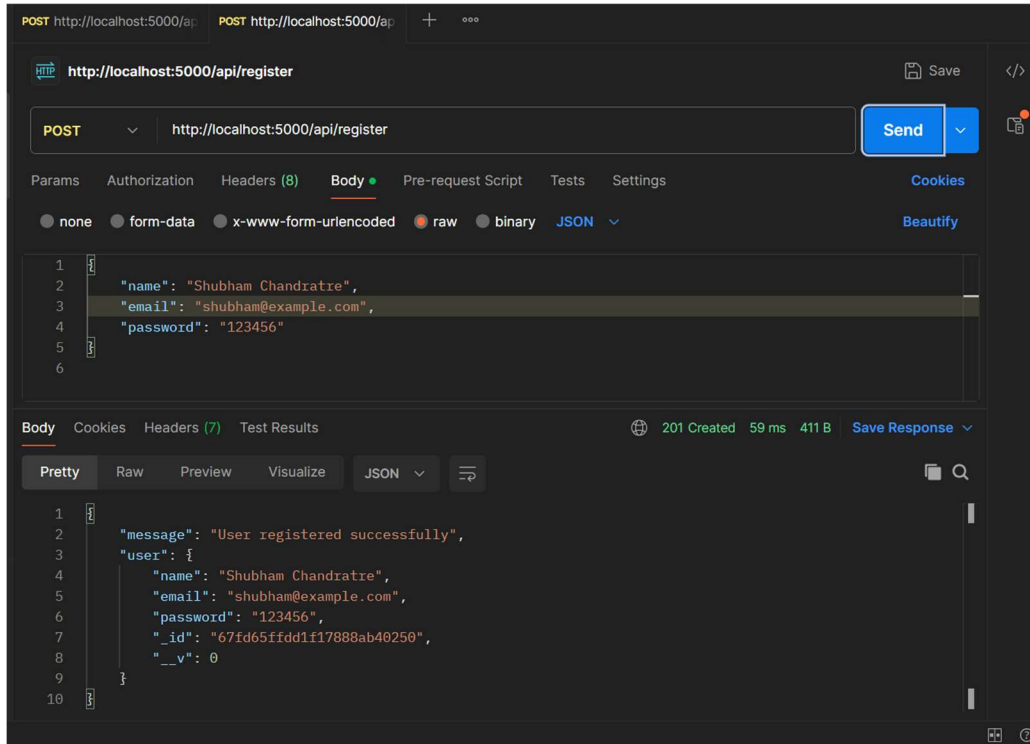
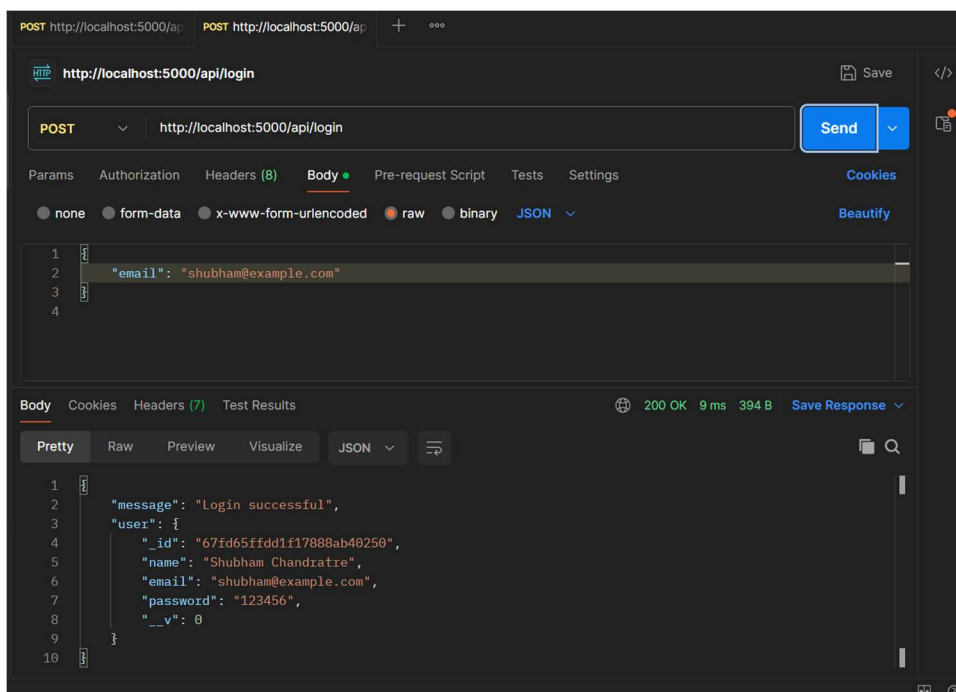


