



ระบบสารสนเทศการเงินสินเชื่อเพื่อรถจักรยานยนต์

## INFORMATION SYSTEM OF LOAN FOR MOTORCYCLES



นายพีรณย์ เมฆอับ รหัส 44370278

นายศักดิ์สิทธิ์ ชันสาคร รหัส 44370419

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 25 พ.ค. 2553 .....
เลขทะเบียน..... 15004149 .....
เลขเรียกหนังสือ..... 15 .....
มหาวิทยาลัยนเรศวร
2548

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร


ปีการศึกษา 2548




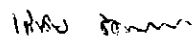
## ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ      ระบบการเงินสินเชื่อ  
ผู้เสนอโครงการ      นายพีรณย์      เมฆอับ      รหัส 44370278  
   นายศักดิ์สิทธิ์      ชันสาคร      รหัส 44370419  
อาจารย์ที่ปรึกษา      ดร.สุรเชษฐ์      กานต์ประชา  
สาขาวิชา      วิศวกรรมคอมพิวเตอร์  
ภาควิชา      วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
ปีการศึกษา      2548

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะกรรมการสอบโครงการวิศวกรรม

  
..... ประธานกรรมการ  
(ดร.สุรเชษฐ์ กานต์ประชา)

  
..... กรรมการ  
(ดร.พนมขวัญ ธิยะมงคล)

  
..... กรรมการ  
(นายแสงชัย มังกรทอง)

หัวข้อโครงการ	ระบบสารสนเทศการเงินสินเชื่อเพื่อรถจักรยานยนต์		
ผู้ดำเนินโครงการ	นายพีรคนย์	เมฆอับ	รหัส 44370278
	นายศักดิ์สิทธิ์	ขันสาคร	รหัส 44370419
อาจารย์ที่ปรึกษา	ดร.สุรเชษฐ์ กานต์ประชา		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

### บทคัดย่อ

โครงการระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์ จัดทำขึ้นเพื่อออกแบบและพัฒนา ระบบการเงินสินเชื่อ ให้มีความสะดวกในการจัดเก็บข้อมูล ค้นหาข้อมูลและการคำนวณทางด้านการเงิน ช่วยเพิ่มประสิทธิภาพการจัดเก็บข้อมูลให้ดียิ่งขึ้น

ในส่วนของ การออกแบบเว็บไซต์ใช้ภาษาPHP(Personal Home Page)ในการเขียน โปรแกรมระบบสารสนเทศการเงินสินเชื่อเพื่อรถจักรยานยนต์ ประกอบด้วย Dreamweaver MX ช่วยจัดตกแต่งหน้าตาเว็บเพจ และ Edit Plus เป็นเครื่องมือจัดการโค้ดของโปรแกรมที่จัดทำขึ้น ส่วนการจัดเก็บข้อมูลนั้นใช้ MySQL เป็นเครื่องมือในการจัดการระบบฐานข้อมูล ซึ่งข้อมูลที่จัดเก็บเป็นรายละเอียดต่างๆของเอกสารการทำสัญญาสินเชื่อ

ผลลัพธ์ที่ได้จากโครงการนี้ คือ ระบบซึ่งเป็นโปรแกรมที่ช่วยในการทำสัญญาสินเชื่อ ให้มีความสะดวกขึ้น เนื่องจากโปรแกรมสามารถ คำนวณทางด้านการเงิน เตือนวันใกล้หมดอายุของ ทะเบียนรถจักรยานยนต์ เก็บประวัติต่างเกี่ยวกับลูกค้าจึงง่ายต่อการสืบค้นและแก้ไข ทั้งยังสรุป ยอดการทำสัญญาสินเชื่อ การชำระเงินและยอดค้างชำระ เพื่อให้เจ้าของกิจการพิจารณาการอนุมัติ สินเชื่อในครั้งต่อไป

**Project Title** INFORMATION SYSTEM OF LOAN FOR MOTORCYCLES  
**Name** Mr.Peeradon Mek\_ub ID 44370278  
Mr.Saksit kansakon ID 44370419  
**Project Advisor** Surachet Kanprachar, Ph.D.  
**Major** Computer Engineering  
**Department** Electrical and Computer Engineering  
**Academic Year** 2548

---

### ABSTRACT

The purpose of this project, Information system of loan for motorcycles, is to design, to develop financial credit system in order to make the process of collecting data more conveniently, and to calculate the amount of money need to be paid for the customer efficiently.

To develop a web page, in this project, PHP (Personal Home Page) language is adopted. Moreover, in order to make the web page more interesting, Dreamweaver MX is used as a tool to help decorating the web page. Edit Plus is used for managing code system of program. For database part, MySQL is adopted.

The result of this project is to help in making a loan for motorcycle much easier since it can calculate the amount of money to be paid in every month. Also, the profile of customers can be easily retrieved from the database and the details of motorcycle can be used more efficiently; for example, the system will automatically warn if any of the motorcycle registration is about to be expired.

## กิตติกรรมประกาศ

ผู้จัดทำขอกราบขอบพระคุณ อาจารย์สุรเชษฐ์ กานต์ประชา ซึ่งเป็นอาจารย์ที่ปรึกษา  
โครงการ ตามที่ท่านได้ให้ความกรุณาแนะนำวิธีในการทำงาน ให้เข้าใจถึงการศึกษาอย่างเป็น  
ระบบขั้นตอน อีกทั้งสละเวลาเพื่อตรวจสอบการทำงานและชี้แนวทางแก้ไขในทุกขั้นตอนตลอด  
การศึกษา

ขอขอบพระคุณ คุณลักขณา คงเทศ พนักงานบริษัทสินเชื่อที่ให้ความรู้เกี่ยวกับการจัดทำ  
สินเชื่อ เพื่อช่วยให้คำปรึกษาโครงการระบบสารสนเทศสินเชื่อเพื่อรถจักรยานยนต์ในการทำ  
โครงการครั้งนี้

ขอขอบพระคุณอาจารย์ทุกท่านที่ช่วยให้คำปรึกษาเกี่ยวกับปัญหาต่างในการจัดทำโครงการ  
ระบบสารสนเทศสินเชื่อเพื่อรถจักรยานยนต์

นายพีรณย์ เมฆอับ  
นายศักดิ์สิทธิ์ ชันสาคร



## สารบัญ

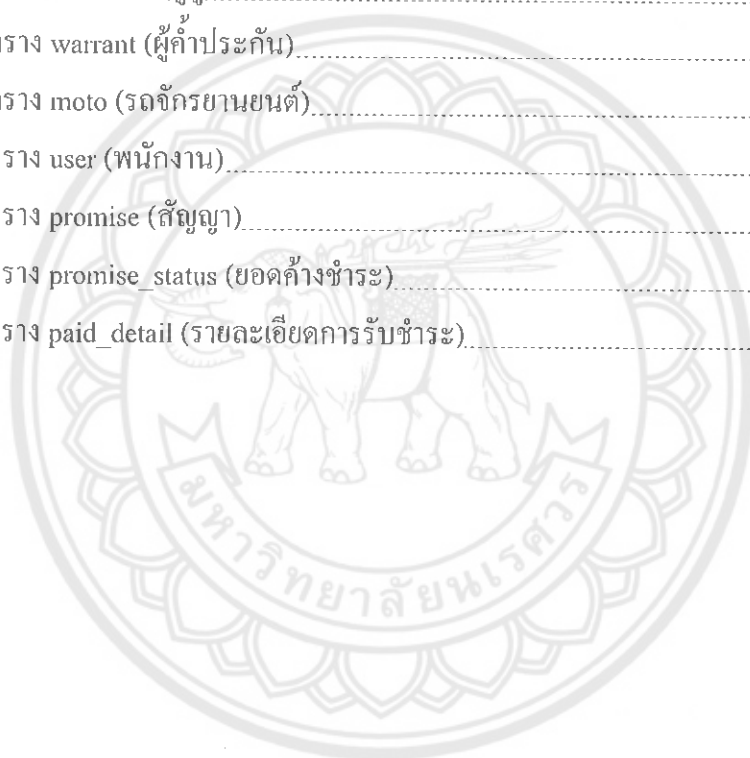
	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
<b>บทที่ 1 บทนำ</b>	
1.1 หลักการและเหตุผล.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบข่ายของโครงการ.....	1
1.4 ขั้นตอนการดำเนินโครงการ.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	2
1.6 รายละเอียดงบประมาณตลอดโครงการ.....	2
<b>บทที่ 2 หลักการและทฤษฎี</b>	
2.1 ทฤษฎีการคำนวณ.....	3
2.2 ภาษา PHP.....	3
2.3 ทำไมภาษา PHP น่าสนใจและน่าใช้.....	4
2.4 ประวัติ PHP.....	4
2.5 เรื่องของตัวแปรและคำสั่งพื้นฐาน.....	5
2.6 ฟังก์ชันเกี่ยวกับสตริง.....	15
2.7 ฟังก์ชันเกี่ยวกับวันและเวลา.....	16
2.8 การสร้างฟังก์ชันไว้ใช้เอง.....	16
2.9 ขอบเขตของตัวแปร (Variable Scope).....	18
2.10 การสร้างตัวแปรแบบ static.....	19
2.11 การรับส่งข้อมูลจากผู้ใช้.....	20
2.12 MYSQL.....	27

## สารบัญ (ต่อ)

	หน้า
2.13 ทำไมถึงเลือกใช้ MYSQL.....	27
2.14 การสร้างฐานข้อมูลด้วย MYSQL.....	27
2.15 Macromedia Dreamweaver MX.....	36
<b>บทที่ 3 วิธีดำเนินการ โครงการ</b>	
3.1 การศึกษาปัญหา.....	38
3.2 รวบรวมข้อมูลและวิเคราะห์ข้อมูล.....	38
3.3 นำข้อมูลและผลการวิเคราะห์มาออกแบบฐานข้อมูล.....	39
3.4 ออกแบบหน้าตาเว็บเพจ.....	40
3.5 นำโปรแกรมที่ทำเสร็จแล้วไปทดลองใช้จริง.....	41
3.6 สรุปผลการทดลอง.....	41
<b>บทที่ 4 ผลการดำเนิน โครงการ</b>	
4.1 ออกแบบระบบการทำงาน.....	42
4.2 ออกแบบฐานข้อมูล.....	44
4.3 ออกแบบหน้าตาเว็บเพจ.....	53
4.4 การเขียน PHP ติดต่อกับฐานข้อมูล.....	56
4.5 ผลการทดลองใช้.....	57
<b>บทที่ 5 บทสรุป</b>	
5.1 สรุปปัญหาที่เกิดขึ้นในการทดลอง.....	59
5.2 ข้อเสนอแนะและแนวทางดำเนินงานต่อ.....	59
เอกสารอ้างอิง.....	60
ภาคผนวก.....	61
ประวัติผู้เขียนโครงการ.....	71

## สารบัญตาราง

ตารางที่	หน้า
1.1 การดำเนินโครงการ.....	2
2.1 ตัวดำเนินการทางตรรกะ(Logical Operators).....	10
2.2 ตัวดำเนินการเชิงเปรียบเทียบ.....	10
2.3 ฟังก์ชันเกี่ยวกับวันและเวลา.....	16
2.4 แสดงตาราง Operator.....	32
4.1 ตาราง customer (ผู้กู้).....	45
4.2 ตาราง warrant (ผู้ค้ำประกัน).....	46
4.3 ตาราง moto (รถจักรยานยนต์).....	47
4.4 ตาราง user (พนักงาน).....	48
4.5 ตาราง promise (สัญญา).....	48
4.6 ตาราง promise_status (ยอดค้างชำระ).....	49
4.7 ตาราง paid_detail (รายละเอียดการรับชำระ).....	50





## สารบัญรูป

รูปที่	หน้า
2.1 แสดงการรับข้อมูลชนิดต่างๆ .....	6
2.2 แสดงผลรันชนิดข้อมูลอาร์เรย์.....	6
2.3 แสดงการรับการแปลงชนิดข้อมูล .....	7
2.4 แสดงการรับตัวดำเนินการกำหนดค่า.....	9
2.5 แสดงการรับตัวดำเนินการสตริง .....	11
2.6 แสดงการรับตามเงื่อนไข if .....	12
2.7 แสดงการรับตามเงื่อนไข else .....	13
2.8 แสดงการรับรูปแบบคำสั่ง for.....	14
2.9 แสดงการรับคำสั่ง do...while.....	15
2.10 แสดงคำสั่งการรับฟังก์ชัน explode และ implode .....	16
2.11 แสดงการรับฟังก์ชันที่สร้างไว้ใช้เอง.....	18
2.12 แสดงการรับฟังก์ชันขอบเขตของตัวแปร .....	19
2.13 แสดงการรับตัวแปรแบบ static.....	20
2.14 แสดงช่องกรอก password.....	21
2.15 แสดงการรับช่องรับรหัส.....	22
2.16 แสดงผลการรับการ (Check Box).....	23
2.17 แสดงผลการรับปุ่มตัวเลือก (Radio Button).....	24
2.18 แสดงผลการรับลิสต์บ็อกซ์ (List Box).....	25
2.19 แสดงผลการรับการใช้ form ในการส่งค่า.....	26
2.20 แสดงการสร้างฐานข้อมูล.....	27
2.21 แสดงการใช้คำสั่ง show database.....	28
2.22 แสดงการใช้คำสั่ง use.....	28
2.23 แสดงการใช้คำสั่ง Create table tablename .....	29
2.24 แสดงการใช้คำสั่ง show database.....	30
2.25 แสดงการใช้คำสั่ง show tables.....	30
2.26 แสดงการใช้คำสั่งเพื่อแสดงคอลัมน์.....	30
2.27 แสดงการใช้คำสั่ง insert.....	31
2.28 แสดงการใช้คำสั่ง select.....	31

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.29 แสดงการใช้คำสั่ง select.....	32
2.30 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข.....	33
2.31 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ between.....	33
2.32 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ in.....	34
2.33 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ not in.....	34
2.34 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ like.....	34
2.35 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ order by.....	34
2.36 แสดงการปรับปรุงโดยใช้คำสั่ง update.....	35
2.37 แสดงการปรับปรุงโดยใช้ delete.....	35
2.38 แสดงการลบตารางออกจากฐานข้อมูลด้วยคำสั่ง drop table.....	36
2.39 แสดงการลบฐานข้อมูลด้วยคำสั่ง drop database.....	36
3.1 หน้าตาหลัก.....	40
3.2 ส่วนของการรับข้อมูล.....	40
3.3 ส่วนของการรายงาน.....	41
4.1 Context diagram ของระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์.....	42
4.2 Data flow diagram ของระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์.....	43
4.3 ER Diagram.....	51
4.4 เว็บไซต์ล็อกอินและเมนูหลัก.....	53
4.5 เว็บไซต์การกรอกประวัติของผู้กู้.....	54
4.6 เว็บไซต์การทำสัญญา.....	55
4.7 เว็บไซต์รายงานการเปิดสัญญา.....	56
1 เว็บไซต์ล็อกอินและเมนูหลัก.....	61
2 การกรอกประวัติส่วนตัวของผู้กู้.....	62
3 การกรอกประวัติส่วนตัวของผู้ค้ำประกัน.....	63
4 การกรอกประวัติรถจักรยานยนต์.....	64
5 การกรอกการทำสัญญา.....	65
6 แสดงหน้าที่จะต้องเข้าไปกรอกรหัสเพื่อดูยอดค่าค้างชำระในแต่ละงวด.....	66
7 แสดงค่าใช้จ่ายในแต่ละงวดวันที่ต้องจ่ายอัตราค่าปรับในแต่ละงวด.....	66

## สารบัญรูป (ต่อ)

รูปที่		หน้า
8	แสดงการค้นหาประวัติของลูกค้าโดยต้องกรอกชื่อของผู้กู้.....	67
9	การค้นหาประวัติของผู้ค้าประกัน.....	67
10	ประวัติผู้กู้ที่ต้องการแก้ไข.....	68
11	แสดงหน้าที่สามารถเลือกช่วงรายงาน.....	69
12	การทำสัญญาในช่วงที่เลือก.....	69
13	การส่งจ่ายผิด.....	70
14	รายละเอียดของการส่งจ่ายผิด.....	70



## บทที่ 1

### บทนำ

#### 1.1 หลักการและเหตุผล

ในสมัยก่อนนั้นการจัดเก็บข้อมูลจะจัดเก็บในรูปแบบของกระดาษ เมื่อคอมพิวเตอร์ได้เข้ามามีบทบาท เราจึงได้นำโปรแกรมสำเร็จมาใช้ เช่น Microsoft Office จนปัจจุบันได้มีการพัฒนาภาษาที่ใช้สำหรับการจัดการข้อมูลโดยเฉพาะ ที่เรียกว่าระบบฐานข้อมูล ทำให้มีประสิทธิภาพในการจัดเก็บข้อมูล และสะดวกรวดเร็วในการค้นหา ปรับปรุงแก้ไขมากขึ้น

การจัดการข้อมูลของระบบการเงินสินเชื่อ แบบเดิมเป็นการจัดเก็บข้อมูลอยู่ในรูปแบบกระดาษ ซึ่งทำให้การค้นหาข้อมูลมีความล่าช้า แก้ไขข้อมูลได้ยาก ข้อมูลนั้นไม่มีความรัดกุมและการคำนวณอัตราดอกเบี้ยอาจเกิดความผิดพลาดได้เนื่องมาจากพนักงานที่ต้องคำนวณอัตราดอกเบี้ยทุกการทำสัญญาสินเชื่อ

#### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างโปรแกรมระบบการเงินสินเชื่อรถจักรยานยนต์
2. เพื่อสามารถออกแบบโปรแกรมในการจัดเก็บข้อมูลของระบบการเงินสินเชื่อ
3. เพื่อช่วยให้เจ้าของกิจการคำนวณรายรับ-รายจ่าย ในการปล่อยเงินสินเชื่อ
4. เพื่อช่วยพนักงานลดเวลาในการสืบค้นข้อมูล
5. เพื่อช่วยพนักงานในการคิดอัตราดอกเบี้ยในแต่ละเดือน
6. เพื่อช่วยเตือนพนักงานเรื่องทะเบียนรถจักรยานยนต์ที่ใกล้หมดอายุ

#### 1.3 ขอบข่ายของโครงการ

1. จัดการระบบการเงินสินเชื่อระบบเก่าให้มาอยู่ในระบบใหม่
2. ออกแบบฐานข้อมูลในการจัดเก็บระบบการเงินสินเชื่อ
3. ศึกษาภาษา php และ mysql เพื่อใช้ในการเขียนโปรแกรม

## 1.4 ขั้นตอนการดำเนินโครงการ

ตารางที่ 1.1 การดำเนินโครงการ

กิจกรรม	ธ.ค.48	ม.ค.49	ก.พ.49	มี.ค.49	เม.ย.49	พ.ค.49
ศึกษาปัญหาระบบการเงิน สินเชื่อบริการยานยนต์						
ศึกษาการเขียนภาษา php, my sql						
ออกแบบเว็บเพจ						
เขียนโปรแกรมและจัดทำ ฐานข้อมูล						
ทดสอบโปรแกรม และ โครงการ						
นำเสนอผลงานและจัดทำ รายงาน						

## 1.5 ผลที่คาดว่าจะได้รับ

1. ได้รับความรู้จากการเขียนโปรแกรม php โดยใช้ mysql เป็นฐานข้อมูล
2. ช่วยลดภาระต่าง ๆ อันเนื่องมาจากระบบเก่าให้แก่พนักงาน
3. เพื่อเป็นพื้นฐานในการพัฒนาระบบต่างๆ ได้ในอนาคต

## 1.6 รายละเอียดงบประมาณตลอดโครงการ

ค่ากระดาษรายงาน	200	บาท
ค่าหนังสือ	300	บาท
ค่าซอฟต์แวร์ / วัสดุ	700	บาท
ค่าถ่ายเอกสาร/ เข้ารูปเล่ม	800	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาท

(สองพันบาทถ้วน)

(หมายเหตุ: ทุกรายการสามารถถัวเฉลี่ยได้)

## บทที่ 2

### หลักการและทฤษฎี

#### 2.1 ทฤษฎีการคำนวณ

ระบบสินเชื่อเพื่อรถจักรยานยนต์นั้นใช้ทฤษฎีการคำนวณโดย นำเงินต้นที่ผู้กู้ได้กู้ไปแล้วนำมาหารด้วยจำนวนงวดที่ผู้กู้จะต้องส่งนามาคูณด้วยอัตราดอกเบี้ย จะได้เป็นเงินที่ผู้กู้จะต้องชำระในแต่ละงวด การคำนวณค่าปรับนั้นอัตราค่าปรับจะเกิดขึ้นก็ต่อเมื่อผู้กู้ชำระเงินไม่ตรงตามวันที่กำหนด ซึ่งผู้กู้สามารถจ่ายล่าช้าได้ 8 วันนับจากวันที่กำหนดจ่ายถ้าเลยจาก 8 วันไปแล้วอัตราดอกเบี้ยปรับจะมีผลใช้ทันที ตามทฤษฎีคำนวณอัตราดอกเบี้ยโดย นำเงินที่ผู้กู้ค้างชำระคูณด้วยอัตราดอกเบี้ยค่าปรับแล้วคูณด้วยจำนวนวันที่เลยกำหนดโดยนับจากวันที่กำหนดจ่าย จะได้อัตราดอกเบี้ยค่าปรับในแต่ละวัน แล้วนำไปคูณกับจำนวนวันที่เลยกำหนดการค้างชำระจะได้อัตราค่าปรับสุทธินำไปบวกกับเงินที่ค้างชำระ จะได้เงินที่ต้องชำระออกมา

$$\text{อัตราค่าปรับ} = (\text{เงินที่ต้องชำระในแต่ละงวด} * \text{ดอกเบี้ยปรับ} * \text{จำนวนวันที่มากกว่า 8 วัน} / 30)$$

$$\text{เงินที่ต้องชำระ} = \text{อัตราค่าปรับ} + \text{เงินที่ต้องชำระในแต่ละงวด}$$

