



โปรเจค : ร้านขายลูกชิ้นผ่านแอปพลิเคชันออนไลน์

จัดทำโดย

นางสาวกัญญาณัฐ ภูมิวัตร 620710291

นางสาวชุติมา เหมือนประสาธ 620710372

นางสาวรวิพร สุภาพันธุ์ชัย 650710714

เสนอ

อ.ดร.อรรรรณ เชาวลิต

โปรเจคนี้เป็นส่วนหนึ่งของการศึกษาวิชาการระบบฐานข้อมูล (Database Systems)

สาขาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์

มหาวิทยาลัยศิลปากร

ปีการศึกษา 2566

กิตติกรรมประกาศ

การทำโปรเจกต์นี้สำเร็จด้วยดี ผู้จัดทำขอกราบขอบพระคุณ อ.ดร.อรรธรณ เซาวลิต ซึ่งเป็นอาจารย์ผู้ควบคุมการทำโปรเจกต์ ที่กรุณาให้แนวคิดและคำแนะนำในการดำเนินงานตลอดจนการแก้ไขปัญหาต่าง ๆ อันเป็นประโยชน์ต่อโปรเจกต์นี้

ประโยชน์อันใดที่เกิดจากโปรเจกต์นี้ย่อมเป็นผลมาจากความกรุณาของท่านดังกล่าวข้างต้น ผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างยิ่งจึงใคร่ขอขอบพระคุณอย่างสูงไว้ ณ โอกาสนี้

คณะผู้จัดทำ

สารบัญ

ที่มาของปัญหา.....	1
Relation	2
Normal.....	3
SQL	5
Crow’s Foot ERD.....	7
หน้าจอ.....	8
1. ระบบสมาชิก	8
2. ระบบคลังสินค้า.....	14
3. ระบบรับคำสั่งซื้อและใบใบเสร็จ	20
บรรณานุกรม	25

ร้านขายลูกชิ้นผ่านแอปพลิเคชันออนไลน์

ที่มาของปัญหา

เนื่องจากร้านลูกชิ้นหรรษาเป็นร้านค้าที่ขายลูกชิ้นหลากหลายชนิดและหลายราคา มีหน้าร้านให้ลูกค้ามาเลือกซื้อลูกชิ้น ซึ่งทางร้านมีลูกค้าเข้ามาตลอด เมื่อลูกค้าเข้ามาพร้อมกันหลายคนเวลาสั่งซื้อก็จะใช้การเขียนใส่กระดาษ หรือพูดปากเปล่ากับพนักงาน เหตุการณ์นี้จึงก่อให้เกิดความผิดพลาดในการรับคำสั่งซื้อ เช่น จำนวนคำสั่งซื้อไม่ครบ

ดังนั้นจึงได้นำระบบการจัดการฐานข้อมูลเข้ามาช่วยในการเก็บข้อมูล เช่น การเรียกดูจำนวนคำสั่งซื้อ ราคารวมแต่ละคำสั่งซื้อ เป็นต้น เพื่อลดความเสี่ยงต่อความผิดพลาดในการดำเนินงาน ทำให้ร้านค้ามีคุณภาพ และความน่าเชื่อถือมากขึ้น

การพัฒนาฐานข้อมูลของระบบร้านลูกชิ้นหรรษา ได้เริ่มจากการไปสำรวจความต้องการของผู้ใช้งานระบบ คือ พนักงาน และเจ้าของร้านค้า จากนั้นทำการรวบรวมข้อมูลเพื่อไปทำการวิเคราะห์ศึกษาการทำงานดังกล่าว

1. รายการคำสั่งซื้อลูกชิ้น : รับคำสั่งซื้อโดยการจดใส่กระดาษ และพูดปากเปล่า
2. จำนวนลูกชิ้น : การตรวจสอบลูกชิ้นในร้านที่มีอยู่ เพียงพอต่อการสั่งซื้อหรือไม่
3. การเตรียมตามคำสั่งซื้อ : การจัดเตรียมให้พร้อมส่ง
4. จัดส่งคำสั่งซื้อ
5. ตรวจสอบการคงเหลือของลูกชิ้น : หลังทำคำสั่งซื้อเสร็จลูกชิ้นจะคงเหลือเท่าใด

Relation

คำสั่งซื้อ (รหัสบิลใบเสร็จ, รหัสสินค้า, ชื่อสินค้า, จำนวนสินค้า, ราคารวม, รับผิดชอบ, รหัสผัก, ใช้น้ำจิ้ม, รหัสน้ำจิ้ม)

Bill_ID	Prod_ID	Prod_Name	Prod_quan	Total	Vegetble_get	Vegetable_id	Sauce_get	Sauce_id
---------	---------	-----------	-----------	-------	--------------	--------------	-----------	----------

ใบเสร็จ (รหัสบิลใบเสร็จ, ชื่อสินค้า, ราคารวม, แต้มที่ได้รับ)

Bill_ID	Prod_Name	Total	Point
---------	-----------	-------	-------

ระบบคลังสินค้า (รหัสสินค้า, ชื่อสินค้า, วันหมดอายุของสินค้า, ราคาขาย, จำนวนสินค้า)

Prod_ID	Prod_Name	Prod_exp	Prod_price	Prod_quan
---------	-----------	----------	------------	-----------

สินค้า (รหัสสินค้า, ชื่อสินค้า, ราคาขาย)

