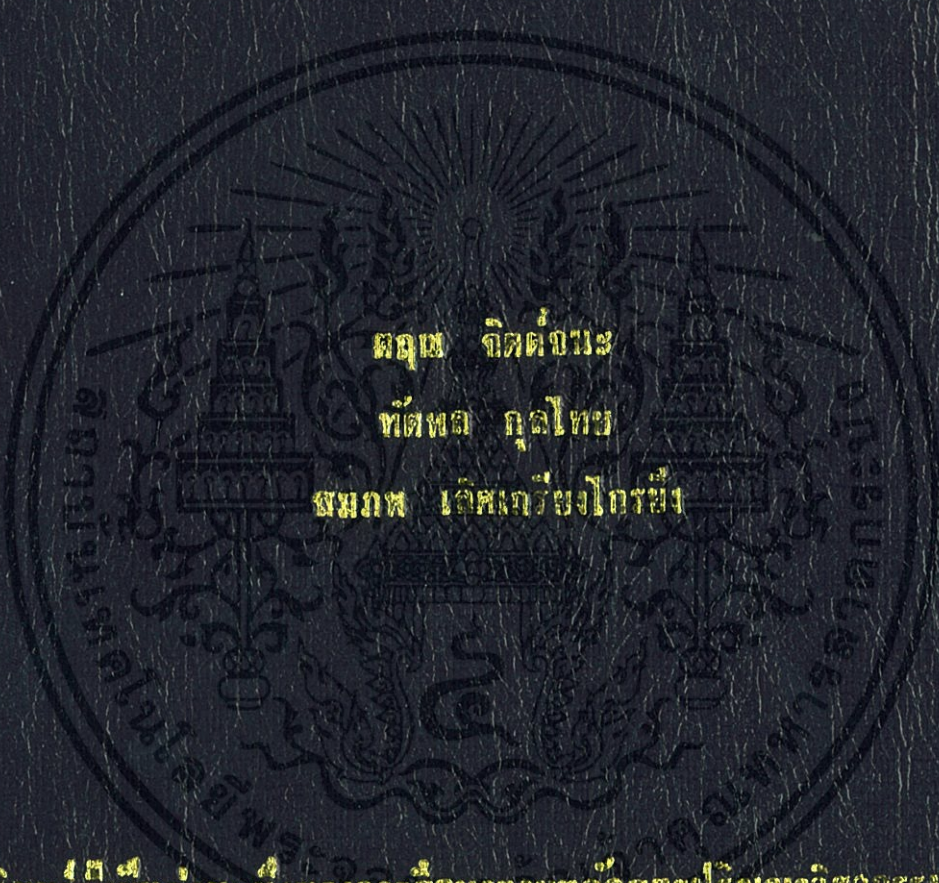


ระบบจัดการบริหารห้องสมุดโปรแกรมมิ่ง
PROGRAMMING LAB MANAGEMENT SYSTEM



ศาสตราจารย์ ดร. จิตต์ฉวี
ทัตพล กุลไทย
สมภพ เลิศเจริญโกกรักษ์

ปริญญาโทพัฒนศาสตร์ สาขาหนึ่งของภาควิชาศึกษาศาสตร์ มหาวิทยาลัยศรีนครินทรวิถียุทธศาสตร์
มหาวิทยาลัยศรีนครินทรวิถียุทธศาสตร์
คณะศึกษาศาสตร์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี กรุงเทพมหานคร
ปีการศึกษา 2555

ระบบจัดการบริหารห้องแล็บโปรแกรมมิ่ง
PROGRAMMING LAB MANAGEMENT SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2555

ปริญญาานิพนธ์ปีการศึกษา 2555

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจัดการห้องแล็บโปรแกรมมิง

PROGRAMMING LAB MANAGEMENT SYSTEM

ผู้จัดทำ

- | | | | |
|-------------|-------------------|--------------|----------|
| 1. นายตฤณ | จิตต์จนะ | รหัสนักศึกษา | 52010389 |
| 2. นายทัตพล | กุลไทย | รหัสนักศึกษา | 52010413 |
| 3. นายสมภพ | เลิศเกรียงไกรยิ่ง | รหัสนักศึกษา | 52011242 |



..... อาจารย์ที่ปรึกษา
(อาจารย์คณัฐ ตั้งติสานนท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดการห้องแลบโปรแกรมมิง

นาย ตฤณ	จิตต์จนะ	52010389
นาย ทัดพล	กุลไทย	52010413
นาย สมภพ	เลิศเกรียงไกรยิ่ง	52011242
อาจารย์คณัฐ ปีการศึกษา 2555	ตั้งติสานนท์	อาจารย์ที่ปรึกษา

บทคัดย่อ

การสอนวิชาที่เกี่ยวข้องกับการโปรแกรมมิง โดยปกติแล้วจะสอนทฤษฎีควบคู่กับการสอนภาคปฏิบัติ ซึ่งการสอนภาคปฏิบัติจะช่วยให้นักศึกษาได้ทดลองทำแบบฝึกหัดด้วยตนเอง โดยมีอาจารย์คอยให้คำปรึกษาและตรวจแบบฝึกหัดให้นักศึกษา โครงการนี้จะพัฒนาระบบจัดการห้องแลบโปรแกรมมิงในรูปแบบของเว็บแอปพลิเคชันโดยใช้ภาษาพีเอชพีเป็นภาษาหลักในการพัฒนาระบบ และใช้เทคโนโลยีคอมไพลเลอร์ในการแปลรหัสต้นฉบับภาษาซี โดยระบบจะแบ่งผู้ใช้งานออกเป็นสามประเภท คือ นักศึกษา อาจารย์และผู้ดูแลระบบ ซึ่งนักศึกษาสามารถส่งรหัสต้นฉบับให้ระบบเพื่อทำการตรวจผลลัพธ์และให้คะแนน อาจารย์สามารถบริหารจัดการแบบฝึกหัดและออกแบบเนื้อหาในชั้นเรียนได้ ส่วนผู้ดูแลระบบสามารถจัดการกับบัญชีผู้ใช้ นำเข้าหรือนำออกฐานข้อมูล และล้างฐานข้อมูล นอกจากนี้ระบบสามารถเก็บประวัติการใช้งานการทำแบบฝึกหัดของนักศึกษา และสร้างเอกสารรายงานผลการทำแบบฝึกหัดของนักศึกษาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMING LAB MANAGEMENT SYSTEM

Mr. Trin	Jitjana	52010389
Mr. Tattapol	Kulthai	52010413
Mr. Somphop	Lertkriangkraiying	52011242
Mr. Kanat	Tangtisanon	Advisor

Academic Year 2012

ABSTRACT

This project attempts to develop programming lab management system based on web application coding mainly in PHP language. Tiny C compiler is used to compile c source file into executable file. The system divides users into three categories. Students must verify the correctness of the program by themselves and then submit c source file to the system. The file will be compiled and run, and then system will generate the mark according to the correctness for each student program. Supervisors manage exercises, set exercise style and assign input to be test case for each exercise. Administrator manages user accounts, export database to csv file for backup or restore database by importing the csv file, and also format database. Furthermore, the system will log every student's usage and generate report for supervisor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลืออย่างยิ่งจาก อาจารย์คณัฐ ตั้งติสานนท์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการที่คอยติดตามดูแลเอาใจใส่ตลอดการทำงาน คอยรับฟังปัญหา อุปสรรคต่าง ๆ และชี้แนะแนวทางในการทำงานให้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณอาจารย์สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ให้คำปรึกษาในเรื่องต่าง ๆ ชี้แนะแนวทางการทำงาน ข้อควรปรับปรุงแก้ไข

ขอขอบพระคุณบุคคลในครอบครัวของข้าพเจ้าที่ให้การสนับสนุนในทุกเรื่องมาโดยตลอด และคอยให้กำลังใจในยามเหน็ดเหนื่อย ท้อแท้ ให้มีกำลังใจในการทำงานให้สำเร็จลุล่วงต่อไป

สุดท้ายนี้ขอขอบคุณรุ่นพี่และเพื่อน ๆ ทุกคนที่คอยให้คำปรึกษา แลกเปลี่ยนความคิดเห็น เป็นกำลังใจและช่วยเหลือในเรื่องเบ็ดเตล็ดต่าง ๆ

นาย ตฤณ	จิตต์จนะ
นาย ทัดพล	กุลไทย
นาย สมภพ	เลิศเกรียงไกรยิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 วิธีการดำเนินการ.....	2
1.4.1 การวางแผน.....	2
1.4.2 การวิเคราะห์และออกแบบระบบ.....	2
1.4.3 การทดลอง.....	2
1.4.4 สรุปผล.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 เทคโนโลยีเว็บ.....	4
2.1.1 วิวัฒนาการของเว็บ.....	4
2.1.2 สถาปัตยกรรมในการติดตั้งเว็บไซต์.....	6
2.1.3 เทคโนโลยีที่สนับสนุน.....	8
2.1.4 ภาษาสคริปต์สำหรับเว็บไซต์.....	8
2.2 ไคลเอนต์ - เซิร์ฟเวอร์ โมเดล.....	10
2.3 ภาษาพีเอชพี.....	11
2.3.1 โครงสร้างการทำงานของพีเอชพี.....	11

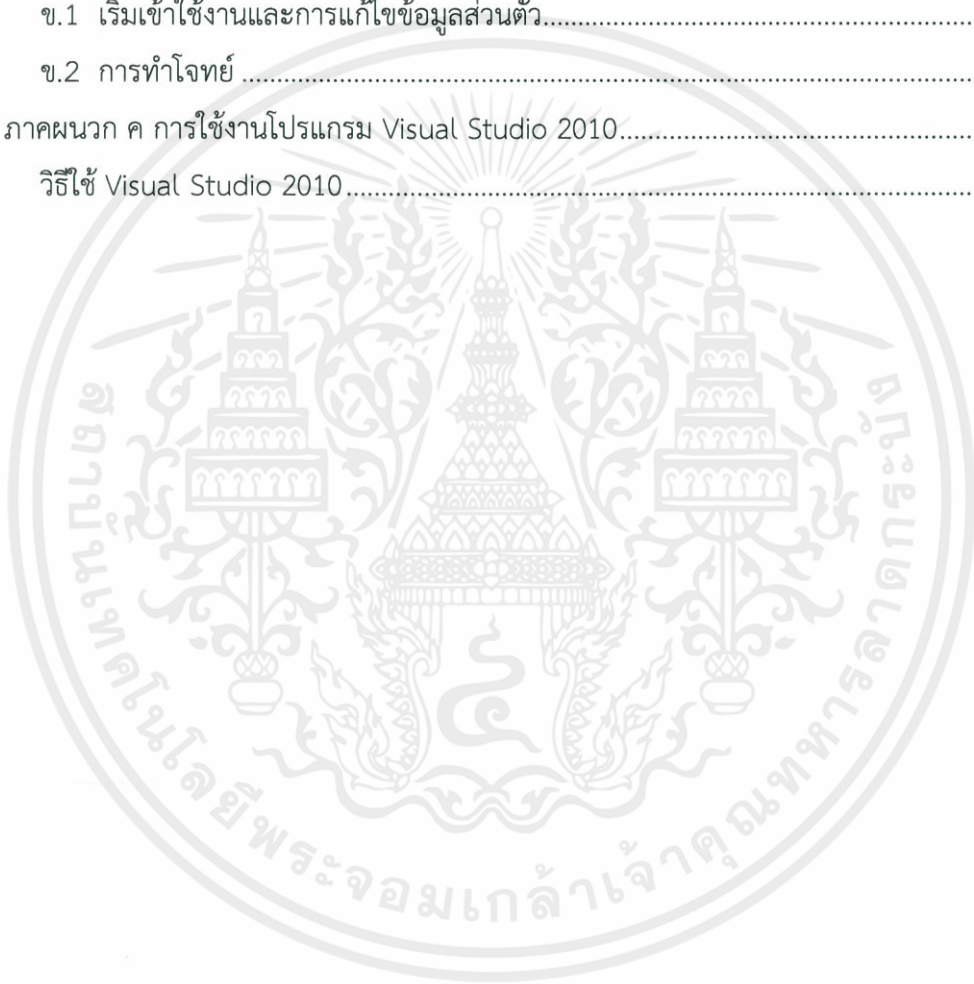
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2	ความรู้พื้นฐานภาษาพีเอชพี.....	13
2.4	ภาษาเอสคิวแอล.....	18
2.4.1	เพิ่มข้อมูล (INSERT INTO).....	18
2.4.2	เลือกข้อมูล (SELECT).....	19
2.4.3	การระบุเงื่อนไข (WHERE).....	19
2.4.4	เรียงลำดับข้อมูล (ORDER BY).....	19
2.4.5	ปรับปรุงข้อมูล (UPDATE).....	20
2.4.6	ลบข้อมูล (DELETE).....	20
2.5	โครงร่างพีเอชพี (PHP Framework).....	21
2.5.1	โครงร่าง (Framework).....	21
2.5.2	โค้ดอิกไนเตอร์ (CodeIgniter).....	21
2.5.3	ซีไอไซต์ (CI site).....	22
2.6	จาวาสคริปต์ (JavaScript).....	25
2.6.1	ลักษณะการทำงานของจาวาสคริปต์.....	26
2.6.2	รูปแบบของจาวาสคริปต์.....	26
2.6.3	ความสามารถของจาวาสคริปต์.....	26
2.6.4	ข้อจำกัดของจาวาสคริปต์.....	27
2.7	เอแจ็กซ์ (AJAX).....	27
2.7.1	ที่มาของปัญหา.....	27
2.7.2	โครงสร้างของเอแจ็กซ์.....	28
2.7.3	การทำงานของเอแจ็กซ์.....	29
2.7.4	ข้อดีของเอแจ็กซ์.....	30
2.8	คอมไพเลอร์ (Compiler).....	30
2.8.1	ความหมายของคอมไพเลอร์.....	31
2.8.2	การทำงานของคอมไพเลอร์.....	31
2.8.3	โปรแกรมคอมไพเตอร์.....	32
2.8.4	ไทนี่ซี คอมไพเลอร์ (Tiny C Compiler).....	33
2.9	นิพจน์ปรกติ (Regular expression).....	34
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า		
ไม่ว่ากรณีใดๆก็ตาม อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้		
	บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	36
3.1	ความต้องการของระบบ และลักษณะสำคัญ.....	36

3.1.1	วิเคราะห์รูปแบบของโจทย์	36
3.1.2	ความต้องการของระบบตรวจผลลัพธ์ของโปรแกรม	36
3.2	รายการตรวจสอบความต้องการของระบบ (System Requirement Checklist)	38
3.2.1	ข้อมูลนำเข้า (input)	38
3.2.2	ข้อมูลส่งออก (output)	38
3.2.3	กระบวนการ (process)	38
3.2.4	ประสิทธิภาพ (performance)	39
3.2.5	ส่วนควบคุม (control)	39
3.3	การใช้ระบบ	39
3.3.1	แผนภาพการใช้ระบบ (Use case diagram)	39
3.3.2	ตารางการใช้งานระบบ	41
3.4	การออกแบบฐานข้อมูล	42
3.5	การออกแบบส่วนติดต่อผู้ใช้	43
3.5.1	ผู้ใช้ประเภทนักศึกษา	43
3.5.2	ผู้ใช้ประเภทอาจารย์	46
3.5.3	ผู้ใช้ประเภทผู้ดูแลระบบ	50
บทที่ 4	การทดลองและผลการทดลอง	51
4.1	สภาพแวดล้อมที่ใช้ในระบบ	51
4.2	ขั้นตอนการทดลอง	51
4.2.1	การเข้าสู่ระบบ	51
4.2.2	การเพิ่มข้อมูลผู้ใช้	52
4.2.3	การจัดการโจทย์	53
4.2.4	การสร้างชั้นเรียน	55
4.2.5	การตรวจผลลัพธ์และการให้คะแนน	57
บทที่ 5	บทสรุป	61
5.1	สรุปผลการทดลอง	61
5.2	ปัญหา อุปสรรค และแนวทางการแก้ไข	61
5.3	แนวทางในการพัฒนาต่อ	62

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดูเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม	63
ภาคผนวก ก การติดตั้ง AppServ	64
ก.1 ความหมายของโปรแกรม AppServ	64
ก.2 ขั้นตอนการติดตั้ง	64
ภาคผนวก ข Direction Lab	68
ข.1 เริ่มเข้าใช้งานและการแก้ไขข้อมูลส่วนตัว	68
ข.2 การทำโจทย์	70
ภาคผนวก ค การใช้งานโปรแกรม Visual Studio 2010	74
วิธีใช้ Visual Studio 2010	74



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 รูปแบบแท็กแสดงส่วนของภาษาพีเอชที.....	13
2.2 อักขระต้องห้าม.....	15
2.3 ตัวดำเนินการต่าง ๆ ในภาษาพีเอชที.....	16
2.4 ฟิลด์ต่าง ๆ ของการเพิ่มข้อมูล.....	19
2.5 ฟิลด์ต่าง ๆ ของการปรับปรุงข้อมูล.....	20
2.6 ฟิลด์ต่าง ๆ ของการลบข้อมูล.....	20
2.7 ประเภทไฟล์และวิธีการใช้ไฟล์ต่าง ๆ ในซีไอไซต์.....	23
2.8 โฟลเดอร์ต่าง ๆ ในซีไอไซต์.....	25
2.9 สรุพออบชันของทีซีซี.....	33
2.10 การอธิบายสายอักขระ.....	34
3.1 การใช้งานระบบแบ่งตามประเภทผู้ใช้งาน.....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 เว็บเพจ และเว็บเบราว์เซอร์.....	5
2.2 การให้บริการข้อมูลผ่านเว็บ	6
2.3 สถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ หรือแบบซิงเกิลเทียร์	7
2.4 สถาปัตยกรรมแบบทูเทียร์.....	7
2.5 สถาปัตยกรรมแบบทรีเทียร์.....	8
2.6 การทำงานของภาษาสคริปต์ที่ฝั่งไคลเอนต์.....	9
2.7 การทำงานของภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์.....	10
2.8 ลำดับขั้นตอนการทำงานของเว็บไซต์แบบพลวัต.....	11
2.10 การไหลของข้อมูลในสถาปัตยกรรมแบบเอ็มซีวี.....	22
2.11 การเปรียบเทียบรูปแบบเว็บแอปพลิเคชันแบบปกติและเอแจ็กซ์.....	28
2.12 สถาปัตยกรรมของเอแจ็กซ์.....	29
2.13 การปรับปรุงส่วนติดต่อผู้ใช้บางส่วนและการสื่อสารแบบไม่เข้าจังหวะ.....	30
2.14 กระบวนการคอมไพล์โค้ดโปรแกรม.....	31
3.1 แผนภาพการทำงานส่วนตรวจโปรแกรมและให้คะแนน.....	37
3.2 แผนภาพการใช้งานระบบจัดการแลบโปรแกรมมิ่ง.....	40
3.3 เอนทิตีและความสัมพันธ์ระหว่างเอนทิตีของฐานข้อมูลในระบบ.....	42
3.4 หน้าต่างหลังของผู้ใช้ประเภทนักศึกษา.....	43
3.5 หน้าต่างรายละเอียดโจทย์และตัวอย่างผลลัพธ์ของโจทย์.....	44
3.6 หน้าต่างรายละเอียดการตรวจผลลัพธ์.....	45
3.7 หน้าต่างหลักของผู้ใช้ประเภทอาจารย์.....	46
3.8 หน้าต่างการเพิ่มโจทย์.....	47
3.9 ช่องกรอกกรณีทดสอบ.....	48
3.10 รูปแบบการตรวจ.....	49
3.11 ตัวเลือกเพิ่มเติมสำหรับการตรวจ.....	50
3.12 หน้าต่างหลักของผู้ใช้ประเภทผู้ดูแลระบบ.....	50
4.1 หน้าต่างกรอกข้อมูลเพื่อเข้าระบบ.....	51
4.2 หน้าต่างเพิ่มผู้ใช้.....	52
4.3 กรอกรายละเอียดของผู้ใช้.....	52

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นโดยคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ไม่ว่ากรณีใดๆ

4.4	รายการผู้ใช้	53
4.5	หน้าต่างเพิ่มโจทย์	53
4.6	หน้าต่างรายการโจทย์	54
4.7	หน้าต่างการแก้ไขโจทย์	54
4.8	หน้าต่างการสร้างชั้นเรียน	55
4.9	หน้าต่างรายการโจทย์ที่สามารถเลือก	56
4.10	หน้าต่างรายการคลาสเซสชัน	57
4.11	หน้าต่างหลักของนักศึกษา	58
4.12	คำสั่งโจทย์และตัวอย่างผลลัพธ์	59
4.13	หน้าต่างตรวจผลลัพธ์จากโปรแกรม	60
ก.1	การติดตั้งโปรแกรม AppServ	64
ก.2	เงื่อนไขการติดตั้ง	65
ก.3	เลือกปลายทางที่ต้องการติดตั้ง	65
ก.4	การเลือก Package ที่ต้องการติดตั้ง	66
ก.5	การกำหนดค่า Config Apache Server	66
ก.6	การกำหนดค่า Config MySQL Database	67
ก.7	การสิ้นสุดการติดตั้งโปรแกรม Appserv	67
ข.1	ใส่ ip ของเครื่อง server	68
ข.2	ใส่ username และรหัสลงไป	69
ข.3	การแก้ไขข้อมูลส่วนตัว	69
ข.5	เข้าสู่ระบบ	70
ข.6	การทำโจทย์	70
ข.7	ตัวอย่างการแสดงผล	71
ข.8	การส่งไฟล์	71
ข.9	ผลการตรวจ	72
ข.10	ประวัติการส่งงาน	72
ข.11	ออกจากระบบ	73
ค.1	การสร้างโปรเจค	74
ค.2	ตั้งชื่อโปรเจค	74
ค.3	ตั้งค่าโปรเจค	75
ค.4	เพิ่ม source file	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.5 เลือกชนิดของ source file.....	76
ค.6 เขียนโปรแกรม.....	76
ค.7 บันทึกโปรแกรม.....	77



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

การเรียนวิชาทฤษฎีโปรแกรมมิ่งต้องมีการเรียนวิชาแลบโปรแกรมมิ่งควบคู่ไป วิชาแลบโปรแกรมมิ่งต้องมีการจัดตารางเรียนแลบก่อน จึงให้นักศึกษาเข้ามาทำแลบเพื่อให้นักศึกษาได้ปฏิบัติเขียนโปรแกรมจริงตามที่ได้เรียนทฤษฎีและสามารถนำมาประยุกต์ใช้ได้ โดยขั้นตอนในการทำแลบโปรแกรมมิ่งคือ นักศึกษาเข้ามาห้องแลบ อาจารย์กำหนดโจทย์ จากนั้นนักศึกษาอ่านโจทย์และทำความเข้าใจเพื่อเขียนโปรแกรมให้ได้ผลลัพธ์ที่สอดคล้องกับโจทย์ที่กำหนด เมื่อนักศึกษาเขียนโปรแกรมเสร็จแล้วจึงเรียกให้อาจารย์มาตรวจผลลัพธ์ว่าถูกต้องตามโจทย์กำหนดหรือไม่ ซึ่งอาจารย์อาจซักถามเพื่อทดสอบความเข้าใจ จึงต้องมีการสร้างระบบเพื่อจัดตารางเรียนแลบโปรแกรมมิ่งเพื่อให้นักศึกษาได้เรียนรู้การเขียนโปรแกรมทั้งด้านไวยากรณ์ (Syntax) และขั้นตอนวิธี (Algorithm) จนสามารถนำมาประยุกต์เขียนโปรแกรมสำหรับโจทย์ต่าง ๆ ได้

การตรวจผลลัพธ์จากโปรแกรมและให้คะแนนภายในห้องแลบโปรแกรมมิ่งจะดำเนินการโดยอาจารย์ ทำให้อาจเกิดความล่าช้าในการตรวจเนื่องจากอัตราส่วนระหว่างอาจารย์และจำนวนนักศึกษาค่อนข้างน้อย

โครงการนี้ได้สร้างขึ้นมาเพื่อจัดตารางเรียนแลบโปรแกรมมิ่งที่เป็นไปตามนโยบายการเรียนวิชาโปรแกรมและเป็นการกระตุ้นให้นักศึกษาฝึกแก้ไขปัญหาในการเขียนโปรแกรมด้วยตนเองได้ อีกทั้งนักศึกษาสามารถตรวจสอบความถูกต้องของผลลัพธ์โปรแกรมได้ผ่านตัวเว็บแอปพลิเคชัน

1.2 วัตถุประสงค์ของโครงการ

วัตถุประสงค์ของโครงการนี้เพื่อจัดตารางเรียนวิชาแลบการเขียนโปรแกรมให้นักศึกษาระบบยังจัดทำเพื่อให้นักศึกษาสามารถส่งโค้ดผ่านเว็บเพื่อตรวจสอบผลลัพธ์ของโค้ดตามโจทย์กำหนด ทำให้อาจารย์ลดเวลาในการตรวจผลลัพธ์ของโปรแกรมของนักศึกษาตามที่โจทย์กำหนดและรายงานผลคะแนนการเรียนวิชาแลบโปรแกรมมิ่งของนักศึกษาแก่อาจารย์

1.3 ขอบเขตของโครงการ

เป็นระบบจัดการห้องแลบโปรแกรมมิ่งที่ผู้ดูแลระบบวิชาได้นำเข้าข้อมูลการลงทะเบียนของนักศึกษาแล้วระบบจะข้อมูลนั้นมาจัดตารางเรียนวิชาแลบโปรแกรมมิ่งให้นักศึกษาที่เรียนวิชา

โปรแกรมมิ่ง ซึ่งผู้ดูแลระบบวิชาสามารถเพิ่ม, ลดห้องเรียนแลบและทำการเปิดกลุ่มเรียนได้เพื่อให้เพียงพอต่อจำนวนนักศึกษาที่ลงทะเบียนเรียนวิชาโปรแกรมมิ่ง ทั้งระบบยังให้นักศึกษาอัปโหลดไฟล์โค้ดโปรแกรมเข้าสู่ระบบเพื่อให้ระบบนำไปคอมไพล์เพื่อให้ได้ผลลัพธ์ของโปรแกรมและนำไปตรวจสอบกับผลลัพธ์ของโจทย์ที่กำหนด จากนั้นแสดงผลการตรวจผลลัพ์โปรแกรม ซึ่งตรวจได้เฉพาะโค้ดโปรแกรมภาษาซี และแสดงประวัติการตรวจงานแต่ละบทเรียนให้แก่อาจารย์

1.4 วิธีการดำเนินการ

1.4.1 การวางแผน

ศึกษาระบบเก่าเพื่อนำมาวิเคราะห์และหาทางปรับปรุงในระบบใหม่ จากนั้นกำหนดวัตถุประสงค์ของโครงการ, ขอบเขตโครงการและระยะเวลาดำเนินโครงการ และศึกษาเกี่ยวกับเทคโนโลยีต่าง ๆ ที่มีความจำเป็นต่อระบบ

1.4.2 การวิเคราะห์และออกแบบระบบ

กำหนดรายการสิ่งที่ระบบต้องมีต่าง ๆ , เครื่องมือและซอฟต์แวร์ที่จำเป็น ออกแบบฐานข้อมูลเพื่อรองรับข้อมูลใหม่ ๆ , แผนภาพการทำงานของระบบและส่วนติดต่อผู้ใช้ให้เป็นไปตามวัตถุประสงค์ของโครงการ

1.4.3 การทดลอง

เขียนโปรแกรมส่วนย่อยเพื่อทดสอบแนวคิดที่ได้วิเคราะห์และออกแบบไว้ และทำการแก้ไขปรับปรุงเพื่อให้ส่วนต่าง ๆ ถูกต้องสมบูรณ์มากยิ่งขึ้น

1.4.4 สรุปผล

สรุปผลการทดลองและบันทึกปัญหาและอุปสรรคต่าง ๆ ที่พบเพื่อนำมาปรับปรุงแก้ไขตัวระบบต่อไปให้มีความเสถียรและพร้อมใช้งานได้จริงและทั้งเพิ่มเติมข้อเสนอแนะต่าง ๆ ที่ควรมีในระบบให้ดีขึ้นสำหรับการนำไปใช้งานจริง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น 1) ได้รับความรู้ความเข้าใจในเรื่องของเว็บเทคโนโลยี, การสื่อสารระหว่างคลเอนต์และเซิร์ฟเวอร์

2) ได้รับความรู้ความเข้าใจถึงกระบวนการพัฒนาระบบ ตั้งแต่การวางแผน, การวิเคราะห์, การออกแบบ จนถึงการสร้างและการนำระบบไปใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 เทคโนโลยีเว็บ

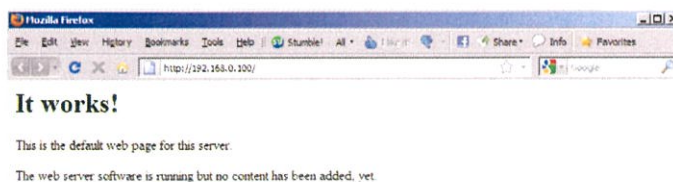
เว็บเป็นเทคโนโลยีของการสื่อสารที่ทำให้ผู้ใช้จากทั่วโลกสามารถเข้าใช้ข้อมูลที่มีให้ได้อย่างสะดวกและรวดเร็ว มีการเก็บเอกสารไว้ที่ฐานข้อมูลซึ่งจะถูกเข้าใช้และแสดงผลโดยเว็บเบราว์เซอร์ซึ่งเป็นโปรแกรมที่ถูกติดตั้งอยู่ที่เครื่องคอมพิวเตอร์ปลายทาง โดยในการใช้เว็บเบราว์เซอร์นี้ ผู้ใช้สามารถร้องขอข้อมูลหรือเอกสารได้จากคอมพิวเตอร์เครื่องอื่นที่อยู่บนกระบวนเครือข่าย และเมื่อได้รับข้อมูลมาแล้ว เว็บเบราว์เซอร์ก็จะแสดงผลข้อมูลเหล่านั้นให้แก่ผู้ใช้

