

Step 1 — Install New Ubuntu Server

Webserver ubuntu

```
devuser@webserver005:~$ hostnamectl
Static hostname: webserver005
Icon name: computer-vm
Chassis: vm
Machine ID: 957435bfff3f044fa9f3a649421013b93
Boot ID: 861632f59c5848ef950b850280873436
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
```

Backend ubuntu

```
devuser@backend005:~$ hostnamectl
Static hostname: backend005
Icon name: computer-vm
Chassis: vm
Machine ID: d52abbb0bc8d478dbcb8b2ad8f92826b
Boot ID: cd713316d84d4cd2b56c0ebc72e6246c
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
```

Step 2 — MySQL Server

ทำการลง mysql server

```
devuser@backend005:~$ 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> █
```

10.211.55.5

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.

ทำการสร้างโฟลเดอร์ **nodelogin** และใช้คำสั่ง **npm init** เพื่อเริ่มทำโปรเจค **nodejs**

```
devuser@webserver005:/var/www/lab4/nodelogin$ 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: (nodelogin)
version: (1.0.0)
description:
entry point: (login.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /var/www/lab4/nodelogin/package.json:

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

Is this OK? (yes) yes
```

```
devuser@webserver005:/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@webserver005:/var/www/lab4/nodelogin$ npm install express-session --save
```

```
added 5 packages, and audited 68 packages in 6s
```

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

```
found 0 vulnerabilities
```

```
devuser@webserver005:/var/www/lab4/nodelogin$ 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
```

```
devuser@webserver005:/var/www/lab4/nodelogin$ ls  
login.html  login.js  package.json  static
```

ไฟล์ทั้งหมดของเว็บ nodelogin

Login.js

```
JS login.js x
nodelogin > JS login.js > ...
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 const app = express();
13 app.use(session({
14   secret: 'secret',
15   resave: true,
16   saveUninitialized: true
17 }));
18 app.use(express.json());
19 app.use(express.urlencoded({ extended: true }));
20 app.use(express.static(path.join(__dirname, 'static')));
21
22 app.get('/', function(request, response) {
23   response.sendFile(path.join(__dirname + '/login.html'));
24 });
25
26 app.post('/auth', function(request, response) {
27   let username = request.body.username;
28   let password = request.body.password;
29   if (username && password) {
30     connection.query('SELECT * FROM accounts WHERE username = ? AND password = ?', [username, password], function(error, results, fields) {
31       if (error) throw error;
32       if (results.length > 0) {
33         request.session.loggedin = true;
34         request.session.username = username;
35         response.redirect('/home');
36       } else {
37         response.send('Incorrect Username and/or Password!');
38       }
39       response.end();
40     });
41   } else {
42     response.send('Please enter Username and Password!');
43     response.end();
44   }
45 });
46
47 app.get('/home', function(request, response) {
48   if (request.session.loggedin) {
49     response.send('Welcome back, ' + request.session.username + '!');
50   } else {
51     response.send('Please login to view this page!');
52   }
53   response.end();
54 });
55
56 app.listen(3003);
```

Login.html

```
JS login.js login.html x
node login > login.html > html > head > link
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>
```

Style.css

```
JS login.js style.css x
nodelogin > static > style.css > *
1 {
2   box-sizing: border-box;
3   font-family: -apple-system, BlinkMacSystemFont, "segoe ui", roboto, oxygen, ubuntu, cantarell, "fira sans", "droid sans", "helvetica neue", Arial, sans-serif;
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;
```

