ระบบรวบรวมข้อมูลอัตโนมัติ

AUTOMATIC CRAWLING SYSTEM

ธีรวัจน์ สันต์สวัสดิ์
เสาวคนธ์ โชติช่วง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบรวบรวมข้อมูลอัตโนมัติ

AUTOMATIC CRAWLING SYSTEM

ธีรวัจน์ สันต์สวัสดิ์

เสาวคนธ์ โชติช่วง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรวบรวมข้อมูลอัตโนมัติ

AUTOMATIC CRAWLING SYSTEM

ผู้จัดทำ

1. นายธีรวัจน์ สันต์สวัสดิ์ รหัสนักศึกษา 57010634
2. นางสาวเสาวคนธ์ โชติช่วง รหัสนักศึกษา 57011439

_____ อาจารย์ที่ปรึกษา

(รศ.ดร.บุญธีร์ เครือตราชู)

_____ อาจารย์ที่ปรึกษาร่วม

(รศ.กฤตวัน ศิริบูรณ์)

ระบบรวบรวมข้อมูลอัตโนมัติ

นายธีรวัจน์	สันต์สวัสดิ์	57010634
นางสาวเสาวคนธ์	โชติช่วง	57011439
รศ.ดร.บุญธีร์	เครือตราชู	อาจารย์ที่ปรึกษา
รศ.กฤตวัน	ศิริบูรณ์	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2560

บทคัดย่อ

โครงงานชิ้นนี้สร้างระบบรวบรวมข้อมูลอัตโนมัติ ที่สามารถรวบรวมข้อมูลจำนวนมากได้จากเว็บไซต์ที่กำหนด โดยการป้อนคำค้นหา (Keyword) ลงในหน้าค้นหาที่ทำบน Web Application เพื่อส่งไปยัง Web server สำหรับทำการค้นหา Keyword ผ่าน Google Search Engine เพื่อเป็นการกำหนดที่อยู่ของหน้าเว็บที่บอกให้บอทที่จะไปทำการเก็บข้อมูล ข้อมูลที่ได้มาจากการทำงานของบอทจะถูกจัดเก็บลงบนฐานข้อมูลเป็นข้อความตัวอักษร (Text) และตัวเลข (Number) ซึ่งผู้ใช้งานสามารถ Export ไฟล์ออกมาเป็นชนิด CSV เพื่อนำไปใช้งานต่อได้ตามความต้องการ นอกจากนี้ ในหน้าแสดงผลการค้นหาข้อมูลของ Web Application ยังแสดงข้อมูลที่บอททำการเก็บมาตามช่วงเวลาที่กำหนดไว้ในรูปแบบของตารางและกราฟ ซึ่งสามารถนำมาใช้สำหรับทำการวิเคราะห์ข้อมูลในเบื้องต้นได้ทันที

Automatic Crawling System

Mr. Theerawat	Sansawat	57010634
Ms. Saowakon	Chotchuang	57011439
Assoc.Prof.Dr. Boontee	Kruatrachue	Advisor
Assoc.Prof. Kritawan	Siriboon	Co-Advisor

Academic Year 2560

ABSTRACT

This project implements automatic crawling system which can collect large data from many websites using keywords. The system uses web application to receive keywords and gather web pages address from Google Search Engine with the keywords. The bot is used to collect data from all web pages address then the collected data will be stored in database by type text and number. User can export to csv. In addition, web application result page can display collected data in table and chart for further used immediately.

กิตติกรรมประกาศ

โครงงานเรื่องการรวบรวมข้อมูลอัตโนมัติฉบับนี้นั้นสามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือทั้งจากทางตรงและทางอ้อมของบุคคลหลายฝ่าย ซึ่งจะไม่สามารถสำเร็จลงได้ถ้าหากปราศจากบุคคลเหล่านี้ อันได้แก่

รศ.ดร.บุญธีร์ เครือตราชู และ รศ.กฤตวัน ศิริบูรณ์ อาจารย์ที่ปรึกษาทั้งสองท่านที่คอยให้คำแนะนำเกี่ยวกับโครงงานตั้งแต่เริ่มต้น ให้คำปรึกษาเมื่อเวลาที่พบเจอปัญหาแล้วไม่สามารถแก้ไขด้วยตนเองได้ คอยชี้แนะแนวทางและข้อคิดเห็นอันเป็นประโยชน์ในการนำไปพัฒนาโครงงานเพื่อต่อยอดให้ดียิ่งขึ้น อีกทั้งยังให้การดูแลอย่างใกล้ชิดเพื่อให้การทำงานต่าง ๆ ดำเนินไปอย่างราบรื่นและเกิดประสิทธภาพสูงที่สุด

บิดา มารดาและครอบครัว ที่ให้การสนับสนุนมาโดยตลอด คอยให้กำลังใจเสมอในเวลาที่ต้องการ อบรมสั่งสอนและเลี้ยงดู รวมทั้งให้ความรู้และคำแนะนำที่มีประโยชน์ต่าง ๆ เพื่อให้ไปถึงเป้าหมายที่ตั้งใจไว้

คณะท่านอาจารย์และบุคลากรในภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความรู้และการสนับสนุนตลอดระยะเวลาที่ผ่านมา สถานที่และสิ่งอื่น ๆ ที่จำเป็นในการพัฒนาโครงงาน ตลอดจนการเรียนรู้ในการทำงานกันเป็นกลุ่ม สามารถทำงานร่วมกับผู้อื่นได้เป็นอย่างดี

จึงขอขอบพระคุณทุกท่านที่กล่าวมา ตลอดจนรุ่นพี่และเพื่อน ๆ รวมถึงผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่ไม่ได้กล่าวถึงไว้ในส่วนนี้ ด้วยความดีและคุณค่าอันเกิดจากประโยชน์ที่พึงมีในโครงงานชิ้นนี้ ขอมอบให้กับผู้ที่มีส่วนเกี่ยวข้องทุกท่านและขอกราบขอบพระคุณมา ณ ที่นี้ ในส่วนของข้อบกพร่อง หากเกิดความผิดพลาดประการใด ต้องขออภัยไว้ ณ โอกาสนี้ด้วย

ธีรวัจน์ สันต์สวัสดิ์
เสาวคนธ์ โชติช่วง

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตาราง **หน้า**

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

รูป หน้า

สารบัญรูป (ต่อ)

รูป **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

การรวบรวมข้อมูลจากเว็บไซต์ ส่วนใหญ่เป็นการดึงข้อมูลจากแหล่งต่าง ๆ หรือทำการคัดลอกแล้ววางลงในเอกสาร ถ้าหากข้อมูลที่ต้องการทำการรวบรวมนั้นมีจำนวนน้อย ก็จะสามารถรวบรวมให้เสร็จสิ้นได้ในเวลาที่กำหนด แต่ถ้าเกิดข้อมูลที่ทำการรวบรวมอยู่นั้นมีจำนวนมาก รายละเอียดย่อยในข้อมูลมีความหลากหลายหรือมีจำนวนหน้าเว็บไซต์ตั้งแต่สิบหน้าหรือร้อยหน้าขึ้นไป การจะรวบรวมข้อมูลจำนวนมากให้เสร็จทันภายในเวลาอันรวดเร็วคงจะเป็นไปได้ยาก แต่ในขณะเดียวกันหากมีเครื่องมือที่สามารถรวบรวมข้อมูลที่มีความถูกต้องและตรงตามความต้องการของผู้ใช้งานได้นั้น แม้ข้อมูลที่ต้องการจะมีจำนวนมากเท่าใดก็สามารถรวบรวมข้อมูลทั้งหมดให้สำเร็จภายในเวลาที่กำหนดได้

ในโครงงานชิ้นนี้จึงมีแนวคิดที่จะเลือกใช้ภาษาไพธอน (Python) ในการทำเครื่องมือสำหรับการรวบรวมข้อมูล เนื่องจากปัจจุบัน การนำภาษาไพธอนมาใช้งานได้รับความนิยมอย่างมาก เพราะนอกจากจะสามารถใช้งานได้ง่าย มีไลบรารีที่หลากหลายให้เลือกใช้ สามารถนำมาใช้งานได้อย่างเหมาะสมตามความต้องการแล้ว หากติดปัญหาหรือมีข้อสงสัยก็สามารถหาคำตอบได้อย่างรวดเร็วจากอินเตอร์เน็ต ซึ่งเป็นข้อดีของการได้รับความนิยมในใช้งาน ณ ปัจจุบันนี้

1.2 วัตถุประสงค์ของโครงงาน

1) สร้างบอท (Bot) เพื่อรวบรวมข้อมูลจากการกำหนดคีย์เวิร์ด (Keyword) ของข้อมูลและยูอาร์แอล (URL) ที่กำหนดได้อย่างมีประสิทธิภาพ ข้อมูลที่ได้ครบถ้วนตามต้องการ
2) ทำการเปรียบเทียบและหาข้อจำกัดของไลบรารีที่สามารถนำมาใช้สร้างบอทสำหรับทำการรวบรวมข้อมูล
3) ข้อมูลที่ได้จากการรวบรวมของบอทจะถูกเก็บอยู่ในรูปแบบและโครงสร้างข้อมูลตามที่ผู้ใช้งานกำหนดอย่างถูกต้อง และไม่มีการซ้ำกันของข้อมูล
4) ลดระยะเวลาและขั้นตอนในการเก็บและจัดเตรียมข้อมูลก่อนการนำไปใช้งาน

1.3 ขอบเขตของโครงงาน

1) สามารถรวบรวมข้อมูลจากคีย์เวิร์ดของข้อมูลและยูอาร์แอลที่กำหนด
2) ข้อมูลที่รวบรวมได้อยู่ในรูปแบบและโครงสร้างข้อมูลที่กำหนดไว้อย่างถูกต้อง

3) ข้อมูลที่รวบรวมได้สามารถทำการส่งออกข้อมูล (Export) เป็นไฟล์ประเภทต่าง ๆ ได้ เช่น ไฟล์ประเภทซีเอสวี (.csv) เป็นต้น เพื่อให้ง่ายต่อการนำไปใช้งาน
4) ข้อมูลที่รวบรวมจะจัดเก็บเป็นข้อมูลชนิดข้อความตัวอักษร (Text) และตัวเลข (Number) เช่น ข้อมูลประเภทราคาจะจัดเก็บเป็นข้อมูลชนิดตัวเลข เป็นต้น

1.4 วิธีการดำเนินงาน

เว็บครอว์เลอร์ (Web crawler) หรือเว็บสไปเดอร์ (Web spider) เป็นบอทที่เก็บข้อมูลทางอินเตอร์เน็ต (Internet bot) ซึ่งใช้ในการเก็บข้อมูลภายในเว็บไซต์ที่กำหนด โดยเลือกจากคีย์เวิร์ด เพื่อที่จะนำข้อมูลมาใช้งานในการวิเคราะห์และเปรียบเทียบ

ในการทำโครงงาน จะมีวิธีการ ดังนี้

1) ศึกษาหาข้อจำกัดและเปรียบเทียบไลบรารีต่าง ๆ ที่ช่วยในการสร้างระบบรวบรวมข้อมูล
2) ทำการเลือกไลบรารีที่ดีและเหมาะสมเพื่อนำมาสร้างระบบรวบรวมข้อมูล
3) สร้างส่วนต่อประสานกับผู้ใช้ (User Interface) สำหรับการใช้งานระบบเป็นเว็บแอปพลิเคชัน (Web Application)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้และความเข้าใจในการนำไลบรารีที่ช่วยในการสร้างบอทสำหรับรวบรวมข้อมูลของภาษาไพธอนมาใช้งาน
2) เข้าใจหลักการการทำงานของเว็บครอว์เลอร์
3) เข้าใจการทำเว็บไซต์และสามารถวิเคราะห์โครงสร้างของเว็บไซต์เพื่อนำไปใช้งานได้
4) เป็นแนวทางในการพัฒนาเกี่ยวกับเว็บครอว์เลอร์หรือเว็บสไปเดอร์

บทที่ 2
ความรู้และเทคโนโลยีที่เกี่ยวข้อง

2.1 ภาษาไพธอน

ไพธอน (Python) เป็นภาษาเขียนโปรแกรมระดับสูง (high-level programming language) ได้รับการพัฒนาขึ้นโดย Guido van Rossum ในช่วงปี คศ. 1985-1990 ที่สถาบันวิจัยคณิตศาสตร์และวิทยาการคอมพิวเตอร์แห่งชาติในประเทศเนเธอร์แลนด์ มีต้นแบบในการพัฒนามาจากภาษาอื่น เช่น ภาษาซี (C) ภาษาซีพลัสพลัส (C++) ยูนิกซ์ เชลล์ (Unix shell) และภาษาสคริปต์อื่น ๆ ไพธอนมีลิขสิทธิ์อยู่ภายใต้สัญญาอนุญาตสาธารณะทั่วไปของ GNU (GNU General Public License หรือ GPL)

รูป 2.1 Python

2.1.1 ลักษณะและการทำงานของภาษาไพธอน

ไพธอนถูกออกแบบมาให้มีโครงสร้างของภาษาที่ไม่ซับซ้อน มีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษาอื่น จะใช้คำหลักที่เป็นภาษาอังกฤษแทนการใช้เครื่องหมาย เมื่อนำไปใช้งานจึงเป็นภาษาที่เรียนรู้และเข้าใจได้ง่ายเหมาะกับผู้เริ่มต้นเขียนโปรแกรม ไพธอนยังมีความสะดวกในการทดสอบและแก้ไขข้อบกพร่องของโปรแกรม เนื่องจากมีการประมวลผลคำสั่งด้วยตัวแปลภาษา (interpreter) ขณะโปรแกรมทำงาน (Runtime) จึงไม่จำเป็นต้องทำการแปลโปรแกรม (compile) ก่อนการทำงาน

ไพธอนรองรับการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented) และมีความสามารถในการเขียนโปรแกรมได้หลายรูปแบบ เนื่องจากมีไลบรารีที่ครอบคลุมการทำงานที่หลากหลาย ใช้งานร่วมกันได้ทั้งบนระบบปฏิบัติการ UNIX Windows และ Macintosh การพัฒนาโปรแกรมร่วมกันกับภาษาซี ภาษาซีพลัสพลัส ภาษาคอม (COM หรือ Component Object Model) ภาษาแอกทีฟเอ็กซ์ (ActiveX) ภาษาคอร์บา (CORBA หรือ Common Object Request Broker Architecture) และภาษาจาวา (Java) ก็ทำได้ง่ายเช่นกัน ในส่วนของฮาร์ดแวร์ (Hardware) ไพธอนสามารถทำงานได้บนแพลตฟอร์ม (platform) ของฮาร์ดแวร์ที่หลากหลายและมีอินเตอร์เฟซเดียวกันบนทุกแพลตฟอร์มอีกด้วย

2.1.2 ไลบรารีของภาษาไพธอนที่นำมาใช้งาน

2.1.2.1 ไลบรารี Scrapy

Scrapy คือ แอปพลิเคชันเฟรมเวิร์ค (application framework) สำหรับการรวบรวมข้อมูลจากเว็บไซต์และการสกัดข้อมูล (extracting structured data) ที่สามารถนำไปใช้ประโยชน์ได้หลากหลาย เช่น การทำเหมืองข้อมูล (Data Mining) การประมวลผลข้อมูล (Information Processing) หรือ ทำที่เก็บประวัติ (Historical Archival) เป็นต้น

รูป 2.2 Scrapy

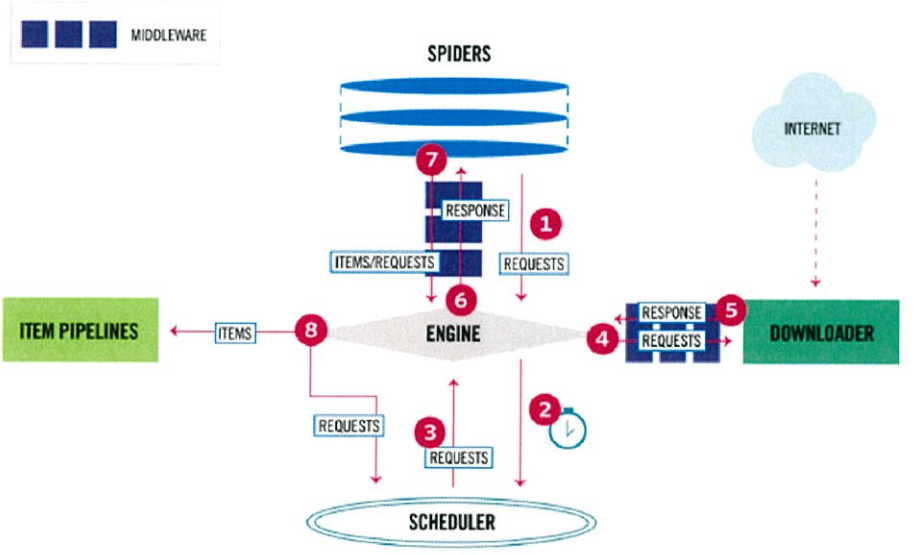
Scrapy เขียนขึ้นจากภาษาไพธอนทั้งหมด โดยมีการนำบางแพ็คเกจที่สำคัญมาใช้งาน ได้แก่

1) lxml นำมาใช้ในการประมวลผลเอชทีเอ็มแอลและเอ็กเอ็มแอล (XML)
2) parsel เป็นไลบรารีที่ใช้สำหรับดึงข้อมูลเอชทีเอ็มแอลและเอ็กเอ็มแอล
3) w3lib ใช้สำหรับจัดการกับยูอาร์แอลและการเข้ารหัสหน้าเว็บ
4) twisted เฟรมเวิร์คของเครือข่ายแบบอะซิงโครนัส (Asynchronous)
5) cryptography และ pyOpenSSL ใช้เพื่อจัดการความปลอดภัยระดับเครือข่ายต่าง ๆ ที่จำเป็น

โครงสร้างการทำงานของสแครปปี้จะแบ่งส่วนประกอบออกเป็น 7 ส่วน ได้แก่

1) เอนจิน (Scrapy Engine) เป็นส่วนที่รับผิดชอบควบคุมการทำงานระหว่างส่วนประกอบต่าง ๆ ภายในระบบ
2) ตัวจัดการลำดับ (Scheduler) จะรับคำร้องขอ (Request) จากเอนจินมาจัดลำดับและรอส่งกลับไปให้เมื่อเอนจินต้องการ
3) ตัวจัดการดาวน์โหลด (Downloader) ทำหน้าที่รับผิดชอบในการดาวน์โหลดหน้าเว็บและส่งกลับไปให้เอนจิน เพื่อให้เอนจินส่งให้สไปเดอร์
4) สไปเดอร์ (Spider) เป็นคลาส (class) ที่เขียนขึ้นจากผู้ใช้งาน Scrapy เพื่อใช้รวบรวมข้อมูลตามความต้องการของผู้ใช้งาน

5) ไอเทมไปป์ไลน์ (Item Pipeline) ทำหน้าที่ประมวลผลรายการข้อมูลที่ได้จากสไปเดอร์ รวมถึงการทำความสะอาดข้อมูล (Data cleansing) การตรวจสอบความถูกต้องของข้อมูล (Data validation) และการเก็บข้อมูลลงฐานข้อมูล
6) ตัวจัดการดาวน์โหลดมิดเดิลแวร์ (Downloader middlewares) ส่วนเชื่อมต่อระหว่างเอนจินกับตัวจัดการดาวน์โหลด ทำหน้าที่ส่งคำร้องขอไปยังตัวจัดการดาวน์โหลดและส่งคำตอบ (Response) กลับไปให้เอนจิน
7) สไปเดอร์มิดเดิลแวร์ (Spider middlewares) ส่วนเชื่อมต่อระหว่างเอนจินกับสไปเดอร์ ทำหน้าที่รับคำตอบเข้ามาและส่งออกไอเทมกับคำร้องขอ



รูป 2.3 โครงสร้างและขั้นตอนการทำงานของ Scrapy

การทำงานภายใน Scrapy จะมีเอนจินเป็นส่วนที่ทำหน้าที่ควบคุมกระบวนการการทำงานทั้งหมด ซึ่งแต่ละขั้นตอนมีการทำงานดังนี้

1) เอนจินได้รับคำร้องขอเพื่อเริ่มต้นรวบรวมข้อมูลจากสไปเดอร์
2) เอนจินกำหนดลำดับคำร้องขอในตัวจัดการลำดับและถามถึงคำร้องขอถัดไปที่จะทำการรวบรวมข้อมูล
3) ตัวจัดการลำดับจะส่งคำร้องขอถัดไปกลับไปให้เอนจิน
4) เอนจินส่งคำร้องขอที่ได้รับไปให้ตัวจัดการดาวน์โหลดผ่านทางตัวจัดการดาวน์โหลดมิดเดิลแวร์

5) เมื่อมีเพจที่ทำการดาวน์โหลดเสร็จสิ้น ตัวจัดการดาวน์โหลดจะสร้างคำตอบและตอบกลับไปที่เอนจินผ่านทางตัวจัดการดาวน์โหลดมิดเดิลแวร์
6) เอนจินได้รับคำตอบจากตัวจัดการดาวน์โหลดและส่งคำตอบที่ได้ผ่านทางสไปเดอร์มิดเดิลแวร์ไปให้สไปเดอร์เพื่อทำการประมวลผล
7) สไปเดอร์ทำการประมวลผลคำตอบที่ได้รับและส่งไอเทมที่ได้กับคำร้องขอ (ที่จะทำต่อ) กลับไปให้เอนจินผ่านทางสไปเดอร์มิดเดิลแวร์
8) เอนจินส่งไอเทมที่ผ่านการประมวลผลแล้วไปที่ไอเอมไปป์ไลน์ จากนั้นส่งคำร้องขอที่สำเร็จแล้วไปที่ตัวจัดการลำดับและถามถึงคำร้องขอถัดไปที่เป็นไปได้ที่จะเริ่มทำการรวบรวมข้อมูล
9) กระบวนการทำงานจะทำซ้ำตั้งแต่ข้อ 1 จนกระทั่งไม่มีคำร้องขอจากตัวจัดการลำดับ

2.1.2.2 ไลบรารี Selenium

ซีลีเนียม (Selenium) เป็นไลบรารีของเครื่องมือที่สามารถจำลองการทำงานของเว็บเบราว์เซอร์ มีลักษณะการทำงานโดยการเปิดเว็บเบราว์เซอร์ขึ้นมาและทำการดำเนินงานตามคำสั่งที่ผู้ใช้งานกำหนดโดยอัตโนมัติ เช่น การกด (click) การนำเข้า (input) การเลือก (select) การนำทาง (navigate) เป็นต้น ซึ่งการใช้งานเบื้องต้นทั่วไปจะนำไปใช้ในการทดสอบระบบอัตโนมัติสำหรับการทำงานของเว็บแอปพลิเคชัน แต่ก็สามารถทำงานแบบอื่นได้ตามแต่ผู้ใช้งานกำหนด

รูป 2.4 Selenium with Python

ซีลีเนียมได้รับความนิยมอย่างมากในปัจจุบันและมีผู้ใช้งานกันอย่างกว้างขวางเนื่องจากสามารถนำมาใช้งานได้ฟรี (open-source) มีความยืดหยุ่น สามารถเพิ่มเติมรูปแบบและลักษณะการทำงานได้ง่าย รองรับการทำงานร่วมกับภาษาอื่น ๆ ได้หลายภาษา ได้แก่ ภาษาไพธอน ภาษาจาวา ภาษาซีชาร์ป (C#) ภาษารูบี้ (Ruby) ภาษาจาวาสคริปต์ (Javascript) และสนับสนุนการทำงานกับหลายเว็บเบราว์เซอร์ที่เป็นที่นิยมในปัจจุบันอีกด้วย

2.1.2.3 กูเกิลเสิร์ชเอพีไอ

รูป 2.5 Google Search

กูเกิลเสิร์ชเอพีไอ (Google Search API) เป็นการนำกูเกิลเสิร์ชเอนจินมาใช้ในโปรแกรมภาษาไพธอน มีฟังก์ชันการทำงาน (Function) 2 ฟังก์ชัน ได้แก่

1) get_page(url) ใช้สำหรับร้องขอเว็บเพจที่ต้องการ โดยการกำหนดยูอาร์แอลลงในฟังก์ชัน
2) search(query, tld='com', lang='en', num=10, start=0, stop=None, pause=2.0) ใช้เพื่อค้นหาข้อความที่กำหนดให้ผ่านกูเกิลเสิร์ช โดยกำหนดข้อความที่จะทำการค้นหาลงในส่วนคิวรี (Query) และสามารถกำหนดค่าอื่น ๆ เพิ่มเติมได้ เช่น โดเมน ภาษา จำนวนผลลัพธ์ที่แสดงต่อหน้า เป็นต้น

2.2 ภาษาเอชทีเอ็มแอล

ภาษาเอชทีเอ็มแอล (HTML หรือ Hypertext Markup Language) เป็นภาษาที่นิยมใช้สำหรับงานเขียนเว็ปเพจกันอย่างกว้างขวาง ซึ่งคำว่าไฮเปอร์เท็กซ์ (Hypertext) อ้างอิงถึงวิธีที่เว็ปเพจมีการเชื่อมโยงกัน ลิงค์ที่ปรากฏอยู่ในแต่ละเว็ปเพจจึงเรียกว่าไฮเปอร์เท็กซ์ ส่วนคำว่าภาษามาร์กอัป (Markup Language) หมายถึงการใช้ภาษาเอชทีเอ็มแอลในการทำเครื่องหมายให้กับไฟล์เอชทีเอ็มแอลด้วยแท็กที่บอกกับเว็บเบราว์เซอร์ว่าจะจัดโครงสร้างของเว็ปให้แสดงอย่างไร

เริ่มต้นนั้น ภาษาเอชทีเอ็มแอลได้รับการพัฒนาขึ้นโดยมีเจตนาที่จะกำหนดโครงสร้างของไฟล์ เช่น ส่วนหัว การจัดย่อหน้า รายการและอื่น ๆ เพื่ออำนวยความสะดวกในการแบ่งปันข้อมูลทางวิทยาศาสตร์ระหว่างนักวิจัย แต่ในปัจจุบันภภาษาอชทีเอ็มแอลถูกใช้งานอย่างแพร่หลายในการ

จัดรูปแบบหน้าเว็บด้วยแท็กต่าง ๆ ที่มีอยู่ในภาษาเอชทีเอ็มแอลซึ่งในขณะนี้ เวอร์ชันล่าสุดของภาษาเอชทีเอ็มแอล คือ เอชทีเอ็มแอลห้า (HTML5) (รูป 2.5)

รูป 2.6 HTML5

2.2.1 โครงสร้างของไฟล์ในภาษาเอชทีเอ็มแอล

การเขียนภาษาเอชทีเอ็มแอล จะประกอบด้วยส่วนประกอบ 2 ส่วน ที่อยู่บริเวณภายในของ <html>.....</html> คือ

1) ส่วนหัว (Header) อยู่บริเวณระหว่าง <head> ... </head> คือส่วนที่จะทำหน้าที่เป็นหัวของหน้าเว็บเพจทั่วไป
2) ส่วนเนื้อหา (Body) อยู่บริเวณระหว่าง <body> ... </body> คือส่วนที่ทำหน้าที่เป็นเนื้อหาของหน้าเว็บเพจนั้น ๆ จะประกอบไปด้วยแท็กคำสั่งในภาษาเอชทีเอ็มแอล หรือส่วนที่ใช้ตกแต่งหน้าเว็บเพจ

ตัวอย่าง 2.1 โครงสร้างทั่วไปของไฟล์ภาษาเอชทีเอ็มแอล

```
<html>

   <head>
      Document header related tags
   </head>

   <body>
      Document body related tags
   </body>

</html>
```

2.2.2 คำสั่งในภาษาเอชทีเอ็มแอล

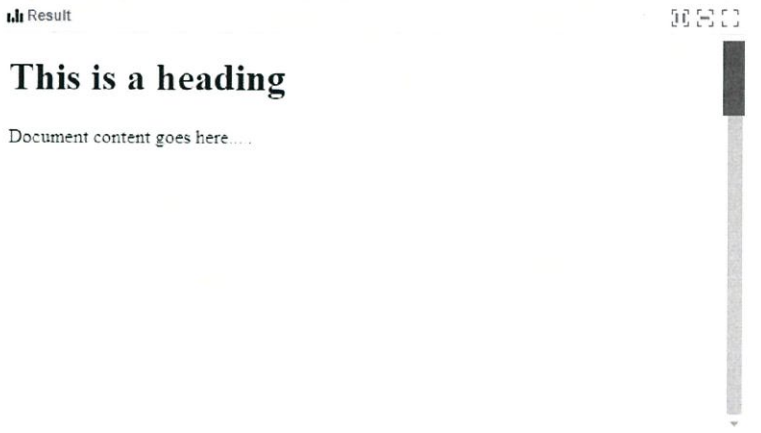
ในภาษาเอชทีเอ็มแอลจะใช้ประโยชน์จากแท็ก (Tag) หรือก็คือคำสั่งต่าง ๆ เพื่อจัดรูปแบบของเนื้อหาตรงส่วนโค้ด ชื่อของแท็กจะอยู่ภายในวงเล็บ ดังนี้ <ชื่อแท็ก> โดยส่วนใหญ่แท็กนั้น ๆ จะมีแท็กปิดที่เป็นชื่อเดียวกัน ยกเว้นในบางแท็กเท่านั้น

ตาราง 2.1 ตัวอย่างแท็กในภาษาเอชทีเอ็มแอล

แท็ก (Tag)	คำอธิบาย
<!DOCTYPE...>	เป็นแท็กที่กำหนดประเภทของไฟล์และเวอร์ชันของภาษาเอชทีเอ็มแอล
<html>	ไฟล์เอชทีเอ็มแอลที่มีความสมบูรณ์จะมีแท็กนี้ครอบเนื้อหาอยู่ ซึ่งจะมีแท็ก <head> ... </ head> และแท็ก <body> ... </ body> ที่ครอบเนื้อหาของไฟล์อยู่ภายใน <html> ... </html> อีกชั้นหนึ่ง
<head>	แท็กนี้แสดงถึงส่วนหัวของไฟล์ ซึ่งจะครอบแท็กอื่น ๆ ไว้ภายในอีก เช่น <title>, <link> เป็นต้น
<title>	เป็นแท็กที่กำหนดชื่อของไฟล์ จะใช้แท็กนี้ภายในแท็ก <head>
<body>	แท็กนี้แสดงถึงส่วนเนื้อหาของไฟล์ ซึ่งจะครอบแท็กอื่น ๆ ไว้ภายในอีก เช่น <h1>, , <p> เป็นต้น
<h1>	เป็นแท็กที่แสดงถึงหัวเรื่อง
<p>	เป็นแท็กที่แสดงถึงการจัดย่อหน้า

โปรแกรม 2.1 ตัวอย่างโปรแกรมพื้นฐานในรูปแบบที่ง่ายของภาษาเอชทีเอ็มแอล

```
<!DOCTYPE html>
<html>
   <head>
      <title>This is document title</title>
   </head>
   <body>
      <h1>This is a heading</h1>
      <p>Document content goes here.....</p>
   </body>
</html>
```



รูป 2.7 ผลลัพธ์การแสดงหน้าเว็บจาก (โปรแกรม 2.1)

