

# ENGSE202 LAB4

## Step 1 — Install New Ubuntu Server

### B. server

```
Last login: Wed Dec 20 02:12:58 2023 from 10.211.55.2
devuser@webserver033:~$ hostname
webserver033
devuser@webserver033:~$
```

### A. server

```
Last login: Wed Jan 3 05:22:12 2024 from 10.211.55.2
devuser@backend033:~$ hostname
backend033
devuser@backend033:~$
```

## Step 2 — MySQL Server

```
devuser@backend033:~$ sudo apt list --upgradable
Listing... Done
distro-info-data/jammy-updates 0.52ubuntu0.6 all [upgradable from: 0.52ubuntu0.4]
distro-info/jammy-updates 1.1ubuntu0.2 arm64 [upgradable from: 1.1ubuntu0.1]
python3-distro-info/jammy-updates 1.1ubuntu0.2 all [upgradable from: 1.1ubuntu0.1]
python3-software-properties/jammy-updates 0.99.22.9 all [upgradable from: 0.99.22.7]
python3-update-manager/jammy-updates 1:22.04.17 all [upgradable from: 1:22.04.10]
software-properties-common/jammy-updates 0.99.22.9 all [upgradable from: 0.99.22.7]
update-manager-core/jammy-updates 1:22.04.17 all [upgradable from: 1:22.04.10]
devuser@backend033:~$ sudo apt upgrade
```

```
devuser@backend033:~$ 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 libfcgioldbl
  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-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgioldbl
  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-8.0 mysql-server-core-8.0
0 upgraded, 27 newly installed, 0 to remove and 2 not upgraded.
Need to get 28.9 MB of archives.
After this operation, 238 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

## Step 3 — Install Nginx

```
devuser@backend033:~$ 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 libxml4 nginx-common nginx-core python3-acme python3-certbot python3-configargparse python3-i18n python3-josepy
  python3-parsedatetime python3-requests-toolbelt python3-rfc3339 python3-zope.component python3-zope.event python3-zope.hookable
Suggested packages:
  python-certbot-doc python3-certbot-apache libgd-tools fcgiwrap nginx-doc ssl-cert python-acme-doc python-certbot-nginx-doc
The following NEW packages will be installed:
  certbot 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 libxml4 nginx-common nginx-core python3-acme python3-certbot python3-configargparse python3-parsedatetime python3-requests-toolbelt
  python3-certbot-nginx python3-configargparse python3-i18n python3-josepy python3-parsedatetime python3-requests-toolbelt
  python3-rfc3339 python3-zope.component python3-zope.event python3-zope.hookable
0 upgraded, 33 newly installed, 0 to remove and 2 not upgraded.
Need to get 3,606 kB of archives.
After this operation, 12.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

## Step 4 — Nodejs Installation

หลังจากติดตั้ง Nodejs เวอร์ชันจะเป็น 12.22.9 จึงได้ทำการแก้ไขให้เป็น 20.10.0 ตามภาพ

```
devuser@backend033:~$ node -v
v12.22.9
devuser@backend033:~$ date
Mon Jan  8 07:59:45 UTC 2024
devuser@backend033:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-1ubuntu1.15).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
devuser@backend033:~$ curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash
% Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100 16555  100 16555    0      0  31022       0  --::-- --::-- --::-- 31060
=> Downloading nvm from git to '/home/devuser/.nvm'
=> Cloning into '/home/devuser/.nvm'...
remote: Enumerating objects: 365, done.
remote: Counting objects: 100% (365/365), done.
remote: Compressing objects: 100% (314/314), done.
remote: Total 365 (delta 43), reused 159 (delta 25), pack-reused 0
Receiving objects: 100% (365/365), 364.78 KiB | 1.28 MiB/s, done.
Resolving deltas: 100% (43/43), done.
* (HEAD detached at FETCH_HEAD)
  master
=> Compressing and cleaning up git repository

=> Appending nvm source string to /home/devuser/.bashrc
=> Appending bash_completion source string to /home/devuser/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
devuser@backend033:~$ source ~/.profile
devuser@backend033:~$ nvm install node
Downloading and installing node v21.5.0...
Downloading https://nodejs.org/dist/v21.5.0/node-v21.5.0-linux-arm64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v21.5.0 (npm v10.2.4)
Creating default alias: default -> node (> v21.5.0)
devuser@backend033:~$ nvm install 20
Downloading and installing node v20.10.0...
Downloading https://nodejs.org/dist/v20.10.0/node-v20.10.0-linux-arm64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v20.10.0 (npm v10.2.3)
devuser@backend033:~$ node -v
v20.10.0
```

