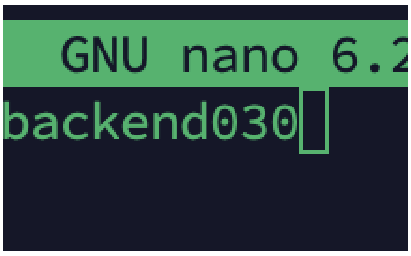
LAB4 back end server

IP:10.211.55.6 webserver030
IP:10.211.55.8 backend030

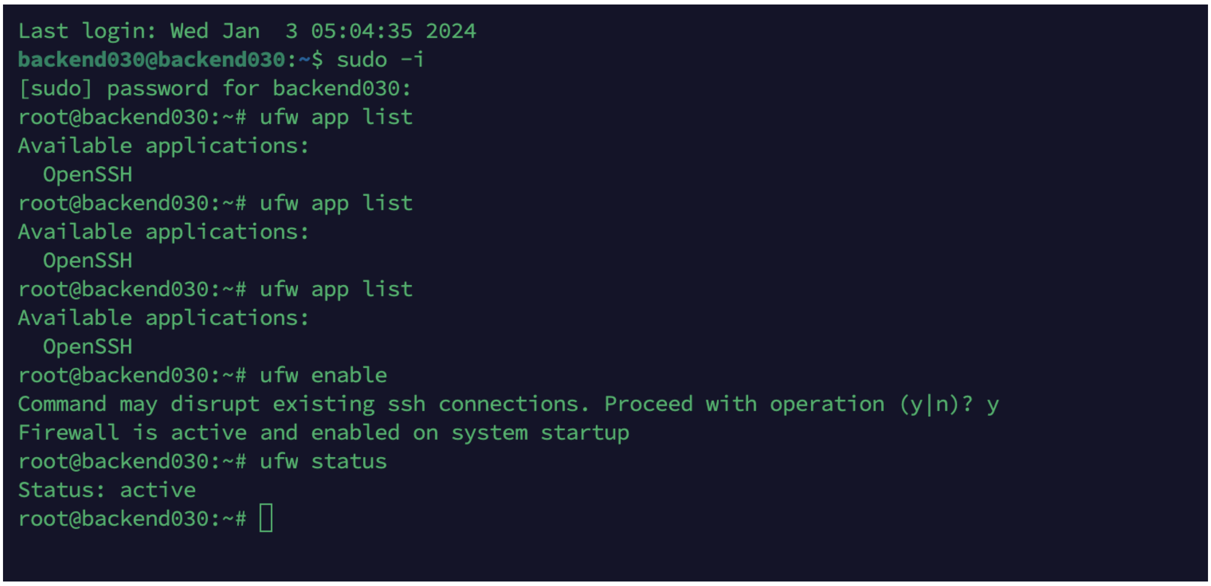
01 Ubuntu Linux Change Hostname (computer name)

```
sudo nano /etc/hostname
sudo nano /etc/hosts
sudo reboot
```



```
  GNU nano 6.2
backend030
```

02 Setting Up a Firewall



```
Last login: Wed Jan  3 05:04:35 2024
backend030@backend030:~$ sudo -i
[sudo] password for backend030:
root@backend030:~# ufw app list
Available applications:
  OpenSSH
root@backend030:~# ufw app list
Available applications:
  OpenSSH
root@backend030:~# ufw app list
Available applications:
  OpenSSH
root@backend030:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@backend030:~# ufw status
Status: active
root@backend030:~#
```

03 Installing MySQL

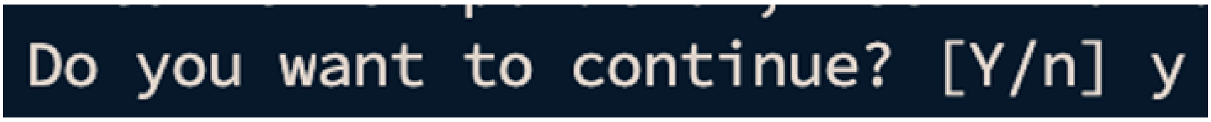
sudo apt update

sudo apt list --upgradable

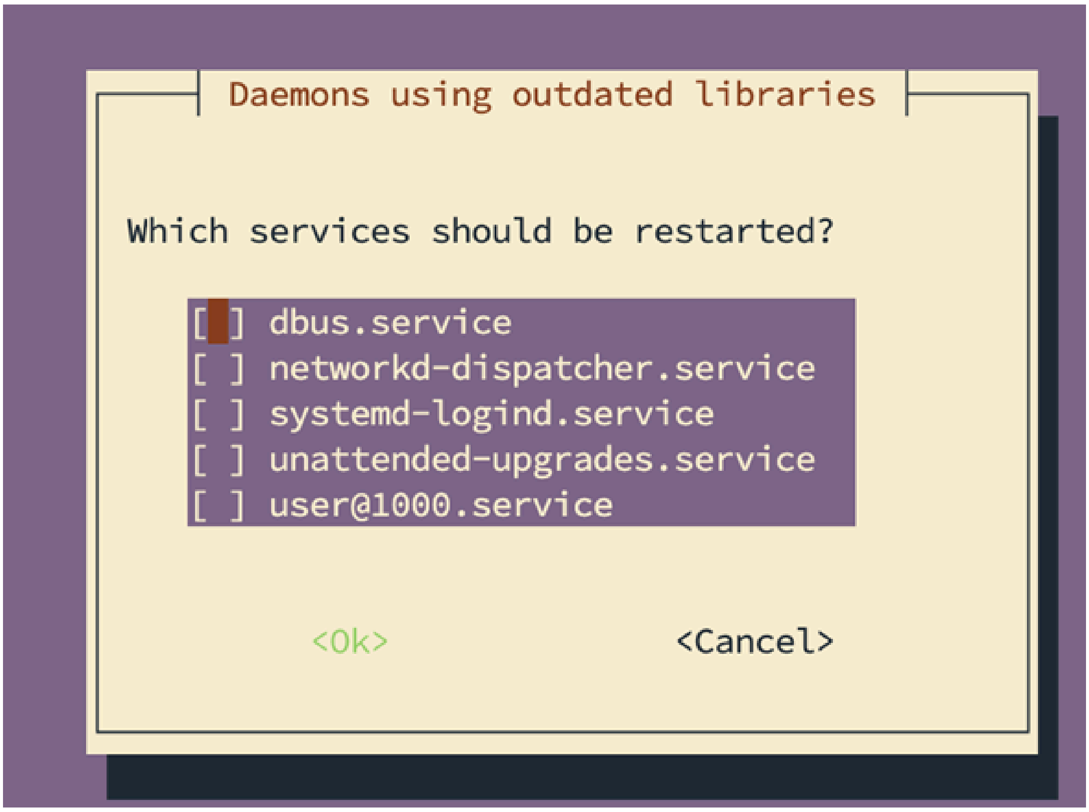
sudo apt upgrade

apt-cache searh mysql-server

sudo apt install mysql-server-8.0



```
Do you want to continue? [Y/n] y
```



```
Daemons using outdated libraries

Which services should be restarted?

[ ] dbus.service
[ ] networkd-dispatcher.service
[ ] systemd-logind.service
[ ] unattended-upgrades.service
[ ] user@1000.service
```

OK

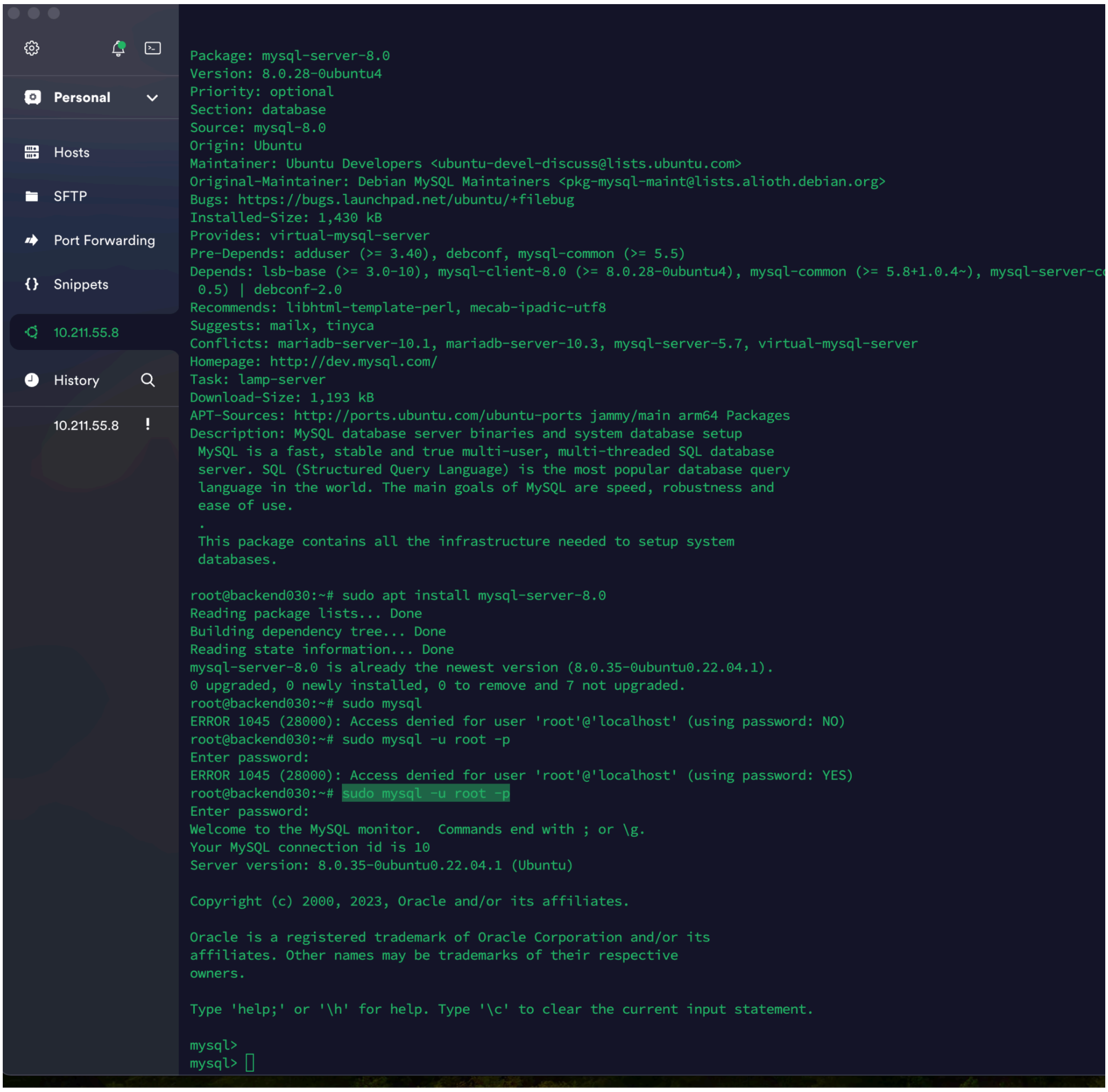
sudo apt update && sudo apt upgrade

sudo mysql

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

exit

sudo mysql -u root -p



```
Package: mysql-server-8.0
Version: 8.0.28-0ubuntu4
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: 1,430 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 (>= 8.0.28-0ubuntu4), mysql-common (>= 5.8+1.0.4~), mysql-server-c
 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: 1,193 kB
APT-Sources: http://ports.ubuntu.com/ubuntu-ports jammy/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.

root@backend030:~# 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 7 not upgraded.
root@backend030:~# sudo mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
root@backend030:~# sudo mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
root@backend030:~# sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
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>
mysql>
```