สร้างฐานข้อมูลเพื่อจัดเก็บข้อมูลผู้ใช้

Mysql

สร้างฐานข้อมูล nodelogin

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

ให้สิทธิ์ฐานข้อมูลกับ example_user

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

สร้างตารางเพื่อเก็บข้อมูลชื่อ accounts เก็บ id username password email

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

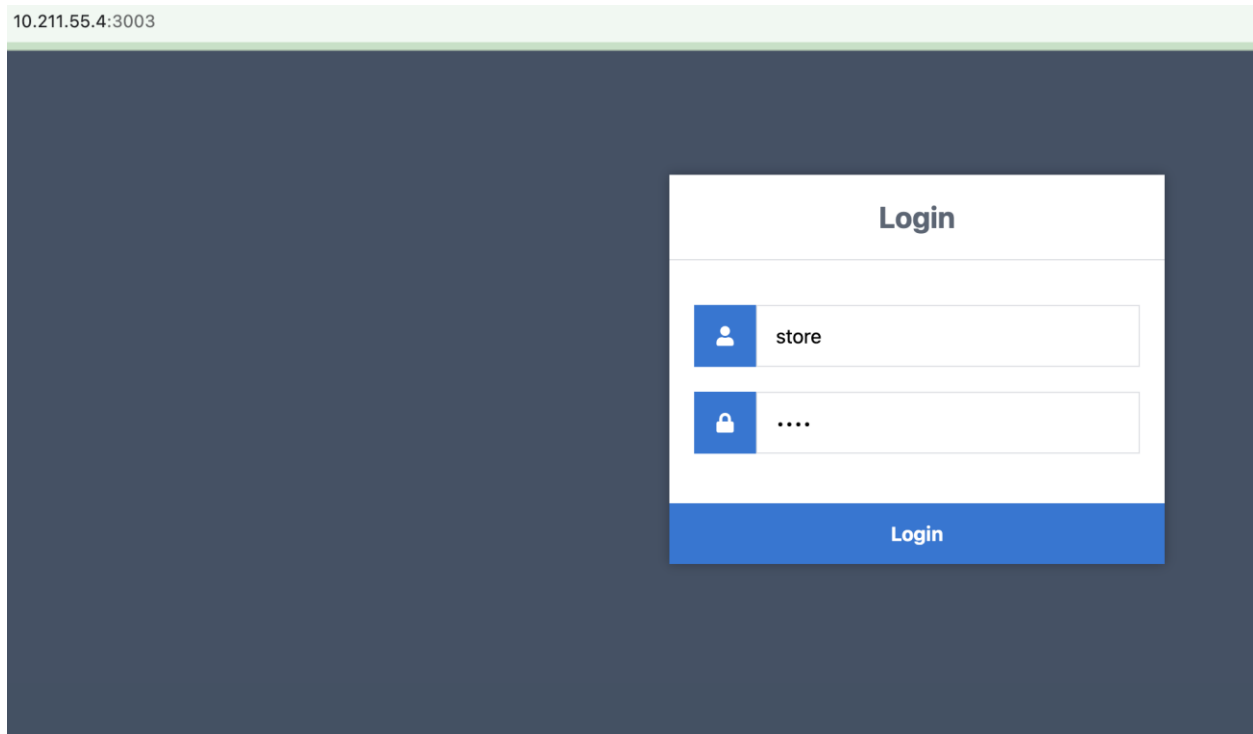
ทดสอบเพิ่มข้อมูลผู้ใช้

```
mysql> INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (1, 'store', '1234', 'jakkrapart_in66@live.rmutl.ac.th');  
Query OK, 1 row affected (0.01 sec)  
  
mysql> INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (2, 'test2', 'test2', 'test2@test.com');  
Query OK, 1 row affected (0.00 sec)  
  
mysql> INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (3, 'test3', 'test3', 'test3@test.com');  
Query OK, 1 row affected (0.00 sec)
```

รันโปรแกรม

```
devuser@webserver005:/var/www/lab4$ cd nodelogin/  
devuser@webserver005:/var/www/lab4/nodelogin$ node login.js
```


