

Lab 4

Step 1

ตั้งชื่อให้ server ใหม่

```
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 backend014
```

```
GNU nano 6.2 /etc/hostname
backend014
```

ตั้งชื่อให้ server เดิมใหม่

```
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 webserver014
```

```
GNU nano 6.2 /etc/hostname
webserver014
```

Step 2

install MySQL server

```
devuser@backend014:~$ apt-cache search mysql-server
mysql-server - MySQL database server (metapackage depending on the latest version)
mysql-server-8.0 - MySQL database server binaries and system database setup
mysql-server-core-8.0 - MySQL database server binaries
default-mysql-server - MySQL database server binaries and system database setup (metapackage)
default-mysql-server-core - MySQL database server binaries (metapackage)
mariadb-server-10.6 - MariaDB database server binaries
mariadb-server-core-10.6 - MariaDB database core server files
devuser@backend014:~$ apt info -a mysql-server-8.0
Package: mysql-server-8.0
Version: 8.0.35-0ubuntu0.22.04.1
Priority: optional
Section: database
Source: mysql-8.0
Origin: Ubuntu
```

```
devuser@backend014:~$ sudo apt install mysql-server-8.0
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7
libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-parser-perl libhtml-tagset-perl
libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 libtimedate-perl liburi-perl
mecab-ipadic_mecab-ipadic-utf8_mecab-utils mysql-client-8.0 mysql-client-core-8.0
mysql-common mysql-server-core-8.0
```

Step 3

Install Nginx

```
devuser@backend014:~$ sudo apt update && sudo apt upgrade
[sudo] password for devuser:
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 Packages [1,104 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports jammy-updates/universe arm64 Packages [944 kB]
Fetched 2,278 kB in 4s (530 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  distro-info distro-info-data python3-distro-info python3-software-properties
  python3-update-manager software-properties-common update-manager-core
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
devuser@backend014:~$ sudo apt install nginx certbot python3-certbot-nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx-common nginx-core python3-acme
  python3-certbot python3-configargparse python3-icu python3-josepy python3-parsedatetime
```

```
no VM guests are running outdated hypervisor (qemu) binaries on
devuser@backend014:~$ sudo ufw allow 'Nginx Full'
Rules updated
Rules updated (v6)
```

Step 4

Nodejs Installation

```
devuser@backend014:~$ sudo apt-get install nodejs -y
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.
Need to get 30.4 MB of archives.
After this operation, 195 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_20.x nodistro/main arm64 nodejs arm64 2
Fetched 30.4 MB in 1s (27.7 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 78443 files and directories currently installed.)
Preparing to unpack .../nodejs_20.10.0-1nodesource1_arm64.deb ...
Unpacking nodejs (20.10.0-1nodesource1) ...
Setting up nodejs (20.10.0-1nodesource1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

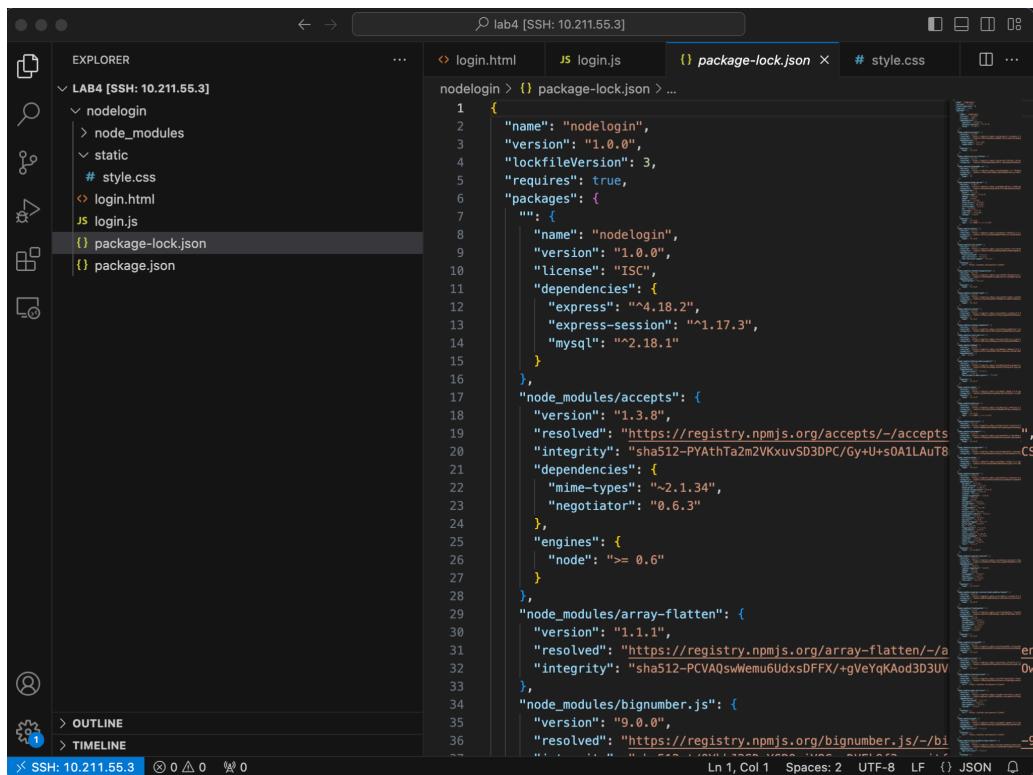
Running kernel seems to be up-to-date.
```

```
devuser@backend014:~$ node -v
v20.10.0
devuser@backend014:~$ npm -v
10.2.3
devuser@backend014:~$ 
```

Step 5

Create Login System with Node.js App

สร้างไฟล์ต่างๆใน VS Code



```
1  {
2   "name": "nodeLogin",
3   "version": "1.0.0",
4   "lockfileVersion": 3,
5   "requires": true,
6   "packages": [
7     {
8       "name": "nodeLogin",
9       "version": "1.0.0",
10      "license": "ISC",
11      "dependencies": {
12        "express": "^4.18.2",
13        "express-session": "^1.17.3",
14        "mysql": "~2.18.1"
15      }
16    },
17    "node_modules/accepts": {
18      "version": "1.3.8",
19      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
20      "integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+s0A1LauT8C5",
21      "dependencies": {
22        "mime-types": "~2.1.34",
23        "negotiator": "0.6.3"
24      },
25      "engines": {
26        "node": ">= 0.6"
27      }
28    },
29    "node_modules/array-flatten": {
30      "version": "1.1.1",
31      "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
32      "integrity": "sha512-PCVAQswWemu6UdxsDFX/+gVeYqKAod3d3UV0w",
33    },
34    "node_modules/bignumber.js": {
35      "version": "9.0.0",
36      "resolved": "https://registry.npmjs.org/bignumbe.js/-/bignumbe.js-9.0.0.tgz"
37    }
38  }
39  "node_modules/accepts": {
40    "version": "1.3.8",
41    "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
42    "integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+s0A1LauT8C5",
43    "dependencies": {
44      "mime-types": "~2.1.34",
45      "negotiator": "0.6.3"
46    },
47    "engines": {
48      "node": ">= 0.6"
49    }
50  },
51  "node_modules/accepts": {
52    "version": "1.3.8",
53    "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
54    "integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+s0A1LauT8C5",
55    "dependencies": {
56      "mime-types": "~2.1.34",
57      "negotiator": "0.6.3"
58    },
59    "engines": {
60      "node": ">= 0.6"
61    }
62  },
63  "node_modules/bignumbe.js": {
64    "version": "9.0.0",
65    "resolved": "https://registry.npmjs.org/bignumbe.js/-/bignumbe.js-9.0.0.tgz"
66  }
67 }
```

