



## **มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี**

โครงการเรื่อง

**การพัฒนาเว็บแอปพลิเคชันฟังก์ชันของ Marketplace**

**วิชา Web Technology and Platform รหัสวิชา DSI 215**

จัดทำโดย

นายชนกฤต สุนทรเมือง รหัสนักศึกษา 65130500407

นางสาวธนัชญา โสภณกิตติกุล รหัสนักศึกษา 65130500408

นางสาวภัทรานิษฐ์ สันติชนวิกรัย รหัสนักศึกษา 65130500417

นายสิริวิชญ์ ธรรมวิหาร รหัสนักศึกษา 65130500426

นายผดุงวัฒน์ คงตุก รหัสนักศึกษา 65130500429

นายสุชาครีย์ ศรีบุญเรือง รหัสนักศึกษา 65130500445

เสนอ

อาจารย์ นันทพงศ์ เจียนดวงจันทร์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ภาคการศึกษาที่ 2 ปีการศึกษาที่ 2566

## ที่มา

ย้อนกลับไปปี 2014 Facebook เริ่มทดสอบฟีเจอร์ "Groups for Sale" ซึ่งเป็นกลุ่มที่ผู้ใช้สามารถโพสต์สินค้าเพื่อขายภายในกลุ่มได้ ฟีเจอร์นี้ได้รับความนิยมอย่างมาก แสดงให้เห็นถึงความต้องการของผู้ใช้ Facebook ในการซื้อขายสินค้า

## ปัญหา

- ตัวเลือกการกรองมีจำกัด เวลาที่เราต้องการหาสินค้า เมื่อใช้ฟิเตอร์การกรอง สามารถกรองสินค้าตามที่เราต้องการได้น้อยและมีจำกัด
- การจัดหมวดหมู่กระจายและไม่เป็นระเบียบ เช่น มีการแบ่ง category ของสินค้าที่เป็นยานพาหนะ แต่รูปที่แสดงขึ้นมาก็คือสินค้าที่ไม่เกี่ยวกับยานพาหนะ
- ผู้ใช้ใหม่มองว่า Market Place ในรูปแบบเก่าใช้งานยากต้องทักหาผู้โพสต์สินค้าก่อนถึงจะทำการซื้อขายสินค้าได้ ทำให้มีขั้นตอนที่เยอะเกินไป
- ในเรื่องของ Interface ที่ไม่เป็นมิตรต่อผู้ใช้งาน : UI บางส่วนทำให้ USER สับสนกับการใช้งานและไม่รู้ว่าจะกดปุ่มไหน หรือ ปุ่มนั้นมีไว้ทำอะไร
- ขั้นตอนการซื้อสินค้าค่อนข้างยุ่งยาก ต้องมีการทักหา, พูดคุยกับผู้ขายก่อนผู้ใช้ถึงจะซื้อสินค้าได้ ไม่สามารถกดซื้อได้เลย ถ้าผู้ใช้ใจร้อนแล้วผู้ขายตอบช้าก็อาจจะทำให้ขายสินค้าไม่ได้

## ทฤษฎีที่เกี่ยวข้อง โดยโปรแกรมหลักๆที่เรานำมาพัฒนา

- **React** เป็น JavaScript library ที่ใช้ในการสร้างส่วนของหน้า Front end ของเว็บไซต์และแอปพลิเคชัน ซึ่งถูกพัฒนาโดย Facebook ใช้ในการสร้าง UI components ที่สามารถนำกลับมาใช้ใหม่ได้ เช่น ปุ่ม, แบบฟอร์ม, และองค์ประกอบอื่นๆ บนหน้าเว็บ
- **Post man** เป็นเครื่องมือที่ใช้ในการทดสอบ API (Application Programming Interface) โดยช่วยให้ผู้พัฒนาสามารถส่งคำขอ HTTP และดูผลลัพธ์ที่ได้จากเซิร์ฟเวอร์ ใช้ในการทดสอบ API เพื่อดูว่าทำงานถูกต้องหรือไม่ เช่น การส่งคำขอ GET, POST, PUT, DELETE ไปยังเซิร์ฟเวอร์ และดูผลลัพธ์ที่ได้
- **MySQL**: MySQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ที่ใช้ SQL (Structured Query Language) ในการจัดการและเรียกดูข้อมูล ใช้ในการเก็บข้อมูลต่างๆ ของแอปพลิเคชัน เช่น ข้อมูลผู้ใช้, ข้อมูลสินค้า, หรือข้อมูลอื่นๆ ที่ต้องการการจัดการแบบเชิงสัมพันธ์ MySQL มีความสามารถในการจัดการข้อมูลขนาดใหญ่และมีความเสถียรสูง
- **Node.js**: Node.js เป็น JavaScript runtime environment ที่ทำงานบนเซิร์ฟเวอร์ ทำให้สามารถเขียนโค้ด JavaScript สำหรับฝั่งเซิร์ฟเวอร์ได้ Node.js มีความสามารถในการพัฒนา back end ของแอปพลิเคชัน เช่น การสร้าง RESTful API, การเชื่อมต่อกับฐานข้อมูล, และการจัดการเซิร์ฟเวอร์ Node.js มีความเร็วและประสิทธิภาพสูงเนื่องจากใช้ V8 engine ของ Google Chrome ในการประมวลผล
- **Figma**: Figma เป็นเครื่องมือออกแบบที่ทำงานบนคลาวด์ ใช้สำหรับการออกแบบ UI/UX ของเว็บไซต์และแอปพลิเคชัน มีจุดเด่นที่สามารถทำงานร่วมกันในทีมแบบเรียลไทม์ ทำให้นักออกแบบหลายคนสามารถทำงานบนโปรเจกต์เดียวกันพร้อมกันได้ Figma ใช้ในการออกแบบโครงร่างหน้าเว็บ, การสร้าง prototype สำหรับการทดสอบการใช้งาน, และการทำงานร่วมกันในทีมออกแบบ

