determines which computers can connect to the project's cluster. Other IP addresses can add to the list from other computers as well. To add the current IP address of your computer, click on 'Add My Current IP Address', and MongoDB will automatically detect it. Otherwise, setting 0.0 0.0/0 to acceptable connections on any IP address assigned to the machine. In other words, the server will listen for

incoming connections on all available network interfaces, including both IPv4 and IPv6 addresses. This is useful when the server needs to be accessible from multiple networks.

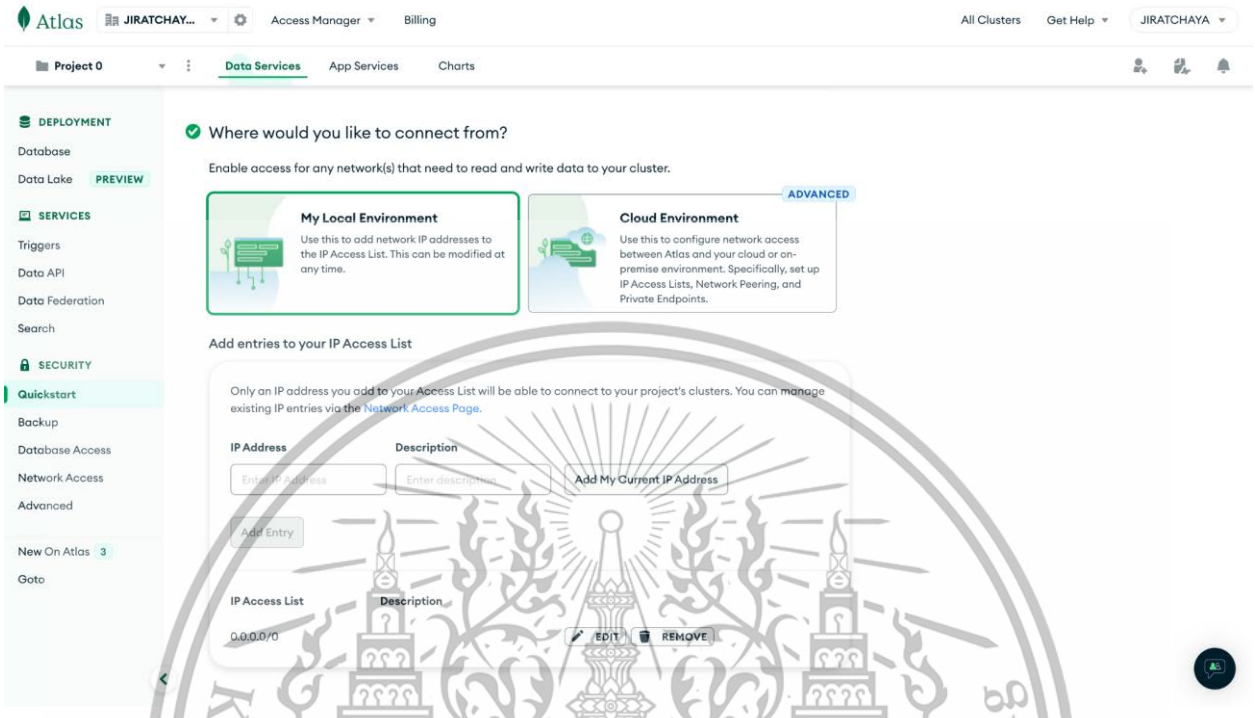

Figure 47 IP addresses acceptable connections

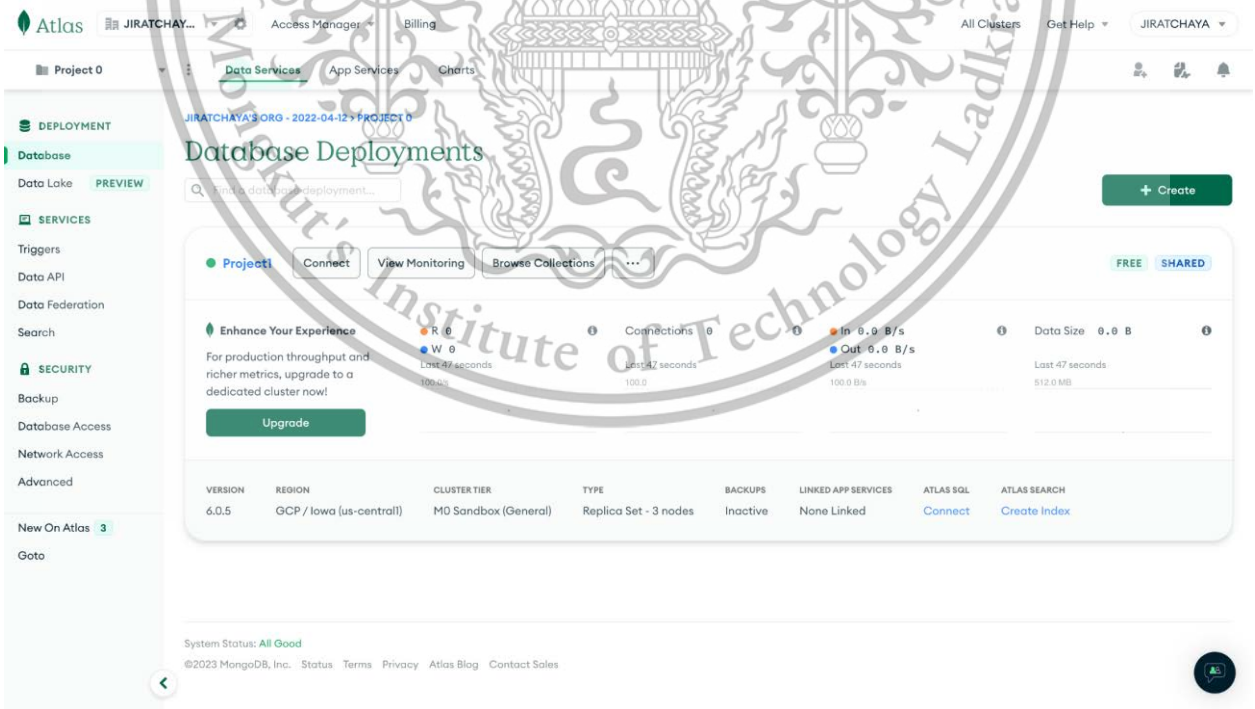

Figure 48 Cluster/project is created

1.2) Create database and collections inside the project

Clicking “Add My Own Data" is a method to import data into their MongoDB database and store it in a collection within the database. To store data, users must first create a database and a collection. Collections are used to manage data within a database, and users can create multiple collections within a single database. Additionally, a single MongoDB cluster can contain multiple databases with its own collections and data.

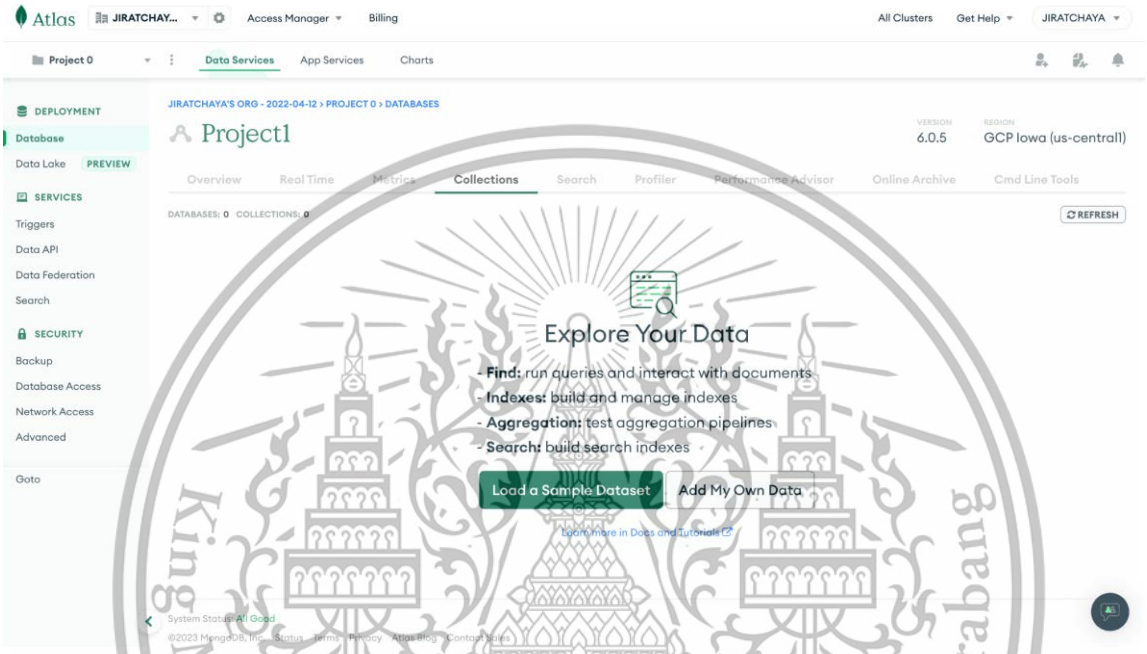

Figure 49 Before create collection of the project

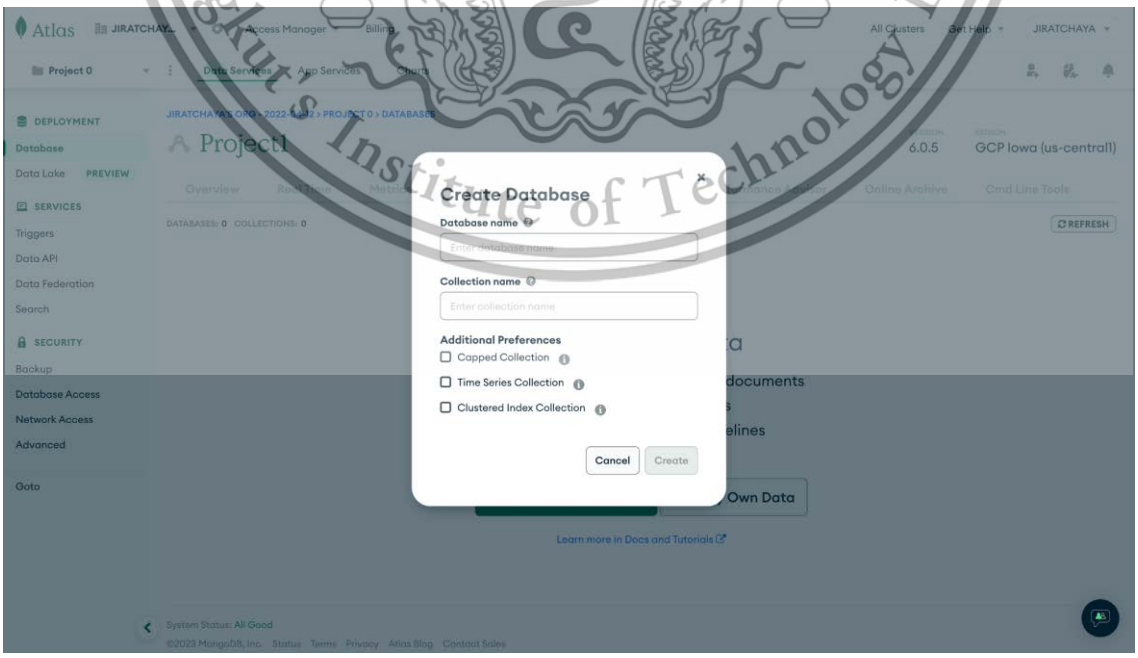

Figure 50 Create a database and a collection

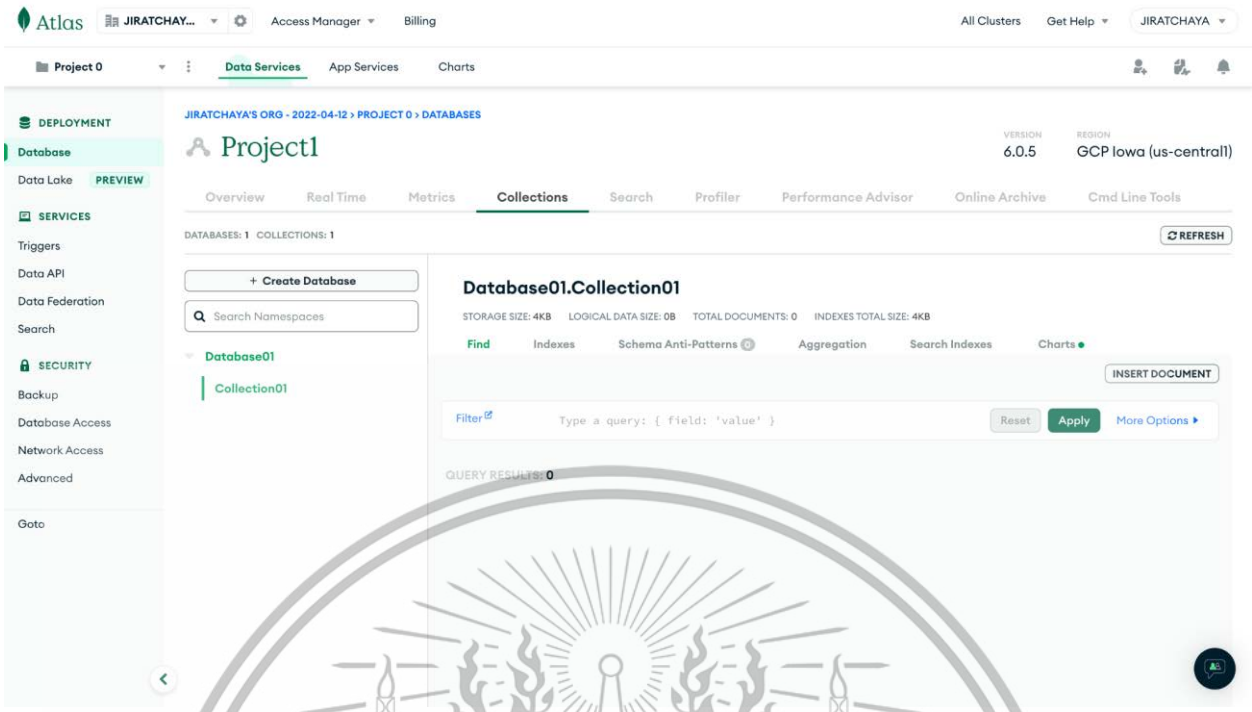

Figure 51 After create database and collection

The data format used for inserting documents into MongoDB is JSON. There are several methods for inserting the documents into a MongoDB database.