2.1.1 วิวัฒนาการของเว็บ

แนวความคิดของการให้บริการเว็บเริ่มต้นจากผู้ให้บริการ ผู้ให้บริการจำเป็นต้องติดตั้งเครื่องบริการ ที่เรียกว่าเว็บเซิร์ฟเวอร์ขึ้นมา โดยอาจเรียกเว็บเซิร์ฟเวอร์นั้นว่าเว็บไซต์ จากนั้นนำข้อมูลที่ต้องการเผยแพร่ขึ้นไปไว้บนเว็บเซิร์ฟเวอร์ที่ได้รับอนุญาตให้สามารถเข้าใช้บริการจากภายนอกได้

เว็บเซิร์ฟเวอร์ที่ตั้งตั้งขึ้นนั้นจะต้องเชื่อมต่อกับระบบอินเทอร์เน็ตด้วยโพรโตคอลที่ซีพีไอพี (TCP/IP: Transmission control protocol/Internet protocol) ผ่านทางพอร์ต 80 และจะต้องสนับสนุนการให้บริการข้อมูลผ่านโพรโตคอลเอชทีทีพี (HTTP: Hypertext transfer protocol) ในชั้นแอปพลิเคชัน ในส่วนของผู้ใช้งานนั้นจำเป็นต้องติดตั้งโปรแกรมเว็บเบราว์เซอร์ ซึ่งในการใช้งานผู้ใช้จะต้องพิมพ์ยูอาร์แอล (URL: Unified Resource Locator) ซึ่งเป็นตำแหน่งที่บ่งบอกถึงเว็บเซิร์ฟเวอร์และข้อมูลที่ต้องการใช้ เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอจากผู้ใช้ก็จะตรวจสอบเงื่อนไขต่าง ๆ และส่งข้อมูลกลับไปให้แก่ผู้ใช้ที่ทำการร้องขอข้อมูลมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 เว็บเพจ และเว็บเบราว์เซอร์

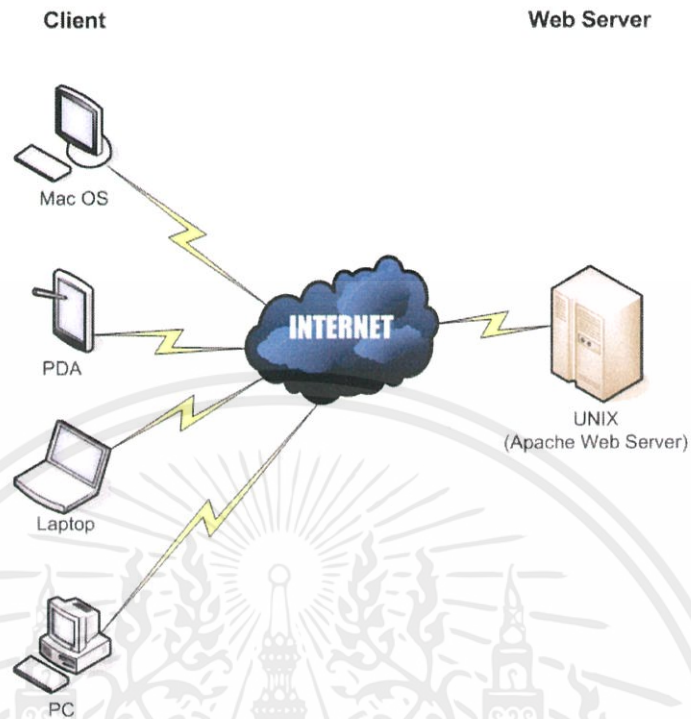
ในส่วนของข้อมูลที่ถูกเก็บอยู่ภายในเซิร์ฟเวอร์นั้น เพื่อให้เกิดความเป็นกลางในการนำเสนอข้อมูลของเว็บ จึงได้มีการจัดตั้งมาตรฐานของรูปแบบข้อมูลสำหรับเว็บขึ้น มีชื่อเรียกว่า เอชทีเอ็มแอล (HTML: Hyper Text Markup Language) โดยผู้ให้บริการต้องจัดรูปแบบข้อมูลของตนให้อยู่ในรูปแบบของเอชทีเอ็มแอล และเมื่อข้อมูลถูกส่งไปยังผู้ใช้ โปรแกรมเว็บเบราว์เซอร์จะทำการแปลงรูปแบบของเอชทีเอ็มแอลให้มีโครงสร้างและหน้าตาตามที่ผู้ให้บริการต้องการ

จากพื้นฐานของการให้บริการเว็บจะเห็นได้ว่า เว็บเป็นการให้บริการข้อมูลในทิศทางเดียว กล่าวคือ จากผู้ให้บริการไปยังผู้ใช้บริการ ซึ่งอินเทอร์เน็ตนั้นมีการเติบโตที่รวดเร็ว โดยสาเหตุหนึ่งก็มาจากการให้บริการเว็บ ทำให้เกิดความสะดวกรวดเร็วในการเข้าถึงข้อมูลต่าง ๆ

เมื่อผู้ใช้เว็บมีจำนวนมากขึ้น จากรูปแบบเริ่มต้นของการให้บริการแบบทิศทางเดียวในรูปแบบของข้อความและรูปภาพ ก็ได้มีการขยายไปสู่ความหลากหลายของชนิดข้อมูล เช่น เสียง และวิดีโอ นอกจากนี้ยังมีการรับข้อมูลโต้ตอบกับผู้ใช้ได้

จากความสามารถในการรับข้อมูลและโต้ตอบกับผู้ใช้ที่ตนเองที่ส่งผลให้เกิดแอปพลิเคชันรูปแบบใหม่ คือ เว็บแอปพลิเคชัน โดยทั่วไปโปรแกรมที่ผู้ใช้ใช้งานกันนั้นจะเป็นโปรแกรมที่ติดตั้งอยู่บนคอมพิวเตอร์ ซึ่งมีข้อจำกัดคือผู้ใช้ต้องติดตั้งโปรแกรมลงบนเครื่องของตนเอง ซึ่งการให้บริการแอปพลิเคชันแบบนี้ก็สามารถช่วยอำนวยความสะดวกได้อีกรูปแบบหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การให้บริการข้อมูลผ่านเว็บ

จากความสามารถของเว็บในการสื่อสารข้อมูลแบบสองทางและความเป็นหนึ่งในการนำเสนอข้อมูล ทำให้นักพัฒนาได้เป็นประโยชน์และเริ่มพัฒนาแอปพลิเคชันบนพื้นฐานของเว็บออกมา ซึ่งมีข้อได้เปรียบกว่าแบบเก่า คือ ผู้ใช้สามารถเข้าถึงแอปพลิเคชันจากที่ใดก็ได้ที่มีโปรแกรมเว็บเบราว์เซอร์ ซึ่งไม่ได้ถูกจำกัดอยู่แต่เพียงคอมพิวเตอร์ตั้งโต๊ะเท่านั้น แอปพลิเคชันนี้ยังสามารถเข้าถึงได้จากโทรศัพท์เคลื่อนที่ รวมถึงอุปกรณ์ต่าง ๆ ได้เช่นเดียวกัน

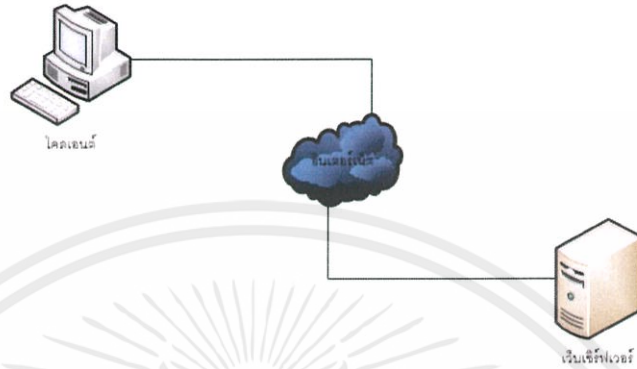
2.1.2 สถาปัตยกรรมในการติดตั้งเว็บไซต์

สถาปัตยกรรมในการติดตั้งเว็บไซต์สามารถแบ่งออกได้เป็น 3 รูปแบบ ตามการให้บริการของเว็บ คือ

- 1) สถาปัตยกรรมแบบไคลเอนต์ - เซิร์ฟเวอร์ (Client - Server) หรือแบบซิงเกิลเทียร์ (Single - tier)
- 2) สถาปัตยกรรมแบบทูเทียร์ (Two - tier)
- 3) สถาปัตยกรรมแบบทรีเทียร์ (Three - tier)

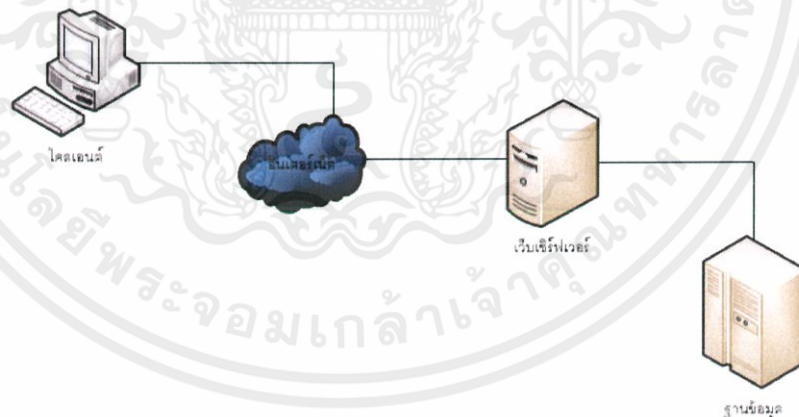
เอกสารนี้เป็นเอกสารที่สำหรับสถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์หรือสถาปัตยกรรมแบบซิงเกิลเทียร์ เป็นที่ไม่ว่ากรณีใด สถาปัตยกรรมที่ประกอบด้วยเว็บเซิร์ฟเวอร์และไคลเอนต์ โดยเว็บเซิร์ฟเวอร์นั้นจะทำหน้าที่ในการ

ให้บริการข้อมูลเว็บ ส่วนไคลเอนต์ทำหน้าที่รับข้อมูลจากเว็บเซิร์ฟเวอร์มาแสดงผลบนเว็บเบราว์เซอร์ และอาจจะทำการส่งข้อมูลจากผู้ใช้กลับไปให้กับเซิร์ฟเวอร์ได้



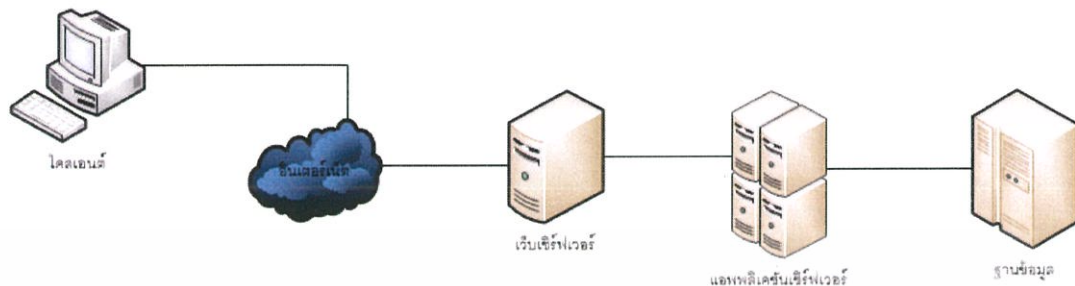
รูปที่ 2.3 สถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ หรือแบบซิงเกิลเทียร์

สถาปัตยกรรมแบบทูเทียร์ จะเป็นสถาปัตยกรรมของเว็บที่มีการเพิ่มระบบฐานข้อมูล (Database) เข้ามา ซึ่งสถาปัตยกรรมแบบนี้จะเหมาะกับเว็บแอปพลิเคชันที่เน้นการเก็บข้อมูลของผู้ใช้เป็นหลัก ซึ่งการนำระบบฐานข้อมูลเข้ามาใช้นั้นจะทำให้การเก็บและดึงข้อมูลสามารถทำได้ผ่านภาษาเอสคิวแอล (SQL) หากเลือกใช้ระบบฐานข้อมูลแบบสัมพันธ์ (Relational DBMS)



รูปที่ 2.4 สถาปัตยกรรมแบบทูเทียร์

สำหรับสถาปัตยกรรมแบบทรีเทียร์นั้น เป็นสถาปัตยกรรมที่ได้แยกระบบออกเป็น ระบบฐานข้อมูล แอปพลิเคชันเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ และไคลเอนต์ ในสถาปัตยกรรมแบบนี้จะมีการแยกส่วนแอปพลิเคชันเซิร์ฟเวอร์กับเว็บเซิร์ฟเวอร์ออกจากกัน โดยส่วนของแอปพลิเคชันเซิร์ฟเวอร์นั้นจะทำหน้าที่ในการประมวลผลหรือทำธุรกรรมต่าง ๆ เป็นหลัก ส่วนเว็บเซิร์ฟเวอร์จะนำผลลัพธ์ที่ได้จากการประมวลผลมาจัดรูปแบบในแบบของเอชทีเอ็มแอล เพื่อส่งไปแสดงผลยังไคลเอนต์



รูปที่ 2.5 สถาปัตยกรรมแบบทรี-tier

2.1.3 เทคโนโลยีที่สนับสนุน

เนื่องจากการให้บริการเว็บในปัจจุบันไม่ได้เป็นเพียงการให้บริการข้อมูลแบบเอชทีเอ็มแอลเท่านั้น เว็บไซต์ยังมีความสามารถในการให้บริการแบบโต้ตอบ การประมวลผล และการจัดเก็บข้อมูลของผู้ใช้ด้วย ดังนั้นผู้พัฒนาจึงจำเป็นต้องมีเครื่องมือและมีภาษาในการเขียนโปรแกรมที่เหมาะสม เพื่อให้การพัฒนาเว็บเป็นไปได้อย่างมีประสิทธิภาพ

สำหรับเครื่องมือที่เกี่ยวข้องกับการพัฒนาเว็บสามารถแบ่งออกได้เป็น 2 ประเภท คือ เครื่องมือสำหรับการสร้างเพจเอชทีเอ็มแอล ซึ่งรวมความสามารถในการวางโครงสร้างเว็บไซต์และเครื่องมือช่วยในการเขียนโปรแกรมซึ่งรวมถึงการติดต่อกับฐานข้อมูลด้วย สำหรับโปรแกรมในการสร้างเพจเอชทีเอ็มแอล และวางโครงสร้างเว็บไซต์ จะเป็นเครื่องมือที่เข้ามาช่วยในการสร้างส่วนติดต่อผู้ใช้เป็นหลัก ตัวอย่างเช่น Adobe Dreamweaver, Microsoft FrontPage เป็นต้น

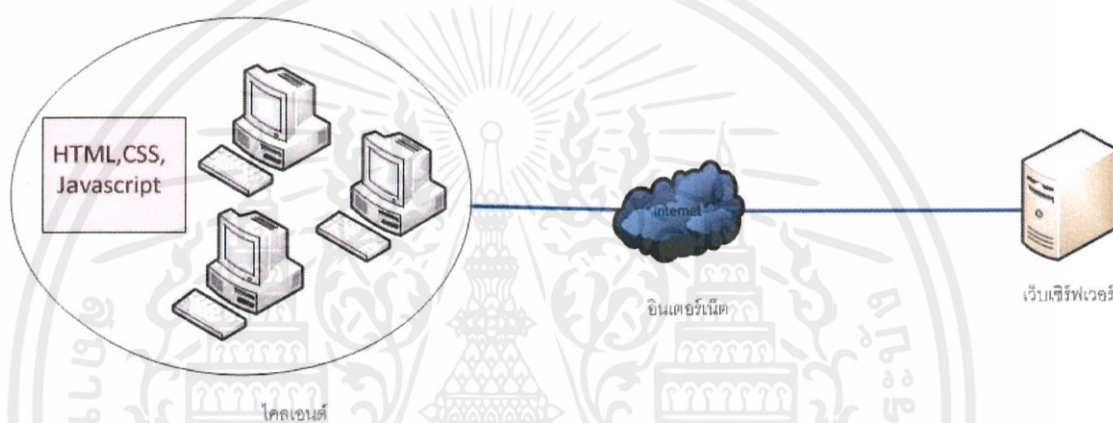
การตัดสินใจเลือกภาษาและเครื่องมือในการพัฒนานั้นก็มีหลากหลายปัจจัยที่เข้ามาเกี่ยวข้อง เช่น หากเว็บที่ต้องการสร้างขึ้นเป็นแอปพลิเคชันพื้นฐานที่มีการทำงานไม่ซับซ้อน และให้บริการต่อผู้ใช้งานจำนวนน้อย การเลือกใช้ภาษาสคริปต์อย่าง เพิร์ล (Perl), พีเอชพี (PHP) หรือเอเอสพี (ASP) ก็เป็นสิ่งที่เหมาะสม เนื่องจากภาษาสคริปต์เป็นสิ่งที่เรียนรู้และใช้งานได้ง่าย แต่มีข้อเสีย คือ ภาษาสคริปต์เหล่านี้มีแนวโน้มที่จะทำให้โครงสร้างของโปรแกรมไม่มั่นคงและยากต่อการแก้ไข ซึ่งไม่เหมาะกับแอปพลิเคชันขนาดใหญ่และซับซ้อน แต่สามารถนำโครงร่าง (Framework) มาใช้งานร่วมเพื่อช่วยในการรักษาโครงสร้างของเว็บแอปพลิเคชันขนาดใหญ่ได้

2.1.4 ภาษาสคริปต์สำหรับเว็บไซต์

ภาษาเอชทีเอ็มแอลนั้นเป็นเพียงภาษาที่ใช้ในการแสดงผล ดังนั้นเพื่อให้เว็บสามารถคำนวณหรือตัดสินใจได้ จำเป็นต้องใช้ภาษาในการเขียนโปรแกรมหรือภาษาสคริปต์อื่น ๆ ช่วยเพิ่มความสามารถลงในเพจเอชทีเอ็มแอล ซึ่งภาษาสคริปต์สามารถแบ่งออกได้เป็น 2 ระดับ คือ ภาษา

สคริปต์ที่ฝั่งไคลเอนต์ (Client - side Script) และภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์ (Server - side Script)

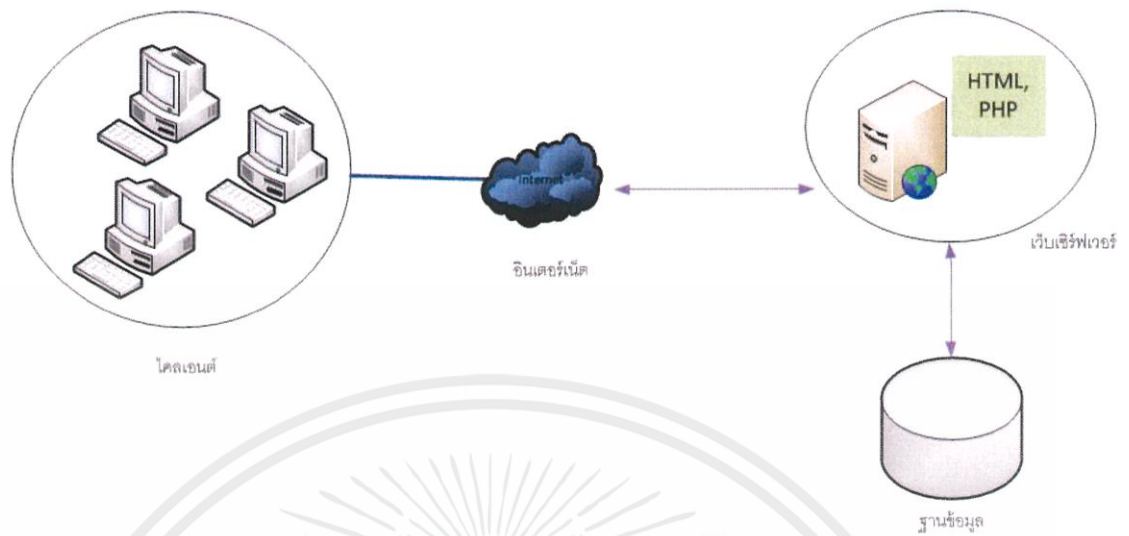
สำหรับภาษาสคริปต์ที่ฝั่งไคลเอนต์นั้น โค้ดของภาษาสคริปต์เหล่านี้จะฝังตัวอยู่ในเพจเอชทีเอ็มแอล ดังนั้น เมื่อผู้ใช้งานดาวน์โหลดเพจเอชทีเอ็มแอลมา ภาษาสคริปต์เหล่านี้ก็จะติดมาด้วย และเมื่อตัวแปรความหมายของภาษาเอชทีเอ็มแอลในโปรแกรมเว็บเบราว์เซอร์พบแท็กที่บ่งบอกว่าเป็นส่วนของโค้ดภาษาสคริปต์ โปรแกรมจะเรียกส่วนของอินเทอร์พรีเตอร์ (Interpreter) ขึ้นมาประมวลผลและทำงานตามภาษาสคริปต์ที่ได้เขียนไว้



รูปที่ 2.6 การทำงานของภาษาสคริปต์ที่ฝั่งไคลเอนต์

ในส่วนของภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์นั้นเป็นภาษาสคริปต์ที่เขียนขึ้นมาเพื่อใช้ในการสร้างเว็บเพจตามข้อมูลหรือตามความต้องการของผู้ใช้ ภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์ก็จะมีฝั่งโค้ดไว้ในเพจเอชทีเอ็มแอลเช่นกัน แต่ในการทำงานนั้น เมื่อมีการร้องขอเพจเอชทีเอ็มแอลเข้ามา เว็บเซิร์ฟเวอร์จะไม่ได้ส่งไฟล์เอชทีเอ็มแอลนั้นไปในทันที แต่จะมีการเรียกอินเทอร์พรีเตอร์ขึ้นมาทำงานก่อน เพื่อที่จะเปลี่ยนแท็กที่ฝังอยู่ในเอชทีเอ็มแอลให้กลายเป็นแท็กเอชทีเอ็มแอลตามที่ได้เขียนโปรแกรมภาษาสคริปต์ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การทำงานของภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์

จากคุณลักษณะของการทำงานที่แตกต่างกัน ทำให้ภาษาสคริปต์ทั้งสองแบบมีความสามารถที่แตกต่างกันด้วย ทั้งนี้ภาษาสคริปต์ที่ฝั่งไคลเอนต์นั้นจะทำงานที่ไคลเอนต์เลย ทำให้จำเป็นต้องมีการจำกัดความสามารถของภาษาสคริปต์ มิฉะนั้นแล้วอาจมีผู้ไม่ประสงค์ดีเขียนโปรแกรมที่อาจทำอันตรายแก่ไคลเอนต์ที่ดาวน์โหลดเพจไปใช้งานได้

จากสาเหตุข้างต้นนั้น ทำให้ภาษาสคริปต์ฝั่งไคลเอนต์จึงมีความสามารถไม่มากนัก ส่วนใหญ่นักพัฒนาเว็บจะใช้ภาษาสคริปต์ในการสร้างลูกเล่นเพื่อทำให้หน้าตาของเว็บไซต์มีความน่าสนใจมากขึ้น

แต่ภาษาสคริปต์ในฝั่งเซิร์ฟเวอร์นั้นจะแตกต่างกัน เนื่องจากภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์นั้นถูกใช้ในการสร้างเพจเอชทีเอ็มแอล ที่ประกอบด้วยข้อมูลตามที่ผู้ใช้ต้องการ และภาษาสคริปต์ก็มีการทำงานอยู่ที่เซิร์ฟเวอร์ ดังนั้นความสามารถของภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์จึงมีความสามารถมากกว่า เช่น สามารถเชื่อมต่อกับระบบฐานข้อมูล และอาจสามารถเชื่อมต่อกับเซิร์ฟเวอร์เครื่องอื่น ๆ ได้ด้วย

2.2 ไคลเอนต์ - เซิร์ฟเวอร์ โมเดล

ไคลเอนต์ - เซิร์ฟเวอร์ เป็นการอธิบายความสัมพันธ์ระหว่างโปรแกรมคอมพิวเตอร์สองโปรแกรม โดยโปรแกรมหนึ่งเป็นไคลเอนต์ จะสร้างคำขอของบริการจากอีกโปรแกรมหรือเซิร์ฟเวอร์ แนวคิดนี้เป็นแนวคิดสำคัญในระบบเครือข่ายแบบจำลองไคลเอนต์ - เซิร์ฟเวอร์ให้แบบแผนการติดต่อภายในโปรแกรมที่ให้ประสิทธิภาพการกระจายข้ามตำแหน่งที่ต่างกัน

นอกจากนี้ยังเป็นอีกหนึ่งสิ่งที่จะต้องรู้หรือที่เรียกว่าการสื่อสารที่จำเป็นซึ่งมีอยู่ทั่วไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น ระบบจำลองไคลเอนต์ - เซิร์ฟเวอร์จะมีเซิร์ฟเวอร์หนึ่งเครื่อง และคอยคำร้องขอของไคลเอนต์ โดยปกติโปรแกรมไคลเอนต์หลายโปรแกรมใช้บริการเซิร์ฟเวอร์ร่วมกัน ซึ่งมักจะเป็นส่วน

หนึ่งของโปรแกรมประยุกต์ขนาดใหญ่ เมื่อเปรียบเทียบกับอินเทอร์เน็ต เว็บเบราว์เซอร์ของผู้ใช้โปรแกรมเป็นไคลเอนต์ที่ขอบริการจากเว็บเซิร์ฟเวอร์ ในทางเทคนิคเรียกว่า Hypertext Transfer Protocol Server ซึ่งเป็นคอมพิวเตอร์อีกเครื่องบนอินเทอร์เน็ต

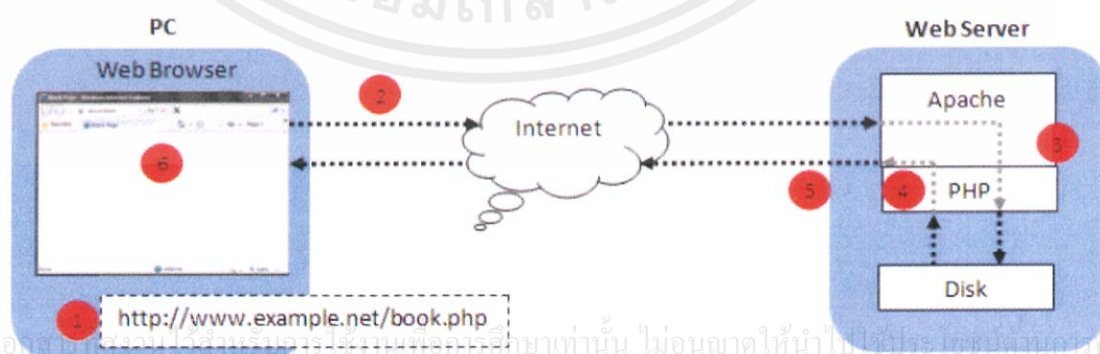
เครือข่ายแบบไคลเอนต์ - เซิร์ฟเวอร์เหมาะกับระบบเครือข่ายที่ต้องการเชื่อมต่อกับเครื่องไคลเอนต์จำนวนมาก โดยการรองรับจำนวนเครื่องไคลเอนต์อาจมากถึงหลักพัน เพราะฉะนั้นเครื่องที่จะนำมาทำหน้าที่ให้บริการจะต้องเป็นเครื่องที่มีประสิทธิภาพสูงเนื่องจากต้องสามารถทนทานต่อความผิดพลาด (Fault Tolerance) และต้องคอยให้บริการทรัพยากรให้กับเครื่องลูกข่ายตลอดเวลา สำหรับองค์กรขนาดใหญ่อาจมีการติดตั้งหลาย ๆ เซิร์ฟเวอร์เพื่อแบ่งภาระการทำงาน

2.3 ภาษาพีเอชพี

พีเอชพี (PHP: PHP Hypertext Preprocessor) เป็นภาษาสคริปต์ฝั่งเซิร์ฟเวอร์ซึ่งจะถูกประมวลผลบนเซิร์ฟเวอร์เป็นเว็บเพจ แล้วส่งเว็บเพจแสดงผลบนเบราว์เซอร์และสามารถเขียนรวมกับเอชทีเอ็มแอล, จาวาสคริปต์ และภาษาสำหรับเว็บอื่น ๆ

2.3.1 โครงสร้างการทำงานของพีเอชพี

เว็บไซต์นั้นแบบออกเป็น 2 ประเภทตามลักษณะของการประมวลข้อมูล คือ เว็บไซต์แบบสถิต (Static) และ เว็บไซต์แบบพลวัต (Dynamic) สำหรับเว็บไซต์แบบสถิตนั้นเป็นเว็บไซต์ที่มีข้อมูลเป็นเอกสารเอชทีเอ็มแอลเพียงอย่างเดียว ส่วนเว็บไซต์แบบพลวัต นั้น นอกจากเอกสารเอชทีเอ็มแอลแล้ว ยังมีการเพิ่มโค้ดของโปรแกรมที่เป็นภาษาสคริปต์ที่ฝั่งเซิร์ฟเวอร์อย่างเช่น พีเอชพี, เจเอสพี หรือ เอเอสพี เข้าไปด้วย ซึ่งช่วยเพิ่มความสามารถในการทำงาน เช่น การรับส่งข้อมูล จากผู้ใช้หรือการติดต่อฐานข้อมูล เป็นต้น



เอกสารนี้เป็นเอกสารที่เผยแพร่โดยกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ ไม่อนุญาติให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิรูปที่ 2.8 ลำดับขั้นตอนการทำงานของเว็บไซต์แบบพลวัต ที่มีการนำไปใช้

1) เริ่มต้นโดยเมื่อเราต้องการเปิดดูเว็บไซต์ เราจะเปิดโปรแกรมเรียกดูเว็บขึ้นมา ก่อน เช่น Internet Explorer จากนั้นพิมพ์ยูอาร์แอลของเว็บไซต์ที่ต้องการ ซึ่งในตัวอย่างคือ <http://www.example.net/book.php>

2) โปรแกรมเรียกดูเว็บจะส่งข้อมูลไปยังเว็บไซต์ www.example.net เพื่อขอข้อมูลในที่นี้คือไฟล์ `book.php`

