**PROJECT TITLE**

**Student Partner**

**Digital alternative to study desks**

**Submitted By**

Guntha Rohith

**Under the guidance of**

K Naga Samurai

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **1.** | **Introduction** | **2** |
| 1.1 | Abstract | **2** |
| 1.2 | Aim | **2** |
| **2.** | **Technology Stack** | **3** |
| **3.** | **Features of the Application** | **4-5** |
| 3.1 | Authenticated User | **4** |
| 3.2 | Authenticated Admin | **5** |
| **4.** | **Core concepts and Libraries Used in Angular** | **5-6** |
| 4.1 | Core concepts used | **5** |
| 4.2 | Libraries used | **6** |
| **5.** | **Project Learning and Outcome** | **6** |
| **6.** | **Project Architecture Diagram** | **7** |
| **7.** | **Component Level Architecture Diagram** | **8** |
| **8** | **Security Architecture Diagram** | **9** |
| **9.** | **Front End Pages** | **10-17** |
| **10.** | **Api testing using Postman** | **18-25** |
| **11.** | **Back End Walk Through** | **26-32** |
| **12.** | **References** | **32** |

**1.Introduction**

**1.1 Abstract**

There is a person who want to learn Data structures and Algorithms.But he is often unorganised about what to learn today! Whenever he starts reading up on a topic, he’ll find dozens of resources making it difficult to choose the right resource to start learning. And he finds it troublesome to track his progress. He wants to divide the syllabus into various topics (like Lists, Trees , Graphs etc.) and gather the resources for each topic in one place. He plans to use flashcards and notes to keep a check on his learning. He also prefers discussing topics and doubts with peers. He is looking for groups and apps to help her find people in the same path as his to study together. In this project I am try to solve these problems.

**1.2 Aim**

In this project , I am going to build a self-learning platform targeted for students and career professionals. This platform will act like a digital alternative to traditional study desks, with all learning equipment in one place.

**1.3 Objective**

The objectives in building this project are to understand the fundamental concepts and usage of each technology in the MEAN stack, as well as their compatibilities and advantages as a complete stack in web application development.

**2. Technology Stack-MEAN**

**M-MONGO**

MongoDB is an open source NoSQL database management program. MongoDB is a tool that can manage document-oriented information, store or retrieve information. With its flexible schema approach, it's popular with development teams using agile methodologies.

**E-EXPRESS**

Express.js is a free and open-source web application framework for Node.js. It is used for designing and building web applications quickly and easily. Express provides methods to specify what function is called for a particular HTTP verb ( GET , POST , SET , etc.) and URL pattern ("Route"), and methods to specify what template ("view") engine is used, where template files are located, and what template to use to render a response.

**A-ANGULAR**

A component-based framework for building scalable web applications.Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

**N-NODEJS**

Node. js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node. js is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind.

**3.Features of the Application**

**3.1 Authenticated User:**

1. Can login to their account using Login option.
2. Can view the list of learning paths available.
3. Can create a new learning path.
4. Can navigate to the detailed learning path.
5. Can view the list of resources.
6. Can create a resource.
7. Can create a notes.
8. Can edit a notes.
9. Can delete a notes.
10. Can view the list of available notes.
11. Can view the profile.
12. Can edit the profile.
13. Can change the password.
14. Can reset the password.
15. Can upload a profile photo.

**3.2 Authenticated Admin**

1. Can login to their account using Login option.
2. Can view the dashboard with the analytics.
3. Can view all the users.
4. Can edit and delete the user.
5. Can view all the learning paths.
6. Can delete the learning path.
7. Can view all the resources.
8. Can delete the resource.
9. Can view and edit the profile.

