

Lab 4

Step1 Install New Ubuntu Server

1. sudo apt update

```
devuser@backend097:~$ sudo apt update
[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]
Fetched 229 kB in 2s (101 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
47 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2. sudo apt upgrade

```
devuser@backend097:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
```

3. sudo ufw app list

```
devuser@backend097:~$ sudo ufw app list
Available applications:
  OpenSSH
```

4. sudo ufw allow OpenSSH

```
devuser@backend097:~$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
```

5. sudo ufw enable

```
devuser@backend097:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

6.sudo ufw status

```
devuser@backend097:~$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
OpenSSH                     ALLOW      Anywhere
OpenSSH (v6)                 ALLOW      Anywhere (v6)
```

วิธีเปลี่ยนชื่อ server

1. sudo nano /etc/hostname

```
devuser@server:~$ sudo nano /etc/hostname
```

เปลี่ยนชื่อจากอันเก่าเป็นชื่อใหม่

```
GNU nano 6.2
backend097
```

2. sudo nano /etc/hosts

```
devuser@server:~$ sudo nano /etc/hosts
```

เปลี่ยนชื่อตั้ง 127.0.1.1 เป็นชื่อใหม่

```
127.0.0.1 localhost
127.0.1.1 backend097

# The following lines are desirable for
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

3. sudo reboot

```
devuser@server:~$ sudo reboot
```

4. เช็คชื่อ server

hostnamectl

```
devuser@backend097:~$ hostnamectl
Static hostname: backend097
Icon name: computer-vm
Chassis: vm
Machine ID: 1a6936cd4e8042199c21e487a525c37e
Boot ID: 29d3a23434ea457188aab3a1d7c22df6
Virtualization: parallels
Operating System: Ubuntu 22.04.3 LTS
Kernel: Linux 5.15.0-91-generic
Architecture: arm64
Hardware Vendor: Parallels International GmbH.
Hardware Model: Parallels ARM Virtual Machine
```

Step2 MySQL Server

1. ค้นหาเซิร์ฟเวอร์ MySQL และแพ็คเกจไดลเอนต์บน Ubuntu 22.04

```
sudo apt-cache search mysql-server
```

```
devuser@backend097:~$ 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
```

2. sudo apt info -a mysql-server-8.0

```
devuser@backend097:~$ 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
```

3. sudo apt install mysql-server-8.0

```
devuser@backend097:~$ 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:
```

4. พิมพ์คำสั่งตามนี้

- sudo mysql
- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd@2023';
- exit

```
devuser@backend097:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
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
```

5. sudo mysql_secure_installation

```
devuser@backend097:~$ sudo mysql_secure_installation
```

6. เปิดใช้งานเซิร์ฟเวอร์ MySQL คำสั่งตามนี้

- sudo systemctl is-enabled mysql.service
- sudo systemctl enable mysql.service
- sudo systemctl status mysql.service

```
devuser@backend097:~$ sudo systemctl is-enabled mysql.service
enabled
devuser@backend097:~$ sudo systemctl enable mysql.service
Synchronizing state of mysql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
devuser@backend097:~$ sudo systemctl status mysql.service
● mysql.service - MySQL Community Server
    Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
      Active: active (running) since Wed 2024-01-03 06:15:51 UTC; 7min ago
        Main PID: 808 (mysqld)
          Status: "Server is operational"
            Tasks: 38 (limit: 2191)
           Memory: 425.7M
              CPU: 4.650s
            CGroup: /system.slice/mysql.service
                    └─808 /usr/sbin/mysqld

Jan 03 06:15:50 backend097 systemd[1]: Starting MySQL Community Server...
Jan 03 06:15:51 backend097 systemd[1]: Started MySQL Community Server.
```

Step3 Install Nginx

1. sudo apt update && sudo apt upgrade

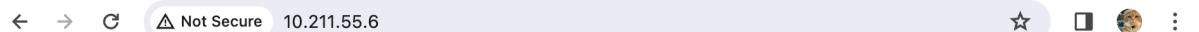
```
devuser@backend097:~$ sudo apt update && sudo apt upgrade
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Fetched 229 kB in 2s (115 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.
```

2. sudo apt install nginx certbot python3-certbot-nginx

```
devuser@backend097:~$ sudo apt install nginx certbot python3-certbot-nginx
```

3. sudo ufw allow 'Nginx Full'

```
devuser@backend097:~$ sudo ufw allow 'Nginx Full'
Rule added
Rule added (v6)
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Step4 Nodejs Installation

1. ดาวน์โหลดและนำเข้าคีย์ Nodesource GPG

- sudo apt-get update
- sudo apt-get install -y ca-certificates curl gnupg
- sudo mkdir -p /etc/apt/keyrings
- curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

```
devuser@backend097:~$ sudo apt-get update
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease
Reading package lists... Done
devuser@backend097:~$ sudo apt-get install -y ca-certificates curl gnupg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.15).
curl set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
devuser@backend097:~$ sudo mkdir -p /etc/apt/keyrings
devuser@backend097:~$ curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg
```

