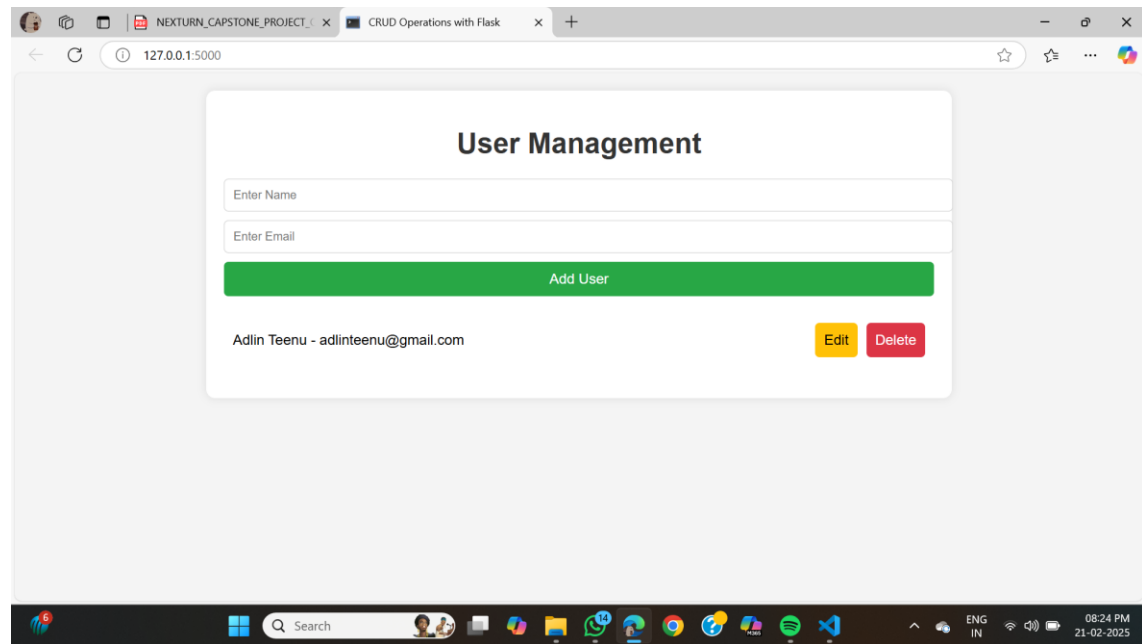
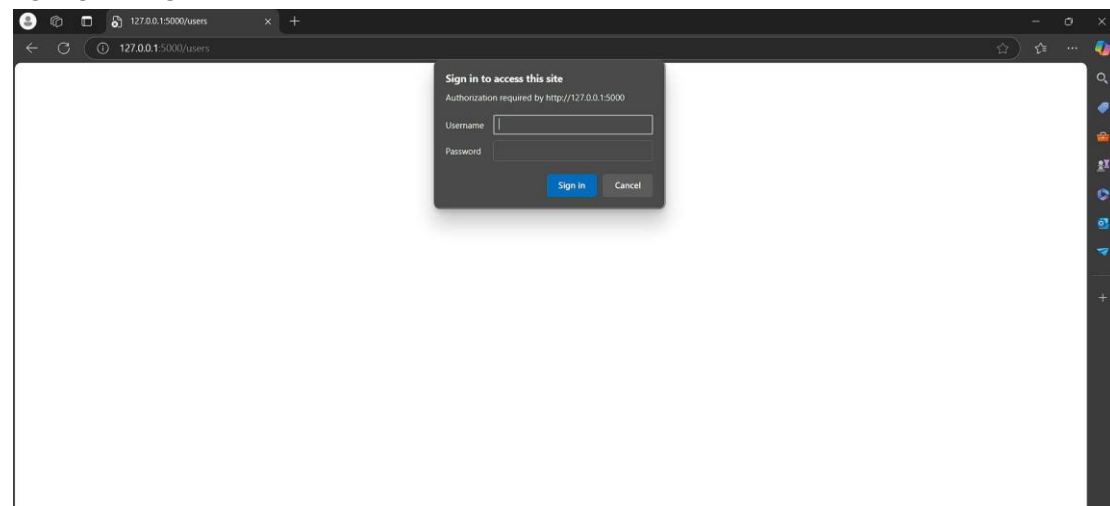


## USER MANAGEMENT



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000'. The page title is 'User Management'. The main content area contains a form with two input fields labeled 'Enter Name' and 'Enter Email'. Below these fields is a green button labeled 'Add User'. At the bottom of the form, there is a table with one user entry: 'Adlin Teenu - adlinteenu@gmail.com'. To the right of this entry are two buttons: 'Edit' (yellow) and 'Delete' (red).

