Assignment

Subject : Service Oriented computing

Reg No: 2022mt93314

Name: Bramhanand G S

Topic: Idea Log

**Introduction**

Idea Log is a versatile note-taking app designed for digital users. With a user-friendly interface, it allows you to effortlessly write your thoughts and ideas. It enhances organization and accessibility of information, boosting productivity and reducing the need for physical paper, making them indispensable in today's digital world.

**Description of Application**

The Idea Log application has the following components.

* **Entities**
  + Users
  + Ideas
* **Features**
  + Registration of users
  + Updating of user details
  + Log new ideas
  + Update ideas
  + Delete ideas

**Design and Architecture, system operations, and services identified**



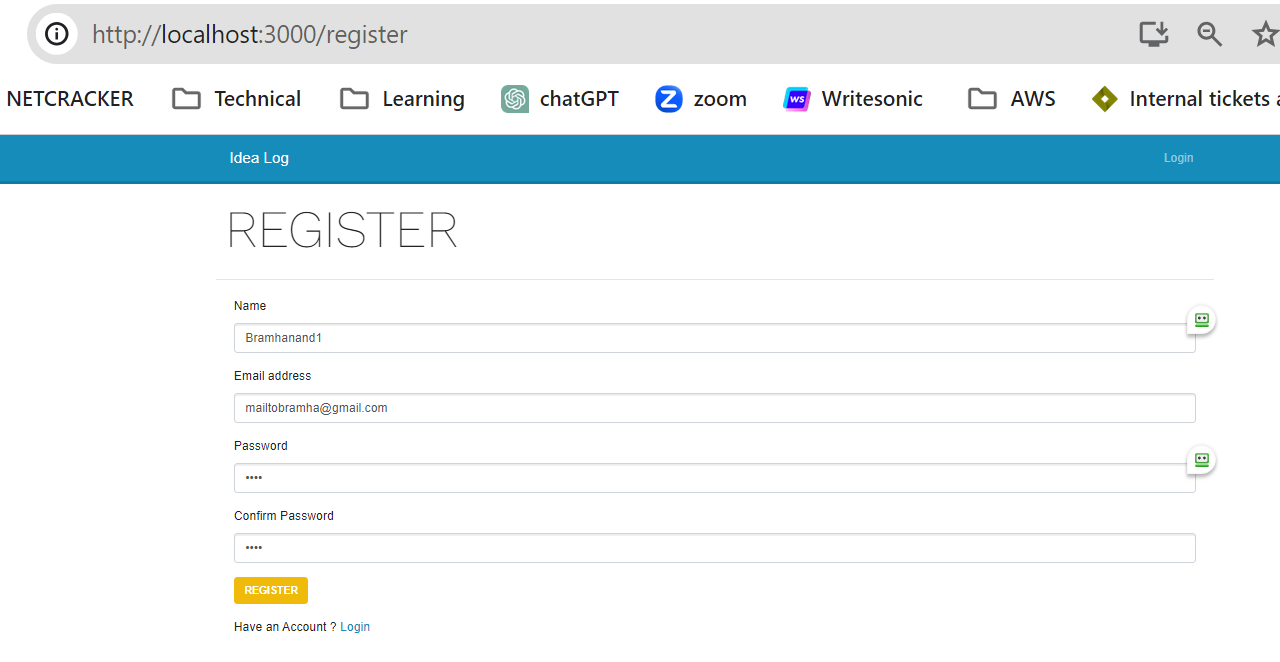
The Design and Architecture of Idea Log is as follows:

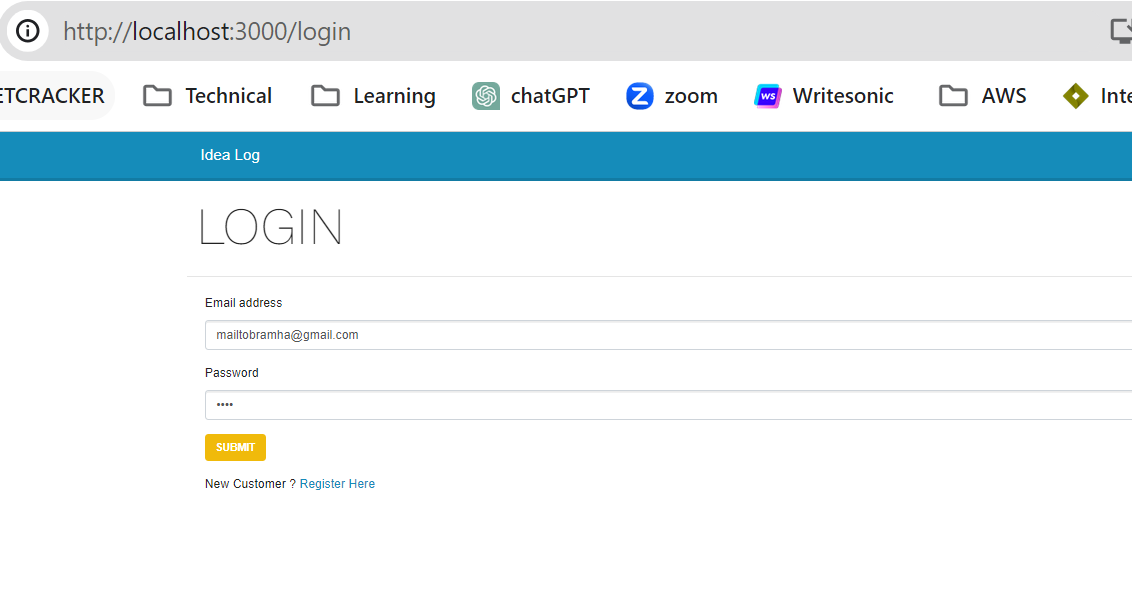
**Frontend:**

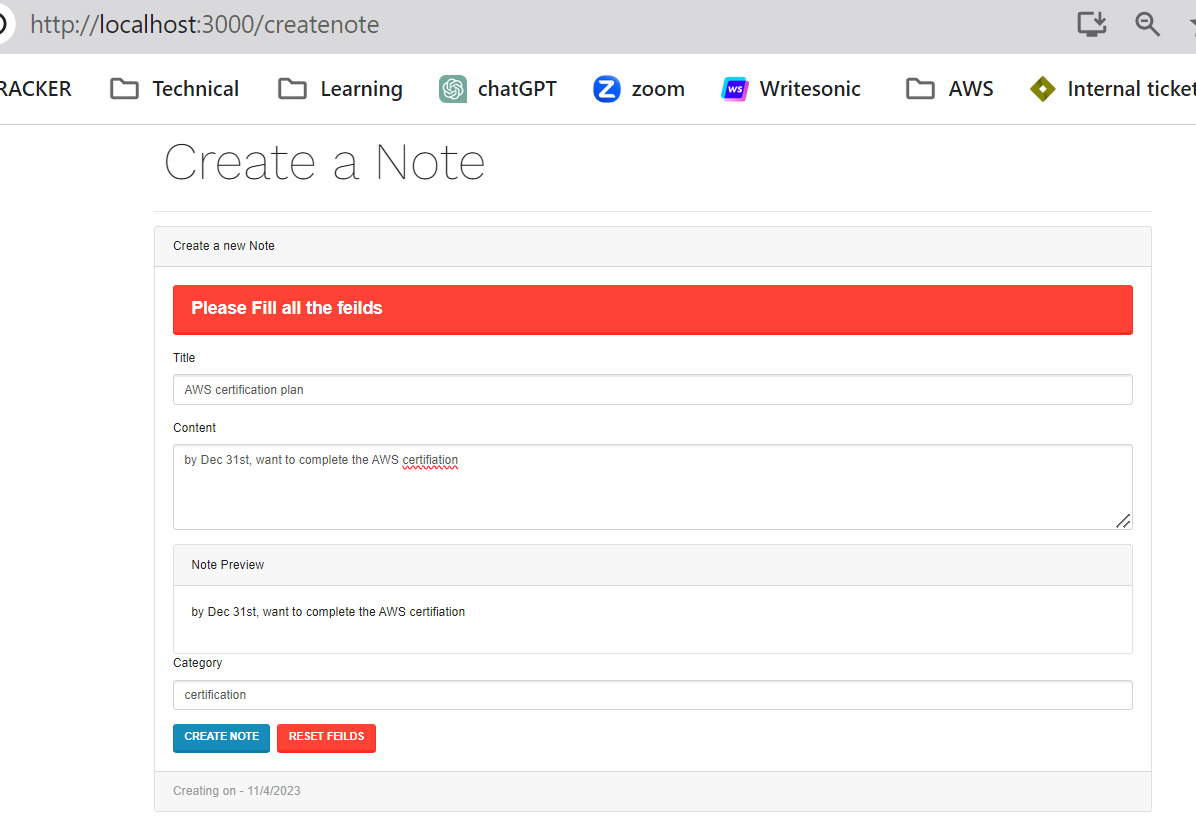
The frontend of the application will be responsible for providing a user-friendly interface to users to log their ideas.