2.3 ภาษาซีเอสเอส

ภาษาซีเอสเอส (CSS หรือ Cascading Style Sheet) เป็นภาษาที่ใช้สำหรับจัดรูปแบบในการแสดงผลของไฟล์ด้วยการกำหนดคุณสมบัติให้กับส่วนประกอบต่าง ๆ ของภาษาเอชทีเอ็มแอล เช่น ลักษณะเส้น สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร การจัดวางข้อความ ระยะห่างหรืออื่น ๆ โดยมีรูปแบบการเขียนวากยสัมพันธ์ (Syntax) ที่เฉพาะ ซึ่งถูกกำหนดมาตรฐานโดย W3C (World Wide Web Consortium) เช่นเดียวกับภาษาเอชทีเอ็มแอล

รูป 2.8 CSS

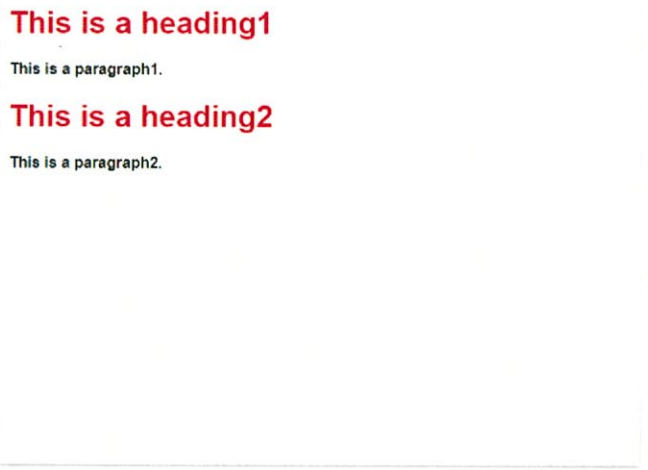
การกำหนดรูปแบบหรือสไตล์ (Style) ในภาษาซีเอสเอสนั้นใช้หลักการของการแยกเนื้อหาออกจากคำสั่งที่ใช้ในการจัดรูปแบบสำหรับการแสดงผลของภาษาเอชทีเอ็มแอล ซึ่งกำหนดให้รูปแบบของการแสดงผลในไฟล์ภาษาเอชทีเอ็มแอลไม่ขึ้นอยู่กับเนื้อหา เพื่อให้ง่ายต่อการจัดรูปแบบให้มีลักษณะเดียวกันทั้งเว็บไซต์หรือแก้ไขลักษณะต่าง ๆ ของเนื้อหาที่อาจมีการแก้ไขบ่อยครั้ง เนื่องจากการแก้ไขคำสั่งของภาษาซีเอสเอสเพียงจุดเดียวก็จะมีผลกับไฟล์ทั้งหมด ไม่ต้องตามแก้ไขตามแท็กต่าง ๆ ทีละจุดในไฟล์ภาษาเอชทีเอ็มแอล

โปรแกรม 2.2 การกำหนดลักษณะโดยไม่ใช้ภาษาซีเอสเอส

```
<html>
     <head>
          <title>This is document title</title>
     </head>
     <body>
          <h1><font color="red" face="Arial">
               This is a heading1</font>
          </h1>
          <p><font color="black" face="Arial"><b>
               This is a paragraph1. </b></font>
          </p>
          <h1><font color="red" face="Arial">
               This is a heading2</font>
          </h1>
          <p><font color="black" face="Arial"><b>
               This is a paragraph2. </b></font>
          </p>
     </body>
</html>
```

โปรแกรม 2.3 การกำหนดลักษณะโดยใช้ภาษาซีเอสเอส

```
<html>
  <head>
    <title>This is document title</title>
    <style type="text/css">
        h1{
            color: red;
            font-family: Arial;
            }
        p {
            color: black;
            font-family: Arial;
            font-weight: bold;
            }
    </style>
  </head>
  <body>
    <h1>This is a heading1</h1>
    <p>This is a paragraph1. </p>

    <h1>This is a heading2</h1>
    <p>This is a paragraph2. </p>
  </body>
</html>
```



รูป 2.9 ผลลัพธ์การแสดงหน้าเว็บจาก (โปรแกรม 2.2, โปรแกรม 2.3)

จากตัวอย่างของโปรแกรมและผลลัพธ์ข้างต้น (รูป 2.9) จะเห็นได้ว่าการใช้ภาษาซีเอสเอส (โปรแกรม 2.2) เพื่อแก้ไขรูปแบบของเนื้อหาในหน้าเว็บไซต์นั้นใช้งานได้ง่าย รวดเร็ว และมีความเรียบร้อยกว่าโปรแกรมที่ไม่ใช้ภาษาซีเอสเอส (โปรแกรม 2.3) เพราะสามารถแก้ไขที่สไตล์ในส่วนของส่วนหัวเพียงจุดเดียว แทนที่จะแก้ไขตามแท็กทีละจุดซึ่งอยู่ในส่วนของส่วนเนื้อหา

2.4 ภาษาจาวาสคริปต์

จาวาสคริปต์ (JavaScript) (รูป 2.10) เป็นภาษาสคริปต์เชิงวัตถุ ใช้ร่วมกับภาษาเอชทีเอ็มแอล เพื่อให้เว็บไซต์มีการเคลื่อนไหว ตอบสนองต่อผู้ใช้งานและมีความน่าสนใจมากยิ่งขึ้น การทำงานของภาษาจาวาสคริปต์นั้นมีวิธีการทำงานในลักษณะการแปลความและดำเนินงานทีละคำสั่ง (interpret) หรือเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming หรือ OOP)

รูป 2.10 JavaScript

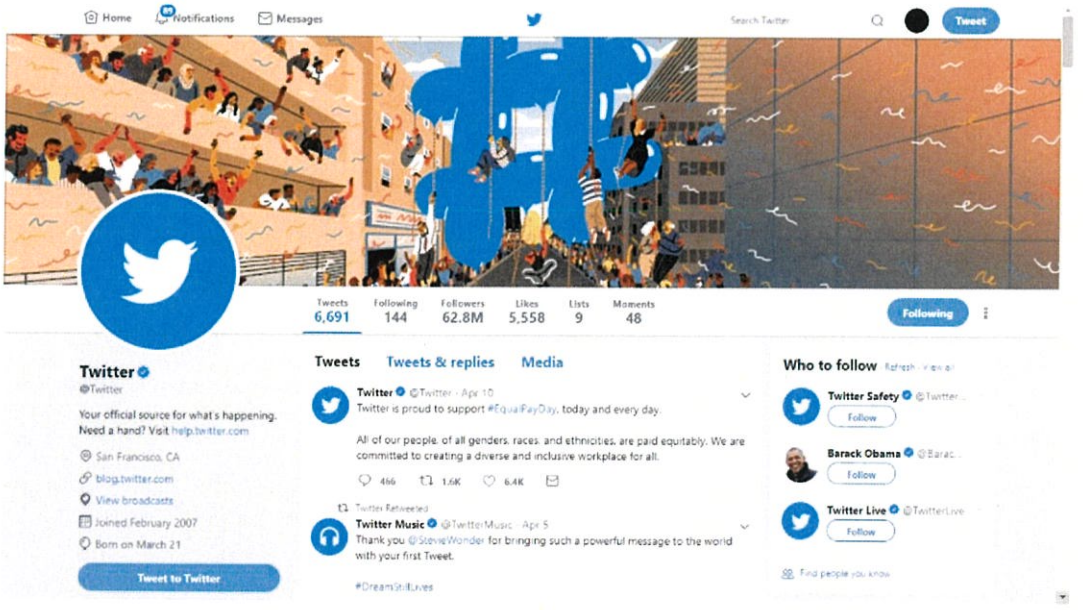
ในปัจจุบันมีการใช้งานภาษาจาวาสคริปต์กันอย่างกว้างขวาง เนื่องจากเป็นภาษาที่สามารถนำไปใช้ได้โดยไม่เสียค่าใช้จ่าย และการแปลความของภาษาจาวาสคริปต์นั้นจะทำโดยสคริปต์

(client-side script) ภายในเว็บเบราว์เซอร์ (Web browser) ที่รองรับภาษา ซึ่งเว็บเบราว์เซอร์ส่วนใหญ่เกือบทั้งหมดในปัจจุบันนั้นได้รองรับภาษาจาวาสคริปต์แล้ว

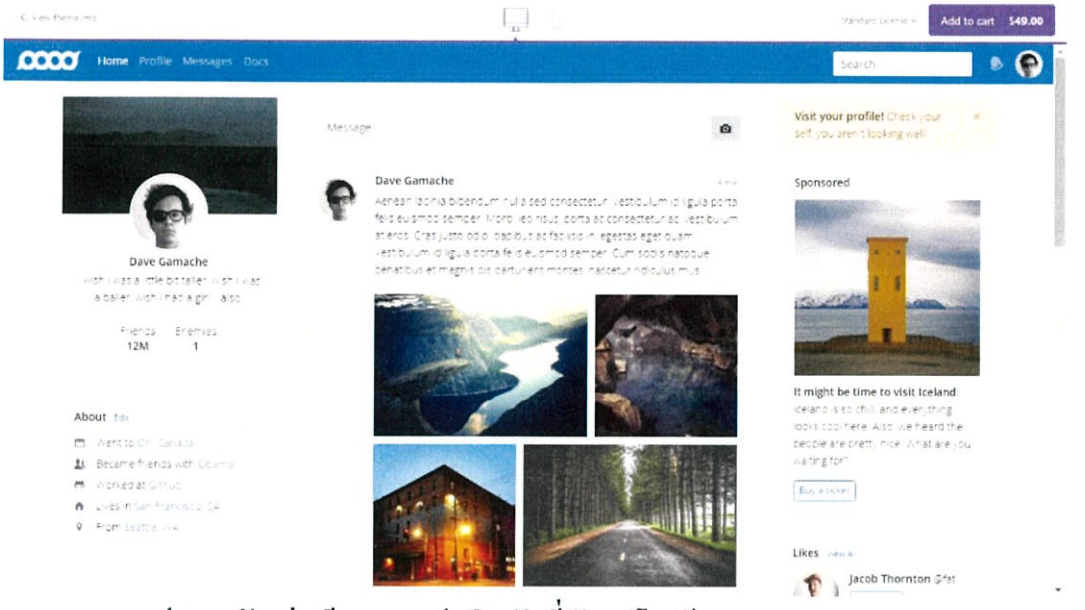
2.5 Bootstrap

รูป 2.11 Bootstrap

Bootstrap (รูป 2.11) เป็นเฟรมเวิร์คทางส่วนหน้าบ้าน (Front-end Framework) ที่ได้รวมภาษาเอชทีเอ็มแอล ภาษาซีเอสเอสและภาษาจาวาสคริปต์เข้าด้วยกัน ใช้สำหรับพัฒนาเว็บไซต์ที่เขียนขึ้นเพียงครั้งเดียวก็สามารถรองรับการทำงานและแสดงผลกับทุกสมาร์ท ดีไวซ์ (Smart Device) หรือเรียกว่า Responsive Web โดย Bootstrap นั้นถูกพัฒนาขึ้นโดยทีมงานจากทวิตเตอร์ (Twitter) (รูป 2.12) ซึ่งจะเห็นได้ว่ามีรูปแบบของเว็บไซต์ที่คล้ายคลึงกันมาก (รูป 2.13)



รูป 2.12 หน้าหลักของทวิตเตอร์

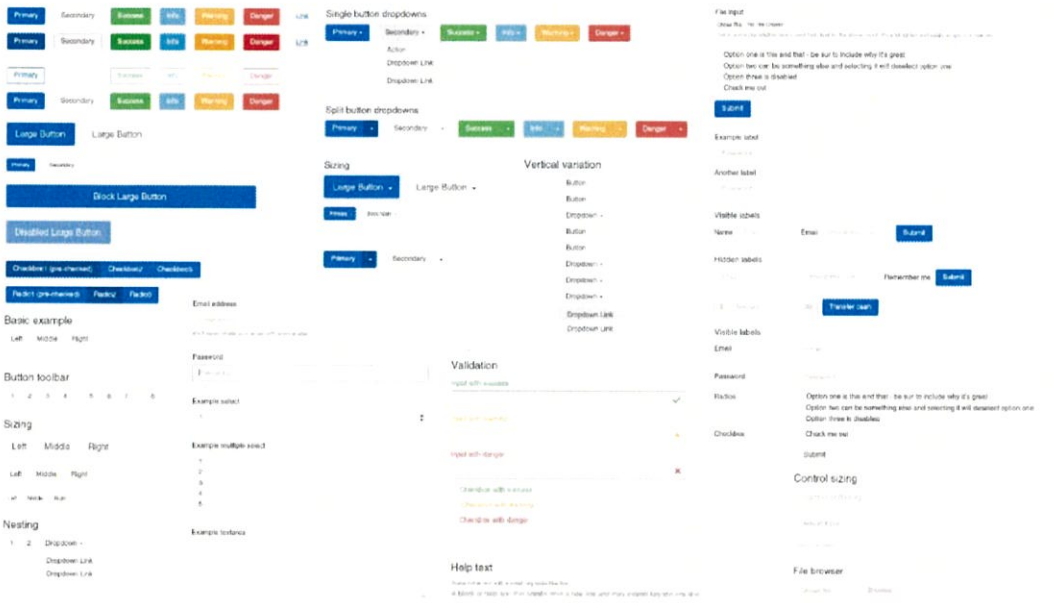


รูป 2.13 ตัวอย่างธีมของแอปพลิเคชันที่พัฒนาโดยทีมงานของ Bootstrap

สมัยก่อนในช่วงที่สมาร์ท ดีไวซ์ยังไม่ถูกพัฒนาไปไกล การออกแบบเว็บไซต์เพื่อให้ตอบสนองต่อการทำงานในทุก ๆ อุปกรณ์จึงยังไม่จำเป็นมาก ซึ่งต่างกับในปัจจุบันที่มีการนำสมาร์ท ดีไวซ์มาใช้งานกันอย่างแพร่หลาย ผู้คนส่วนใหญ่เปิดดูเว็บไซต์จากสมาร์ทโฟน (Smart Phone) เป็นหลัก การออกแบบและพัฒนาเว็บไซต์ให้ตอบสนองต่อหน้าจอของอุปกรณ์ที่มีความหลากหลายจึงมีความยาก แต่ก็จำเป็นอย่างมากเช่นกัน

ด้วยเหตุนี้ ทวิตเตอร์จึงได้พัฒนา Bootstrap ขึ้นมา เพื่อตอบโจทย์ในด้านการออกแบบเว็บไซต์ที่สามารถรองรับการทำงานบนหน้าจออุปกรณ์ได้หลากหลาย (Responsive Web Design) โดยเฉพาะระบบกริด (Grid) (รูป 2.14) ที่มีอยู่ใน Bootstrap ซึ่งมีการคำนวนค่าอัตราส่วนหน้าจอของอุปกรณ์พร้อมกับช่วยปรับขนาดของเว็บไซต์ให้แสดงผลกับทุก ๆ หน้าจอโดยอัตโนมัติ ซึ่งผู้พัฒนาเว็บไซต์สามารถปรับแต่งให้แต่ละหน้าจอมีการแสดงผลที่ต่างกันได้ตามลักษณะและขนาดของหน้าจอ

รูป 2.14 ระบบกริดพื้นฐานใน Bootstrap



รูป 2.15 Bootstrap UI

นอกจากนี้ Bootstrap ยังมีส่วนต่อประสานกับผู้ใช้ (รูป 2.15) ที่ออกแบบมาได้อย่างสวยงาม และมีความทันสมัย ให้ผู้พัฒนาสามารถเลือกใช้ได้หลากหลายตามความต้องการ ความสามารถที่มากมายนี้เองทำให้ Bootstrap ได้รับความนิยมอย่างกว้างขวางในส่วนของการพัฒนาด้านหน้าบ้าน เพราะมีเครื่องมือที่พร้อมสนับสนุนการทำงานและมีรูปแบบที่สามารถทำความเข้าใจและใช้งานได้ง่าย

2.6 Django

แจงโก้ (Django) (รูป 2.16) เป็นเฟรมเวิร์คที่พัฒนาขึ้นจากภาษาไพธอนเพื่อใช้ในการสร้างเว็บแอปพลิเคชันในฝั่งของหลังบ้าน (Backend) โดยระบบภายในของ Django นั้นมีการทำงานทุกอย่าง

ที่จำเป็นตั้งแต่การเชื่อมต่อฐานข้อมูลไปจนถึงการเร็นเดอร์ข้อมูล (render) ออกมาเพื่อส่งให้ทางฝั่งหน้าบ้านแสดงผลข้อมูล

รูป 2.16 Django

การทำงานของ Django ได้แบ่งออกเป็นส่วน ๆ แยกระหว่างส่วนออกแบบ (Design) และตรรกะ (Logic) (ตัวอย่าง 2.2) มีการทำงานแบบ Object-relational mapper คือการกำหนดโครงสร้างของข้อมูล (Data Model) ในภาษาไพธอน ซึ่งเป็นส่วนของการทำงานที่เกี่ยวข้องกับฐานข้อมูลและสนับสนุน dynamic database-access API

ตัวอย่าง 2.2 โครงสร้างโปรเจคของ Django

```
mysite/
     manage.py
     mysite/
          __init__.py
          settings.py
          urls.py
          wsgi.py
     home/
          __init__.py
          admin.py
          apps.py
          migrations/
               __init__.py
          models.py
          tests.py
          views.py
```

นอกจากแจงโก้เฟรมเวิร์คจะมีระบบการทำงานที่แบ่งเป็นสัดส่วนแล้ว ยังมีระบบของแคช (Cache system) ที่ทำหน้าที่บันทึกหรือก็คือการจัดการเก็บข้อมูลที่มีการดาวน์โหลดไปแล้ว เพื่อเพิ่มประสิทธิภาพการทำงานของเว็บไซต์ในด้านความเร็วได้อีกด้วย

2.7 MongoDB

รูป 2.17 MongoDB

มองโกดีบี (MongoDB) (รูป 2.17) เป็นฐานข้อมูลแบบ No-SQL หรือฐานข้อมูลแบบไม่มีความสัมพันธ์ของโครงตาราง (Non-Relation) ซึ่งเป็นคนละชนิดกับฐานข้อมูลแบบเอสคิวแอลทั่วไป โดยที่ mongoDB จะเก็บข้อมูลเป็นคีย์ (key) หรือฟิลด์ (field) และค่า (value) หรือเรียกว่าเจสัน (JSON หรือ JavaScript Object Notation) (รูป 2.18) แทนการเก็บข้อมูลเป็นรายการ (record) ในตาราง ซึ่งใน mongoDB จะเรียกการเก็บข้อมูลแบบเจสันว่าเอกสาร (Document) และจะถูกเก็บไว้ในคอลเลกชัน (Collection)

```
{
  name: "sue",                          ←— field: value
  age: 26,                              ←— field: value
  status: "A",                          ←— field: value
  groups: [ "news", "sports" ]          ←— field: value
}
```

รูป 2.18 โครงสร้างเอกสารใน MongoDB

MongoDB เป็นฐานข้อมูลแบบไม่มีความสัมพันธ์ของโครงตาราง ประเภทฐานข้อมูลแบบเอกสาร (Document Database) ที่มีความสามารถในการยืดหยุ่นและปรับขนาดได้ตามที่ต้องการด้วยการทำคิวรีและอินเด็กซ์ (Index) ทำให้ MongoDB มีคุณสมบัติที่น่าสนใจดังนี้

- MongoDB มีความยืดหยุ่นในการจัดเก็บข้อมูลเช่นเดียวกับไฟล์ประเภทเจสัน คือ ฟิลด์ของข้อมูลในแต่ละเอกสารไม่จำเป็นต้องเหมือนกัน เพราะว่าโครงตาราง (schema) ของข้อมูลสามารถเปลี่ยนแปลงได้ตลอดเวลา
- โครงสร้างของเอกสารจะจับคู่กับออบเจ็กต์ (Object) ในโค้ดของแอปพลิเคชันและยังมีการสืบค้นข้อมูลทันที (Ad hoc queries) การจัดทำดรรชนี (indexing) และ real time aggregation ช่วยทำให้เข้าถึงข้อมูลและใช้งานข้อมูลได้ง่ายมากยิ่งขึ้น
- MongoDB สามารถใช้งานฐานข้อมูลแบบกระจาย (distributed database) เป็นหลักได้โดยใช้ได้ทั้งแบบ horizontal scaling และแบบ geographic distribution ซึ่งใช้งานได้ง่ายและมีความพร้อมสำหรับการใช้งาน

- MongoDB สามารถใช้งานได้โดยไม่มีค่าใช้จ่าย เพราะเป็นซอฟท์แวร์ฟรีซึ่งอยู่ในการควบคุมของ GNU Affero General Public License

2.8 Mongoengine

รูป 2.19 Mongoengine

มองโกเอนจิน (MongoEngine) (รูป 2.19) คือ ตัวแปลงออบเจ็กต์ในภาษาจาวาสคริปต์ให้เป็นเอกสารใน MongoDB (Object-Document Mapper) ที่เขียนขึ้นจากภาษาไพธอนเพื่อใช้สำหรับการทำงานร่วมกับ MongoDB ซึ่งสามารถกำหนดโครงตารางและตรวจสอบความถูกต้องสำหรับเอกสารที่จะจัดเก็บลง MongoDB เพื่อให้ข้อมูลที่จะทำการจัดเก็บมีโครงสร้างตามที่ผู้ใช้ต้องการ ช่วยลดปัญหาในการทำงาน เช่น การคิวรีข้อมูลจากฐานข้อมูล เป็นต้น สามารถทำได้โดยการสร้างคลาสของเอกสารและทำการกำหนดฟิลด์ของออบเจ็กต์ลงภายในคลาส (โปรแกรม 2.4) ซึ่งเทียบได้กับโครงสร้างเอกสารในรูปแบบของเจสัน (ตัวอย่าง 2.3) ที่เก็บลงใน MongoDB

โปรแกรม 2.4 ตัวอย่างการสร้างคลาสของเอกสารและกำหนดฟิลด์ของออบเจ็กต์

```
from mongoengine import *

class Metadata(EmbeddedDocument):
     tags = ListField(StringField())
     revisions = ListField(IntField())

class WikiPage(Document):
     title = StringField(required=True)
     text = StringField()
     metadata = EmbeddedDocumentField(Metadata)
```

ตัวอย่าง 2.3 โครงสร้างเอกสารในรูปแบบของเจสัน

```
{
     title: String,
     text: String,
     metadata: [
          {
```

ตัวอย่าง 2.3 โครงสร้างเอกสารในรูปแบบของเจสัน (ต่อ)

```
                tags: [String],
                revisions: [Int]
            }
        ]
}
```

การเรียกคิวรีข้อมูลจาก MongoEngine นั้นจะทำผ่าน QuerySetManager เนื่องจากจะแปลงเอกสารใน MongoDB ให้อยู่ในรูปแบบของออบเจ็กต์ โดยจะสร้างและคืนค่าเป็นออบเจ็กต์เซตของการคิวรี (QuerySet object) จากฐานข้อมูล การคิวรีข้อมูลนั้นจะมี Query operator ที่ใช้กับ MongoEngine โดยเฉพาะ เช่น หากต้องการผู้ใช้งานที่มีอายุน้อยกว่าหรือเท่ากับ 18 ปี จะใช้ lte หรือ less than or equal to (ตัวอย่าง 2.4) เป็นต้น

ตัวอย่าง 2.4 การคิวรีด้วย operator lte โดย MongoEngine

```
# Only find users whose age is 18 or less
young_users = Users.objects(age__lte=18)
```

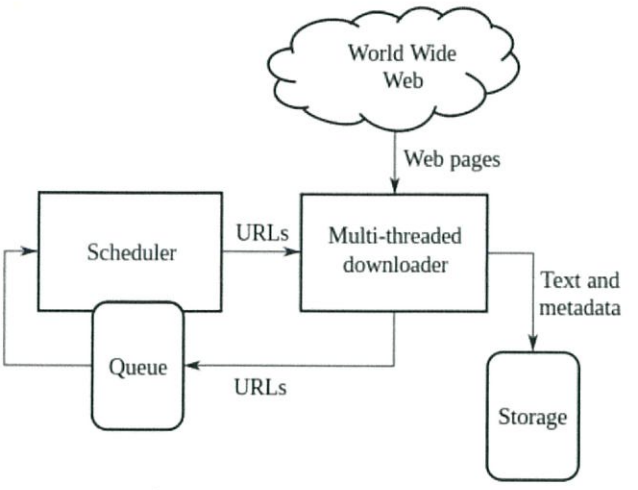
2.9 Search Engine

เสิร์ชเอนจิน (Search Engine) หรือเว็บเสิร์ชเอนจิน (Web search engine) เป็นระบบของซอฟท์แวร์ที่ถูกออกแบบมาเพื่อการค้นหาข้อมูลบนเวิลด์ไวด์เว็บ (WWW หรือ World Wide Web) ซึ่งผลลัพธ์ของการค้นหาโดยทั่วไปจะถูกแสดงในหน้าผลการค้นหา (Search engine results page หรือ SERPs) ข้อมูลที่พบจากเว็บเพจอาจประกอบไปด้วยข้อความ รูปภาพและไฟล์ชนิดต่าง ๆ เสิร์ชเอนจินในบางผู้ให้บริการอาจมีข้อมูลในฐานข้อมูลหรือไดเรกทอรีแบบเปิด มีการเก็บรักษาข้อมูลแบบเรียลไทม์โดยใช้การทำงานของอัลกอริทึมที่อยู่บนเว็บครอว์เลอร์ ซึ่งแตกต่างจากเว็บไดเรกทอรีที่ดูแลโดยมนุษย์เพียงอย่างเดียว

2.9.1 การทำงานของเว็บเสิร์ชเอนจิน

เสิร์ชเอนจินมีการทำงานที่ใกล้เคียงกับเวลาจริงตามขั้นตอนการทำงานดังต่อไปนี้

1) การรวบรวมข้อมูลจากเว็บ (Web crawling)
2) การจัดทำดรรชนี (Indexing)
3) การค้นหา (Searching)



รูป 2.20 โครงสร้างของโปรแกรมรวบรวมข้อมูลเว็บมาตรฐาน

เว็บเสิร์ชเอนจินจะเก็บรวบรวมข้อมูลจากไซต์หนึ่งไปยังอีกไซต์หนึ่ง โดยสไปเดอร์จะทำการตรวจสอบไฟล์มาตรฐานที่ชื่อว่า robots.txt ว่ามีไฟล์อยู่ที่ไซต์นั้น ๆ ก่อนที่จะส่งข้อมูลบางอย่างเช่น ชื่อ เนื้อหาของเพจ จาวาสคริปต์ ซีเอสเอสและส่วนหัว เป็นต้น ส่งกลับไปเพื่อจัดทำดรรชนีเพื่อเป็นหลักฐานของเอชทีเอ็มแอลมาตรฐานของเนื้อหาที่ให้ข้อมูล หรือเมตาดาต้าในแท็กเมตาของเอชทีเอ็มแอล

ส่วนของการจัดทำดรรชนี เป็นการเชื่อมโยงคำและโทเคน (token) ที่ระบุได้ที่พบบนหน้าเว็บไปยังชื่อโดเมนและส่วนที่เป็นเอชทีเอ็มแอล การเชื่อมโยงจะทำในฐานข้อมูลสาธารณะที่มีไว้สำหรับการใช้คำสำคัญเพื่อค้นหาเว็บ ซึ่งคำสำคัญสำหรับการค้นหาจากผู้ใช้งานอาจเป็นคำเดียวหรือเป็นประโยคก็ได้ ทั้งนี้ ดรรชนีจะช่วยให้การค้นหาข้อมูลที่มีความสัมพันธ์กับคำสำคัญถูกค้นหาพบได้ตรงตามความต้องการและรวดเร็วที่สุดเท่าที่จะเป็นไปได้

ในปัจจุบัน เสิร์ชเอนจิ้น เช่น กูเกิล (Google) มีแนวคิดที่ทำให้ผู้ใช้สามารถค้นหาข้อมูลจากบนเว็บไซต์ได้ตรงความต้องการ คือการใช้อัลกอริทึมที่ชื่อว่าเพจแรงค์ (PageRank) เพื่อจัดลำดับผลการค้นหาที่ได้จากสไปเดอร์ โดยมีปัจจัยที่ใช้กำหนดลำดับของเว็บเพจ คือ

1) จำนวนครั้งและที่ที่พบคีย์เวิร์ดในแต่ละเว็บเพจ
2) ระยะเวลาของเว็บเพจตั้งแต่เริ่มสร้างจนถึงปัจจุบัน
3) จำนวนของเว็บเพจที่เชื่อมต่อมายังเพจที่มีความเกี่ยวข้อง

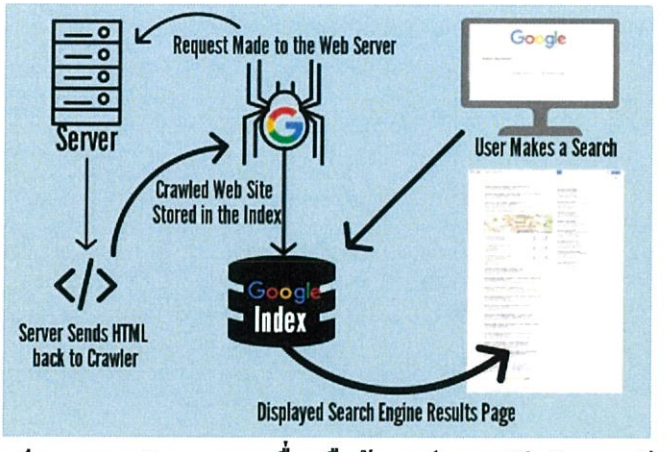
2.9.2 ประเภทของเครื่องมือค้นหา

เครื่องมือค้นหาแต่ละประเภทมีวิธีการจัดทำดรรชนีและการจัดเก็บฐานข้อมูลที่แตกต่างกันในแต่ละประเภทของเครื่องมือค้นหา แต่ละเว็บไซต์ที่เลือกวิธีการรวบรวมข้อมูลมาใช้จึงมีความแตกต่างกันไป โดยส่วนใหญ่จะเลือกใช้วิธีการทำด้วยโปรแกรมคอมพิวเตอร์หรือการเลือกใช้วิธีการที่ทำการรวบรวมขึ้นด้วยมนุษย์ อาจมีบางเครื่องมือค้นหาที่ใช้ร่วมกันทั้งสองวิธี ด้วย

เหตุนี้ การที่จะทำการรวบรวมข้อมูลจากเว็บไซต์โดยใช้วิธีการค้นหานั้น จึงควรรู้ประเภทที่แต่ละเว็บไซต์เลือกใช้ เนื่องจากเครื่องมือแต่ละประเภทมีวิธีและขั้นตอนการจัดการที่แตกต่างกัน ซึ่งสามารถจำแนกประเภทของเครื่องมือค้นหาได้ดังต่อไปนี้

2.9.2.1 เครื่องมือค้นหาประเภทจัดทำดรรชนี

เครื่องมือค้นหาประเภทจัดทำดรรชนี (Keyword search engines หรือ Index search engines) เป็นเครื่องมือค้นหาที่จัดทำดรรชนีและฐานข้อมูลแบบอัตโนมัติด้วยคอมพิวเตอร์ โดยจะทำการสำรวจแต่ละเว็บไซต์เพื่อค้นหาและรวบรวมข้อมูลมาทำเป็นดรรชนีแล้วจัดเก็บลงฐานข้อมูล ลักษณะของเครื่องมือค้นหาในแต่ละผู้ให้บริการจะแตกต่างกันไป เช่น ขนาดของฐานข้อมูล เนื้อหา ความถี่ในการเพิ่มข้อมูล การใช้งานและลักษณะความสามารถเฉพาะตัว เป็นต้น แต่การทำงานทั่วไปของเครื่องมือค้นหาประเภทนี้ คือ จะนำคำสำคัญที่ถูกกำหนดจากผู้ใช้งานมาเปรียบเทียบกับคำที่ถูกจัดทำเป็นดรรชนีในฐานข้อมูลว่ามีความคล้ายคลึงมากหรือน้อยเพียงใด จากนั้นจะทำการจัดลำดับผลการค้นหาแล้วแสดงผล



