

ขั้นตอนการสร้าง GoDrink App ด้วย Next.js และ Google Sheets บน VS Code

เอกสารนี้จะแนะนำขั้นตอนการสร้างโปรเจกต์ GoDrink สำหรับใช้งานบนเครื่องของคุณ (Local Project) โดยใช้ Next.js, Tailwind CSS และดึงข้อมูลสินค้ามาจาก Google Sheets โดยตรง

สิ่งที่ต้องมีก่อนเริ่ม (Prerequisites)

1. **Node.js:** ตรวจสอบว่าคุณติดตั้ง Node.js เวอร์ชัน 18.17 หรือสูงกว่า
2. **Visual Studio Code (VS Code):** โปรแกรม Code Editor
3. **บัญชี Google:** สำหรับสร้าง Google Sheets

ขั้นตอนที่ 1: เตรียมข้อมูลใน Google Sheets

เราจะสร้างไฟล์ Google Sheets เพื่อใช้เป็นฐานข้อมูล (Database) จำลองสำหรับรายการเครื่องดื่มของเรา

1. สร้าง Google Sheet ใหม่:

- ไปที่ sheets.google.com และสร้างเอกสารใหม่ (Blank spreadsheet)
- ตั้งชื่อชีตว่า GoDrink-Products
- ในชีตแรก (ตั้งชื่อว่า Sheet1) ให้สร้างตารางข้อมูลตามนี้ และคัดลอกข้อมูลตัวอย่างไปวางได้เลย:

id	name	thaiName	description	thaiDescription	price	category	image
1	Signature Cold Brew	ซิกเนเจอร์ โคลด์ บรูว์	Our smooth and rich cold brew.	โคลด์ บรูว์ รสชาติเข้มข้น กลมกล่อม	120	Coffee	https://images.pexels.com/photos/312418/pexels-photo-312418.jpeg
2	Tropical Smoothie	ทรอปิคอล สมูทตี้	A refreshing blend of mango and passion fruit.	การผสมผสานที่สดชื่นของมะม่วงและเสาวรส	100	Smoothies	https://images.pexels.com/photos/1484516/pexels-photo-1484516.jpeg

3	Rose Milk Tea	ชานมกุหลาบ	Fragrant rose tea with creamy milk.	ชากุหลาบหอมกรุ่นกับนมรสนุ่ม	90	Tea	https://images.pexels.com/photos/3755710/pexels-photo-3755710.jpeg
4	Sparkling Lychee	ลิ้นจี่โซดา	Sweet lychee with a refreshing fizz.	ลิ้นจี่หวานชื่นใจพร้อมโซดาซ่า	85	Special	https://images.pexels.com/photos/1293268/pexels-photo-1293268.jpeg
5	Caramel Macchiato	คาราเมลมัคคิอาโต้	Rich espresso with vanilla syrup and caramel.	เอสเพรสโซเข้มข้นกับไซรัปวานิลลาและคาราเมล	110	Coffee	https://images.pexels.com/photos/2983101/pexels-photo-2983101.jpeg
6	Matcha Latte	มัทฉะลาเต้	Ceremonial grade matcha with steamed milk.	มัทฉะเกรดพิธีชงกับนมอุ่น	115	Tea	https://images.pexels.com/photos/461428/pexels-photo-461428.jpeg
7	Berry Blast Smoothie	เบอร์รี่บลาสต์สมูทตี้	A mix of strawberries, blueberries, and raspberries.	ส่วนผสมของสตรอเบอร์รี่บลูเบอร์รี่และราสเบอร์รี่	125	Smoothies	https://images.pexels.com/photos/109275/pexels-photo-109275.jpeg

							09275.jpeg
8	Butterfly Pea Limeade	อัญชัน มะนาว	A magical color-changing drink.	เครื่องดื่ม เปลี่ยนสี ได้สุด มหัศจรรย์	95	Special	https://images.pexels.com/photos/8140366/pexels-photo-8140366.jpeg

2. เปิดการเข้าถึง (Share):

- คลิกที่ปุ่ม **"Share"** (แชร์) มุมขวบน
- ในส่วน **"General access"** (การเข้าถึงทั่วไป) ให้เปลี่ยนจาก "Restricted" (จำกัด) เป็น **"Anyone with the link"** (ทุกคนที่มีลิงก์)
- ตั้งค่าสิทธิ์เป็น **"Viewer"** (ผู้มีสิทธิ์อ่าน)
- คลิก **"Done"** (เสร็จสิ้น)