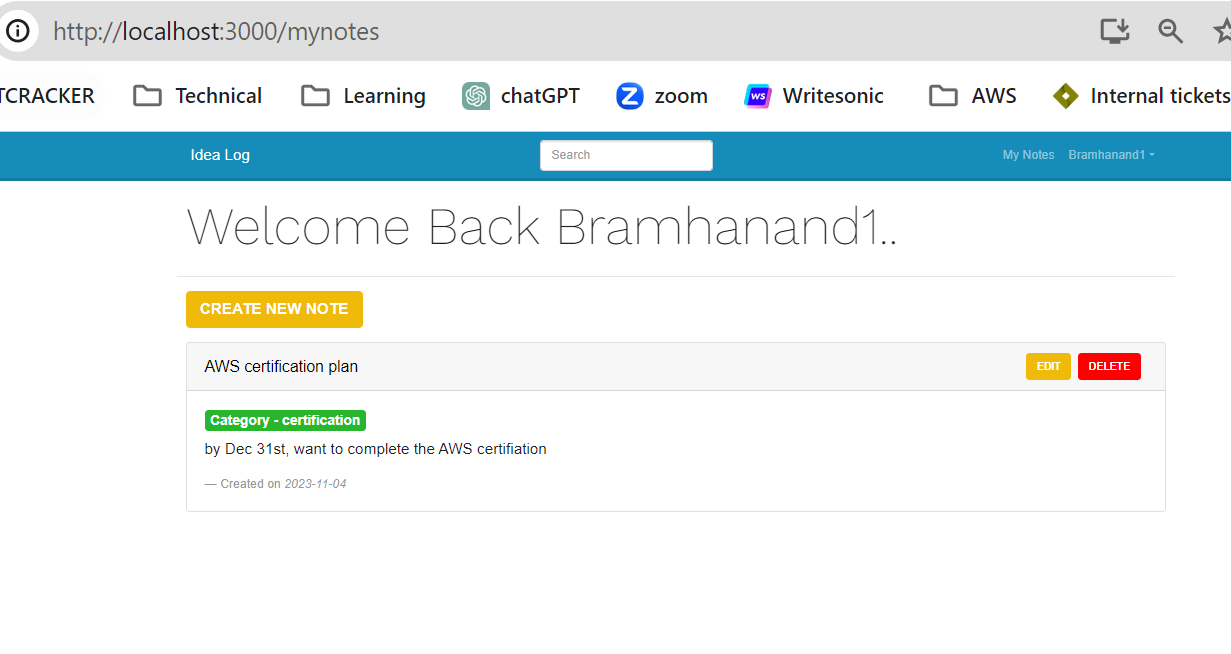
The frontend will communicate with the backend server through RESTful APIs to perform actions like adding notes, updating notes and deleting notes.