รูป 2.21 การทำงานของเครื่องมือค้นหาประเภทจัดทำดรรชนี

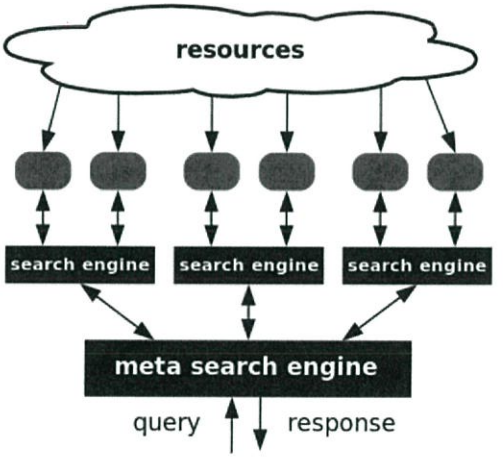
2.9.2.2 เครื่องมือค้นหาประเภทนามานุกรม

เครื่องมือค้นหาประเภทนามานุกรม (Subject directories search engines หรือ Web directory) เป็นเครื่องมือการค้นหาที่ทำการรวบรวมข้อมูลจากเว็บมาจัดทำเป็นฐานข้อมูลขึ้นโดยมนุษย์ ในการจัดทำจะมีการประเมินความน่าเชื่อถือของข้อมูล จากนั้นจะทำการแบ่งหัวข้อและสรุปเนื้อหาของข้อมูล มีการจัดเรียงเนื้อหาให้เป็นหมวดหมู่ ทำให้เห็นความสัมพันธ์หรือโครงสร้างของข้อมูลได้ชัดเจน จึงง่ายต่อการค้นหาแม้ผู้ใช้งานไม่มีคำถามที่แน่ชัด เครื่องมือค้นหาประเภทนี้จะมีปริมาณและขนาดของฐานข้อมูลที่เล็กกว่าประเภทแรกเป็นอย่างมาก เนื่องจากมีขั้นตอนและใช้เวลาในการประเมินมาก นอกจากนี้ยังสามารถทำการค้นหาแบบใช้คำสำคัญได้ แต่

ต่างกันตรงที่จะค้นหาจากคำสำคัญที่ถูกประเมินเอาไว้แล้ว ไม่ใช่ค้นหาจากคำที่ค้นเจอในเอกสาร เช่นในเครื่องมือค้นหาประเภทแรก

2.9.2.3 เครื่องมือค้นหาประเภทเมต้า

เครื่องมือค้นหาประเภทเมต้า (Meta search engines หรือ Multi search engines) หลักการในการค้นหานั้นจะค้นหาโดยใช้แท็กเมต้าในภาษาเอชทีเอ็มแอลที่ได้มีการรวบรวมข้อมูลไว้แล้วของเครื่องมือแต่ละประเภท เพราะเครื่องมือค้นหาประเภทนี้ไม่มีฐานข้อมูลเป็นของตนเอง เมื่อทำการค้นหาจึงให้แต่ละเครื่องมือทำการค้นหาแล้วส่งผลลัพธ์กลับมา ผลการค้นหาที่ได้จะนำมาสรุปผลและทำการเรียงใหม่ ข้อดีของเครื่องมือค้นหาประเภทเมต้า คือ ความเร็วในการค้นหา เพราะใช้หลายเครื่องมือในการค้นหาในเวลาเดียวกัน แต่การจัดลำดับการค้นหา การลบข้อมูลที่มีความซ้ำซ้อนกัน และการค้นหาแบบซับซ้อนจะแตกต่างจากเครื่องมือประเภทอื่น



รูป 2.22 โครงสร้างของเครื่องมือการค้นหาประเภทเมต้า

บทที่ 3

การออกแบบและพัฒนา

ในปัจจุบัน ข้อมูลในชีวิตประจำวันมีจำนวนมหาศาล การค้นหาข้อมูลจึงสามารถทำได้หลายช่องทาง เช่น หนังสือพิมพ์ โทรทัศน์ สื่อสังคมออนไลน์หรืออินเตอร์เน็ต ในขณะที่ขอบเขตของข้อมูลบางอย่างอาจจำกัดอยู่เพียงเฉพาะบุคคลหรืออยู่แค่ภายในองค์กร การสืบค้นหรือรวบรวมข้อมูลทั่วไปที่เป็นที่นิยมนำมาใช้งานกันอย่างแพร่หลายในปัจจุบัน คือ การค้นหาข้อมูลผ่านอินเตอร์เน็ตโดยใช้เสิร์ชเอ็นจิน การใช้เสิร์ชเอ็นจินจากผู้ให้บริการอย่างกูเกิล (Google) จึงเป็นทางเลือกแรก ๆ ของการสืบค้นข้อมูลในยุคนี้ เพราะไม่มีการจำกัดสิทธิ์ในการใช้งาน ทำให้มีความสะดวกและรวดเร็วในการค้นหา พบผลลัพธ์ได้หลากหลายมุมมองและไม่มีค่าใช้จ่ายในการใช้งาน

หลักการทำงานในการรวบรวมเพจที่มีความเกี่ยวข้องกับคีย์เวิร์ด ทั้งกูเกิลเสิร์ชหรือเสิร์ชเอนจินจากผู้ให้บริการอื่นก็มีหลักการเดียวกัน คือการใช้สไปเดอร์หรือครอว์เลอร์ไปทำการเก็บรวบรวมข้อมูลอินเด็กซ์ของคีย์เวิร์ดขนาดใหญ่และที่ที่สามารถพบคีย์เวิร์ดนั้น ๆ ได้มาแสดงเป็นผลการค้นหาตามที่ผู้ใช้งานต้องการ แต่ผลลัพธ์ของข้อมูลที่พบอาจมีเนื้อหาไม่ตรงหรือมีความซ้ำซ้อน ไม่สามารถนำไปใช้งานได้โดยตรง การนำข้อมูลที่รวบรวมได้มาทำการคัดกรองจึงเป็นสิ่งที่จำเป็น ในทางตรงกันข้าม หากว่าข้อมูลที่รวบรวมมาได้นั้นผ่านการคัดกรองข้อมูลมาแล้วและถูกจัดอยู่ในรูปแบบหรือโครงสร้างของข้อมูลที่เหมาะสม สามารถนำไปใช้งานได้ทันที จะทำให้สามารถเพิ่มความสะดวกในการทำไปใช้งานได้มากยิ่งขึ้น

3.1 ภาพรวมของระบบ

เว็บแอปพลิเคชันที่สามารถแสดงผลลัพธ์การเก็บรวบรวมข้อมูลจากบอทในรูปแบบของตารางและกราฟ ภายในเป็นระบบการเก็บรวบรวมข้อมูลโดยใช้บอทที่สร้างขึ้นจากภาษาไพธอน มีการใช้กูเกิลเสิร์ชเอนจินเป็นเครื่องมือสำหรับค้นหายูอาร์แอลเพื่อกำหนดเว็บไซต์ปลายทางที่ไปทำการเก็บข้อมูลให้แก่บอท ซึ่งบอทจะต้องทำงานร่วมกับเว็บเซิร์ฟเวอร์ที่สร้างจากแจงโก้เฟรมเวิร์ค จากนั้นข้อมูลจากการทำงานของบอทจะถูกเก็บลงในฐานข้อมูลมองโกดีบีซึ่งเป็นฐานข้อมูลแบบไม่มีความสัมพันธ์กับโครงตาราง (Non-Relation) นอกจากนั้นยังสามารถทำการส่งออกข้อมูลในรูปแบบของไฟล์ซีเอสวีเพื่ออำนวยความสะดวกในการนำไปใช้งานต่อตามความต้องการ

เนื่องจากข้อมูลปัจจุบันภายในอินเตอร์เน็ตมีมากมายหลายประเภท ในโครงงานชิ้นนี้จึงเน้นไปที่การเก็บข้อมูลประเภทของราคา โดยเลือกเป็นราคาของโรงแรม เพราะมีจำนวนมากและมีการเปลี่ยนแปลงบ่อยตามแต่ละโรงแรม สถานที่ ซึ่งในเบื้องต้นจึงทำการเลือกเว็บไซต์ที่ได้รวบรวม

ข้อมูลของราคาโรงแรมไว้แล้ว คือ Hotels.com และได้กำหนดข้อมูลที่จะให้บอททำการเก็บกลับมา คือ ไอดีของโรงแรม ชื่อโรงแรม ระดับดาวและราคา แต่เนื่องจากว่าราคาของโรงแรมนั้นมีการเปลี่ยนแปลงตลอดเวลา จึงจำเป็นต้องสั่งการให้บอทเก็บข้อมูลตามช่วงเวลาที่สั่งการในแต่ละวัน และมีการกำหนดให้บอทเก็บข้อมูลล่วงหน้าด้วย

3.1.1 ส่วนประกอบของระบบ

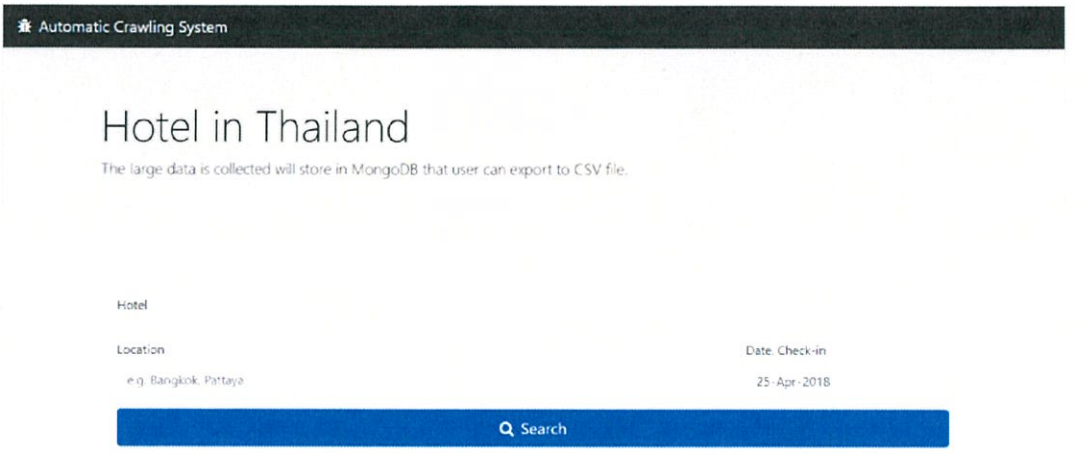
ระบบรวบรวมข้อมูลอัตโนมัติจะแบ่งการทำงานออกเป็น 3 ส่วน ซึ่งทุกส่วนต้องทำงานร่วมกันเพื่อให้ระบบสามารถทำงานได้อย่างต่อเนื่อง ดังนี้

3.1.1.1 ส่วนติดต่อผู้ใช้งาน

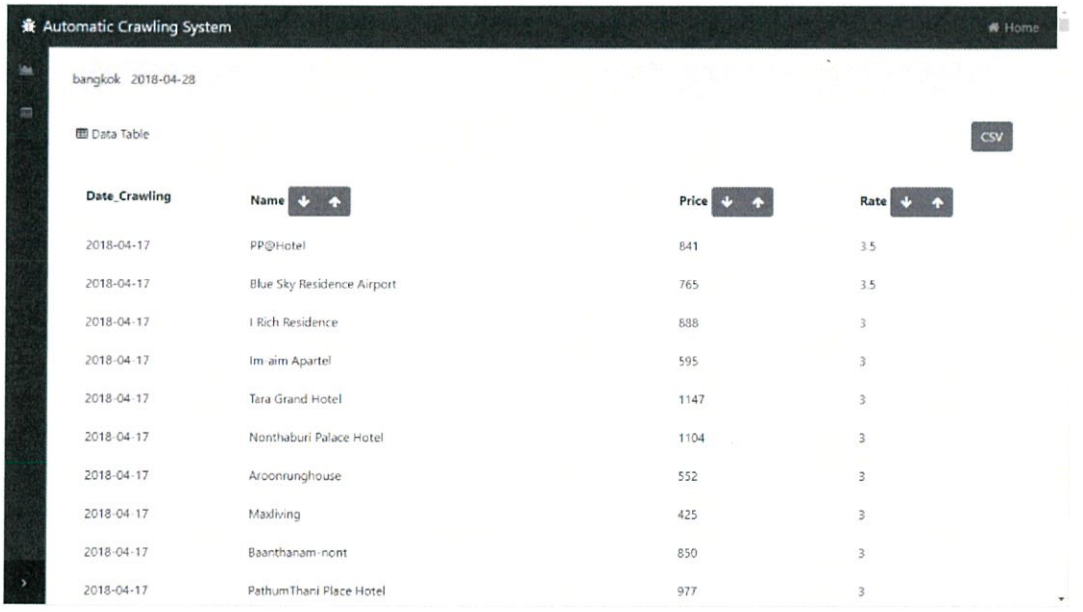
ส่วนติดต่อผู้ใช้งานหรือส่วนหน้าบ้าน (Front-end) เป็นส่วนที่ติดต่อกับผู้ใช้งานและส่วนจัดการระบบ (รูป 3.1) ทำหน้าที่ในการรับค่าอินพุทที่ต้องการค้นหาข้อมูลภายในฐานข้อมูลจากผู้ใช้งาน (รูป 3.2) ได้แก่สถานที่ (Location) และวันที่ (Date) เพื่อส่งต่อไปยังส่วนจัดการระบบ และทำหน้าที่รับข้อมูลที่เป็นผลลัพธ์ของการค้นหากลับมาแสดงเป็นตาราง (รูป 3.3) และกราฟ (รูป 3.4)

รูป 3.1 การติดต่อภายในระบบของส่วนติดต่อผู้ใช้งาน

การใช้งานของระบบนั้น เมื่อเริ่มต้น ผู้ใช้งานเข้าสู่หน้าค้นหาของเว็บแอปพลิเคชันจากนั้นจะกรอกสถานที่และวันที่ต้องการค้นหาโรมแรมและราคา (รูป 3.2) เมื่อกดค้นหา หน้าของเว็บแอปพลิเคชันจะเปลี่ยนหน้าการทำงานไปที่หน้าแสดงผลลัพธ์การค้นหา (รูป 3.3) ซึ่งในหน้าแสดงผลลัพธ์จะแสดงตารางข้อมูล หากกดที่ชื่อของโรงแรมที่ผู้ใช้งานสนใจ จะเป็นการเปลี่ยนหน้าของเว็บแอปพลิเคชันจากหน้าแสดงตารางไปยังหน้าแสดงกราฟราคาของโรงแรม (รูป 3.4) ซึ่งสามารถดูแนวโน้มราคาของโรงแรมซึ่งเป็นราคาที่สัมพันธ์กับวันที่ได้

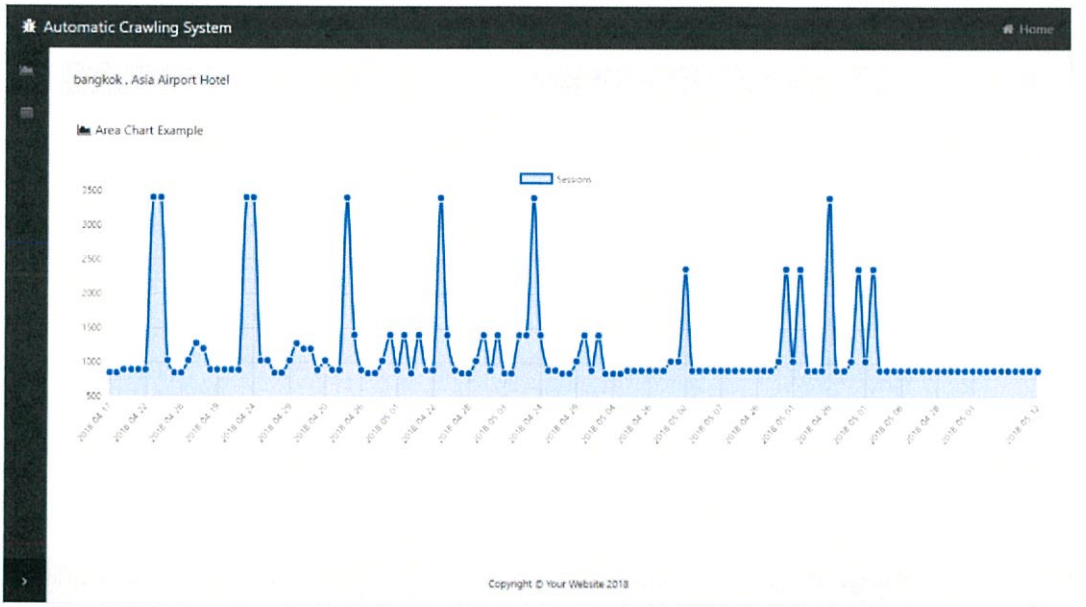


รูป 3.2 หน้าเว็บสำหรับค้นหาโรงแรม



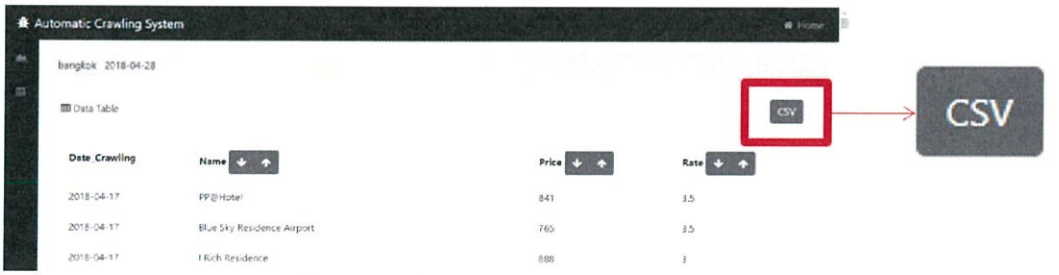
Date_Crawling	Name	Price	Rate
2018-04-17	PP@Hotel	841	3.5
2018-04-17	Blue Sky Residence Airport	765	3.5
2018-04-17	I Rich Residence	888	3
2018-04-17	Im-aim Apartel	595	3
2018-04-17	Tara Grand Hotel	1147	3
2018-04-17	Nonthaburi Palace Hotel	1104	3
2018-04-17	Aroonrunghouse	552	3
2018-04-17	Maxliving	425	3
2018-04-17	Baanthanam-nont	850	3
2018-04-17	PathumThani Place Hotel	977	3

รูป 3.3 หน้าแสดงตารางผลลัพธ์การค้นหาโรงแรม



รูป 3.4 หน้าแสดงกราฟผลลัพธ์การค้นหาโรงแรม

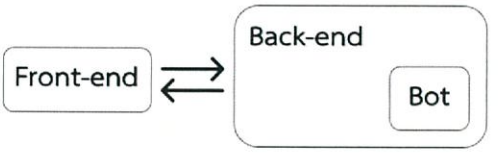
นอกจากนี้ ในหน้าแสดงผลลัพธ์การค้นหาในรูปแบบของตาราง ยังสามารถทำการส่งออกข้อมูลในรูปแบบของไฟล์ซีเอสวีเพื่ออำนวยความสะดวกในการนำไปใช้งานต่อตามความต้องการ โดยกดปุ่ม CSV ที่อยู่บริเวณมุมขวาบนของตาราง (รูป 3.5) จะเป็นการดาวน์โหลดข้อมูลที่แสดงอยู่ในตารางลงในเครื่องคอมพิวเตอร์ของผู้ใช้งานในรูปแบบของไฟล์ซีเอสวี



รูป 3.5 ปุ่มสำหรับส่งออกข้อมูลในรูปแบบของไฟล์ซีเอสวี

3.1.1.2 ส่วนจัดการระบบ

ส่วนจัดการระบบหรือส่วนหลังบ้าน (Back-end) จะทำหน้าที่เป็นส่วนกลางในการติดต่อระหว่างส่วนติดต่อผู้ใช้งานกับส่วนรวบรวมข้อมูล (รูป 3.6) ทำหน้าที่รับอินพุทของผู้ใช้งานที่มาจากส่วนติดต่อผู้ใช้งาน ส่งต่อไปยังฐานข้อมูลเพื่อทำการค้นหาผลลัพธ์และส่งกลับไปแสดงผลยังเว็บแอปพลิเคชัน และทำหน้าที่ติดต่อกับส่วนรวบรวมข้อมูลเพื่อรับข้อมูลที่บอททำการเก็บมาได้ลงในฐานข้อมูล



รูป 3.6 การติดต่อภายในระบบของส่วนจัดการระบบ

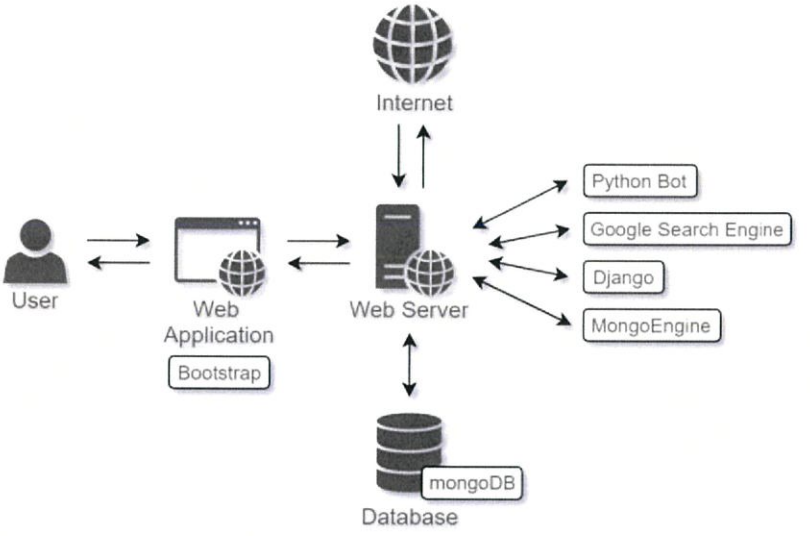
3.1.1.3 ส่วนรวบรวมข้อมูล (Bot)

ส่วนการรวบรวมข้อมูลหรือบอท เป็นส่วนที่ทำการเก็บรวบรวมข้อมูลจากเว็บไซต์ที่กำหนด ซึ่งจะทำการค้นหายูอาร์แอลผ่านผ่านเครื่องมือการค้นหา ซึ่งในโครงงานชิ้นนี้ได้ใช้กูเกิลเสิร์ชเอนจิน (รูป 3.7) จากนั้นจะทำการเลือกยูอาร์แอลที่ได้มาจากเครื่องมือสำหรับการค้นหา มากำหนดให้สไปเดอร์หรือบอทไปทำการเก็บข้อมูลตามยูอาร์แอล เมื่อได้ข้อมูลมาแล้วจะทำการติดต่อไปยังส่วนจัดการระบบเพื่อจัดเก็บข้อมูลที่ได้มาลงฐานข้อมูลเพื่อรอการนำไปใช้งานต่อไป

รูป 3.7 ส่วนประกอบของบอท

3.1.2 การทำงานของระบบ

ระบบเก็บข้อมูลจะใช้คีย์เวิร์ดที่กำหนดจากผู้ใช้งาน ได้แก่ สถานที่และวันที่เป็นคำสำคัญเพื่อนำไปค้นหาข้อมูลจากฐานข้อมูลเพื่อนำมาแสดงผล โดยจะมีส่วนของบอทที่ทำการเก็บข้อมูลซึ่งใช้กูเกิลเสิร์ชเอนจินเพื่อรวบรวมยูอาร์แอลมากำหนดที่อยู่ของข้อมูลให้แก่บอทเพื่อทำการรวบรวมข้อมูลต่อไป โดยข้อมูลที่บอทได้มาจะถูกเก็บเป็นประเภทของข้อความและตัวเลขในรูปแบบโครงสร้างข้อมูลและชนิดไฟล์ข้อมูลที่กำหนดลงในฐานข้อมูล สามารถส่งออกข้อมูลเป็นไฟล์ซีเอสวีจากหน้าเว็บแอปพลิเคชันหรือจากฐานข้อมูลโดยตรงเพื่อนำไปใช้ในการวิเคราะห์ต่อไปได้



รูป 3.8 ภาพรวมการทำงานของระบบรวบรวมข้อมูลอัตโนมัติ

ในขั้นตอนการทำงาน บอทจะถูกสั่งการเพื่อให้บอทไปทำการเก็บข้อมูลที่คาดว่าเป็นข้อมูลที่ต้องการจากผู้ใช้งานมาจัดเก็บลงฐานข้อมูลล่วงหน้า ในขณะใช้งาน ผู้ใช้งานจะป้อนคำค้นหา ได้แก่ สถานที่และวันที่ ลงในหน้าค้นหาของเว็บแอปพลิเคชัน จากนั้นคำค้นหาจะถูกส่งไปที่ส่วนจัดการระบบเพื่อทำการดึงข้อมูลที่ตรงกับคำค้นหาจากฐานข้อมูลออกมาเพื่อส่งผลลัพธ์กลับไปแสดงผลที่หน้าแสดงผลของเว็บแอปพลิเคชันต่อไป

3.2 เครื่องมือที่ใช้ในการพัฒนา

ระบบรวบรวมข้อมูลได้มีการกำหนดภาษาและไลบรารีที่เหมาะสมซึ่งจำเป็นในการทำงานของระบบ เครื่องมือในการพัฒนาที่มีส่วนช่วยอำนวยความสำดวกสำหรับการทำงานควบคู่กัน ดังนี้

3.2.1 ภาษาที่ใช้ในการพัฒนา

1) ภาษาไพธอน ใช้ในการเขียนโปรแกรมเพื่อสร้างสไปเดอร์สำหรับทำการเก็บข้อมูลและการทำงานในส่วนจัดการระบบ
2) ภาษาเอชทีเอ็มแอล เป็นภาษาที่ใช้ในการสร้างเว็บเพจ ในการพัฒนาระบบจะใช้วิเคราะห์โครงสร้างของเว็บไซต์เมื่อจะทำการเก็บข้อมูลที่ต้องการจากเพจนั้น ๆ
3) ภาษาซีเอสเอส ใช้ร่วมกับภาษาเอชทีเอ็มแอลเพื่อให้เว็บไซต์มีความสวยงามจัดรูปแบบได้ง่าย
4) ภาษาจาวาสคริปต์ ใช้ร่วมกับภาษาเอชทีเอ็มแอลเพื่อให้เว็บไซต์มีการเคลื่อนไหว ตอบสนองต่อผู้ใช้งานและมีความน่าสนใจมากยิ่งขึ้น

3.2.2 เทคโนโลยีที่ใช้ในการพัฒนา

มีการใช้เทคโนโลยีต่าง ๆ เพื่อช่วยอำนวยความสะดวกและลดขั้นตอนในการทำงาน ดังนี้

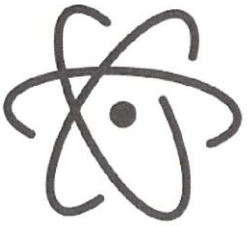
1) ไลบรารี Scrapy และ Selenium จะใช้ในการสร้างบอทเพื่อนำไปเก็บข้อมูล
2) ไลบรารีกูเกิล เป็นการเรียกใช้งานกูเกิลเสิร์ชเอนจินสำหรับการค้นหายูอาร์แอล
3) Bootstrap คือ เฟรมเวิร์คสำหรับพัฒนาเว็บไซต์ เพื่อช่วยให้เขียนเว็บไซต์ได้ง่าย สะดวก และสวยงาม
4) Django ใช้สำหรับการจัดการระบบหลังบ้านและติดต่อกับฐานข้อมูล
5) MongoDB เป็นฐานข้อมูลแบบเอกสารสำหรับจัดเก็บข้อมูล
6) Mongoengine ใช้สำหรับแปลงออบเจ็กต์ในภาษาจาวาสคริปต์ให้เป็นเอกสารใน MongoDB

3.2.3 เครื่องมือที่ใช้ในการพัฒนา

1) PyCharm เครื่องมือที่ใช้สำหรับการพัฒนาภาษาไพธอน ช่วยอำนวยความสะดวกในการทำงานและพัฒนาระบบได้อย่างมีประสิทธิภาพ สามารถค้นหาและติดตั้งไลบรารีที่ต้องการจากโปรแกรมได้

รูป 3.9 PyCharm

2) Anaconda Prompt เป็น command-line tool สำหรับอำนวยความสะดวกในการใช้งานเครื่องมือของบางไลบรารีและใช้กับคำสั่งทั่วไปในภาษาไพธอนที่ Command Prompt ของ Windows ไม่สามารถทำได้
3) Atom เป็น Text Editor ใช้แก้ไขโค้ด สามารถเปิดหลายโปรเจคหรือหลายหน้าต่างพร้อมกันได้



รูป 3.10 Atom text editor

4) Swagger API เป็นเครื่องมือที่ช่วยในการสร้าง API (Application Programming Interface) สำหรับใช้เป็นช่องทางการสื่อสารระหว่างส่วนหน้าบ้านและส่วนหลังบ้าน

รูป 3.11 Swagger API

3.3 รายละเอียดเชิงเทคนิค

3.3.1 ส่วนติดต่อผู้ใช้งาน

ในส่วนนี้จะเป็นส่วนที่ใช้ติดต่อกับผู้ใช้งาน ซึ่งจะทำการติดต่อและแสดงผลผ่านหน้าเว็บของเว็บแอปพลิเคชัน โดยจะใช้ภาษาเอชทีเอ็มแอลสำหรับวางโครงสร้างและแบ่งส่วนตามแท็กต่าง ๆ ภาษาซีเอสเอสสำหรับการกำหนดรูปแบบและลักษณะของเว็บไซต์ให้ดูสวยงามและเป็นไปในทิศทางเดียวกัน และภาษาจาวาสคริปต์สำหรับการทำให้เว็บไซต์มีการตอบสนองต่อผู้ใช้งานและเป็นส่วนที่กำหนดการเชื่อมต่อกับระบบหลังบ้าน จึงได้นำ Bootstrap มาใช้ เพื่อช่วยในการสร้างเว็บแอปพลิเคชันให้มีความทันสมัย ใช้งานได้ง่ายและสวยงาม

ซึ่งภายในส่วนติดต่อผู้ใช้งานจะแบ่งส่วนออกเป็น 2 ส่วนย่อยตามลักษณะการติดต่อกับผู้ใช้งาน คือ

1) ส่วนการรับข้อมูลหรือส่วนรับคำค้นหาจากผู้ใช้งาน ในส่วนนี้จะประกอบไปด้วยปุ่มสำหรับกดค้นหา (Search) และช่องในการเติมคำสำหรับค้นหา ได้แก่
   - ช่องสำหรับกรอกชื่อสถานที่ (Location) ซึ่งจำเป็นต้องกรอกเป็นข้อความตัวอักษร เช่น bangkok หรือ pattaya เป็นต้น
   - ช่องสำหรับกรอกวันที่ (Date) สามารถกรอกเป็นเลขของวัน-เดือน-ปี (DD-MMM-YYYY) หรือสามารถเลือกวันที่จากปฏิทินที่แสดงบนหน้าจอได้

2) ส่วนการแสดงผลลัพธ์การค้นหาแก่ผู้ใช้งาน เป็นส่วนที่แสดงข้อมูลจากการค้นหาภายในฐานข้อมูลที่ได้รับมาจากส่วนหลังบ้าน ซึ่งการแสดงข้อมูลในส่วนนี้จะแสดงเป็นตารางของข้อมูลโรงแรมในสถานที่ที่ผู้ใช้งานค้นหาและกราฟความสัมพันธ์ของราคาและวันที่จากโรงแรมที่ผู้ใช้งานสนใจ