2. สร้างพื้นที่เก็บข้อมูล deb

- NODE_MAJOR=20
- echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_\$NODE_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list

```
devuser@backend097:~$ NODE_MAJOR=20
devuser@backend097:~$ echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list
deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_20.x nodistro main
```

3. Run Update and Install

- sudo apt-get update
- sudo apt-get install nodejs -y

```
devuser@backend097:~$ sudo apt-get update
Get:1 https://deb.nodesource.com/node_20.x nodistro InRelease [12.1 kB]
Get:2 https://deb.nodesource.com/node_20.x nodistro/main arm64 Packages [4,685 B]
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease
Hit:5 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease
Hit:6 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease
Fetched 16.8 kB in 1s (12.1 kB/s)
Reading package lists... Done
devuser@backend097:~$ sudo apt-get install nodejs -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
```

4. ทดสอบรีบิก node และ npm

- node -v
 - npm -v
- โดยการเช็คเวอร์ชันของ node และ npm

Step5 Create Login System with Node.js App

ทำใน Server เก่า (10.211.55.5)

1. sudo mkdir -p /var/www/lab4
cd /var/www/lab4

```
devuser@webserver097:~$ sudo mkdir -p /var/www/lab4
devuser@webserver097:~$ cd /var/www/lab4
```

2. sudo mkdir -p nodelogin

```
devuser@webserver097:/var/www/lab4$ sudo mkdir -p nodelogin
```

3. sudo chown -R \$USER:\$USER /var/www/lab4

```
devuser@webserver097:/var/www/lab4$ sudo chown -R $USER:$USER /var/www/lab4
```

4. cd nodelogin

```
devuser@webserver097:/var/www/lab4$ cd nodelogin
```

5. sudo npm init

```
devuser@webserver097:/var/www/lab4/nodelogin$ sudo npm init
```

6. sudo npm install express --save

```
devuser@webserver097:/var/www/lab4/nodelogin$ sudo npm install express --save
changed 1 package, and audited 63 packages in 6s

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

found 0 vulnerabilities
```

7. sudo npm install express-session --save

```
devuser@webserver097:/var/www/lab4/nodelogin$ sudo npm install express-session --save
added 5 packages, and audited 68 packages in 995ms

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

found 0 vulnerabilities
```

8. sudo npm install mysql --save

```
devuser@webserver097:/var/www/lab4/nodelogin$ sudo npm install mysql --save
added 12 packages, and audited 80 packages in 1s

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

found 0 vulnerabilities
```

9. Connect VS Code

ดูวิธี Connect ได้ใน lab2

10. DB MySQL

- sudo mysql -u root -p

```
devuser@webserver097:/var/www/lab4/nodelogin$ sudo mysql -u root -p
```

- CREATE USER 'example_user'@'%' IDENTIFIED BY 'P@ssw0rd@2023';

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

- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd@2023';

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

- CREATE DATABASE IF NOT EXISTS `nodelogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

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

- GRANT ALL ON nodelogin.* TO 'example_user'@'%';

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

- USE nodelogin;

```
mysql> USE nodelogin;
Database changed
```

- 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;

```
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 (0.02 sec)
```

- INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (1, 'test', 'test', 'test@test.com');

```
mysql> INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (1, 'test', 'test',
-> @test.com');
Query OK, 1 row affected (0.00 sec)
```

- EXIT

```
mysql> EXIT
Bye
devuser@webserver097:/var/www/lab4/nodelogin$
```

11. เพิ่มไฟล์ใน VSCode

```
└─ NODELOGIN [SSH: 10.211.55.5]
    └── node_modules
    └── static
        # style.css
        < login.html
        JS login.js
        {} package-lock.json
        {} package.json
```

- เพิ่มโค้ดไฟล์ตามลักษณะ

<https://codeshack.io/basic-login-system-nodejs-express-mysql/>

- แก้ไขข้อมูลบางส่วนในไฟล์ login.js (ตามที่ไฮไลต์)

```
JS login.js > [e] connection
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
```

```
71
72     app.listen(3003);
```

- แก้ไขข้อมูลบางส่วนในไฟล์ package.json (ตามที่ไฮไลต์)

```
{} package.json > ...
1  {
2      "name": "nodelogin",
3      "version": "1.0.0",
4      "description": "",
5      "main": "login.js",
6      > Debug
7      "scripts": {
8          "test": "echo \"Error: no test specified\" && exit 1"
9      },
10     "author": "",
11     "license": "ISC",
12     "dependencies": {
13         "express": "^4.18.2",
14         "express-session": "^1.17.3",
15         "mysql": "^2.18.1"
16     }
17 }
```