ทดลองเข้าเว็บ



ไฟล์ config nginx ให้ผู้ใช้เข้าได้ผ่าน domain name

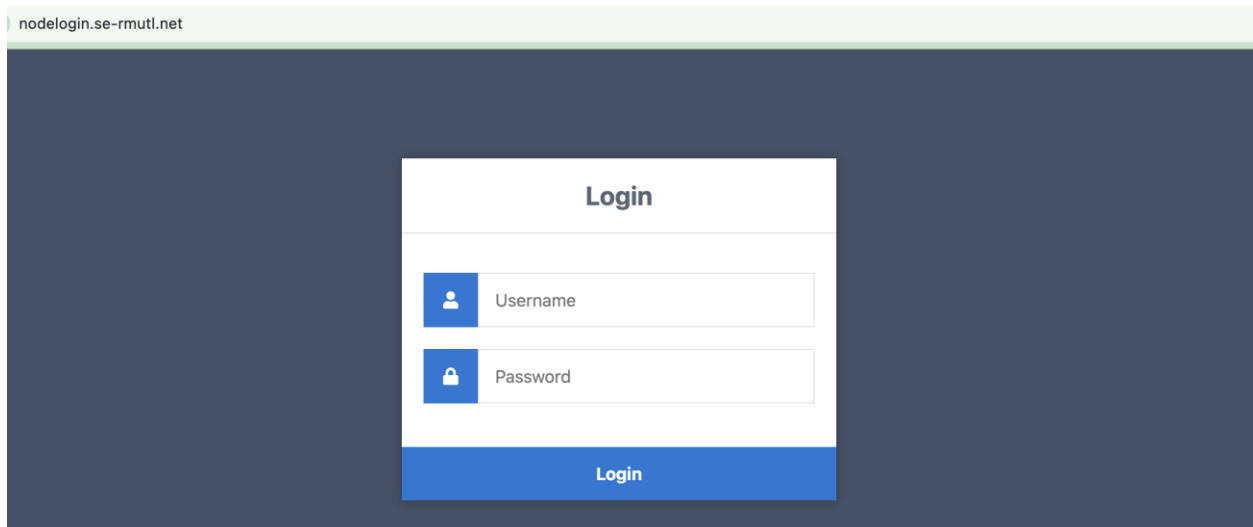
```
devuser@webserver005:~$ more /etc/nginx/sites-available/nodelogin
server {
    listen 80;
    listen [::]:80;
    server_name nodelogin.se-rmutl.net;

    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;
    }
}
```

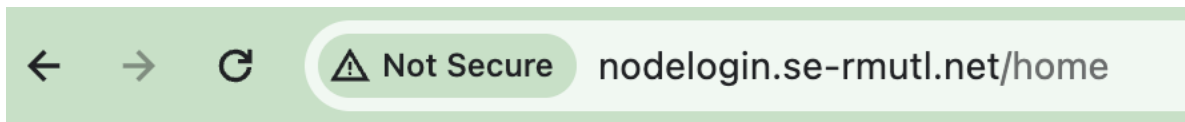
จักรภัทร อินสุก 66543210005-5

```
devuser@webserver005:~$ sudo nginx -t
[sudo] password for devuser:
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
devuser@webserver005:~$ sudo systemctl restart nginx.service
```

ทดสอบเว็บผ่าน domain name



ทดลอง login จากผู้ใช้ที่เพิ่มเข้าไป



Welcome back, store!

STEP 6

สร้างโปรเจกต์ react weblogin

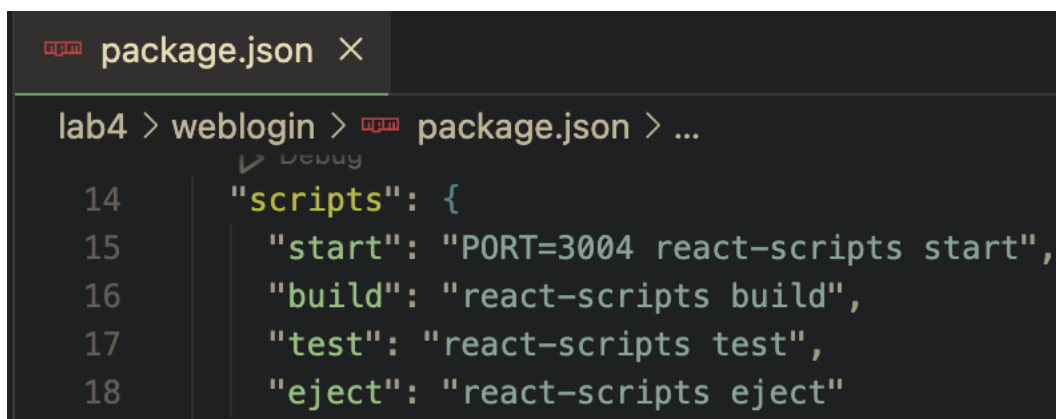
```
devuser@webserver005:/var/www/lab4$ npx create-react-app weblogin
```

Creating a new React app in `/var/www/lab4/weblogin`.

Installing packages. This might take a couple of minutes.

Installing `react`, `react-dom`, and `react-scripts` with `cra-template`...

