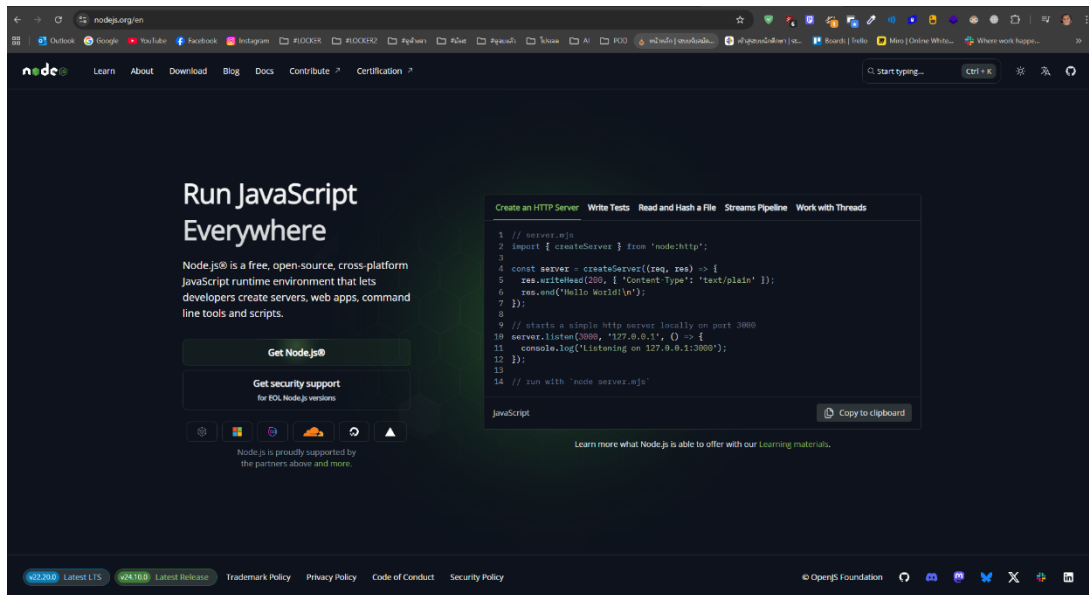


คู่มือการติดตั้งระบบบันทึกและคำนวณต้นทุนงานใบไม้รัฐวิสาหกิจชุมชนป่าต้นผึ้ง

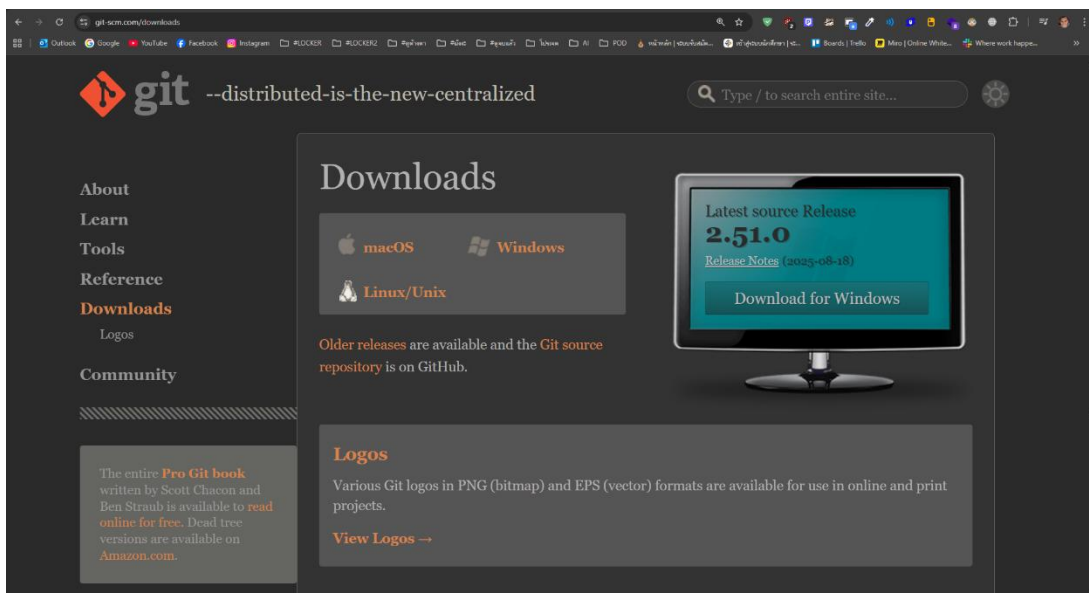
Leaf Plate Cost Management System Pa Ton Phueng Community Enterprise