Firstly, directly update data using a web interface. This can be done by clicking the "insert document" button and using the JSON data format to input the document. This method is easy and quick insertion of documents without additional programming or software. However, it may not be the most efficient method for large amounts of data or automated insertion of documents.

```json
{
    "invoiceNumber": "INV-001",
    "orderNumber": "0123456789",
    "purchaseDate": "2023-04-01",
    "customerName": "John Doe",
    "customerTaxID": "1234567890123",
    "itemList": [
      {
        "itemNo": 1,
        "itemName": "Product A",
        "itemQuantity": 5,
        "itemPrice": 10
      },
      {
        "itemNo": 2,
        "itemName": "Product B",
        "itemQuantity": 2,
        "itemPrice": 20
      }
    ],
    "discount": 10,
    "total": 100
}
```


Figure 52 Sample of JSON Data format

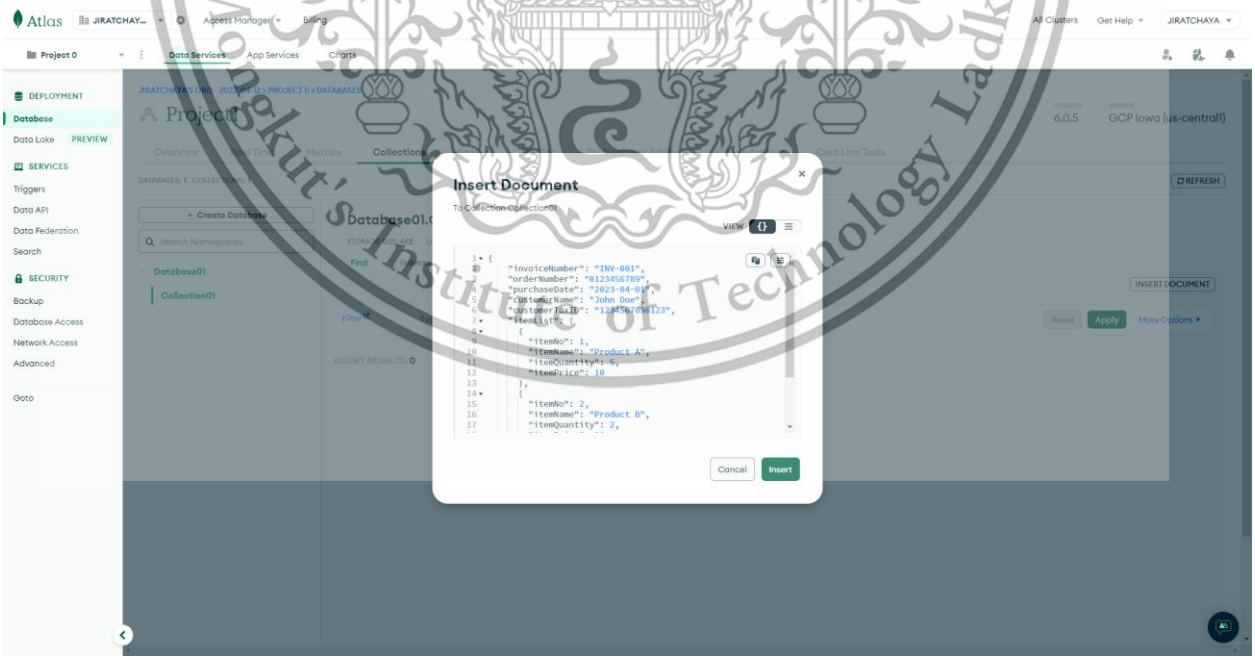

Figure 53 Insert document via web interface

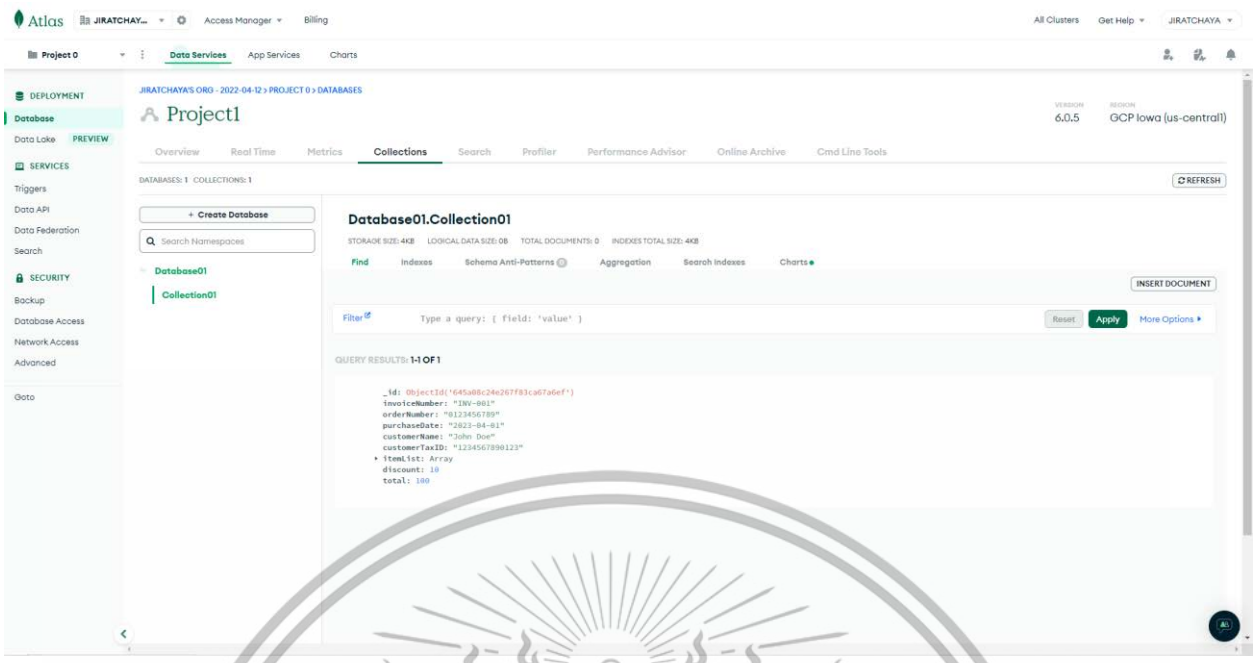

Figure 54 Finished update/upload data to database

Secondly, data can be updated programmatically using languages such as Node.js, Python, or Java by establishing a connection with the MongoDB database using a connection URL.

```
// Connect to the MongoDB database
const url = 'mongodb+srv://<username>:<password>@testmongo.*******.mongodb.net/InvoiceData'

// Define the schema for the collection
const userSchema = new mongoose.Schema({
  name: String,
  age: Number,
  email: String
});

// Define the model for the collection
const User = mongoose.model('User', userSchema);

// Create a new document using the model's create() method
const newUser = {
  name: 'John Doe',
  age: 35,
  email: 'johndoe@example.com'
};
User.create(newUser, (err, result) => {
  if (err) throw err;
  console.log(`Inserted new document with ID ${result._id}`);
```

```
  mongoose.connection.close();
});
```

The example code to update data on MongoDB using node.js

2) Smart E-Tax invoice generator project with MongoDB

2.1) Customerdatas collection

Collections named "Customerdatas" have been created to store essential customer data required to complete the E-tax invoice. This data includes the name, address, and tax ID of the purchaser. Phone number is an optional field that can also be included, while the order number is required for transaction tracking.

In order to update the "customerdatas" collection, it is necessary for purchasers to submit their data via the webpage to MongoDB. The fundamental requirement for requesting transaction details for the E-tax invoice is the order number, which must be provided within the form submission.

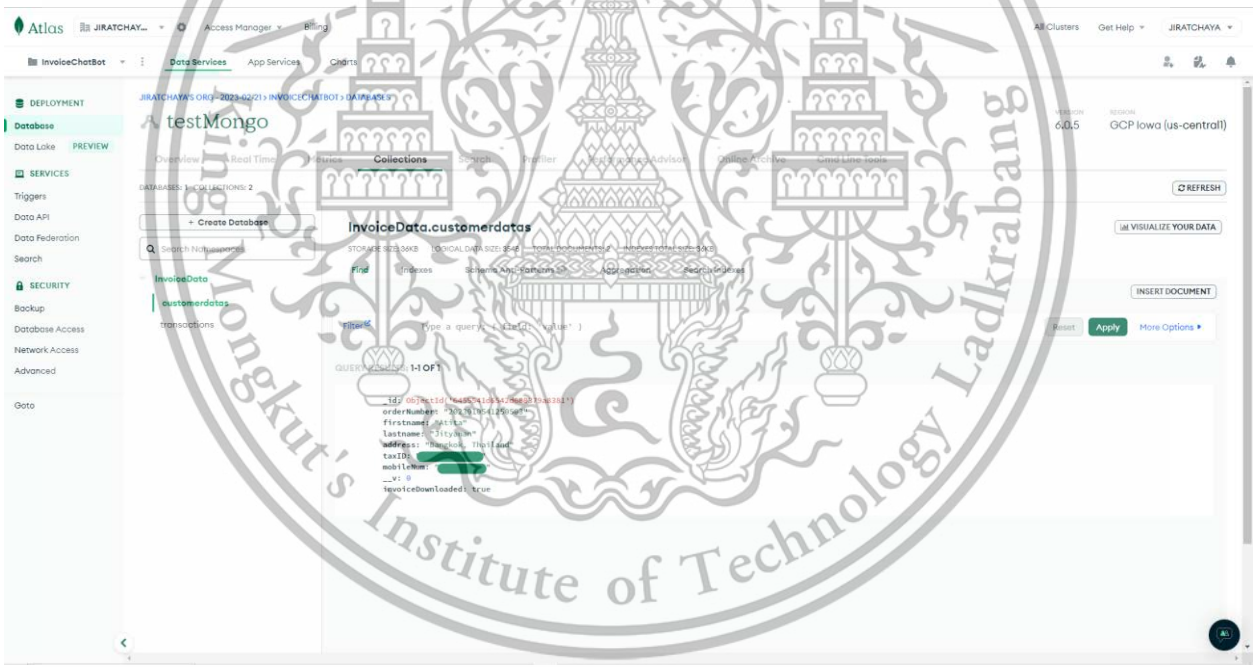

Figure 55 Collection “customerdatas” updated from purchaser

```
// connection URL
const url = 'mongodb+srv://<username>:<password>@testmongo.*******.mongodb.net/InvoiceData'

// database connection
mongoose.set("strictQuery", false);
mongoose.connect(url, {
   useNewUrlParser: true,
   useUnifiedTopology: true
 })
```

```
// request data from purchaser input
app.post('/request-submitted', (req,res) => {
   orderNumber = req.body.OrderNumber;
   let NewCustomerData = new CustomerData({
     orderNumber: req.body.OrderNumber,
     firstname: req.body.first_name,
     lastname: req.body.last_name,
     address: req.body.Address,
     taxID: req.body.ID_Number
  });
```

Update data in MongoDB using post method

```
<input type="text" name="OrderNumber" id="OrderNumber" class="form-control" required>

<input type="text" name="ID_Number" id="ID_Number" class="form-control">

<input type="text" name="first_name" id="first_name" class="form-control" aria-label="First name">

<input type="text" name="last_name" id="last_name" class="form-control" aria-label="Last name">

<input type="text" name="Address" id="address" class="form-control">
```

HTML form structure

2.2) Transactions collection

Collections named "Transactions" have been created to store essential data related to LINE store transactions, each of which is uniquely identified by an order number. To update the details of these transactions, LINE Shopping provides the required order information, which is subsequently used to add a document in JSON format to MongoDB for database updates.

The store can access a comprehensive summary of sales data by specifying the desired date range for the report. This sales report can be downloaded in Excel format. The report includes detailed information on orders, including those paid via bank transfer, paid on delivery, or via Rabbit LINE Pay, as well as all products currently in stock and all available discount codes. The store can utilize this information for further analysis and decision-making purposes.

