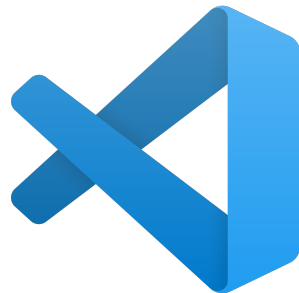


Docker วิธีใช้/แก้ปัญหา ในคาบ lab

ที่ต้องติดตั้ง มี : node js / vs code / docker



กรณี ถ้าไม่ลง ตัว node js มันก็จะขึ้น แบบ นี้ เวลา เรา พิมพ์ *npm* ใน *vs code* ⇒

```
PS D:\ทดลอง> npm
npm : The term 'npm' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the
spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ npm
+ ~~~
+ CategoryInfo          : ObjectNotFound: (npm:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

⇒ แก้ต้องไป ลง node js

ถ้าลง เสร็จละ ให้ปิด ตัว vs code แล้วเปิดใหม่ นะ ถึง จะติด

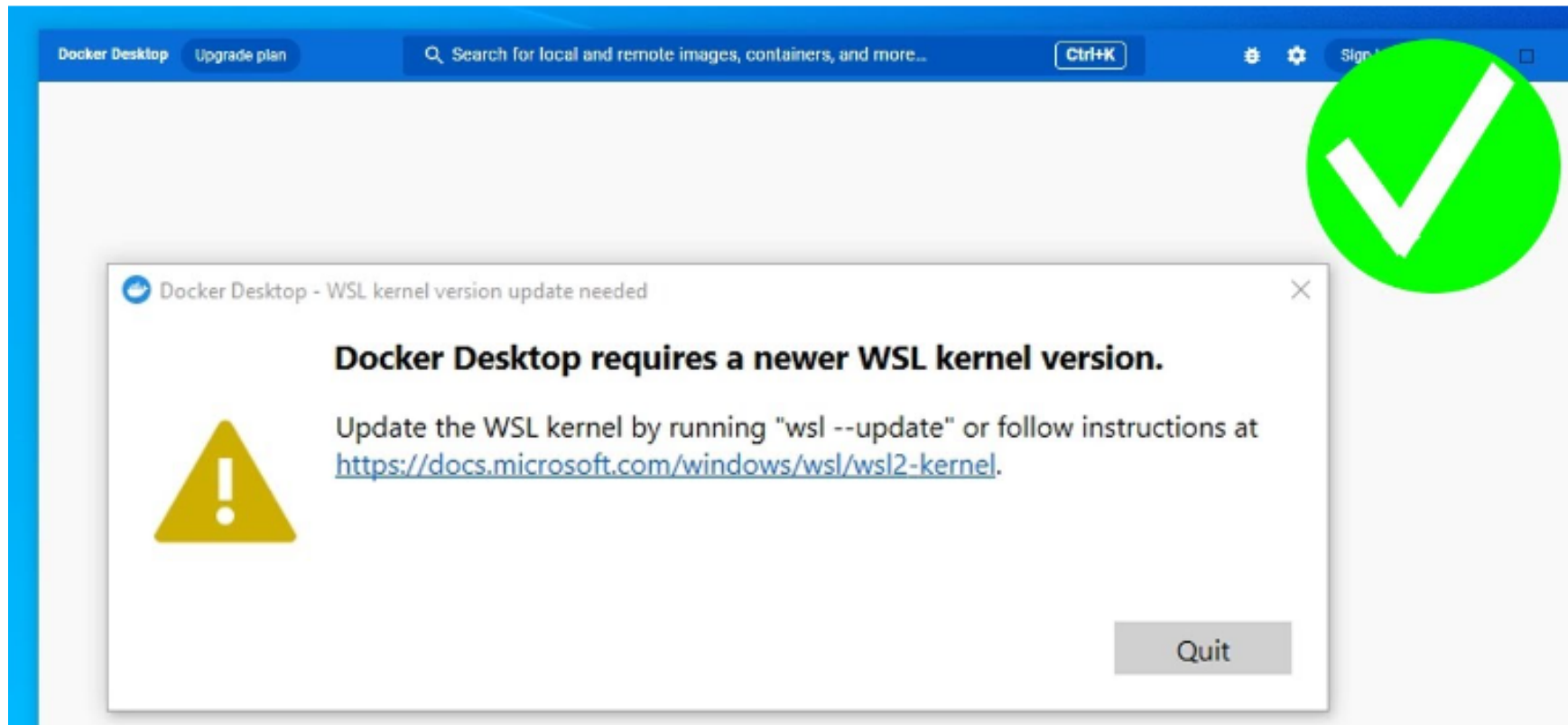
ถ้าติดเป็นแบบนี้ :

```
Usage:

npm install          install all the dependencies in your project
npm install <foo>    add the <foo> dependency to your project
npm test            run this project's tests
npm run <foo>        run the script named <foo>
npm <command> -h     quick help on <command>
npm -l              display usage info for all commands
npm help <term>      search for help on <term> (in a browser)
npm help npm         more involved overview (in a browser)

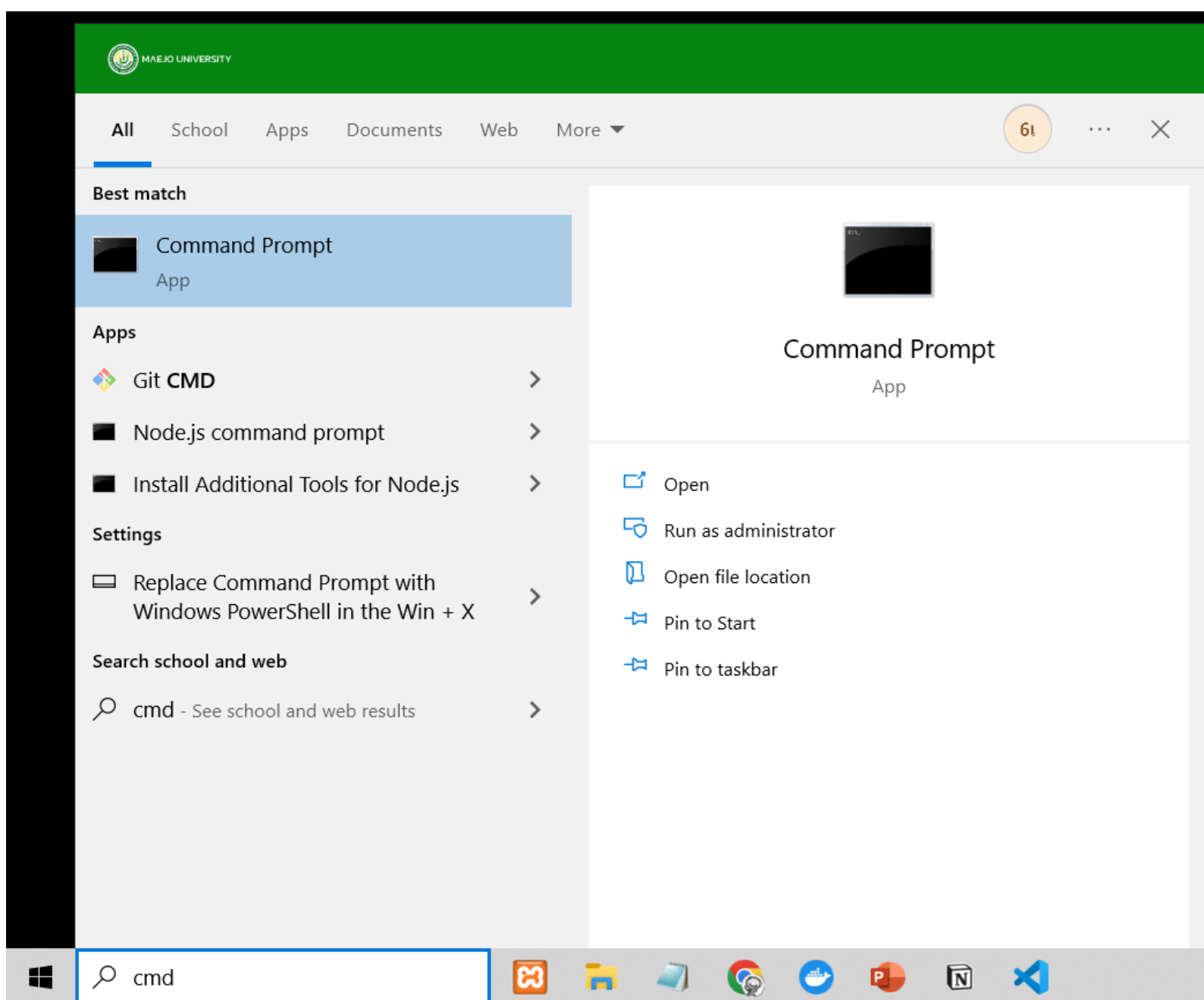
All commands:
```

เมื่อเราติดตั้ง ตัว Docker เสร็จ แล้ว เปิดโปรแกรมขึ้นมาแล้วเป็น แบบนี้ :

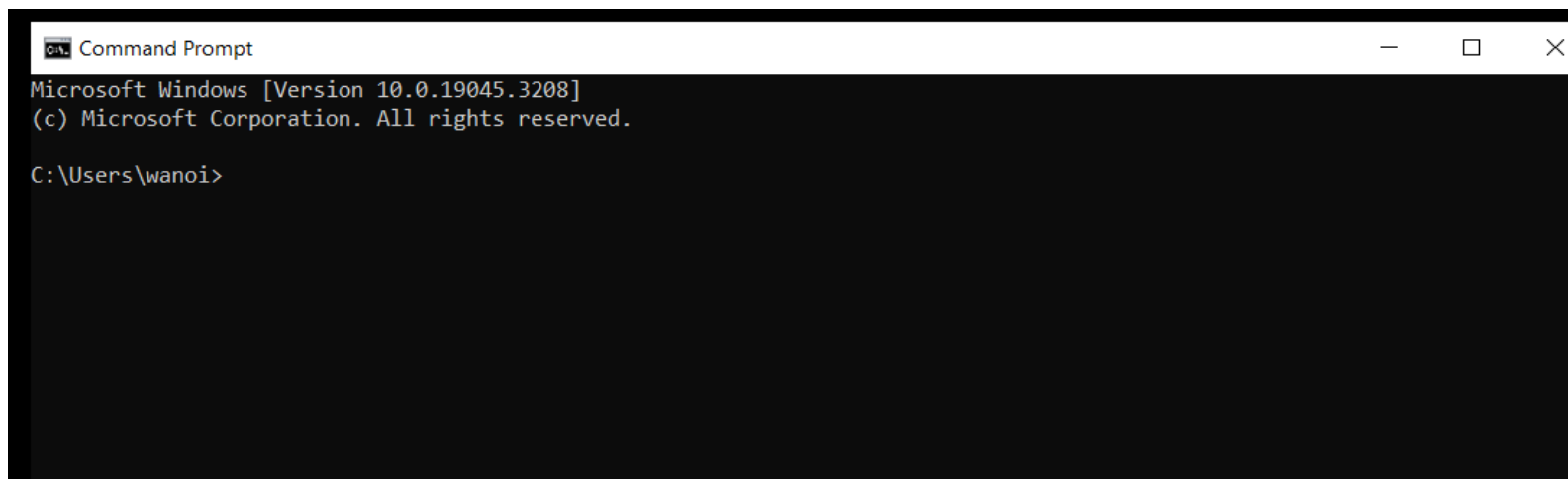


ปัญหานี้ คือ มันบอกว่า ให้เรา ไป update ตัว WSL โดยการที่เรา จะ update นั้น ⇒

ให้เราไป ที่ ช่องค้นหา แล้ว : พิมพ์ cmd แล้วกด เข้าไป เลย / หรือ กด window + R แล้ว พิมพ์ cmd ก็ได้