อนุญาต port 3000 ให้ใช้งานได้

```
devuser@webserver014:/var/www/lab4/nodeLogin$ sudo ufw status
[sudo] password for devuser:
Status: active
```

To	Action	From
--	-----	-----
OpenSSH	ALLOW	Anywhere
Apache	ALLOW	Anywhere
Nginx Full	ALLOW	Anywhere
3333	ALLOW	Anywhere
3000	ALLOW	Anywhere
3001	ALLOW	Anywhere
3002	ALLOW	Anywhere
3003	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)
Nginx Full (v6)	ALLOW	Anywhere (v6)
3333 (v6)	ALLOW	Anywhere (v6)
3000 (v6)	ALLOW	Anywhere (v6)
3001 (v6)	ALLOW	Anywhere (v6)
3002 (v6)	ALLOW	Anywhere (v6)
3003 (v6)	ALLOW	Anywhere (v6)

สร้าง Database

```
devuser@webserver014:/var/www/lab4/nodelogin$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'example_user'@'%' IDENTIFIED BY 'P@ssw0rd@2023';
Query OK, 0 rows affected (0.03 sec)

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd@2023';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CREATE DATABASE IF NOT EXISTS `nodelogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0.00 sec)

mysql> GRANT ALL ON nodelogin.* TO 'example_user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> USE nodelogin;
Database changed
mysql> CREATE TABLE IF NOT EXISTS `accounts` (
    ->   `id` int(11) NOT NULL AUTO_INCREMENT,
    ->   `username` varchar(50) NOT NULL,
    ->   `password` varchar(255) NOT NULL,
    ->   `email` varchar(100) NOT NULL,
    ->   PRIMARY KEY (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

อนุญาตให้ user เข้ามาจัดการได้

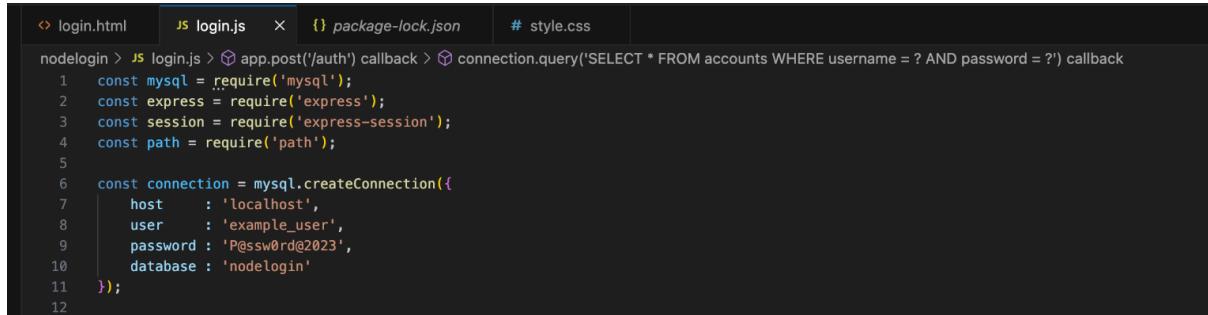
```
mysql> INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (1, 'test', 'test', 'test@test.com');
Query OK, 1 row affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

ใส่ข้อมูลและตรวจสอบ

```
mysql> SELECT * FROM accounts;
+----+-----+-----+-----+
| id | username | password | email      |
+----+-----+-----+-----+
| 1  | test     | test     | test@test.com |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

Remote จาก VSCode เข้า Ubuntu และไปที่ไฟล์เดอร์ lab4/login.js เปลี่ยน user และ password



```
login.html      JS login.js      package-lock.json      style.css
nodelogin > JS login.js > app.post('/auth') callback > connection.query('SELECT * FROM accounts WHERE username = ? AND password = ?') callback
1  const mysql = require('mysql');
2  const express = require('express');
3  const session = require('express-session');
4  const path = require('path');
5
6  const connection = mysql.createConnection({
7    host     : 'localhost',
8    user     : 'example_user',
9    password : 'P@ssw0rd@2023',
10   database : 'nodeLogin'
11 });
12
```

เปลี่ยนเลข port เป็น 3003

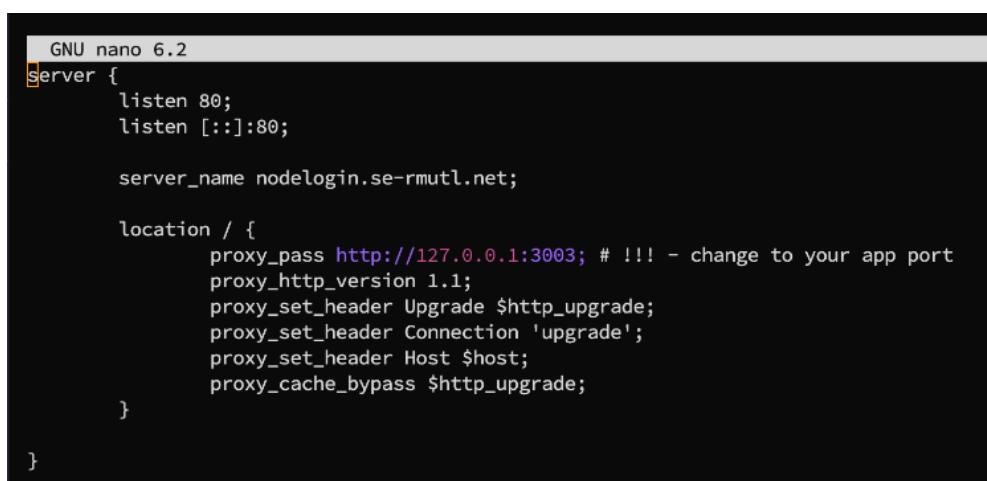
```
2     app.listen(3003);
```

ใช้คำสั่ง ufw allow 3003 อนุญาตให้รันโดยไม่ติด firewall

```
devuser@webserver014:/var/www/lab4/nodelogin$ sudo ufw allow 3003
Rule added
Rule added (v6)
```

เข้าไปกำหนดเลข port และชื่อโดเมน

```
devuser@webserver014:/var/www/lab4/nodelogin$ sudo nano /etc/nginx/sites-available/nodelogin
```



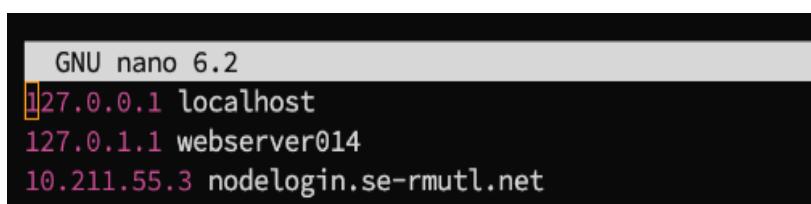
```
GNU nano 6.2
server {
    listen 80;
    listen [::]:80;

    server_name nodelogin.se-rmutil.net;

    location / {
        proxy_pass http://127.0.0.1:3003; # !!! - change to your app port
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

เพิ่มโดเมนข้างใน

```
devuser@webserver014:/var/www/lab4/nodelogin$ sudo nano /etc/hosts
```



