

# OMOBIO

## LOGIN AND REGISTRATION SYSTEM

Technologies: PHP, React JS, MySQL, REST APIs

Email : ranasinghemalith20@gmail.com

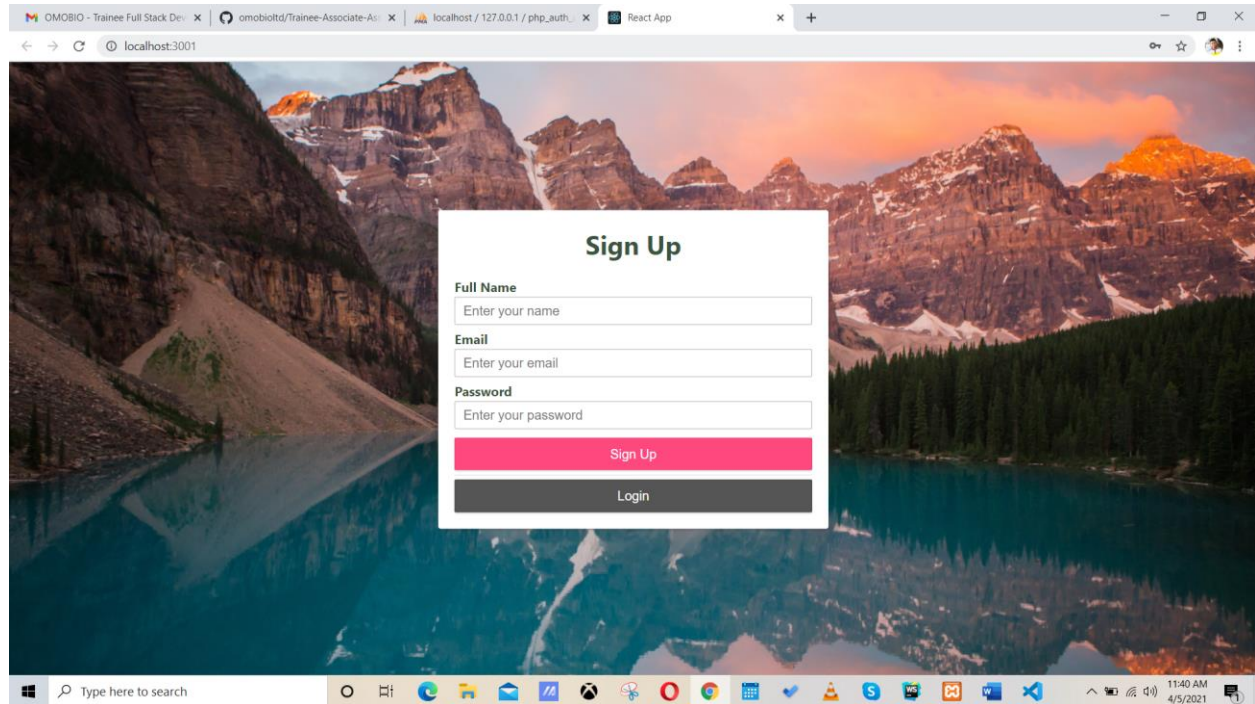
Created By Malith Ranasinghe

Date : 2021-04-05

## Contents

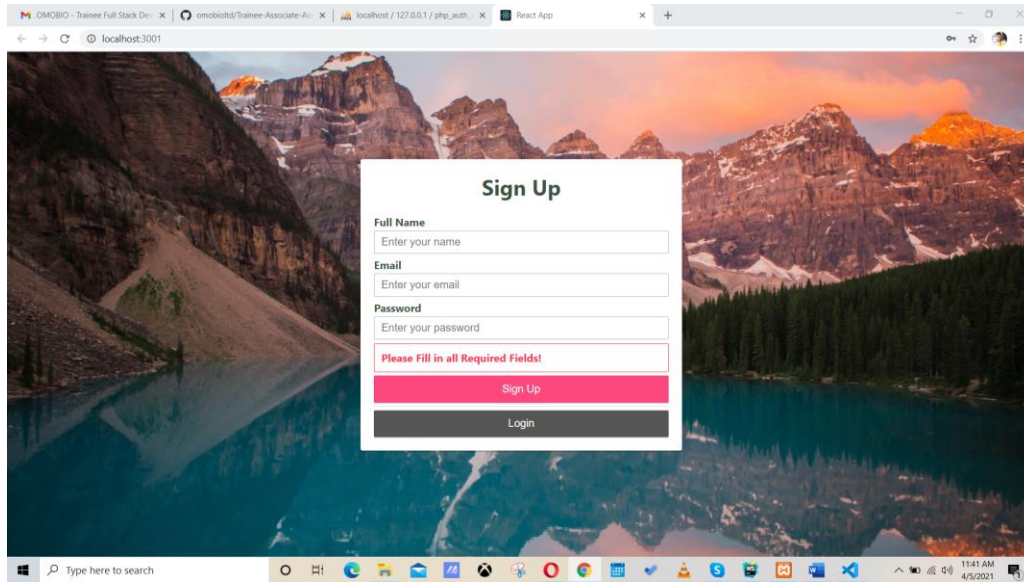
Technologies: PHP, React JS, MySQL, REST APIs .....	1
Registration Page.....	3
Validation check .....	4
1. Empty Fields Check .....	4
1. Check Same Email Registration.....	4
2. Password Characters Check .....	5
3. Invalid Email Check .....	5
4. Successfully Registered Message.....	6
Home Page .....	6
Logout User .....	7
Login Page.....	7
Check Validation .....	8
1. Empty fields check. ....	8
2. Invalid email address check .....	8
3. Invalid password check .....	9
Successfully Login .....	9
Testing the APIs .....	10
API URLs – .....	10
1. Register .....	10
2. Login.....	10
3. Get User Info by giving token.....	10
Register a user API.....	10
Login the user API.....	11
Get the user information by giving the token .....	12
Password encrypted. ....	12

## Registration Page

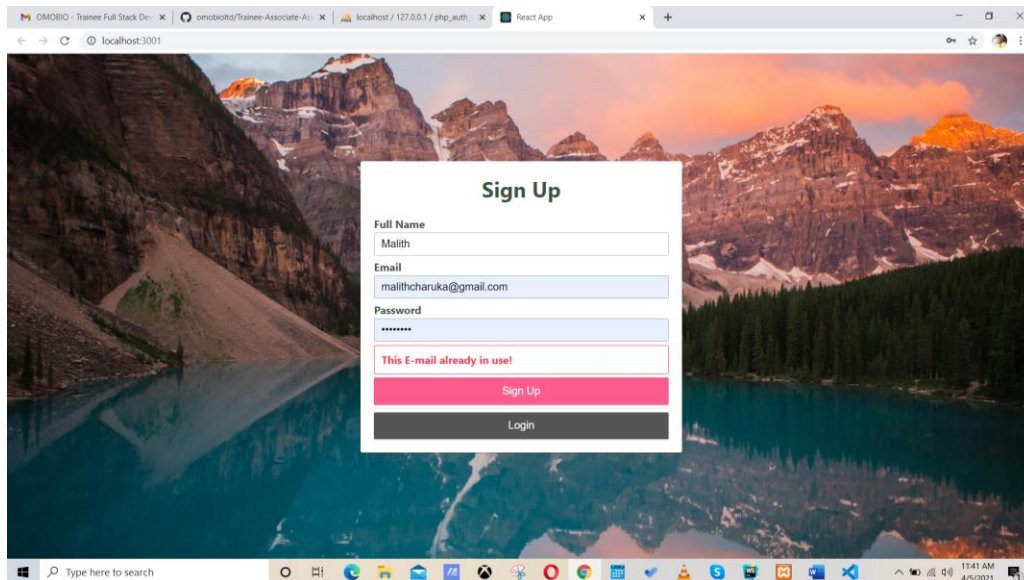


## Validation check

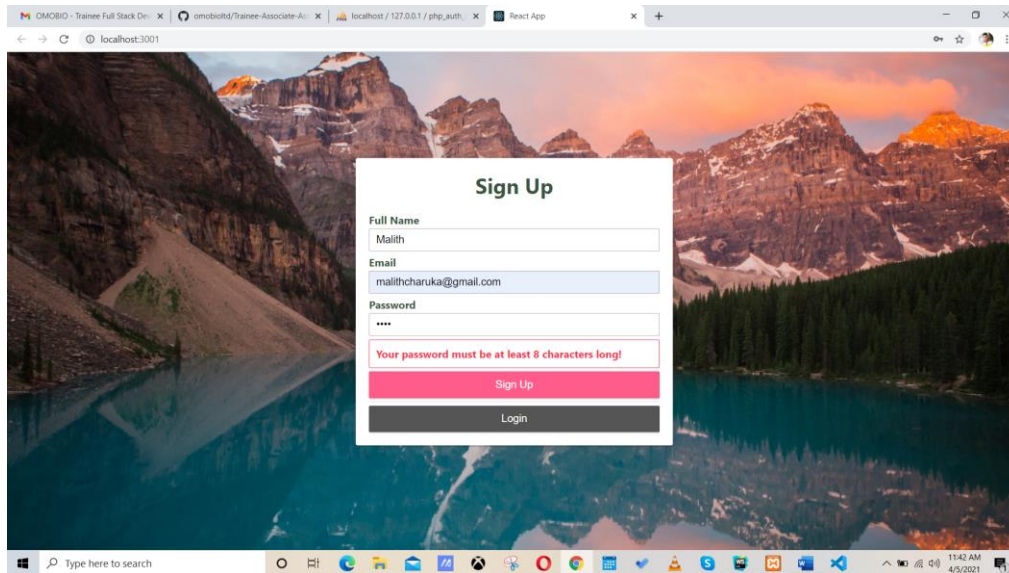
### 1. Empty Fields Check



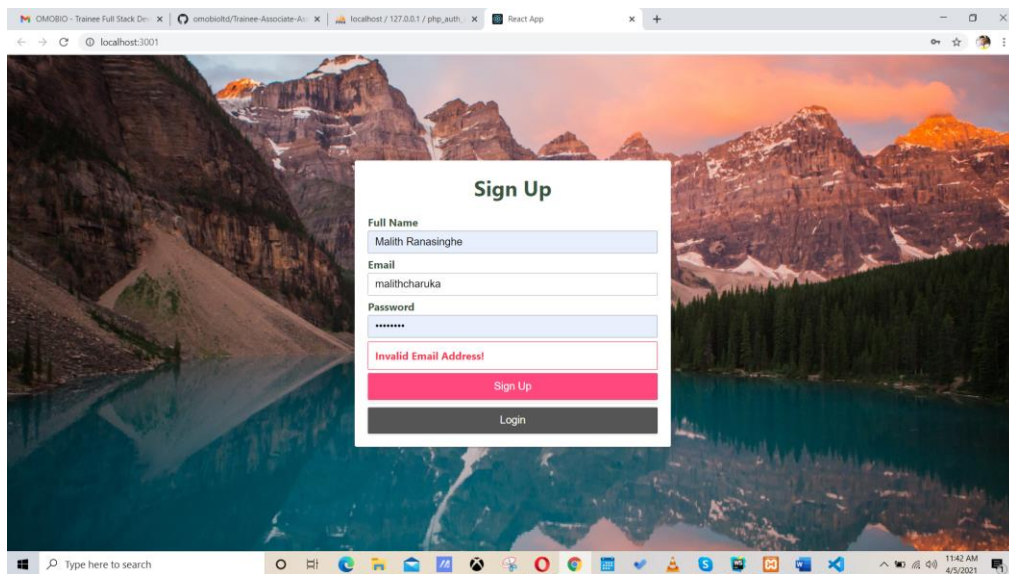
### 1. Check Same Email Registration



## 2. Password Characters Check

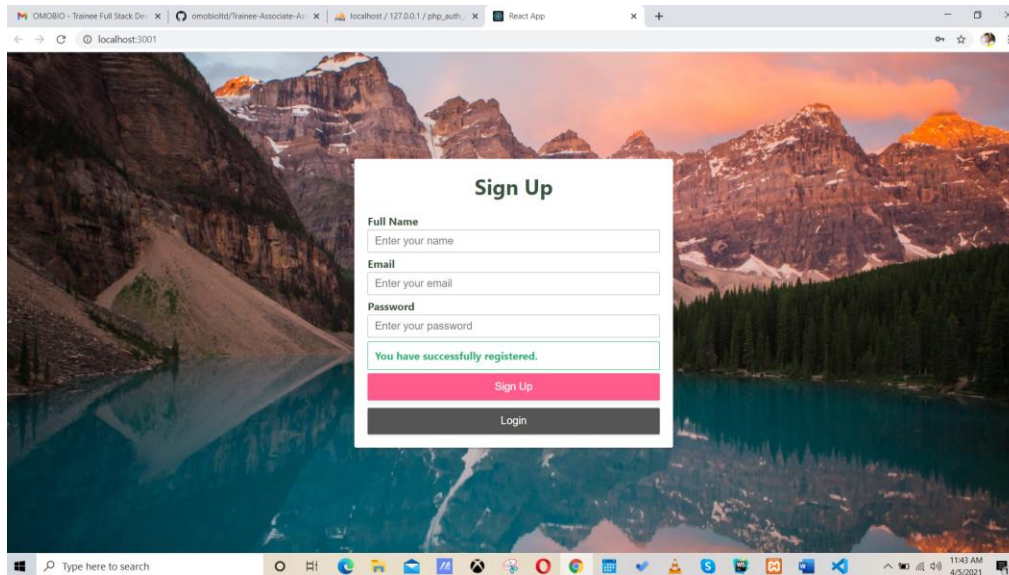