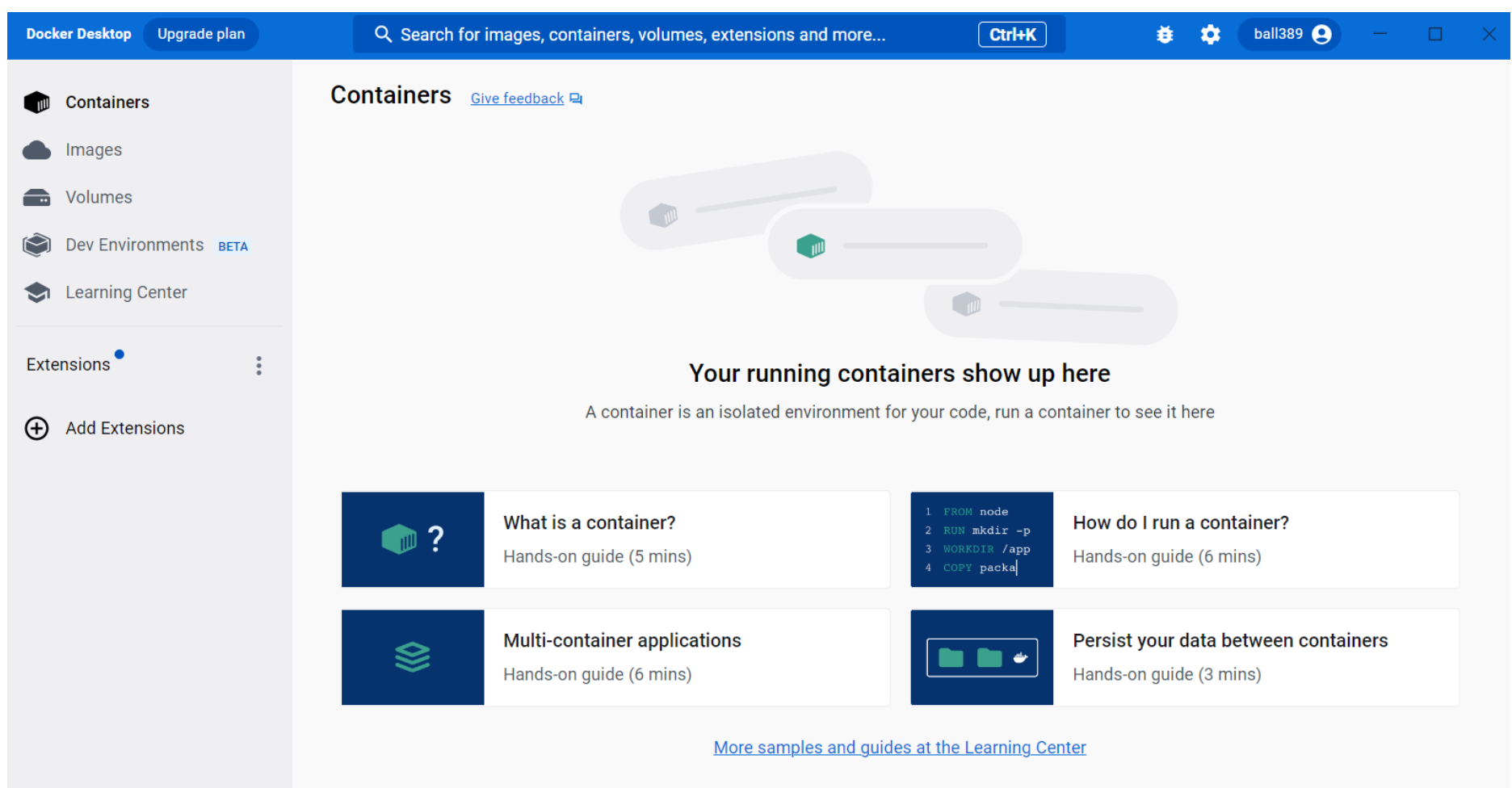
กดเข้ามา ขึ้นแบบนี้ ⇒



ให้เราพิมพ์ **wsl --update** แล้วกด Enter รอจน รันเสร็จ

แล้ว ลอง ปิด เปิด docker ใหม่

ถ้าขึ้นแบบนี้ :



แบบนี้แสดง ว่า ติดตั้ง เรียบร้อยละ

ถ้ายังไม่ขึ้นแบบนี้ ให้ **รีสตาร์ท** เครื่อง แล้วเปิดใหม่

ถ้า เกิดมันขึ้น แบบนี้ :

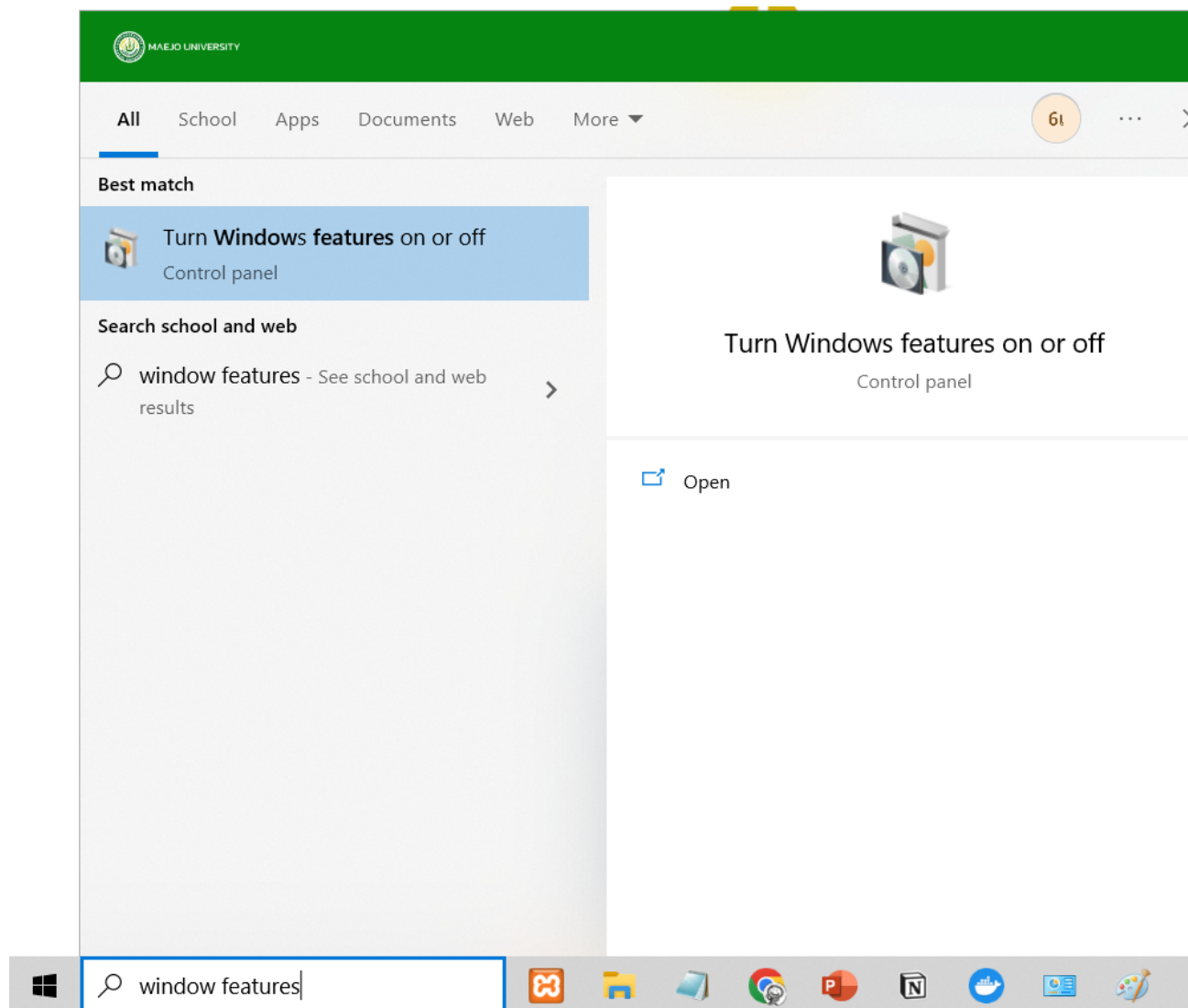
An error occurred



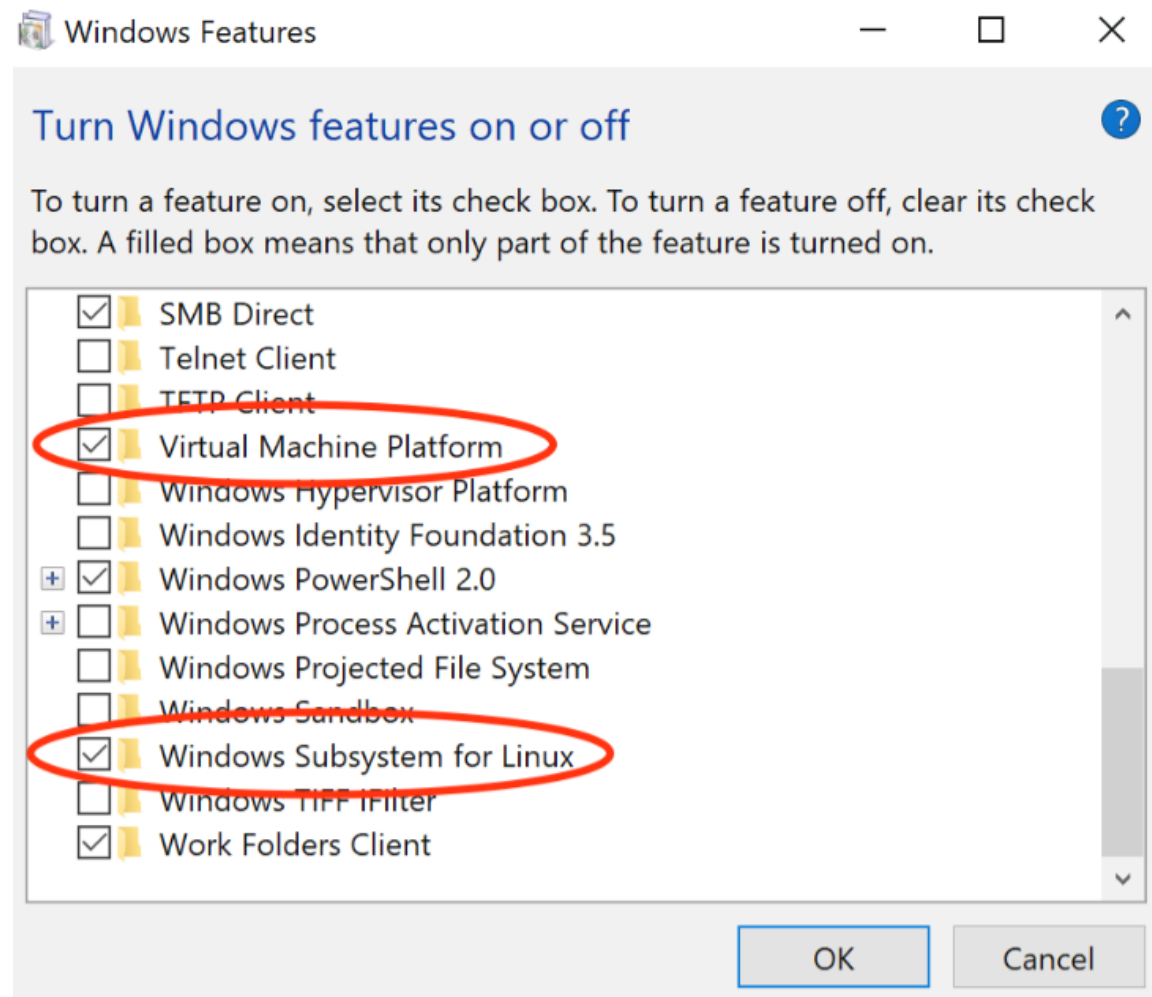
Hardware assisted virtualization and data execution protection must be enabled in the BIOS. See <https://docs.docker.com/docker-for-windows/troubleshoot/#virtualization>