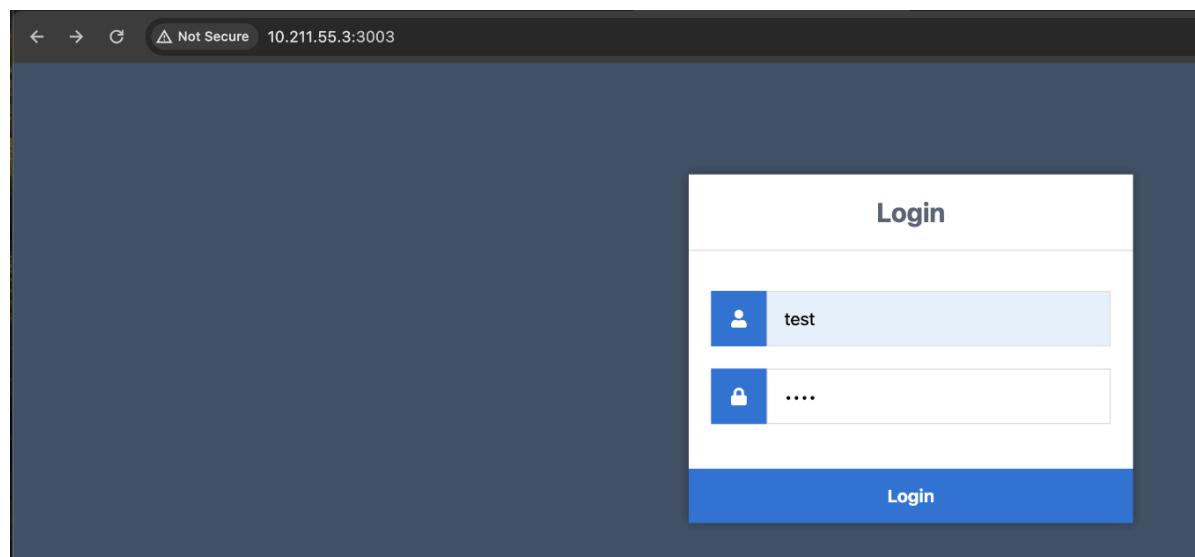
```
GNU nano 6.2
127.0.0.1 localhost
127.0.1.1 webserver014
10.211.55.3 nodelogin.se-rmutil.net
```

เพิ่มโดเมนในเครื่อง mac

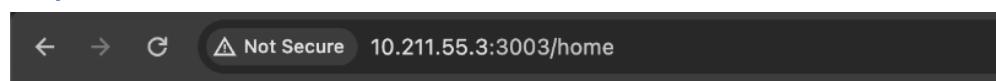
```
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost engse203.lab1  
255.255.255.255 broadcasthost  
:1              localhost  
10.211.55.3    ubuntu-linux--se-.shared ubuntu-linux--se- #prl_hostonly shared  
10.211.55.3    nodelogin.se-rmutl.net  
10.211.55.3    myapp.se-rmutl.net expressjs-example.se-rmutl.net my-react-app.se-rmutl.net ubuntu-next-app.se-rmutl.net  
  
SE  
SE
```

รัน

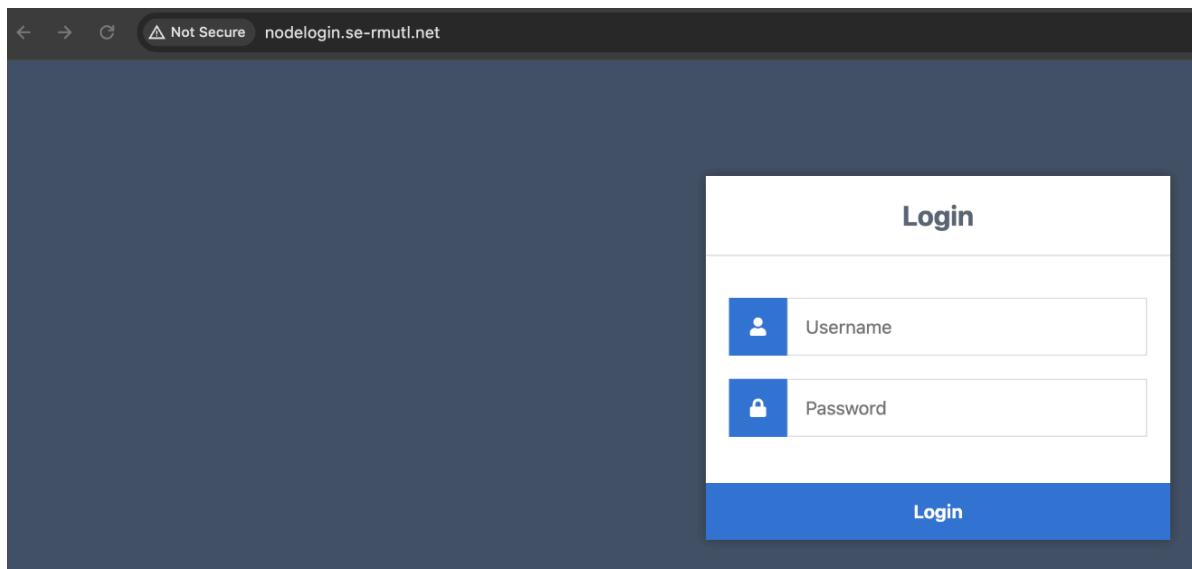
<http://10.211.55.3:3003/>



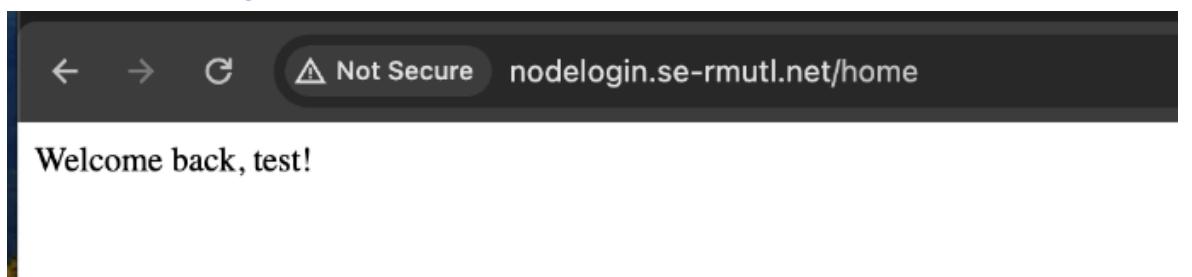
<http://10.211.55.3:3003/home>



<http://nodelogin.se-rmutil.net/>



<http://nodelogin.se-rmutil.net/home>



step 6.1

ติดตั้ง Client ที่เครื่อง webserver014

```
devuser@webserver014:/var/www/lab4$ mkdir client
```

```
devuser@webserver014:/var/www/lab4$ cd client
```

รีโนท์ vscode เข้าไปในโฟลเดอร์ที่พึงสร้างมา

A screenshot of the VS Code code editor. The left sidebar shows a file tree with a folder named 'CLIENT [SSH: 10.211.55.3]' containing files like App.css, App.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json, package.json, and README.md. The right pane shows the content of the 'package.json' file. The file contains script definitions for 'start', 'build', 'test', and 'eject'.