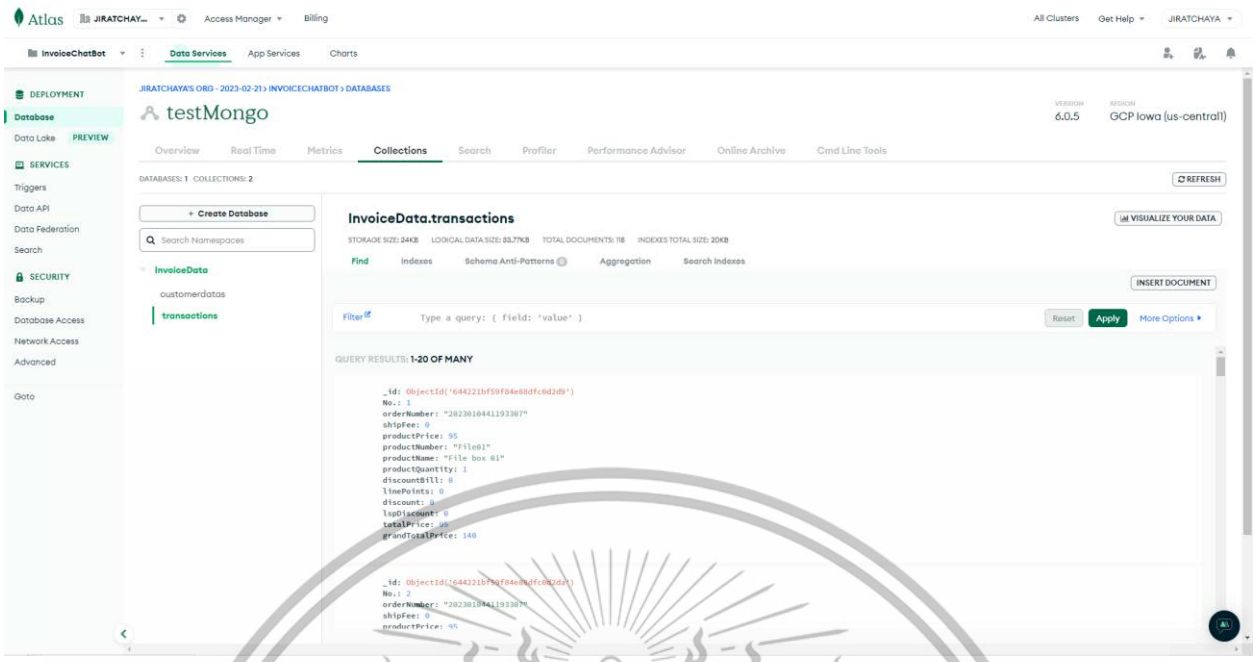

Figure 56 Collection “transactions” updated from store

3.2.3.6 Hosting

Hosting is a service type that provides for presenting data online. When someone wants to publish data online, allow others to view data, download it, or perform other actions. Creating a website is the priority. Then, after the website is created, it can be uploaded to a hosting service that provides the space for the website to be stored and accessed online. Dynamic DNS is a technique that enables a domain name to be linked to an IP address, regardless of whether it is static or dynamic. It updates the IP address automatically whenever it changes, guaranteeing that the domain stays connected to the correct server. The server is typically set up at home and hosts the website on a computer. Using the DDNS service from TRUE DDNS requires following these steps.

1) Register the TRUE DDNS service

Visiting the https://trueddns.com/login browser the register the service

Figure 57 TRUE DDNS registration page

2) The system will check the user's privileges to verify whether they can use the TRUE DDNS system.

Figure 58 Confirm the user's privileges

3) Insert information for a new register

The system will automatically enter the Internet number and select the customer type, usually an individual. Then, you should enter your national ID number as registered for the DDNS service activation.

Figure 59 New register page

4) Enter the registration information

Figure 60 The registration page for entering your information

5) Confirm registration in the Email

The confirmation of registration is important to complete within 24 hours after the completion of the registration process. The system will verify the information provided during registration and send a confirmation email from the address trueddns@trueddns.com.

Figure 61 Check Email confirmation

6) Sign in to TRUE DDNS after registering

After we have successfully verified the registration, the system will take around 5 seconds to create a TRUE DDNS user account and display a login page for accessing the system. Then, re-login with the login credentials that were registered before.

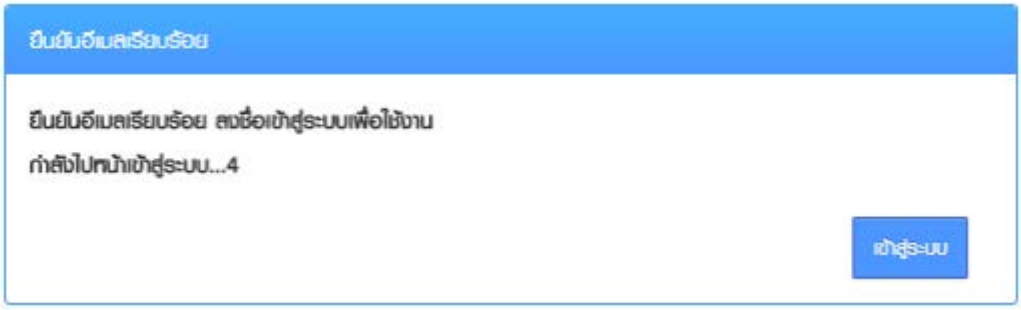

Figure 62 Sign in

Figure 63 Log in to access the service

7) Domain name setting

A domain name refers to a website's address on the internet which users type in the browser's search to visit the website directly. Each domain name is unique and cannot be shared between other websites. Moreover, once a domain name is set up, it cannot be changed. So, it's better to create a domain name that is easy to remember.

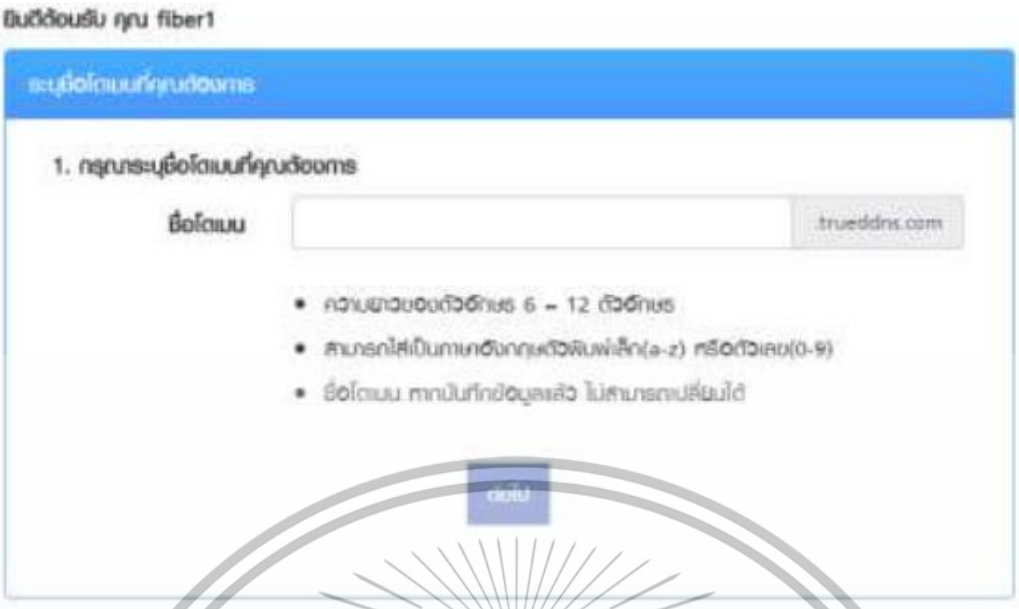

Figure 64 Set domain name

8) Port setting

The TRUE DDNS system allows users to add or remove up to 10 ports. Once the user has identified the desired number of ports, the system will display the previously entered information for the user to review and confirm before creating a new account.

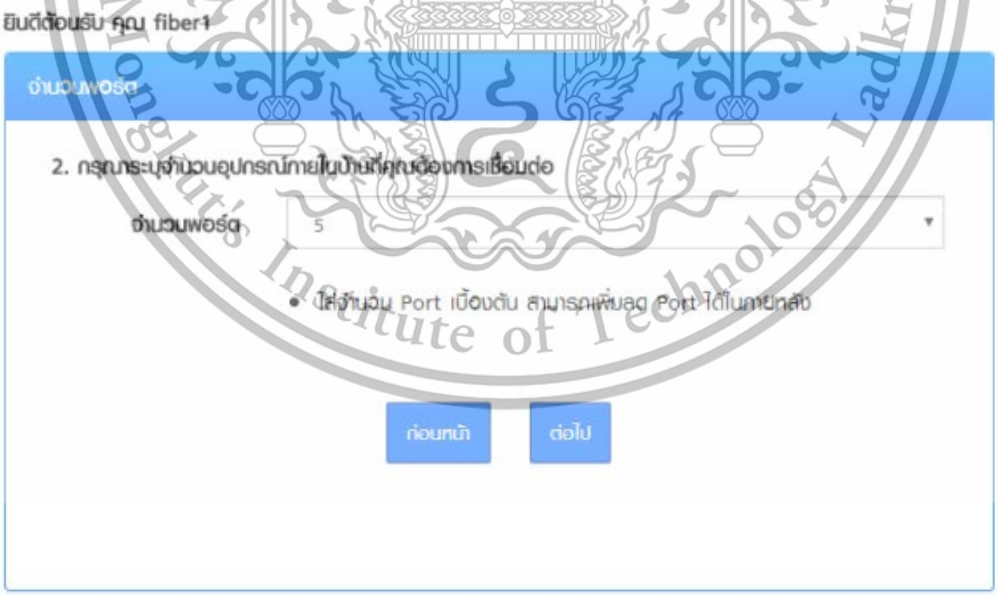

Figure 65 Port selection

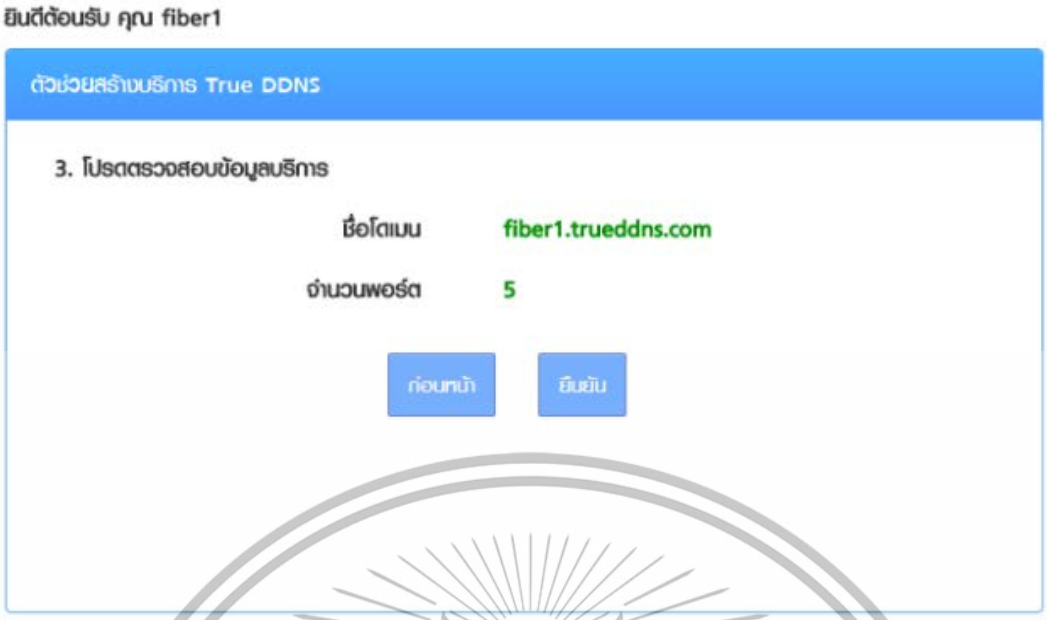

Figure 66 Recheck information

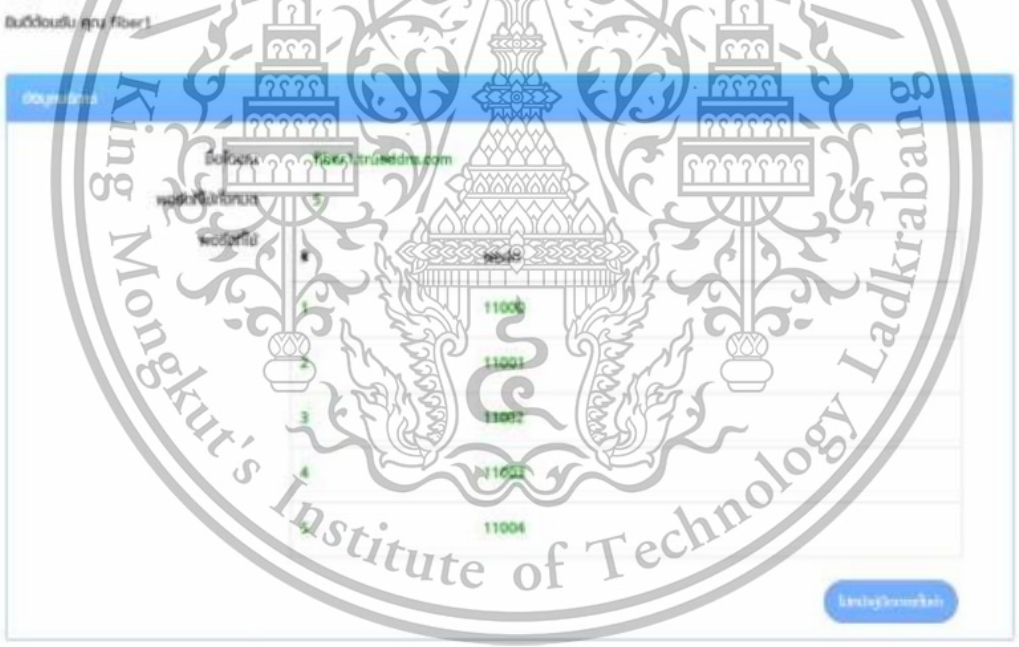

Figure 67 Port that can forward with router

9) Visit link http://192.168.1.1/ to set the internal host

Username is on the internet router or look at the manual/box or back of router

Password is the key below (34*****M)