# 1. ส่วนของการใช้งานเว็บแอปพลิเคชัน

## 1.1 Navbar

ในส่วนของ แถบ Navbar เราจะตัวเลือกฟังก์ชันหลักๆ ในกับผู้ใช้งาน ได้เลือกใช้ โดยจะแบ่งเป็นสามส่วนหลักๆ

ส่วนที่1 ฟังก์ชัน เป็นส่วนของโลโก้เว็บไซต์

ส่วนที่2 ตรงกลางมีตัวเลือก Home, Add Product, Shop

ส่วนที่3 และฟังก์ชันเป็น Login, Cart, Profile โดยที่เรามีการระบุ ลิงก์เอาไว้เพื่อให้ผู้ใช้งาน สามารถคลิกเพื่อสามารถเปลี่ยนไปยังหน้าต่างที่ต้องการได้ และจากในภาพเรามีการทำ Login มีให้มีขนาดตัวที่สามารถมองเห็นได้ชัด ไว้สำหรับเวลาที่ผู้ใช้งาน ต้องการที่จะเข้าสู่ระบบ



Home

Add Product

Shop

Login



Profile

## 1.2 SIGN UP

หน้านี้เป็นหน้า SIGN UP สำหรับผู้ใช้งาน ที่ยังไม่มีบัญชีในระบบของเว็บไซต์เรา โดยเราจะมีกรเก็บข้อมูลของผู้ใช้งาน ทั้งหมด 4 ส่วนหลักๆ โดยมี **ชื่อ, นามสกุล, อีเมลล์, รหัสผ่าน** โดยที่เราจะมีกำหนด Pattern ในการลงทะเบียนให้กับผู้ใช้งาน ไว้ โดย Pattern ที่ตั้งไว้คือหากผู้ใช้งาน ไม่มีการกรอกข้อมูลมาหรือข้อมูลเป็นค่าว่างเราจะทำการรีเทิร์นค่ากลับไปโดยแจ้งให้ผู้ใช้งาน กรอกข้อมูลให้ครบทุกช่อง และ มีการกำหนด Pattern ของ รหัสผ่าน เพื่อความปลอดภัยของผู้ใช้งานเอง โดยเราจะมีการระบุว่าผู้ใช้งาน ควรกรอกรหัสผ่าน ที่มี A-a 0-9 และอักขระพิเศษ ซึ่งหากผู้ใช้งานกรอกข้อมูลไม่ตรงกับค่าที่มีการตั้งไว้เราจะทำการรีเทิร์นค่ากลับไปยังผู้ใช้งานและมีคำอธิบายข้างล่างช่องกรอกว่าควรกรอกข้อมูลอะไรบ้าง และหลังจากที่ผู้ใช้งานกรอกข้อมูลการลงทะเบียนเสร็จจะมรปุ่มกดยืนยันเพื่อแจ้งไปสู่หน้าของการ LOG IN เข้าสู่ระบบต่อ

**SIGN UP**

First Name

Last Name

E-mail

Password

**SIGN UP** **LOG IN**

I have an account ? [Login](#)