#### 2.2 ภาษา PHP [1]

PHP คือภาษา script อย่างหนึ่งที่เป็น server-side script ซึ่งจะทำงานในฝั่ง server แล้วส่งการแสดงผลมายัง browser ของตัว Client และนอกจากนี้ มันยังเป็น script ที่ embed บน HTML อีกด้วย

PHP เป็นภาษาจําพวก script language คำสั่งต่าง ๆ จะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปต์เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่น ๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาโดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded Scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่งซึ่งช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพมากขึ้น

PHP ได้รับการเผยแพร่เป็นครั้งแรกในปี ค.ศ. 1994 โดยคุณ Rasmus Lerdorf ต่อมาได้มีนักโปรแกรมเมอร์เข้ามาช่วยในการพัฒนาต่อมาตามลำดับ เป็นเวอร์ชันต่าง ๆ จนกระทั่งถึงเวอร์ชันล่าสุดซึ่งเป็นเวอร์ชัน 5 นักพัฒนาสำคัญของเวอร์ชัน 4 และ เวอร์ชัน 5 คือคุณ Zeev Suraski

และคุณ Andi Gutmans ในขณะนี้มีเว็บเซิร์ฟเวอร์ประมาณ 16 ล้าน โดเมน (domains) ที่ใช้ PHP เราสามารถตรวจสอบจำนวนของ domains ที่ใช้ PHP ได้ที่ ในตอนแรก PHP ย่อมาจาก Personal Home Page แต่ต่อมาได้เปลี่ยนชื่อให้ตรงตามกฎเกณฑ์ของ GNU ชื่อในปัจจุบันของ PHP นั้นย่อมาจาก Hypertext Preprocessor รายละเอียดต่าง ๆ ของ PHP เราสามารถเข้าไปค้นหาเพิ่มเติมได้ที่ เว็บไซต์ของ PHP

### 2.3 ทำไมภาษา PHP น่าสนใจและน่าใช้ [1]

ภาษาอื่นที่ทำหน้าที่คล้าย ๆ กับภาษา PHP คือ Perl, Microsoft Active Server Pages (ASP), Java Server Page (JSP), และ Allaire ColdFusion

ถ้าเปรียบเทียบภาษา PHP กับ ภาษาอื่น ๆ เหล่านี้เราจะพบว่าภาษา PHP มีข้อได้เปรียบหลายอย่างดังต่อไปนี้

- สามารถติดต่อกับหลายประเภทของฐานข้อมูลอย่างเช่น MySQL, PostgreSQL, mSQL, Oracle, Informix, Sybase และสามารถใช้อินเตอร์เฟซ Open Database Connectivity Standard (ODBC) เพื่อติดต่อกับผลิตภัณฑ์ฐานข้อมูล ของ Microsoft
- ไม่ต้องเสียค่าใช้จ่ายในการใช้ โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ
- เรียนรู้และใช้งานง่าย โดยเฉพาะถ้าเรารู้ภาษา C, C++, Perl, และ Java อยู่แล้ว
- สามารถใช้ PHP ได้บนหลายระบบปฏิบัติการ โดยที่ไม่ต้องเปลี่ยนโปรแกรม

### 2.4 ประวัติ PHP [1]

ภาษา PHP ถูกสร้างโดย Rasmus Lerdorf ในปี ค.ศ.1994 เนื่องมาจากการต้องการพัฒนาโปรแกรมเพื่อเก็บข้อมูลของผู้ใช้ที่แวะเวียนเข้ามาเยี่ยมชมโฮมเพจส่วนตัว โดยเวอร์ชันแรกได้ถูกพัฒนาและเปิดเผยให้ผู้ต้องการใช้ศึกษาในปี ค.ศ. 1995 โดยเรียกว่า “Personal Home Page Tool” ซึ่งเป็นที่มาของคำว่า PHP เมื่อประมาณกลางปี 1995 Rasmus ได้คิดค้นและพัฒนาให้ PHP/FI หรือ PHP เวอร์ชัน 2 มีความสามารถจัดการเกี่ยวกับแบบฟอร์มข้อมูลที่ถูกสร้างมาจากภาษา HTML และสนับสนุนการติดต่อกับโปรแกรมจัดการฐานข้อมูล mSQL ได้ จึงทำให้มีผู้ใช้ PHP มากขึ้น โดยในปลายปี ค.ศ.1996 PHP ถูกนำไปใช้ประมาณ 15,000 เว็บไซต์ทั่วโลก และเพิ่มขึ้นเรื่อยๆ จนเป็น 50,000 เว็บไซต์ ในราวกลางปี ค.ศ.1997 PHP ได้นำพื้นฐานของ PHP/FI มาพัฒนาใหม่เป็น PHP เวอร์ชัน 3 ซึ่งมีความสมบูรณ์มากขึ้นโดยสามารถทำงานกับ C2's Strong Hold web Server และ Red Hat Linux ต่อมา PHP ถูกพัฒนาเป็น PHP 4 หรือ เวอร์ชัน 4 ที่ใช้อย่างแพร่หลายในปัจจุบัน

## 2.5 เรื่องของตัวแปรและคำสั่งพื้นฐาน [1]

ตัวแปร (variable) หมายถึง ตัวแทนของข้อมูลซึ่งอยู่ในหน่วยความจำ ของการเขียนโปรแกรมจำเป็นต้องมีการใช้ตัวแปรประเภทต่างๆ เพื่อพิกเก็บระหว่างการทำงานของโปรแกรม

### 2.5.1 การประกาศตัวแปร

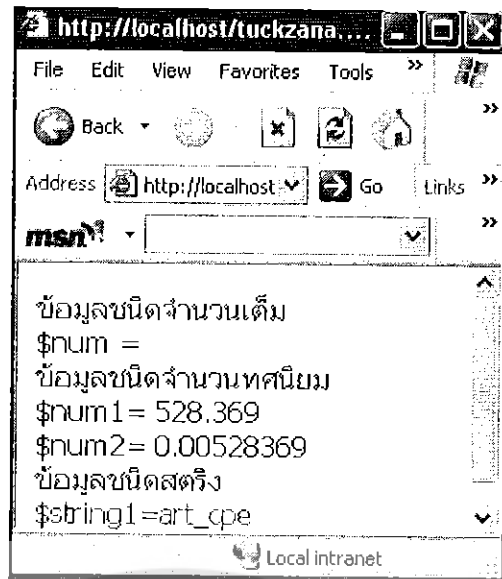
ใน php นั้นการประกาศตัวแปรต้องขึ้นต้นด้วยเครื่องหมาย \$ (dollar sing) เสมอแล้วจึงตามด้วยชื่อตัวแปร โดย php จะให้ความสำคัญกับตัวพิมพ์เล็กตัวพิมพ์ใหญ่ (case sensitive) ซึ่งตัวแปรนั้นต้องขึ้นต้นด้วยอักษรหรือเครื่องหมาย Underscore(\_) เท่านั้น

ความแตกต่างที่สำคัญระหว่าง PHP กับภาษาโปรแกรมอื่นๆก็คือ เรามักจำเป็นต้องกำหนดชนิดข้อมูลในขณะที่ประกาศตัวแปร เพราะ PHP นั้นจะพิจารณาข้อมูลของตัวแปรจากค่าของตัวแปร ดังตัวอย่างต่อไปนี้

```
<?php
echo "ข้อมูลชนิดจำนวนเต็ม<br>";
$num = 125;
echo "\$num = <br>";
echo "ข้อมูลชนิดจำนวนทศนิยม<br>";
$num1= 528.369;
$num2= 528.369e-5;
echo "\$num1= $num1<br>";
echo "\$num2= $num2<br>";
echo "ข้อมูลชนิดสตริง<br>";
$string1 = "art_cpe";
echo "<font color='red'\>\$string1=\$string1</font><br>";
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.1





รูปที่ 2.1 แสดงการรับข้อมูลชนิดต่างๆ

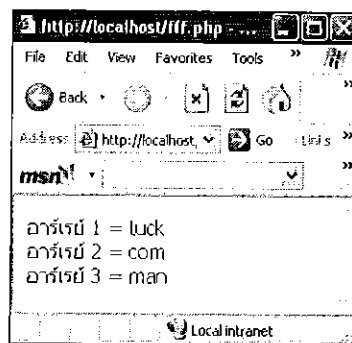
### 2.5.2 ข้อมูลชนิดอาร์เรย์ (Array)

เป็นข้อมูลที่เก็บค่าตั้งแต่ 1 ค่าขึ้นไปในชื่อเดียวกันเพื่อให้ง่ายต่อการจดจำนำไปใช้งาน โดยมี index เป็นตัวระบุตำแหน่งของแต่ละข้อมูลอยู่ภายใน

```

<?php
$array1 = array("tuck","com","man");
echo "อาร์เรย์ตัวที่ 1 = ".$array1[0]."<br>";
echo "อาร์เรย์ตัวที่ 2 = ".$array1[1]."<br>";
echo "อาร์เรย์ตัวที่ 3 = ".$array1[2]."<br>";
?>
  
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.2



รูปที่ 2.2 แสดงผลรันชนิดข้อมูลอาร์เรย์

### 2.5.3 การแปลงชนิดข้อมูล

โดยปกติเมื่อมีการสร้างตัวแปรขึ้นมา ชนิดข้อมูลของตัวแปรจะเป็นไปตามข้อมูลที่กำหนดให้กับตัวแปรนั้น แต่หากต้องการเปลี่ยนชนิดข้อมูลใหม่เป็นไปตามที่เราต้องการ เช่น เปลี่ยนข้อมูลชนิดตัวเลขไปเป็นข้อมูลชนิดสตริง ก็สามารทำได้ 2 วิธีคือ การแปรข้อมูลด้วยวิธีการ Cast และการใช้ฟังก์ชัน Settype()

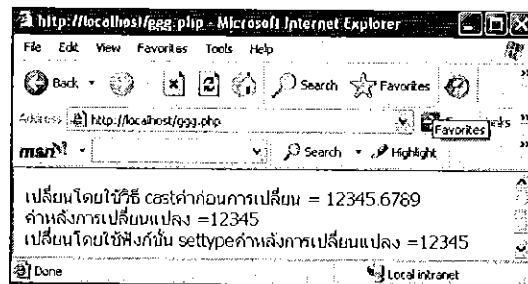
-การแปรข้อมูลด้วยวิธีการ Cast ซึ่งเป็นการระบุชนิดข้อมูลที่ต้องการไว้หน้าตัวแปร โดย

- (int),(integer) ใช้แปลงข้อมูลชนิดจำนวนเต็ม
- (real),(double),(float) ใช้แปลงข้อมูลชนิดทศนิยม
- (string) ใช้แปลงข้อมูลชนิดสตริง
- (array) ใช้แปลงข้อมูลชนิดอาร์เรย์

-การแปลงข้อมูลด้วยฟังก์ชัน settype() เป็นฟังก์ชันที่ใช้แปลงชนิดข้อมูล ซึ่งมีข้อมูลชนิดการใช้งานดังนี้

```
<?php
echo "เปลี่ยน โดยใช้วิธี cast<br>";
$num = 12345.6789;
echo "ค่าก่อนการเปลี่ยน" = ".$num."<br>";
$num1= (int)$num;
echo "ค่าหลังการเปลี่ยน =" . $num1."<br>";
echo " เปลี่ยน โดยใช้ฟังก์ชันsettype<br>";
settype ($num,"int");
echo "ค่าหลังการเปลี่ยน =" . $num;
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.3



รูปที่ 2.3 แสดงการรันการแปลงชนิดข้อมูล

### 2.5.4 ฟังก์ชันเกี่ยวข้องกับตัวแปร

-ฟังก์ชัน `gettype()` เป็นฟังก์ชันที่ใช้ตรวจสอบชนิดข้อมูลของตัวแปร

รูปแบบ `gettype(ตัวแปร)`

-ฟังก์ชัน `empty()` เป็นฟังก์ชันที่ใช้ตรวจสอบข้อมูลตัวแปรว่าเก็บข้อมูลไว้หรือไม่ โดยถ้าเก็บข้อมูลไว้จะคืนค่าเท็จ (`false`) แต่ถ้าไม่เก็บข้อมูลใดๆ ไว้(ตัวแปรไม่มีค่าเป็น `Null`) ก็จะส่งคืนค่า (`true`) มาให้

-ฟังก์ชัน `isset()` เป็นฟังก์ชันที่ตรวจสอบว่ามีตัวแปรตามชื่อที่ระบุไว้หรือไม่ ถ้ามีจะคืนค่าจริงมาให้ มิฉะนั้นจะคืนค่าเท็จ นอกจากนี้ถ้าตัวแปรนั้นถูกกำหนดค่าเป็น `Null` ฟังก์ชัน `isset()` ก็จะคืนค่าเท็จออกมาให้

รูปแบบ `isset(ตัวแปร)`

-ฟังก์ชัน `unset()` เป็นฟังก์ชันที่ใช้ทลายตัวแปร และคืนค่าหน่วยความจำในส่วนของตัวแปรนั้นให้แก่วระบบ

รูปแบบ `unset(ตัวแปร)`

### 2.5.5 ค่าคงที่ (Constant)

เราสามารถกำหนดค่าที่ขึ้นมาในโปรแกรมได้โดยใช้คำสั่ง `define()` เพื่อความสะดวกในการอ้างค่าใดค่าหนึ่งบ่อยๆ

รูปแบบ `define(ชื่อค่าคงที่,ค่า)`

### 2.5.6 ตัวดำเนินการทางคณิตศาสตร์(Arithmetic Operators)

ประกอบด้วย

- + การบวก(addition)
- การลบ(subtraction)
- \* การคูณ(multiplication)
- / การหาร(division)
- % การ โมดูลัส(modulus)คือการหาเศษที่ได้จากการหาร

เราสามารถเพิ่มหรือลดค่าของตัวแปรทีละหนึ่งตามรูปแบบที่นิยมใช้ในภาษาC และ Java โดยใช้ตัวดำเนินการต่อไปนี้

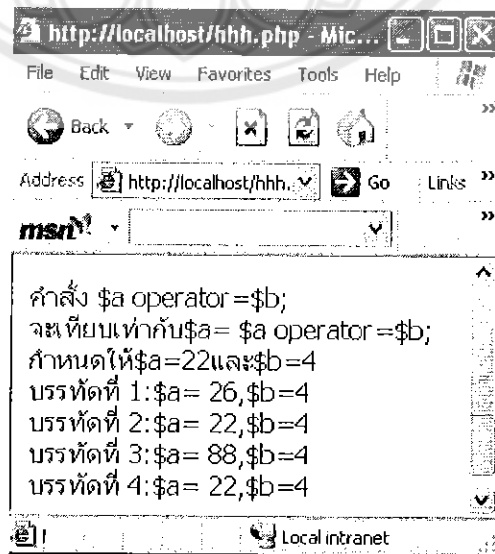
- ++ เพิ่มค่าขึ้น 1 (incrementing)
- ลดค่าลง ไป 1 (decrementing)

## 2.5.7 ตัวดำเนินการกำหนดค่า (Assignment Operators)

เป็นตัวดำเนินการที่ใช้ในการกำหนดค่าให้แก่ตัวแปร ซึ่งนอกจากตัวดำเนินการ=แล้วยังมี +=, -=, \*=, /= และอื่นๆดังตัวอย่างต่อไปนี้

```
<?php
echo "คำสั่ง \$a operator=\$b;<br>";
echo "จะเปรียบเทียบ\$a= \$a operator=\$b;<br>";
$a=22; $b=4;
echo "กำหนดให้ \$a=22&A\$b=4<br>";
$a+=$b;// $a=$a+$b
echo "บรรทัดที่ 1:\$a= $a,\$b=$b<br>";
$a-=$b;// $a=$a-$b
echo "บรรทัดที่ 2:\$a= $a,\$b=$b<br>";
$a*=$b;// $a=$a*$b
echo " บรรทัดที่3:\$a= $a,\$b=$b<br>";
$a/=$b;// $a=$a/$b
echo " 4:\$a= $a,\$b=$b<br>";
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.4



รูปที่ 2.4 แสดงการรันตัวดำเนินการกำหนดค่า

### 2.5.8 ตัวดำเนินการทางตรรกะ(Logical Operators)

เป็นสัญลักษณ์ที่ใช้ดำเนินการกับค่าตรรกะ(จริงเท็จ)ของตัวแปรหรือนิพจน์ดังรายละเอียดในตารางที่ 2.1

ตารางที่ 2.1 ตัวดำเนินการทางตรรกะ(Logical Operators) [1]

ตัวดำเนินการ	ชื่อ	ตัวอย่างการใช้งาน	ความหมาย
&& หรือ and	และ(and)	$\$a \& \& \$b$	ให้เป็นค่าจริงเมื่อทั้งคู่เป็นจริง
หรือ or	หรือ(or)	$\$a    \$b$	ให้เป็นค่าจริงเมื่อ \$a หรือ \$b หรือทั้งคู่เป็นจริง
!	นิเสธ (not)	$! \$a$	ให้ค่าเป็นจริงเมื่อตัวใดตัวหนึ่งเป็นจริง

### 2.5.9 ตัวดำเนินการเชิงเปรียบเทียบ (Comparison Operators)

เป็นตัวดำเนินการที่ใช้เปรียบเทียบตัวแปรหรือนิพจน์ 2 ตัวแปร/นิพจน์ โดยจะให้ค่าจริงหรือเท็จอย่างใดอย่างหนึ่งออกมาดังตัวอย่างตารางที่ 2.2

ตารางที่ 2.2 ตัวดำเนินการเชิงเปรียบเทียบ (Comparison Operators) [1]

ตัวดำเนินการ	ชื่อ	ตัวอย่างการใช้งาน	ความหมาย
==	เท่ากับ	$\$a == \$b$	ให้ค่าจริงเมื่อ \$a เท่ากับ \$b
!= หรือ <>	ไม่เท่ากับ	$\$a != \$b$	ให้ค่าจริงเมื่อ \$a ไม่เท่ากับ \$b
<	น้อยกว่า	$\$a < \$b$	ให้ค่าจริงเมื่อ \$a น้อยกว่า \$b
>	มากกว่า	$\$a > \$b$	ให้ค่าจริงเมื่อ \$a มากกว่า \$b

### 2.5.10 ตัวดำเนินการสตริง (String Operator)

เป็นดำเนินการสตริงมี 1 ตัวคือ ตัวดำเนินการที่ใช้ในการต่อสตริง 2 สตริงเข้าด้วยกัน (Concatenation operation) มีสัญลักษณ์เป็นเครื่องหมายจุด(.)

```
<?php
```

```
$a="<b>it_man:</b>";
```

```
$b="information technology man";
```

```
?>
```

```
<table>
```

```

</td></tr>
<?
echo $a.$b;
?>
</td></tr>
</table>

```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.5



รูปที่ 2.5 แสดงการรันตัวดำเนินการสตริง

### 2.5.11 การทำงานตามเงื่อนไข (Condition Execution)

ภาษา PHP มีคำสั่งที่ใช้ตรวจสอบเงื่อนไขเพื่อเลือกทำงานดังต่อไปนี้

คำสั่ง if

ภาษา PHP มีคำสั่งต่างๆที่อยู่ภายใน if ก็ต่อเมื่อเงื่อนไขมีค่าทางตรรกะเป็นจริง

รูปแบบ

```

<?php
$a=10;$b = 30;
$ans=" ";
if($a>$b){
    $ans = "A0;Çè0";
}
if($a<$b){

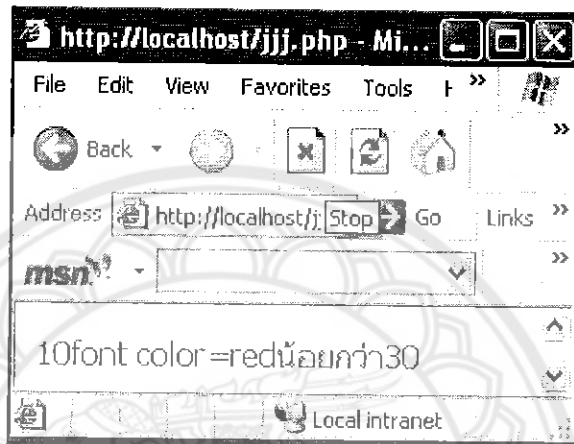
```

```

    $ans = "éíÃ;ÇèÒ";
    echo $a."font color=red".$ans."</font>".$b;
}
?>

```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.6



รูปที่ 2.6 แสดงการรันตามเงื่อนไข if

คำสั่ง else

ภาษา PHP else เป็นคำสั่งที่ใช้ร่วมกับ if ซึ่ง PHP จะทำคำสั่งภายใน else เมื่อเงื่อนไขหลัง if มีค่าทางตรรกะเป็นเท็จ

รูปแบบ

```
<?php
```

```
$a=5;$b=-5;$c=3;
```

```
if($b>$c){
```

```
    echo "\$bมีค่ามากกว่า \$c<br>";
```

```
}
```