ไปที่ไฟล์ package.json เพิ่มพอร์ต 3004 เข้าไป

```
# App.css      JS App.js      JS index.js      {} package.json ×
{} package.json > {} dependencies
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": "^1.6.4",
10     "react": "^18.2.0",
11     "react-dom": "^18.2.0",
12     "react-scripts": "5.0.1",
13     "web-vitals": "^2.1.4"
14   },
15   ▷ Debug
16   "scripts": {
17     "start": "PORT=3004 react-scripts start",
18     "build": "react-scripts build",
19     "test": "react-scripts test",
20     "eject": "react-scripts eject"
21   },
22 }
```

ใช้คำสั่ง ufw allow 3004 อนุญาติให้รัน

```
devuser@webserver014:/var/www/lab4/client$ sudo ufw allow 3004
Rule added
Rule added (v6)
```

รัน

```
Compiled successfully!
```

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

```
Local:          http://localhost:3004
On Your Network: http://10.211.55.3:3004
```

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

```
webpack compiled successfully
```



Registration

Username

password

Login

Username...

Password...

จากนั้นใช้คำสั่ง sudo nano /etc/nginx/sites-available/client

```
devuser@webserver014:/var/www/lab4/client$ sudo nano /etc/nginx/sites-available/client
[sudo] password for devuser:
```

เข้าไปกำหนดเลข port และชื่อโดเมน

```
GNU nano 6.2                               /etc/nginx/sites-available/client
server {
    listen 80;
    listen [::]:80;
    server_name weblogin.se-rmutil.net;

    location / {
        proxy_pass http://127.0.0.1:3004; # !!! - change to your app port
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

เข้าไปที่ /etc/hosts เข้าไปเพิ่มโดเมนใน ip เพื่อให้ mac สามารถเปิดโดเมนนี้ได้

```
UW PICO 5.09                                         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
10.211.55.3    ubuntu-linux--se-.shared ubuntu-linux--se- #prl_hostonly shared
10.211.55.3    weblogin.se-rmutil.net
10.211.55.3    nodelogin.se-rmutil.net
10.211.55.3    myapp.se-rmutil.net expressjs-example.se-rmutil.net my-react-app.se-rmutil.net ubuntu-next-app.se-$
```

เข้าไปเพิ่ม ip และชื่อโดเมน

```
devuser@webserver014:/var/www/lab4/client$ sudo nano /etc/hosts
```

```
GNU nano 6.2                               /etc/hosts
127.0.0.1 localhost
127.0.1.1 webserver014
10.211.55.3 weblogin.se-rmutl.net
10.211.55.3 nodelogin.se-rmutl.net
```

จากนั้น cd เพื่อย้อนกลับแล้วใช้ restart nginx ใหม่

```
devuser@webserver014:$ sudo systemctl restart nginx
```

รัน

The screenshot shows a web browser window with the URL `Not Secure weblogin.se-rmutl.net`. The page contains two forms: a registration form and a login form.

Registration Form:

- Label: Username
- Input field: (empty)
- Label: password
- Input field: (empty)
- Button: Register

Login Form:

- Label: Username...
- Input field: (empty)
- Label: Password...
- Input field: (empty)
- Button: Login

step 6.2

ติดตั้ง Server ที่ เครื่อง webserver014

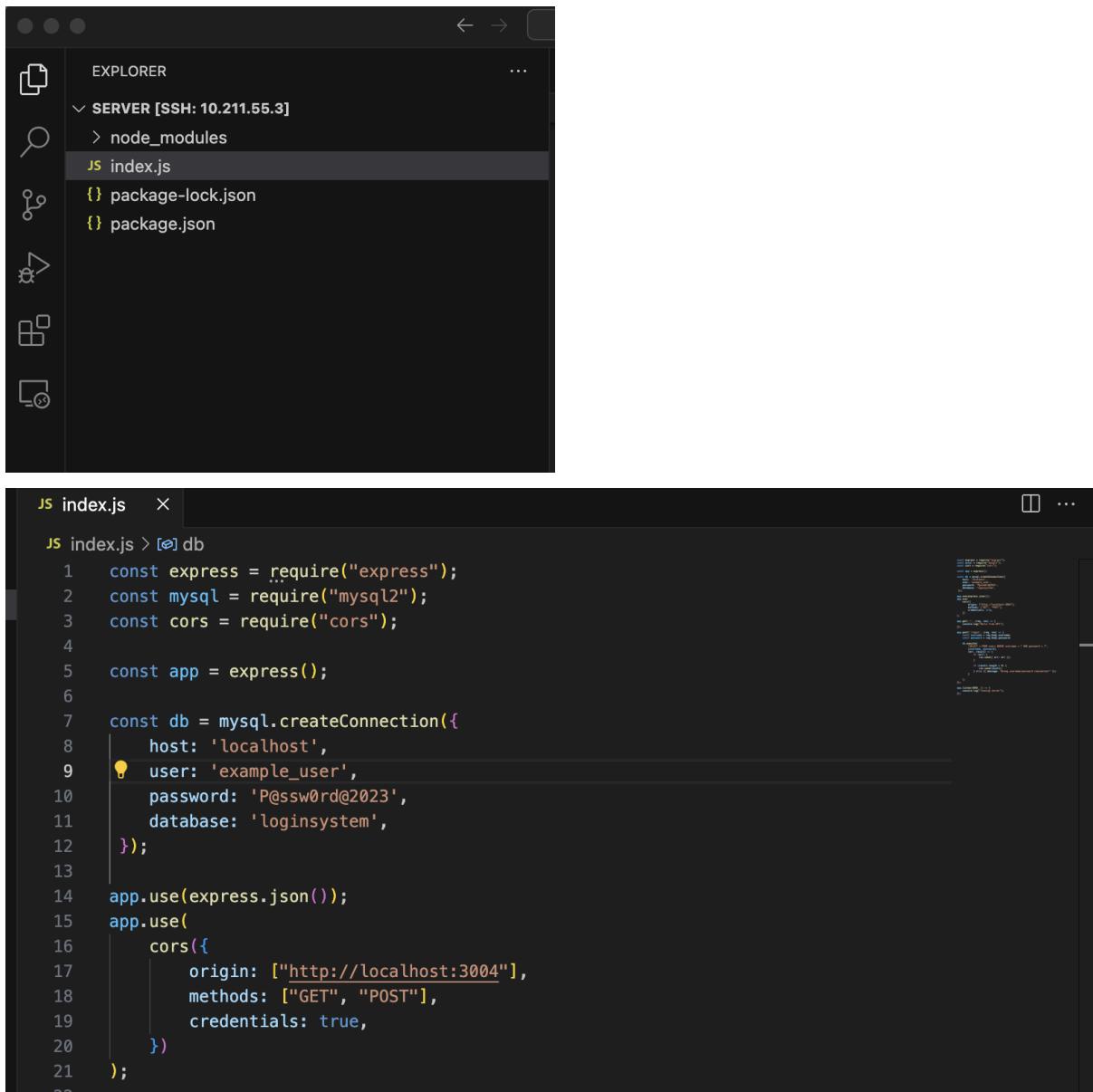
```
devuser@webserver014:/var/www/lab4/client$ cd ..  
devuser@webserver014:/var/www/lab4$ mkdir server  
devuser@webserver014:/var/www/lab4$ cd server
```

เข้าไปใน server สร้างไฟล์ package.json ด้วย npm init และติดตั้งส่วนอื่นๆ

```
devuser@webserver014:/var/www/lab4/server$ 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: (server)  
version: (1.0.0)  
description:  
entry point: (index.js)  
test command:  
git repository:  
keywords:  
author:  
license: (ISC)  
About to write to /var/www/lab4/server/package.json:  
  
{  
  "name": "server",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC"  
}  
  
Is this OK? (yes)  
devuser@webserver014:/var/www/lab4/server$ npm install express --save  
  
added 62 packages, and audited 63 packages in 1s  
  
11 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities
```

```
devuser@webserver014:/var/www/lab4/server$ npm install express --save  
  
added 62 packages, and audited 63 packages in 1s  
  
11 packages are looking for funding  
  run `npm fund` for details
```

ทำการสร้าง index.js



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a file tree for a project on a remote server (SSH: 10.211.55.3). The tree includes a folder named 'node_modules', a file named 'index.js' which is currently selected, and two empty files named 'package-lock.json' and 'package.json'. The main editor area on the right contains the code for 'index.js'. The code initializes an Express application, connects to a MySQL database, and sets up CORS middleware. The code is color-coded for syntax highlighting.