**4. Core Concepts and Libraries Used in Angular**

**4.1 Core Concepts Used**

* Used separate guards for users and admin in order to restrict access to unauthorised users.
* Create different modules for different type of users.
* Used Lazy-loading concept to load the admin module which will improve the application loading.
* Created services for the main features which consume the apis created in the backend.
* Used reactive forms for handling and validating user input.
* Neat and clean design with responsiveness.
* Created models for all the main objects of the features.
* Used angular material tables with sort, filter and pagination
* Reset password was implemented where user gets a mail for changing his password.
* File uploading was used to upload user’s profile image.

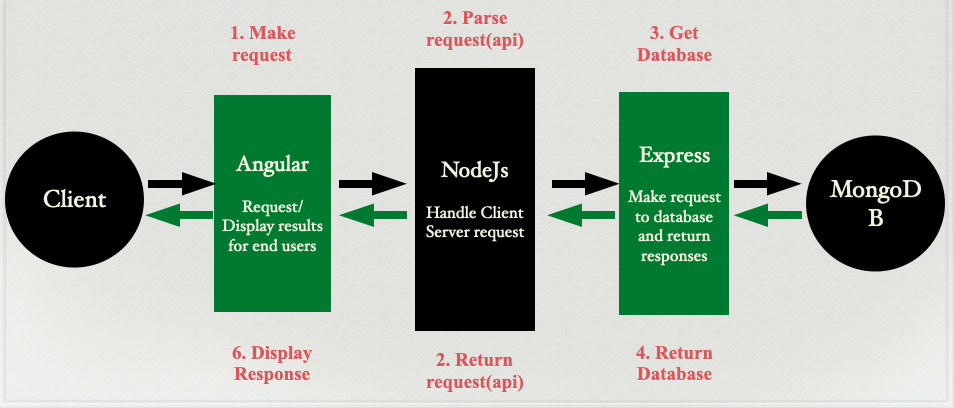
**4.2 Libraries Used**

* Bootstrap
* Angular Material
* Multer (for file uploading)
* Jsonwebtoken
* Nodemailer (for sending emails)
* Mongoose (for interacting with mongodb)

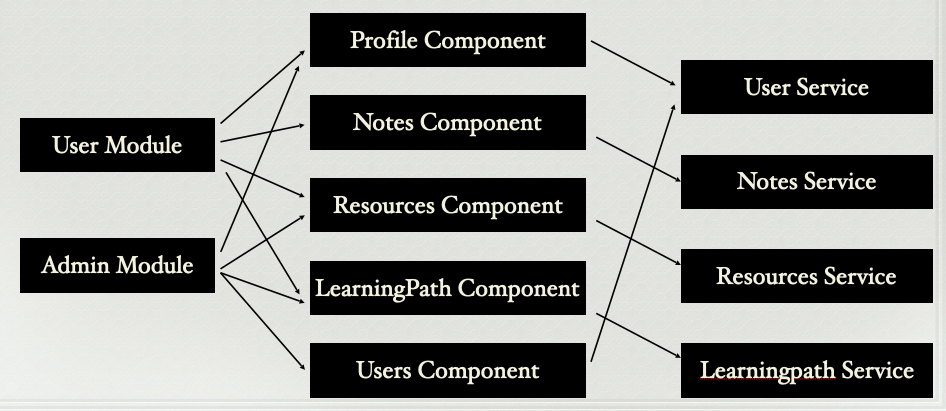
**5. Project Learning and Outcome**

* Building an end to end Real World Web Application using MEAN Stack.
* Integration of Frontend with Backend.
* Understanding and Implementing the core Features of Angular.
* CRUD operations with the persisting data.
* Implementing the API calls from Frontend.
* Implementing Security features including Authentication and Authorization as well as protection of routes using guards based on role (Admin/User).
* Handling input forms with Validations.
* Fetching and sending the response from Backend.
* Implementing file uploading to upload profile image.
* Implementing email notifications for reset-password and reset password successful.