กำหนด port ให้เป็น 3004



```
lab4 > weblogin > package.json > ...  
14   "scripts": {  
15     "start": "PORT=3004 react-scripts start",  
16     "build": "react-scripts build",  
17     "test": "react-scripts test",  
18     "eject": "react-scripts eject"
```

เพิ่ม firewall ให้กับ server

```
devuser@webserver005:~$ sudo ufw allow 3004  
[sudo] password for devuser:  
Rule added  
Rule added (v6)
```

ลงโมดูล react เพิ่มเติม

```
devuser@webserver005:/var/www/lab4/client$ npm i --save axios
added 3 packages, and audited 1558 packages in 2s

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

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

ไฟล์ทั้งหมดของโปรเจ็ค weblogin

App.js

```
JS index.js  JS App.js  ×
lab4 > client > src > JS App.js > App
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
16     const register = () => {
17         Axios.post("http://10.211.55.4:4000/register", {
18             username: usernameReg,
19             password: passwordReg,
20         }).then((response) => {
21             console.log(response);
22             if (response) {
23                 console.log("regis suscess");
24             } else {
25                 console.log("fail");
26             }
27         });
28     };
29
30     const login = () => {
31         Axios.post("http://10.211.55.4:4000/login", {
32             username: username,
33             password: password,
34         }).then((response) => {
35             if (!response.data.message) {
36                 setLoginStatus("'+response.data[0].username+' was logged in.");
37                 // console.log("1");
38             } else {
39                 setLoginStatus(response.data.message);
40                 // console.log("2");
41             }
42         });
43     };
44 }
```

```

return (
  <div className="App">
    <div className="registration">
      <h1>Registration</h1>
      <label>Username</label>
      <input
        type="text"
        onChange={(e) => {
          setUsernameReg(e.target.value);
        }}
      /><br/>
      <label>password</label>
      <input
        type="text"
        onChange={(e) =>{
          setPasswordReg(e.target.value);
        }}
      /> <br />
      <button onClick={register} > Register</button>
    </div>

    <div className="login">
      <h1>Login</h1>
      <input
        type="text"
        placeholder="Username..."
        onChange = { (e) => {
          setUsername (e.target.value);
        }}
      /> <br/>
      <input
        type="password"
        placeholder="Password..."
        onChange = { (e) => {
          setPassword (e.target.value);
        }}
      />
      <button onClick={login}>Login</button>
    </div>
    <h1> {loginStatus}</h1>
  </div>
);
}

export default App;

```

Index.css

JS index.js

JS App.js

App.css

×

lab4 > client > src > App.css > .App

```
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  }
```

Index.js

```
JS index.js × JS App.js
lab4 > client > src > JS index.js
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  );
```


สร้าง api server จำลองในเครื่อง

สร้างเว็บสำหรับเป็น api

```
devuser@webserver005:/var/www/lab4$ mkdir server
devuser@webserver005:/var/www/lab4$ cd server/
devuser@webserver005:/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) **yes**

ลงส่วนเสริมสำหรับทำ api

```
devuser@webserver005:/var/www/lab4/server$ npm install express --save
added 62 packages, and audited 63 packages in 1s

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

found 0 vulnerabilities
devuser@webserver005:/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@webserver005:/var/www/lab4/server$ npm i --save cors
added 2 packages, and audited 76 packages in 684ms

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

found 0 vulnerabilities
```

```
devuser@webserver005:/var/www/lab4/server$ sudo npm install -g nodemon
[sudo] password for devuser:

changed 33 packages in 1s

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

```
devuser@webserver005:/var/www/lab4/server$ sudo ufw allow 4000
Rule added
Rule added (v6)
```

ไฟล์ api server
Index.js in server

```
lab4 > server > JS index.js > ...
 1  const express = require("express");
 2  const mysql = require("mysql2");
 3  const cors = require("cors");
 4
 5  const app = express();
 6  const db = mysql.createConnection({
 7      user: "root",
 8      host: "localhost",
 9      password: "password",
10      database: "loginsystem",
11  });
12
13  app.use(express.json());
14  app.use(
15      cors({
16          origin: "http://weblogin.se-rmutl.net", // Update with your frontend URL
17          methods: ["GET", "POST"],
18          credentials: true,
19      })
20  );
21
22
23  app.post('/register', (req, res)=> {
24      const username = req.body.username;
25      const password = req.body.password;
26      db.execute(
27          "INSERT INTO users (username, password) VALUES (?,?)",
28          [username, password],
29          (err, result)=> {
30              if (err) {
31                  res.send({err: err});
32              }
33              res.send(result);
34          }
35      );
36  });
37
```

```
app.post('/login', (req, res) => {
  const username = req.body.username;
  const password = req.body.password;
  db.execute(
    "SELECT * FROM users WHERE username = ? AND password = ?",
    [username, password],
    (err, result) => {
      if (err) {
        res.send({err: err});
      }

      if (result.length > 0) {
        res.send(result);
      } else {
        res.send({ message: "Wrong username/password combination!" });
      }
    }
  );
});

app.listen(4000, () => {
  console.log("running server");
});
```

ทำการสร้างฐานข้อมูลไว้เก็บข้อมูลผู้ใช้

