Backend Assignment

Submitted by: Akash Biswas (Roll no.: 200074)

Assumptions

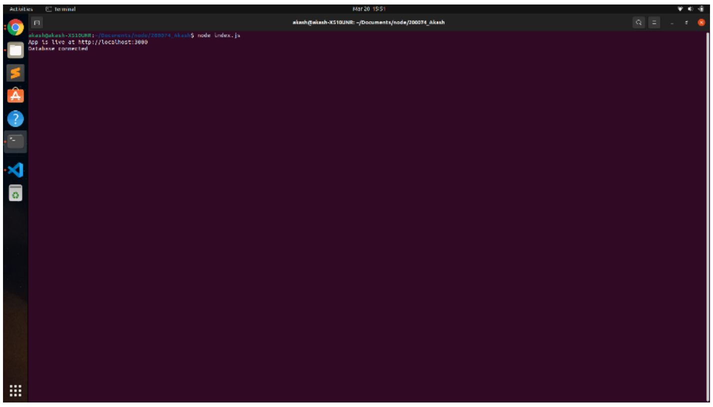
1. While registering, a new user will have to fill the following details in the following json format shown

   ```
   {
           “email”:”user@domain.com (must be unique)”,
           “password”:”password string of minimum length 8”,
           “name”:”Full name of the user”,
           “phone”:”Phone number of the user”,
           “address”:{
                   “line1”:”Address line 1”,
                   “line2”:”Address line 2 is optional”,
                   “city”:”city of the User’s address”,
                   “state”:”state of the User”,
                   “pin”:”6-digit pin code”
           }
   }
   ```

   While registering, one must provide all the details (except address line 2, as it is optional).
2. I have used **MongoDB** as my database system.
3. GET request for: /, /api/profile, /api/logout
4. POST request for: /api/register, /api/login
5. Default port used is 3000.

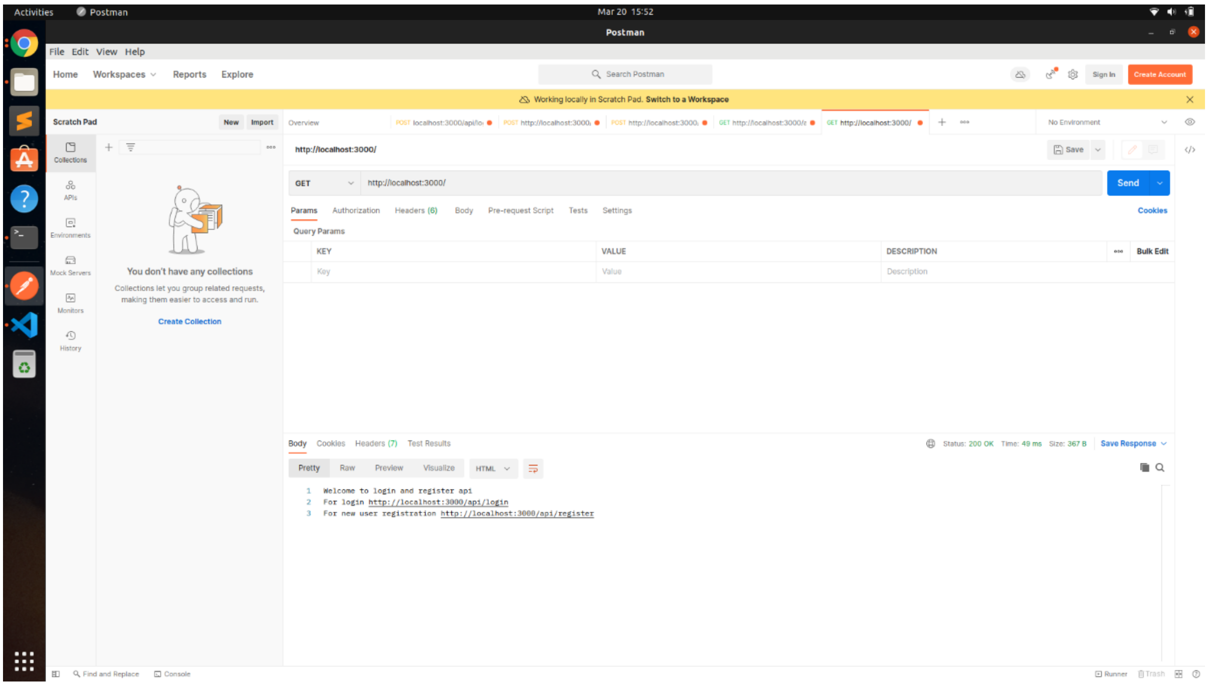
Setup

After installing the dependencies with `npm install`, we can run the program by `node index.js`



```
akash@akash-X510UNR:~/Documents/node/200074_Akash$ node index.js
App is live at http://localhost:3000
Database connected
```

On doing so, we will get prompted that app is live and whether the database is connected. Once we send a **GET** request to http://localhost:PORT/, we get the following response which basically gives the links for login and registration.