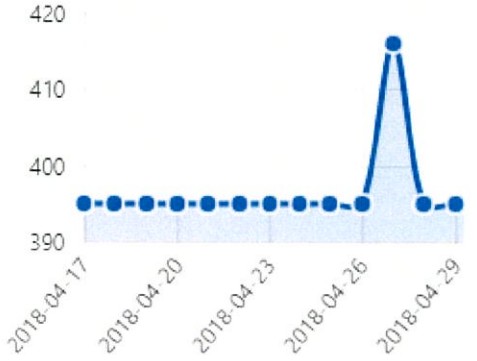
Date_Crawling	Name	Price	Rate
2018-04-17	PP@Hotel	841	3.5
2018-04-17	Blue Sky Residence Airport	765	3.5
2018-04-17	I Rich Residence	888	3
2018-04-17	Im-aim Apartel	595	3
2018-04-17	Tara Grand Hotel	1147	3

รูป 3.12 ข้อมูลโรงแรมจากการค้นหาสถานที่ซึ่งแสดงในรูปแบบของตาราง

โดยที่ตารางจะแสดงข้อมูล ได้แก่ วันที่เก็บราคาโรงแรม (Date_Crawling) ชื่อของโรงแรม (Name) ราคาโรงแรม (Price) และระดับดาวของโรงแรม (Rate) (รูป 3.12) นอกจากนั้น ในส่วนนี้ผู้ใช้งานยังสามารถส่งออกไฟล์ข้อมูลที่แสดงในตารางออกมาเป็นไฟล์ซีเอสวีได้โดยกดที่ปุ่ม CSV บริเวณมุมขวาบนของตาราง (รูป 3.13)

รูป 3.13 ปุ่มสำหรับการส่งออกข้อมูลในรูปแบบของไฟล์ซีเอสวี

และในส่วนของหน้าเว็บแอปพลิเคชันที่แสดงกราฟ จะแสดงความสัมพันธ์ระหว่างวันที่และราคาของโรงแรมออกมาเป็นกราฟเส้น ซึ่งมีแกน X เป็นแกนของวันที่ ส่วนแกน Y เป็นแกนของราคาโรงแรม (รูป 3.14) ซึ่งข้อมูลที่นำมาแสดงจะเป็นข้อมูลของโรงแรมนั้น ๆ ที่ทำการเลือกโดยการกดที่ชื่อของโรงแรมที่สนใจภายในตารางจากหน้าแสดงตารางของเว็บแอปพลิเคชัน (รูป 3.15) ซึ่งกราฟที่ได้จะแสดงว่าในแต่ละวันนั้น ราคาของโรงแรมมีราคาเท่าใด ในภาพรวมมีแนวโน้มของราคาเป็นอย่างไร ผู้ใช้งานสามารถสังเกตได้จากกราฟในเบื้องต้น



รูป 3.14 กราฟแสดงความสัมพันธ์ของวันที่ (แกน X) และราคา (แกน Y)

Date_Crawling	Name	Price	Rate
2018-04-17	PP@Hotel	841	3.5
2018-04-17	Blue Sky Residence Airport	765	3.5
2018-04-17	I Rich Residence	888	3
2018-04-17	Im-aim Apartel	595	3
2018-04-17	Tara Grand Hotel	1147	3
2018-04-17	Nonthaburi Palace Hotel	1104	3
2018-04-17	Aroonrunghouse	552	3
2018-04-17	Maxliving	425	3

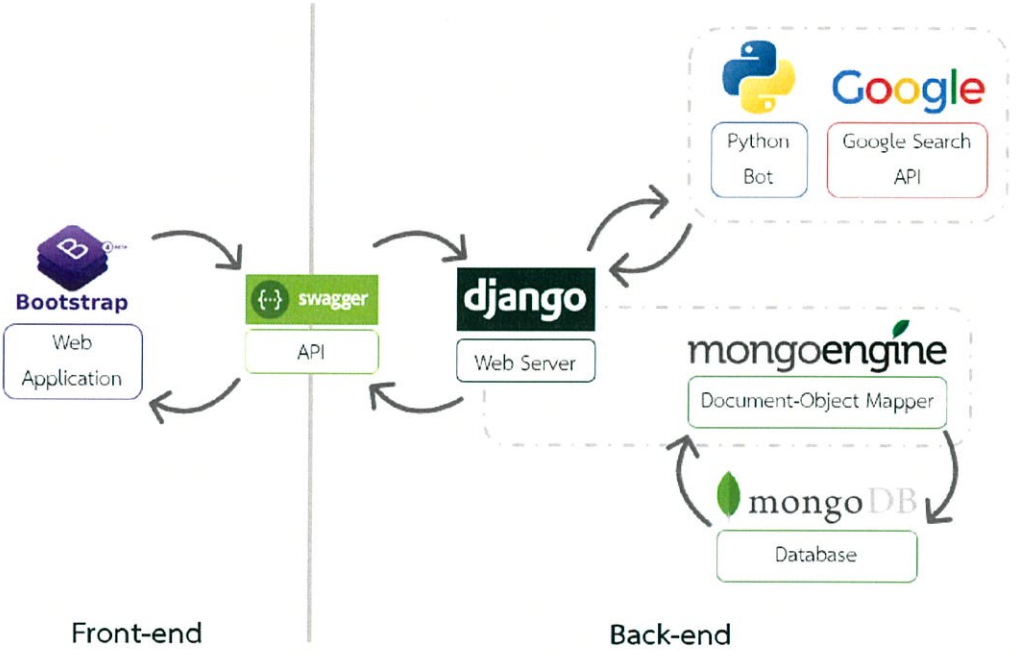
รูป 3.15 การเลือกโรงแรมเพื่อแสดงกราฟราคา

3.3.2 ส่วนจัดการระบบ

ส่วนจัดการระบบเป็นส่วนที่ทำงานอยู่เบื้องหลังของเว็บแอปพลิเคชัน มีหน้าที่หลักในการจัดการระบบเพื่อให้เว็บแอปพลิเคชันสามารถทำงานผ่านเว็บเบราว์เซอร์ได้ และยังใช้ในการติดต่อกับส่วนต่าง ๆ ทั้งภายในและภายนอกของส่วนจัดการระบบ ได้แก่

1) การติดต่อภายในส่วนจัดการระบบ เป็นการติดต่อกับส่วนย่อยต่าง ๆ ที่อยู่ภายใน ได้แก่ บอท ฐานข้อมูล และเว็บเซิร์ฟเวอร์ ซึ่งเว็บเซิร์ฟเวอร์จะเป็นตัวกลางในการติดต่อกับบอทและฐานข้อมูล ในส่วนของบอท แม้ว่าบอทจะถูกจัดให้เป็นส่วนการทำงานหลัก แต่ก็ยังคงอยู่ภายในส่วนจัดการระบบ ซึ่งส่วนจัดการระบบจะทำหน้าที่ในการนำข้อมูลที่บอทเก็บมาได้นั้นจัดเก็บลงในฐานข้อมูล และต้องทำการค้นหาข้อมูลจากฐานข้อมูลเพื่อเตรียมสำหรับส่งไปยังส่วนติดต่อผู้ใช้งาน
2) การติดต่อภายนอกส่วนจัดการระบบ เป็นการติดต่อกับเว็บแอปพลิเคชันในส่วนติดต่อผู้ใช้งานผ่านการใช้เอพีไอเพื่อรับส่งข้อมูลระหว่างกัน เพื่อให้สามารถนำ

ข้อมูลที่ผู้ใช้งานต้องการส่งไปแสดงผลยังหน้าแสดงผลลัพธ์ของเว็บแอปพลิเคชันได้



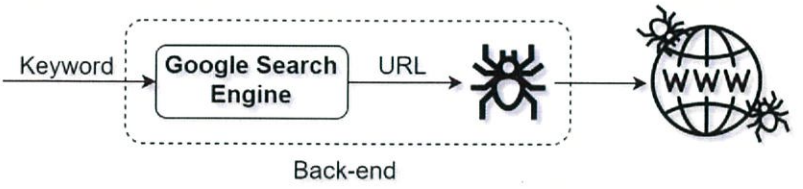
รูป 3.16 การติดต่อภายในระบบรวบรวมข้อมูล

ในโครงงานชิ้นนี้จะใช้แจงโก้เฟรมเวิร์คสำหรับทำหน้าที่เป็นเว็บเซิร์ฟเวอร์เพื่อให้เว็บแอปพลิเคชันสามารถใช้งานได้ มีการเลือกใช้มองโกดีบีเป็นฐานข้อมูล ฐานข้อมูลที่เลือกใช้นั้นจะเป็นฐานข้อมูลประเภทเอกสาร (Document Database) ที่เป็นฐานข้อมูลที่ไม่มีความสัมพันธ์ของโครงตาราง โดยปกติแล้วแจงโก้เฟรมเวิร์คจะไม่สามารถทำงานร่วมกับฐานข้อมูลประเภทนี้ได้ จึงได้นำมองโกเอนจินมาใช้สำหรับเป็นตัวกลางในการทำงานร่วมกับฐานข้อมูล ได้แก่ การเก็บและการค้นหาข้อมูล เพราะมองโกเอนจินเป็นเครื่องมือที่ช่วยในการแมพข้อมูลเพื่อทำการเก็บข้อมูลลงฐานข้อมูลหรือทำการค้นหาข้อมูลจากฐานข้อมูลมาได้

นอกจากนี้ยังมีตัวกลางในการติดต่อระหว่างส่วนติดต่อผู้ใช้งานและส่วนจัดการระบบ ซึ่งเป็นตัวกลางสำหรับใช้รับหรือส่งข้อมูล ซึ่งจะใช้สแวกเกอร์เอพีไอเป็นตัวกลาง เพราะสามารถใช้งานได้ง่ายเนื่องจากมีส่วนประสานกับผู้ใช้ที่สามารถใช้สำหรับทดสอบการรับส่งข้อมูล และเชื่อมต่อนั้นสามารถทำได้รวดเร็วและไม่ซับซ้อน เพียงแค่นำยูอาร์แอลที่ได้จากเอพีไอไปใส่ในส่วนการรับส่งข้อมูลของทั้งทางฝั่งหน้าบ้านและหลังบ้าน ก็จะสามารถทำการติดต่อระหว่างกันได้

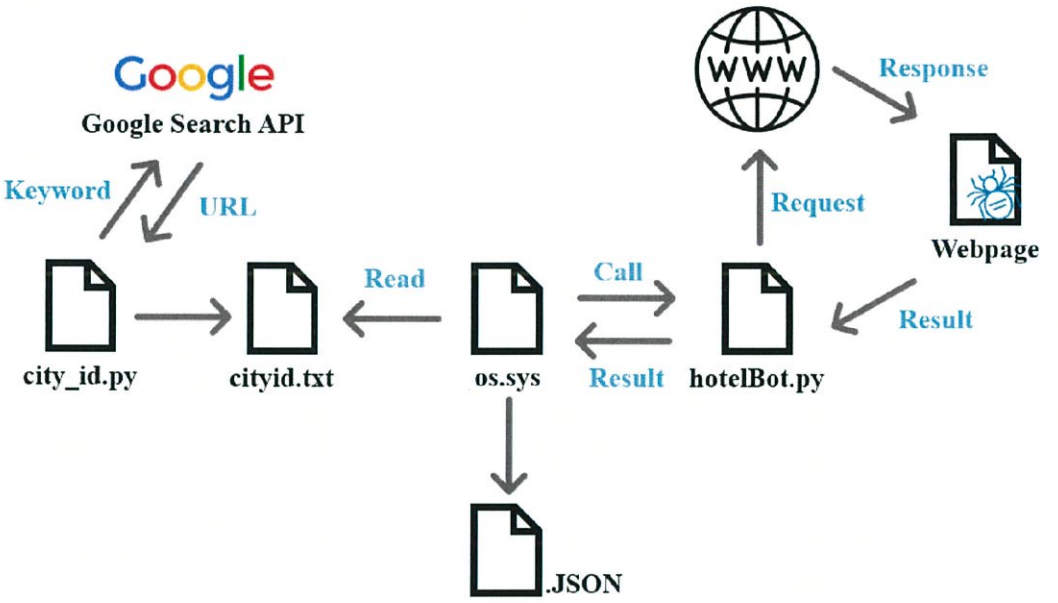
3.3.3 ส่วนรวบรวมข้อมูล

ส่วนการรวบรวมข้อมูลหรือส่วนของบอทในระบบรวบรวมข้อมูลอัตโนมัติ จะอยู่ภายในส่วนจัดการระบบ โดยลักษณะการทำงานของบอทจะถูกกำหนดไว้ในเบื้องต้นเพื่อให้บอททำการเก็บข้อมูลตามที่กำหนด โดยจะนำคีย์เวิร์ดไปค้นหาในกูเกิลเสิร์ชแล้วจะได้ผลลัพธ์เป็นรายการของยูอาร์แอลที่เกี่ยวข้องออกมา จากนั้นจะเลือกยูอาร์แอลจากเว็บไซต์ที่กำหนดไว้ เมื่อได้ยูอาร์แอลแล้วจะทำการสั่งงานให้บอทไปยังที่อยู่นั้น ๆ เพื่อทำการเก็บข้อมูล ข้อมูลที่ได้มาก็จะถูกส่งไปเก็บยังฐานข้อมูลเพื่อรอการค้นหาจากผู้ใช้งานต่อไป



รูป 3.17 การค้นหายูอาร์แอลสำหรับกำหนดให้แก่บอท

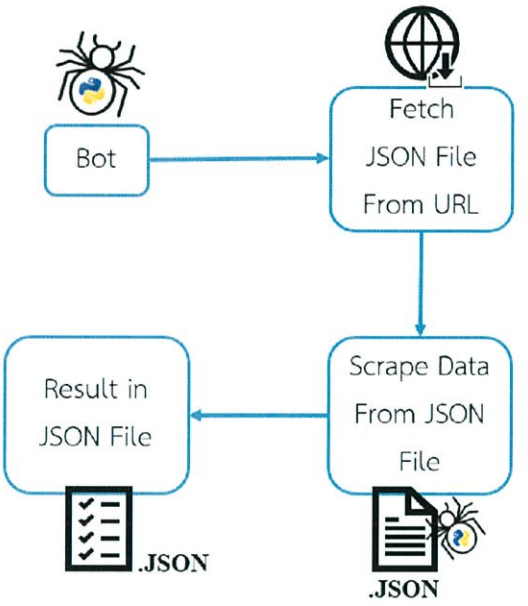
ในโครงงานชิ้นนี้ เป็นการเก็บข้อมูลของโรงแรมจากเว็บไซต์ Hotels.com ซึ่งจะเลือกเก็บเฉพาะโรงแรมในประเทศไทยตามจังหวัดต่าง ๆ ในขั้นตอนการทำงานเริ่มต้นจึงเป็นการหาไอดีและยูอาร์แอลของจังหวัด โดยจะใส่อินพุทเป็นรายชื่อจังหวัดลงในไฟล์ city_id.py ซึ่งเป็นไฟล์สำหรับค้นหาไอดีและยูอาร์แอลผ่านกูเกิลเสิร์ชที่ใช้ชื่อของจังหวัดในประเทศไทยเป็นคีย์เวิร์ด เมื่อทำงานสำเร็จจะนำผลลัพธ์ที่ได้เขียนลงในไฟล์ cityid.txt เพื่อรอให้คำสั่งในไฟล์ os.sys มาทำการอ่านข้อมูลไอดีและยูอาร์แอลของแต่ละจังหวัด จากในไฟล์ cityid.txt เพื่อไปกำหนดยูอาร์แอลให้แก่บอทหรือก็คือไฟล์ hotelBot.py โดยบอทจะไปยังยูอาร์แอลที่ได้เพื่อเก็บข้อมูลตามที่กำหนดไว้ในไฟล์ hotelBot.py จากนั้นจะนำข้อมูลที่เป็นผลลัพธ์จากการทำงานของบอทส่งกลับไปยังไฟล์ os.sys เพื่อทำการเขียนลงไฟล์ผลลัพธ์ประเภทเจสันสำหรับจัดเก็บลงฐานข้อมูลในขั้นตอนลำดับถัดไป



รูป 3.18 การทำงานของบอท

ในโครงงานชิ้นนี้ วิธีการทำงานเพื่อเก็บข้อมูลของบอทจะเลือกใช้ตามความเหมาะสมของแต่ละเว็บไซต์ เพื่อให้บอทเก็บข้อมูลได้อย่างมีประสิทธิภาพ โดยจะใช้วิธีเก็บข้อมูลด้วยกัน 2 วิธี คือ

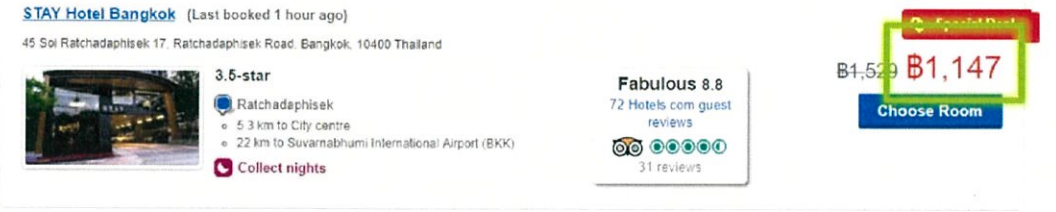
1) การเก็บข้อมูลแบบเก็บจากไฟล์เจสัน โดยที่จะให้บอททำการดักเก็บไฟล์ข้อมูลประเภทเจสัน ก่อนที่ระบบหลังบ้านของเว็บไซต์นั้น ๆ จะส่งข้อมูลมาถึงแล้วแสดงผลในหน้าเว็บ ซึ่ง ไฟล์เจสันที่ได้จะมีข้อมูลของเว็บไซต์ที่ดึงมาจากฐานข้อมูลอยู่รวมกันมากมาย จึงต้องทำการกรองเอาเฉพาะข้อมูลที่ต้องการซึ่งได้กำหนดไว้แล้ว เช่น หากเป็นข้อมูลจากโรงแรม จะเลือกเก็บไอดีของโรงแรม ชื่อโรงแรม ราคา ระดับดาว เป็นต้น



รูป 3.19 การเก็บข้อมูลแบบเก็บจากไฟล์เจสัน

รูป 3.20 ส่วนสำหรับค้นหายูอาร์แอลของไฟล์เจสันจากเว็บเบราว์เซอร์

2) การเก็บข้อมูลจากแท็กในภาษาเอชทีเอ็มแอล วิธีนี้เป็นวิธีที่บอทจะทำการเก็บข้อมูลจากหน้าเว็บไซต์ที่ถูกแสดงแล้ว โดยจะเลือกข้อมูลที่จะทำการเก็บก่อน แล้วสำรวจดูว่าข้อมูลนั้น ๆ อยู่บริเวณส่วนใดของหน้าเว็บและมีแท็กชื่อว่าอะไร แล้วจึงค่อยกำหนดให้บอทไปเก็บข้อมูลตรงส่วนนั้น ๆ กลับมาได้




```
▼<div class="price">
  ▼<a href="/ho633829184/?pa=4&q-check-out=2018-04-
  30&tab=description&q-room-0-adults…
  9&MGT=1&WOE=1&WOD=7&ZSX=0&SYE=3&q-room-0-children=0#rooms-and-
  rates-anchor" target="_blank">
    ▼<span class="old-price-cont">
       <del data-reason="DRR-446">฿1,529</del>
       <ins class="special-deal-animation">฿1,147</ins>
     </span>
   </a>
</div>
```

รูป 3.21 ตัวอย่างการสำรวจชื่อแท็กในภาษาเอชทีเอ็มแอลจากเว็บไซต์

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองส่วนติดต่อผู้ใช้งาน

4.1.1 การส่งและรับข้อมูลของเว็บแอปพลิเคชัน

การส่งและรับข้อมูลของเว็บแอปพลิเคชันจะใช้ในการติดต่อกับส่วนจัดการระบบ เมื่อผู้ใช้งานต้องการค้นหาข้อมูลจากฐานข้อมูล โดยที่เว็บแอปพลิเคชันจะนำคีย์เวิร์ดที่ได้รับจากผู้ใช้งานส่งไปยังส่วนจัดการระบบและรอส่วนจัดการระบบส่งข้อมูลที่เป็นผลลัพธ์กลับมาเพื่อแสดงให้แก่ผู้ใช้งาน

ในการสร้างช่องทางเพื่อติดต่อระหว่างส่วนติดต่อผู้ใช้งานกับส่วนจัดการระบบจะใช้ภาษาจาวาสคริปต์ในการเชื่อมต่อกับเอพีไอที่เป็นตัวกลางในการติดต่อ ทำได้โดยการเพิ่มสคริปต์คำสั่งลงในไฟล์เอชทีเอ็มแอลของหน้าเว็บไซต์ที่ต้องการรับหรือส่งข้อมูล ในส่วนของเอพีไอจะเลือกใช้สแวกเกอร์เอพีไอซึ่งมีส่วนต่อประสานกับผู้ใช้งานที่ทำให้สามารถใช้งานได้สะดวก

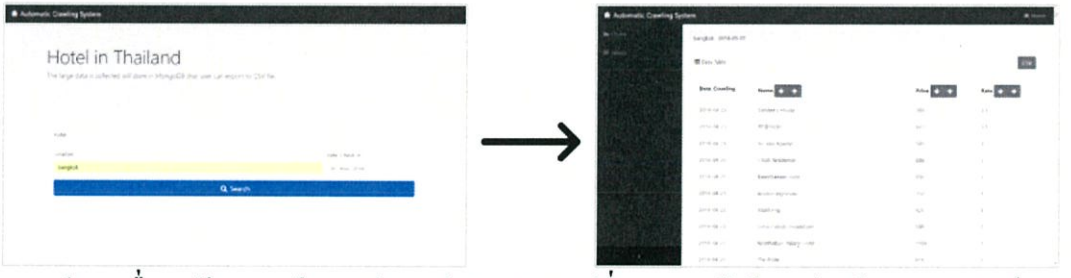
โปรแกรม 4.1 การส่งข้อมูลสำหรับค้นหาไปที่ Back-end

```
<script>
     $(document).ready(function (){
          $("#sumbitSearch").click(function(){
               $.post("http://localhost:8000/api/hotel/",
               {
                    location:
                         document.getElementById(
                         "InputLocation").value,
                    date_checkin :
                         document.getElementById(
                         "InputDate").value,
               },
               function(data, status){
                    if(data.length > 0){
                         localStorage.setItem('location',
                              document.getElementById(
                              "InputLocation").value)
                         localStorage.setItem('date',
                              document.getElementById(
                              "InputDate").value)
                    window.location.href = "tables.html"
               }
```

โปรแกรม 4.1 การส่งข้อมูลสำหรับค้นหาไปที่ Back-end (ต่อ)

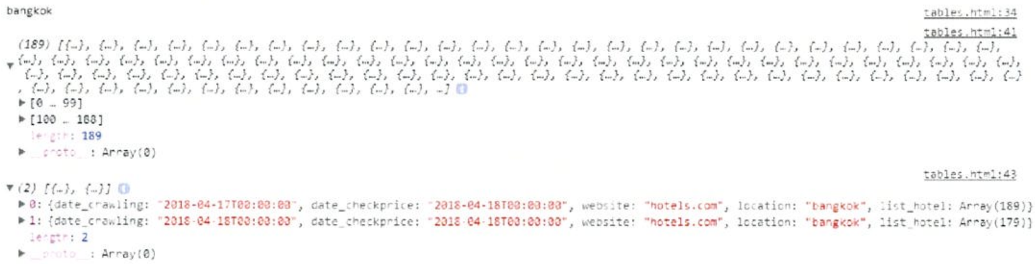
```
                else {
                    alert("not work");
                }
                });
                event.preventDefault();
            })
    });
</script>
```

จากสคริปต์คำสั่ง (โปรแกรม 4.1) เป็นคำสั่งในการส่งค่าข้อมูลของสถานที่และวันที่จาก "InputLocation" และ "InputDate" ไปค้นหาข้อมูลของโรงแรมที่ฐานข้อมูล โดยมีการติดต่อกับเอพีไอที่ยูอาร์แอล http://localhost:8000/api/hotel/ เมื่อเอพีไอได้รับการร้องขอจะส่งคำร้องขอไปที่ส่วนของระบบหลังบ้านเพื่อให้ทำการค้นหาข้อมูลและส่งกลับมาให้แก่ส่วนของระบบหน้าบ้านของเว็บแอปพลิเคชันเพื่อแสดงข้อมูลให้แก่ผู้ใช้งานและเปลี่ยนไปที่หน้าแสดงข้อมูลในรูปแบบตาราง (รูป 4.1) จากคำสั่ง window.location.href = "tables.html"

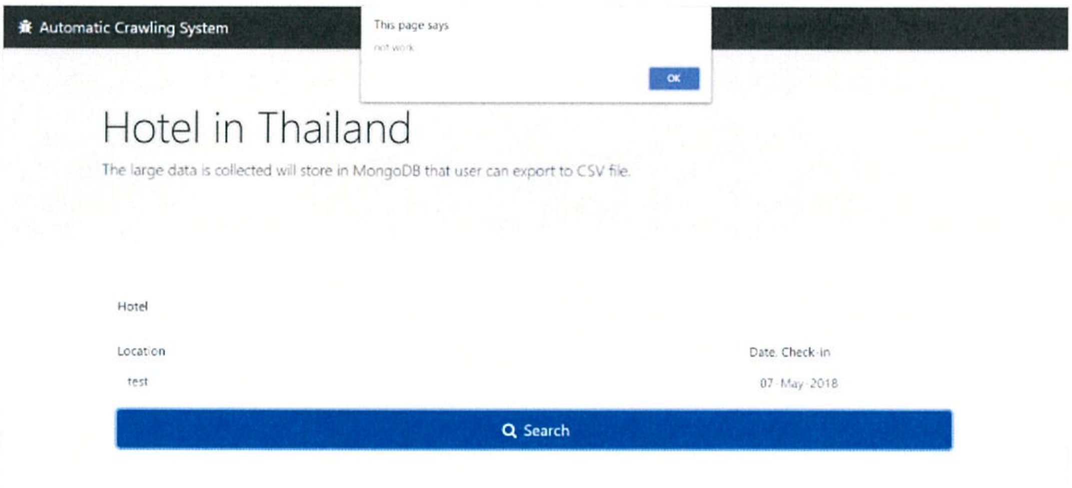


รูป 4.1 เมื่อกดค้นหา หน้าของเว็บแอปพลิเคชันจะเปลี่ยนจากหน้าค้นหาสู่หน้าแสดงผลลัพธ์ในรูปแบบตาราง

ถ้าคำร้องขอนั้นถูกต้องจะได้รับข้อมูลกลับมาในรูปแบบของอาร์เรย์ของออบเจ็กต์ (Array of Object) (รูป 4.2) แต่ถ้าหากว่าผู้ใช้งานป้อนคีย์เวิร์ดไม่ตรงกับเงื่อนไขที่กำหนด เช่น คำว่า "test" ซึ่งไม่ใช่ชื่อของสถานที่ เมื่อกดค้นหา ที่หน้าจอจึงแสดงข้อความ "not work" และจะไม่เปลี่ยนไปที่หน้าแสดงตาราง (รูป 4.3)



รูป 4.2 ผลลัพธ์การค้นหาข้อมูลที่ได้รับจากส่วนระบบหลังบ้านที่ console ในเว็บเบราว์เซอร์



รูป 4.3 การแจ้งเตือนเมื่อผู้ใช้งานกรอกชื่อสถานที่ไม่ถูกต้อง

4.1.2 การแสดงผลลัพธ์การค้นหาด้วยตาราง

การแสดงผลลัพธ์การค้นหาด้วยตารางเป็นการนำข้อมูลที่ได้จากฐานข้อมูลโดยการค้นหาข้อมูลที่ต้องการจากผู้ใช้งานมาแสดงข้อมูลในรูปแบบของตาราง เมื่อหน้าของเว็บแอปพลิเคชันสำหรับแสดงตารางได้รับข้อมูลของการค้นหา ซึ่งอยู่ในรูปของอาร์เรย์ของออบเจ็กต์ ในการนำข้อมูลมาแสดงในตารางจะทำได้โดยการสร้างคำสั่งด้วยภาษาจาวาสคริปต์ (โปรแกรม 4.2) ลงในไฟล์เอชทีเอ็มแอลของหน้าแสดงตาราง เพื่อแสดงข้อมูลตามโครงสร้างตารางที่ได้กำหนดไว้ (โปรแกรม 4.3)

โปรแกรม 4.2 การนำข้อมูลมาแสดงในตารางด้วยคำสั่งในภาษาจาวาสคริปต์

```
dataTables = listHotel
     $.each(listHotel, function(index, hotelValue){
          $("#dataTable").append(
               "<tr><td>" + hotelValue.date_crawling +
               "</td><td>" + hotelValue.hotel_name +
```

โปรแกรม 4.2 การนำข้อมูลมาแสดงในตารางด้วยคำสั่งในภาษาจาวาสคริปต์ (ต่อ)

```
                "</td><td>" + hotelValue.price +
                "</td><td>" + hotelValue.rate +
                "</td></tr>"
            );
        })
```

โปรแกรม 4.3 การสร้างตารางในภาษาเอชทีเอ็มแอล

```
<table id="dataTable" width="100%" class="dataTable" >
    <tr>
    <th>ID</th>
    <th>Name
        <div style="display:inline-block;"></div>
    </th>
    <th>Price
        <div style="display:inline-block;"></div>
    </th>
    <th>Rate
        <div style="display:inline-block;"></div>
    </th>
    </tr>
    <tbody>
    </tbody>
</table>
```

เนื่องจากว่า ต้องนำข้อมูลที่ได้รับจากฐานข้อมูลมาแสดงในตารางที่ชื่อ "dataTable" จึงต้องใช้คำสั่ง append เพื่อเป็นการนำข้อมูลใส่ลงในตาราง ซึ่งจะทำการแบ่งข้อมูลทีละเซลล์ (cell) ด้วยแท็กในรูปแบบของ "<td>" + ข้อมูลแต่ละเซลล์ + "</td>" วนจนครบทุกข้อมูลในโรงแรมเดียวกัน ได้แก่

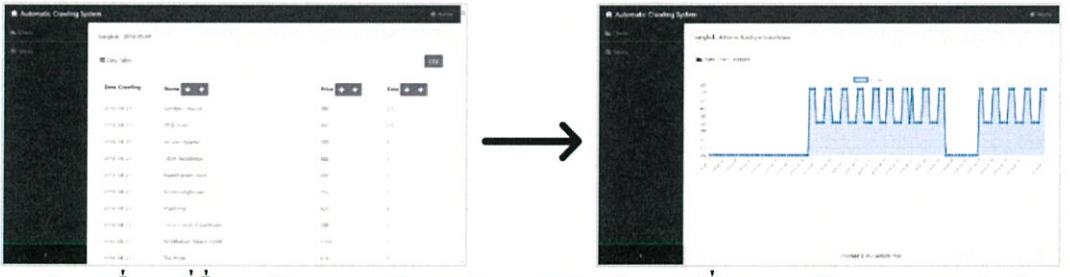
- วันที่เก็บข้อมูล (hotelValue.date_crawling)
- ชื่อของโรงแรม (hotelValue.hotel_name)
- ราคาของโรงแรม (hotelValue.price)
- ระดับดาวของโรงแรม (hotelValue.rate)