12. sudo ufw allow 3003

```
devuser@webserver097:/var/www/lab4/nodelogin$ sudo ufw allow 3003
Rule added
Rule added (v6)
```

13. sudo mysql -u root -p

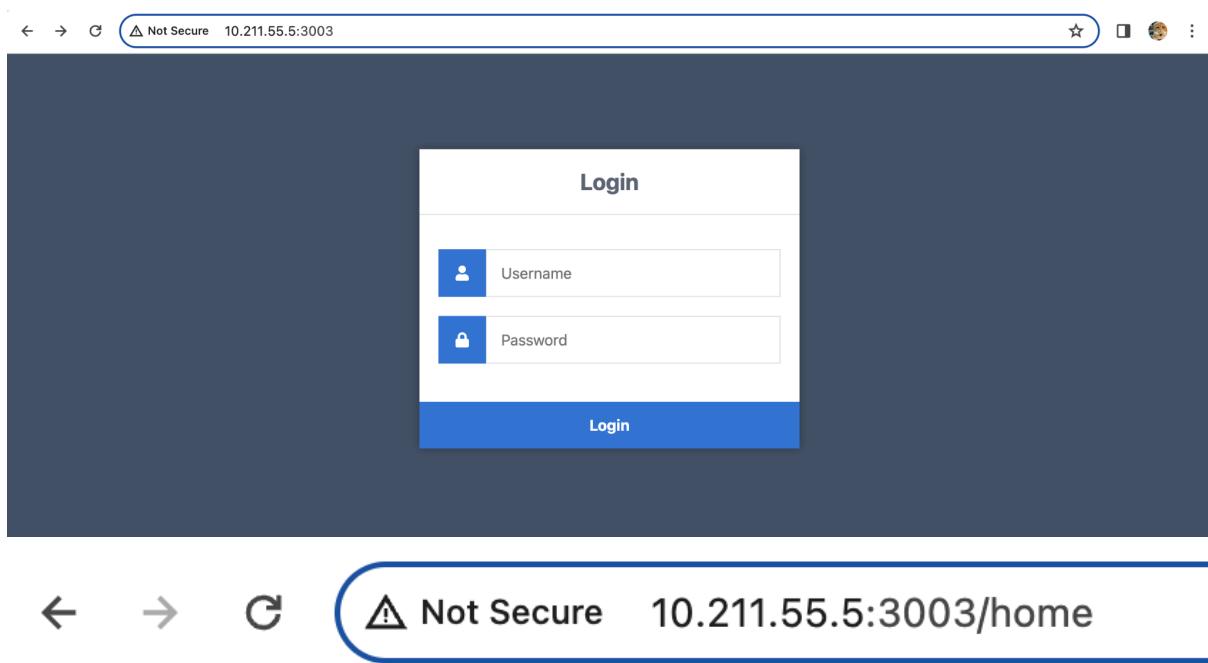
```
devuser@webserver097:/var/www/lab4/nodelogin$ sudo mysql -u root -p
```

14. ALTER USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd@2023';

```
mysql> ALTER USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd@2023';
Query OK, 0 rows affected (0.01 sec)
```

15. node login.js

```
devuser@webserver097:/var/www/lab4/nodelogin$ node login.js
```



Welcome back, test!

16. cd /etc/nginx/sites-available

```
devuser@webserver097:~$ cd /etc/nginx/sites-available
```

17. sudo nano nodelogin

```
devuser@webserver097:/etc/nginx/sites-available$ sudo nano nodelogin
```

18. sudo systemctl restart nginx

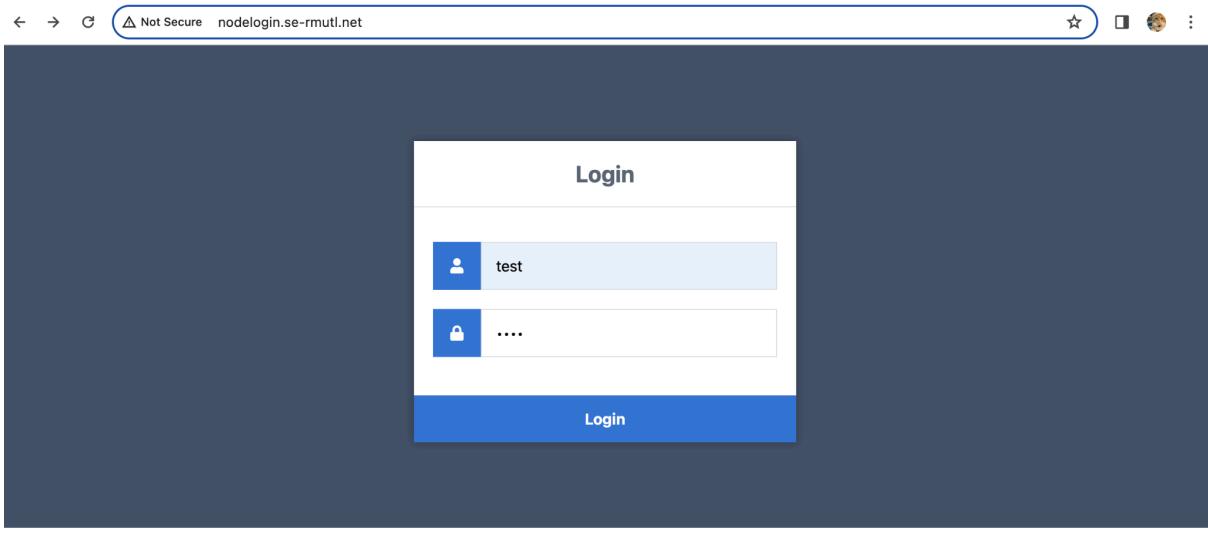
```
devuser@webserver097:/etc/nginx/sites-available$ sudo systemctl restart nginx
```