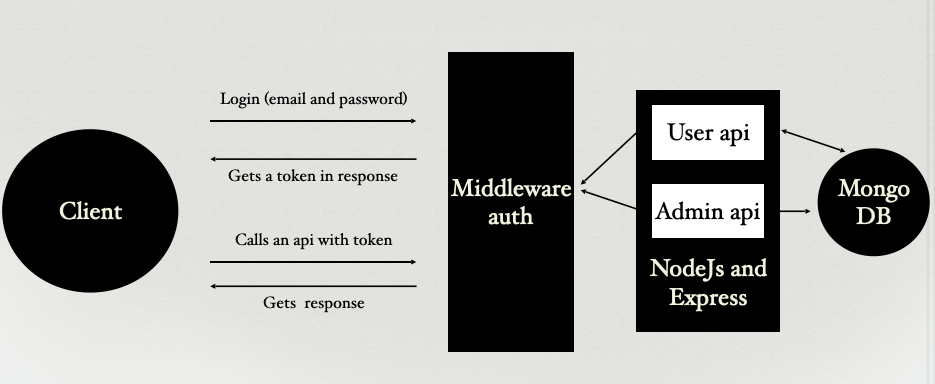
**6. Project Architecture Diagram**

****

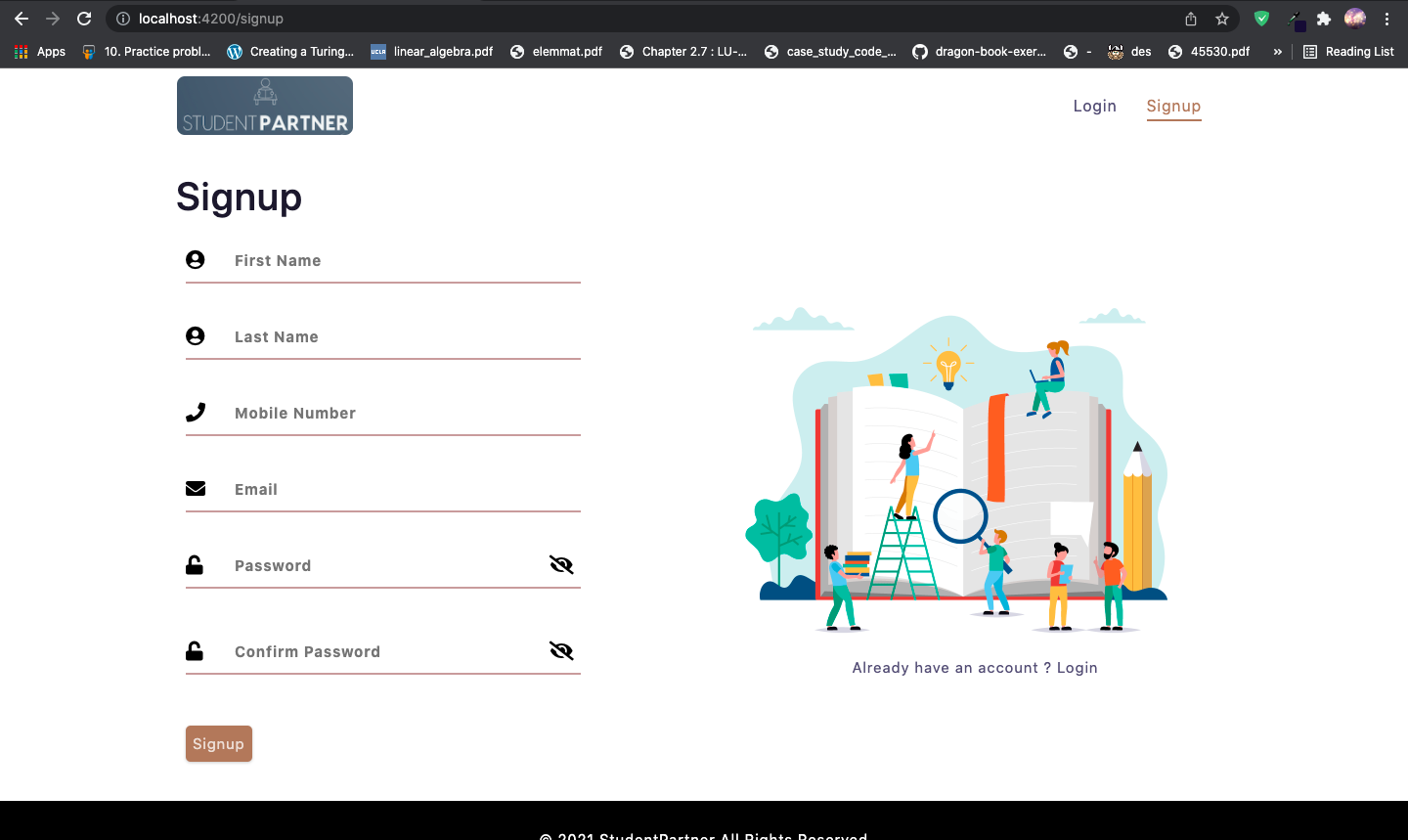
**7. Component Level Architecture Diagram**