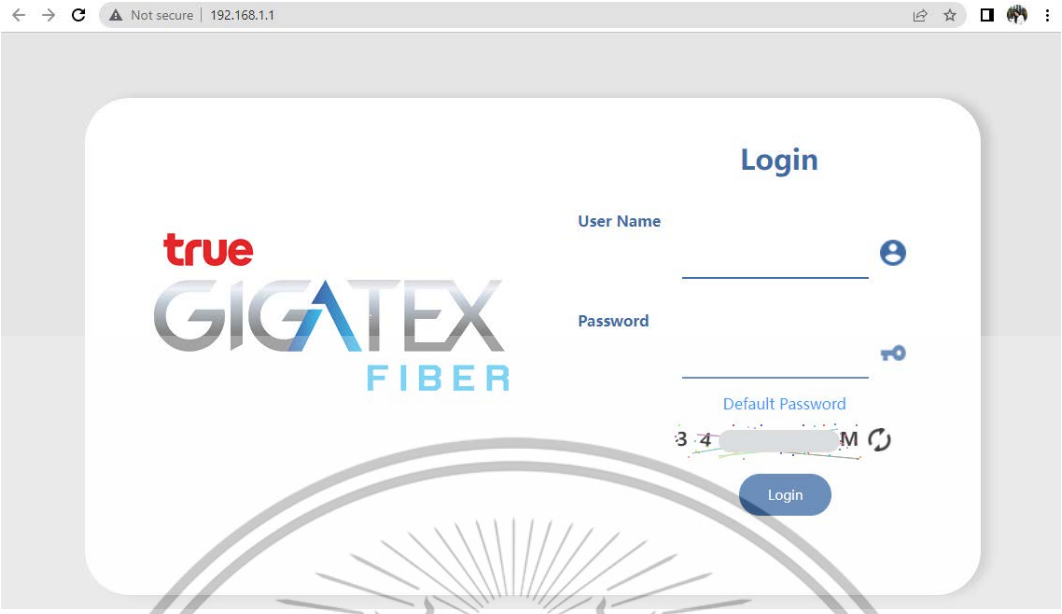

Figure 68 link http://192.168.1.1/

10) Select IPv4 Mapping

To create a new mapping project, first choose a project name and enter the internal host name by IP address from your own computer. You can obtain your computer's IPv4 address by running the following command in the Command Prompt: 'ipconfig'. IPv4 address will be the internal host.

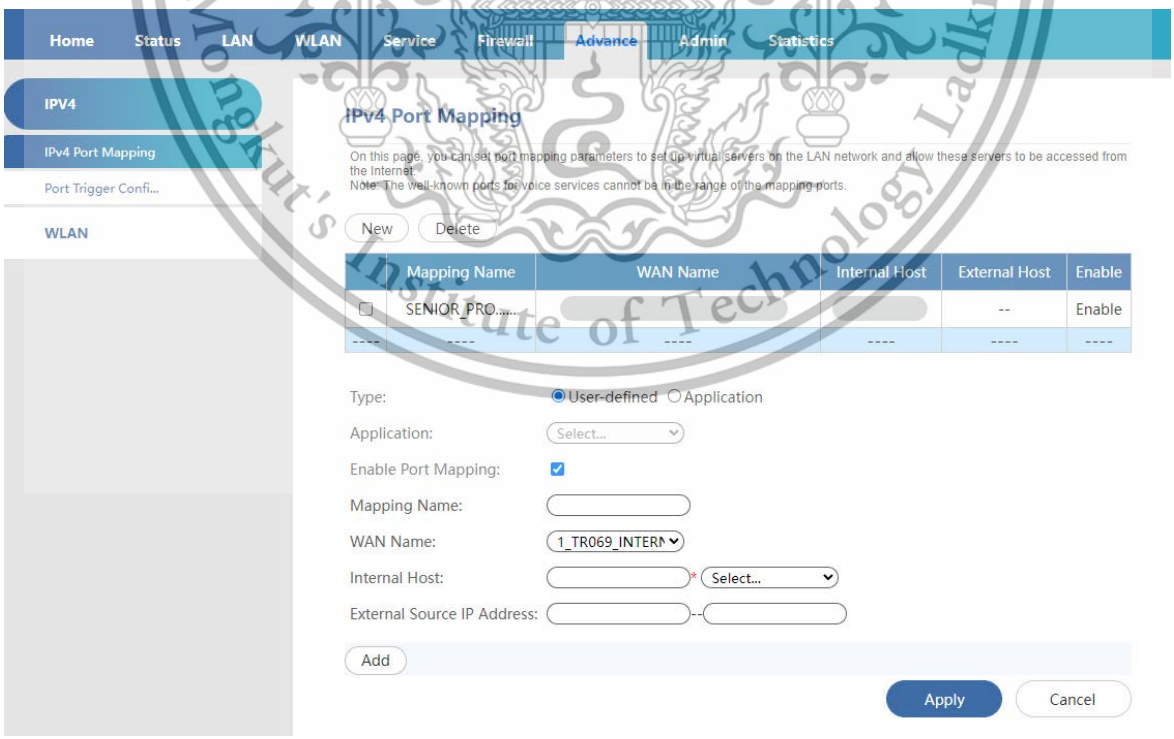

Figure 69 Create new IPv4 mapping

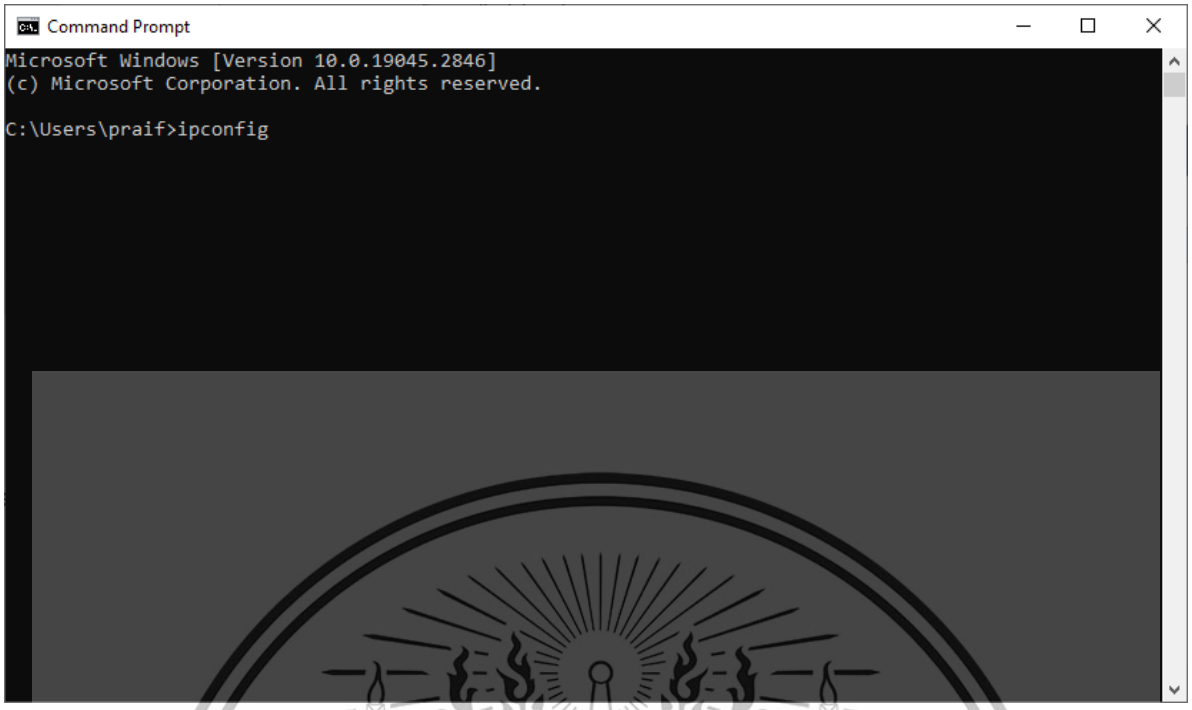

Figure 70 Navigate to the Ip address with command line

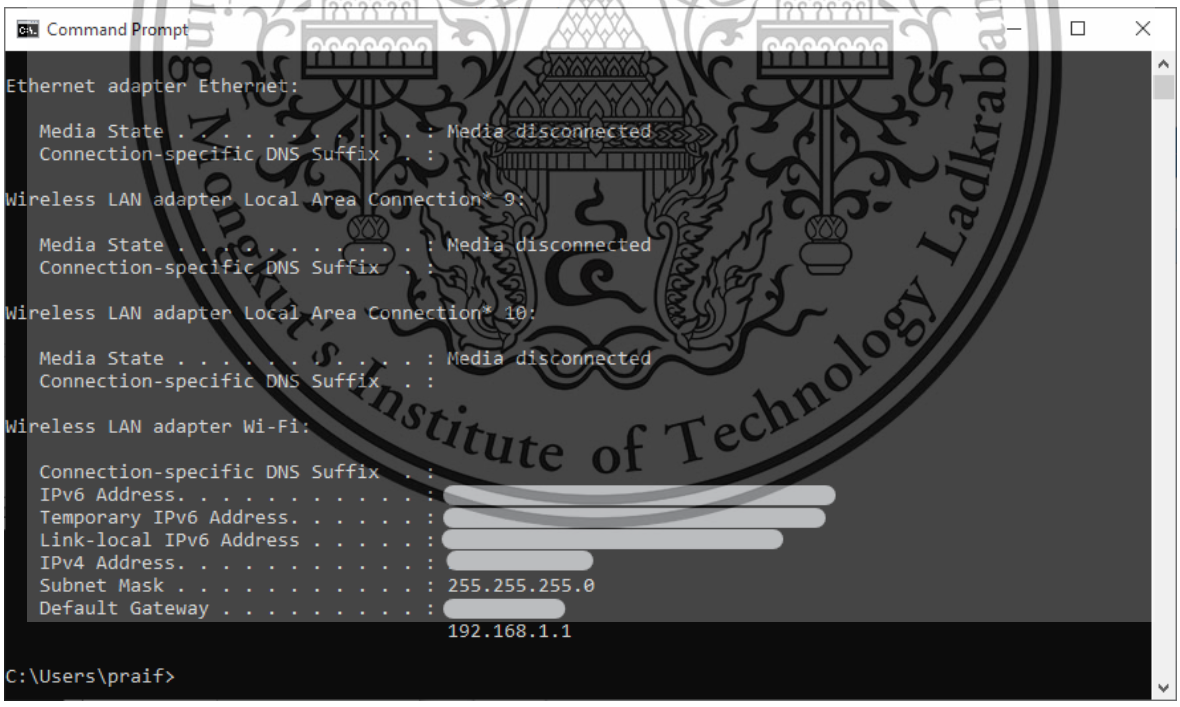

Figure 71 Get the IPv4 address

11)Add the internal port number and external port number

An internal port number refers to a specified port on the router to which incoming requests from the internet will be forwarded to a device with the same port number. An

external port number is assigned by TRUE DDNS to a created port and is used to route traffic from the internet to the internal port on the device.

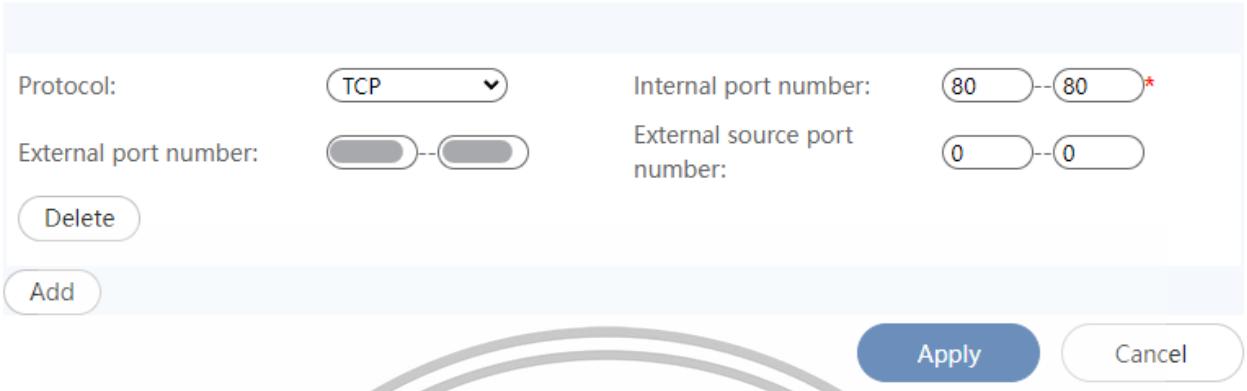

Figure 72 Add the internal port number and external port number

12)Set in backend server

The project will run on a domain name that has been created using the TRUE DDNS service.

```
// Define server port
const port = process.env.PORT || 80;

// Run app on DDNS domail name
app.listen(port,'0.0.0.0')
```

Server code setting

Chapter 4
Research Result

4.1 Web application

4.1.1 Web application first iteration

The Smart E-tax Invoice generator project deploys a web application with three versions. The first version is a prototype testing version without any interface decoration. It is used to test the connection between the MongoDB database and the frontend user. During testing, it was discovered that users needed to input the order number to match the transaction details. However, this version does not include any data security measures, such as encryption and private keys in the e-tax invoice file. Additionally, it does not have the functionality to generate PDF files.

