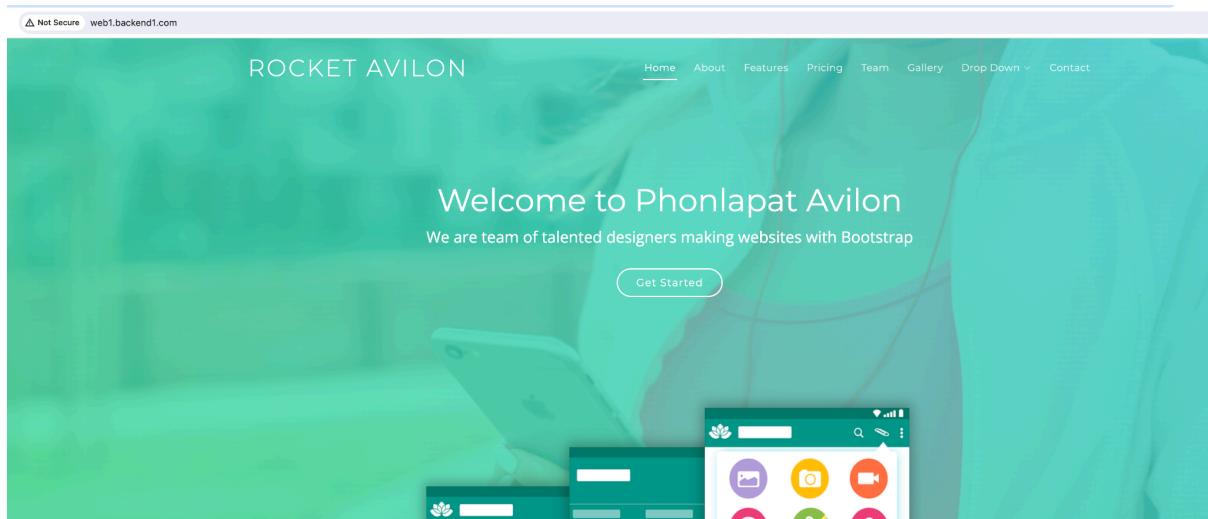


LAB 3 Step 6-11

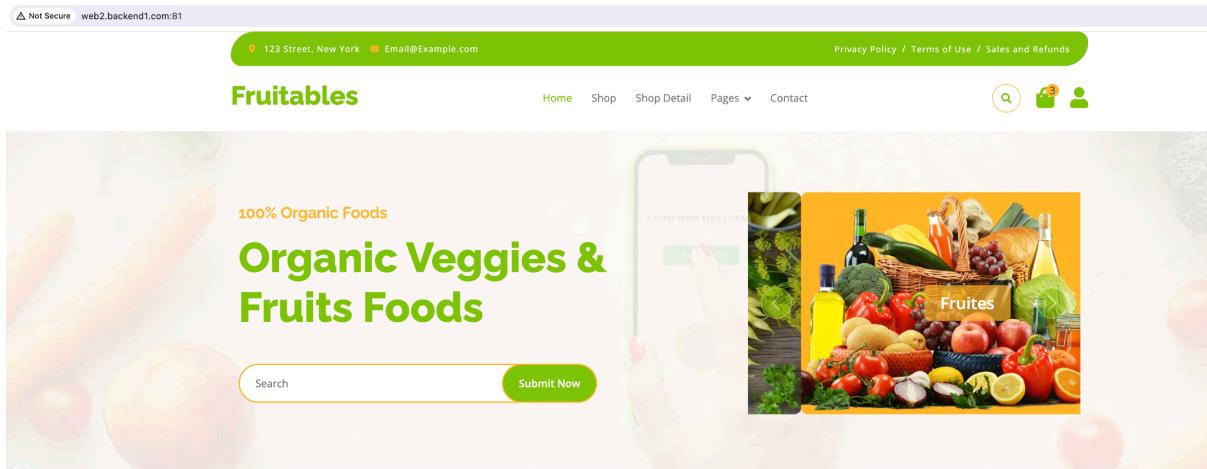
```
logged in.  
devuser@backend1-66543210020:~$ su phonlapat  
Password:  
To run a command as administrator (user "root"), use  
See "man sudo_root" for details.  
  
phonlapat@backend1-66543210020:/home/devuser$
```

```
devuser@backend2-66543210020:~$ su phonlapat  
Password:  
To run a command as administrator (user "root"), use "sudo <command>  
See "man sudo_root" for details.  
  
phonlapat@backend2-66543210020:/home/devuser$
```