3) เว็บเซิร์ฟเวอร์ที่ทำงานอยู่บนเว็บไซต์ปลายทาง (ในที่นี้คือ Apache) ได้รับข้อมูลและตรวจสอบพบว่าไฟล์ที่ต้องการเป็นไฟล์พีเอชพี จึงส่งให้พีเอชพีเอนจิน (PHP engine) ทำการประมวลผล

4) พีเอชพีเอนจินอ่านไฟล์ข้อมูล `book.php` และทำการประมวลผลข้อมูลตามโค้ดที่เขียนไว้ในไฟล์นี้ (ในขั้นตอนนี้อาจมีการประมวลผลและติดต่อกับบริการอื่น ๆ เช่น ติดต่อกับฐานข้อมูลหรือติดต่อกับไดเรกทอรีเซิร์ฟเวอร์ เป็นต้น) จากนั้นจึงส่งผลลัพธ์ที่ได้ซึ่งอยู่ในรูปแบบของเอชทีเอ็มแอลกลับไปยัง Apache

5) Apache ส่งผลลัพธ์กลับไปให้โปรแกรมเรียกดูเว็บที่ฝั่งไคลเอนต์

6) เมื่อโปรแกรมเรียกดูเว็บที่ได้รับข้อมูลซึ่งเป็นเอกสารเอชทีเอ็มแอลก็จะแสดงผลลัพธ์ให้
ผู้ชม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ความรู้พื้นฐานภาษาพีเอชพี

2.3.2.1 การเปิด - ปิดแท็กพีเอชพี

การเริ่มต้นเขียนภาษาพีเอชพีนั้นจะต้องมีการกำหนดขอบเขตส่วนของภาษาพีเอชพีไว้ในเพจ เพราะเพจหนึ่งอาจมีหลายภาษาประกอบกัน จึงต้องทำการแบ่งส่วนให้ชัดเจนเพื่อปกป้องไม่ให้อินเทอร์เน็ตเวิร์กของแต่ละภาษาเกิดความสับสนในการนำชุดคำสั่งไปแปลภาษา ซึ่งรูปแบบการเปิด - ปิดแท็กนั้นมี 4 รูปแบบ คือ

ตารางที่ 2.1 รูปแบบแท็กแสดงส่วนของภาษาพีเอชพี

รูปแบบแท็ก	เปิดแท็กพีเอชพี	ปิดแท็กพีเอชพี
แบบมาตรฐาน	<?php	?>
แบบสั้น	<?	?>
แบบ ASP	<%	%>
แบบ Script	<script language="PHP">	</script>

2.3.2.2 รูปแบบคำสั่ง (statement)

เมื่อเขียนคำสั่งของภาษาพีเอชพีแล้วจะต้องปิดท้ายด้วยเครื่องหมายอัฒภาค (;) ทุกครั้งเพื่อบ่งบอกถึงการสิ้นสุดคำสั่ง

โปรแกรม 2.1 โค้ดแสดงผลลัพธ์ "Hello world"

```
<HTML>
<BODY>
<?php
    echo "Hello, World";
?>
</BODY>
</HTML>
```

2.3.2.3 ตัวแปร (Variable)

การประกาศตัวแปรของภาษาพีเอชพีนั้นมีการตั้งชื่อที่คล้ายคลึงกับภาษาโปรแกรมมิ่งอื่น ๆ ซึ่งจะต้องเริ่มต้นด้วยอักษรภาษาอังกฤษ หรือเครื่องหมายเส้นใต้อักษร (Underscore) แต่ตัวแปรของภาษาพีเอชพีนั้นต้องมีเครื่องหมายสกุลเงินดอลลาร์ (\$) นำหน้าด้วย

โปรแกรม 2.2 การประกาศตัวแปรที่ถูกต้องและไม่ถูกต้อง

```
// Correct variable declaration
$total;
$_cell1;
$_length_of_string

// Wrong variable declaration
total;
$1_total;
$2_length;
```

2.3.2.4 หมายเหตุ (Comment)

หมายเหตุมีไว้สำหรับเขียนคำอธิบายคำสั่ง หรือชุดคำสั่ง โดยส่วนที่เป็นหมายเหตุจะไม่ถูกนำไปประมวลผลโดยอินเทอร์พรีเตอร์

2.3.2.5 การกำหนดค่าให้ตัวแปร (variable initialization)

โดยปกติแล้วการกำหนดค่าให้แก่ตัวแปรจะกำหนดผ่านเครื่องหมายเท่ากับ (=) ซึ่งมีทั้งการกำหนดค่าให้ตัวเลขและสายอักขระ โดยที่การกำหนดค่าให้ตัวแปรนั้นสามารถทำได้พร้อมกับการประกาศตัวแปรทันที

โปรแกรม 2.3 การกำหนดค่าให้ตัวแปร

```
<?php
$total = 10;
?>
// Difference between single quote and double quote
<php
$total = 10;
$example1 = 'The total is $total';
$example2 = "The total is $total";
echo $ example1
echo $ example2
// Output of $example1 is The total is $total
// Output of $example2 is The total is 10
?>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.6 การเชื่อมสายอักขระ (String concatenation)

สามารถเชื่อมต่อสายอักขระได้มากกว่าหนึ่งสาย โดยใช้เครื่องหมายมหัพภาค (.)

โปรแกรม 2.4 การเชื่อมต่อสายอักขระ

```
<php
$a = 'apples';
$b = 'bananas';
$c = '';
$c = $a . ' and ' . $b;
?>
```

2.3.2.7 อักขระต้องห้าม (Escaped characters)

อักขระต้องห้าม คือ อักขระพิเศษที่ทำหน้าที่ต่างจากอักขระทั่วไป วิธีการใช้งานจะต้องนำหน้าอักขระเหล่านี้ด้วยเครื่องหมายทับย้อนหลัง (\)

ตารางที่ 2.2 อักขระต้องห้าม

Character	Escaped Character	Description
ไม่มี	\n	ป้อนบรรทัด
ไม่มี	\r	ปิดแคร์
ไม่มี	\t	เพิ่มย่อหน้า
\	\\	แสดงเครื่องหมาย \
\$	\\$	แสดงเครื่องหมาย \$
"	\"	แสดงเครื่องหมาย "

2.3.2.8 แกลลุ่ม (Array)

คือตัวแปรชนิดหนึ่งที่สามารถเก็บค่าได้หลายค่าในตัวแปรเดียวกัน การสร้างตัวแปรแกลลุ่มให้เรียกใช้ฟังก์ชัน array() ซึ่งจะถูกชี้ตำแหน่งโดยคีย์ โดยปกติคีย์จะมีค่าเป็นตัวเลข หากตัวแปรแกลลุ่มใด ๆ มีคีย์เป็นสายอักขระจะเรียกตัวแปรแกลลุ่มนั้นว่าแกลลุ่มเชื่อมโยง (Associative array) ซึ่งมีลักษณะคล้ายโครงสร้างข้อมูลแบบพจนานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 2.5 การประกาศและกำหนดค่าให้ตัวแปรแถวลำดับ

```
$World = array(1 => "moon", 2 => "sun");
$Computer = new array();
$Computer[0] = "Mouse";
$Computer['cpu'] = "Processor";
```

2.3.2.9 ตัวดำเนินการ (Operator)

ตัวดำเนินการใช้สัญลักษณ์ต่าง ๆ ในการกำหนด ซึ่งมีทั้งการกำหนดค่าให้แก่ตัวแปร, การคำนวณทางคณิตศาสตร์, การเปรียบเทียบเชิงตรรกศาสตร์ เป็นต้น

ตารางที่ 2.3 ตัวดำเนินการต่าง ๆ ในภาษาพีเอชพี

Operator	ความหมาย
==	เท่ากับ
!= , <>	ไม่เท่ากับ
< , >	น้อยกว่า, มากกว่า
<= , >=	น้อยกว่าหรือเท่ากับ, มากกว่าหรือเท่ากับ

2.3.2.10 คำสั่งควบคุม (Control structure)

คำสั่งควบคุมใช้ในการตรวจสอบเงื่อนไขต่าง ๆ รวมถึงการซ้ำเพื่อให้สอดคล้องและทำงานตามแนวคิดของผู้เขียนโปรแกรม คำสั่งควบคุมมีหลายชนิด เช่น if - else condition, for loop และ while loop แต่คำสั่ง for each loop จะอ่านตัวแปรแถวลำดับทุกส่วนย่อยแบบเรียงลำดับโดยไม่ต้องมีเงื่อนไข หากในคำสั่งควบคุมมีรูปแบบคำสั่ง (statement) มากกว่า 1 คำสั่ง จะต้องครอบด้วยเครื่องหมายปีกกาเปิดและปีกกาปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 2.6 คำสั่งควบคุมต่าง ๆ ในภาษา PHP

```
<?php
if ($apples > $bananas) echo "You have more apples than
bananas!"; else if ($apples == $bananas) echo "You have apples
equal to bananas!"; else echo "You have less apples than
bananas!";

$arrayAmpur = array( "Muang" , "Bangkruy" , "Bangyai" );
for ( $i = 0; $i < count($arrayAmpur); $i ++)
{
echo " $i : " . $arrayAmpur[$i] . "<BR>";
}

$arrayAmpur = array( "1201" => " Muang " , "1202" => "
Bangkruy " );
foreach ( $arrayAmpur as $kAmpur => $vAmpur)
{
echo " $kAmpur : $vAmpur <BR>";
}

$rs = mysql_query("SELECT * FROM employees");
while ( $row_rs = mysql_fetch_array($rs) )
{
echo "Employee ID: " . $row_rs['employeeid'] . "<BR>";
echo "First Name: " . $row_rs['firstname'] . "<BR>";
echo "Last Name: " . $row_rs['lastname'] . "<BR>";
}
?>
```

2.3.2.11 ฟังก์ชัน (Function)

ฟังก์ชันมีไว้สำหรับเก็บรูปแบบคำสั่งที่ใช้งานบ่อยครั้ง เพื่อให้ง่ายต่อการเรียกใช้งานและเก็บรูปแบบคำสั่งให้เป็นระเบียบ ลดจำนวนบรรทัดของโค้ด ซึ่งมีผลให้ขนาดของไฟล์ลดลงได้

โปรแกรม 2.7 การประกาศและเรียกใช้ฟังก์ชัน (Function declaration and invocation)

```
<?php
// Declaration
function check_length($data){
    return strlen($data);
}
// Invocation
echo check_length("Project") // 7
?>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ภาษาเอสคิวแอล

ภาษาเอสคิวแอล (SQL: Structured Query Language) เป็นภาษามาตรฐานที่ใช้ติดต่อกับระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ซึ่งระบบฐานข้อมูลในปัจจุบันเกือบทั้งหมดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ เช่น Oracle, Sybase, Microsoft SQL Server, Microsoft Access, DB2 เป็นต้น ถึงแม้ว่าระบบจัดการฐานข้อมูลเชิงสัมพันธ์ทุกระบบ จะสามารถใช้คำสั่งพื้นฐานของเอสคิวแอลเหมือนกัน แต่ทุกระบบอาจจะมีคำสั่งพิเศษของตัวเองที่ ระบบอื่นอาจจะใช้ไม่ได้

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นโครงสร้างฐานข้อมูลที่สามารถกำหนดความสัมพันธ์ขึ้นเมื่อใดก็ได้ โดยที่ไม่จำเป็นต้องกำหนดไว้ตั้งแต่เริ่มต้นออกแบบฐานข้อมูล และเปลี่ยนโครงสร้างได้ง่าย ฐานข้อมูลเก็บข้อมูลและสารสนเทศในออบเจกต์เรียกว่า ตาราง (Table)

ตารางประกอบด้วยสดมภ์ (Column) และแถวข้อมูล (Row) โดยสดมภ์ประกอบด้วยชื่อสดมภ์ ประเภทข้อมูล และคุณสมบัติอื่น ๆ (เช่น การกำหนดเป็นคีย์หลัก, ค่าเริ่มต้น) ส่วนแถวเก็บข้อมูลของสดมภ์

เอสคิวแอลสามารถแบ่งคำสั่งออกเป็น 4 กลุ่ม คือ

- 1) Data Manipulate (DML) เป็นคำสั่งจัดการข้อมูล ได้แก่ INSERT, UPDATE, DELETE, ROLLBACK, COMMIT
- 2) Data Definition (DDL) เป็นคำสั่งจัดการกับไฟล์ในฐานข้อมูล ได้แก่ CREATE, ALTER, DROP
- 3) Query เป็นคำสั่งการเรียกดูข้อมูล คือ SELECT
- 4) Data Control เป็นคำสั่งจัดการความปลอดภัย

2.4.1 เพิ่มข้อมูล (INSERT INTO)

เพิ่มระเบียนหนึ่ง หรือหลายระเบียนลงในตาราง

โปรแกรม 2.8 ไวยากรณ์ของการเพิ่มข้อมูล

```
INSERT INTO target [(field1[, field2[, ...]])]
VALUES (value1[, value2[, ...]])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ฟิลด์ต่าง ๆ ของการเพิ่มข้อมูล

ส่วน	คำอธิบาย
target	ชื่อของตารางหรือแบบสอบถามที่ต้องการผนวกข้อมูลเข้า
field1, field2	ชื่อของเขตข้อมูลที่ต้องการผนวกข้อมูลเข้าถ้าตามหลังอาร์กิวเมนต์ target หรือชื่อของเขตข้อมูลที่ต้องการจะนำข้อมูลมาถ้าตามหลังอาร์กิวเมนต์ source
source	ชื่อของตารางหรือแบบสอบถามที่ต้องการคัดลอกระเบียบมา
value1, value2	ค่าที่ต้องการจะแทรกลงในเขตข้อมูลที่ระบุของระเบียบใหม่ แต่ละค่าจะถูกแทรกลงในเขตข้อมูลที่ตรงกับตำแหน่งของค่าในรายการ เช่น value1 จะถูกแทรกลงใน field1 ของระเบียบใหม่ และ value2 จะถูกแทรกลงใน field2 และเป็นเช่นนี้เรื่อย ๆ คุณจะต้องคั่นระหว่างค่าต่าง ๆ ด้วยเครื่องหมายจุลภาค (.) และใส่ขอบเขตข้อมูลข้อความไว้ในเครื่องหมายอัญประกาศ ('')

2.4.2 เลือกข้อมูล (SELECT)

เป็นคำสั่งที่ใช้สำหรับการเรียกดูข้อมูลในตาราง คำสั่งเอสคิวแอล SELECT สามารถเรียกได้ทั้งตาราง หรือสามารถระบุฟิลด์ที่ต้องการเรียกดูข้อมูลได้

โปรแกรม 2.10 ไวยากรณ์ของการเลือกข้อมูล

```
SELECT Column1, Column2, Column3,... FROM [Table-Name]
```

2.4.3 การระบุเงื่อนไข (WHERE)

เป็นคำสั่งที่ใช้สำหรับการระบุเงื่อนไขการเลือกข้อมูลในตาราง คำสั่งเอสคิวแอล WHERE สามารถระบุเงื่อนไขในการเลือกข้อมูลได้ตั้งแต่หนึ่งเงื่อนไขเป็นต้นไป

โปรแกรม 2.11 ไวยากรณ์ของการระบุเงื่อนไข

```
SELECT Column1, Column2, Column3,... FROM Table-Name
WHERE [Field] = 'Value' ;
```

2.4.4 เรียงลำดับข้อมูล (ORDER BY)

คำสั่งเอสคิวแอล ORDER BY เป็นคำสั่งที่ใช้สำหรับการระบุเงื่อนไขในการเลือกข้อมูลในตาราง โดยจะจัดเรียงข้อมูลตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาด้านใน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 2.12 ไวยากรณ์ของการเรียงลำดับข้อมูล

```
SELECT Column1,Column2,Culumn3,... FROM [Table-Name] ORDER BY
[Field] [ASC/DESC],[Field]
[ASC/DESC],...
```

2.4.5 ปรับปรุงข้อมูล (UPDATE)

คำสั่งเอสคิวแอล UPDATE เป็นคำสั่งที่ใช้สำหรับแก้ไขข้อมูลในตาราง โดยสามารถทำการแก้ไขได้หลายฟิลด์และหลายระเบียน ภายในคำสั่งหนึ่งคำสั่ง ทั้งนี้ขึ้นอยู่กับเงื่อนไขที่ผู้ใช้ได้เขียนขึ้น

โปรแกรม 2.13 ไวยากรณ์ของการปรับปรุงข้อมูล

```
UPDATE table SET newvalue WHERE criteria;
```

ตารางที่ 2.5 ฟิลด์ต่าง ๆ ของการปรับปรุงข้อมูล

ส่วน	คำอธิบาย
table	ชื่อของตารางที่มีข้อมูลที่คุณต้องการจะปรับเปลี่ยน
newvalue	นิพจน์ที่ระบุค่าที่จะถูกแทรกลงในบางเขตข้อมูลในระเบียนที่มีการปรับปรุง
criteria	นิพจน์ที่กำหนดระเบียนที่จะถูกปรับปรุง ระเบียนที่เป็นไปตามนิพจน์เท่านั้นที่จะถูกปรับปรุง

2.4.6 ลบข้อมูล (DELETE)

คำสั่งเอสคิวแอล DELETE เป็นคำสั่งที่ใช้สำหรับลบข้อมูลในตาราง โดยสามารถทำการลบได้หลายระเบียนภายในคำสั่งเดียว ทั้งนี้ขึ้นอยู่กับเงื่อนไขที่ผู้ใช้เขียนขึ้นด้วย

โปรแกรม 2.14 ไวยากรณ์ของการลบข้อมูล

