

```
1 import { Button, ScrollView, StyleSheet, Text, View } from 'react-native';
2 import Header from './Header';
3 import { useState, useEffect } from 'react';
4 const initItems = [
5   { id: '1', name: 'ข้าวผัด', price: '80' },
6   { id: '2', name: 'ผัดไทย', price: '60' },
7   { id: '3', name: 'ไก่', price: '45' },
8 ];
9
10 export default function FoodOrderApp() {
11   const [cart, setCart] = useState([]);
12   const [totalPrice, setTotalPrice] = useState(0);
13   const [totalPriceWithVat, setTotalPriceWithVat] = useState(0);
14
15   const addToCart = (item) => {
16     const newCart = [...cart];
17     const index = newCart.findIndex((cartItem) => cartItem.id === item.id);
18     if (index !== -1) {
19       newCart[index].quantity += 1;
20     } else {
21       newCart.push({ ...item, quantity: 1 });
22     }
23     setCart(newCart);
24   };
25
26   useEffect(() => {
27     const vatRate = 0.07;
28     const newTotalPrice = cart.reduce((total, item) => total + item.price * item.quantity, 0);
29     const vatAmount = newTotalPrice * vatRate;
30     const totalPriceWithVat = newTotalPrice + vatAmount;
31     setTotalPrice(newTotalPrice);
32     setTotalPriceWithVat(totalPriceWithVat);
33   }, [cart]);
34 }
```

สร้าง item ด้วยการใช้ objects array

สร้างstateที่เอาไว้เก็บ

- สินค้าใหม่ที่ถูกเลือก
- ราคา
- ราคาที่รวมvat

## useState

cart setCart ในการเก็บ itemหรือสินค้าที่เราเลือก

ใน ฟังก์ชัน addToCart รับพารามิเตอร์ item เป็นobject ที่แทนสินค้าที่ต้องการเพิ่มลงตะกร้า

เพื่อไม่ให้แก้ไข cart เดิม

เราสร้าง array ใหม่ชื่อ newCart ใช้(...) spread operator ในการคัดลอก elementมา

และ สร้าง index ทำการ findIndex หาตำแหน่ง (index) ของสินค้าใน newCart

และเปรียบเทียบ id ของใน (cartItem.id) กับ id ของใน (item.id)

และใน if else

ทำการเปรียบเทียบ(index !== -1) ถ้าindex ไม่เท่ากับ-1

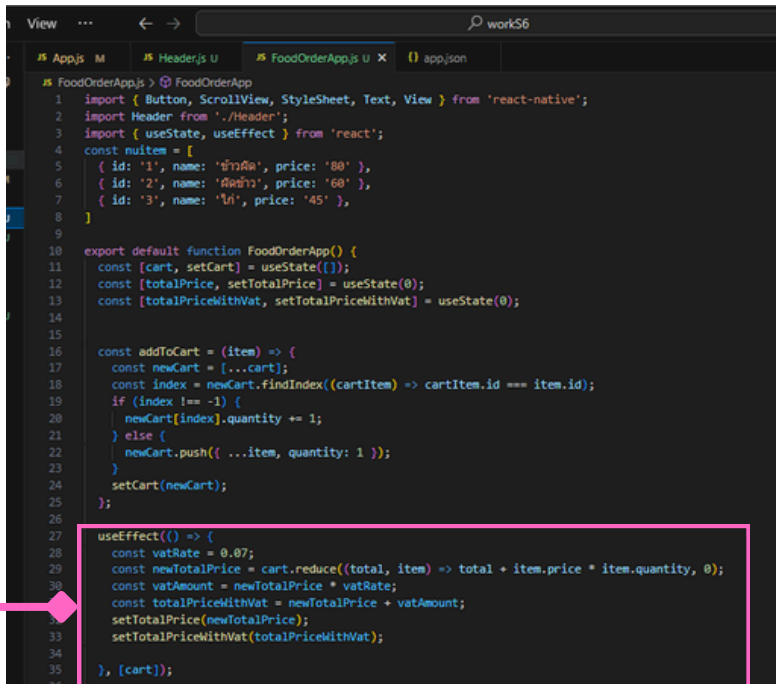
ก็ทำการเพิ่ม quantity(จำนวน) ของสินค้าที่มีอยู่ใน newCart +1