Prod_ID	Prod_Name	Prod_price
---------	-----------	------------

ผัก (รหัสผัก, ชื่อผัก)

Vegetable_id	Vegetable_name
--------------	----------------

น้ำจิ้ม (รหัสน้ำจิ้ม, ชื่อน้ำจิ้ม)

Sauce_id	Sauce_name
----------	------------

รับผิดชอบ (รหัสผัก, ความต้องการ)

Vegetable_id	Vegetable_get
--------------	---------------

ใช้น้ำจิ้ม (รหัสน้ำจิ้ม, ความต้องการ)

Sauce_id	Sauce_get
----------	-----------

Normalization

คำสั่งซื้อ

1NF : รหัสบิลใบเสร็จ, รหัสสินค้า, ชื่อสินค้า, จำนวนสินค้า, ราคารวม, รับפק, รหัสפק, รับน้ำจิ้ม, รหัสน้ำจิ้ม

2NF : รหัสบิลใบเสร็จ, รหัสสินค้า, จำนวนสินค้า, ราคารวม, รับפק, รหัสפק, รับน้ำจิ้ม, รหัสน้ำจิ้ม

3NF : เป็นแล้ว

ใบเสร็จ รหัสบิลใบเสร็จ, ชื่อสินค้า, ราคารวม, แต้มที่ได้รับ

1NF : รหัสบิลใบเสร็จ, ชื่อสินค้า, ราคารวม, แต้มที่ได้รับ

2NF : เป็นแล้ว

3NF : เป็นแล้ว

ระบบคลังสินค้า

1NF : รหัสสินค้า, ชื่อสินค้า, วันหมดอายุของสินค้า, ราคาขาย, จำนวนสินค้า

2NF : เป็นแล้ว

3NF : เป็นแล้ว

สินค้า

1NF : รหัสสินค้า, ชื่อสินค้า, ราคาขาย

2NF : เป็นแล้ว

3NF : เป็นแล้ว

פק

1NF : รหัสפק, ชื่อפק

2NF : เป็นแล้ว

3NF : เป็นแล้ว

น้ำจิ้ม

1NF : รหัสน้ำจิ้ม, ชื่อน้ำจิ้ม

2NF : เป็นแล้ว

3NF : เป็นแล้ว

รับผัก(รหัสผัก, ความต้องการ)