1. ขั้นตอนการติดตั้งแบบโหนดนักพัฒนา

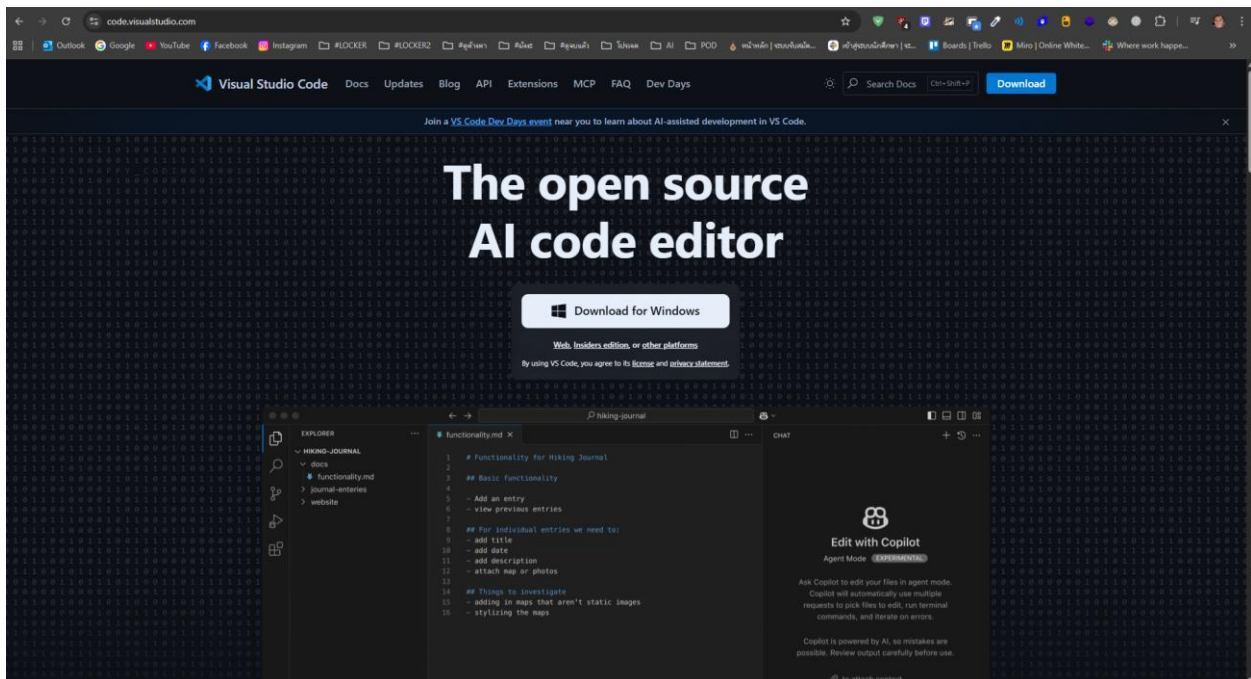
1.1 เข้าเว็บไซต์ <https://nodejs.org/en> เพื่อดาวน์โหลด nodejs และติดตั้ง



1.2 เข้าเว็บไซต์ <https://git-scm.com/downloads> เพื่อดาวน์โหลด git และ ติดตั้ง



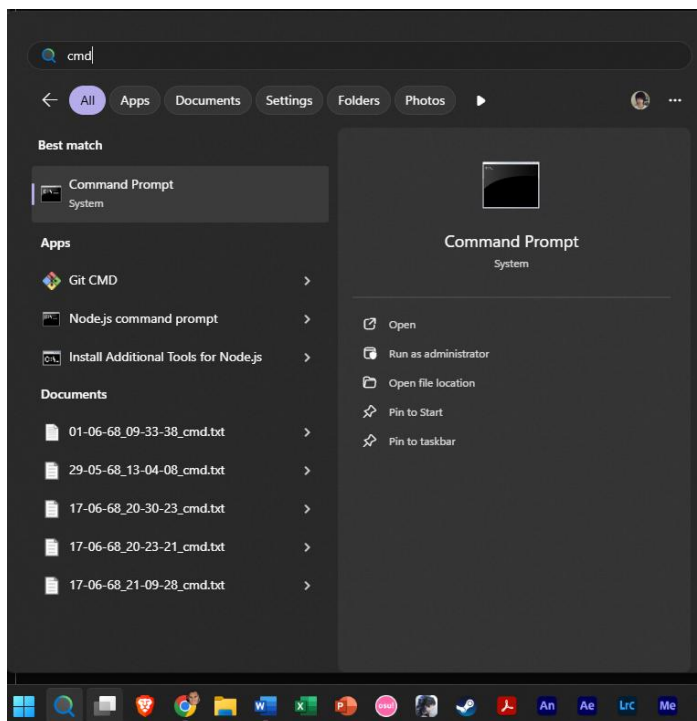
1.3 เข้าเว็บไซต์ <https://code.visualstudio.com> เพื่อดาวน์โหลด vscode และติดตั้ง



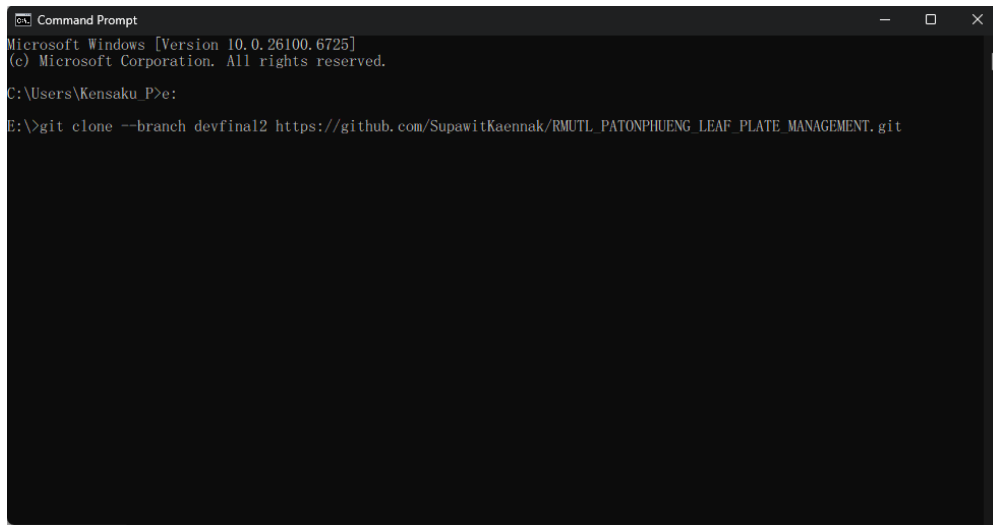
1.3 ให้ทำการ copy url ดังกล่าว

https://github.com/SupawitKaennak/RMUTL_PATONPHUENG_LEAF_PLATE_MANAGEMENT.git

1.4 เข้า cmd ใน computer (windows)