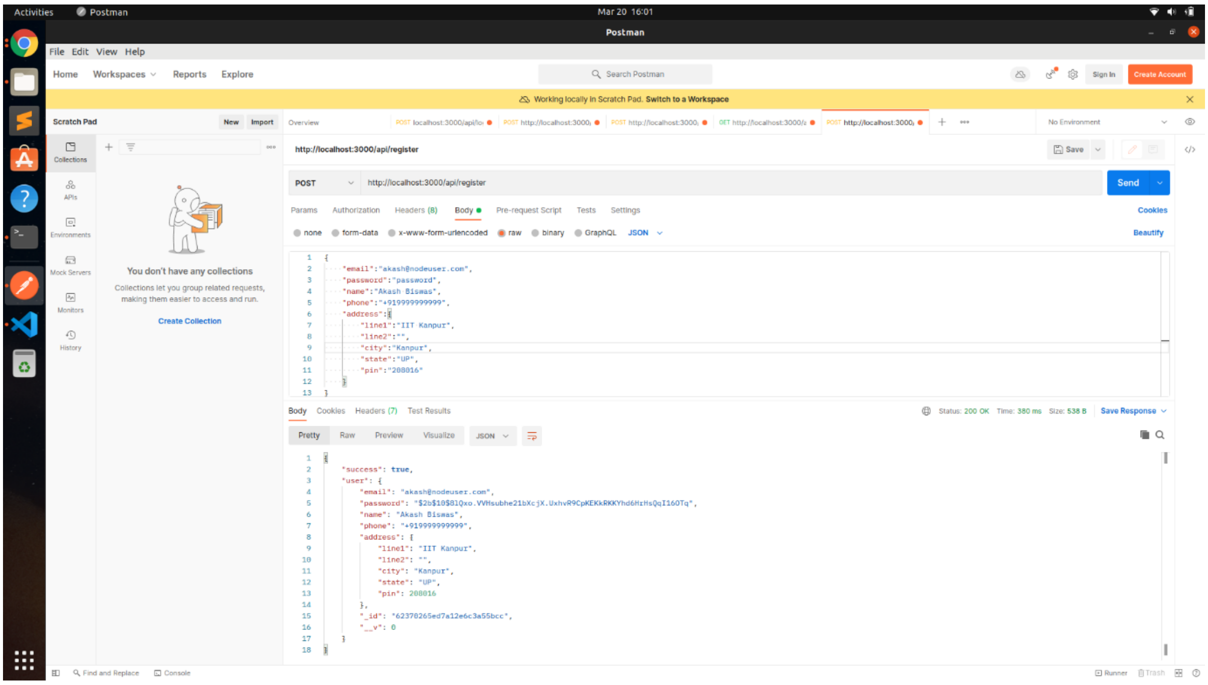
GET http://localhost:3000/

Status: 200 OK Time: 49 ms Size: 367 B

```
Welcome to login and register api
For login http://localhost:3000/api/login
For new user registration http://localhost:3000/api/register
```

Registration (Signup)

We can **POST** request as follows to http://localhost:PORT/api/register to register a new user.