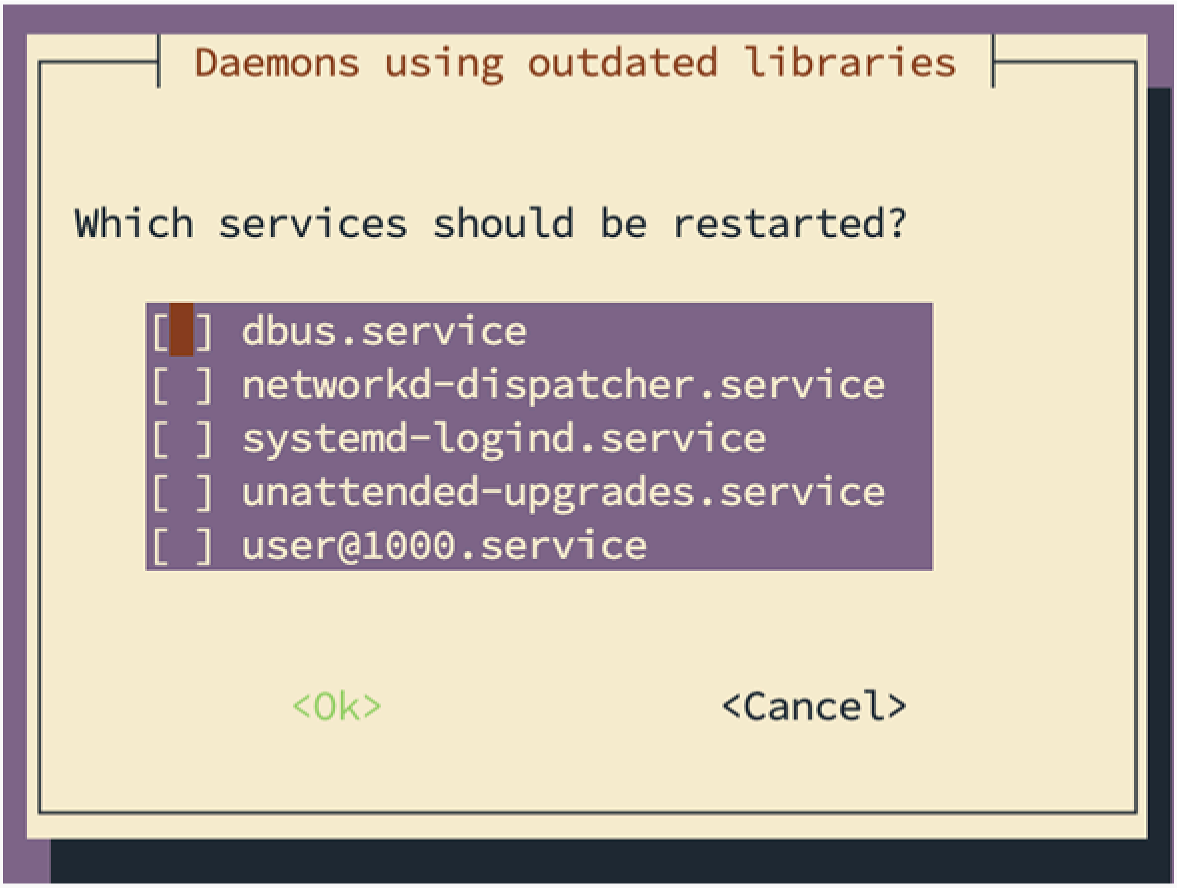
04 Install Nginx

— Update your system

```
sudo apt update && sudo apt upgrade
```

— Install Nginx, Certbot and Python3-Certbot-Nginx

```
sudo apt install nginx certbot python3-certbot-nginx
```



```
Daemons using outdated libraries

Which services should be restarted?

[ ] dbus.service
[ ] networkd-dispatcher.service
[ ] systemd-logind.service
[ ] unattended-upgrades.service
[ ] user@1000.service

<Ok>                    <Cancel>
```

— UFW Firewall Configuration

```
sudo ufw allow 'Nginx Full'
```

ทำการเพิ่ม nginx เข้า firewall

```
sudo apt-get update
```

แล้วก็ติดตั้ง certificate

sudo apt-get install -y ca=certificates curl gnupg

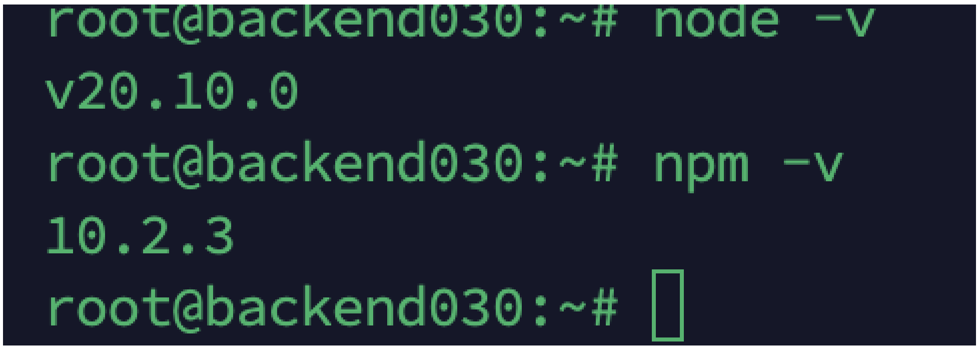
แล้วทำการสร้าง directory keyring

sudo mkdir -p /etc/apt/keyrimgs

Nodejs Installation

sudo apt-get update

sudo apt-get install nodejs -y

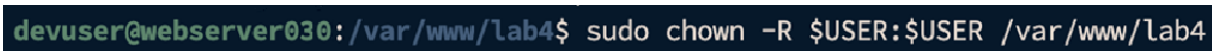


```
root@backend030:~# node -v
v20.10.0
root@backend030:~# npm -v
10.2.3
root@backend030:~# 
```

ทำการสร้าง lab4 และ nodelogin

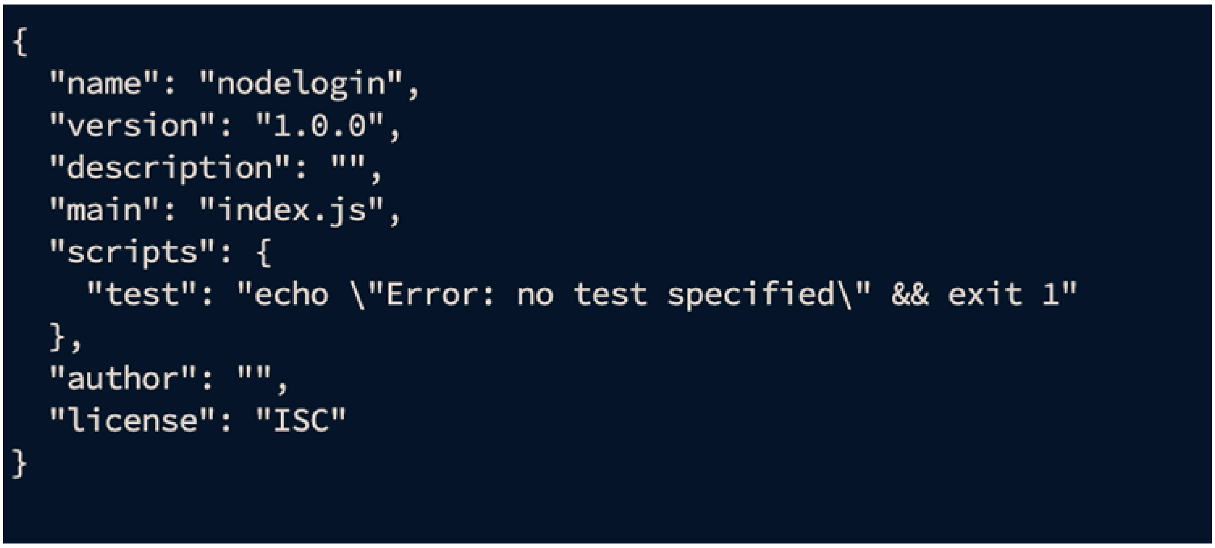
sudo mkdir -p /var/www/lab4/nodelogin

ต่อไปทำการ อนุญาติให้ user ธรรมดา แก้ไขไฟล์ต่างๆใน directory lab4 ได้

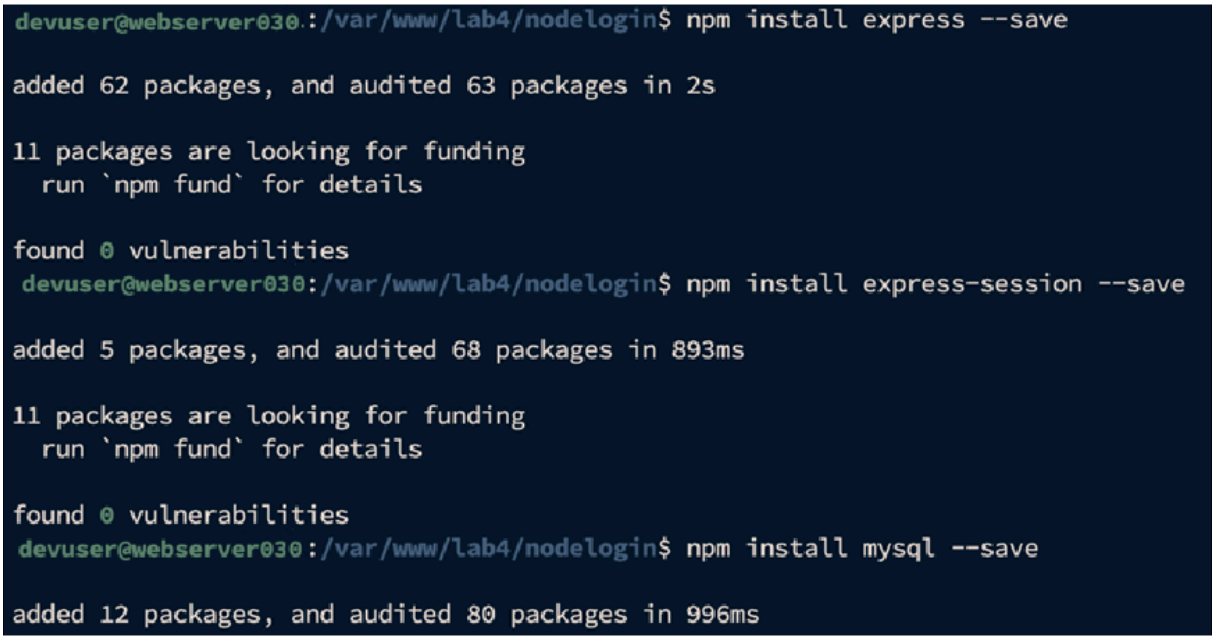


```
devuser@webserver030:/var/www/lab4$ sudo chown -R $USER:$USER /var/www/lab4
```