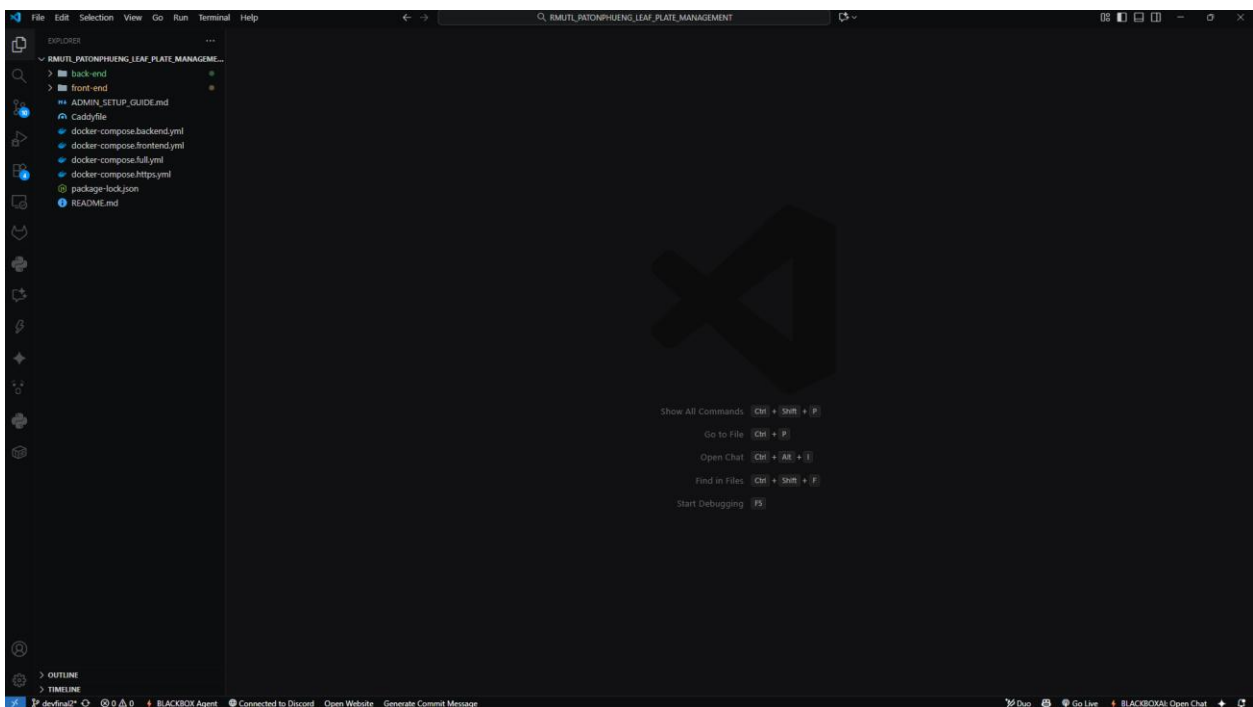
1.5 เข้า path ที่เราต้องการติดตั้ง แล้วทำการ clone โปรเจคลงมา “git clone --branch devfinal2 https://github.com/SupawitKaennak/RMUTL_PATONPHUENG_LEAF_PLATE_MANAGEMENT.git”



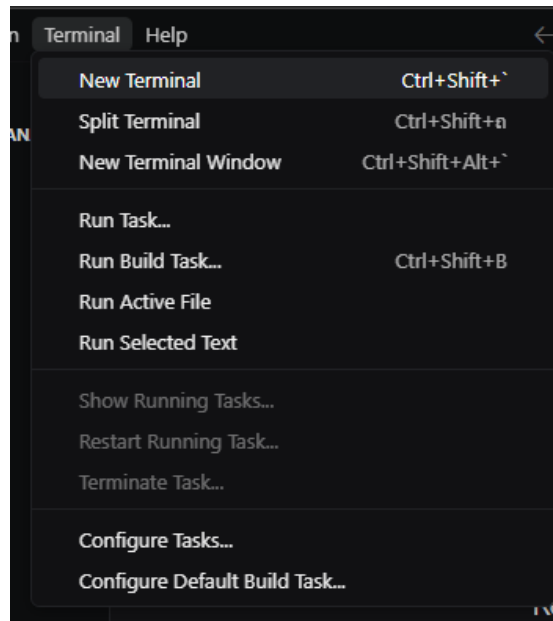
```
Command Prompt
Microsoft Windows [Version 10.0.26100.6725]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kensaku>P>e:
E:\>git clone --branch devfinal2 https://github.com/SupawitKaennak/RMUTL_PATONPHUENG_LEAF_PLATE_MANAGEMENT.git
```