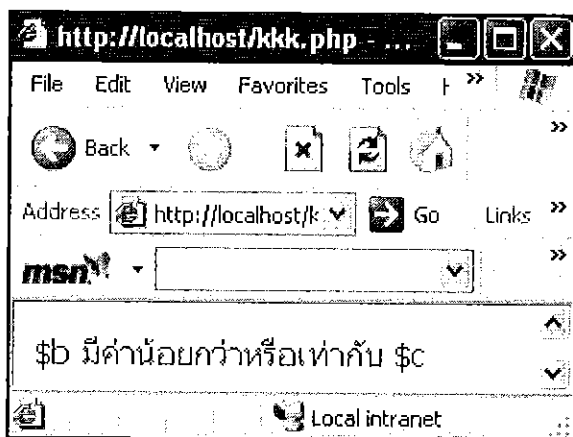
```
else{
```

```
echo "\$b มีค่าน้อยกว่าหรือเท่ากับ \$c<br>";
```

```
}
```

```
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.7



รูปที่ 2.7 แสดงการรันคำสั่ง else

### 2.5.12 การทำงานแบบวนซ้ำ(Iteration)

เมื่อเรามีคำสั่งในการ check เงื่อนไขของมันจะต้องมีการทำแบบวนซ้ำ

คำสั่ง for

เป็นคำสั่งที่ใช้ในการสร้าง โปรแกรมมันจะงานตามจำนวนครั้งที่เรต้องการโดยนิพจน์

- 1 นั้นจะทำครั้งแรกและครั้งเดียวตอนเข้าสู่ลูป จากนั้นในแต่ละรอบของการซ้ำ นิพจน์ 2 จะถูกตรวจสอบ โดยถ้ามีค่าทางตรรกะเป็นจริงก็จะทำคำสั่งที่อยู่ภายใน แล้วจึงทำนิพจน์ที่ 3 แต่ถ้านิพจน์ 2 เป็นเท็จก็จะหลุดออกจากคำสั่ง for ทั้งนี้จากตัวอย่างที่ผ่านมา เราสามารถนำมาเขียนใหม่ด้วยคำสั่ง for ได้ดังนี้

รูปแบบ

```
<?php
for($num=0;$num<=2;$num++){
    echo"$num : ผ่าน<br>";
}
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.8





รูปที่ 2.8 แสดงการรันรูปแบบคำสั่ง for

คำสั่ง do...while

คำสั่ง do...while จะเข้ามาในคำสั่งก่อนอย่างน้อย 1 ครั้งเสมอหลังจากนั้นถึงจะ check เงื่อนไข ถ้าคำสั่งเป็นจริงก็จะวนไปทำคำสั่งนั้นอีก แต่ถ้าเป็นเท็จก็จะออกจากลูปทันที

รูปแบบ

```
<?php
```

```
$art=1;
```

```
do{
```

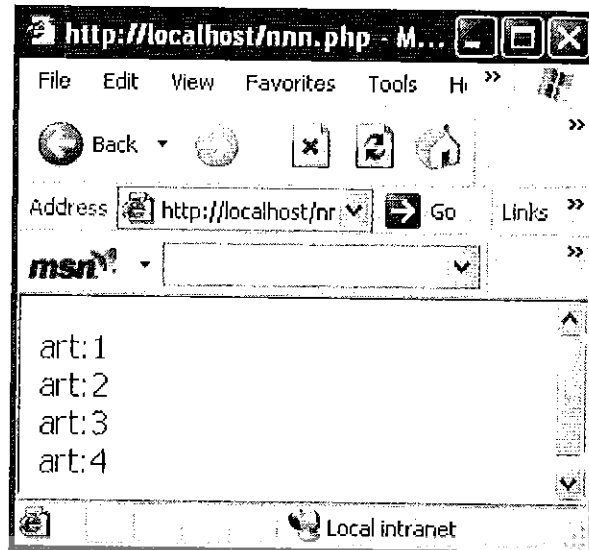
```
    echo"art:". $art."<br>";
```

```
    $art++;
```

```
}while($art<5)
```

```
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.9



รูปที่ 2.9 แสดงการรันคำสั่ง do...while

## 2.6 ฟังก์ชันเกี่ยวกับสตริง [1]

เนื่องจากสตริงเป็นข้อมูลที่สร้างปัญหาให้เราค่อนข้างมาก เนื่องจากเราไม่สามารถใช้ตัวดำเนินการปกติกับข้อมูลประเภทนี้ได้ เพราะเหตุนี้ php จึงได้จัดเตรียมฟังก์ชันจำนวนมากไว้ให้กัน โดยฟังก์ชันที่สำคัญๆมี ดังนี้

### 2.6.1 ฟังก์ชัน explode() และ implode()

explode() เป็นฟังก์ชันที่ใช้แบ่งข้อความออกเป็นส่วนย่อยๆ โดยขึ้นอยู่กับตัวแยก(delimiter) ที่เรากำหนดและส่งผลลัพธ์กลับมาเป็น อาร์เรย์ ที่มีข้อความย่อยๆเหล่านั้นเป็นสมาชิก

รูปแบบ

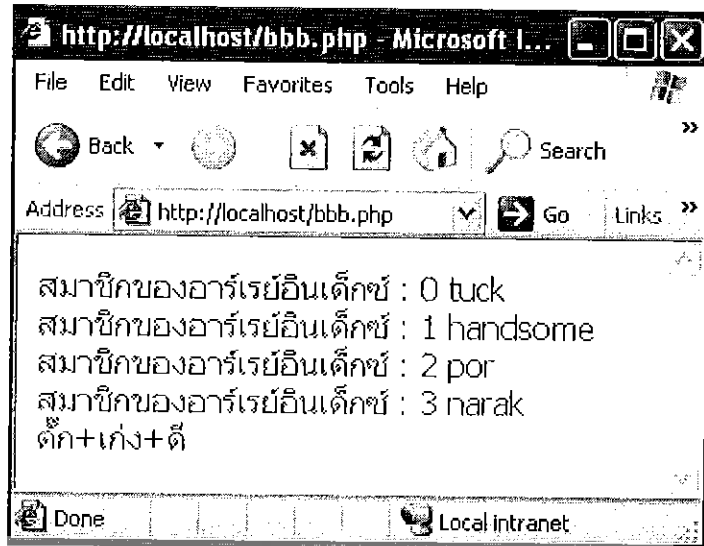
explode(สัญลักษณ์ที่ใช้แยก,ข้อความ)

implode() เป็นฟังก์ชันที่นำข้อมูลออกจากอาร์เรย์มาประกอบกันเป็นข้อมูลเดียวกัน ซึ่งจะตรงกันข้ามกับฟังก์ชัน explode()

รูปแบบ

implode(สัญลักษณ์ใช้คั่น,อาร์เรย์)

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.10



รูปที่ 2.10 แสดงการรันฟังก์ชัน explode และ implode

### 2.7 ฟังก์ชันเกี่ยวกับวันและเวลา [1]

ฟังก์ชัน date() จะส่งข้อมูลชนิดสตริงที่แสดงวัน/เวลาตามที่รูปแบบที่กำหนดโดยพิจารณาวัน/เวลาจากวัน/เวลาปัจจุบันของเครื่อง หรือจากอาร์กิวเมนต์ตัวที่ 2 (ถ้ามีการระบุ) รูปแบบใช้งานง่าย ๆ คือ

รูปแบบ

date(สตริง[, timestamp])

โดยที่แปรที่มีส่วนเกี่ยวข้องกับวันที่และเวลาดังแสดงได้ดังตารางที่ 2.3

ตารางที่ 2.3 ฟังก์ชันเกี่ยวกับวันและเวลา [1]

ตัวอักษร	ความหมาย
d	วันที่ โดยมีศูนย์นำหน้านั้นคือ "01", "31"
D	ชื่อวันในสัปดาห์ เฉพาะตัวอักษร 3 ตัวแรก เช่น "Sun", "Mon"
m	หมายเลขของเดือน โดยมีศูนย์นำหน้านั้นคือ "01", "02"
M	ชื่อเดือนเฉพาะตัวอักษร 3 ตัวแรก เช่น "Jan", "Feb"
y	เลขปีแบบ 2 หลัก เช่น "02" เป็นต้น
Y	เลขปีแบบ 4 หลัก เช่น "2005" เป็นต้น

### 2.8 การสร้างฟังก์ชันไว้ใช้เอง [1]

การประกาศฟังก์ชันมีรูปแบบดังนี้

Function ชื่อฟังก์ชัน([อาร์กิวเมนต์1,อาร์กิวเมนต์2,.....]){

คำสั่ง;

}

ชื่อฟังก์ชัน เป็นชื่อที่ตั้งขึ้นสำหรับเรียกใช้ฟังก์ชันนั้น

อาร์กิวเมนต์ เป็นตัวแปรชนิดต่างๆ โดยระบุไว้ในวงเล็บเปิดกับปิด เพื่อใช้ในการรับข้อมูลเข้ามาประมวลผล และ อาจกลับส่งข้อมูลกลับออกไป

คำสั่ง เป็นฟังก์ชันที่เราสร้างขึ้นมาใช้ในการทำอะไร

```
<?php
```

```
function find_area($dia,$mode){
```

```
    if($mode == 1){
```

```
        $sarea=(22/7)*$dia*$dia;
```

```
        return $sarea;
```

```
    }
```

```
    elseif($mode == 0){
```

```
        $sarea = $dia*$dia;
```

```
        return $sarea;
```

```
    }
```

```
}
```

```
$diameter = 14;
```

```
echo"Radius of Circle = $diameter";
```

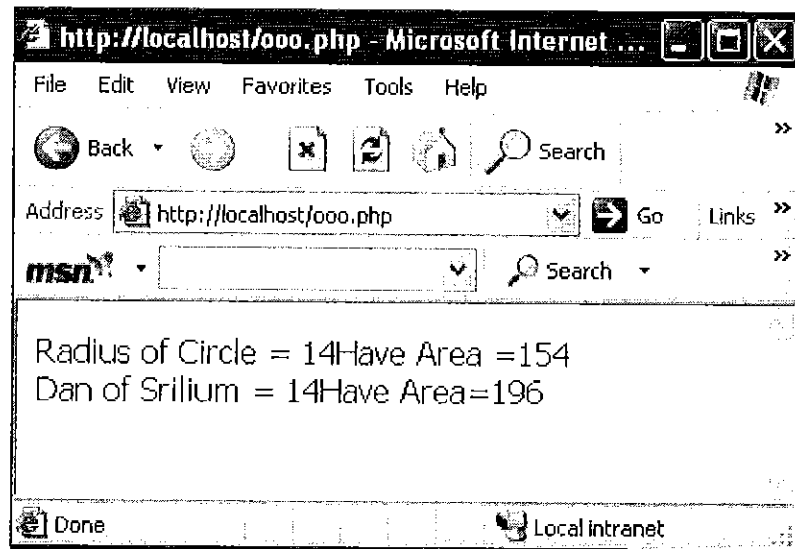
```
echo"Have Area =" .find_area($diameter,1);
```

```
echo"<br>Dan of Srilium = $diameter";
```

```
echo"Have Area=" .find_area($diameter,0);
```

```
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.11



รูปที่ 2.11 แสดงการรันฟังก์ชันที่สร้างไว้ใช้เอง

## 2.9 ขอบเขตของตัวแปร (Variable Scope) [1]

ขอบเขตของตัวแปรตัวแปรที่สร้างไว้ในกับนอกฟังก์ชัน ถึงแม้ว่าจะเป็นตัวแปรชื่อเดียวกัน แต่เราจะถือว่ามันเป็นตัวแปรคนละตัวและมีตำแหน่งในหน่วยความจำคนละที่กัน  
รูปแบบ

<?php

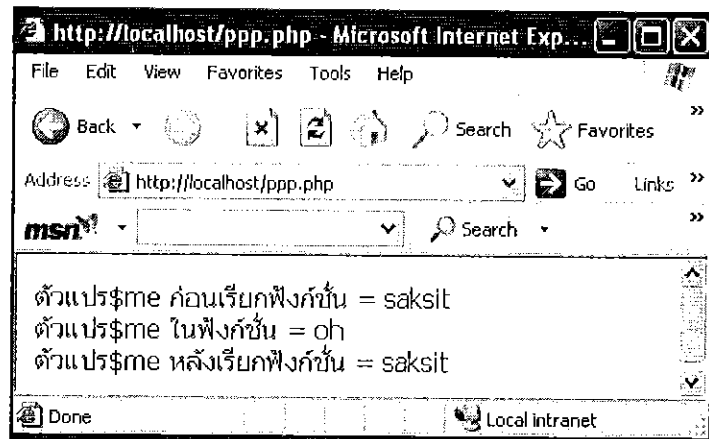
```

$me = "saksit";
function testvar() {
    $me = "oh";
    return $me;
}
echo"ตัวแปร \$me ก่อนเรียกฟังก์ชัน! = $me<br>";
echo"ตัวแปร Å\$me ในฟังก์ชัน= ".testvar()."<br>";
echo"ตัวแปร Å\$me หลังเรียกฟังก์ชัน= $me";

```

?>

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.12



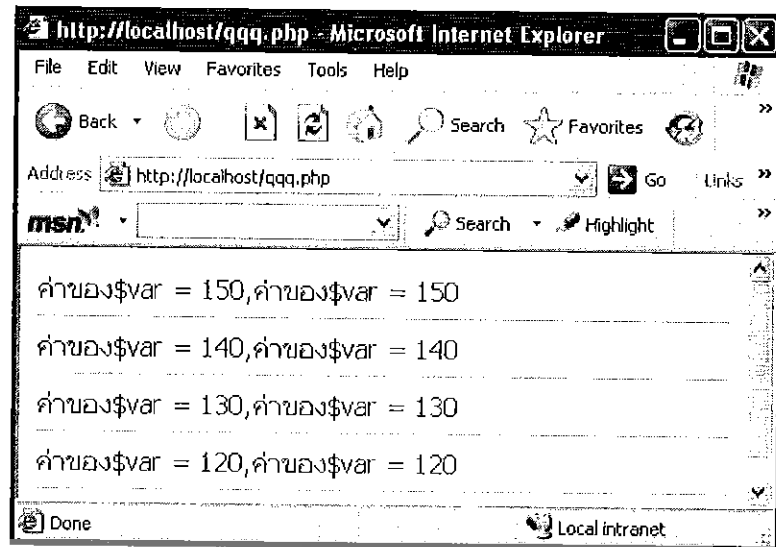
รูปที่ 2.12 แสดงการรันฟังก์ชันขอบเขตของตัวแปร

## 2.10 การสร้างตัวแปรแบบ static [1]

ทุกครั้งหลังจากที่ได้เรียกใช้ฟังก์ชัน ตัวแปรที่เราประกาศไว้ในตัวฟังก์ชันจะถูกทำลายทันทีหลังจากที่ทำงานของฟังก์ชันนั้นสิ้นสุดลง แล้วมันก็จะถูกสร้างขึ้นมาใหม่ทุกครั้งที่เราเรียกใช้ฟังก์ชัน ดังนั้นจะต้องสร้างตัวแปรแบบ static ดังตัวอย่าง

```
<?php
function testloop(){
    static $var = 150;
    static $str = "A";
    echo "เมื่อใช้ $var = $var, ";
    echo "เมื่อใช้ $var = $var<br>";
    $var-=10;
    $str++;
}
for($i = 0 ;$i<4;$i++){
    testloop();
}
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.13



รูปที่ 2.13 แสดงการรันตัวแปรแบบ static

## 2.11 การรับส่งข้อมูลจากผู้ใช้ [1]

เราจำเป็นต้องมีการกรองข้อมูลอะไรบางอย่างลงในหน้า web page เพื่อใช้ในการส่งข้อมูลหรือยืนยันว่าผู้ใช้งานเป็นใคร

### 2.11.1 input Objectg ประเภทต่างๆใน HTML

#### ช่องรับข้อความ (Text Field)

เป็นออบเจกต์ที่ใช้รับข้อความจากผู้ใช้ โดยข้อความเหล่านี้ก็คือ ตัวอักษร,ตัวเลข และสัญลักษณ์พิเศษต่างๆ  
รูปแบบ

```
<input type = "text" name = "ชื่อ" size = "ความกว้าง" maxlength = "จำนวนตัวอักษร" value = "ค่าเริ่มต้น">
```

โดยที่ type = "text"

name = "ชื่อ"

size = "ความกว้าง"

maxlength = "จำนวนตัวอักษร"

value = "ค่าเริ่มต้น"

กำหนดว่าเป็น input object ประเภทช่องรับข้อความ

กำหนดชื่อของช่องรับข้อความ

กำหนดความกว้างของช่องรับข้อความ

กำหนดจำนวนตัวอักษรสูงสุดที่ยอมให้ป้อนในช่อง

กำหนดค่าเริ่มต้นที่แสดงในช่องรับข้อความ

#### ช่องรับรหัส (Password Field)

### รูปแบบ

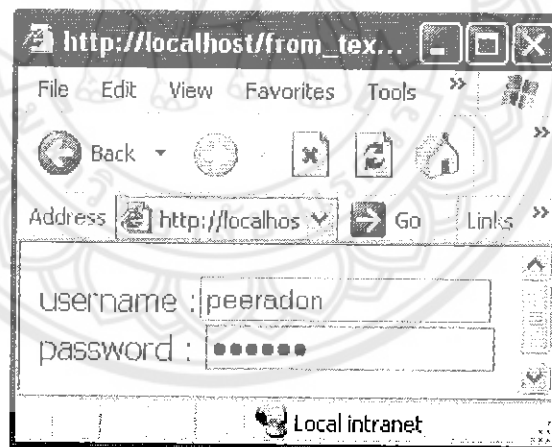
```
<input type = "password" name = "ชื่อ" size = "ความกว้าง" maxlength = "จำนวนตัวอักษร" value = "ค่าเริ่มต้น">
```

โดยที่ type = "password"	กำหนดว่าเป็น input object ประเภทช่องรับรหัส
name = "ชื่อ"	กำหนดชื่อของช่องรับข้อความ
size = "ความกว้าง"	กำหนดความกว้างของช่องรับข้อความ
maxlength = "จำนวนตัวอักษร"	กำหนดจำนวนตัวอักษรสูงสุดที่ยอมให้ป้อนในช่อง
value = "ค่าเริ่มต้น"	กำหนดค่าเริ่มต้นที่แสดงในช่องรับข้อความ

### Ex1 form\_textfield.php

```
username :<input type = "text" name = "text1" size = "20" maxlength = "50" value = "peeradon">
password : <input type = "password" name = "pass1" size = "20" maxlength = "50" value = "mek_up">
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.14



รูปที่ 2.14 แสดงช่องกรอก password

### ช่องรับรหัส (Password Field)

เป็นออบเจ็กต์ที่ใช้รับข้อความหลายบรรทัด

### รูปแบบ

```
<textarea name="ชื่อ" cols="ความกว้าง" rows="ความสูง">ค่าเริ่มต้น</textarea>
```

โดยที่ <textarea></textarea>	กำหนด input object ประเภทช่องรับข้อความแบบหลายบรรทัด
name="ชื่อ"	กำหนดชื่อของช่องรับข้อความ



cols="ความกว้าง"	กำหนดความกว้างของช่องรับข้อความ
rows="ความสูง"	กำหนดความสูงของช่องรับข้อความ
ค่าเริ่มต้น	กำหนดค่าเริ่มต้นที่แสดงในช่องรับข้อความ

### Ex2 form\_textarea.php

```
<textarea name="text1" cols="40" rows="3">
```

```
pee
```

```
ra
```

```
don
```

```
</textarea>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.15



รูปที่ 2.15 แสดงการรันช่องรับรหัส

### เช็คบ็อกซ์ (Checkbox)

เป็นออบเจกต์ที่ใช้สำหรับกำหนดทางเลือกซึ่งเป็นไปได้ 2 ทาง โดยการคลิกเลือกหรือไม่เลือก

#### รูปแบบ

```
<input type="checkbox" name="ชื่อ" value="ค่า" checked>
```

โดยที่ input type="checkbox"	กำหนด input object ประเภทเช็คบ็อกซ์
name="ชื่อ"	กำหนดชื่อของเช็คบ็อกซ์
value="ค่า"	กำหนดค่าที่ต้องการส่ง
checked	กำหนดให้เช็คบ็อกซ์ถูกเลือกไว้ในตอนต้น

### Ex3 form\_checkbox.php

```
<input type="checkbox" name="pec" > : CheckBox1<br>
```

```
<input type="checkbox" name="ra" checked> : CheckBox2<br>
```

```
<input type="checkbox" name="don" checked> : CheckBox3<br>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.16



รูปที่ 2.16 แสดงผลการรันการ CheckBox

#### 2.11.2 ปุ่มตัวเลือก (Radio Button)

เป็นออบเจกต์ที่ใช้สำหรับกำหนดทางเลือกคล้ายกับเช็ลบอกซ์ แต่จะต้องใช้ 2 อันขึ้นไปโดยเมื่อเรากำหนดชื่อของปุ่มตัวเลือกใดๆ ไว้เหมือนกัน ผู้ใช้จะไม่สามารถคลิกปุ่มตัวเลือกเหล่านี้ได้ทีละหลายอัน

#### รูปแบบ

```
<input type="radio" name="ชื่อ" value="ค่า" checked>
```

โดยที่ type="radio"	กำหนด input object ประเภทปุ่มตัวเลือก
name="ชื่อ"	กำหนดชื่อของปุ่มตัวเลือก
value="ค่า"	กำหนดค่าที่ต้องการส่ง
checked	กำหนดให้เช็ลบ็อกซ์ถูกเลือกไว้ในตอนต้น

### Ex4 form\_radio.php

```
เพศ<br>
```

```
<input type="radio" name="sex" value="m" checked>ชาย <br>
```

```
<input type="radio" name="sex" value="f">หญิง
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.17



รูปที่ 2.17 แสดงผลการรันปุ่มตัวเลือก Radio Button

### 2.11.3 ลิสต์บ็อกซ์ (Listbox)

เป็นออบเจกต์ที่แสดงรายการให้ผู้ใช้เลือก ซึ่งกำหนดได้ทั้งแบบให้เลือกได้เพียงรายการเดียว หรือ หลายรายการพร้อมกัน (multiple selection)

รูปแบบ

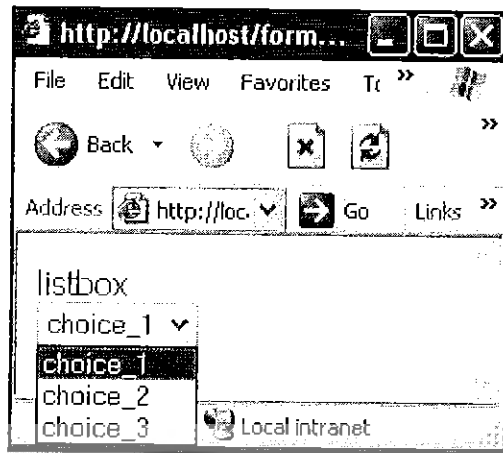
```
<select name="ชื่อ" size="ความสูง" >
  <option value="">ข้อความ</option>
</select>
```

โดยที่	< select ></ select >	กำหนด input object ประเภทลิสต์บ็อกซ์
	name="ชื่อ"	กำหนดชื่อของลิสต์บ็อกซ์
	size="ความสูง"	กำหนดความสูงของลิส
	<option></option>	เป็นการสร้างรายการเลือกของลิสต์บ็อกซ์
	value="ค่า"	กำหนดค่าที่ต้องการส่ง

#### Ex5 form\_listbox.php

```
listbox<br>
<select name="b" size="1" >
  <option value="p1">choice_1</option>
  <option value="p2">choice_2</option>
  <option value="p3">choice_3</option>
</select>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.18



25.  
พ.พ.พ.พ.  
2548.  
15004949.