```
DELETE [table.*] FROM table
WHERE criteria
```

ตารางที่ 2.6 ฟิลด์ต่าง ๆ ของการลบข้อมูล

ส่วน	คำอธิบาย
table	ชื่อของตารางที่ระเบียนจะถูกลบออกซึ่งไม่จำเป็นต้องมีก็ได้
table	ชื่อของตารางที่ระเบียนจะถูกลบออก
criteria	นิพจน์ ที่กำหนดว่าระเบียนใดบ้างที่จะถูกลบ

2.5 โครงร่างพีเอชพี (PHP Framework)

2.5.1 โครงร่าง (Framework)

โครงร่าง คือ ชุดของไลบรารี (Libraries) หรือคลาส (Classes) สำหรับใช้ในการพัฒนาโปรแกรม โดยมีโครงสร้างของโปรแกรมที่เป็นมาตรฐาน โดยที่ภายใน โครงร่างนั้น จะมีโค้ดที่นำกลับมาใช้ใหม่ได้อยู่เป็นจำนวนมาก ซึ่งจะทำให้ประหยัดเวลาในการทำแอปพลิเคชันหนึ่ง ๆ ได้อย่างมาก เนื่องจากไม่จำเป็นต้องเขียนโค้ดเดิมซ้ำ ๆ ทุกครั้งที่พัฒนาแอปพลิเคชันใหม่ หรือแอปพลิเคชันที่ทำงานคล้าย ๆ กัน จึงควรเก็บโค้ดซ้ำ ๆ ไว้เป็นฟังก์ชันในไฟล์ย่อย ๆ แล้วใช้วิธีการรวม (include) ไฟล์เข้ามาใช้งาน ซึ่งเป็นหลักการของโครงร่าง

2.5.2 โค้ดอิกไนเตอร์ (CodeIgniter)

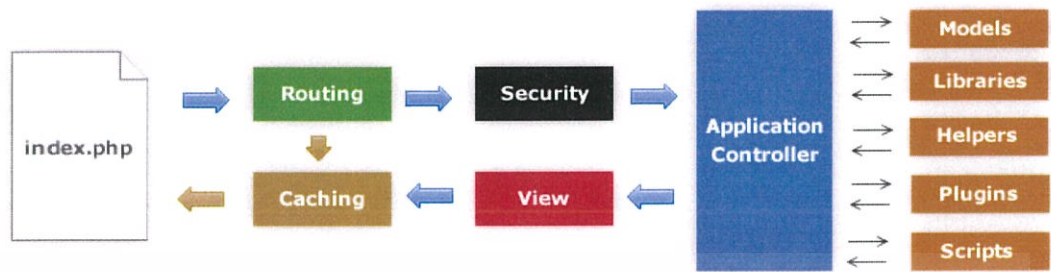
เป็นโครงร่างพีเอชพี ซึ่งพัฒนาขึ้นโดย Rick Ellis ใช้รูปแบบการเขียนโปรแกรมแบบเอ็มซีวี (MCV: Model-View-Controller) ซึ่งจะแยกเอาส่วนของการแสดงผลออกจากส่วนประมวลผล ทำให้โค้ดที่ได้มีความเป็นระเบียบเป็นระบบและง่ายต่อการบำรุงรักษา เนื่องจากตัวโครงร่าง เองนั้นมาพร้อมกับฟังก์ชัน หรือ คลาสที่อยู่ในรูปของไลบรารีช่วยเหลือ (library helper) หรือส่วนเสริม (plugin) ซึ่งสิ่งเหล่านี้มีส่วนช่วยให้นักพัฒนาใช้เวลาในการพัฒนาระบบลดลงสามารถศึกษาทำความเข้าใจได้ในระยะเวลาสั้น ๆ

โค้ดอิกไนเตอร์ถูกสร้างบนพื้นฐานการพัฒนาแบบเอ็มซีวี ซึ่งเป็นวิธีการทางซอฟต์แวร์ที่แยกระบบการประมวลผลของแอปพลิเคชันออกจากส่วนแสดงผล ในทางปฏิบัติ มันอนุญาตให้หน้าเว็บมีสคริปต์เล็ก ๆ เนื่องจากส่วนแสดงต้องแยกออกจากสคริปต์พีเอชพี

สามารถอธิบายรูปแบบการเขียนโปรแกรมแบบเอ็มซีวีได้ดังต่อไปนี้

- 1) โมเดล (Model) อธิบายโครงสร้างข้อมูล แบบฉบับของคลาสแบบจำลองจะมีฟังก์ชันที่ช่วยให้สามารถ รับ, ใส่, ปรับปรุงข้อมูล ในฐานข้อมูล
- 2) วิว (View) คือข้อมูลที่ถูกแสดงให้กับผู้ใช้ ส่วนใหญ่จะเป็นเว็บเพจ แต่ในโค้ดอิกไนเตอร์, วิวสามารถเป็นเพียงบางส่วนของหน้าได้ เช่น ส่วนหัว ส่วนท้ายได้
- 3) คอลโทรลเลอร์ (Controller) ถูกใช้เป็น สื่อกกลาง ระหว่างโมเดล, วิว และแหล่งที่มาอื่น ๆ ที่ต้องการในการทำงานร้องขอเอชทีทีพี (HTTP request) และการสร้างเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 การไหลของข้อมูลในสถาปัตยกรรมแบบเอ็มซีวี

สามารถอธิบายขั้นตอนการไหลของข้อมูลของสถาปัตยกรรมแบบเอ็มซีวีได้ดังนี้

- 1) หน้า index.php เป็นตัวควบคุมส่วนหน้า, สร้างทรัพยากรพื้นฐานที่ต้องการในการรันโค้ดอีกในเตอร์
- 2) ส่วน Routing ทำการตรวจสอบการร้องขอเอชทีทีพีที่กำหนดว่าจะทำอะไร
- 3) ถ้ามีไฟล์แคช (cache) อยู่ Caching จะส่งข้อมูลกลับไปยังเบราว์เซอร์ทันที
- 4) ก่อนที่จะไหลตัวควบคุมของแอปพลิเคชัน (Application Controller) การร้องขอเอชทีทีพีและผู้ใช้ใด ๆ ที่ส่งข้อมูลมาจะถูกกรองข้อมูลที่ Security เพื่อความปลอดภัย
- 5) โหลดแบบจำลอง (Models), ไลบรารีหลัก (Libraries), ส่วนเสริม (Plugins), ผู้ช่วย (Helpers) และทรัพยากรอื่น ๆ ที่จำเป็นในการทำงานที่ถูกร้องขอมาผ่านคอนโทรลเลอร์
- 6) สุดท้าย View ปฏิบัติงานและถูกส่งกลับไปยังเบราว์เซอร์เพื่อโชว์หน้าจอ ถ้าระบบแคชถูกใช้งาน หน้าจอจะถูกแคชก่อนแล้วจึงค่อยส่งสิ่งที่ร้องขอมาเป็นลำดับถัดไป

2.5.3 ซีไอไซด์ (CI site)

ภายในโฟลเดอร์ application จะมีโฟลเดอร์ย่อยที่ทำหน้าที่ต่างกันอีกหลายโฟลเดอร์ ในหัวข้อก่อนหน้านี้ เราได้ทำการศึกษาหน้าที่การทำงานของโฟลเดอร์ controller, config และ views กันแล้ว สำหรับในหัวข้อนี้จะเป็นการศึกษาชนิดของไฟล์ หรือคลาส และหน้าที่การทำงานของโฟลเดอร์อื่น ๆ ซีไอไซด์ในทางเทคนิคแล้ว เราจะสามารถเขียนโค้ดพีเอชพีแล้วบันทึกไว้ในโฟลเดอร์ใดก็ได้ภายในซีไอไซด์ แต่โค้ดอีกในเตอร์ได้ทำการสร้างโครงสร้างโฟลเดอร์ไว้เพื่อทำหน้าที่ที่แตกต่างกันเพื่อความสะดวกในการพัฒนา ดูแลรักษา และการปรับปรุงซีไอไซด์

ตัวอย่างที่จะกล่าวต่อไป นี้ จะเป็นการเรียกใช้งานส่วนของโค้ดที่เรียกว่า “display” โดยอาจจะเรียกใช้งานผ่านฟังก์ชัน “mainpage” โดยมี 4 วิธีการที่แตกต่างกันแต่ให้ผลลัพธ์เหมือนกัน ได้แก่การสร้างไฟล์ของโค้ดอีกในเตอร์เป็นประเภทต่าง ๆ ดังนี้ ได้แก่ model, library, helper และ plug-in

ตารางที่ 2.7 ประเภทไฟล์และวิธีการใช้ไฟล์ต่าง ๆ ในซีไอไซด์

ประเภทไฟล์	วิธีการใช้
model	<p>เป็นคลาสไฟล์ (ตามหลักการเขียนโปรแกรมแบบเชิงวัตถุ)</p> <ul style="list-style-type: none"> • การ Load : <code>\$this->load->model('display');</code> • การใช้ : <code>\$this->display->mainpage();</code> • คลาสไฟล์นี้ต้องเป็นคลาสที่สืบทอดมาจาก model (โดยเริ่มต้นด้วย class Display extends Model) • คลาสจะต้องมี constructor ที่มีส่วนของโค้ดดังนี้ <pre>function display() { parent::Model(); }</pre> <p>และมีฟังก์ชันที่ชื่อ mainpage()</p> <pre>Function mainpage(){}</pre> <ul style="list-style-type: none"> • โดยหลักการ Model เป็นคลาสที่ถูกออกแบบมาเพื่อทำงานกับฐานข้อมูล
Library	<ul style="list-style-type: none"> • Library มี 2 โพลเดอร์คือ ในโพลเดอร์ system/libraries และ application/libraries • การ Load : <code>\$this->load->library('display');</code> • การใช้ : <code>\$this->display->mainpage();</code> • ไฟล์ library เป็นคลาสไฟล์ (ไม่ต้องสืบทอดคลาส Model และ ไม่จำเป็นต้องมี constructor) <pre>class Display() { function mainpage(){ // code here } }</pre> <ul style="list-style-type: none"> • libraries จะเป็นที่สำคัญสำหรับเก็บโค้ดที่เราเขียนขึ้นเพื่อทำงานเฉพาะด้าน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้เอาไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 (ต่อ)

plug-in	<p>plug-in มี 2 โพลเดอร์คือ ในโพลเดอร์ system/helper และ application/helper ไฟล์ plug-in จะเป็นแบบ procedural ไม่ใช่แบบ OOP</p> <ul style="list-style-type: none"> • การ Load : <code>\$this->load->plugin('display');</code> • การใช้โดยเรียกฟังก์ชันโดยตรง: <code>mainpage();</code> • ชื่อไฟล์ต้องเพิ่ม <code>_pi.php</code> คือ <code>display_pi.php</code> • <code>mainpage()</code> ต้องเป็นฟังก์ชัน ไม่ใช่คลาส • ข้อแตกต่างระหว่าง helper และ plug-in คือ helper เป็นที่รวบรวม ฟังก์ชันหลาย ๆ ฟังก์ชัน แต่ plug-in เป็นที่เก็บฟังก์ชันเดียวต่อหนึ่งไฟล์
---------	---

ไฟล์ต่าง ๆ ที่กล่าวถึงข้างต้นจะต้องไม่สามารถเรียกใช้งานได้โดยตรง ข้อสังเกตคือ helpers, plug-in และ libraries มี 2 ชุดเพื่อประโยชน์ดังนี้

- 1) ในโพลเดอร์ system เป็นส่วนหนึ่งของโค้ดซีโคโนเตอร์ (CI Core) จะถูกแชร์ให้ application ทั้งหมด สามารถทำการอัปเดตโค้ดอีกในเตอร์เป็นเวอร์ชันใหม่โดยการเขียนทับ (overwrite) โพลเดอร์ system
- 2) ในโพลเดอร์ application จะใช้งานได้เฉพาะ application นั้น ๆ เท่านั้น ถ้าต้องการอัปเดตเวอร์ชันของโค้ดอีกในเตอร์เป็นเวอร์ชันใหม่ โพลเดอร์ application จะไม่ถูกเขียนทับ
- 3) เมื่อเราทำการโหลด helper, plug-in หรือ library ของโค้ดอีกในเตอร์จะไปค้นหาไฟล์จากทั้งสองที่โดยอัตโนมัติ โดยทำการค้นหาไฟล์ในโพลเดอร์ system ก่อน ถ้าไม่เจอไฟล์ก็จะไปหาในโพลเดอร์ application เป็นลำดับถัดไป
- 4) จากข้อ 3 ทำให้สามารถ Overwrite Core, library, helpers และ plug-in ได้โดยการสร้างไฟล์ที่มีชื่อไฟล์เหมือนกัน แต่เก็บในโพลเดอร์ application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 โฟลเดอร์ต่าง ๆ ในซีไอไซด์

โฟลเดอร์	คำอธิบาย
Config	เก็บข้อมูลพื้นฐานเกี่ยวกับซีไอไซด์
Controllers	ส่วนควบคุม
Errors	เก็บไฟล์แจ้งเตือนข้อผิดพลาด(Error) ที่เกิดขึ้น
Hooks	ควบคุมการโหลดไฟล์
Libraries	เก็บไฟล์ไลบรารี
Models	เก็บไฟล์ส่วนของการทำงาน
Views	เก็บไฟล์ส่วนการแสดงผล
Cache	ใช้เก็บไฟล์แคช
Codeigniter	เก็บไฟล์ระบบ
Database	ไฟล์ไลบรารีสำหรับการทำงานกับฐานข้อมูล
Fonts	เก็บฟอนต์
Helpers	ไฟล์ผู้ช่วยระบบ
Plugins	เก็บไฟล์ส่วนเสริม
Scaffolding	ไลบรารีที่ช่วยในการจัดการกับฐานข้อมูล

โค้ดอิกไนเตอร์เป็นโครงร่างพีเอชพีที่มีขนาดเล็ก ใช้งานง่าย มีรูปแบบการเขียน โปรแกรมแบบเอ็มซีวี ซึ่งจะแยกเอาส่วนของการแสดงผลออกจากส่วนประมวลผล ทำให้โค้ดที่ได้ออกมา มีความเป็นระเบียบ เป็นระบบและง่ายต่อการพัฒนา ช่วยให้ผู้ใช้ เรียนรู้สามารถศึกษาทำความเข้าใจได้ในระยะเวลาอันสั้น เพราะมีคู่มือช่วยในการศึกษา ไม่ว่าจะเป็นการติดตั้ง การแนะนำ การใช้งาน รูปแบบการเขียนโค้ดต่าง ๆ เป็นต้น และ เนื่องจากตัวโครงร่างมาพร้อมกับฟังก์ชัน หรือ คลาสที่อยู่ในรูปของ ไลบรารีผู้ช่วย (library helper) หรือส่วนเสริม (plugin) จึงมีส่วนช่วยในการเขียนและพัฒนาโปรแกรมโดยใช้เวลาในการพัฒนาระบบลดลง

2.6 จาวาสคริปต์ (JavaScript)

จาวาสคริปต์นั้นเป็นภาษาสคริปต์เชิงวัตถุ (Object-Oriented) ภาษาหนึ่งสำหรับการเขียนโปรแกรม เพื่อนำเสนอความสามารถ รวมถึงลูกเล่นของเว็บไซต์ผ่านทางเว็บ โดยสามารถเพิ่มคำสั่งจาวาสคริปต์ไปได้เพื่อใช้ในประโยชน์ในด้านต่าง ๆ สำหรับจาวาสคริปต์นี้ใช้สำหรับการสร้างเว็บไซต์เพื่อให้มีลักษณะพลวัต กล่าวคือ สามารถตอบโต้กับผู้ใช้ได้ทันที อีกทั้งยังต้องมีความเกี่ยวข้องกับไฟล์

ที่เป็นเอชทีเอ็มแอลอีกด้วย ดังนั้นเพื่อให้การเขียนจาวาสคริปต์เป็นไปได้ด้วยดี ผู้เขียนควรที่จะมีความรู้เกี่ยวกับภาษาเอชทีเอ็มแอลด้วย

2.6.1 ลักษณะการทำงานของจาวาสคริปต์

1) นาวิเกเตอร์จาวาสคริปต์ (Navigator JavaScript) เป็นจาวาสคริปต์ฝั่งลูกข่าย (Client-Side JavaScript) ซึ่งหมายถึงจาวาสคริปต์ที่ถูกแปลทางฝั่งไคลเอนต์ จึงมีความเหมาะสมต่อการใช้งานของผู้ใช้ทั่วไปเป็นส่วนใหญ่

2) โลฟไวร์จาวาสคริปต์ (LiveWire JavaScript) เป็นจาวาสคริปต์ฝั่งแม่ข่าย (Server-Side JavaScript) ซึ่งหมายถึงจาวาสคริปต์ที่ถูกแปลทางฝั่งเซิร์ฟเวอร์ สามารถใช้ได้เฉพาะกับโลฟไวร์ของเน็ตสเคปโดยตรง

2.6.2 รูปแบบของจาวาสคริปต์

การเขียนจาวาสคริปต์เป็นการเขียนโปรแกรมเสริมเข้าไปในโปรแกรมที่เป็นโครงสร้างหลักของเว็บเพจที่เขียนด้วยเอชทีเอ็มแอล เพื่อช่วยให้การทำงานมีความสามารถมากขึ้นจากที่เคยทำได้ด้วยเอชทีเอ็มแอล การใช้จาวาสคริปต์จะช่วยทำให้การทำเว็บเพจหรือโปรแกรมเชิงวัตถุมีความสามารถมากขึ้น

2.6.3 ความสามารถของจาวาสคริปต์

1) จาวาสคริปต์ทำให้สามารถเขียนโปรแกรมแบบง่าย ๆ ได้โดยไม่ต้องพึ่งภาษาอื่น เช่น พีเอชพี โดยส่วนใหญ่จะเป็นรูปแบบของการแสดงผล

2) จาวาสคริปต์มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่น เมื่อผู้ใช้คลิกที่ปุ่ม สามารถสั่งให้เปิดหน้าต่างใหม่ได้ ทำให้เว็บไซต์มีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น

3) จาวาสคริปต์สามารถเขียนหรือเปลี่ยนแปลงองค์ประกอบของเอชทีเอ็มแอล (HTML Element) ได้ สามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้

4) จาวาสคริปต์มีความสามารถในการตรวจสอบ เปรียบเทียบ ตัดสินใจ ประมวลผล และสามารถสร้างฟังก์ชันเองได้

5) จาวาสคริปต์สามารถใช้ในการตรวจสอบผู้ใช้ได้ เช่น ตรวจสอบเว็บเบราว์เซอร์ของผู้ใช้

6) จาวาสคริปต์มีลักษณะเป็นอินเทอร์พรีเตอร์ (Interpreter) แบบไฟล์ข้อความฝังอยู่ในเอกสารเอชทีเอ็มแอล ทำให้สามารถทำงานบนเว็บเบราว์เซอร์ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศีกษาเท่านั้น ในอนาคตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4 ข้อจำกัดของจาวาสคริปต์

- 1) ไม่สามารถติดต่อ หรือทำงานบนฝั่งของเครื่องแม่ข่ายได้
- 2) ไม่สามารถสร้างส่วนต่อประสาน (Interface) ของเว็บแอปพลิเคชันได้โดยตรง ต้องใช้งานร่วมกับโปรแกรมภาษาอื่น

2.7 เอแจ็กซ์ (AJAX)

เอแจ็กซ์ (AJAX: Asynchronous JavaScript and XML) ไม่ใช่ชื่อของการเขียนโปรแกรม หรือเป็นชื่อของภาษาที่ใช้ในการโปรแกรม แต่เป็นชุดของเทคโนโลยีต่าง ๆ มาจากการทำงานร่วมกันของจาวาสคริปต์ (JavaScript) และเอ็็กเอ็มแอล (XML) แบบไม่เข้าจังหวะ (Asynchronous) มีหลักการทำงาน 2 ประเด็นคือการอัปเดต หน้าจอแบบบางส่วนและการติดต่อสื่อสารกับเซิร์ฟเวอร์ โดยใช้หลักการไม่เข้าจังหวะ ทำให้ผู้ใช้ไม่ต้องหยุดการทำงานเพื่อรอการประมวลผลจากเซิร์ฟเวอร์ รวมถึงการโหลดและการรีเฟรชหน้าจอ ของเบราว์เซอร์ทางฝั่งไคลเอนต์มีการใช้เอแจ็กซ์ โดยการเพิ่มเลย์เออร์ระหว่างเบราว์เซอร์ผู้ใช้กับเซิร์ฟเวอร์ทำให้ผู้ใช้สามารถทำงานได้โดยไม่ต้องรอให้ไคลเอนต์ติดต่อไปยังเซิร์ฟเวอร์รวมถึงการโหลดและการรีเฟรชหน้าจอทั้งหมดด้วย ดังนั้นผู้ใช้สามารถใช้งานแอปพลิเคชันได้อย่างมีประสิทธิภาพมากขึ้น

เอแจ็กซ์เอนจิน (AJAX engine) ทำหน้าที่เป็นตัวกลางระหว่างไคลเอนต์และเซิร์ฟเวอร์ขณะนั้น เมื่อไคลเอนต์มีต้องการร้องขอ แทนที่จะการร้องขอส่งเอชทีทีพี (HTTP request) ไปยังเซิร์ฟเวอร์โดยตรงไคลเอนต์จะส่งจาวาสคริปต์คอล (JavaScript call) ไปเอแจ็กซ์เอนจิน เพื่อโหลดข้อมูลผู้ใช้ต้องการ และหากเอแจ็กซ์เอนจินต้องการข้อมูลเพิ่มเติมในการตอบสนองต่อผู้ใช้ เอแจ็กซ์เอนจินจะส่งการร้องขอไปยังเซิร์ฟเวอร์โดยใช้เอ็็กเอ็มแอล โดยส่วนประกอบจำเป็นขั้นพื้นฐานที่ขาดไม่ได้ในเอแจ็กซ์ ได้แก่ เอชทีเอ็มแอล, เอ็็กเอ็มแอล, ดีโอเอ็ม (DOM) และจาวาสคริปต์

2.7.1 ที่มาของปัญหา

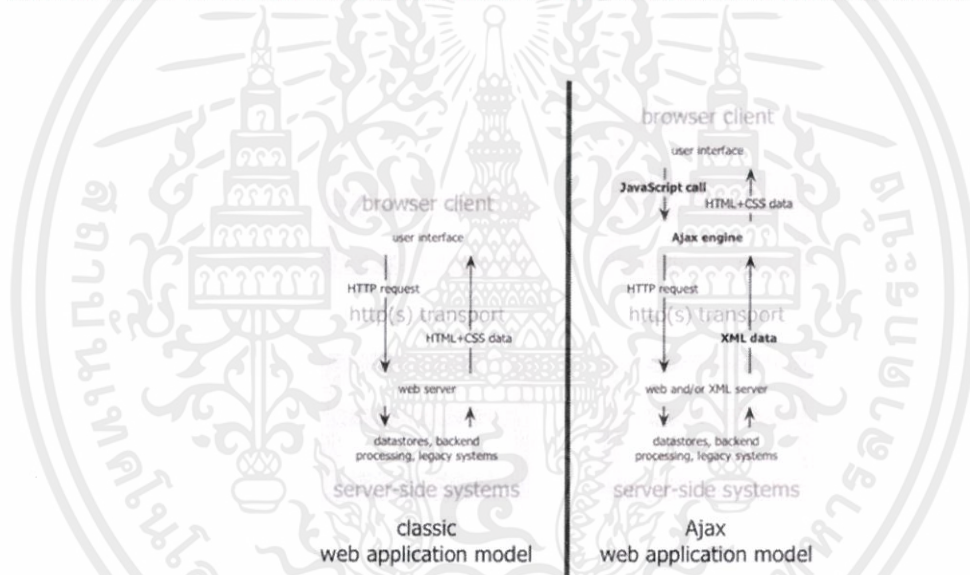
เนื่องจากแอปพลิเคชันที่ใช้งานในปัจจุบันนี้ มีหลักการที่ทำงานแล้วเกิดการสูญเสียเวลาและทรัพยากรของผู้ใช้ในการรอคอยการทำงานต่าง ๆ ทำให้ผู้ใช้ต้องหยุดคอย ดังนั้นการทำงานของใช้จึงเป็นไปอย่างไม่ต่อเนื่อง ซึ่งหลักการดังกล่าวคือ

- 1) User interaction paradigm การที่เบราว์เซอร์ตอบสนองต่อการทำงานของผู้ใช้ โดยจะทิ้งหน้าเว็บที่แสดงอยู่ในขณะนั้น แล้วไปทำการส่งการร้องขอส่งเอชทีทีพีกลับไปยังเซิร์ฟเวอร์แทน ซึ่งทำให้ผู้ใช้ไม่สามารถทำอะไรได้ นอกจากการรอคอย เมื่อเซิร์ฟเวอร์ทำการประมวลผลเสร็จก็จะส่งหน้าเอชทีเอ็มแอลกลับมายังเบราว์เซอร์ ต่อจากนั้นเบราว์เซอร์ก็จะรีเฟรชและแสดงหน้าเอชทีเอ็มแอล

หน้าใหม่ และเหตุนี้เองที่ทำให้ผู้ใช้สามารถใช้งานต่อไปได้ จะเห็นว่า ผู้ใช้มีช่วงเวลาของการหยุดรอคอยเป็นเวลานานสำหรับการประมวลผลของเซิร์ฟเวอร์และการรีเฟรชหน้าเอชทีเอ็มแอลใหม่ทั้งหน้า ซึ่งเป็นสิ่งที่ไม่ดีประสิทธิภาพในเชิงพลวัต ของการทำงานบนเว็บแอปพลิเคชัน

2) Synchronous request and response communication mode การที่เบราว์เซอร์เริ่มทำการร้องขอข้อมูล และเซิร์ฟเวอร์ก็ตอบสนองเฉพาะการร้องขอที่เบราว์เซอร์ร้องขอมา เซิร์ฟเวอร์จะไม่สามารถส่งข้อมูลได้ถ้าเบราว์เซอร์ไม่ได้ร้องขอข้อมูลในขณะนั้น ซึ่งถือว่าเป็นการติดต่อสื่อสารเป็นแบบทิศทางเดียว

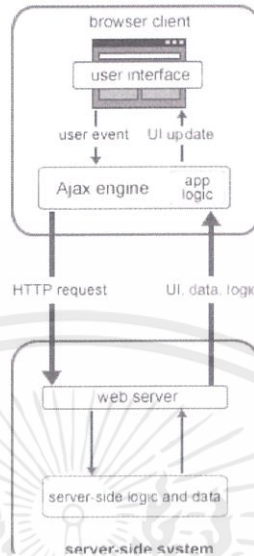
วงจรการร้องขอและตอบสนองแบบเข้าจังหวะ (Synchronous) คือ การทำงานแบบประสานจังหวะระหว่างเบราว์เซอร์กับเซิร์ฟเวอร์ทำให้เกิดความล่าช้าในการทำงานทำให้ผู้ใช้ทำอะไรไม่ได้อีก นอกจากการคอยการตอบสนองกลับมาจากเซิร์ฟเวอร์เมื่อเซิร์ฟเวอร์ประมวลผลเสร็จ



รูปที่ 2.11 การเปรียบเทียบรูปแบบเว็บแอปพลิเคชันแบบปกติและเอแจ็กซ์

2.7.2 โครงสร้างของเอแจ็กซ์

มุมมองของโครงสร้างทางซอฟต์แวร์ของเอแจ็กซ์ต่างจากเว็บแอปพลิเคชันในทุกวันนี้ เนื่องจากมีการเพิ่มเอนจินทางฝั่งไคลเอนต์ เอแจ็กซ์เอนจินนี้อยู่ส่วนติดต่อผู้ใช้ (User interface) กับเซิร์ฟเวอร์ซึ่งจะมองว่าเป็นการทำงานที่ไคลเอนต์การทำงานต่าง ๆ ของผู้ใช้ โปรแกรมจะไปเรียกเอแจ็กซ์เอนจินตัวนี้ขึ้นมา แทนที่การร้องขอหน้าเว็บจากเซิร์ฟเวอร์โดยตรง และจะใช้โครงสร้างข้อมูลแบบเอ็เอ็มแอลในการขนย้ายข้อมูลระหว่างเซิร์ฟเวอร์กับเอแจ็กซ์เอนจิน เมื่อเบราว์เซอร์ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์ นอกจากนี้เอแจ็กซ์เอนจินไม่ต้องทำการติดตั้ง ไม่ใช่ส่วนเสริม (plugin) และไม่สามารถดาวน์โหลด (download) ได้ เพราะเอแจ็กซ์ เป็นแนวคิดในการแก้ปัญหาการหยุดชะงักการทำงานของผู้ใช้



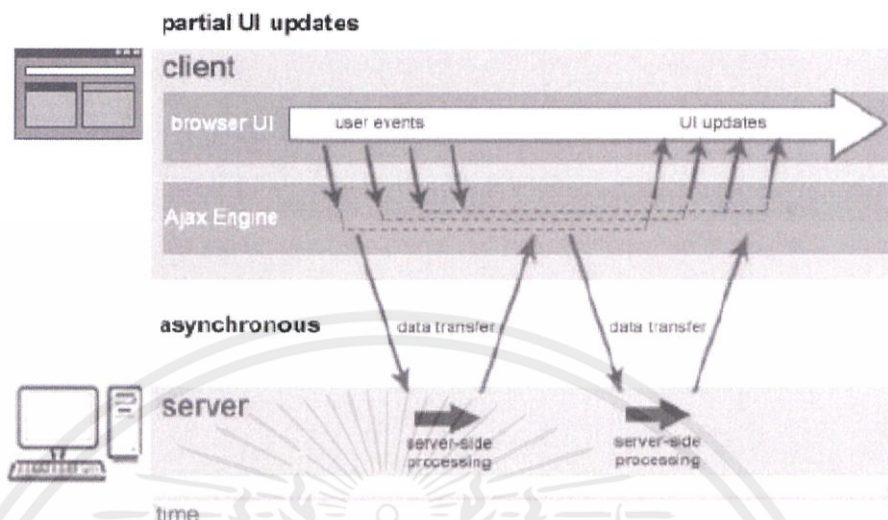
รูปที่ 2.12 สถาปัตยกรรมของเอแจ็กซ์

2.7.3 การทำงานของเอแจ็กซ์

เอแจ็กซ์จะช่วยลดการติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยในการโหลดหน้าเว็บนั้น เบราวเซอร์จะโหลดข้อมูลเอแจ็กซ์เอนจิน แทนการร้องขอข้อมูลจากเซิร์ฟเวอร์โดยตรง นอกจากเอแจ็กซ์จะทำหน้าที่ทั้งการแสดงผลส่วนติดต่อกับผู้ใช้และติดต่อไปยังเซิร์ฟเวอร์แล้ว เอแจ็กซ์เอนจินอนุญาตให้การกระทำต่าง ๆ ในเว็บแอปพลิเคชันเป็นแบบไม่เข้าจังหวะ คือความเป็นอิสระในการติดต่อไปยังเซิร์ฟเวอร์นั่นเอง

การอัปเดตหน้าจอบางส่วน แทนที่การ "click, wait, and refresh" ระหว่างที่เกิด การทำงานแบบการติดต่อสื่อสารของผู้ใช้ ส่วนติดต่อผู้ใช้ที่ต้องนำมาแสดงซ้ำในหน้าเว็บที่ร้องขอไปยังเซิร์ฟเวอร์จะถูกจัดเป็นข้อมูลใหม่เมื่อถูกอัปเดตแล้ว การหยุดชะงักส่วนติดต่อผู้ใช้จึงไม่เกิดขึ้น เพราะหน้าเว็บนั้นยังคงถูกแสดงอยู่และสามารถใช้งานได้โดยปราศจากการหยุดชะงักการทำงานของผู้ใช้ การอัปเดตหน้าเว็บบางส่วนสามารถทำให้หน้าเว็บทำงานต่อไปได้

การติดต่อแบบไม่เข้าจังหวะ เข้ามาแทนที่การ "synchronous request and response model" สำหรับเอแจ็กซ์ การร้องขอและตอบสนองจะทำแบบไม่เข้าจังหวะ ซึ่งคือการติดต่อสื่อสารกับเซิร์ฟเวอร์แบบอิสระโดยทำการลดการติดต่อระหว่างเบราว์เซอร์ กับเซิร์ฟเวอร์ผลที่ได้ก็คือผู้ใช้สามารถใช้งานเว็บแอปพลิเคชันได้ในขณะที่ไคลเอนต์ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์อยู่เบื้องหลัง (การทำงานแบบพร้อมกันแต่มองเป็น 2 ฝั่ง เช่นหน้าร้านกับหลังร้าน) เมื่อข้อมูลเดินทางมาถึงเบราว์เซอร์ ก็จะอัปเดตส่วนติดต่อผู้ใช้ที่ต้องการข้อมูลใหม่ ส่วนติดต่อผู้ใช้ที่ไม่ต้องการอัปเดตก็จะแสดงส่วนนั้นต่อไป



รูปที่ 2.13 การปรับปรุงส่วนติดต่อผู้ใช้บางส่วนและการสื่อสารแบบไม่เข้าจังหวะ

2.7.4 ข้อดีของเอแจ็กซ์

- 1) ตอบสนองต่อผู้ใช้ได้อย่างรวดเร็วเนื่องจากการอัปเดตแบบบางส่วน
- 2) ผู้ใช้ไม่ต้องหยุดรอคอยการประมวลผลของเซิร์ฟเวอร์เนื่องจากการติดต่อแบบ

Asynchronous

- 3) รองรับกับเว็บเบราว์เซอร์หลัก ๆ ที่สามารถใช้จาวาสคริปต์ได้
- 4) ทำให้การประมวลผลที่เซิร์ฟเวอร์มีความเร็วขึ้นเนื่องจากการประมวลผลที่เซิร์ฟเวอร์

ลดลง

- 5) ไม่ต้องทำการติดตั้ง หรือใช้ส่วนเสริม
- 6) ไม่ยึดติดกับแพลตฟอร์ม (Platform) หรือภาษาที่ใช้ในการเขียนโปรแกรม
- 7) เป็นเทคโนโลยีใหม่ที่ไม่ได้เป็นของนักพัฒนาเว็บแอปพลิเคชันคนใด กล่าวคือทุกคนมีสิทธิ์

เข้ามาพัฒนาแอปพลิเคชันตัวนี้

2.8 คอมไพเลอร์ (Compiler)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

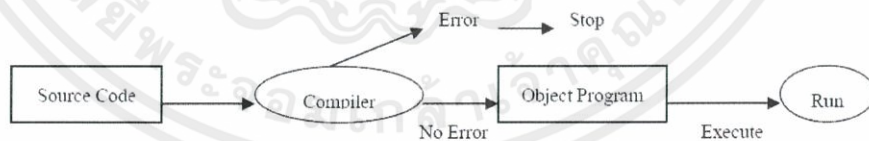
2.8.1 ความหมายของคอมไพเลอร์

คอมไพเลอร์ เป็นโปรแกรมแปลภาษาที่ทำหน้าที่แปลภาษาระดับสูงมาเป็นภาษาเป้าหมาย (Object Program) ซึ่งอาจหมายถึง ภาษาเครื่อง หรือภาษาแอสเซมบลี หากเกิดข้อผิดพลาดเกิดขึ้น คอมไพเลอร์จะหยุดทำการการแปลเพื่อให้ผู้ใช้งานทำการแก้ไขข้อผิดพลาด แต่ถ้าไม่มีข้อผิดพลาดก็จะทำงานต่อไปจนจบ และเกิดภาษาเป้าหมายขึ้น เพื่อสื่อสารกับเครื่องคอมพิวเตอร์ต่อไป

ในการพัฒนาซอฟต์แวร์คอมพิวเตอร์นั้น โปรแกรมเมอร์ (Programmer) จะเขียนโปรแกรมในภาษาคอมพิวเตอร์แบบต่าง ๆ ตามความชำนาญของแต่ละคน โปรแกรมที่ได้จะเรียกว่า โปรแกรมต้นฉบับ (Source Code) ซึ่งมนุษย์จะอ่านโปรแกรมต้นฉบับนี้ได้ แต่เครื่องคอมพิวเตอร์จะไม่สามารถเข้าใจคำสั่งเหล่านั้น เนื่องจากคอมพิวเตอร์เข้าใจแต่ภาษาเครื่อง (Machine Language) ซึ่งมีลักษณะเป็นชุดของบิตที่สร้างขึ้นจากระบบเลขฐานสอง (เลข 0 และ 1 เท่านั้น) ที่หมายถึงสถานะของไฟฟ้าที่มีสองสถานะ คือ เปิดและปิด นั้นทำให้การเขียนโปรแกรมเป็นเรื่องไม่สะดวก เนื่องจากจะต้องจำรหัสคำสั่ง ผู้เขียนจะเข้าใจได้ยาก ทำการตรวจสอบหาข้อผิดพลาดของโปรแกรมได้ยาก ฉะนั้นจึงต้องอาศัยโปรแกรมตัวแปลภาษาคอมพิวเตอร์ (Translator) ในการแปลภาษาคอมพิวเตอร์ภาษาต่าง ๆ ไปเป็นภาษาเครื่อง โปรแกรมที่แปลจากโปรแกรมต้นฉบับ แล้วจะเรียกว่ารหัสจุดหมาย (Object Code) ซึ่งจะประกอบด้วย รหัสคำสั่งที่คอมพิวเตอร์สามารถเข้าใจ และนำไปปฏิบัติต่อได้

2.8.2 การทำงานของคอมไพเลอร์

ถ้าต้องการรันโปรแกรม ต้องสั่งงานคอมไพเลอร์ให้ทำการแปลภาษานั้นแล้วทำการ เอ็กซีคิวต์ (Execute) โปรแกรมต้นฉบับ



รูปที่ 2.14 กระบวนการคอมไพล์โค้ดโปรแกรม

การทำงานของคอมไพเลอร์จะประกอบด้วย 5 ขั้นตอนหลัก

1) Lexical Analyzer เป็นขั้นตอนในการตรวจจับและจัดเรียงคำโปรแกรมต้นฉบับที่ต้องการแปลภาษา

2) Syntax Analyzer เป็นขั้นตอนในการตรวจสอบไวยากรณ์ของภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ โดยจะคืนให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Semantic Analyzer เป็นขั้นตอนในการตรวจสอบชนิดของข้อมูลที่จะนำมาประมวลผล และทำการเปลี่ยนรูปแบบของภาษาให้อยู่ในรูปของรูปแบบขั้นกลาง (Intermediate form) เพื่อรอการแปลงให้เป็นรหัสจุดหมาย

4) Code Generation เป็นการแปลงรูปแบบขั้นกลางให้เป็นรหัสจุดหมาย ซึ่งส่วนใหญ่จะเป็นภาษาแอสเซมบลี

5) Code Optimization เป็นการลดขั้นตอนการทำงานที่ซ้ำซ้อนกันของภาษาเครื่อง เพื่อให้การประมวลผลโดยเครื่องคอมพิวเตอร์ทำงานได้เร็วขึ้น

2.8.3 โปรแกรมคอมพิวเตอร์

หลักการการทำงานโดยทั่วไปของคอมพิวเตอร์จะต้องมีการป้อนข้อมูล ให้แก่คอมพิวเตอร์เสียก่อน แล้วคอมพิวเตอร์จึงจะสามารถทำการประมวลผล (Process) ตามโปรแกรมที่เราได้สร้างขึ้นมา เมื่อประมวลผลเสร็จ คอมพิวเตอร์จะนำผลลัพธ์ที่ได้ออกมาแสดงผลในรูปแบบต่าง ๆ ส่วนประกอบของโปรแกรมคอมพิวเตอร์ โดยทั่วไปจะมีส่วนประกอบ 3 ส่วนดังนี้

2.8.3.1 ข้อมูลนำเข้า (Input)

ส่วนที่คอมพิวเตอร์มีไว้สำหรับรับนำข้อมูลเข้ามาใช้ในการประมวลผล ในการเขียนโปรแกรมถ้าจะต้องมีการใช้ข้อมูลจากภายนอก ก็จะต้องมีการกำหนดหน่วยรับข้อมูล ไว้ในโปรแกรมเสียก่อน เช่น อาจจะใช้แป้นพิมพ์ (Keyboard) เป็นต้น

2.8.3.2 กระบวนการ (Process)

การเขียนชุดคำสั่งในการดำเนินงานของโปรแกรมเพื่อให้เครื่องคอมพิวเตอร์ดำเนินงานตามวัตถุประสงค์ของผู้เขียน ในส่วนนี้อาจจะมีการสร้างตัวแปรในรูปแบบต่าง ๆ ขึ้นมาเพื่อรองรับข้อมูลที่จะนำเข้ามาใช้ในการประมวลผล

2.8.3.3 ข้อมูลส่งออก (Output)

ส่วนที่คอมพิวเตอร์มีไว้สำหรับนำผลลัพธ์ที่ได้จากการประมวลผลออกมาแสดงผลหรือจัดเก็บ โดยจะต้องมีการกำหนดหน่วยที่เป็นข้อมูลส่งออกไว้ในส่วนของโปรแกรมเสียก่อน ตัวอย่างหน่วยข้อมูลส่งออก ที่เห็นได้ง่าย คือ จอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.4 ไทนี่ซี คอมไพเลอร์ (Tiny C Compiler)

ในการพัฒนาโครงการนี้ ได้เลือกใช้ทีซีซี (TCC: Tiny C Compiler) เพราะมีขนาดเล็กและสามารถทำงานได้กับระบบปฏิบัติการวินโดวส์ และทำงานได้ทั้งสถาปัตยกรรมหน่วยประมวลผลแบบ x86 และ x64 อีกทั้ง พีเอชพีฟังก์ชันในการเรียกใช้คำสั่งบนคอมมานด์พรอมท์ (command prompt) ได้โดยตรง คุณสมบัติที่โดดเด่นกว่าคอมไพเลอร์ตัวอื่น ๆ คือ สามารถคอมไพล์, ลิงค์ออบเจกต์โค้ดและได้โปรแกรมสกุล .exe ภายในคำสั่งเดียว

ทีซีซีมีคุณลักษณะอื่น ๆ อีก เช่น เมื่อโค้ดโปรแกรมที่นำไปคอมไพล์มีลักษณะเหมือนกันทุกประการ ทีซีซีใช้เวลาคอมไพล์น้อยกว่าจีซีซี (GCC) ถึง 9 เท่า, สามารถใช้ C dynamic library ได้โดยตรงและมีฟีเจอร์ (Feature) หลากหลายที่เป็นไปตามมาตรฐาน ISO/IEC 9899:1999 และ GNU C extensions และมีทุกฟีเจอร์ตามมาตรฐาน ANSI C extension

ในการคอมไพล์ด้วยคำสั่งทีซีซีสามารถทำได้โดยใช้คำสั่ง `tcc source.c` โดยที่ `source.c` คือ โปรแกรมต้นฉบับที่จะทำการคอมไพล์หลังจากนั้นจะได้ไฟล์ `source.exe` ซึ่งเป็น executable file พร้อมนำไปรันได้ รูปแบบการใช้คำสั่งโดยทั่วไปของ TCC คือ `tcc [options] [infile1 infile2...] ['-run' infile args...]` และป้อนอินพุตเข้าโปรแกรมโดยคำสั่ง `echo` ผ่านเครื่องหมายไพป์ (`|`) โดยรูปแบบคำสั่ง คือ `echo input1 input2 ... | source.exe` ซึ่งการไพป์คือการนำผลลัพธ์ที่ได้จากโปรแกรมหนึ่ง ไปเป็นอินพุตของอีกโปรแกรมหนึ่ง

ตารางที่ 2.9 สรุปออปชันของทีซีซี

Option	คำอธิบาย
-v	แสดงเวอร์ชันปัจจุบันของทีซีซี
-c	สร้างออบเจกต์ไฟล์ (object file)
-o outfile	ป้อนออบเจกต์ไฟล์, executable หรือ dll ให้ "outfile"
-Bdir	กำหนดเส้นทาง (path) ที่ไลบรารีภายในของทีซีซี (tcc internal libraries) ถูกเก็บไว้
-bench	ส่งออกสถิติการคอมไพล์
-run source [args...]	คอมไพล์ไฟล์ "source" และรันด้วยอาร์กิวเมนต์แบบคอมมานด์ไลน์ "args"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 นิพจน์ปรกติ (Regular expression)

นิพจน์ปรกติ (regular expression) คือ สายอักขระ (string) ที่อธิบายถึงรูปแบบของสายอักขระตามโครงสร้างรูปแบบที่กำหนด นิพจน์ปรกตินั้นมีการใช้อย่างแพร่หลายในโปรแกรมประเภท text editor ในการค้นหาและปรับเปลี่ยนข้อความ

นิพจน์ปรกติใช้อธิบายรูปแบบของสายอักขระ โดยการอธิบายจะกระชับและรัดกุมโดยไม่จำเป็นต้องอธิบายเป็นรายชื่อทั้งหมด เช่น "ex(pres|pan)sion" มีความหมายว่า expression หรือ expansion

ตารางที่ 2.10 การอธิบายสายอักขระ

	คำอธิบาย
	เครื่องหมายไพบี (pipe) ใช้สำหรับสร้างทางเลือกที่จะใช้ค้นหา นิพจน์ เช่น "express expand" มีความหมายว่า express หรือ expand
()	เครื่องหมายวงเล็บเหลี่ยม ใช้ในการกำหนดลำดับและขอบเขตของนิพจน์ เช่น "ex(pres pan)sion" มีความหมายว่า expression หรือ expansion
?	เครื่องหมายปริศนา ใช้บ่งบอกว่ามีนิพจน์ก่อนหน้านี้จำนวน 0 หรือ 1 นิพจน์ (มีนิพจน์) ปร" เช่น (หรือไม่มีก็ได้?กติก ปกติ หรือ ปกติมีความหมายว่า "
*	เครื่องหมายดอกจัน ใช้บ่งบอกว่ามีนิพจน์ก่อนหน้านี้จำนวน 0 นิพจน์หรือมากกว่า เช่น "go*gle" มีความหมายว่า ggle gogle google gooogle เป็นต้น
+	เครื่องหมายบวก ใช้บ่งบอกว่ามีนิพจน์ก่อนหน้านี้จำนวน 1 นิพจน์หรือมากกว่า เช่น "go+gle" มีความหมายว่า gogle gogle gooogle เป็นต้น แต่ไม่เท่ากับ ggle
{n}	วงเล็บปีกกาใช้บ่งบอกว่าต้องมีนิพจน์ก่อนหน้านี้จำนวน n นิพจน์ เช่น "go{5}gle" มีความหมายว่า goooogle เพียงตัวเดียว
{m,n}	วงเล็บปีกกาที่มีจุลภาคข้างใน ใช้บ่งบอกว่าต้องมีนิพจน์ก่อนหน้านี้จำนวน m นิพจน์ถึง n นิพจน์ เช่น "go{2,4}gle" มีความหมายว่า gooogle gooogle และ goooogle หากไม่ระบุตัวเลข n จะหมายถึงต้องมีนิพจน์จำนวนตั้งแต่ m นิพจน์เป็นต้นไป
^	เครื่องหมายหมวก ใช้เขียนเป็นตัวแรกสุดของนิพจน์ปรกติ เพื่อบ่งบอกว่านิพจน์ที่ต้องการค้นหาอยู่ต้นบรรทัดหรือต้นสตริงเท่านั้น
\$	เครื่องหมายสกุลเงิน ดอลลาร์ ใช้เขียนเป็นตัวท้ายสุดของนิพจน์ปรกติ เพื่อบ่งบอกว่านิพจน์ที่ต้องการค้นหาอยู่ท้ายบรรทัดหรือท้ายสตริงเท่านั้น

ตารางที่ 2.10 (ต่อ)

□	วงเล็บเหลี่ยม ใช้เขียนแทนอักขระหนึ่งตัวที่อยู่ในขอบเขต ซึ่งระบุไว้ภายในวงเล็บ เช่น "[bcdf]an" มีความหมายว่า ban can dan และ fan
^	เครื่องหมายหมวก ใช้เขียนภายในวงเล็บเหลี่ยมเป็นตัวแรก เพื่อบ่งบอกว่าเป็นอักขระหนึ่งตัวที่ไม่ได้อยู่ในขอบเขต เช่น "[^bcdf]an" มีความหมายว่า aan ean gan han ian เป็นต้น
-	เครื่องหมายยัติภังค์ ใช้เขียนภายในวงเล็บเหลี่ยมระหว่างอักขระสองตัว เพื่อบ่งบอกว่าเป็นอักขระหนึ่งตัวที่มีรหัสแอสกีหรือยูนิโคดอยู่ในช่วงนั้น เช่น "[a-z]an" มีความหมายว่า aan ban can เรื่อยไปจนถึง zan
.	เครื่องหมายมหัพภาค ใช้บ่งบอกอักขระในตำแหน่งนั้นของนิพจน์จะเป็นตัวอักษรอะไรก็ได้ เช่น ".an" มีความหมายว่า aan ban can 0an 1an -an เป็นต้น
\	เครื่องหมายแบ็กสแลช ใช้ระบุสัญลักษณ์ที่ต้องการใช้ในนิพจน์ แต่ซ้ำกับสัญลักษณ์ที่กล่าวมาทั้งหมดด้านบน เช่น "[0-9]\.[0-9]\\$" มีความหมายว่า 2.3\$ 9.0\$ 1.7\$ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 ความต้องการของระบบ และลักษณะสำคัญ

3.1.1 วิเคราะห์รูปแบบของโจทย์

โจทย์ต่างๆ ที่อาจารย์จะมอบหมายให้นักศึกษาทำมีหลากหลายรูปแบบ ก่อนการออกแบบและสร้างระบบตรวจสอบผลลัพธ์ของโปรแกรมควรต้องวิเคราะห์รูปแบบการป้อนอินพุตและส่งออกเอาต์พุตของแต่ละโจทย์ ซึ่งมีรูปแบบของโจทย์ดังนี้

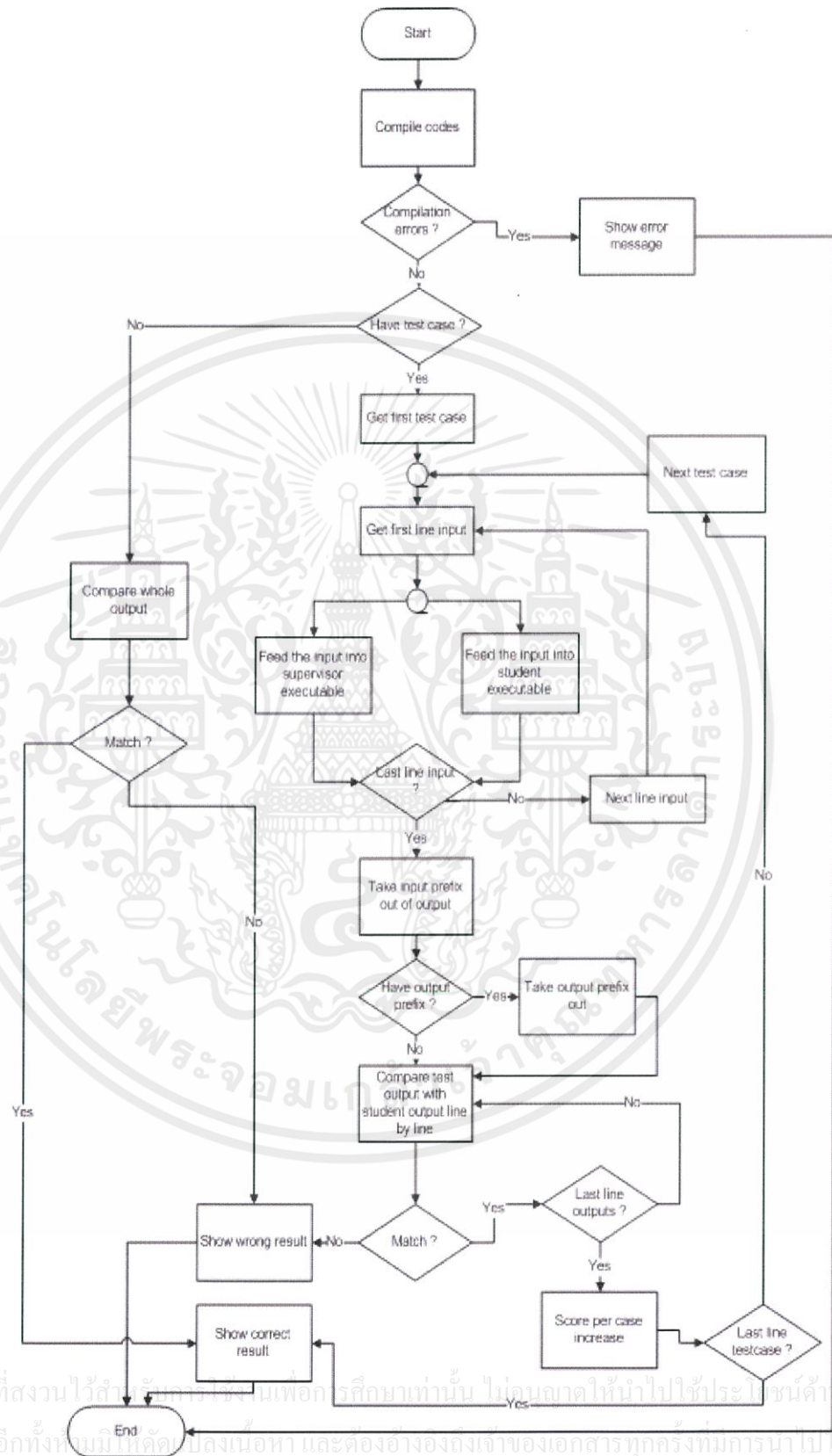
- 1) โจทย์ที่ไม่ต้องการรับอินพุต เช่น ให้แสดงจำนวนระหว่าง 1 ถึง 10
- 2) โจทย์ที่รับอินพุตค่าเดียว เช่น ให้แสดงผลคูณระหว่าง 1 ถึง ตัวเลขที่รับเข้ามา
- 3) โจทย์ที่รับอินพุตมากกว่า 1 ค่า เช่น หาค่าที่น้อยที่สุดจากเลขจำนวนเต็มทั้งหมดที่รับเข้ามา
- 4) โจทย์ที่ส่งออกเอาต์พุตค่าเดียว เช่น ให้หาค่าที่มากที่สุด
- 5) โจทย์ที่ส่งออกเอาต์พุตมากกว่า 1 ค่า เช่น ให้คำนวณพื้นที่แล้วเส้นรอบวงของรูปสี่เหลี่ยมจากความยาวและความกว้างที่ป้อนเข้ามา

3.1.2 ความต้องการของระบบตรวจสอบผลลัพธ์ของโปรแกรม

ความต้องการของระบบตรวจสอบผลลัพธ์ของโปรแกรมหามีดังต่อไปนี้

- 1) ระบบต้องสามารถตรวจสอบผลลัพธ์ของโปรแกรมภาษาซีได้
- 2) ตรวจสอบโปรแกรมที่รันบนคอมมานด์พรอมท์ (command prompt) ได้
- 3) สามารถตรวจสอบความถูกต้องของผลลัพธ์ของโปรแกรมจากนักศึกษาด้วยการเทียบผลการรันกับผลลัพธ์จากโค้ดโปรแกรมตัวอย่างที่เหมือนกันที่อาจารย์ส่งเข้าระบบ
- 4) ระบบต้องสามารถตรวจจับข้อผิดพลาด (error) และแจ้งผลการตรวจโปรแกรมได้
- 5) ระบบมีการแสดงผลการตรวจหลังจากส่งโค้ดโปรแกรมทันที โดยหลังจากส่งงานแล้วจะมีหน้าต่างแจ้งผลการตรวจขึ้น ผลลัพธ์ของโปรแกรมจากโค้ดของนักศึกษาแต่ละกรณีทดสอบ, วันเวลาที่ตรวจ, ผลการคอมไพล์, คะแนน, หน่วยความจำที่ใช้
- 6) สามารถตรวจโปรแกรมที่มีหลายอินพุตและหลายเอาต์พุตได้
- 7) ระบบต้องสามารถตรวจสอบนามสกุลของโค้ดโปรแกรมที่นักศึกษาส่งเข้ามาสู่ระบบได้

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น



รูปที่ 3.1 แผนภาพการทำงานส่วนตรวจโปรแกรมและให้คะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดูบนเว็บไซต์ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 รายการตรวจสอบความต้องการของระบบ (System Requirement Checklist)

3.2.1 ข้อมูลนำเข้า (input)

- 1) นักศึกษาต้องอัปโหลดไฟล์โค้ดโปรแกรมภาษาซีเข้าสู่ระบบเพื่อทำการตรวจและให้คะแนน
- 2) นักศึกษาสามารถปรับปรุงโค้ดโปรแกรมภาษาซีและอัปโหลดเข้าสู่ระบบได้ หากโจทย์ข้อนั้นได้คะแนนไม่เป็นที่น่าพอใจ
- 3) อาจารย์ต้องเพิ่มโจทย์ให้ระบบพร้อมทั้งกำหนดกรณีทดสอบและโค้ดโปรแกรมคำตอบ
- 4) อาจารย์สามารถแก้ไขโจทย์, กรณีทดสอบและโค้ดโปรแกรมคำตอบได้แล้วส่งกลับไปให้ระบบ
- 5) อาจารย์ต้องกำหนดเซสชัน (session) สำหรับการส่งโค้ดโปรแกรมภาษาซีให้นักศึกษาในแต่ละบทเรียน โดยในการสร้างเซสชัน สามารถกำหนดชื่อ, วันและเวลาที่เริ่ม, วันและเวลาที่สิ้นสุด, รูปแบบไฟล์ เป็นต้น
- 6) อาจารย์สามารถทำการทดสอบความถูกต้องของผลลัพธ์ของโจทย์ได้โดยการอัปโหลดโค้ดโปรแกรมเพื่อคอมไพล์และรันโปรแกรม

3.2.2 ข้อมูลส่งออก (output)

- 1) ระบบต้องแสดงรายชื่อนักศึกษาเรียนแลบโปรแกรมมิง
- 2) ระบบต้องแสดงคะแนนและผลการตรวจผลลัพธ์ของโปรแกรมของนักศึกษา

3.2.3 กระบวนการ (process)

- 1) ระบบต้องตรวจผลลัพธ์ของโปรแกรมและให้คะแนนได้
- 2) ระบบสามารถเตรียมโจทย์และกรณีทดสอบพร้อมกำหนดคะแนนแต่ละอินพุตเพื่อให้นักศึกษาทำโจทย์ได้
- 3) ระบบสามารถติดต่อกับฐานข้อมูลผ่านเว็บไซต์ได้
- 4) ระบบสามารถแจ้งเตือนเวลาที่เหลือที่นักศึกษาสามารถทำโจทย์หรือสามารถแจ้งเตือนเมื่อส่งโค้ดโปรแกรมเกินจำนวนครั้งที่กำหนดไว้
- 5) ระบบมีการแสดงตัวอย่างผลลัพธ์ของโปรแกรมเพื่อให้นักศึกษาสามารถทำความเข้าใจคำสั่งของโจทย์ได้ง่ายขึ้น

3.2.4 ประสิทธิภาพ (performance)

- 1) ระบบต้องสามารถตรวจตรวจผลลัพธ์ของโปรแกรมให้คะแนนได้ทันทีหลังจากได้รับไฟล์โค้ดโปรแกรมจากนักศึกษา
- 2) ระบบต้องสามารถรองรับผู้ใช้งานจำนวนมากได้ขณะที่เข้าเว็บไซต์ในเวลาเดียวกัน
- 3) ระบบสามารถแจ้งเตือนเวลาที่เหลือที่นักศึกษาสามารถทำโจทย์หรือสามารถแจ้งเตือนเมื่อส่งโค้ดโปรแกรมเกินจำนวนครั้งที่กำหนดไว้

3.2.5 ส่วนควบคุม (control)

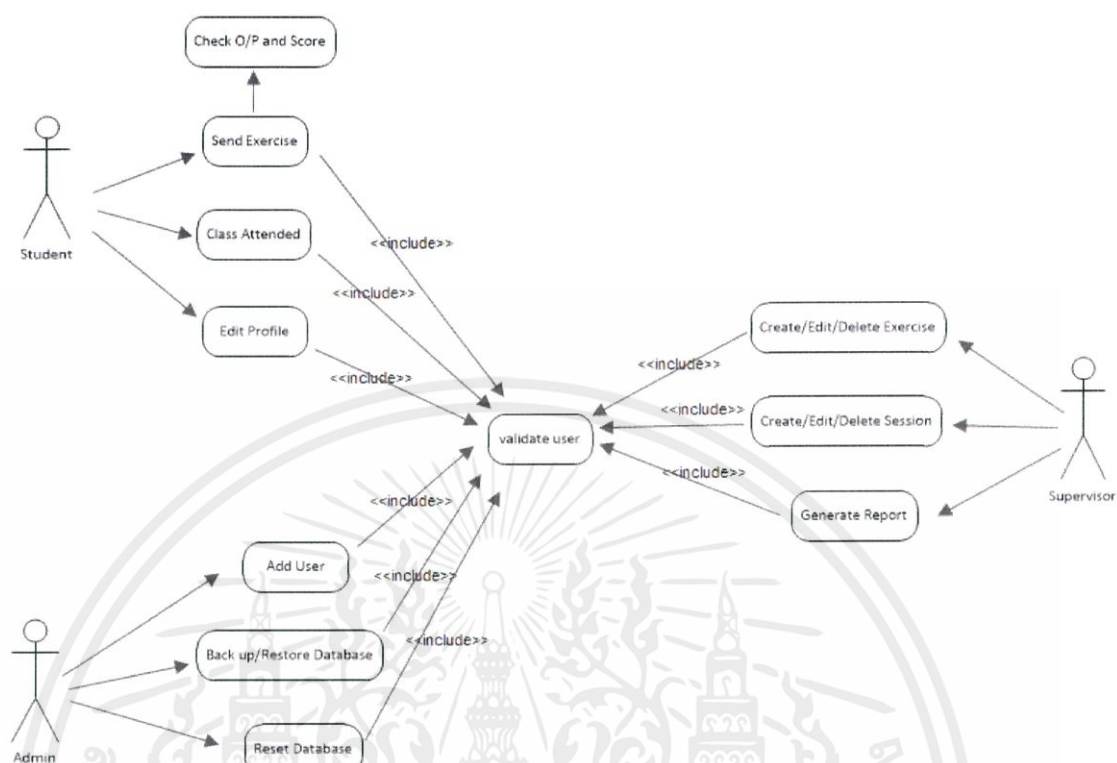
- 1) โค้ดโปรแกรมที่นักศึกษาส่งเข้าสู่ระบบไม่ควรเกิดการผิดพลาดขณะคอมไพล์ (compilation error) และการผิดพลาดขณะรัน (runtime error) หลังจากผ่านการคอมไพล์และเอ็กซิคิวต์ในระบบ
 - 2) โค้ดโปรแกรมต้องอยู่ในรูปแบบของไฟล์สกุล .c เท่านั้น
 - 3) ผู้ใช้ระบบต้องป้อนชื่อผู้ใช้และรหัสผ่านเพื่อเข้าสู่ระบบ
 - 4) ชื่อผู้ใช้และรหัสผ่านต้องถูกเพิ่ม, แก้ไขหรือลบโดยผู้ดูแลระบบเท่านั้น
 - 5) ระบบต้องแบ่งสิทธิของผู้ใช้เพื่อกำหนดตารางฐานข้อมูลที่สามารถเข้าถึงได้
 - 6) ระบบคอมไพล์และรันโปรแกรมบนคอมมานด์พรอมท์
 - 7) ระบบต้องสามารถสร้างล็อกไฟล์ (log file) เพื่อเก็บข้อมูลการเข้าใช้ระบบสำหรับการตรวจสอบการกระทำที่เกิดขึ้นกับระบบในภายหลังได้

3.3 การใช้ระบบ

3.3.1 แผนภาพการใช้ระบบ (Use case diagram)

แผนภาพการใช้ระบบเป็นแบบจำลองเพื่อใช้อธิบายหน้าที่ต่าง ๆ ประกอบไปด้วย หน้าที่ (Use case), ตัวแสดง (Actor) และระบบ (System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนภาพการใช้งานระบบจัดการแลบโปรแกรมมิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

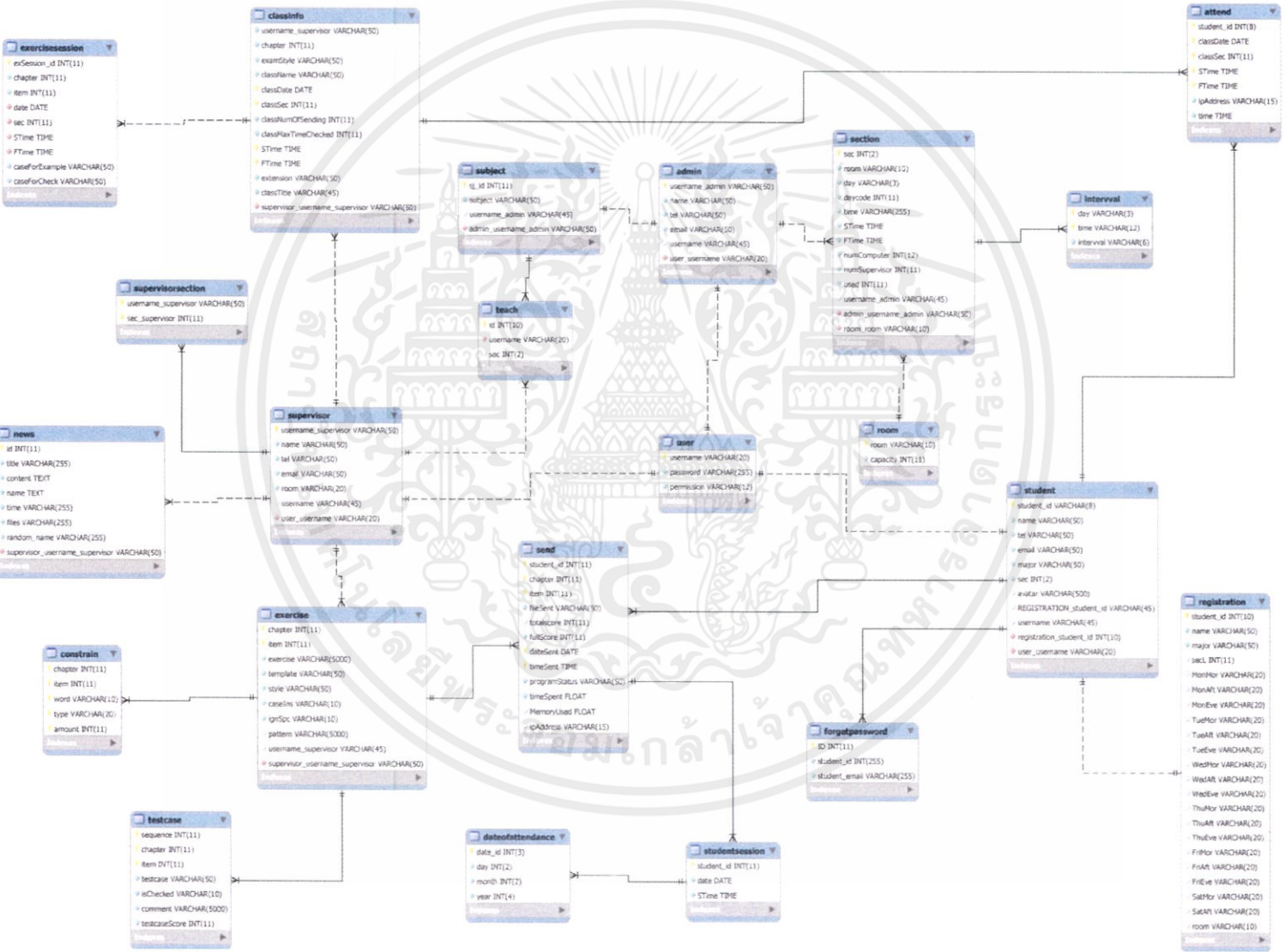
3.3.2 ตารางการใช้งานระบบ

ตารางที่ 3.1 การใช้งานระบบแบ่งตามประเภทผู้ใช้งาน

นักศึกษา	อัปโหลดไฟล์โค้ดโปรแกรมภาษาซีเข้าสู่ระบบ ตรวจผลลัพธ์และคะแนนจากไฟล์โค้ดโปรแกรมภาษาซีที่อัปโหลด ตรวจประวัติการส่งย้อนหลัง
อาจารย์	สร้าง, แก้ไขหรือลบเซสชัน สร้าง, แก้ไขหรือลบโจทย์
ผู้ดูแลระบบ	เพิ่มหรือลบผู้ใช้งานในระบบ นำเข้าฐานข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบฐานข้อมูล



รูปที่ 3.3 เอนทิตีและความสัมพันธ์ระหว่างเอนทิตีของฐานข้อมูลในระบบ

3.5 การออกแบบส่วนติดต่อผู้ใช้

3.5.1 ผู้ใช้ประเภทนักศึกษา



รูปที่ 3.4 หน้าต่างหลังของผู้ใช้ประเภทนักศึกษา

นักศึกษาคือผู้ใช้ระบบที่ทำการส่งโค้ดโปรแกรมเข้าสู่ระบบเพื่อตรวจสอบผลลัพธ์ของโปรแกรมที่สอดคล้องกับคำสั่งของโจทย์ที่ได้รับ ที่หน้าเว็บหลังจะมีการแสดงรายละเอียดของชั้นเรียน คือ ชื่อชั้นเรียน, วันและเวลา, เลขที่กำหนด, เรื่อง, กลุ่มเรียน, จำนวนครั้งในการส่งสูงสุด, ชนิดไฟล์ที่อนุญาต และรูปแบบชื่อไฟล์ที่กำหนด นักศึกษาสามารถใช้งานส่วนต่าง ๆ ได้ คือ

- 1) ทำโจทย์ เพื่อเข้าสู่โจทย์ตัวอย่างและส่งโค้ดโปรแกรมเพื่อตรวจสอบผลลัพธ์
- 2) ประวัติการส่งงาน เพื่อดูรายละเอียดการส่งโค้ดโปรแกรมที่ผ่านมาของตนเอง
- 3) ตารางเรียนส่วนบุคคล เพื่อดูตารางเรียนเลขซึ่งแจ้งวันเวลาและกลุ่มเรียน
- 4) แจ้งปัญหาหรือข้อเสนอ เพื่อให้นักศึกษาแจ้งผู้ดูแลระบบได้
- 5) ข้อมูลส่วนตัว เพื่อปรับปรุงข้อมูลของนักศึกษา สามารถเพิ่มเติมข้อมูลหมายเลขโทรศัพท์และอีเมล อีกทั้งยังสามารถเปลี่ยนรหัสผ่านได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Programming Lab Management System

ระบบบริหารจัดการห้องปฏิบัติการเขียนโปรแกรม



ชื่อ : นายชนันต์ สิงใจเขียน
ID : 52010000
SEC : 4

[Home](#)

[Logout](#)

Class Information

ชื่อ Class :	Computer and Programming Laboratory	แลปที่	4
เรื่อง :	การเขียนโปรแกรมคำสั่งเงื่อนไข	กลุ่มเรียน	4
วันเวลา :	13/03/2013 17:00:00 ถึง 13/03/2013 19:00:00	จำนวนครั้งในการส่งสูงสุด	ไม่จำกัด

▼ Menu

- ทำโจทย์
- ประวัติการส่งงาน
- ตารางเรียนส่วนบุคคล
- แจ้งปัญหา ข้อสงสัยและ
- ข้อมูลส่วนตัว

ทำแบบทดสอบ

แลปที่ 4 ข้อที่ 3

จงเขียนโปรแกรมเพื่อรับค่าตัวแปรเป็นตัวเลขจำนวนเต็ม 1 จำนวนแล้วแสดงผลสลับทวิว่าตัวเลขที่รับเข้ามาเป็นเลข คู่หรือเลขคี่

ตัวอย่างการแสดงผล