The UI is responsive, allowing users to access the application from both desktop and mobile devices.









**Backend:**

The backend will handle all the business logic, process requests from the frontend and interact with the database.

The backend exposes RESTful APIs to be consumed by the frontend and any other potential clients.

**Database:**

The application's data is stored in the MongoDB database.

It is used to store the user’s details and ideas.

**Authentication:**

JWT (JSON Web Token) authentication is used for user’s authentication. It is a secure and efficient method implemented in my project to verify user identity. It enables users to access resources with a token that contains their identity information, enhancing data protection and authorization while simplifying user management. JWT's self-contained structure and digital signatures make it a trusted choice for securing access to the project's features and data.

**Notes Management:**

Implemented a search functionality that allows users to search their ideas.

Provided an user friendly interface for users to add, update or delete their ideas.

**REST API endpoints with URI and HTTP methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial No** | **Entity** | **Http Method** | **URI** |
| 1 | User | POST | **/users** |
| POST | **/users/login** |
| PUT | **/users/profile** |

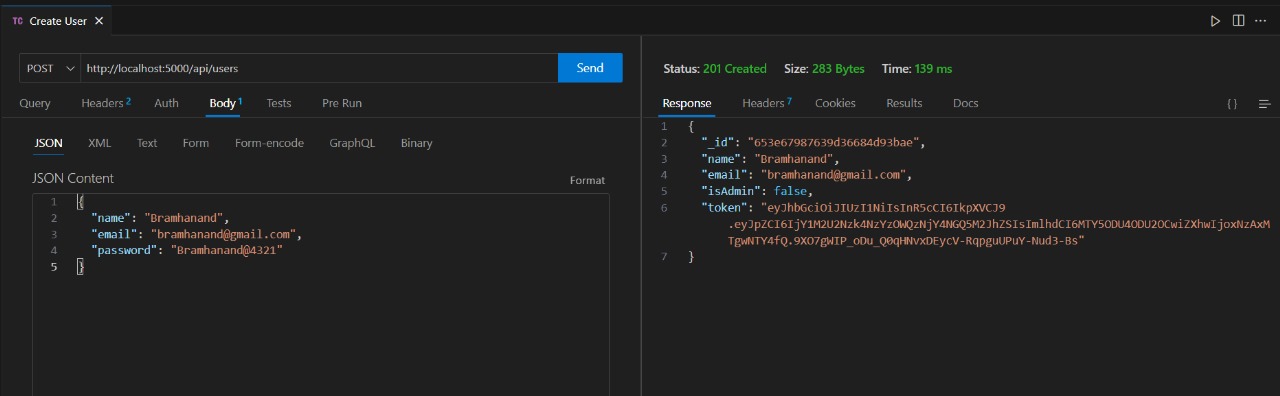
|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Ideas | POST | **/notes/create** |
| GET | **/notes** |
| PUT | **/notes/:id** |
| DELETE | **/notes/:id** |

**Representation design in request and response**

Sample Request and Response header for the above-mentioned HTTP method:

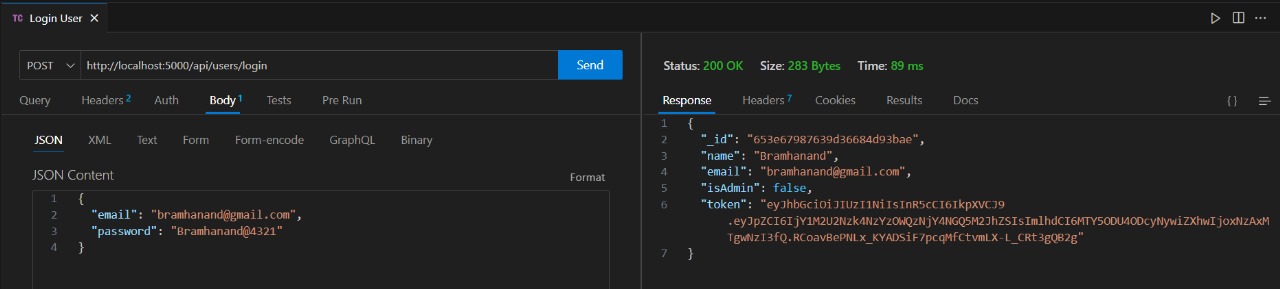
1. **POST /users**

* Request:
  + **Method: POST**
  + **URI: /users**
* Response:
  + **Status Code: 201 CREATED**
  + **Headers: Content-Type: application/json**
  + **Body:**



1. **POST /users/login**

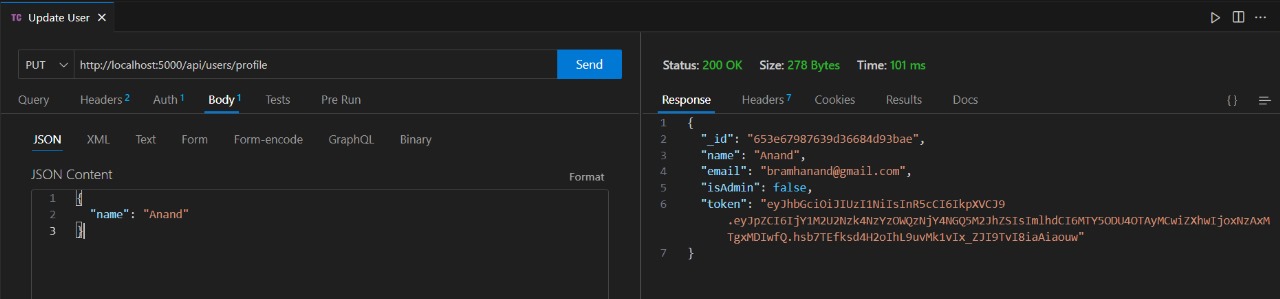
* Request:
  + **Method: POST**
  + **URI: /users/login**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json**
  + **Body:**



1. **PUT /users/profile:**

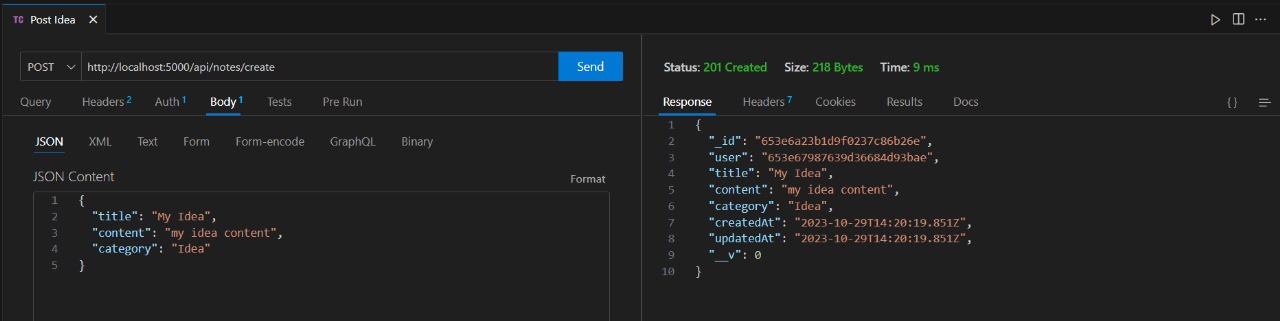
* Request:
  + **Method: PUT**
  + **URI: /users/profile**
* Response:
  + **Status Code: 201 CREATED**
  + **Headers: Content-Type: application/json, Authorisation: Bearer Token**