รูปที่ 2.18 แสดงผลการรันลิสต์ป๊อปอัพ Listbox

#### 2.11.4 Hidden Field

เป็นออบเจกต์ที่ใช้ส่งค่าจากอีก ไฟล์หนึ่ง ไปยังอีกไฟล์หนึ่ง โดยไม่ให้ผู้รับรู้รูปแบบ

```
<input type="hidden" name="ชื่อ" value="ค่า">
```

type="hidden"	กำหนด input object ประเภท hidden field
name="ชื่อ"	กำหนดชื่อเป็น hidden field
value="ค่า"	กำหนดค่าของ hidden field

#### 2.11.5 ปุ่มคำสั่ง (Command Button)

เป็นออบเจกต์ที่มีลักษณะเป็นปุ่มกด สำหรับให้ผู้ใช้คลิกเพื่อสั่งงาน ซึ่งมีอยู่ 3 แบบ คือ

- ปุ่มส่งข้อมูล (submit) ใช้ส่งข้อมูลทั้งหมดไปยัง server
- ปุ่มล้างข้อมูล (reset) ล้างข้อมูลทั้งหมดภายในฟอร์มให้เป็นค่าเริ่มต้น
- ปุ่มธรรมดา (button) นิยมใช้ร่วมกับสคริปต์ที่ทำงานร่วมกับฝั่งไคลเอนต์

รูปแบบ

```
<input type="submit" name="ชื่อ" value="ข้อความ">
```

```
<input type="reset" name="ชื่อ" value="ข้อความ">
```

โดยที่ type="submit" หรือ type=" reset " กำหนด input object ประเภทส่งข้อมูลหรือล้างข้อมูล

name="ชื่อ"	กำหนดชื่อของปุ่ม
value="ข้อความ"	กำหนดข้อความที่ปรากฏบนปุ่ม

### 2.11.6ฟอร์ม (form)

เป็นส่วนสำคัญที่ใช้กำหนดรายละเอียดของการส่งข้อมูลไปยัง server ซึ่งในเว็บเพจหนึ่งๆ สามารถมีฟอร์มได้มากกว่า 1 โดยเมื่อผู้ใช้คลิกปุ่มส่งข้อมูลภายในฟอร์มใด บราวเซอร์ก็จะส่งค่าของ input object ทั้งหมดที่อยู่ในฟอร์มนั้นไปให้ server

รูปแบบ

```
<form action="ไฟล์"method="วิธีส่งข้อมูล"enctype="ประเภทข้อมูล">
```

input object ต่าง

```
</form>
```

โดยที่ <form></form>

เป็นแท็กที่ใช้สร้างฟอร์ม

action="ไฟล์"

กำหนดไฟล์ php ที่บราวเซอร์จะเรียกและส่งค่า

method="วิธีส่งข้อมูล"

กำหนดวิธีการส่งข้อมูลไปยัง server โดยมีอยู่ 2 แบบ คือ get และ post

**Ex6** การใช้formในการส่งค่า

**form\_input.php**

```
<form action="form_show.php"method="post">
```

```
<input type="text" name="peeradon"size="20">
```

```
<input type="submit" value="submit">
```

```
</form>
```

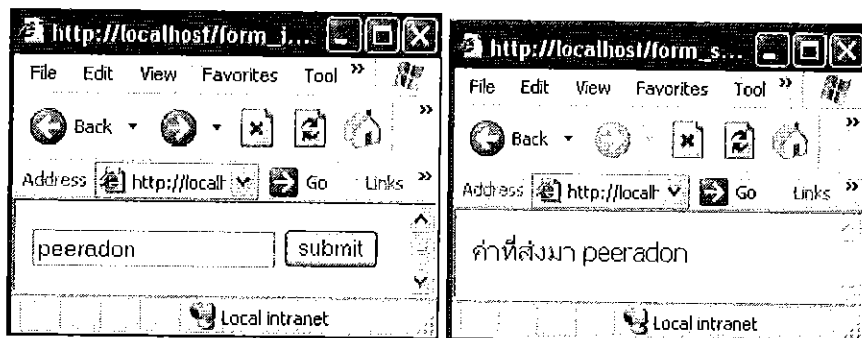
**form\_show.php**

```
<?
```

```
echo "ค่าที่ส่งมา ".$peeradon;
```

```
?>
```

หลังจากที่ได้ Run Program มีผลเป็นดังรูปที่ 2.19



รูปที่ 2.19 แสดงผลการรันการใช้ form ในการส่งค่า

## 2.12 MySQL [1]

MySQL เป็นฐานข้อมูลแบบ open source ที่ได้รับความนิยมในการใช้งานสูงสุดโปรแกรมหนึ่งบนเครื่องให้บริการ มีความสามารถในการจัดการกับฐานข้อมูลด้วยภาษา SQL (Structures Query Language) อย่างมีประสิทธิภาพ มีความรวดเร็วในการทำงาน รองรับการทำงานจากผู้ใช้งานหลายๆ คนและหลายๆ งานได้ในขณะเดียวกัน ฐานข้อมูลมีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูล การที่จะเพิ่มเติม เข้าถึงหรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นต้องอาศัยระบบจัดการฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอปพลิเคชันอื่นๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูลและระบบจัดการฐานข้อมูล

## 2.13 ทำไมถึงเลือกใช้ MySQL [1]

- MySQL เป็นระบบจัดการฐานข้อมูลแบบ relational คือ จะทำการจัดเก็บข้อมูลทั้งหมดอยู่ในรูปแบบตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์เดียว จะทำให้ทำงานได้รวดเร็วขึ้นและมีความยืดหยุ่น
- MySQL แจกจ่ายให้ใช้งานแบบ open source คือ ผู้ใช้งาน MySQL ทุกคนสามารถใช้งานและปรับแต่งการทำงานได้ตามต้องการ สามารถดาวน์โหลดโปรแกรม MySQL ได้จากอินเทอร์เน็ตและนำมาใช้งานโดยไม่มีค่าใช้จ่ายใดๆ

## 2.14 การสร้างฐานข้อมูลด้วย MySql [1]

### 2.14.1 การสร้างฐานข้อมูล

ฐานข้อมูลของ MySql เป็น Database Server ซึ่งสามารถรองรับฐานข้อมูลที่มีความแตกต่างกันได้สามารถสร้างฐานข้อมูลได้ที่ Mysql command prompt โดยพิมพ์โค้ดได้ดังนี้

รูปแบบ

```
Create database dbname;
```

โดยที่ *dbname* หมายถึง ชื่อของฐานข้อมูล

**Ex1 การสร้างฐานข้อมูล**

แสดงการสร้างฐานข้อมูล ดังรูป 2.20

```
mysql> create database dbblissing;
Query OK, 1 row affected (0.00 sec)
```

รูปที่ 2.20 แสดงการสร้างฐานข้อมูล

หลังจากนั้นใช้คำสั่ง `show databases` เพื่อดูฐานข้อมูลทั้งหมดที่มี ซึ่งจะเห็นว่า มี database `dbblissing` ที่สร้างไว้ ดังรูป

แสดงการใช้คำสั่ง `show databases` ดังรูปที่ 2.21

```
mysql> show databases;
+-----+
| Database |
+-----+
| compasstravel |
| db_blissing |
| dbblissing |
| dddd |
| iccoms |
| mysql |
| test |
+-----+
7 rows in set (0.01 sec)
```

รูปที่ 2.21 แสดงการใช้คำสั่ง `show databases`

#### 2.14.2 การสร้างตารางภายในฐานข้อมูล

ก่อนการสร้างตารางภายในฐานข้อมูลต้องใช้คำสั่ง `use` เพื่อเรียกฐานข้อมูลที่ต้องการก่อน โดยมีรูปแบบคำสั่งดังนี้

รูปแบบ

`Use dbname;`

โดยที่ `dbname` หมายถึง ชื่อของฐานข้อมูล

Ex2 การใช้คำสั่ง `use`

แสดงการใช้คำสั่ง `use` ดังรูปที่ 2.22

```
mysql> use dbblissing;
Database changed
mysql>
```

รูปที่ 2.22 แสดงการใช้คำสั่ง `use`

หลังจากที่ใช้คำสั่ง `use` แล้วจะทำให้สามารถสร้างตารางภายในฐานข้อมูลที่เราเลือกได้ โดยใช้คำสั่งดังนี้

รูปแบบ

`Create table tablename(column1 type, column2 type,...);`

โดยที่ `tablename` หมายถึง ชื่อตาราง

`Column1` หมายถึง คอลัมน์ที่ต้องการสร้าง

`Type` หมายถึง ชนิดของคอลัมน์

Ex3 การใช้คำสั่ง `Create table tablename`

แสดงการใช้คำสั่ง Create table tablename ดังรูปที่ 2.23

```
mysql> create table customer(c_id int(5) unsigned not null auto_increment primary key,c_fname char(30) not null,c_lname char(30) not null);
Query OK, 0 rows affected (0.02 sec)
```

รูปที่ 2.23 แสดงการใช้คำสั่ง Create table tablename

ใช้คำสั่ง create เพื่อสร้างตารางใหม่ชื่อ customer ประกอบด้วย 3 คอลัมน์ คือ c\_id, c\_fname และ c\_lname กำหนดให้ c\_id เป็น primary key จึงกำหนดเป็น not null คือเป็นค่าที่ห้ามว่างและกำหนดให้เพิ่มแบบ อัตโนมัติ (auto-increment)

### 2.14.3 คำอธิบายคีย์เวิร์ด (keyword)

- Not null หมายถึง ค่าของข้อมูลในคอลัมน์นี้ห้ามเป็นค่าว่าง จะต้องมียังข้อมูลอยู่
- Auto\_increment หมายถึง เป็นคำสั่งพิเศษใน MySQL ใช้กับคอลัมน์ที่เป็นเลขจำนวนเต็ม (integer) เมื่อเพิ่มแถวในตารางค่านี้จะเพิ่มทีละ 1 โดยอัตโนมัติ ซึ่งค่าเหล่านี้จะไม่ซ้ำกันกับข้อมูลในคอลัมน์ที่ใช้ auto\_increment จะต้องกำหนดเป็น index (หรือ primary key) ซึ่งแสดงลำดับสำหรับอ้างอิงค่าข้อมูลด้วยเสมอ
- Primary key หมายถึง เป็นการกำหนดค่าของคอลัมน์ให้เป็น primary key
- Unsigned หมายถึง เป็นการกำหนดค่าชนิดเลขจำนวนเต็มให้เป็นค่าที่ไม่มีเครื่องหมาย

เครื่องหมาย

### 2.14.4 การแสดงรายละเอียดในฐานข้อมูล

การแสดงข้อมูลของตารางในฐานข้อมูล ทำได้โดยใช้คำสั่ง show ดังนี้

#### Ex3 การใช้คำสั่ง show databases แสดงชื่อฐานข้อมูล

รูปแบบ

```
Show databases;
```

โดยที่ databases หมายถึง ชื่อฐานข้อมูล

แสดงการแสดงผลรายละเอียดในฐานข้อมูลดังรูปที่ 2.24



```
mysql> show databases;
+-----+
| Database |
+-----+
| compasstravel |
| db_lissing |
| dblissing |
| dddd |
| iccoms |
| mysql |
| test |
+-----+
7 rows in set (0.00 sec)
```

รูปที่ 2.24 แสดงการใช้คำสั่ง show databases

**Ex4 การใช้คำสั่ง show tables แสดงตาราง**

รูปแบบ

Show tables;

โดยที่ tables หมายถึง ชื่อตาราง

แสดงการใช้คำสั่ง show tables แสดงตารางดังรูปที่ 2.25

```
mysql> show tables;
+-----+
| Tables_in_dblissing |
+-----+
| customer |
+-----+
1 row in set (0.00 sec)
```

รูปที่ 2.25 แสดงการใช้คำสั่ง show tables

**Ex5 การใช้คำสั่งเพื่อแสดงคอลัมน์ (โครงสร้างของตาราง)**

รูปแบบ

Show columns from *tablename* from *databasename*;โดยที่ *tablename* หมายถึงชื่อตาราง*Databasename* หมายถึงชื่อฐานข้อมูล

แสดงการใช้คำสั่งเพื่อแสดงคอลัมน์ (โครงสร้างของตาราง)

```
mysql> show columns from customer from dblissing;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| c_id | int(5) unsigned | | PRI | NULL | auto_increment |
| c_fname | char(30) | | | | |
| c_lname | char(30) | | | | |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

รูปที่ 2.26 แสดงการใช้คำสั่งเพื่อแสดงคอลัมน์

### 2.14.5 การเพิ่มข้อมูลในฐานข้อมูลด้วยคำสั่ง insert

รูปแบบ

```
INSERT [into] table[(column1, column2, column3,...)]
```

```
VALUES(value1, value2, value3,...)
```

โดยที่ table หมายถึง ชื่อตาราง

Column หมายถึง ชื่อคอลัมน์จะมีหรือไม่ก็ได้

Value หมายถึง ค่าของข้อมูลที่ต้องการเพิ่ม

Ex6 การใช้คำสั่ง insert

แสดงการใช้คำสั่ง insert ดังรูปที่ 2.27

```
mysql> insert into customer values('1', 'peeradkon', 'mekub');
Query OK, 1 row affected (0.00 sec)
```

รูปที่ 2.27 แสดงการใช้คำสั่ง insert

### 2.14.6 การเรียกใช้ข้อมูลด้วยคำสั่ง select

หากต้องการเรียกดูข้อมูลหรือค้นหาข้อมูล ทำได้โดยใช้คำสั่ง SELECT ซึ่งระบุเงื่อนไขต่างๆร่วมด้วยโดยการใช้คำสั่ง SELECT สามารถแบ่งเป็นหัวข้อต่างๆตามลักษณะการใช้งานได้ดังนี้

รูปแบบ

```
SELECT items FROM tables;
```

โดยที่ items หมายถึง ชื่อคอลัมน์

Tables หมายถึง ชื่อตาราง

Ex6 การใช้คำสั่ง select แบบเลือกคอลัมน์ที่ต้องการแสดง

แสดงการใช้คำสั่ง select แบบเลือกคอลัมน์ที่ต้องการแสดงดังรูปที่ 2.28

```
mysql> select c_id,c_fname from customer;
+----+-----+
| c_id | c_fname |
+----+-----+
| 2    | suksit  |
| 1    | peeradkon |
+----+-----+
2 rows in set (0.00 sec)
```

รูปที่ 2.28 แสดงการใช้คำสั่ง select

Ex7 การใช้คำสั่ง select แบบเลือกคอลัมน์ทั้งหมดแสดง

แสดงการใช้คำสั่ง select แบบเลือกคอลัมน์ทั้งหมดแสดงดังรูปที่ 2.29

```
mysql> select * from customer;
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 2    | suksit  | kunsakon |
| 1    | peeradkon | mekub    |
+----+-----+-----+
2 rows in set (0.00 sec)
```

รูปที่ 2.29 แสดงการใช้คำสั่ง select

#### 2.14.7 การเรียกใช้ข้อมูลแบบมีเงื่อนไข

เป็นการใช้คำสั่ง SELECT เพื่อเรียกดูข้อมูลภายในตาราง ตามเงื่อนไขที่กำหนด

##### รูปแบบ

```
FROM tables
[WHERE condition]
[GROUP BY group_type]
[ORDER BY order_type]
```

โดยที่ tables หมายถึง ชื่อตาราง  
 Condition หมายถึง เงื่อนไขที่ต้องการทำ  
 Group\_type หมายถึง รายชื่อของคอลัมน์ที่ใช้ในการกำหนดกลุ่ม  
 Order\_type หมายถึง ชนิดของการเรียงลำดับ

WHERE ใช้เรียกดูข้อมูลแบบมีเงื่อนไข เป็นการระบุค่าเฉพาะข้อมูลที่ต้องการเรียกดู อาจใช้เงื่อนไขเพื่อดึงข้อมูลบางแถวจากตารางก็ได้ โดยการใช้คำสั่ง WHERE ซึ่งภายในคำสั่ง WHERE จะประกอบด้วยส่วนสำคัญดังนี้

- ชื่อคอลัมน์
- Operator การเปรียบเทียบ
- ข้อมูลเฉพาะที่ต้องการแสดงเป็นเงื่อนไขของคอลัมน์ที่ระบุ

หลังคำสั่ง WHERE จะใช้ operator สำหรับการกำหนดเงื่อนไขเพื่อเรียกใช้ข้อมูล ซึ่งมี operator ดังตารางที่ 2.4

ตารางที่ 2.4 แสดงตาราง Operator [1]

Operator	ชื่อ
=	เท่ากับ
>	มากกว่า

<	น้อยกว่า
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ
!= หรือ <>	ไม่เท่ากับ
IS NOT NULL	ไม่ว่าง
IS NULL	ว่าง
BETWEEN	ค่าที่อยู่ระหว่าง
IN	ค่าที่อยู่ในคอลัมน์ที่กำหนด
NOT IN	ค่าที่ไม่อยู่ในคอลัมน์ที่กำหนด
LIKE	เหมือนกับรูปแบบที่กำหนด
NOT LIKE	ไม่เหมือนกับรูปแบบที่กำหนด
REGEXP	Regular expression

Ex8 การใช้คำสั่ง select แบบมีเงื่อนไข

แสดงการใช้คำสั่ง select แบบมีเงื่อนไข ดังรูปที่ 2.30

```
mysql> select * from customer where c_id = '1';
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 1 | peeradkon | mekub |
+----+-----+-----+
1 row in set (0.00 sec)
```

รูปที่ 2.30 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข

Ex9 การใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ between

แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ between ดังรูปที่ 2.31

```
mysql> select * from customer where c_id between '2' and '3';
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 2 | suksit | kunsakon |
| 3 | somasuk | meeluk |
+----+-----+-----+
2 rows in set (0.00 sec)
```

รูปที่ 2.31 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ between

Ex10 การใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ in

แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ in ดังรูปที่ 2.32

```
mysql> select * from customer where c_fname in('peeradkon');
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 1 | peeradkon | mekub |
+----+-----+-----+
1 row in set (0.00 sec)
```

รูปที่ 2.32 แสดงการใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ in

Ex11 การใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ not in  
แสดงการใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ not in ดังรูปที่ 2.33

```
mysql> select * from customer where c_fname not in('peeradkon');
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 2 | suksit | kunsakon |
| 3 | somasuk | meeluk |
+----+-----+-----+
2 rows in set (0.00 sec)
```

รูปที่ 2.33 แสดงการใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ not in

Ex12 การใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ like  
แสดงการใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ like ดังรูปที่ 2.34

```
mysql> select * from customer where c_fname like('%ra%');
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 1 | peeradkon | mekub |
+----+-----+-----+
1 row in set (0.00 sec)
```

รูปที่ 2.34 แสดงการใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ like

Ex13 การใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ order by  
แสดงการใช้คำสั่ง select แบบมีเงื่อนไขโดยใช้ order by ดังรูปที่ 2.35

```
mysql> select * from customer order by c_id asc;
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 1    | peeradkon | mekub   |
| 2    | suksit   | kunsakon |
| 3    | sonsuk   | meeluk  |
+----+-----+-----+
3 rows in set (0.28 sec)

mysql> select * from customer order by c_id desc;
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 3    | sonsuk   | meeluk  |
| 2    | suksit   | kunsakon |
| 1    | peeradkon | mekub   |
+----+-----+-----+
3 rows in set (0.00 sec)
```

รูปที่ 2.35 แสดงการใช้คำสั่ง select แบบมีเงื่อนไข โดยใช้ order by

#### 2.14.8 การปรับปรุงแถวในฐานข้อมูลด้วยคำสั่ง update

การเรียกใช้ข้อมูลในฐานข้อมูล เพื่อนำมาปรับปรุงใหม่สามารถทำได้ดังนี้  
รูปแบบ

UPDATE tablename

SET column1= expression1, expression2, expression3,...

[WHERE condition]

โดยที่ tablename หมายถึง ชื่อตาราง

Column หมายถึง ชื่อคอลัมน์

Expression หมายถึง รูปแบบการคำนวณหรือค่าใหม่ที่ต้องการกำหนดก็ได้

#### Ex14 การปรับปรุงโดยใช้คำสั่ง update

แสดงการปรับปรุงโดยใช้คำสั่ง update ดังรูปที่ 2.36