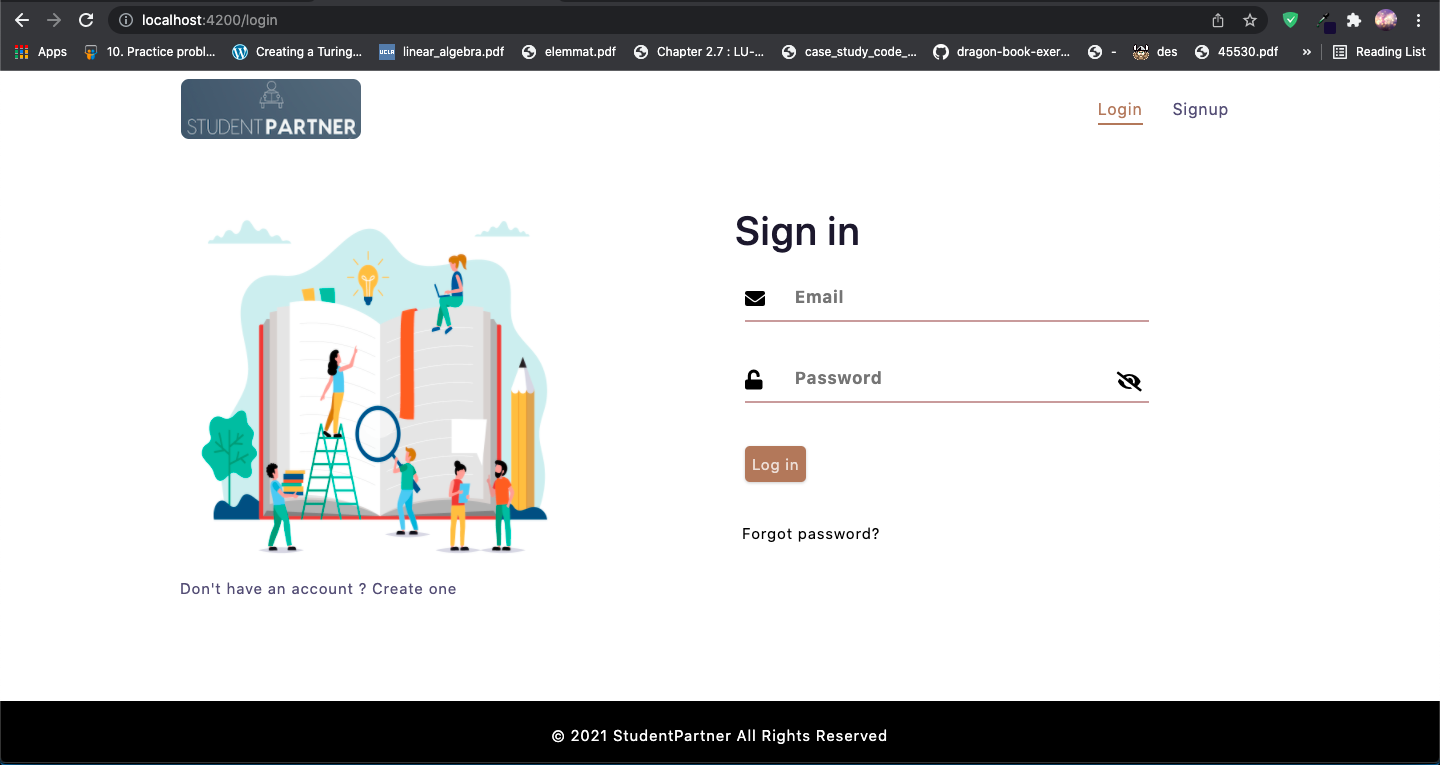
**8. SECURITY ARCHITECTURE DIAGRAM**