19. sudo nginx -t

```
devuser@webserver097:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

20. sudo nano /etc/hosts

```
10.211.55.5      nodelogin.se-rmutil.net
```



← → G Not Secure nodelogin.se-rmutil.net/home

Welcome back, test!

Step6 Create Login System with Node.js App (Client/Server)

My Source code :

<https://github.com/D5en15/A.Thanit-Lab4.git>

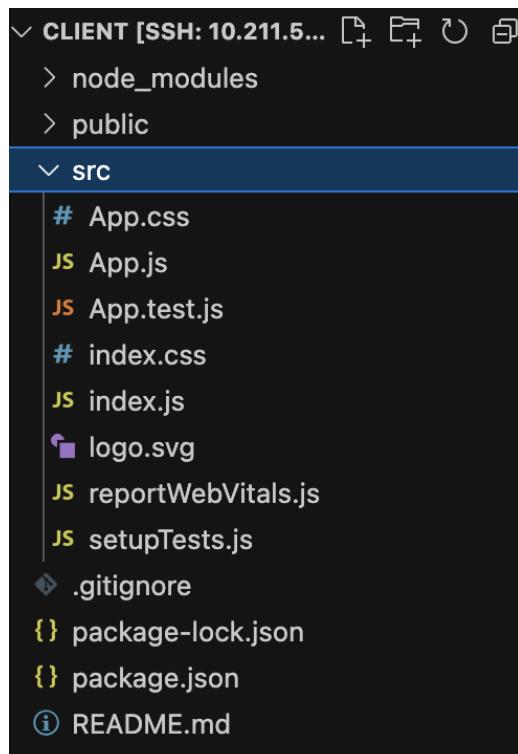
6.1 Client

1. เข้า cd /var/www/lab4

- mkdir client
- cd client
- npx create-react-app .

```
devuser@webserver097:/var/www/lab4$ mkdir client
devuser@webserver097:/var/www/lab4$ ls
client nodelogin
devuser@webserver097:/var/www/lab4$ cd client
devuser@webserver097:/var/www/lab4/client$ npx create-react-app .
```

2. Connect VS Code



3. นำโค้ดเข้าใส่ใน VSCode

<https://github.com/gsandamali/Part-1-User-registration-and-login-without-JWT>

client/src/

```
# App.css
JS App.js

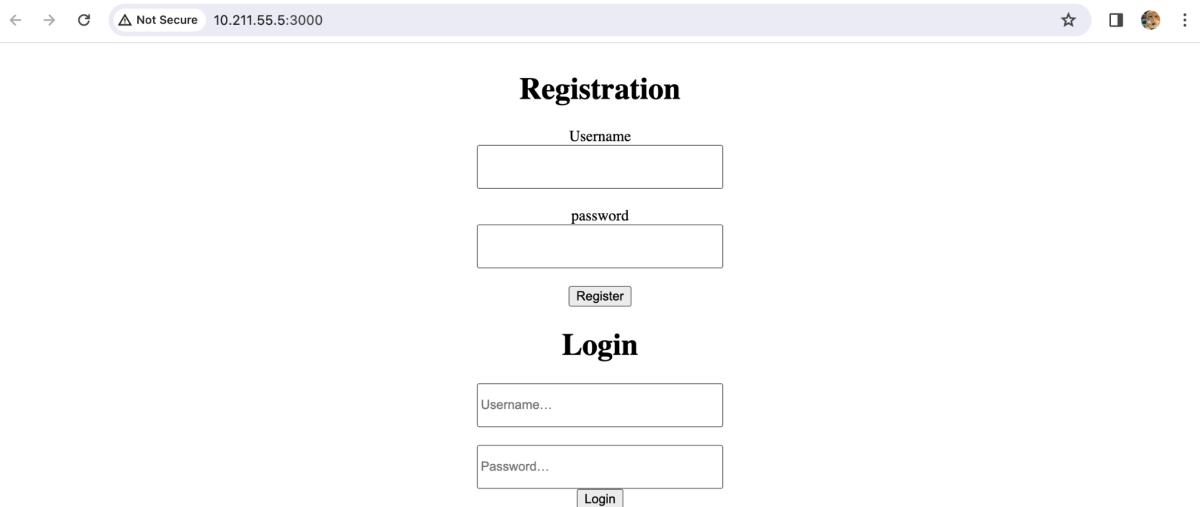
JS index.js
```

4. เพิ่ม npm i --save axios