Figure 73 Smart e-tax invoice generator first iteration

4.1.2 Web application second iteration

The second version incorporates interface decoration based on the store's concept design. It takes into account the feedback and errors identified in the first version. This version includes an order number filling form where users can enter their order number to ensure accurate retrieval of transaction details. Despite the design changes, the connection between the MongoDB database and user input remains functional. Version 2 version also includes the functionality to generate PDF files. However, it is important to note that this version, like the previous one, does not implement data security measures such as encryption and private keys.

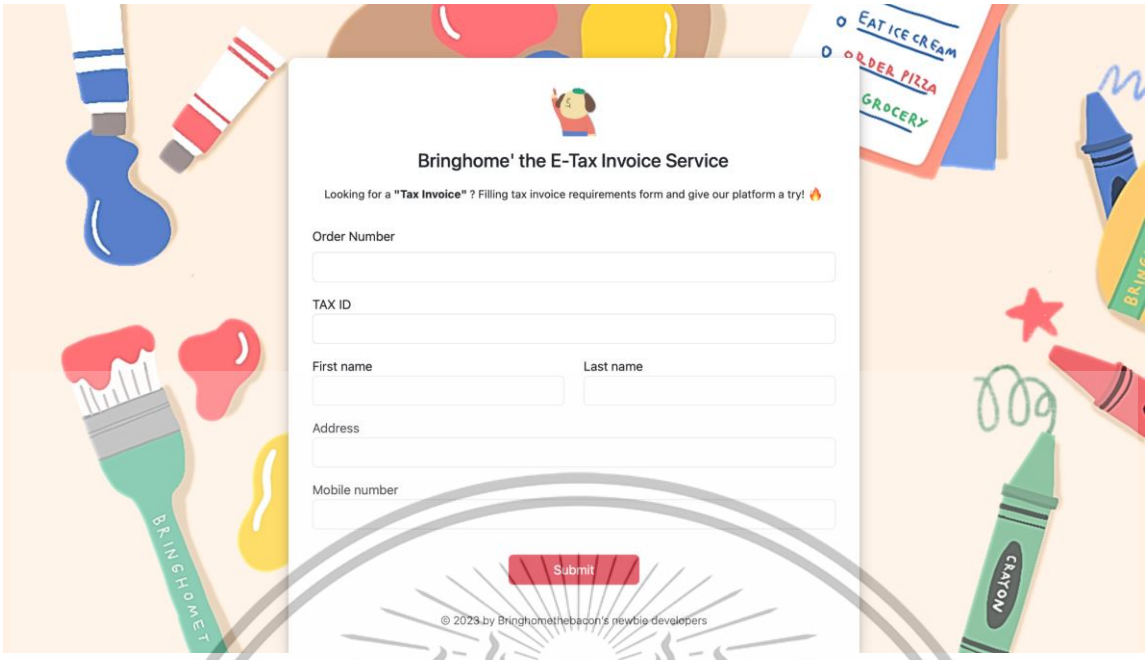

Figure 74 Smart e-tax invoice generator second iteration

4.1.3) Web application final iteration

In the final version of the Smart E-tax Invoice generator project, the order number filling form has a condition that requires users to fill it out. It is a one-time filling and cannot be left blank. As previous versions' limitations regarding data security, the final version enhances security by implementing encryption and electronic signatures in the e-tax invoice file. These measures provide an additional layer of protection to ensure the document's integrity and security

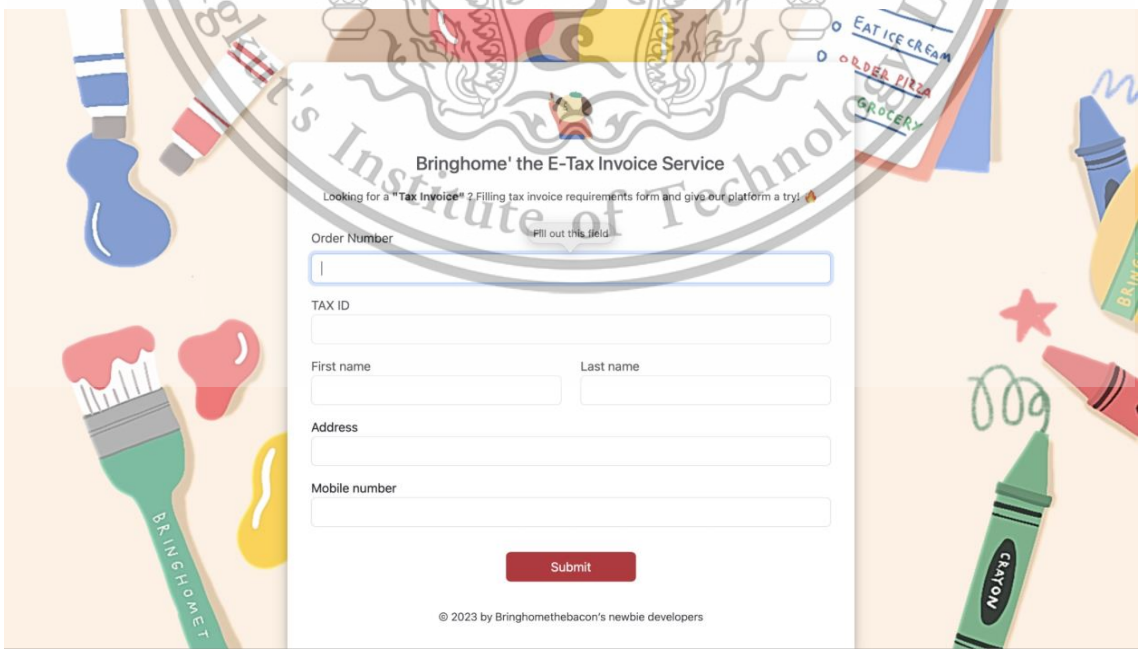

Figure 75 Smart e-tax invoice generator final iteration

The XML format of tax invoice is also created but before sending the response back to the customer while it can be recorded for further tax operation for the company. The following code snippet is an XML tax invoice with digital signature according to regulation of the Revenue Department.

```
<?xml version="1.0"?>
<TAXINV_2023010441193307 Id="_0">
 <InvoiceHeader>
  <Vendor>
   <Name>Bringhome.theBacon</Name>
   <Address>24/56 Bang Na-Trat Rd, Tambon Bang Sao Thong, Amphoe Bang Sao Thong, Chang Wat
Samut Prakan 10540</Address>
   <Phone type="telNumber">0987490837</Phone>
   <TaxId schemaID="TaxID">123456789012</TaxId>
  </Vendor>
  <Customer>
   <CustomerInfo>
    <CustomerName>
     <FirstName>Atita</FirstName>
     <LastName>Jityanan</LastName>
    </CustomerName>
    <Address>55/495 Aspire Sathorn-Taksin (Timber zone) Ratchapruk Rd., Bang ko, Bang ko, BKK,
10150</Address>
    <Phone>0909530385</Phone>
    <TaxId>1104700031805</TaxId>
   </CustomerInfo>
   <OrderHeaderInfo>
    <IssuedDateTime>5/16/2023</IssuedDateTime>
    <TaxInvoiceNumber>0.5581633080508854</TaxInvoiceNumber>
    <OrderNumber>2023010441193307</OrderNumber>
   </OrderHeaderInfo>
  </Customer>
 </InvoiceHeader>
 <OrderDetails>
  <ProductDetails id="1">
   <ProductID>File01</ProductID>
   <ProductName>File box 01</ProductName>
   <UnitPrice>95.00</UnitPrice>
   <Quantity>1</Quantity>
   <TotalPrice>95.00</TotalPrice>
  </ProductDetails>
  <ProductDetails id="2">
   <ProductID>File02</ProductID>
   <ProductName>File box 02</ProductName>
   <UnitPrice>95.00</UnitPrice>
```

```
    <Quantity>3</Quantity>
    <TotalPrice>285.00</TotalPrice>
   </ProductDetails>
   <ProductDetails id="3">
    <ProductID>Pos</ProductID>
    <ProductName>Spaghetti poster (A5)</ProductName>
    <UnitPrice>50.00</UnitPrice>
    <Quantity>1</Quantity>
    <TotalPrice>50.00</TotalPrice>
   </ProductDetails>
   <ProductDetails id="4">
    <ProductID>BH08</ProductID>
    <ProductName>Mini Postcard</ProductName>
    <UnitPrice>50.00</UnitPrice>
    <Quantity>1</Quantity>
    <TotalPrice>50.00</TotalPrice>
   </ProductDetails>
  </OrderDetails>
  <EndofBillCalculation>
   <TotalBeforeVAT>525.00</TotalBeforeVAT>
   <Discount>0.00</Discount>
   <ShipFee>45.00</ShipFee>
   <VAT>36.75</VAT>
   <GrandTotal>443.25</GrandTotal>
  </EndofBillCalculation>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                <Reference URI="#_0">
                <Transforms><Transform
        Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/></Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <DigestValue>GUez7XH+22jGiQFAgi3Z21rvH0I=</DigestValue></Reference>
        </SignedInfo>
        <SignatureValue>vzm5pe5HX6a86OZv7+ny9mBq5DyStRIn+3QcuVyGwcpFy9cAPfdld+eNCoGqR
m3hLbIWj3TXWHWuWCQaCHVHTnSeIPHbDdLkwhMEpLCRjQVDbjF1CqJrU85QwBdJuyBpyT9j3Xu338fHt
3UwWw40tDBmL+h31A1K/p4ddEeaBPkueWSoko2/sehbi+wfF585YV04pjhvwV2izLWMjl6cUkSWyoCzA1
mo5kOAB73vHyee3UKMQaSCtLVrvV1B21j/L0squBk1VjMgNb9KtRRavwXL0U1m5MbGBZ8hFMA67gM89
9SK4LoePgba6AQIuIjE2pSjn0+KOZDpADNCCLlTAw==
        </SignatureValue>
        <KeyInfo>
                <X509Data>
                        <X509Certificate>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUR2RENDD
                QXFRQ0NRQ3FWc3FPNDdRZUJEQU5CZ2txaGtpRzl3MEJBUXNGQURDQm56RUxNQWtH
                QTFVRUJoTUMKZEdneEVEQU9CZ05WQkFnTUlySmhibWRyYjJzeEVEQU9CZ05WQkFjTUI
```

zTmhkR2h2Y200eEd6QVpCZ05WQkFvTQpFbUp5YVc1bmFHOXRaUzUwYUdWaVlXTnZia
kVPTUF3R0ExVUVDd3dGYTIxcGRHd3hHekFaQmdOVkJBTU1FbUp5CmFXNW5hRzl0Wl
M1MGFHVmlZV052YmpFaU1DQUdDU3FHU0liM0RRRUpBUllUWVhScGRHRXVhbWwwU
UdkdFlXbHMKTG1OdmJUQWVGdzB5TXpBMU1EUXdPREUyTXpKYUZ3MHlOREExTURNd
09ERTJNekphTUlHZk1Rc3dDUVlEVlFRRwpFd0owYURFUU1BNEdBMVVFQ0F3SFltRnVa
MnR2YXpFUU1BNEdBMVVFQnd3SGMyRjBhRzl5YmpFYk1Ca0dBMVVFCkNnd1NZbkpwY
m1kb2IyMWxMblJvWldKaFkyOXVNUTR3REFZRFZRUUxEQVZyYldsMGJERWJNQmtHQTF
VRUF3d1MKWW5KcGJtZG9iMjFsTG5Sb1pXSmhZMjl1TVNJd0lBWUpLb1pJaHZjTkFRa0J
GaE5oZEdsMFlTNXFhWFJBWjIxaAphV3d1WTI5dE1JSUJakFOQmdrcWhraUc5dzBCQVF
FRkFBT0NBUThBTUlJQkNnS0NBUUVBMFhBK1hrVytFNG5KCmdzbllWZjgvc2g5MkxaRjlt
SFdQOFZoOHZGQU8rR2dOay9UaHpVNHc0NFgwM3I3REFTLzgzSjFpWE5XV3R2djcKUW
ZySEpnUjF1R0dRUjlFK1JCVFhVNVcrVm4xZCs2OXp4Yk4yb0xlZFRSRENWSjlGUWFPVjVP
dXgyVkxqeUppSApidGNjd0tUYWVsMDNlK0NMc2FMd1VyQW1OQVJjanJaMnFEQ2p6b1k
1azNMYkduTlo1RTY3SFAwVzhrUlN5NS9mCjA5Z0R3djNWVjhIai9lVmZaQkdiTC92RU8xb
XFKR1dXdlBubHRncW5mcy9ZZFJBR3RWQ1hCMDhKNklUdng0SEYKZVRQaUpCSkVheSt
rWmorL3E4NTBUVTRUS2xnbkU3bXVLQ2xBTnQyVlQ0dm5sQmVwckFoTFFPQ0g0UTRvQ
k51dApEdWxWQzd3cmdRSURBUUFCTUEwR0NTcUdTSWIzRFFFQkN3VUFBNElCQVFDZn
BacGpkYm9kWWYrdFdsYk9OZGVmCnEyeU9OWGtIUVVsSlI3RXZQRWhETnpacjdTUzdvdl
ViRCtXeXJ2TVpKTXhkdWkwdFdEQ3U0RTF0bTl6U3krSkMKNkhzbzFOZ0ZLK3lqeGpzSVY
0Mkp1RUdFUkRJaFVnYktRcXV1VkYrR0pQWmZQeGVOS3RXL2xMMldaUzk3b3F4VApOa
3dnQlFwMXBTSVNMVzFIZG9GTGcrWGdDZjlqdkwrQjdTUnJEODVhNHlxWCt0UEVlUWpT
K2dpNElJZjVuaGFkCkVlREx3UHpCTVpYVUNGSnJqbWg1a3ZiTkNmdFlhY0kyTk5LL2g2S
Fg2VnRibVRieXh2MnV5d3hyRmd4N1k5cWcKa3FtQUUyRVpoblFZTW9DTFdJdEJUQVpG
Y3BjWUdpT1MvQmtNR0FUSno5SHBYb3RueHlJRVFudFhSbW9hNjZITQotLS0tLUVORCB
DRVJUSUZJQ0FURS0tLS0tCg==
</X509Certificate>
</X509Data>
</KeyInfo>
<Object>
<xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#"
Target="#xmldsig-69b3d670-04e4-419f-8d45-938ea867f571">
<xades:SignedProperties Id="xmldsig-69b3d670-04e4-419f-8d45-938ea867f571-
signedprops">
<xades:SignedSignatureProperties>
<xades:SigningTime>2023-05-15T18:06:14.866Z</xades:SigningTime>
<xades:SigningCertificate>
<xades:Cert>
<xades:CertDigest>
<DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
/><DigestValue>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUR
2RENDQXFRQ0NRQ3FWc3FPNDdRZUJEQU5CZ2txaGtpRzl3MEJBUXNG
QURDQm56RUxNQWtHQTFVRUJoTUMKZEdneEVEQU9CZ05WQkFnTUI
ySmhibWRyYjJzeEVEQU9CZ05WQkFjTUIzTmhkR2h2Y200eEd6QVpCZ0
5WQkFvTQpFbUp5YVc1bmFHOXRaUzUwYUdWaVlXTnZiakVPTUF3R0E
xVUVDd3dGYTIxcGRHd3hHekFaQmdOVkJBTU1FbUp5CmFXNW5hRzl0