POST http://localhost:3000/api/register

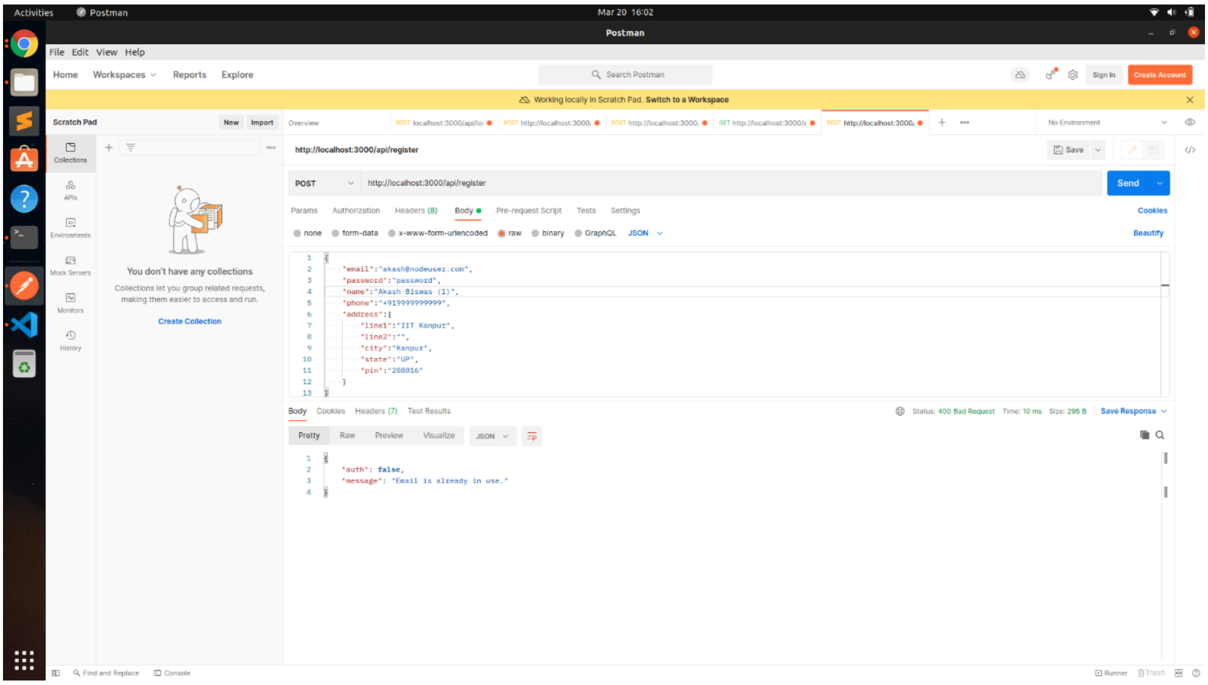
```
{
    "email":"akash@nodeuser.com",
    "password":"password",
    "name":"Akash Biswas",
    "phone":"+919999999999",
    "address":{
        "line1":"IIT Kanpur",
        "line2":"",
        "city":"Kanpur",
        "state":"UP",
        "pin":"208016"
    }
}
```

Status: 200 OK Time: 380 ms Size: 538 B

```
{
    "success": true,
    "user": {
        "email": "akash@nodeuser.com",
        "password": "$2b$10$81Qxo.VVHsubhe21bXcjX.UxhvR9CpKEKkRKKYhd6HrHsQqI160Tq",
        "name": "Akash Biswas",
        "phone": "+919999999999",
        "address": {
            "line1": "IIT Kanpur",
            "line2": "",
            "city": "Kanpur",
            "state": "UP",
            "pin": 208016
        },
        "_id": "62370265ed7a12e6c3a55bcc",
        "__v": 0
    }
}
```

It successfully saves the user's email, password, name, phone no. and address into database and returns "success":true. But it doesn't return "success":true, in case it doesn't save the user's detail.

For example, if we try to register another user with the same email



POST http://localhost:3000/api/register

```
{
    "email":"akash@nodeuser.com",
    "password":"password",
    "name":"Akash Biswas (1)",
    "phone":"+919999999999",
    "address":{
        "line1":"IIT Kanpur",
        "line2":"",
        "city":"Kanpur",
        "state":"UP",
        "pin":"208016"
    }
}
```