```
mysql> update customer set c_fname = 'dddddd' where c_id = '1';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

รูปที่ 2.36 แสดงการปรับปรุงโดยใช้คำสั่ง update

#### 2.14.9 การลบแถวจากตารางด้วยคำสั่ง delete

คำสั่ง delete เป็นคำสั่งที่ใช้ลบแถวออกจากตาราง โดยมีรูปแบบของคำสั่งดังนี้  
รูปแบบ

DELETE FROM table

[WHERE condition]

โดยที่ table หมายถึง ชื่อตาราง

Condition หมายถึง เงื่อนไขที่กำหนด

### Ex15 การปรับปรุงโดยใช้คำสั่ง delete

แสดงการปรับปรุงโดยใช้คำสั่ง delete ดังรูปที่ 2.37

```
mysql> delete from customer where c_id = '3';
Query OK, 1 row affected (0.02 sec)

mysql> select * from customer;
+----+-----+-----+
| c_id | c_fname | c_lname |
+----+-----+-----+
| 2 | suksit | kunsakon |
| 1 | ddddd | mekub |
+----+-----+-----+
2 rows in set (0.00 sec)
```

รูปที่ 2.37 แสดงการปรับปรุงโดยใช้คำสั่ง delete

### 2.14.10 การลบตารางและฐานข้อมูลด้วยคำสั่ง drop

เป็นคำสั่งที่ใช้สำหรับลบตารางที่ต้องการออกจากฐานข้อมูล หรือลบฐานข้อมูลทั้งหมด

รูปแบบ

```
DROP TABLE tablename;
```

โดยที่ tablename หมายถึง ชื่อตาราง

### Ex16 การลบตารางออกจากฐานข้อมูลด้วยคำสั่ง drop table

แสดงการลบตารางออกจากฐานข้อมูลด้วยคำสั่ง drop table ดังรูปที่ 2.38

```
mysql> drop table customer;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
Empty set (0.01 sec)
```

รูปที่ 2.38 แสดงการลบตารางออกจากฐานข้อมูลด้วยคำสั่ง drop table

รูปแบบ

```
DROP DATABASE databasename;
```

โดยที่ databasename หมายถึง ชื่อฐานข้อมูล

### Ex17 การลบฐานข้อมูลด้วยคำสั่ง drop database

แสดงการลบฐานข้อมูลด้วยคำสั่ง drop database ดังรูปที่ 2.39

```
mysql> drop database dbblissing;
Query OK, 0 rows affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| compasstravel |
| db_lissing |
| dddd |
| iccoms |
| mysql |
| test |
+-----+
6 rows in set (0.00 sec)
```

รูปที่ 2.39 แสดงการลบฐานข้อมูลด้วยคำสั่ง drop database

## 2.15 Macromedia Dreamweaver MX [2]

Dreamweaver เป็นเครื่องมือในการสร้างเว็บเพจที่มีประสิทธิภาพสูง เป็นผลิตภัณฑ์ของบริษัท Macromedia ซึ่งเป็นบริษัทที่พัฒนาและออกแบบโปรแกรมทางด้านกราฟฟิก รวมถึงเครื่องมือสร้างเว็บเพจต่างๆ ในโครงการนี้เราจะใช้ Dreamweaver เวอร์ชัน MX โดยความสามารถของ Dreamweaver จะคล้ายกับโปรแกรมประเภท FrontPage ,NetObject Fusion, HomeSite จุดเด่นของโปรแกรม Macromedia Dreamweaver MX ได้แก่

ผู้ใช้ไม่จำเป็นต้องศึกษาภาษา HTML มาก่อน ก็สามารถสร้างเอกสารเว็บได้ เพราะตัวโปรแกรมมีฟังก์ชันการทำงานแบบ HTML Generator บังคับควบคุมการทำงาน ได้จัดแบ่งเป็นหมวดหมู่ ช่วยให้การสั่งงานกระทำได้สะดวก และรวดเร็ว ใช้งานกับภาษาไทยได้ดี

สามารถสร้างภาพเคลื่อนไหว (Animation) โดยใช้รูปแบบของ Macromedia Director ด้วยคุณสมบัติ Animate Netscape และ CSS-P Layers ทำให้ได้ภาพเคลื่อนไหวบนเบราว์เซอร์ 4.0 โดยไม่ต้องอาศัย Plugin ใดๆ สามารถในการสร้างตาราง โดยการอิมพอร์ตจาก Text File สามารถปรับปรุงดูแลรักษาไซต์ เช่น การตรวจสอบลิงก์, สร้างรายงานแสดงผลการทดสอบการทำงาน มีฟังก์ชันในการโอนถ่ายข้อมูล (FTP) ขึ้นเครื่องแม่ข่าย (Server)

จากขั้นตอนและรายละเอียดทั้งหมด สามารถวิเคราะห์หาข้อมูลในการคำนวณอัตราดอกเบี้ย, อัตราค่าปรับ และศึกษาวิธีการออกแบบโปรแกรมโดยใช้ภาษา PHP และฟังก์ชันต่างๆ ของ PHP รวมถึงการใช้ภาษา Sql ในการออกแบบฐานข้อมูล เพื่อมารองรับการทำงานของฐานข้อมูล และ Macromedia Dreamweaver MX ช่วยในการออกแบบรูปร่างหน้าตาต่างๆ ได้อย่างมีประสิทธิภาพ



## บทที่ 3

### วิธีดำเนินการโครงการ

#### 3.1 การศึกษาปัญหา

ระบบสินเชื่อเป็นระบบที่ต้องการเก็บข้อมูลเป็นจำนวนมากทั้งจากประวัติของผู้กู้ ประวัติของผู้ค้ำประกัน ประวัติรถจักรยานยนต์และการทำสัญญา

3.1.1 พนักงานต้องคำนวณอัตราดอกเบี้ยชำระและค่าปรับเมื่อมีการชำระเงินช้า

3.1.2 ลูกค้ำจะทำสัญญาได้ที่ละครั้งคือ ทำสัญญาหนึ่งครั้งแล้วต้องชำระหนี้ให้หมดก่อนจึงจะทำสัญญาครั้งต่อไปได้

3.1.3 การสืบค้นข้อมูลในการหาประวัติผู้กู้ที่หนีการชำระ ซึ่งระบบเก่านี้ต้องพึ่งเอกสารในการค้นหาประวัติต่างๆรวมทั้งประวัติจักรยานยนต์ที่ต้องคอยตรวจสอบการหมดอายุของทะเบียนรถ

3.1.4 เจ้าของกิจการต้องการรู้ยอดการให้ออมมติสินเชื่อ ยอดการรับชำระหนี้แต่ละเดือนและที่สำคัญคือผู้ค้ำชำระหนี้เป็นระยะเวลานาน

#### 3.2 รวบรวมข้อมูลและวิเคราะห์ข้อมูล

หลังจากศึกษาปัญหาแล้วได้รวบรวมข้อมูลเพื่อเขียนระบบการทำงานโดยที่ การทำสัญญาแต่ละครั้งต้องประกอบไปด้วยข้อมูลผู้กู้ ข้อมูลผู้ค้ำประกัน ข้อมูลรถจักรยานยนต์ จำนวนเงินกู้ อัตราดอกเบี้ย อัตราเบี้ยปรับ จำนวนงวดที่ลูกค้ำต้องการผ่อนชำระ และวิเคราะห์การทำงานของระบบแบ่งตามขั้นตอนได้ดังนี้

3.2.1 การกรอกข้อมูลใหม่ของลูกค้ำ เมื่อลูกค้ำต้องการทำสัญญา ลูกค้ำจะให้ข้อมูลผู้กู้ ข้อมูลผู้ค้ำประกัน ข้อมูลรถจักรยานยนต์ จำนวนเงินที่ต้องการกู้ จำนวนงวดที่ต้องการผ่อนชำระให้พนักงาน พนักงานจะทำการตรวจสอบลูกค้ำว่าเคยทำสัญญาหรือไม่ โดยตรวจสอบที่ข้อมูลลูกค้ำ ข้อมูลผู้ค้ำประกัน ข้อมูลรถจักรยานยนต์ ถ้าข้อมูลใดข้อมูลหนึ่งไม่เคยทำสัญญามา พนักงานจะทำการกรอกข้อมูลนั้นๆใหม่ แต่ถ้าข้อมูลนั้นมีแล้ว พนักงานจะนำรหัสนั้นมาทำสัญญาต่อไป

3.2.2 การทำสัญญา พนักงานจะนำรหัสสัญญา รหัสผู้กู้ รหัสผู้ค้ำประกัน รหัสรถจักรยานยนต์ จำนวนเงินกู้ อัตราดอกเบี้ย อัตราเบี้ยปรับ จำนวนงวดที่ต้องการชำระ มากรอกในแบบฟอร์มการทำสัญญาและตั้งรหัสสัญญาขึ้นมา เพื่อให้ลูกค้ำนำเงินมาชำระตามกำหนดต่อไป

3.2.3 การรับชำระหนี้ ลูกค้ำนำรหัสสัญญามาให้พนักงานตามกำหนดผ่อนชำระ พนักงานทำการบันทึกข้อมูลการรับชำระเงินเพื่อเป็นหลักฐานให้กับลูกค้ำ

3.2.4 รายงานผลต่อเจ้าของกิจการ ทุกสิ้นเดือนพนักงานจะรายงานต่อเจ้าของกิจการว่าแต่ละเดือนมีลูกค้ามาทำสัญญาที่รายเป็นจำนวนเงินเท่าไร มีการรับชำระเงินเป็นจำนวนเงินเท่าไร และจำนวนผู้ชำระเงิน

### 3.3 นำข้อมูลและผลการวิเคราะห์มาออกแบบฐานข้อมูล

เมื่อทำการรวบรวมข้อมูลของระบบและได้วิเคราะห์ข้อมูลแล้วนั้นต้องนำผลการวิเคราะห์นั้นมาออกแบบฐานข้อมูลเพื่อเตรียมจัดทำโปรแกรมต่อไป โดยแบ่งประเภทข้อมูลสำคัญๆได้ดังนี้

1. ข้อมูลผู้กู้
2. ข้อมูลผู้ค้ำประกัน
3. ข้อมูลรถจักรยานยนต์
4. ข้อมูลการทำสัญญา
5. ข้อมูลการรับชำระหนี้

เมื่อทราบข้อมูลหลักๆแล้วนำข้อมูลมาวิเคราะห์เพื่อแก้ปัญหาต่างๆ ได้ดังนี้

3.3.1 ข้อมูลผู้กู้ ต้องเก็บข้อมูลสำคัญๆคือ ชื่อ-สกุลผู้กู้ เลขที่บัตรประจำตัวประชาชน ที่อยู่ปัจจุบัน สถานที่ทำงาน รายได้เฉลี่ยต่อวันต่อเดือนและสร้างรหัสผู้กู้

3.3.2 ข้อมูลผู้ค้ำประกัน ต้องเก็บข้อมูลสำคัญๆคือ ชื่อ-สกุลผู้ค้ำประกัน เลขที่บัตรประจำตัวประชาชน ที่อยู่ปัจจุบัน สถานที่ทำงาน รายได้เฉลี่ยต่อวันต่อเดือนสร้างรหัสผู้ค้ำประกัน

3.3.3 ข้อมูลรถจักรยานยนต์ ต้องเก็บข้อมูลสำคัญๆคือ หมายเลขตัวถัง หมายเลขเครื่อง หมายเลขทะเบียนรถ ประเภทรถ ผู้ถือกรรมสิทธิ์ วันหมดอายุทะเบียนรถ

3.3.4 ข้อมูลการทำสัญญา การทำสัญญาแต่ละครั้งต้องประกอบไปด้วย ข้อมูลผู้กู้ ข้อมูลผู้ค้ำประกัน ข้อมูลรถจักรยานยนต์ จำนวนเงินกู้ อัตราดอกเบี้ย อัตราเบี้ยปรับ จำนวนงวดผ่อนชำระและสร้างรหัสสัญญา จึงจะทำสัญญาได้ แต่จำนวนงวดแต่ละงวดต้องมีการนับวันเพื่อหาค่าปรับที่เกิดจากการชำระหนี้ล่าช้าของลูกค้าจึงต้องแยกการเก็บข้อมูลของจำนวนงวดได้โดย เก็บวันที่ต้องรับชำระหนี้ไว้เพื่อนับวันที่แต่ละงวด เก็บจำนวนวันที่ลูกค้าไม่มาชำระหนี้เพื่อคำนวณค่าปรับต่อไป

3.3.5 ข้อมูลการรับชำระหนี้ การรับชำระหนี้ต้องเก็บจำนวนเงิน รหัสสัญญา วันที่รับชำระหนี้ เลขที่ใบเสร็จ ที่ลูกค้านำมาชำระและเพื่อเป็นหลักฐานกับลูกค้าว่าลูกค้านั้นได้ทำการรับชำระหนี้แล้ว

### 3.4 ออกแบบหน้าตาเว็บเพจ

หลังจากออกแบบฐานข้อมูลแล้วทำให้ทราบว่าต้องเก็บข้อมูลส่วนใดบ้างนั้นในส่วนของการออกแบบเว็บเพจจึงมีความสำคัญในส่วนรับข้อมูลจากระบบภายนอกออกแบบหน้าตาได้ดังนี้

3.4.1 หน้าตาเว็บเพจหลัก ประกอบไปด้วยเฟรมหลักและเฟรมรอง เฟรมหลักคือหน้าตาแสดงผลจากการเข้ามาของเฟรมรอง เฟรมรองคือเมนูของโปรแกรม โดยหน้าตาเว็บเพจหลักมีหน้าตาเป็นดังรูปที่ 3.1

รูปที่ 3.1 หน้าตาหลัก

3.4.2 หน้าตาเว็บเพจการรับข้อมูล เป็นการออกแบบเพื่อการรับข้อมูลเข้าในตัวอย่างรูปที่ 3.2 เป็นตัวอย่างการรับข้อมูลของเมนู กรอกประวัติรถจักรยานยนต์

รูปที่ 3.2 ส่วนของการรับข้อมูล

3.4.3 หน้าตาเว็บเพจการแสดงผลรายงาน ในรูปที่ 3.3 เป็นตัวอย่างหน้าตาเว็บเพจการออกแบบรายงานของการทำสัญญาเพื่อสรุปให้เจ้าของกิจการ

#### รายงานการเปิดสัญญา

ระหว่างวันที่ ๒๕๖ ถึง วันที่ ๒๕๖

รหัสสัญญา	วันที่ทำสัญญา	เงินกู้	ดอกเบี้ย	เงินผลประโยชน์
๒๕๖	๒๕๖	๒๕๖	๒๕๖%	๒๕๖
	รวม	๒๕๖		๒๕๖

Back

### รูปที่ 3.3 ส่วนของการรายงาน

### 3.5 นำโปรแกรมที่ทำเสร็จแล้วไปทดลองใช้จริง

หลังจากทำโปรแกรมเสร็จจึงนำโปรแกรมทดลองงานจริง เพื่อศึกษาผลการทำงานของโปรแกรมที่จัดทำขึ้นว่าเป็นไปตามความต้องการของกิจการหรือไม่มากนักในและเก็บข้อมูลมาวิเคราะห์หาจุดบกพร่องในการแก้ไขครั้งต่อไป

### 3.6 สรุปผลการทดลอง

โปรแกรมที่จัดทำขึ้นต้องสามารถใช้งานได้จริง โดยแบ่งหน้าที่การใช้งานหลักๆ ได้ดังนี้

- 3.6.1 ต้องเก็บข้อมูลประวัติผู้กู้ ประวัติผู้ค้ำประกัน ประวัติรถจักรยานยนต์และสัญญาได้
- 3.6.2 โปรแกรมต้องแสดงวันหมดอายุของรถจักรยานยนต์ทำสัญญาอยู่ได้
- 3.6.3 โปรแกรมต้องคำนวณการคิดอัตราดอกเบี้ย ผลประโยชน์ของกิจการได้
- 3.6.4 โปรแกรมต้องรายงานการทำสัญญา รายงานรับชำระเงินและรายงานผู้ค้างชำระได้

จากทั้งหมดที่กล่าวมาแล้วในข้างต้น ไม่ว่าจะเป็นการวิเคราะห์และออกแบบระบบงานชื่อของรถจักรยานยนต์และการศึกษาการออกแบบเว็บเพจ เราจะนำไปออกแบบระบบสืบเชื้อออกแบบระบบฐานข้อมูลและจัดทำรูปแบบหน้าตาของเว็บไซต์ซึ่งจะกล่าวในบทต่อไป

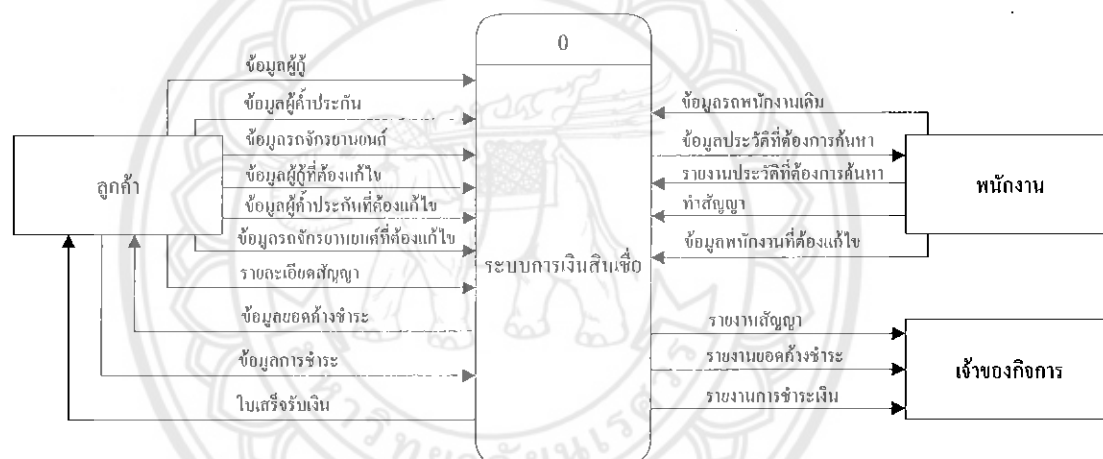
## บทที่ 4

### ผลการดำเนินโครงการ

หลังจากผ่านวิธีการดำเนินโครงการแล้วนั้น ขั้นตอนต่อไปนี้เป็นกรนำขั้นตอนที่ได้มาจากบทที่แล้วนำมาทำการวิเคราะห์และออกแบบข้อมูล เพื่อให้ได้ผลลัพธ์ออกมา โดยได้แบ่งเป็นขั้นตอนดังนี้

#### 4.1 ออกแบบระบบการทำงาน [3]

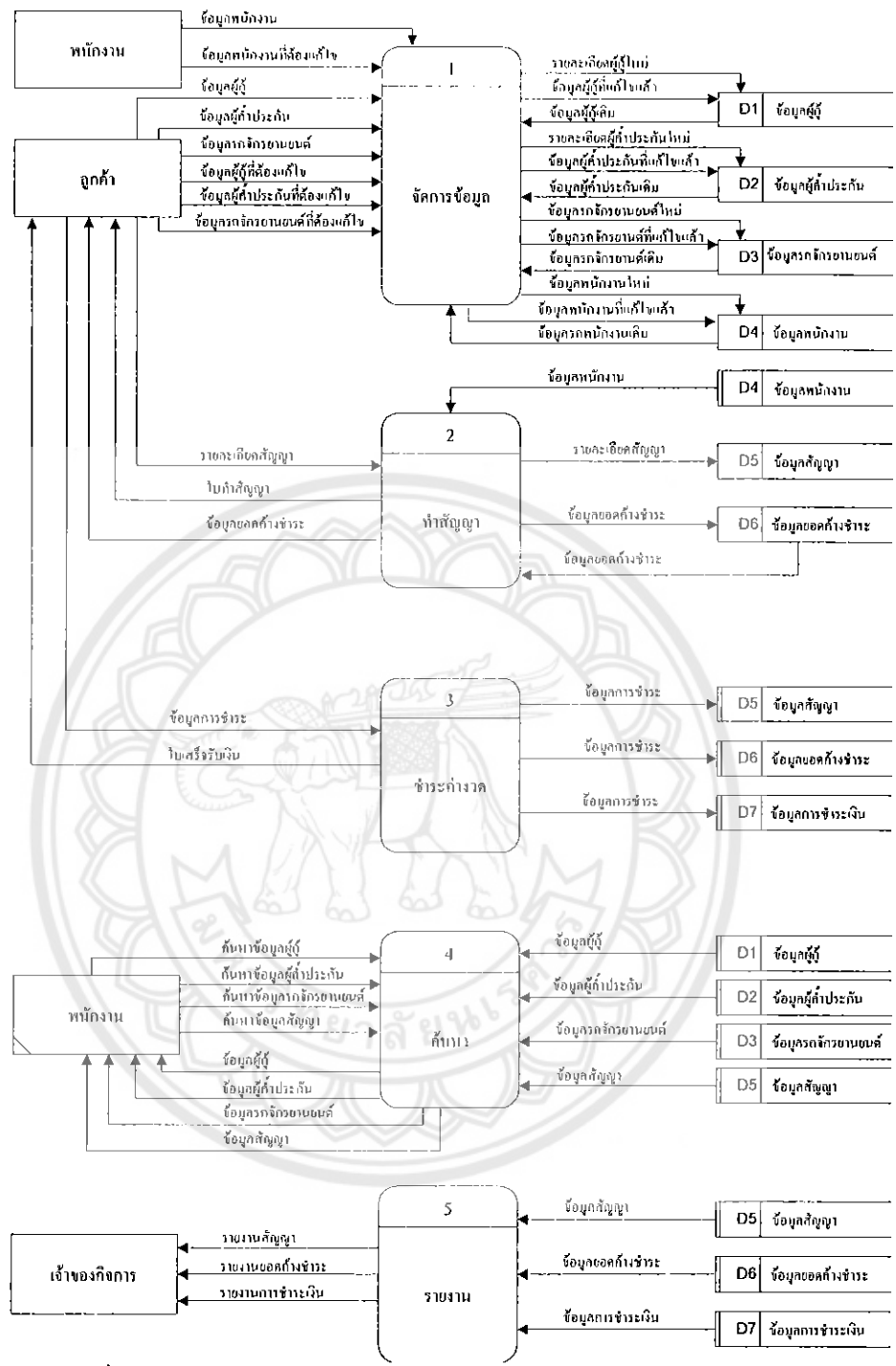
จากที่ได้ศึกษาปัญหาและข้อมูลแล้วนั้น สามารถนำมาสรุปได้เป็นลักษณะ context diagram ของระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์ได้ดังนี้