WAD Assignment – 3b Database CRUD Operations

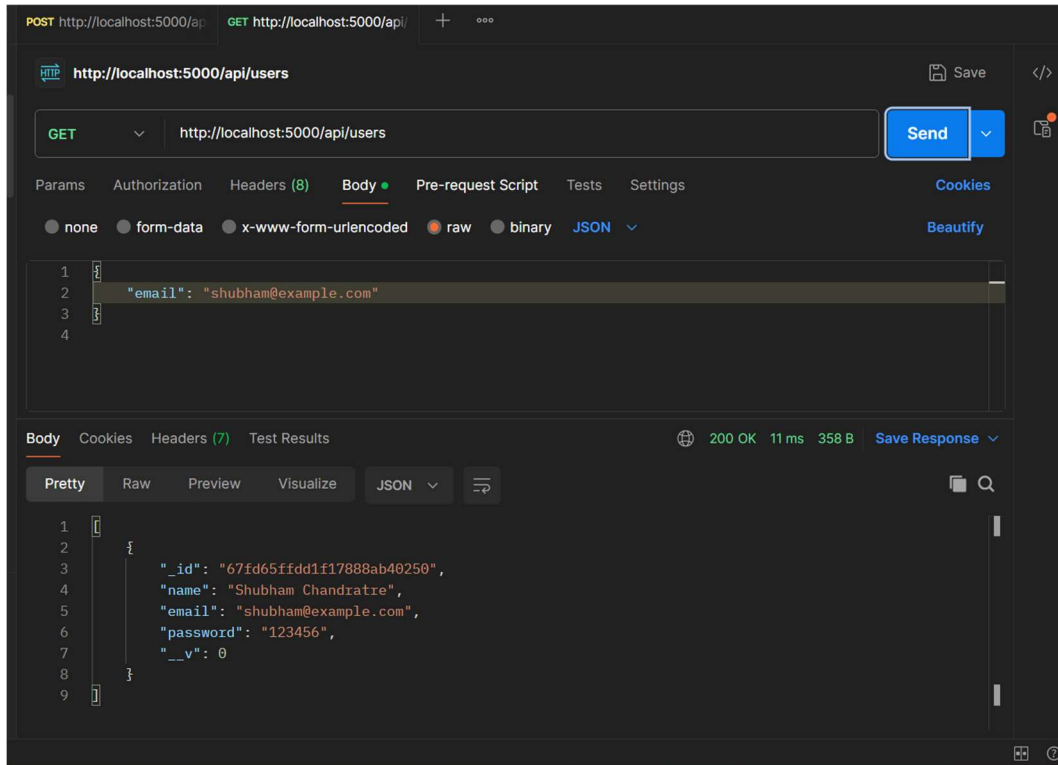
1.Register



2. login



3. display users



POST http://localhost:5000/api/ GET http://localhost:5000/api/ + ...

HTTP http://localhost:5000/api/users Save </>

GET http://localhost:5000/api/users Send

Params Authorization Headers (8) Body • Pre-request Script Tests Settings Cookies Beautify