3. คัดลอก Spreadsheet ID:

- ดูที่ URL ของเบราว์เซอร์ จะมีลักษณะดังนี้:
`https://docs.google.com/spreadsheets/d/SPREADSHEET_ID/edit#gid=0`
- คัดลอกค่า SPREADSHEET_ID เก็บไว้ (จะเป็นสตริงยาวๆ)

ขั้นตอนที่ 2: สร้างโปรเจกต์ Next.js

1. เปิด Terminal (หรือ Command Prompt) ขึ้นมา

2. ใช้คำสั่งเพื่อสร้างโปรเจกต์ Next.js ใหม่:

```
npx create-next-app@latest godrink-app
```

3. ตอบคำถามตามนี้:

- Would you like to use TypeScript? **No**
- Would you like to use ESLint? **Yes**
- Would you like to use Tailwind CSS? **Yes**
- Would you like to use 'src/' directory? **No**
- Would you like to use App Router? **Yes**
- Would you like to customize the default import alias? **No**

4. เข้าสู่โฟลเดอร์โปรเจกต์:

```
cd godrink-app
```

5. เปิดโปรเจกต์ใน VS Code:

```
code .
```

ขั้นตอนที่ 3: สร้างฟังก์ชันสำหรับดึงข้อมูล

เราจะใช้วิธี fetch API มาตรฐานเพื่อดึงข้อมูลจาก Google Sheets โดยตรง ไม่ต้องติดตั้ง Library เพิ่มเติม

1. สร้างไฟล์สำหรับดึงข้อมูล:

- สร้างโฟลเดอร์ใหม่ชื่อ lib ใน root ของโปรเจกต์
- ในโฟลเดอร์ lib สร้างไฟล์ใหม่ชื่อ sheets.js
- ใส่โค้ดนี้ลงใน lib/sheets.js:

```
export async function getProducts() {
  try {
    const SPREADSHEET_ID = 'YOUR_SPREADSHEET_ID'; // <--- วาง SPREADSHEET_ID
    ของคุณที่นี่
    const SHEET_NAME = 'Sheet1';
    const URL =
    `https://docs.google.com/spreadsheets/d/${SPREADSHEET_ID}/gviz/tq?tqx=out:json&s
    heet=${SHEET_NAME}`;

    const res = await fetch(URL);
    if (!res.ok) {
      throw new Error(`Failed to fetch sheet data: ${res.statusText}`);
    }

    // Google Sheets API คืนค่ามาเป็น Text ที่ต้องตัดส่วนที่ไม่ใช่ JSON ออก
    let text = await res.text();
    const jsonString = text.match(/google\.visualization\.Query\.setResponse\(((.*)\))/s)[1];
    const data = JSON.parse(jsonString);

    // แปลงข้อมูลจากโครงสร้างของ Google Sheets ให้เป็น Array ของ Object ที่เราใช้งานได้
    const headers = data.table.cols.map(col => col.label);
    const rows = data.table.rows;

    const products = rows.map(row => {
      const product = {};
      headers.forEach((header, index) => {
        const cell = row.c[index];
        product[header] = cell ? cell.v : null;
      });
      return product;
    });

    // แปลง price ให้เป็นตัวเลข
    const formattedProducts = products.map(product => ({
```

```
...product,  
price: Number(product.price) || 0,  
});
```

```
return formattedProducts;
```

```
} catch (error) {  
  console.error('Error fetching from Google Sheets:', error);  
  return []; // คืนค่าเป็น array ว่างถ้าเกิดข้อผิดพลาด  
}  
}
```

สำคัญ: อย่าลืมเปลี่ยน YOUR_SPREADSHEET_ID เป็น ID ที่คุณคัดลอกมาจากขั้นตอนที่ 1

ขั้นตอนที่ 4: นำโค้ด React มาปรับใช้และแสดงผล

ตอนนี้เราจะนำโค้ด React จากที่คุณมีอยู่ มาใส่ในโปรเจกต์ Next.js ของเรา

1. ล้างไฟล์ app/page.js:

เปิดไฟล์ app/page.js และลบโค้ดทั้งหมดออก แล้วแทนที่ด้วยโค้ดนี้:

```
import { getProducts } from '../lib/sheets';  
import ClientPage from './ClientPage'; // เราจะสร้างไฟล์นี้ต่อไป
```

```
export default async function HomePage() {  
  // Fetch data on the server  
  const products = await getProducts();  
  
  // Pass data to a Client Component  
  return <ClientPage initialProducts={products} />;  
}
```

Next.js 13+ (App Router) จะทำการดึงข้อมูลบนฝั่ง Server ก่อน ซึ่งดีต่อ SEO และประสิทธิภาพ

2. สร้าง Client Component:

เนื่องจากหน้าเว็บของเราจะมีการโต้ตอบกับผู้ใช้ (เช่น การกดปุ่ม, การจัดการ state) เราต้องสร้างเป็น Client Component

- สร้างไฟล์ใหม่ในโฟลเดอร์ app ชื่อ ClientPage.js
- คัดลอกโค้ด React เดิม (จาก immersive godrink-react-app) มาวางในไฟล์นี้
- แก้ไขส่วนบนสุดของไฟล์ **ClientPage.js** ดังนี้:

```
'use client'; // บรรทัดนี้สำคัญมาก! เพื่อบอก Next.js ว่าเป็น Client  
Component
```

```
import React, { useState, useMemo } from 'react';
```

```
// --- SVG Icons ---

// Using inline SVGs for portability. These could be replaced with
// an icon library like Lucide-React.

const HomeIcon = ({ className }) => <svg
  xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor"
  strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
  className={className}><path d="m3 9 9-7 9 7v11a2 2 0 0 1-2 2H5a2 2
  0 0 1-2-2z"></path><polyline points="9 22 9 12 15 12 15
  22"></polyline></svg>;

const CoffeeIcon = ({ className }) => <svg
  xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor"
  strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
  className={className}><path d="M10 2v2a4 4 0 0 1-4
  4H2"></path><path d="M10 2v6a4 4 0 0 0 4 4h8"></path><path d="M12
  12h8a4 4 0 0 1 4 4v2a2 2 0 0 1-2 2H2v-2a4 4 0 0 1
  4-4h4Z"></path></svg>;

const ShoppingCartIcon = ({ className }) => <svg
  xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor"
  strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
  className={className}><circle cx="8" cy="21" r="1"></circle><circle
  cx="19" cy="21" r="1"></circle><path d="M2.05 2.05h21.66 12.42a2 2
  0 0 0 2 1.58h9.78a2 2 0 0 0
  1.95-1.57l11.65-7.43H5.16"></path></svg>;

const UserIcon = ({ className }) => <svg
  xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor"
  strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
  className={className}><path d="M19 21v-2a4 4 0 0 0-4-4H9a4 4 0 0
  0-4 4v2"></path><circle cx="12" cy="7" r="4"></circle></svg>;

const SearchIcon = ({ className }) => <svg
```

```

xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor"
strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
className={className}><circle cx="11" cy="11" r="8"></circle><line
x1="21" y1="21" x2="16.65" y2="16.65"></line></svg>;
const PlusCircleIcon = ({ className }) => <svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor"
strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
className={className}><circle cx="12" cy="12" r="10"></circle><line
x1="12" y1="8" x2="12" y2="16"></line><line x1="8" y1="12" x2="16"
y2="12"></line></svg>;
const MinusCircleIcon = ({ className }) => <svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor"
strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
className={className}><circle cx="12" cy="12" r="10"></circle><line
x1="8" y1="12" x2="16" y2="12"></line></svg>;
const ChevronLeftIcon = ({ className }) => <svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor"
strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
className={className}><polyline points="15 18 9 12 15
6"></polyline></svg>;
const SpecialDrinkIcon = ({ className }) => <svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor"
strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
className={className}><path d="m12 2 3.09 6.26L22 9.27l-5 4.87 1.18
6.88L12 17.77l-6.18 3.25L7 14.14 2 9.27l6.91-1.01L12
2z"></path></svg>;
const TeaIcon = ({ className }) => <svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor"