เปลี่ยนชื่อ host , vm และ เพิ่ม user phonlapat



<http://web1.backend1.com/80>



<http://web2.backend1.com/81>

```
● phonlapat@backend1-66543210020:/var/www/nginx$ sudo mkdir react1
● phonlapat@backend1-66543210020:/var/www/nginx$ sudo mkdir react2
```

```
● phonlapat@backend1-66543210020:/var/www/nginx$ sudo chown -R $USER /var/www/nginx/react1
● phonlapat@backend1-66543210020:/var/www/nginx$ sudo chown -R $USER /var/www/nginx/react2
● phonlapat@backend1-66543210020:/var/www/nginx$
```

สร้างโฟลเดอร์ใหม่แล้วเปลี่ยน Owner

```
root@backend1-66543210020:~# ufw allow 3010
Rule added
Rule added (v6)
root@backend1-66543210020:~# ufw allow 3011
Rule added
Rule added (v6)
```

เพิ่ม port 3010 , 3011 ให้ใช้ได้

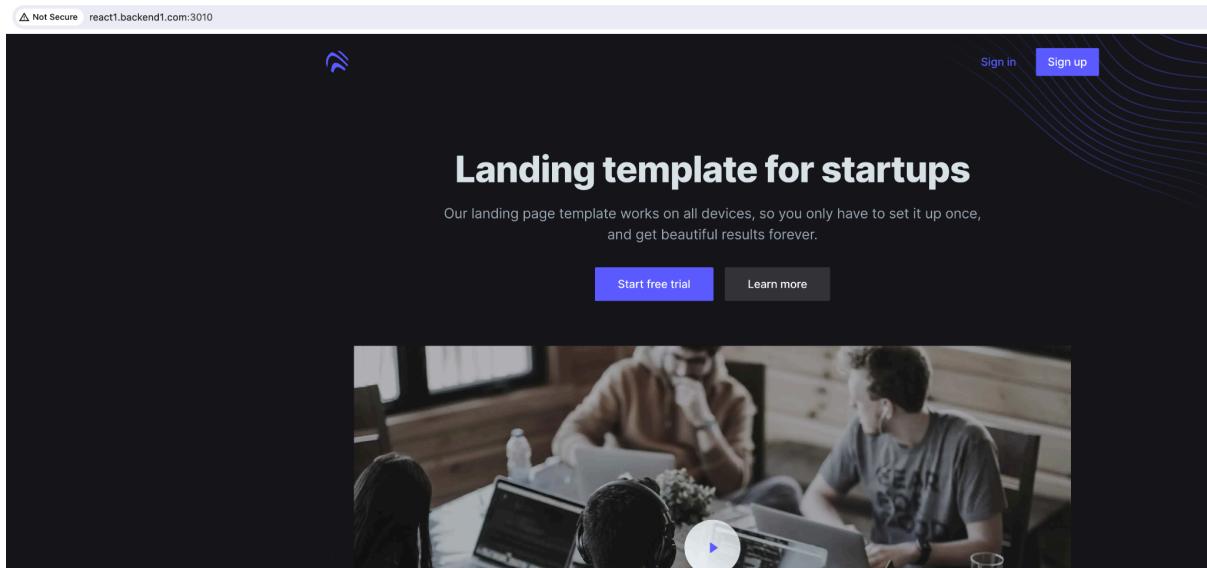
```
{  
  "name": "open-pro-next",  
  "version": "0.1.0",  
  "private": true,  
  "scripts": {  
    "dev": "next dev",  
    "build": "next build",  
    "start": "next start -p 3010",  
    "lint": "next lint"  
  },  
  "homepage": "https://demos.creative-tim.com/argon-",  
  "scripts": {  
    "start": "PORT=3011 react-scripts start",  
    "build": "react-scripts build",  
    "test": "react-scripts test".  
},
```