## 3. Invalid Email Check

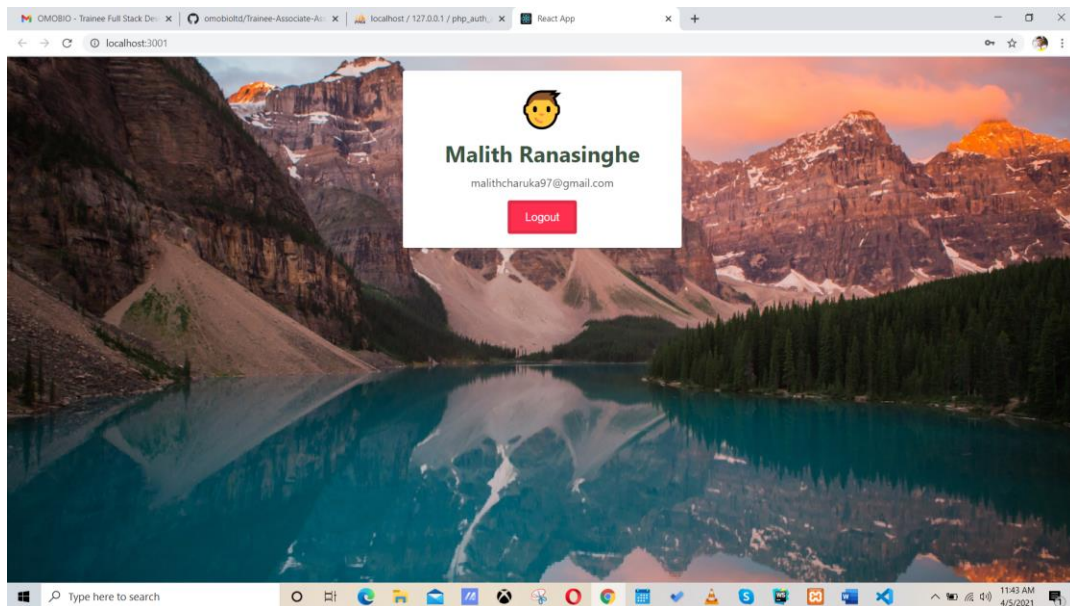




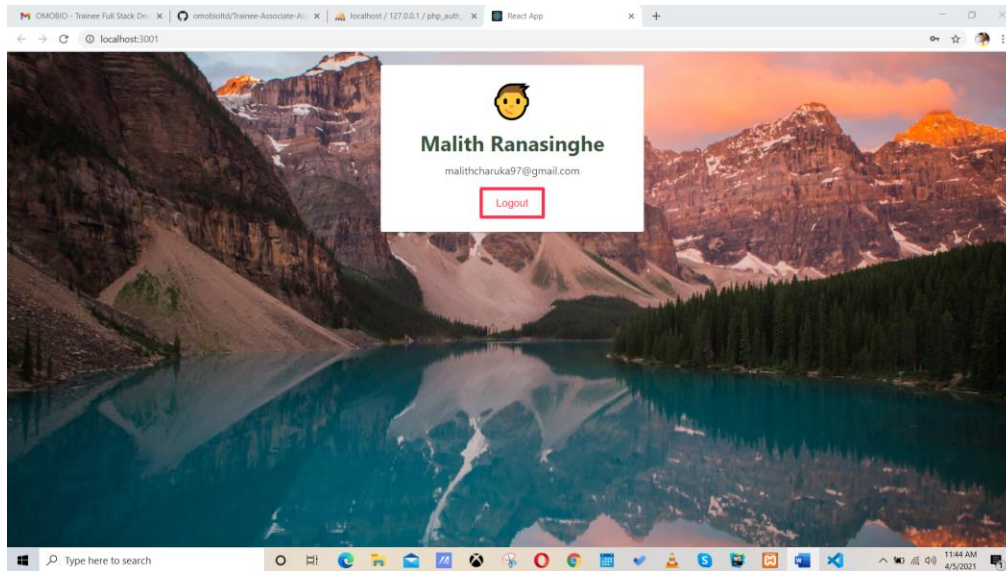
#### 4. Successfully Registered Message



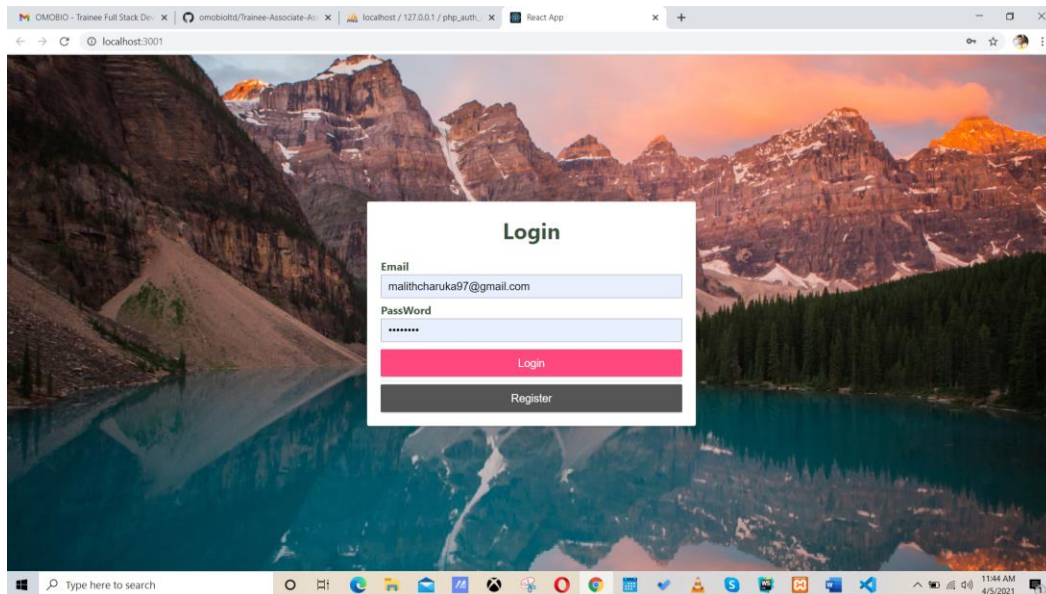
#### Home Page



## Logout User



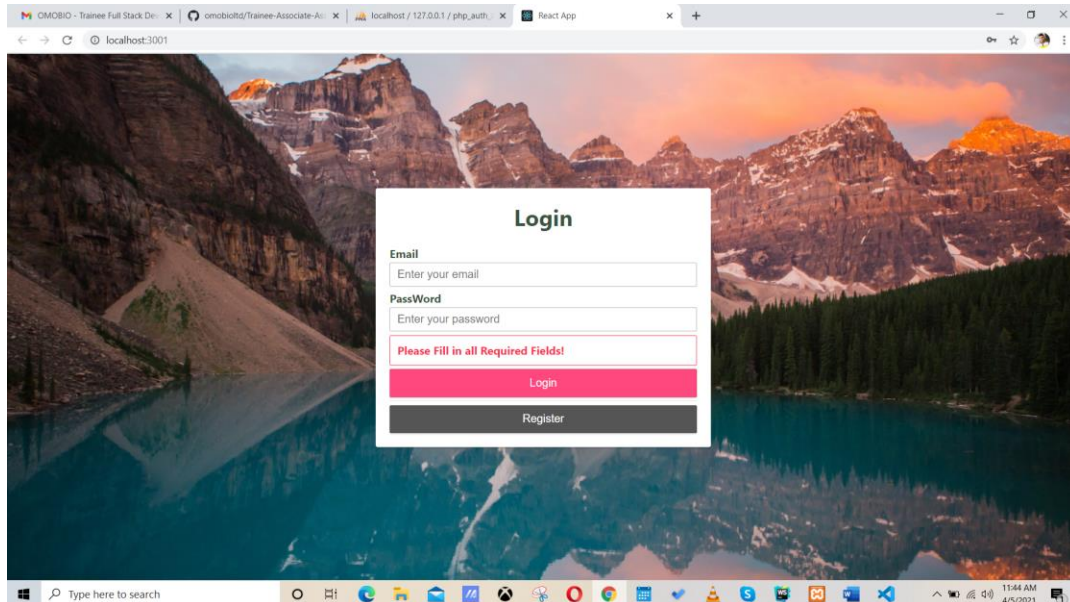
