

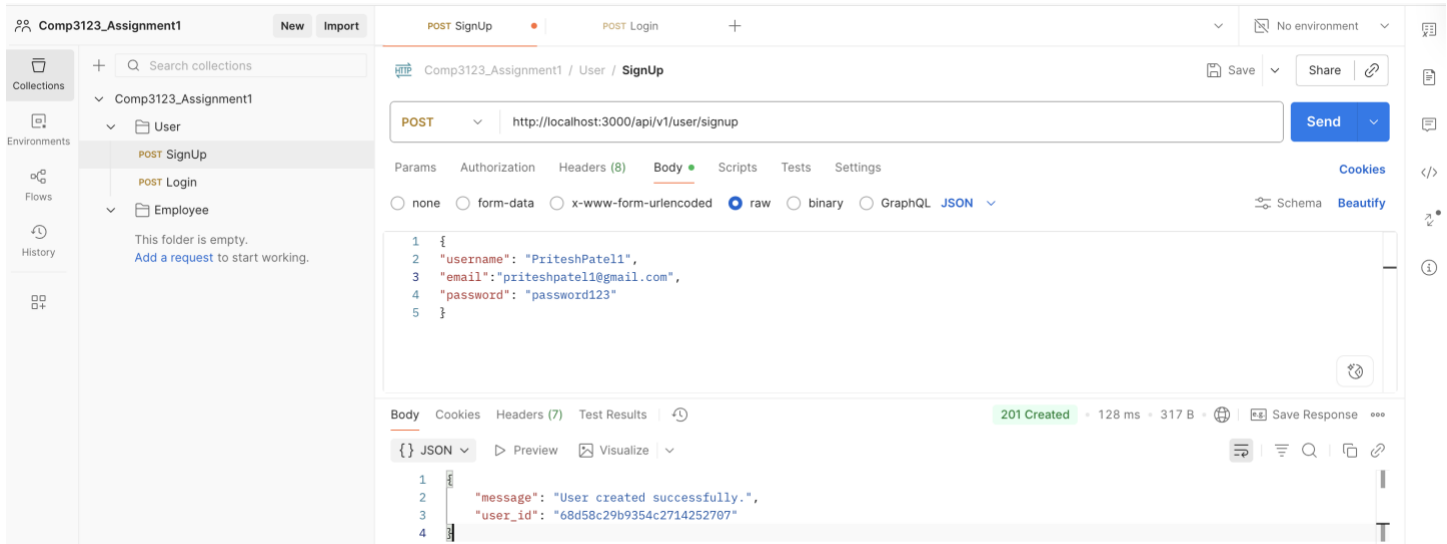
Name: Breno Lopes Mafra

Student ID: 101485572

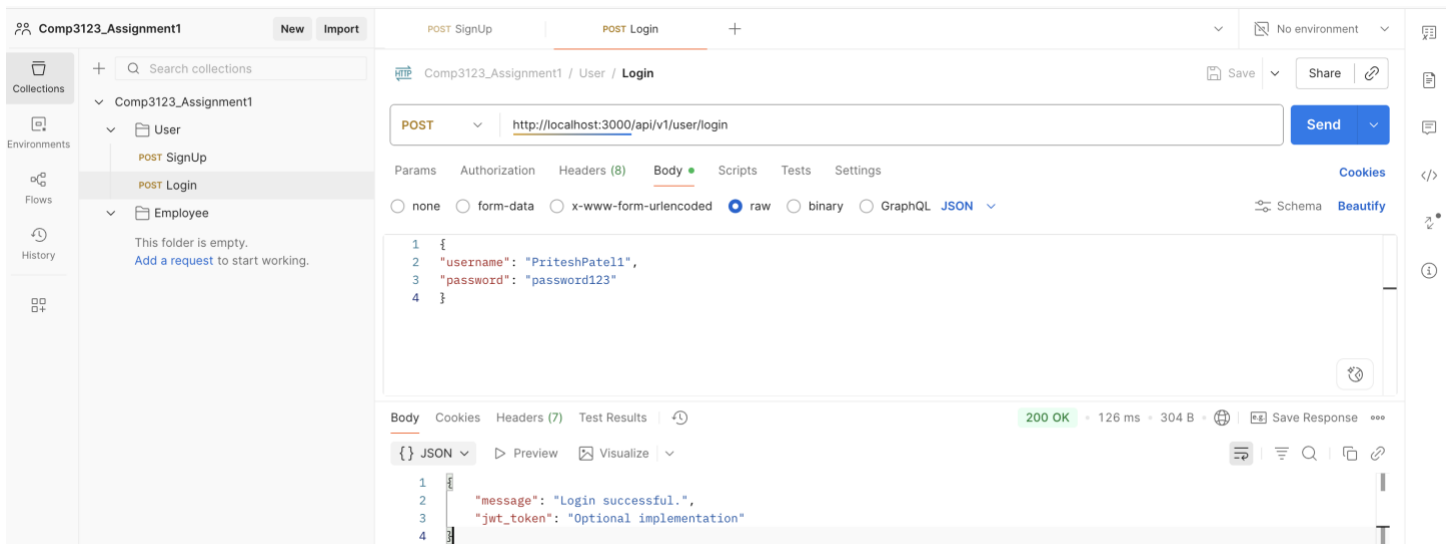
Assignment 1 – Screenshots

- **Postman tests for all endpoints + responses:**

1. <http://localhost:3000/api/v1/user/signup> (Create User)



