**Week 2:**

Enhance the API with authentication and logging.

Deliver:

* User authentication feature using JWT.
* Logging functionality.
* Exception handling for invalid requests or errors.
* Execution screenshot along with logs.

Code Repository: <https://github.com/Akriti987/Weekly_Tasks.git>

**Execution Documentation for Node.js To-Do Application**

This document outlines the execution steps and functionalities of the provided Node.js application for managing tasks.

**Prerequisites:**

* Node.js and npm installed on your system.
* A code editor or IDE for editing JavaScript code. (VS Code)
* **Key Features:**
  + CRUD operations for tasks.
  + User authentication with JWT.
  + Data validation and error handling.
  + Logging functionality for users.

**1. Setting Up the Project:**

1. Clone the repository.
2. Install dependencies: run ‘**npm install’** to install the required dependencies listed in the ‘**package.json’** file.

**2.** Database Used **(MongoDB)**

**3. Running the Application:**

Start the server by **node server.js** in the terminal inside the **/Week2/src** folder. This will start the Node.js on the specified port (default: 3000).

**4. API Endpoints:**

The application provides several API endpoints for managing tasks:

* **GET /tasks:** Retrieves a list of all tasks stored in the database.
* **GET /tasks/:id:** Retrieves a single task by its ID.
* **POST /tasks:** Creates a new task. The request body should contain **title** and **description** properties for the new task. (middleware validation)
* **PUT /tasks/:id:** Updates an existing task by its ID. The request body should contain the updated properties for the task.
* **DELETE /tasks/:id:** Deletes a task by its ID.
* **POST /signup:** Registers the user by their email-Id and password.
* **POST /login:** Logs in the user after verifying email-Id and password.

By following these steps and understanding the functionalities, you can effectively use this Node.js application to manage your tasks through API requests.

**Execution Screenshots:**

1. Running the App

A computer screen with white text

Description automatically generated

1. Requests (POST, GET, PUT, DELETE)

**1.POST /signup:**

A screenshot of a computer

Description automatically generated

**2.POST /login:**

A screenshot of a computer

Description automatically generated

**3.POST /tasks: (CREATE)**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Authorization token used while creating a task