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

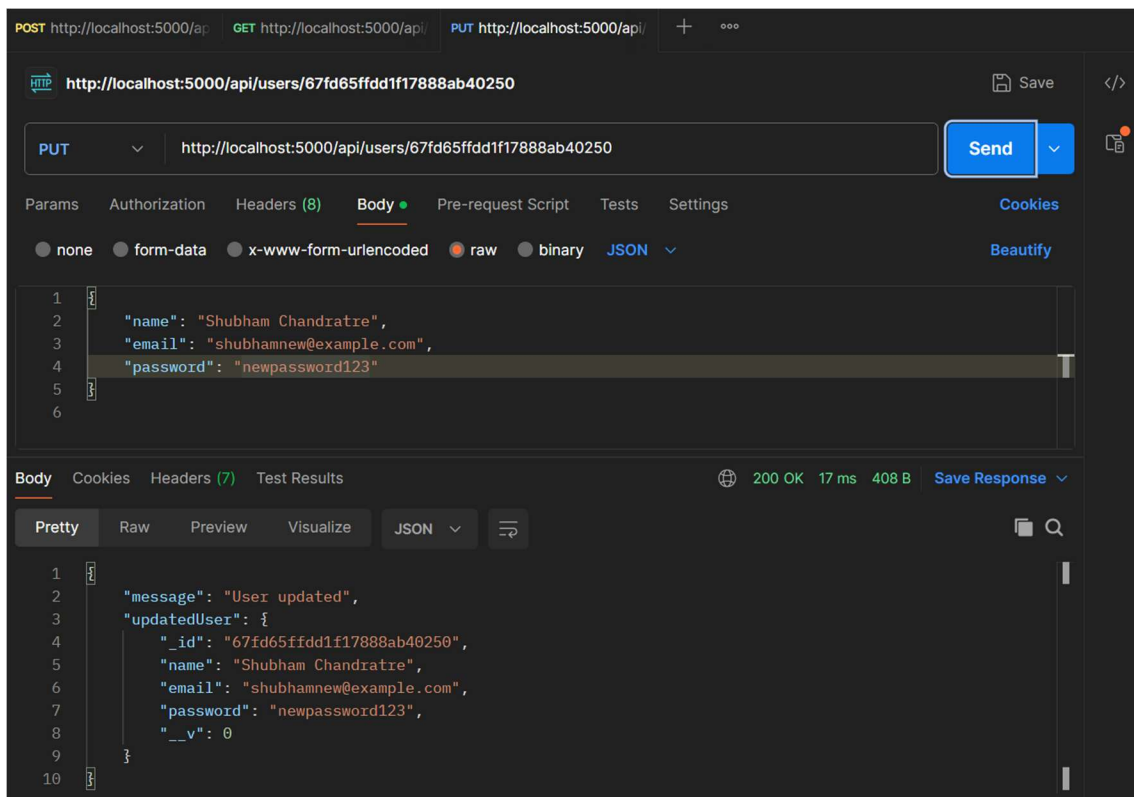
```
1 {
2   "email": "shubham@example.com"
3 }
4
```

Body Cookies Headers (7) Test Results 200 OK 11 ms 358 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "67fd65ffdd1f17888ab40250",
3   "name": "Shubham Chandratre",
4   "email": "shubham@example.com",
5   "password": "123456",
6   "__v": 0
7 }
8
9
```

4. update user



POST http://localhost:5000/api/ GET http://localhost:5000/api/ PUT http://localhost:5000/api/ + ...

HTTP http://localhost:5000/api/users/67fd65ffdd1f17888ab40250 Save </>

PUT http://localhost:5000/api/users/67fd65ffdd1f17888ab40250 Send

Params Authorization Headers (8) Body • Pre-request Script Tests Settings Cookies Beautify

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

```
1 {
2   "name": "Shubham Chandratre",
3   "email": "shubhamnew@example.com",
4   "password": "newpassword123"
5 }
6
```

Body Cookies Headers (7) Test Results 200 OK 17 ms 408 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "User updated",
3   "updatedUser": {
4     "_id": "67fd65ffdd1f17888ab40250",
5     "name": "Shubham Chandratre",
6     "email": "shubhamnew@example.com",
7     "password": "newpassword123",
8     "__v": 0
9   }
10 }
```

5. Delete User

The screenshot displays a REST client interface with a dark theme. At the top, a tab bar shows four requests: POST, GET, PUT, and the active DELETE request. The DELETE request is configured with the URL `http://localhost:5000/api/users/67fd65fdd1f17888ab40250` and the method `DELETE`. A `Send` button is visible next to the URL. Below the URL bar, tabs for `Params`, `Authorization`, `Headers (6)`, `Body`, `Pre-request Script`, `Tests`, and `Settings` are shown. The `Body` tab is active, showing a single empty line. Below the body, radio buttons for `none`, `form-data`, `x-www-form-urlencoded`, `raw` (selected), and `binary` are present, along with a `JSON` dropdown and a `Beautify` button. The bottom section shows the response, with tabs for `Body`, `Cookies`, `Headers (7)`, and `Test Results`. The `Body` tab is active, displaying a `200 OK` status, a response time of `17 ms`, and a size of `274 B`. The response body is shown in `JSON` format, displaying `"message": "User deleted successfully"`. The interface includes various utility icons like `Save`, `Send`, `Beautify`, and `Save Response`.

POST http://localhost:5000/api/ GET http://localhost:5000/api/ PUT http://localhost:5000/api/ DEL http://localhost:5000/api/ + ...

HTTP http://localhost:5000/api/users/67fd65fdd1f17888ab40250 Save </>

DELETE http://localhost:5000/api/users/67fd65fdd1f17888ab40250 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

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

1

Body Cookies Headers (7) Test Results 200 OK 17 ms 274 B Save Response

Pretty Raw Preview Visualize JSON

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