Status: 400 Bad Request Time: 10 ms Size: 295 B

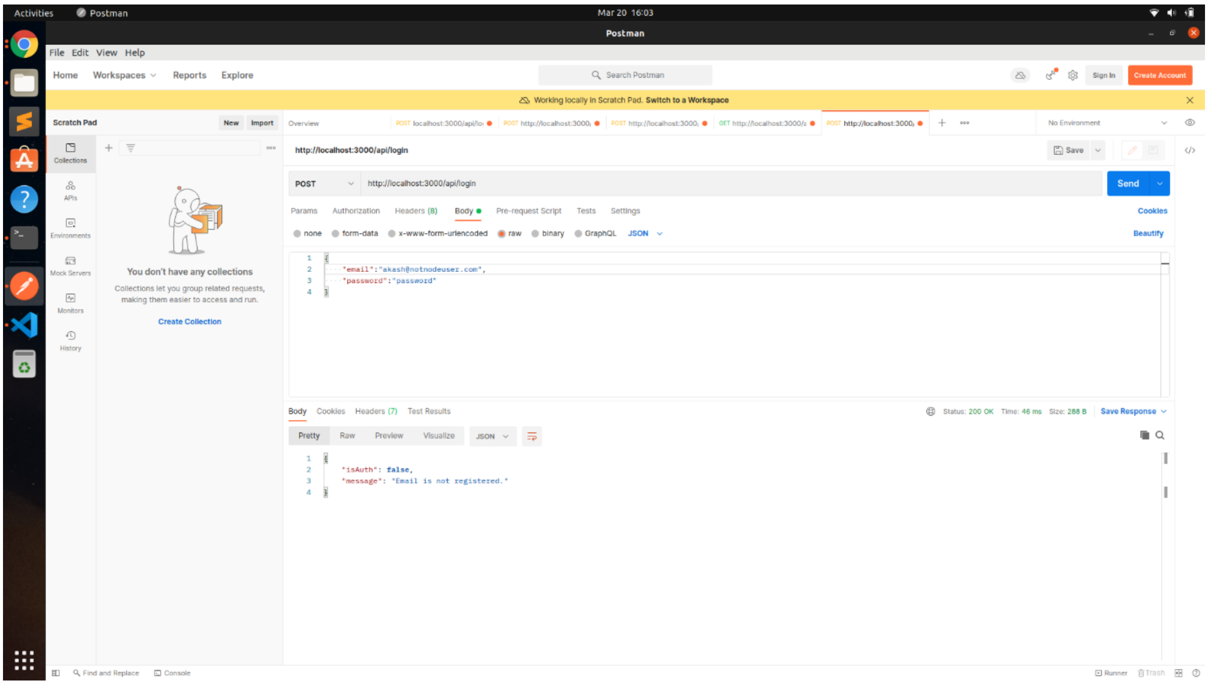
```
{
    "auth": false,
    "message": "Email is already in use."
}
```

Here we return "auth":false, as it first authenticates whether the email is already in use . Also we get a message saying the same.

Login

As, we have created a user, we can now **POST** request as follows to http://localhost:PORT/api/ to sign in. But first we test the incorrect credentials.

In this, we have given a email with which no user is registered.



POST http://localhost:3000/api/login

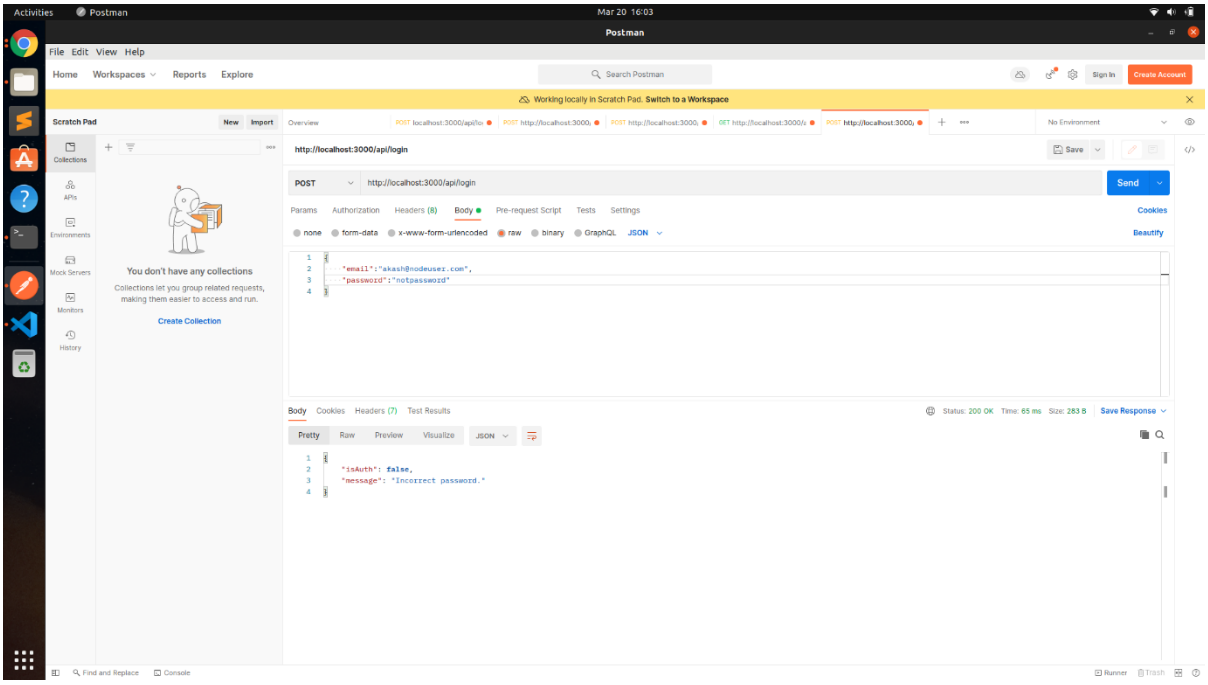
```
{
    "email":"akash@notnodeuser.com",
    "password":"password"
}
```

Status: 200 OK Time: 46 ms Size: 288 B

```
{
    "isAuth": false,
    "message": "Email is not registered."
}
```

As expected, authentication fails, and we get a message saying, “Email is not registered”.