## Step 5 — Create Login System with Node.js App

```
devuser@webserver033:/var/www/lab4$ cd nodelogin
devuser@webserver033:/var/www/lab4/nodelogin$ npm int
devuser@webserver033:/var/www/lab4/nodelogin$ npm -v
10.2.3
```

สร้าง database nodelogin

```
devuser@webserver033:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
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 `nodelogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0.01 sec)
```

ใส่ข้อมูลลงใน nodelogin

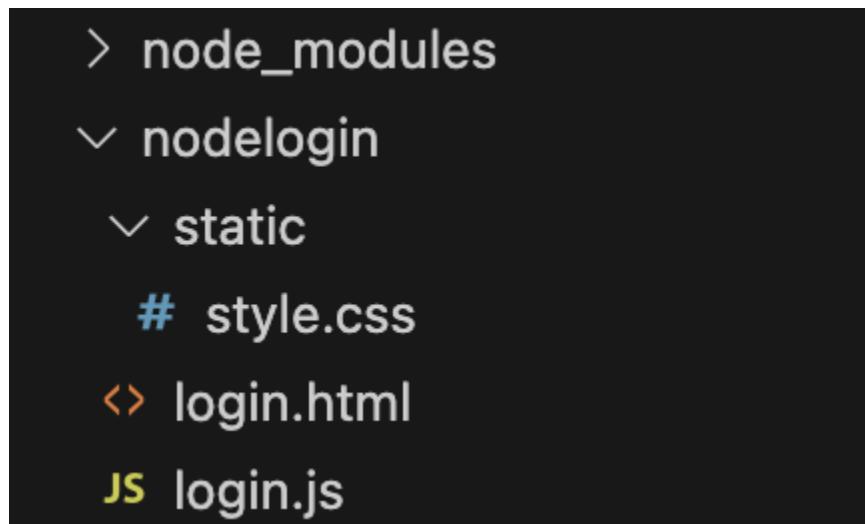
```
mysql> CREATE USER 'example_user'@'%' IDENTIFIED BY 'P@ssw0rd@2023';
Query OK, 0 rows affected (0.03 sec)

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

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

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

เข้าไปสร้างไฟล์ใน /var/www/lab4/nodelogin โดยใช้ remote ssh vscode



```
login.html × JS login.js # style.css {} package.json

nodelogin > login.html > html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width,minimum-scale=1">
6      <title>Login</title>
7      <!-- the form awesome library is used to add icons to our form -->
8      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
9      <!-- include the stylesheet file -->
10     <link href="/style.css" rel="stylesheet" type="text/css">
11  </head>
12  <body>
13      <div class="login">
14          <h1>Login</h1>
15          <form action="/auth" method="post">
16              <label for="username">
17                  <!-- font awesome icon -->
18                  <i class="fas fa-user"></i>
19              </label>
20              <input type="text" name="username" placeholder="Username" id="username" required>
21              <label for="password">
22                  <i class="fas fa-lock"></i>
23              </label>
24              <input type="password" name="password" placeholder="Password" id="password" required>
25              <input type="submit" value="Login">
26          </form>
27      </div>
28  </body>
29 </html>
```