```
กรณีที่ 1
Enter 1st num : 9
Num : Odd
กรณีที่ 2
Enter 1st num : 3
Num : Odd
กรณีที่ 3
Enter 1st num : 7
Num : Odd
```

เวลาเริ่มต้น : 17:00:00 น.

เวลาสิ้นสุด : 19:00:00 น.

เวลาปัจจุบัน : 18:36:30 น.

เวลาที่เหลือ : 1 ชั่วโมง 36 นาที

Source code No file chosen

*ส่งมาแล้ว 0 ครั้ง

รูปที่ 3.5 หน้าต่างรายละเอียดโจทย์และตัวอย่างผลลัพธ์ของโจทย์

ที่รายการทำแบบทดสอบ แสดงรายละเอียดของชั้นเรียนบริเวณบนของหน้าต่างและแสดงรายละเอียดของโจทย์ซึ่งประกอบไปด้วยหมายเลขแลบและหมายเลขข้อ, คำสั่ง, เวลาเริ่มและสิ้นสุด, เวลาปัจจุบันและเวลาที่เหลือ พร้อมปุ่มเลือกโปรแกรมต้นฉบับเพื่อให้นักศึกษาเลือกไฟล์โค้ดโปรแกรมมาส่ง และปุ่มส่งโปรแกรมต้นฉบับเพื่อทำการส่งโค้ดโปรแกรมสู่ระบบ โดยการแสดงผลตัวอย่างผลลัพธ์จะถูกจัดแสดงในรูปแบบพื้นหลังสีดำและตัวอักษรสีขาว เพื่อให้มีลักษณะคล้ายกับการรันบนเทอร์มินอล แบ่งการแสดงผลแยกแต่ละกรณีทดสอบซึ่งจะมีอินพุตที่แตกต่างกันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Programming Lab Management System

ระบบจัดการห้องปฏิบัติการเขียนโปรแกรม



ชื่อ นายชนม์ ตั้งใจเย็น
ID: 52010000
SEC: 4

Home

Logout

Class Information

ชื่อ Class :	Computer and Programming Laboratory	แลปที่ :	4
เรื่อง :	การเขียนโปรแกรมสาสิ่งเงินไข	กลุ่มเรียน :	4
วันเวลา :	13/03/2013 17:00:00 ถึง 13/03/2013 19:00:00	จำนวนครั้งในการส่งสูงสุด :	ไม่จำกัด

Menu

- ทาโจทย์
- ประสิทธิภาพสงงาน
- ตารางเรียนสวขบุคคล
- แจงปัญหา ขอเสนอแนะ
- ขอมูลสวนตัว

ผลการตรวจ

โจทย์บที่ 4 ข้อที่ 3