```
JS index.js > [?] db
1  const express = require("express");
2  const mysql = require("mysql2");
3  const cors = require("cors");
4
5  const app = express();
6
7  const db = mysql.createConnection({
8      host: 'localhost',
9      user: 'example_user',
10     password: 'P@ssw0rd@2023',
11     database: 'loginsystem',
12   });
13
14 app.use(express.json());
15 app.use(
16   cors({
17     origin: ["http://localhost:3004"],
18     methods: ["GET", "POST"],
19     credentials: true,
20   })
21 );
22
```

ทำการลง middleware

```
devuser@webserver014:/var/www/lab4/server$ npm install --save mysql2
added 11 packages, and audited 74 packages in 8s

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

found 0 vulnerabilities
devuser@webserver014:/var/www/lab4/server$ npm i --save cors
added 2 packages, and audited 76 packages in 3s

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

found 0 vulnerabilities
devuser@webserver014:/var/www/lab4/server$ sudo npm install -g nodemon
[sudo] password for devuser:

changed 33 packages in 3s

3 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New patch version of npm available! 10.2.3 -> 10.2.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.5
npm notice Run npm install -g npm@10.2.5 to update!
npm notice
```

ทำการสร้าง database ชื่อ loginsystem

```
devuser@webserver014:/var/www/lab4/server$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0.01 sec)

mysql> USE loginsystem;
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, 1 warning (0.03 sec)

mysql> USE loginsystem;
Database changed
mysql> INSERT INTO `users`(`username`, `password`) VALUES ('test', 'test');
Query OK, 1 row affected (0.00 sec)

mysql> select * from users;
+-----+-----+
| username | password |
+-----+-----+
| test     | test      |
+-----+-----+
1 row in set (0.00 sec)

mysql> 
```

```
devuser@webserver014:/var/www/lab4/server$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> GRANT ALL ON loginsystem.* TO 'example_user'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
devuser@webserver014:/var/www/lab4/server$ sudo node index.js
[sudo] password for devuser:
running server
Hello from API

```

ติดตั้ง noddedemon และ เปิดport:4000

```
devuser@webserver014:/var/www/lab4/server$ sudo ufw allow 4000  
[sudo] password for devuser:  
Rule added  
Rule added (v6)
```

```
devuser@webserver014:/var/www/lab4/server$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 26  
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2023, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> GRANT ALL ON loginsystem.* TO 'example_user'@'%';  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> exit  
Bye
```

รันและทดสอบ

```
devuser@webserver014:/var/www/lab4/server$ sudo node index.js  
[sudo] password for devuser:  
running server  
Hello from API  
□
```

Compiled successfully!

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

Local: http://localhost:**3004**
On Your Network: http://10.211.55.3:**3004**

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

webpack compiled **successfully**

□

Registration

Username

test2

password

test2

Register

Login

Username...

Password...

Login

'test2' was inserted.

Registration

Username

password

Register

Login

test2

.....

Login

'test2' was logged in.



Registration

Username

password

Login

Username...

Password...

Duplicate entry 'test' for key 'users.PRIMARY'



Registration

Username

password

Login

test4

.....

Wrong username/password comination!

index.js code

```
JS index.js > ...
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  const db = mysql.createConnection({
8    host: 'localhost',
9    user: 'example_user',
10   password: 'P@ssw0rd@2023',
11   database: 'loginsystem',
12 });
13 app.use(express.json());
14 app.use(
15   cors({
16     origin: ["http://weblogin.se-rmutil.net"],
17     methods: ["GET", "POST"],
18     credentials: true,
19   })
20 );
21
22 app.get('/', (req, res) => {
23   console.log("Hello from API");
24 });
25
26 app.post('/login', (req, res) => {
27   let username = req.body.username;
28   let password = req.body.password;
29
30   console.log("body: "+ JSON.stringify(req.body));
31
32   console.log("username: "+username);
33   console.log("password: "+password);
34
35   db.execute(
36     "SELECT * FROM users WHERE username = ? AND password = ?",
37     [username, password],
38     (err, result) => {
39       if (err) {
40         res.send({ err: err });
41       }
42       if (result.length > 0) {
43         res.send(result);
44       } else {
45         res.send({ message: "Wrong username/password combination!" });
46       }
47     }
48   );
49 });
50 app.post('/register', (req, res) => {
51   let username = req.body.username;
52   let password = req.body.password;
53
54   console.log("body: "+ JSON.stringify(req.body));
55
56   console.log("username: "+username);
57   console.log("password: "+password);
58
59   db.execute(
60     "INSERT INTO users (username, password) VALUES (?,?)",
61     [username, password],
62     (err, result) => {
63       if (err) {
64         res.send({ err: err });
65       }
66       res.send(result);
67     }
68   );
69 });
70
71 app.listen(4000, () => {
72   console.log("running server");
73 });
74 })!
```

app.js code

```
JS App.js  X
client > src > JS App.js > App > register > then() callback
  1 import React, { useEffect, useState } from "react";
  2 import Axios from 'axios';
  3 import './App.css';
  4 function App() {
  5   const [usernameReg, setUsernameReg] = useState("");
  6   const [passwordReg, setPasswordReg] = useState ("");
  7   const [username, setUsername] = useState("");
  8   const [password, setPassword] = useState ("");
  9   const [loginStatus, setLoginStatus] = useState("");
10   const register = () => {
11     Axios.post("http://10.211.55.3:4000/register", {
12       username: usernameReg,
13       password: passwordReg,
14     }).then((response) => {
15       console.log(response);
16
17       if (!response.data.err) {
18         setLoginStatus(""+usernameReg + "' was inserted.");
19       } else {
20         setLoginStatus(response.data.err.sqlMessage);
21       }
22     });
23   };
24   const login = () => {
25     Axios.post("http://10.211.55.3:4000/login", {
26       username: username,
27       password: password,
28     }).then((response) => {
29       console.log(response);
30       //console.log("response.data.message: "+response.data[0].message);
31
32
33       if (!response.data.message) {
34         setLoginStatus(""+response.data[0].username + "' was logged in.");
35       } else {
36         setLoginStatus(response.data.message);
37       }
38     });
39   };
40
41   return (
42     <div className="App">
43       <div className="registration">
44         <h1>Registration</h1>
45         <label>Username</label>
46         <input
47           type="text"
48           onChange={(e) => {
49             setUsernameReg(e.target.value);
50           }}
51         /><br/>
52         <label>password</label>
53         <input
54           type="text"
55           onChange={(e) =>{
56             setPasswordReg(e.target.value);
57           }}
58         /> <br />
59         <button onClick={register}> Register</button>
60       </div>
61       <div className="login">
62         <h1>Login</h1>
63         <input
64           type="text"
65           placeholder="Username..."
66           onChange = { (e) => {
67             setUsername (e.target.value);
68           }}
69         /> <br/>
70
71
72         <input
73           type="password"
74           placeholder="Password..."
75           onChange = { (e) => {
76             setPassword (e.target.value);
77           }}
78         />
79         <button onClick={login}>Login</button>
80       </div>
81     <h1> {loginStatus}</h1>
82   </div>
83 );
84
85
86 export default App;
```

Step 7 Create Login System with Node.js App

- เข้าสู่ระบบเป็น root
- ให้สิทธิ์ผู้ใช้งานเทียบเท่า root
- อนุญาติให้ใช้ openSSH
- เปิดไฟล์วอล์
- ตรวจสอบสถานะ