```
devuser@webserver097:/var/www/lab4/client$ npm i --save axios
```

5. npm start

```
devuser@webserver097:/var/www/lab4/client$ npm start
```



6. เปลี่ยน PORT เป็น 3004

```
{ package.json }
```

```
"scripts": {  
  "start": "PORT=3004 react-scripts start",
```

```
sudo ufw allow 3004
```

7. cd /etc/nginx/sites-available

```
sudo cp default client
```

```
sudo nano client
```

```
devuser@webserver097:~$ cd /etc/nginx/sites-available  
devuser@webserver097:/etc/nginx/sites-available$ sudo cp default client  
devuser@webserver097:/etc/nginx/sites-available$ sudo nano 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;  
  }  
}
```

```

    }

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;
    }
}

```

8. sudo systemctl restart nginx

sudo nginx -t

```

devuser@webserver097:/etc/nginx/sites-available$ sudo systemctl restart nginx
devuser@webserver097:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

```

9. sudo nano /etc/hosts

10.211.55.5 weblogin.se-rmutil.net

10. cd /var/www/lab4/client/

npm start

```

Compiled successfully!

You can now view client in the browser.

Local:          http://localhost:3004
On Your Network: http://10.211.55.5:3004

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

webpack compiled successfully

```

Registration

Username	<input type="text"/>
password	<input type="password"/>
<input type="button" value="Register"/>	

Login

Username...
Password...
<input type="button" value="Login"/>

6.2 Server

1. cd /var/www/lab4

- mkdir server
- cd server
- npm init

```
devuser@webserver097:~$ cd /var/www/lab4
devuser@webserver097:/var/www/lab4$ mkdir server
devuser@webserver097:/var/www/lab4$ cd server
devuser@webserver097:/var/www/lab4/server$ npm init
```

2. npm install express --save

- npm install --save mysql2
- npm i --save cors

```
devuser@webserver097:/var/www/lab4/server$ npm install express --save
added 62 packages, and audited 63 packages in 3s

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

found 0 vulnerabilities
devuser@webserver097:/var/www/lab4/server$ 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@webserver097:/var/www/lab4/server$ npm i --save cors
added 2 packages, and audited 76 packages in 1s

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

found 0 vulnerabilities
```

3. Connect VS Code

```
✓ SERVER [SSH: 10.211.55.5]
  > node_modules
    JS index.js
    {} package-lock.json
    {} package.json
```

4. sudo ufw allow 4000

```
devuser@webserver097:/var/www/lab4/server$ sudo ufw allow 4000
[sudo] password for devuser:
Rule added
Rule added (v6)
```

5. sudo mysql -u root -p

```
devuser@webserver097:/var/www/lab4/server$ sudo mysql -u root -p
```

- CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
- USE loginsystem;
- CREATE TABLE IF NOT EXISTS `users`(`username` varchar(50) NOT NULL, `password` varchar(500) NOT NULL, PRIMARY KEY (`username`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
- USE loginsystem;
- INSERT INTO `users`(`username`, `password`) VALUES ('test', 'test');
- GRANT ALL ON loginsystem.* TO 'example_user'@'%';

```
mysql> CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 3 warnings (0.00 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> USE loginsystem;
Database changed
mysql> INSERT INTO `users`(`username`, `password`) VALUES ('test', 'test');
Query OK, 1 row affected (0.00 sec)

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

mysql> exit
Bye
```

6. node index.js

ตรวจสอบว่าติดต่อกับ Server ได้หรือไม่

```
devuser@webserver097:/var/www/lab4/server$ node index.js
running server
Hello from API
Hello from API
```

7. แก้ไข port ใน App.js ฝั่ง client ให้เป็น port 4000