```
กรณีที่ 1
Enter 1st num : 9
Num : Odd
กรณีที่ 2
Enter 1st num : 3
Num : Odd
กรณีที่ 3
Enter 1st num : 7
Num : Odd
```

ผลการคอมไพล์: No error

คะแนน: 3 / 3

```
#include<stdio.h>
int main()
{
    int a;
    printf("Enter 1st num : ");
    scanf("%d",&a);
    if(a%2 == 1)
        printf("Num : Odd");
    else
        printf("Num : Even");
    return 0;
}
```

รูปที่ 3.6 หน้าต่างรายละเอียดการตรวจผลลัพธ์

เมื่อนักศึกษาทำการส่งโค้ดโปรแกรมเข้าสู่ระบบ จะปรากฏหน้าต่างแสดงรายละเอียดของการตรวจผลลัพธ์ ซึ่งประกอบไปด้วย วัน เวลาที่ตรวจ, ผลการคอมไพล์, คะแนน, เวลาและหน่วยความจำที่ใช้ในการคอมไพล์ พร้อมรายละเอียดโค้ดที่นักศึกษาเขียนและตัวอย่างของผลลัพธ์ของโปรแกรมตามกรณีทดสอบ หากกรณีทดสอบใดให้ผลลัพธ์ที่ถูกต้องตามโจทย์กำหนดก็จะแสดงเครื่องหมายถูก และหากผิดก็จะแสดงเครื่องหมายกากบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 ผู้ใช้ประเภทอาจารย์

รูปที่ 3.7 หน้าต่างหลักของผู้ใช้ประเภทอาจารย์

อาจารย์คือผู้ใช้ระบบเพื่อจัดการสร้างชั้นเรียนพร้อมทั้งกำหนดโจทย์ ที่หน้าหลักนั้นสามารถสร้างชั้นเรียนได้ อาจารย์สามารถใช้งานส่วนต่าง ๆ ได้ เช่น

- 1) สร้างชั้นเรียน เพื่อกำหนดรายละเอียดเขาสชั้นและโจทย์
- 2) รายการเขาสชั้น เพื่อเลือกรูปแบบโจทย์และสามารถแก้ไขหรือลบเขาสชั้นได้
- 3) สร้างโจทย์ เพื่อเพิ่มโจทย์สำหรับเตรียมให้นักศึกษาทำ
- 4) รายการโจทย์ / แก้ไขโจทย์ เพื่อดูโจทย์ทั้งหมดที่มี และเพื่อแก้ไขหรือลบโจทย์
- 5) ประกาศ เพื่อแจ้งข้อมูลข่าวสารต่าง ๆ
- 6) จองกลุ่มเรียนคุ่มแลบ เพื่อให้อาจารย์เลือกกลุ่มเรียนที่จะคุ่มแลบ

ก่อนที่นักศึกษาจะสามารถทำโจทย์ได้นั้น อาจารย์ต้องทำการสร้างชั้นเรียนขึ้นก่อน โดยกำหนดที่หน้าต่างหลักซึ่งประกอบไปด้วย เขาสชั้น, หัวเรื่อง, ห้อง, วันเวลาเริ่มต้นและสิ้นสุด, แลบที่กำหนด, กลุ่มเรียน, จำนวนครั้งในการส่งสูงสุด, ชนิดหรือสกุลของไฟล์ที่อนุญาตให้ส่งได้ โดยสามารถทำการแก้ไขหรือลบรายละเอียดชั้นเรียน และกำหนดรูปแบบโจทย์ในภายหลังได้ที่รายการ "รายการเขาสชั้น"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming Lab Management System

Home Logout Username : super

Menu

- สร้างเซสชัน
- รายการเซสชัน
- สร้างโจทย์
- รายการโจทย์/แก้ไขโจทย์
- ทลสอบโจทย์
- ประวัติการสงงาน

สร้างโจทย์

เลขที่:

ข้อที่:

คำสั่ง:

รูปแบบการตรวจ:

การตรวจ:

```
Enter num : 7
Result : 5040
Enter num : 3
***
***
```

 *ตัวอย่างการแสดงผล

มีเฉพาะ Output ไม่มี Input

กรณีทดสอบ: *คลิกที่ checkbox ก่อนลบ

กรณีทดสอบ:

เงื่อนไข:

คำสั่ง:

ตัวเลือก: การตรวจคำตอบ ไม่สนใจเรื่องตัวพิมพ์เล็ก พิมพ์ใหญ่ (Case insensitive)
 การตรวจคำตอบ ไม่สนใจเรื่องการเว้นวรรคระหว่างคำตอบ (Ignorance of space)

ซอร์สโค้ด: No file chosen

*ซอร์สโค้ดต้องให้ผลลัพธ์ตรงกับรูปแบบการตรวจไว้แล้ว

รูปที่ 3.8 หน้าต่างการเพิ่มโจทย์

หน้าต่าเพิ่มโจทย์นั้นประกอบด้วยช่องกรอกข้อมูลต่าง ๆ เพื่อกำหนดรายละเอียดโจทย์ ดังนี้

- 1) หมายเลขแลบและหมายเลขข้อ
- 2) คำสั่ง เพื่อเป็นคำอธิบายให้นักศึกษาทราบถึงผลลัพธ์ที่โจทย์ต้องการ
- 3) รูปแบบการตรวจเฉพาะแต่ละโจทย์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อย่างไรก็ตามขอสงวนสิทธิ์ในชื่อและเครื่องหมายของเอกสารที่ปรากฏในภาพนี้ไว้

- 4) กรณีทดสอบ เพื่อเป็นการกำหนดอินพุตและคะแนนในแต่ละกรณีทดสอบ แล้วนำอินพุตที่กำหนดป้อนให้โปรแกรมที่นักศึกษาส่ง

- 5) เงื่อนไขเพิ่มเติมที่เป็นข้อบังคับของโจทย์แต่ละข้อ
- 6) ตัวเลือกเพื่อกำหนดรายละเอียดเพิ่มเติมในการตรวจผลลัพธ์
- 7) ปุ่มอัปโหลดโค้ดโปรแกรมของอาจารย์เพื่อใช้เทียบผลลัพธ์ของโปรแกรมนักศึกษา

ในการกรอกกรณีทดสอบนั้น ในช่องอินพุท 1 บรรทัด ต่อ 1 ครั้งที่รับอินพุท ซึ่งจะมีจำนวนเท่าที่โปรแกรมต้องการ ใช้การเว้นบรรทัดแบ่งจำนวนครั้งที่ป้อน

กรณีทดสอบ:

กรณีทดสอบ	ทดสอบเรื่อง	คะแนน
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

รูปที่ 3.9 ช่องกรอกกรณีทดสอบ

รูปแบบการตรวจมี 3 รูปแบบคือ

- 1) ต้องมีค่านำหน้าอินพุท (input prefix) ก่อนการรับค่า และมีค่านำหน้าเอาต์พุท (output prefix) หลังผลลัพธ์ โดยที่หลัง input prefix และ output prefix ต้องตามด้วยเครื่องหมายทวิภาคเสมอ เช่น โปรแกรมรับเลข 3 จำนวน จงหาวานำมาสร้างเป็นรูปสามเหลี่ยมได้หรือไม่ หากเป็น เป็นสามเหลี่ยมมุมฉากหรือไม่ ตัวอย่างอินพุทและเอาต์พุท คือหลังค่านำหน้าอินพุท และค่านำหน้าเอาต์พุทต้องตามด้วยเครื่องหมายทวิภาค (:) เสมอ

ตัวอย่าง 1.1 ข้อมูลจากการตรวจแบบที่ 1

```
//input
Enter nums : 3 4 5
//output
```

- 2) มีเฉพาะค่านำหน้าอินพุทแต่ไม่มีค่านำหน้าเอาต์พุทหลังผลลัพธ์ เช่น โปรแกรมรับเลข 1 จำนวนแล้ววาดรูปสี่เหลี่ยมมุมฉากด้วยเครื่องหมายดอกจัน หลังค่านำหน้าอินพุทต้องตามด้วยเครื่องหมายทวิภาคเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง 1.2 ข้อมูลจากการตรวจแบบที่ 2

```
//input
Enter nums : 2
//output
**
**
```

3) ตรวจเฉพาะอินพุท ในกรณีที่เอาท์พุทแต่ละโปรแกรมแตกต่างกัน อาจมีการกรอกนิพจน์ปรกติ (regular expression) ที่ช่องรูปแบบการตรวจเพื่อใช้ตรวจจับผลลัพธ์ที่โจทย์กำหนด เช่น จงเขียนโปรแกรม แสดงรหัสนักศึกษา ต้องกรอกนิพจน์ปรกติ ดังนี้ [0-9]{8} เพื่อตรวจจับเลข 8 ตัว หรือหากเลข 4 หลักแรกถูกกำหนดให้คงที่ นิพจน์ปรกติคือ (5201)[0-9]{4} เพื่อตรวจจับเลข 5201 แล้วตามด้วยเลข 4 ตัว

ตัวอย่าง 1.3 ข้อมูลจากการตรวจแบบที่ 3

```
// Output 1
52010413
// Output 2
52010389
// Output 3
52011139
```

โจทย์บางโจทย์อาจมีการกำหนดคำคีย์เวิร์ด (Keyword) ที่ห้ามใช้ สามารถเพิ่มเงื่อนไขได้โดยการเลือกคีย์เวิร์ด, ประเภทของเงื่อนไข, และจำนวนที่กำหนด



รูปที่ 3.10 รูปแบบการตรวจ

บางกรณีอาจไม่สนใจการเว้นวรรคระหว่างคำตอบหรือพิมพ์เล็กพิมพ์ใหญ่ จึงมีตัวเลือกเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวเลือก : การตรวจสอบ ไม่สนใจเรื่องตัวพิมพ์เล็ก พิมพ์ใหญ่ (Case insensitive)
 การตรวจสอบ ไม่สนใจเรื่องการเว้นวรรคระหว่างคำต่อ (Ignorance of space)

รูปที่ 3.11 ตัวเลือกเพิ่มเติมสำหรับการตรวจ

3.5.3 ผู้ใช้ประเภทผู้ดูแลระบบ



รูปที่ 3.12 หน้าต่างหลักของผู้ใช้ประเภทผู้ดูแลระบบ

ผู้ดูแลระบบคือผู้ใช้ที่จัดการดูแลระบบโดยรวม มีหน้าที่หลักในการจองกลุ่มเรียนสำหรับการจัดตารางเรียนวิชาแล็บโปรแกรมมิ่ง ผู้ดูแลระบบสามารถเข้าใช้งานส่วนต่าง ๆ ได้ คือ

- 1) จัดการข้อมูลผู้ใช้เพื่อเพิ่ม, แก้ไขหรือลบข้อมูลผู้ใช้
- 2) นำเข้าข้อมูลลงทะเบียนและจัดตารางเรียนแล็บ
- 3) เพิ่ม / ลบห้องเรียนที่จะใช้เป็นห้องแล็บ
- 4) เปิด / ปิดกลุ่มเรียนเพื่อใช้สำหรับจัดตารางเรียนแล็บ
- 5) จัดตารางเรียนแล็บโปรแกรมมิ่ง เพื่อส่งจัดตารางเรียนแล็บ
- 6) แก้ไขรหัสวิชา ในกรณีทีวิชาอาจมีการเปลี่ยนรหัสวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 สภาพแวดล้อมที่ใช้ในระบบ

ในการพัฒนาและทดลองระบบจัดการห้องแล็บโปรแกรมมิ่ง ต้องจัดสภาพแวดล้อมและเครื่องมือรวมถึงซอฟต์แวร์ต่างๆ ดังนี้

- 1) ระบบปฏิบัติการ: Windows XP Service pack 3, Windows 7
- 2) Web server: Apache 2.5.10 for Windows
- 3) Database Server: PhpMyAdmin
- 4) Web browser: Google Chrome
- 5) คอมพิวเตอร์พกพาเพื่อใช้ในการจำลองเป็นตัวเซิร์ฟเวอร์

4.2 ขั้นตอนการทดลอง

4.2.1 การเข้าสู่ระบบ

ในการใช้งานระบบจัดการห้องแล็บโปรแกรมมิ่ง ผู้ใช้จะต้องทำการเข้าสู่ระบบโดยการกรอกชื่อผู้ใช้และรหัสผ่าน ซึ่งหากกรอกชื่อผู้ใช้หรือรหัสผ่านผิด ก็จะไม่สามารถเข้าระบบได้ โดยผู้ใช้นั้นมี 3 ประเภท คือ นักศึกษา, อาจารย์ และผู้ดูแลระบบ



รายการ session ที่เปิดวันนี้ (13 March 2013)

กลุ่มที่ 2 บทที่ 5

เวลา 08:00:00 ถึง 10:00:00

รูปที่ 4.1 หน้าต่างกรอกข้อมูลเพื่อเข้าระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับคณาจารย์เพื่อใช้ในการสอนเท่านั้น เมื่อผู้ใดนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การเพิ่มข้อมูลผู้ใช้

การเพิ่มข้อมูลผู้ใช้นั้นสามารถกระทำได้โดยผู้ใช้ที่มีสิทธิ์เป็นผู้ดูแลระบบเท่านั้น

1) เลือกรายการเพิ่มผู้ใช้ จะปรากฏช่องกรอกรายละเอียดของผู้ใช้

▼ Menu

เพิ่มข้อมูลผู้ใช้

- จัดการห้องและกลุ่มเรียนแลบ
 - นำเข้าข้อมูลลงทะเบียน
 - เพิ่ม/ลบห้องแลบ
 - เปิด/ปิดกลุ่มเรียน
 - จัดการตารางเรียนแลบโปรแกรมมิ่ง
 - เพิ่มข้อมูลผู้ใช้**
 - แก้ไขข้อมูลผู้ใช้
 - แก้ไขรหัสวิชา

ชื่อผู้ใช้:

รหัสผ่าน:

ยืนยันรหัสผ่าน:

ชื่อ:

เบอร์โทร:

อีเมล:

สิทธิผู้ใช้:

รูปที่ 4.2 หน้าต่างเพิ่มผู้ใช้

2) กรอกรายละเอียดต่าง ๆ ของผู้ใช้ เช่น ชื่อผู้ใช้, รหัสผ่าน, หมายเลขโทรศัพท์, อีเมล พร้อมทั้งกำหนดสิทธิผู้ใช้ด้วย

▼ Menu

เพิ่มข้อมูลผู้ใช้

- จัดการห้องและกลุ่มเรียนแลบ
 - นำเข้าข้อมูลลงทะเบียน
 - เพิ่ม/ลบห้องแลบ
 - เปิด/ปิดกลุ่มเรียน
 - จัดการตารางเรียนแลบโปรแกรมมิ่ง
 - เพิ่มข้อมูลผู้ใช้
 - แก้ไขข้อมูลผู้ใช้
 - แก้ไขรหัสวิชา

ชื่อผู้ใช้: 52010413 ✓

รหัสผ่าน: ✓

ยืนยันรหัสผ่าน:

ชื่อ: นายทัตพล กุลไทย

เบอร์โทร: 0879221611

อีเมล: k.tattapol@gmail.com

สิทธิผู้ใช้: นักศึกษา

สาขาวิชา: วิศวกรรมคอมพิวเตอร์

กลุ่มเรียน: 7

รูปที่ 4.3 กรอกรายละเอียดของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) สามารถแก้ไขหรือลบผู้ใช้ได้

Menu

- จัดการห้องและกลุ่มเรียนแลบ
 - นำเข้าข้อมูลลงทะเบียน
 - เพิ่ม/ลบห้องแลบ
 - เปิด/ปิดกลุ่มเรียน
 - จัดการตารางเรียนแลบโปรแกรมมิ่ง
- เพิ่มข้อมูลผู้ใช้
- แก้ไขข้อมูลผู้ใช้
- แก้ไขรหัสวิชา

รายชื่อผู้ใช้ประเภทนักศึกษา

ชื่อผู้ใช้	ชื่อสกุล	โทร.	e-mail	สาขาวิชา	กลุ่มเรียน	แก้ไข	ลบ
49012324	นายนพพล			วิศวกรรมการ			
	ธราชีริน			วัดคุม (รับ	2	แก้ไข	ลบ
50010491	นายอิฐพล			วิศวกรรมการ			
	รัตธรรมกุล			วัดคุม	2	แก้ไข	ลบ
50011571	นายศิริโรตม			วิศวกรรม			
	จักรศ่า			โทรคมนาคม	2	แก้ไข	ลบ
51010046	นายกษิต			วิศวกรรม			
	วรรณอนม			โยธา	2	แก้ไข	ลบ

รูปที่ 4.4 รายการผู้ใช้

4.2.3 การจัดการโจทย์

1) เลือกรายการเพิ่มแบบฝึกหัดจะปรากฏช่องรายละเอียดการสร้างโจทย์ และทำการกรอกรายละเอียดของโจทย์

Menu

- สร้างเซสชัน
- รายการเซสชัน
- สร้างโจทย์
- รายการโจทย์/แก้ไขโจทย์
- ทดสอบโจทย์
- ประวัติการส่งงาน

Menu (ส่วนบุคคล)

- จองกลุ่มเรียนแลบ
- ตารางคะแนน
- ประกาศ
- นักศึกษาที่มีรหัสผ่าน
- ข้อมูลส่วนตัว

สร้างโจทย์

แลบที่ : 2

ข้อที่ : 1

คำสั่ง : พิมพ์ศึกษาแสดงผลประโยค "Hello World" ออกหน้าจอ

รูปแบบ มี Input/Output

การตรวจ: Input ถูก : 7
Result : 5044

```

        Input ถูก : 3
        ***
        ***
        ***
        
```

* ตัวอักษรบางตัวแสดงผล

มีเฉพาะ Output ไม่มี Input

เป็นแบบฝึกหัด การนำ Regular expression หรือรหัสพิเศษบางอย่างไปในแต่ละบุคคล

กรณีทดสอบ: เพิ่มกรณีทดสอบ ลบกรณีทดสอบ *คลิกที่ checkbox ก่อนลบ

กรณีทดสอบ: ทดสอบเรื่อง คะแนน

*บอกโจทย์ต้องกรอกอินพุต ให้ป้อนอินพุต 1 บรรทัดต่อการกรอกอินพุต 1 ครั้งไปและกรณีทดสอบ

เงื่อนไข:

คำสั่ง: ประเภท: จำนวน:

ตัวเลือก: การตรวจสอบ ไม่สนใจเรื่องตัวพิมพ์เล็ก พิมพ์ใหญ่ (Case insensitive)
 การตรวจสอบ ไม่สนใจเรื่องการเว้นวรรคระหว่างคำตอบ (Ignorance of space)

ขอสไลด์: เลือกไฟล์:


*คลิกที่ปุ่มเพื่อเพิ่มไฟล์หรือลบไฟล์ที่เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้เพื่อการค้าหรือเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 หน้าต่างเพิ่มโจทย์

2) สามารถแก้ไขหรือลบโจทย์ที่เคยเพิ่มไว้แล้วได้

รายการโจทย์

แลบ ข้อ	คำสั่ง	แก้ไข โจทย์	ลบ
2 1	ให้นักศึกษาแสดงผลประโยค "Hello World" ออกทางหน้าจอ		
2 4	ให้นักศึกษาเขียนโปรแกรมเพื่อแสดงผลลัพธ์เป็นชื่อและนามสกุลของนักศึกษา โดยให้ชื่อปรากฏอยู่ที่มบน ข้ายของจอภาพ และนามสกุลปรากฏอยู่ที่มบนขวาของจอภาพ		

รูปที่ 4.6 หน้าต่างรายการโจทย์

3) เมื่อกดแก้ไขโจทย์ก็จะปรากฏหน้าต่างเช่นเดียวกันกับขณะสร้างโจทย์ แต่จะมีข้อความของโจทย์เดิมคงอยู่แล้ว



Menu

- สร้างเซสชัน
- รายการเขาสืบ
- สร้างโจทย์
- รายการโจทย์/แก้ไขโจทย์
- ทดสอบโจทย์
- ประวัติการลงบันทึก
- Menu (ส่วนบุคคล)
- จองคลื่นเรียนคอมพิวเตอร์
- ตารางคอมพิวเตอร์
- ประกาศ
- นักศึกษาที่สมัครเรียน
- ข้อมูลส่วนตัว

แก้ไขโจทย์

บทที่ : 2

ข้อที่ : 1

คำสั่ง : ให้นักศึกษาแสดงผลประโยค "Hello World" ออกทางหน้าจอ

รูปแบบการ มี Input/Output

ตรวจ ตรวจสอบเฉพาะ Output

เพิ่มแบบผลลัพธ์ (กรอก formula, expression กรณีที่ผลลัพธ์อาจนำไปใช้และ)

Hello World

กรณีทดสอบ:

*คลิก checkbox ก่อนลบ

กรณีทดสอบ ทดสอบเรื่อง คะแนน

บทแก้ไขต้องกรอกอินพุต ให้ป้อนอินพุต 1 บรรทัดผลการกรอกอินพุต 1 กรณีในและกรณีในเฉลย

เงื่อนไข:

คีย์เวิร์ด ประเภท จำนวน

ตัวเลือก:

- การตรวจสอบ ไม่สนใจเรื่องตัวพิมพ์เล็ก พิมพ์ใหญ่ (Case insensitive)
- การตรวจสอบ ไม่สนใจเรื่องการเว้นวรรคระหว่างคำคอม (Ignorance of space)

ซอร์สโค้ด: ใช้โค้ดเดิม

*ซอร์สโค้ดต้องไม่มีล้นขีด นิยมรูปแบบการตรวจที่เลือก

รูปที่ 4.7 หน้าต่างการแก้ไขโจทย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การสร้างชั้นเรียน

1) เมื่อเลือกรายการสร้างเซสชัน จะปรากฏหน้าต่าง Class Information เพื่อใส่รายละเอียดของชั้นเรียนลงไป

Menu

- สร้างเซสชัน
- รายการเซสชัน
- สร้างโจทย์
- รายการโจทย์/แก้ไขโจทย์
- ทดสอบโจทย์
- ประวัติการส่งงาน

Class Information

ชื่อ Class : Computer and Programming Laboratory

เรื่อง : Hello World

วัน : 27/09/2012

เวลา : 16:30 ถึง 21:30

แลปที่ : 2

กลุ่มเรียน : 7

จำนวนครั้งในการส่งสูงสุด : 7

เวลาสูงสุดที่ใช้ตรวจ : 5 วินาที

ชนิดของไฟล์ที่ส่งได้ : .c .cpp

สร้างเซสชัน

Menu (ส่วนบุคคล)

- จองกลุ่มเรียนคอมพิวเตอร์
- ตารางคอมพิวเตอร์
- ประกาศ
- นักศึกษาที่ลี้ภัย
- ข้อมูลส่วนตัว

รูปที่ 4.8 หน้าต่างการสร้างชั้นเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เมื่อกดสร้างเซชันแล้ว จะแสดงรายละเอียดที่กรอกไปรวมถึงโจทย์ตามแลบที่กรอกไปข้างต้น

เลือกโจทย์/แก้ไขการเลือกโจทย์

*กรุณาเลือกหรือแก้ไขการเลือกโจทย์สำหรับเซชันนี้ โดยเลือกข้อและกรณีสอดคล้องต้องการ

แลบที่ 5

- ข้อที่ 1 จงเขียนโปรแกรมรับจำนวนเต็ม 1 จำนวน แล้วแสดงผล คูณหลังตัวเลข 1 ถึง จำนวนที่รับมา
- ข้อที่ 2 จงเขียนโปรแกรมรับทุกเป็นตัวเลขจำนวนเต็ม 1 จำนวนแล้วคำนวณหาผลคูณของตัวเลขตั้งแต่ 2 จนถึงตัวเลขที่ รับเข้าโดยวิธีคำสั่ง for เท่านั้น
- ข้อที่ 3 จงเขียนโปรแกรมรับทุกเป็นตัวเลขจำนวนเต็ม 1 จำนวนแล้วคำนวณหาผลคูณของตัวเลขตั้งแต่ 2 จนถึงตัวเลขที่ รับเข้าโดยวิธีคำสั่ง while เท่านั้น
- ข้อที่ 4 จงเขียนโปรแกรมรับทุกเป็นตัวเลขจำนวนเต็ม 1 จำนวนแล้วคำนวณหาผลคูณของตัวเลขตั้งแต่ 2 จนถึงตัวเลขที่ รับเข้าโดยวิธีคำสั่ง do-while เท่านั้น
- ข้อที่ 5 จงเขียนโปรแกรมรับตัวเลขสองไปได้นับว่า จำนวนที่รับมาตั้งแต่ 0 ขึ้นถึงสิบหรือสิบสอง กราของจำนวนที่รับมา รับจนจบ

เลือกข้อนี้

ตัวอย่างโค้ด

```
#include<stdio.h>
int main(){
    int a;
    int sum=0;
    do{
        printf("Enter num : ");
        scanf("%d",&a);
        sum+=a;
    }
```

กรณีข้อสอบ

เพื่อไม่แสดงคำถาม เพื่อใช้ตรวจคำตอบของโค้ด

1

2

0

a

a

a

กรณีข้อสอบ

เพื่อไม่แสดงคำถาม เพื่อใช้ตรวจคำตอบของโค้ด

1

2

3

0

b

b

b

เพื่อไม่แสดงคำถาม เพื่อใช้ตรวจคำตอบของโค้ด

1

2

3

4

0

c

c

c

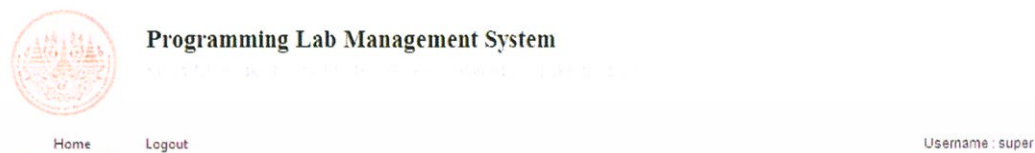
ส่วนที่กรอกผลลัพธ์

```
กรณีที่ 1
Enter num : 1
Enter num : 2
Enter num : 0
Result : 3
กรณีที่ 2
Enter num : 1
Enter num : 2
Enter num : 3
Enter num : 0
Result : 6
กรณีที่ 3
Enter num : 1
Enter num : 2
Enter num : 3
Enter num : 4
Enter num : 0
Result : 10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และข้อมูลในเอกสารฉบับนี้ของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 หน้าต่างรายการโจทย์ที่สามารถเลือก

3) เมื่อกดปุ่มกำหนดโจทย์แล้วเซสชันจะเพิ่มขึ้นมายังรายการคลาสเซสชันโดยสามารถแก้ไขชั้นเรียนในภายหลังได้



รูปที่ 4.10 หน้าต่างรายการคลาสเซสชัน

4.2.5 การตรวจผลลัพธ์และการให้คะแนน

หลังจากที่อาจารย์สร้างชั้นเรียนและกำหนดโจทย์สำหรับนักศึกษาสำเร็จเป็นที่เรียบร้อยแล้ว นักศึกษาจึงสามารถเข้าสู่ระบบเพื่อทำโจทย์

1) หลังจากนักศึกษาเข้าสู่ระบบแล้ว หน้าต่างหลักจะแสดงรายละเอียดของชั้นเรียนให้นักศึกษาทราบถึงการทดลองที่อาจารย์กำหนดให้ แล้วนักศึกษาเลือกรายการทำโจทย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Programming Lab Management System

Copyright © 2013. All rights reserved. | [Privacy Policy](#) | [Terms of Service](#)



ชื่อ: นายณัฐ สัจใจเงิน
ID: E2010000
SEC: 4

[Home](#)

[Logout](#)

Menu

- ทำใจทมิ
- ประวัติการส่งงาน
- ตารางเรียนส่วนบุคคล
- แจ้งปัญหา ข้อเสนอแนะ
- ข้อมูลส่วนตัว