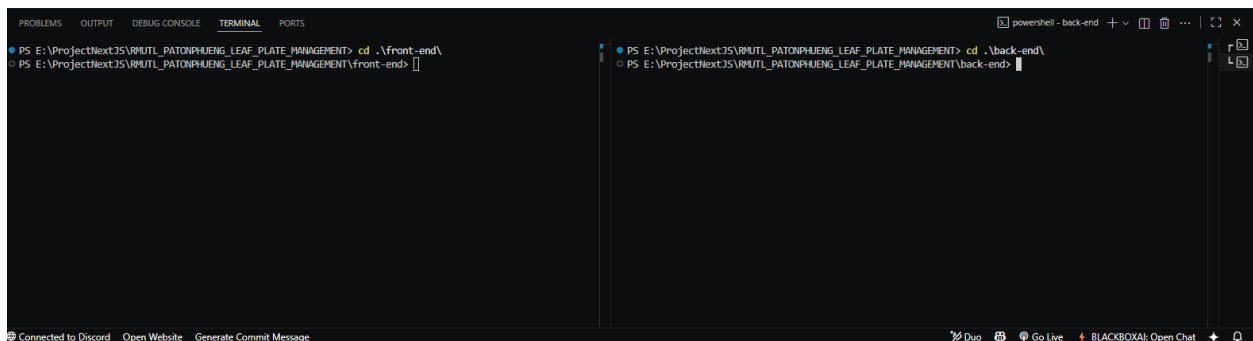
1.6 เข้า vscode แล้วทำการเปิดโปรเจคที่ clone มา



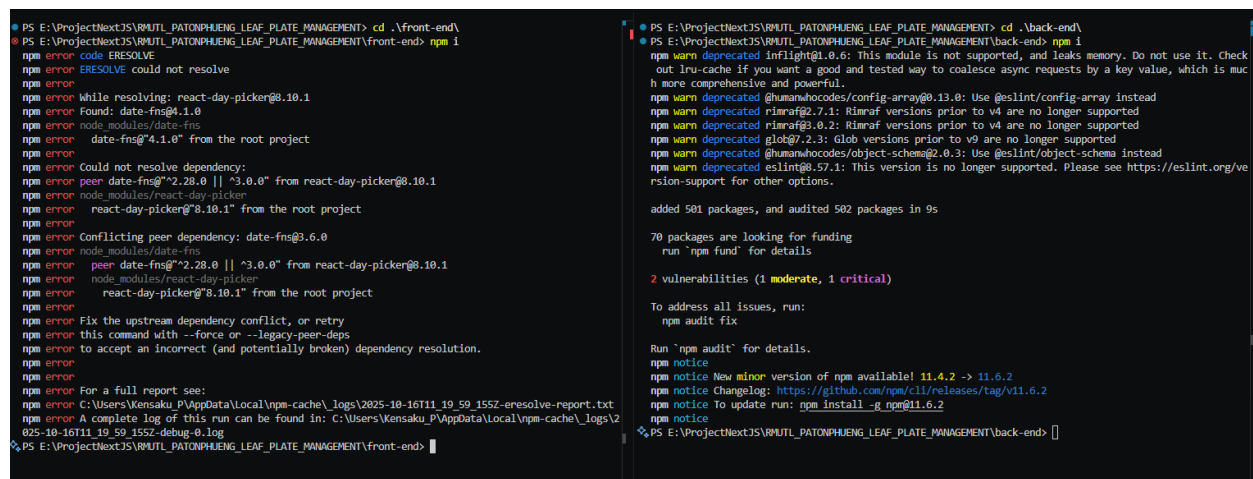
1.7 เปิด terminal ใหม่



1.8 ทำการ split terminal มา 2 อัน สำหรับ frontend และ backend



1.9 ทำการติดตั้ง dependencies ของทั้ง frontend และ backend คำสั่ง “npm i”



1.10 หาก npm i แล้วของ frontend error ให้เติมคำสั่ง “npm i --legacy-peer-deps”

```
PS E:\ProjectNextJS\RMUTL_PATONPHUENG_LEAF_PLATE_MANAGEMENT\front-end> npm i --legacy-peer-deps
added 480 packages, and audited 481 packages in 23s

43 packages are looking for funding
  run `npm fund` for details

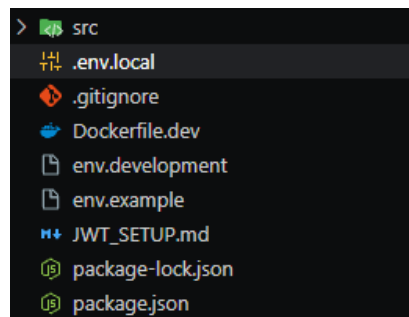
2 vulnerabilities (1 high, 1 critical)

To address all issues possible, run:
  npm audit fix --force

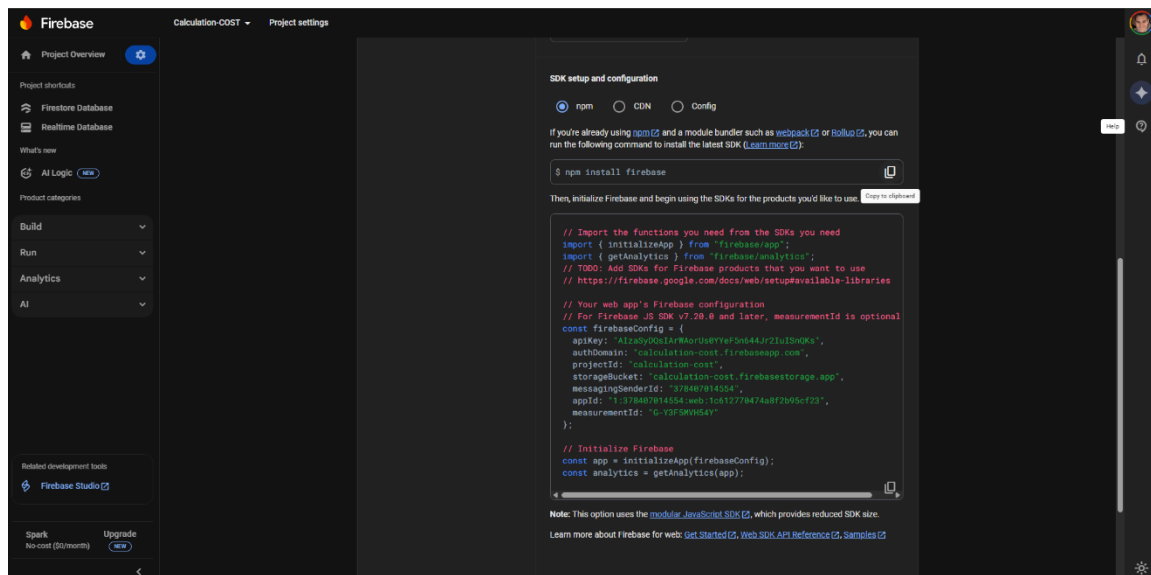
Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
```