WlM1MGFHVmlZV052YmpFaU1DQUdDU3FHU0liM0RRRUpBUllUWVhS
cGRHRXVhbWwwUUdkdFlXbHMKTG1OdmJUQWVGdzB5TXpBMU1EUX
dPREUyTXpKYUZ3MHlOREExTURNd09ERTJNekphTUlHZk1Rc3dDUVlE
VlFRRwpFd0owYURFUU1BNEdBMVVFQ0F3SFltRnVaMnR2YXpFUU1BN
EdBMVVFQnd3SGMyRjBhRzl5YmpFYk1Ca0dBMVVFCkNnd1NZbkpwYm
1kb2IyMWxMblJvWldKaFkyOXVNUTR3REFZRFZRUUxEQVZyYldsMGJER
WJNQmtHQTFVRUF3d1MKWW5KcGJtZG9iMjFsTG5Sb1pXSmhZMjl1TV
NJd0lBWUpLb1pJaHZjTkFRa0JGaE5oZEdsMFlTNXFhWFJBWjIxaAphV3
d1WTI5dE1JSUJJakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBTUlJQ
kNnS0NBUUVBMFhBK1hrVytFNG5KCmdzbllWZjgvc2g5MkxaRjltSFdQO
FZoOHZGQU8rR2dOay9UaHpVNHc0NFgwM3I3REFTLzgzSjFpWE5XV3R
2djcKUWZySEpnUjF1R0dRUjlFK1JCVFhVNVcrVm4xZCs2OXp4Yk4yb0xl
ZFRSRENWSjlGUWFPVjVPdXgyVkxqeUppSApidGNjd0tUYWVsMDNlK0N
Mc2FMd1VyQW1OQVJjanJaMnFEQ2p6b1k1azNMYkduTlo1RTY3SFAwV
zhrUlN5NS9mCjA5Z0R3djNWVjhJai9lVmZaQkdiTC92RU8xbXFKR1dXdl
BubHRncW5mcy9ZZFJBR3RWQ1hCMDhKNklUdng0SEYKZVRQaUpCSk
VheStrWmorL3E4NTBUVTRUS2xnbkU3bXVLQ2xBTnQyVlQ0dm5sQmV
wckFoTFFPQ0g0UTRvQk51dApEdWxWQzd3cmdRSURBUUFCTUEwR0N
TcUdTSWIzRFFFQkN3VUFBNElCQVFDZnBacGpkYm9kWWYrdFdsYk9QZ
GVmCnEyeU9OWGtlUVVsSll3RXZQRWhETnpacjdTUzdvdlViRCtXeXJ2T
VpKTXhkdWkwdFdEQ3U0RTF0bTl6U3krSkMKNkhzbzFOZ0ZLK3lqeGpz
SVY0Mkp1RUdFUkRJaFVnYktRcXV1VkYrR0pQWmZOeGVOS3RXL2xMM
ldaUzk3b3F4VApOa3dnQlFwMXBTSVNMVzFIZG9GTGcrWGdDZjlqdkwr
OjdTUnJEODVhNHlxWCt0UEVlUWpTK2dpNElJZjVuaGFkCkVlREx3UHp
CTVpYVUNGSnJqbWg1a3ZiTkNmdFlhY0kyTk5LL2g2SFg2VnRibVRieXh
2MnV5d3hyRmd4N1k5cWcKa3FtQUUyRVpoblFZTW9DTFdJdEJUQVpG
Y3BjWUdpT1MvQmtNR0FUSno5SHBYb3RueHlJRVFudFhSbW9hNjZITQ
otLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==</DigestValue>
</xades:CertDigest>
<xades:IssuerSerial>
<X509IssuerName>C=th, ST=bangkok, L=sathorn,
O=bringhome.thebacon, OU=kmitl, CN=bringhome.thebacon,
E=atita.jit@gmail.com</X509IssuerName>
<X509SerialNumber>00aa56ca8ee3b41e04</X509SerialNum
ber>
</xades:IssuerSerial>
</xades:Cert>
</xades:SigningCertificate>
</xades:SignedSignatureProperties>
</xades:SignedProperties>
</xades:QualifyingProperties>
</Object>
</Signature>
</TAXINV_2023010441193307>

4.2 PDF E-tax invoice

4.2.1 PDF E-tax invoice first version

The first version of tax invoice PDF format consisted of all the necessary information for tax invoice and represented all the order transactions matched with customer information input accurately. However, there's room for improvements since some of the details need to be fixed such as the layout of customer information presentation, the position of signature as well as the encryption process that need to be added.

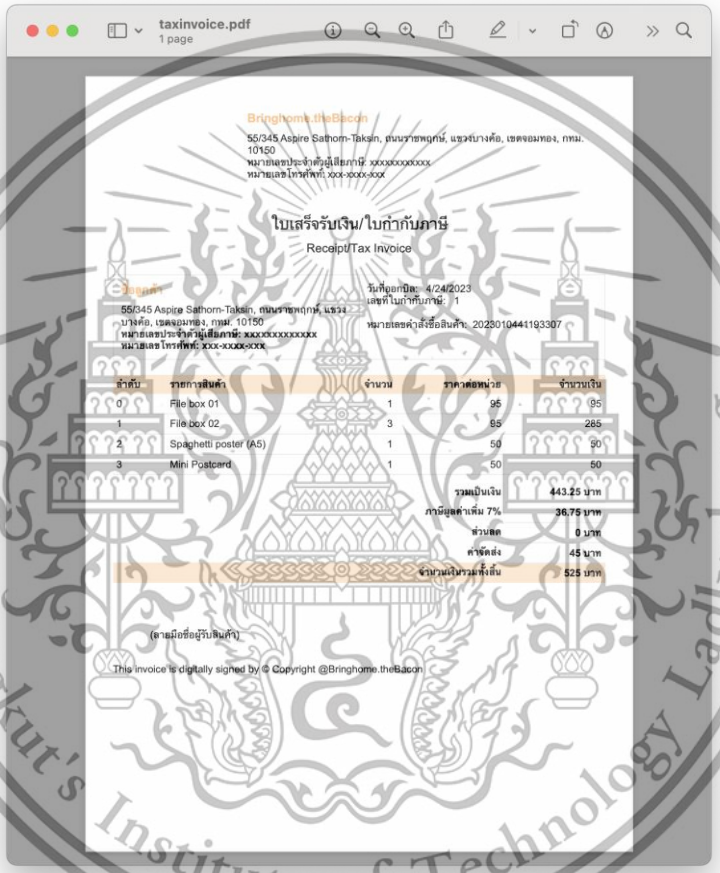

Figure 76 PDF E-tax invoice first version

4.2.2 PDF E-tax invoice second version

The second version was improved according to the above suggestions and already encrypted the tax invoice before sending the response as a tax invoice PDF file to the customer.

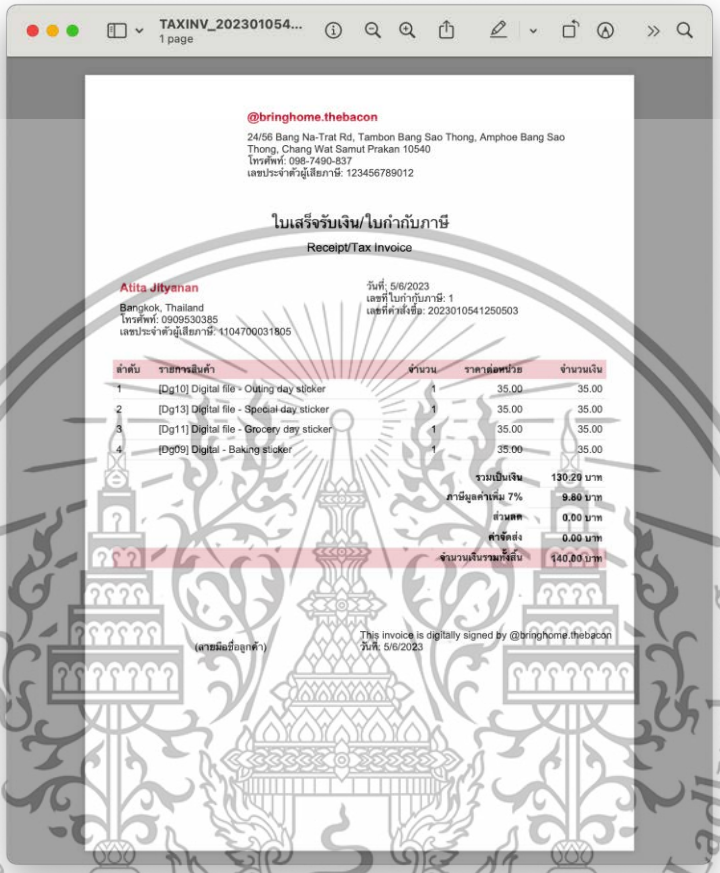

Figure 77 PDF E-tax invoice second version

4.2.3 PDF E-tax invoice final version

Finally, the final version of the tax invoice is fully approved by the company, with some revised details such as the company's name after registration and tax invoice number recording.

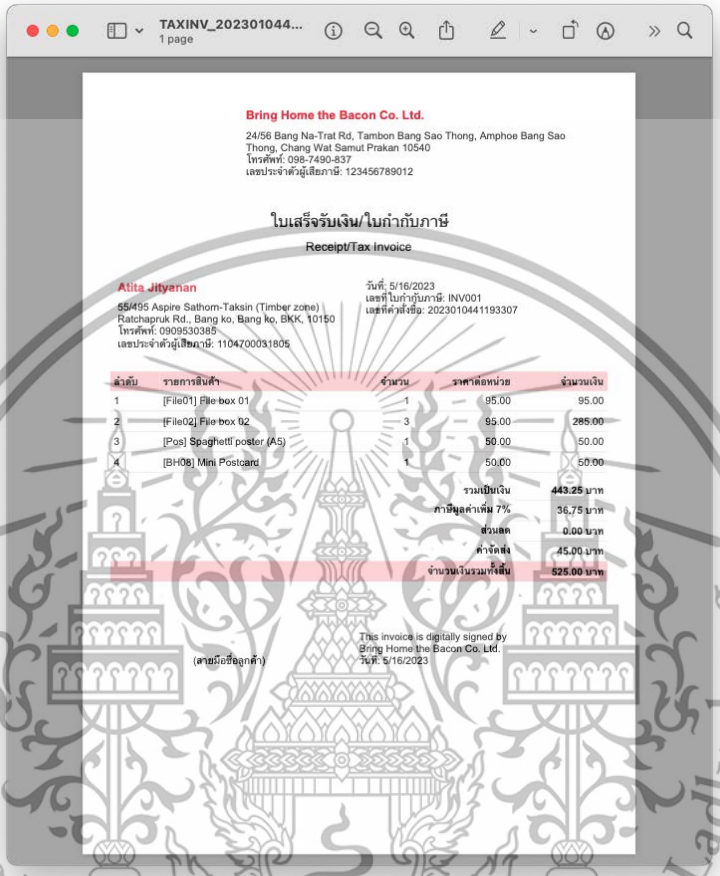

Figure 78 PDF E-tax invoice final version

Chapter 5
Summary of Research Results and Recommendations

5.1 Summary

Smart e-Tax Invoice web application aims to streamline and automate the process of generating tax invoices, ensuring accuracy, compliance with tax regulations, and increased efficiency for organizations. With features such as automated data entry, from customer, order transaction data integration and digitally signed document, it streamlines the invoicing process, saves time and costs, enhances accuracy, and improves the overall customer experience. Implementing such an application can lead to increased productivity, reduced errors, and improved financial management for organizations.