* **Body:**



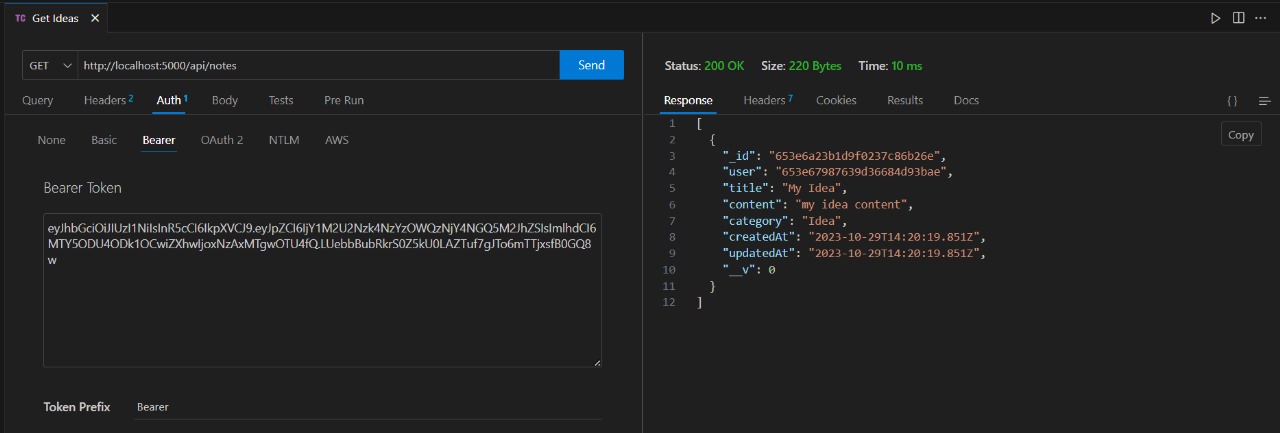
1. **POST /notes/create:**

* Request:
  + **Method: POST**
  + **URI: /notes/create**
* Response:
  + **Status Code: 201 CREATED**
  + **Headers: Content-Type: application/json, Authorisation: Bearer Token**
  + **Body:**



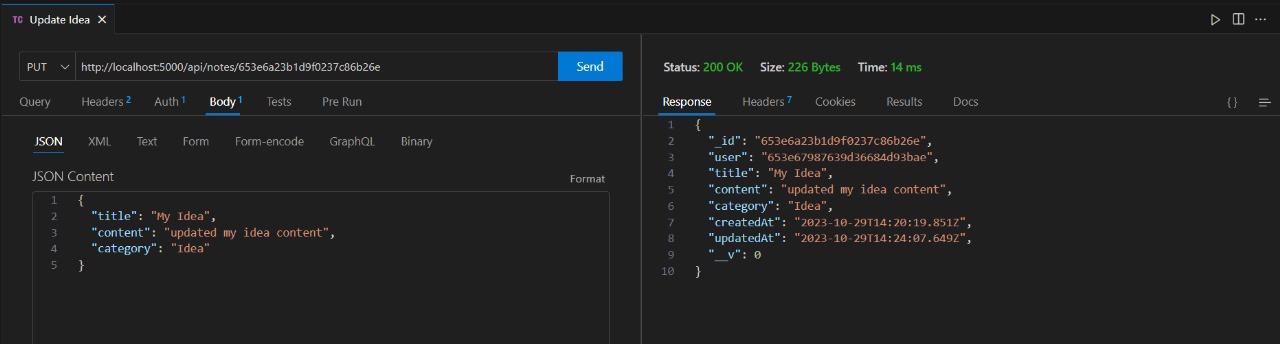
1. **GET /notes:**

* Request:
  + **Method: GET**
  + **URI: /notes**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json, Authorisation: Bearer Token**
  + **Body:**

****

1. **PUT /notes/:id :**

* Request:
  + **Method: PUT**
  + **URI: /notes/: id**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json, Authorisation: Bearer Token**
  + **Body:**

****

1. **DELETE /notes/:id :**

* Request:
  + **Method: DELETE**
  + **URI: /notes/:id**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json, Authorisation: Bearer Token**
  + **Body:**