```
login.html  login.js  # style.css  package.json

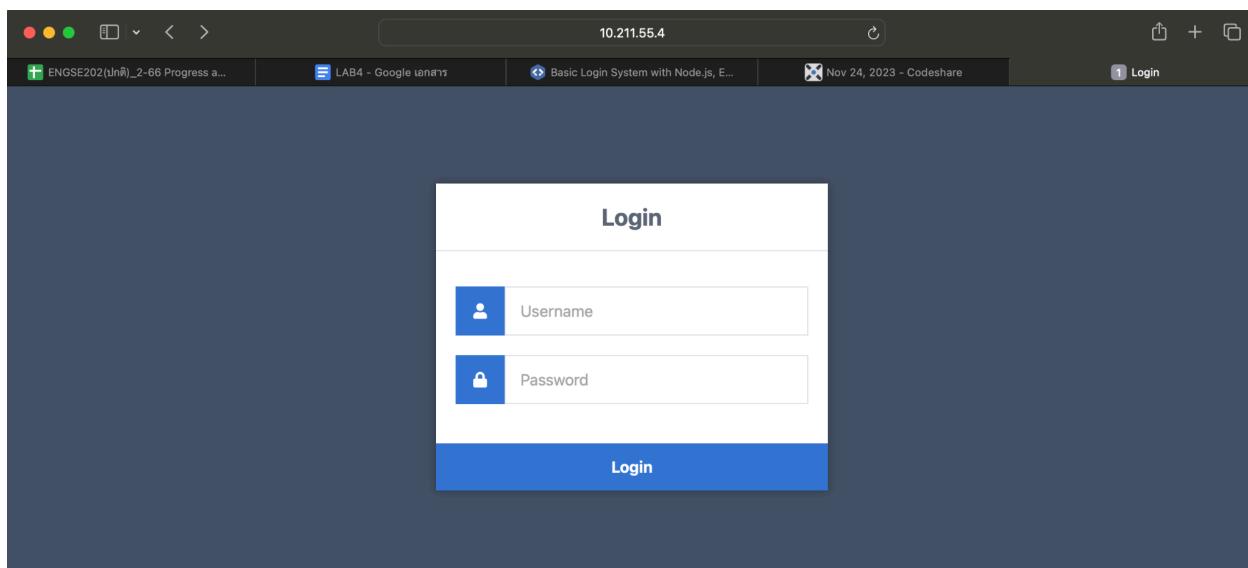
nodeLogin > login.js > saveUninitialized
  1  const mysql = require('mysql');
  2  const express = require('express');
  3  const session = require('express-session');
  4  const path = require('path');
  5
  6  const connection = mysql.createConnection({
  7    host      : 'localhost',
  8    user      : 'example_user',
  9    password  : '@ssw0rd@2023',
 10    database  : 'nodeLogin'
 11  });
 12
 13  const app = express();
 14
 15  app.use(session({
 16    secret: 'secret',
 17    resave: true,
 18    saveUninitialized: true
 19  }));
 20
 21  app.use(express.json());
 22  app.use(express.urlencoded({ extended: true }));
 23  app.use(express.static(path.join(__dirname, 'static')));
 24
 25 // http://localhost:3000/
 26 app.get('/', function(request, response) {
 27   // Render login template
 28   response.sendFile(path.join(__dirname + '/login.html'));
 29 });
 30
 31 // http://localhost:3000/auth
 32 app.post('/auth', function(request, response) {
 33   // Capture the input fields
 34   let username = request.body.username;
 35   let password = request.body.password;
 36   // Ensure the input fields exists and are not empty
 37   if (username && password) {
 38     // Execute SQL query that'll select the account from the database based on the specified username and password
 39     connection.query('SELECT * FROM accounts WHERE username = ? AND password = ?', [username, password], function(error, results, fields) {
 40       // If there is an issue with the query, output the error
 41       if (error) throw error;
 42       // If the account exists
 43       if (results.length > 0) {
 44         // Authenticate the user
 45         request.session.loggedin = true;
 46         request.session.username = username;
 47         // Redirect to home page
 48         response.redirect('/home');
 49       } else {
 50         response.send('Incorrect Username and/or Password!');
 51       }
 52     });
 53   } else {
 54     response.send('Please enter Username and Password!');
 55     response.end();
 56   }
 57 });
 58
 59 // http://localhost:3000/home
 60 app.get('/home', function(request, response) {
 61   // If the user is loggedin
 62   if (request.session.loggedin) {
 63     // Output username
 64     response.send('Welcome back, ' + request.session.username + '!');
 65   } else {
 66     // Not logged in
 67     response.send('Please login to view this page!');
 68   }
 69   response.end();
 70 });
 71
 72 app.listen(3003);
```

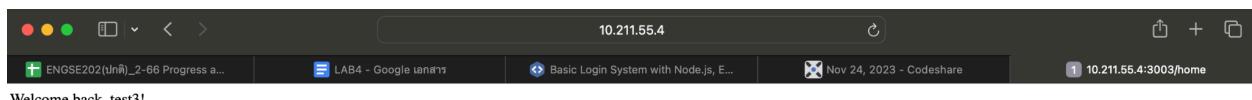
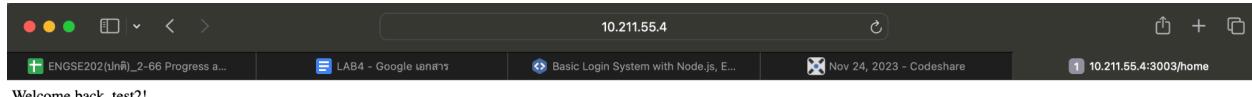
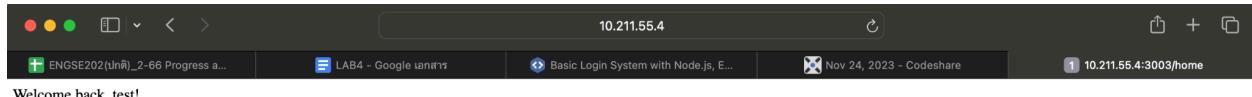
```
login.html | JS login.js | # style.css | {} package.json
node:login > static > # style.css > .login form input[type="submit"]:hover
1  * {
2    box-sizing: border-box;
3    font-family: -apple-system, BlinkMacSystemFont, "segoe ui", roboto, oxygen, ubuntu, cantarell, "fira sans", "droid sans", "helvetica neue",
4    font-size: 16px;
5  }
6  body {
7    background-color: #435165;
8  }
9  .login {
10   width: 400px;
11   background-color: #ffffff;
12   box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);
13   margin: 100px auto;
14 }
15 .login h1 {
16   text-align: center;
17   color: #5b6574;
18   font-size: 24px;
19   padding: 20px 0 20px 0;
20   border-bottom: 1px solid #dee0e4;
21 }
22 .login form {
23   display: flex;
24   flex-wrap: wrap;
25   justify-content: center;
26   padding-top: 20px;
27 }
28 .login form label {
29   display: flex;
30   justify-content: center;
31   align-items: center;
32   width: 50px;
33   height: 50px;
34   background-color: #3274d6;
35   color: #ffffff;
36 }
37 .login form input[type="password"], .login form input[type="text"] {
38   width: 310px;
39   height: 50px;
40   border: 1px solid #dee0e4;
41   margin-bottom: 20px;
42   padding: 0 15px;
43 }
44 .login form input[type="submit"] {
45   width: 100%;
46   padding: 15px;
47   margin-top: 20px;
48   background-color: #3274d6;
49   border: 0;
50   cursor: pointer;
51   font-weight: bold;
52   color: #ffffff;
53   transition: background-color 0.2s;
54 }
55 .login form input[type="submit"]:hover {
56   background-color: #2868c7;
57   transition: background-color 0.2s;
58 }
```