เปลี่ยนพอร์ตของ template ที่โหลดมา

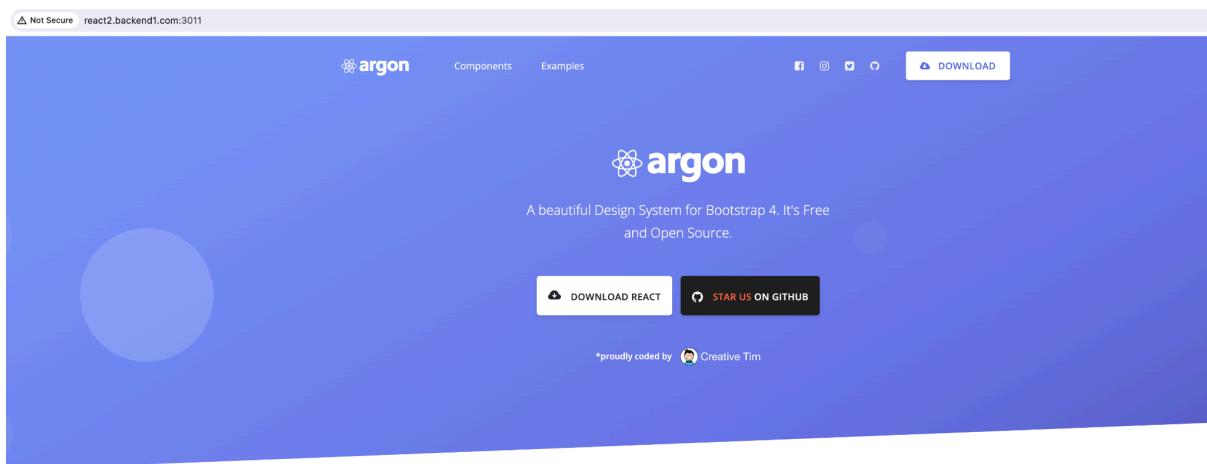
```
File: /etc/hosts
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost engse203.lab1  
255.255.255.255 broadcasthost  
::1            localhost  
10.211.55.4    myapp.se-rmutl.net expressjs-example.se-rmutl.net nodelogin.se-$  
10.211.55.4    my-react-app.se-rmutl.net ubuntu-next-app.se-rmutl.net  
10.211.55.4    weblogin.se-rmutl.net web1.backend1.com  
10.211.55.4    web2.backend1.com react1.backend1.com  
10.211.55.6    backend_server_ip_address api.midterm-engse202-020.com
```

```
File: /etc/hosts Modified
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost engse203.lab1  
255.255.255.255 broadcasthost  
::1            localhost  
10.211.55.4    myapp.se-rmutl.net expressjs-example.se-rmutl.net nodelogin.se-$  
10.211.55.4    my-react-app.se-rmutl.net ubuntu-next-app.se-rmutl.net  
10.211.55.4    weblogin.se-rmutl.net web1.backend1.com  
10.211.55.4    web2.backend1.com react1.backend1.com react2.backend1.com  
10.211.55.4    backend_server_ip_address api.midterm-engse202-020.com
```

ใส่ domain ให้เครื่อง



<http://react1.backend1.com/3010>



Basic Elements

Buttons

[BUTTON](#) [WITH ICON](#) [ICON](#)

<http://react2.backend1.com/3011>

```
root@backend1-66543210020:~# ufw allow 4040  
Rule added  
Rule added (v6)
```

```
root@backend1-66543210020:~# ufw allow 3012  
Rule added  
Rule added (v6)
```

เพิ่ม port 4040 , 3012

```
JS index.js ●
lab4 > server > JS index.js > ...
1 const express = require("express");
2 const mysql = require("mysql2");
3 const cors = require("cors");
4
5 const app = express();
6
7
8 app.use(express.json());
9 app.use(
10   cors({
11     origin: ["http://api.backend2.com:4040/"],
12     methods: ["GET", "POST"],
13     credentials: true,
14   })
15 );
16
17 app.get('/', (req, res) => {
18   console.log("Hello from API");
19   res.json({message: 'Phonlapat Suanghiran'});
20 });
21
22
23
24 app.listen(4040, () => {
25   console.log("running server");
26 });
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Hello from API
Hello from API
^C
○ phonlapat@backend2-66543210020:/var/www/lab4/server$ node index.js
running server
Hello from API
```

เข้า VScode ลง remote เข้า เครื่อง 10.211.55.6
ใส่โค้ดแล้วลองทดสอบกับ Postman

GET localhost:4040

localhost:4040

GET http://api.backend2.com:4040

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize HTML

```

1 <!DOCTYPE html>
2 <html lang="en">
3

```

Sending request...