```
devuser@webserver005:/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 63

Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

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

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

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

```
mysql> CREATE DATABASE IF NOT EXISTS `loginsystem` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected, 2 warnings (0.03 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 (0.04 sec)
```

```
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)

ทดสอบรันweblogin เพื่อเชื่อมต่อกับ api

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**



รัน api server

```
devuser@webserver005:/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
```

ทดสอบเว็บ



Registration

Username

password

Register

Login

Username...

Password...

Login

ลองใส่รหัสผิด

Login

test
....
Login

Wrong username/password combination!

ทดลองเพิ่มข้อมูลผู้ใช้

Registration

Username
test
password
test
Register

Login

test
....
Login

'test' was inserted.

ทดลองเข้าสู่ระบบ

Login

'test' was logged in.

Step 7

ทดลองเอา api มาตั้งอยู่ server site เครื่องเพื่อให้สามารถสื่อสารกับ webogin ได้

```
devuser@backend005:~$ sudo mkdir /var/www/lab4/server
devuser@backend005:~$ sudo chown -R $USER:$USER /var/www/lab4
devuser@backend005:~$ cd /var/www/lab4/server
devuser@backend005: /var/www/lab4/server$
```

สร้าง โปรเจกเหมือน step ที่แล้วทุกอย่าง

```
devuser@backend005: /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) yes
```

```
devuser@backend005:/var/www/lab4/server$ npm install express --save
```

```
added 62 packages, and audited 63 packages in 1s
```

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

```
found 0 vulnerabilities
```

```
devuser@backend005:/var/www/lab4/server$ npm i --save cors
```

```
added 2 packages, and audited 76 packages in 745ms
```

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

```
found 0 vulnerabilities
```

```
mysql> CREATE USER 'example_user'@ '%' IDENTIFIED BY 'P@ssword@2023';  
Query OK, 0 rows affected (0.05 sec)
```

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

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

```
devuser@backend005:/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
```

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

const app = express();
const db = mysql.createConnection({
  user: "example_user",
  host: "localhost",
  password: "P@ssw0rd@2023",
  database: "loginsystem",
});
```

ทำการเปลี่ยนของส่วน origin ใน index.js ของ server site

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

และเปลี่ยน axios ใน webserver ให้ทำงานผ่าน ip ฟัง server ทั้ง login และ register

```
const register = () => {
  Axios.post("http://10.211.55.5:4000/register", {
    username: usernameReg,
    password: passwordReg,
  }).then((response) => {
    console.log(response);
    if (!response.data.err) {
      setLoginStatus('"' + usernameReg + '" was inserted.');
    } else {
      setLoginStatus(response.data.err.sqlMessage);
    }
  });
};

const login = () => {
  Axios.post("http://10.211.55.5:4000/login", {
    username: username,
    password: password,
  }).then((response) => {
    if (!response.data.message) {
      setLoginStatus('"' + response.data[0].username + '" was logged in.');
      // console.log("1");
    } else {
      setLoginStatus(response.data.message);
      // console.log("2");
    }
  });
};
```

```
app.post('/register', (req, res) => {
  const username = req.body.username;
  const password = req.body.password;
  db.execute(
    "INSERT INTO users (username, password) VALUES (?,?)",
    [username, password],
    (err, result) => {
      if (err) {
        res.send({err: err});
      }
      res.send(result);
    }
  );
});
```

```
app.post('/login', (req, res) => {
  const username = req.body.username;
  const password = req.body.password;
  db.execute(
    "SELECT * FROM users WHERE username = ? AND password = ?",
    [username, password],
    (err, result) => {
      if (err) {
        res.send({err: err});
      }

      if (result.length > 0) {
        res.send(result);
      } else {
        res.send({ message: "Wrong username/password combination!" });
      }
    }
  );
});

app.listen(4000, () => {
  console.log("running server");
});
```

```
devuser@backend005:/var/www/lab4/server$ node index.js  
running server  
█
```

ทดสอบweblogin ผ่าน server site api

เพิ่มข้อมูลผู้ใช้

Registration

Username

jakkrapart

password

1234

Register

Login

few4

....

Login

'jakkrapart' was inserted.

ทดสอบเข้าสู่ระบบ

Login

Login

'jakkrapart' was logged in.