OK

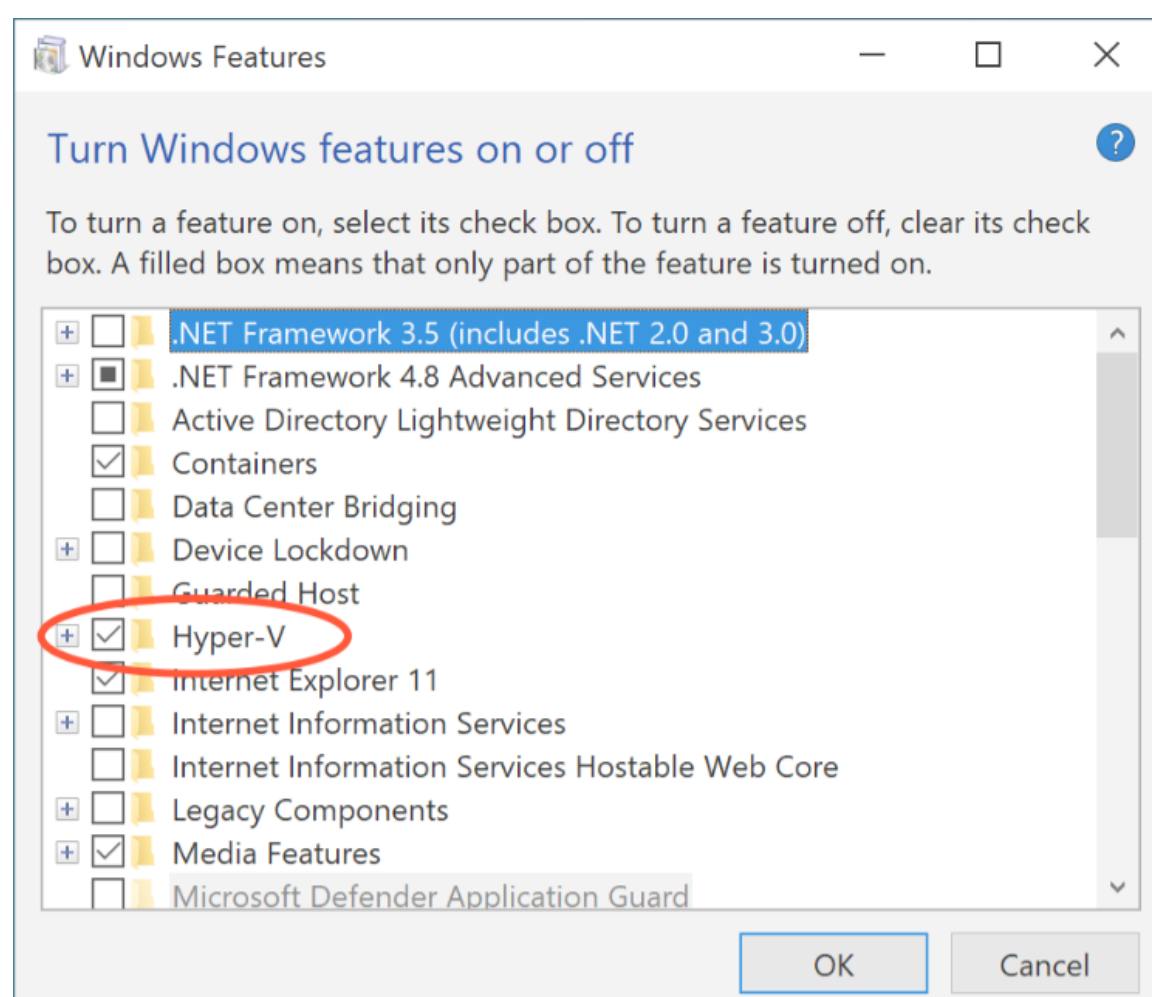
ให้ พิมพ์ ตรงช่องค้นหา ใน window



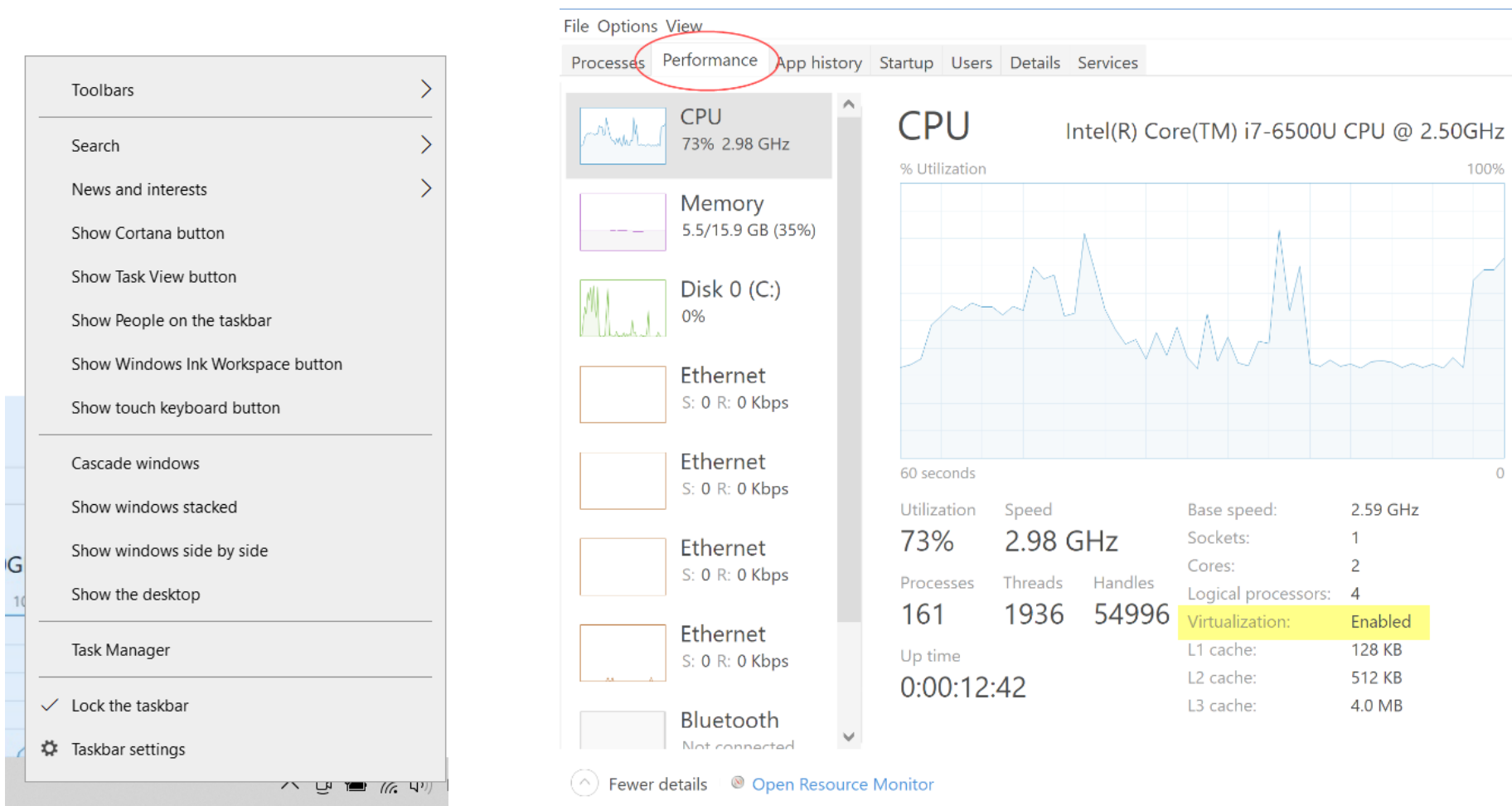
คลิก เลือก turn windows features on or off



ให้เลือก ตึก ตาม ที่ วง นะ



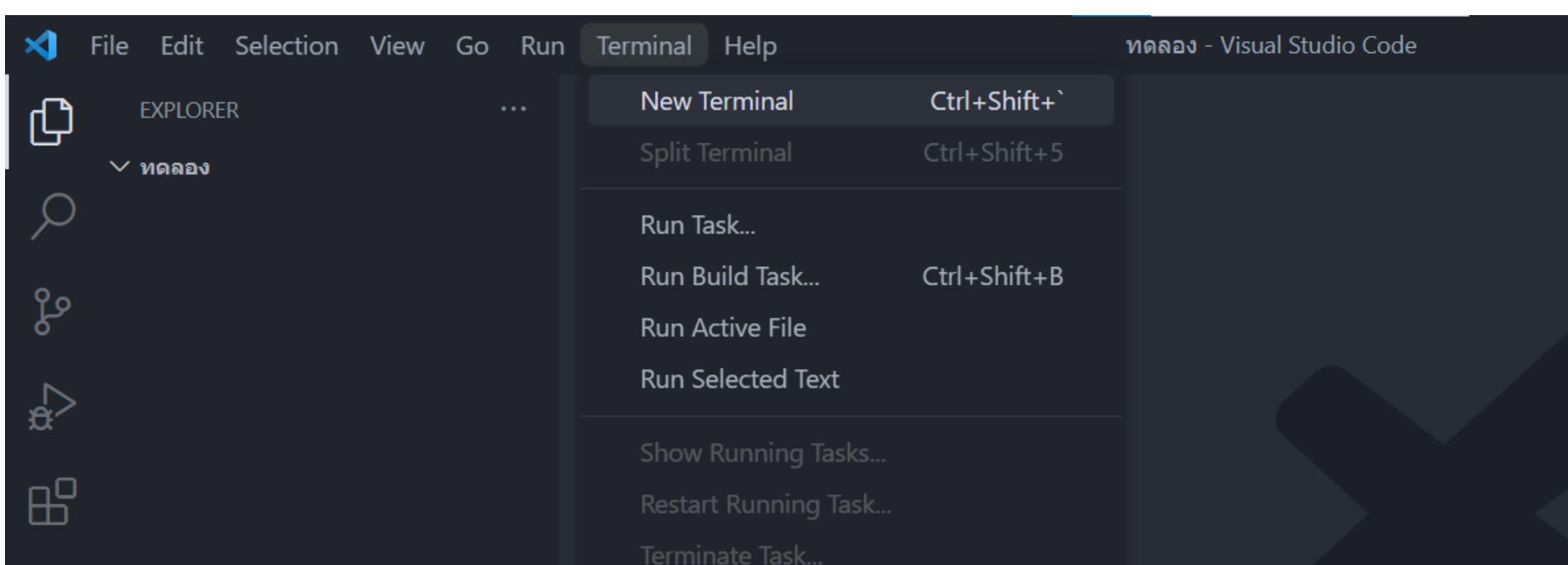
หลังจากนั้น ให้ คลิก ขวา ตรง แถบจอ เมนู ด้านล่าง เลือก Task manager



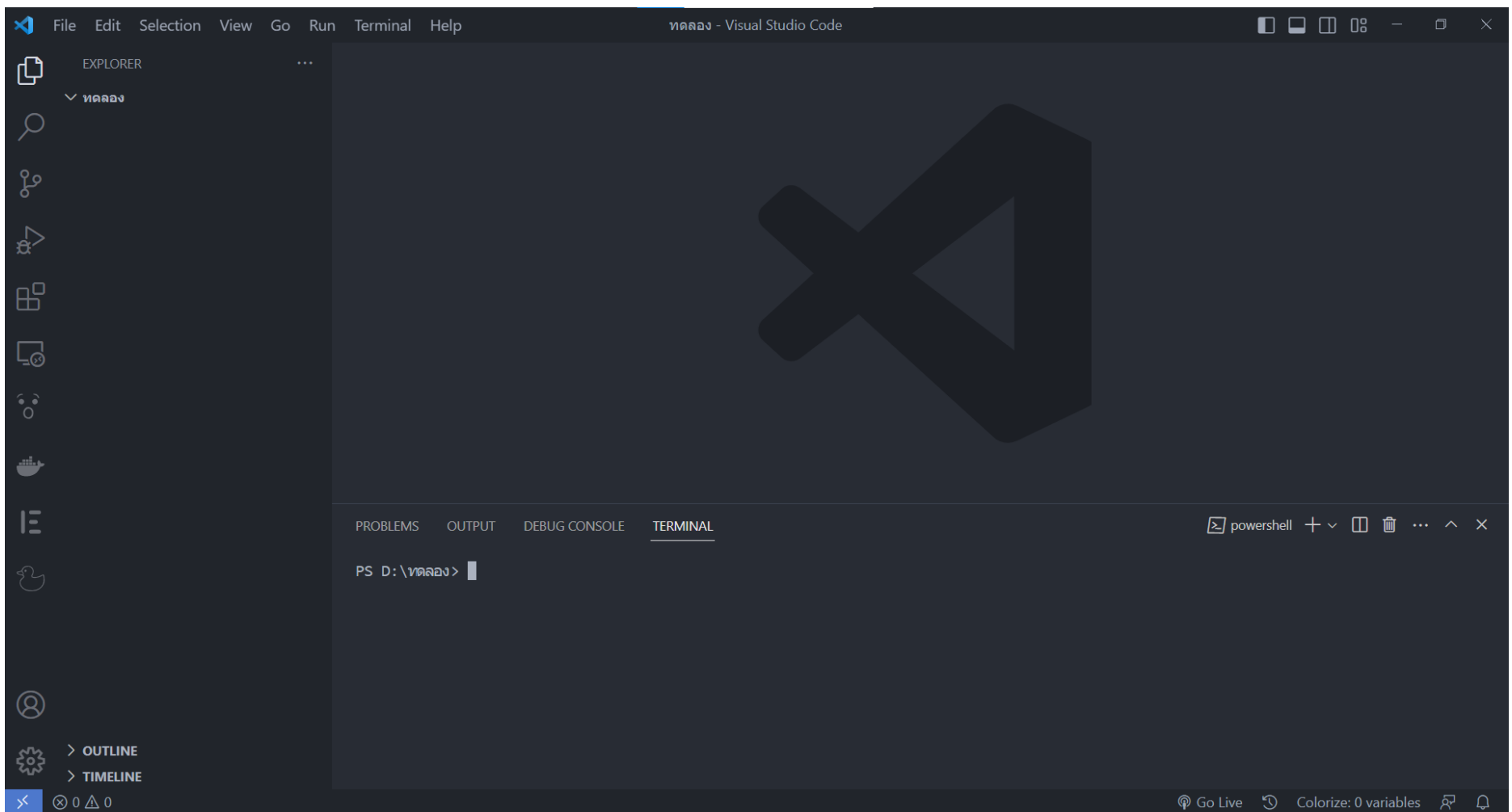
ก็จะเห็นว่า ตัว virtualization เปิด ใช้งานแล้ว ลอง ปิดเครื่องเปิด ใหม่ ดู
 ถ้ายังแก้ไม่ได้ : https://www.youtube.com/watch?v=AK7hvJ_xolc หรือ
<https://www.youtube.com/watch?v=HHuQv2yQloo>

เมื่อ เข้า vs code มา ให้ กด

Terminal ⇒ New Terminal



มันก็จะขึ้น แถบ ตัว terminal ขึ้นมา



ที่เราต้อง ติดตั้ง npm ใน Terminal มี \Rightarrow

`npm init`

`npm install`

`npm install express --save`

`npm install body-parser`

พิมพ์ `npm init` \Rightarrow จะขึ้นแบบนี้


```
PS D:\ทดลอง> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

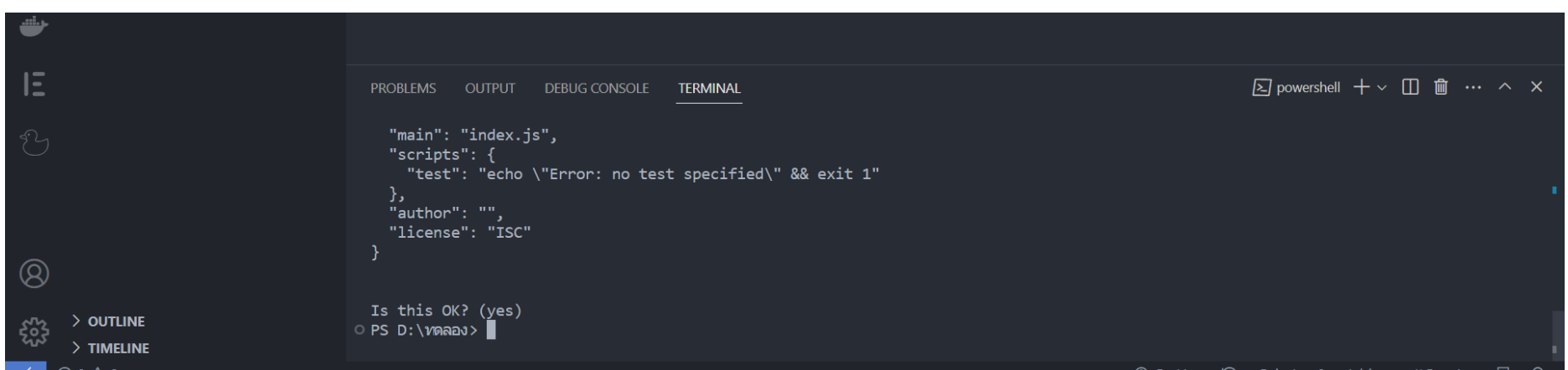
Press ^C at any time to quit.
package name: (ทดลอง) 
```