Status: 404 Not Found Time: 12 ms Size: 400 B Save as example

Postman Agent

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "err": {
3     "code": "ER_DUP_ENTRY",
4     "errno": 1062,
5     "sqlState": "23000",
6     "sqlMessage": "Duplicate entry '' for key 'users.PRIMARY'",
7     "sql": "INSERT INTO users (username, password) VALUES (?,?)"
8   }
9 }

```

Status: 200 OK Time: 16 ms Size: 471 B Save as example

Update Postman Agent

body: {"username": "", "password": ""}
username:
password:

ทดสอบ register

กลับไปที่เครื่อง backend1 เข้าไปสร้าง database ดังรูป

```

mysql> CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 3 warnings (0.01 sec)

mysql> USE loginsystem;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE TABLE IF NOT EXISTS `users`(
    ->   `username` varchar(50) NOT NULL,
    ->   `password` varchar(500) NOT NULL,
    ->   PRIMARY KEY (`username`)
    -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected, 2 warnings (0.00 sec)

mysql> INSERT INTO `users`(`username`, `password`) VALUES ('test', 'test');
ERROR 1062 (23000): Duplicate entry 'test' for key 'users.PRIMARY'
mysql> select * from users;
+-----+-----+
| username | password |
+-----+-----+
| test     | test      |
+-----+-----+
1 row in set (0.00 sec)

```

The screenshot shows a VS Code interface connected to a remote host at 10.211.55.4. The terminal window displays MySQL commands to create a database and table, and insert data. The code editor shows a React component (App.js) with logic for user registration and login, utilizing Axios for API calls.

```

EXPLORER          ... JS App.js X JS index.js JS App.test.js package.json
www [SSH: 10.211.55.4]  App.js  App  login
10.211.55.4
apache
expressjs-example
html
lab2
lab3
lab4
myapp
nginx
client
build
node_modules
public
src
.gignore
package-lock.json
package.json
README.md
Part-1-User-registration-and-log...
react1
react2
web2

ngin > client > src > JS App.js > App.js > App > login
1 import React, { useEffect, useState } from 'react';
2 import Axios from 'axios';
3 import './App.css';
4
5 function App() {
6
7   const [usernameReg, setUsernameReg] = useState("");
8   const [passwordReg, setPasswordReg] = useState("");
9
10  const [username, setUsername] = useState("");
11  const [password, setPassword] = useState("");
12
13  const [loginStatus, setLoginStatus] = useState("");
14
15  const register = () => {
16    Axios.post("http://api.backend2.com:4040/register", {
17      username: usernameReg,
18      password: passwordReg,
19    }).then((response) => {
20      console.log(response);
21
22      if (!response.data.err) {
23        setLoginStatus(`${usernameReg} was inserted.`);
24      } else {
25        setLoginStatus(response.data.err.sqlMessage);
26      }
27
28    });
29  };
30
31  const login = () => {
32    Axios.post("http://api.backend2.com:4040/login", {
33      username: username,
34      password: password,
35    }).then((response) => {
36      console.log(response);
37      //console.log(`"response.data.message": ${response.data[0].message}`);
38
39      if (!response.data.message) {
40        setLoginStatus(`${username} was logged in.`);
41      } else {
42        setLoginStatus(response.data.message);
43      }
44
45    });
46  };

```

เข้า VScode remote เข้าเครื่อ 10.211.55.4
ใส่โค้ดและติดตั้ง package