## Login Page



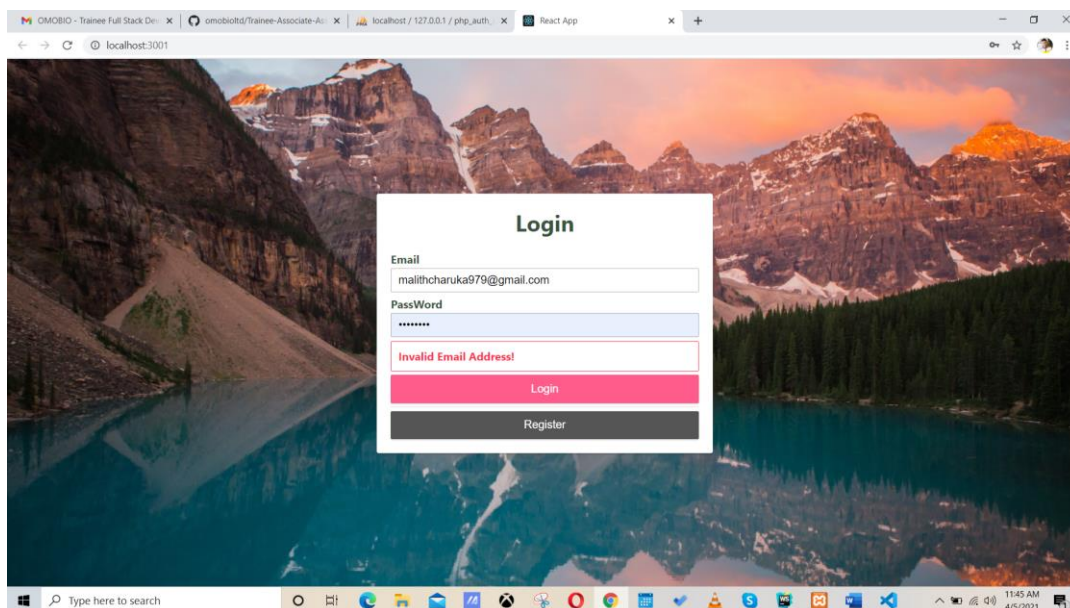


## Check Validation

### 1. Empty fields check.

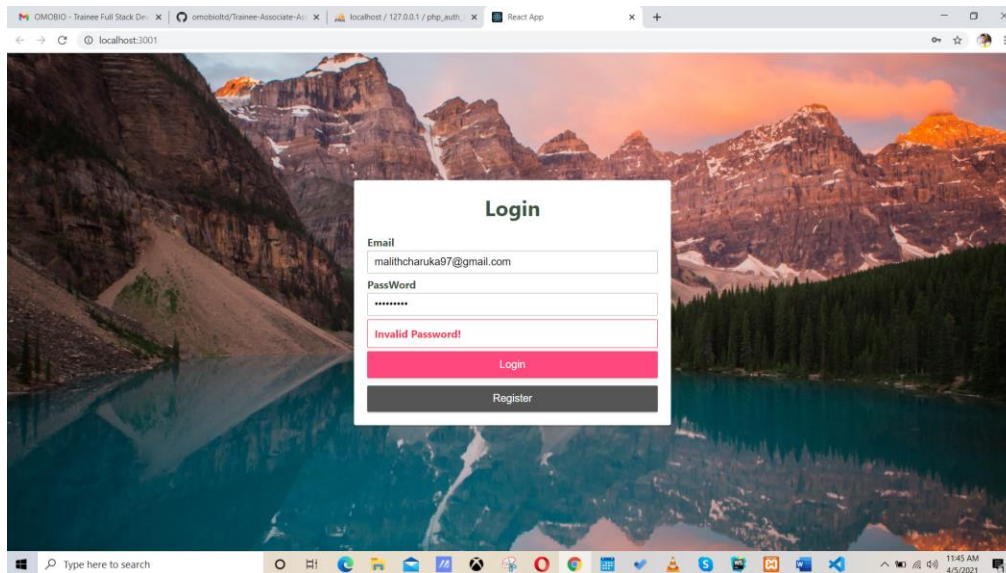


### 2. Invalid email address check

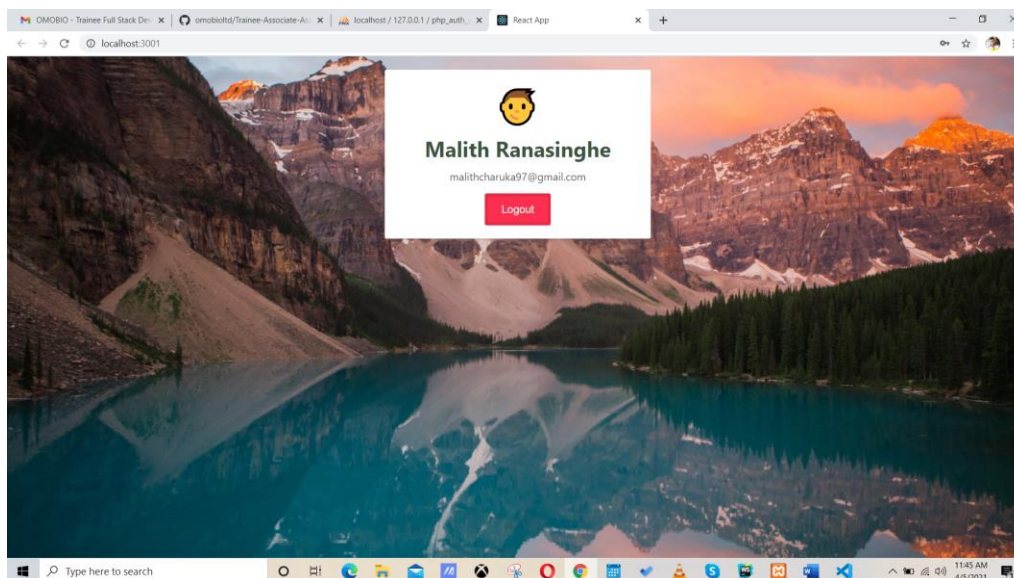




### 3. Invalid password check



### Successfully Login



## Testing the APIs

### API URLs –

1. Register  
<http://localhost/php-login-registration-api/register.php>
2. Login  
<http://localhost/php-login-registration-api/login.php>
3. Get User Info by giving token.  
<http://localhost/php-login-registration-api/user-info.php>

### Register a user API

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** <http://localhost/php-login-registration-api/register.php>
- Body Type:** raw
- Content Type:** JSON
- Request Body:**

```
1 {  
2   "name": "John Doe",  
3   "email": "johndoe@gmail.com",  
4   "password": "MyTestPass"  
5 }
```
- Response Status:** 201 Created
- Response Time:** 383ms
- Response Size:** 512 B
- Response Body:**

```
1 {  
2   "success": 1,  
3   "status": 201,  
4   "message": "You have successfully registered."  
5 }
```

