

Internet Programming

TP8: VueJS, NodeJS

(Authentication frontend and APIs)

Name: Aing Hongsin

Id: e20190004

Group: I4 GIC A

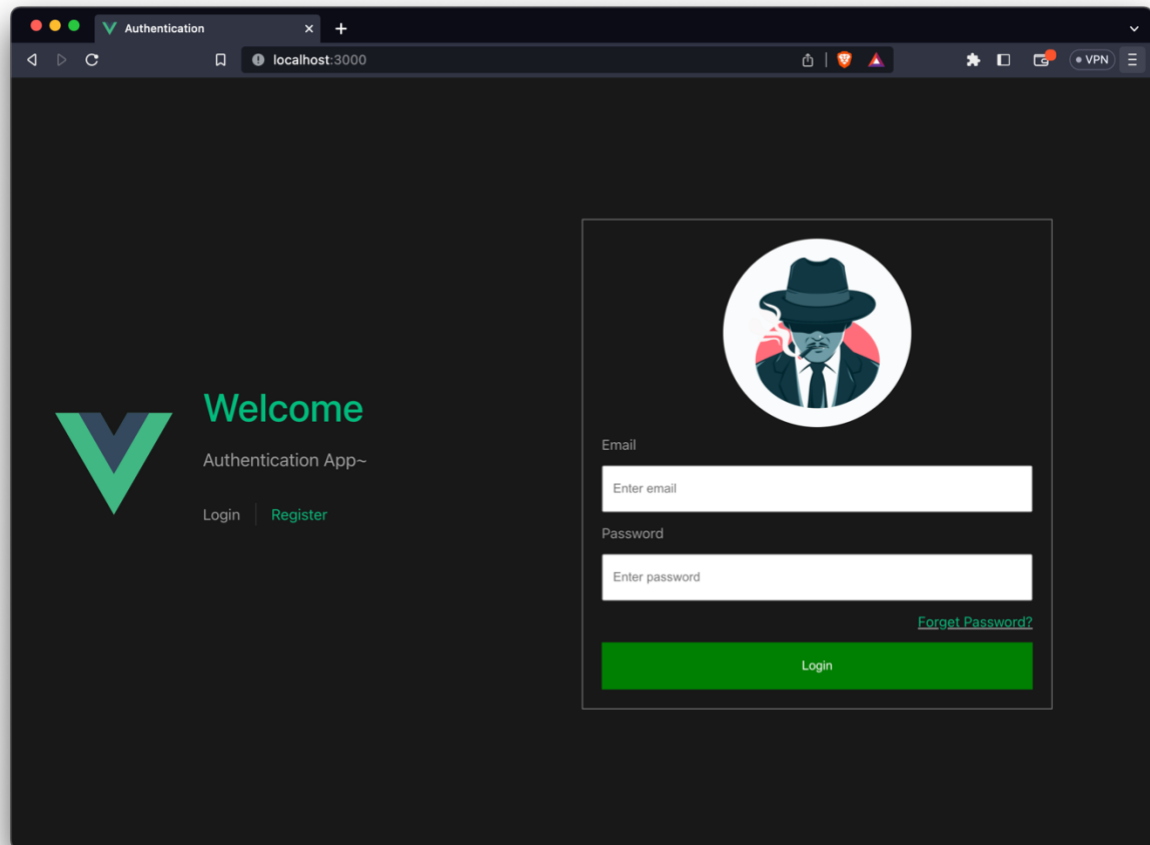
Github: [AingHongSin](#)

Table of Contents

Exercise1: Auth screens in VueJS.....	2
Login	2
Register	3
Exercise2: NodeJS with ExpressJS	3
Register	3
Login	4
Exercise 3: UI and APIs Integration	5

Exercise1: Auth screens in VueJS

Login



Register

The screenshot shows a web browser window with the address bar displaying `localhost:3000/Register`. The page has a dark theme. On the left, there is a green logo consisting of two overlapping 'V' shapes, followed by the text 'Welcome' in green and 'Authentication App~' in white. Below this, there are two links: 'Login' and 'Register', with 'Register' being the active link. On the right, there is a 'Sign Up' form with the instruction 'Please fill in this form to create an account.' The form contains five input fields: 'Email' (placeholder 'Enter email'), 'Username' (placeholder 'Username'), 'First name' (placeholder 'First name'), 'Last name' (placeholder 'Last name'), and 'Password' (placeholder 'Create your password'). A green 'Login' button is at the bottom of the form.

Exercise2: NodeJS with ExpressJS

Register

The screenshot shows a REST client interface. The top bar indicates a POST request to `http://localhost:3001/register`. The 'Body' tab is selected, showing a JSON payload:

```
1 {
2   "email": "abc@gmail.com",
3   "username": "abc",
4   "firstname": "ab",
5   "lastname": "ab",
6   "password": "123456"
7 }
```

The bottom section shows the response, with a status of 200 OK, time of 20 ms, and size of 327 B. The response body is:

```
1 {
2   "success": true,
3   "data": {}
4 }
```

Login

The screenshot displays a REST client interface with a dark theme. At the top, there are tabs for 'Overview', 'New Collection', and 'POST New Request'. The 'POST New Request' tab is active, showing a request to 'http://localhost:3001/login'. The request method is 'POST'. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is selected, and the request body is shown in a text editor with the following JSON:

```
1 {
2   "email": "lmn@gmail.com",
3   "password": "123456"
4 }
```

Below the request body, there are radio buttons for 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (selected), 'binary', and 'GraphQL'. The 'JSON' button is also visible. A 'Send' button is located to the right of the request body.

Below the request editor, there are tabs for 'Body', 'Cookies', 'Headers (9)', and 'Test Results'. The 'Body' tab is selected, and the response body is shown in a text editor with the following JSON:

```
1 {
2   "success": true,
3   "data": [
4     {
5       "email": "abc@gmail.com",
6       "username": "abc",
7       "firstname": "ab",
8       "lastname": "ab",
9       "password": "123456"
10    },
11    {
12      "email": "bcd@gmail.com",
13      "username": "bcd",
14      "firstname": "bc",
15      "lastname": "bc",
16      "password": "123456"
17    },
18    {
19      "email": "efg@gmail.com",
20      "username": "efg",
21      "firstname": "ef",
22      "lastname": "ef",
23      "password": "123456"
24    },
25    {
26      "email": "fgh@gmail.com",
27      "password": "123456"
28    }
29  ]
30 }
```

At the bottom of the response editor, there are tabs for 'Pretty', 'Raw', 'Preview', and 'Visualize'. The 'Pretty' tab is selected. The status bar at the bottom shows 'Status: 200 OK', 'Time: 13 ms', 'Size: 880 B', and a 'Save Response' button.

Exercise 3: UI and APIs Integration