รูปที่ 4.1 Context diagram ของระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์

จากรูปที่ 4.1 Context diagram ของระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์ สามารถอธิบายได้โดยเริ่มจากลูกค้าเมื่อต้องการทำสัญญาจำเป็นต้องยื่นใบทำสัญญาให้กับระบบ โดยที่ใบทำสัญญานั้นจะประกอบไปด้วย ข้อมูลผู้กู้ ข้อมูลผู้ค้ำประกัน ข้อมูลรถจักรยานยนต์และรายละเอียดสัญญา จากนั้นพนักงานจึงจัดทำกรบันทึกข้อมูลการทำสัญญา และให้ข้อมูลยอดค้ำชำระให้แก่ลูกค้า เมื่อลูกค้าต้องการชำระเงิน ลูกค้าจะให้ข้อมูลการชำระเงินแก่ระบบ จากนั้นพนักงานจะทำการบันทึกข้อมูลให้แล้วระบบจะให้ใบเสร็จรับเงินคืนกับลูกค้า ส่วนเจ้าของกิจการจำเป็นต้องได้รับรายงานการทำสัญญา รายงานยอดค้ำชำระและรายงานการชำระเงินจากระบบ

จาก context diagram ข้างต้นนำมาเขียน data flow diagram เพื่อทำให้เห็นถึงระบบการทำงานให้ละเอียดขึ้น ซึ่งจะแบ่งเป็น 5 กระบวนการ ได้แก่ จัดการข้อมูล ทำสัญญา ชำระงวดค้ำหน้าและรายงาน ดังต่อไปนี้



รูปที่ 4.2 Data flow diagram ของระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์

จากรูปที่ 4.2 Data flow diagram ของระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์ สามารถอธิบายการทำงานของแต่ละ Process ได้ดังต่อไปนี้

- Data flow diagram process 1 จัดการข้อมูล ถูกทำให้ข้อมูลผู้กู้ ข้อมูลผู้ค้าประกันและข้อมูลรถจักรยานยนต์ ให้กับระบบจัดการข้อมูล พนักงานให้ข้อมูลพนักงานกับระบบจัดการข้อมูล ระบบจัดการข้อมูลก็จะทำการบันทึกหรือแก้ไขข้อมูลนั้นในฐานข้อมูล
- Data flow diagram process 2 ทำสัญญา ถูกทำให้รายละเอียดสัญญาแก่ระบบ ระบบจะทำการบันทึกสัญญาโดยมีข้อมูลของพนักงานเพื่อยืนยันว่าพนักงานคนไหนเป็นคนรับทำสัญญา และระบบจะบันทึกข้อมูลยอดค้ำชำระไว้ด้วยเพื่อเป็นระบบจะได้ค่าปรับในแต่ละงวดได้แล้ว ระบบจะให้ใบทำสัญญาและยอดค้ำชำระแก่ลูกค้า
- Data flow diagram process 3 ชำระค่างวด ถูกทำให้ข้อมูลการชำระ(รหัสสัญญาจำนวนเงิน)แก่ระบบ ระบบชำระค่างวดจะทำการปรับปรุงข้อมูลใน ข้อมูลสัญญา ข้อมูลยอดค้ำชำระ ข้อมูลการชำระเงิน
- Data flow diagram process 4 ค้นหา พนักงานสามารถค้นหาข้อมูลผู้กู้ ข้อมูลผู้ค้าประกัน ข้อมูลรถจักรยานยนต์ ข้อมูลสัญญา โดยพนักงานกรอกคำสำคัญให้กับระบบ ระบบจะให้ข้อมูลที่พนักงานต้องการค้นหาเกี่ยวกับค้นหา
- Data flow diagram process 5 รายงาน ระบบจะให้รายงานการทำสัญญา รายงานยอดค้ำชำระและรายงานการชำระ ให้กับเจ้าของกิจการ

## 4.2 ออกแบบฐานข้อมูล [4]

เมื่อทราบการทำงานของระบบแล้วโดยดูได้จาก Data flow diagram ข้างต้นนั้น สามารถออกแบบฐานข้อมูล โดยแบ่งการเก็บข้อมูลตาม Data flow diagram เป็น 7 ตารางและได้เขียนให้อยู่ในรูปของ Datadictionary เพื่อให้ง่ายต่อการทำความเข้าใจดังนี้

### 4.2.1 Datadictionary

1. customer (ผู้กู้) ฐานข้อมูลของผู้กู้ที่มีการจัดเก็บข้อมูลประวัติของลูกค้าเพื่อสะดวกในการค้นหาชื่อ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น โดยมีการจัดเก็บความสำคัญเช่น

- รหัสผู้กู้(c\_id) เป็นคีย์หลักและเก็บข้อมูลเป็นประเภท varchar(6) เพราะรหัสลูกค้านั้นมี การเก็บข้อมูลในลักษณะดังนี้ e00000 e00001 เป็นต้น
- ชื่อ-สกุล(c\_name) เก็บข้อมูลเป็นประเภท varchar(50) ซึ่งเป็นการเก็บข้อมูลตัวอักษร ทั้งชื่อและนามสกุล
- เลขที่บัตรประจำตัวประชาชน(c\_p\_id) เก็บข้อมูลเป็นประเภท decimal(13,0) เลขที่ บัตรประจำตัวประชาชนเป็นตัวเลข 13 หลักและเป็นข้อมูลที่ใช้ป้องกันความซ้ำซ้อนลูกค้า

ส่วนข้อมูลที่เหลือของฐานข้อมูลผู้กู้ที่ไม่ได้อธิบายไว้จากข้างต้นนั้นสามารถรายละเอียดเพิ่มได้จากตารางที่ 4.1 ดังนี้

ตารางที่ 4.1 ตาราง customer (ผู้กู้)

NO.	Attribute Name	Type	Description	Key
1	c_id	varchar(6)	รหัส	PK
2	c_name	varchar(50)	ชื่อ - สกุล	
3	c_age	decimal(2,0)	อายุ	
4	c_status	varchar(5)	สถานะผู้กู้	
5	c_m_name	varchar(50)	ชื่อคู่สมรส	
6	c_m_age	decimal(2,0)	อายุคู่สมรส	
7	c_h_add	varchar(100)	ที่อยู่ปัจจุบัน	
8	c_h_tel	varchar(50)	เบอร์โทรศัพท์	
9	c_h_near	varchar(50)	สถานที่ใกล้เคียง	
10	c_p_type	varchar(50)	บัตรแสดงตนถือบัตร	
11	c_p_id	decimal(13,0)	เลขที่บัตรแสดงตน	
12	c_p_by	varchar(30)	ออกให้โดย	
13	c_p_stdte	date	วันออกบัตร	
14	c_p_endate	date	บัตรหมดอายุ	
15	c_p_add	varchar(100)	ที่อยู่ในบัตรประจำตัว	
16	c_w	varchar(50)	อาชีพ	
17	c_w_class	varchar(50)	ตำแหน่ง	
18	c_w_shop	varchar(50)	แผนก	
19	c_w_name	varchar(50)	ชื่อสถานที่ทำงาน	
20	c_w_add	varchar(100)	ที่ตั้งที่ทำงาน	
21	c_w_tel	varchar(20)	เบอร์โทรศัพท์ที่ทำงาน	
22	c_re_d	decimal(10,0)	รายได้ผู้กู้ต่อวัน	
23	c_re_m	decimal(10,0)	รายได้ผู้กู้ต่อเดือน	
24	c_m_re_d	decimal(10,0)	รายคู่สมรสต่อวัน	
25	c_m_re_m	decimal(10,0)	รายได้คู่สมรสต่อเดือน	
26	c_re_other	varchar(10)	รายได้จากอื่น	
27	c_re_other_m	decimal(10,0)	รายได้เฉลี่ยต่อเดือน	

2. warrant (ผู้ค้ำประกัน) ฐานข้อมูลของผู้ค้ำประกันนั้นมีการจัดเก็บข้อมูลประวัติของผู้ค้ำประกันเพื่อสวดอกในการค้นหาชื่อ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น โดยมีการจัดเก็บความสำคัญเช่น

- รหัสผู้ค้ำประกัน(w\_id) เป็นคีย์หลักและเก็บข้อมูลเป็นประเภท varchar(6) เพราะรหัสผู้ค้ำประกันนั้นมีการเก็บข้อมูลในลักษณะดังนี้ e00000 e00001 เป็นต้น



- ชื่อ-สกุล(w\_name) เก็บข้อมูลเป็นประเภท varchar(50) ซึ่งเป็นการเก็บข้อมูลตัวอักษรทั้งชื่อและนามสกุล
  - เลขที่บัตรประจำตัวประชาชน(w\_p\_id) เก็บข้อมูลเป็นประเภท decimal(13,0) เลขที่บัตรประจำตัวประชาชนเป็นตัวเลข 13 หลักและเป็นข้อมูลที่ใช้ป้องกันความซ้ำซ้อนผู้ค้าประกัน
- ส่วนข้อมูลที่เหลือของฐานข้อมูลผู้ค้าประกันที่ไม่ได้อธิบายไว้จากข้างต้นนั้นสามารถดูรายละเอียดเพิ่มเติมได้จากตารางที่ 4.2 ดังนี้

ตารางที่ 4.2 ตาราง warrant (ผู้ค้าประกัน)

NO.	Attribute Name	Type	Description	Key
1	w_id	varchar(6)	รหัส	PK
2	w_name	varchar(50)	ชื่อ - สกุล	
3	w_age	decimal(2,0)	อายุ	
4	w_status	varchar(5)	สถานะผู้ค้าประกัน	
5	w_m_name	varchar(50)	ชื่อคู่สมรส	
6	w_m_age	decimal(2,0)	อายุคู่สมรส	
7	w_h_add	varchar(100)	ที่อยู่ปัจจุบัน	
8	w_h_tel	varchar(50)	เบอร์โทรศัพท์	
9	w_h_near	varchar(50)	สถานที่ใกล้เคียง	
10	w_p_type	varchar(50)	บัตรแสดงตนถือบัตร	
11	w_p_id	decimal(13,0)	เลขที่บัตรแสดงตน	
12	w_p_by	varchar(30)	ออกให้โดย	
13	w_p_stdate	date	วันออกบัตร	
14	w_p_endate	date	บัตรหมดอายุ	
15	w_p_add	varchar(100)	ที่อยู่ในบัตรประจำตัว	
16	w_w	varchar(50)	อาชีพ	
17	w_w_class	varchar(50)	ตำแหน่ง	
18	w_w_shop	varchar(50)	แผนก	
19	w_w_name	varchar(50)	ชื่อสถานที่ทำงาน	
20	w_w_add	varchar(100)	ที่ตั้งที่ทำงาน	
21	w_w_tel	varchar(20)	เบอร์โทรศัพท์ที่ทำงาน	
22	w_re_d	decimal(10,0)	รายได้ผู้ค้าประกันต่อวัน	
23	w_re_m	decimal(10,0)	รายได้ผู้ค้าประกันต่อเดือน	
24	w_m_re_d	decimal(10,0)	รายคู่สมรสต่อวัน	
25	w_m_re_m	decimal(10,0)	รายได้คู่สมรสต่อเดือน	

26	w_re_other	varchar(10)	รายได้จากอื่น	
27	w_re_other_m	decimal(10,0)	รายได้เฉลี่ยต่อเดือน	

3. moto (รถจักรยานยนต์) ฐานข้อมูลรถจักรยานยนต์นั้นมีการจัดเก็บข้อมูลรายละเอียดต่างๆเกี่ยวกับรถจักรยานยนต์ที่นำมาทำสัญญาสินเชื่อและได้ทำการจัดเก็บข้อมูลดังนี้

- หมายเลขตัวถัง(mo\_id) เป็นคีย์หลักและเป็นข้อมูลประเภท varchar(20) ที่เป็นคีย์หลัก เพราะหมายเลขตัวถังรถนั้นเป็นตัวอักษรที่ไม่ซ้ำกันจึงง่ายต่อการตรวจสอบว่าลูกค้านำรถคันเดียวกันมาทำสัญญาซ้ำซ้อนกันหรือไม่

- วันหมดอายุป้ายวงกลม(mo\_endate) เป็นข้อมูลประเภท date ซึ่งเป็นการเก็บข้อมูลวันหมดอายุของทะเบียนรถ เพื่อให้โปรแกรมมีการเตือนว่ารถจักรยานยนต์คันใดบ้างที่ทำสัญญาอยู่นั้นทะเบียนรถใกล้หมดอายุ

ส่วนข้อมูลที่เหลือของฐานข้อมูลรถจักรยานยนต์ที่ไม่ได้อธิบายไว้จากข้างต้นนั้นสามารถดูรายละเอียดเพิ่มได้จากตารางที่ 4.3 ดังนี้

ตารางที่ 4.3 ตาราง moto (รถจักรยานยนต์)

NO	Attribute Name	Type	Description	Key
1	mo_band	varchar(20)	ยี่ห้อรถจักรยานยนต์	
2	mo_model	varchar(20)	รุ่นรถจักรยานยนต์	
3	mo_type	varchar(20)	แบบรถจักรยานยนต์	
4	mo_col	varchar(10)	สีรถจักรยานยนต์	
5	mo_id_reg	varchar(20)	หมายเลขทะเบียน	
6	mo_id_mach	varchar(20)	หมายเลขเครื่อง	
7	mo_id	varchar(20)	หมายเลขตัวถัง	PK
8	mo_age	decimal(2,0)	อายุรถจักรยานยนต์	
9	mo_endate	date	วันหมดอายุป้ายวงกลม	
10	mo_owner	varchar(50)	ผู้ถือกรรมสิทธิ์	
11	mo_owner_endate	date	บัตรประชาชนผู้ถือกรรมสิทธิ์หมดอายุ	

4. user (พนักงาน) ฐานข้อมูลพนักงานเก็บข้อมูลเกี่ยวกับ ชื่อ รหัสผ่านและระดับงาน มีไว้เพื่อให้โปรแกรมตรวจสอบผู้ใช้งานโดยมีรายละเอียดในการจัดเก็บดังนี้

- ชื่อในการlogin(usrname) เป็นคีย์หลักและเป็นข้อมูลประเภท varchar(10) ซึ่งเป็นคีย์หลักเพราะให้ชื่อในการ login ไม่ซ้ำกัน และเก็บข้อมูลเป็นตัวหนังสือ

- ระดับผู้ใช้งาน(level) เป็นข้อมูลประเภท decimal(1,0) เพราะเก็บตัวเลข 1 หลัก มีไว้เพื่อแบ่งระดับการใช้งานโปรแกรมคือ ถ้า level = 1 เป็นการใช้งานทั่วไปได้แต่ไม่สามารถสั่งคืนใบเสร็จที่ส่งจ่ายผิด ถ้า level = 2 นั้นสามารถคืนใบเสร็จที่ส่งจ่ายผิดได้

ส่วนข้อมูลที่เหลือของฐานข้อมูลพนักงานที่ไม่ได้อธิบายไว้จากข้างต้นนั้นสามารถดูรายละเอียดเพิ่มได้จากตารางที่ 4.4 ดังนี้

ตารางที่ 4.4 ตาราง user (พนักงาน)

NO.	Attribute Name	Type	Description	Key
1	username	varchar(10)	ชื่อในการ login	PK
2	password	varchar(10)	รหัสในการ login	
3	name	varchar(50)	ชื่อผู้ใช้งาน	
4	level	decimal(1,0)	ระดับผู้ใช้งาน 1 = พนักงาน 2 = เจ้าของกิจการ	

5. promise (สัญญา) ฐานข้อมูลสัญญาเป็นการนำคีย์หลักของข้อมูลผู้กู้ ข้อมูลผู้ค้ำประกัน ข้อมูลรถจักรยานยนต์ มาทำสัญญากันโดยที่มีการกำหนดเงินกู้ อัตราดอกเบี้ย อัตราเบี้ยปรับ จำนวนการผ่อนชำระ โดยมีรายละเอียดในการจัดเก็บข้อมูลดังนี้

- รหัสสัญญา(p\_id) เป็นคีย์หลักและเป็นข้อมูลประเภท varchar(6) ซึ่งเป็นการเก็บข้อมูลเลขที่สัญญาดังตัวอย่างเช่น สัญญาเลขที่ e00002 e00003 เป็นต้น

- สถานะทำสัญญา (p\_status) เป็นข้อมูลประเภท decimal(1,0) เพราะเป็นการเก็บข้อมูลของสถานะการทำสัญญาว่า ถ้าเป็น 1 ยังทำสัญญาอยู่ ถ้าเป็น 2 คือปิดสัญญาแล้ว

ส่วนข้อมูลที่เหลือของฐานข้อมูลสัญญาที่ไม่ได้อธิบายไว้จากข้างต้นนั้นสามารถดูรายละเอียดเพิ่มได้จากตารางที่ 4.5 ดังนี้

ตารางที่ 4.5 ตาราง promise (สัญญา)

NO.	Attribute Name	Type	Description	Key
1	p_id	varchar(6)	รหัสสัญญา	PK
2	c_id	varchar(6)	รหัสผู้กู้	FK
3	w_id	varchar(6)	รหัสผู้ค้ำประกัน	FK
4	p_relation	varchar(50)	ความสัมพันธ์ผู้กู้กับผู้ค้ำประกัน	
5	mo_id	varchar(30)	หมายเลขตัวถังรถจักรยานยนต์	FK
6	p_start	date	วันเริ่มทำสัญญา	

7	p_money	decimal(10,0)	จำนวนเงินกู้	
8	p_dok	decimal(5,0)	อัตราดอกเบี้ยกู้	
9	p_dok_afine	decimal(5,0)	อัตราดอกเบี้ยค่าปรับ	
10	p_ben	decimal(10,0)	เงินผลประโยชน์	
11	p_rev_total	decimal(10,0)	รวมเงินที่ได้รับ	
12	p_time	char(3)	ระยะเวลาในการผ่อนชำระ	
13	p_rev_mo	decimal(10,0)	เงินรับชำระรายเดือน	
14	p_rev_time	date	วันที่เริ่มรับเงินผ่อนชำระ	
15	username	varchar(50)	พนักงานรับทำสัญญา	FK
16	p_status	decimal(1,0)	สถานะทำสัญญา 1=ชำระเงินยังไม่ครบทุกงวด 2=ชำระเงินครบทุกงวด	

6. promise\_status (ยอดค้างชำระ) เป็นการเก็บข้อมูลเกี่ยวกับจำนวนเงินของแต่ละงวดว่าคงค้างชำระเท่าไร เพื่อสามารถตรวจสอบได้ว่าลูกค้านั้นชำระเงินช้าจนทำให้เกิดค่าปรับหรือไม่ โดยมีรายละเอียดการเก็บข้อมูลดังนี้

- รหัสสัญญา(p\_id)และงวดที่(ps\_count\_nguad) เป็นคีย์คู่อันดับกัน โดยที่รหัสสัญญาเก็บข้อมูลเป็นประเภท varchar(6) เหมือนดังที่ได้กล่าวมาจากตารางที่ 4.5 ส่วนงวดที่นั้นมีการเก็บข้อมูลเป็นประเภท decimal(2,0) เพราะเป็นการเก็บเลขที่งวดเพียง 2 หลัก

- วันที่ต้องมาชำระ(ps\_date) เก็บข้อมูลเป็นประเภท date เพื่อให้ทราบว่าวันที่ต้องชำระเงินแต่ละงวดเป็นวันที่เท่าไรเพราะถ้าลูกค้าชำระเงินช้าเกินกว่า 8 วันนั้นจะได้มีการปรับเกิดขึ้น

- สถานะในแต่ละงวด(ps\_id) เก็บข้อมูลเป็นประเภท decimal(1,0) เพื่อให้ทราบสถานะการชำระเงินจากลูกค้าว่าแต่ละมีการชำระเงินอย่างไร คือถ้าเป็น 1 ยังมีการรับชำระเงิน ถ้าเป็น 2 มีการชำระเงินแต่ยังชำระไม่หมด ถ้าเป็น 3 คือชำระเงินในงวดนั้นๆหมดแล้ว

ส่วนข้อมูลที่เหลือของฐานข้อมูลยอดค้างชำระที่ไม่ได้อธิบายไว้จากข้างต้นนั้นสามารถดูรายละเอียดเพิ่มได้จากตารางที่ 4.6 ดังนี้

ตารางที่ 4.6 ตาราง promise\_status (ยอดค้างชำระ)

NO.	Attribute Name	Type	Description	Key
1	p_id	varchar(6)	รหัสสัญญา	PK,FK
2	ps_count_nguad	decimal(2,0)	งวดที่	PK
3	ps_date	date	วันที่ต้องมาชำระ	
4	ps_money	decimal(10,2)	เงินต้นที่ต้องชำระ	
5	ps_dok	decimal(10,2)	เงินดอกเบี้ยที่ต้องชำระ	
6	ps_time_afine	decimal(30,0)	จำนวนวันปรับชำระช้า	