```
devuser@backend014:~$ sudo -i
[sudo] password for devuser:
root@backend014:~# adduser non
Adding user `non' ...
Adding new group `non' (1001) ...
Adding new user `non' (1001) with group `non' ...
Creating home directory `/home/non' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for non
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] y
root@backend014:~# usermod -aG sudo non
root@backend014:~# ufw allow OpenSSH
Rules updated
Rules updated (v6)
root@backend014:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@backend014:~# ufw status
Status: active

To                         Action      From
--                         --          --
Nginx Full                 ALLOW      Anywhere
OpenSSH                     ALLOW      Anywhere
Nginx Full (v6)             ALLOW      Anywhere (v6)
OpenSSH (v6)                ALLOW      Anywhere (v6)

root@backend014:~#
```

- ทำการ update
- ทำการ upgrade

```
devuser@backend014:~$ sudo apt update
Hit:1 https://deb.nodesource.com/node_20.x nodistro InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:3 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Hit:4 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Reading package lists... Done
```

```
devuser@backend014:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  python3-update-manager update-manager-core
The following packages will be upgraded:
  distro-info distro-info-data python3-distro-info python3-software-properties
  software-properties-common
5 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
Need to get 73.3 kB of archives.
After this operation, 5,120 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 distro-info-data all 0.52ubuntu0
[656B]
```

- ติดตั้ง mysql
- ตรวจสอบการติดตั้ง mysql

```
devuser@backend014:~$ sudo apt-cache search mysql-server
mysql-server - MySQL database server (metapackage depending on the latest version)
mysql-server-8.0 - MySQL database server binaries and system database setup
mysql-server-core-8.0 - MySQL database server binaries
default-mysql-server - MySQL database server binaries and system database setup (metapackage)
default-mysql-server-core - MySQL database server binaries (metapackage)
mariadb-server-10.6 - MariaDB database server binaries
mariadb-server-core-10.6 - MariaDB database core server files
devuser@backend014:~$ sudo apt info -a mysql-server-8.0
Package: mysql-server-8.0
Version: 8.0.35-0ubuntu0.22.04.1
Priority: optional
Section: database
Source: mysql-8.0
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian MySQL Maintainers <pkg-mysql-maint@lists.alioth.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 1,402 kB
Provides: virtual-mysql-server
Pre-Depends: adduser (>= 3.40), debconf, mysql-common (>= 5.5)
Depends: lsb-base (>= 3.0-10), mysql-client-8.0 (>= 8.0.35-0ubuntu0.22.04.1), mysql-common (>= 5.8+1
.0.4~), mysql-server-core-8.0 (= 8.0.35-0ubuntu0.22.04.1), passwd, perl:any (>= 5.6), psmisc, debconf
f (>= 0.5) | debconf-2.0
Recommends: libhtml-template-perl, mecab-ipadic-utf8
Suggests: mailx, tinyca
Conflicts: mariadb-server-10.1, mariadb-server-10.3, mysql-server-5.7, virtual-mysql-server
Homepage: http://dev.mysql.com/
Task: lamp-server
Download-Size: 1,238 kB
APT-Manual-Installed: yes
APT-Sources: http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 Packages
Description: MySQL database server binaries and system database setup
MySQL is a fast, stable and true multi-user, multi-threaded SQL database
server. SQL (Structured Query Language) is the most popular database query
language in the world. The main goals of MySQL are speed, robustness and
ease of use.
.
This package contains all the infrastructure needed to setup system
databases.
```

ติดตั้ง mysql 8.0

```
devuser@backend014:~$ sudo apt install mysql-server-8.0
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server-8.0 is already the newest version (8.0.35-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

เข้าไปทำงาน mysql

กำหนดรหัสผ่านในการเข้า mysql ครั้งต่อไป และออก

```
devuser@backend014:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd@2023';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

ติดตั้งตัวสร้างความปลอดภัยให้กับ mysql

```
devuser@backend014:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.
Using existing password for root.

Estimated strength of the password: 100
Change the password for root ? ((Press y|Y for Yes, any other key for No) : y

New password:

Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) :

New password:

Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
  - Dropping test database...
Success.

  - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
devuser@backend014:~$
```

ติดตั้งตัวสร้างความปลอดภัยให้กับ mysql|ทำการอัพเดทและอัพเกรด

ติดตั้ง nginx

อนุญาติให้เข้าถึงไฟล์วอลล์ของ nginx

```
devuser@backend016:~$ sudo apt update && sudo apt upgrade
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Fetched 229 kB in 2s (104 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  distro-info distro-info-data python3-distro-info python3-software-properties
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

```
devuser@backend014:~$ sudo apt install nginx certbot python3-certbot-nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
certbot is already the newest version (1.21.0-1build1).
python3-certbot-nginx is already the newest version (1.21.0-1).
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

```
devuser@backend016:~$ sudo ufw allow 'Nginx Full'
Rule added
Rule added (v6)
```

ติดตั้ง nodejs

```
devuser@backend014:~$ curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash - && \
sudo apt-get install -y nodejs
2024-01-08 11:01:00 - Installing pre-requisites
Hit:1 https://deb.nodesource.com/node_20.x nodistro InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Reading package lists... Done
E: Release file for http://ports.ubuntu.com/ubuntu-ports/dists/jammy-updates/InRelease is not valid
yet (invalid for another 17h 10min 0s). Updates for this repository will not be applied.
E: Release file for http://ports.ubuntu.com/ubuntu-ports/dists/jammy-security/InRelease is not valid
yet (invalid for another 17h 9min 13s). Updates for this repository will not be applied.
```

ทดสอบเรียก node และ npm

```
devuser@backend014:~$ node -v  
v20.10.0  
devuser@backend014:~$ npm -v  
10.2.3
```

ติดตั้ง Server ที่ เครื่อง backend014

```
devuser@backend014:~$ sudo mkdir -p /var/www/lab4  
[sudo] password for devuser:  
devuser@backend014:~$ cd /var/www/lab4  
devuser@backend014:/var/www/lab4$ ls -l  
total 0
```

```
devuser@backend014:/var/www/lab4$ sudo mkdir server  
[sudo] password for devuser:  
devuser@backend014:/var/www/lab4$ cd server
```

เข้าไปใน server สร้างไฟล์ package.json ด้วย npm init และติดตั้งส่วนอื่นๆ

```
devuser@backend014:/var/www/lab4/server$ sudo 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: (server)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /var/www/lab4/server/package.json:

{
  "name": "server",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
npm notice
npm notice New patch version of npm available! 10.2.3 -> 10.2.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.5
npm notice Run npm install -g npm@10.2.5 to update!
npm notice
```

ทำการลง middleware

```
devuser@backend014:/var/www/lab4/server$ sudo npm install express --save
added 62 packages, and audited 63 packages in 2s

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

found 0 vulnerabilities
```

```
devuser@backend014:/var/www/lab4/server$ sudo npm install --save mysql2
added 11 packages, and audited 74 packages in 2s