```
login.html      JS login.js      # style.css      {} package.json X
{} package.json > {} dependencies
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   }

```

devuser@webserver033:/var/www/lab4/nodelogin\$ node login.js





เปลี่ยน ip

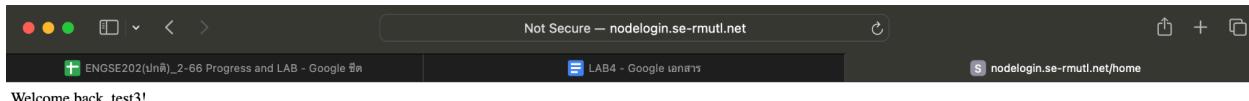
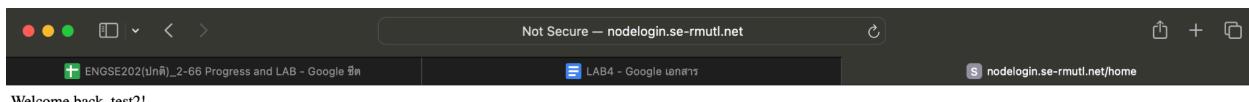
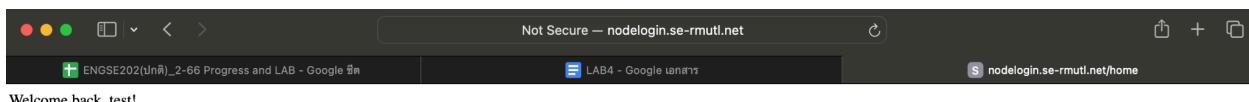
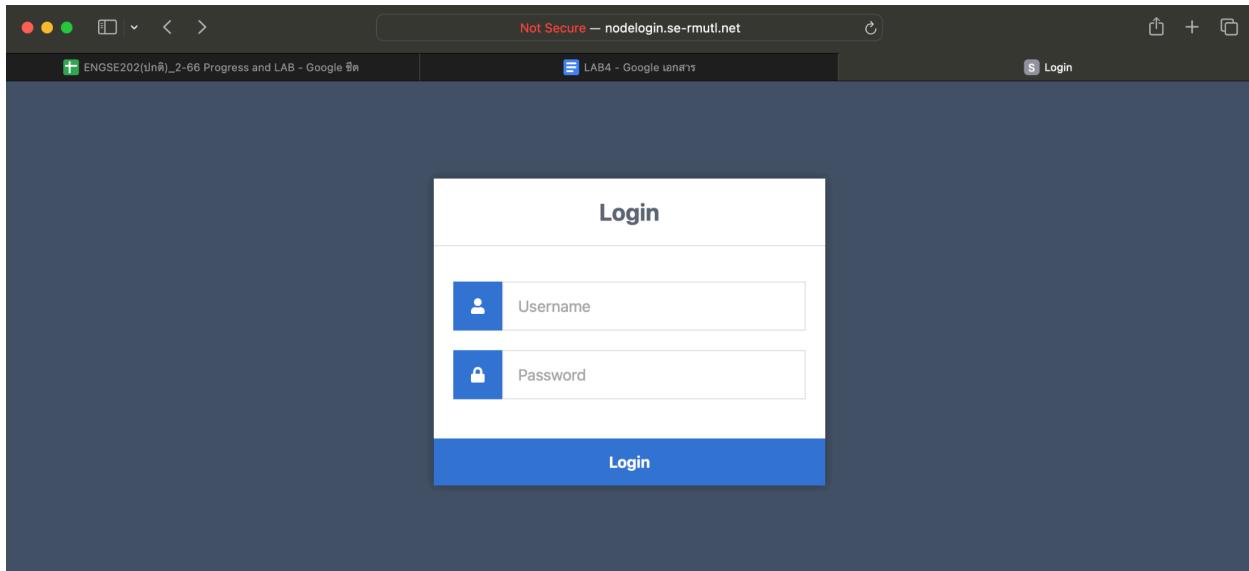
```
devuser@webserver033:~$ cd /etc/nginx/sites-available  
devuser@webserver033:/etc/nginx/sites-available$ sudo cp myapp nodelogin
```