1.11 ทำการ copy ไฟล์ env.development ทั้ง frontend และ backend และเปลี่ยนชื่อไฟล์เป็น .env.local



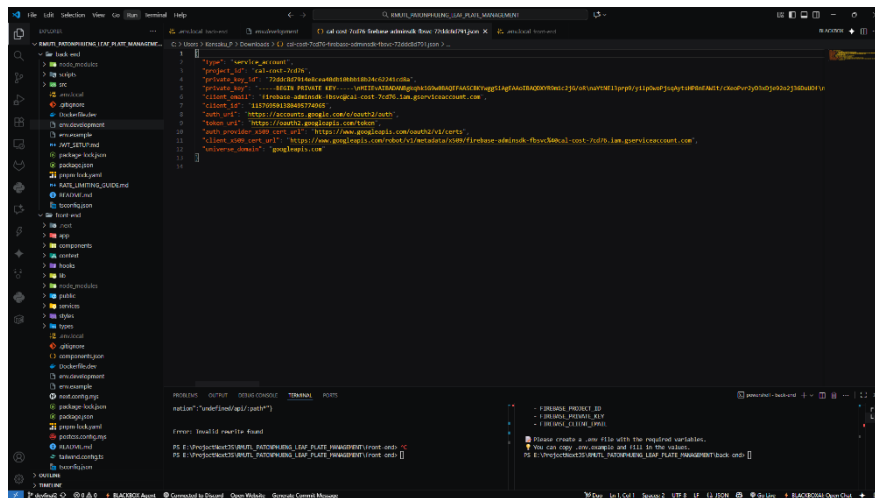
1.12 เข้า firebase ใน browser แล้วไปที่ project setting ของตนเอง เลือกแถบ general แล้ว copy sdk ของ app ที่เราสร้างแล้วนำไปใส่ใน .env.local ของ frontend



```
front-end > env.development

1 # =====
2 # FRONT-END ENVIRONMENT VARIABLES (.env.local)
3 # =====
4 # ⚠️ ไฟล์นี้จะไม่ถูก commit ไปยัง Git repository
5 # ⚠️ กรุณาแทนที่ค่าเหล่านี้ด้วยข้อมูลจริงจาก Firebase Console
6
7 # =====
8 # FIREBASE CLIENT CONFIGURATION
9 # =====
10 # ค่าเหล่านี้ได้จาก Firebase Console > Project Settings > General > Your apps > Web app
11 NEXT_PUBLIC_FIREBASE_API_KEY=AiZaSyD-YourApiKeyHere
12 NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN=your-project-id.firebaseio.com
13 NEXT_PUBLIC_FIREBASE_PROJECT_ID=project-id
14 NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET=your-project-id.appspot.com
15 NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID=1234567890
16 NEXT_PUBLIC_FIREBASE_APP_ID=1:1234567890:web:abcdef123456
17
18
19 # =====
```

1.13 ใน firebase เข้าแถบ Service accounts ทำการ generate new private key จากนั้นจะได้ไฟล์ json มาให้ copy ในไฟล์ json ไปใส่ใน .env.local ของ backend



```
# =====
# Firebase Configuration
# =====
# Firebase Project ID
FIREBASE_PROJECT_ID=your-actual-project-id

# Firebase Service Account Private Key ID
FIREBASE_PRIVATE_KEY_ID=your-actual-private-key-id

# Firebase Service Account Private Key (ต้องใส่ \n สำหรับ newlines)
FIREBASE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----\nYour Actual Private Key Here\n-----END PRIVATE KEY-----\n"

# Firebase Service Account Client Email
FIREBASE_CLIENT_EMAIL=your-actual-service-account@your-project.iam.gserviceaccount.com

# Firebase Service Account Client ID
FIREBASE_CLIENT_ID=your-actual-client-id

# Firebase Auth URIs
FIREBASE_AUTH_URI=https://accounts.google.com/o/oauth2/auth
FIREBASE_TOKEN_URI=https://oauth2.googleapis.com/token
```

1.14 ใช้คำสั่ง npm run dev ของทั้งสอง terminal ทั้ง frontend และ backend

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
43 packages are looking for funding
  run `npm fund` for details

2 vulnerabilities (1 high, 1 critical)
  -
PS E:\Project\text35\PMUTIL_PATONPHUENG_LEAF_PLATE_MANAGEMENT\front-end> ^C
PS E:\Project\text35\PMUTIL_PATONPHUENG_LEAF_PLATE_MANAGEMENT\front-end> npm run dev