### 1.3 LOG IN

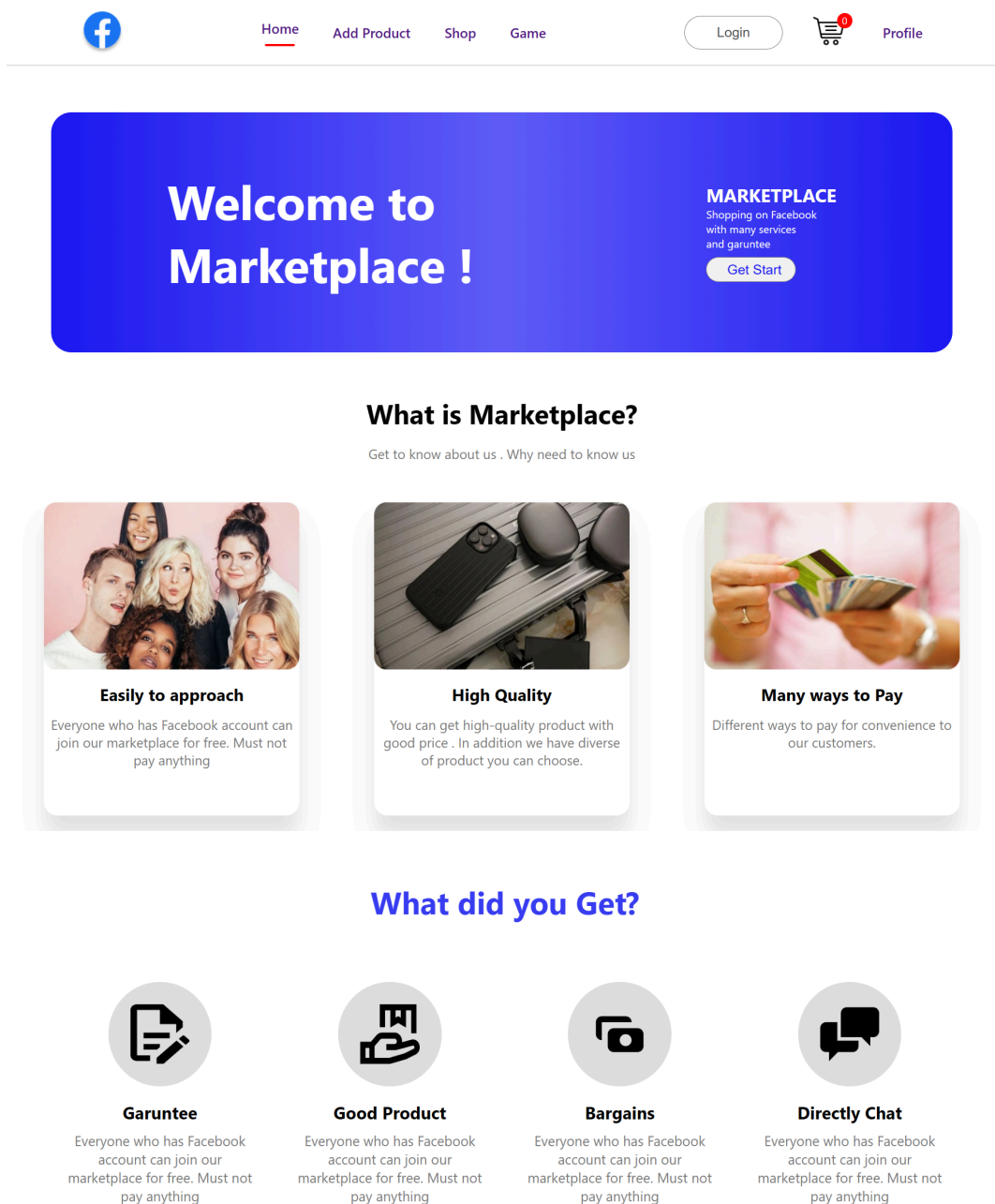
หน้า LOG IN คือหน้าที่ผู้ใช้งานต้องการที่จะเข้าสู่ระบบเว็บไซต์ของเรา โดยข้อมูลที่จะต้องกรอกในการเข้าสู่ระบบคือ **อีเมลล์ และ รหัสผ่านของผู้ใช้งาน** ซึ่งตัวของการ LOG IN เรามีการ SELECT ไปยังตัวของ SIGN UP เพื่อตรวจสอบว่าค่าที่ user กรอกมานั้นตรงกับที่ได้ทำการลงทะเบียนไว้ และในหน้าของ LOG IN เราได้มีการตั้ง Pattern ไว้เช่นเดียวกับหน้าของการลงทะเบียนผู้ใช้งานใหม่ โดยห้ามใส่ค่าว่าง และ ต้องใส่รหัสผ่าน A-a 0-9 และอักขระพิเศษ และหลังจากที่ผู้ใช้งานกรอกข้อมูลเรียบร้อยแล้วจะมีปุ่มกดยืนยันเพื่อเข้าสู่หน้าต่างของเว็บไซต์เรา โดยที่ในหน้าของการ LOG IN เราจะมีการเก็บ SESSION ของผู้ใช้งานเอาไว้เพื่อยืนยันตัวในการเข้าใช้งานในระบบเว็บไซต์