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

found 0 vulnerabilities
devuser@backend014:/var/www/lab4/server$ sudo npm i --save cors

added 2 packages, and audited 76 packages in 876ms

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

found 0 vulnerabilities
```

```
devuser@backend014:/var/www/lab4/server$ sudo npm install -g nodemon
added 33 packages in 2s

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

ทำการสร้าง database ชื่อ loginsystem

```
devuser@backend014:/var/www/lab4/server$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0.02 sec)

mysql> USE loginsystem;
Database changed
mysql> INSERT INTO `users` (`username`, `password`) VALUES ('test', 'test');
ERROR 1146 (42S02): Table 'loginsystem.users' doesn't exist
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, 1 warning (0.03 sec)

mysql> USE loginsystem;
Database changed
mysql> INSERT INTO `users` (`username`, `password`) VALUES ('test', 'test');
Query OK, 1 row affected (0.02 sec)

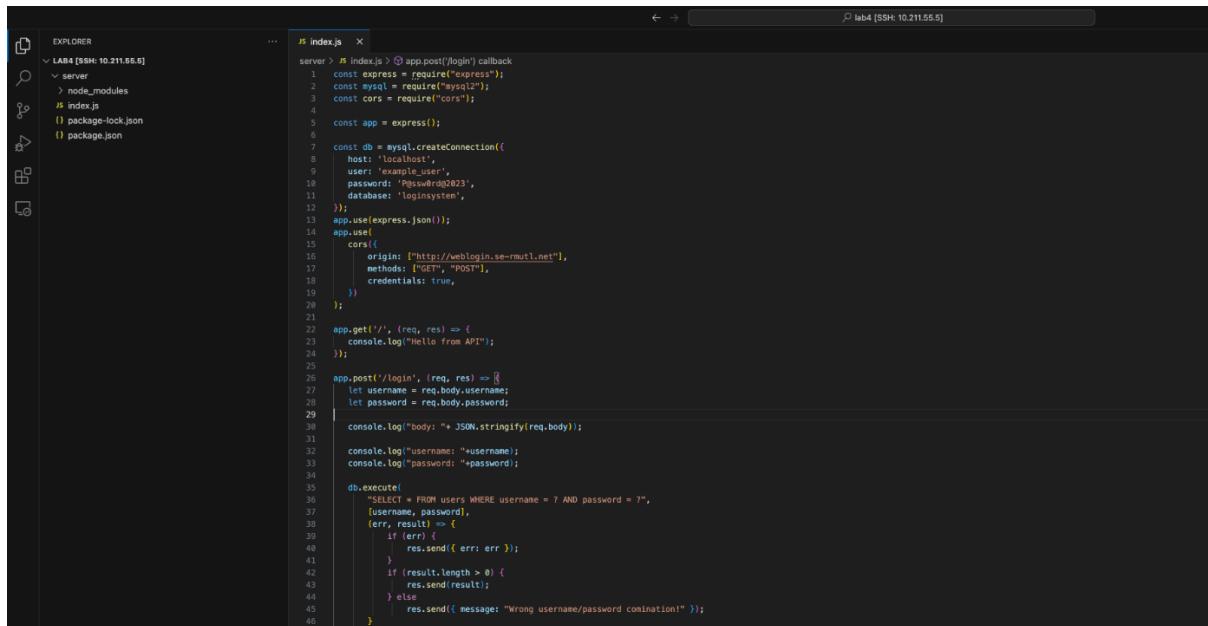
mysql> select * from users;
+-----+-----+
| username | password |
+-----+-----+
| test     | test      |
+-----+-----+
1 row in set (0.00 sec)

mysql> exit
Bye
devuser@backend014:/var/www/lab4/server$
```

ໃຫ້ຄໍາສັ່ງ ufw allow 4000 ອນໝາດໃຫ້ຮັນ

```
devuser@backend014:/var/www/lab4/server$ sudo ufw allow 4000
[sudo] password for devuser:
Rule added
Rule added (v6)
```

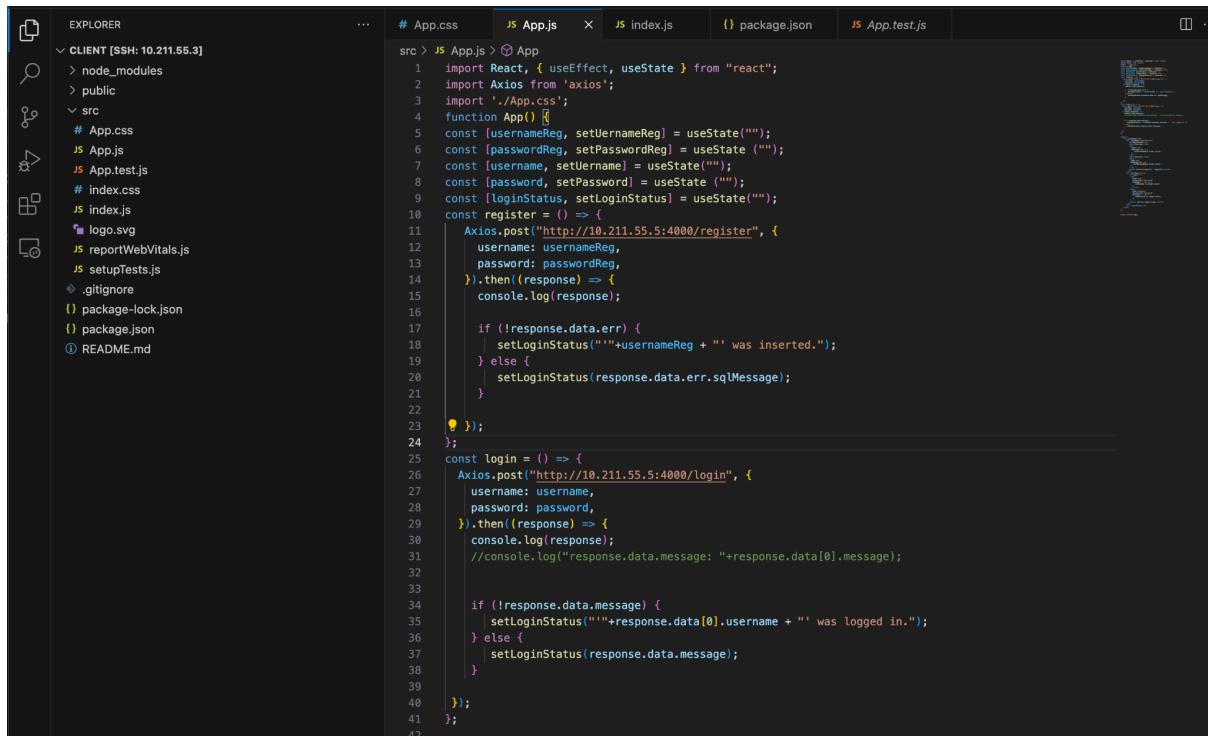
ຮຶນໂທ vscode ເຂົ້າກັນ backend server ແລ້ວໃຫ້ສ້າງໄຟລ് index.js ໄປໃສ່ເນຳ
ໂຄດຂອງ web server ເຂົ້າມາ



```
EXPLORER JS index.js
server > # index.js > ④ app.post('/login') callback
1 const express = require('express');
2 const mysql = require('mysql2');
3 const cors = require('cors');
4
5 const app = express();
6
7 const db = mysql.createConnection({
8   host: 'localhost',
9   user: 'hexacode_user',
10  password: 'P@ssw0rd#023',
11  database: 'loginsystem',
12 });
13 app.use(express.json());
14 app.use(
15   cors({
16     origin: ['http://weblogin.se-rmuil.net'],
17     methods: ['GET', 'POST'],
18     credentials: true,
19   })
20 );
21
22 app.get('/', (req, res) => {
23   console.log('Hello from API');
24 });
25
26 app.post('/login', (req, res) => {
27   let username = req.body.username;
28   let password = req.body.password;
29
30   console.log(`body: ${JSON.stringify(req.body)}`);
31
32   console.log(`username: ${username}`);
33   console.log(`password: ${password}`);
34
35   db.execute(
36     'SELECT * FROM users WHERE username = ? AND password = ?',
37     [username, password],
38   ).then(result => {
39     if (err) {
40       res.send({ err: err });
41     }
42     if (result.length > 0) {
43       res.send(result);
44     } else {
45       res.send({ message: "Wrong username/password combination!" });
46     }
47   );
48 });

JS index.js
```