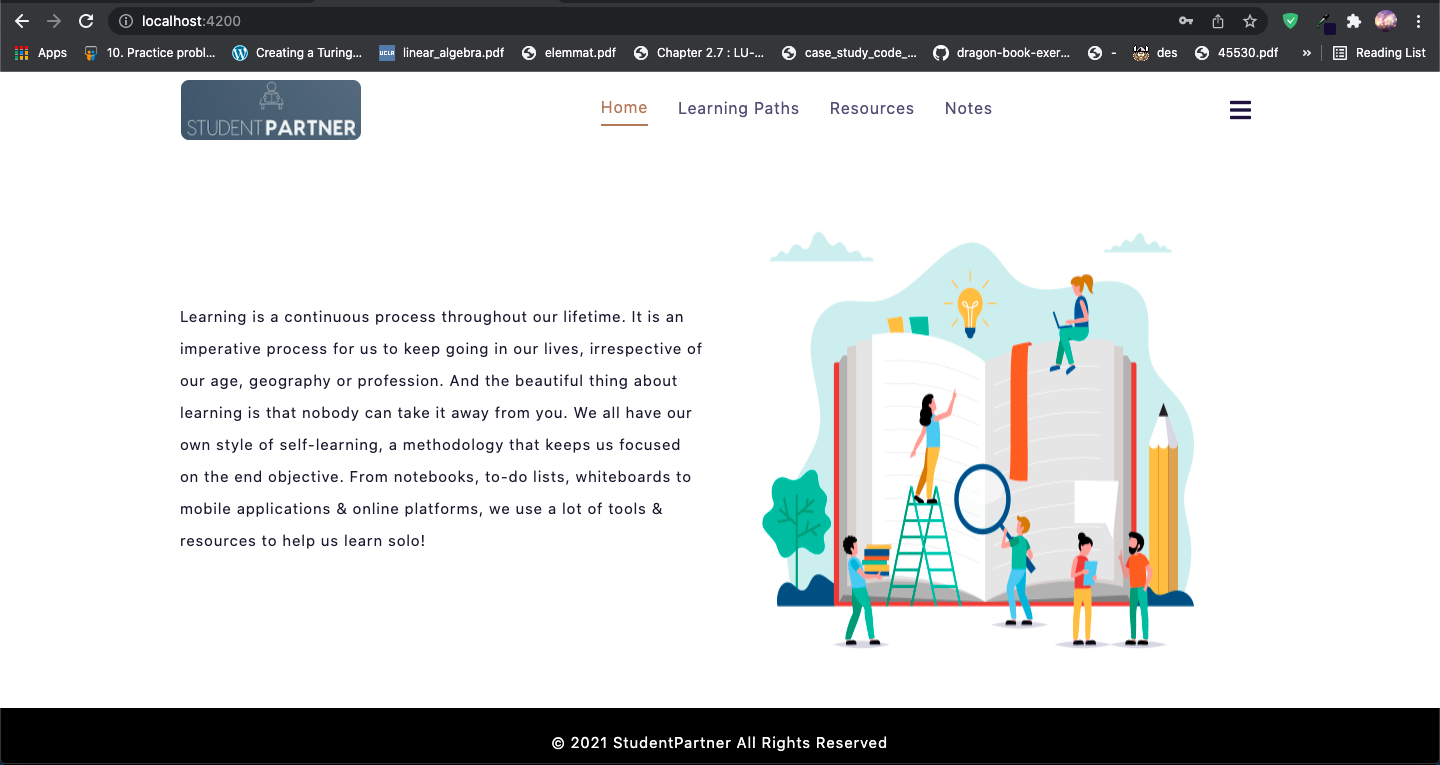
****

**9. Front End pages.**

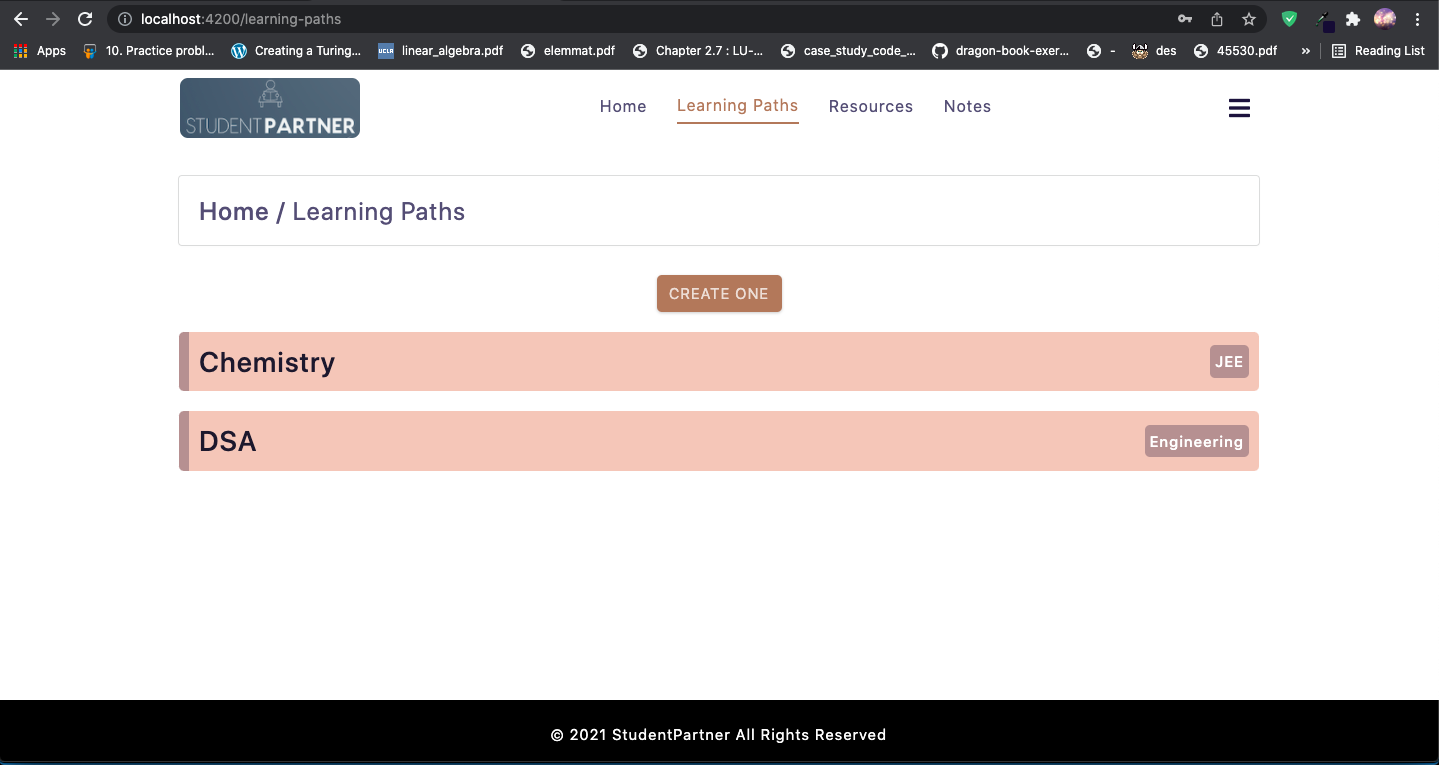
**Signup Page:**

**Login Page:**

