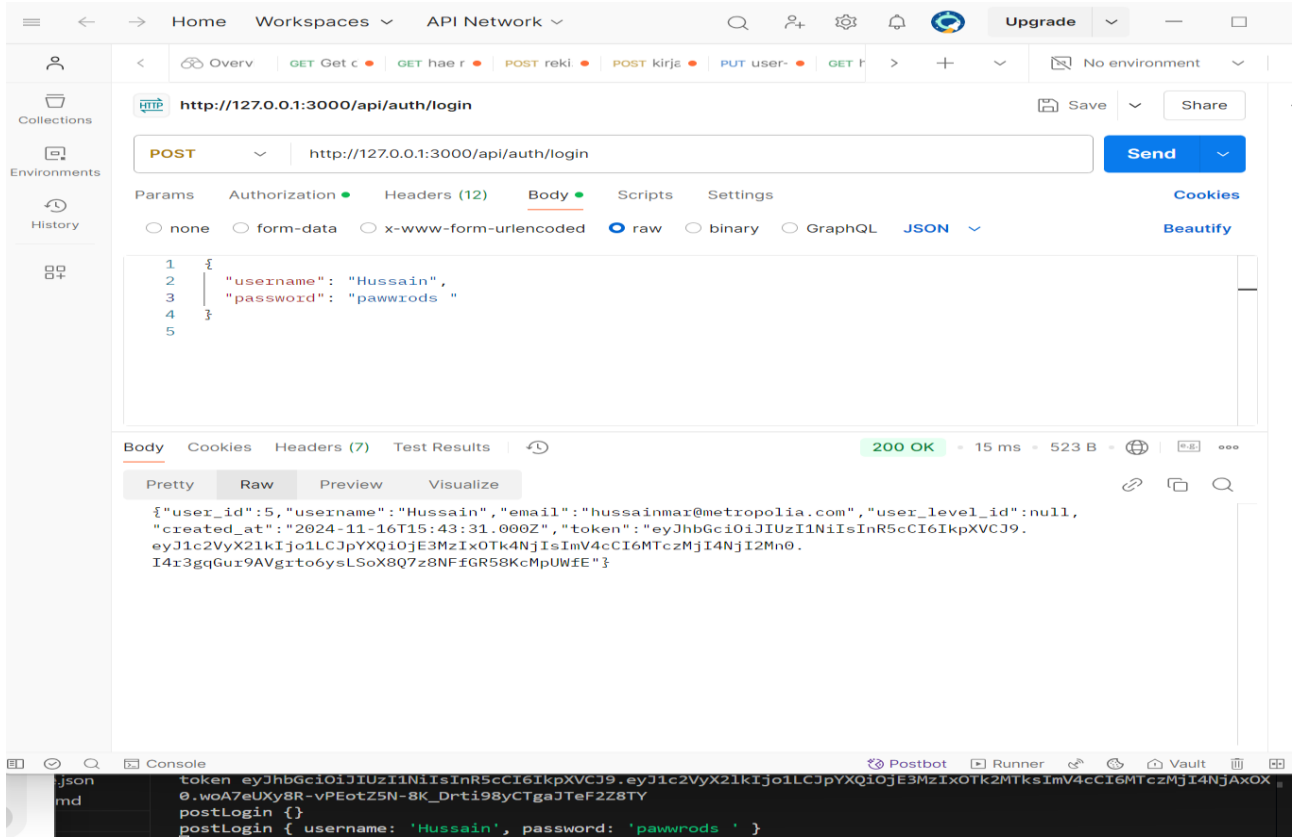
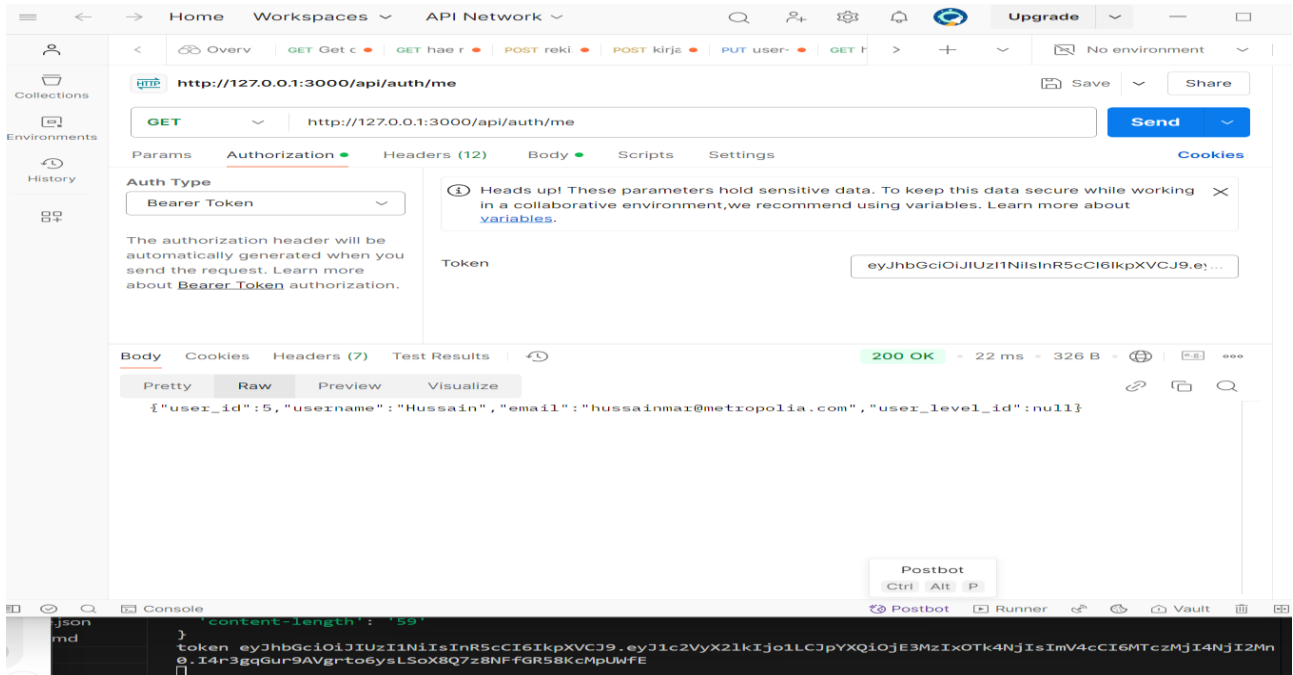