อาจารย์จะบอกว่า จะให้ เขียนชื่ออะไร ถ้าพิมพ์ไรเสร็จ ก็ กด enter

```
Sorry, name can only contain one friendly characters.
package name: (ทดลอง) hello-ball
version: (1.0.0) 
```

สมมติ เราตั้งชื่อ package name : ว่า hello-ball

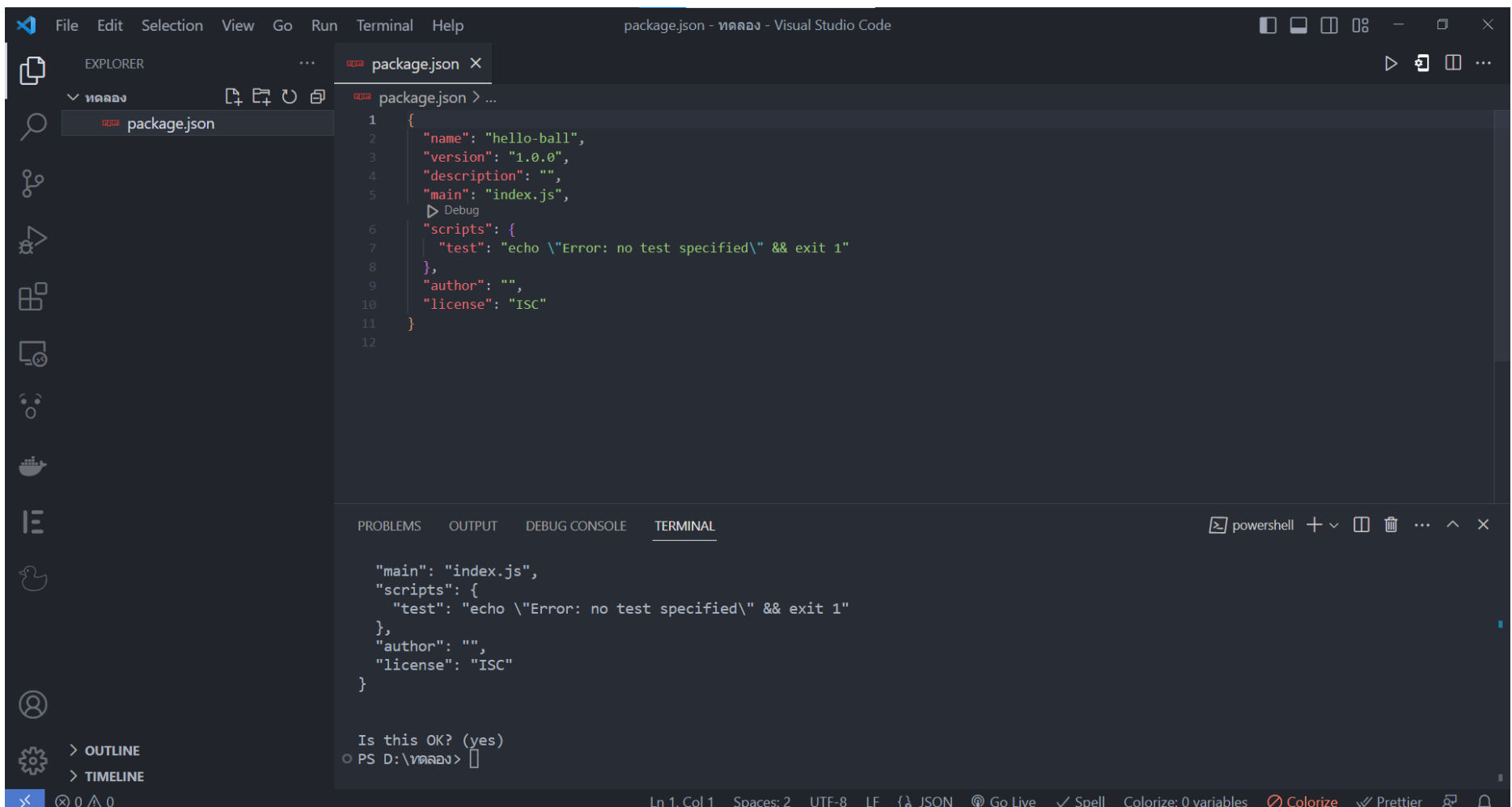
มันจะมีให้เขียน อะไร อีกเยอะ แต่ สมมติเราเขียนหมดละ กด enter จน กลับมา ปกติ



```
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

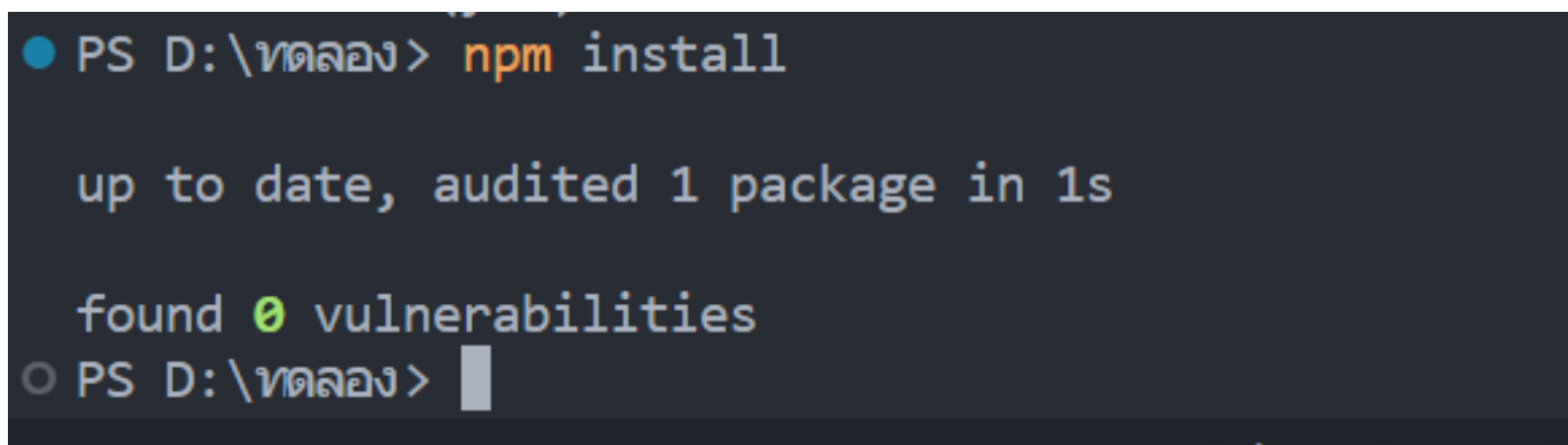
Is this OK? (yes)
PS D:\ทดลอง>
```

จะเห็นว่าจะมีไฟล์ \Rightarrow package.json มา

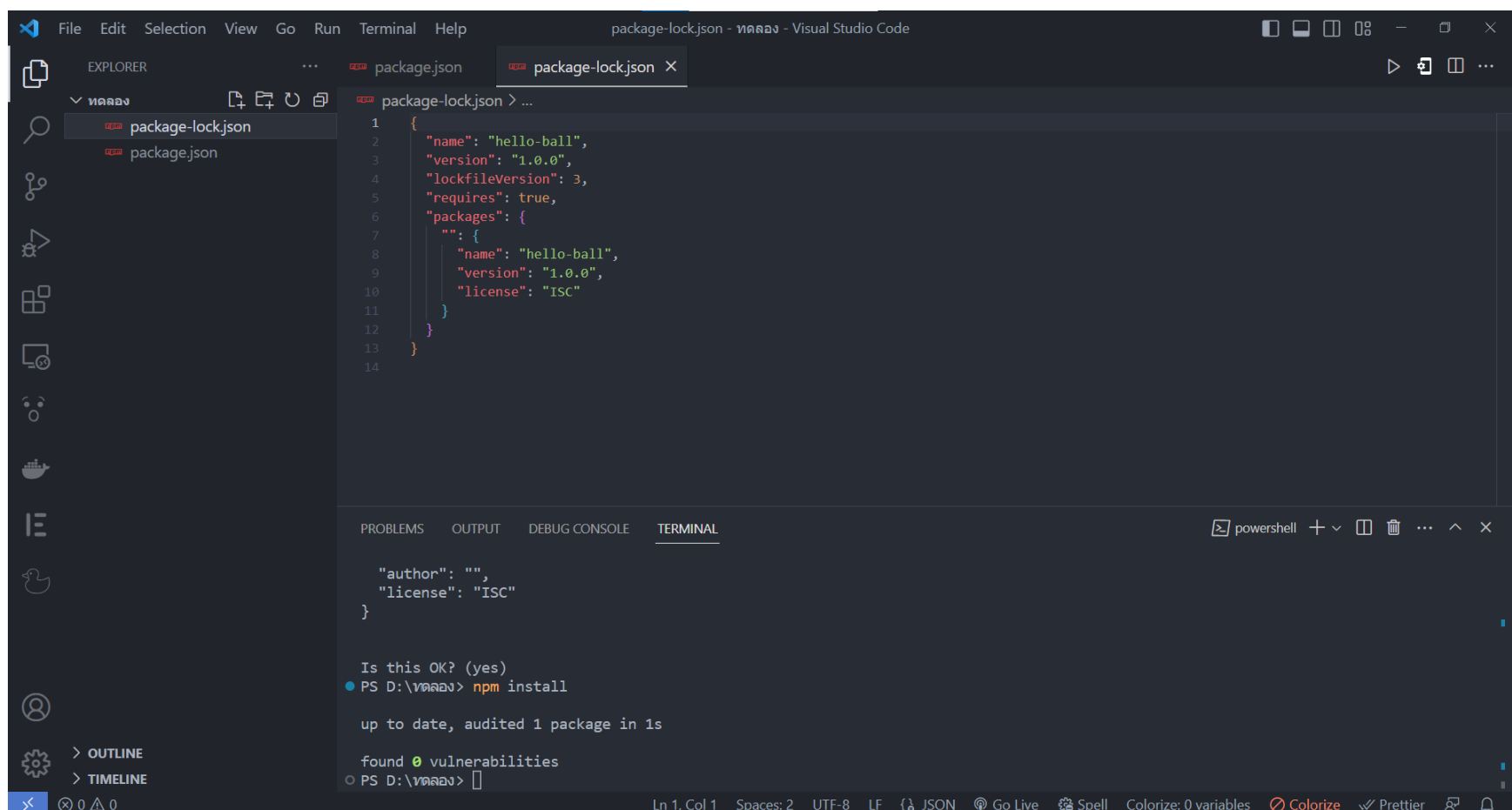


ไม่ต้องไป แก่ หรือ ทำไร มัน นะ ปล่อย มันอยู่ สงบๆ

ต่อ มาให้เรา พิมพ์ **npm install**



ก็จะมีไฟล์ ใหม่เพิ่มมา คือ \Rightarrow package-lock.json



ต่อ มาให้ พิมพ์ **npm install express --save**

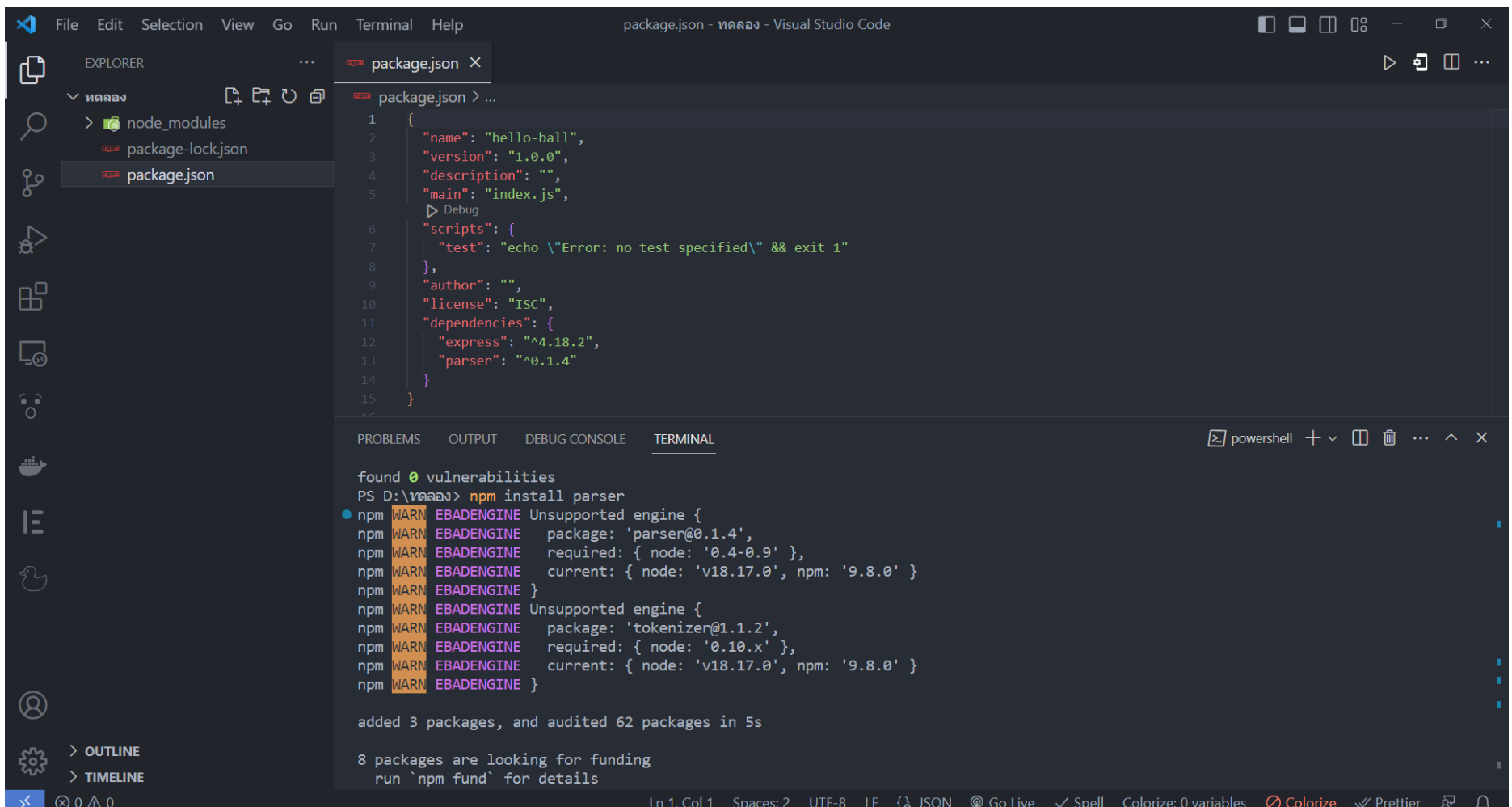
```
PS D:\ทดลอง> npm install express --save

added 58 packages, and audited 59 packages in 5s

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

found 0 vulnerabilities
PS D:\ทดลอง>
```