แต่ถ้าไม่เจอ ให้ทำการ push เพิ่มสินค้าใหม่เข้าไปใน newCart

spread operator(...item) คัดลอกทุก property ของ item  
เพิ่ม property quantity: 1 ให้สินค้าใหม่

และ setCart(newCart);

- อัปเดต state ของ cart ด้วย newCart ที่แก้ไขแล้ว



```

1 import { Button, ScrollView, StyleSheet, Text, View } from 'react-native';
2 import Header from './Header';
3 import { useState, useEffect } from 'react';
4 const menuItem = [
5   { id: '1', name: 'ข้าวผัด', price: '80' },
6   { id: '2', name: 'ผัดไทย', price: '60' },
7   { id: '3', name: 'ไก่', price: '45' },
8 ]
9
10 export default function FoodOrderApp() {
11   const [cart, setCart] = useState([]);
12   const [totalPrice, setTotalPrice] = useState(0);
13   const [totalPriceWithVat, setTotalPriceWithVat] = useState(0);
14
15   const addToCart = (item) => {
16     const newCart = [...cart];
17     const index = newCart.findIndex((cartItem) => cartItem.id === item.id);
18     if (index !== -1) {
19       newCart[index].quantity += 1;
20     } else {
21       newCart.push({ ...item, quantity: 1 });
22     }
23     setCart(newCart);
24   };
25
26   useEffect(() => {
27     const vatRate = 0.07;
28     const newTotalPrice = cart.reduce((total, item) => total + item.price * item.quantity, 0);
29     const vatAmount = newTotalPrice * vatRate;
30     const totalPriceWithVat = newTotalPrice + vatAmount;
31     setTotalPrice(newTotalPrice);
32     setTotalPriceWithVat(totalPriceWithVat);
33   }, [cart]);
34 }

```

useEffect จะทำงานเมื่อ component render ครั้งแรก และเมื่อค่าที่อยู่ใน dependency array ([]) มีการเปลี่ยนแปลง

ใช้ useEffect เพื่อคำนวณราคาสินค้า ทุกครั้งที่ข้อมูลใน cart มีการเปลี่ยนแปลง

กำหนด vatRate ไว้ 0.07

cart.reduce() reduce เป็นฟังก์ชันที่ใช้ในการรวมค่าต่างๆ ใน array ให้เหลือค่าเดียว

- total เป็นค่าที่เก็บระหว่างการวนลูป
- item คือวัตถุ (object) แต่ละชิ้นใน card

จะคำนวณ  $item.price * item.quantity$

คือ ราคาคูณกับจำนวนสินค้า หาผลค่าสินค้านั้น ๆ และนำไปรวมกับ total ราคาที่มีอยู่จากการคำนวณสินค้า

vatAmount คือเก็บ ราคาสินค้า\*ค่าvat

และ totalPriceWithVat คือ ราคาสินค้า+ค่าvatของสินค้า

เมื่อคำนวณราคาเสร็จ จะเรียก setTotalPrice เพื่ออัปเดตสถานะ (totalPrice) ใน component เมื่อมีการอัปเดตจะทำให้ component render ใหม่พร้อมกับราคารวมที่คำนวณได้

```
36 return (
37   <View style={styles.container}>
38     <Text style={styles.title}>เมนูอาหาร</Text>
39     <ScrollView horizontal={true}>
40       {items.map(item => (
41         <View key={item.id} style={styles.card}>
42           <Text style={{ fontSize: 18, color: 'white' }}>{item.name}</Text>
43           <Text style={{ fontSize: 16, color: 'white' }}>{item.price}</Text>
44           <Button title="เพิ่มสินค้า" color="#2ecc71" onPress={() => addToCart(item)} />
45         </View>
46       ))}
47     </ScrollView>
48   </View>
49 );
50
51
52
53
54
55 <View style={styles.containerCard2}>
56   <Text style={{ fontSize: 20, fontWeight: 'bold' }}>รายการสินค้า</Text>
57   <ScrollView horizontal={false}>
58     {cart.map(item => (
59       <View key={item.id} style={styles.cardSelect}>
60         <Text style={{ fontSize: 18, color: 'black' }}>{item.name}</Text>
61         <Text style={{ fontSize: 14, color: 'black' }}>{item.quantity} x {item.price} <Text>
62       </View>
63     ))}
64   </ScrollView>
65   <Text style={{ color: 'red' }}>รวม: {totalPrice}</Text>
66
67   <View style={styles.linnn}>
68     <Text style={{ fontSize: 14, }}>รวม:</Text>
69     <Text style={{ fontSize: 14, }}>{totalPrice} บาท</Text>
70   </View>
71
72   <View style={styles.linnn}>
73     <Text style={{ fontSize: 14, }}>รวม:</Text>
74     <Text style={{ fontSize: 14, }}>{totalPrice} บาท</Text>
75   </View>
76
77   <View style={styles.linnn}>
78     <Text style={{ fontSize: 18, }}>รวม:</Text>
79     <Text style={{ fontSize: 18, }}>{totalPriceWithWat} บาท</Text>
80   </View>
81 </View>
82 );
83 }
```