Now, we try an incorrect password with the email with which we registered the user.



POST http://localhost:3000/api/login

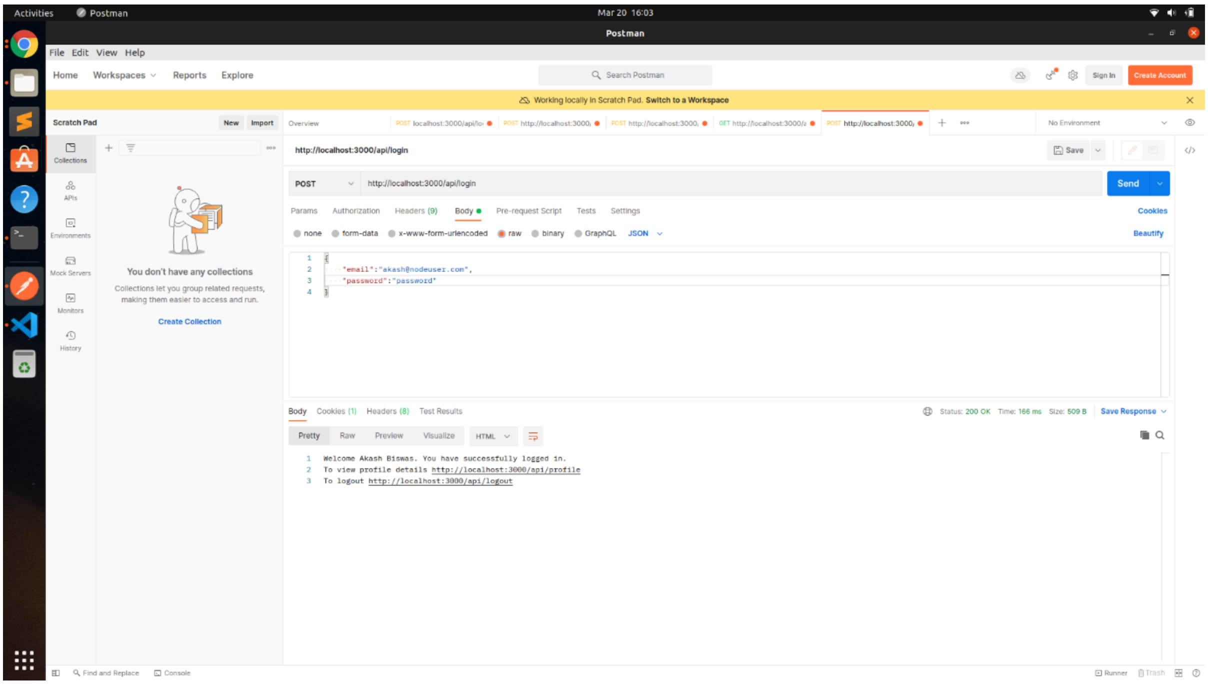
```
{
    "email":"akash@nodeuser.com",
    "password":"notpassword"
}
```

Status: 200 OK Time: 65 ms Size: 283 B

```
{
    "isAuth": false,
    "message": "Incorrect password."
}
```

Here too the authentication fails, and we get the message explaining the cause which is “Incorrect password”.

Now we give the correct credentials.