```
client > src > App.js > index.js
1 import React, { 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      if (!response.data.err) {
17        setLoginStatus(""+usernameReg + "' was inserted.");
18      } else {
19        setLoginStatus(response.data.err.sqlMessage);
20      }
21    });
22  };
23  const login = () => {
24    Axios.post("http://10.211.55.5:4000/login", {
25      username: username,
26      password: password,
27    }).then((response) => {
28      console.log(response);
29      //console.log("response.data.message: "+response.data[0].message);
30
31      if (!response.data.message) {
32        setLoginStatus(""+response.data[0].username + "' was logged in.");
33      } else {
34        setLoginStatus(response.data.message);
35      }
36    });
37  };
38  };
39
40  return (
41    <div className="App">
42      <div className="registration">
43        <h1>Registration</h1>
44        <label>Username</label>
45        <input type="text" value={usernameReg} onChange={(e) => setUsernameReg(e.target.value)} />
46      </div>
47      <div className="login">
48        <h1>Login</h1>
49        <input type="text" value="test" onChange={(e) => setUsername(e.target.value)} />
50        <input type="password" value="...." onChange={(e) => setPassword(e.target.value)} />
51        <button type="button" onClick={() => login()}>Login</button>
52      </div>
53    </div>
54  );
55}
56
57
```

8. เข้ารันในเว็บ และให้ลองล็อกอิน

- [server] node index.js
- [client] npm start

← → ⚡ Not Secure 10.211.55.5:3004



Registration

Username

password

Register

Login

Login

'test' was logged in.

9. ให้รันแบบโดยเม้นโดยเข้าไปแก้พอร์ตใน server/index.js

origin: ["http://10.211.55.5:3004"], => origin: ["<http://weblogin.se-rmutil.net>"],

The screenshot shows a terminal window titled "lab4 [SSH: 10.211.55.5]". The left pane is an "EXPLORER" view showing a project structure under "LAB4 [SSH: 10.211.55.5]". The "server" folder contains "index.js", which is the current file being edited. The right pane shows the code for "index.js". The code sets up an Express app, connects to a MySQL database, and handles CORS for requests from "http://weblogin.se-rmutil.net". It also defines a route for the root path that logs "Hello from API".

```
server > ↗ index.js > ⚡ app.post('/register') callback
  1  const express = require("express");
  2  const mysql = require("mysql2");
  3  const cors = require("cors");
  4
  5
  6  const app = express();
  7
  8
  9  const db = mysql.createConnection({
 10    host: 'localhost',
 11    user: 'example_user',
 12    password: 'P@ssw0rd@2023',
 13    database: 'loginsystem',
 14  });
 15  app.use(express.json());
 16  app.use(
 17    cors({
 18      origin: ["http://weblogin.se-rmutil.net"],
 19      methods: ["GET", "POST"],
 20      credentials: true,
 21    })
 22  );
 23
 24  app.get('/', (req, res) => {
 25    console.log("Hello from API");
 26  });
 27
```

← → ⚡ Not Secure weblogin.se-rmutil.net ☆ 🌐 ⋮

Registration

Username

password

Register

Login

Login

'test' was logged in.

Step7 Create Login System with Node.js App (Server)

1. เพิ่ม User ใน backend ***ถ้ามีปัญหา***

- sudo -i
- adduser fang
- usermod -aG sudo fang
- ufw allow OpenSSH
- ufw enable
- ufw status

```
root@backend097:~# adduser fang
Adding user `fang' ...
Adding new group `fang' (1001) ...
Adding new user `fang' (1001) with group `fang' ...
Creating home directory `/home/fang' ...
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 fang
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@backend097:~# usermod -aG sudo fang
root@backend097:~# ufw allow OpenSSH
Skipping adding existing rule
Skipping adding existing rule (v6)
root@backend097:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@backend097:~# ufw status
Status: active

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

root@backend097:~# exit
logout
```

2. cd /var/www/lab4

- sudo chown -R \$USER:\$USER /var/www/lab4
- mkdir server
- cd server
- npm init

```
devuser@backend097:~$ sudo chown -R $USER:$USER /var/www/lab4
[sudo] password for devuser:
devuser@backend097:~$ cd /var/www/lab4
devuser@backend097:/var/www/lab4$ mkdir server
devuser@backend097:/var/www/lab4$ cd server
devuser@backend097:/var/www/lab4/server$ npm init
```

3. npm install express --save

- npm install --save mysql2
- npm i --save cors

```
devuser@backend097:/var/www/lab4/server$ 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@backend097:/var/www/lab4/server$ npm install --save mysql2
added 11 packages, and audited 74 packages in 898ms

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

found 0 vulnerabilities
devuser@backend097:/var/www/lab4/server$ npm i --save cors
added 2 packages, and audited 76 packages in 660ms

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

found 0 vulnerabilities
```

4. sudo mysql -u root -p

- CREATE USER 'example_user'@'%' IDENTIFIED BY 'P@sswOrd@2023';
- ALTER USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrd@2023';
- CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
- USE loginsystem;
- CREATE TABLE IF NOT EXISTS `users`(`username` varchar(50) NOT NULL, `password` varchar(500) NOT NULL, PRIMARY KEY (`username`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
- USE loginsystem;
- INSERT INTO `users` (`username`, `password`) VALUES ('test', 'test');
- GRANT ALL ON loginsystem.* TO 'example_user'@'%';