```
devuser@webserver033:/etc/nginx/sites-available$ ls  
default expressjs-example myapp next-app nodelogin react-app recat-app
```

```
GNU nano 6.2                                              nodelogin *  
  
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;  
    }  
}
```

```
● ● ● admin — pico - sudo — 80x24  
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  
10.211.55.4    myapp.se-rmutl.net expressjs-example.se-rmutl.net  
10.211.55.4    my-react-app.se-rmutl.net  
10.211.55.4    ubuntu-next-app.se-rmutl.net  
10.211.55.4    nodelogin.se-rmutl.net  
10.211.55.4    ubuntu-linux.shared ubuntu-linux #prl_hostonly shared  
10.211.55.5    ubuntu-linux-api.shared ubuntu-linux-api #prl_hostonly shared  
  
^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
```

```
devuser@webserver033:~$ sudo systemctl stop apache2  
devuser@webserver033:~$ sudo systemctl restart nginx
```

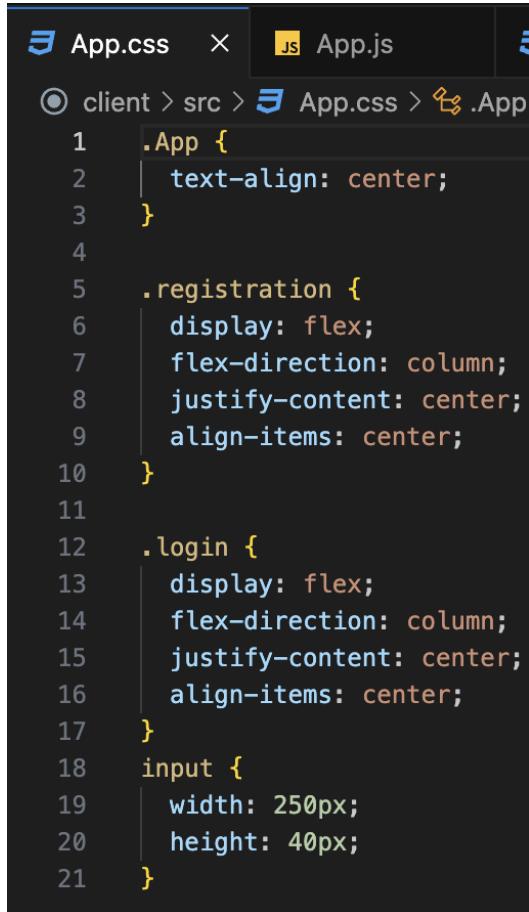


## step 6.1

สร้างไดเรกทอรีและติดตั้ง react