User Authentication

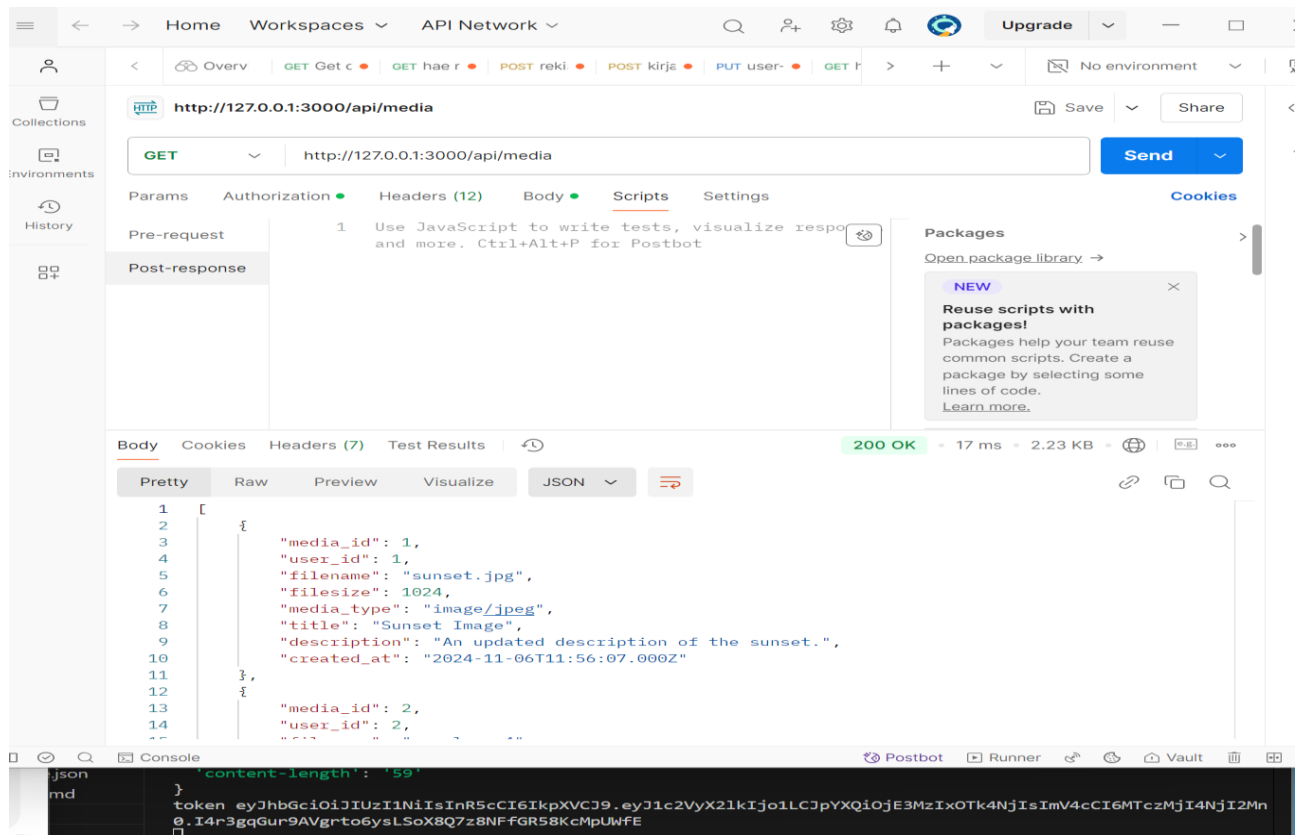
POST: <http://localhost:3000/api/auth/login>