2. <http://localhost:3000/api/v1/user/login> (Login with Username + Password)



3. <http://localhost:3000/api/v1/user/login> (Login with Email + Password)

The screenshot shows the Postman interface for a collection named 'Comp3123_Assignment1'. The 'User' folder is expanded, showing 'POST SignUp' and 'POST Login'. The 'POST Login' request is selected. The URL is `http://localhost:3000/api/v1/user/login`. The request body is a JSON object: `{ "email": "priteshpatel1@gmail.com", "password": "password123" }`. The response status is '200 OK' with a response time of 134 ms and a body size of 304 B. The response body is a JSON object: `{ "message": "Login successful.", "jwt_token": "Optional implementation" }`.

```
1 {
2   "email": "priteshpatel1@gmail.com",
3   "password": "password123"
4 }
```

```
1 {
2   "message": "Login successful.",
3   "jwt_token": "Optional implementation"
4 }
```

4. <http://localhost:3000/api/v1/emp/employees> (Create Employee)

The screenshot shows the Postman interface for a collection named 'Comp3123_Assignment1'. The 'Employee' folder is expanded, showing 'GET List Employees' and 'POST Create Employee'. The 'POST Create Employee' request is selected. The URL is `http://localhost:3000/api/v1/emp/employees`. The request body is a JSON object: `{ "first_name": "Alice", "last_name": "Johnson", "email": "alice.johnson@example.com", "position": "Designer", "salary": 85000, "date_of_joining": "2023-08-10T00:00:00.000Z", "department": "Design" }`. The response status is '201 Created' with a response time of 64 ms and a body size of 325 B. The response body is a JSON object: `{ "message": "Employee created successfully.", "employee_id": "68d58dc2b9354c271425270c" }`.

```
1 {
2   "first_name": "Alice",
3   "last_name": "Johnson",
4   "email": "alice.johnson@example.com",
5   "position": "Designer",
6   "salary": 85000,
7   "date_of_joining": "2023-08-10T00:00:00.000Z",
8   "department": "Design"
9 }
```

```
1 {
2   "message": "Employee created successfully.",
3   "employee_id": "68d58dc2b9354c271425270c"
4 }
```

5. <http://localhost:3000/api/v1/emp/employees> (List Employees)

The screenshot shows the Postman interface for a collection named 'Comp3123_Assignment1'. The 'Employee' folder is expanded, showing 'GET List Employees' and 'POST Create Employee'. The 'GET List Employees' request is selected. The URL is `http://localhost:3000/api/v1/emp/employees`. The response status is '200 OK' with a response time of 9 ms and a body size of 463 B. The response body is a JSON array: `[{ "employee_id": "68d58dc2b9354c271425270c", "first_name": "Alice", "last_name": "Johnson", "email": "alice.johnson@example.com", "position": "Designer", "salary": 85000, "date_of_joining": "2023-08-10T00:00:00.000Z", "department": "Design" }]`.

```
1 [
2   {
3     "employee_id": "68d58dc2b9354c271425270c",
4     "first_name": "Alice",
5     "last_name": "Johnson",
6     "email": "alice.johnson@example.com",
7     "position": "Designer",
8     "salary": 85000,
9     "date_of_joining": "2023-08-10T00:00:00.000Z",
10    "department": "Design"
11  }
12 ]
```

6. <http://localhost:3000/api/v1/emp/employees/{employeeid}> (Get one employee by ID)

The screenshot shows the Postman interface for a collection named 'Comp3123_Assignment1'. The 'GET Get employee by ID' request is selected. The URL is `http://localhost:3000/api/v1/emp/employees/68d58dc2b9354c271425270c`. The response is a 200 OK status with a JSON body containing employee details.

Request:

- Method: GET
- URL: `http://localhost:3000/api/v1/emp/employees/68d58dc2b9354c271425270c`

Response:

```
1 {
2   "employee_id": "68d58dc2b9354c271425270c",
3   "first_name": "Alice",
4   "last_name": "Johnson",
5   "email": "alice.johnson@example.com",
6   "position": "Designer",
7   "salary": 85000,
8   "date_of_joining": "2023-08-10T00:00:00.000Z",
9   "department": "Design"
10 }
```

7. <http://localhost:3000/api/v1/emp/employees/{employeeid}> (Update one employee by ID)

The screenshot shows the Postman interface for a collection named 'Comp3123_Assignment1'. The 'PUT New Request' is selected. The URL is `http://localhost:3000/api/v1/emp/employees/68d58dc2b9354c271425270c`. The request body is a JSON object with updated employee details. The response is a 200 OK status with a message.