POST http://localhost:3000/api/login

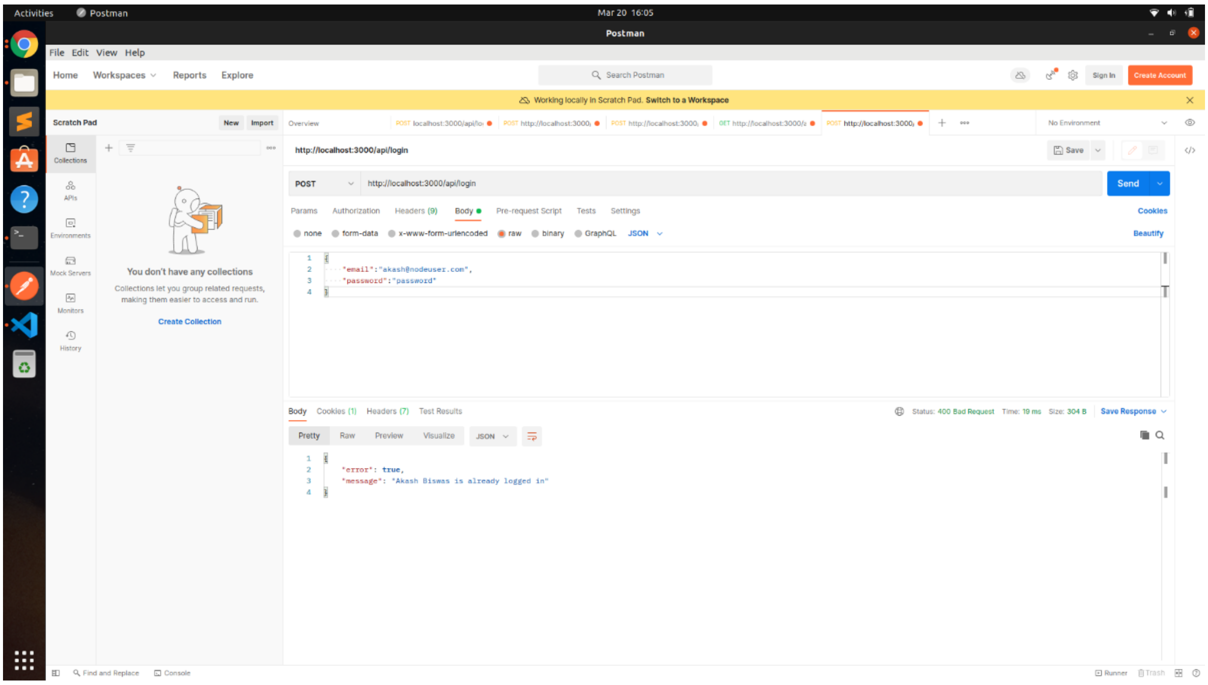
```
{
    "email":"akash@nodeuser.com",
    "password":"password"
}
```

Status: 200 OK Time: 166 ms Size: 509 B

```
Welcome Akash Biswas. You have successfully logged in.
To view profile details http://localhost:3000/api/profile
To logout http://localhost:3000/api/logout
```

After verifying the credentials, we get a welcome message greeting the user by his **name**. Also, we get links to the profile and logout page.

Also, if we again try to login without first logging out, we get the following error and a message saying that the user is already logged in.



Activities
Postman
Mar 20 16:05
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Sign In
Create Account
Working locally in Scratch Pad. Switch to a Workspace
Scratch Pad
New
Import
You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection
No Environment
Save
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
Beautify
Pretty Raw Preview Visualize JSON
Save Response
Find and Replace
Console
Runner
Trash

http://localhost:3000/api/login

POST http://localhost:3000/api/login

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

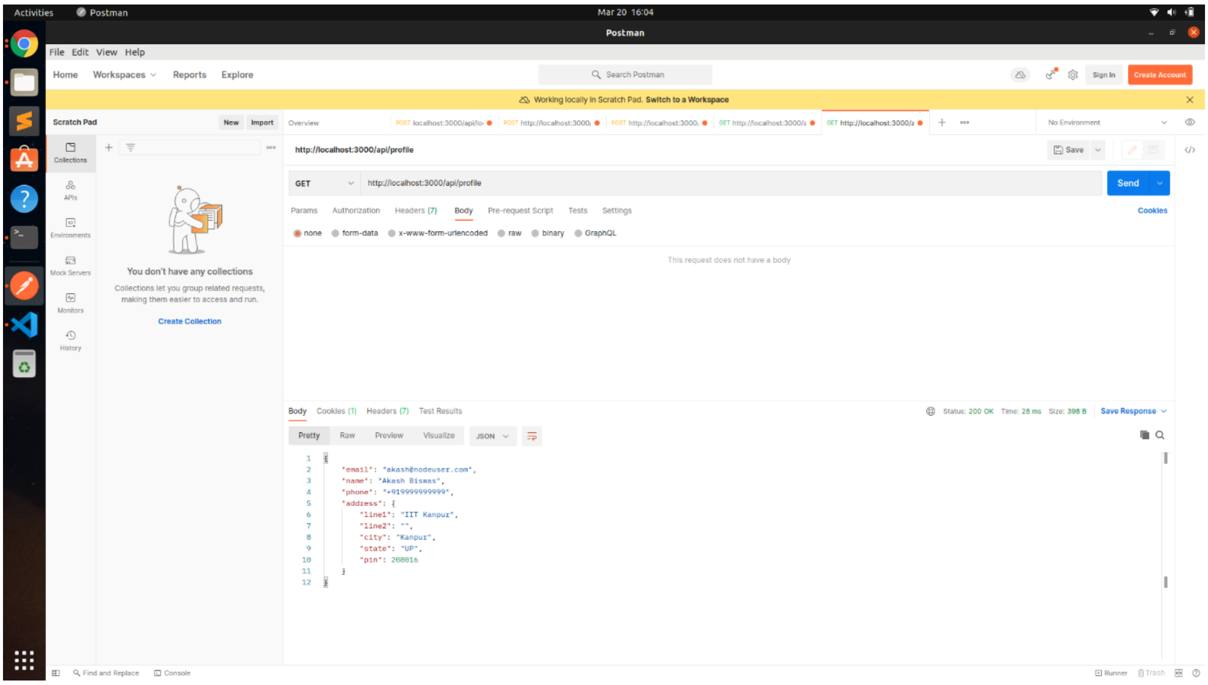
```
{
    "email":"akash@nodeuser.com",
    "password":"password"
}
```