7	ps_afine	decimal(10,2)	ค่าปรับ	
8	ps_id	decimal(1,0)	สถานะในแต่ละงวด 1=ยังไม่มีชำระเงิน 2=มีการชำระเงินแต่ไม่ครบ 3=ชำระเงินครบ	

7. paid\_detail (รายละเอียดการรับชำระ) เป็นการเก็บข้อมูลการรับชำระเงินของลูกค้าและเพื่อให้ทราบว่าลูกค้าชำระเงินงวดที่เท่าไร จำนวนเงินเท่าไร ใบเสร็จเลขที่เท่าไร พนักงานคนใดเป็นคนรับเรื่อง โดยมีรายละเอียดในการจัดเก็บข้อมูลดังนี้

- เลขที่ใบเสร็จ(pd\_idmoney) เป็นคีย์หลักและเก็บข้อมูลเป็นประเภท int(6) เพื่อเก็บเลขที่ใบเสร็จจำนวน 6 หลัก และยังเป็น auto\_increment อีกด้วย

- หมายเหตุ(pd\_note) เป็นการเก็บข้อมูลประเภท varchar(20) เพื่อเก็บหมายเหตุการสั่งชำระเงินผิด

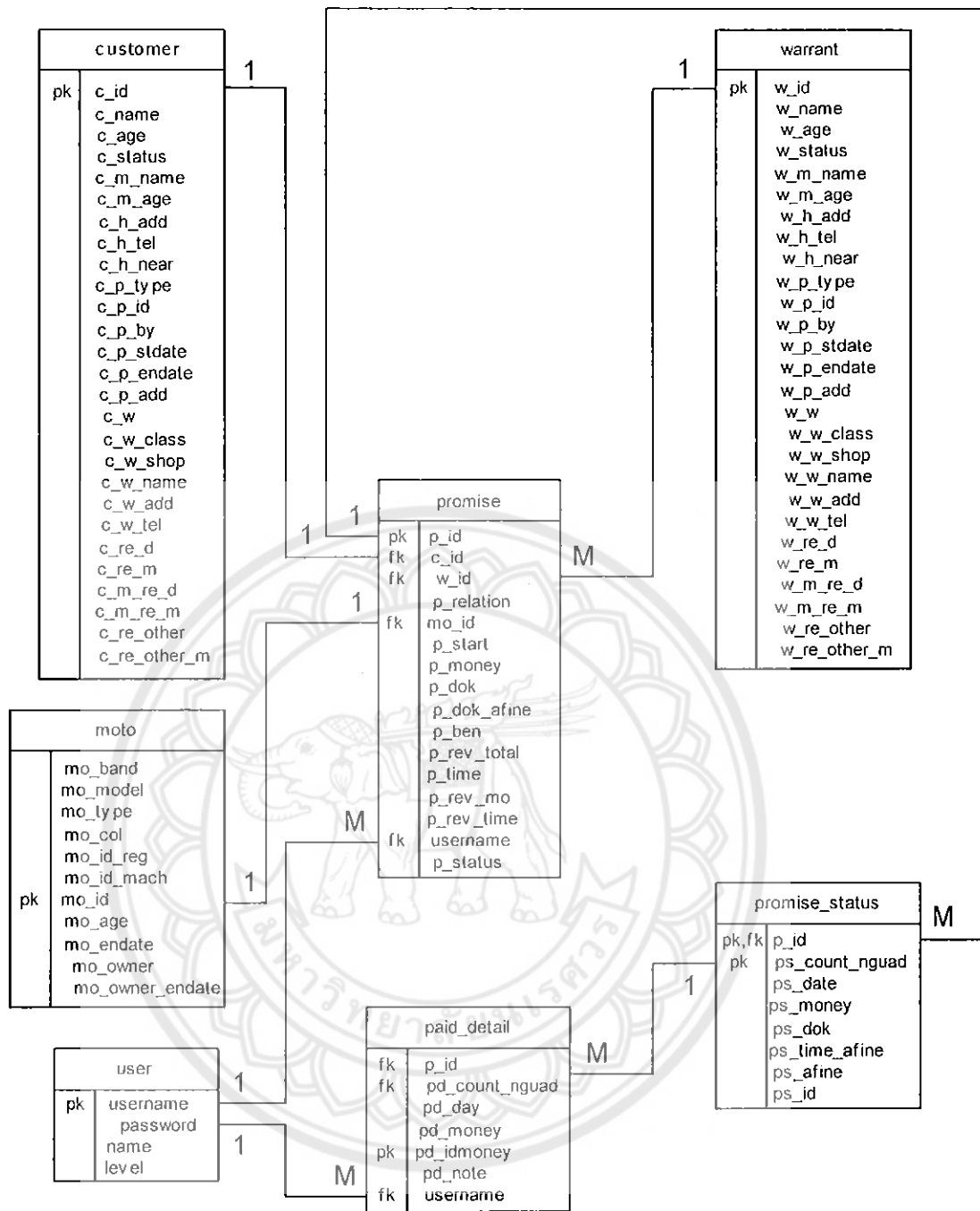
ส่วนข้อมูลที่เหลือของฐานข้อมูลยกเว้นชำระที่ไม่ได้อธิบายไว้จากข้างต้นนั้นสามารถดูรายละเอียดเพิ่มได้จากตารางที่ 4.7 ดังนี้

ตารางที่ 4.7 ตาราง paid\_detail (รายละเอียดการรับชำระ)

NO.	Attribute Name	Type	Description	Key
1	p_id	varchar(6)	รหัสสัญญา	FK
2	pd_count_nguad	decimal(3,0)	เลขที่งวดที่รับชำระ	FK
3	pd_day	date	วันที่รับชำระ	
4	pd_money	decimal(10,2)	จำนวนเงิน	
5	pd_idmoney	int(6)	เลขที่ใบเสร็จ	PK
6	pd_note	varchar(20)	หมายเหตุ	
7	username	varchar(50)	พนักงานรับเงิน	FK

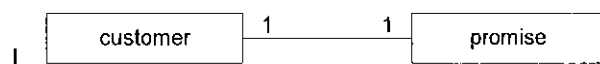
#### 4.2.2 ER Diagram

หลังจากที่ได้ทำการเขียน data dictionary เสร็จแล้วนั้นจึงนำมาเขียนความสัมพันธ์ของตารางแต่ละตารางเพื่อให้ง่ายต่อการเขียนโปรแกรมติดต่อฐานข้อมูล โดยเขียนความสัมพันธ์ได้ดังรูปต่อไปนี้

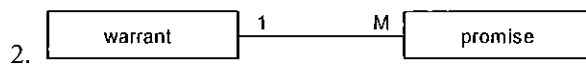


รูปที่ 4.3 ER Diagram

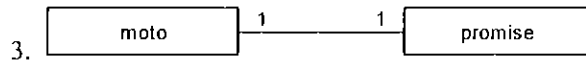
จากรูปที่ 4.3 ER Diagram ข้างต้นอธิบายความสัมพันธ์ได้ดังนี้



ความสัมพันธ์ระหว่างตาราง customer และตาราง promise สามารถอธิบายได้ดังนี้ ผู้ใช้หนึ่งคนสามารถทำสัญญาได้หนึ่งครั้ง โดยในส่วนของตาราง customer จะมี c\_id เป็นคีย์หลัก และในส่วนของตาราง promise จะมี p\_id เป็นคีย์หลักและ c\_idของตาราง customer เป็นคีย์นอก



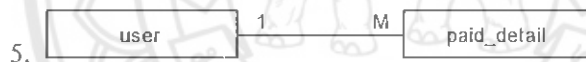
ความสัมพันธ์ระหว่างตาราง warrant และตาราง promise สามารถอธิบายได้ดังนี้ ผู้ค้าประกันหนึ่งคนสามารถค้าประกันได้หลายสัญญา โดยในส่วนของตาราง warrant จะมี w\_id เป็นคีย์หลัก และในส่วนของตาราง promise จะมี p\_id เป็นคีย์หลักและ w\_id ของตาราง warrant เป็นคีย์นอก



ความสัมพันธ์ระหว่างตาราง moto และตาราง promise สามารถอธิบายได้ดังนี้ รถจักรยานยนต์หนึ่งคันสามารถทำสัญญาได้หนึ่งสัญญา โดยในส่วนของตาราง moto จะมี mo\_id เป็นคีย์หลัก และในส่วนของตาราง promise จะมี p\_id เป็นคีย์หลักและ mo\_id ของตาราง moto เป็นคีย์นอก



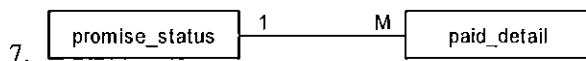
ความสัมพันธ์ระหว่างตาราง user และตาราง promise สามารถอธิบายได้ดังนี้ พนักงานหนึ่งคนสามารถบันทึกข้อมูลการทำสัญญาได้สัญญา โดยในส่วนของตาราง user จะมี username เป็นคีย์หลัก และในส่วนของตาราง promise จะมี p\_id เป็นคีย์หลักและ username ของตาราง user เป็นคีย์นอก



ความสัมพันธ์ระหว่างตาราง user และตาราง paid\_detail สามารถอธิบายได้ดังนี้ พนักงานหนึ่งคนสามารถบันทึกข้อมูลการรับชำระหนี้ได้หลายครั้ง โดยในส่วนของตาราง user จะมี username เป็นคีย์หลัก และในส่วนของตาราง paid\_detail จะมี pd\_idmoney เป็นคีย์หลักและ username ของตาราง user เป็นคีย์นอก



ความสัมพันธ์ระหว่างตาราง promise และตาราง promise\_status สามารถอธิบายได้ดังนี้ สัญญาหนึ่งสัญญาต้องมีรายละเอียดยอดค้างชำระได้หลายงวด โดยในส่วนของตาราง promise จะมี p\_id เป็นคีย์หลัก และในส่วนของตาราง promise\_status จะมี p\_id และ ps\_count\_nguad เป็นคีย์คู่อันดับและ p\_id ของตาราง promise เป็นคีย์นอก



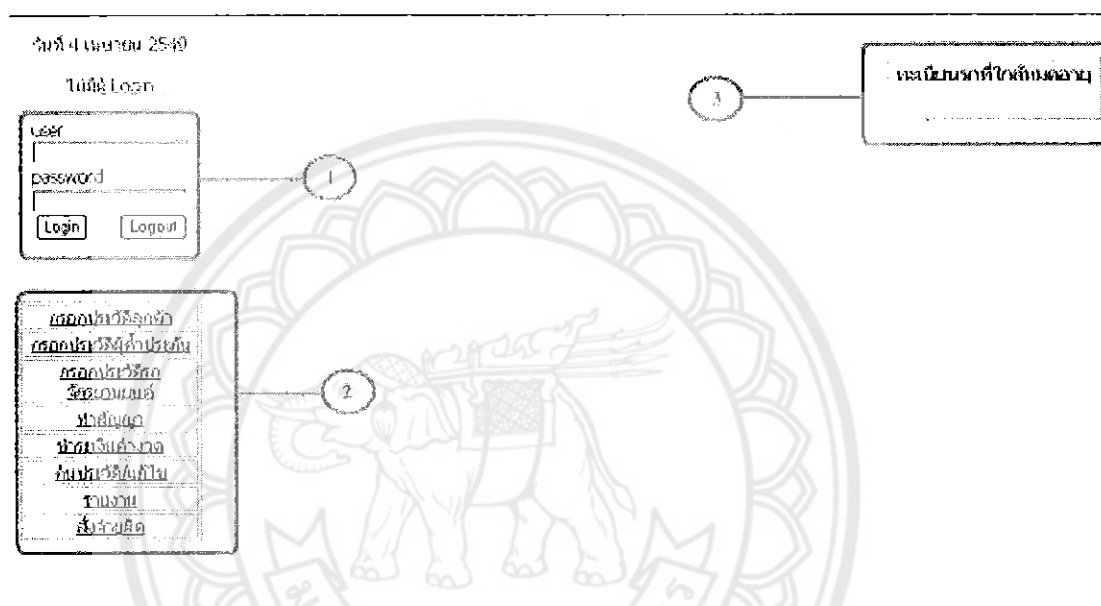
ความสัมพันธ์ระหว่างตาราง promise\_status และตาราง paid\_detail สามารถอธิบายได้ดังนี้ ยอดค้างชำระแต่ละงวดสามารถผ่อนชำระได้มากกว่าหนึ่งครั้ง โดยในส่วนของตาราง promise\_status จะมี p\_id และ ps\_count\_nguad เป็นคีย์คู่อันดับ และในส่วนของ

ตาราง paid\_detail จะมี ps\_count\_nguad เป็นคีย์หลักและ p\_id และ ps\_count\_nguad ของตาราง promise\_status เป็นคีย์นอก

#### 4.3 ออกแบบหน้าตาเว็บเพจ

เมื่อทราบการทำงานของระบบและการเก็บข้อมูลแล้วนั้น นำมาวิเคราะห์ข้อมูลเพื่อจัดทำหน้าตาเว็บเพจได้ดังนี้

##### 4.3.1 เว็บเพจในส่วนของการล็อกอินและเมนูหลัก



รูปที่ 4.4 เว็บเพจล็อกอินและเมนูหลัก

รูปที่ 4.4 ประกอบด้วย 3 ส่วน โดยมีรายละเอียดดังนี้

1. การ Login ถ้าต้องการเข้าใช้ระบบต้องทำการ Login เพื่อทำงานในระบบและเมื่อต้องการออกจากระบบให้กด Logout เพื่อออกจากระบบ

2. เมนูการกรอประวัติมี 8 ส่วนดังนี้

- กรอกประวัติลูกค้า
- กรอกประวัติผู้ค้าประกัน
- กรอกประวัติรถจักรยานยนต์
- ทำสัญญา
- ชำระค่างวด
- คืนประวัติ/แก้ไข
- รายงาน





- โทรศัพท์
- อาชีพ
- บัตรแสดงตน
- สถานที่ใกล้เคียง
- ตำแหน่ง

#### 4.3.3 เว็บเพจการทำสัญญา

หาสัญญาผู้

หาสัญญาวันที่	1	ตุลาคม	2548
รหัสสัญญาผู้			
รหัสผู้กู้			
รหัสผู้ค้ำประกัน			
ความสัมพันธ์ผู้กู้กับผู้ค้ำประกัน			
หมายเลขตัวถัง			
วงเงินกู้			บาท
อัตราดอกเบี้ย			%
อัตราดอกเบี้ยปรับ			%
จำนวนงวด			งวด
เริ่มจ่ายงวดแรกวันที่	1	ตุลาคม	2548
<input type="button" value="Submit"/>			

รูปที่ 4.6 เว็บเพจการทำสัญญา

จากรูปที่ 4.6 แสดงเว็บเพจการทำสัญญาราถูเงินจะต้องกรอกข้อมูลดังต่อไปนี้

- ทำสัญญาวันที่
- วงเงินกู้
- รหัสสัญญาผู้
- อัตราดอกเบี้ย
- รหัสผู้กู้
- อัตราดอกเบี้ยปรับ
- รหัสผู้ค้ำประกัน
- จำนวนงวด
- ความสัมพันธ์ของผู้ค้ำประกัน
- เริ่มจ่ายงวดแรก
- หมายเลขตัวถัง

#### 4.3.4 เว็บเพจรายงานการเปิดสัญญา

รายงานการเปิดสัญญา  
ระหว่างวันที่ 9 ต.ค. 2530 ถึง วันที่ 1 ต.ค. 2555

รหัสสัญญา	วันที่ทำสัญญา	เงินกู้	ดอกเบี้ย	เงินผลประโยชน์	ผู้รับเรื่อง	สถานะ
e22222	22 ก.ย. 2548	1200000.00	2%	288000.00	หลงซื่อ	ทำสัญญาอยู่
e11111	22 ก.ย. 2548	12000.00	2%	2880.00		ปิดสัญญา
e7777	22 ก.ย. 2548	22000.00	2%	4400.00	administrator	ทำสัญญาอยู่
e33333	23 ก.ย. 2548	40000.00	3%	6000.00	พีรดาณย์ เมฆอับ	ทำสัญญาอยู่
e6666	28 ก.ย. 2548	10000.00	2%	800.00	administrator	ทำสัญญาอยู่
e4444	30 ก.ย. 2548	20000.00	2%	2400.00	administrator	ทำสัญญาอยู่
e5555	30 ก.ย. 2548	15000.00	2%	1500.00	administrator	ทำสัญญาอยู่
	รวม	1319000.00		305980.00		

[Back](#)

รูปที่ 4.7 เว็บเพจรายงานการเปิดสัญญา

รูปที่ 4.7 แสดงเว็บเพจรายงานการเปิดสัญญา โดยรายงานนี้จะแสดงรหัสของสัญญาเงินกู้ต่างๆวันที่ทำสัญญาแสดงจำนวนเงินที่กู้ อัตราดอกเบี้ย เงินผลประโยชน์ที่จะได้รับ แสดงผู้รับเรื่อง และสถานของสัญญา รวมถึงแสดงยอดเงินรวมทั้งหมด

ส่วนรายละเอียดเพิ่มเติมของการใช้งาน โปรแกรมการเงินสินเชื่อเพื่อรถจักรยานยนต์ทั้งหมดสามารถดูได้จากภาคผนวก

#### 4.4 การเขียน PHP ติดต่อกับฐานข้อมูล

หลังจากออกแบบหน้าตาเว็บเพจเสร็จแล้วจึงได้ทำการเขียน Code PHP เพื่อให้เว็บเพจที่เราได้ออกแบบไว้แล้วนั้นสามารถติดต่อกับฐานข้อมูลได้และยังสามารถทำให้เว็บเพจเราสามารถบันทึก แก้ไข ลบและค้นหาข้อมูลได้

ตัวอย่างการเขียน PHP ติดต่อกับฐานข้อมูล

```
<?php
// ชื่อ file startconnect.php

$hostname_startconnect = "localhost";           //การกำหนด server
$db_startconnect = "db_lissing";               //การกำหนด database
$username_startconnect = "root";               //การกำหนด user
$password_startconnect = "";                   //การกำหนด password

$startconnect = mysql_pconnect($hostname_startconnect, $username_startconnect,
$password_startconnect) or die(mysql_error()); //การส่งข้อมูลข้างต้นติดต่อกับ database
```

?>

ตัวอย่างการเขียน PHP ติดต่อกับฐานข้อมูลสามารถอธิบายได้ดังนี้ เริ่มจากการกำหนดค่าของ server database user และ password เมื่อกำหนดค่าได้แล้วจึงทำการติดต่อกับฐานข้อมูลโดยใช้คำสั่ง mysql\_connect(server,user,password) โดยตัวอย่างข้างต้นสามารถดูเพิ่มเติมได้ในภาคผนวก

#### ตัวอย่างการเขียน PHP เพื่อบันทึกข้อมูล

```
<?php
include("startconnect.php"); $sql = "INSERT INTO
moto(mo_band,mo_model,mo_type,mo_col,mo_id,mo_id_reg,mo_id_mach,mo_age,mo_endate,
mo_owner,mo_owner_endate)
VALUES('$mo_band','$mo_model','$mo_type','$mo_col','$mo_id','$mo_id_reg','$mo_id_mach','
$mo_age','$mo_endate','$mo_owner','$mo_owner_endate')";
$result = mysql_query($sql,$conn) or die() //การส่งข้อมูลที่ต้องการบันทึกลง database
?>
```

ตัวอย่างการเขียน PHP เพื่อบันทึกข้อมูลสามารถอธิบายได้ดังนี้ เริ่มจากการใช้คำสั่ง insert into moto(mo\_band) values('\$mo\_band') คือให้บันทึก ค่าของ mo\_band ลงในฟิลด์ mo\_band ของตาราง moto และใช้คำสั่ง mysql\_query เพื่อทำการส่งค่าที่กำหนดไว้ทำการติดต่อกับฐานข้อมูลเพื่อเพิ่มข้อมูลในตาราง moto โดยตัวอย่างข้างต้นสามารถดูเพิ่มเติมได้ในภาคผนวก

#### 4.5 ผลการทดลองใช้

หลังจากทำโปรแกรมเสร็จแล้วจึงนำมาทดลองใช้งานจริง โดยแบ่งเป็นข้อๆมีผลดังนี้

1. การจัดเก็บข้อมูลผู้กู้ ข้อมูลผู้ค้าประกัน ข้อมูลรถจักรยานยนต์และสัญญาสินเชื่อ ปรากฏว่าสามารถจัดเก็บข้อมูลได้และยังสามารถแก้ไขข้อมูลรวมถึงลบข้อมูลได้
2. การคำนวณอัตราดอกเบี้ยของการผ่อนชำระในแต่ละงวด โปรแกรมสามารถคำนวณอัตราดอกเบี้ยออกมาได้ถูกต้องจึงช่วยลดเวลาในการคิดคำนวณให้กับพนักงานได้
3. การชำระเงินในแต่ละงวด โปรแกรมสามารถชำระเงินในแต่ละงวดของลูกค้าและบอกได้ว่าลูกค้ามียอดค้างชำระเท่าไร โปรแกรมสามารถคืนยอดชำระที่เกิดจากพนักงานชำระเงินผิดได้
4. การคำนวณอัตราดอกเบี้ยปรับเมื่อเกิดการผ่อนชำระช้าของลูกค้า โปรแกรมสามารถคำนวณได้โดยอัตโนมัติว่าลูกค้าผ่อนชำระช้าเป็นจำนวนกี่วันและมีค่าปรับที่เกิดจากการผ่อนชำระช้าเป็นจำนวนเงินเท่าไร
5. การสืบค้นข้อมูลประวัติลูกค้า ประวัติผู้ค้าประกันและประวัติรถจักรยานยนต์ โปรแกรมสามารถสืบค้นได้จึงทำให้ช่วยลดเวลาพนักงานในการหาเอกสารต่างๆได้