## AUTHORIZATION



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/users'. A modal dialog box is centered on the screen. The dialog box has a title 'Sign in to access this site' and a subtitle 'Authorization required by http://127.0.0.1:5000'. It contains two input fields: 'Username' and 'Password'. At the bottom of the dialog box are two buttons: 'Sign in' (blue) and 'Cancel' (gray).

## CRUD OPERATIONS

### 1. GET ALL USERS

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:5000/users`. The request is saved and the response is displayed in the Body tab, showing a JSON array of two users.

**Request:**

- Method: GET
- URL: `http://127.0.0.1:5000/users`

**Response:**

```
200 OK - 11 ms - 257 B
```

```
{
  "id": 1,
  "username": "harish"
},
{
  "id": 2,
  "username": "k"
}
```

### 2. SEARCH BY ID

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:5000/users/2`. The request is configured with Basic Auth and the response is displayed in the Body tab, showing a JSON object for the user with id 2.

**Request:**

- Method: GET
- URL: `http://127.0.0.1:5000/users/2`
- Auth Type: Basic Auth
- Username: `harish`
- Password: `harish14`

**Response:**

```
200 OK - 11 ms - 198 B
```

```
{
  "id": 2,
  "username": "k"
}
```

### 3. UPDATE USER

The screenshot shows the Postman interface with a PUT request to `http://127.0.0.1:5000/users/3`. The request body is a JSON object: `{ "username": "new username", "password": "s" }`. The response is a 200 OK status with a response time of 25 ms and a body of `{ "message": "User updated successfully!" }`.

Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import Overview Getting started PUT http://127.0.0.1:5000/us + No environment

Collections + Your collection + Authorization Type new

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

PUT http://127.0.0.1:5000/users/3 Save

Params Authorization Headers (10) Body Scripts Settings Cookies Beautify

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

```
1 {
2   "username": "new username",
3   "password": "s"
4 }
```

Body Cookies Headers (5) Test Results 200 OK 25 ms 211 B

JSON Preview Visualize

```
1 {
2   "message": "User updated successfully!"
3 }
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

### 4. DELETE

The screenshot shows the Postman interface with a DELETE request to `http://127.0.0.1:5000/users/3`. The request body is a JSON object: `{ "username": "new username", "password": "s" }`. The response is a 200 OK status with a response time of 28 ms and a body of `{ "message": "User deleted successfully!" }`.

Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import Overview Getting started DEL http://127.0.0.1:5000/us + No environment

Collections + Your collection + Authorization Type new

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

DELETE http://127.0.0.1:5000/users/3 Save

Params Authorization Headers (10) Body Scripts Settings Cookies Beautify

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

```
1 {
2   "username": "new username",
3   "password": "s"
4 }
```

Body Cookies Headers (5) Test Results 200 OK 28 ms 211 B

JSON Preview Visualize

```
1 {
2   "message": "User deleted successfully!"
3 }
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

## TESTING

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

127.0.0.1 - - [21/Feb/2025 00:09:03] "GET /users/2 HTTP/1.1" 200 -
127.0.0.1 - - [21/Feb/2025 00:09:34] "PUT /users/3 HTTP/1.1" 200 -
127.0.0.1 - - [21/Feb/2025 00:10:40] "DELETE /users/3 HTTP/1.1" 200 -
(myenv) PS H:\NEXTURN CAPSTONE PROJ> cd tests
(myenv) PS H:\NEXTURN CAPSTONE PROJ\tests> pytest tests_routes.py
===== test session starts =====
platform win32 -- Python 3.12.9, pytest-8.3.4, pluggy-1.5.0
rootdir: H:\NEXTURN CAPSTONE PROJ\tests
collected 1 item

tests_routes.py . [100%]

===== 1 passed in 1.11s =====
(myenv) PS H:\NEXTURN CAPSTONE PROJ\tests>
```

## DOCKER