> my-v0-project@0.1.0 dev
> next dev

▲ Next.js 14.2.16
- Local:    http://localhost:3000
- Environments: .env.local

✓ Starting...
✓ Ready in 2.2s
  |

run `npm fund` for details

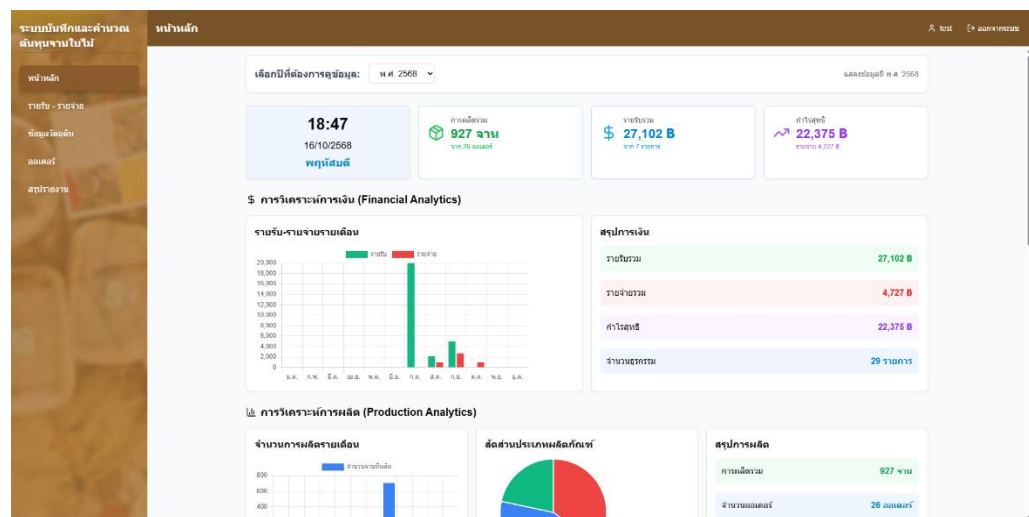
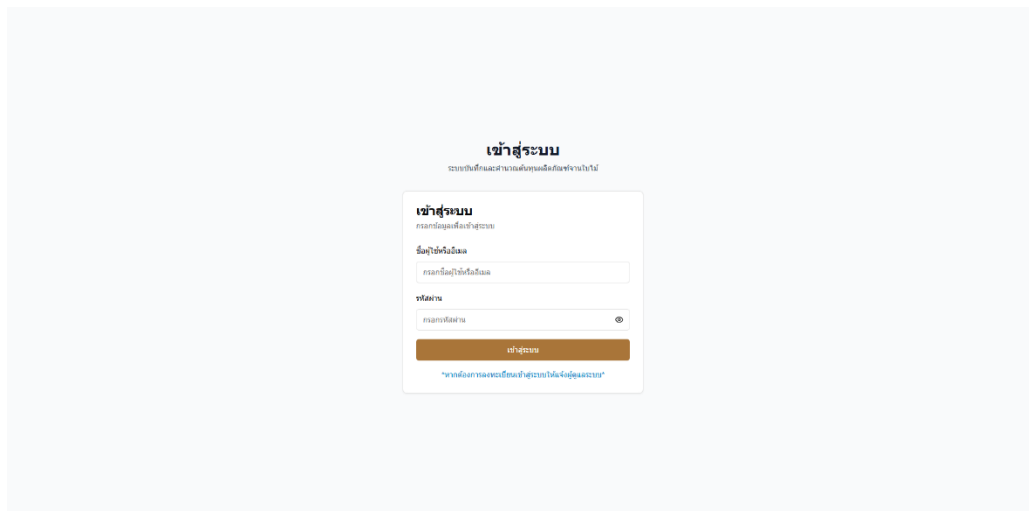
2 vulnerabilities (1 moderate, 1 critical)

To address all issues, run: ...
  You can copy .env.example and fill in the values.
PS E:\Project\text35\PMUTIL_PATONPHUENG_LEAF_PLATE_MANAGEMENT\back-end> npm run dev

> income-expense-backend@1.0.0 dev
> ts-node-dev --respawn --transpile-only src/server.ts