**4. GET /tasks/id:** (**READ**)

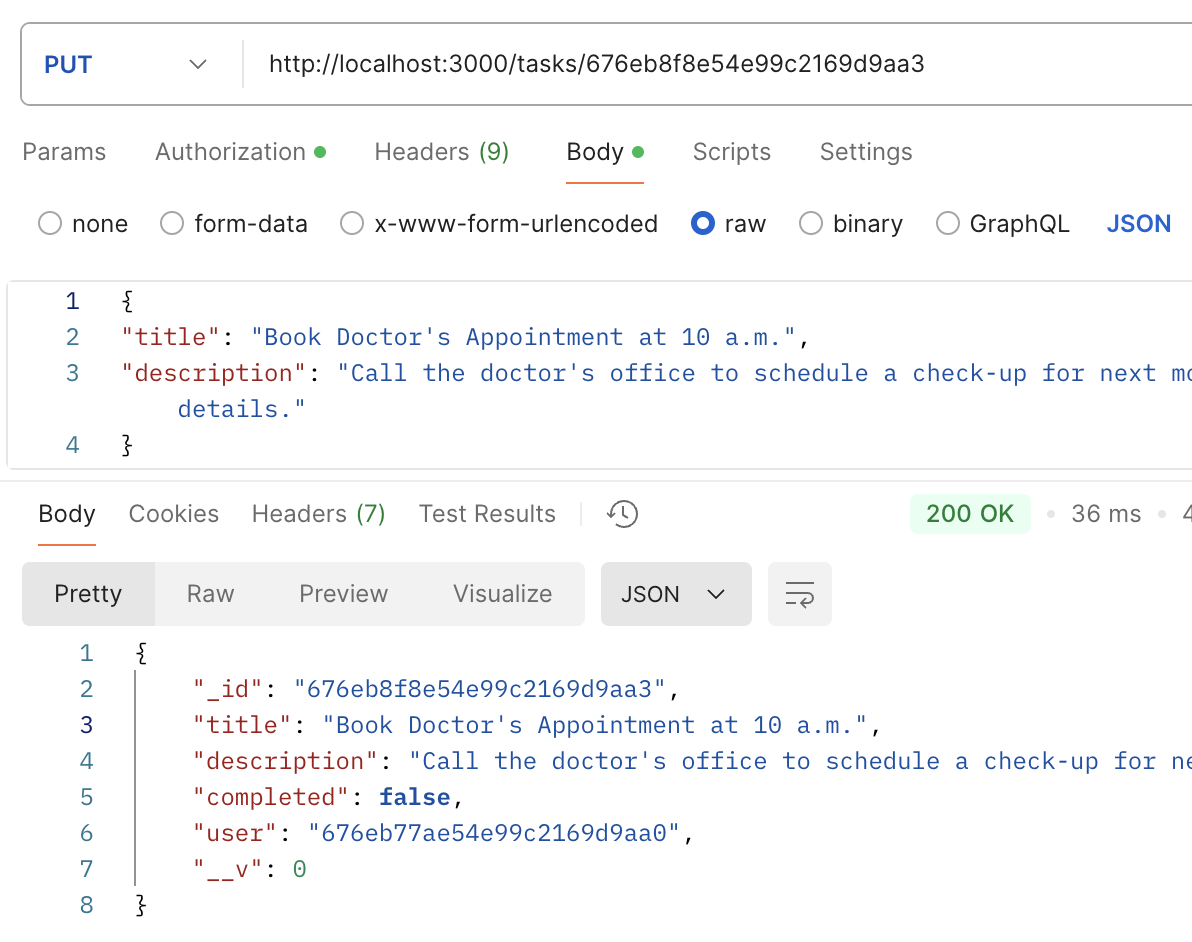
Id of specific task of the user

A screenshot of a computer

Description automatically generated

**5.PUT /tasks/id: (UPDATE)**

Id of specific task of the user

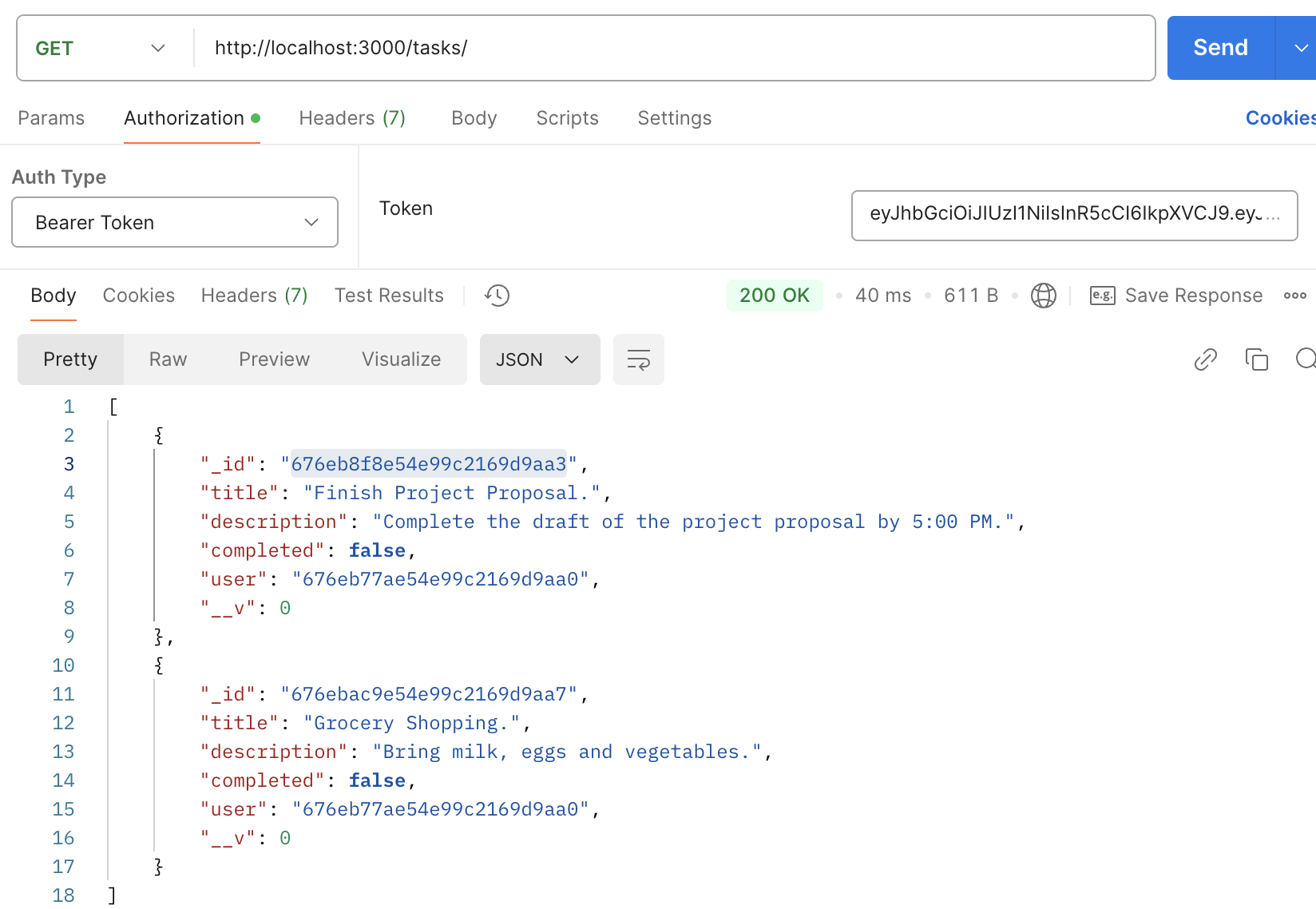


**6. DELETE /tasks/id: (DELETE)**

A screenshot of a computer

Description automatically generated

**7.GET /tasks: (GET ALL TASKS )**



3.Creation of entries in Database

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

4.Code Snippets

A screen shot of a computer program

Description automatically generated

A computer screen with text on it

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated