

โปรเจค : ร้านขายลูกชิ้นผ่านแอปพลิเคชันออนไลน์

จัดทำโดย

นางสาวกัญญาณัฐ ภูมิวัตร 620710291

นางสาวชุติมา เหมือนประสาท 620710372

นางสาวรวีพร สุภาพันธุ์ชัย 650710714

เสนอ

อ.ดร.อรวรรณ เชาวลิต

โปรเจคนี้เป็นส่วนหนึ่งของการศึกษาวิชาระบบฐานข้อมูล ( Database Systems )

สาขาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์

มหาวิทยาลัยศิลปากร

ปีการศึกษา   2566

**กิตติกรรมประกาศ**

การทำโปรเจคนี้สำเร็จด้วยดี ผู้จัดทำขอกราบขอบพระคุณ อ.ดร.อรวรรณ เชาวลิต ซึ่งเป็นอาจารย์ผู้ควบคุมการทำโปรเจค ที่กรุณาให้แนวคิดและคำแนะนำในการดำเนินงานตลอดจนการแก้ไขปัญหาต่าง ๆ อันเป็นประโยชน์ต่อโปรเจคนี้

ประโยชน์อันใดที่เกิดจากโปรเจคนี้ย่อมเป็นผลมาจากความกรุณาของท่านดังกล่าวข้างต้น ผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างยิ่งจึงใคร่ขอขอบพระคุณอย่างสูงไว้ ณ โอกาสนี้

คณะผู้จัดทำ

**สารบัญ**

[ที่มาของปัญหา 1](#_Toc163151885)

[Relation 2](#_Toc163151888)

[Normal 3](#_Toc163151889)

SQL [5](#_Toc163151887)

Crow’s Foot ERD [7](#_Toc163151887)

หน้าจอ 8

1. ระบบสมาชิก [8](#_Toc163151887)

2. ระบบคลังสินค้า [14](#_Toc163151887)

3. ระบบรับคำสั่งซื้อและบิลใบเสร็จ 20

[บรรณานุกรม 25](#_Toc163151890)

**ร้านขายลูกชิ้นผ่านแอปพลิเคชันออนไลน์**

**ที่มาของปัญหา**

เนื่องจากร้านลูกชิ้นหรรษาเป็นร้านค้าที่ขายลูกชิ้นหลากหลายชนิดและหลายราคา มีหน้าร้านให้ลูกค้ามาเลือกซื้อลูกชิ้น ซึ่งทางร้านมีลูกค้าเข้ามาตลอด เมื่อลูกค้าเข้ามาพร้อมกันหลายคนเวลาสั่งซื้อก็จะใช้การเขียนใส่กระดาษ หรือพูดปากเปล่ากับพนักงาน เหตุการณ์นี้จึงก่อให้เกิดความผิดพลาดในการรับคําสั่งซื้อ เช่น จํานวนคําสั่งซื้อไม่ครบ

ดังนั้นจึงได้นําระบบการจัดการฐานข้อมูลเข้ามาช่วยในการเก็บข้อมูล เช่น การเรียกดูจํานวน

คําสั่งซื้อ ราคารวมแต่ละคําสั่งซื้อ เป็นต้น เพื่อลดความเสี่ยงต่อความผิดพลาดในการดําเนินงาน ทําให้ร้านค้ามีคุณภาพ และความน่าเชื่อถือมากขึ้น

การพัฒนาฐานข้อมูลของระบบร้านลูกชิ้นหรรษา ได้เริ่มจากการไปสํารวจความต้องการของผู้ใช้งานระบบ คือ พนักงาน และเจ้าของร้านค้า จากนั้นทําการรวบรวมข้อมูลเพื่อไปทําการวิเคราะห์ศึกษาการทํางานดังกล่าว

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

## **Relation**

คำสั่งซื้อ (รหัสบิลใบเสร็จ, รหัสสินค้า ,ชื่อสินค้า, จำนวนสินค้า, ราคารวม ,รับผัก ,รหัสผัก, รับน้ำจิ้ม,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 |

รับผัก(รหัสผัก, ความต้องการ)

|  |  |
| --- | --- |
| Vegetable\_id | Vegetable\_get |

รับน้ำจิ้ม(รหัสน้ำจิ้ม, ความต้องการ)

|  |  |
| --- | --- |
| Sauce\_id | Sauce\_get |

**Normalization**

คำสั่งซื้อ

1NF : รหัสบิลใบเสร็จ, รหัสสินค้า ,ชื่อสินค้า, จำนวนสินค้า, ราคารวม ,รับผัก ,รหัสผัก, รับน้ำจิ้ม,รหัสน้ำจิ้ม

2NF : รหัสบิลใบเสร็จ, รหัสสินค้า,จำนวนสินค้า, ราคารวม ,รับผัก ,รหัสผัก, 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

ผัก

1NF : รหัสผัก, ชื่อผัก

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

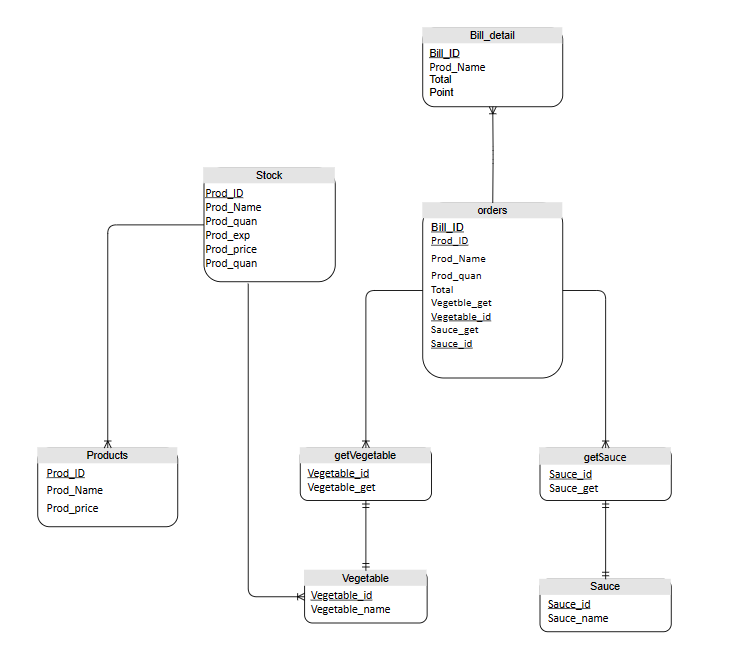
2NF : เป็นแล้ว

3NF : เป็นแล้ว

### **SQL**

|  |  |
| --- | --- |
| ตาราง order  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  ); | ตาราง Stock  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  ); |
| ตาราง Bill\_detail  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)  ); | ตาราง Products  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  ); |
| ตาราง getVegetable  CREATE TABLE getVegetable (  Vegetable\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,  Vegetable\_get VARCHAR(100) NOT NULL  ); | ตาราง Vegetable  CREATE TABLE Vegetable (  Vegetable\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,  Vegetable\_name VARCHAR(100) NOT NULL  ); |
| ตาราง getSauce  CREATE TABLE getSauce (  Sauce\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,  Sauce\_get VARCHAR(100) NOT NULL  ); | ตาราง Sauce  CREATE TABLE Sauce (  Sauce\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,  Sauce\_name VARCHAR(100) NOT NULL  ); |

## **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,

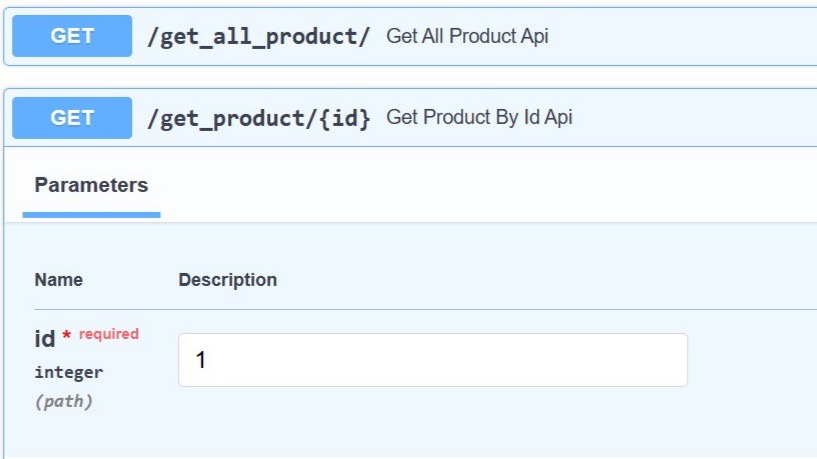
}

**คำสั่ง SQL**

INSERT INTO `product ` (`id`, `product\_name`, `description`, `price`)

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

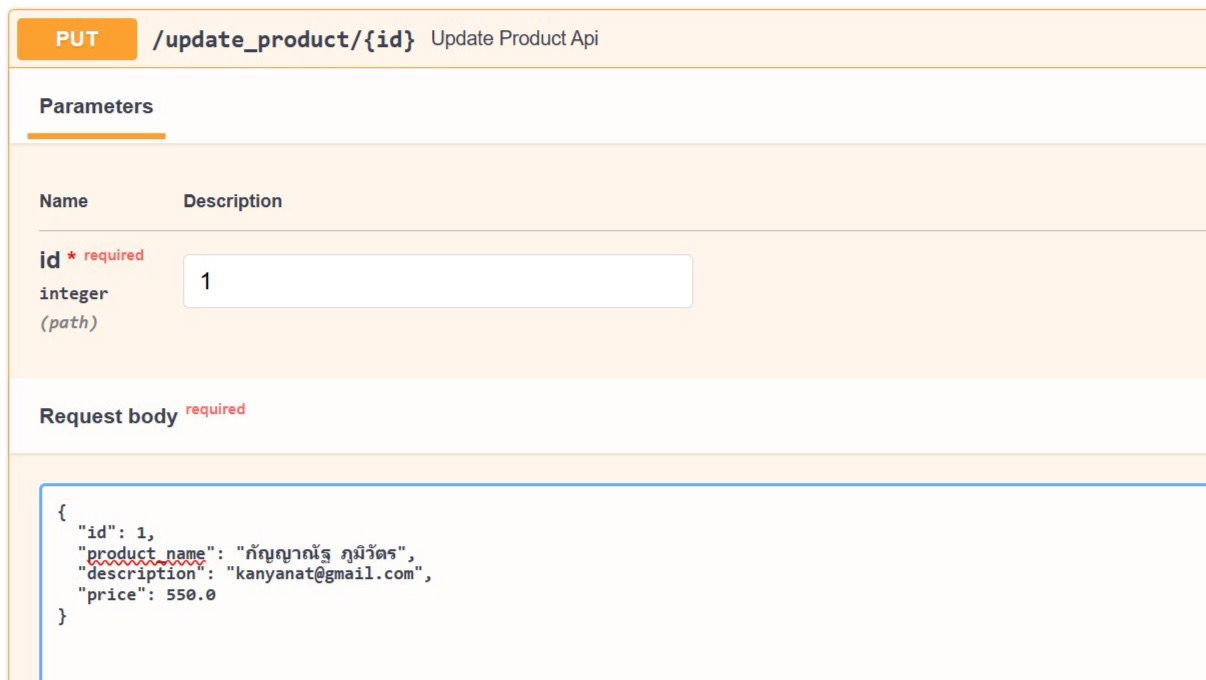
* **Get Product – เรียกดูข้อมูล**



**คำสั่ง SQL**

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

* **Put Product – อัปเดตแก้ไขข้อมูล**



**คำสั่ง SQL**

[UPDATE](http://localhost:8040/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) `product ` [SET](http://localhost:8040/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) `price` = '550.0' WHERE `product`.`id` = '1'

* **Delete Product - ลบข้อมูล**

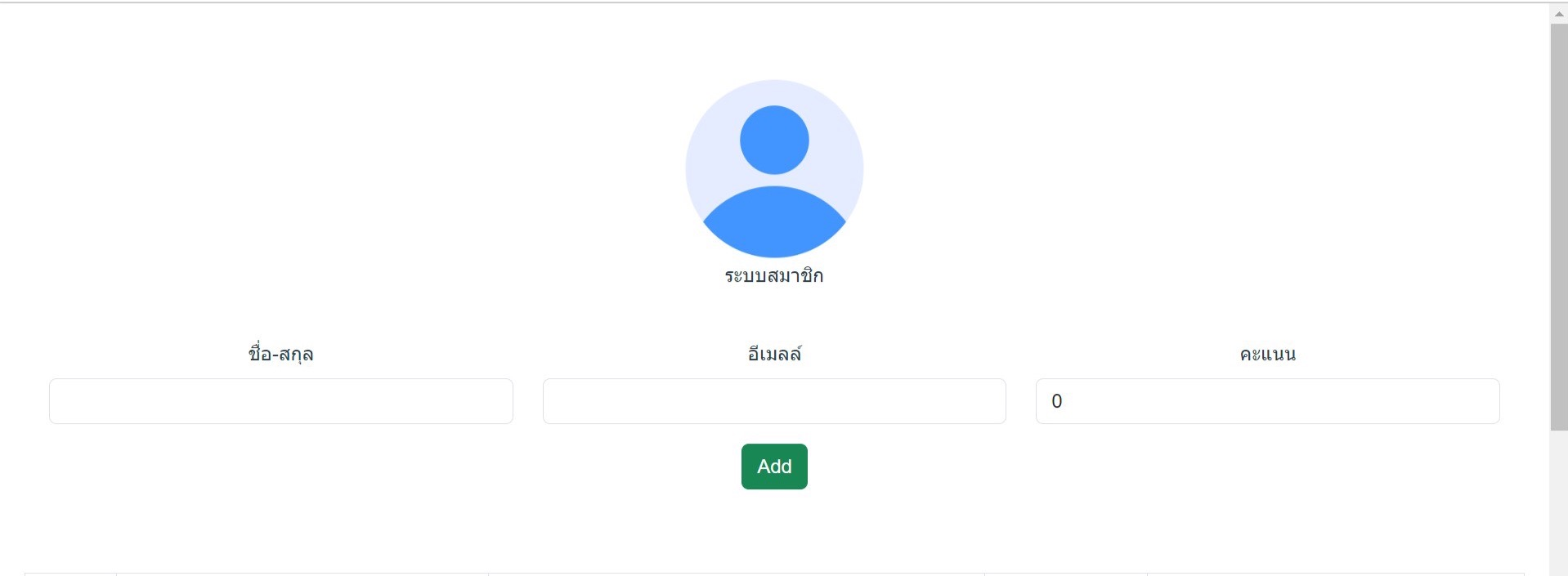


**คำสั่ง SQL**

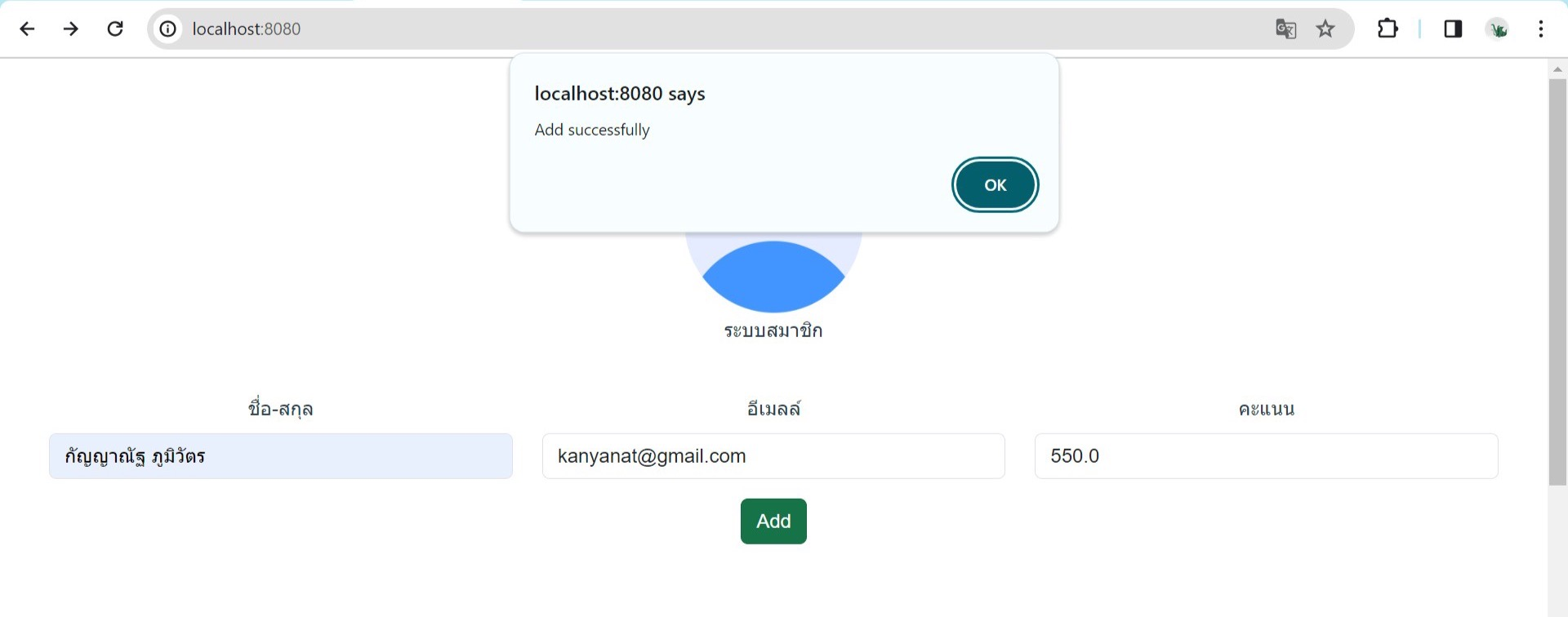
DELETE FROM `product ` WHERE id = '1';

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

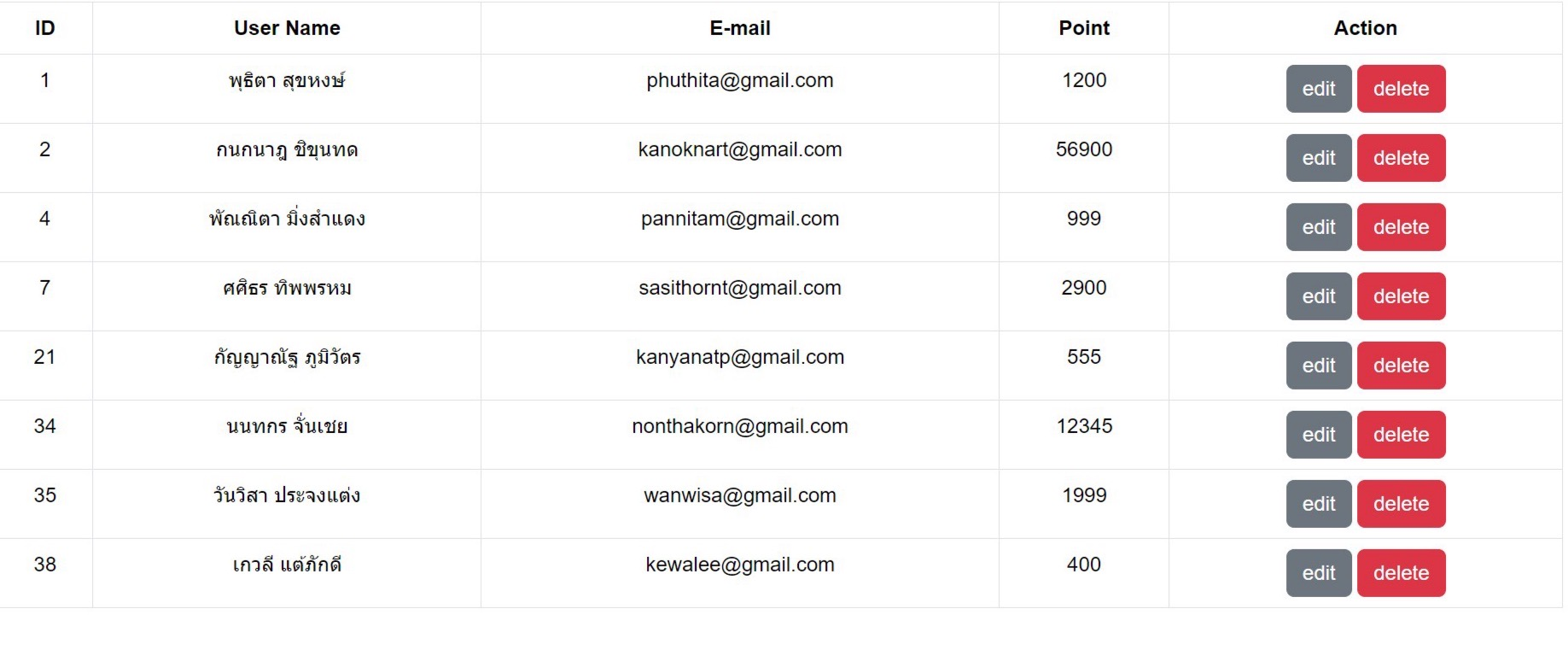
1.หน้า Vue ระบบสมาชิก



2.เพิ่มข้อมูล

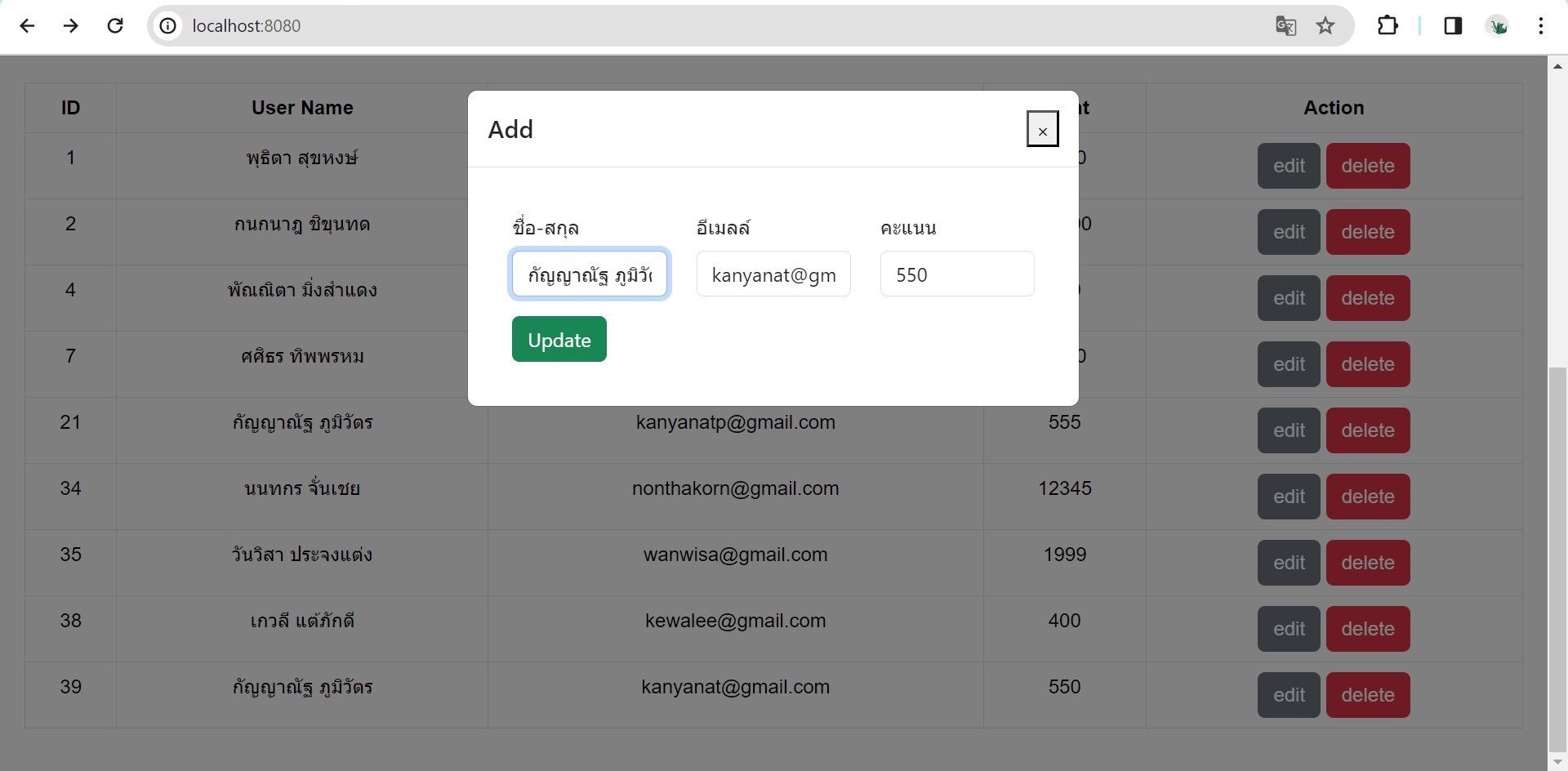


เมื่อเพิ่มข้อมูลแล้วจะขึ้น alert บอกว่าเพิ่มข้อมูลสำเร็จ “Add successfully” หลังจากกด “OK” ข้อมูลจะเพิ่มลงไป

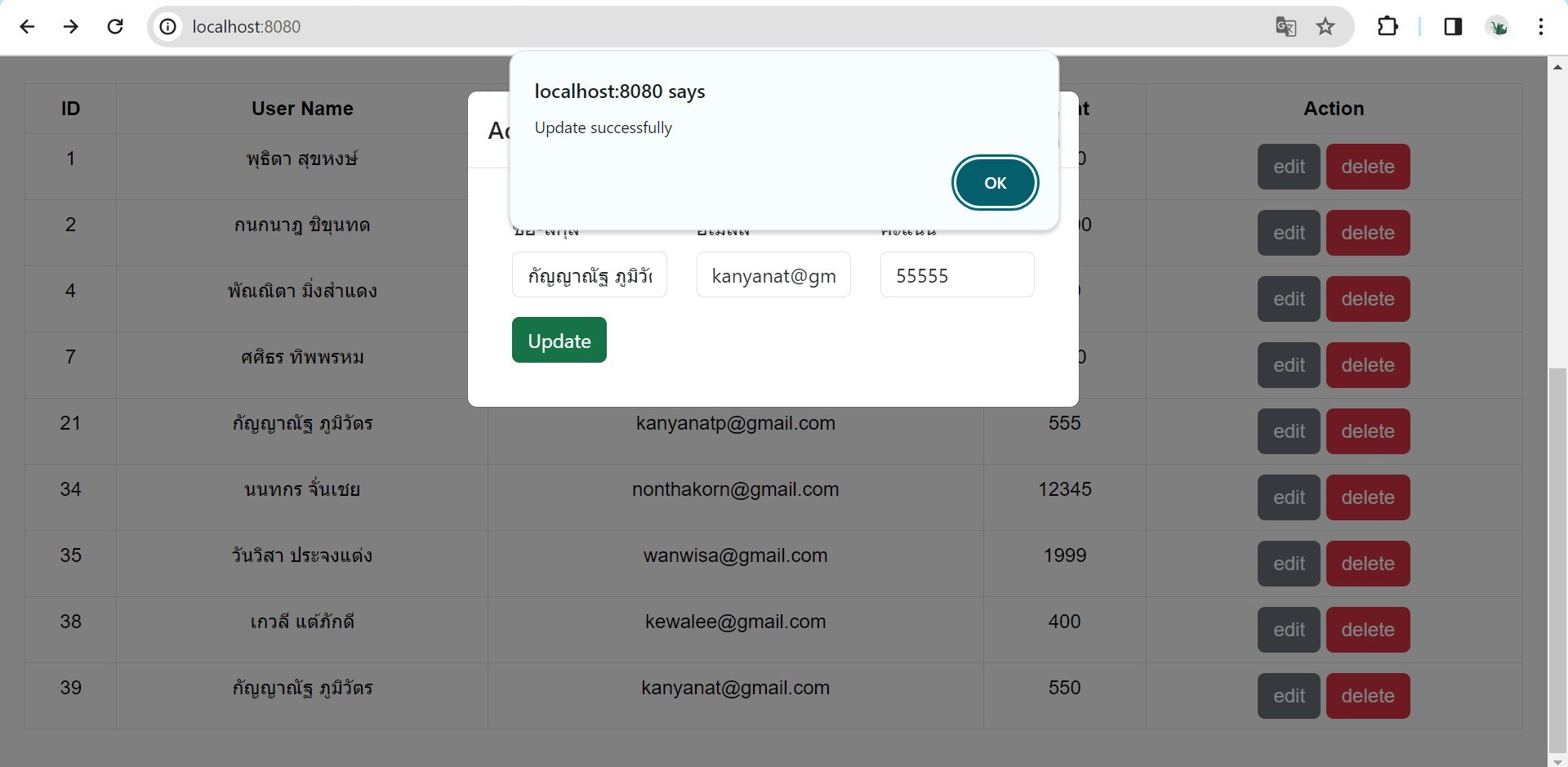


ข้อมูลจะเพิ่มลงมาในหน้านี้

3.แก้ไขข้อมูล



“edit” เพื่อแก้ไขข้อมูลใหม่ไปแทนที่ข้อมูลเก่าที่ต้องการเปลี่ยนแปลง

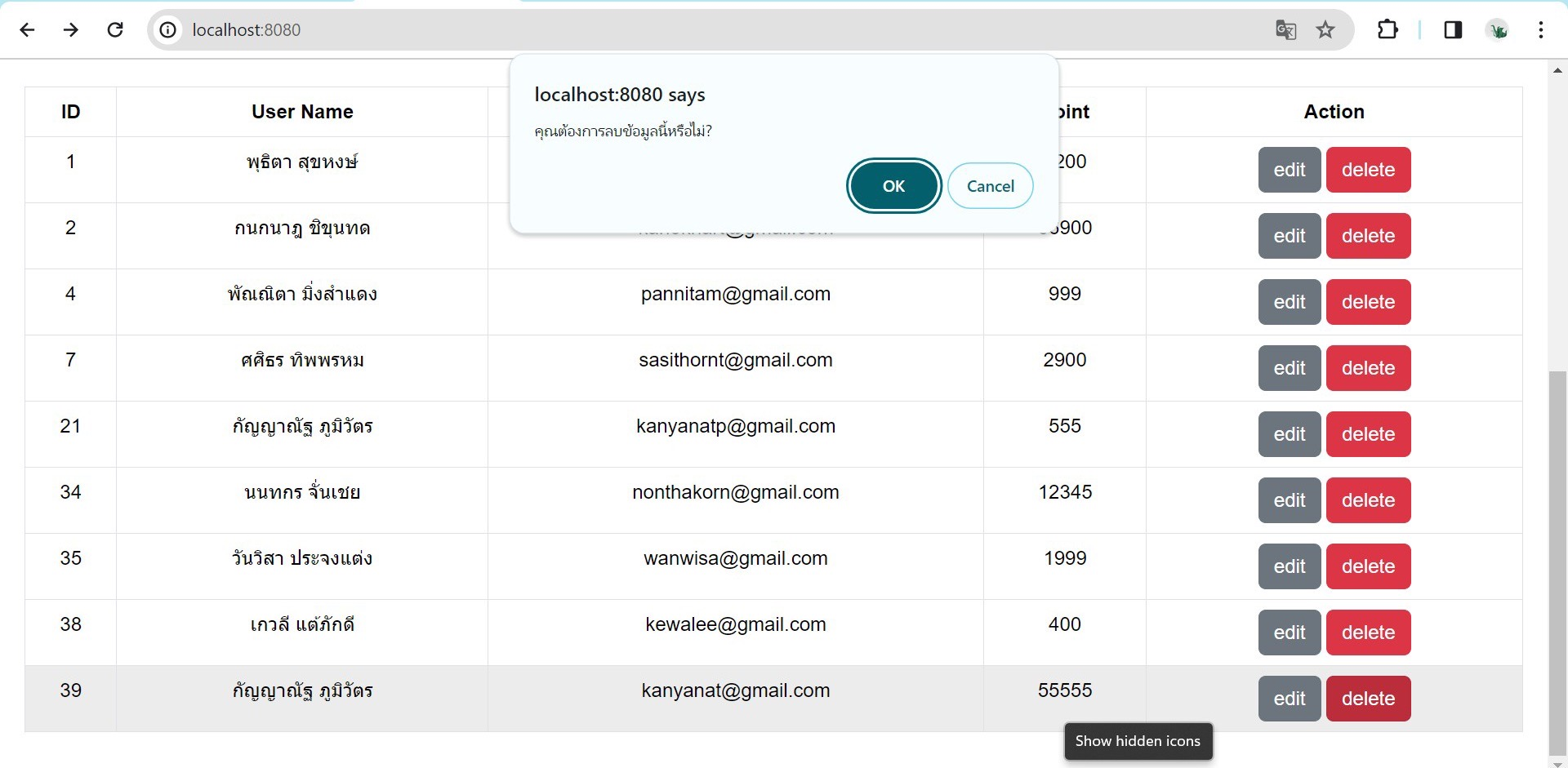


เมื่อกด “Update” เพื่อแก้ไขข้อมูล จะมี alert แจ้งว่าสำเร็จด้วยข้อความ “Update successfully” ข้อมูลทุกอย่างที่แก้ไข จะเปลี่ยนแปลงเมื่อกดปุ่ม “OK”

4.ลบข้อมูล

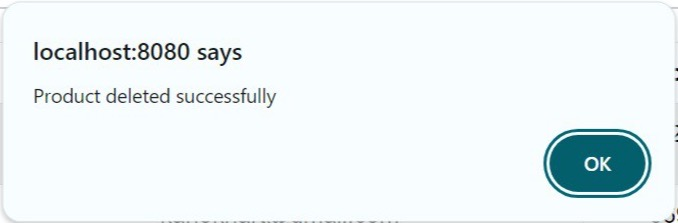


กดเลือกข้อมูลที่ต้องการลบที่ปุ่ม “delete ” จากนั้น

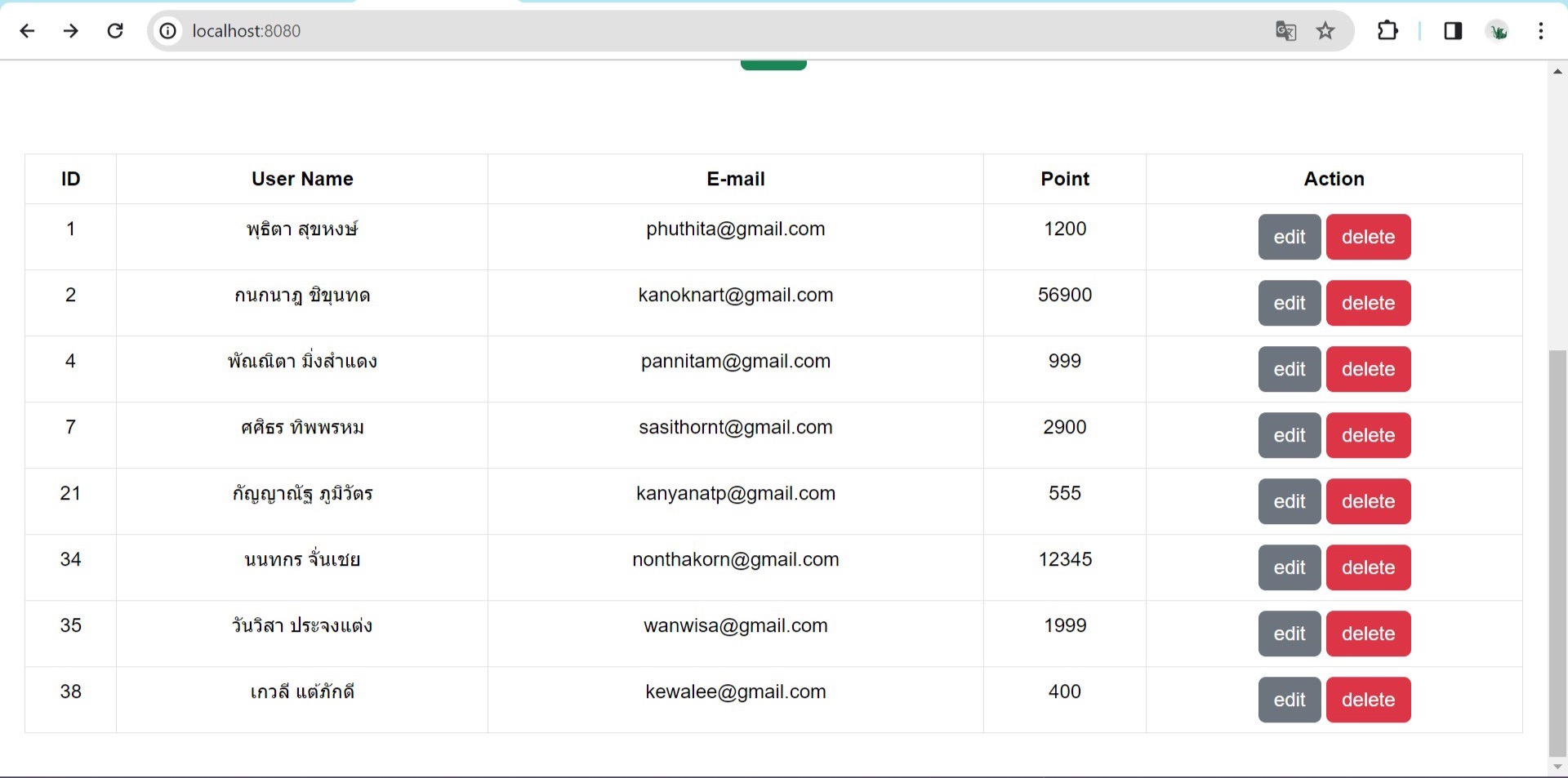


หลังจากกดปุ่ม “delete” แล้วจะขึ้น alert พร้อมข้อความถามว่า “คุณต้องการลบข้อมูลนี้หรือไม่?”

หากต้องการลบให้กดปุ่ม “OK” เพื่อทำการลบข้อมูลที่ไม่ต้องการ แต่หากไม่ต้องการลบกดปุ่ม “cancel” ด้านข้าง



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



# 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": "ปูอัด",

"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', 'ปูอัด', '2024-04-12', '5.0', '50');

* **Get Product – เรียกดูข้อมูล**

**<http://localhost:8001/get_product/P004>**

****

รูปที่ 1 Get Product

**คำสั่ง SQL**

SELECT \* FROM `product\_system` WHERE prod\_id ='P004';

* **Put Product – อัปเดตแก้ไขข้อมูล**

**<http://localhost:8001/update_product/P004>**

****

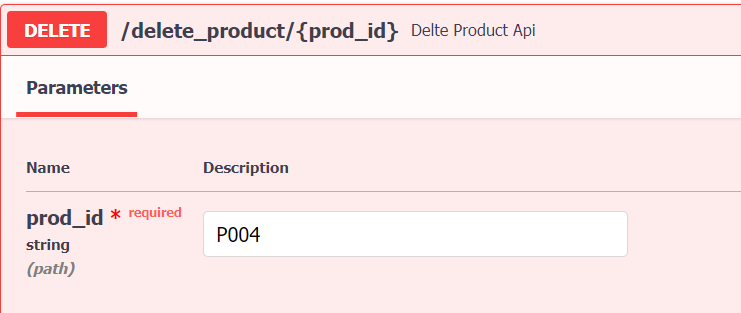
รูปที่ 2 Put Product

**คำสั่ง SQL**

[UPDATE](http://localhost:8040/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) `product\_system` [SET](http://localhost:8040/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) `prod\_quan` = '20' WHERE `product\_system`.`prod\_id` = 'P004'

* **Delete Product - ลบข้อมูล**

**<http://localhost:8001/delete_product/P004>**

****

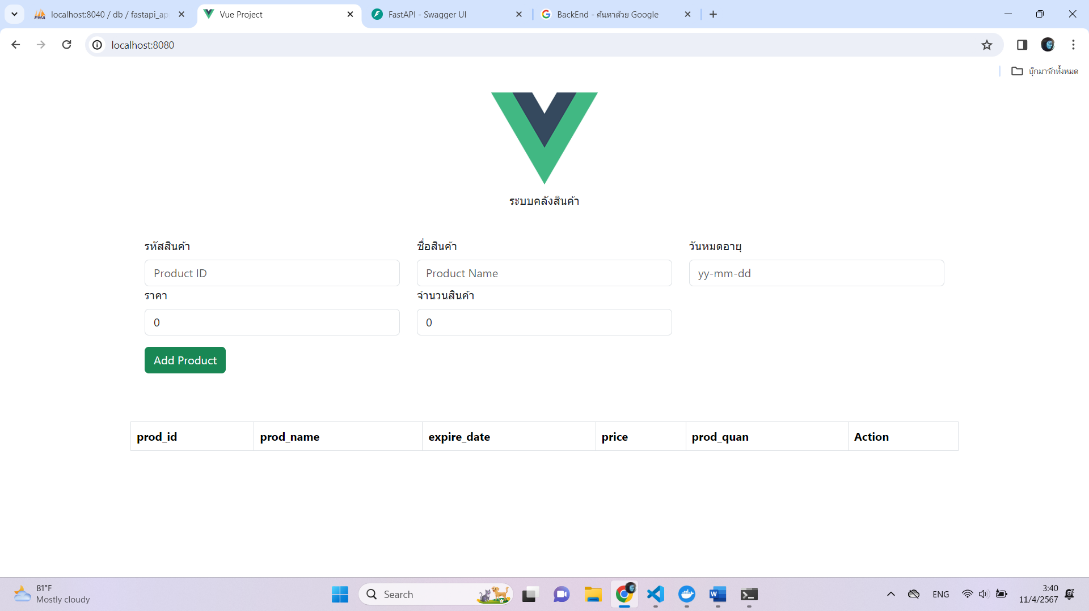
รูปที่ 3 Delete Product

**คำสั่ง SQL**

DELETE FROM `product\_system` WHERE prod\_id = 'P004';

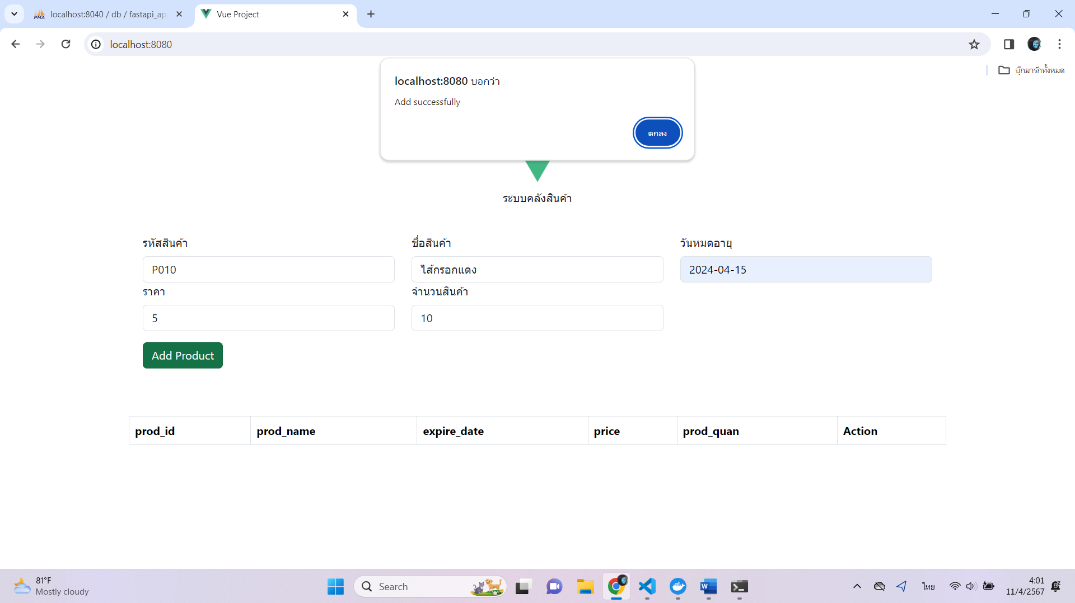
## Front End ใช้ Vue.js(วิว) สามารถสร้างข้อมูล แก้ไขข้อมูล ลบข้อมูลได้ในหน้าเดียวกัน โดยข้อมูลจะเชื่อมต่อกับ Back End และใช้งานผ่าน Docker

1. หน้า Vue ระบบคลังสินค้า



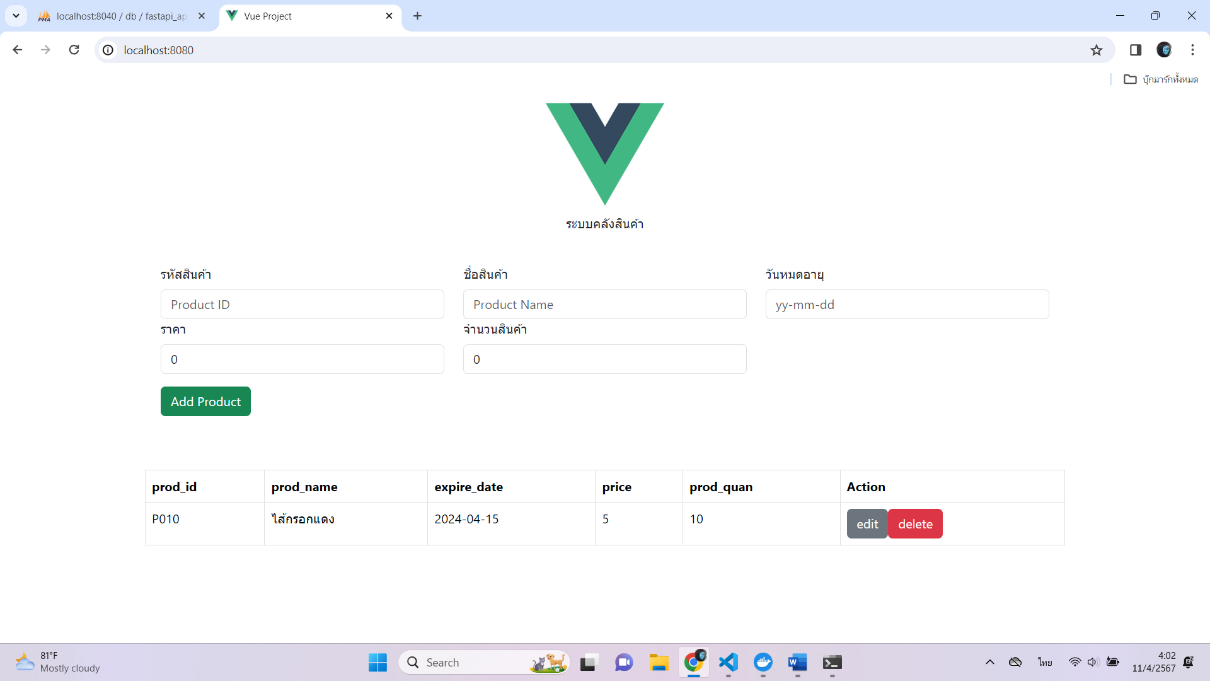
รูปที่ 4 ระบบคลังสินค้า

1. เพิ่มข้อมูล



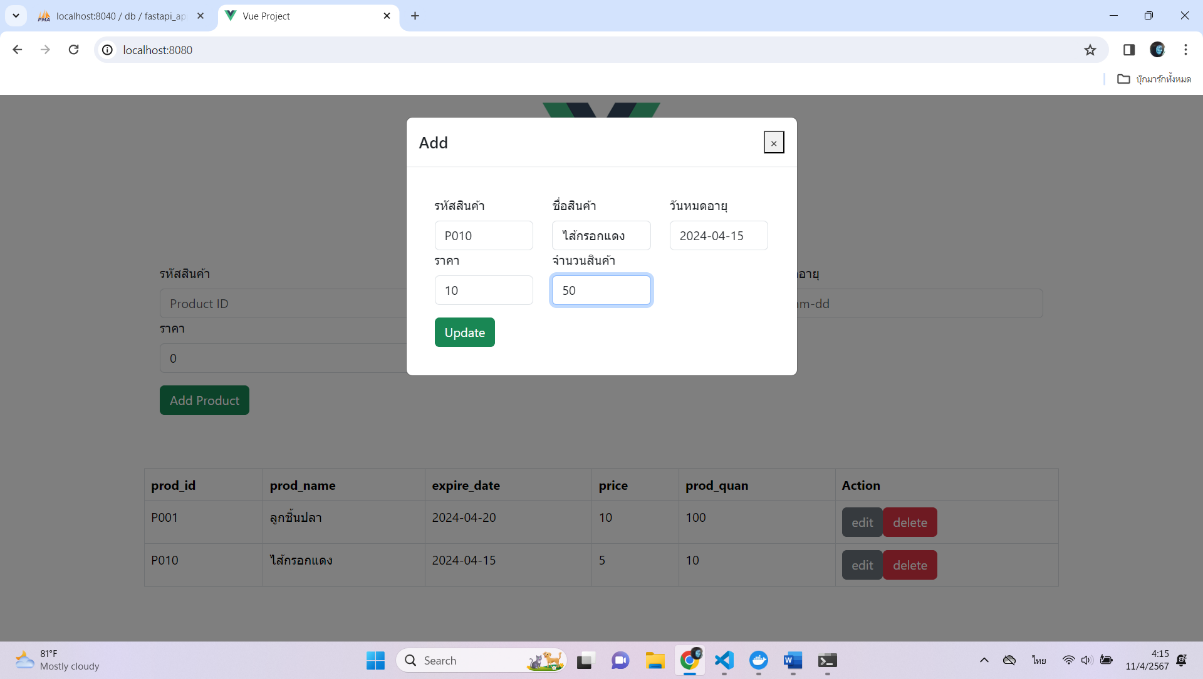
รูปที่ 5 แจ้งเตือนการเพิ่มข้อมูล

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



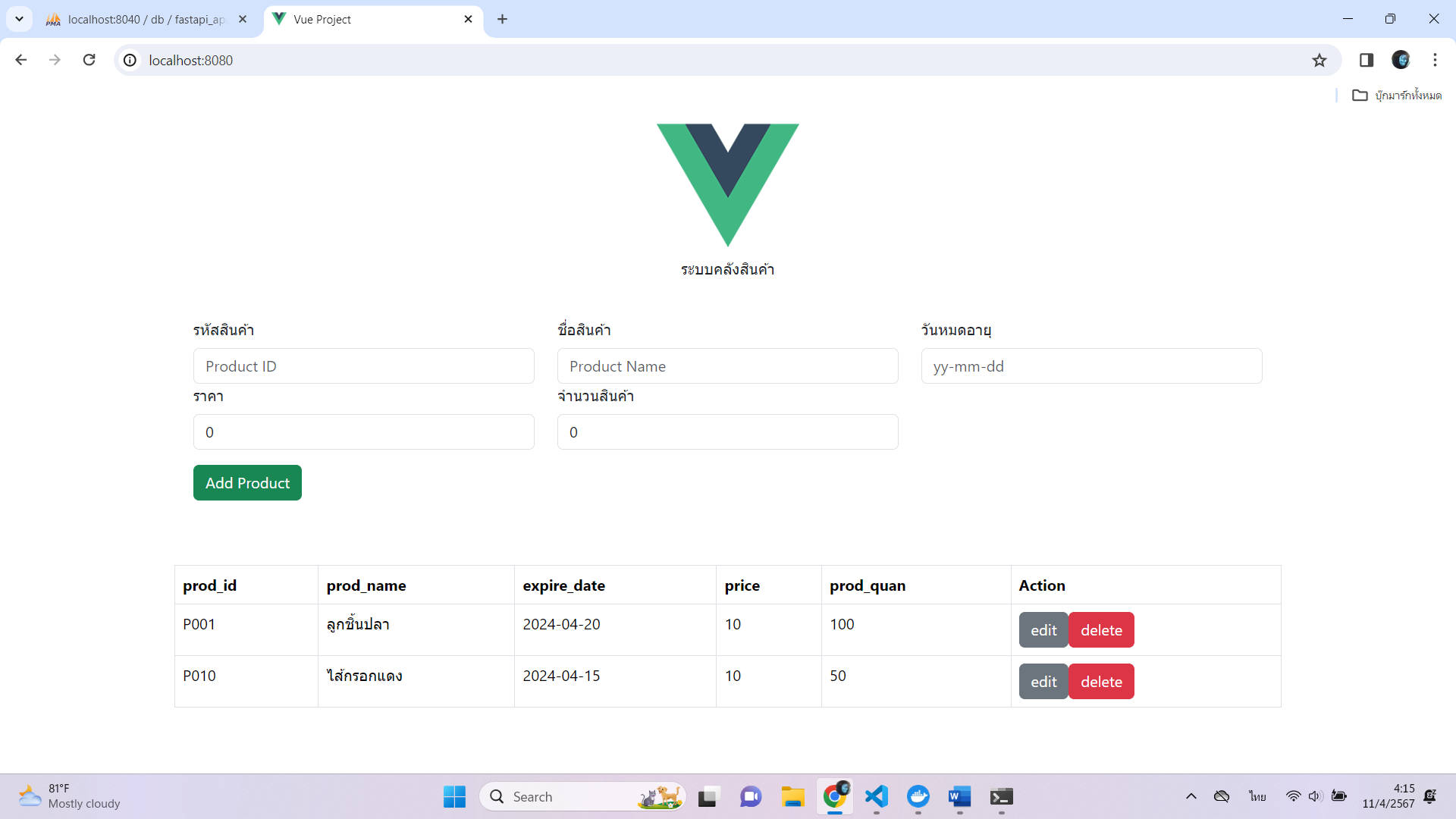
รูปที่ 6 ข้อมูลถูกเพิ่มในระบบ

1. แก้ไขข้อมูล



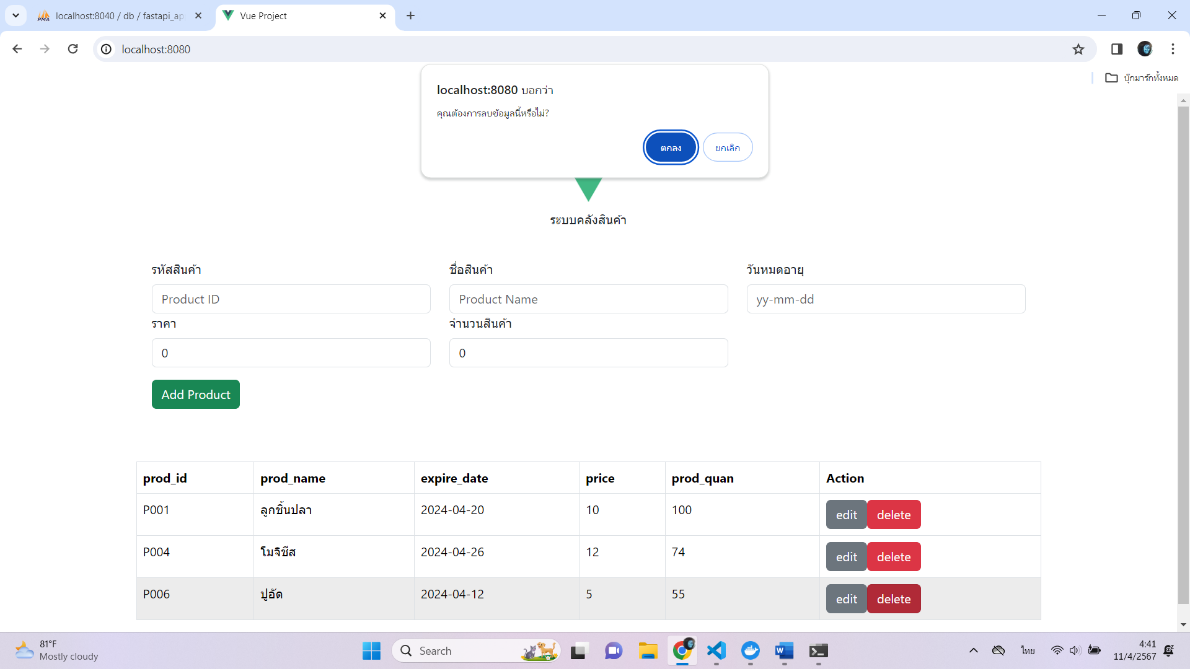
รูปที่ 7 การแก้ไขข้อมูล

เมื่อกดปุ่ม **edit** จะมีหน้าให้แก้ไขข้อมูล จากตัวอย่างมีการอัปเดตข้อมูลที่ **ราคา(price)** และ**จำนวนสินค้า(prod\_quan)** จะได้ผลลัพธ์ดังนี้

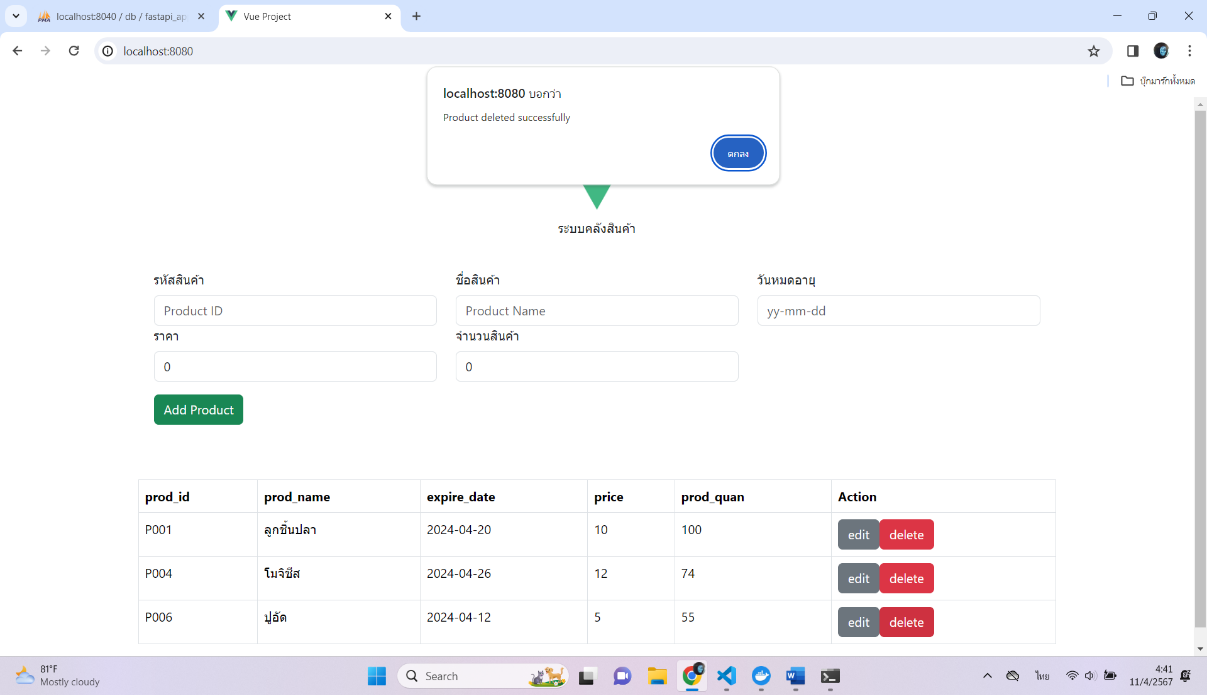


รูปที่ 8 ข้อมูลถูกแก้ไขแล้ว

1. ลบข้อมูล



รูปที่ 9 การแจ้งเตือนลบข้อมูล



รูปที่ 10 แสดงข้อความว่าข้อมูลถูกลบแล้ว

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

ผลลัพธ์ที่ได้คือ prod\_id = P006 หายไป



รูปที่ 11 ข้อมูลที่เหลือในระบบ

**ระบบรับคำสั่งซื้อและบิลใบเสร็จ**

**Back End** ใช้ Rest API เชื่อมต่อกับ MySQL ด้วย Gin Framework

ตาราง bill\_detail

- สร้างตาราง bill\_detail

CREATE TABLE bill\_detail (

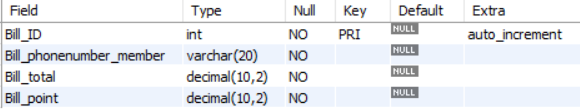
Bill\_ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Bill\_phonenumber\_member VARCHAR(20) NOT NULL,

Bill\_total DECIMAL(10,2) NOT NULL,

Bill\_point DECIMAL(10,2) NOT NULL );

describe bill\_detail ;



**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) ;

**การแสดงบิลใบเสร็จ – Get**

A white background with blue text

Description automatically generated

**คำสั่ง SQL**

select

Bill\_ID,

Bill\_phonenumber\_member,

Bill\_total,

Bill\_point

from bill\_detail ;