ແກ້ໄຂ ip ໃນ App.js ຜົ່ງ client ໃຫ້ເປັນ ip 10.211.55.5 ຕາມຂອງ server



```
EXPLORER # App.css JS App.js JS index.js () package.json JS App.test.js
src > JS App.js > ④ App
1 import React, { useEffect, useState } from "react";
2 import Axios from 'axios';
3 import './App.css';
4 function App() {
5   const [usernameReg, setUsernameReg] = useState("");
6   const [passwordReg, setPasswordReg] = useState("");
7   const [username, setUsername] = useState("");
8   const [password, setPassword] = useState("");
9   const [loginStatus, setLoginStatus] = useState("");
10  const register = () => {
11    Axios.post("http://10.211.55.5:4000/register", {
12      username: usernameReg,
13      password: passwordReg,
14    }).then(response) => {
15      console.log(response);
16
17      if (!response.data.err) {
18        setLoginStatus(`"${usernameReg}" was inserted.`);
19      } else {
20        setLoginStatus(response.data.err.sqlMessage);
21      }
22    };
23  };
24  const login = () => {
25    Axios.post("http://10.211.55.5:4000/login", {
26      username: username,
27      password: password,
28    }).then(response) => {
29      console.log(response);
30      //console.log(`response.data.message: ${response.data[0].message}`);
31
32      if (!response.data.message) {
33        setLoginStatus(`"${username}" was logged in.`);
34      } else {
35        setLoginStatus(response.data.message);
36      }
37    };
38  };
39}
40
41

JS App.js
```

ให้รันทั้ง webserver และ backend server และให้รันแบบโดเมน และให้เข้าสู่ระบบดูว่าหรือไม่

Not Secure weblogin.se-rmuf.net

Registration

Username
test

Password
test

Login

Username...
test

Password...
test

Duplicate entry 'test' for key 'users.PRIMARY'

```
devuser@backend014:/var/www/lab4/server$ nodemon index.js
[nodemon] 3.0.2
[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
body: {"username":"test","password":"test"}
username: test
password: test
```

ทำให้ client ต่อ กับ server ของ backend014

จากนั้นใช้คำสั่ง cd /etc/nginx/sites-available และทำการก็อปปี้ตัว default ที่ใช้ได้มา

```
devuser@backend014:/var/www/lab4/server$ cd /etc/nginx/sites-available
```

```
devuser@backend014:/etc/nginx/sites-available$ sudo cp default server
```

เข้าไปกำหนดเลข port และชื่อโดเมน

```
devuser@backend014:/etc/nginx/sites-available$ sudo nano server
```

```
GNU nano 6.2                                     server
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80;
    listen [::]:80;
    server_name backend_server_ip_address;

    location / {
        proxy_pass http://127.0.0.1:4000; # !!! - change to your app port
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

เข้าไปเพิ่ม ip และชื่อโดเมน

```
devuser@backend014:/etc/nginx/sites-available$ sudo nano /etc/hosts
```

```
GNU nano 6.2                                     /etc/hosts *
127.0.0.1 localhost
127.0.1.1 backend014
10.211.55.5 backend_server_ip_address
# The following lines are desirable for IPv6 capable hosts
```

จากนั้น cd เพื่อย้อนกลับแล้วใช้ restart nginx ใหม่ จากนั้นก็เข้า terminal

```
devuser@backend014:/etc/nginx/sites-available$ sudo systemctl restart nginx
```

เข้าไปที่ /etc/hosts เข้าไปเพิ่มโดเมนใน ip เพื่อให้ mac สามารถเปิดโดเมนนี้ได้

```
UW PICO 5.09                                         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
10.211.55.3    ubuntu-linux--se-.shared ubuntu-linux--se- #prl_hostonly shared
10.211.55.5    backend_server_ip_address
10.211.55.3    weblogin.se-rmutil.net
10.211.55.3    nodelogin.se-rmutil.net
10.211.55.3    myapp.se-rmutil.net expressjs-example.se-rmutil.net my-react-app.$

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit       ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

แก้ไข ip ใน App.js ผัง client ให้เป็น
http://backend_server_ip_address:4000 ตามรูป

```
# App.css      JS App.js  X  JS index.js  {} package.json  JS App.test.js
src > JS App.js > ⚡ App > ⚡ register > ⚡ then() callback
1  import React, { useEffect, useState } from "react";
2  import Axios from 'axios';
3  import './App.css';
4  function App() {
5    const [usernameReg, setUsernameReg] = useState("");
6    const [passwordReg, setPasswordReg] = useState ("");
7    const [username, setUsername] = useState("");
8    const [password, setPassword] = useState ("");
9    const [loginStatus, setLoginStatus] = useState("");
10   const register = () => {
11     Axios.post("http://backend_server_ip_address:4000/register", {
12       username: usernameReg,
13       password: passwordReg,
14     }).then((response) => [
15       console.log(response);
16
17       if (!response.data.err) {
18         setLoginStatus("'" +usernameReg + "' was inserted.");
19       } else {
20         setLoginStatus(response.data.err.sqlMessage);
21       }
22     ]);
23   };
24
25   const login = () => {
26     Axios.post("http://backend_server_ip_address:4000/login", {
27       username: username,
28       password: password,
29     }).then((response) => {
30       console.log(response);
31       //console.log("response.data.message: "+response.data[0].message);
32     });
33   };
34 }
```

แล้วทำการรัน

```
Compiled successfully!

You can now view client in the browser.

Local:          http://localhost:3004
On Your Network: http://10.211.55.3:3004

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

webpack compiled successfully
[
```

```
devuser@backend014:/var/www/lab4/server$ nodemon index.js
[nodemon] 3.0.2
[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
body: {"username":"test","password":"test"}
username: test
password: test
body: {"username":"test","password":"test"}
username: test
password: test
Hello from API
Hello from API
body: {"username":"test5","password":"test5"}
username: test5
password: test5
body: {"username":"test5","password":"test5"}
username: test5
password: test5
[
```

The screenshot shows a web browser window with two tabs open. The active tab is titled "backend_server_ip_address" and displays a "Registration" form. The form has two input fields: "Username" containing "test5" and "password" containing "test5". Below the form is a "Register" button. The second tab is titled "React App" and shows a "Login" form with two input fields: "Username..." and "Password...", and a "Login" button. At the bottom of the page, a message reads "'test5' was inserted."

Not Secure weblogin.se-rrmutt.net

Registration

Username

password

Login

'test5' was logged in.