

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

เอกสารนี้จะแนะนำขั้นตอนการสร้างโปรเจกต์ GoDrink สำหรับใช้งานบนเครื่อง (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 และ Google Cloud Project

## ขั้นตอนที่ 1: เตรียมข้อมูลใน Google Sheets

เราจะสร้างไฟล์ Google Sheets เพื่อใช้เป็นฐานข้อมูล (Database) จำลองสำหรับรายการเครื่องดื่มของเรา

- 1. **สร้าง Google Sheet ใหม่:**
  - ไปที่ [sheets.google.com](https://sheets.google.com) และสร้างเอกสารใหม่ (Blank spreadsheet)
  - ตั้งชื่อชีตว่า GoDrink-Products
  - ในชีตแรก (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    | <a href="https://images.pexels.com/photos/312418/pexels-photo-312418.jpeg">https://images.pexels.com/photos/312418/pexels-photo-312418.jpeg</a>     |
| 2  | Tropical Smoothie   | ทรอปิคอล สมูทตี้       | A refreshing blend of mango and passion fruit. | การผสมผสานที่สดชื่นของมะม่วงและเสาวรส | 100   | Smoothies | <a href="https://images.pexels.com/photos/1484516/pexels-photo-1484516.jpeg">https://images.pexels.com/photos/1484516/pexels-photo-1484516.jpeg</a> |
| 3  | Rose                | ชานม                   | Fragrant                                       | ชากุหลาบ                              | 90    | Tea       | <a href="https://i">https://i</a>   |

|   |                      |                       |  |   |     |           |   |
|---|----------------------|-----------------------|--|---|-----|-----------|---|
|   | Milk Tea             | กุหลาบ                | rose tea with creamy milk.                           | หอมกรุ่นกับนมรสนุ่ม                               |     |           | <a href="https://images.pexels.com/photos/3755710/pexels-photo-3755710.jpeg">images.pexels.com/photos/3755710/pexels-photo-3755710.jpeg</a>         |
| 4 | Sparkling Lychee     | ลิ้นจี่โซดา           | Sweet lychee with a refreshing fizz.                 | ลิ้นจี่หวานชื่นใจพร้อมโซดาซ่า                     | 85  | Special   | <a href="https://images.pexels.com/photos/1293268/pexels-photo-1293268.jpeg">https://images.pexels.com/photos/1293268/pexels-photo-1293268.jpeg</a> |
| 5 | Caramel Macchiato    | คาราเมลมัคคิอาโต้     | Rich espresso with vanilla syrup and caramel.        | เอสเพรสโซเข้มข้นกับไซร์วานิลลาและคาราเมล          | 110 | Coffee    | <a href="https://images.pexels.com/photos/2983101/pexels-photo-2983101.jpeg">https://images.pexels.com/photos/2983101/pexels-photo-2983101.jpeg</a> |
| 6 | Matcha Latte         | มัทฉะลาเต้            | Ceremonial grade matcha with steamed milk.           | มัทฉะเกรดพิธีชงกับนมอุ่น                          | 115 | Tea       | <a href="https://images.pexels.com/photos/461428/pexels-photo-461428.jpeg">https://images.pexels.com/photos/461428/pexels-photo-461428.jpeg</a>     |
| 7 | Berry Blast Smoothie | เบอร์รี่บลาสต์สมูทตี้ | A mix of strawberries, blueberries, and raspberries. | ส่วนผสมของสตรอว์เบอร์รี่บลูเบอร์รี่และราสเบอร์รี่ | 125 | Smoothies | <a href="https://images.pexels.com/photos/109275/pexels-photo-109275.jpeg">https://images.pexels.com/photos/109275/pexels-photo-109275.jpeg</a>     |

|   |                             |                 |  |   |    |         |   |
|---|-----------------------------|-----------------|--|---|----|---------|---|
|   |                             |                 |  |   |    |         | <a href="#">eg</a>  |
| 8 | Butterfly<br>Pea<br>Limeade | อัญชัน<br>มะนาว | A<br>magical<br>color-changing<br>drink. | เครื่องดื่ม<br>เปลี่ยนสี<br>ได้สุด<br>มหัศจรรย์ | 95 | Special | <a href="https://images.pexels.com/photos/8140366/pexels-photo-8140366.jpeg">https://images.pexels.com/photos/8140366/pexels-photo-8140366.jpeg</a> |