1NF : รหัสผัก, ความต้องการ

2NF : เป็นแล้ว

3NF : เป็นแล้ว

รับน้ำจิ้ม

1NF : รหัสน้ำจิ้ม, ความต้องการ

2NF : เป็นแล้ว

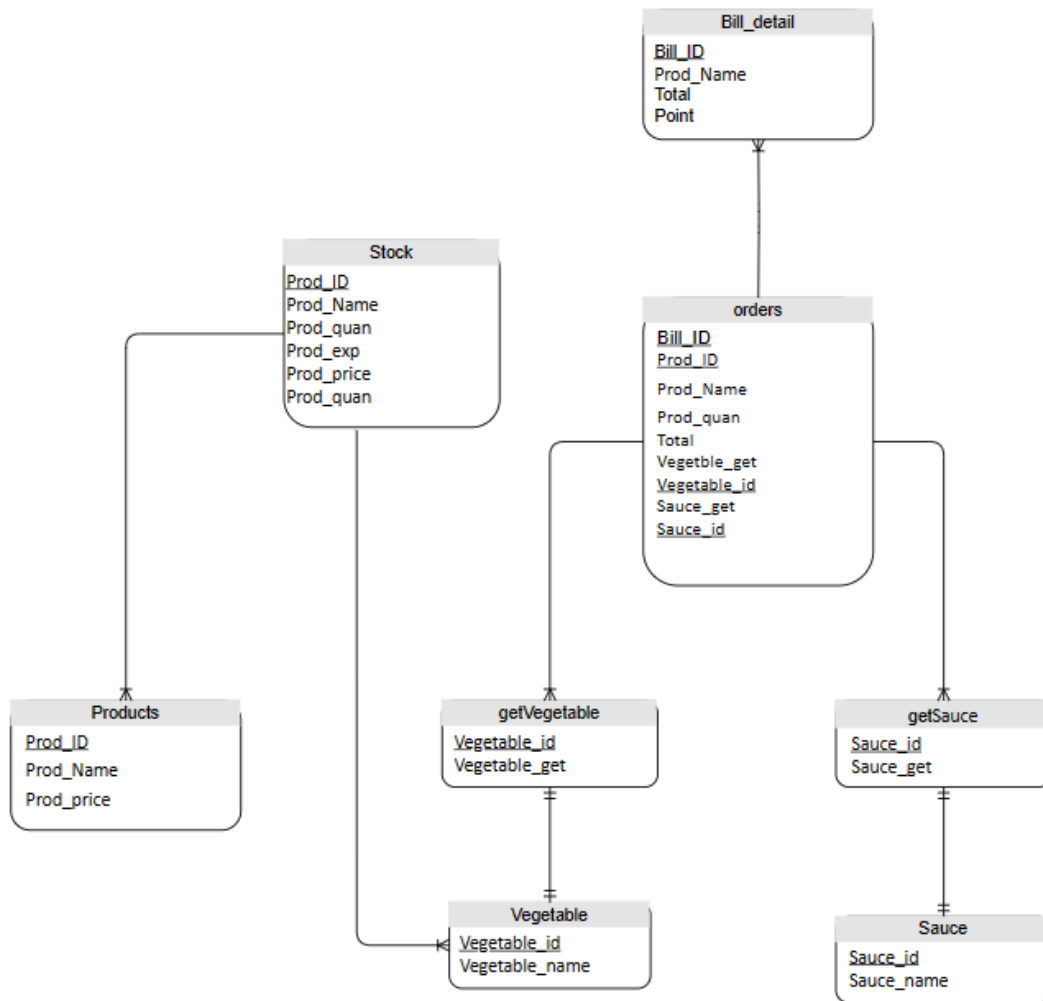
3NF : เป็นแล้ว

SQL

<p>ตาราง order</p> <pre> CREATE TABLE orders (Order_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Prod_ID INT NOT NULL, Prod_Name VARCHAR(100) NOT NULL, Prod_quan INT NOT NULL, Total DECIMAL(10,2) NOT NULL, Vegetable_get VARCHAR(100), Vegetable_id INT, Sauce_get VARCHAR(100), Sauce_id INT); </pre>	<p>ตาราง Stock</p> <pre> CREATE TABLE Stock (Prod_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Prod_Name VARCHAR(100) NOT NULL, Prod_quan INT NOT NULL, Prod_exp DATE NOT NULL, Prod_price DECIMAL(10,2) NOT NULL); </pre>
<p>ตาราง Bill_detail</p> <pre> CREATE TABLE Bill_detail (Bill_ID INT NOT NULL, Prod_Name VARCHAR(100) NOT NULL, Total DECIMAL(10,2) NOT NULL, Point DECIMAL(10,2) NOT NULL, PRIMARY KEY (Bill_ID, Prod_Name)); </pre>	<p>ตาราง Products</p> <pre> CREATE TABLE Products (Prod_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Prod_Name VARCHAR(100) NOT NULL, Prod_price DECIMAL(10,2) NOT NULL); </pre>

<p>ตาราง getVegetable</p> <pre>CREATE TABLE getVegetable (Vegetable_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Vegetable_get VARCHAR(100) NOT NULL);</pre>	<p>ตาราง Vegetable</p> <pre>CREATE TABLE Vegetable (Vegetable_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Vegetable_name VARCHAR(100) NOT NULL);</pre>
<p>ตาราง getSauce</p> <pre>CREATE TABLE getSauce (Sauce_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Sauce_get VARCHAR(100) NOT NULL);</pre>	<p>ตาราง Sauce</p> <pre>CREATE TABLE Sauce (Sauce_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Sauce_name VARCHAR(100) NOT NULL);</pre>

Crow's Foot ERD



หน้าจอ

ระบบสมาชิก

Back End ใช้ Fast API เชื่อมต่อกับ PHP MyAdmin ใช้งานผ่าน Docker

โครงสร้างฐานข้อมูล ระบบสมาชิก product

CREATE TABLE 'product'(

'id' int not null **PRIMARY KEY**,

'product_name' varchar(30) not null,

'description' varchar(50) not null,

'price' float not null

);

ตัวอย่างข้อมูล

- Post Product – สร้างข้อมูล

```
{
  "id": 1,
  "product_name": "กัญญาณัฐ ภูมิวัตร",
  "description": "kanyanat@gmail.com",
  "price": 550.0,
}
```

คำสั่ง SQL

```
INSERT INTO `product` (`id`, `product_name`, `description`, `price`)
```

```
VALUES ('1', 'กัญญาณัฐ ภูมิวัตร', kanyanat@gmail.com, '550.0');
```

- Get Product – เรียกดูข้อมูล

GET /get_product/{id} Get Product By Id Api

Parameters

Name	Description
id * required integer (path)	1

คำสั่ง SQL

SELECT * FROM `product` WHERE id ='1';

- Put Product – อัปเดตแก้ไขข้อมูล

PUT /update_product/{id} Update Product Api

Parameters

Name	Description
id * required integer (path)	1

Request body required

```
{
  "id": 1,
  "product_name": "คัพเค้กกล้วย ภูเก็ต",
  "description": "kanyanat@gmail.com",
  "price": 550.0
}
```

คำสั่ง SQL

UPDATE `product` SET `price` = '550.0' WHERE `product`.`id` = '1'

- Delete Product - ลบข้อมูล

DELETE /delete_product/{id} Delete Product Api

Parameters

Name	Description
id * required integer (path)	1

Servers

คำสั่ง SQL

DELETE FROM `product` WHERE id = '1';

Front End ใช้ Vue.js(วิว) สามารถสร้างข้อมูล แก้ไขข้อมูล ลบข้อมูลได้ในหน้าเดียวกัน โดยข้อมูลจะเชื่อมต่อกับ Back End และใช้งานผ่าน Docker

1.หน้า Vue ระบบสมาชิก

2.เพิ่มข้อมูล

เมื่อเพิ่มข้อมูลแล้วจะขึ้น alert บอกว่าเพิ่มข้อมูลสำเร็จ “Add successfully” หลังจากกด “OK” ข้อมูลจะเพิ่มลงไป

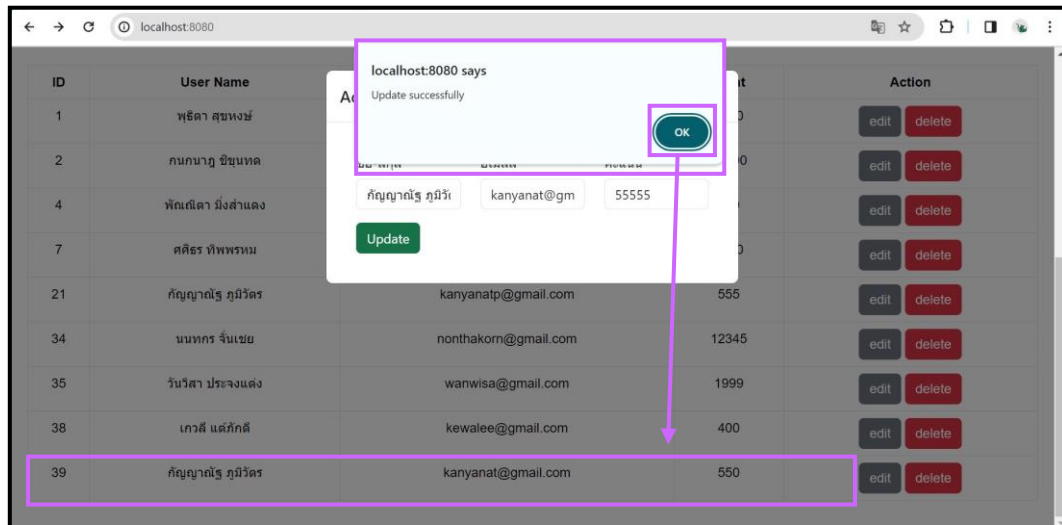
ID	User Name	E-mail	Point	Action
1	พริดา สุขหงษ์	phuthita@gmail.com	1200	<button>edit</button> <button>delete</button>
2	กนกนาฏ ชื่นนท	kanoknart@gmail.com	56900	<button>edit</button> <button>delete</button>
4	พัฒนไดา มิ่งสำแดง	pannitam@gmail.com	999	<button>edit</button> <button>delete</button>
7	ศศิธร ทิพรหม	sasithornt@gmail.com	2900	<button>edit</button> <button>delete</button>
21	กัญญากร ไกรวัตร	kanyanatp@gmail.com	555	<button>edit</button> <button>delete</button>
34	นนทกร จันทชัย	nonthakorn@gmail.com	12345	<button>edit</button> <button>delete</button>
35	วันวิสา ประจแสง	wanwisa@gmail.com	1999	<button>edit</button> <button>delete</button>
38	เกวลี แด่ภักดี	kewalee@gmail.com	400	<button>edit</button> <button>delete</button>

ข้อมูลจะเพิ่มลงมาในหน้านี้

3.แก้ไขข้อมูล

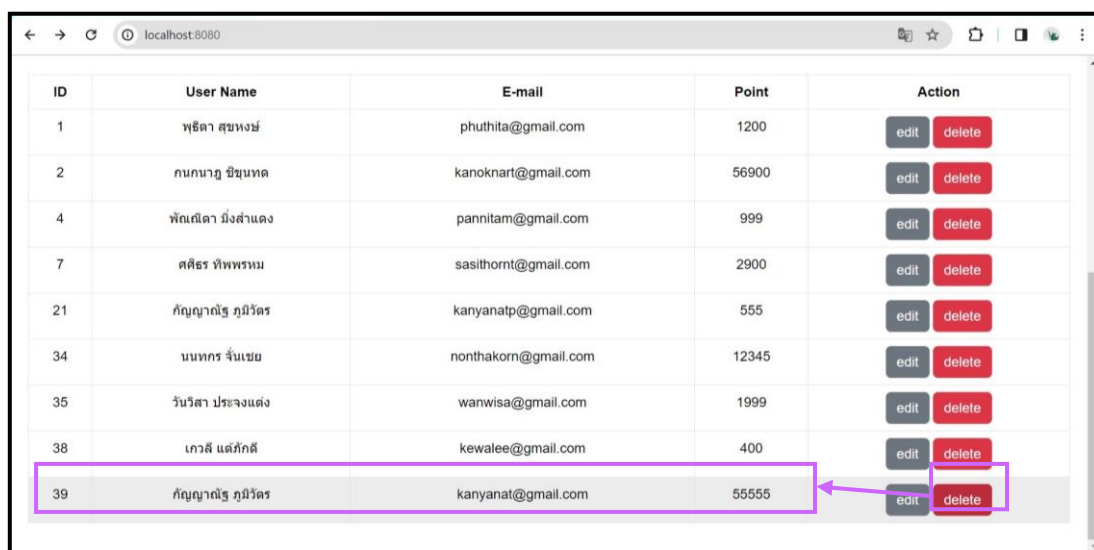
The screenshot shows a web application interface. In the background, there is a table with columns: ID, User Name, E-mail, Point, and Action. The table contains several rows of user data. Overlaid on the table is a modal form titled 'Add'. The modal form has three input fields: 'ชื่อ-สกุล' (Name-Surname) with the value 'กัญญากร ไกรวัตร', 'อีเมลล์' (Email) with the value 'kanyanat@gm', and 'คะแนน' (Score) with the value '550'. Below these fields is a green 'Update' button. A red box highlights the 'Add' button in the modal, and a red arrow points from the 'edit' button in the table's 'Action' column to this 'Add' button.

“edit” เพื่อแก้ไขข้อมูลใหม่ไปแทนที่ข้อมูลเก่าที่ต้องการเปลี่ยนแปลง

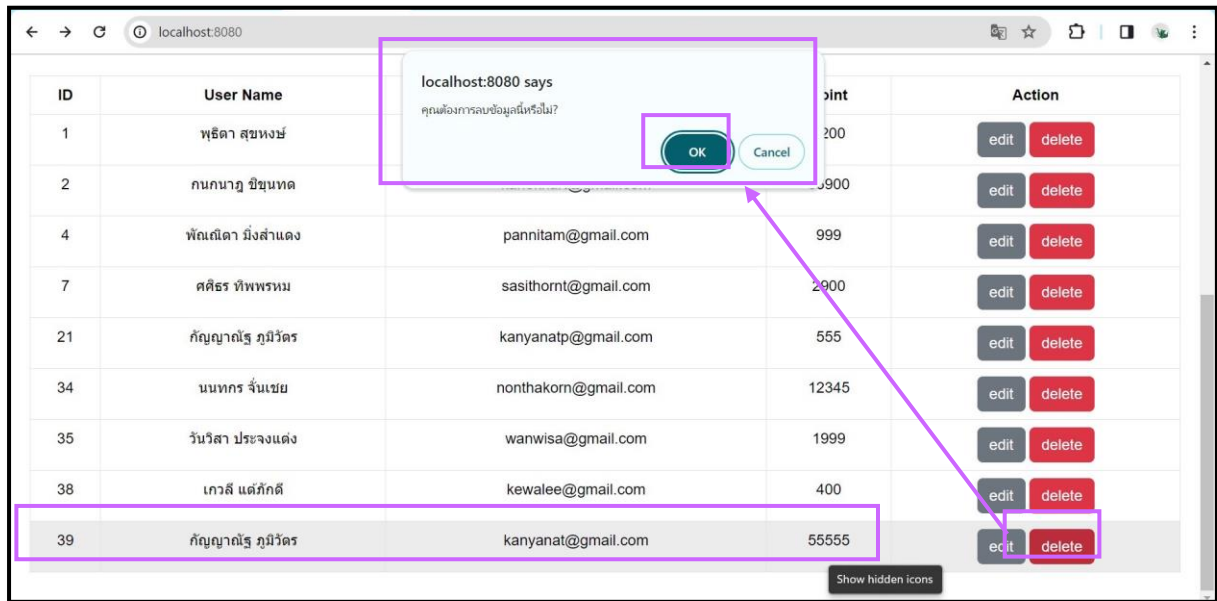


เมื่อกด “Update” เพื่อแก้ไขข้อมูล จะมี alert แจ้งว่าสำเร็จด้วยข้อความ “Update successfully” ข้อมูลทุกอย่างที่แก้ไข จะเปลี่ยนแปลงเมื่อกดปุ่ม “OK”

4.ลบข้อมูล

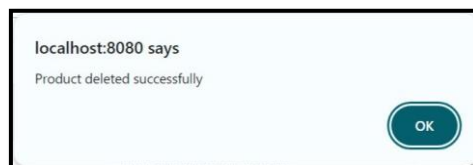


กดเลือกข้อมูลที่ต้องการลบที่ปุ่ม “delete ” จากนั้น



หลังจากกดปุ่ม “delete” แล้วจะขึ้น alert พร้อมข้อความถามว่า “คุณต้องการลบข้อมูลนี้หรือไม่?”

หากต้องการลบให้กดปุ่ม “OK” เพื่อทำการลบข้อมูลที่ไม่ต้องการ แต่หากไม่ต้องการลบกดปุ่ม “cancel” ด้านข้าง



แล้ว alert ถัดมาจะแจ้งข้อความ “Product delete successfully” เพื่อบอกว่าลบข้อความที่กดยืนยันเรียบร้อยแล้ว จากนั้นกดปุ่ม “OK” ข้อความนั้นจะหายไปจะเหลือข้อมูลเพียงภาพถัดไป

ID	User Name	E-mail	Point	Action
1	พฐิตา สุขหงษ์	phuthita@gmail.com	1200	edit delete
2	กนกนาฏ ชื่นทนต์	kanoknart@gmail.com	56900	edit delete
4	พัดเดิดา มิ่งสำแดง	pannitam@gmail.com	999	edit delete
7	ศศิธร ทิพพรหม	sasithornt@gmail.com	2900	edit delete
21	กัญญาภาณุ ภูมิวิตร	kanyanatp@gmail.com	555	edit delete
34	นนทกร จันทชัย	nonthakorn@gmail.com	12345	edit delete
35	วันวิสา ประจงแต่ง	wanwisa@gmail.com	1999	edit delete
38	เกวลี แด้ภักดี	kewalee@gmail.com	400	edit delete

ระบบคลังสินค้า

Back End ใช้ Fast API เชื่อมต่อกับ PHP MyAdmin ใช้งานผ่าน Docker

โครงสร้างฐานข้อมูล ระบบคลังสินค้า product_system

```
CREATE TABLE `product_system`(  
  `prod_id` varchar(4) not null,  
  `prod_name` varchar(30) not null,  
  `expire_date` date,  
  `price` float not null,  
  `prod_quan` int,  
  PRIMARY KEY (`prod_id`)  
);
```

ตัวอย่างข้อมูล

- Post Product – สร้างข้อมูล

```
{  
  "prod_id": "P004",  
  "prod_name": "ปุ๋ยัด",  
  "expire_date": "2024-04-12",  
  "price": 5.0,  
  "prod_quan": 50  
}
```

คำสั่ง SQL

```
INSERT INTO `product_system` (`prod_id`, `prod_name`, `expire_date`, `price`,  
`prod_quan`) VALUES ('P004', 'ปุ๋ยัด', '2024-04-12', '5.0', '50');
```

- Get Product – เรียกดูข้อมูล

http://localhost:8001/get_product/P004

GET /get_product/{prod_id} Get Product By Id Api	
Parameters	
Name	Description
prod_id * required string (path)	P004

รูปที่ 1 Get Product

คำสั่ง SQL

SELECT * FROM `product_system` WHERE prod_id ='P004';

- Put Product – อัปเดตแก้ไขข้อมูล

http://localhost:8001/update_product/P004

PUT /update_product/{prod_id} Update Product Api	
Parameters	
Name	Description
prod_id * required string (path)	P004
Request body * required	
<pre>{ "prod_id": "P004", "prod_name": "ปลาด", "expire_date": "2024-04-12", "price": 5, "prod_quan": 20 }</pre>	

รูปที่ 2 Put Product

คำสั่ง SQL

UPDATE `product_system` SET `prod_quan` = '20' WHERE `product_system`.`prod_id` = 'P004'

- Delete Product - ลบข้อมูล

http://localhost:8001/delete_product/P004

DELETE /delete_product/{prod_id} Delete Product Api	
Parameters	
Name	Description
prod_id * required string (path)	<input type="text" value="P004"/>

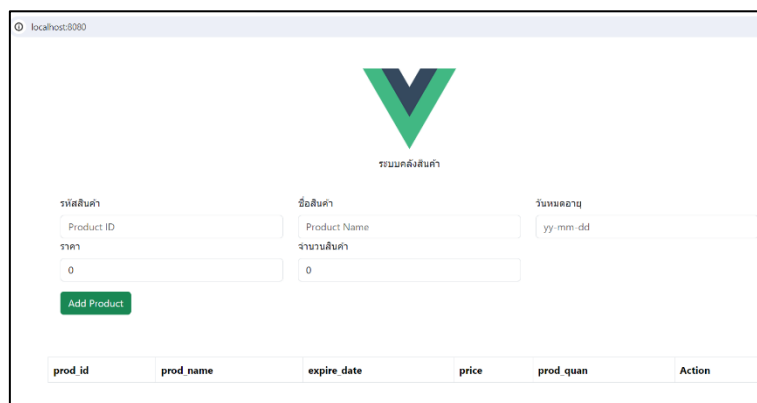
รูปที่ 3 Delete Product

คำสั่ง SQL

```
DELETE FROM `product_system` WHERE prod_id = 'P004';
```

Front End ใช้ Vue.js(วิว) สามารถสร้างข้อมูล แก้ไขข้อมูล ลบข้อมูลได้ในหน้าเดียวกัน โดยข้อมูลจะเชื่อมต่อกับ Back End และใช้งานผ่าน Docker

1. หน้า Vue ระบบคลังสินค้า



localhost:8080

ระบบคลังสินค้า

รหัสสินค้า: Product ID
ราคา: 0

ชื่อสินค้า: Product Name
จำนวนสินค้า: 0

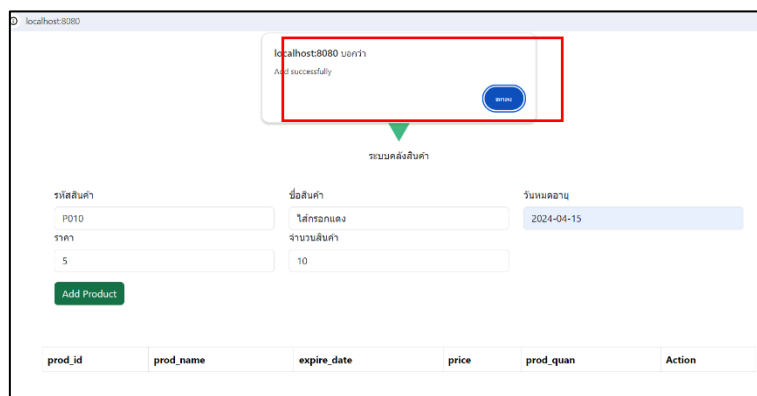
วันหมดอายุ: yy-mm-dd

Add Product

prod_id	prod_name	expire_date	price	prod_quan	Action
---------	-----------	-------------	-------	-----------	--------

รูปที่ 4 ระบบคลังสินค้า

2. เพิ่มข้อมูล



localhost:8080 บอกว่า
Add successfully

ระบบคลังสินค้า

รหัสสินค้า: P010
ราคา: 5

ชื่อสินค้า: ไก่ทอดแดง
จำนวนสินค้า: 10

วันหมดอายุ: 2024-04-15

Add Product

prod_id	prod_name	expire_date	price	prod_quan	Action
---------	-----------	-------------	-------	-----------	--------

รูปที่ 5 แจ้งเตือนการเพิ่มข้อมูล

เมื่อกรอกข้อมูลเสร็จ กด **Add Product** แล้ว จะมีการแจ้งเตือนบนเว็บว่า **Add successfully** จากนั้นข้อมูลก็เข้าสู่ระบบ ดังรูปด้านล่าง

ระบบคลังสินค้า

รหัสสินค้า:

ชื่อสินค้า:

วันหมดอายุ:

ราคา:

จำนวนสินค้า:

prod_id	prod_name	expire_date	price	prod_quan	Action
P010	ไส้กรอกแดง	2024-04-15	5	10	<input type="button" value="edit"/> <input type="button" value="delete"/>

รูปที่ 6 ข้อมูลถูกเพิ่มในระบบ

3. แก้ไขข้อมูล

Add

รหัสสินค้า:

ชื่อสินค้า:

วันหมดอายุ:

ราคา:

จำนวนสินค้า:

prod_id	prod_name	expire_date	price	prod_quan	Action
P001	ลูกชิ้นปลา	2024-04-20	10	100	<input type="button" value="edit"/> <input type="button" value="delete"/>
P010	ไส้กรอกแดง	2024-04-15	5	10	<input type="button" value="edit"/> <input type="button" value="delete"/>

รูปที่ 7 การแก้ไขข้อมูล

เมื่อกดปุ่ม **edit** จะมีหน้าต่างแก้ไขข้อมูล จากตัวอย่างมีการอัปเดตข้อมูลที่ ราคา(price) และจำนวนสินค้า(prod_quan) จะได้ผลลัพธ์ดังนี้

P010	ไส้กรอกแดง	2024-04-15	10	50
------	------------	------------	----	----

รูปที่ 8 ข้อมูลถูกแก้ไขแล้ว

4. ลบข้อมูล

localhost:8080 บอกรว่า
คุณต้องการลบข้อมูลนี้หรือไม่?

ลบออก ยกเลิก

ระบบคลังสินค้า

รหัสสินค้า:
ราคา:

ชื่อสินค้า:
จำนวนสินค้า:

วันหมดอายุ:

Add Product

prod_id	prod_name	expire_date	price	prod_quan	Action
P001	ลูกชิ้นปลา	2024-04-20	10	100	edit delete
P004	โน้ตบุ๊ก	2024-04-26	12	74	edit delete
P006	บุ๊ค	2024-04-12	5	55	edit delete

รูปที่ 9 การแจ้งเตือนลบข้อมูล

localhost:8080 บอกรว่า
Product deleted successfully

ลบออก

ระบบคลังสินค้า

รหัสสินค้า:
ราคา:

ชื่อสินค้า:
จำนวนสินค้า:

วันหมดอายุ:

Add Product

prod_id	prod_name	expire_date	price	prod_quan	Action
P001	ลูกชิ้นปลา	2024-04-20	10	100	edit delete
P004	โน้ตบุ๊ก	2024-04-26	12	74	edit delete
P006	บุ๊ค	2024-04-12	5	55	edit delete

รูปที่ 10 แสดงข้อความว่าข้อมูลถูกลบแล้ว

เมื่อกดปุ่ม **delete** จะมีข้อความบนเว็บแสดงว่า *คุณต้องการลบข้อมูลนี้ใช่หรือไม่* เมื่อกด **ตกลง** ก็จะมีข้อความแสดงอีกว่า *Product delete successfully*

ผลลัพธ์ที่ได้คือ prod_id = P006 หายไป

prod_id	prod_name	expire_date	price	prod_quan	Action
P001	ลูกชิ้นปลา	2024-04-20	10	100	edit delete
P004	โน้ตบุ๊ก	2024-04-26	12	74	edit delete

รูปที่ 11 ข้อมูลที่เหลือในระบบ

ระบบรับคำสั่งซื้อและบิลใบเสร็จ

Back End ใช้ Rest API เชื่อมต่อกับ MySQL ด้วย Gin Framework

ตาราง bill_detail

- สร้างตาราง bill_detail

```
CREATE TABLE bill_detail (
```

```
Bill_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```
Bill_phonenumber_member VARCHAR(20) NOT NULL,
```

```
Bill_total DECIMAL(10,2) NOT NULL,
```

```
Bill_point DECIMAL(10,2) NOT NULL );
```

```
describe bill_detail ;
```

Field	Type	Null	Key	Default	Extra
Bill_ID	int	NO	PRI	HULL	auto_increment
Bill_phonenumber_member	varchar(20)	NO		HULL	
Bill_total	decimal(10,2)	NO		HULL	
Bill_point	decimal(10,2)	NO		HULL	

การสร้างบิลใบเสร็จ – Post

ตัวอย่างข้อมูล

```
{
  "bill_phonenumber_member": "0983787681",
  "bill_total": 50,
  "bill_point": 1
}
```

คำสั่ง SQL

เพิ่มข้อมูลลงในตาราง

```
INSERT INTO bill_detail (Bill_phonenumber_member, Bill_total, Bill_point)
```

```
VALUES ('0123456789', 100.50, 20.00),
```

```
('0983787681', 50.00, 1.00) ;
```

การแสดงผลใบเสร็จ – Get

BillID	CustomerPhoneNumber	Total	Point
1	0897453625	5	7
2	0856963221	20	2
3	0983787681	100	2
4	0860552848	200	4

คำสั่ง SQL

```
select
Bill_ID,
Bill_phonenumber_member,
Bill_total,
Bill_point
from bill_detail ;
```

การลบใบเสร็จ – Delete

BillID	CustomerPhoneNumber	Total	Point	Action
1	0897453625	5	7	Delete
2	0856963221	20	2	Delete
3	0983787681	100	2	Delete
4	0860552848	200	4	Delete