```
mysql> CREATE USER 'example_user'@'%' IDENTIFIED BY 'P@sswOrd@2023';
Query OK, 0 rows affected (0.02 sec)

mysql> ALTER USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrd@2023';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0.00 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 (0.01 sec)

mysql> USE loginsystem;
Database changed
mysql> INSERT INTO `users` (`username`, `password`) VALUES ('test', 'test');
Query OK, 1 row affected (0.01 sec)

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

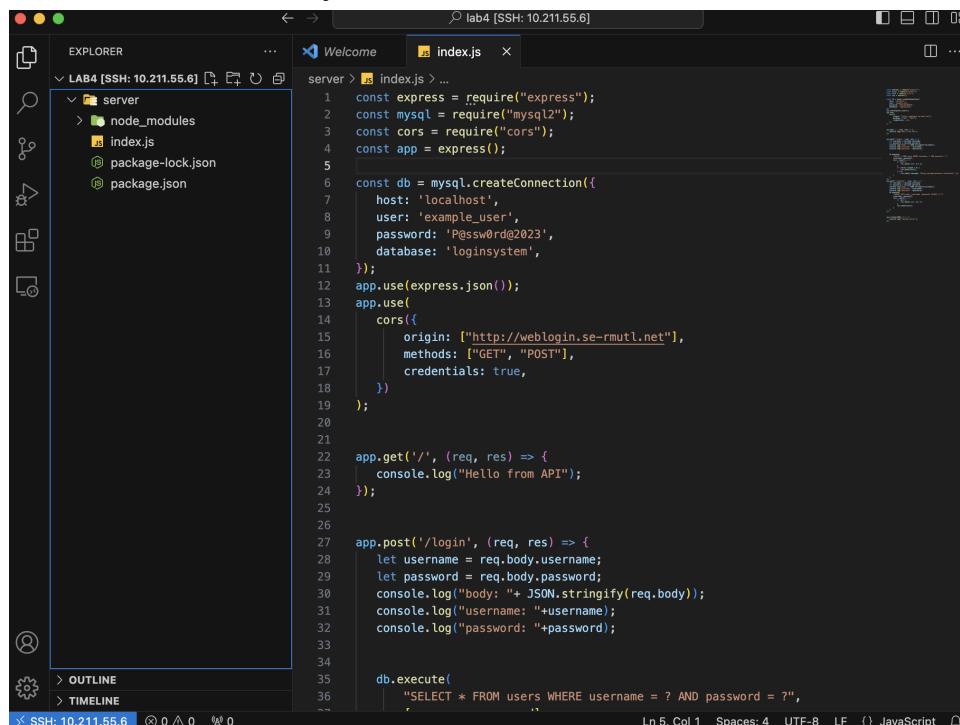
mysql> exit
Bye
```

5. sudo ufw allow 4000

```
devuser@backend097:/var/www/lab4/server$ sudo ufw allow 4000
Rule added
Rule added (v6)
```

6. Connect VS Code

สร้างไฟล์ index.js นำโค้ดของ web server เข้ามา



The screenshot shows the VS Code interface connected via SSH to a server at 10.211.55.6. The Explorer sidebar shows a folder named 'LAB4' containing 'server', 'node_modules', 'index.js', 'package-lock.json', and 'package.json'. The main editor tab is titled 'index.js' and contains the following code:

```
const express = require("express");
const mysql = require("mysql2");
const cors = require("cors");
const app = express();

const db = mysql.createConnection({
  host: 'localhost',
  user: 'example_user',
  password: '@ssw0rd2023',
  database: 'loginsystem',
});

app.use(express.json());
app.use(cors({
  origin: ["http://weblogin.ser-mutl.net"],
  methods: ["GET", "POST"],
  credentials: true,
}));