npm init



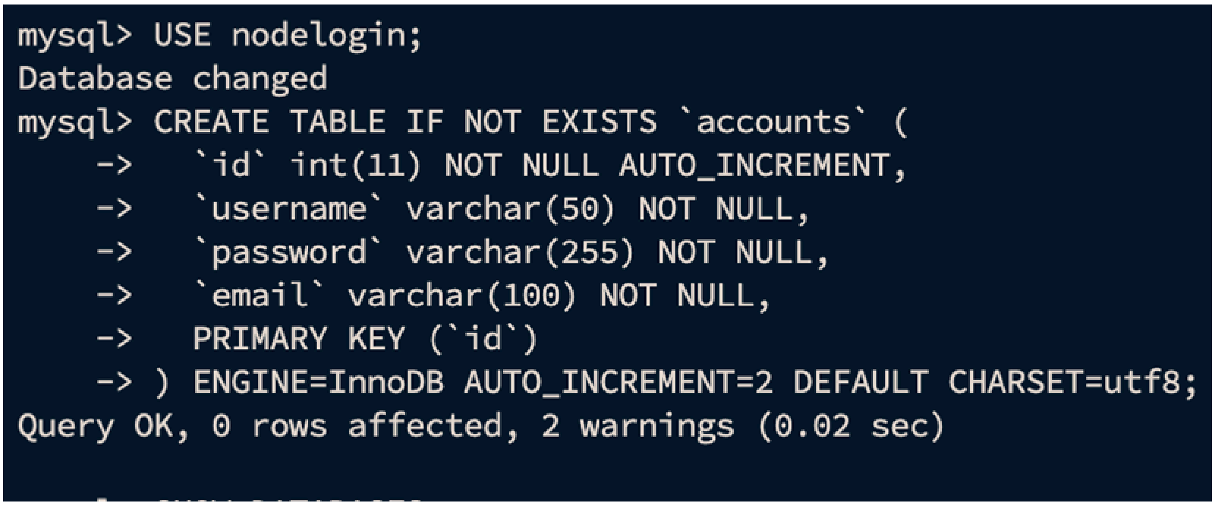
```
{
  "name": "nodelogin",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```



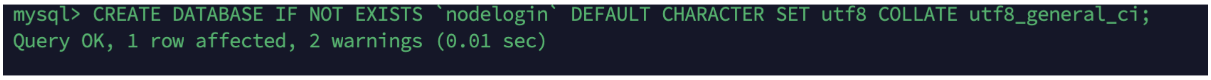
```
devuser@webserver030:/var/www/lab4/nodelogin$ npm install express --save

added 62 packages, and audited 63 packages in 2s

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

found 0 vulnerabilities
 devuser@webserver030:/var/www/lab4/nodelogin$ npm install express-session --save

added 5 packages, and audited 68 packages in 893ms

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

found 0 vulnerabilities
devuser@webserver030:/var/www/lab4/nodelogin$ npm install mysql --save

added 12 packages, and audited 80 packages in 996ms
```

เข้า mysql ด้วย user exmaple_user

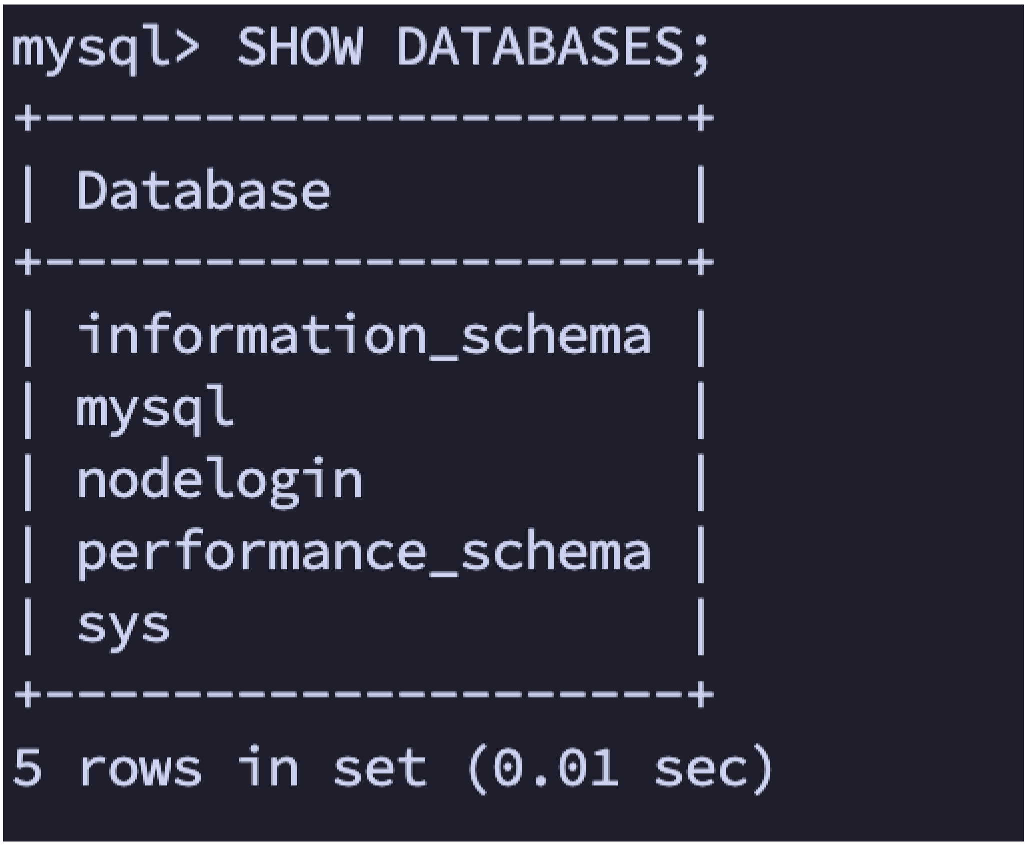
sudo mysql -u example_user -p



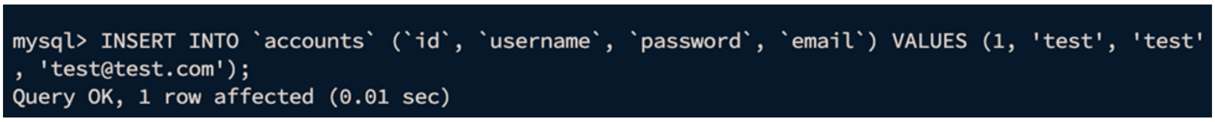
```
mysql> USE nodelogin;
Database changed
mysql> CREATE TABLE IF NOT EXISTS `accounts` (
    ->   `id` int(11) NOT NULL AUTO_INCREMENT,
    ->   `username` varchar(50) NOT NULL,
    ->   `password` varchar(255) NOT NULL,
    ->   `email` varchar(100) NOT NULL,
    ->   PRIMARY KEY (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
Query OK, 0 rows affected, 2 warnings (0.02 sec)
```



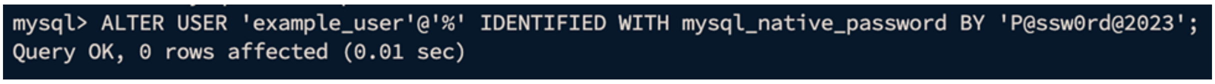
```
mysql> CREATE DATABASE IF NOT EXISTS `nodelogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0.01 sec)
```



```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| nodelogin          |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)
```

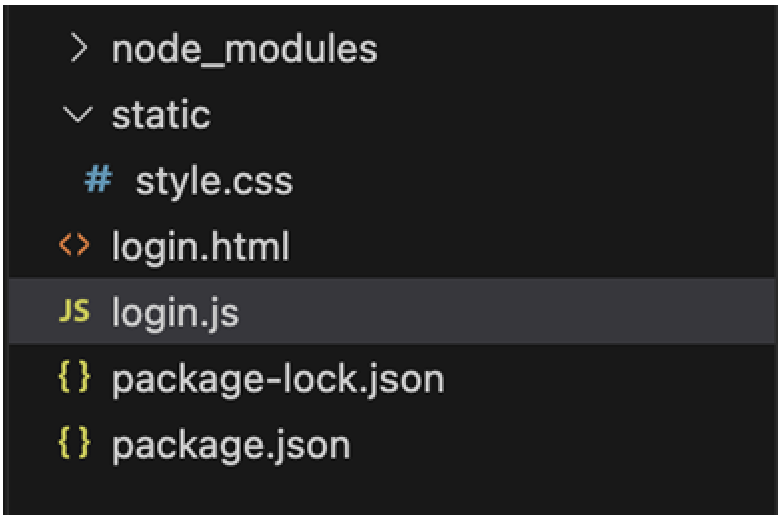


```
mysql> INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (1, 'test', 'test'
, 'test@test.com');
Query OK, 1 row affected (0.01 sec)
```

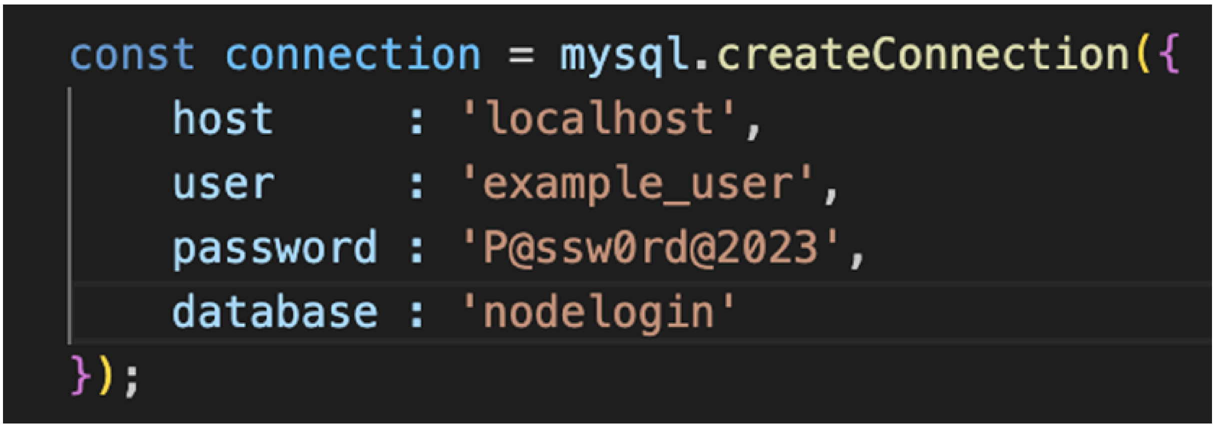


```
mysql> ALTER USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd@2023';
Query OK, 0 rows affected (0.01 sec)
```

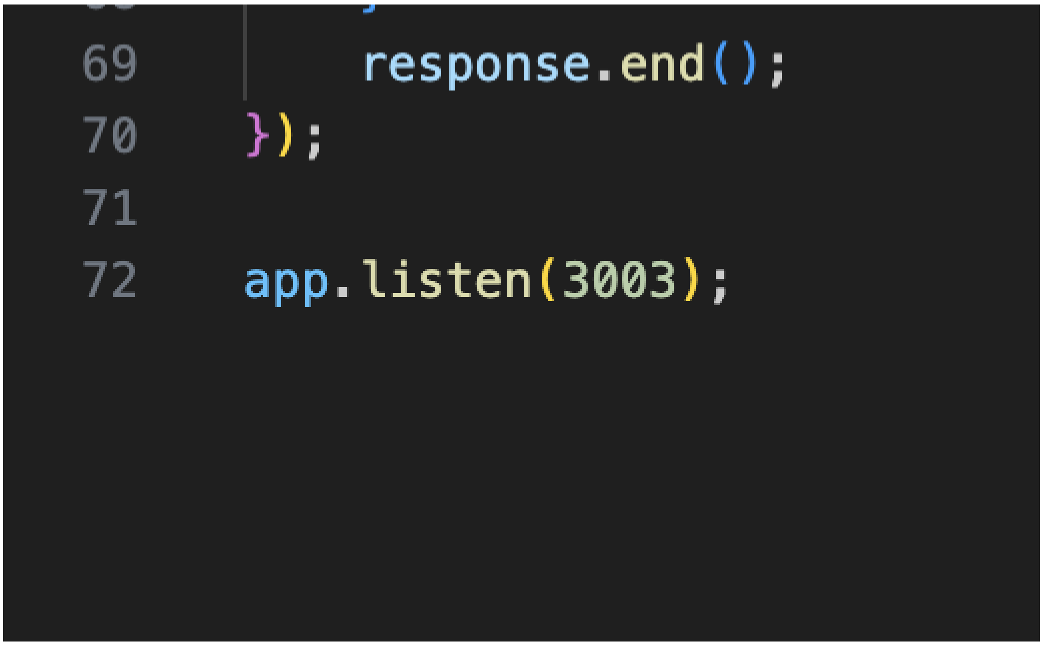
ทำการเชื่อม ssh 10.211.55.6 ใน VScode



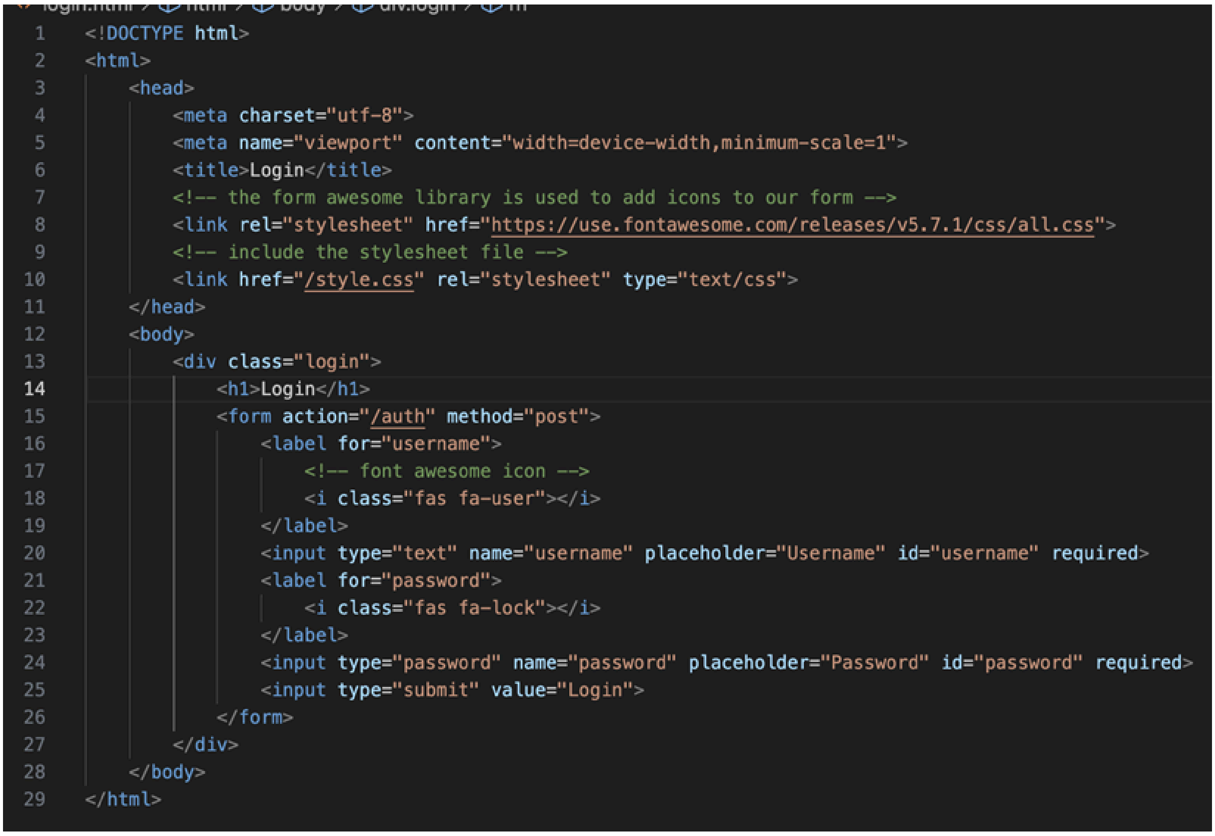
```
> node_modules
v static
   # style.css
<> login.html
JS login.js
{} package-lock.json
{} package.json
```



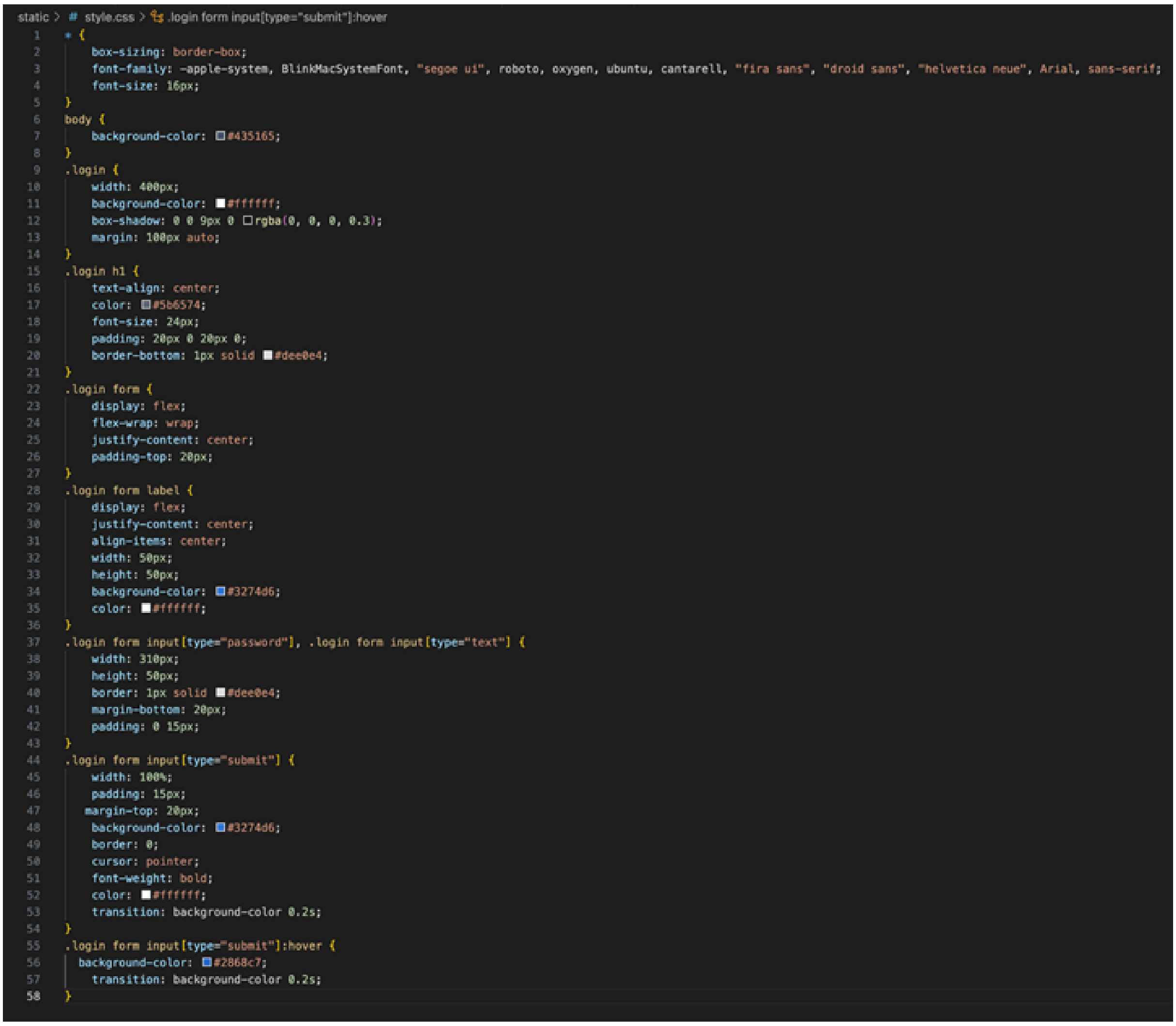
```
const connection = mysql.createConnection({
    host     : 'localhost',
    user     : 'example_user',
    password : 'P@ssw0rd@2023',
    database : 'nodelogin'
});
```



```
        response.end();
});

app.listen(3003);
```



```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,minimum-scale=1">
        <title>Login</title>
        <!-- the form awesome library is used to add icons to our form -->
        <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
        <!-- include the stylesheet file -->
        <link href="/style.css" rel="stylesheet" type="text/css">
    </head>
    <body>
        <div class="login">
            <h1>Login</h1>
            <form action="/auth" method="post">
                <label for="username">
                    <!-- font awesome icon -->
                    <i class="fas fa-user"></i>
                </label>
                <input type="text" name="username" placeholder="Username" id="username" required>
                <label for="password">
                    <i class="fas fa-lock"></i>
                </label>
                <input type="password" name="password" placeholder="Password" id="password" required>
                <input type="submit" value="Login">
            </form>
        </div>
    </body>
</html>
```

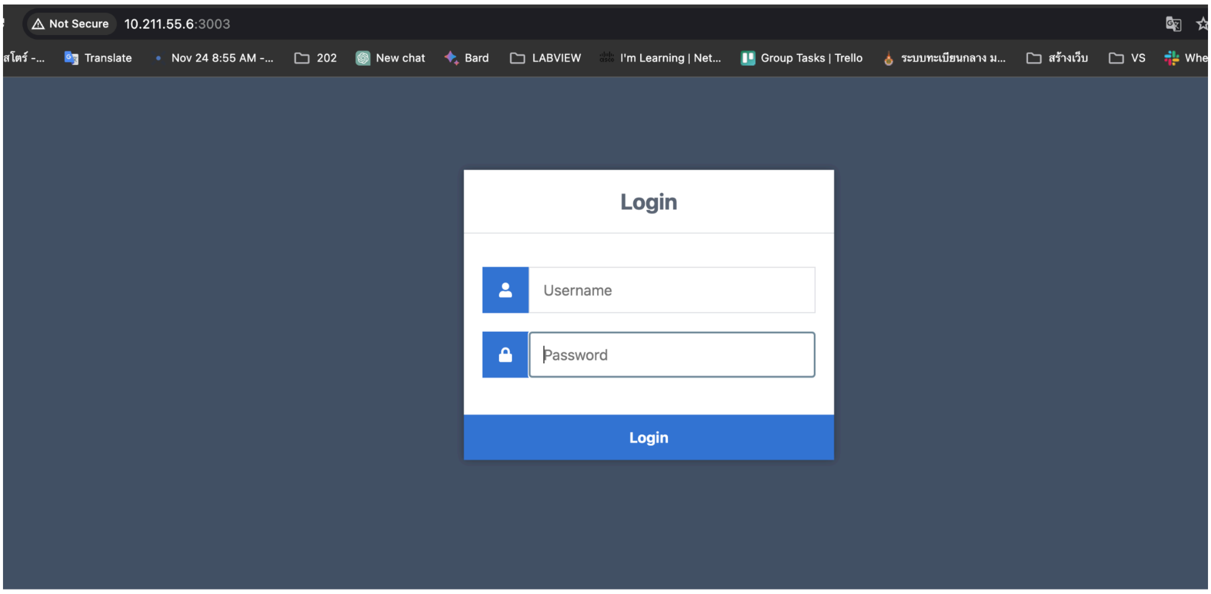


static > # style.css > .login form input[type="submit"]:hover

```css
* {
    box-sizing: border-box;
    font-family: -apple-system, BlinkMacSystemFont, "segoe ui", roboto, oxygen, ubuntu, cantarell, "fira sans", "droid sans", "helvetica neue", Arial, sans-serif;
    font-size: 16px;
}
body {
    background-color: #435165;
}
.login {
    width: 400px;
    background-color: #ffffff;
    box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);
    margin: 100px auto;
}
.login h1 {
    text-align: center;
    color: #5b6574;
    font-size: 24px;
    padding: 20px 0 20px 0;
    border-bottom: 1px solid #dee0e4;
}
.login form {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    padding-top: 20px;
}
.login form label {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 50px;
    height: 50px;
    background-color: #3274d6;
    color: #ffffff;
}
.login form input[type="password"], .login form input[type="text"] {
    width: 310px;
    height: 50px;
    border: 1px solid #dee0e4;
    margin-bottom: 20px;
    padding: 0 15px;
}
.login form input[type="submit"] {
    width: 100%;
    padding: 15px;
   margin-top: 20px;
    background-color: #3274d6;
    border: 0;
    cursor: pointer;
    font-weight: bold;
    color: #ffffff;
    transition: background-color 0.2s;
}
.login form input[type="submit"]:hover {
  background-color: #2868c7;
    transition: background-color 0.2s;
}
```

```
sudo ufw allow 3003
```

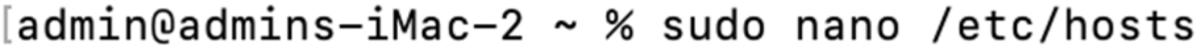
```
node login.js
```



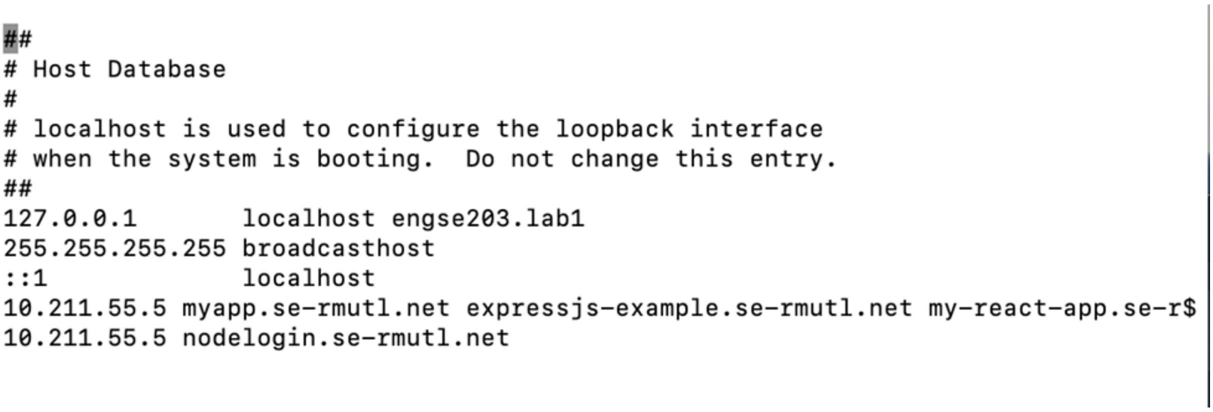
cd /etc/nginx/sites-available

sudo cp default nodelogin

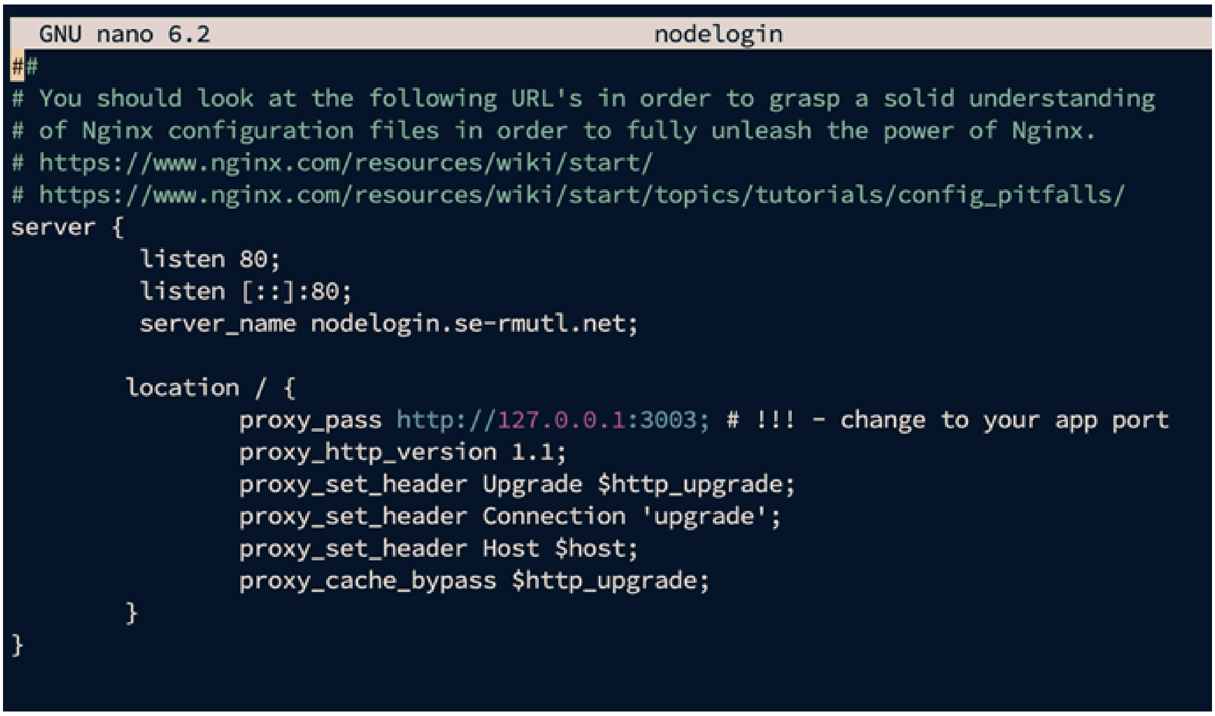
sudo nano nodelogin



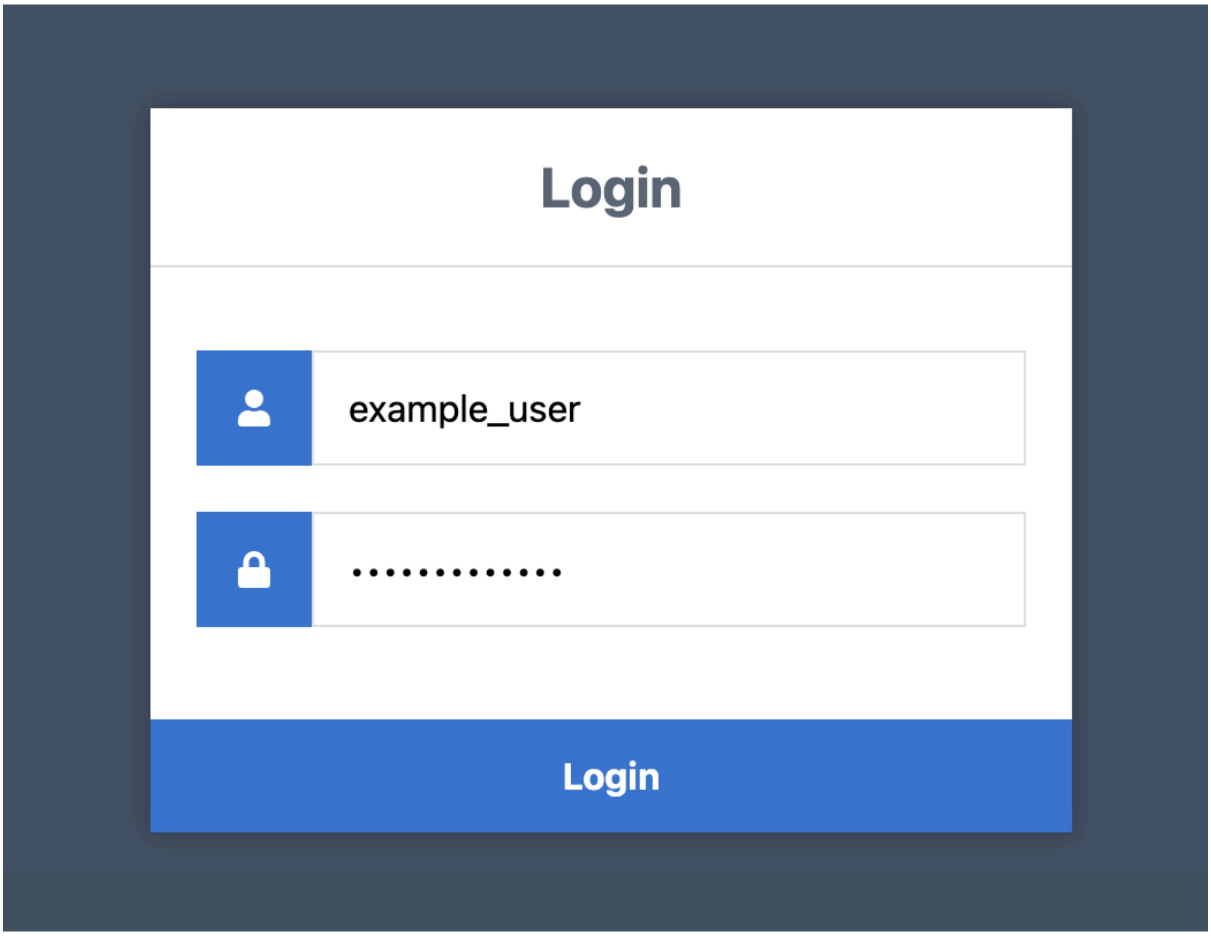
```
[admin@admins-iMac-2 ~ % sudo nano /etc/hosts
```



```
##
# 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
10.211.55.5 myapp.se-rmutl.net expressjs-example.se-rmutl.net my-react-app.se-r$
10.211.55.5 nodelogin.se-rmutl.net
```

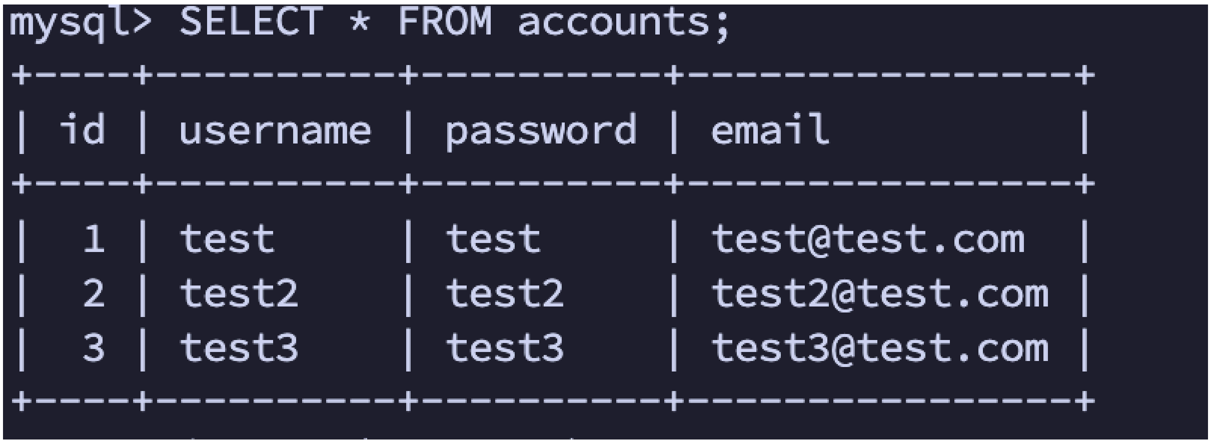


```
  GNU nano 6.2                                      nodelogin
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
server {
         listen 80;
         listen [::]:80;
         server_name nodelogin.se-rmutl.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;
        }
}
```



example_user

•••••••••••••



```
mysql> SELECT * FROM accounts;
+----+----------+----------+----------------+
| id | username | password | email          |
+----+----------+----------+----------------+
|  1 | test     | test     | test@test.com  |
|  2 | test2    | test2    | test2@test.com |
|  3 | test3    | test3    | test3@test.com |
+----+----------+----------+----------------+
```

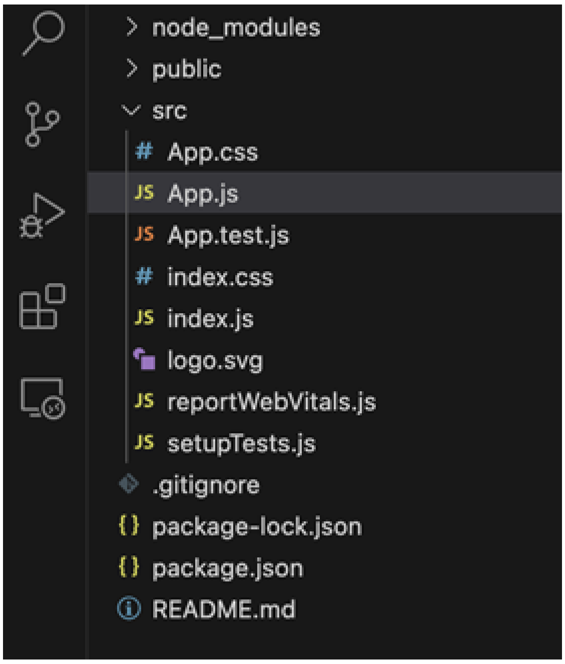
mkdir client

cd client

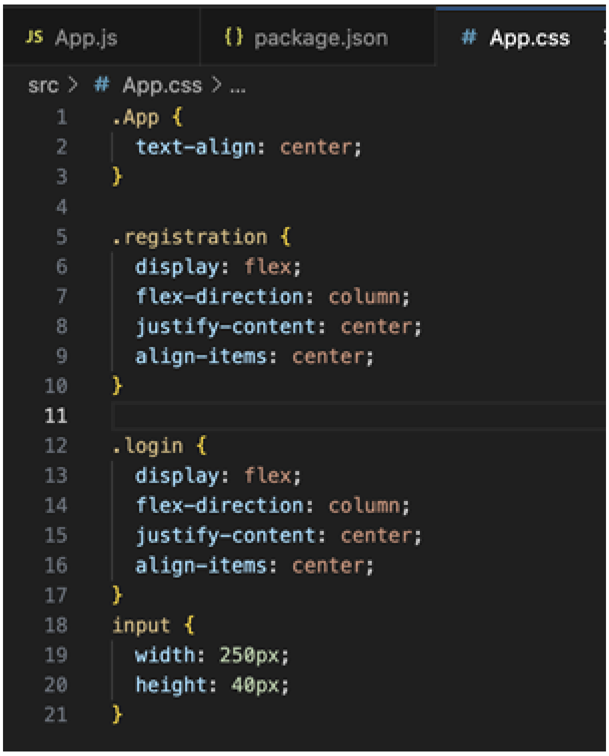
npx creale-react-app .

npm i –save axios

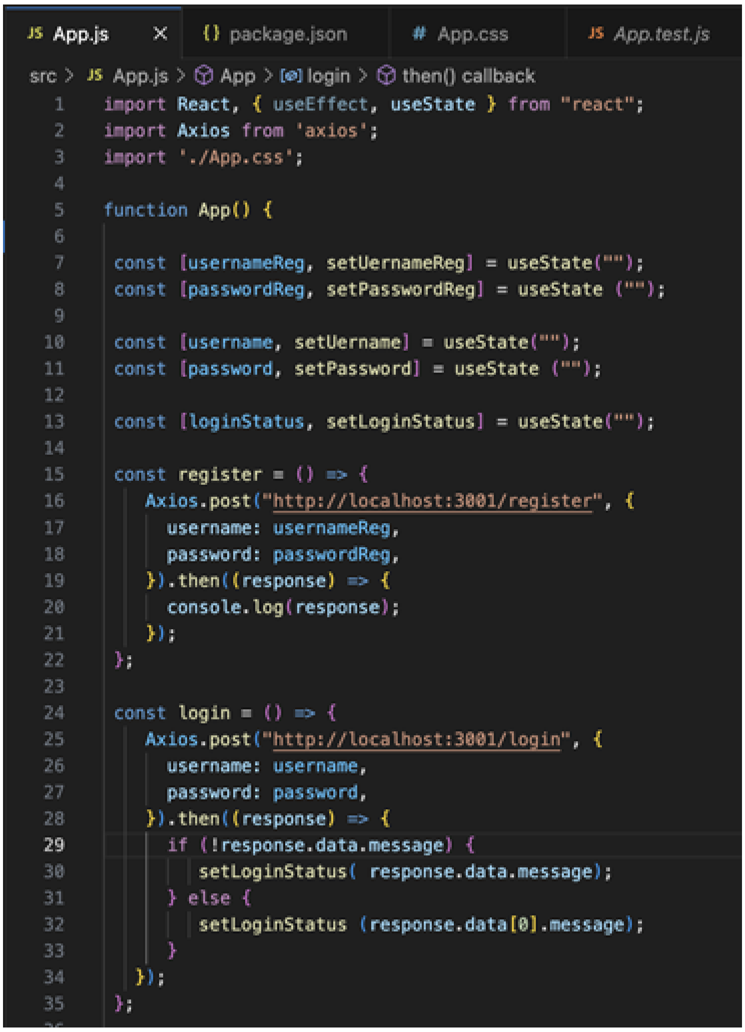
npm install express –sace



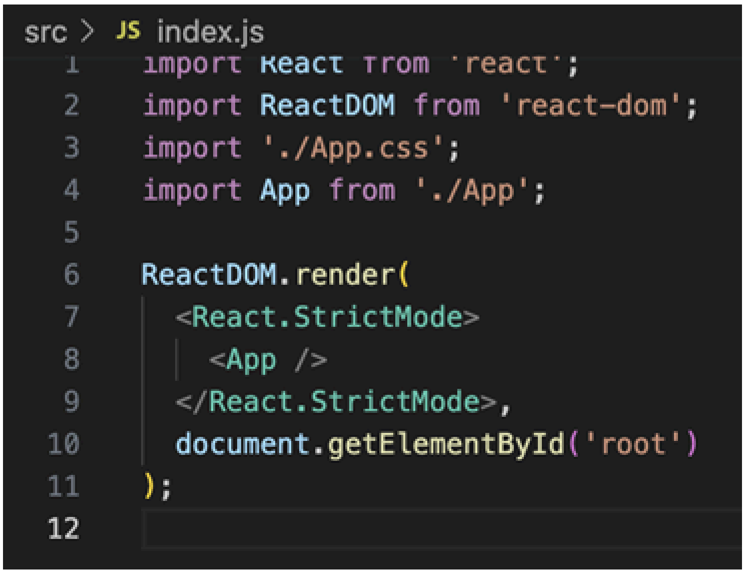
```
> node_modules
> public
v src
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```



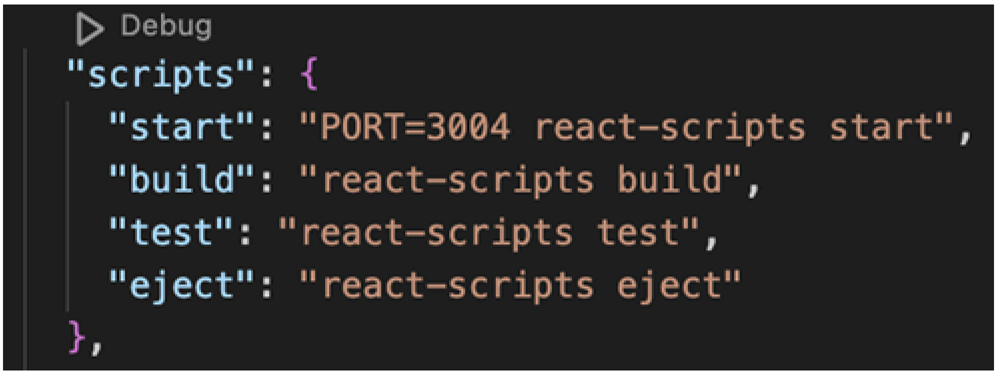
```css
.App {
  text-align: center;
}

.registration {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

.login {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
input {
  width: 250px;
  height: 40px;
}
```



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

function App() {

 const [usernameReg, setUernameReg] = useState("");
 const [passwordReg, setPasswordReg] = useState ("");

 const [username, setUername] = useState("");
 const [password, setPassword] = useState ("");

 const [loginStatus, setLoginStatus] = useState("");

 const register = () => {
    Axios.post("http://localhost:3001/register", {
      username: usernameReg,
      password: passwordReg,
    }).then((response) => {
      console.log(response);
    });
 };

 const login = () => {
    Axios.post("http://localhost:3001/login", {
      username: username,
      password: password,
    }).then((response) => {
      if (!response.data.message) {
         setLoginStatus( response.data.message);
      } else {
         setLoginStatus (response.data[0].message);
      }
   });
 };
```



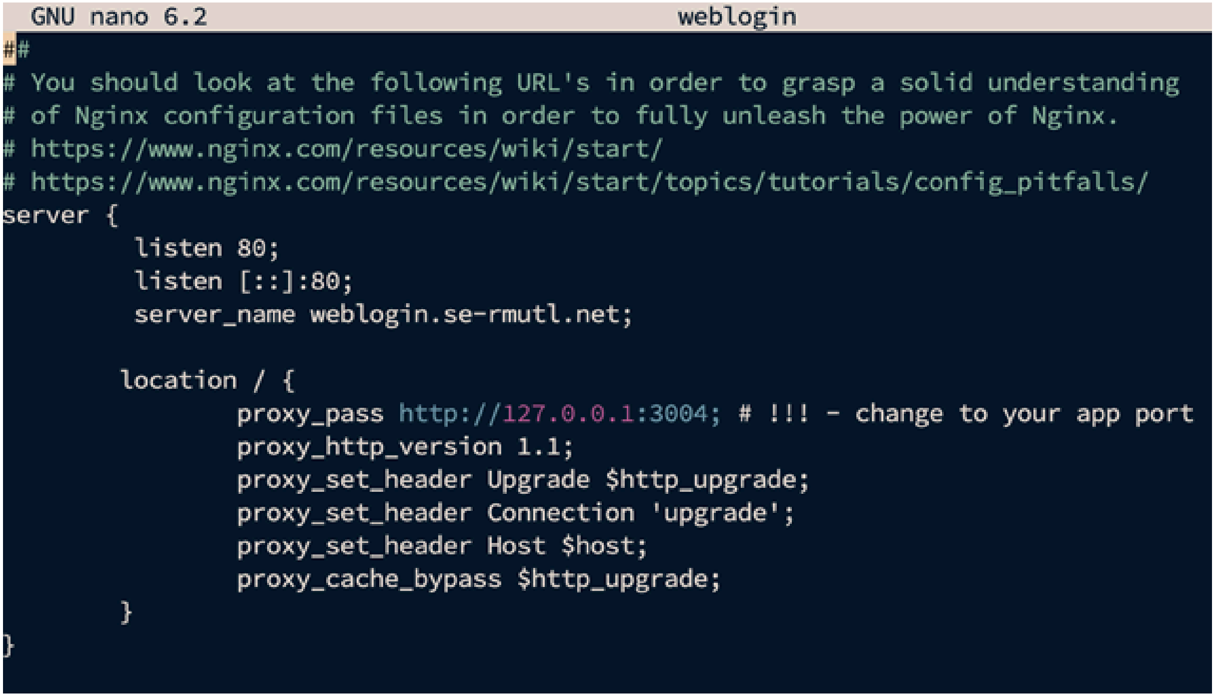
```
import React from 'react';
import ReactDOM from 'react-dom';
import './App.css';
import App from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

```



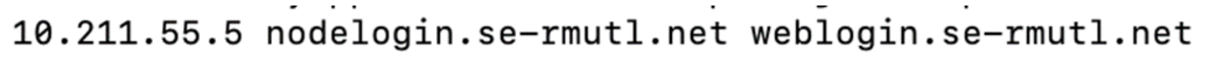
```
"scripts": {
  "start": "PORT=3004 react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
```

```
cd /etc/nginx/sites-availabe
```

```
sudo cp nodelogin weblogin
sudo nano weblogin
```

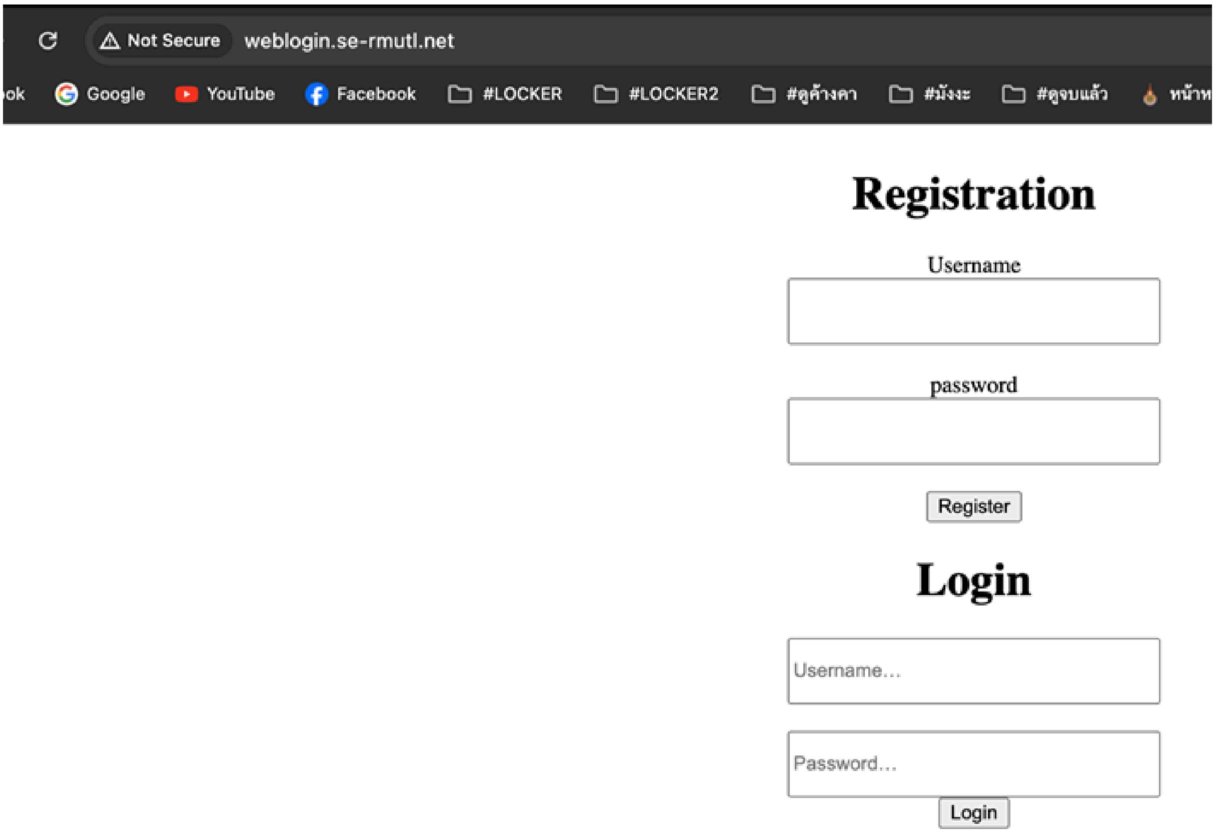


```
  GNU nano 6.2                                weblogin
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
server {
         listen 80;
         listen [::]:80;
         server_name weblogin.se-rmutl.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;
        }
}
```



```
10.211.55.5 nodelogin.se-rmutl.net weblogin.se-rmutl.net
```

sudo ufw allow 3004
npm start



Registration

Username

password

Register

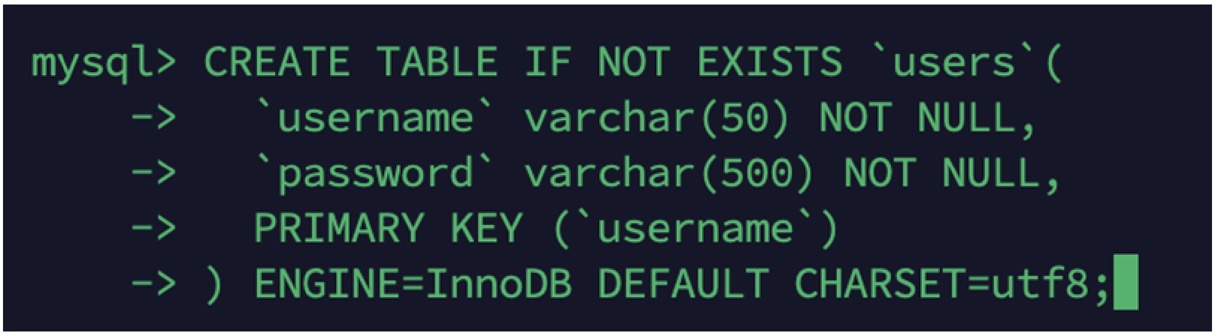
Login

Username...

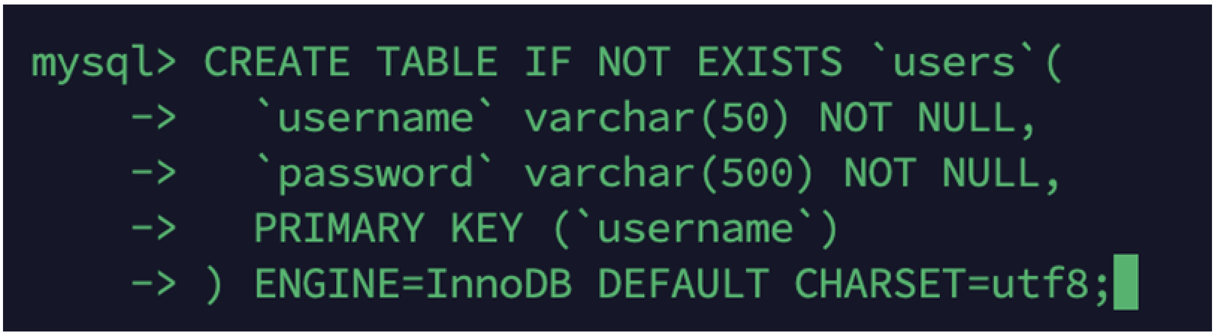
Password...

Login

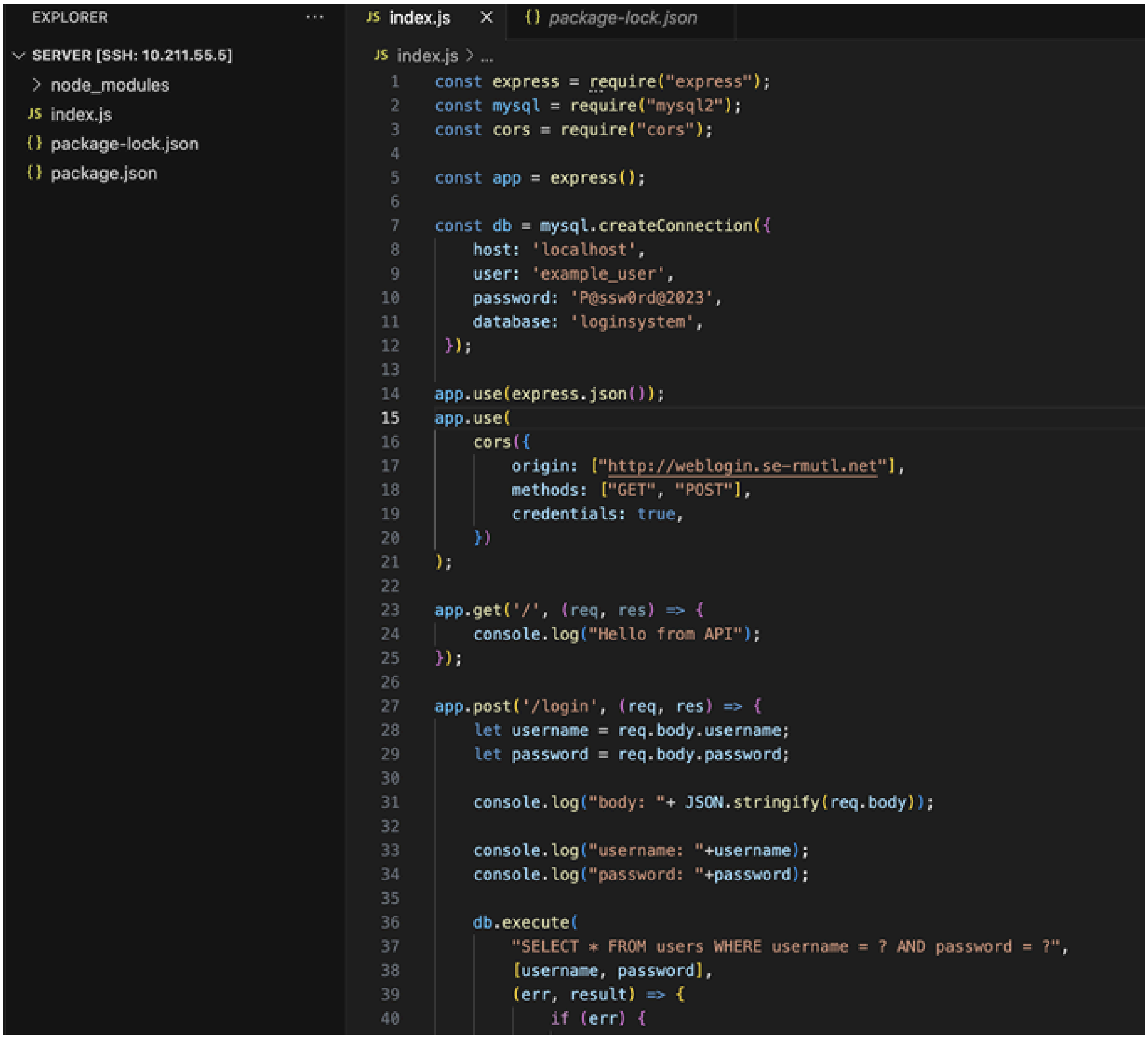
```
mkdir server
cd server
npm init
npm i –sace cors
install express –save
npm install –save mysql2
sudo npm install -g nodemon
```



```
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;
```



```
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;
```



```javascript
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: 'P@ssw0rd@2023',
    database: 'loginsystem',
 });

app.use(express.json());
app.use(
    cors({
        origin: ["http://weblogin.se-rmutl.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],
        (err, result) => {
            if (err) {
```

server2

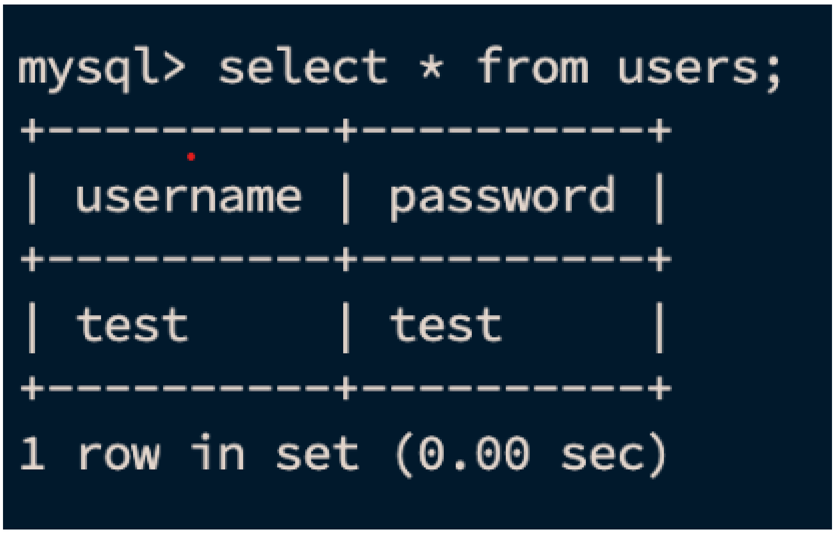
ส่วนนี้จะเป็นของ backend เป็นต้นไป สร้าง directory lab4, เข้าไปใน directory lab4, สร้าง directory server ข้างใน directory lab4 จากนั้น npm init สร้างเพื่อ package

sudo mkdir lab4
cd lab4
sudo mkfir server
cd server
npm init

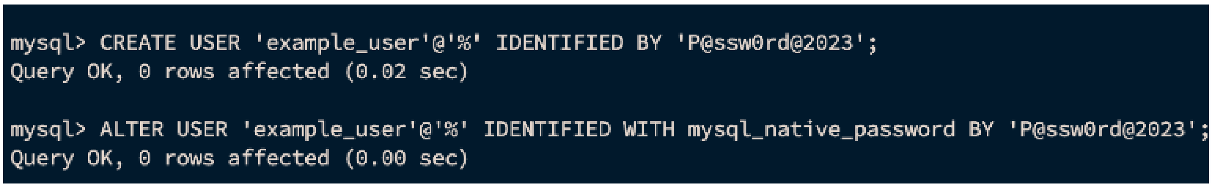
ติดตั้ง express และ mysql2 โดยใช้ npm

sudo npm install express –sace
sudo npm install –save mysql2
sudo npm i –save cors

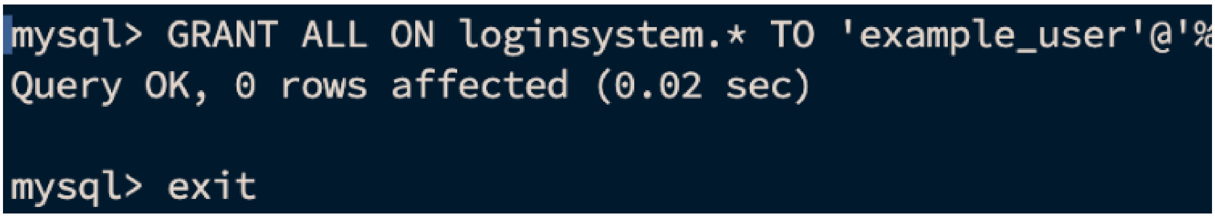
ติดตั้ง nodemon โดยใช้ npm
sudo npm install -g nodemon
sudo mysql



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

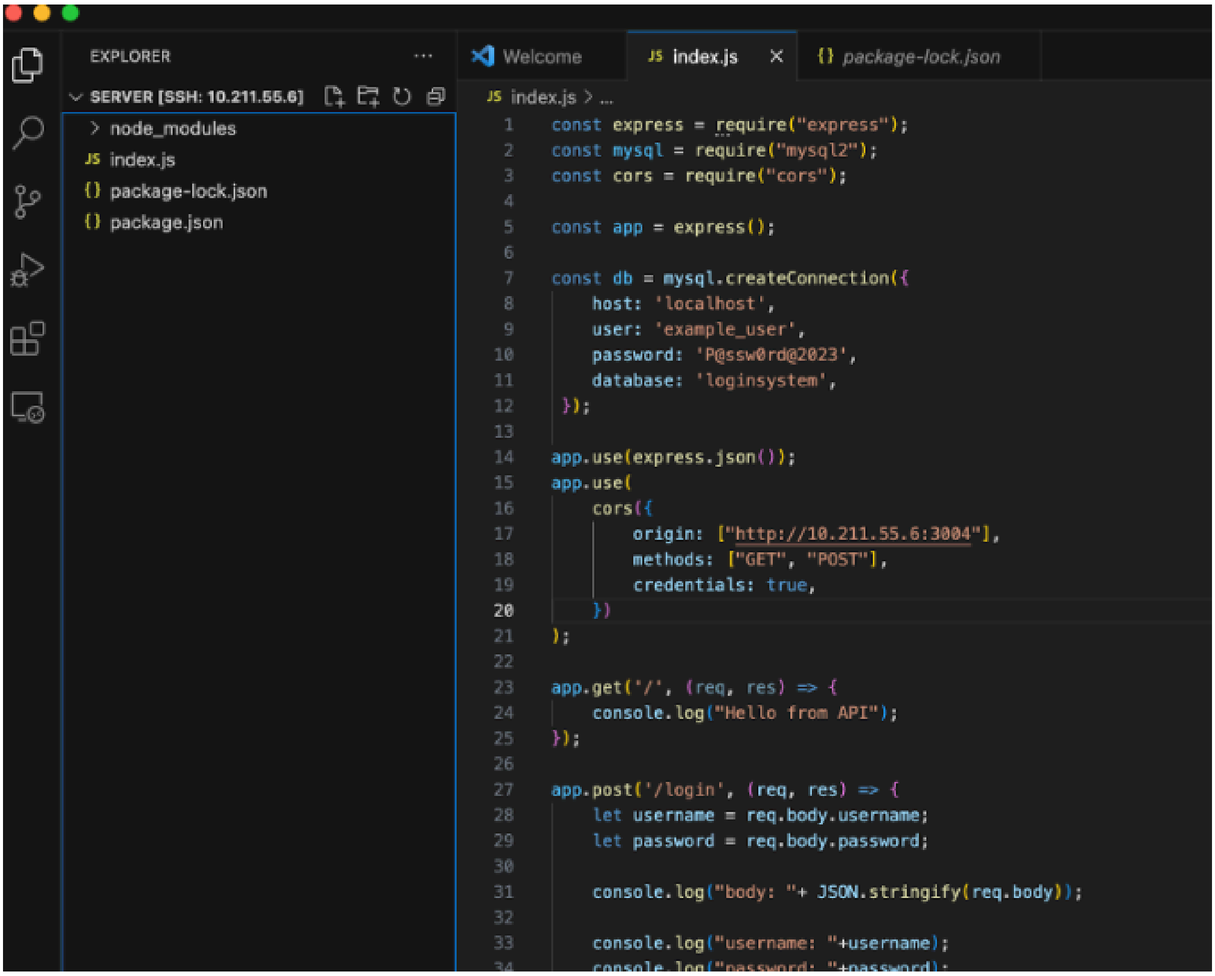


```
mysql> CREATE USER 'example_user'@'%' IDENTIFIED BY 'P@ssw0rd@2023';
Query OK, 0 rows affected (0.02 sec)

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

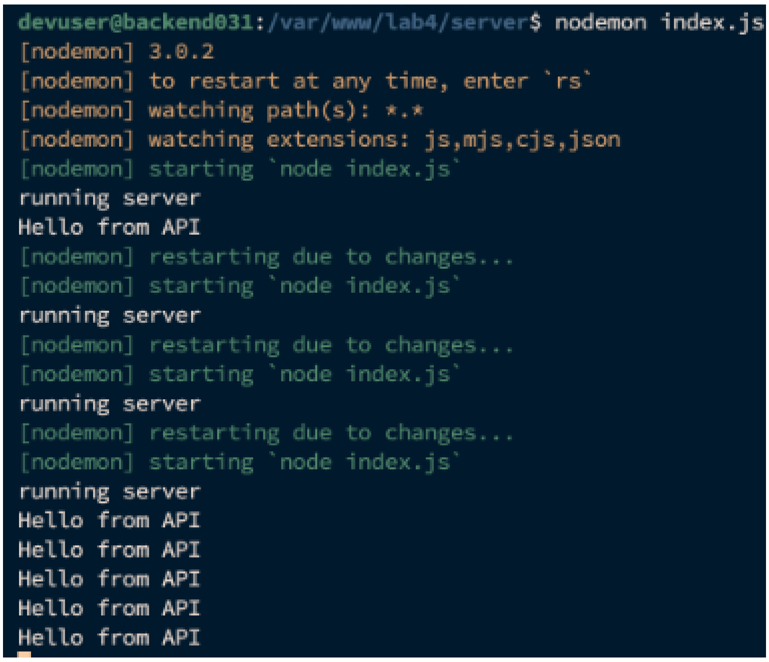


```
mysql> GRANT ALL ON loginsystem.* TO 'example_user'@'%
Query OK, 0 rows affected (0.02 sec)

mysql> exit
```

อนุญาติให้ user ธรรมดา แก้ไขไฟล์ต่างๆใน directory lab4 ได้
sude chown -R $USER:$USER /var/www/lab4



```
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: 'P@ssw0rd@2023',
    database: 'loginsystem',
 });

app.use(express.json());
app.use(
    cors({
        origin: ["http://10.211.55.6:3004"],
        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);
```



```
devuser@backend031:/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
Hello from API
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
running server
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
running server
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
running server
Hello from API
Hello from API
Hello from API
Hello from API
Hello from API
```