GET: <http://localhost:3000/api/auth/me>



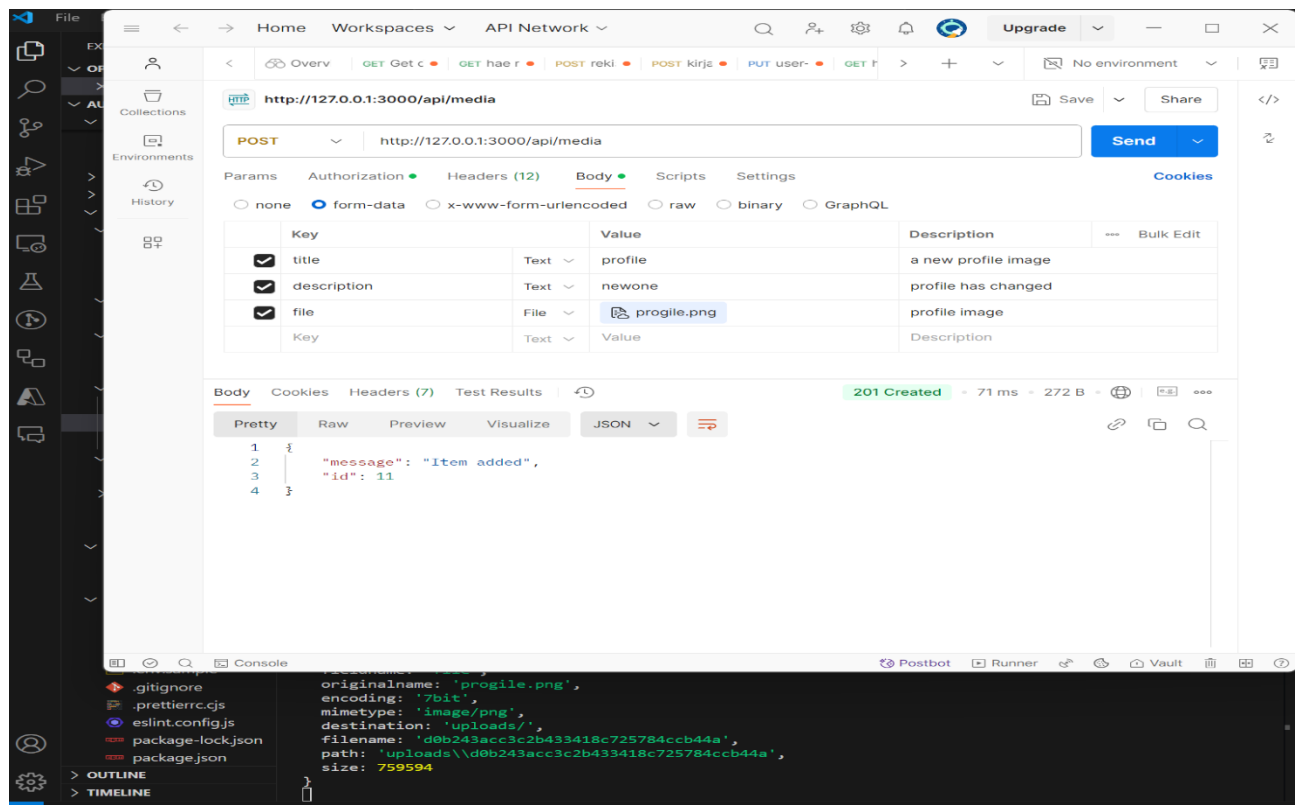
GET: <http://localhost:3000/api/media>



Postman interface showing a GET request to `http://127.0.0.1:3000/api/media`. The response is a 200 OK status with a 17 ms response time and 2.23 KB of data. The body is a JSON array of two media objects.

```
1 [
2   {
3     "media_id": 1,
4     "user_id": 1,
5     "filename": "sunset.jpg",
6     "filesize": 1024,
7     "media_type": "image/jpeg",
8     "title": "Sunset Image",
9     "description": "An updated description of the sunset.",
10    "created_at": "2024-11-06T11:56:07.000Z"
11  },
12  {
13    "media_id": 2,
14    "user_id": 2,
15    "filename": "profile.png",
16    "filesize": 759594,
17    "media_type": "image/png",
18    "title": "Profile Image",
19    "description": "A new profile image",
20    "created_at": "2024-11-06T11:56:07.000Z"
21  }
22 ]
```

POST: <http://localhost:3000/api/media>



Postman interface showing a POST request to `http://127.0.0.1:3000/api/media`. The response is a 201 Created status with a 71 ms response time and 272 B of data. The body is a JSON object with a message and an id.

Key	Value	Description
title	profile	a new profile image
description	newone	profile has changed
file	profile.png	profile image
Key	Value	Description

```
1 {
2   "message": "Item added",
3   "id": 11
4 }
```

The screenshot displays the Postman interface for an API client. At the top, the URL bar shows `http://127.0.0.1:3000/api/media/11`. The request method is set to **PUT**. The request body is a JSON object:

```
{
  "title": "A new profile",
  "description": "again profile has changed"
}
```

The response status is **200 OK** with a response time of 39 ms and a size of 271 B. The response body is a JSON object:

```
{
  "message": "Item updated",
  "id": "11"
}
```

The bottom of the image shows a terminal window with the following output:

```
insertId: 0,
info: 'Rows matched: 1 Changed: 1 Warnings: 0',
serverStatus: 2,
warningStatus: 0,
changedRows: 1
},
undefined
```