**การลบบิลใบเสร็จ – Delete**

A white background with blue text

Description automatically generated

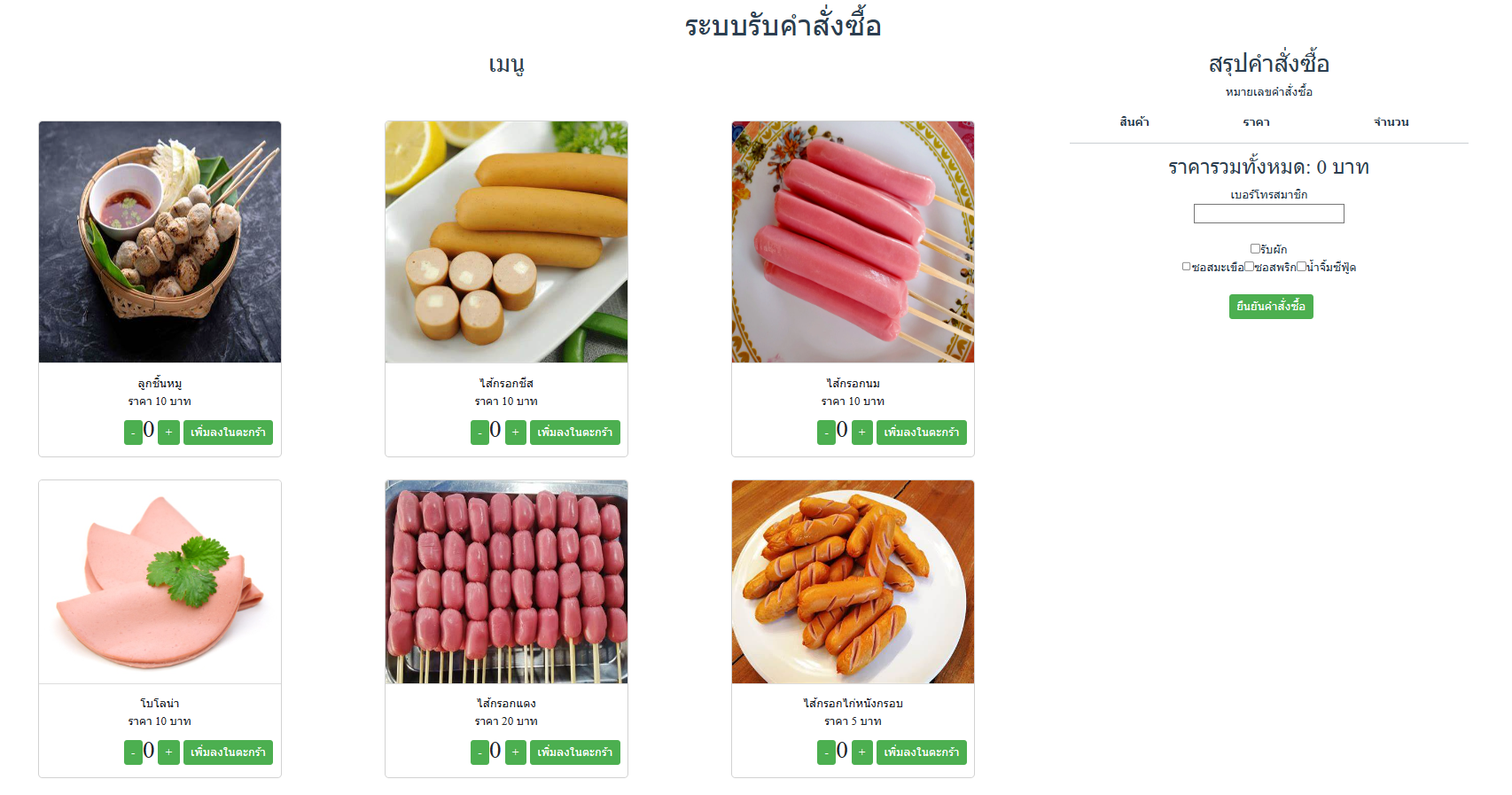
**คำสั่ง SQL**

DELETE FROM bill\_detail

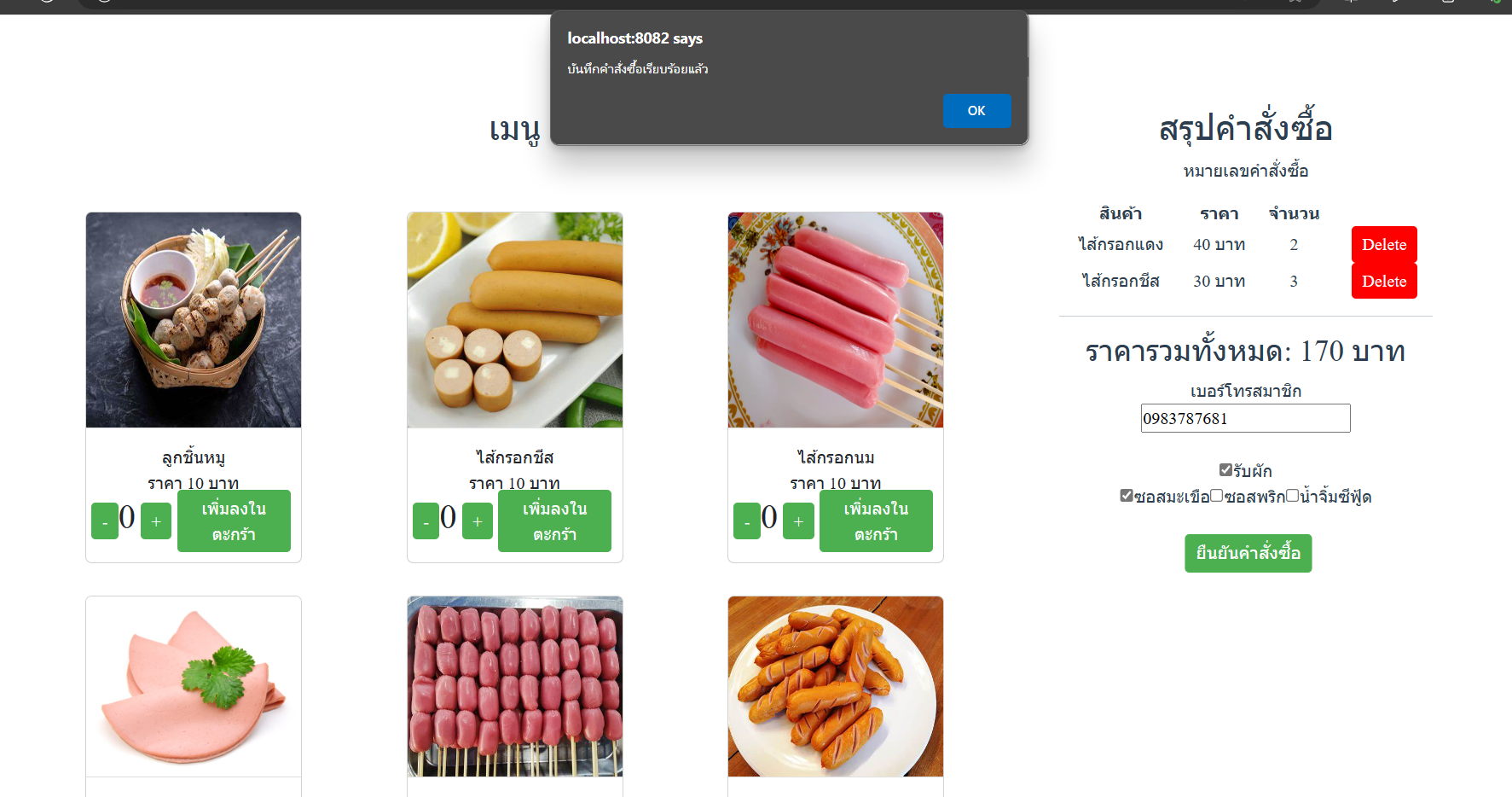
WHERE Bill\_ID = ?

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

1. หน้ารับคำสั่งซื้อ

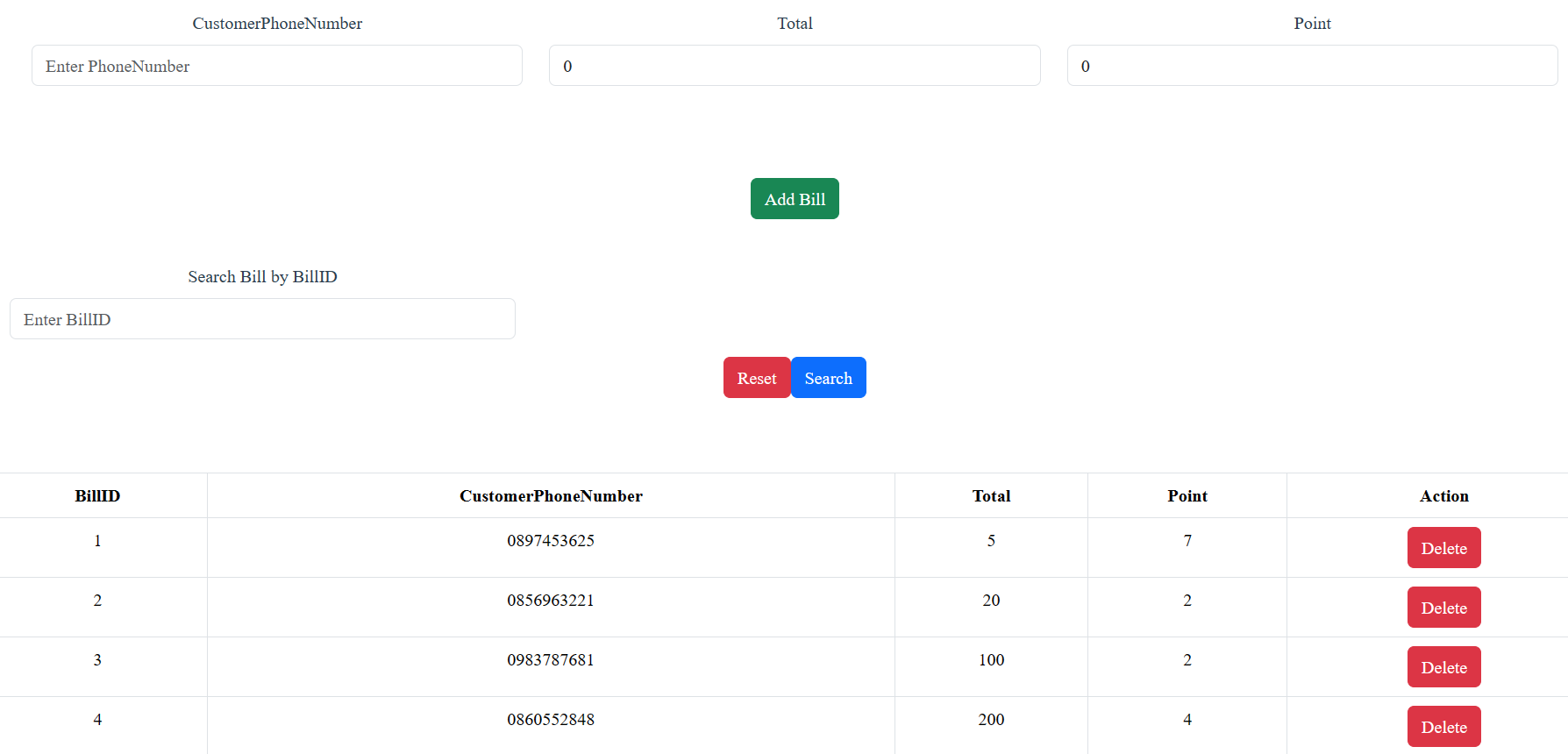


1. หน้าการบันทึกการสั่งซื้อและสร้างบิลใบเสร็จ

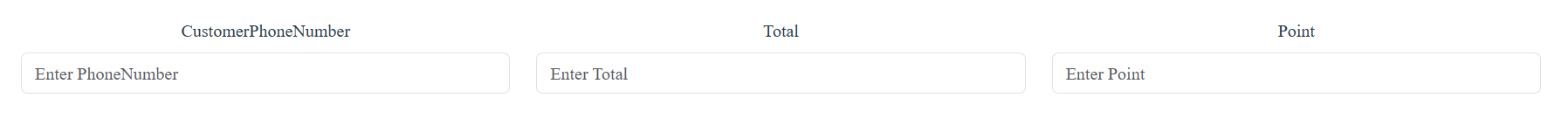


เมื่อกด “ยืนยันคำสั่งซื้อ” ข้อมูลจะถูกนำไปสร้างบิลใบเสร็จ

1. หน้าบิลใบเสร็จ



3.1 การเพิ่มข้อมูลบิลใบเสร็จ

3.2 การค้นหาบิลใบเสร็จจาก Bill\_ID

A white rectangular object with a blue line

Description automatically generated with medium confidence

* 1. การแสดงข้อมูลบิลใบเสร็จทั้งหมดA person pointing at something

     Description automatically generated
  2. การลบข้อมูลจากในฐานข้อมูล

A person pointing at something

Description automatically generated

**บรรณานุกรม**

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