```
devuser@webserver033:/var/www/lab4/client$ ls
node_modules  package.json  package-lock.json  public  README.md  src
```

เข้าสู่ vscode และ เข้าไปแก้ไขไฟล์ app.css , app.js , index.js



```
App.css  X  App.js  ⏪
client > src > App.css > .App
1  .App {
2    text-align: center;
3  }
4
5  .registration {
6    display: flex;
7    flex-direction: column;
8    justify-content: center;
9    align-items: center;
10 }
11
12 .login {
13   display: flex;
14   flex-direction: column;
15   justify-content: center;
16   align-items: center;
17 }
18 input {
19   width: 250px;
20   height: 40px;
21 }
```

```
App.css      App.js      index.js
client > src > App.js > App > login > then() callback
  1 import React, { useEffect, useState } from "react";
  2 import Axios from 'axios';
  3 import './App.css';
  4
  5 function App() {
  6
  7   const [usernameReg, setUsernameReg] = useState("");
  8   const [passwordReg, setPasswordReg] = useState("");
  9
 10  const [username, setUsername] = useState("");
 11  const [password, setPassword] = useState("");
 12
 13  const [loginStatus, setLoginStatus] = useState("");
 14
 15  const register = () => {
 16    Axios.post("http://10.211.55.5:4000/register", {
 17      username: usernameReg,
 18      password: passwordReg,
 19    }).then((response) => {
 20      console.log(response);
 21
 22      if (!response.data.err) {
 23        setLoginStatus(`"${usernameReg}" was inserted.`);
 24      } else {
 25        setLoginStatus(response.data.err.sqlMessage);
 26      }
 27
 28    });
 29  };
 30
 31  const login = () => {
 32    Axios.post("http://10.211.55.5:4000/login", {
 33      username: username,
 34      password: password,
 35    }).then((response) => {
 36      console.log(response);
 37      //console.log("response.data.message: "+response.data[0].message);
 38
 39      if (!response.data.message) {
 40        setLoginStatus(`"${response.data[0].username}" was logged in.`);
 41      } else {
 42        setLoginStatus(response.data.message);
 43      }
 44
 45    });
 46  };
 47
 48  return (
 49    <div className="App">
 50      <div className="registration">
 51        <h1>Registration</h1>
 52        <label>Username</label>
 53        <input
 54          type="text"
 55          onChange={(e) => {
 56            setUsernameReg(e.target.value);
 57          }}
 58        /><br/>
 59        <label>password</label>
 60        <input
 61          type="text"
 62          onChange={(e) => {
 63            setPasswordReg(e.target.value);
 64          }}
 65        /> <br />
 66        <button onClick={register} > Register</button>
 67      </div>

```

```
69      <div className="login">
70        <h1>Login</h1>
71        <input
72          type="text"
73          placeholder="Username..."
74          onChange = { (e) => {
75            setUsername (e.target.value);
76          }
77          /> <br/>
78        <input
79          type="password"
80          placeholder="Password..."
81          onChange = { (e) => {
82            setPassword (e.target.value);
83          }
84          />
85        <button onClick={login}>Login</button>
86      </div>
87      <h1> {loginStatus}</h1>
88    </div>
89  );
90}
91
92 export default App;
```

The screenshot shows a code editor interface with three tabs at the top: "App.css", "App.js", and "index.js". The "index.js" tab is active, indicated by a blue bar and bold text. Below the tabs, there is a breadcrumb navigation bar showing the path: "client > src > index.js". The main area displays the code for the "index.js" file:

```
1 import React from "react";
2 import ReactDOM from "react-dom";
3 import "./App.css";
4 import App from "./App";
5
6 ReactDOM.render(
7   <React.StrictMode>
8     <App />
9   </React.StrictMode>,
10  document.getElementById("root")
11);
```

เปลี่ยน port 3004