## 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`

ตัวอย่างของอาจารย์

`https://docs.google.com/document/d/1fmlOdyFmooMNSgAJmutnkOB_gYAkRjeM1q69wwVGX8o/edit?usp=sharing`

- 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** (เลือก Yes หากถนัด)
- 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: ติดตั้ง Library และดึงข้อมูล

เราจะใช้ papr-js ซึ่งเป็น library ที่ใช้ง่ายสำหรับอ่านข้อมูลจาก Google Sheets

#### 1. ติดตั้ง papr-js:

```
npm install papr-js
```

#### 2. สร้างไฟล์สำหรับดึงข้อมูล:

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

```
import papr from 'papr-js';
```

```
export async function getProducts() {  
  try {  
    const SPREADSHEET_ID = 'YOUR_SPREADSHEET_ID'; // <--- วาง SPREADSHEET_ID  
    ของคุณที่นี่
```

```
    // ดึงข้อมูลจากชีตสาธารณะ
```

```
    const productsData = await papr.get(SPREADSHEET_ID);
```

```
    // papr-js จะคืนค่าเป็น array ของ object โดยอัตโนมัติ
```

```
    // โดยใช้แถวแรกเป็น key ของ object
```

```
    // เราต้องแปลง price จาก string เป็น number
```

```
    const formattedProducts = productsData.map(product => ({  
      ...product,  
      price: Number(product.price) || 0, // แปลงเป็นตัวเลข, ถ้าผิดพลาดให้เป็น 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:

เนื่องจากหน้าเว็บของเรามีการโต้ตอบกับผู้ใช้ (เช่น การกดปุ่ม, การจัดการ state) เราต้องสร้างเป็น Client Component

- สร้างไฟล์ใหม่ในโฟลเดอร์ app ชื่อ ClientPage.js
- คัดลอกโค้ดทั้งหมดจาก godrink-react-app ที่คุณมีอยู่ มาวางในไฟล์นี้
- แก้ไขส่วนบนสุดของไฟล์ **ClientPage.js** ดังนี้:

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

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

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

```
// (โค้ด SVG ทั้งหมดของคุณอยู่ที่นี้... ไม่ต้องเปลี่ยนแปลง)
```

```
const HomeIcon = ({ className }) => { /* ... */ };
```

```
// ... ไอคอนอื่นๆทั้งหมด
```

```
// --- Mock Data ---
```

```
// ลบ const products และ const categories ออกไปได้เลย
```

```
// เพราะเราจะรับข้อมูลผ่าน props แทน
```

```
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 ---
// (โค้ด Component ทั้งหมดของคุณอยู่ที่นี้... ไม่ต้องเปลี่ยนแปลง)
const Header = () => { /* ... */ };
// ... Component อื่นๆ ทั้งหมด

// --- Main App Component ---
// แก้ไข Component หลัก (ที่เคยชื่อ App) เป็น ClientPage
export default function ClientPage({ initialProducts }) {
  const [activeView, setActiveView] = useState('home');
  const [cart, setCart] = useState([]);
  const [selectedCategory, setSelectedCategory] = useState(null);

  // ใช้ initialProducts ที่ได้รับมาจาก Server
  const products = initialProducts;

  const handleAddToCart = (productToAdd) => {
    // (โค้ดส่วนนี้เหมือนเดิม)
  };

  const handleUpdateCart = (productToUpdate, newQuantity) => {
    // (โค้ดส่วนนี้เหมือนเดิม)
  };

  const filteredProducts = useMemo(() => {
    if (!selectedCategory) return products;
    return products.filter(p => p.category === selectedCategory);
  }, [selectedCategory, products]); // เพิ่ม products เข้าไปใน dependency array

  const handleSelectCategory = (categoryName) => {
    // (โค้ดส่วนนี้เหมือนเดิม)
  }

  const renderView = () => {
    // (โค้ดส่วนนี้เหมือนเดิม)
  };

  return (
    // (โค้ด JSX ที่ return เหมือนเดิม)
  );
}

```

### 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. สร้างไฟล์ 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>

ตอนนี้คุณควรจะได้เห็นเว็บแอปพลิเคชัน GoDrink ของคุณทำงานบนเครื่อง โดยดึงข้อมูลรายการสินค้ามาจาก Google Sheets ที่คุณสร้างไว้เรียบร้อยแล้วครับ!