## เมนูอาหาร

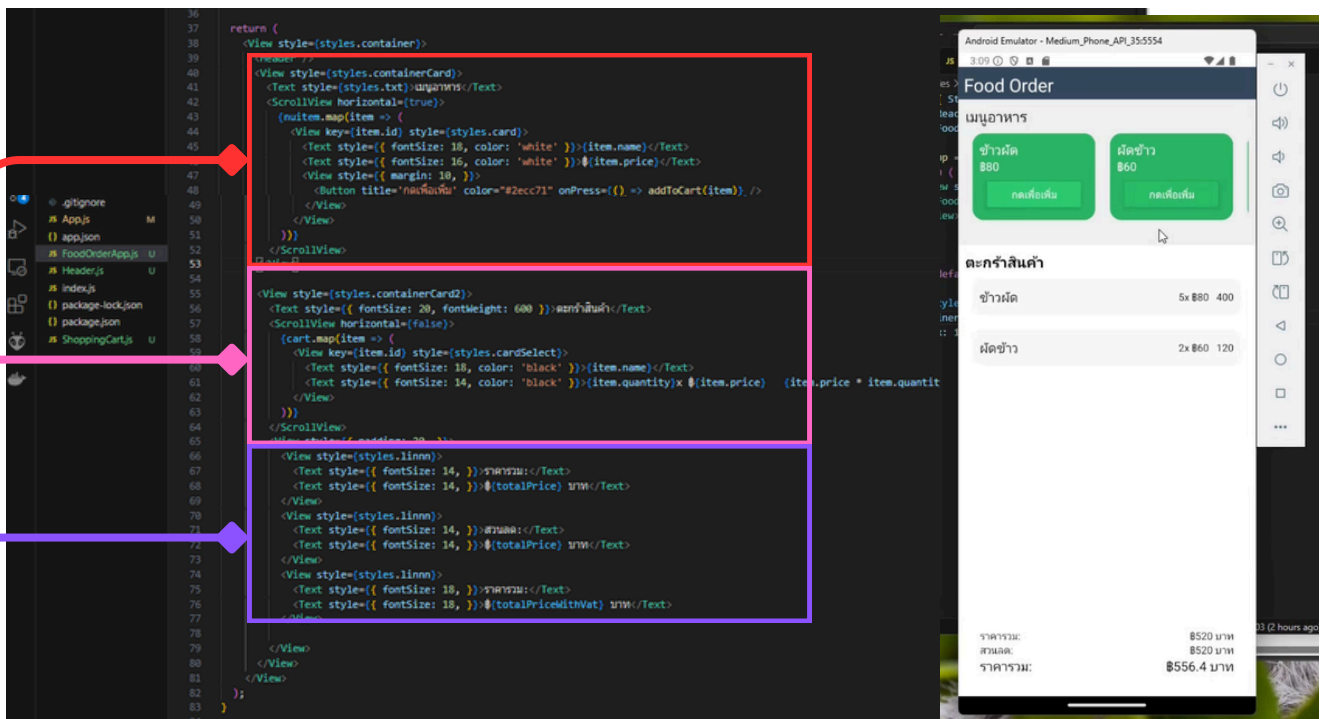
- ใช้ ScrollView แบบแนวนอน (horizontal={true}) เพื่อเลื่อนดูเมนูอาหาร
- แต่ละcard แสดงชื่อ (name) และราคา (price)
- ปุ่มเมื่อกดจะเรียกใช้ฟังก์ชัน addToCart(item) เพื่อเพิ่มสินค้าไปยังตะกร้า