```
JS App.js | JS index.js | JS App.test.js | {} package.json X

nginx > client > {} package.json > {} scripts > start

1  {
2    "name": "client",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.17.0",
7      "@testing-library/react": "^13.4.0",
8      "@testing-library/user-event": "^13.5.0",
9      "axios": "^1.6.5",
10     "cors": "^2.8.5",
11     "react": "^18.2.0",
12     "react-dom": "^18.2.0",
13     "react-scripts": "5.0.1",
14     "web-vitals": "^2.1.4"
15   },
16   ▶ Debug
17   "scripts": {
18     "start": "PORT=3012 react-scripts start",
19     "build": "react-scripts build",
20     "test": "react-scripts test",
21     "eject": "react-scripts eject"
22   },
23   "eslintConfig": {
24     "extends": [
25       "react-app",
26       "react-app/jest"
27     ],
28     "browserslist": {
29       "production": [
30         ">0.2%",
31         "not dead",
32         "not op_mini all"
33       ],
34       "development": [
35         "last 1 chrome version",
36         "last 1 firefox version",
37         "last 1 safari version"
38       ]
39     }
40   }
41 }
```

เปลี่ยน port เป็น 3012 และเริ่มทดสอบ

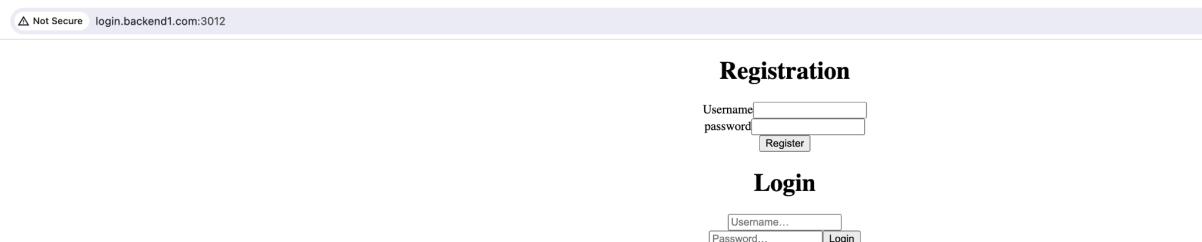
Compiled successfully!

You can now view **client** in the browser.

Local: http://localhost:3012
On Your Network: http://10.211.55.4:3012

Note that the development build is not optimized.
To create a production build, use **npm run build**.

webpack compiled **successfully**



A screenshot of a web browser window. The address bar shows "Not Secure login.backend1.com:3012". The page contains two forms: a "Registration" form with fields for "Username" and "password", and a "Login" form with fields for "Username..." and "Password...", and a "Login" button.

Registration

Username
password

Register

Login

Username... Password... Login

<http://login.backend1.com:3012/>

```
○ phonlapat@backend2-66543210020:/var/www/lab4/server$ nodemon index.js
[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
running server
```



<http://api.backend2.com:4040/>

A screenshot of the VS Code terminal interface. The tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal output shows the command "nodemon index.js" being run, along with the logs from nodemon and the Node.js application. The application logs include "Hello from API" and an empty square bracket "[]".

```
phonlapat@backend2-66543210020:/var/www/lab4/server$ nodemon index.js
[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
running server
Hello from API
[]
```

ทดสอบลองใช้งาน

Registration

Username

password

Login

'test' was logged in.

```
○ phonlapat@backend2-66543210020:/var/www/lab4/server$ nodemon index.js
[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
running server
Hello from API
body: {"username":"test","password":"test"}
username: test
password: test
```

ทดสอบ Login

Registration

Username

password

Registration

Username

password

Login

'dev' was inserted.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
○ phonlapat@backend2-66543210020:/var/www/lab4/server$ nodemon index.js
[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
running server
Hello from API
body: {"username":"test","password":"test"}
username: test
password: test
body: {"username":"dev","password":"123"}
username: dev
password: 123
□
```

ทดสอบ register

```
1 { ...
2 ... "username": "API",
3 ... "password": "123"
4 }
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ↻

Status: 200 OK Time: 31 ms Size: 397 B Save as example

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "info": "",
6   "serverStatus": 2,
7   "warningStatus": 0,
8   "changedRows": 0
9 }
```

```
○ phonlapat@backend2-66543210020:/var/www/lab4/server$ nodemon index.js
[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
running server
Hello from API
body: {"username":"test","password":"test"}
username: test
password: test
body: {"username":"dev","password":"123"}
username: dev
password: 123
body: {"username":"API","password":"123"}
username: API
password: 123
□
```

ทดสอบ register จาก postman

```
mysql> USE loginsystem;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from users;
+-----+-----+
| username | password |
+-----+-----+
| API      | 123      |
| dev      | 123      |
| gay      | 123      |
| ket      | 112233   |
| rocket   | 123      |
| test     | test     |
| test123  | test123  |
| test2    | test2    |
| tester   | 123      |
+-----+-----+
10 rows in set (0.00 sec)
```

เช็คใน mysql ว่าข้อมูลเข้าใหม่