Body Cookies (1) Headers (7) Test Results — Status: 400 Bad Request Time: 19 ms Size: 304 B

```
{
    "error": true,
    "message": "Akash Biswas is already logged in"
}
```

User profile details

After successfully logging in, we can send a **GET** request to http://localhost:PORT/api/profile which sends the user details in json format in response.



Activities
Postman
Mar 20 16:04
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Sign In
Create Account
Working locally in Scratch Pad. Switch to a Workspace
Scratch Pad
New
Import
You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection
No Environment
Save
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
Pretty Raw Preview Visualize JSON
Save Response
Find and Replace
Console
Runner
Trash

http://localhost:3000/api/profile

GET http://localhost:3000/api/profile

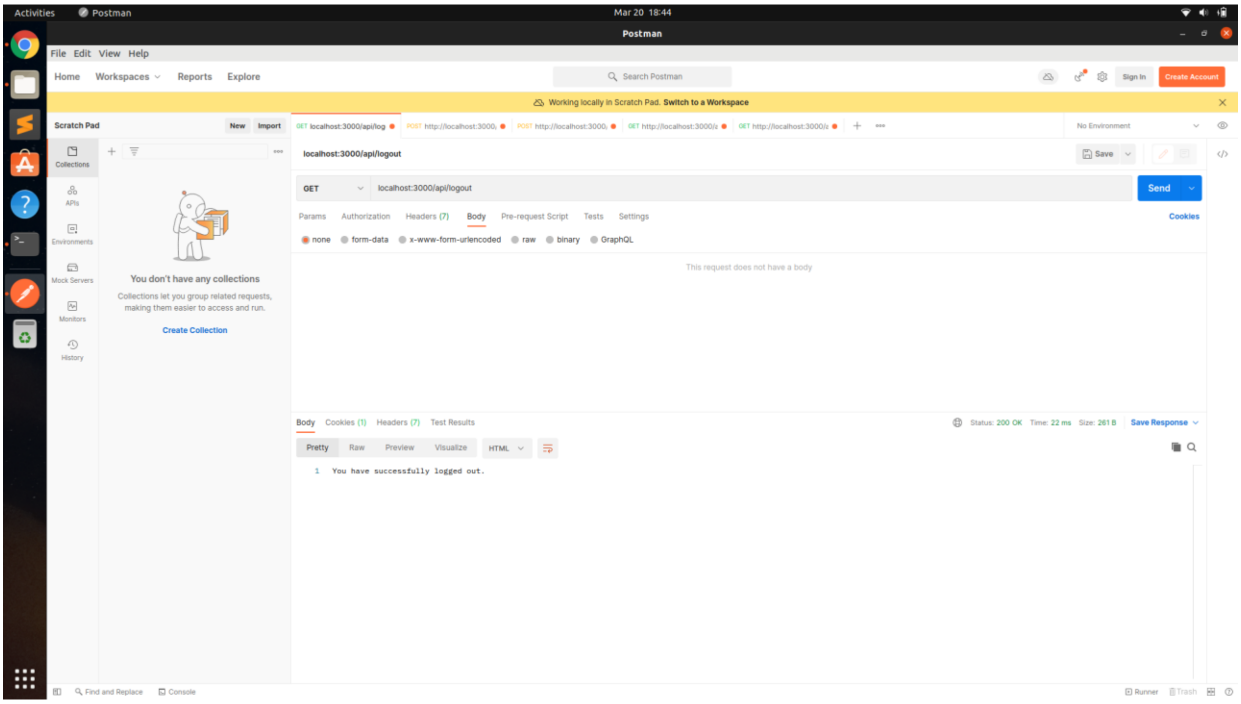
Body: ● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