Class Information

ชื่อ Class : Computer and Programming Laboratory

เรื่อง : การเขียนโปรแกรมต่างเงื่อนไข

วันที่ : 13/03/2013

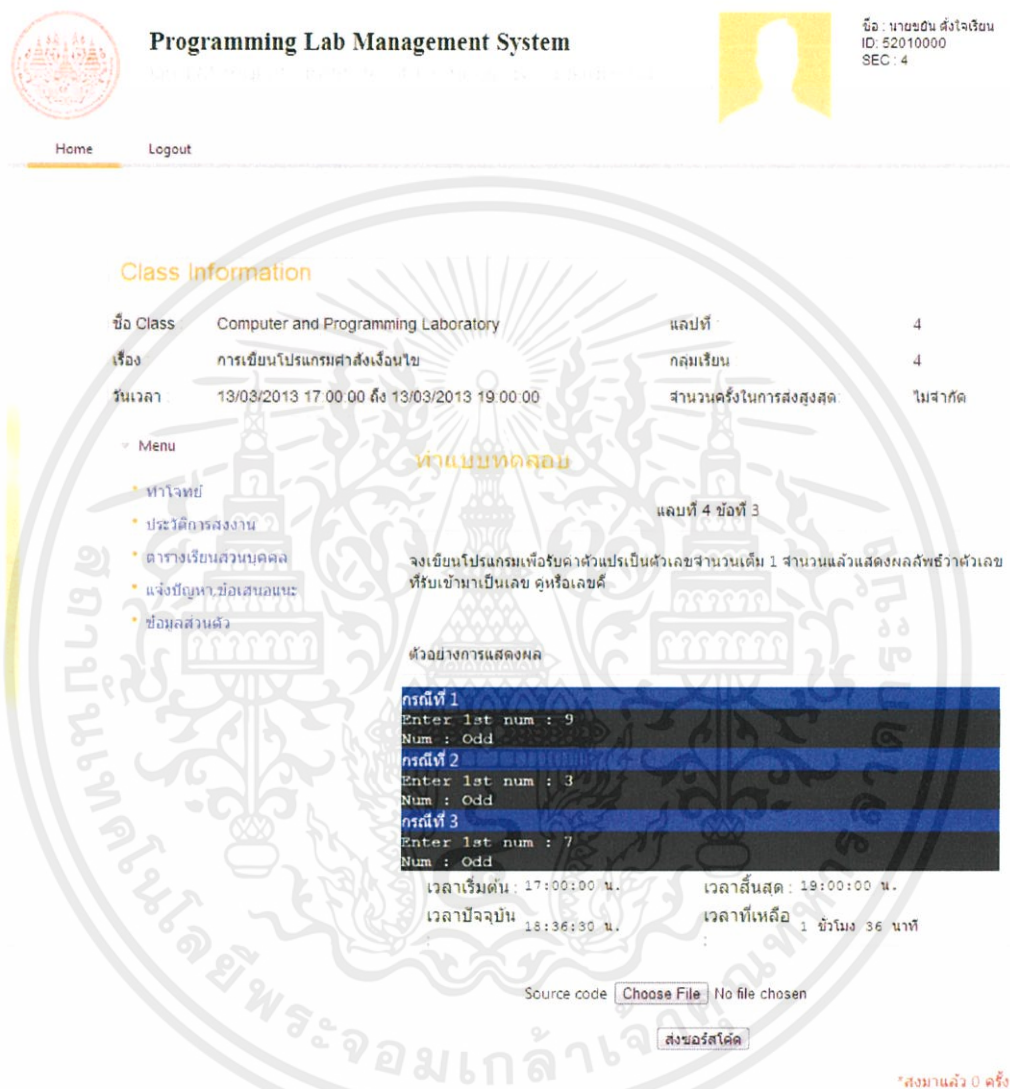
เวลา : 17:00:00 ถึง 19:00:00

ฉบับที่ : 4

รูปที่ 4.11 หน้าต่างหลักของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) หลังจากนักศึกษาเข้ามายังรายการทำโจทย์แล้ว หน้าจอจะแสดงคำสั่งและตัวอย่างผลลัพธ์ให้นักศึกษาทราบ

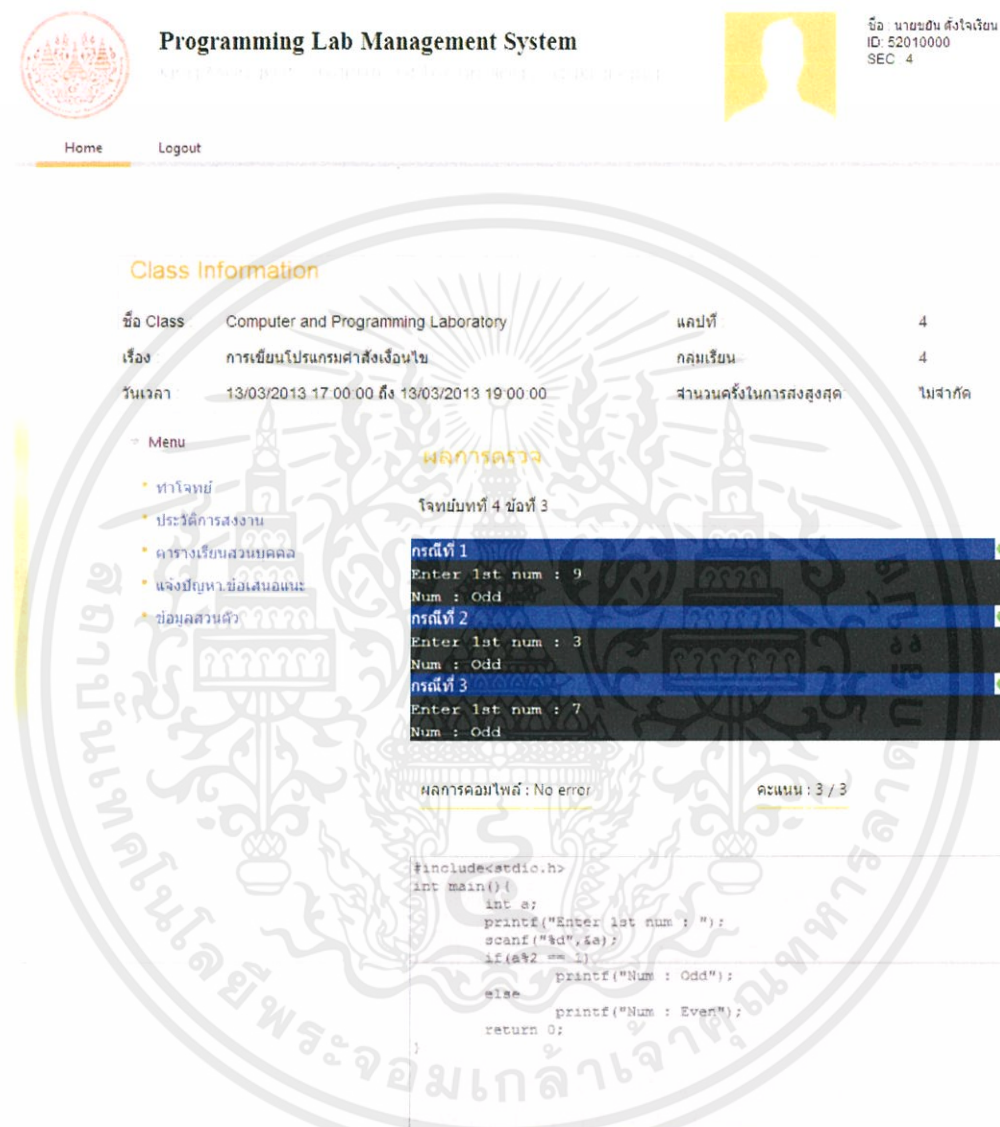


The screenshot displays the 'Programming Lab Management System' interface. At the top, there is a navigation bar with 'Home' and 'Logout' links. The main content area is titled 'Class Information' and shows details for the 'Computer and Programming Laboratory' class, including its schedule and a menu of activities. A specific exercise is highlighted, titled 'หาแบบทดสอบ' (Find test questions), which asks the user to write a program to check if a number is odd or even. The interface includes a 'ตัวอย่างการแสดงผล' (Sample output) section showing a terminal window with three test cases: 9 (Odd), 3 (Odd), and 7 (Odd). Below the terminal, there are fields for 'Source code' and a 'ส่งขอรหัสโค้ด' (Submit code) button. The bottom right corner indicates the user's session status: '*สมาแล้ว 0 ครั้ง'.

รูปที่ 4.12 คำสั่งโจทย์และตัวอย่างผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) หลังจากส่งซอร์สโค้ดไปตรวจ หน้าต่างจะแสดงผลลัพธ์จากโปรแกรม หากกรณีทดสอบใด ให้ผลลัพธ์ถูกต้องจะแสดงเครื่องหมายถูก และหากผิดก็จะแสดงเครื่องหมายกากบาท



The screenshot displays the 'Programming Lab Management System' interface. At the top, there is a logo on the left, the system name in the center, and a user profile icon on the right with the text 'ชื่อ นายธน ดั่งใจเย็น ID: 52010000 SEC 4'. Below the header are 'Home' and 'Logout' links. The main content area is titled 'Class Information' and contains a table with the following data:

ชื่อ Class	Computer and Programming Laboratory	แลปที่	4
เรื่อง	การเขียนโปรแกรมคำสั่งเงื่อนไข	กลุ่มเรียน	4
วันเวลา	13/03/2013 17:00:00 ถึง 13/03/2013 19:00:00	จำนวนครั้งในการส่งสูงสุด	ไม่จำกัด

Below the table is a 'Menu' section with several options, including 'ผลการตรวจ' (Execution Results). The 'ผลการตรวจ' section shows the results of a code execution:

ผลการตรวจ
โจทย์วันที่ 4 ข้อที่ 3

กรณีที่ 1
Enter 1st num : 9
Num : Odd ✓

กรณีที่ 2
Enter 1st num : 3
Num : Odd ✓

กรณีที่ 3
Enter 1st num : 7
Num : Odd ✓

ผลการคอมไพล์ : No error คะแนน : 3 / 3

The code being executed is as follows:

```
#include<stdio.h>
int main(){
    int a;
    printf("Enter 1st num : ");
    scanf("%d",&a);
    if(a%2 == 1)
        printf("Num : Odd");
    else
        printf("Num : Even");
    return 0;
}
```

รูปที่ 4.13 หน้าต่างตรวจผลลัพธ์จากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

5.1 สรุปผลการทดลอง

- 1) สามารถสร้างคลาสเซสชันเพื่อกำหนดเวลาในการทำโจทย์ให้นักศึกษาได้ สามารถเลือกโจทย์ในแต่ละแลบให้นักศึกษาทำได้ตามต้องการ พร้อมทั้งเลือกกรณีทดสอบที่จะใช้เป็นตัวอย่างหรือใช้เพื่อแสดงผลลัพธ์ได้
- 2) โค้ดโปรแกรมที่นักศึกษาส่งเข้าไปตรวจ หากมีการรับอินพุท จำเป็นต้องมีคำนำหน้าอินพุท (input prefix) แต่ไม่จำเป็นต้องเหมือนกับตัวอย่าง และจะมีคำนำหน้าเอาต์พุท (output prefix) หรือไม่ก็ได้แล้วแต่โจทย์กำหนด แต่การรับอินพุททุกครั้งต้องมีเครื่องหมายทวิภาค (:) เสมอเพื่อใช้สำหรับตรวจจับอินพุท
- 3) ทดลองส่งโค้ดโปรแกรมเข้าสู่ระบบ พบว่าระบบสามารถตรวจผลลัพธ์จากโปรแกรมและให้คะแนนได้อย่างถูกต้อง กรณีที่เอาต์พุทของโปรแกรมนักศึกษาเหมือนกันกับเอาต์พุท ของโปรแกรมอาจารย์ หากเอาต์พุทไม่เหมือนกันก็จะไม่ได้คะแนน
- 4) การตรวจผลลัพธ์จะสามารถรับรองผลได้ว่านักศึกษาคิดและเขียนโปรแกรมได้อย่างถูกต้อง หากอาจารย์กำหนดกรณีทดสอบที่มากพอและนักศึกษาได้คะแนนเต็ม แสดงให้เห็นถึงโปรแกรมที่นักศึกษาเขียนมีขั้นตอนวิธีที่ถูกต้อง
- 5) บรรทัดสำหรับป้อนอินพุทหลังคำนำหน้าอินพุทต้องอยู่ติดกันและบรรทัดสำหรับการแสดงคำนำหน้าเอาต์พุทต้องอยู่ติดกัน
- 6) ในการตรวจแบบเฉพาะเอาต์พุทนั้น ถ้าเอาต์พุทแต่ละโปรแกรมมีความแตกต่างกัน อาจารย์จำเป็นต้องกำหนดรูปแบบในการตรวจเอาต์พุทโดยใช้ regular expression ซึ่งทำให้เกิดความยุ่งยากต่ออาจารย์และไม่ครอบคลุมโจทย์ทุกรูปแบบ

5.2 ปัญหา อุปสรรค และแนวทางการแก้ไข

- 1) ระบบไม่สามารถตรวจรูปแบบการเขียนโค้ดโปรแกรมของนักศึกษาได้ แต่จุดประสงค์ของการเขียนโค้ดที่ดี นอกจากได้ผลลัพธ์ที่ถูกต้องแล้ว โค้ดโปรแกรมควรจะมีระเบียบเพื่อจะได้อ่านทำความเข้าใจได้ง่าย ดังนั้นควรมีการตรวจรูปแบบการเขียนโค้ดโปรแกรมเพื่อฝึกให้นักศึกษาเป็นนักพัฒนาโปรแกรมผู้มีทัศนคติที่ดีต่อการพัฒนาโปรแกรม
- ผู้พัฒนาโปรแกรม มิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ระบบยังไม่สามารถตรวจโค้ดโปรแกรมที่มีข้อห้ามหรือเงื่อนไขเพิ่มเติมได้ เช่น ต้องใช้ฟังก์ชัน (function), ให้ใช้ตัวแปรประเภท pointer เท่านั้น เป็นต้น ดังนั้นระบบควรเพิ่มความสามารถโดยใช้ Lexical analyzer ในการวิเคราะห์โครงสร้างของโปรแกรมหรือไวยากรณ์ของโปรแกรม

3) เนื่องจากโปรแกรมมีผลลัพธ์หลากหลายรูปแบบมาก ทำให้ระบบตรวจได้ไม่ครอบคลุมทุกโจทย์ ดังนั้นควรเพิ่มความสามารถในการตรวจรูปแบบผลลัพธ์ที่หลากหลายมากขึ้น

4) รหัสผ่านของนักศึกษาโดยค่าเริ่มต้นจะเหมือนกับ Username ซึ่งง่ายต่อการสร้างรหัสผ่านให้นักศึกษา แต่สามารถถูกบุกรุกได้ง่าย ดังนั้นควรเปลี่ยนรหัสผ่านเริ่มต้นของนักศึกษาเป็นชุดตัวอักษรที่สุ่มขึ้นมา

5.3 แนวทางในการพัฒนาต่อ

1) สามารถตรวจได้นอกเหนือจากโค้ดโปรแกรมภาษาซี เช่น ภาษาซีพลัสพลัส, ภาษาจาวา ดังนั้นจึงอาจมีการติดตั้งคอมไพเลอร์หรือเขียนโปรแกรมเพิ่มให้แก่ระบบให้มีความสามารถมากยิ่งขึ้น

2) สามารถตรวจรูปแบบการเขียนโค้ดโปรแกรมเพื่อฝึกให้นักศึกษาเป็นผู้พัฒนาโปรแกรมที่ดีต่อไปในอนาคตได้

3) ระบบสามารถใช้ Lexical analyzer ในการวิเคราะห์โครงสร้างของโปรแกรมหรือไวยากรณ์ของโปรแกรมได้

4) เพิ่มความสามารถในการตรวจรูปแบบผลลัพธ์ที่หลากหลายมากขึ้นได้ เช่น กรณีที่ผลลัพธ์ของแต่ละบุคคลไม่จำเป็นต้องเหมือนกัน ซึ่งโจทย์วัดการจัดรูปแบบการแสดงผลของผลลัพธ์ จึงอาจมีผลลัพธ์ที่แตกต่างกันไปได้

5) สามารถเรียกดูประวัติ การทำแบบฝึกหัดย้อนหลังของนักศึกษาแต่ละบุคคลได้ เพื่อดูพัฒนาการในการพัฒนาโปรแกรมของนักศึกษาแต่ละบุคคลได้

6) ในการตรวจแบบเฉพาะเอชท์พูนั้น ถ้าเอชท์พูนแต่ละโปรแกรมมีความแตกต่างกัน อาจารย์จำเป็นต้องกำหนดรูปแบบในการตรวจเอชท์พูนโดยใช้ regular expression ซึ่งทำให้เกิดความยุ่งยากต่ออาจารย์และไม่ครอบคลุมโจทย์ทุกรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1]อภิเนตร อุนากุล. 2546. กระบวนการพัฒนาเว็บแอปพลิเคชันโดยใช้ UML. กรุงเทพฯ :
แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [2]วศิน เพิ่มทรัพย์ และวิโรจน์ ชัยมูล. 2548. ความรู้เบื้องต้นเกี่ยวกับคอมพิวเตอร์และเทคโนโลยี
สารสนเทศ. กรุงเทพฯ : โปรวีชั่น.
- [3]อดิศักดิ์ จันทร์มิน. 2548. สร้าง Web application อย่างมืออาชีพด้วย PHP ฉบับ Workshop.
กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- [4]อาณัติ รัตนธิรกุล. 2549. ก้าวสู่อำชีผู้ดูแลระบบเครือข่ายคอมพิวเตอร์ในองค์กร (ภาคปฏิบัติ).
กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- [5]สุชาติ คุ่มมะณี และธวัชชัย ชมศิริ. 2550. เรียนรู้เครือข่ายและอุปกรณ์ Cisco ด้วยโปรแกรม
Simulation. กรุงเทพฯ : โปรวีชั่น.
- [6]บัญชา ปะสีละเตสัง. 2555. พัฒนาเว็บแอปพลิเคชันด้วย PHP ร่วมกับ MySQL และ
Dreamweaver. กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- [7]2012. "Model-View-Controller" [Online].
Available : <http://msdn.microsoft.com/en-us/library/ff649643.aspx>
- [8]Refsnes Data. 1999. PHP Tutorial. [Online].
Available : <http://www.w3schools.com/php/default.asp>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การติดตั้ง AppServ

ก.1 ความหมายของโปรแกรม AppServ

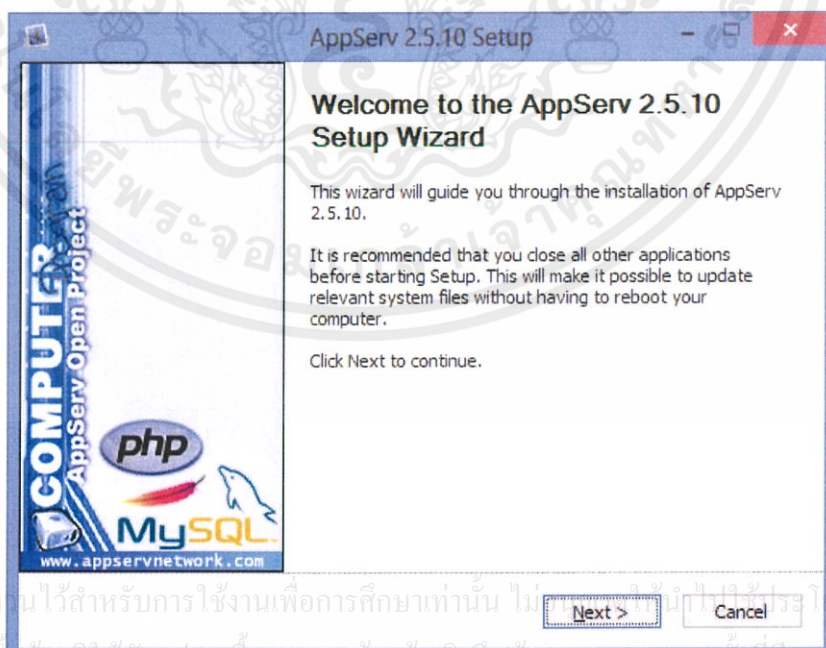
AppServ คือโปรแกรมที่รวบรวมเอา Open Source Software หลากๆ อย่างมารวมกัน โดยมีPackage หลักดังนี้

- 1) Apache ใช้สำหรับเป็น Web server
- 2) PHP ใช้สำหรับเป็นหน่วยประมวลผลการทำงานของภาษา PHP
- 3) MySQL ใช้สำหรับเป็น Database server
- 4) phpMyAdmin ใช้สำหรับเป็น Database manager

ก.2 ขั้นตอนการติดตั้ง

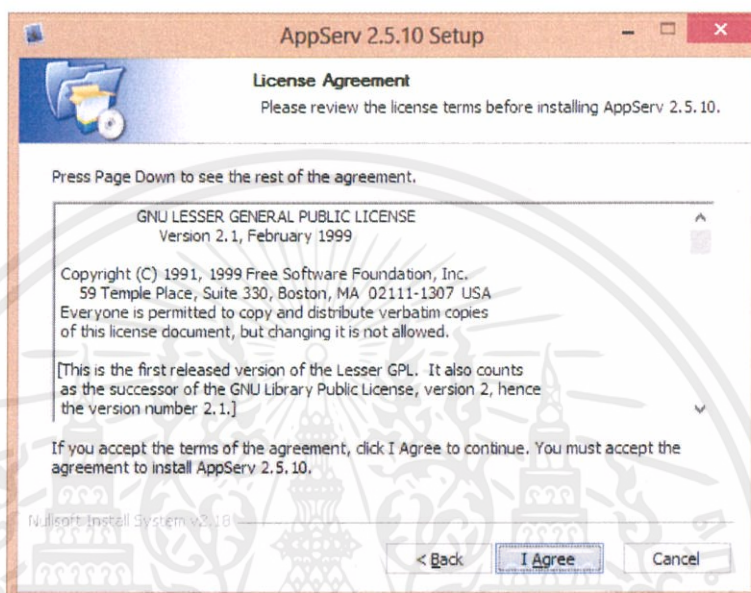
ดาวน์โหลดโปรแกรม AppServ จากเว็บไซต์ <http://www.appservnetwork.com> โดยเลือกเวอร์ชันที่ต้องการติดตั้งระหว่างเวอร์ชัน 2.4.x และ 2.5.x ขั้นตอนการติดตั้งมีดังต่อไปนี้

- 1) ดับเบิลคลิกไฟล์ appserv-win32-x.x.x.exe เพื่อทำการติดตั้ง



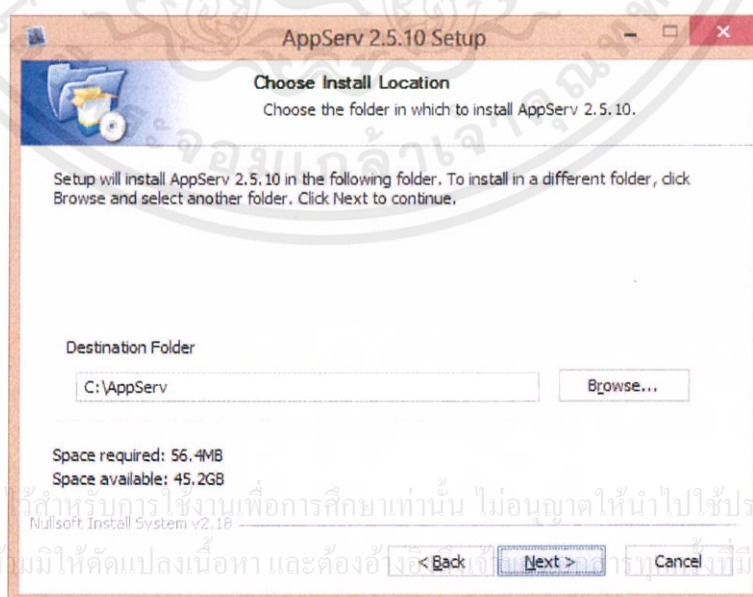
รูปที่ ก.1 การติดตั้งโปรแกรม AppServ

2) เข้าสู่ขั้นตอนเงื่อนไขการใช้งานโปรแกรม เมื่ออ่านเงื่อนไขต่างๆ เสร็จสิ้นแล้ว หากยอมรับเงื่อนไขให้กด Next เพื่อเข้าสู่การติดตั้งในขั้นต่อไป หากไม่ยอมรับให้กด Cancel เพื่อออกจากกรติดตั้งโปรแกรม



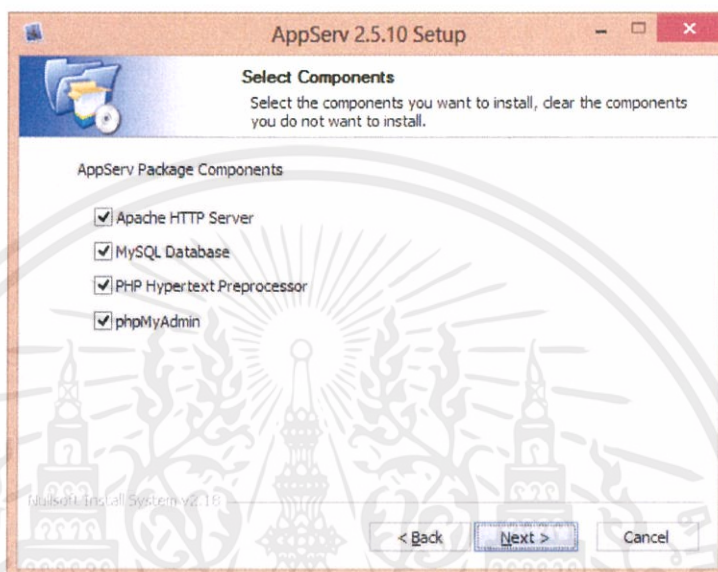
รูปที่ ก.2 เงื่อนไขการติดตั้ง

3) เลือกปลายทางที่ต้องการติดตั้ง โดยค่าเริ่มต้นปลายทางที่ติดตั้งจะเป็น C:\AppServ หากต้องการเปลี่ยนปลายทางที่ติดตั้ง ให้กด Browse แล้วเลือกปลายทางที่ต้องการ จากนั้นกด Next



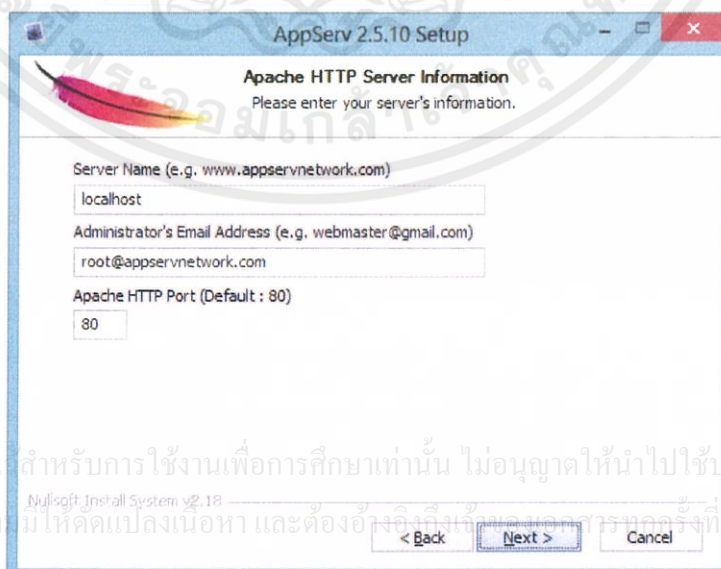
รูปที่ ก.3 เลือกปลายทางที่ต้องการติดตั้ง

4) เลือก Package Components ที่ต้องการติดตั้ง โดยค่าเริ่มต้นนั้นจะให้เลือกลงทุก Package แต่หากว่าผู้ใช้งานต้องการเลือกเฉพาะบาง Package ก็สามารถเลือกตามข้อที่ต้องการออก เมื่อทำการเลือก Package เรียบร้อยแล้ว ให้กด Next



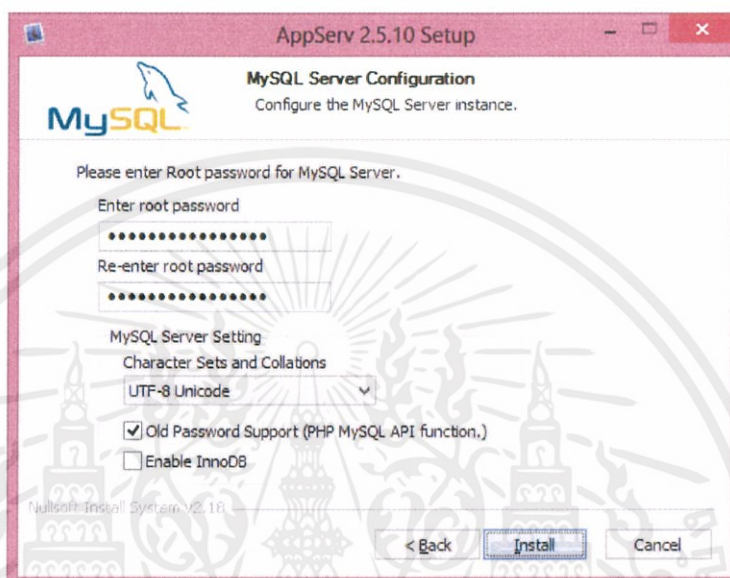
รูปที่ ก.4 การเลือก Package ที่ต้องการติดตั้ง

5) กำหนดค่าคอนฟิกของ Apache Web Server มีอยู่ด้วยกันทั้งหมด 3 ส่วน ตามรูปที่ 5 คือ Server Name คือช่องสำหรับป้อนชื่อ Web Server ของท่าน Admin Email คือช่องสำหรับป้อนข้อมูล อีเมลผู้ดูแลระบบ HTTP Port คือช่องสำหรับระบุ Port ที่จะเรียกใช้งาน