ต่อ มาให้ พิมพ์ **npm install body-parser**

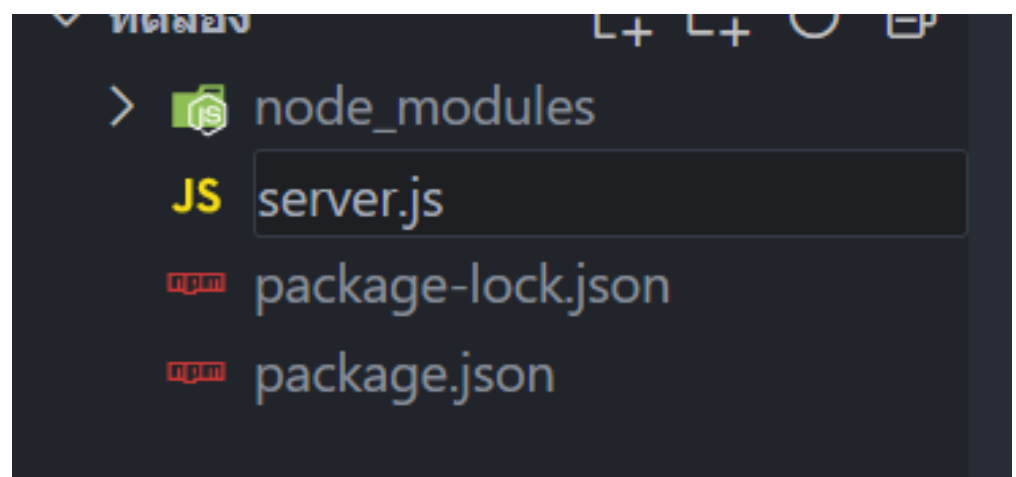
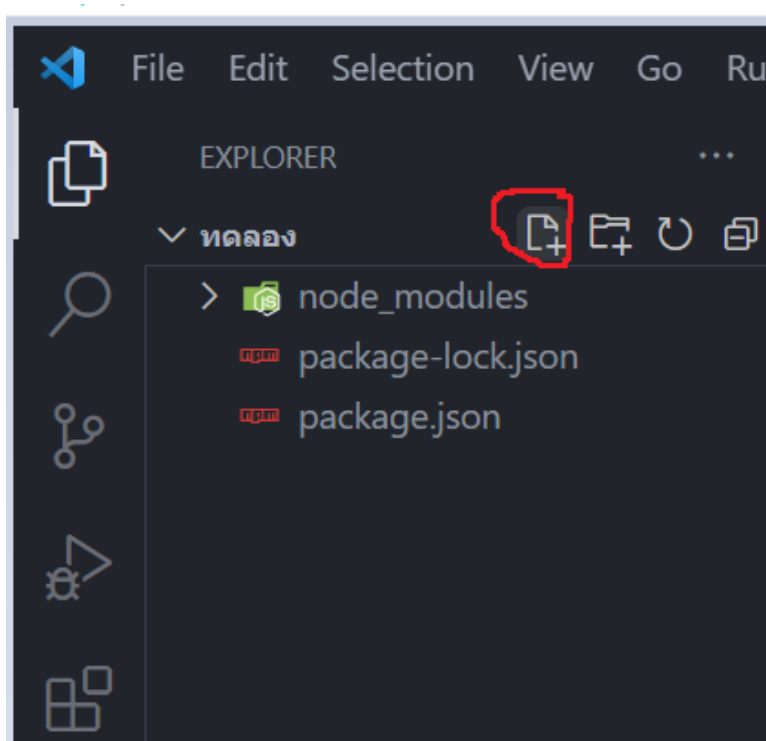


The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure with 'node_modules', 'package-lock.json', and 'package.json'. The main editor shows the 'package.json' file with the following content:

```
{
  "name": "hello-ball",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2",
    "parser": "^0.1.4"
  }
}
```

The terminal at the bottom shows the output of the command 'npm install parser'. It displays several warnings about unsupported engines and the successful installation of the 'parser' package.

เท่านี้ก็เรียบร้อยละ จะเห็น ว่า มีไฟล์ node_modules เพิ่มมาละ ต่อมาเราก็ พร้อมที่จะ เขียนโค้ดแล้ว



กดสร้าง ไฟล์ ให้ตั้งชื่อ ว่า server.js

แล้ว enter เมื่อ พิมพ์เสร็จ

ให้เขียนโค้ด ตามนี้ นะ =>

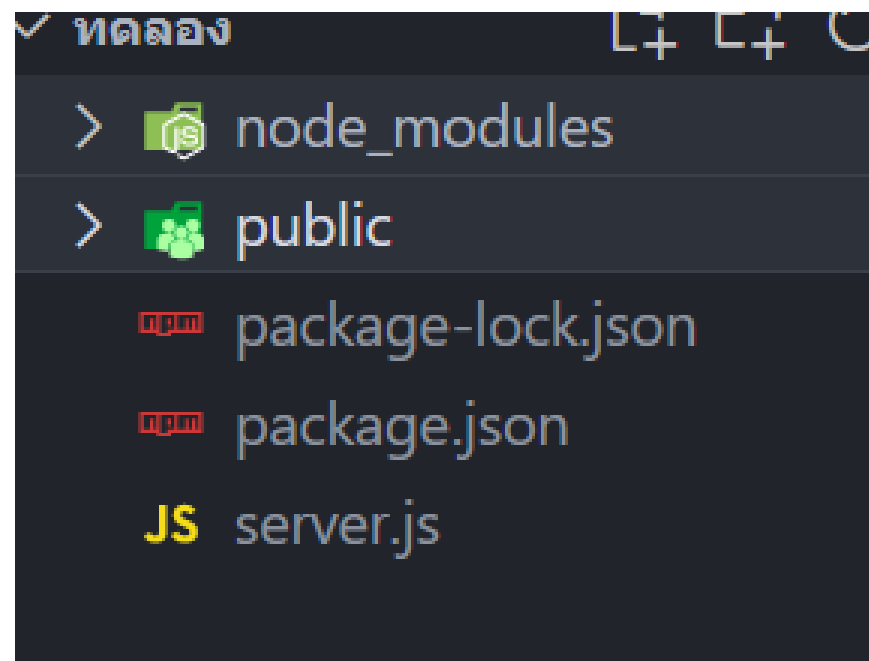
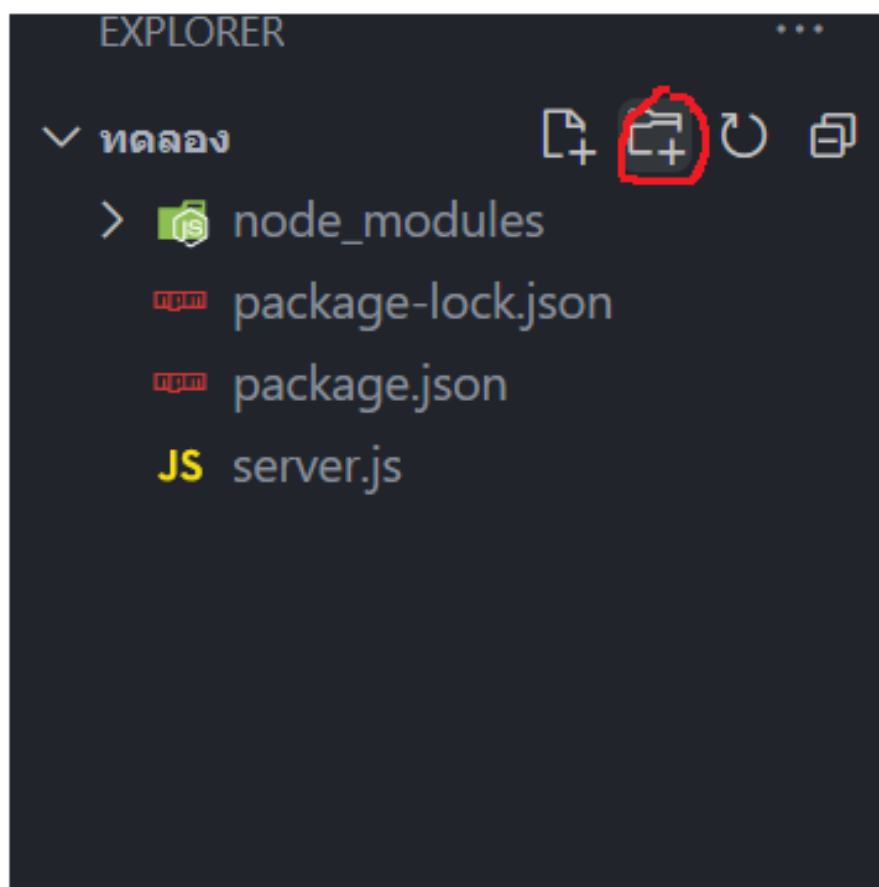
```

const express = require('express');
const bodyParser = require('body-parser');
const app = express();
let userGoal = 'Learn Docker!';
app.use(
  bodyParser.urlencoded({
    extended: false,
  })
);
app.use(express.static('public'));
app.get('/', (req, res) => {
  res.send(`
    <html>
      <head>
        <link rel="stylesheet" href="styles.css">
      </head>
      <body>
        <section>
          <h2>My Course Goal</h2>
          <h3>${userGoal}</h3>
        </section>
        <form action="/store-goal" method="POST">
          <div class="form-control">
            <label>Course Goal</label>
            <input type="text" name="goal">
          </div>
          <button>Set Course Goal</button>
        </form>
      </body>
    </html>
  `);
});
app.post('/store-goal', (req, res) => {
  const enteredGoal = req.body.goal;
  console.log(enteredGoal);
  userGoal = enteredGoal;
  res.redirect('/');
});
app.listen(80);

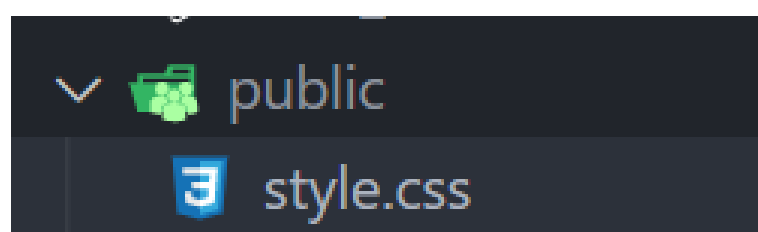
```

```
JS server.js ×
JS server.js > app.get('/') callback
1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const app = express();
4  let userGoal = 'Learn Docker!';
5  app.use(
6    bodyParser.urlencoded({
7      extended: false,
8    })
9  );
10 app.use(express.static('public'));
11 app.get('/', (req, res) => {
12   res.send(`
13     <html>
14       <head>
15         <link rel="stylesheet" href="style.css">
16       </head>
17       <body>
18         <section>
19           <h2> หน้ดี </h2>
20           <h3>${userGoal}</h3>
21         </section>
22         <form action="/store-goal" method="POST">
23           <div class="form-control">
24             <label>Course Goal</label>
25             <input type="text" name="goal">
26           </div>
27           <button>Set Course Goal</button>
28         </form>
29       </body>
30     </html>
31   `);
32 });
33 app.post('/store-goal', (req, res) => {
34   const enteredGoal = req.body.goal;
35   console.log(enteredGoal);
36   userGoal = enteredGoal;
```