#### 1.4 PROFILE

ในส่วน of หน้า Profile หลังจากที่ ผู้ใช้งานกดเข้าสู่ระบบมาแล้วจะพบกับหน้านี้เพื่อ ยืนยันข้อมูลว่า เราได้ทำการเข้าสู่ระบบเรียบร้อยแล้ว โดยที่ส่วน of หน้าต่างที่จะมีการ โชว์ ข้อมูล ชื่อ, อีเมลล์ และ รหัสผ่าน ของ ผู้ใช้งานสามารถที่จะกด Edit ข้อมูลส่วนตัว โดยที่หลังจากกดก็จะมีหน้าต่างแสดงให้เรารอกข้อมูลใหม่ลงไปได้และหลังจากกดจะมีข้อมูลยืนยันว่ามีการเปลี่ยนแปลงข้อมูลเรียบร้อยแล้ว และดึงกลับไป หน้าแสดงข้อมูลส่วนตัวเพื่อยืนยันกับ ผู้ใช้งานว่าระบบทำการแก้ไขข้อมูลให้เรียบร้อยแล้ว หรือสามารถกด Delete บัญชีทิ้งได้หาก ผู้ใช้งาน ไม่ต้องการแล้ว โดยในส่วน of การลบบัญชีเราจะมี Pop up เด้งขึ้นมาให้ ผู้ใช้งาน กดยืนยันอีกครั้ง เพื่อป้องกันการกดโดนโดยที่ไม่ได้ตั้งใจ และหาก ผู้ใช้งาน กดยืนยันว่าต้องการลบบัญชีก็ จะมีการขึ้นข้อความว่าลบบัญชีเสร็จสิ้นและดึงกลับไปยัง หน้าของการ SIGN UP


## 1.5 HOME

ในส่วนของหน้า HOME จะเป็นการแนะนำฟีเจอร์ของ Marketplace หรือเว็บไอร์แลนด์ที่เราได้ทำขึ้น โดยหน้านี้เราจะแบ่งเป็นทั้งหมด 3 Sections โดยส่วนที่ 1 จะเป็นส่วนของ banner และมีปุ่มกดทางขวามือเพื่อเริ่มต้นลงทะเบียนใช้งานของผู้ใช้งาน ต่อมาเป็นส่วนที่ 2 จะเป็นการอธิบายถึงฟีเจอร์ Marketplace ว่าคืออะไร และใช้งานยังไง และส่วนสุดท้าย ส่วนที่ 3 คือการอธิบายและบอกให้กับผู้ใช้งานได้ทราบถึงประโยชน์การใช้งานในส่วนของ Marketplace ว่าเขาจะได้รับอะไรบ้าง



## 1.6 Add Product

หน้า Add Product คือหากผู้ใช้งานคนไหนที่มีความสนใจที่จะลงขายสินค้ากับทางเว็บไซต์เรา หลังจากที่เข้าสู่ระบบมาแล้ว จะสามารถกดเข้ามาที่ฟังก์ชันการลงขายสินค้าได้ โดยในหน้านี้จะมีบอกข้อมูลรายละเอียดต่างๆ ที่ต้องใช้ในการลงขายสินค้าโดยจะต้องกรอกข้อมูลทั้งหมดคือ รูปภาพสินค้า, ชื่อของสินค้า, รายละเอียดของสินค้า, ราคาของสินค้าที่ต้องการขาย, สต็อกของสินค้าว่ามีกี่ชิ้น, สถานะของสินค้าว่าพร้อมขายหรือไม่ หรือว่าสินค้าหมดแล้ว, หมวดหมู่ โดยที่ผู้ใช้งานจะต้องเลือกหมวดหมู่ให้ตรงกับสินค้าที่ต้องการลงขาย และหลังจากที่กรอกข้อมูลเรียบร้อยแล้วจะมีข้อความแจ้งเตือนว่า คุณได้ทำการลงขายสินค้าเรียบร้อยแล้ว