```

```

strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
className={className}><path d="M12 2a4 4 0 0 0-4 4v4a4 4 0 0 0 4
4h4a4 4 0 0 0 4-4V6a4 4 0 0 0-4-4h-4z"></path><path d="M20 14h-2a2
2 0 0 1-2-2V8"></path><path d="M12 14v8"></path><path d="M8
18h8"></path></svg>;

const SmoothieIcon = ({ className }) => <svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor"
strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"
className={className}><path d="M12 22a4 4 0 0 0 4-4H8a4 4 0 0 0 4
4z"></path><path d="M14 18h-4a2 2 0 0 1-2-2V7a2 2 0 0 1 2-2h4a2 2 0
0 1 2 2v9a2 2 0 0 1-2 2z"></path><path d="M18 10h2a1 1 0 0 1 1
1v2a1 1 0 0 1-1 1h-2"></path><path d="M6 5 8 2"></path></svg>;

const categories = [
  { name: 'Coffee', thaiName: 'กาแฟ', icon: CoffeeIcon, color:
'bg-pink-100' },
  { name: 'Tea', thaiName: 'ชา', icon: TeaIcon, color:
'bg-sky-100' },
  { name: 'Smoothies', thaiName: 'สมูทตี้', icon: SmoothieIcon,
color: 'bg-green-100' },
  { name: 'Special', thaiName: 'เมนูพิเศษ', icon: SpecialDrinkIcon,
color: 'bg-yellow-100' },
];

// --- Components ---

const Header = () => (
  <header className="sticky top-0 z-20 bg-gradient-to-r
from-purple-500 to-purple-600 text-white p-4 flex justify-between
items-center shadow-lg">
    <h1 className="text-2xl font-bold
tracking-wider">GoDrink</h1>
    <div className="flex items-center space-x-4">

```



```

        <SearchIcon className="h-6 w-6 cursor-pointer
hover:opacity-80 transition-opacity" />
        <UserIcon className="h-6 w-6 cursor-pointer
hover:opacity-80 transition-opacity" />
    </div>
</header>
);

const WelcomeBanner = () => (
    <div className="p-6 sm:p-8 bg-gradient-to-r from-purple-500
to-purple-600 text-white rounded-xl mb-8 shadow-lg">
        <h2 className="text-3xl font-bold mb-2">ยินดีต้อนรับ!</h2>
        <p className="text-lg opacity-90">ส่งเครื่องดื่มแก้วโปรดของคุณได้ง่ายๆ
ที่นี่</p>
    </div>
);

const CategoryNav = ({ onSelectCategory }) => (
    <div className="mb-8">
        <h3 className="text-2xl font-bold text-gray-800 mb-4">สำรวจ
หมวดหมู่</h3>
        <div className="grid grid-cols-2 sm:grid-cols-4 gap-4">
            {categories.map((cat) => (
                <div key={cat.name} onClick={() =>
onSelectCategory(cat.name)} className={` ${cat.color} p-4 rounded-xl
text-center shadow-md hover:shadow-lg hover:-translate-y-1
transition-all duration-300 cursor-pointer`} >
                    <cat.icon className="h-8 w-8 mx-auto mb-2
text-purple-600" />
                    <span className="font-semibold
text-gray-700">{cat.thaiName}</span>
                </div>
            ))}
        </div>
    </div>
);

```

```

    </div>
  );

  const ProductCard = ({ product, onAddToCart }) => (
    <div className="bg-white rounded-xl shadow-md overflow-hidden
transform hover:-translate-y-2 transition-transform duration-300
ease-in-out group">
      <img src={product.image} alt={product.name}
className="w-full h-48 object-cover" onError={(e) => {
e.target.onerror = null; e.target.src =
'https://placeholder.co/400x300/e2e8f0/64748b?text=Image+Not+Found';
}} />
      <div className="p-4 flex flex-col h-full">
        <h4 className="text-lg font-bold
text-gray-800">{product.name}</h4>
        <p className="text-sm text-gray-500
mb-3">{product.thaiName}</p>
        <p className="text-gray-600 text-sm
flex-grow">{product.description}</p>
        <div className="mt-4 flex justify-between items-center">
          <span className="text-xl font-extrabold text-purple-
600">฿{product.price}</span>
          <button onClick={() => onAddToCart(product)}
className="bg-purple-500 text-white rounded-full p-2
group-hover:bg-purple-600 transition-colors">
            <PlusCircleIcon className="w-6 h-6" />
          </button>
        </div>
      </div>
    </div>
  );

  const BottomNav = ({ activeView, setActiveView }) => {
    const navItems = [

```

```

    { name: 'Home', thaiName: 'หน้าหลัก', icon: HomeIcon, view:
'home' },
    { name: 'Menu', thaiName: 'เมนู', icon: CoffeeIcon, view:
'menu' },
    { name: 'Cart', thaiName: 'ตะกร้า', icon: ShoppingCartIcon,
view: 'cart' },
    { name: 'Profile', thaiName: 'โปรไฟล์', icon: UserIcon, view:
'profile' },
  ];
  return (
    <nav className="fixed bottom-0 left-0 w-full bg-white/80
backdrop-blur-sm shadow-[0_-2px_10px_rgba(0,0,0,0.05)] md:hidden
z-20">
      <div className="flex justify-around items-center h-16">
        {navItems.map(item => (
          <button key={item.name} onClick={() =>
setActiveView(item.view)} className={`flex flex-col items-center
justify-center space-y-1 transition-colors ${activeView ===
item.view ? 'text-purple-600' : 'text-gray-500'}`}>
            <item.icon className="w-6 h-6" />
            <span className="text-xs
font-medium">{item.thaiName}</span>
          </button>
        ))}
      </div>
    </nav>
  );
};

const CartView = ({ cart, onUpdateCart, onBack }) => {
  const total = useMemo(() => cart.reduce((sum, item) => sum +
item.price * item.quantity, 0), [cart]);

  const handleQuantityChange = (product, change) => {

```

```

    const existingItem = cart.find(item => item.id ===
product.id);
    if (existingItem) {
        const newQuantity = existingItem.quantity + change;
        if (newQuantity > 0) {
            onUpdateCart(product, newQuantity);
        } else {
            onUpdateCart(product, 0); // This will remove the
item
        }
    }
};

return (
    <div className="p-4 bg-gray-50 min-h-screen">
        <div className="flex items-center mb-6">
            <button onClick={onBack} className="p-2 mr-2
rounded-full hover:bg-gray-200">
                <ChevronLeftIcon className="w-6 h-6
text-gray-700"/>
            </button>
            <h2 className="text-3xl font-bold text-gray-800">
ตะกร้าของคุณ</h2>
        </div>

        {cart.length === 0 ? (
            <div className="text-center py-20">
                <ShoppingCartIcon className="w-16 h-16 mx-auto
text-gray-300 mb-4"/>
                <p className="text-gray-500 text-lg">ตะกร้าของคุณ
ว่างเปล่า</p>
                <button onClick={onBack} className="mt-6
bg-purple-500 text-white font-semibold py-2 px-6 rounded-lg
shadow-md hover:bg-purple-600 transition-colors">

```

เลือกซื้อสินค้าต่อ

```
        </button>
      </div>
    ) : (
      <div className="space-y-4">
        {cart.map(item => (
          <div key={item.id} className="bg-white p-4
rounded-xl shadow-sm flex items-center">
            <img src={item.image} alt={item.name}
className="w-20 h-20 rounded-lg object-cover mr-4" />
            <div className="flex-grow">
              <h3 className="font-bold
text-gray-800">{item.name}</h3>
              <p className="text-purple-600 font-
semibold">฿{item.price}</p>
            </div>
            <div className="flex items-center
space-x-3">
              <button onClick={() =>
handleQuantityChange(item, -1)} className="text-purple-500
rounded-full hover:bg-purple-100 p-1">
                <MinusCircleIcon className="w-6
h-6" />
              </button>
              <span className="font-bold text-lg
w-8 text-center">{item.quantity}</span>
              <button onClick={() =>
handleQuantityChange(item, 1)} className="text-purple-500
rounded-full hover:bg-purple-100 p-1">
                <PlusCircleIcon className="w-6
h-6" />
              </button>
            </div>
          </div>
        )
      </div>
    )
  </div>
```

```

    )))

    <div className="mt-8 pt-6 border-t-2
border-gray-200 border-dashed">
      <h3 className="text-xl font-bold mb-4">สรุป
รายการสั่งซื้อ</h3>

      <div className="space-y-2 text-gray-700">
        <div className="flex justify-between">
          <span>ราคาสินค้า</span>
          <span>฿{total.toFixed(2)}</span>
        </div>
        <div className="flex justify-between">
          <span>ค่าจัดส่ง</span>
          <span>ฟรี</span>
        </div>
        <div className="flex justify-between
font-bold text-xl text-gray-800 pt-2 border-t mt-2">
          <span>ยอดรวมทั้งหมด</span>
          <span>฿{total.toFixed(2)}</span>
        </div>
      </div>

      <button className="w-full mt-6 bg-purple-600
text-white font-bold py-3 rounded-xl shadow-lg hover:bg-purple-700
transition-colors text-lg">
        ชำระเงิน
      </button>
    </div>
  </div>
  ))
</div>
);
};

// --- Main App Component ---

```

```

export default function ClientPage({ initialProducts }) {
  const [activeView, setActiveView] = useState('home');
  const [cart, setCart] = useState([]);
  const [selectedCategory, setSelectedCategory] = useState(null);

  // Use initialProducts received from the server
  const products = initialProducts;

  const handleAddToCart = (productToAdd) => {
    setCart(prevCart => {
      const existingItem = prevCart.find(item => item.id ===
productToAdd.id);
      if (existingItem) {
        return prevCart.map(item =>
          item.id === productToAdd.id ? { ...item,
quantity: item.quantity + 1 } : item
        );
      }
      return [...prevCart, { ...productToAdd, quantity: 1 }];
    });
  };

  const handleUpdateCart = (productToUpdate, newQuantity) => {
    if (newQuantity <= 0) {
      setCart(prevCart => prevCart.filter(item => item.id !==
productToUpdate.id));
    } else {
      setCart(prevCart => prevCart.map(item =>
        item.id === productToUpdate.id ? { ...item,
quantity: newQuantity } : item
      ));
    }
  };
};

```

```

const filteredProducts = useMemo(() => {
  if (!products) return []; // Add a guard clause for when
products are not yet loaded
  if (!selectedCategory) return products;
  return products.filter(p => p.category ===
selectedCategory);
}, [selectedCategory, products]);

const handleSelectCategory = (categoryName) => {
  setSelectedCategory(categoryName);
  setActiveView('menu'); // Switch to menu view when a
category is selected
}

const renderView = () => {
  switch (activeView) {
    case 'cart':
      return <CartView cart={cart}
onUpdateCart={handleUpdateCart} onBack={() =>
setActiveView('home')} />;
    case 'menu':
    case 'home':
    default:
      return (
        <div className="p-4 sm:p-6">
          <WelcomeBanner />
          <CategoryNav
onSelectCategory={handleSelectCategory} />
          <div>
            <div className="flex justify-between
items-center mb-4">
              <h3 className="text-2xl font-bold
text-gray-800">{selectedCategory ? `${selectedCategory} Drinks` : '
เมนูแนะนำ'}</h3>

```



```

                {selectedCategory && (
                    <button onClick={() =>
setSelectedCategory(null)} className="text-sm font-semibold
text-purple-600 hover:underline">
                        แสดงทั้งหมด
                    </button>
                )}
            </div>
            <div className="grid grid-cols-1
sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-6">
                {filteredProducts.map(product => (
                    <ProductCard key={product.id}
product={product} onAddToCart={handleAddToCart} />
                ))}
            </div>
        </div>
    </div>
    </div>
    );
}
};

return (
    <div className="bg-gray-50 min-h-screen font-sans pb-20
md:pb-0">
        <Header />
        <main>
            {renderView()}
        </main>
        <BottomNav activeView={activeView}
setActiveView={setActiveView} />
    </div>
    );
}

```

สรุปการแก้ไขใน ClientPage.js:

- เพิ่ม 'use client'; ที่บรรทัดแรกสุด
 - ลบ const products ที่เป็น mock data ออก
 - เปลี่ยนชื่อ Component หลักจาก App เป็น ClientPage และรับ initialProducts ผ่าน props
 - กำหนด const products = initialProducts; เพื่อให้โค้ดส่วนที่เหลือทำงานได้
 - ใน useMemo ของ filteredProducts ให้เพิ่ม products เข้าไปใน dependency array
3. ล้างไฟล์ app/globals.css:
เปิดไฟล์ app/globals.css และลบโค้ด CSS ที่มีอยู่ทั้งหมดออก เหลือไว้แค่ 3 บรรทัดนี้:
@tailwind base;
@tailwind components;
@tailwind utilities;

ขั้นตอนที่ 5: รันโปรเจกต์

1. กลับไปที่ Terminal ใน VS Code
2. รันคำสั่งเพื่อเริ่ม Development Server:
npm run dev
3. เปิดเบราว์เซอร์แล้วไปที่: <http://localhost:3000>