The screenshot displays the Postman application interface. At the top, there's a navigation bar with tabs for Home, Workspaces, and API Collection. Below this, a sidebar on the left contains icons for Collections, Environments, History, and Settings. The main workspace is divided into several sections. The top section shows the request details: a DELETE request to the URL `http://127.0.0.1:3000/api/media/11`. Below the URL bar, there's a section for Params, Authorization (set to Bearer Token), Headers (12), Body, Scripts, and Settings. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables." The Token field shows `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjo1LCJpYXQiOjE3MzIxOTk4NjI1ImV4cCI6MTczMjI4NjI2Mn0.IGR3ggGur9AVgrto6ysLSOX8Q7z8NfFGR58KcMpUWFE`. The bottom section shows the response details: a 200 OK status, 34 ms response time, and a JSON body: `{ "message": "Item deleted", "id": "11" }`. The bottom of the screen shows a file explorer on the left with a list of files including `.env`, `.env.sample`, `.gitignore`, `.prettierrc.cjs`, `eslint.config.js`, `package-lock.json`, and `package.json`. The bottom right corner shows a console with the command `curl -XDELETE http://127.0.0.1:3000/api/media/11` and its output.

PUT: <http://127.0.0.1:3000/api/users>

The screenshot displays the Postman application interface. At the top, the 'API Network' tab is active, showing a list of requests. The selected request is a PUT to `http://127.0.0.1:3000/api/users`. The request body is a JSON object:

```
1 {
2   "username": "UpdatedUserName",
3   "email": "updated_email@example.com"
4 }
```

The response is a 200 OK status with a 70 ms response time and 273 B of data. The response body is a JSON object:

```
1 {
2   "message": "User information updated"
3 }
```

The bottom panel shows the console output, which includes the request headers and the response body. The headers are:

```
authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjo1LCJpYXQiOjE3MzIxOTk4NjIsImV4cCI6MTczMjI4NjI2Mn0.I4r3gqGur9AVgrto6ysLSOX8Q7z8NFfGR58KcMpUwFE',
'user-agent': 'PostmanRuntime/7.42.0',
accept: '*/*',
'postman-token': 'd40b25b4-2c3a-4a5a-94a3-f341e4d8bc92',
host: '127.0.0.1:3000',
'accept-encoding': 'gzip, deflate, br',
connection: 'keep-alive',
'content-length': '84'
```

The response body is:

```
token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjo1LCJpYXQiOjE3MzIxOTk4NjIsImV4cCI6MTczMjI4NjI2Mn0.I4r3gqGur9AVgrto6ysLSOX8Q7z8NFfGR58KcMpUwFE
{
  "message": "User information updated"
}
```