เนื่องจากข้อมูลของโรงแรมเดียวกันจำเป็นต้องอยู่ในแถวเดียวกัน จึงทำได้โดยการใช้แท็ก <tr> ในการเริ่มแถวใหม่และใช้แท็ก </tr> ในการสิ้นสุดแถวปัจจุบัน เมื่อทำคำสั่งจนเสร็จสิ้นจึงสามารถแสดงตารางให้แก่ผู้ใช้งานได้ (รูป 4.4)

รูป 4.4 ข้อมูลของโรงแรมที่แสดงในตาราง

4.1.3 การแสดงผลลัพธ์การค้นหาด้วยกราฟ

การแสดงผลลัพธ์การค้นหาด้วยกราฟเป็นการนำข้อมูลที่ได้จากฐานข้อมูล ได้แก่ ราคาและวันที่ของโรงแรมที่ผู้ใช้งานสนใจมาแสดงในรูปแบบของกราฟเส้น เพื่อดูแนวโน้มและความเปลี่ยนแปลงจากความสัมพันธ์ของราคาและวันที่ โดยที่ผู้ใช้งานสามารถเลือกโรงแรมที่ผู้ใช้งานสนใจจากรายชื่อของโรงแรมที่หน้าแสดงตาราง เมื่อผู้ใช้งานกดที่ชื่อของโรงแรม หน้าแสดงตารางของเว็บแอปพลิเคชันจะส่งไอดีและสถานที่ของโรงแรมไปยังส่วนหลังบ้านเพื่อทำการค้นหาจากฐานข้อมูลแล้วจะส่งผลลัพธ์การค้นหากลับมาในรูปของอาร์เรย์ของออบเจ็กต์ในหน้าแสดงกราฟของเว็บแอปพลิเคชันเพื่อแสดงผลให้แก่ผู้ใช้งาน (รูป 4.5)



รูป 4.5 เมื่อกดที่ชื่อของโรงแรม หน้าของเว็บแอปพลิเคชันจะเปลี่ยนจากหน้าแสดงตารางสู่หน้าแสดงผลลัพธ์ในรูปแบบของกราฟ

ในการนำข้อมูลมาแสดงในรูปแบบของกราฟเส้นจะทำได้โดยการสร้างคำสั่งด้วยภาษาจาวาสคริปต์ (โปรแกรม 4.4) ลงในไฟล์เอชทีเอ็มแอลของหน้าแสดงกราฟ เพื่อแสดงข้อมูลตามโครงสร้างของกราฟที่ได้กำหนดไว้ คือ แกน X แสดงวันที่ และแกน Y แสดงราคา

โปรแกรม 4.4 คำสั่งการแสดงข้อมูลด้วยกราฟในภาษาจาวาสคริปต์

```
function(data, status){
     console.log(data)
```

โปรแกรม 4.4 คำสั่งการแสดงข้อมูลด้วยกราฟในภาษาจาวาสคริปต์ (ต่อ)

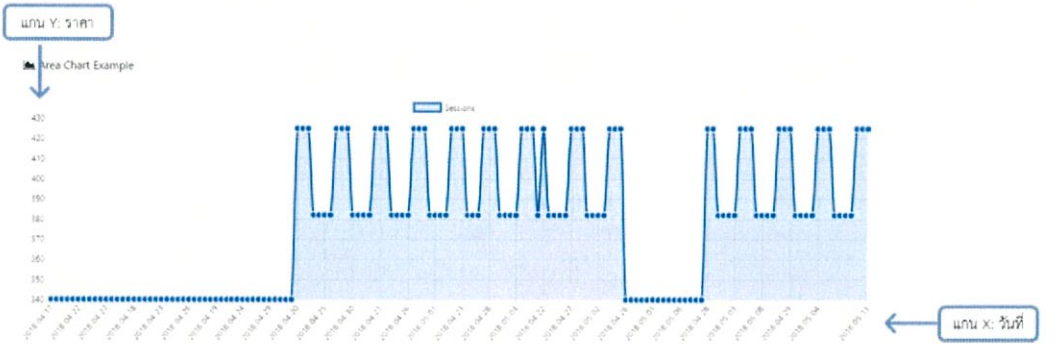
```
      $.each(data, function(key ,value){
            checkDateData.push(
                  value.date_checkprice.slice(0,10))
            $.each(value.list_hotel, function(index,
                  hotelValue){
                  priceData.push(hotelValue.price)
            })
      });
}).then(()=>{
      var ctx = document.getElementById("hotelChart");
      var myLineChart = new Chart(ctx, {
            type: 'line',
            data: {
                  labels: checkDateData,
                  datasets: [{
                        label: "Sessions",

                        ...,

                        data: priceData,
                  }],
            },
            options: {}
      });
})
```

โปรแกรม 4.5 คำสั่งของส่วนการแสดงกราฟในไฟล์เอชทีเอ็มแอล

```
<div class="card mb-3">
      <div class="card-header">
            <i class="fa fa-area-chart"></i>
            Area Chart Example
      </div>
      <div class="card-body">
            <canvas id="hotelChart" width="100%"
                  height="30"></canvas>
      </div>
</div>
```

สำหรับการแสดงกราฟจะใช้คำสั่งจากจาวาสคริปต์เป็นหลัก ในส่วนของเอชทีเอ็มแอลจะใช้สำหรับการกำหนดขนาดและบริเวณที่จะแสดงผลเท่านั้น (โปรแกรม 4.5) เนื่องจากมีไลบรารีของภาษาจาวาสคริปต์ที่ชื่อว่า Chart.js ซึ่งช่วยอำนวยความสะดวกในการแสดงกราฟอยู่ จึงได้นำมาใช้แทนการเขียนคำสั่งแสดงกราฟด้วยภาษาเอชทีเอ็มแอล

ในส่วนของคำสั่งการแสดงข้อมูลด้วยกราฟในภาษาจาวาสคริปต์ (โปรแกรม 4.4) สามารถสร้างกราฟด้วยคำสั่ง var myLineChart = new Chart() และกำหนดประเภทของกราฟให้เป็นกราฟเส้นด้วยการกำหนด type: 'line' สำหรับการนำข้อมูลของโรงแรมมาพล็อตกราฟ ได้มีการนำข้อมูลของวันที่ (value.date_checkprice) เก็บลงในตัวแปร checkDateData และข้อมูลของราคา (hotelValue.price) เก็บลงในตัวแปร priceData ด้วยคำสั่ง push เพื่อใช้สำหรับนำมาแสดงในกราฟ โดยที่ตัวแปร checkDateData จะถูกกำหนดให้แก่ labels หรือค่าในแกน X และตัวแปร priceData จะถูกกำหนดให้แก่ data หรือค่าในแกน Y เมื่อคำสั่งทำงาน จะทำการกำหนดจุดและแสดงข้อมูลที่ได้รับมาจนครบบนหน้าแสดงกราฟของเว็บแอปพลิเคชัน (รูป 4.6)



รูป 4.6 ข้อมูลของโรงแรมที่แสดงในกราฟ

4.1.4 การส่งออกข้อมูลเป็นไฟล์ประเภทซีเอสวี

การส่งออกข้อมูลเป็นไฟล์ประเภทซีเอสวีจะใช้วิธีการกดปุ่มเพื่อดาวน์โหลดไฟล์ที่มีข้อมูลของโรงแรมจากการค้นหาลงภายในเครื่อง โดยจะนำข้อมูลผลลัพธ์มาสร้างอีกตารางซึ่งถูกซ่อนไว้ (โปรแกรม 4.6) เนื่องจากว่าตารางที่แสดงผลกับตารางที่ซ่อนไว้สำหรับส่งออกข้อมูลมีข้อมูลไม่เหมือนกัน จึงไม่สามารถใช้ตารางเดียวกันได้ ดังนั้น เมื่อผู้ใช้งานกดปุ่มดาวน์โหลด จะเป็นการดาวน์โหลดข้อมูลจากตารางที่ถูกซ่อนไว้มาแทน ซึ่งข้อมูลของโรงแรมที่อยู่ภายในตารางที่ถูกซ่อนไว้สำหรับส่งออกข้อมูลมีดังนี้

- วันที่เก็บข้อมูล (value.date_crawling)
- วันที่ต้องการดูราคา (value.date_checkprice)
- ชื่อโรงแรม (hotelValue.hotel_name)
- ราคาของโรงแรม (hotelValue.price)
- ระดับดาวของโรงแรม (hotelValue.rate)
- สถานที่ (location)

โปรแกรม 4.6 การเตรียมข้อมูลลงในตารางด้วยคำสั่งในภาษาจาวาสคริปต์

```
function(data, status){
     console.log(data);
     var listHotel = []
     var location = localStorage.getItem("location")
     dataCSV = data
     $.each(data, function(key,value){
          $.each(value.list_hotel,function(index,
                 hotelValue){
               $("#dataTableCSV").append(
                    "<tr><td>" + value.date_crawling +
                    "</td><td>" + value.date_checkprice +
                    "</td><td>" + hotelValue.hotel_name +
                    "</td><td>" + hotelValue.price +
                    "</td><td>" + hotelValue.rate +
                    "</td><td>" + location + "</td></tr>"
               );
          })
     })
});
```

โปรแกรม 4.7 การซ่อนตารางในภาษาเอชทีเอ็มแอล

```
<table id="dataTableCSV" width="100%"
class="dataTableCSV" style="display:none;">
     <tr>
          <th>Date_Crawling</th>
          <th>Date_Checkprice</th>
          <th>Name</th>
          <th>Price</th>
          <th>Rate</th>
          <th>Location</th>
     </tr>
     <tbody>
     </tbody>
</table>
```

ในการนำข้อมูลลงในตารางจะใช้วิธีเดียวกันกับตารางที่ใช้แสดงผล แต่จะต่างกันตรงที่ตารางที่ถูกซ่อนไว้จะใช้คำสั่ง style="display:none;" เพิ่มเข้ามาเพื่อไม่ให้ผู้ใช้งานเห็นในหน้าแสดงตาราง (โปรแกรม 4.7)

สำหรับการดาวน์โหลดไฟล์ซีเอสวีลงในเครื่องจำเป็นต้องแปลงข้อมูลในตารางให้อยู่ในรูปแบบของข้อความตัวอักษรก่อนการดาวน์โหลด โดยจะใช้คำสั่ง (โปรแกรม 4.8) เพื่อเรียกใช้ฟังก์ชั่น prepCSVRow() สำหรับแปลงข้อมูลในตาราง

โปรแกรม 4.8 การแปลงข้อมูลให้อยู่ในรูปข้อความตัวอักษร (String)

```
/* Convert our data to CSV string */
var CSVString = prepCSVRow(titles, titles.length, '');
CSVString = prepCSVRow(data, titles.length, CSVString);
```

เมื่อทำการแปลงข้อมูลเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นคำสั่งเพื่อทำให้สามารถดาวน์โหลดไฟล์ได้ โดยจะเป็นการสร้างออบเจ็กต์และกำหนดยูอาร์แอลสำหรับดาวน์โหลด การกำหนดชื่อของไฟล์ซีเอสวีเมื่อดาวน์โหลดลงในเครื่อง

โปรแกรม 4.9 การกำหนดยูอาร์แอลและชื่อของไฟล์สำหรับดาวน์โหลด

```
/* Make CSV downloadable */
var downloadLink = document.createElement("a");
var blob = new Blob(["\ufeff", CSVString]);
var url = URL.createObjectURL(blob);
downloadLink.href = url;
downloadLink.download = "data.csv";
```

เมื่อทำคำสั่ง (โปรแกรม 4.9) สำเร็จแล้ว จึงเป็นการดาวน์โหลดไฟล์ซีเอสวีจริงลงในเครื่อง (โปรแกรม 4.10) ผ่านเว็บเบราว์เซอร์จากยูอาร์แอลที่ได้กำหนดไว้ ทำให้ได้ไฟล์ชื่อว่า data.csv ซึ่งภายในไฟล์จะมีข้อมูลของโรงแรมตามการค้นหาจากสถานที่และวันที่ในหน้าค้นหาของเว็บแอปพลิเคชัน (รูป 4.7)

โปรแกรม 4.10 การดาวน์โหลดไฟล์ซีเอสวี

```
/* Actually download CSV */
document.body.appendChild(downloadLink);
downloadLink.click();
document.body.removeChild(downloadLink);
```

	A	B	C	D	E	F
1	Date_Crawling	Date_Checkprice	Name	Price	Rate	Location
2	2018-04-17T00:00:00	2018-04-28T00:00:00	PP@Hotel	841	3.5	bangkok
3	2018-04-17T00:00:00	2018-04-28T00:00:00	Blue Sky Residence Airport	765	3.5	bangkok
4	2018-04-17T00:00:00	2018-04-28T00:00:00	Pinehurst Golf Club & Hotel	1500	3.5	bangkok
5	2018-04-17T00:00:00	2018-04-28T00:00:00	I Rich Residence	888	3	bangkok
6	2018-04-17T00:00:00	2018-04-28T00:00:00	Im-aim Apartel	595	3	bangkok
7	2018-04-17T00:00:00	2018-04-28T00:00:00	Tara Grand Hotel	1147	3	bangkok
8	2018-04-17T00:00:00	2018-04-28T00:00:00	Nonthaburi Palace Hotel	1104	3	bangkok
9	2018-04-17T00:00:00	2018-04-28T00:00:00	Aroonrunghouse	552	3	bangkok
10	2018-04-17T00:00:00	2018-04-28T00:00:00	Maxliving	425	3	bangkok
11	2018-04-17T00:00:00	2018-04-28T00:00:00	Baanthanam-nont	850	3	bangkok
12	2018-04-17T00:00:00	2018-04-28T00:00:00	PathumThani Place Hotel	977	3	bangkok
13	2018-04-17T00:00:00	2018-04-28T00:00:00	Lotus Condo Downtown	598	2	bangkok
14	2018-04-17T00:00:00	2018-04-28T00:00:00	Sakun Place	586	2.5	bangkok
15	2018-04-17T00:00:00	2018-04-28T00:00:00	Bay Hotel Suvarnabhumi Airport	1019	3	bangkok
16	2018-04-17T00:00:00	2018-04-28T00:00:00	Mida Hotel Ngamwongwan	1840	3.5	bangkok
17	2018-04-17T00:00:00	2018-04-28T00:00:00	The Pride	819	3	bangkok
18	2018-04-17T00:00:00	2018-04-28T00:00:00	Paeva Luxury Serviced Residence	1444	3.5	bangkok
19	2018-04-17T00:00:00	2018-04-28T00:00:00	Grand Ratchapruek Hotel	1954	3.5	bangkok
20	2018-04-17T00:00:00	2018-04-28T00:00:00	13 Coins Bangyai Hotel	794	3	bangkok

รูป 4.7 ตัวอย่างข้อมูลบางส่วนในไฟล์ซีเอสวี

4.2 การทดลองส่วนจัดการระบบ

4.2.1 การรันเว็บเซิร์ฟเวอร์สำหรับการทำงานของเว็บแอปพลิเคชัน

การใช้งานเว็บแอปพลิเคชันจำเป็นต้องรันเว็บเซิร์ฟเวอร์เพื่อให้ส่วนหน้าบ้านและส่วนหลังบ้านติดต่อกันเพื่อรับหรือส่งข้อมูลระหว่างกันได้ โดยจะใช้แจงโก้เฟรมเวิร์คเป็นเว็บเซิร์ฟเวอร์ ซึ่งสามารถสั่งการให้เว็บเซิร์ฟเวอร์ทำงานได้ด้วยคำสั่ง (ตัวอย่าง 4.1) ที่โฟลเดอร์โปรเจคของแจงโก้เฟรมเวิร์คที่สร้างไว้แล้ว เมื่อเซิร์ฟเวอร์สามารถรันได้สำเร็จ (รูป 4.8) ก็จะสามารถใช้งานเว็บแอปพลิเคชันได้

ตัวอย่าง 4.1 คำสั่งรันเว็บเซิร์ฟเวอร์ที่คอมมานด์ไลน์

```
python manage.py runserver
```

```
(noojaneenv) λ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
May 07, 2018 - 19:30:07
Django version 2.0.2, using settings 'webcrawling.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

รูป 4.8 ภายในคอมมานด์ไลน์เมื่อรันเซิร์ฟเวอร์สำเร็จ

4.2.2 การสร้างช่องทางสำหรับติดต่อกับเว็บแอปพลิเคชัน

การติดต่อระหว่างส่วนติดต่อผู้ใช้งานของเว็บแอปพลิเคชันกับส่วนการจัดการระบบจะใช้การสื่อสารกันผ่านเอพีไอ (รูป 4.9) ซึ่งในโครงงานชิ้นนี้ได้เลือกใช้สแวกเกอร์เอพีไอที่มีส่วนต่อประสานกับผู้ใช้ให้ใช้งาน (รูป 4.10) สามารถเข้าใช้งานได้จากเว็บเบราว์เซอร์ผ่านยูอาร์แอล http://localhost:8000/api#/

รูป 4.9 การติดต่อระหว่าง Front-end กับ Back-end ด้วย API

รูป 4.10 ส่วนต่อประสานกับผู้ใช้ของสแวกเกอร์เอพีไอ

การใช้งานสแวกเกอร์เอพีไอจะทำการตั้งค่าพาท (path) และรูปแบบโครงสร้างของข้อมูลที่จะทำการรับและส่งระหว่างส่วนหน้าบ้านกับส่วนหลังบ้าน โดยจะทำการเพิ่มคำสั่งต่าง ๆ ลงในไฟล์ webcrawling\settings.py (โปรแกรม 4.11) และในไฟล์ webcrawling\urls.py (โปรแกรม 4.12) ซึ่งไฟล์ทั้งคู่นั้นจะอยู่ในโปรเจคของแจงโก้ เมื่อสำเร็จจะสามารถเข้าไปที่หน้าส่วนต่อประสานกับผู้ใช้ของสแวกเกอร์เอพีไอได้ด้วยยูอาร์แอล http://localhost:8000/api#/ ผ่านเว็บเบราเซอร์

ไฟล์ settings.py (โปรแกรม 4.11) เป็นไฟล์สำหรับตั้งค่าในการนำส่วนที่ต้องใช้งานเพิ่มเติมนอกเหนือจากที่แจงโก้เฟรมเวิร์คมีให้จากภายนอกเข้ามาใช้งาน โดยจะทำการกำหนดโดยนำชื่อของสิ่งที่จำเป็นต้องใช้งานระบุไว้ใน INSTALLED_APPS ก็จะสามารถนำส่วนที่เพิ่มเติมเข้ามาใช้งานภายในโปรเจคได้

ไฟล์ urls.py (โปรแกรม 4.12) จะเป็นการสร้างสแวกเกอร์เอพีไอสำหรับนำมาใช้งานจากคำสั่ง get_swagger_view() ที่กำหนดชื่อของเอพีไอจากคำสั่ง title="webcrawling API" สามารถสร้างช่องทางการรับ-ส่งข้อมูลจากแต่ละเว็บเพจของเว็บแอปพลิเคชันด้วยคำสั่ง router.register() และกำหนดพาทสำหรับการเข้าใช้งานส่วนต่อประสานกับผู้ใช้ของสแวกเกอร์เอพีไอที่ urlpatterns

โปรแกรม 4.11 การตั้งค่าสแวกเกอร์เอพีไอที่ไฟล์ settings.py ในโปรเจคของแจงโก้เฟรมเวิร์ค

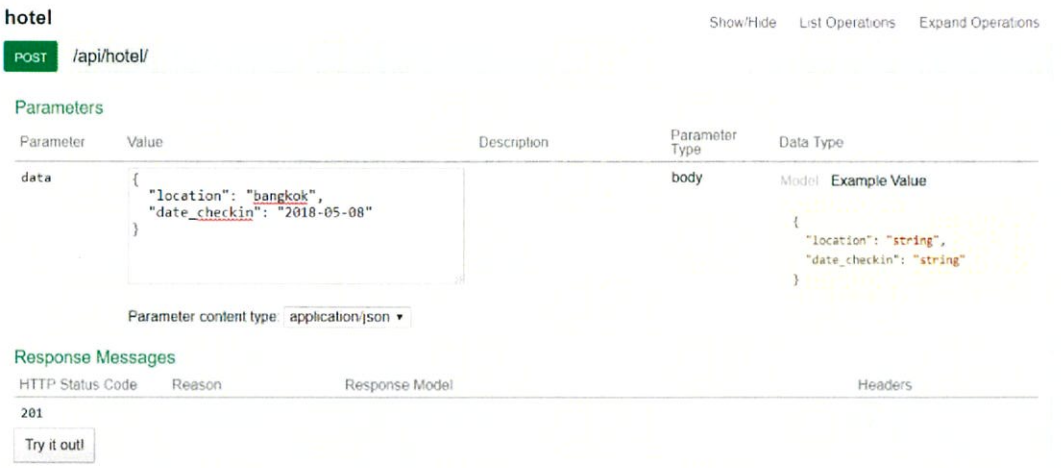
```
INSTALLED_APPS = [

     ...,

    'rest_framework',
    'rest_framework_swagger'
]
```

โปรแกรม 4.12 การตั้งค่าและนำสแวกเกอร์เอพีไอมาใช้งานที่ไฟล์ urls.py ในโปรเจคของแจงโก้เฟรมเวิร์ค

```
from rest_framework_swagger.views import get_swagger_view
from rest_framework.routers import DefaultRouter

from hotel.views import HotelViewSet, ChartViewSet

schema_view = get_swagger_view(title="webcrawling API")
router = DefaultRouter()

router.register(r'hotel', HotelViewSet)
router.register(r'chart', ChartViewSet)

urlpatterns = [
    path('api', schema_view),
    path('api/', include(router.urls))
 ]
```

สามารถทดสอบการรับหรือส่งข้อมูลระหว่างส่วนติดต่อผู้ใช้งานของเว็บแอปพลิเคชันกับระบบหลังบ้านได้จากส่วนต่อประสานกับผู้ใช้ของสแวกเกอร์เอพีไอ โดยสามารถเลือกส่วนที่จะทำการทดสอบจากหน้าจอได้

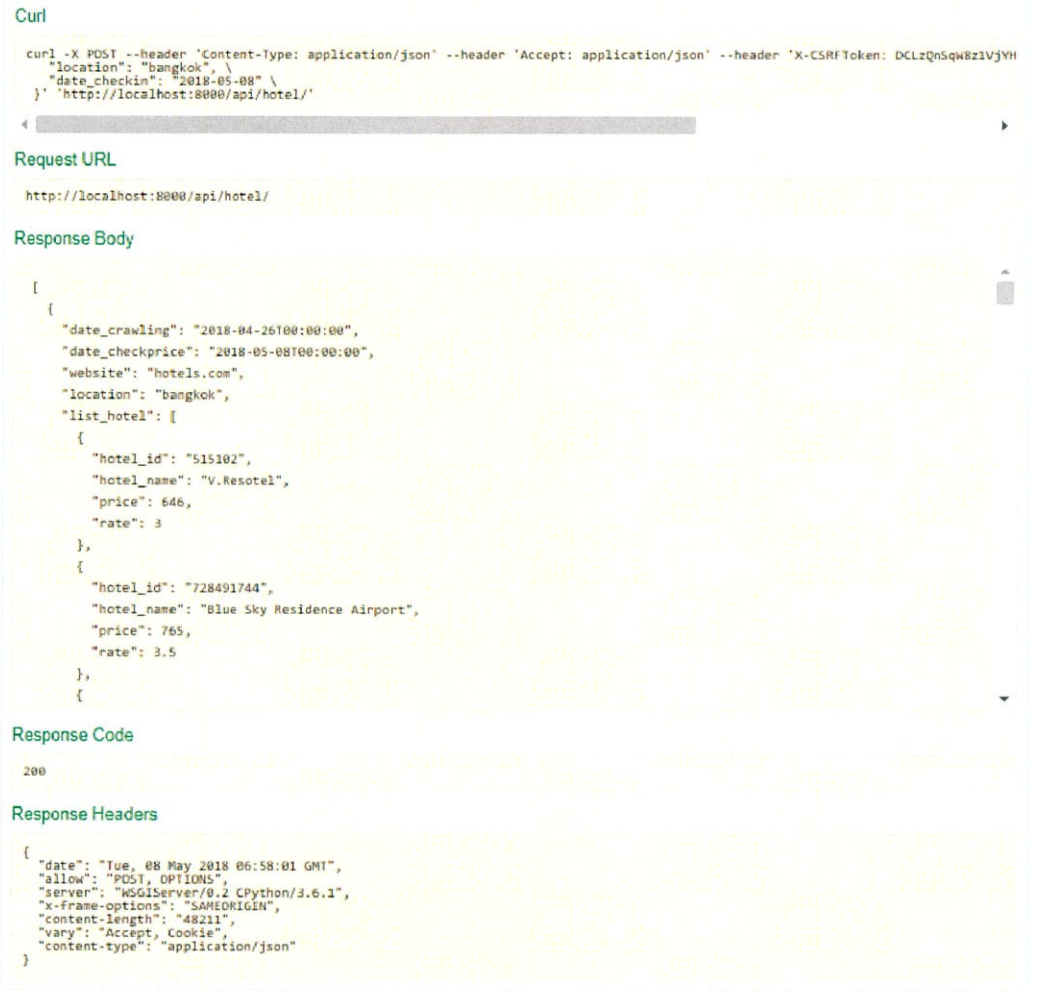
ในที่นี้เลือก hotel ซึ่งเป็นส่วนที่ใช้ติดต่อของหน้าค้นหาเพื่อส่งข้อมูลไปยังหน้าแสดงตารางของเว็บแอปพลิเคชัน จะเห็นว่าสามารถทดสอบได้โดยกำหนดค่าลงในช่อง Value ในส่วนของ data ซึ่งมีโครงสร้างของข้อมูลที่ถูกกำหนดไว้เพื่อใช้ส่งไปที่ส่วนหลังบ้าน หากกำหนดค่าของข้อมูลโดยให้ "location": "bangkok" และ "date_checkin": "2018-05-08" (รูป 4.11) แล้วกดที่ปุ่ม “Try it out!” จะเป็นการส่งข้อมูลไปที่ส่วนหลังบ้าน



รูป 4.11 การทดสอบการส่งข้อมูลโดยกำหนดค่าลงในช่อง Value ในส่วนของ data ในส่วนต่อประสานกับผู้ใช้ของสแวกเกอร์เอพีไอ

หลังจากนั้น ในส่วนหลังบ้านจะนำข้อมูลที่ได้ไปทำตามเงื่อนไขของอัลกอริทึมที่กำหนดไว้ หากข้อมูลที่ได้รับมานั้นเป็นไปตามเงื่อนไขใด ก็จะทำการส่งข้อมูลกลับไปที่ส่วนหน้าบ้านตามเงื่อนไขนั้น ซึ่งในการทดสอบนี้ ต้องการข้อมูลของโรงแรมที่อยู่ในพื้นที่ bangkok ของวันที่ 8 พฤษภาคม 2018 จากฐานข้อมูล ส่วนจัดการระบบจึงต้องนำข้อมูลไปค้นหาในฐานข้อมูลและเมื่อได้ผลลัพธ์จะทำการคืนค่ากลับไปเป็นอาร์เรย์ของออบเจ็กต์ตามที่ได้กำหนดไว้ในเงื่อนไขของการค้นหาข้อมูลจากฐานข้อมูล

ในหน้าจอของสแวกเกอร์เอพีไอจะแสดงข้อมูลที่ได้รับกลับมาจากส่วนหลังบ้านในส่วนของ Response Body ถ้าการทดสอบนั้นถูกต้อง สามารถนำยูอาร์แอลจากส่วน Request URL ไปใช้งานได้ทันที หากข้อมูลที่ได้นั้นไม่ถูกต้อง ควรตรวจสอบทั้งในส่วนคำร้องขอของเว็บแอปพลิเคชันและเงื่อนไขการค้นหาจากส่วนหลังบ้านให้ถูกต้องก่อน



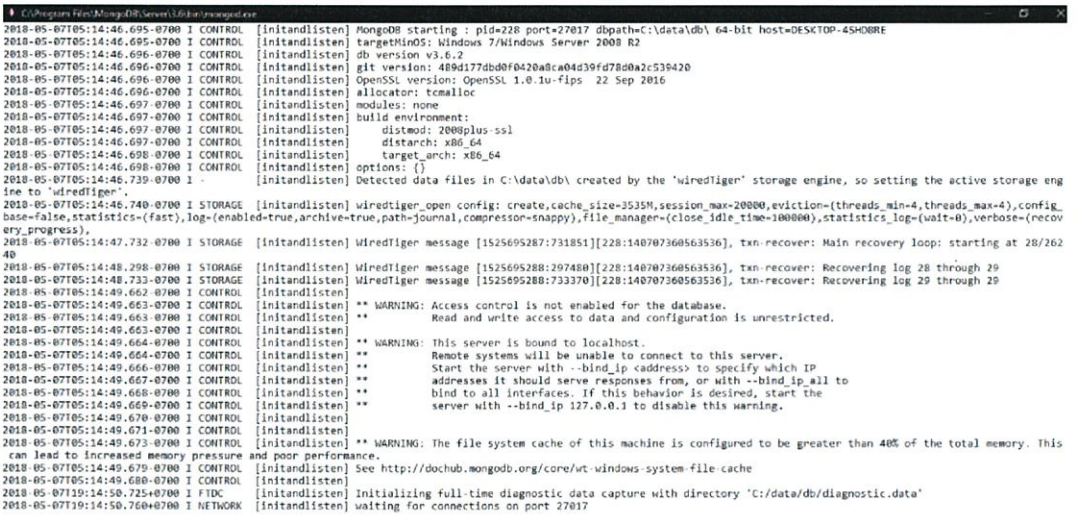
รูป 4.12 การทดสอบการรับข้อมูลในช่อง Response Body ในส่วนต่อประสานกับผู้ใช้ของสแวกเกอร์เอพีไอ

4.2.3 การเชื่อมต่อกับฐานข้อมูล

การใช้แจงโก้เชื่อมต่อกับมองโกดีบีนั้นสามารถทำได้โดยการนำมองโกเอนจินมาใช้ในโปรเจคด้วยคำสั่ง from mongoengine import * ที่ไฟล์ webcrawling\webcrawling\settings.py แล้วทำการเชื่อมต่อกับฐานข้อมูลด้วยคำสั่ง connect('ชื่อฐานข้อมูล') (โปรแกรม 4.8) จากนั้นจึงลองเชื่อมต่อกันฐานข้อมูลด้วยการรันคอมมานด์ mongod ขึ้นมา ถ้าหากเชื่อมต่อสำเร็จ (รูป 4.13) จะสามารถใช้งานฐานข้อมูลได้

โปรแกรม 4.13 การเชื่อมต่อกับฐานข้อมูล

```
from mongoengine import *
connect('mydb')
```



รูป 4.13 การเชื่อมต่อกับฐานข้อมูลที่สำเร็จ

4.2.4 การกำหนดโครงสร้างของข้อมูล

เป็นการกำหนดโครงสร้างของข้อมูลก่อนที่จะทำการเก็บข้อมูลลงบนฐานข้อมูลมองโกดีบี เพื่อแก้ปัญหาที่อาจเกิดขึ้นเกี่ยวกับการคิวรีข้อมูล เช่น อาจหาข้อมูลไม่พบเนื่องจากชื่อของข้อมูลไม่ถูกต้อง เพราะว่ามองโกดีบีเป็นฐานข้อมูลแบบไม่มีความสัมพันธ์ของโครงตาราง ทำให้สามารถเก็บข้อมูลได้แม้ว่าโครงสร้างข้อมูลจะไม่เหมือนกัน จึงได้นำมองโกเอนจินมาเพื่อใช้สำหรับการกำหนดโครงสร้างและตรวจสอบความถูกต้องสำหรับเอกสารที่จะจัดเก็บลงมองโกดีบี เพื่อให้ข้อมูลที่จะทำการจัดเก็บมีโครงสร้างที่ถูกต้องและเป็นรูปแบบเดียวกันทั้งหมด

การกำหนดโครงสร้างของข้อมูลนั้น สามารถเขียนคำสั่งลงได้ที่ webcrawling\hotel\models.py (โปรแกรม 4.14) และในการกำหนดโครงสร้างของข้อมูลจะกำหนดตามโครงสร้างของภาษาเจสัน (ตัวอย่าง 4.2)

โปรแกรม 4.14 คำสั่งเพื่อกำหนดโครงสร้างของข้อมูลโรงแรมด้วย mongoengine

