

វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា

Institute of Technology of Cambodia

Department of Information and Communication



Subject: Internet Programming

Professor: HOK TIN

Assignment: TP10

Name: LAY Vicha

ID: e20190494

Group: I4-GIC-B

2022-2023


## Contents

1.	Git url : <a href="https://github.com/LAY-VICHA/TP_IP_year4/tree/main/TP10">https://github.com/LAY-VICHA/TP_IP_year4/tree/main/TP10</a> .....	1
2.	EX1 – Connect node project to Mongodb .....	1
3.	EX2 – add more API to node project (me, update password, user, delete) .....	3

1. Git url : [https://github.com/LAY-VICHA/TP\\_IP\\_year4/tree/main/TP10](https://github.com/LAY-VICHA/TP_IP_year4/tree/main/TP10)

## 2. EX1 – Connect node project to MongoDB

← → ↻ http://localhost:3000/Register



# Welcome

Authentication App~

[Login](#) | [Register](#)

### Sign Up

Please fill in this form to create an account.

Email

Username

First name


Last name

Password

Login

- After register, our application will route to login page


← → ↻ http://localhost:3000



# Welcome

Authentication App~

[Login](#) | [Register](#)




Email

Password

[Forget Password?](#)

Login


← → ↻ http://localhost:3000



## Welcome

Authentication App~

[Login](#) | [Register](#)



Email


Password

[Forget Password?](#)

Login

- It will route to home page after a success login

← → ↻ http://localhost:3000/home



## Welcome

Authentication App~

[Login](#) | [Register](#)

### You're in Home page

well done 🎉

[back to login](#)

### 3. EX2 – add more API to node project (me, update password, user, delete)

The image displays two screenshots of a REST client interface, likely Postman, showing API requests and responses.

**Top Screenshot: GET Request**

- Method:** GET
- URL:** localhost:3001/me
- Body:**

```
1 {
2   "email": "sok@gmail.com",
3   "password": "123456"
4 }
```
- Status:** 200 OK
- Time:** 16 ms
- Size:** 494 B
- Body (JSON):**

```
1 {
2   "_id": "647df9dfe0511af07d8408e7",
3   "email": "sok@gmail.com",
4   "username": "sok",
5   "firstname": "sok",
6   "lastname": "lay",
7   "password": "$2a$10$uwhn1hUYPEgu822bx.4YaeJgzQ5D6m2LwBvqN19TyvZzztLfoqWqq",
8   "__v": 0
9 }
```

**Bottom Screenshot: POST Request**

- Method:** POST
- URL:** localhost:3001/update-password
- Body:**

```
1 {
2   "email": "sok@gmail.com",
3   "password": "1234567"
4 }
```
- Status:** 200 OK
- Time:** 165 ms
- Size:** 518 B
- Body (JSON):**

```
1 {
2   "success": true,
3   "data": {
4     "_id": "647df9dfe0511af07d8408e7",
5     "email": "sok@gmail.com",
6     "username": "sok",
7     "firstname": "sok",
8     "lastname": "lay",
9     "password": "$2a$10$dKXhYztm6RR7bFvJaFwCF0k2FV2Gag0r/SzNw8TkPsbncQdZaRwqy",
10    "__v": 0
11  }
12 }
```

- After update the user's password

The first screenshot shows a POST request to `localhost:3001/login`. The body is raw text containing JSON data: `{ "email": "sok@gmail.com", "password": "1234567" }`. The response is a 200 OK status with a time of 208 ms and size of 679 B. The response body is JSON: `{ "success": true, "data": { "_id": "647df9dfe0511af07d8408e7", "email": "sok@gmail.com", "username": "sok" } }`.

The second screenshot shows a POST request to `localhost:3001/update-user`. The body is raw text containing JSON data: `{ "email": "sok@gmail.com", "firstname": "tola", "lastname": "sao" }`. The response is a 200 OK status with a time of 16 ms and size of 519 B. The response body is JSON: `{ "success": true, "data": { "_id": "647df9dfe0511af07d8408e7", "email": "sok@gmail.com", "username": "sok", "firstname": "tola", "lastname": "sao", "password": "$2a$10$dKXhYztm6RR7bFvJaFwCF0k2FV2Gag0x/SzNw8TkPsbncQdZaRwqy", "__v": 0 } }`.

The screenshot shows a REST client interface with a POST request to `localhost:3001/delete-user`. The request body is a JSON object with the following fields:

```
1 {
2   "email": "sok@gmail.com",
3   "firstname": "tola",
4   "lastname": "sao"
5 }
```

The response is a 200 OK status with a time of 31 ms and a size of 586 B. The response body is a JSON object:

```
1 {
2   "success": true,
3   "data": {
4     "_id": "647df9dfe0511af07d8408e7",
5     "email": "sok@gmail.com",
6     "username": "sok",
7     "firstname": "tola",
8     "lastname": "sao",
9     "password": "$2a$10$dKXhYztm6RR7bFvJaFwCF0k2FV2Gag0x/SzNw8TkPsbncQdZaRwqy",
10    "__v": 0
11  }
12 }
```

- After delete the user

The screenshot shows a REST client interface with a POST request to `localhost:3001/login`. The request body is a JSON object with the following fields:

```
1 {
2   "email": "sok@gmail.com",
3   "password": "1234567"
4 }
```

The response is a 200 OK status with a time of 13 ms and a size of 348 B. The response body is a JSON object:

```
1 {
2   "success": false,
3   "err": "Email is incorrected."
4 }
```