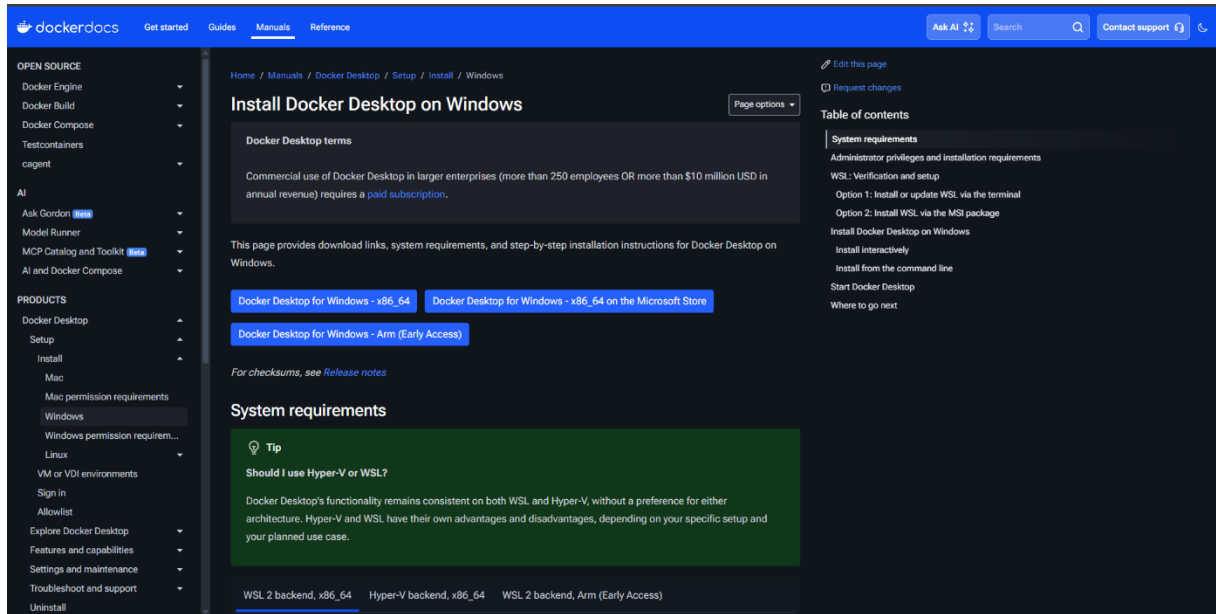
[INFO] 18:43:19 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.8.3)
  Using .env.local configuration
  Firebase Config Status:
  - Environment: development
  - Project ID: calculation-cost
  - Client Email: firebase-adminsdk-fbsvc@calculation-cost.iam.gserviceaccount.com
  - Using Real Credentials: ✓
  - Using Real Credentials: ✓
  - Firebase Admin initialized successfully!
  - Server running on port 8080
  - Environment: development
  - JWT Secret configured: Yes
  - JWT Expires In: 30m
  - CORS Origins: http://localhost:3000, http://127.0.0.1:3000
  - Rate Limiting: Development Mode (Very Lenient)
  - Dev Rate Limit: 1000 requests per 10 seconds
  - Health check: http://localhost:8080/health
```

1.15 ลองเข้าเว็บไซต์ด้วย <http://localhost:3000/> หรือ <http://127.0.0.1:3000/>

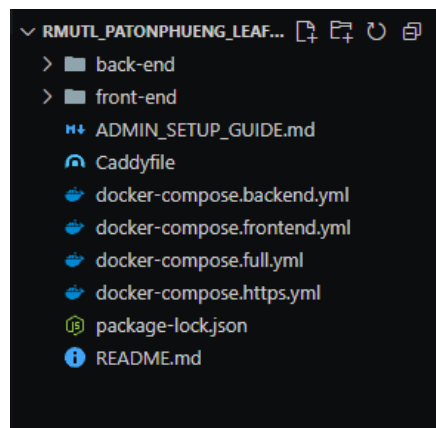


2. ขั้นตอนการติดตั้งแบบใช้งานจริงผ่าน docker

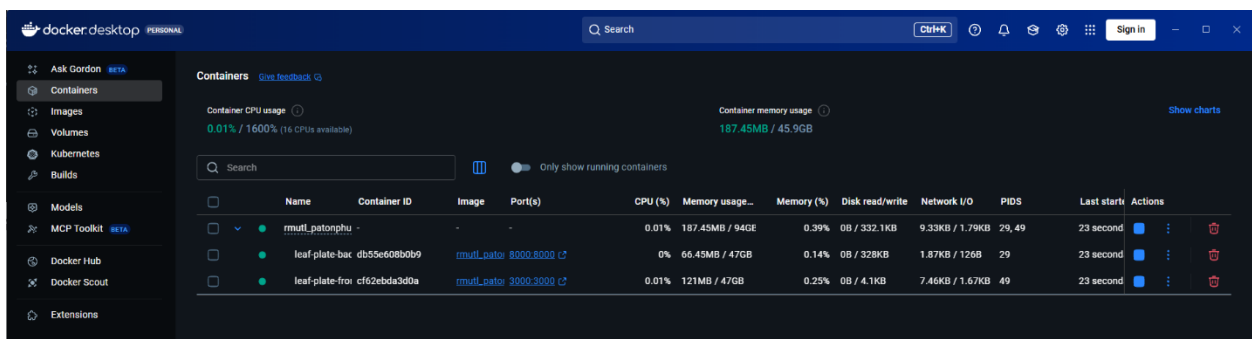
2.1 ทำการเข้าเว็บไซต์ <https://docs.docker.com/desktop/setup/install/windows-install/> เพื่อโหลด docker desktop