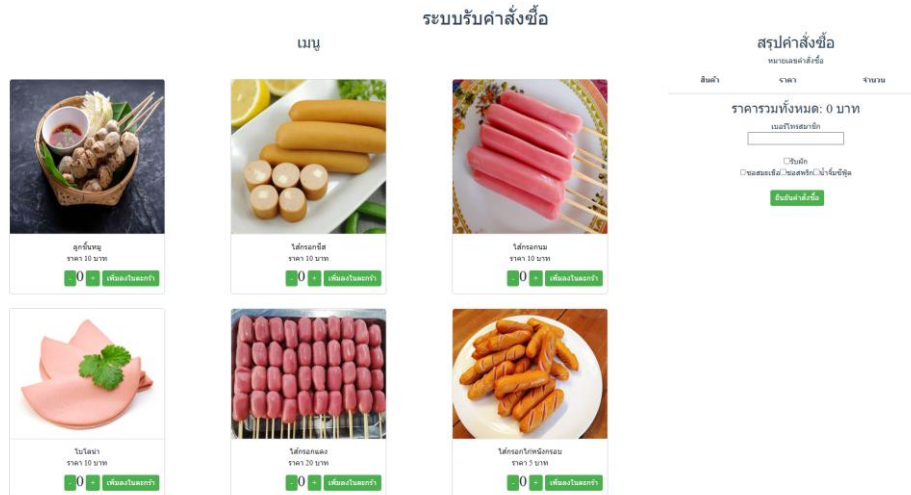
คำสั่ง SQL

```
DELETE FROM bill_detail
```

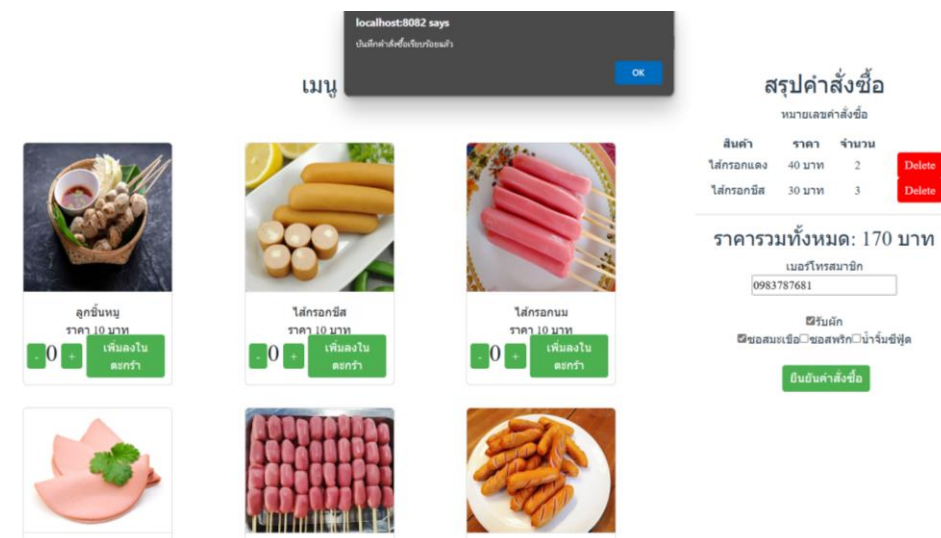
```
WHERE Bill_ID = ?
```


Front End ใช้ Vue.js ซึ่งสามารถแสดงข้อมูล สร้างข้อมูล แก้ไขข้อมูล ลบข้อมูลได้ โดยจะเชื่อมต่อกับ Back End เพื่อนำข้อมูลจากฐานข้อมูลมาใช้

1. หน้ารับคำสั่งซื้อ



2. หน้าการบันทึกการสั่งซื้อและสร้างบิลใบเสร็จ



เมื่อกด “ยืนยันคำสั่งซื้อ” ข้อมูลจะถูกนำไปสร้างบิลใบเสร็จ

3. หน้าบิลใบเสร็จ

CustomerPhoneNumber

Total

Point

Enter PhoneNumber

0

0

Add Bill

Search Bill by BillID

Enter BillID

Reset

Search

BillID	CustomerPhoneNumber	Total	Point	Action
1	0897453625	5	7	Delete
2	0856963221	20	2	Delete
3	0983787681	100	2	Delete
4	0860552848	200	4	Delete

3.1 การเพิ่มข้อมูลบิลใบเสร็จ

CustomerPhoneNumber

Total

Point

Enter PhoneNumber

Enter Total

Enter Point

3.2 การค้นหาบิลใบเสร็จจาก Bill_ID

Search Bill by BillID

Enter BillID

Reset

Search

3.3 การแสดงข้อมูลบิลใบเสร็จทั้งหมด

BillID	CustomerPhoneNumber	Total	Point
1	0897453625	5	7
2	0856963221	20	2
3	0983787681	100	2
4	0860552848	200	4

3.4 การลบข้อมูลจากในฐานข้อมูล

BillID	CustomerPhoneNumber	Total	Point	Action
1	0897453625	5	7	Delete
2	0856963221	20	2	Delete
3	0983787681	100	2	Delete
4	0860552848	200	4	Delete

บรรณานุกรม

- [1] POS . [ออนไลน์]. Available: <https://www.businessplus.co.th/Product/business-plus-pos>