```
GNU nano 6.2
[{"name": "client",
"version": "0.1.0",
"private": true,
"dependencies": {
"@testing-library/jest-dom": "^5.17.0",
"@testing-library/react": "^13.4.0",
"@testing-library/user-event": "^13.5.0",
"axios": "^1.6.5",
"react": "^18.2.0",
"react-dom": "^18.2.0",
"react-scripts": "5.0.1",
"web-vitals": "^2.1.4"
},
"scripts": {
"start": "PORT=3004 react-scripts start",
"build": "react-scripts build",
"test": "react-scripts test",
"eject": "react-scripts eject"
},
"eslintConfig": {
"extends": [
"react-app",
"react-app/jest"
]
},
"browserslist": {
"production": [
">0.2%",
"not dead",
"not op_mini all"
],
"development": [
"last 1 chrome version",
"last 1 firefox version",
"last 1 safari version"
]
}
}]
```

```
GNU nano 6.2                                     weblogin *
```

```
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;  
    }  
}
```

```
admin — pico ◁ sudo — 80x24  
File: /etc/hosts  
UW PICO 5.09  
  
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost engse203.lab1  
255.255.255.255 broadcasthost  
::1            localhost  
10.211.55.4    myapp.se-rmutil.net expressjs-example.se-rmutil.net  
10.211.55.4    my-react-app.se-rmutil.net  
10.211.55.4    ubuntu-next-app.se-rmutil.net  
10.211.55.4    nodelogin.se-rmutil.net  
10.211.55.4    weblogin.se-rmutil.net  
10.211.55.5    ubuntu-linux-api.shared ubuntu-linux-api #prl_hostonly shared  
10.211.55.4    ubuntu-linux.shared ubuntu-linux #prl_hostonly shared  
  
^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
```

**Registration**

Username

password

**Register**

**Login**

Username...

Password...

**Login**

## 6.2 Server

ทำการสร้างไดเรกทอรี server ใน lab 4

เข้าไปใน server สร้างไฟล์ package.json ด้วย npm init

```
devuser@webserver033:/var/www/lab4$ mkdir server
devuser@webserver033:/var/www/lab4$ ls
client nodelogin server
```

```
devuser@webserver033:/var/www/lab4$ cd server/
devuser@webserver033:/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.

## ทำการสร้าง index.js

```
js index.js  X
server > index.js > app.post('/register') callback
  1 const mysql = require("mysql2");
  2 const cors = require("cors");
  3
  4
  5 const app = express();
  6
  7
  8
  9 const db = mysql.createConnection({
 10   host: 'localhost',
 11   user: 'example_user',
 12   password: 'P@ssw0rd@2023',
 13   database: 'loginsystem',
 14 });
 15
 16 app.use(express.json());
 17 app.use(
 18   cors({
 19     origin: ["http://weblogin.sermutl.net"],
 20     methods: ["GET", "POST"],
 21     credentials: true,
 22   })
 23 );
 24
 25
 26 app.get('/', (req, res) => {
 27   console.log("Hello from API");
 28 });
 29
 30
 31 app.post('/login', (req, res) => {
 32   let username = req.body.username;
 33   let password = req.body.password;
 34
 35
 36   console.log("body: "+ JSON.stringify(req.body));
 37
 38
 39   console.log("username: "+username);
 40   console.log("password: "+password);
 41
 42
 43   db.execute(
 44     "SELECT * FROM users WHERE username = ? AND password = ?",
 45     [username, password],
 46     (err, result) => {
 47       if (err) {
 48         res.send({ err: err });
 49       }
 50       if (result.length > 0) {
 51         res.send(result);
 52       } else {
 53         res.send({ message: "Wrong username/password combination!" });
 54       }
 55
 56     );
 57   });
 58 }
 59
 60
 61 app.post('/register', (req, res) => {
 62   let username = req.body.username;
 63   let password = req.body.password;
 64
 65
 66   console.log("body: "+ JSON.stringify(req.body));
 67
 68   console.log("username: "+username);
```

```
70     console.log("password: "+password);
71
72
73     db.execute(
74         "INSERT INTO users (username, password) VALUES (?,?)",
75         [username, password],
76         (err, result) => {
77             if (err) {
78                 res.send({ err: err });
79             }
80             res.send(result);
81         }
82     );
83
84 );
85 );
86
87
88 app.listen(4000, () => {
89     console.log("running server");
90 })
```