5.2 Recommendations

Recommendations of this research was received from the company itself and advisor for further research and improvement as the following listed.

5.2.1 Enhance server-side validation by adding additional features to improve the security and reliability of the customer form. This could involve implementing validation checks for data types, such as identification numbers, to ensure that only valid and compatible data is accepted.

5.2.2 Deploy the web application as HTTPS to enhance the security of user data and protect against potential cyber threats. This ensures that all communication between the user's browser and the web application is encrypted, providing a secure browsing experience.

5.2.3 Write real-time data to the database using LINE Shopping API instead of manual process in order to improve further performance.

Bibliography

Mozilla Developer Network. (n.d.). HTML. Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTML

Mozilla Developer Network. (n.d.). CSS. Retrieved from https://developer.mozilla.org/en-US/docs/Web/CSS

Mozilla Developer Network. (n.d.). JavaScript. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

Revenue Department. (n.d.). Value Added Tax (VAT) 360. Retrieved from https://www.rd.go.th/fileadmin/user_upload/SMEs/infographic/13-1.vat_360.pdf

Revenue Department. (n.d.). e-Tax Invoice Flipbook. Retrieved from https://etax.rd.go.th/etax_staticpage/app/emag/flipbook/flipbook01/#p=1

IBM. (n.d.). Security: Asymmetric Algorithm Public Key Cryptography. Retrieved from https://www.ibm.com/docs/en/zos/2.1.0?topic=security-asymmetric-algorithm-public-key-cryptography

TechTarget. (n.d.). GitHub. Retrieved from https://www.techtarget.com/searchitoperations/definition/GitHub

Geekflare. (n.d.). How to Delete a GitHub Repository? Retrieved from https://geekflare.com/delete-github-repository/

PTTRNS. (n.d.). Figma vs Adobe XD: Which Design Tool Should You Go With? Retrieved from https://www.pttrns.com/figma-vs-adobe-xd-which-design-tool-should-you-go-with/

MakeUseOf. (n.d.). What Is Figma Used For? Retrieved from https://www.makeuseof.com/what-is-figma-used-for/

Revenue Department. (n.d.). Digital Signature for e-Tax Invoice. Retrieved from https://etax.rd.go.th/etax_staticpage/app/emag/flipbook/06_Digital-Signature.pdf

Electronic Transactions Development Agency (ETDA). (n.d.). Digital Signature. Retrieved from https://www.etda.or.th/th/Our-Service/Digital-Trusted-services-Infrastructure/TEDA/Digital-Signature.aspx

Hashnode. (n.d.). Digital Signature in Node.js. Retrieved from https://hashnode.com/post/digital-signature-in-node-js-cks3ewmq40vqrhqs1axb9h4ci

PeculiarVentures/xadesjs. (n.d.). GitHub Repository. Retrieved from https://github.com/PeculiarVentures/xadesjs

Electronic Transactions Development Agency (ETDA). (n.d.). ประกาศกรมสื่อสารทางอิเล็กทรอนิกส์ เรื่อง แนวทางการลงนามทางอิเล็กทรอนิกส์ สำหรับสัญญาเช่าฯ. Retrieved from https://www.etda.or.th/th/Useful-Resource/%E0%B8%81%E0%B8%8F%E0%B8%AB%E0%B8%A1%E0%B8%B2%E0%B8%A2-HTML/%E0%B8%9E%E0%B8%A3%E0%B8%B0%E0%B8%A3%E0%B8%B2%E0%B8%8A%E0%B8%9A%E0%B8%8D%E0%B8%8D%E0%B8%95%E0%B8%A7%E0%B8%B2%E0%B8%94%E0%B8%A7%E0%B8%A2%E0%B8%98%E0%B8%A3%E0%B8%81%E0%B8%A3%E0%B8%A3%E0%B8%A1%E0%B8%97%E0%B8%B2%E0%B8%87%E0%B8%AD%E0%B9%80%E0%B8%A5%E0%B8%81%E0%B8%97%E0%B8%A3%E0%B8%AD%E0%B8%99%E0%B8%8

1%E0%B8%AA/%E0%B8%9E%E0%B8%A3%E0%B8%B0%E0%B8%A3%E0%B8%B2%E0%B8%8A%E0%B8%9A%E0%B8%8D%E0%B8%8D%E0%B8%95%E0%B8%A7%E
Microsoft. (n.d.). Digital Signatures. Retrieved from https://learn.microsoft.com/en-us/windows/win32/seccrypto/digital-signatures
European Union Agency for Cybersecurity (ENISA). (n.d.). Public Key Infrastructure (PKI). Retrieved from https://www.enisa.europa.eu/topics/incident-response/glossary/public-key-infrastructure-pki
Deistercloud. (n.d.). PKCS7. Retrieved from https://docs.deistercloud.com/content/Axional%20development%20libraries.20/Server%20side%20javascript.50/AX%20Library.10/crypt/PKCS.10/PKCS7.xml?embedded=true
Revenue Department. (n.d.). Tax Incentives. Retrieved from https://www.rd.go.th/english/6043.html
IBM. (n.d.). SQL vs. NoSQL. Retrieved from https://www.ibm.com/cloud/blog/sql-vs-nosql#:~:text=SQL%20and%20NoSQL%20differ%20in,row%20transactions%20or%20unstructured%20data.
Pragim Technologies. (n.d.). MongoDB Tutorial. Retrieved from https://www.pragimtech.com/blog/mongodb-tutorial/relational-and-non-relational-databases
Simplilearn. (n.d.). What is Express.js? Retrieved from https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js#:~:text=Express%20is%20a%20node%20js,helps%20manage%20servers%20and%20routes.
GeeksforGeeks. (n.d.). Why to Use Node.js for Backend Development? Retrieved from https://www.geeksforgeeks.org/why-to-use-node-js-for-backend-development/
TechTarget. (n.d.). Database. Retrieved from https://www.techtarget.com/searchdatamanagement/definition/database
Simplilearn. (n.d.). What Is a Database? Retrieved from https://www.simplilearn.com/tutorials/dbms-tutorial/what-is-a-database#what_is_a_database
Database Town. (n.d.). Relational Database vs. Object-Oriented Database: Key Differences. Retrieved from https://databasetown.com/relational-database-vs-object-oriented-database-key-differences/
IBM. (n.d.). Relational Databases. Retrieved from https://www.ibm.com/topics/relational-databases
IBM. (n.d.). NoSQL Databases. Retrieved from https://www.ibm.com/topics/nosql-databases
MongoDB. (n.d.). What is MongoDB? Retrieved from https://www.mongodb.com/what-is-mongodb
IBM. (n.d.). MongoDB. Retrieved from https://www.ibm.com/topics/mongodb
MongoDB. (n.d.). MongoDB Schemas. Retrieved from https://www.mongodb.com/docs/atlas/app-services/schemas/#:~:text=A%20schema%20is%20a%20JSON,created%2C%20changed%2C%20or%20deleted
TutorialsTeacher.com. (n.d.). MongoDB Tutorial. Retrieved from https://www.tutorialsteacher.com/mongodb/what-is-mongodb

Baeldung. (n.d.). IP All-Zeroes. Retrieved from https://www.baeldung.com/cs/ip-all-zeroes#:~:text=0.0%20means%20it%20will%20accept,be%20accessible%20

Biography

Atita Jityanan was born on January 21st, 2001. She is a Robotics and AI Engineering student in the faculty of School of International & Interdisciplinary Engineering at the King Mongkut's Institute of Technology Ladkrabang (KMITL). She took a role as a Chatbot Developer Intern as her first internship at LINE Company (Thailand) in Summer 2022. After that, she joined 4 months internship at ExxonMobil IT as Customer Engagement Intern. Atita has area of interests fall under IT product development, business, and product management. In this project, she is responsible for three main tasks: E-tax invoice generation, digital signature generation, and UI design. The last part of her responsibilities involves designing the user interface, including a grammar check feature.

The author, Jiratchaya Fuengthanakul, was born on November 8th, 2000, in Bangkok, Thailand. She is currently a fourth-year student pursuing a bachelor's degree in Robotic and Artificial Intelligence Engineering at King Mongkut's Institute of Technology Ladkrabang. Throughout her academic journey, Jiratchaya has gained practical experience by interning at various companies. In the summer of 2022, she worked as a cloud customer solutions consultant intern at Google Thailand for a duration of 4 months. Subsequently, she joined ExxonMobil as an IT business consultant intern for another 4-month cooperative internship period. In this project, she is responsible for the database connection with the backend, which connects to the DDNS backend deployment. Additionally, she is responsible for the HTML structure of the pages where users can download and request invoices.

SMART E-TAX INVOICE GENERATOR

Atita Jityanan, Jiratchaya Fuengthanakul
Robotic and AI Department of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
atita.jit@gmail.com, fueng.jiratchaya@gmail.com

***Abstract*— This thesis presents a Smart e-Tax Invoice Generator, which is designed to streamline the process of generating tax invoices electronically. This web-based solution offers a range of features and functionalities to simplify and automate invoice generation, ensuring compliance with tax regulations. Key features of the e-tax invoice generator include accurately generated tax invoices matched with the customer's order as requested, automated tax calculations, certified generated tax invoices with the digital signature, and secure storage and retrieval of generated invoices for further tax operation for the company. These features collectively enable business, Bringhome.theBacon our company participants who are under the process of Company Limited registration, to save time and effort in generating accurate tax invoices and further operation after completing the registration process to adhere to tax guidance. The project was tested using actual customer order transactions and accurately generated tax invoices for individual customers.**

Keywords— Tax invoice, Web application, Digital Signature

I. BACKGROUND AND SIGNIFICANT

With the rapid advancement of technology and the increasing accessibility of the internet, global e-commerce has become one of the fastest-growing industries in the world. The shift towards online shopping has been driven by factors such as convenience, greater product variety, and competitive pricing. According to Lertad Supadhiloke, Director of e-Commerce at LINE Thailand, speaking at the LINE Shopping Social Commerce 2022 event, "the social commerce market of LINE SHOPPING is having Gross Merchandise Value around 252% in 2022" ('ก้าวกระโดดอย่างมั่นใจในยุค Social Commerce ด้วย LINE SHOPPING' session on the main stage, August 22nd, 2022). The predicted statistic isn't temporary growth but customer behavior on digital transformation was consistent. E-commerce has revolutionized the way businesses operate, enabling them to reach a wider customer base and sell their products and services online.

1) Research Objectives

The main objective is to develop a web application that automates the process of generating tax invoices for businesses. The web application will be designed to generate electronic invoices with digitally signed signatures that are compliant with the tax laws and regulations of the jurisdiction and capable of verification.

2) Expected benefits

Aim to help businesses save time and resources by automating the invoicing process and ensuring accuracy and consistency in the generation of invoices compliant with the Revenue Department of Thailand and Improve the overall efficiency of the invoicing process by reducing errors and streamlining the documentation and record-keeping processes.

II. RESEARCH

1) Tax Invoice Research

The process of how a customer requests a tax invoice from a seller involves several steps to ensure accurate and compliant documentation. It begins when the customer contacts the seller and expresses their need for a tax invoice. Upon receiving the customer's request, the seller carefully collects the necessary information to generate the tax invoice. The customer is asked to provide key details such as their full name, complete address, and tax identification number. Once the customer's information is obtained, the seller proceeds to verify the request. This involves reviewing the provided information to ensure its accuracy and completeness. After verifying the request, the seller moves on to the tax invoice generation stage. Using the customer's information and the relevant details of the purchase or transaction, the seller creates a comprehensive tax invoice. This document typically includes the customer's name and address, details of the products or services purchased, any applicable taxes or fees, and the total amount payable. Once the tax invoice is generated, the seller prepares it for delivery to the customer. The mode of delivery may depend on the agreed-upon method between the parties involved. It could be a physical document, where the seller prints and hands over the tax invoice to the customer. Alternatively, the tax invoice may be sent electronically in the form of a PDF file, which can be conveniently shared via email or any other digital communication platform. By following this detailed process, the customer can successfully request a tax invoice from the seller. This ensures compliance with tax regulations and provides the customer with accurate and reliable documentation for their purchase.

2) Solution initiation

The result of user research with the company and tax invoice process lead to workflow and features of Smart e-Tax Invoice Generator as stated below.

1) Customer filling form: Developing a medium for receiving customer information with request form user interface, to gather customer's information including name, address and tax identification number.

2) e-Tax Invoice generation: generating tax invoice customer's information and order transaction details from the database as PDF file.

3) Encryption: encrypting tax invoice with XAdES digital signature.

The workflow and features include a customer request form interface to gather customer information, e-Tax invoice generation from the database as a PDF file, and encryption of the invoice with XAdES digital signature.