### ลงขายสินค้า

**รายละเอียดสินค้า**

**Upload Files**

Choose File No file chosen

**Product Name**

Product Name

**Description**

Description

**Product Price**

Product Price

**Stock Quantity**

Stock Quantity

**Product Status**

Yes

**Category**

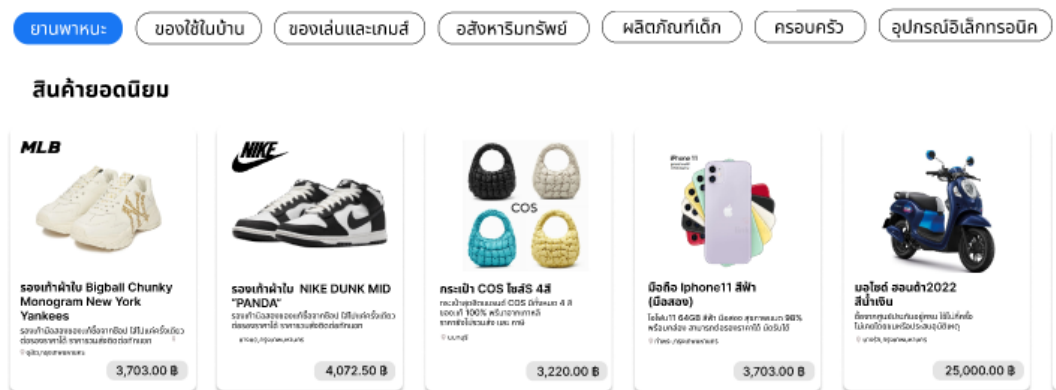
ยานพาหนะ

Submit



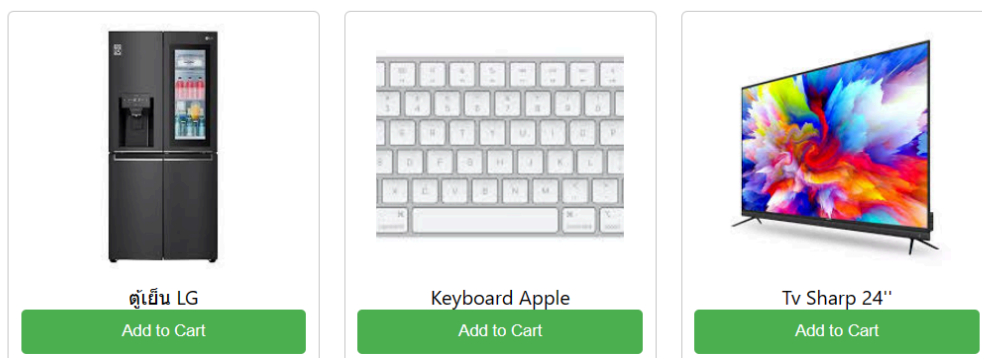
## 1.7 Shop

ส่วนของหน้า Shop จะเป็นการไปดึงข้อมูลจากที่ผู้ใช้งานอื่นๆ มีการลงขายสินค้ามาแสดง ให้กับคนอื่นได้เห็น โดยภายในหน้าดังนี้ จะมีการแสดงข้อมูลรูปภาพสินค้า ชื่อสินค้า ราคาสินค้า และ สถานะสินค้า ก่อนที่ผู้ใช้งานจะตัดสินใจเลือกซื้อ และหากผู้ใช้งานคนใด สนใจสินค้าในเว็บไซต์เรา ก็สามารถที่จะกดถูกใจ หรือบันทึกไว้ก่อนได้แล้วค่อยกลับมาซื้อ หรือตัดสินใจอีกครั้ง และเรายังมีฟีเจอร์การกรองหมวดหมู่ที่หากผู้ใช้งานคนไหนสนใจที่จะเลือกหมวดหมู่ที่เราตั้งไว้ ก็สามารถกดเลือกได้และหลังจากกดแล้ว เว็บเราจะทำการดึงข้อมูลแต่ละหมวดหมู่ออกมาโชว์ให้กับผู้ใช้งานได้เห็น และมีการระบุว่าสินค้าแต่ละหมวดหมู่นี้มีจำนวนกี่ชิ้น