```
from mongoengine import *
from datetime import datetime

class Room(EmbeddedDocument):
    hotel_id = StringField(max_length=200, required=True)
    HotelName = StringField(max_length=200,
                            required=True)
    Price = FloatField(required=True)
    rate = FloatField(required=True)
```

โปรแกรม 4.14 คำสั่งเพื่อกำหนดโครงสร้างของข้อมูลโรงแรมด้วย mongoengine (ต่อ)

```
class Hotel(Document):
    dateCrawling = DateTimeField(default=datetime.now())
    dateCheckPrice = DateTimeField(default=
                                    datetime.now())
    website = StringField(max_length=200, required=True)
    location = StringField(max_length=200, required=True)
    listHotel = ListField(EmbeddedDocumentField(Room))
```

รายละเอียดของแต่ละข้อมูลที่เก็บตามโครงสร้าง มีดังต่อไปนี้

1) dateCrawling: Date (วันที่ให้บอททำการเก็บข้อมูล)
2) dateCheckPrice: Date (วันที่ต้องการดูราคาโรงแรม)
3) website: String (เว็บไซต์ที่ทำการเก็บข้อมูล)
4) location: String (สถานที่ที่ต้องการค้นหาโรงแรม)
5) listHotel: [Document] (อาร์เรย์ของรายการโรงแรม)

ซึ่ง Document ของรายการโรงแรมจะประกอบไปด้วยข้อมูล ดังนี้

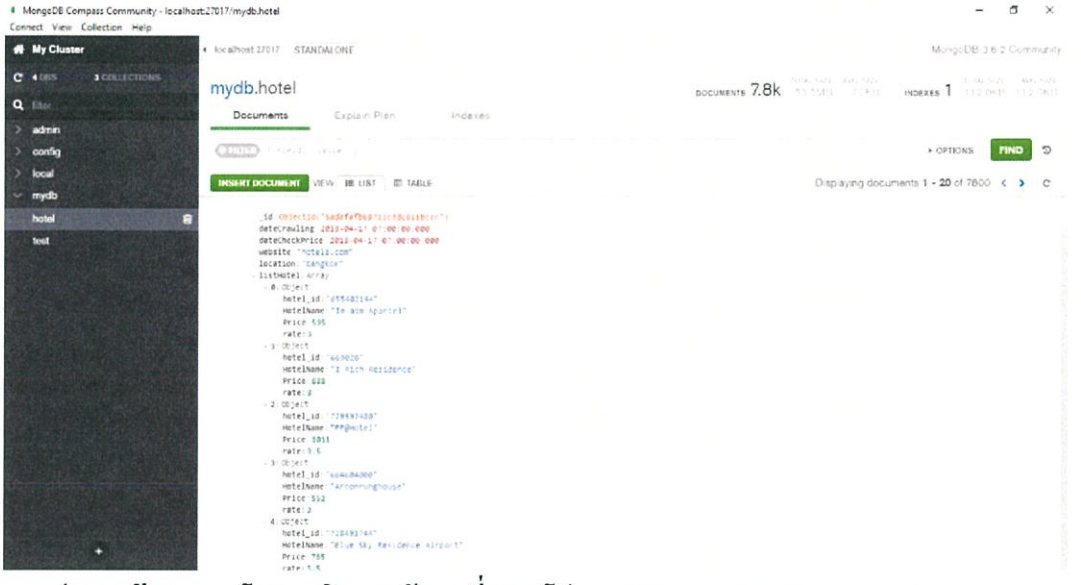
1) hotel_id: String (ไอดีของโรงแรม)
2) HotelName: String (ชื่อโรงแรม)
3) Price: Float (ราคาของโรงแรม)
4) rate: Float (ระดับดาวของโรงแรม)

ตัวอย่าง 4.2 โครงสร้างข้อมูลของข้อมูลโรงแรมแบบเจสัน

```
dateCrawling: Date,
dateCheckPrice: Date,
website: String,
location: String,
listHotel: {
     [
          hotel_id: String,
          HotelName: String,
          Price: Float,
          rate: Float
     ]
}
```

เมื่อทำการเก็บข้อมูลลงในฐานข้อมูลแล้ว สามารถดูข้อมูลภายในฐานข้อมูลได้จากโปรแกรม MongoDB Compass Community แล้วเลือกฐานข้อมูลที่ใช้เก็บ ซึ่งในโครงงานชิ้นนี้จะใช้ฐานข้อมูล mydb.hotel เมื่อกดเข้าไปในฐานข้อมูลจะพบว่า ข้อมูลภายในฐานข้อมูลจะถูกจัดเก็บ

ในรูปแบบของออบเจ็กต์ตามวันที่ให้บอทไปทำการเก็บข้อมูลมา และแต่ละออบเจ็กต์จะมีออบเจ็กต์ย่อยถูกเก็บในอาร์เรย์ของ listHotel ซึ่งเก็บข้อมูลของแต่ละโรงแรมอยู่



รูป 4.14 ข้อมูลของโรงแรมในฐานข้อมูลที่ดูจากโปรแกรม MongoDB Compass Community

4.3 การทดลองส่วนรวบรวมข้อมูล

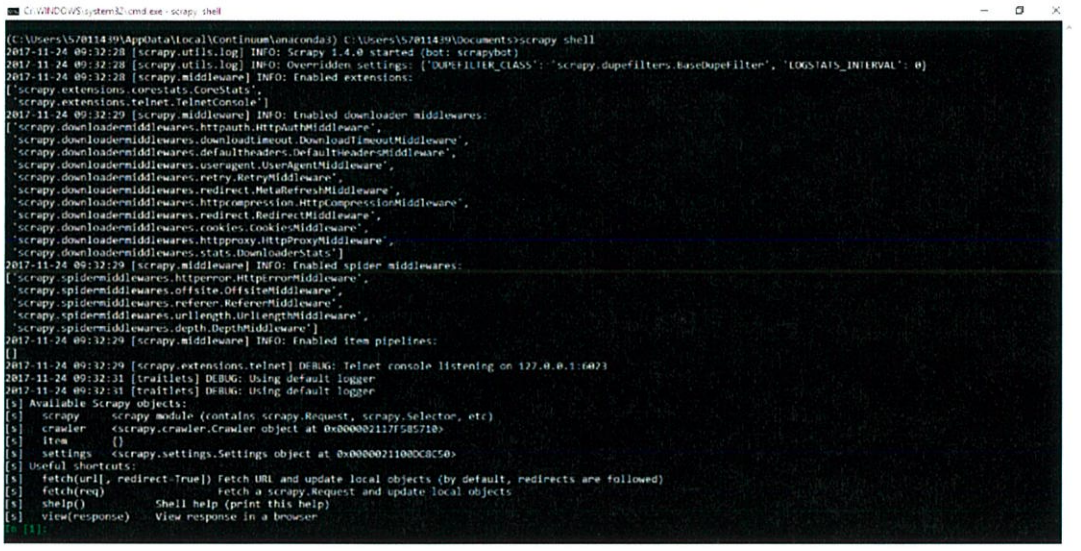
การทดลองสร้างบอทเพื่อให้บอทสามารถเก็บข้อมูลได้ตามที่ต้องการ ตามโครงสร้างของข้อมูลและเว็บไซต์ที่ได้กำหนดไว้แล้ว

4.3.1 การทดลองการเรียกใช้บอท

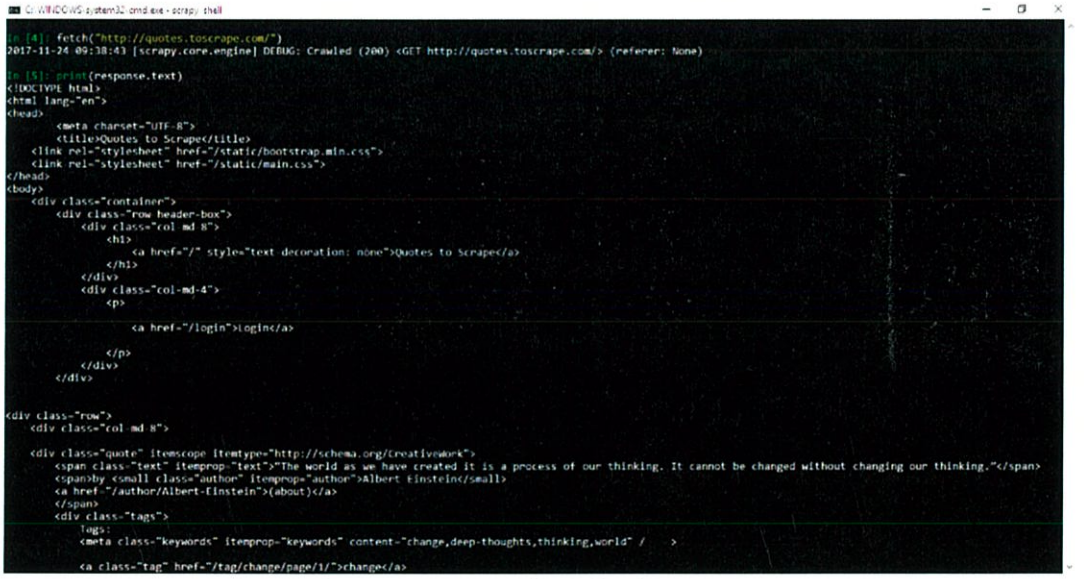
เป็นการเรียกใช้งานสไปเดอร์เพื่อให้ไปทำการเก็บข้อมูล ซึ่งสามารถเรียกใช้งานสไปเดอร์ได้ด้วยวิธี ดังต่อนี้

4.3.1.1 Scrapy shell

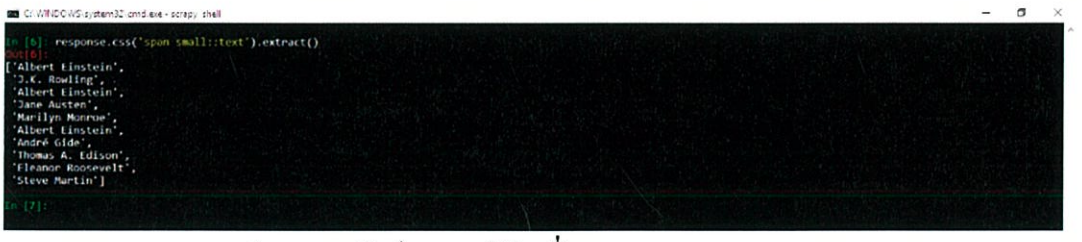
เป็นการเรียกใช้สไปเดอร์ผ่าน scrapy shell (รูป 4.15) ซึ่งจะทำการ fetch หน้าเว็บที่ต้องการลงมาเก็บในตัวแปร response (รูป 4.16) จากนั้นจะใช้คำสั่ง response.css().extract() เพื่อทำการแยกข้อมูลที่ต้องการออกมา (รูป 4.17)



รูป 4.15 Scrapy shell



รูป 4.16 การ Fetch หน้าเว็บไซต์และแสดงผล



รูป 4.17 ผลลัพธ์ของการใช้คำสั่ง response.css().extract()

4.3.1.2 สไปเดอร์จากไลบรารี Scrapy

สามารถเรียกใช้งานได้ 2 วิธี ดังนี้

1) Call by command

เป็นการสร้างสไปเดอร์ผ่านคอมมานด์ไลน์ โดยจะสร้างโปรเจค scrapy ด้วยคำสั่ง scrapy startproject projectname จากนั้นจะทำการสร้างสไปเดอร์เพื่อให้ไปเก็บข้อมูลที่ยูอาร์แอลนั้น ๆ ด้วยคำสั่ง scrapy genspider spidername URL หลังจากนั้น จะทำการแก้ไขโค้ดในส่วนของตัวสไปเดอร์ ก่อนจะทำการสั่งให้สไปเดอร์เริ่มเก็บข้อมูลด้วยคำสั่ง scrapy crawl spidername

```
(C:\Users\57011439\AppData\Local\Continuum\anaconda3) C:\Users\57011439\Documents>scrapy startproject test3
New Scrapy project 'test3', using template directory 'C:\\Users\\57011439\\AppData\\Local\\Continuum\\anaconda3
\\lib\\site-packages\\scrapy\\templates\\project', created in:
    C:\Users\57011439\Documents\test3

You can start your first spider with:
    cd test3
    scrapy genspider example example.com
```

รูป 4.18 การสร้างโปรเจคและการสร้างสไปเดอร์

2) Call by script

แบ่งได้ 2 วิธี คือ

- **Process** โดยปกติ สไปเดอร์จะรัน 1 ตัวต่อ 1 โปรเซส แต่หากทำการเพิ่มโค้ดในไฟล์ spiders.py ดังโปรแกรม 4.15 จะทำให้สไปเดอร์รันได้หลายตัวต่อ 1 โปรเซส

โปรแกรม 4.15 การรันสไปเดอร์หลายตัวต่อ 1 โปรเซส

```
import scrapy
from scrapy.crawler import CrawlerProcess

class MySpider1(scrapy.Spider):
    # Your first spider definition
 ...

class MySpider2(scrapy.Spider):
    # Your second spider definition
 ...

process = CrawlerProcess()
```

โปรแกรม 4.15 การรันสไปเดอร์หลายตัวต่อ 1 โปรเซส (ต่อ)

```
process.crawl(MySpider1)
process.crawl(MySpider2)
process.start() # the script will block here until all
crawling jobs are finished
```

- **Command-line** คล้ายกับการ Call by command แต่เป็นการสั่งผ่านสคริปต์ ดังโปรแกรม 4.16 ต่อไปนี้

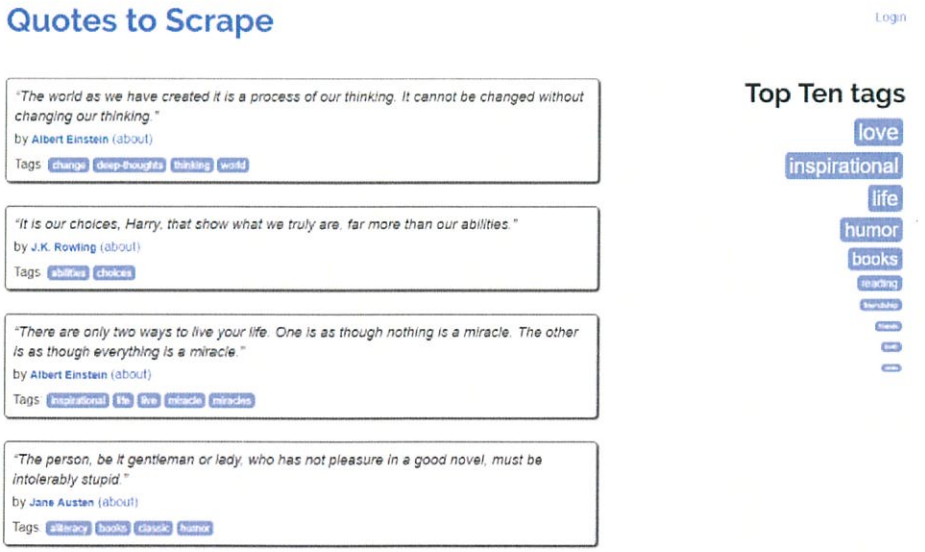
โปรแกรม 4.16 การสั่งสไปเดอร์ผ่านสคริปต์

```
cmdline.execute(("scrapy crawl ylpBot -a
start_url=https://www.yellowpages.com/search?search_terms
=hotel&geo_location_terms="+keyWord).split())
```

4.3.2 การทดสอบไลบรารีที่ใช้สำหรับรวบรวมข้อมูล

ทดสอบเพื่อเปรียบเทียบไลบรารีทั้ง 2 ได้แก่ scrapy และ pyspider เพื่อเลือกไลบรารีที่เหมาะแก่การนำไปใช้ในโครงงาน พัฒนาได้ง่าย นำไปใช้ได้หลายรูปแบบ

โดยทำการทดลองเก็บข้อมูล ได้แก่ ข้อความ(text) ชื่อผู้แต่ง(author) และแท็ก(tags) จากเว็บไซต์ที่กำหนด คือ http://quotes.toscrape.com/ (รูป 4.19) แล้วนำผลลัพธ์ที่ได้มาเปรียบเทียบกัน



รูป 4.19 เว็บไซต์ http://quotes.toscrape.com/

รูป 4.20 การสำรวจแท็ก (tag) เพื่อเก็บข้อมูล

โปรแกรม 4.17 โค้ดตัวอย่างการรวบรวมข้อมูลของ scrapy

```
class TestbotSpider(scrapy.Spider):
    name = 'testbot'

    allowed_domains = ['http://quotes.toscrape.com/']
    start_urls = ['http://quotes.toscrape.com/']

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text':
quote.css('span.text::text').extract(),
                'author': quote.css('span
small::text').extract(),
                'tags': quote.css('div.tags
a.tag::text').extract(),
            }

        next_page = response.css('li.next
a::attr(href)').extract.first()
        if next_page is not None:
            yield response.follow(next_page,
callback=self.parse)
```

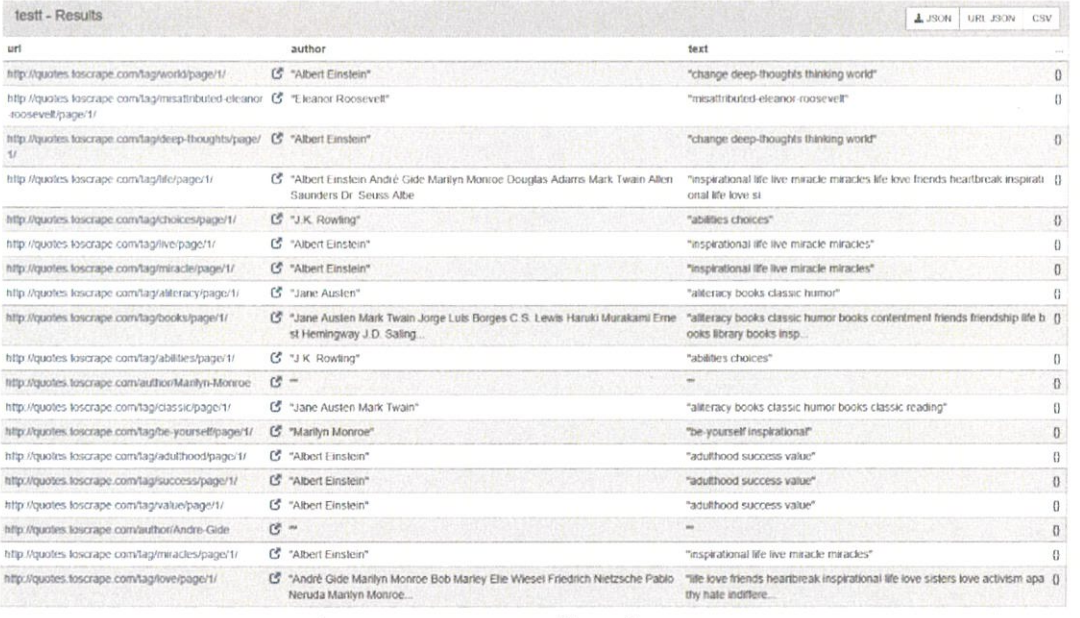
โปรแกรม 4.18 โค้ดตัวอย่างการรวบรวมข้อมูลของ pyspider

```
from pyspider.libs.base_handler import *


 class Handler(BaseHandler):
    crawl_config = {
    }

    @every(minutes=24*60)
    def on_start(self):
        self.crawl('http://quotes.toscrape.com/',
callback=self.index_page)

    @config(age=10*24*60*60)
    def index_page(self, response):
        for each in
response.doc('a[href^="http"]').items():
            self.crawl(each.attr.href,
callback=self.detail_page)

    @config(priority=2)
    def detail_page(self, response):
        return {
   "text": response.doc('span.text').text(),
   "author": response.doc('span small').text(),
   "tags": response.doc('div.tags a.tag').text(),
        }
```

จากการทดสอบ จะได้ตัวอย่างข้อมูลที่ได้จากการรวบรวมของไลบรารีทั้ง 2 มีดังนี้

text	author	tags
"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."	Albert Einstein	change,deep-thoughts,thinking,world
"It is our choices, Harry, that show what we truly are, far more than our abilities."	J.K. Rowling	abilities,choices
"There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle."	Albert Einstein	inspirational,life,live,miracle,miracles
"The person, be it gentleman or lady, who has not pleasure in a good novel, must be intolerably stupid."	Jane Austen	aliteracy,books,classic,humor
"Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than absolutely boring."	Marilyn Monroe	be-yourself,inspirational
"Try not to become a man of success. Rather become a man of value."	Albert Einstein	adulthood,success,value
"It is better to be hated for what you are than to be loved for what you are not."	André Gide	life,love
"I have not failed. I've just found 10,000 ways that won't work."	Thomas A. Edison	edison,failure,inspirational,paraphrased
"A woman is like a tea bag; you never know how strong it is until it's in hot water."	Eleanor Roosevelt	misattributed-eleanor-roosevelt
"A day without sunshine is like, you know, night."	Steve Martin	humor,obvious,simile

รูป 4.21 ผลการรวบรวมข้อมูลด้วย scrapy



testt - Results

JSON | URL JSON | CSV

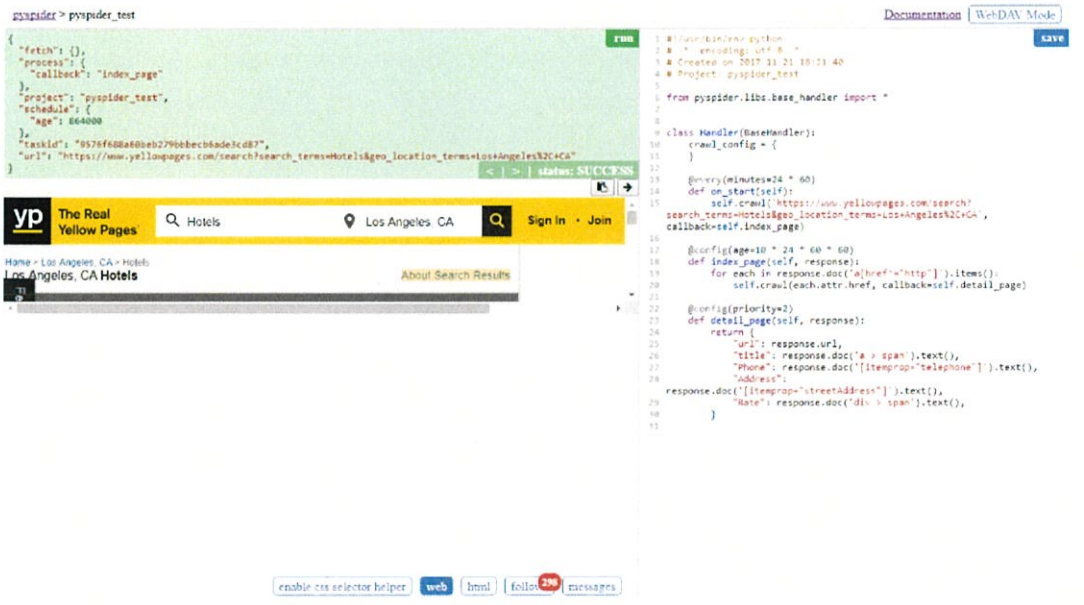
url	author	text	...
http://quotes.toscrape.com/tag/world/page/1/	"Albert Einstein"	"change deep-thoughts thinking world"	{}
http://quotes.toscrape.com/tag/misattributed-eleanor-roosevelt/page/1/	"Eleanor Roosevelt"	"misattributed-eleanor-roosevelt"	{}
http://quotes.toscrape.com/tag/deep-thoughts/page/1/	"Albert Einstein"	"change deep-thoughts thinking world"	{}
http://quotes.toscrape.com/tag/life/page/1/	"Albert Einstein André Gide Marilyn Monroe Douglas Adams Mark Twain Allen Saunders Dr. Seuss Albe	"inspirational life live miracle miracles life love friends heartbreak inspirational life love si	{}
http://quotes.toscrape.com/tag/choices/page/1/	"J.K. Rowling"	"abilities choices"	{}
http://quotes.toscrape.com/tag/live/page/1/	"Albert Einstein"	"inspirational life live miracle miracles"	{}
http://quotes.toscrape.com/tag/miracle/page/1/	"Albert Einstein"	"inspirational life live miracle miracles"	{}
http://quotes.toscrape.com/tag/aliteracy/page/1/	"Jane Austen"	"aliteracy books classic humor"	{}
http://quotes.toscrape.com/tag/books/page/1/	"Jane Austen Mark Twain Jorge Luis Borges C.S. Lewis Haruki Murakami Ernest Hemingway J.D. Saling...	"aliteracy books classic humor books contentment friends friendship life books library books insp...	{}
http://quotes.toscrape.com/tag/abilities/page/1/	"J.K. Rowling"	"abilities choices"	{}
http://quotes.toscrape.com/author/Marilyn-Monroe	""	""	{}
http://quotes.toscrape.com/tag/classic/page/1/	"Jane Austen Mark Twain"	"aliteracy books classic humor books classic reading"	{}
http://quotes.toscrape.com/tag/be-yourself/page/1/	"Marilyn Monroe"	"be-yourself inspirational"	{}
http://quotes.toscrape.com/tag/adulthood/page/1/	"Albert Einstein"	"adulthood success value"	{}
http://quotes.toscrape.com/tag/success/page/1/	"Albert Einstein"	"adulthood success value"	{}
http://quotes.toscrape.com/tag/value/page/1/	"Albert Einstein"	"adulthood success value"	{}
http://quotes.toscrape.com/author/Andre-Gide	""	""	{}
http://quotes.toscrape.com/tag/miracles/page/1/	"Albert Einstein"	"inspirational life live miracle miracles"	{}
http://quotes.toscrape.com/tag/love/page/1/	"André Gide Marilyn Monroe Bob Marley Elie Wiesel Friedrich Nietzsche Pablo Neruda Marilyn Monroe...	"life love friends heartbreak inspirational life love sisters love activism apathy hate indifere...	{}

รูป 4.22 ผลการรวบรวมข้อมูลด้วย pyspider

จากผลการค้นหา พบว่า ข้อมูลที่เก็บได้จากบอทของไลบรารี scrapy มีความถูกต้องตามที่ต้องการทั้งหมด แต่ข้อมูลที่เก็บได้จากบอทของไลบรารี pyspider ยังมีข้อมูลบางส่วนที่ไม่ตรงตาม

ความต้องการ เนื่องจากสไปเดอร์ของไลบรารี pyspider ไม่ได้ทำการเก็บข้อมูลทีละหน้าเหมือนกับบอทของไลบรารี scrapy แต่จะทำการ Follow link ที่พบไปเรื่อย ๆ หากต้องการเก็บข้อมูลที่ใช้บอทจากไลบรารี pyspider เพื่อให้ได้ข้อมูลที่ตรงตามความต้องการ จำเป็นจะต้องทำการคัดกรองข้อมูลที่บอทเก็บมาก่อนการนำไปใช้งาน

รูป 4.23 ส่วนต่อประสานกับผู้ใช้ของไลบรารี pyspider ส่วนมอนิเตอร์จัดการโปรเจค



รูป 4.24 ส่วนต่อประสานกับผู้ใช้ของไลบรารี pyspider ส่วนการเขียนโค้ด

ในด้านการใช้งาน ไลบรารี pyspider จะมีส่วนต่อประสานกับผู้ใช้สำหรับการสร้างโปรเจคมาให้ (รูป 4.23) สามารถเขียนโค้ดที่ส่วนต่อประสานกับผู้ใช้ (รูป 4.24) เพื่อกำหนดลักษณะการเก็บข้อมูลของบอท ทำให้มีการใช้งานได้ง่าย ไม่จำเป็นต้องหาแท็กเอชทีเอ็มแอลของข้อมูลที่ต้องการก่อนเหมือนไลบรารี scrapy แต่ในขณะเดียวกันก็ทำให้ไลบรารีนำมาประยุกต์ใช้กับระบบให้เป็นไปตามความต้องการได้ยากกว่าการนำไลบรารี scrapy มาใช้ เพราะไลบรารี scrapy สามารถตั้งค่าได้มากกว่า มีความยืดหยุ่นของตัวสไปเดอร์มากกว่า จึงนำไปใช้งานตามความต้องการและทำงาน

ร่วมกันกับโมดูลอื่นได้มากกว่า แต่เนื่องจากผู้พัฒนาโครงงานไม่สามารถทดสอบประสิทธิภาพในตอนนี้ได้ จึงเลือกที่จะใช้ไลบรารี scrapy มากกว่า ในอนาคตอาจมีการทดสอบเปรียบเทียบไลบรารีอีกครั้ง

4.3.3 ทำการสำรวจความแตกต่างของยูอาร์แอลในการค้นหาโรงแรมจากตัวอย่างเว็บไซต์ที่กำหนด

เพื่อสร้างรูปแบบการค้นหาจากยูอาร์แอล เนื่องจากในการค้นหา จะทำการกำหนดคีย์เวิร์ดแล้วนำไปใส่ในยูอาร์แอลเพื่อไปยังหน้าเว็บไซต์นั้น ๆ จึงทำการเลือกบางเว็บไซต์ที่ใช้สำหรับจองโรงแรมเพื่อสร้างรูปแบบของยูอาร์แอลตัวอย่าง ซึ่งจะเลือกเว็บไซต์จองโรงแรมที่ได้รับความนิยมจำนวน 10 เว็บไซต์ ดังนี้

1) Booking.com: http://www.booking.com
2) Hotels.com: http://www.hotels.com
3) Agoda: http://www.agoda.com
4) Tripadvisor: http://www.tripadvisor.com
5) Trivago: http://www.trivago.co.th
6) Traveloka: http://www.traveloka.com
7) Expedia.co.th: http://www.expedia.co.th
8) HotelsCombined: http://www.hotelscombined.co.th
9) HOTELHUNTER: http://www.hotelhunter.com
10) Skyscanner: http://www.skyscanner.co.th/hotels

โดยทำการกำหนดคีย์เวิร์ดของการค้นหาให้เหมือนกันทุกเว็บไซต์ ได้แก่ สถานที่ วันที่เช็คอิน-เช็คเอาท์ จำนวนคืนที่เข้าพัก จำนวนคนและจำนวนห้องพักแล้วทำการค้นหา จากนั้นจะนำยูอาร์แอลจากหน้าผลลัพธ์การค้นหาจากแต่ละเว็บไซต์มาทำการวิเคราะห์ส่วนประกอบ เพื่อหาค่าที่จำเป็นต้องกำหนดของส่วนประกอบของยูอาร์แอลในการค้นหาแต่ละครั้ง

จากผลการทดลองที่ได้ พบว่า ค่าที่จำเป็นต้องกำหนดในแต่ละเว็บไซต์มีดังต่อไปนี้

1) Booking.com: วันที่เช็คอิน-เช็คเอาท์ เลขไอดีของสถานที่(เลขที่กำหนดให้สถานที่เฉพาะเว็บไซต์นี้เท่านั้น)และประเภทของสถานที่
2) Hotels.com: สถานที่
3) Agoda: เลขไอดีของสถานที่
4) Tripadvisor: เลขไอดีของสถานที่และสถานที่
5) Trivago: เลขไอดีของสถานที่

6) Traveloka: วันที่เช็คอิน-เช็คเอาท์ จำนวนคืน จำนวนห้อง เลขไอดีของสถานที่ สถานที่และจำนวนคน
7) Expedia.co.th: สถานที่
8) HotelsCombined: สถานที่และวันที่เช็คอิน-เช็คเอาท์
9) HOTELHUNTER: สถานที่และวันที่เช็คอิน-เช็คเอาท์
10) Skyscanner: สถานที่และเลขไอดีของสถานที่

ตาราง 4.1 ค่าที่จำเป็นต้องกำหนดในแต่ละเว็บไซต์สำหรับค้นหาโรงแรม

เว็บไซต์	วันที่เช็คอิน - เช็คเอาท์	สถานที่	ประเภทสถานที่	ไอดีสถานที่	จำนวนคืน	จำนวนห้อง	จำนวนคน
Booking.com	✓		✓	✓			
Hotels.com		✓					
Agoda				✓			
Tripadvisor		✓		✓			
Trivago				✓			
Traveloka	✓	✓		✓	✓	✓	✓
Expedia.co.th		✓					
HotelsCombined	✓	✓					
HOTELHUNTER	✓	✓					
Skyscanner		✓		✓			

จากค่าที่จำเป็นต้องกำหนดในแต่ละเว็บไซต์ จะเห็นได้ว่า แต่ละเว็บไซต์มีรูปแบบของยูอาร์แอลที่เป็นของหน้าแสดงผลลัพธ์การค้นหาแตกต่างกัน ทำให้ไม่สามารถกำหนดรูปแบบตายตัวของยูอาร์แอลเพื่อทำการรวบรวมข้อมูลได้ เนื่องจากบางเว็บไซต์มีการป้องกันการรวบรวมข้อมูลจากบอท เช่น เลขไอดีของสถานที่ ที่ใช้เฉพาะภายในเว็บไซต์เท่านั้น การที่จะสร้างรูปแบบของยูอาร์แอล จึงไม่สามารถทำได้เพราะไม่มีข้อมูลที่ใช้เฉพาะภายในเว็บไซต์นั้น ๆ

4.3.4 การทดลองนำกูเกิลเสิร์ชเอนจินมาใช้ภายในระบบ

เพื่อหาวิธีที่จะนำกูเกิลเสิร์ชเอนจินมาใช้ภายในระบบ ในส่วนของการค้นหายูอาร์แอลจากคำสำคัญ ก่อนที่จะนำไปให้สไปเดอร์ทำการเก็บข้อมูล โดยทำการรันโค้ดต่อไปนี้ตามเวอร์ชันของไพธอนและไลบรารีที่จำเป็นต้องใช้ จากนั้นจึงนำไปทดสอบการใช้งานร่วมกับไลบรารี Scrapy

โปรแกรม 4.19 โค้ดไพธอนเวอร์ชัน 2.7