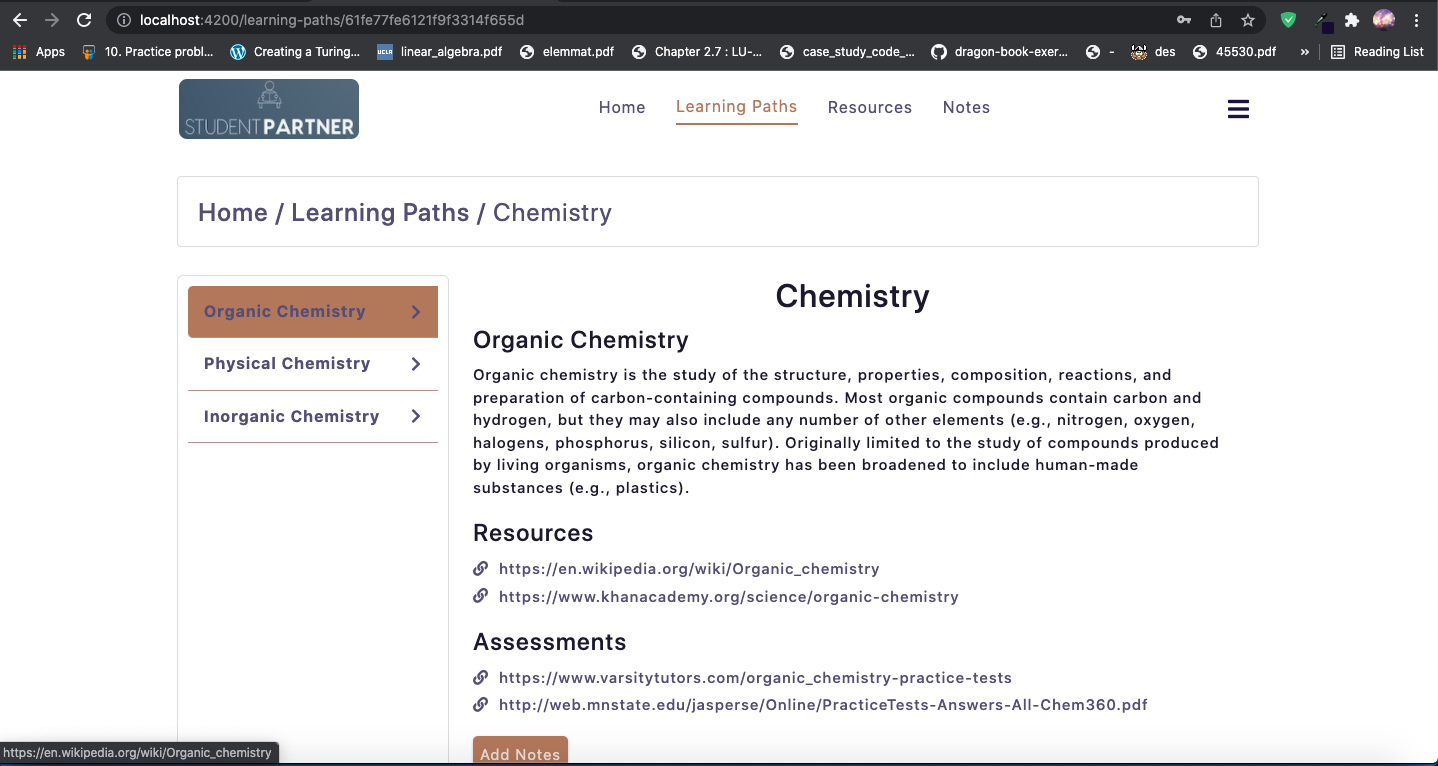
****

**User Home Page:**