## ตะกร้าสินค้า:

- ใช้ ScrollView แนวตั้ง (horizontal={false}) สำหรับแสดงรายการสินค้าในตะกร้า
- แต่ละรายการแสดงชื่อ จำนวน(quantity) ราคา(price) และราคารวมของสินค้านั้น
- 

## ราคารวมทั้งหมด:

- แสดงราคารวมทั้งหมดจากตัวแปร totalPrice



### เมนูสินค้า

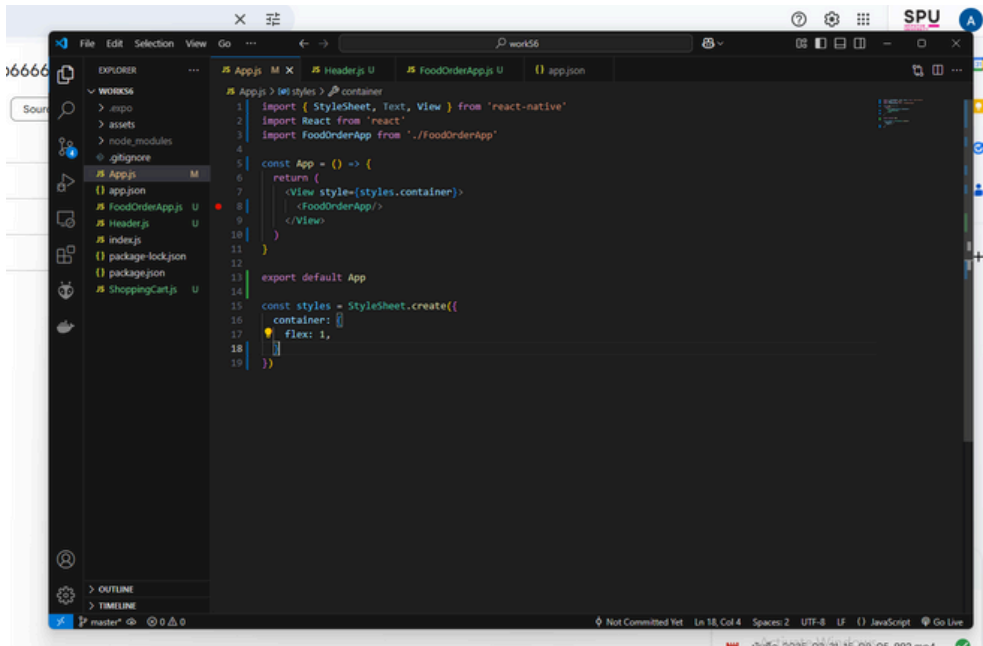
ใช้ .map เพื่อวนลูปรายการเมนูใน(nuitem) แสดงชื่อและราคาใน Text การทำงานของ.map

Method Map คือค่าใน object Array แต่ละตัวจะทำการนำไปผ่าน Function ที่เรากำหนดไว้และ return ค่าออกมา ในโค้ดนี้คือใช้เพื่อสร้าง Component ซ้ำๆ เช่น รายการสินค้า, จะสร้าง <View> สำหรับแต่ละสินค้าขึ้นมา และปุ่ม Button เรียกใช้ฟังก์ชัน addToCart(item) เมื่อกด เพื่อเพิ่มสินค้าในตะกร้า cart

### ตะกร้าสินค้า

วนลูปข้อมูลใน cart เพื่อแสดงรายการสินค้าที่ถูกเลือก ใช้.map เหมือนเมนูสินค้า จะแสดงชื่อสินค้า จำนวนสินค้า (quantity), ราคา และราคารวมของสินค้านั้น (item.price \* item.quantity)

### แสดง ราคารวมทั้งหมด ของสินค้าในตะกร้า



ส่วนของ หน้าApp  
และ  
ของ header