```
from google import search
import urllib
from bs4 import BeautifulSoup

def google_scrape(url):
    thepage = urllib.urlopen(url)
    soup = BeautifulSoup(thepage, "html.parser")
    return soup.title.text

i = 1
query = 'search this'
for url in search(query, stop=10):
    a = google_scrape(url)
    print str(i) + ". " + a
    print url
    print " "
    i += 1
```

โปรแกรม 4.20 ไพธอนเวอร์ชัน 3.6

```
from google import search
for url in search('google 1.9.1 python', tld='com.pk',
lang='es', stop=5):
print(url)
```

จากการทดลองโดยใช้วิธีที่เลือก เริ่มต้นจะใช้โค้ดไพธอนเวอร์ชัน 2.7 ซึ่งถ้าหากทำงานควบคู่กับไลบรารี urllib จะสามารถใช้งานกูเกิลเสิร์ชเอนจินได้ง่ายและรวดเร็ว

ตัวอย่าง 4.3 ผลจากการรันโค้ดด้วยไพธอนเวอร์ชัน 2.7

```
1. search: search this website
https://www.laits.utexas.edu/tex/gr/search.html

2. Top 10 Things You Shouldn't Search On Google - Part 3
- YouTube
https://www.youtube.com/watch?v=vujEJb9a-IM
```

ตัวอย่าง 4.3 ผลจากการรันโค้ดด้วยไพธอนเวอร์ชัน 2.7 (ต่อ)

```
3. Search by Image - YouTube
https://www.youtube.com/watch?v=t99BfDnBZcI

4. WHY DO PEOPLE SEARCH THIS?! (Google Feud) - YouTube
https://www.youtube.com/watch?v=EdWLCdWKQuQ

5. Site Search - StudySpanish.com
https://studyspanish.com/search

6. How to search the contents of the current page for
text or links | Firefox Help
https://support.mozilla.org/en-US/kb/search-contents-
current-page-text-or-links

7. How to search this site | Department of Employment
https://www.employment.gov.au/how-search-site

8. Search this Website | Rare and Manuscript Collections
https://rare.library.cornell.edu/about/search


Process finished with exit code 0
```

แต่การนำมาใช้งานกับไลบรารี Scrapy นั้นไม่สามารถทำได้ เนื่องจาก Scrapy เป็นไลบรารีที่รองรับเทคโนโลยีใหม่จึงไม่สามารถทำงานบนไพธอนเวอร์ชัน 2.7 ได้ ทำให้ต้องเปลี่ยนมาใช้ไพธอนเวอร์ชัน 3.6

ตัวอย่าง 4.4 ผลจากการรันโค้ดด้วยไพธอนเวอร์ชัน 2.7 กับไลบรารี Scrapy

```
Traceback (most recent call last):
  File "C:/Users/tawan/Desktop/oneTest/test.py", line 13,
in <module>
    a = google_scrape(url)
  File "C:/Users/tawan/Desktop/oneTest/test.py", line 6,
in google_scrape
    thepage = urllib.urlopen(url)
AttributeError: module 'urllib' has no attribute
'urlopen'

Process finished with exit code 1
```

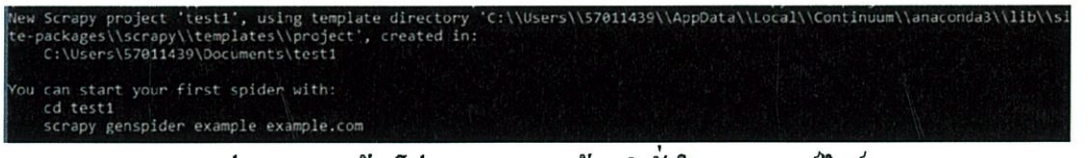
แต่ไพธอนเวอร์ชัน 3.6 ไม่รองรับการใช้งานของไลบรารี urllib ทำให้ต้องศึกษาไลบรารีของกูเกิลเสิร์ชเอนจินเพิ่มเติมแล้วนำโค้ดมาประยุกต์ ถึงสามารถค้นหาผ่านกูเกิลเสิร์ชเอนจินและทำการเก็บยูอาร์แอลเพื่อนำไปใช้งานต่อในส่วนของไลบรารี Scrapy ตามที่ต้องการได้

ตัวอย่าง 4.5 ผลจากการรันโค้ดด้วยไพธอนเวอร์ชัน 3.6

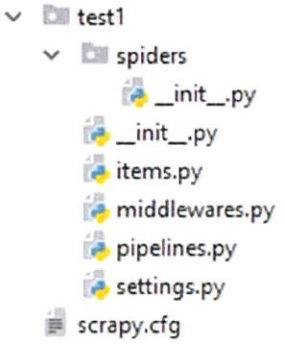
```
https://www.linguee.com/english-
spanish/translation/search+this+site.html
https://www.linguee.com/english-
spanish/translation/you+can+search+this.html
http://context.reverso.net/traduccion/ingles-
espanol/search+this
https://www.laits.utexas.edu/tex/gr/search.html
https://studyspanish.com/search
https://support.mozilla.org/en-US/kb/search-contents-
current-page-text-or-links
https://www.tineye.com/
https://www.education.gov.au/how-search-site
https://addons.opera.com/es-
419/extensions/details/ampare-search-this-on-
youtube/?display=en&reports&resolved

Process finished with exit code 0
```

4.3.5 การทดสอบการเรียกใช้งานสไปเดอร์จากไลบรารี Scrapy

เพื่อเรียกใช้งานสไปเดอร์ให้ไปทำการเก็บข้อมูลตามยูอาร์แอลและคีย์เวิร์ดที่กำหนด โดยทำการสร้างโปรเจคจากไลบรารี Scrapy ด้วยคำสั่ง scrapy startproject test ในคอมมานด์ไลน์จะได้ผลดังรูป 4.25 และมีโครงสร้างของโปรเจคดังรูป 4.26



รูป 4.25 การสร้างโปรเจค Scrapy ด้วยคำสั่งในคอมมานด์ไลน์



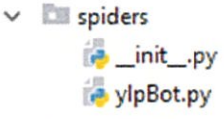
รูป 4.26 โครงสร้างของโปรเจค Scrapy

ไปที่ settings.py เพิ่มโค้ดเพื่อทำการส่งออกข้อมูลเป็นไฟล์ประเภทซีเอสวี

โปรแกรม 4.21 โค้ดสำหรับการ export ข้อมูลเป็นไฟล์ .csv

```
#Export as CSV Feed
FEED_FORMAT = "csv"
FEED_URI = "testyellowpage.csv"
```

สร้างไฟล์ ylpBot.py ในโฟลเดอร์ spiders/ (รูป 4.27) แล้วทำการเขียนโค้ดตามโปรแกรม 4.22



รูป 4.27 บอทสไปเดอร์ในโฟลเดอร์ spiders

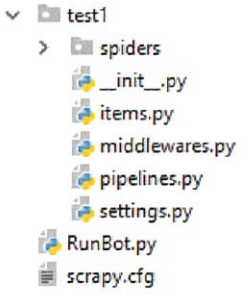
โปรแกรม 4.22 สไปเดอร์สำหรับเว็บไซต์ https://www.yellowpages.com/

```
import scrapy
class ylpbotSpider(scrapy.Spider):
    name = 'ylpBot'

    def __init__(self, *args, **kwargs):
        super( ylpbotSpider, self).__init__(*args,
**kwargs)
        self.start_urls = [kwargs.get('start_url')]

    def parse(self, response):
        titles =
response.xpath('//h2/a/span/text()').extract()
        address1=response.css('.street-
address::text').extract()
```

โปรแกรม 4.22 สไปเดอร์สำหรับเว็บไซต์ https://www.yellowpages.com/ (ต่อ)

```
        address2=
response.css('.locality::text').extract()
        address2=[w.replace('\xa0', '') for w in address2]
        address3=
response.xpath('//span[@itemprop="addressRegion"]/text()'
).extract()
        address4=
response.xpath('//span[@itemprop="postalCode"]/text()').e
xtract()
        phones =
response.css('.phones.phone.primary::text').extract()
        #Give the extracted content row wise
        next_page = response.css('.next.ajax-
page::attr(href)').extract_first()
        if next_page is not None:
           next_page = response.urljoin(next_page)
           yield scrapy.Request(next_page,
callback=self.parse)

        for item in
zip(titles,address1,address2,address3,address4,phones):
            #create a dictionary to store the scraped
info
            scraped_info = {
                'title' : item[0],
                'Street address' : item[1],
                'Location address' : item[2],
                'Region' : item[3],
                'Postal' : item[4],
                'Phones' : item[5],
            }
            #yield or give the scraped info to scrapy
            yield scraped_info
```

สร้างโปรแกรมสำหรับรันสไปเดอร์ (ylpBot) โดยสร้าง RunBot.py ไว้ในโฟลเดอร์ test1\



รูป 4.28 สร้าง RunBot.py ในโฟลเดอร์ test1

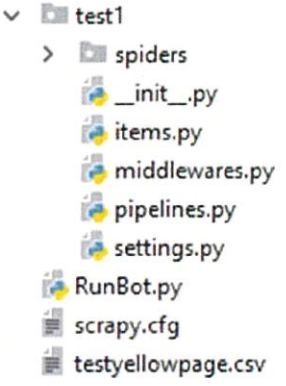
โปรแกรม 4.23 การรับคีย์เวิร์ดและกำหนดยูอาร์แอลเพื่อทำการ call spider

```
from scrapy import cmdline

keyWord = input("Enter keyword : ")
keyWord= keyWord.replace(" ", "%20")
print (keyWord)
print (("scrapy crawl ylpBot -a
start_url=https://www.yellowpages.com/search?search_terms
=hotel&geo_location_terms="+keyWord).split())
cmdline.execute(("scrapy crawl ylpBot -a
start_url=https://www.yellowpages.com/search?search_terms
=hotel&geo_location_terms="+keyWord).split())
```

ซึ่งผลการทดลอง จะได้ไฟล์ .csv ซึ่งเป็นไฟล์ที่สไปเดอร์ไปเก็บข้อมูลมา

```
Enter keyword : Rome
Rome
['scrapy', 'crawl', 'ylpBot', '-a', 'start_url=https://www.yellowpages.com/search?search_terms=hotel&geo_location_terms=Rome']
2017-11-24 07:23:27 [scrapy.utils.log] INFO: Scrapy 1.4.0 started (bot: test1)
2017-11-24 07:23:27 [scrapy.utils.log] INFO: Overridden settings: {'BOT_NAME': 'test1', 'FEED_FORMAT': 'csv', 'FEED_URI': 'testyellowpage.csv', 'NEWSPIDER_MODULE': 'test1.spiders',
2017-11-24 07:23:27 [scrapy.middleware] INFO: Enabled extensions:
['scrapy.extensions.corestats.CoreStats',
 'scrapy.extensions.telnet.TelnetConsole',
 'scrapy.extensions.feedexport.FeedExporter',
 'scrapy.extensions.logstats.LogStats']
2017-11-24 07:23:27 [scrapy.middleware] INFO: Enabled downloader middlewares:
['scrapy.downloadermiddlewares.robotstxt.RobotsTxtMiddleware',
 'scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',
 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',
 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware',
 'scrapy.downloadermiddlewares.retry.RetryMiddleware',
 'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware',
 'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware',
 'scrapy.downloadermiddlewares.redirect.RedirectMiddleware',
 'scrapy.downloadermiddlewares.cookies.CookiesMiddleware',
 'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware',
 'scrapy.downloadermiddlewares.stats.DownloaderStats']
2017-11-24 07:23:27 [scrapy.middleware] INFO: Enabled spider middlewares:
['scrapy.spidermiddlewares.httperror.HttpErrorMiddleware',
 'scrapy.spidermiddlewares.offsite.OffsiteMiddleware',
 'scrapy.spidermiddlewares.referer.RefererMiddleware',
 'scrapy.spidermiddlewares.urllength.UrlLengthMiddleware',
 'scrapy.spidermiddlewares.depth.DepthMiddleware']
2017-11-24 07:23:27 [scrapy.middleware] INFO: Enabled item pipelines:
[]
2017-11-24 07:23:27 [scrapy.core.engine] INFO: Spider opened
```

รูป 4.29 ผลการรวบรวมข้อมูลจากสไปเดอร์



รูป 4.30 ไฟล์ .csv จะพบในโฟลเดอร์ test1

	A	B	C	D	E	F
1	title	Street address	Location address	Region	Postal	Phones
2						
3	Holiday Inn Express & Suites Rome	35 Hobson Way	Rome,	GA	30161	(800) 345-8082
4						
5	Days Inn Rome	840 Turner Mcall Blvd SW	Rome,	GA	30161	(706) 295-0400
6						
7	Country Inn & Suites	15 Hobson Way	Rome,	GA	30161	(706) 232-3380
8						
9	Americas Best Value Inn	2973 Cedartown Hwy SE	Rome,	GA	30161	(706) 235-1717
10						
11	Best Western Executive Inn	217 Highway 411 SE	Rome,	GA	30161	(706) 234-3161
12						
13	La Quinta Inn & Suites Rome	15 Chateau Dr SE	Rome,	GA	30161	(866) 670-5933
14						
15	Hawthorn Suites by Wyndham Rome	100 W 2nd Ave	Rome,	GA	30161	(706) 378-4837
16						
17	Garden Inn & Suites	2541 Shorter Ave SW	Rome,	GA	30165	(706) 235-2330
18						
19	Hampton Inn & Suites Rome	875 W 1st St	Rome,	GA	30161	(706) 622-5631
20						
21	Courtyard by Marriott	320 West 3rd	Rome,	GA	30165	(706) 295-7006
22						
23	Baymont Inn & Suites	21 Chateau Dr SE	Rome,	GA	30161	(706) 232-9551
24						
25	Studio 6	20 Chateau Dr SE	Rome,	GA	30161	(706) 291-2447
26						
27	Comfort Suites Rome	23 Chateau Dr SE	Rome,	GA	30161	(706) 232-6055

testyellowpage

รูป 4.31 ผลของการรวบรวมข้อมูลในไฟล์ .csv

4.3.6 เปรียบเทียบไลบรารี Scrapy และ Selenium

เป็นการนำไลบรารี Scrapy และไลบรารี Selenium มาทำการเปรียบเทียบเพื่อหาไลบรารีที่เหมาะสมกับการใช้เก็บข้อมูลจากแต่ละเว็บไซต์ ในการเปรียบเทียบนั้นจะให้ไลบรารีทั้งสองทำการเก็บข้อมูลที่เหมือนกันจากเว็บไซต์เดียวกัน เพื่อเปรียบเทียบในเรื่องต่าง ๆ ดังต่อไปนี้

- ความเร็ว
- ปริมาณข้อมูล
- ความถูกต้อง
- ลักษณะการเก็บ (เทคนิค)
- ความแตกต่างในการทำงาน

ในการเปรียบเทียบนั้น จะทดลองทำการเก็บข้อมูลจากหลายเว็บไซต์ ซึ่งจะเริ่มต้นที่เว็บไซต์ Booking.com ก่อน ซึ่งจะใช้ในการเปรียบเทียบด้านความถูกต้องของข้อมูลโดยผลลัพธ์จากการทดลองให้ทั้งสองไลบรารีทำการเก็บข้อมูล ไลบรารี Scrapy นั้นไม่สามารถทำการเก็บข้อมูลบางข้อมูลได้ (รูป 4.32) เนื่องจากว่า ไลบรารี Scrapy จะทำการ Fetch ข้อมูลจากหน้าเว็บไซต์มาก่อนที่ฐานข้อมูลของเว็บไซต์นั้น ๆ จะส่งข้อมูลที่ต้องแสดงมาถึง ทำให้ข้อมูลจากหน้าเว็บไซต์ที่ Fetch มาด้วยไลบรารี Scrapy นั้นอาจมีข้อมูลที่ต้องการไม่ครบ

รูป 4.32 ตัวอย่างส่วนของข้อมูลที่ต้องการเก็บจากหน้าเว็บไซต์ Booking.com จากการ Fetch โดย Scrapy

แต่สำหรับไลบรารี Selenium นั้น จะทำการเปิดเว็บไซต์ขึ้นมาด้วยเว็บเบราว์เซอร์เสมือนว่าผู้ใช้งานเปิดเว็บไซต์ขึ้นมาด้วยตนเอง ข้อมูลที่ปรากฏบนหน้าเว็บไซต์ขณะแสดงผลในเว็บเบราว์เซอร์กับข้อมูลที่มีในหน้าเว็บไซต์ขณะที่บอททำการเก็บข้อมูลจึงเป็นข้อมูลที่เหมือนกัน (รูป 4.33) ข้อมูลที่เก็บได้จึงมีความถูกต้องและครบถ้วนตามความต้องการ

รูป 4.33 ตัวอย่างส่วนของข้อมูลที่ต้องการเก็บจากหน้าเว็บไซต์ Booking.com ที่แสดงผลจริงขณะใช้งาน

เว็บไซต์ต่อไปที่จะใช้สำหรับทำการเปรียบเทียบไลบรารี คือเว็บไซต์ Hotels.com ซึ่งจะใช้ในการเปรียบเทียบด้านความเร็วและปริมาณข้อมูล โดยผลลัพธ์จากการทดลองเก็บข้อมูลโดยใช้ไลบรารี Scrapy พบว่าสามารถเก็บข้อมูลได้อย่างรวดเร็วประมาณ 0.143 วินาทีต่อ 1 โรงแรม (ตัวอย่าง 4.6)

ตัวอย่าง 4.6 ข้อมูล log การเก็บข้อมูลของไลบรารี Scrapy

```
2018-05-14 00:09:51 [scrapy.utils.log] INFO: Scrapy 1.5.0
started

...

2018-05-14 00:09:57 [scrapy.extensions.feedexport] INFO:
Stored json feed (42 items)

...

2018-05-14 00:09:57 [scrapy.core.engine] INFO: Spider
closed (finished)
```

และในส่วนของผลลัพธ์จากการทดลองเก็บข้อมูลโดยใช้ไลบรารี Selenium พบว่าสามารถเก็บข้อมูลได้เร็วประมาณ 1.524 วินาทีต่อ 1 โรงแรม

จะเห็นได้ว่า การเก็บข้อมูลโดยใช้ไลบรารี Scrapy สามารถเก็บข้อมูลได้รวดเร็วกว่าไลบรารี Selenium มาก ทั้งนี้ ก็ต้องขึ้นอยู่กับความเร็วของอินเตอร์เน็ตด้วย แต่ในขณะเดียวกันข้อมูลที่ถูกเก็บโดยใช้ไลบรารี Selenium กลับสามารถเก็บข้อมูลจากหน้าเว็บกลับมาได้ครบถ้วนตามที่ต้องการมากกว่าการใช้ไลบรารี Scrapy เนื่องจากไลบรารี Selenium จะมีการทำงานด้วยการเปิดหน้าเว็บไซต์ขึ้นมาด้วยเว็บเบราว์เซอร์ แล้วจึงจะเริ่มทำการเก็บข้อมูลเมื่อองค์ประกอบภายในหน้าเว็บไซต์ถูกดาวน์โหลดมาแสดงจนครบ ทำให้ข้อมูลที่เก็บมาได้เป็นข้อมูลเดียวกันที่แสดงภายในหน้าเว็บไซต์

แต่สำหรับไลบรารี Scrapy นั้นกลับไม่สามารถเก็บข้อมูลมาได้อย่างครบถ้วนในบางเว็บไซต์ เพราะในการทำงานของไลบรารีนั้นจะไม่สนใจว่าข้อมูลหรือองค์ประกอบในหน้าเว็บไซต์นั้นได้ถูก Fetch มาจนครบหรือไม่ จะสนใจเพียงว่าสามารถ Fetch หน้าเว็บไซต์มาได้หรือไม่ได้เท่านั้น จึงทำให้ขณะที่ทำการเก็บข้อมูลไม่สามารถทราบได้เลยว่าผลลัพธ์ของข้อมูลที่ได้จะครบถ้วนหรือไม่ ซึ่งจะเห็นได้จากการทดลองเก็บข้อมูลจากเว็บไซต์ ซึ่งการเก็บข้อมูลจากเว็บไซต์ Booking.com นั้นไม่สามารถเก็บข้อมูลบางส่วนได้ ทำให้ข้อมูลไม่สมบูรณ์และไม่สามารถนำไปใช้งานได้ แต่การเก็บข้อมูลจากเว็บไซต์ Hotels.com นั้นสามารถทำได้และได้ข้อมูลครบถ้วน

จากการทดลองเปรียบเทียบไลบรารี Scrapy และ Selenium จึงสามารถสรุปได้ว่า (ตาราง 4.2) หากต้องการความเร็วในการเก็บข้อมูลที่มีปริมาณมาก ควรเลือกใช้ไลบรารี Scrapy ในการเก็บข้อมูล แต่หากต้องการความถูกต้อง ควรเลือกใช้ไลบรารี Selenium ซึ่งให้ผลลัพธ์ของการเก็บข้อมูลที่ครบถ้วนและตรงตามความต้องการมากกว่า

ตาราง 4.2 เปรียบเทียบไลบรารี Scrapy และ Selenium

	Scrapy	Selenium
ความเร็ว	ไวกว่า	ช้ากว่า
ปริมาณข้อมูล	มากกว่า	น้อยกว่า
ความถูกต้อง	น้อยกว่า	มากกว่า
ลักษณะการเก็บ (เทคนิค)	Xpath, CSS	Xpath, CSS
ความแตกต่างในการทำงาน	สนใจแต่การ fetch หน้าเว็บไซต์ ซึ่งจะได้มาแต่โครงสร้างเว็บไซต์ในภาษาเอชทีเอ็มแอล ทำให้ไม่ได้ข้อมูลจากฐานข้อมูลโดยตรง	พยายามเปิดหน้าเว็บไซต์ขึ้นมาด้วยเว็บเบราว์เซอร์ จึงสามารถเก็บข้อมูลจากฐานข้อมูลที่ไป crawl มาโดยตรง

4.3.7 ส่วนการเก็บไอดีของสถานที่

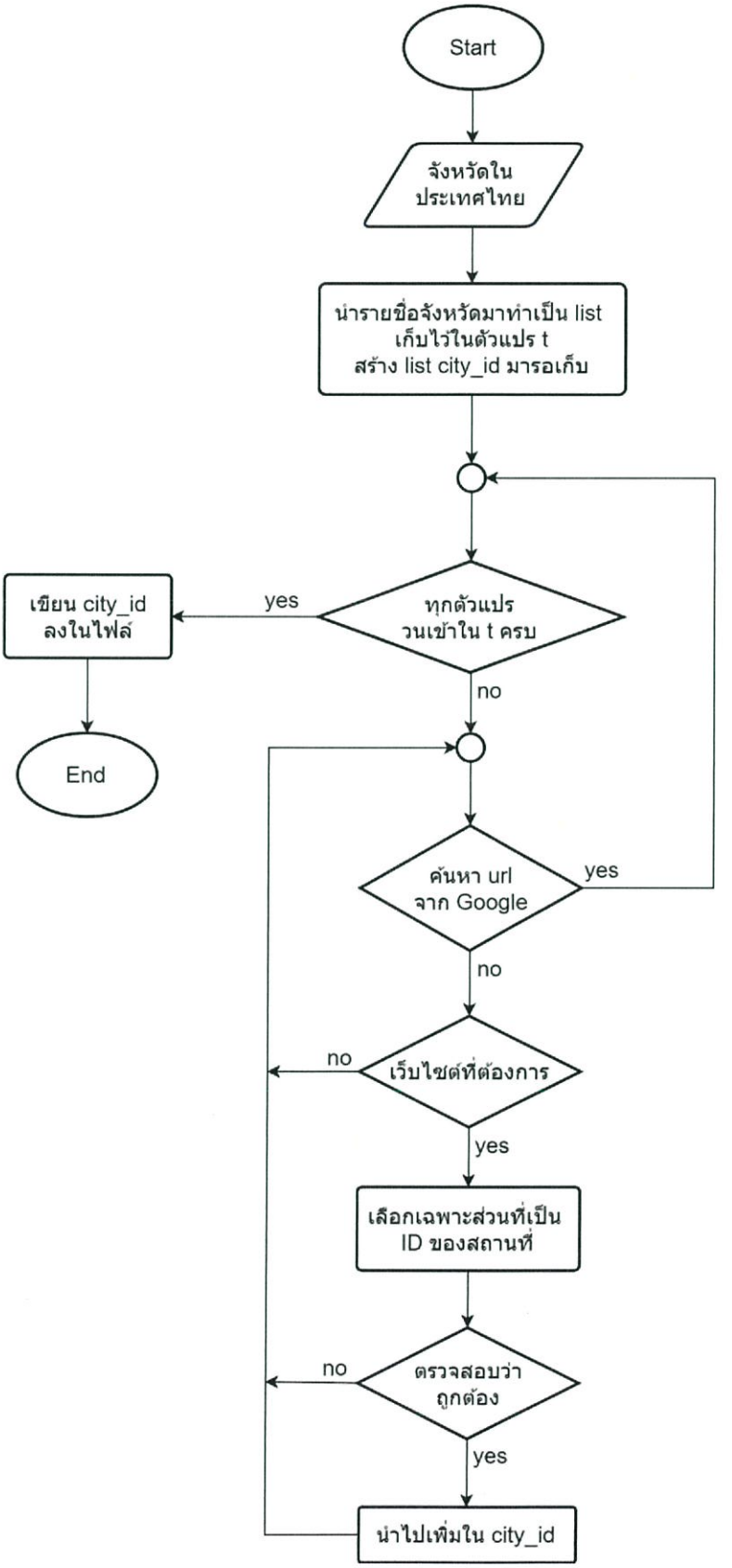
สร้างไฟล์คำสั่งสำหรับทำการเก็บไอดีของสถานที่ที่ต้องการจะค้นหาโรงแรม โดยใช้เอพีไอของกูเกิลช่วยในการเก็บข้อมูลไอดีของสถานที่แล้วเขียนลงในไฟล์ .txt เพื่อสามารถนำไปใช้งานในส่วนต่อไป

การเก็บไอดีของสถานที่ที่ต้องการเก็บข้อมูล จะใช้กูเกิลเสิร์ชเอพีไอ ซึ่งการนำกูเกิลเสิร์ชเอพีไอมาใช้นั้นจะช่วยทำให้การทำงานของบอทช้าลง แต่เนื่องจากว่าบอทไม่สามารถที่จะหาไอดีของสถานที่มาเองได้ แต่ในขณะเดียวกัน ไอดีของสถานที่นั้นไม่ได้เปลี่ยนแปลงค่าตลอดเวลา จึงสามารถใช้สคริปต์คำสั่งในการจัดเก็บไอดีของสถานที่มาก่อนโดยที่ไม่ต้องทำการค้นหาในกูเกิลเสิร์ชทุกครั้ง

สำหรับขั้นตอนการทำงานของไฟล์ cityid.py (รูป 4.34) ซึ่งเป็นไฟล์คำสั่งที่ใช้ในการเก็บไอดีของสถานที่ เริ่มต้นนั้นจะทำการสร้างรายการของสถานที่เก็บไว้ในในตัวแปร จากนั้นจะนำชื่อของสถานที่ที่อยู่ในรายการไปทำการค้นหายูอาร์แอลโดยใช้กูเกิลเสิร์ชเอนจินแล้วทำการเลือกเฉพาะยูอาร์แอลจากเว็บไซต์ที่ต้องการ เพื่อเข้าไปทำการเก็บไอดีของสถานที่จากเว็บไซต์นั้น ๆ กลับมาเพิ่มลงในไฟล์ city_id.txt (ตัวอย่าง 4.7)

ตัวอย่าง 4.7 ไอดีของสถานที่ภายในไฟล์ city_id.txt