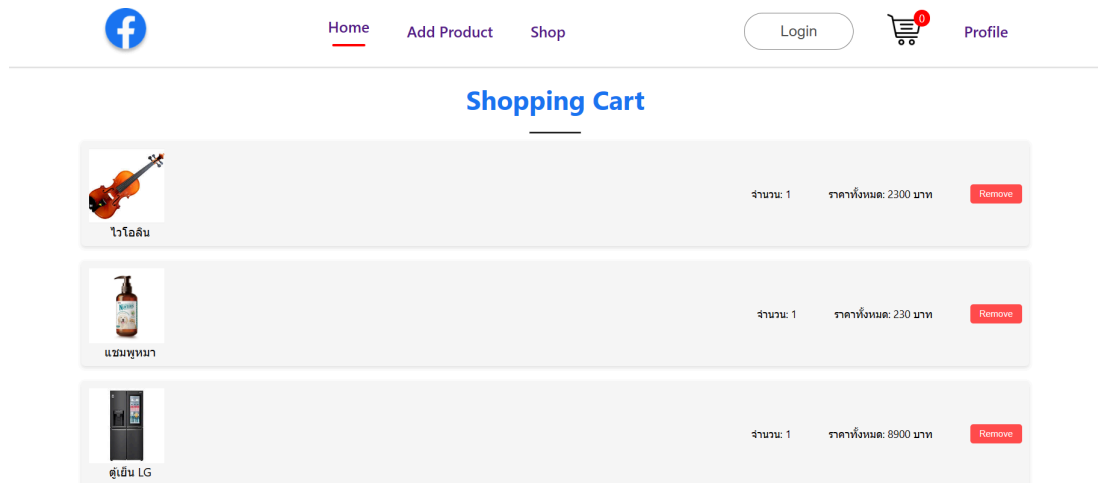
หน้าที่ระบุสินค้าแต่ละหมวดหมู่

Show Out of 3 Products

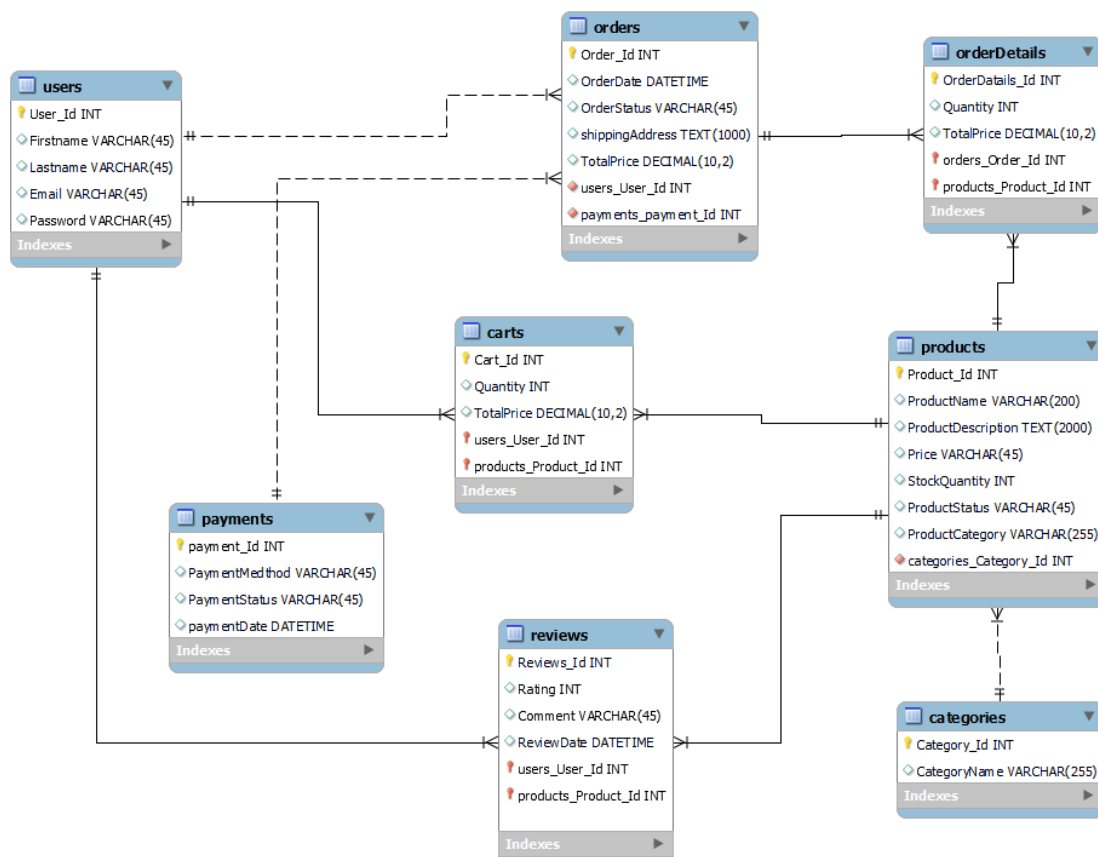


## 1.8 Cart

ในหน้าของตะกร้าสินค้าหลังจากที่มีผู้ใช้งานคนไหนกดดูใจสินค้า สินค้านั้นๆก็จะถูกปรากฏขึ้นภายในตะกร้าที่เราได้สร้างขึ้นไว้ให้ โดยเราจะแสดงข้อมูลรูปภาพสินค้า ชื่อสินค้า จำนวนสินค้าที่ผู้ใช้กด และราคารวมสินค้า โดยที่ภายในตะกร้าเราจะมีปุ่ม Remove คือผู้ใช้งานสามารถลบสินค้าออกได้หากไม่ต้องการที่จะเก็บไว้แล้ว



## 2. ER DIAGRAM



## ER Diagram นี้แสดงถึงโครงสร้างของฐานข้อมูลสำหรับระบบ E-commerce โดยประกอบด้วยตารางต่าง ๆ

### 1. Users (ผู้ใช้)

- User\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับผู้ใช้แต่ละคน
- Firstname (VARCHAR(45)): ชื่อของผู้ใช้
- Lastname (VARCHAR(45)): นามสกุลของผู้ใช้
- Email (VARCHAR(45)): อีเมลของผู้ใช้
- Password (VARCHAR(45)): รหัสผ่านของผู้ใช้