## Login the user API

The screenshot displays a REST client interface for a login API. The top section shows the method **POST** and the URL `http://localhost/php-login-registration-api/login.php`. The **Send** button is highlighted. Below the URL bar, the **Body** tab is selected, and the **raw** radio button is chosen. The **JSON** format is selected from the dropdown menu. The body content is a JSON object: `{ "email": "johndoe@gmail.com", "password": "MyTestPass" }`. The bottom section shows the response status **200 OK**, time **84ms**, and size **778 B**. The **Body** tab is selected, and the **JSON** format is chosen. The response is a JSON object: `{ "success": 1, "message": "You have successfully logged in.", "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi1wvXC9sb2NhOGhvc3RcL3BocF9hdXRoX2FwaVwvIiwiaXVkiOiJoiaHR0cDpcL1wvbg9jYWxob3N0XC9waHBFYXV0aF9hcG1cLyIsIm1hdCI6MTU4NDZMDUxOSwiZXhwIjojNTg0MDM0MTE5LCJkYXRhIjp7InVzZXJfaWQiOiIxIn19.tOreXeguLDRmIuwkA3q8QCgCr9ucHLtvwAXLdJ1x0RE" }`. A watermark **Activate Windows** is visible in the bottom right corner.

POST `http://localhost/php-login-registration-api/login.php` Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded **raw** binary GraphQL **JSON** Beautify

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

Body Cookies Headers (10) Test Results Status: 200 OK Time: 84ms Size: 778 B Save Response

Pretty Raw Preview Visualize **JSON**

```
1 {  
2   "success": 1,  
3   "message": "You have successfully logged in.",  
4   "token":  
5     "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi1wvXC9sb2NhOGhvc3RcL3BocF9hdXRoX2FwaVwvIiwiaXVkiOiJoiaHR0cDpcL1wvbg9jYWxob3N0XC9waHBFYXV0aF9hcG1cLyIsIm1hdCI6MTU4NDZMDUxOSwiZXhwIjojNTg0MDM0MTE5LCJkYXRhIjp7InVzZXJfaWQiOiIxIn19.tOreXeguLDRmIuwkA3q8QCgCr9ucHLtvwAXLdJ1x0RE"  
}
```

Activate Windows

Get the user information by giving the token

The screenshot shows a REST client interface. The method is GET and the URL is `http://localhost/php-login-registration-api/user-info.php`. The Headers tab is selected, showing an Authorization header with the value `Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2Nhbnhvc3R3BocF9hdXR0X2FwaVwvliwiYXVkljoiaHR0cDpcL1wvYXV0aF9hcGcLyS1mlhdCI6MTU4NDZlMDUxOSwiZXhwLjoxNTg0MDM0MTE5LCJkYXRhIjp7InVzZXJfaWQiOiIxNjE5Lj0eXegulDRmWuWkA3q8QCgCr9ucHLtvwAXLdJlX0RE`. The response body is shown in JSON format:

```
{
  "success": 1,
  "status": 200,
  "user": {
    "name": "John Doe",
    "email": "johndoe@gmail.com"
  }
}
```

Password encrypted.

The screenshot shows a database query results interface. The table has columns: id, name, email, and password. The password column contains encrypted values.

id	name	email	password
8	Test new	testemail@gmail.com	\$2y\$10\$xWx/b0lmW9XAJNmg06N8P.85C7rrcVXVQFp8sofS6UI...
9	kasun	kasun@gmail.com	\$2y\$10\$sc16T4k.n1Y493dW16WIRbeWxjVsCKrBz7G4/8pBhcSE...
10	nimal	nimal@gmail.com	\$2y\$10\$IfvwiKtnrZzqhMVnFYGqcOjdwfB4QChWNUmAI0f4s7...
11	Malith Ranasinghe	malith@gmail.com	\$2y\$10\$Np8MQ4XK25gxxTiE.Aeye0o7pBmJXAIWvOgqxXKMLf...
12	Malith Ranasinghe	malith123@gmail.com	\$2y\$10\$KaSLR.BqvJt0bBCi7SfBvePsT4LaH3ciNk3aqdAKcfj...