3.) *Workflow design*

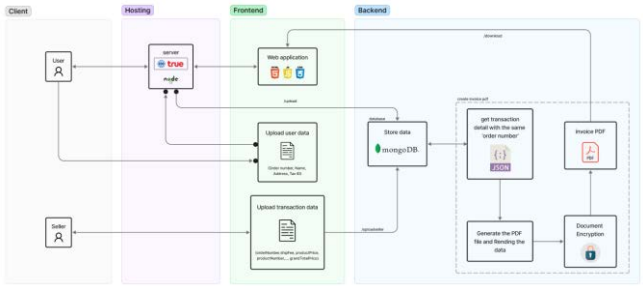

Fig. 1. Workflow diagram of Web Application

Since the received data is separated into two sides, the store's side and the customer's side, with the store's data being divided into different order numbers, and the customer's side having data stored with different order numbers as well, these two sets of data will need to be used together by grouping them based on the same order number. This is necessary to create a PDF file from this combined data.

After evaluating the data obtained from the store and customer data, it was determined that a NoSQL database would be the better choice. This is because the data obtained from the store is in the form of an Excel file, which is a document format. Meanwhile, the data obtained from customers is stored in the form of a JSON file, which has a large number of customer data components combined together. NoSQL databases offer greater flexibility than traditional SQL databases because they do not require a fixed schema. This means that you can store data in any format you choose, without having to worry about predefined data types and structures. In addition to flexibility, NoSQL databases can be faster than SQL databases in certain situations because they are optimized for specific use cases. They also use distributed architectures and in-memory caching to provide faster query response times. Based on the data obtained from the store and customer data, a NoSQL database is the recommended solution. Its flexibility and performance advantages make it a better choice than a traditional SQL database.

III. DEVELOPMENT

In the development process, the first step is to thoroughly understand and analyze the data in the documents. Due to the abundance of data in the document, it is necessary to clean the data before using it, as some columns may be unnecessary in the e-tax invoice file. Therefore, they need to be removed before proceeding. Understanding the structure of the e-tax invoice is crucial since it contains sensitive data. Researching e-tax invoice standards is also essential during development to find the best solution for users.

The front end serves as the customer-facing platform, showcasing the unique design of the store to customers. Additionally, user experience plays a vital role, including a user-friendly design and error handling that effectively guides users in case of missing or repetitive data entry.

1.) *Transaction data*

The raw data related to order transactions is stored in a CSV (Comma-Separated Values) file format. In the order transaction data structure, each row of the CSV file represents a single order, and each cell within that row represents a specific product associated with that order. If an order contains multiple products, each product will be represented by its own cell within the same row as the order number. For example, if an order with the number "20230515" contains three products, there will be three cells in the same row as "20230515", with each cell representing a different product. Not only the file type that is not in the right format, but the data also includes key fields in Thai and null values, it becomes crucial to manage the data properly to ensure accurate tracking and reporting. The figure below shows the raw data of order transactions.

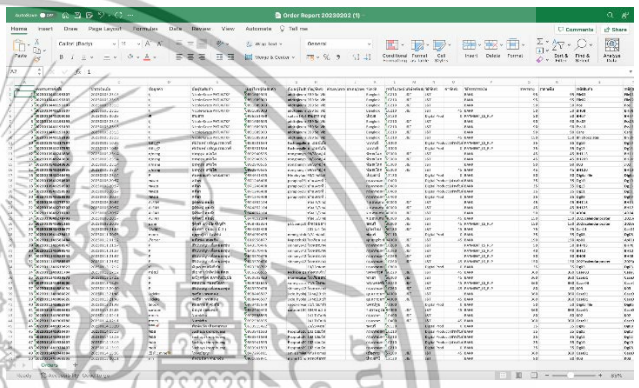

Fig. 2. Transaction data in .csv format.

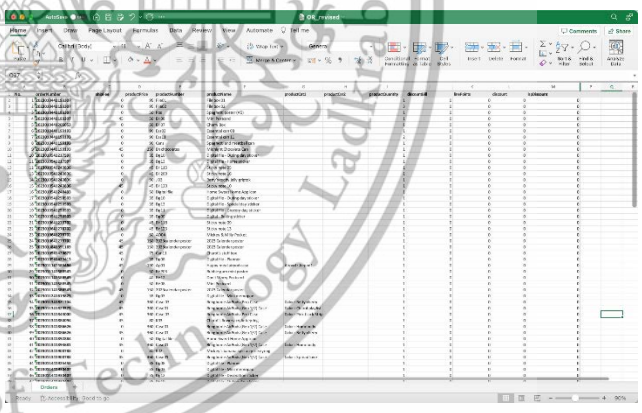

Fig. 3. Organized Order Transaction data in .csv format

2.) *E-tax Invoice Operation.*

In this research, authors used Vanilla JavaScript, NodeJS, and Handlebars in generating dynamic PDF tax invoices with both static and dynamic data on a NodeJS app running on a web browser. PDF tax invoice mainly consists of two parts, a data model, and an e-tax invoice template. The data used is separated into static and dynamic data. The static data is the company's information including company name, company tax issuer identification number, address, and so on. Conversely, customer data received from filled forms and order transaction data retrieved from MongoDB are dynamic

data. The template of the e-Tax invoice was created using Handlebars to create dynamic PDF documents like tax invoices.

3.) *Digital Signature*

Digital signatures are widely used to ensure the integrity and authenticity of electronic documents. A digital signature provides assurance that the document has not been altered since it was signed and that the signer is who they claim to be. Type of digital signature used in this research which also meets the recommendation of the Revenue Department is the XML Advanced Electronic Signatures (XAdES) signature. The XAdES signature consists of two main components: the signature itself and the signed properties. The signature contains the cryptographic information that ensures the integrity and authenticity of the signed document, while the signed properties provide additional information about the signature, such as the signing time and the signer's identity. The tool should specify the signing algorithm and provide a certificate for signing. The signing process involves the following steps

i. The document to be signed is converted to an XML format.
ii. The signing tool calculates a cryptographic hash of the document and signs the hash using the private key associated with the signer's certificate.
iii. The signature and signed properties are added to the XML document.
iv. The signed XML document can then be verified by a recipient using the signer's public key.

a. *Process of adding digital signature*

In this research, the XAdES process involves generating a private key, calculating a hash value of the XML document, creating a digital signature using the private key, including the public key certificate and signing time, and incorporating the signature and qualifying properties into the XML document. Base 64 encoding will be used for specific elements like the digest value and public key certificate for proper representation and transmission of the signature-related data.

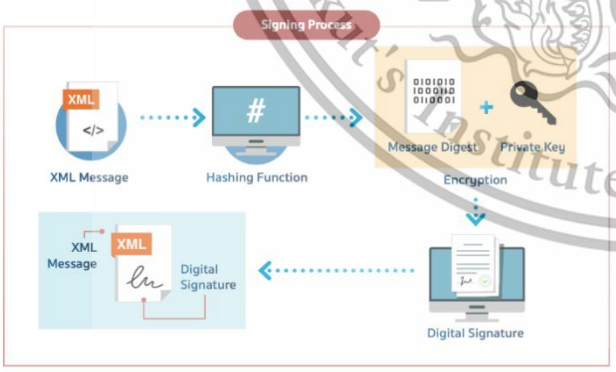

Fig. 4. Signing Process

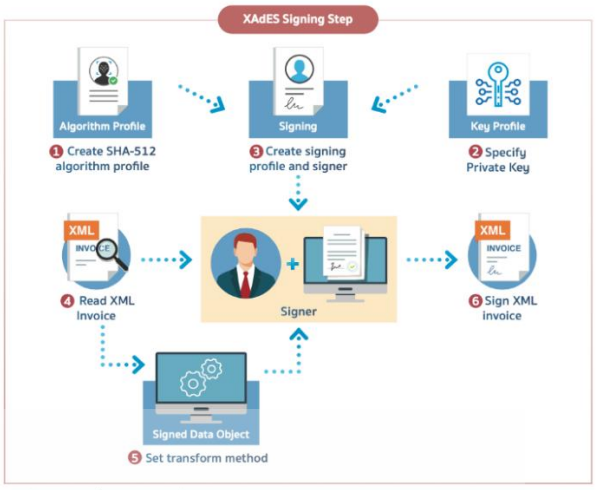

Fig. 5. XAdES Signing Step

4.) *Database*

In this research, authors selected MongoDB, which MongoDB (link resides outside IBM) is an open source, non relational database management system (DBMS) that uses flexible documents instead of tables and rows to process and store various forms of data. As a NoSQL database solution, MongoDB does not require a relational database management system (RDBMS), so it provides an elastic data storage model that enables users to store and query multivariate data types with ease. This not only simplifies database management for developers but also creates a highly scalable environment for cross-platform applications and services.

Fig. 6. MongoDB

5.) *TRUE DDNS*

True Corporation offers a DDNS service that allows users to access their network devices or services using a domain name that stays the same even if their IP address changes. This service is commonly used for remote access to home networks, hosting websites or servers at home, and accessing networked cameras or other devices. The True DDNS service works by associating a domain name with a user's current IP address, which can change periodically as a result of changes to the user's internet connection. Users can sign up for the service and configure their router or device to send updates to the True DDNS service with their current IP address. The True DDNS service then maps the user's domain name to their current IP address, allowing users to access their devices or services using a human-readable domain name rather than an IP address. Overall, True DDNS is a useful service for users who have a dynamic IP address that changes frequently and need to maintain a consistent domain name for accessing their network devices or services. However, it's important to note that DDNS services may not always be reliable, and changes to DNS records can take time to propagate, so it may

not be suitable for high-availability or mission-critical applications.

IV. Research Result

1.) *Web application*

The Smart E-tax Invoice generator project deploys a web application with three versions. The first version is a prototype testing version without any interface decoration. It is used to test the connection between the MongoDB database and the frontend user. During testing, it was discovered that users needed to input the order number to match the transaction details. However, this version does not include any data security measures, such as encryption and private keys in the e-tax invoice file. Additionally, it does not have the functionality to generate PDF files. The second version incorporates interface decoration based on the store's concept design. It takes into account the feedback and errors identified in the first version. This version includes an order number filling form where users can enter their order number to ensure accurate retrieval of transaction details. Despite the design changes, the connection between the MongoDB database and user input remains functional. Version 2 version also includes the functionality to generate PDF files. However, it is important to note that this version, like the previous one, does not implement data security measures such as encryption and private keys. In the final version of the Smart E-tax Invoice generator project, the order number filling form has a condition that requires users to fill it out. It is a one-time filling and cannot be left blank. As previous versions' limitations regarding data security, the final version enhances security by implementing encryption and electronic signatures in the e-tax invoice file. These measures provide an additional layer of protection to ensure the document's integrity and security.

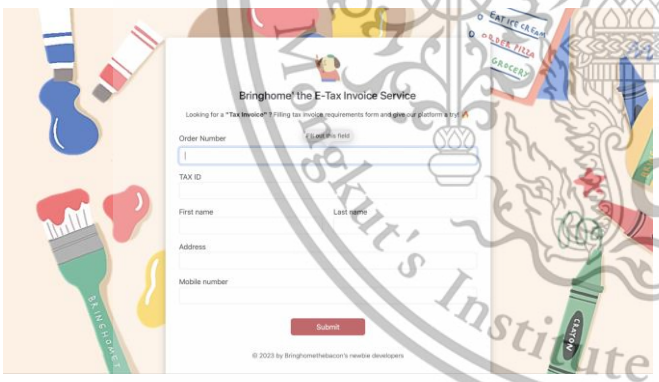

Fig. 7. Web Application

2.) *PDF E-tax invoice*

The first version of tax invoice PDF format consisted of all the necessary information for tax invoice and represented all the order transactions matched with customer information input accurately. However, there's room for improvements since some of the details need to be fixed such as the layout of customer information presentation, the position of signature as well as the encryption process that need to be added. The second version was improved according to the above suggestions and already encrypted the tax invoice before sending the response as a tax invoice PDF file to the customer. Finally, the final version of the tax invoice is fully approved by the company, with some revised details such as the company's name after registration and tax invoice number recording.

Fig. 8. PDF E-tax invoice final version

V. Summary of Research Results

Smart e-Tax Invoice web application aims to streamline and automate the process of generating tax invoices, ensuring accuracy, compliance with tax regulations, and increased efficiency for organizations. With features such as automated data entry, from customers, order transaction data integration, and digitally signed document, it streamlines the invoicing process, saves time and costs, enhances accuracy, and improves the overall customer experience. Implementing such an application can lead to increased productivity, reduced errors, and improved financial management for organizations. *Transaction data*

References

[1] Mozilla Developer Network. (n.d.). HTML. Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTML

[2] Mozilla Developer Network. (n.d.). CSS. Retrieved from https://developer.mozilla.org/en-US/docs/Web/CSS

[3] Mozilla Developer Network. (n.d.). JavaScript. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

[4] Revenue Department. (n.d.). Value Added Tax (VAT) 360. Retrieved from https://www.rd.go.th/fileadmin/user_upload/SMEs/infographic/13-1.vat_360.pdf

[5] Revenue Department. (n.d.). e-Tax Invoice Flipbook. Retrieved from https://etax.rd.go.th/etax_staticpage/app/emag/flipbook/flipbook01/#p=1

[6] IBM. (n.d.). Security: Asymmetric Algorithm Public Key Cryptography. Retrieved from https://www.ibm.com/docs/en/zos/2.1.0?topic=security-asymmetric-algorithm-public-key-cryptography

[7] TechTarget. (n.d.). GitHub. Retrieved from https://www.techtarget.com/searchitoperations/definition/GitHub

[8] Geekflare. (n.d.). How to Delete a GitHub Repository? Retrieved from https://geekflare.com/delete-github-repository/