จากนั้น ให้ กด สร้าง Folder ⇒



ข้างใน folder public ให้สร้าง ไฟล์ **style.css**



ในไฟล์ css ให้ เขียนโค้ดตามนี้ ⇒

```
html {
  font-family: sans-serif;
}

body {
  margin: 0;
}

section,
form {
  padding: 1rem;
  border-radius: 12px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.26);
  margin: 2rem auto;
  max-width: 40rem;
}

.form-control {
  margin: 0.5rem 0;
}

input {
  font: inherit;
```

```

}

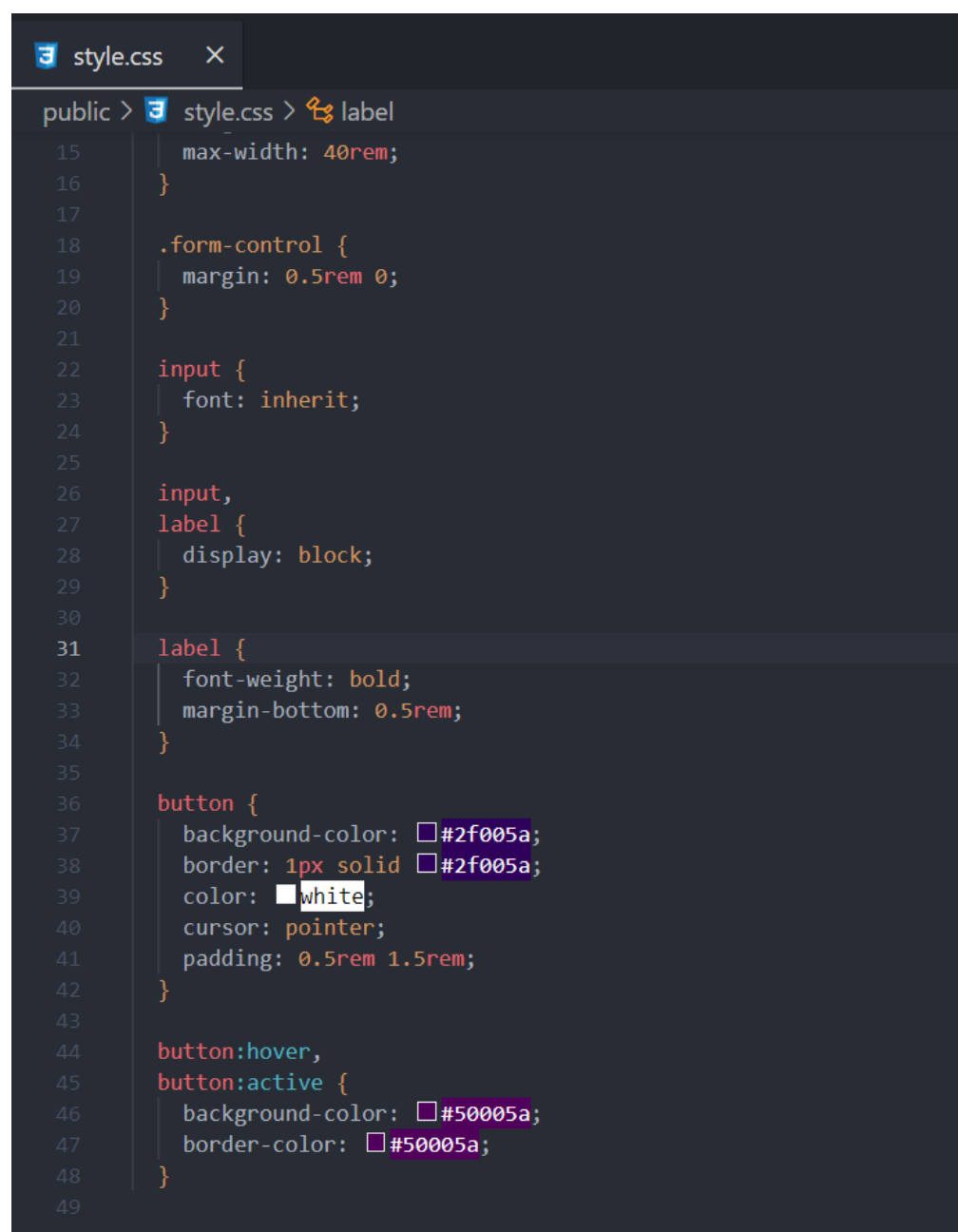
input,
label {
  display: block;
}

label {
  font-weight: bold;
  margin-bottom: 0.5rem;
}

button {
  background-color: #2f005a;
  border: 1px solid #2f005a;
  color: white;
  cursor: pointer;
  padding: 0.5rem 1.5rem;
}

button:hover,
button:active {
  background-color: #50005a;
  border-color: #50005a;
}

```



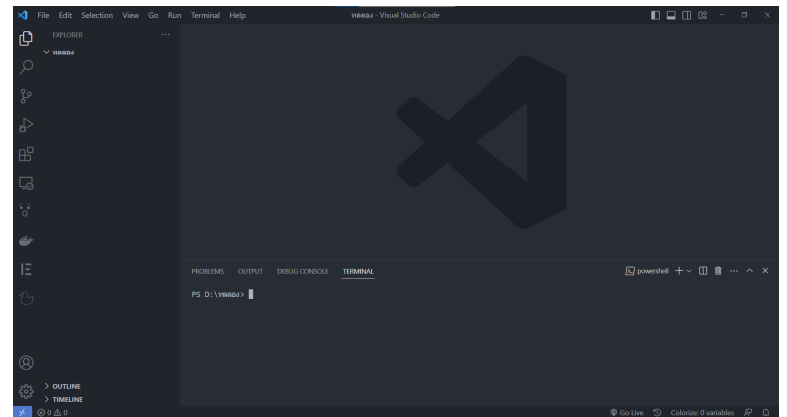
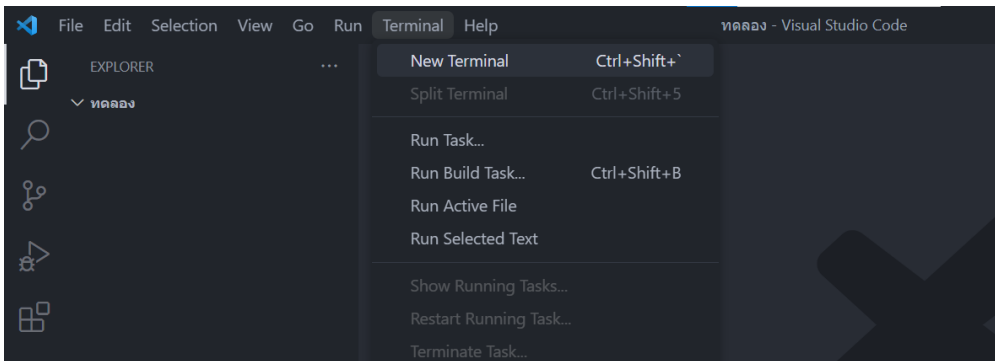
```

style.css
public > style.css > label
15 |   max-width: 40rem;
16 | }
17 |
18 | .form-control {
19 |   margin: 0.5rem 0;
20 | }
21 |
22 | input {
23 |   font: inherit;
24 | }
25 |
26 | input,
27 | label {
28 |   display: block;
29 | }
30 |
31 | label {
32 |   font-weight: bold;
33 |   margin-bottom: 0.5rem;
34 | }
35 |
36 | button {
37 |   background-color: #2f005a;
38 |   border: 1px solid #2f005a;
39 |   color: white;
40 |   cursor: pointer;
41 |   padding: 0.5rem 1.5rem;
42 | }
43 |
44 | button:hover,
45 | button:active {
46 |   background-color: #50005a;
47 |   border-color: #50005a;
48 | }
49 |

```


ต่อมาเราจะมาทำในส่วน ของ Docker และ node

ให้เปิด ตัว Terminal มา :



พิมพ์ **docker --version** เพื่อ ตรวจสอบว่า เจ้า Docker มัน ทำงาน ใน vs code ไหม

ถ้าติด ⇒

```
PS D:\ทดลอง> docker --version
Docker version 24.0.2, build cb74dfc
PS D:\ทดลอง>
```

พิมพ์ node ตามชื่อไฟล์ ⇒ คือ เราตั้งชื่อ ไฟล์ ชื่อ ว่า server ใช้ไหม ก็ พิมพ์ node server.js ไปใช้เพื่อ ตรวจสอบว่า ตัว node มันติดไหม

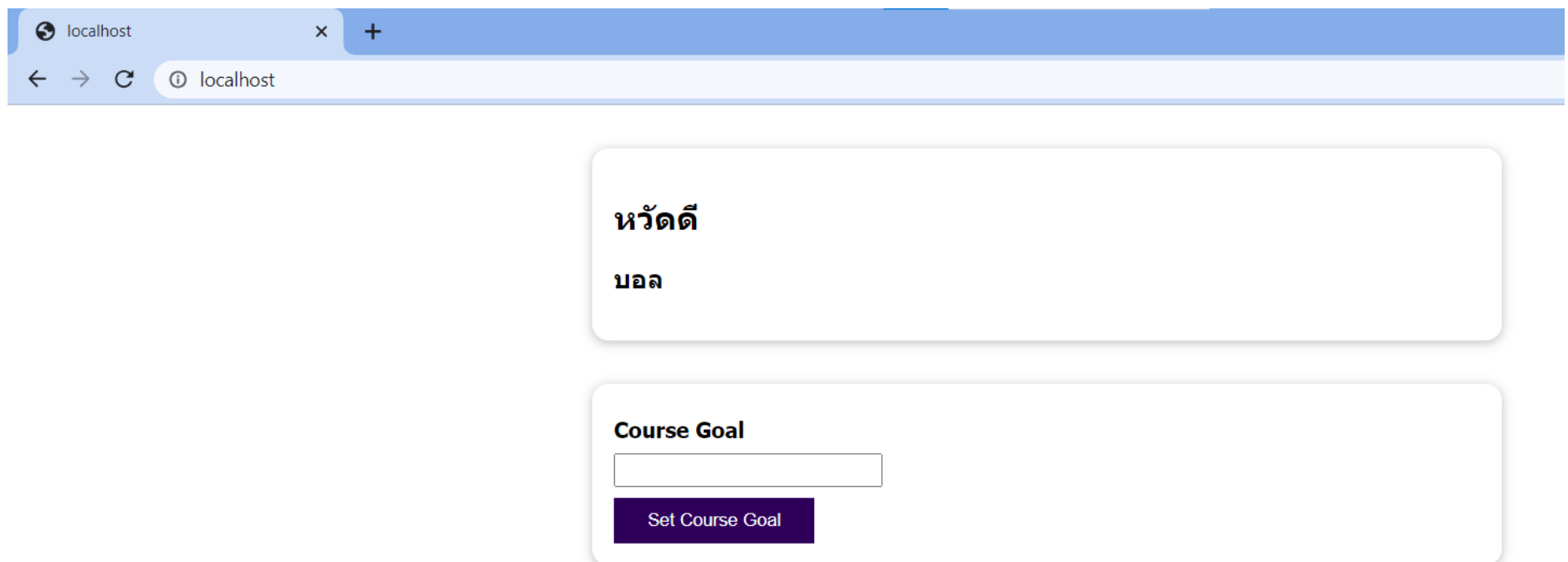
ถ้าติด มันก็ เหมือนจะ ค้าง พิมพ์ ไร ไม่ได้ ไม่ขึ้นอะไรเลย กด enter ไม่ได้ แสดงว่าถูกทางละ

```
Docker version 24.0.2, build cb74dfc
PS D:\ทดลอง> node server.js
```

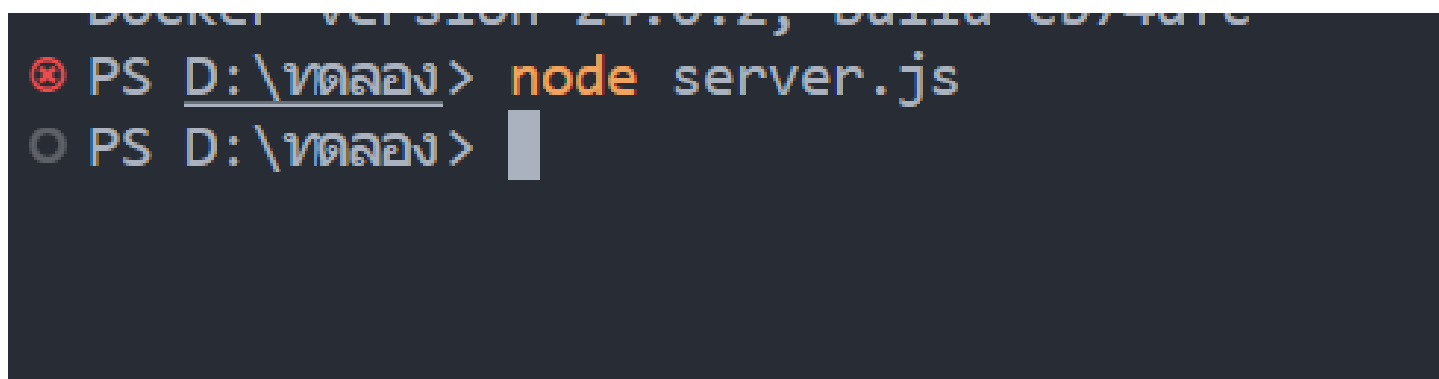
ให้เรา ไป ที่ browser ของเรา เช่น ไปที่ google