### 2. Orders (คำสั่งซื้อ)

- Order\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับคำสั่งซื้อแต่ละรายการ
- OrderDate (DATETIME): วันที่และเวลาที่สั่งซื้อ
- OrderStatus (VARCHAR(45)): สถานะของคำสั่งซื้อ
- shippingAddress (TEXT): ที่อยู่ในการจัดส่ง
- TotalPrice (DECIMAL(10,2)): ราคารวมของคำสั่งซื้อ
- payments\_payment\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Payments
- users\_User\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Users

### 3. OrderDetails (รายละเอียดคำสั่งซื้อ)

- OrderDetails\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับรายละเอียดคำสั่งซื้อแต่ละรายการ
- Quantity (INT): จำนวนสินค้าที่สั่งซื้อ
- TotalPrice (DECIMAL(10,2)): ราคารวมของสินค้านั้น ๆ
- orders\_Order\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Orders
- products\_Product\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Products

### 4. Carts (รถเข็นสินค้า)

- Cart\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับรถเข็นแต่ละรายการ
- Quantity (INT): จำนวนสินค้าที่อยู่ในรถเข็น
- TotalPrice (DECIMAL(10,2)): ราคารวมของสินค้าที่อยู่ในรถเข็น
- users\_User\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Users

- products\_Product\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Products

## 5. Payments (การชำระเงิน)

- payment\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับการชำระเงินแต่ละครั้ง
- PaymentMethod (VARCHAR(45)): วิธีการชำระเงิน
- PaymentStatus (VARCHAR(45)): สถานะของการชำระเงิน
- paymentDate (DATETIME): วันที่และเวลาที่ชำระเงิน

## 6. Products (สินค้า)

- Product\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับสินค้าแต่ละรายการ
- ProductName (VARCHAR(200)): ชื่อของสินค้า
- ProductDescription (TEXT): คำอธิบายของสินค้า
- Price (VARCHAR(45)): ราคาของสินค้า
- StockQuantity (INT): จำนวนสินค้าที่มีในสต็อก
- ProductStatus (VARCHAR(45)): สถานะของสินค้า
- ProductCategory (VARCHAR(255)): หมวดหมู่ของสินค้า
- categories\_Category\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Categories

## 7. Reviews (รีวิว)

- Reviews\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับรีวิวแต่ละรายการ
- Rating (INT): คะแนนของรีวิว
- Comment (VARCHAR(45)): ความคิดเห็นในรีวิว
- ReviewDate (DATETIME): วันที่และเวลาที่รีวิว
- users\_User\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Users
- products\_Product\_Id (INT): คีย์ภายนอก อ้างอิงไปยังตาราง Products

## 8. Categories (หมวดหมู่)

- Category\_Id (INT): คีย์หลัก เป็นตัวระบุที่ไม่ซ้ำกันสำหรับหมวดหมู่แต่ละรายการ
- CategoryName (VARCHAR(255)): ชื่อของหมวดหมู่

## ความสัมพันธ์ระหว่างตาราง

- Users และ Orders มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยผู้ใช้นี้หนึ่งคนสามารถมีหลายคำสั่งซื้อ
- Orders และ OrderDetails มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยคำสั่งซื้อหนึ่งรายการสามารถมีหลายรายการสินค้า
- Products และ OrderDetails มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยสินค้าหนึ่งรายการสามารถอยู่ในหลายคำสั่งซื้อ
- Users และ Carts มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยผู้ใช้นี้หนึ่งคนสามารถมีหลายรายการในรถเข็น
- Products และ Carts มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยสินค้าหนึ่งรายการสามารถอยู่ในหลายรถเข็น
- Users และ Reviews มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยผู้ใช้นี้หนึ่งคนสามารถเขียนหลายรีวิว
- Products และ Reviews มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยสินค้าหนึ่งรายการสามารถมีหลายรีวิว
- Categories และ Products มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many) โดยหมวดหมู่หนึ่งหมวดสามารถมีหลายสินค้า