ลง middleware

```
devuser@webserver033:/var/www/lab4/server$ npm install express --save
added 62 packages, and audited 63 packages in 6s
11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
devuser@webserver033:/var/www/lab4/server$ npm install --save mysql2
added 11 packages, and audited 74 packages in 7s
11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
devuser@webserver033:/var/www/lab4/server$ npm i -save cors
added 2 packages, and audited 76 packages in 693ms
11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

## สร้าง database ชื่อ loginsystem

```
devuser@webserver033:/var/www/lab4/server$ sudo 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 (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> 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> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

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

ติดตั้ง nodedemon allow port:4000 และทดสอบ

```
devuser@webserver033:/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`
[nodemon] clean exit - waiting for changes before restart
```

Compiled successfully!

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

**Local:** http://localhost:**3004**

**On Your Network:** http://**10.211.55.4:3004**

Note that the development build is not optimized.

To create a production build, use **npm run build**.

webpack compiled **successfully**



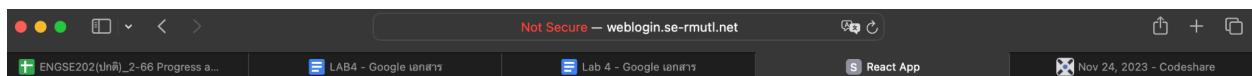
## Registration

Username

password

## Login

**'test' was logged in.**



## Registration

Username

password

## Login

**Wrong username/password combination!**



## Registration

Username

password

## Login

Username...

Password...

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



## Registration

Username

password

## Login

Username...

Password...

**'test5' was inserted.**

## step7

ทำการสร้างไดเรกทอรี server

```
devuser@backend033:/var/www/lab4$ ls  
server
```

เข้าไปใน server สร้างไฟล์ package.json ด้วย npm init

```
devuser@backend033:/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)
```

ลง middleware express mysql2 cors

```
devuser@backend033:/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@backend033:/var/www/lab4/server$ npm install --save mysql2
added 11 packages, and audited 74 packages in 3s
11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
devuser@backend033:/var/www/lab4/server$ npm i --save cors
added 2 packages, and audited 76 packages in 724ms
11 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

ลง nodemon

```
devuser@backend033:/var/www/lab4/server$ npm install -g nodemon
added 33 packages in 12s
3 packages are looking for funding
  run `npm fund` for details
```

## setup database

```
devuser@backend033:/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 11
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 (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 (0.01 sec)

mysql> NSERT INTO `users` (`username`, `password`) VALUES ('test', 'test');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your My
yntax to use near 'NSERT INTO `users` (`username`, `password`) VALUES ('test', 'test')' at line 1
mysql> INSERT INTO `users` (`username`, `password`) VALUES ('test', 'test');
Query OK, 1 row affected (0.01 sec)

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

The screenshot shows a Visual Studio Code interface with the following layout:

- EXPLORER**: Shows the project structure with files: `Welcome`, `index.js`, `node_modules`, `index.js`, `package-lock.json`, and `package.json`.
- WELCOME**: Displays the message "Welcome".
- index.js**: The active tab, showing the following code:

```
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 app.get('/', (req, res) => {
24   console.log("Hello from API");
25 });
26 app.post('/login', (req, res) => {
27   let username = req.body.username;
28   let password = req.body.password;
29   console.log("body: " + JSON.stringify(req.body));
30   console.log("username: " + username);
31   console.log("password: " + password);
32   db.execute(
33     "SELECT * FROM users WHERE username = ? AND password = ?",
34     [username, password],
35     (err, result) => {
36       if (err) {
37         res.send({ err: err });
38       }
39       if (result.length > 0) {
40         res.send(result);
41       } else {
42         res.send({ message: "Wrong username/password combination!" });
43       }
44     }
45   );
46 app.post('/register', (req, res) => {
47   let username = req.body.username;
48   let password = req.body.password;
49   console.log("body: " + JSON.stringify(req.body));
50   console.log("username: " + username);
51   console.log("password: " + password);
52   db.execute(
53     "INSERT INTO users (username, password) VALUES (?,?)",
54     [username, password],
55     (err, result) => {
56       if (err) {
57         res.send({ err: err });
58       }
59       res.send(result);
60     }
61   );
62 });
63 );
64 );
65 app.listen(4000, () => {
66   console.log("running server");
67 });
```



## Registration

Username

password

[Register](#)

## Login

Username...

Password...

[Login](#)

**Wrong username/password combination!**