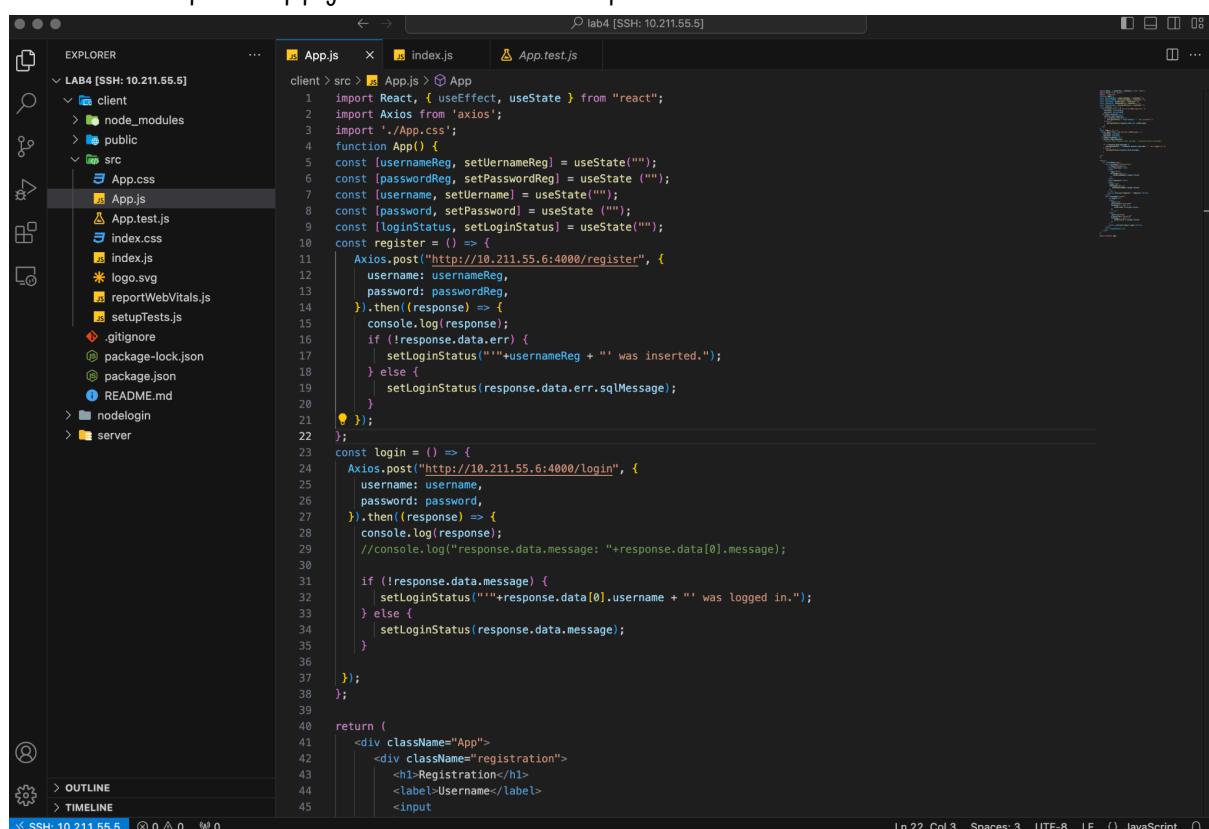
app.get('/', (req, res) => {
  console.log("Hello from API");
});

app.post('/login', (req, res) => {
  let username = req.body.username;
  let password = req.body.password;
  console.log("body: " + JSON.stringify(req.body));
  console.log("username: " + username);
  console.log("password: " + password);

  db.execute(
    "SELECT * FROM users WHERE username = ? AND password = ?",
    [username, password]
  );
  res.json({ message: "User logged in" });
});

db.on("error", (err) => {
  console.error(`Database error: ${err.message}`);
});
```

แก้ไข ip ใน App.js ผัง client ให้เป็น ip 10.211.55.6 ตามของ server



The screenshot shows the VS Code interface connected via SSH to a client at 10.211.55.5. The Explorer sidebar shows a folder named 'LAB4' containing 'client', 'node_modules', 'public', and 'src'. Under 'src', there are files: 'App.css', 'App.js', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js', '.gitignore', 'package-lock.json', 'package.json', and 'README.md'. The main editor tab is titled 'App.js' and contains the following code:

```
import React, { useState } from "react";
import Axios from "axios";
import './App.css';

function App() {
  const [usernameReg, setUsernameReg] = useState("");
  const [passwordReg, setPasswordReg] = useState("");
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const [loginStatus, setLoginStatus] = useState("");

  const register = () => {
    Axios.post("http://10.211.55.6:4000/register", {
      username: usernameReg,
      password: passwordReg,
    }).then((response) => {
      console.log(response);
      if (!response.data.err) {
        setUsernameReg("") + usernameReg + " was inserted.");
      } else {
        setLoginStatus(response.data.err.sqlMessage);
      }
    });
  };

  const login = () => {
    Axios.post("http://10.211.55.6:4000/login", {
      username: username,
      password: password,
    }).then((response) => {
      console.log(response);
      //console.log("response.data.message: "+response.data[0].message);

      if (!response.data.message) {
        setLoginStatus(""+response.data[0].username + " was logged in.");
      } else {
        setLoginStatus(response.data.message);
      }
    });
  };
}

return (
  <div className="App">
    <div className="registration">
      <h1>Registration</h1>
      <label>Username</label>
      <input type="text" value={usernameReg} onChange={(e) => setUsernameReg(e.target.value)} />
    </div>
  </div>
);
```

7. รันทิ้ง webserver และ backend server และให้รันแบบโดเมน และให้เข้าสู่ระบบดู

The screenshot shows a web browser window with the URL `weblogin.se-rmutil.net`. The page contains two forms: a registration form and a login form.

Registration Form:

- Username:
- password:
-

Login Form:

- Username:
- password:
-

'test' was logged in.

```
devuser@backend097:/var/www/lab4/server$ node index.js
running server
body: {"username":"test","password":"test"}
username: test
password: test

```