```
1313946, bangkok
1714341, nakhon-pathom
1314845, nonthaburi
1317578, pathum-thani
...
```



รูป 4.34 การทำงานของไฟล์ cityid.py

4.3.8 การสร้างไฟล์สคริปต์เพื่อเปิดการทำงานของบอท

เป็นคำสั่งสคริปต์ที่จะอ่านไอดีและชื่อของสถานที่จากไฟล์ city_id.txt เพื่อนำมาสร้างและใส่วันที่ลงในยูอาร์แอล เพื่อกำหนดยูอาร์แอลให้บอทนำไป run ด้วยคำสั่ง scrapy crawl ในคอมมานด์ไลน์ผ่านไลบรารี subprocess ฟังก์ชัน popen โดยที่คำสั่ง scrapy crawl สามารถใช้ option -o เพื่อสร้างไฟล์ตามที่ไลบรารี Scrapy กำหนดได้ และในส่วนของ option -a start_url=urls เป็นการกำหนด start_urls ที่คอมมานด์ไลน์ได้

ในการทำงาน จำเป็นต้องนำไลบรารีเหล่านี้มาใช้งาน ได้แก่

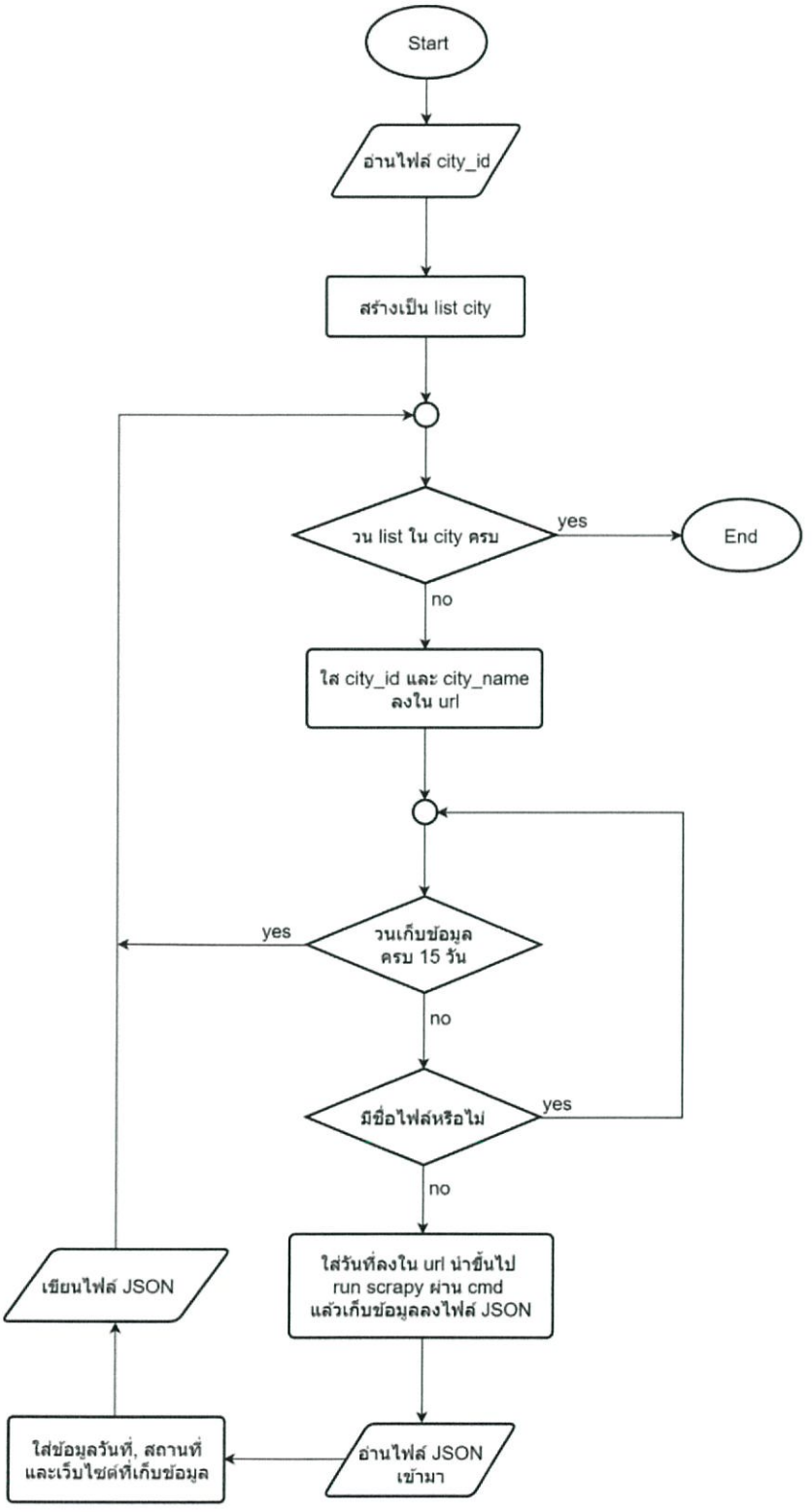
- Subprocess เพื่อใช้คำสั่งในคอมมานด์ไลน์ได้
- Datetime เพื่อใช้งานเกี่ยวกับวันที่
- JSON เพื่ออ่านและเขียนไฟล์เจสัน
- Path เพื่อสร้างพาท

```
import subprocess
from datetime import date,timedelta
import json
from pathlib import Path
```

รูป 4.35 การ import ไลบรารีมาใช้ในไฟล์ os.sys.py

```
import subprocess
from datetime import date,timedelta
import json
from pathlib import Path
print(subprocess.Popen("dir", shell=True, stdout=subprocess.PIPE).stdout.read())

# [illegible] cityid
city  = []
with open("C:\\Users\\tawan\\Desktop\\oneTest\\cityid.txt") as line:
    tempCity = line.readlines()

for i in tempCity:
    city.append(i.replace('\n','').split(','))

for cityid,cityname in city :
    temp1 = 'https://www.hotels.com/search/listings.json?destination-id=#CITYCODE&sort-order=BEST_SELLER&q-check-out=' \
            '#DATEOUT&q-destination=#CITYNAME&q-room-0-adults=2&pg=3&q-rooms=1&q-check-in=#DATEIN&resolved-location=' \
            'CITY:#CITYCODE:UNKNOWN:UNKNOWN&q-room-0-children=0&pn='
    temp2 = temp1.replace('#CITYCODE',cityid).replace('#CITYNAME',cityname.replace('-','%20'))
    for i in range (15):
        my_file = Path("C:\\Users\\tawan\\Desktop\\oneTest\\DATA" + str(date.today()) + "\\" +
                       str(cityname.replace('-', '_'))+ str(date.today() + timedelta(days=i)) + ".json")
        if not my_file.is_file():
            temp3 = temp2.replace('#DATEOUT',str(date.today()+timedelta(days=i+1)))\
                .replace('#DATEIN',str(date.today()+timedelta(days=i)))
            urls = temp3.replace("&",'^&')
            print(subprocess.Popen("scrapy crawl hotelbot -o "+"DATA"+str(date.today())+
                                   "\\"+str(cityname.replace('-','_'))+str(date.today()+timedelta(days=i))+".json"
                                   +" -a start_url="+urls, shell=True,stdout=subprocess.PIPE).stdout.read())

            with open("C:\\Users\\tawan\\Desktop\\oneTest\\DATA" + str(date.today()) + "\\"
                      + str(cityname.replace('-', '_'))+str(date.today()
                      + timedelta(days=i)) + ".json") as json_file:
                forJsonData = json.load(json_file)

            tempDateForJson1 = str(date.today())
            tempDateForJson2 = str(date.today() + timedelta(days=i))

            tempForJson = [
                {
                    "dateCrawling": tempDateForJson1,
                    "dateCheckPrice": tempDateForJson2,
                    "website": 'hotels.com',
                    "location" : str(cityname),
                    "listHotel": forJsonData
                }
            ]
            print(tempForJson)
            with open("C:\\Users\\tawan\\Desktop\\oneTest\\DATA" + str(date.today()) + "\\"
                      + str(cityname.replace('-', '_'))
                      + str(date.today() + timedelta(days=i)) + ".json", 'w') as outfile:
                json.dump(tempForJson, outfile)

```

รูป 4.36 โปรแกรม os.sys.py

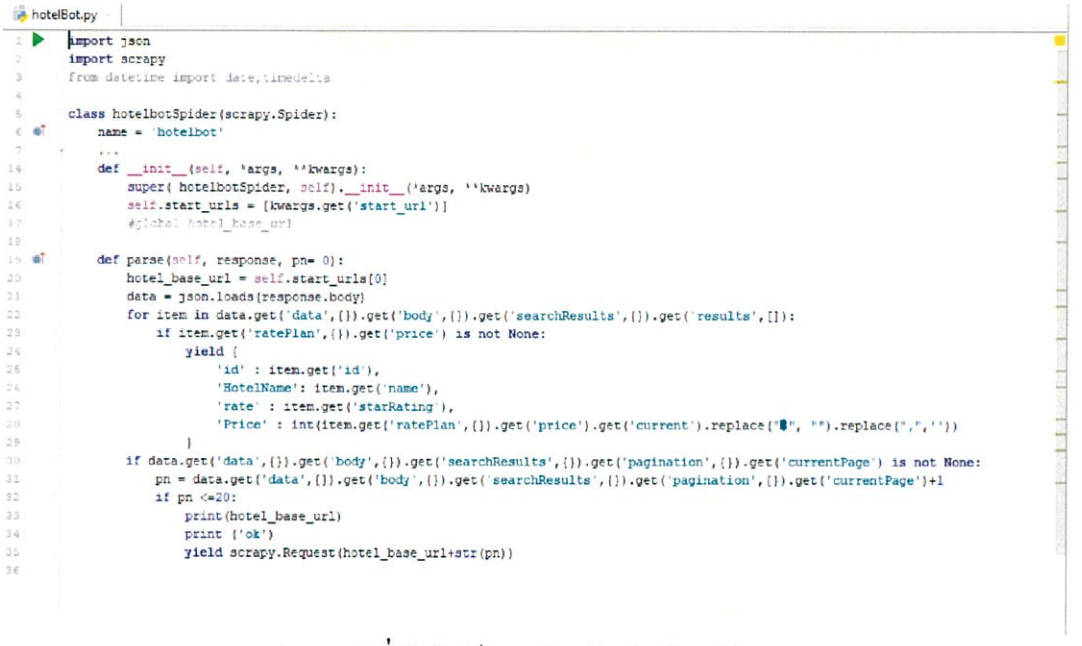


รูป 4.37 การทำงานของคำสั่งสคริปต์เพื่อเรียกให้บอททำงาน

4.3.9 การทำบอทเก็บข้อมูลโรงแรม

เพื่อทำบอทที่ตรงตามความต้องการใช้งาน คือการเก็บข้อมูลของโรงแรม ได้แก่ ไอดีของโรงแรม ชื่อโรงแรม ระดับของโรงแรม และราคา จะต้อง import ไลบรารีมาใช้งาน โดยมี

- ไลบรารี Scrapy สำหรับทำบอท
- ไลบรารี JSON สำหรับสร้างไฟล์ข้อมูลที่เก็บมาได้เป็นไฟล์เจสัน
- ไลบรารี datetime สำหรับจัดการเกี่ยวกับวันที่



รูป 4.38 คำสั่งในไฟล์บอทสำหรับเก็บข้อมูลโรงแรม

ในการสร้างบอท จำเป็นต้องมีการสร้างคลาสและภายในคลาสจะมี 2 ฟังก์ชัน ได้แก่

1) __init__() เป็นเป็นส่วนที่ช่วยให้สามารถกำหนด start_urls ในการเรียกบอทผ่านคอมมานด์ไลน์ได้ (โปรแกรม 4.18) จากคำสั่ง (ตัวอย่าง 4.5)

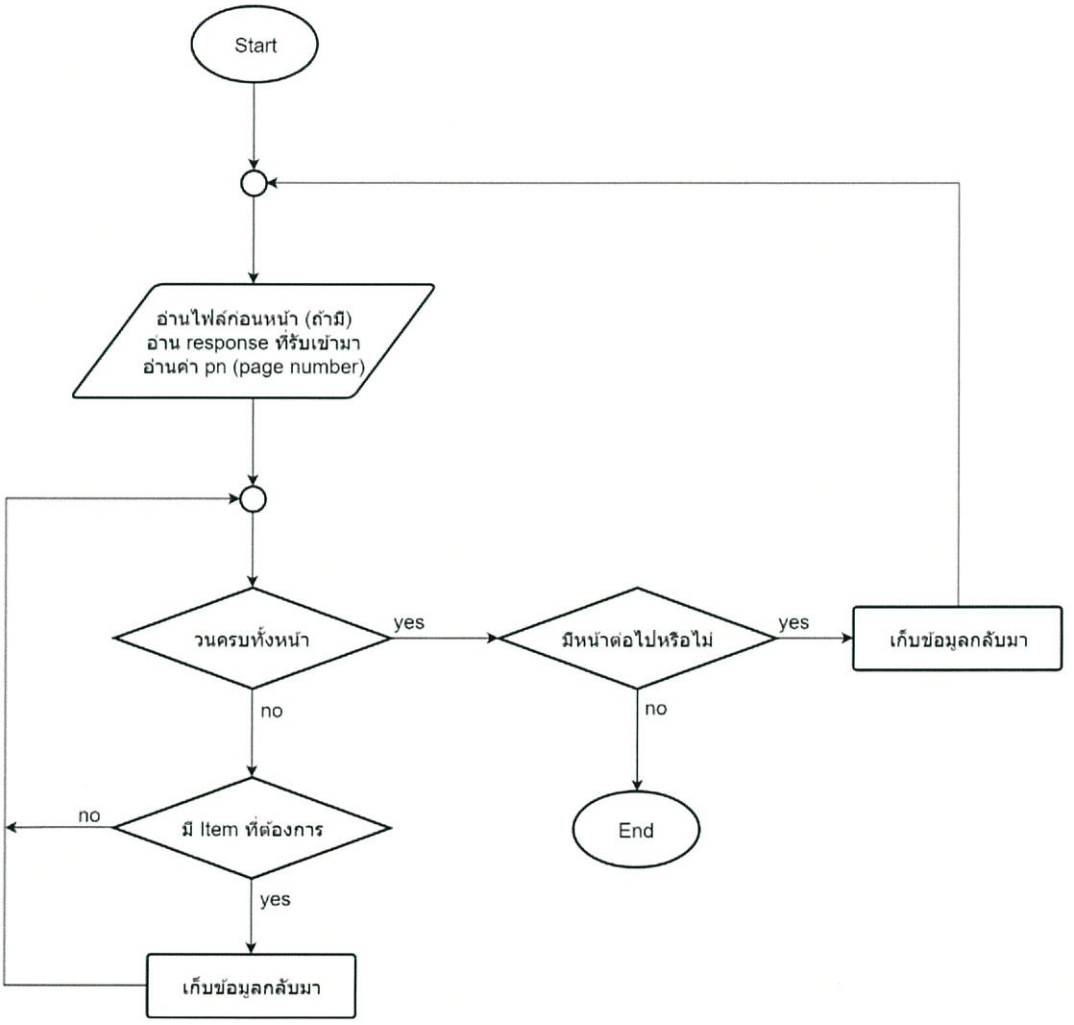
โปรแกรม 4.24 คำสั่งเพื่อการกำหนด start_urls ให้บอทผ่าน command-line ได้

```
def __init__(self, *args, **kwargs):
    super( hotelbotSpider, self).__init__(*args, **kwargs)
    self.start_urls = [kwargs.get('start_url')]
```

ตัวอย่าง 4.8 การกำหนด start_urls ที่ command-line

```
scrapy crawl hotelbot -a start_url=urls
```

2) parse() (รูป 4.24) เป็นส่วนที่กำหนดว่าบอทต้องเก็บข้อมูลจากส่วนใด (รูป 4.25) และหน้าถัดไป (รูป 4.26) อย่างไรบ้าง



รูป 4.39 การเก็บข้อมูลของ Bot

```
def parse(self, response, pn= 0):
    hotel_base_url = self.start_urls[0]
    data = json.loads(response.body)
    for item in data.get('data',{}).get('body',{}).get('searchResults',{}).get('results',[]):
        if item.get('ratePlan',{}).get('price') is not None:
            yield {
                'id' : item.get('id'),
                'HotelName': item.get('name'),
                'rate' : item.get('starRating'),
                'Price' : int(item.get('ratePlan',{}).get('price').get('current').replace("฿", "").replace(",",''))
            }
```

รูป 4.40 คำสั่งเพื่อทำการคัดเลือกข้อมูล

```
if data.get('data',{}).get('body',{}).get('searchResults',{}).get('pagination',{}).get('currentPage') is not None:
    pn = data.get('data',{}).get('body',{}).get('searchResults',{}).get('pagination',{}).get('currentPage')+1
    if pn <=20:
        print(hotel_base_url)
        print ('ok')
        yield scrapy.Request(hotel_base_url+str(pn))
```

รูป 4.41 คำสั่งเพื่อให้บอทไปทุกหน้าที่เป็นผลลัพธ์

การสร้างบอทด้วยวิธีนี้นั้น จะเป็นการไปที่ยูอาร์แอลของข้อมูลเพื่อเก็บข้อมูลที่เป็นชนิดเจสัน จึงจำเป็นต้องมีคำสั่งเพื่อทำการคัดเลือกแต่ข้อมูลที่ต้องการเก็บเท่านั้น (รูป 4.40) และเนื่องจากว่าข้อมูลนั้นมีหลายหน้า จึงต้องมีส่วนที่เป็นคำสั่งให้บอทไปยังหน้าอื่นทุก ๆ หน้าที่เป็นผลลัพธ์ (รูป 4.41) เพื่อเก็บข้อมูลด้วย

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุปของโครงงาน

เว็บแอปพลิเคชันที่ทำงานร่วมกับบอทจากภาษาไพธอน สามารถเก็บข้อมูลจำนวนมากได้ในเวลาที่ลดลงอย่างมากเมื่อเทียบกับการเก็บข้อมูลแบบคัดลอกทั่วไป ข้อมูลที่ได้อยู่ในรูปแบบที่ถูกต้องตามที่ถูกกำหนดโดยผู้ใช้งาน ไม่มีข้อมูลที่ซ้ำกัน อีกทั้งข้อมูลในไฟล์จะถูกจัดเก็บเป็นข้อความตัวอักษร (Text) และตัวเลข (Number) ในรูปแบบของไฟล์เจสันที่สามารถจัดเก็บลงฐานข้อมูลแบบเอกสาร นอกจากนี้ยังสามารถส่งออกเป็นไฟล์ประเภทซีเอสวีจากหน้าภายในหน้าของเว็บแอปพลิเคชันเพื่อนำไปใช้ต่อในงานอื่น ๆ ได้

แต่เนื่องจากว่าในแต่ละเว็บไซต์นั้นมีส่วนประกอบของโครงสร้างที่แตกต่างกันและบางเว็บไซต์ยังมีการเปลี่ยนแปลงหน้าเว็บเสมอ การทำบอทเพียงครั้งเดียวจึงไม่สามารถเก็บข้อมูลจากทุกเว็บไซต์ได้ จึงจำเป็นต้องกำหนดลักษณะการทำงานของบอทให้แต่ละเว็บไซต์ และจากการสำรวจ พบว่าข้อมูลประเภทราคานั้นมีการเปลี่ยนแปลงตลอดเวลา การเก็บข้อมูลเพียงครั้งเดียวจึงไม่เพียงพอ แต่ว่าบอทไม่สามารถทำงานตลอดเวลาได้ จึงต้องตั้งค่าให้บอททำงานตามช่วงเวลาต่าง ๆ เพื่อให้เก็บข้อมูลได้ตามครั้งที่เปลี่ยนแปลงใกล้เคียงกับข้อมูลจริงมากที่สุดด้วย

นอกจากนี้ โครงงานชิ้นนี้เป็นการทำบอทเพื่อเก็บข้อมูลประเภทราคา เป็นไปได้ว่าจะสามารถเก็บข้อมูลของราคาสินค้าได้เช่นเดียวกัน แต่อาจต้องตั้งค่าบอทแตกต่างออกไป เช่น ข้อมูลที่จะเก็บ หรือ ช่วงเวลาที่บอททำงาน ซึ่งอาจจะมีจำนวนครั้งไม่ถี่เท่ากับการเก็บราคาโรงแรม เป็นต้น แต่ถึงแม้ว่าการใช้งานอาจจะดูยุ่งยาก แต่บอทนั้นสามารถทำงานได้รวดเร็ว จึงช่วยเพิ่มประสิทธิภาพและลดเวลาในการเก็บข้อมูลจำนวนมากได้

5.2 ปัญหา อุปสรรค และแนวทางแก้ไข

1) เนื่องจากเป็นการเก็บข้อมูลจากหน้าของเว็บไซต์ การกำหนดคำสั่งให้บอทเก็บข้อมูลตลอดเวลาหรือมีการเก็บข้อมูลเมื่อผู้ใช้งานกดค้นหา วิธีนี้จำเป็นต้องสร้างบอทที่สามารถเก็บข้อมูลได้รวดเร็วมากเพื่อให้ผู้ใช้งานสามารถดูข้อมูลได้ทันทีโดยไม่ต้องรอนานเกินไปจนกว่าจะแสดงผลการค้นหา แต่การที่บอททำงานเร็วมากจนเกินไปอาจทำให้เซิร์ฟเวอร์ของเว็บไซต์ที่ไปทำการเก็บข้อมูลนั้นสรุปว่าบอททำการโจมตีเพื่อปั่นป่วนภายในเว็บไซต์หรือก่อกวนการทำงานของเซิร์ฟเวอร์ที่เป็นเป้าหมาย โดยที่เซิร์ฟเวอร์นั้นจะทำการจำกัดสิทธิ์การเข้าถึงของบอท ทำให้เกิดเป็น error 503 (ตัวอย่าง 5.1) ซึ่งเป็น error จากการที่

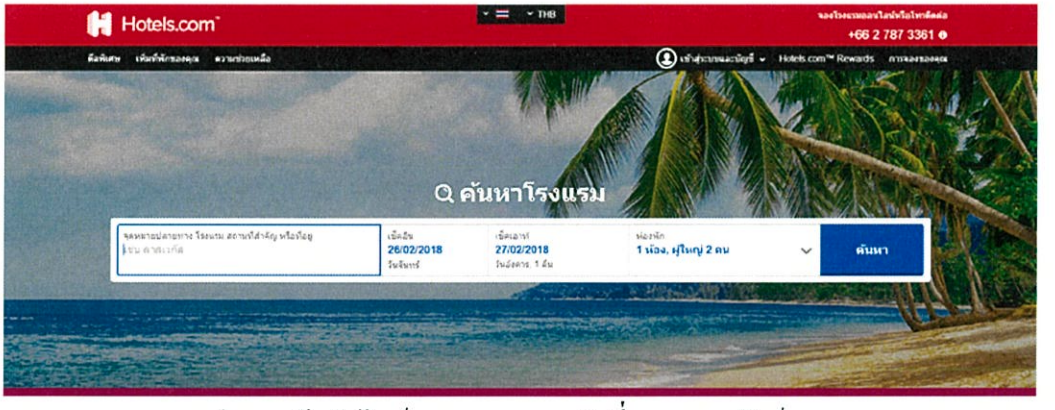
บอทไม่สามารถเข้าถึงเว็บไซต์เพื่อเก็บข้อมูลที่ต้องการได้ บอทจึงไม่สามารถทำงานเร็วเกินไปเพื่อป้องกันการโดนจำกัดสิทธิ์จากเซิร์ฟเวอร์ของเว็บไซต์ปลายทางนั่นเอง

ตัวอย่าง 5.1 ปัญหาการโดนเซิร์ฟเวอร์จำกัดสิทธิ์

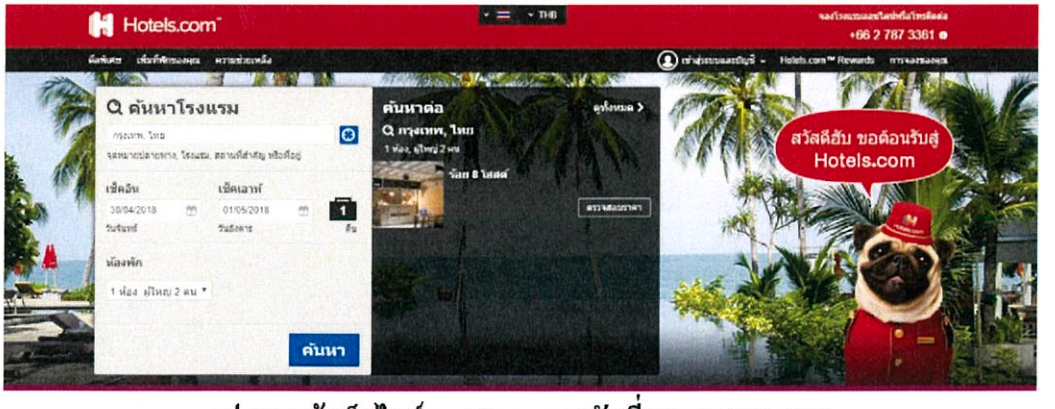
```
urllib.error.HTTPError: HTTP Error 503: Service Unavailable
```

แนวทางในการแก้ปัญหานี้จึงใช้วิธีการกำหนดให้บอทไปเก็บข้อมูลที่คาดว่าผู้ใช้งานจะต้องการมาเก็บลงฐานข้อมูลไว้ล่วงหน้าก่อน ทำให้ขณะที่บอทกำลังทำการเก็บข้อมูลจะสามารถหน่วงความเร็วในการทำงานของบอทให้ช้าลงได้โดยที่ไม่มีผลกระทบต่อการแสดงผลให้แก่ผู้ใช้งาน

2) เนื่องจากเว็บไซต์ที่จำเป็นต้องมีการเข้าใช้งานผ่านอินเตอร์เน็ตจะมีการปรับปรุงหรือเปลี่ยนแปลงรูปแบบของหน้าเว็บไซต์ โครงสร้างของหน้าเว็บไซต์จึงเกิดการเปลี่ยนไปในบางครั้ง ทำให้การเก็บข้อมูลจากหน้าเว็บไซต์โดยตรงผ่าน Xpath หรือ CSS จึงมีโอกาสเกิดปัญหาที่บอทไม่สามารถเก็บข้อมูลกลับมาได้สูง



รูป 5.1 หน้าเว็บไซต์ hotels.com ณ วันที่ 26 กุมภาพันธ์ 2561



รูป 5.2 หน้าเว็บไซต์ hotels.com ณ วันที่ 30 เมษายน 2561

แนวทางในการแก้ปัญหานี้นั้นสามารถทำได้โดยการดักเก็บข้อมูลที่ถูกส่งเข้ามาภายในเว็บไซต์ได้โดยตรงจากยูอาร์แอล แต่วิธีนี้สามารถแก้ปัญหาได้เพียงบางเว็บไซต์เฉพาะเว็บไซต์ที่มีการรับส่งข้อมูลไม่ซับซ้อนเท่านั้น บอทจึงจะสามารถเข้าถึงที่อยู่และทำการเก็บข้อมูลได้

3) ในบางเว็บไซต์ การใช้บอทที่สร้างจากไลบรารี Scrapy นั้นไม่สามารถเก็บข้อมูลได้ด้วยการ Fetch ข้อมูลโครงสร้างจากหน้าเว็บไซต์มาโดยตรงได้ ซึ่งการทำวิธีดังกล่าวจะได้มาเพียงส่วนโครงสร้างของภาษาเอชทีเอ็มแอลเท่านั้นและอาจไม่มีข้อมูลในส่วนที่ต้องการอยู่ภายในนั้นเลย

การแก้ปัญหานี้จึงทำได้โดยการเพิ่มไลบรารี Selenium เข้ามา โดยที่ไลบรารี Selenium จะทำการจำลองเสมือนว่าบอทได้เข้าไปที่เว็บไซต์ปลายทางแล้วจริงโดยการเปิดหน้าเว็บไซต์นั้น ๆ ผ่านเว็บเบราเซอร์แล้วจึงทำการเก็บข้อมูล ด้วยขั้นตอนที่จำเป็นต้องเปิดเว็บไซต์ผ่านเว็บเบราเซอร์จึงทำให้การเก็บโดยใช้บอทจากไลบรารี Selenium ช้ากว่าบอทจากไลบรารี Scrapy เป็นอย่างมาก

5.3 แนวทางในการพัฒนาต่อ

5.3.1 ส่วนการติดต่อกับผู้ใช้งาน

สร้างส่วนติดต่อกับผู้ใช้งานเป็น UI ที่สามารถตั้งค่าลักษณะการเก็บข้อมูลและกำหนดโครงสร้างของข้อมูลที่ต้องการจากเว็บได้โดยผู้ใช้งาน และสามารถทำการส่งออกข้อมูลเหล่านั้นออกมาเป็นไฟล์ได้หลายชนิดมากยิ่งขึ้น

5.3.2 ส่วนการเก็บข้อมูล หรือ บอท

ทำการปรับปรุงประสิทธิภาพของบอทให้สามารถเก็บข้อมูลได้ยืดหยุ่นมากยิ่งขึ้นและบอทเองยังมีปัญหาเรื่องการใช้งานในภาษาไทยซึ่งสามารถพัฒนาต่อไปได้ให้ไฟล์ของข้อมูลสามารถ encode และ decode เป็นภาษาไทยได้

บรรณานุกรม

คณิตศาสตร์และคอมพิวเตอร์ในชีวิตประจำวัน. **การสร้างโฮมเพจด้วยภาษา HTML.** [Online]. Available: https://goo.gl/nohC4k

นัฐลิกา เพ็งรักษ์. **ความหมายของ Search Engine.** [Online]. Available: https://goo.gl/E6xakU

โปรแกรมค้นหา. [Online]. Available: https://goo.gl/xXsC3R

มานพ กองอุ่น. 2559. **Bootstrap คืออะไร?.** [Online]. Available: https://goo.gl/xYrmKq

วศิน เที่ยงคุณากฤต. 2560. **เริ่มพัฒนา Web Application กับภาษา Python ด้วย Django Framework.** [Online]. Available: https://goo.gl/RjNgjk

AmplySoft. 2556. **เริ่มต้นการเขียนเว็บไซต์ ทำเว็บไซต์ด้วยภาษา Python กับ Django Framework.** [Online]. Available: https://goo.gl/Y37PYo

Baker, Jason. **3 Python web scrapers and crawlers.** [Online]. Available: https://goo.gl/H68XbZ

Chai Phonbopit. 2558. **MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น.** [Online]. Available: https://goo.gl/Q6C2GD

CSS คืออะไร?. [Online]. Available: https://goo.gl/MjqU4b

CSS คืออะไร ซีเอสเอส คือ ภาษาที่ใช้ในการจัดรูปแบบเอกสาร HTML ให้มีความสวยงาม. [Online]. Available: https://goo.gl/drw8Ds

Django Software Foundation. **Django documentation.** [Online]. Available: https://goo.gl/TNDBeb

HTML. [Online]. Available: https://goo.gl/CJCxkZ

Miller, David. **SB Admin.** [Online]. Available: https://goo.gl/rFa328

Module google. [Online]. Available: https://goo.gl/c1xNHX

Mongoengine. **MongoEngine User Documentation.** [Online].

Available: https://goo.gl/peyrXx

Pyspider. **Level 1: HTML and CSS Selector.** [Online].

Available: https://goo.gl/6jrhxm

Python. [Online].

Available: https://goo.gl/BjDMT5

Python Software Foundation. **urllib — URL handling modules.** [Online].

Available: https://goo.gl/km37wc

Richardson, Leonard. **Beautiful Soup Documentation.** [Online].

Available: https://goo.gl/nG2iF7

Scrapy. **Architecture overview.** [Online].

Available: https://goo.gl/MFV3Vj

Scrapy. **Scrapy at a glance.** [Online].

Available: https://goo.gl/9Y13oa

Selenium Project. **Selenium Documentation.** [Online].

Available: https://goo.gl/NEq7RZ

Softmelt. **การใช้งาน Bootstrap Framework : ประโยชน์ และขั้นตอนการติดตั้ง Bootstrap.** [Online]. Available: https://goo.gl/4fx51C

THE CURIOUS DEVELOPER. 2561. **Swagger API Documentation.** [Online].

Available: https://goo.gl/xxNGd6

Wikipedia. **Metasearch engine.** [Online].

Available: https://goo.gl/utavqg

Wikipedia. **Web search engine.** [Online].

Available: https://goo.gl/FXtUFg

W3TRAINING SCHOOL. **Learn How Search Engines Like Google Works.** [Online].

Available: https://goo.gl/ftp6UP