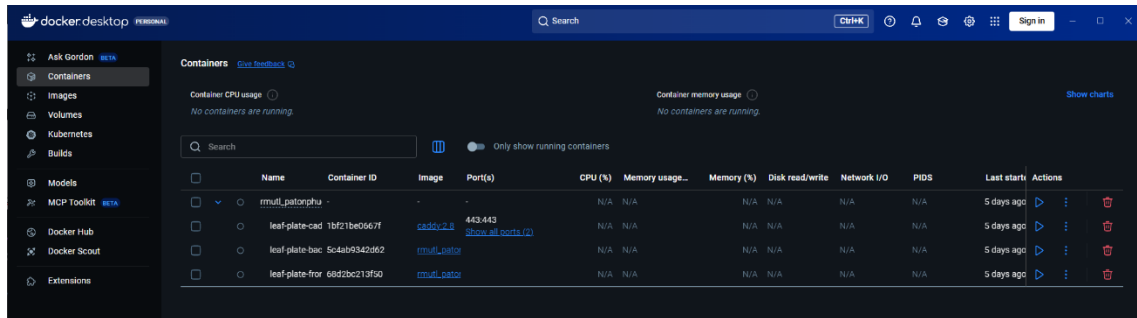
2.2 กลับมา path แรกสุด จะมีไฟล์ docker อยู่ 4 ไฟล์ เราจะใช้แค่ 2 ไฟล์ คือ แบบ full และแบบ https



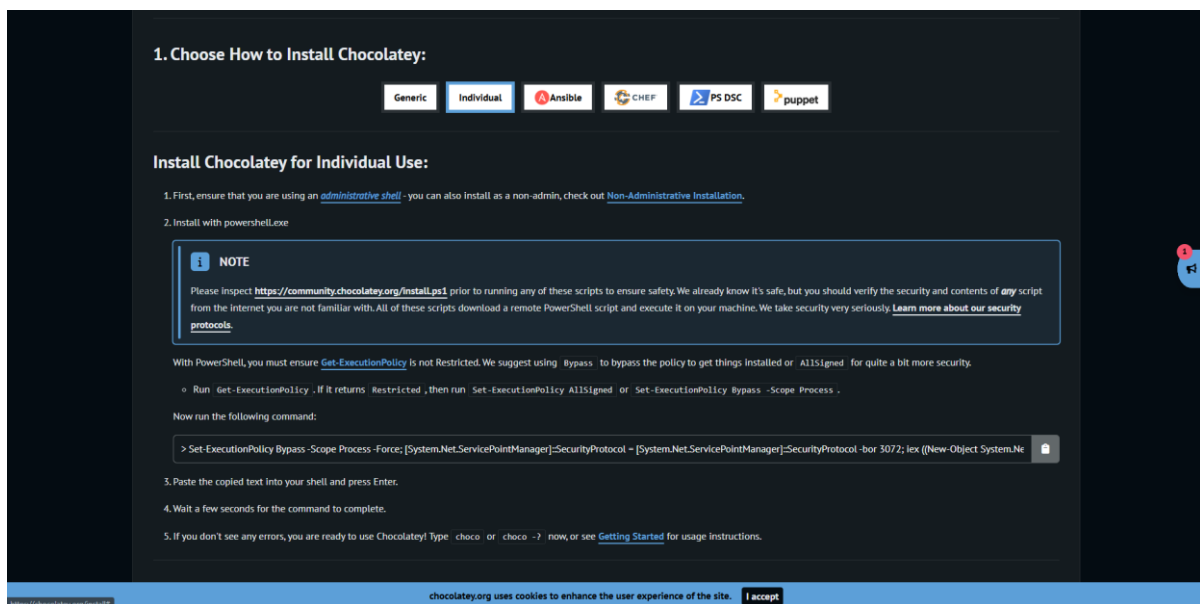
2.3 ลงแบบ full จะเป็น http ปกติ ใช้คำสั่ง “docker compose -f docker-compose.full.yml up -d --build”



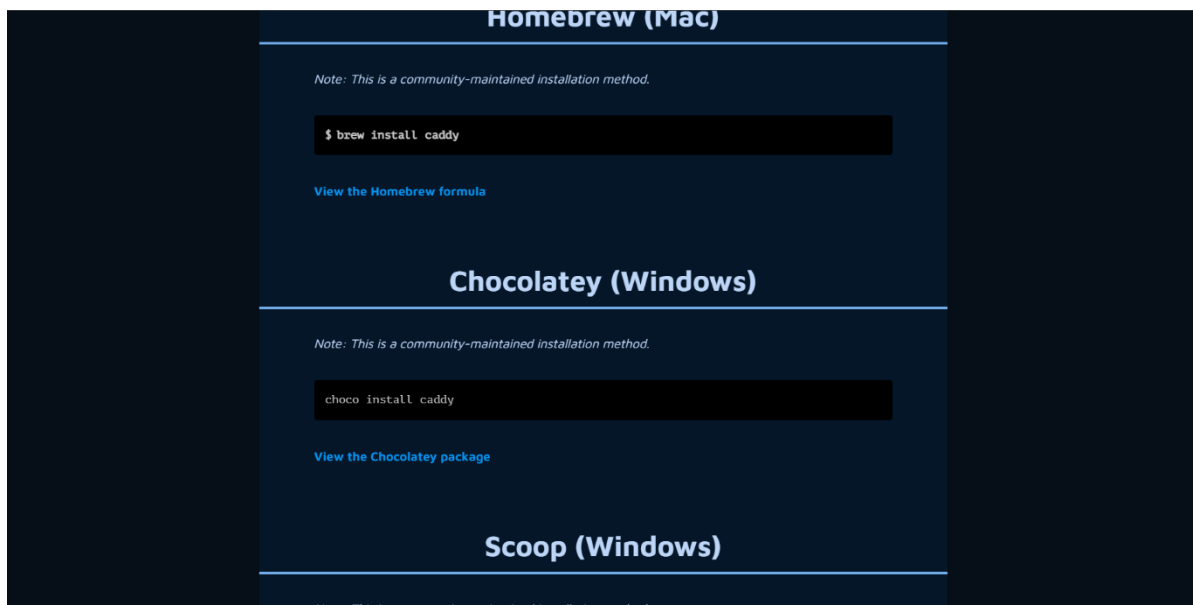
2.4 ลงแบบ https แบบเข้ารหัส ใช้คำสั่ง “docker compose -f docker-compose.https.yml up -d --build”



2.5 ทำการเข้าเว็บไซต์ <https://chocolatey.org/install> เพื่อติดตั้ง terminal choco (chocolatey)



2.6 ทำการเข้าเว็บไซต์ <https://caddyserver.com/docs/install> เพื่อติดตั้ง cert เข้ารหัสเว็บไซต์แบบ root



2.7 ใช้คำสั่ง docker cp leaf-plate-caddy:/data/caddy/pki/authorities/local/root.crt . เพื่อ copy cert มาติดตั้ง

```
PS E:\ProjectNextJS\RMUTL_PATONPHUENG_LEAF_PLATE_MANAGEMENT> docker cp leaf-plate-caddy:/data/caddy/pki/authorities/local/root.crt .  
Successfully copied 2.56kB to E:\ProjectNextJS\RMUTL_PATONPHUENG_LEAF_PLATE_MANAGEMENT\.
```