แล้ว พิมพ์ ตรง ช่อง url \Rightarrow localhost

ก็จะขึ้นเว็บที่เราเขียน ออกมา



หลังจาก นั้น ให้เรา กด ctrl + c เพื่อ ออก จากการทำงาน ก็จะกลับมาเป็น ปกติ



เว็บก็จะปิดไป

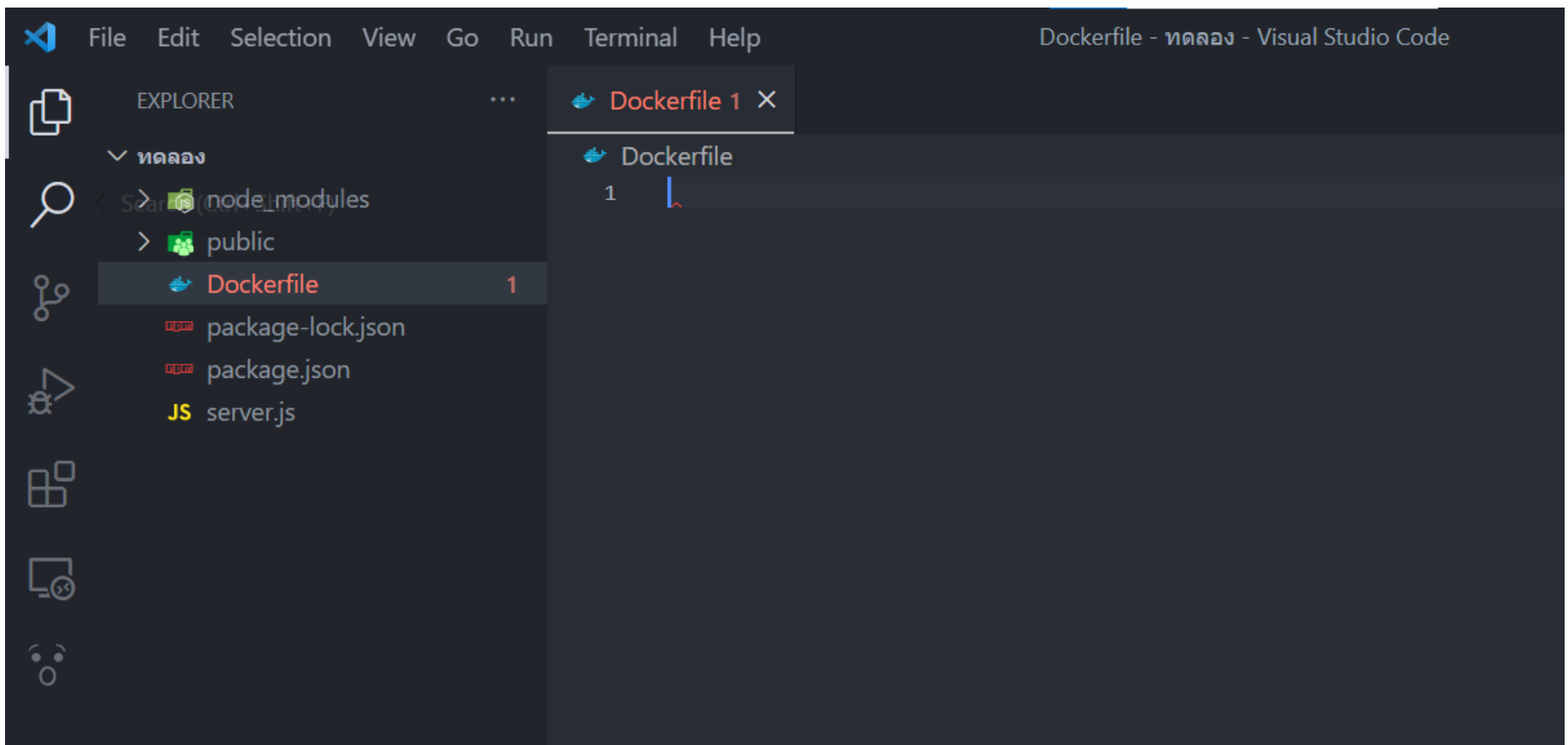
เท่านี้ก็ ดูได้แล้ว ว่า node ทำงาน เรามาดู docker ต่อดีกว่า

อันนี้สำคัญ

ให้สร้างไฟล์ Docker ตั้งชื่อ ไฟล์ว่า Dockerfile

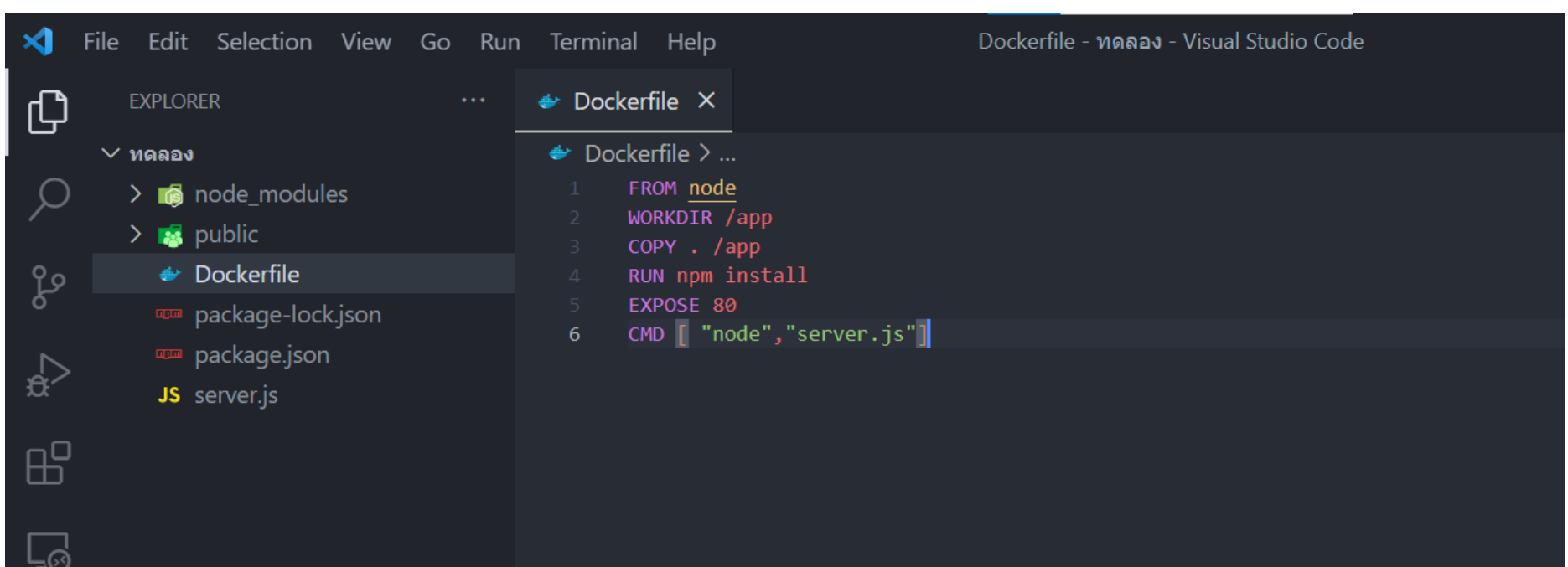
*** ย้ำเลย ***

ให้ พิมพ์ ตามนี้ ให้ถูกๆ นะ ไม่ขึ้นรันไม่ได้

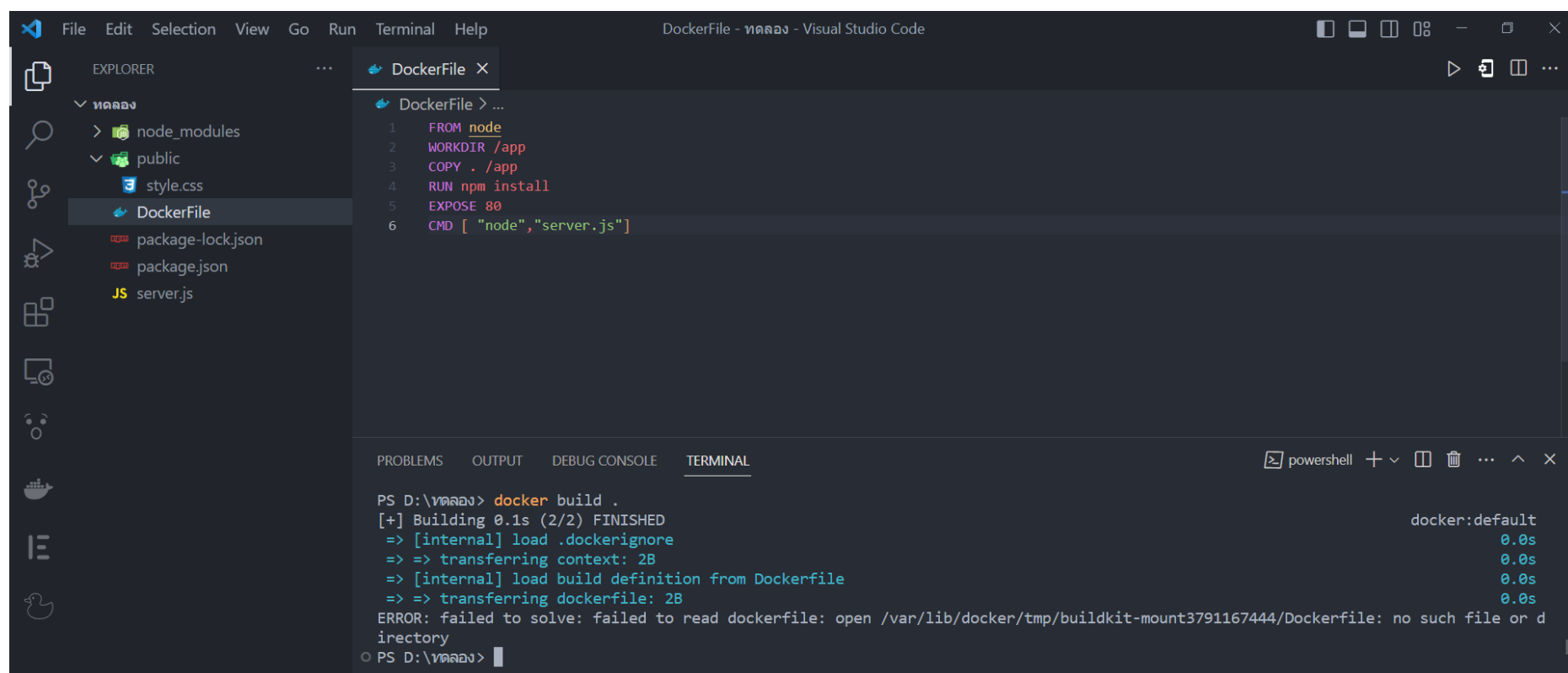


ข้างในให้พิมพ์ โค้ดตามนี้ นะ :

```
FROM node
WORKDIR /app
COPY . /app
RUN npm install
EXPOSE 80
CMD [ "node","server.js"]
```



ถ้า เราพิมพ์ ชื่อ Dockerfile ผิด จะเป็น แบบนี้ :