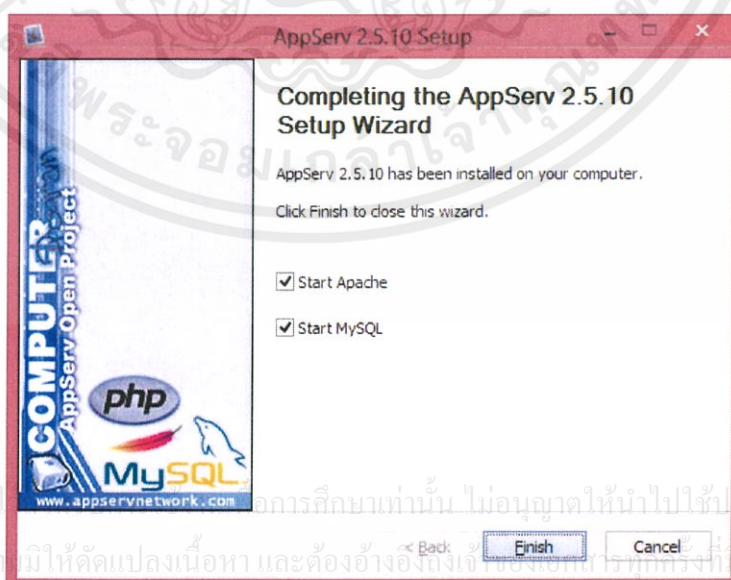
รูปที่ ก.5 การกำหนดค่า Config Apache Server

6) กำหนดค่าคอนฟิกของ MySQL Database คือ Root Password คือช่องสำหรับป้อนรหัสผ่านการใช้งานฐานข้อมูลให้ระบุ user คือ root, Character Sets ใช้ในการกำหนดค่าระบบภาษาที่ใช้ในการติดต่อฐานข้อมูล, Enable InnoDB หากต้องการใช้งานฐานข้อมูลในรูปแบบ InnoDB



รูปที่ ก.6 การกำหนดค่า Config MySQL Database

7) สิ้นสุดขั้นตอนการติดตั้งโปรแกรม AppServ กดปุ่ม Finish เพื่อเสร็จสิ้นการติดตั้งโปรแกรม



รูปที่ ก.7 การสิ้นสุดการติดตั้งโปรแกรม Appserv

ภาคผนวก ข

Direction Lab

ข.1 เริ่มเข้าใช้งานและการแก้ไขข้อมูลส่วนตัว

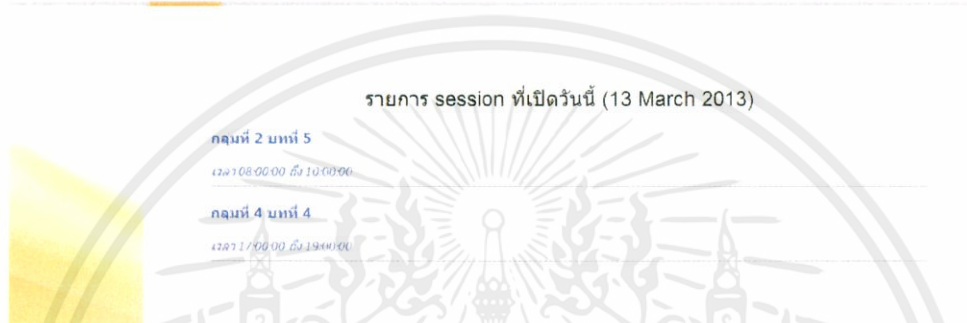
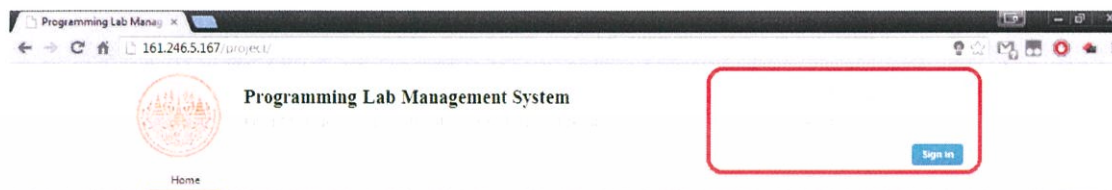
- 1) เปิดเว็บเบราว์เซอร์แล้วพิมพ์ xxx.xxx.xxx.xxx/project ที่ช่อง Address เพื่อเข้าระบบจัดการบริหารห้อง Lab Programming



รูปที่ ข.1 ใส่ ip ของเครื่อง server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) คลิกที่ปุ่ม Sign In ที่มุมขวาบน แล้วทำการ login โดยใช้ username และ password เป็นรหัสนักศึกษาของตนเอง จากนั้น คลิก Sign In



รูปที่ ข.2 ใส่ username และรหัสลงไป

3) คลิกที่ menu แก้ไขข้อมูลส่วนตัว ทำการแก้ไขข้อมูลส่วนตัว ชื่อ รหัสผ่าน เบอร์โทร E-mail เมื่อเสร็จให้ใส่รหัสผ่านเดิม แล้วกด แก้ไข

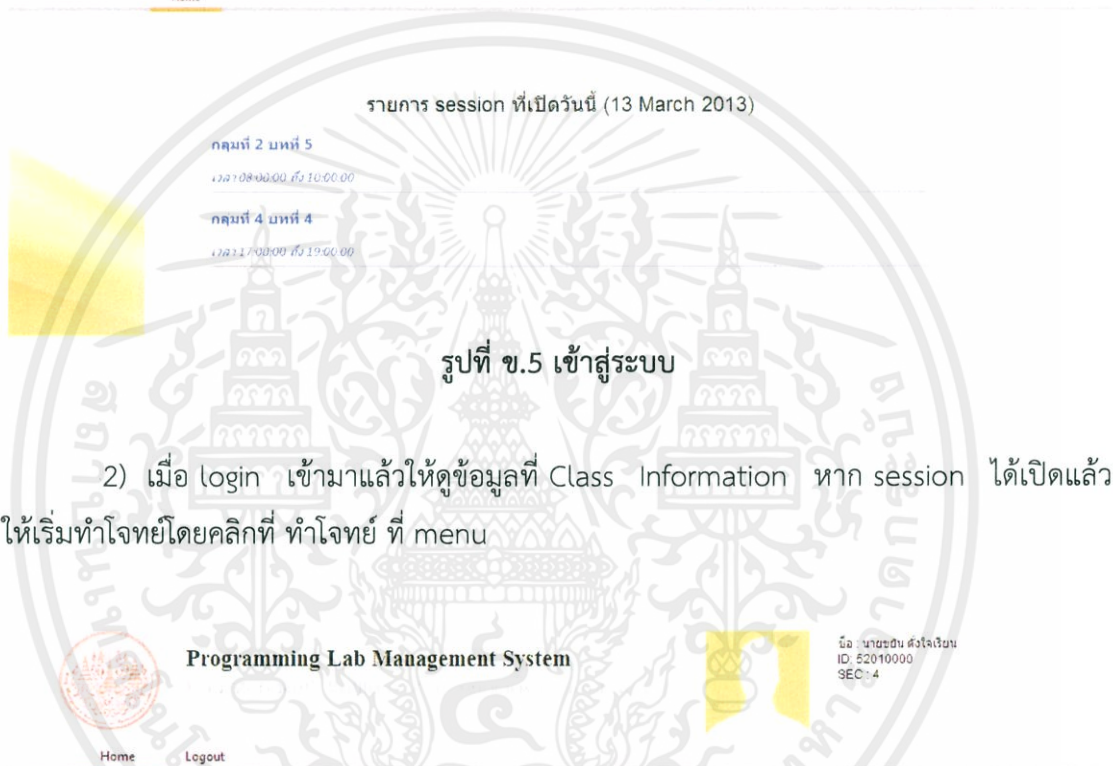
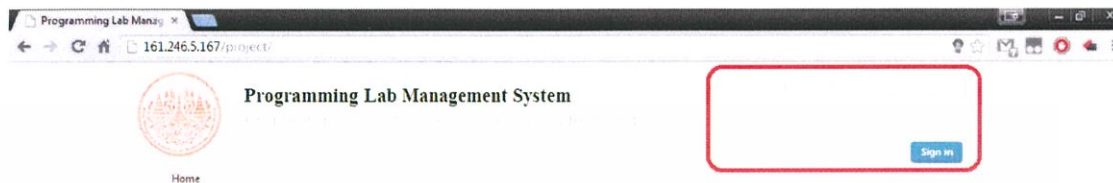


รูปที่ ข.3 การแก้ไขข้อมูลส่วนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

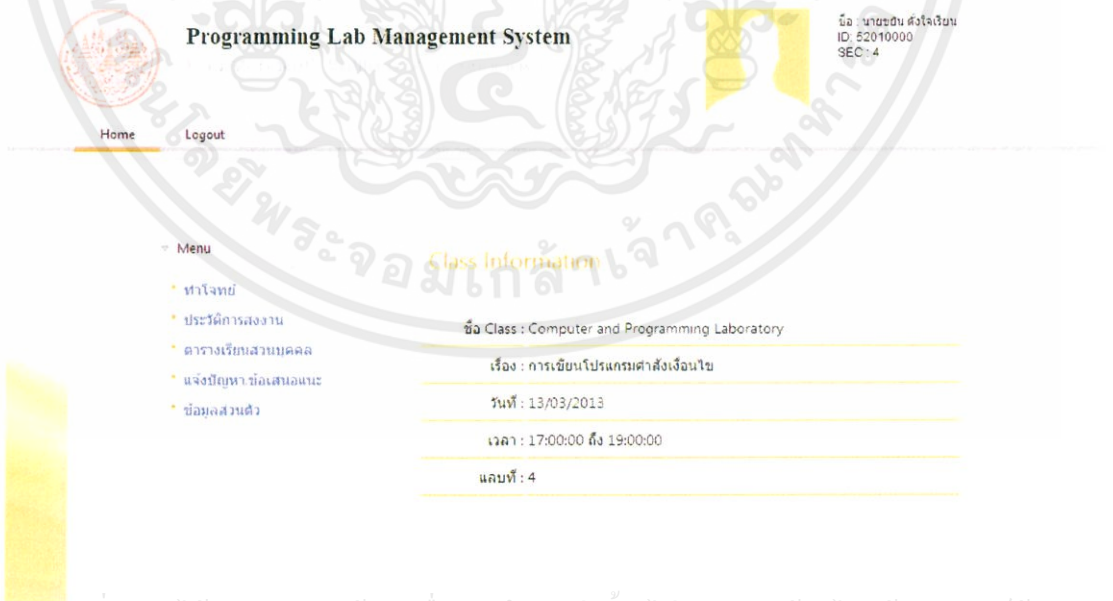
ข.2 การทำโจทย์

1) ทำการ login ด้วย username และ password ของตนเอง



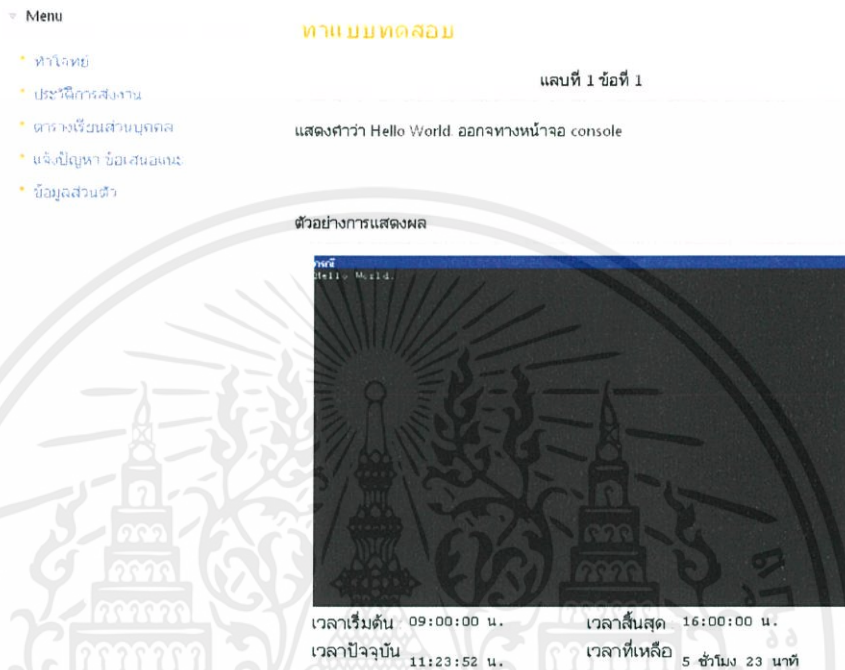
รูปที่ ข.5 เข้าสู่ระบบ

2) เมื่อ login เข้ามาแล้วให้ดูข้อมูลที่ Class Information หาก session ได้เปิดแล้ว ให้เริ่มทำโจทย์โดยคลิกที่ ทำโจทย์ ที่ menu



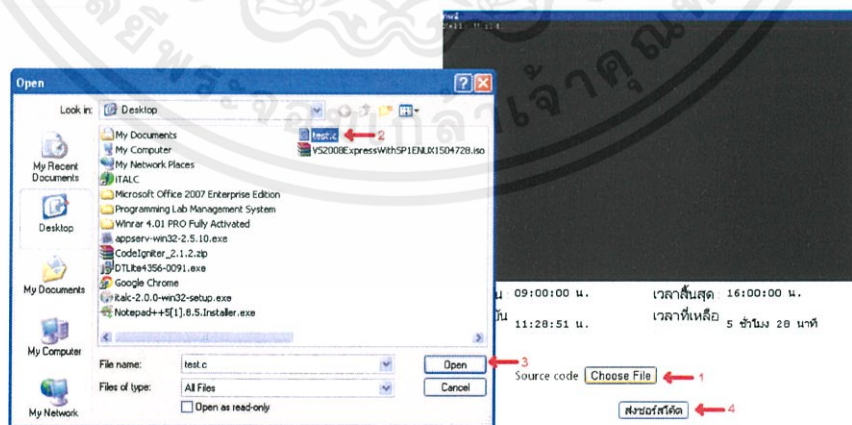
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ ข.6 การทำโจทย์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) อ่านโจทย์ให้เข้าใจ และดูรูปแบบในการแสดงผล จากนั้นทดลองทำโจทย์โดยใช้ VisualStudio2010



รูปที่ ข.7 ตัวอย่างการแสดงผล

4) เมื่อทำโจทย์เสร็จและตรวจสอบมั่นใจว่าถูกต้องแล้ว ให้ทำการ upload source code เพื่อให้ระบบตรวจ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่รูปที่ ข.8 การส่งไฟล์ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เมื่อทำการ upload เสร็จเรียบร้อยแล้ว ระบบจะทำการตรวจแล้วรายงานผลการตรวจ

ผลการตรวจ

โจทย์บทยที่ 1 ข้อที่ 1

วันที่ตรวจ	: 23/08/2012	เวลาตรวจ	: 12:35:57
ผลการคอมไพล์	: No error	คะแนน	: 0 / 1
เวลาที่ใช้	: 0.001 s	หน่วยความจำที่ใช้	: 12.8 MB/s

```
#include<stdio.h>
void main()
{
    printf("Hello World.");
}
```

รูปที่ ข.9 ผลการตรวจ

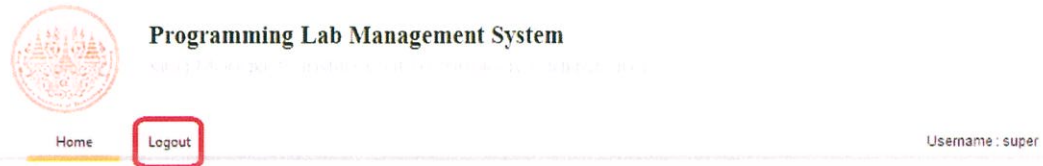
6) ตรวจสอบประวัติการส่งงานได้ที่คำสั่ง menu

Menu		ประวัติการส่งงาน					
แรม	ชื่อไฟล์	คะแนน	วันที่ส่ง	เวลาที่ส่ง	ผลการคอมไพล์	IP address	
1	test3.c	0/1	23/08/2012	12:32:10	No error	127.0.0.1	
1	test4.c	0/1	23/08/2012	12:35:57	No error	127.0.0.1	

รูปที่ ข.10 ประวัติการส่งงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) เมื่อเสร็จเรียบร้อยแล้วให้ทำการ logout ทุกครั้ง



รูปที่ ข.11 ออกจากระบบ



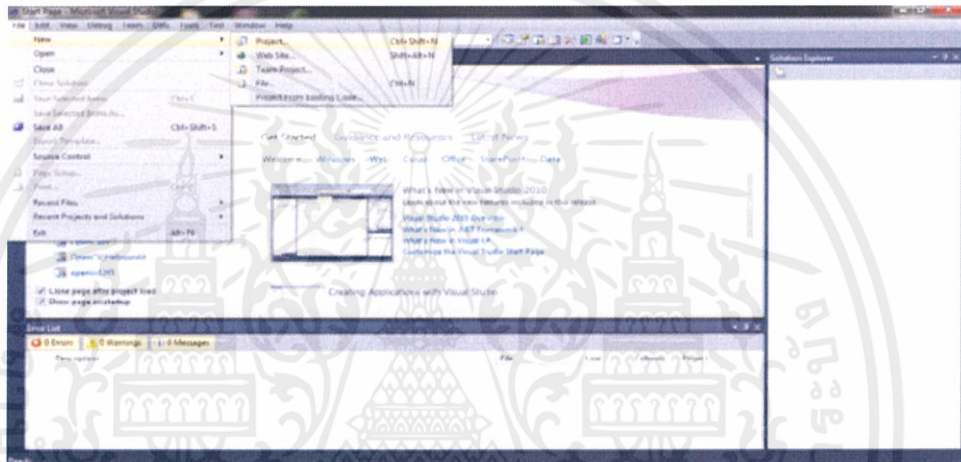
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

การใช้งานโปรแกรม Visual Studio 2010

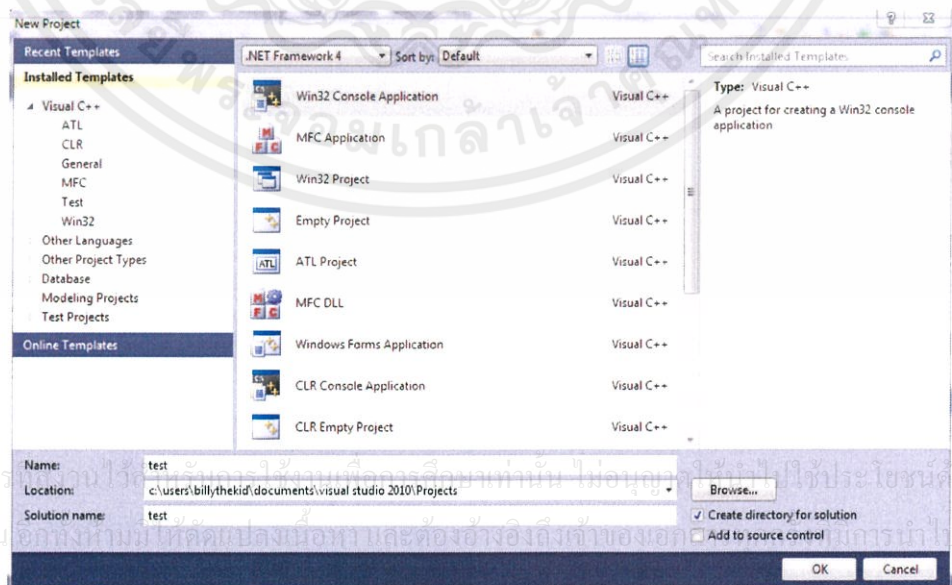
วิธีใช้ Visual Studio 2010 จำเป็นที่จะต้องสร้างโปรเจกต์และตั้งค่าดังนี้

1) คลิกที่ File -> New -> Project



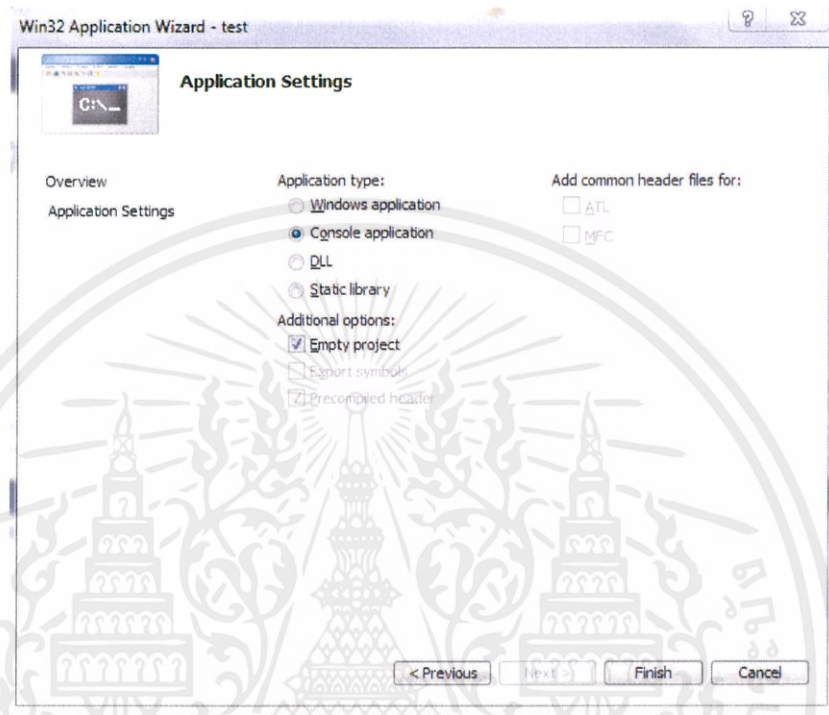
รูปที่ ค.1 การสร้างโปรเจกต์

2) เลือก Win32 Console Application แล้วตั้งชื่อโปรเจกต์ และคลิก OK



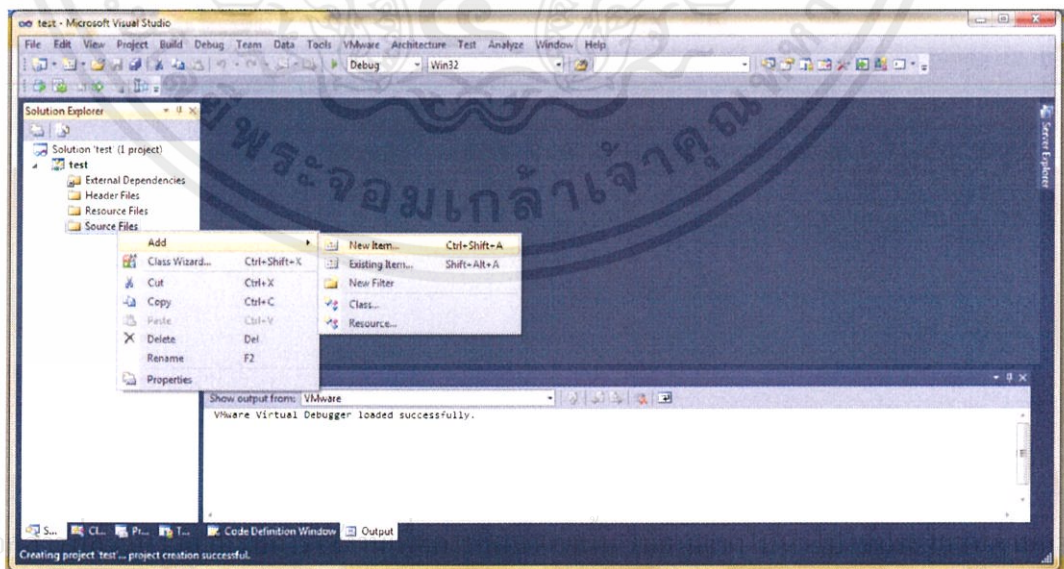
รูปที่ ค.2 ตั้งชื่อโปรเจกต์

3) Application Settings คลิกปุ่ม Next : Application type เลือก console Application ส่วน Additional options เลือก Empty Project แล้วคลิกที่ปุ่ม Finish



รูปที่ ค.3 ตั้งค่าโปรเจค

4) คลิกขวาที่ Source Files -> Add -> New Item



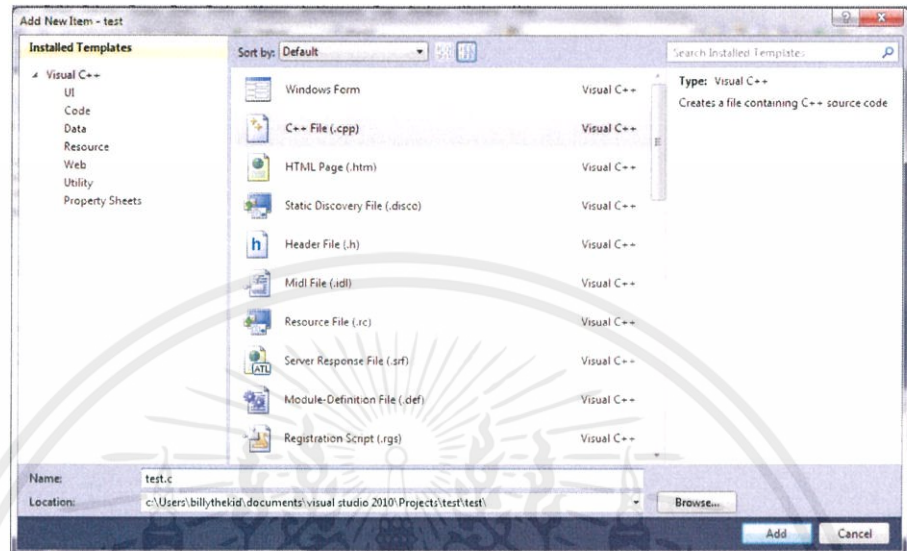
รูปที่ ค.4 เพิ่ม source file

เอกสารนี้เป็น

รค้า

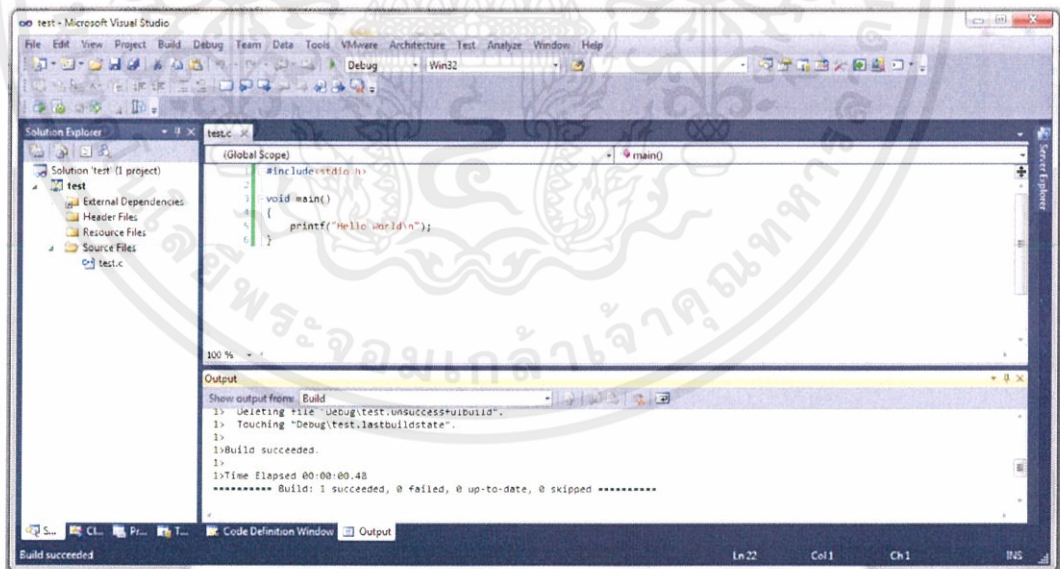
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อรูปที่ ค.4 เพิ่ม source file เอกสารทุกครั้งที่มีการนำไปใช้

5) เลือก C++ File(.cpp) -> ตั้งชื่อไฟล์ (สำหรับภาษา C ใช้ .c) -> Add



รูปที่ ค.5 เลือกชนิดของ source file

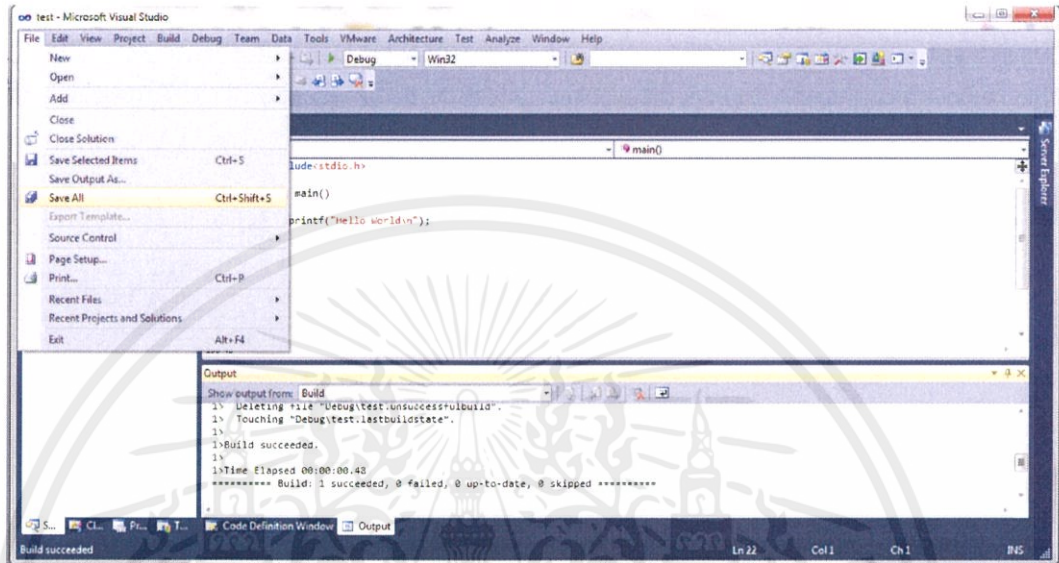
6) ทำการเขียนโปรแกรมตามที่โจทย์สั่ง



รูปที่ ค.6 เขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) Save Project โดยเลือกเมนู File -> Save All หรือ คีย์ลัด Ctrl+Shift+S ซึ่งตอน Save จะไปอยู่ที่ Folder ชื่อ Visual Studio 2010 ที่ Documents Library



รูปที่ ค.7 บันทึกโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้