Request:

- Method: PUT
- URL: `http://localhost:3000/api/v1/emp/employees/68d58dc2b9354c271425270c`
- Body: raw

Response:

```
1 {
2   "employee_id": "68d58dc2b9354c271425270c",
3   "first_name": "Alice",
4   "last_name": "Johnson",
5   "email": "alice.johnson@example.com",
6   "position": "Manager",
7   "salary": 85000,
8   "date_of_joining": "2023-08-10T00:00:00.000Z",
9   "department": "Management"
10 }
```

Response Body:

```
1 {
2   "message": "Employee details updated successfully."
3 }
```

8. <http://localhost:3000/api/v1/emp/employees?employeeid={employeeid}> (Delete employee ID)

The screenshot shows the Postman interface for a collection named 'Comp3123_Assignment1'. The 'DEL Delete employee by ID' request is selected. The URL is `http://localhost:3000/api/v1/emp/employees?employeeid=68d58dc2b9354c271425270c`. The response is a 204 No Content status.

Request:

- Method: DELETE
- URL: `http://localhost:3000/api/v1/emp/employees?employeeid=68d58dc2b9354c271425270c`

Response:

```
1
```

- MongoDB showing the database and collections:

→ ↺ ⚠ Not Secure 0.0.0.0:8081 ☆ 📦 | Work Relaunch to update ⋮

Mongo Express Database ▾

Mongo Express

Databases

Database Name + Create Database

| | | |
|-----------------|----------------------|----------------|
| <div>View</div> | admin | <div>Del</div> |
| <div>View</div> | comp3123_assignment1 | <div>Del</div> |
| <div>View</div> | config | <div>Del</div> |
| <div>View</div> | local | <div>Del</div> |

Server Status

| | | | |
|---------------------|-------------------------------|-----------------------|---------------------|
| Hostname | Brenos-Air | MongoDB Version | 7.0.23 |
| Uptime | 1111142 seconds (12 days) | Node Version | 18.20.3 |
| Server Time | Thu, 25 Sep 2025 19:07:50 GMT | V8 Version | 10.2.154.26-node.37 |
| | | | |
| Current Connections | 6 | Available Connections | 51194 |
| Active Clients | 0 | Queued Operations | 0 |
| Clients Reading | 0 | Clients Writing | 0 |
| Read Lock Queue | 0 | Write Lock Queue | 0 |
| | | | |
| Disk Flushes | | Last Flush | |
| Time Spent Flushing | ms | Average Flush Time | ms |

Mongo Express Database: comp3123_assignment1 ▾

Viewing Database: comp3123_assignment1

Collections

Collection Name + Create collection

| | | | | | |
|-----------------|-------------------|-------------------|-------------------|-----------|----------------|
| <div>View</div> | <div>Export</div> | <div>[JSON]</div> | <div>Import</div> | employees | <div>Del</div> |
| <div>View</div> | <div>Export</div> | <div>[JSON]</div> | <div>Import</div> | users | <div>Del</div> |

Database Stats

| | |
|---------------------------------------|-----------|
| Collections (incl. system.namespaces) | 2 |
| Data Size | 629 Bytes |
| Storage Size | 73.7 KB |
| Avg Obj Size # | 210 Bytes |
| Objects # | 3 |
| Indexes # | 4 |
| Index Size | 123 KB |

Viewing Collection: employees

[New Document](#)[New Index](#)[Simple](#)[Advanced](#)[Delete all 1 documents retrieved](#)

| _id | first_name | last_name | email | position | salary | date |
|-------------------------------------------------------------------------|------------|-----------|---------------------------|----------|--------|----------------------------------------------------------------|
| Link Delete 68d593c7b9354c2714252719 | Alice | Johnson | alice.johnson@example.com | Designer | 85000 | Thu Jan 01 2020 00:00:00 GMT+0000 (Coordinated Universal Time) |

Viewing Collection: users

[New Document](#)[New Index](#)[Simple](#)[Advanced](#)[Delete all 3 documents retrieved](#)

| _id | username | email | password |
|-------------------------------------------------------------------------|-----------------|---------------------------|-----------------------------|
| Link Delete 68d58491b9354c27142526fa | brenolopesmafra | brenolopesmafra@gmail.com | \$2b\$10\$j7wv6Nd4aDaFHlVTE |
| Link Delete 68d58727b9354c27142526fc | PriteshPatel | priteshpatel@gmail.com | \$2b\$10\$z1Jp5Ukvz.Z8xQ8LY |