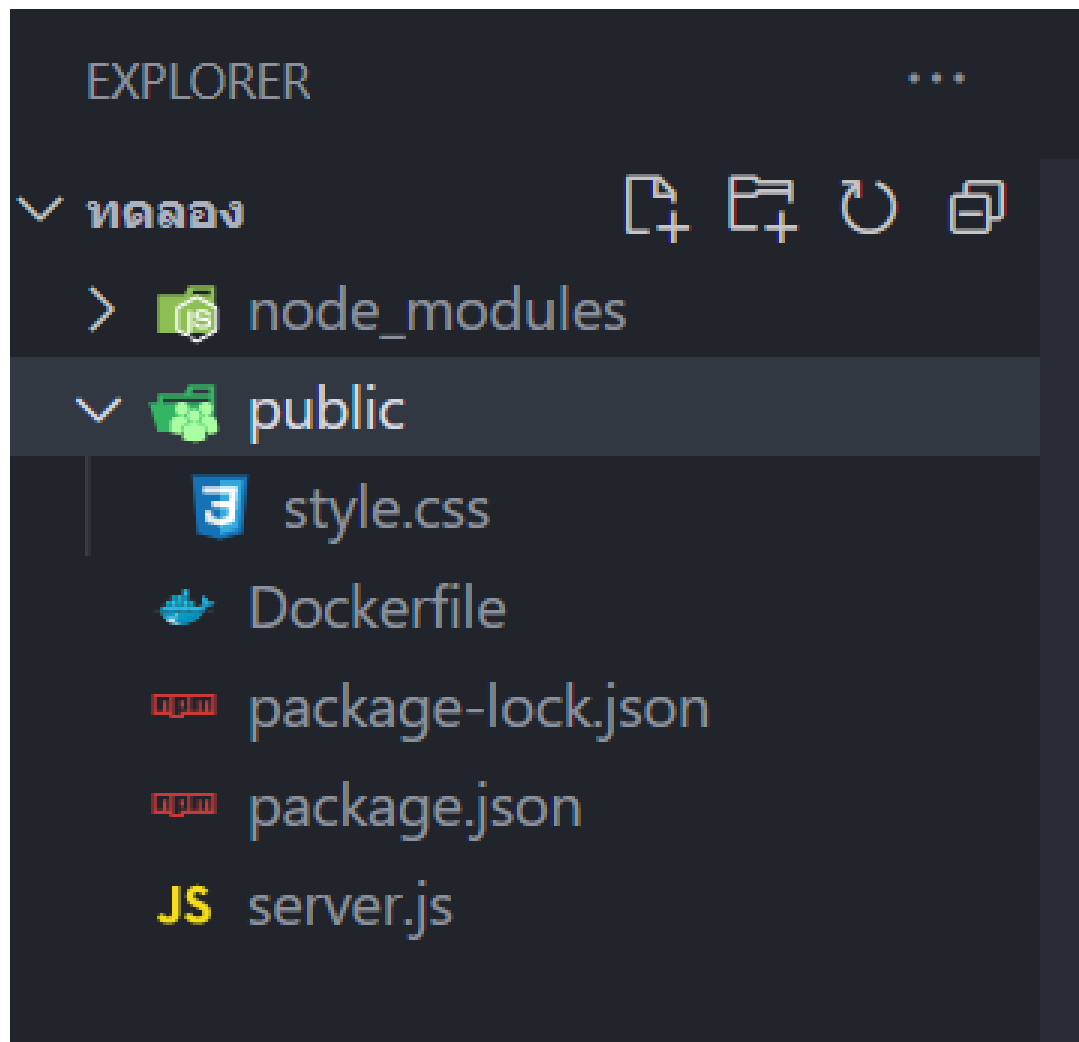


เราพิมพ์ เป็น ตัว F ใหญ่ก็ไม่ได้ ให้ แก้ โดย การ เปลี่ยนชื่อเป็น Dockerfile

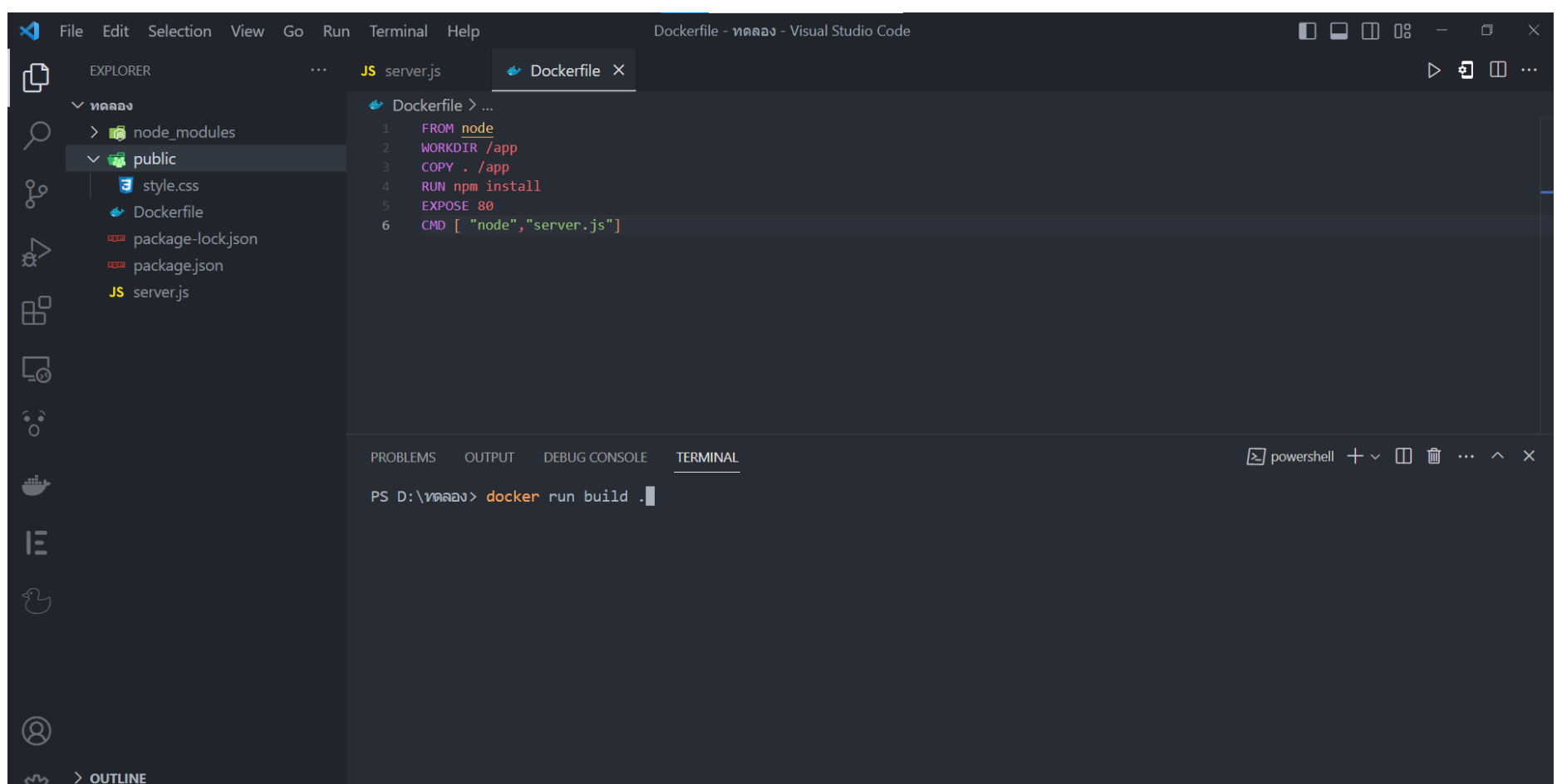
ต่อมา ทำการ build มันขึ้นมา เป็น images โดยการ พิมพ์ ในช่อง Terminal ว่า :

Docker run build .

***** ย้ำ ว่า โครงสร้างไฟล์ เป็นแบบนี้ละ *****



มาทำการ build



แล้ว กด enter

ถ้าใครขึ้นแบบนี้

```
PS D:\ทดลอง> docker run build .
Unable to find image 'build:latest' locally
docker: Error response from daemon: pull access denied for build, repository does not exist or may require 'docker login': denied:
requested access to the resource is denied.
See 'docker run --help'.
PS D:\ทดลอง>
```

ให้ออก vs code เข้าใหม่ หรือ login docker ใหม่ใน โปรแกรม

เมื่อ กด รัน จะได้หน้าแบบนี้

```
PS D:\ทดลอง> docker build .
[+] Building 47.7s (5/9)
=> => transferring context: 2.11MB
=> [1/4] FROM docker.io/library/node@sha256:b3ca7d32f0c12291df6e45a914d4ee60011a3fce4a978df5e609e356a4a2cb88
=> => resolve docker.io/library/node@sha256:b3ca7d32f0c12291df6e45a914d4ee60011a3fce4a978df5e609e356a4a2cb88
=> => sha256:b098c9ebef91eecac0c61865ac2f8fc639a6dc91662be1a8000ebdaa510ec836 7.24kB / 7.24kB
=> => sha256:d52e4f012db158bb7c0fe215b98af1facaddcbaee530efd69b1bae07d597b711 49.55MB / 49.55MB
=> => sha256:b3ca7d32f0c12291df6e45a914d4ee60011a3fce4a978df5e609e356a4a2cb88 1.21kB / 1.21kB
=> => sha256:cb6b3b7ccb94f3f4b071aef1e4452f82baea3e5442d8df00316b069d274b04c6 2.00kB / 2.00kB
=> => sha256:7dd206bea61ff3e3b54be1c20b58d8475ddd6f89df176146ddb7a2fd2c747ea2 19.92MB / 24.03MB
=> => sha256:2320f9be4a9c605d1ac847cf67cec42b91484a7cf7c94996417a0c7c316deadc 64.11MB / 64.11MB
=> => sha256:6e5565e0ba8dfce32b9049f21ceeb212946e0bb810d94cbd2db94ca61082f657 211.00MB / 211.00MB
=> => extracting sha256:d52e4f012db158bb7c0fe215b98af1facaddcbaee530efd69b1bae07d597b711
=> => sha256:5f1526a28cf91707a0af45c5624979f67215b1330efd66c05a0f94839c96f0bc 3.37kB / 3.37kB
=> => sha256:b9c7405b482f54514258ccb5082cd81cfb44b1c17eb724d00abaa2abb579458 47.49MB / 47.49MB
=> => sha256:9db0bc99587b3093ef483ee42f3eb21728d02f47bf77a8009fcf2c155b43a95a 2.27MB / 2.27MB
=> => sha256:8e1c8c1907a57a97428736f263d852d6b3c80a69f3162661650d3d91e43b4150 450B / 450B
```

อาจจะใช้เวลานาน น้อย ขนาด 1 GB ต่อ การ build ไฟล์

```
=> => exporting layers
=> => writing image sha256:9d0496df15d4262d0e66035c2a750475657792f471a58221a8f541e93f2f1fc8

What's Next?
View summary of image vulnerabilities and recommendations → docker scout quickview
PS D:\ทดลอง>
```

นี่คือเราสร้าง images เสร็จแล้ว ให้ จำเลขหน้า 3 ตัว ที่ขีดเส้นให้ติ

อันนี้ของเราจะเป็น เลข 9d0 เลขของ แต่ละคนไม่เหมือนกันนะ ไป ดูของ ใครของมันนะ

ต่อมาเรา ดูว่ามันสร้าง images ได้จริงๆ ไหม ดู โดยการ พิมพ์ **docker images**

```
PS D:\ทดลอง> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
<none> <none> 9d0496df15d4 3 minutes ago 1.1GB
```

นี้ก็ขึ้นมาแล้ว

จากนี้เราก็จะสั่งให้มัน รัน ตัวมันกลายเป็น container ที่พร้อมใช้งาน โดยการพิมพ์ :

docker run -d -p 3000:80 ตามด้วย 3 ตัวหน้า ที่เราบอก

เช่นของเรา ⇒ **docker run -d -p 3000:80 9d0**

แล้วกด enter เลย

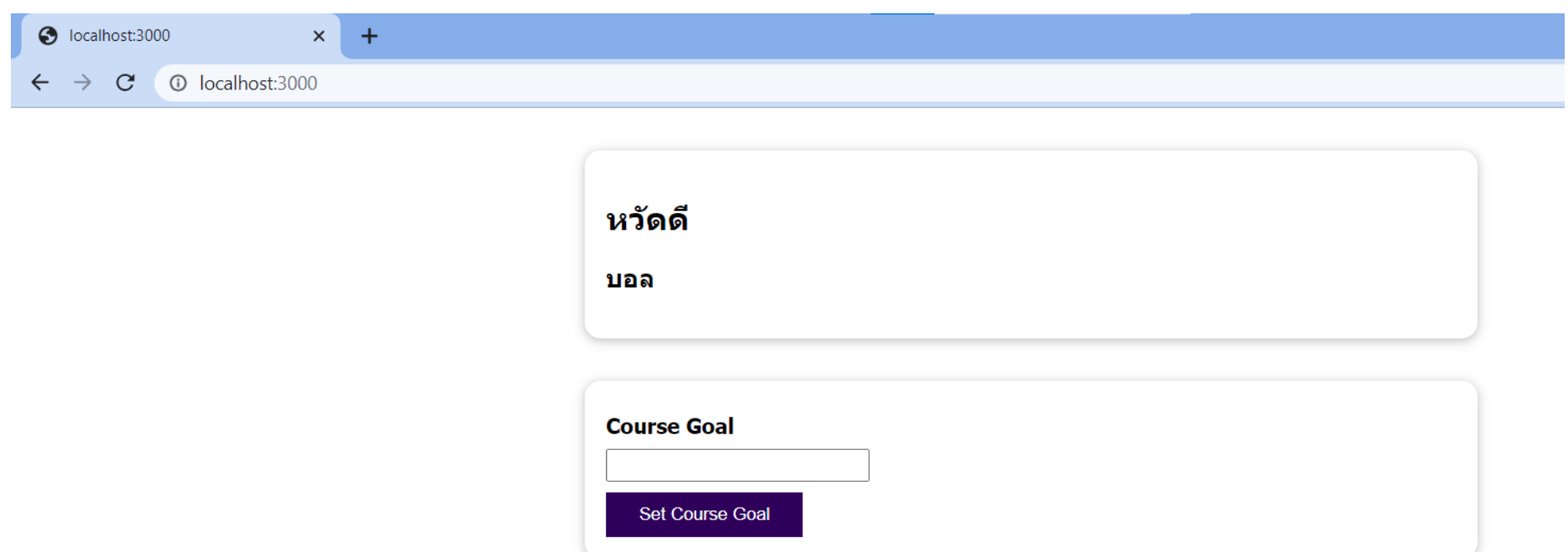
```
PS D:\ทดลอง> docker run -d -p 3000:80 9d0
2eee2f0e3cbdb3cb4cb56cfad259d53bbd31a6e08fa377035a21600ad4eb065
PS D:\ทดลอง>
```

นี่คือ รัน เรียบร้อยละ ต่อมาเราจะดูว่ามัน ติดไหม โดยการพิมพ์ :

docker ps

```
PS D:\ทดลอง> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
2eee2f0e3cbe   9d0       "docker-entrypoint.s..." 50 seconds ago Up 49 seconds 0.0.0.0:3000->80/tcp    priceless_poitras
```

นี่คือ ว่า มัน ติด แล้ว ต่อ มาเรา ก็ไปที่ google เลย ไป พิมพ์ ตรง url ว่า localhost:3000 ได้เลย



localhost:3000

หัวข้อดี

บอล

Course Goal

Set Course Goal

***** ขอรืบาย ตรง *****

-p คือ การกำหนด port ที่เราจะไป แสดงใน หน้าเว็บ

เอาง่ายๆ คือ : เราจะ เอาเลขอะไรพิมพ์ บนเว็บ แล้ว แสดง เว็บเรา เช่น อย่างที่เรา เขียน 3000 ใช้ไหม ก็คือ เรา จะ พิมพ์ localhost:3000 เพื่อเปิดเว็บของเรา หรือ คนอื่นๆ ก็สามารถแก้ไขเลขเป็นอะไรก็ได้ เช่น 1000 หรือ 3001 หรือ อื่นๆ แก้ได้ ตรงนี้ เท่านั้น นะ

พอเวลาเปิด เว็บมาให้ พิมพ์ตามเลขที่เรา กำหนดไว้ ส่วนเลข :80 ไม่ต้องไปแก้มัน เพราะมันคือค่า ที่ กำหนด ไว้แล้ว ถ้าแก้มันจะทำงานไม่ได้

-p ตรงนี้แก้ได้(อันนี้คือเลขที่เราจะใส่ตามต้องการ):80

-d คือ เราจะให้ มัน รัน ใน background ไม่ต้องมาแสดงๆ ตรงหน้าต่าง terminal

เอาง่ายๆ เหมือน : **เราไม่อยากเจอ หน้าเธอ** แต่ขอให้ เธอรักเราต่อไป

(ถึงไม่เห็น มัน แต่มันยังคงทำงานต่อไป เหมือนเดิม)



ตามติดชีวิตนักศึกษา
@DekmahalaiTH

ขณะที่หลงใหลในวันนั้น คือขณะที่
ทรมานชิปหายในวันนี้ 5555 #ตาม
ติดชีวิตนักศึกษา

19:08 หลังเที่ยง · 16 ก.ค. 23



ตามติด
ชีวิต
นักศึกษา

เรียนแบบไม่รู้เรื่องอะไรเลย จะถามก็ไม่ได้
เพราะกูไม่รู้ด้วยซ้ำ ว่ากูไม่รู้อะไร



BIGHIT
MUSIC

3:47

정국 (Jung Kook) 'Seven (feat. Latto)' Official
MV

การดู 75 ล้าน ครั้ง • 5 วันที่ผ่านมา

 HYBE LABELS ✓

정국 (Jung Kook) 'Seven (feat. Latto)' Official MV 'Seven (feat. Latto)' Release
Listen to 'Seven': <https://ingrv.es/Seven Credits: ...>

ใหม่ คำบรรยาย

<https://www.youtube.com/watch?v=QU9c0053UAU>