### 3. SQL มีอะไรบ้าง ที่นำมาใช้ใน Web App

1. การลงทะเบียนผู้ใช้ (/register)
  - ใช้คำสั่ง INSERT INTO เพื่อเพิ่มข้อมูลผู้ใช้ใหม่ลงในตาราง users
  - ทำการแฮชผ่านก่อนบันทึกลงฐานข้อมูลเพื่อความปลอดภัย
2. การเข้าสู่ระบบ (/login)
  - ใช้คำสั่ง SELECT เพื่อดึงข้อมูลผู้ใช้จากตาราง users
  - ตรวจสอบความถูกต้องของรหัสผ่านโดยใช้ bcrypt.compare
3. การแก้ไขโปรไฟล์ (/profile/:id)
  - ใช้คำสั่ง UPDATE เพื่อปรับปรุงข้อมูลผู้ใช้ตาม User\_Id
  - แฮชรหัสผ่านใหม่ก่อนบันทึกลงฐานข้อมูล
4. การลบผู้ใช้ (/profile/:id)
  - ใช้คำสั่ง DELETE เพื่อลบข้อมูลผู้ใช้จากตาราง users ตาม User\_Id
5. การเพิ่มสินค้า (/products)
  - ใช้คำสั่ง INSERT INTO เพื่อเพิ่มข้อมูลสินค้าใหม่ลงในตาราง products
  - บันทึกชื่อไฟล์ภาพที่อัปโหลดลงในฐานข้อมูล
6. การดึงข้อมูลสินค้าทั้งหมด (/products)
  - ใช้คำสั่ง SELECT เพื่อดึงข้อมูลสินค้าทั้งหมดจากตาราง products
7. การเพิ่มสินค้าในตะกร้า (/cart)
  - ใช้คำสั่ง INSERT INTO พร้อมกับ ON DUPLICATE KEY UPDATE เพื่อเพิ่มหรืออัปเดตข้อมูลสินค้าที่อยู่ในตะกร้าของผู้ใช้
8. การดึงข้อมูลสินค้าในตะกร้า (/cart)
  - ใช้คำสั่ง SELECT พร้อมกับ JOIN เพื่อดึงข้อมูลสินค้าจากตาราง carts และ products โดยเชื่อมโยงข้อมูลตาม Product\_Id
9. การลบสินค้าในตะกร้า (/cart/:id)
  - ใช้คำสั่ง DELETE เพื่อลบข้อมูลสินค้าจากตะกร้าของผู้ใช้ตาม Product\_Id และ User\_Id

# สรุปรายงานปัญหาที่พบ

## ปัญหา:

1. การแบ่งงานไม่ชัดเจนเนื่องจากภาษา React เป็นภาษาที่ใหม่สำหรับกลุ่ม:
  - ทีมงานไม่คุ้นเคยกับภาษา React ทำให้ไม่สามารถแบ่งงานได้อย่างเหมาะสม
  - ความไม่แน่ใจในความสามารถของสมาชิกทีมแต่ละคนในการใช้งาน React
2. เกิดปัญหาเรื่องการทำงานภายในทีมเล็กน้อย:
  - ความไม่เข้าใจในเทคโนโลยีและวิธีการทำงานทำให้เกิดความสับสนและขัดแย้งภายในทีม
3. งานที่ดั่งสโคปไว้ตอนแรกค่อนข้างมีสเกลใหญ่เกินไป:
  - ฟังก์ชันที่กำหนดไว้ตอนแรกมีจำนวนมากเกินไปสำหรับเวลาที่มีอยู่

## วิธีแก้ไข:

1. การแบ่งงานไม่ชัดเจนเนื่องจากภาษา React เป็นภาษาที่ใหม่สำหรับกลุ่ม:
  - นัดสมาชิกทีมทุกคนมาทำงานร่วมกัน
  - ให้สมาชิกแต่ละคนเรียนรู้เพิ่มเติมเกี่ยวกับการเขียน React ด้วยตนเอง
  - นัดกันรวมกลุ่มและสอน React เบื้องต้นให้กับสมาชิกในทีม
  - หลังจากเรียนรู้เพิ่มเติมแล้ว แบ่งงานโดยการแบ่ง Components ต่าง ๆ ให้แต่ละคนรับผิดชอบ
2. เกิดปัญหาภายในทีมเล็กน้อย:
  - ใช้การทำงานร่วมกันผ่าน Live Share เพื่อให้ทุกคนสามารถช่วยกันแก้ไขปัญหาและทำงานได้พร้อมกัน
  - จัดการประชุมประจำเพื่อสื่อสารและแก้ไขปัญหาที่เกิดขึ้น
3. งานที่ดั่งสโคปไว้ตอนแรกค่อนข้างมีสเกลใหญ่เกินไป:
  - ทบทวนสโคปงานและฟังก์ชันที่ตั้งไว้ตอนแรก
  - ตัดฟังก์ชันที่ไม่จำเป็นออกเพื่อให้งานสามารถเสร็จได้ภายในเวลาที่กำหนด
  - มุ่งเน้นการพัฒนาฟังก์ชันที่จำเป็นและสำคัญในการออกแบบและดำเนินการต่อ