6. โปรแกรมสามารถออกรายงาน การรับชำระเงินและการทำสัญญาให้กับเจ้าของกิจการ
7. โปรแกรมสามารถเตือนวันหมดอายุของทะเบียนรถจักรยานยนต์ที่อยู่ระหว่างการทำสัญญาอยู่ได้จึงช่วยให้ทราบว่าทะเบียนรถจักรยานยนต์คันในควรรต่อทะเบียน

หลังจากที่ได้ทำตามขั้นตอนข้างต้นตั้งแต่ออกแบบระบบ ออกแบบฐานข้อมูล ออกแบบหน้าตาเว็บเพจและเขียน Code PHP ติดต่อฐานข้อมูลแล้วนั้นจะได้โปรแกรมระบบการเงินสินเชื่อเพื่อรถจักรยานยนต์และหาผลทดลองโปรแกรมแล้ว ขั้นตอนต่อไปคือสรุปผลการทำโครงการการเงินสินเชื่อเพื่อรถจักรยานยนต์ต่อไป



## บทที่ 5

### บทสรุป

ในการจัดทำโครงการระบบสารสนเทศการเงินสินเชื่อเพื่อรถจักรยานยนต์ ได้นำความรู้เกี่ยวกับภาษา PHP ติดต่อฐานข้อมูล MySQL เป็นตัวจัดการฐานข้อมูลและ Dreamwaver MX จัดรูปร่างหน้าตาเว็บเพจ ซึ่งทำให้โครงการนี้สำเร็จได้ โครงการที่จัดทำขึ้นนั้น สามารถช่วยแก้ปัญหาที่เกิดขึ้นจากความล่าช้าในการคำนวณอัตราดอกเบี้ยของพนักงาน อีกทั้งจัดเก็บข้อมูลเกี่ยวกับลูกค้าทำให้มีความสะดวกต่อพนักงานในการค้นหาข้อมูลของลูกค้า โปรแกรมคิดคำนวณค่าปรับได้โดยอัตโนมัติและสามารถเดือนวันหมดอายุของทะเบียนรถที่กำลังทำสัญญาอยู่ได้

เพื่อให้ทราบว่าโครงการที่จัดทำขึ้นนี้สามารถใช้งานได้จริงมีประสิทธิภาพหรือไม่จึงได้ทดสอบโดยให้พนักงานสินเชื่อทดลองใช้งานโปรแกรม ปรากฏว่าโปรแกรมสามารถแก้ปัญหาที่เกิดขึ้นจากการจัดเก็บข้อมูล คำนวณอัตราดอกเบี้ย รายงานการทำสัญญาและเดือนวันหมดอายุของทะเบียนรถได้จริง

#### 5.1 สรุปปัญหาที่เกิดขึ้นในการทดลอง

ในการทดลองนั้นได้พบปัญหา คือ

1. ในการออกใบเสร็จรับเงินนั้น ภาษา PHP มีปัญหาในการจัดการเรื่องการใส่ printer โดยตรงแต่ในส่วนของโปรแกรมที่จัดทำขึ้นนั้นแก้ปัญหาโดยการใช้ browser ช่วยในการ print ใบเสร็จ
2. การสั่งคืนใบเสร็จนั้นมีปัญหาตรงที่ ถ้ามีค่าปรับเกิดขึ้นจะทำให้การคำนวณค่าปรับนั้นผิดพลาดจากความจริง

#### 5.2 ข้อเสนอแนะและแนวทางดำเนินงานต่อ

1. ทำการศึกษาภาษา PHP เพิ่มในเรื่องเกี่ยวกับการใช้ function printer เพื่อสะดวกในการออกใบเสร็จ
2. ทำการวิเคราะห์เรื่องการสั่งคืนใบเสร็จใหม่ว่า ปัญหาที่เกิดขึ้นนั้นเกิดจากอะไร

## เอกสารอ้างอิง

- [1] กิตติ ภัคตีวัฒนะกุล, คัมภีร์ PHP. กรุงเทพฯ : เคทีพี คอมพ์ แอนคอนซัลท์, 2547
- [2] จิรธิ กำไร, Dream Weaver MX 2004. กรุงเทพฯ : เอส.พี.ซี.พรินติ้ง, 2547
- [3] กิตติมา เจริญหิรัญ, การวิเคราะห์และออกแบบระบบ. กรุงเทพฯ : ท็อป, 2546
- [4] โภภาส เอี่ยมศิริวงศ์, การออกแบบและจัดการฐานข้อมูล. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2546

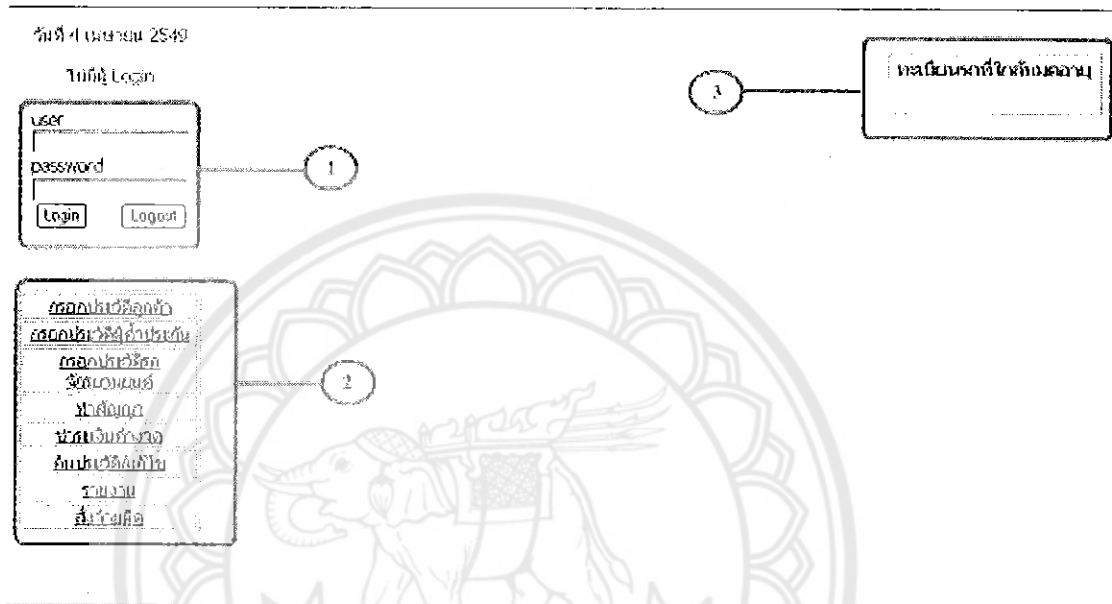


## ภาคผนวก

### การใช้งานโปรแกรม

#### 1. ระบบการเข้าใช้งาน

เมื่อเปิดเข้าใช้ โปรแกรมหน้าแรกที่โปรแกรมแสดงเป็นดังรูปที่ 1



รูปที่ 1 เว็บเพจล็อกอินและเมนูหลัก

จากรูปที่ 1 แบ่งออกเป็น 3 ส่วนคือ

1. ส่วนของล็อกอินเป็นการใส่ user และ password ของผู้ใช้ระบบ

2. ส่วนของเมนูหลักแบ่งเป็น 8 ส่วนคือ

- |                              |                   |
|------------------------------|-------------------|
| - การกรอกประวัติลูกค้า       | -ชำระเงินต่างวด   |
| -การกรอกประวัติผู้ค้าประกัน  | -คืนประวัติ/แก้ไข |
| -การกรอกประวัติรถจักรยานยนต์ | -รายงาน           |
| -การทำสัญญา                  | -ส่งจ่ายผิด       |

3. ส่วนของการแจ้งเตือนทะเบียนที่ทำสัญญาอยู่รถใกล้หมดอายุ





### 3. การกรอกประวัติผู้ค้าประกัน

แสดงการกรอกประวัติผู้ค้าประกันดังแสดงในรูปที่ 3

ประวัติผู้ค้าประกัน

รหัสผู้ค้าประกัน ๑00004

ชื่อ-สกุล ผู้ทำสัญญา ศักดิ์ชัย งามน้อย อายุ 55 ปี

สถานภาพ โสด

ชื่อ-สกุล คู่สมรส อายุ ปี

ที่อยู่ปัจจุบัน 10 หมู่ 7 ต.โคกสว่าง อ.ตาดลี จ.นครสวรรค์

โทรศัพท์ 056328328

สถานที่ใกล้เคียง

บัตรแสดงตนถือบัตร ประชาชน

เลขที่ 3262589745879

ออกให้โดย กรมการปกครองกระทรวงมหาดไทย

วันออกบัตร 31 กันยายน 2545

บัตรหมดอายุ 31 กันยายน 2550

ที่อยู่ใบบัตรประจำตัว 10 หมู่ 7 ต.โคกสว่าง อ.ตาดลี จ.นครสวรรค์

อาชีพ คำขาย

ตำแหน่ง

แผนก

ชื่อสถานที่ทำงาน

ที่ตั้งที่ทำงาน

โทรศัพท์

ผู้ขอสินเชื่อ รายได้วันละ 500 บาท เดือนละ 35000 บาท

คู่สมรส รายได้วันละ บาท เดือนละ บาท

รายได้อื่นๆจาก เลี้ยงเดือนละ บาท

รูปที่ 3 การกรอกประวัติส่วนของผู้ค้าประกัน

จากรูปที่ 3 เป็นหน้าตาการกรอกประวัติผู้ค้าประกัน มีดังนี้

- |                     |                        |                       |
|---------------------|------------------------|-----------------------|
| -รหัสผู้ค้าประกัน   | -บัตรแสดงตนถือบัตร     | -ตำแหน่ง              |
| -ชื่อสกุลผู้ทำสัญญา | -เลขที่                | -แผนก                 |
| -สถานภาพ            | -ออกให้โดย             | -ชื่อสถานที่ทำงาน     |
| -ชื่อสกุลคู่สมรส    | -วันออกบัตร            | -ที่ตั้งที่ทำงาน      |
| -ที่อยู่ปัจจุบัน    | -บัตรหมดอายุ           | -โทรศัพท์             |
| -โทรศัพท์           | -ที่อยู่ใบบัตรประจำตัว | -รายได้เฉลี่ยต่อเดือน |
| -สถานที่ใกล้เคียง   | -อาชีพ                 |                       |

#### 4. การกรอกประวัติรถจักรยานยนต์

แสดงการกรอกประวัติรถจักรยานยนต์ดังแสดงในรูปที่ ก.4

รถจักรยานยนต์

รถจักรยานยนต์ยี่ห้อ  รุ่น  
แบบ  สี

หมายเลขทะเบียน   
หมายเลขเครื่อง

หมายเลขตัวถัง

อายุรถ  ปี

วันที่หมดอายุป้ายวงกลม 1 ▼ ตุลาคม ▼ 2548 ▼

ชื่อผู้ถือกรรมสิทธิ์

บัตรประชาชนผู้ถือกรรมสิทธิ์หมดอายุ 1 ▼ ตุลาคม ▼ 2548 ▼

Submit

รูปที่ 4 กรอกประวัติของรถจักรยานยนต์

จากรูปที่ 4 เป็นหน้าตาการกรอกประวัติของรถจักรยานยนต์ มีดังนี้

- |                                     |                         |
|-------------------------------------|-------------------------|
| -รถจักรยานยนต์ยี่ห้อ                | -หมายเลขตัวถัง          |
| -แบบ                                | -อายุรถ                 |
| -หมายเลขทะเบียน                     | -วันที่หมดอายุป้ายวงกลม |
| -หมายเลขเครื่อง                     | -ชื่อผู้ถือกรรมสิทธิ์   |
| -บัตรประชาชนผู้ถือกรรมสิทธิ์หมดอายุ |                         |

## 5. การทำสัญญา

แสดงการทำสัญญาที่ตั้งแสดงในรูปที่ ก.5

ทำสัญญา

ทำสัญญาวันที่

รหัสสัญญา

รหัสผู้กู้

รหัสผู้ค้ำประกัน

ความสัมพันธ์ผู้กู้กับผู้ค้ำประกัน

หมายเลขตัวถัง

วงเงินกู้  บาท

อัตราดอกเบี้ย  %

อัตราดอกเบี้ยปรับ  %

จำนวนงวด  งวด

เริ่มจ่ายงวดแรกวันที่

Submit

รูปที่ 5 การกรอกการทำสัญญา

จากรูปที่ 5 เป็นหน้าตาการกรอกการทำสัญญา มีดังนี้

- |                                    |                    |
|------------------------------------|--------------------|
| -ทำสัญญาวันที่                     | -วงเงินกู้         |
| -รหัสสัญญา                         | -อัตราดอกเบี้ย     |
| -รหัสผู้กู้                        | -อัตราดอกเบี้ยปรับ |
| -รหัสผู้ค้ำประกัน                  | -จำนวนงวด          |
| -ความสัมพันธ์ผู้กู้กับผู้ค้ำประกัน | -เริ่มจ่ายงวดแรก   |
| -หมายเลขตัวถัง                     |                    |

## 6. การชำระค่างวด

แสดงการชำระค่างวดดังแสดงในรูปที่ 6

ชำระค่างวด

ใส่รหัสสัญญา

Submit

รูปที่ 6 แสดงหน้าที่จะต้องเข้าไปกรอกรหัส เพื่อยุติยอดค่าค้างชำระในแต่ละงวด

## 7. ค่าใช้จ่ายในแต่ละงวด

แสดงค่าใช้จ่ายในแต่ละงวดดังแสดงในรูปที่ 7

ชำระเงินค่างวด

รหัสสัญญา e5555

ชื่อผู้วิ สมปอง แสงคม

งวดที่ค้างชำระ	วันที่ต้องชำระ	เงินชำระต่องวด	จำนวนวันปรับ	ค่าปรับ	รวมชำระ
1	31 ต.ค. 2548	3300.00	0	0.00	3300.00
2	1 ธ.ค. 2548	3300.00	0	0.00	3300.00
3	31 ธ.ค. 2548	3300.00	0	0.00	3300.00
4	31 ม.ค. 2549	3300.00	0	0.00	3300.00
5	3 มี.ค. 2549	3300.00	0	0.00	3300.00

งวดที่ชำระ: 1 ▼

จำนวนเงินที่ชำระ:  บาท

เลขที่ใบเสร็จ:

วันที่ชำระ: 2 ▼ ตุลาคม ▼ 2548 ▼

Back

Submit

รูปที่ 7 แสดงค่าใช้จ่ายในแต่ละงวดวันที่ต้องจ่ายอัตราค่าปรับในแต่ละงวด

## 8. การค้นหาประวัติ

แสดงการเข้าไปค้นหาประวัติดังแสดงในรูปที่ 8

ค้นหา

ค่าที่ต้องการค้น

ค้นตามประวัติ

รูปที่ 8 แสดงการค้นหาประวัติของลูกค้า โดยจะต้องกรอกชื่อของผู้กู้เพื่อค้นหาประวัติ

## 9. ประวัติผู้ค้าประกัน

แสดงการค้นหาประวัติของผู้ค้าประกันดังแสดงในรูปที่ 9

### ประวัติผู้ค้าประกัน

รหัส	ชื่อ-นามสกุล	แก้ไข / ลบ
e00000	พีรณย์ เมฆอับ	แก้ไข ลบ
e00001	ศักดิ์สิทธิ์ ชันสาคร	แก้ไข ลบ
e00002	ก้องเกียรติ กรีฑาเวทย์	แก้ไข ลบ
e00003	อนุชา ส่องแสง	แก้ไข ลบ
e00004	ศักดิ์ชัย งามน้อย	แก้ไข ลบ
e00005	นกน้อย แสงจันทร์	แก้ไข ลบ
e00006	กนก ชัยราช	แก้ไข ลบ
e00007	เมืองแมน พานิช	แก้ไข ลบ

รูปที่ 9 การค้นหาประวัติของผู้ค้าประกัน

### 10. การแก้ไขประวัติ

แสดงการเข้าไปแก้ไขประวัติดังแสดงในรูปที่ 10

รูปที่ 10 ประวัติผู้กู้ที่ต้องการแก้ไข

รูปที่ 10 แสดงประวัติผู้กู้ที่จะเข้ามาแก้ไขจะต้องให้ Admin เป็นผู้แก้ไขจึงต้อง Login โดย Admin เพื่อแก้ไขประวัติ

## 11. การรายงาน

แสดงการรายงานของช่วงเวลาดังแสดงในรูปที่ 11

ตั้งแต่วันที่ 1 ตุลาคม 2548  
ถึงวันที่ 1 ตุลาคม 2548

ชำระเงิน  
แบบรายงาน  สัญญา  
 ค้างชำระ มากกว่า 8 วัน

Submit

รูปที่ 11 แสดงหน้าที่สามารถเลือกช่วงรายงาน

## 12. ช่วงการทำสัญญา

แสดงช่วงการทำสัญญาดังแสดงในรูปที่ 1

รายงานการเปิดสัญญา

ระหว่างวันที่ 9 ต.ค. 2530 ถึง วันที่ 1 ต.ค. 2555

รหัสสัญญา	วันที่ทำสัญญา	เงินกู้	ดอกเบี้ย	เงินผลประโยชน์	ผู้รับเรื่อง	สถานะ
e22222	22 ก.ย. 2548	1200000.00	2%	288000.00	หลงซื่อ	ทำสัญญาอยู่
e11111	22 ก.ย. 2548	12000.00	2%	2880.00		ปิดสัญญา
e7777	22 ก.ย. 2548	22000.00	2%	4400.00	administrator	ทำสัญญาอยู่
e33333	23 ก.ย. 2548	40000.00	3%	6000.00	พีรณย์ เมฆอัม	ทำสัญญาอยู่
e6666	28 ก.ย. 2548	10000.00	2%	800.00	administrator	ทำสัญญาอยู่
e4444	30 ก.ย. 2548	20000.00	2%	2400.00	administrator	ทำสัญญาอยู่
e5555	30 ก.ย. 2548	15000.00	2%	1500.00	administrator	ทำสัญญาอยู่
รวม		1319000.00		305980.00		

Back

รูปที่ 12 การทำสัญญาในช่วงที่เลือก

รูปที่ 12 แสดงการทำสัญญาในช่วงที่เลือกของแต่ละช่วง

## 13 การส่งจ่ายผิด

การส่งจ่ายผิดเป็นการคืนใบเสร็จจำนวนเงินที่เกิดจากการชำระเงินผิดดังแสดงในรูปที่ 13



## แก้ไขการส่งจ่ายเงิน

รหัสสัญญา	<input type="text"/>	<input type="button" value="Submit"/>
เลขที่ใบเสร็จ	<input type="text"/>	

## รูปที่ 13 การส่งจ่ายผิด

รูปที่ 13 เป็นการกรอกรหัสสัญญาและเลขที่ใบเสร็จของการส่งจ่ายผิด เมื่อกรอกข้อมูลเสร็จแล้วจะเป็นการแสดงว่าใบเสร็จที่ส่งจ่ายผิดนั้นเป็นของงวดที่เท่าไร จำนวนเงินเท่าไรและทำการใส่หมายเหตุของการส่งจ่ายผิดดังแสดงฯ ได้ดังรูปที่ 14

ค้ายลดเงิน

รหัสสัญญา	e11111
เลขที่ใบเสร็จ	123456
งวดที่	1
วันที่ชำระ	25 กันยายน 2548
จำนวนเงิน	1000.00
หมายเหตุ	<input type="text"/> *

ตรวจสอบให้แน่ใจก่อน คลิกปุ่ม 'Submit'

รูปที่ 14 รายละเอียดของการส่งจ่ายผิด

รูปที่ 14 เป็นการแสดงรายละเอียดของการส่งจ่ายผิด เมื่อใส่หมายเหตุของการส่งจ่ายผิดแล้วจึงจะสามารถทำรายการต่อไปได้

## ประวัติผู้เขียนโครงการ



ชื่อ นามสกุล นายพีรณย์ เมฆอับ  
 เกิดวันที่ 5 มีนาคม 2526  
 ภูมิลำเนา 84/57 ถนนพระร่วง ซอย 2 ตำบลในเมือง  
 อำเภอเมือง จังหวัดพิษณุโลก 65000

### ประวัติการศึกษา

- จบการศึกษาชั้นประถมศึกษาจาก เทศบาล 2 วัดคูหา

สวรรค อำเภอเมือง จังหวัดพิษณุโลก

- จบการศึกษาชั้นมัธยมศึกษาตอนต้นจาก โรงเรียนพิษณุโลก  
 พิตยาคม อำเภอเมือง จังหวัดพิษณุโลก

- จบการศึกษาประกาศนียบัตรวิชาชีพจากวิทยาลัยเทคนิค

พิษณุโลก อำเภอเมือง จังหวัดพิษณุโลก

- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยนเรศวร

E-Mail : artblue\_p@hotmail.com



ชื่อ นามสกุล นายศักดิ์สิทธิ์ ชันสาคร

เกิดวันที่ 15 เมษายน 2526

ภูมิลำเนา 20 ซอยเรืองยศ ตำบลตาคี อำเภอตาคี  
 จังหวัดนครสวรรค์ 60140

### ประวัติการศึกษา

- จบการศึกษาชั้นประถมศึกษาจาก โรงเรียนลาซาล ไซตริวี  
 นครสวรรค์ อำเภอเมือง จังหวัดนครสวรรค์

- จบการศึกษาชั้นมัธยมศึกษาจาก โรงเรียนตาคีประชา  
 สวรรค อำเภอตาคี จังหวัดนครสวรรค์

- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยนเรศวร

E-Mail : tuck\_zana@hotmail.com