



```

1 import { Button, ScrollView, StyleSheet, Text, View } from 'react-native';
2 import Header from './Header';
3 import { useState, useEffect } from 'react';
4 const menu = [
5   { id: '1', name: 'พิซซ่า', price: '80' },
6   { id: '2', name: 'โรตีไก่', price: '60' },
7   { id: '3', name: ' Uri', 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
16   const addToCart = (item) => {
17     const newCart = [...cart];
18     const index = newCart.findIndex(cartItem => cartItem.id === item.id);
19     if (index !== -1) {
20       newCart[index].quantity += 1;
21     } else {
22       newCart.push({ ...item, quantity: 1 });
23     }
24     setCart(newCart);
25   };
26
27   useEffect(() => {
28     const vatRate = 0.07;
29     const newTotalPrice = cart.reduce((total, item) => total + item.price * item.quantity, 0);
30     const vatAmount = newTotalPrice * vatRate;
31     const totalPriceWithVat = newTotalPrice + vatAmount;
32     setTotalPrice(newTotalPrice);
33     setTotalPriceWithVat(totalPriceWithVat);
34   }, [cart]);
35 }

```

สร้าง item ด้วยการใช้ objects array

สร้างstateที่เอ้าไว้เก็บ

- สินค้าใหม่ที่ถูกเลือก
- ราคา
- ราคาก่อรวมvat



**useState**

cart setCart ในการเก็บ item หรือสินค้าที่เราเลือก

ในฟังก์ชัน addToCart รับพารามิเตอร์ item เป็นobject ที่แทนสินค้าที่ต้องการเพิ่มลงตะกร้า

เพื่อไม่ให้แก้ไข cart เดิม

เราสร้าง array ใหม่ชื่อ newCart ใช้(...) spread operator ในการคัดลอก element มา

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) ของสินค้าที่เมื่อยู่ใน newCart +1

แต่ถ้าไม่เจอ ให้ทำการ push เพิ่มสินค้าใหม่เข้าไปใน newCart

spread operator(...item) คัดลอกทุก property ของ item

เพิ่ม property quantity: 1 ให้สินค้าใหม่

และ setCart(newCart);

- อัพเดต 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 menu = [
5   { id: '1', name: 'พิซซ่า', price: '80' },
6   { id: '2', name: 'โรตีไก่', price: '60' },
7   { id: '3', name: 'ไข่', 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
16   const addToCart = (item) => {
17     const newCart = [...cart];
18     const index = newCart.findIndex(cartItem => cartItem.id === item.id);
19     if (index !== -1) {
20       newCart[index].quantity += 1;
21     } else {
22       newCart.push({ ...item, quantity: 1 });
23     }
24     setCart(newCart);
25   };
26
27   useEffect(() => {
28     const vatRate = 0.07;
29     const newTotalPrice = cart.reduce((total, item) => total + item.price * item.quantity, 0);
30     const vatAmount = newTotalPrice * vatRate;
31     const totalPriceWithVat = newTotalPrice + vatAmount;
32     setTotalPrice(newTotalPrice);
33     setTotalPriceWithVat(totalPriceWithVat);
34   }, [cart]);
35
36 }

```

`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 หากมูลค่าสินค้าบันทึก และนำไปรวมกับ `total` ราคาก็มีอยู่จากการคำนวณสินค้า

`vatAmount` คือเก็บ ราคาสินค้า \* ค่า vat

และ `totalPriceWithVat` คือ ราคาสินค้า + ค่า vat ของสินค้า

เมื่อคำนวณราคาเสร็จ จะเรียก `setTotalPrice` เพื่ออัปเดตสถานะ (`totalPrice`) ใน component เมื่อมีการอัพเดตจะทำให้ component render ใหม่พร้อมกับราคารวมที่คำนวณได้

The screenshot shows a mobile application interface. On the left, there is a file navigation sidebar with files like App.js, Header.js, index.js, package-lock.json, and ShoppingCart.js. The main screen displays a food menu with items like "กุ้งเผา" and "ไข่เจียว". Below the menu is a shopping cart summary showing "จำนวนทั้งหมด 2 รายการ" and "ราคารวมทั้งหมด 20 บาท". The code in the screenshot is as follows:

```

36
37   return (
38     <View style={styles.container}>
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'}}>${item.price}</Text>
44             <View style={{margin: 10, }}>
45               <Button title="Add To Cart" color="#zecc71" onPress={() => addToCart(item)} />
46             </View>
47           </View>
48         )))
49       </ScrollView>
50     </View>
51   );
52 }
53
54
55 <View style={styles.containerCard2}>
56   <Text style={{fontSize: 20, fontWeight: 600}}>รวมทั้งหมด</Text>
57   <ScrollView horizontal={false}>
58     {cart.map(item => (
59       <View key={item.id} style={styles.cardSelect}>
60         <Text style={{fontSize: 18, color: 'black'}}>{item.name}</Text>
61         <Text style={{fontSize: 14, color: 'black'}}>${item.quantity} x ${item.price} = ${item.price * item.quantity}</Text>
62         </View>
63     )))
64   </ScrollView>
65   <View style={styles.linn}>
66     <Text style={{fontSize: 14, }}>รวมทั้งหมด</Text>
67     <Text style={{fontSize: 14, }}>${totalPrice}</Text>
68   </View>
69   <View style={styles.linn}>
70     <Text style={{fontSize: 14, }}>จำนวน</Text>
71     <Text style={{fontSize: 14, }}>${totalPrice}</Text>
72   </View>
73   <View style={styles.linn}>
74     <Text style={{fontSize: 18, }}>ราคารวม</Text>
75     <Text style={{fontSize: 18, }}>${totalPriceWithVat}</Text>
76   </View>
77 </View>
78 </View>
79 </View>
80 </View>
81   );
82 }
83 </View>

```

## เมนูอาหาร

- ใช้ ScrollView แบบแนวนอน (horizontal={true}) เพื่อเลื่อนดูเมนูอาหาร
- แต่ละ cart แสดงชื่อ (name) และราคา (price)
- บุํมเมือกดจะเรียกใช้ฟังก์ชัน addToCart(item) เพื่อเพิ่มสินค้าไปยังตะกร้า

## ตะกร้าสินค้า:

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

## ราคารวมทั้งหมด:

- แสดงราคารวมทั้งหมดจากตัวแปร totalPrice

```

36
37   return (
38     <View style={styles.container}>
39       <Header />
40       <View style={styles.containerCard}>
41         <Text style={styles.txt}>เมนูอาหาร</Text>
42         <ScrollView horizontal={true}>
43           {menu.map(item => (
44             <View key={item.id} style={styles.card}>
45               <Text style={{fontSize: 18, color: 'white'}}>{item.name}</Text>
46               <Text style={{fontSize: 16, color: 'white'}}>${item.price}</Text>
47               <View style={{margin: 10,}}>
48                 <Button title="เพิ่มสินค้า" color="#zecc71" onPress={() => addToCart(item)} />
49               </View>
50             </View>
51           ))
52         </ScrollView>
53       </View>
54     </View>
55   <View style={styles.containerCard2}>
56     <Text style={{fontSize: 20, fontWeight: 600}}>รายการสินค้า</Text>
57     <ScrollView horizontal={false}>
58       {cart.map(item => (
59         <View key={item.id} style={styles.cardSelect}>
60           <Text style={{fontSize: 18, color: 'black'}}>{item.name}</Text>
61           <Text style={{fontSize: 14, color: 'black'}}>${item.quantity} x ${item.price} = ${item.quantity * item.price}</Text>
62           <View style={styles.linn}>
63             <Text style={{fontSize: 14,}}>จำนวน:</Text>
64             <Text style={{fontSize: 14,}}>${item.quantity}</Text>
65           </View>
66           <View style={styles.linn}>
67             <Text style={{fontSize: 14,}}>ราคารวม:</Text>
68             <Text style={{fontSize: 14,}}>${item.totalPrice}</Text>
69           </View>
70           <View style={styles.linn}>
71             <Text style={{fontSize: 14,}}>จำนวนรวม:</Text>
72             <Text style={{fontSize: 14,}}>${totalPrice}</Text>
73           </View>
74           <View style={styles.linn}>
75             <Text style={{fontSize: 18,}}>ราคารวม:</Text>
76             <Text style={{fontSize: 18,}}>${totalPriceWithVat}</Text>
77           </View>
78         </View>
79       ))
80     </ScrollView>
81   </View>
82 </View>
83 )

```

## เมนูสินค้า

ใช้ .map เพื่อวนลูปรายการเมนูใน(guitem) และแสดงชื่อและราคาใน Text

การทำงานของ.map

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

## ตารางร้านค้า

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

แสดง ราคารวมกันหมด ของสินค้าในตะกร้า

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** File, Edit, Selection, View, Go, work56
- Sidebar (Left):**
  - EXPLORER: WORKS56, .expo, assets, node\_modules, .gitignore
  - Source Control: 6666
  - OUTPUT: (empty)
  - PROBLEMS: (empty)
  - RECENT FILES: App.js, Header.js, FoodOrderApp.js, app.json
- Code Editor (Center):** The file `App.js` is open, showing the following code:

```
1 import { StyleSheet, Text, View } from 'react-native';
2 import React from 'react'
3 import FoodOrderApp from './FoodOrderApp'
4
5 const App = () => {
6   return (
7     <View style={styles.container}>
8       <FoodOrderApp/>
9     </View>
10   )
11 }
12
13 export default App
14
15 const styles = StyleSheet.create({
16   container: {
17     flex: 1,
18   }
19 })
```
- Bottom Status Bar:** Not Committed Yet, Ln 18, Col 4, Spaces: 2, UTF-8, LF, JavaScript, Go Live

# ស៊ុនខែង ហ្មាត់App និង header



```
import { StyleSheet, Text, View } from 'react-native'
const Header = () => {
  return (
    <View style={styles.container}>
      <Text style={{color: 'white', fontSize: 25,}}>Food Order</Text>
    </View>
  )
}
export default Header

const styles = StyleSheet.create({
  container: {
    height: 45,
    backgroundColor: '#34495e',
    padding: 8,
  }
})
```