Body Cookies (1) Headers (7) Test Results — Status: 200 OK Time: 28 ms Size: 398 B

```
{
    "email": "akash@nodeuser.com",
    "name": "Akash Biswas",
    "phone": "+919999999999",
    "address": {
        "line1": "IIT Kanpur",
        "line2": "",
        "city": "Kanpur",
        "state": "UP",
        "pin": 208016
    }
}
```

Now to check whether the user's details are accessible only after signing in or not, we first logout from the current session by sending a **GET** request to http://localhost/PORT/api/logout



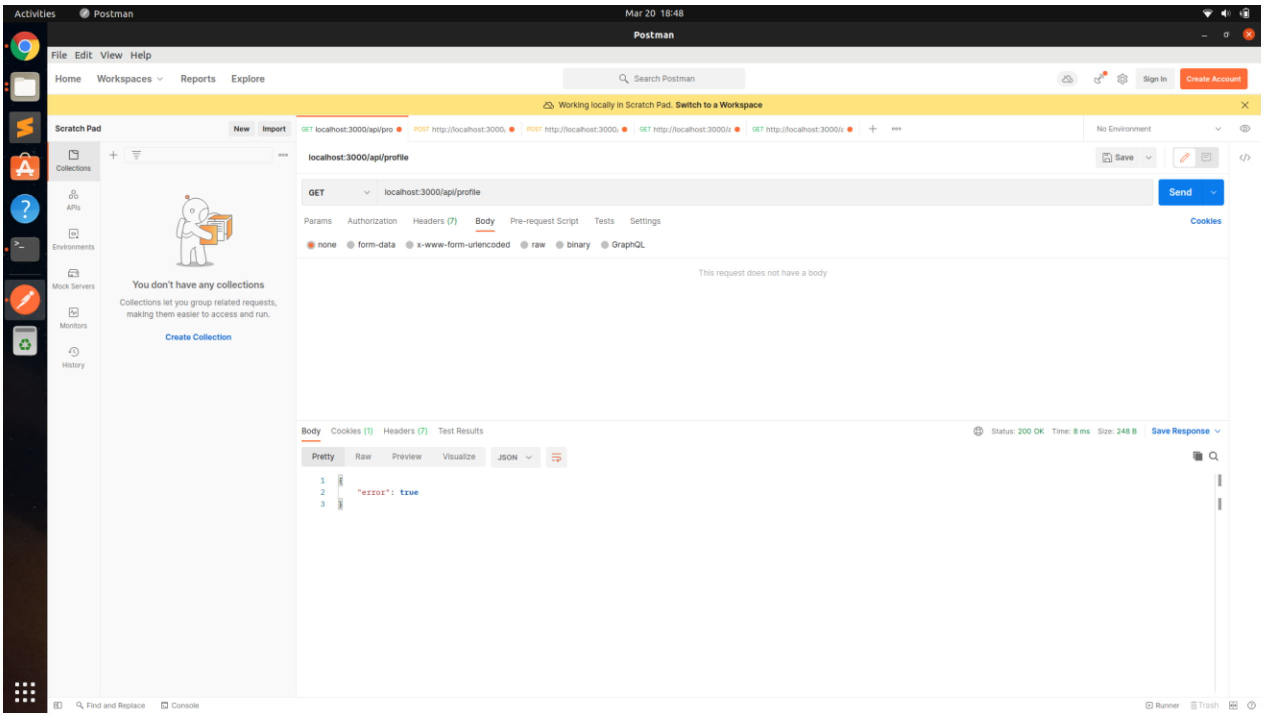
GET localhost:3000/api/logout

Body: none

Status: 200 OK Time: 22 ms Size: 261 B

```
You have successfully logged out.
```

As, it shows the message saying that the user has successfully logged out, we now try sending a **GET** request to http://localhost/PORT/api/profile



GET localhost:3000/api/profile

Body: none

Status: 200 OK Time: 8 ms Size: 248 B

```
{
    "error": true
}
```

As evident, we get an error if we try to access user details after logging out or without signing in.