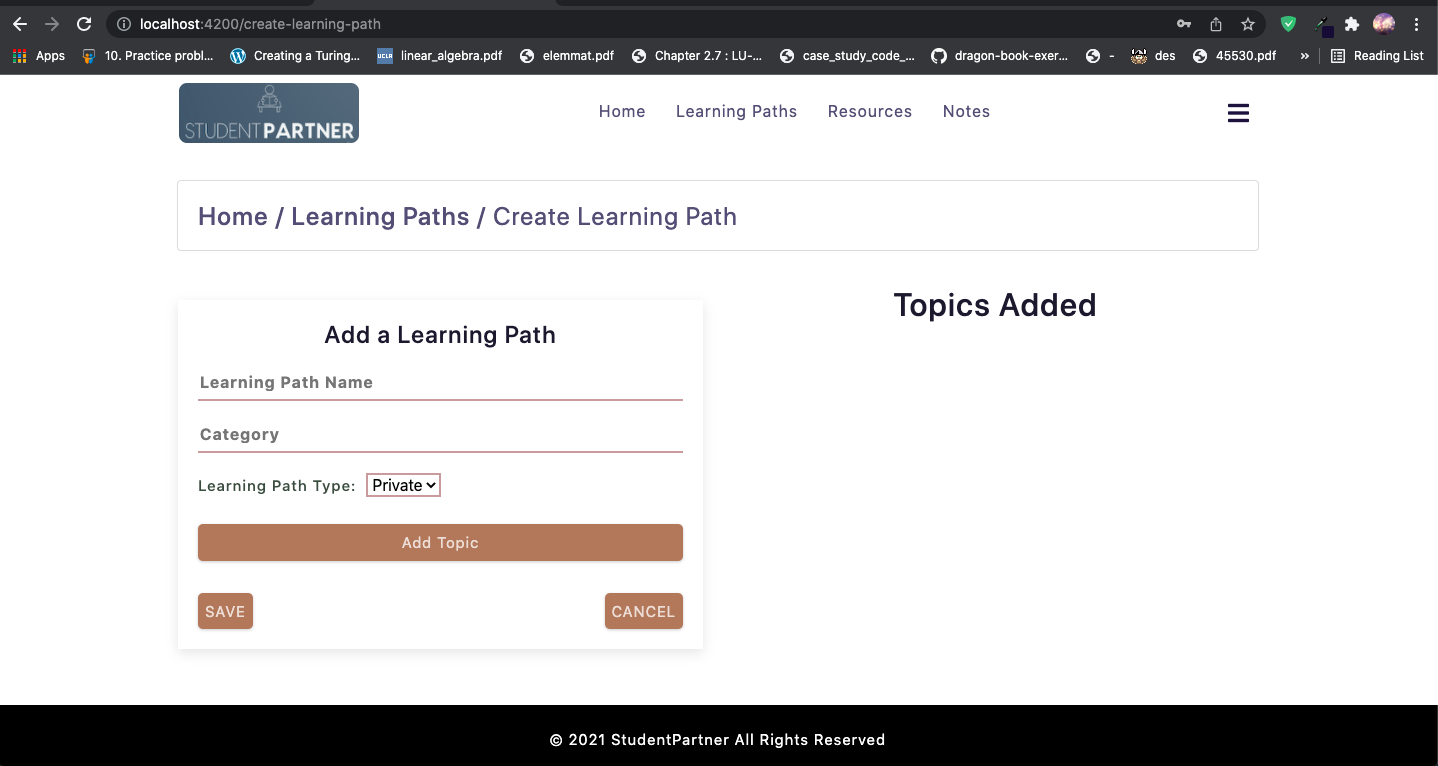
**User Learning Paths Page:**

****

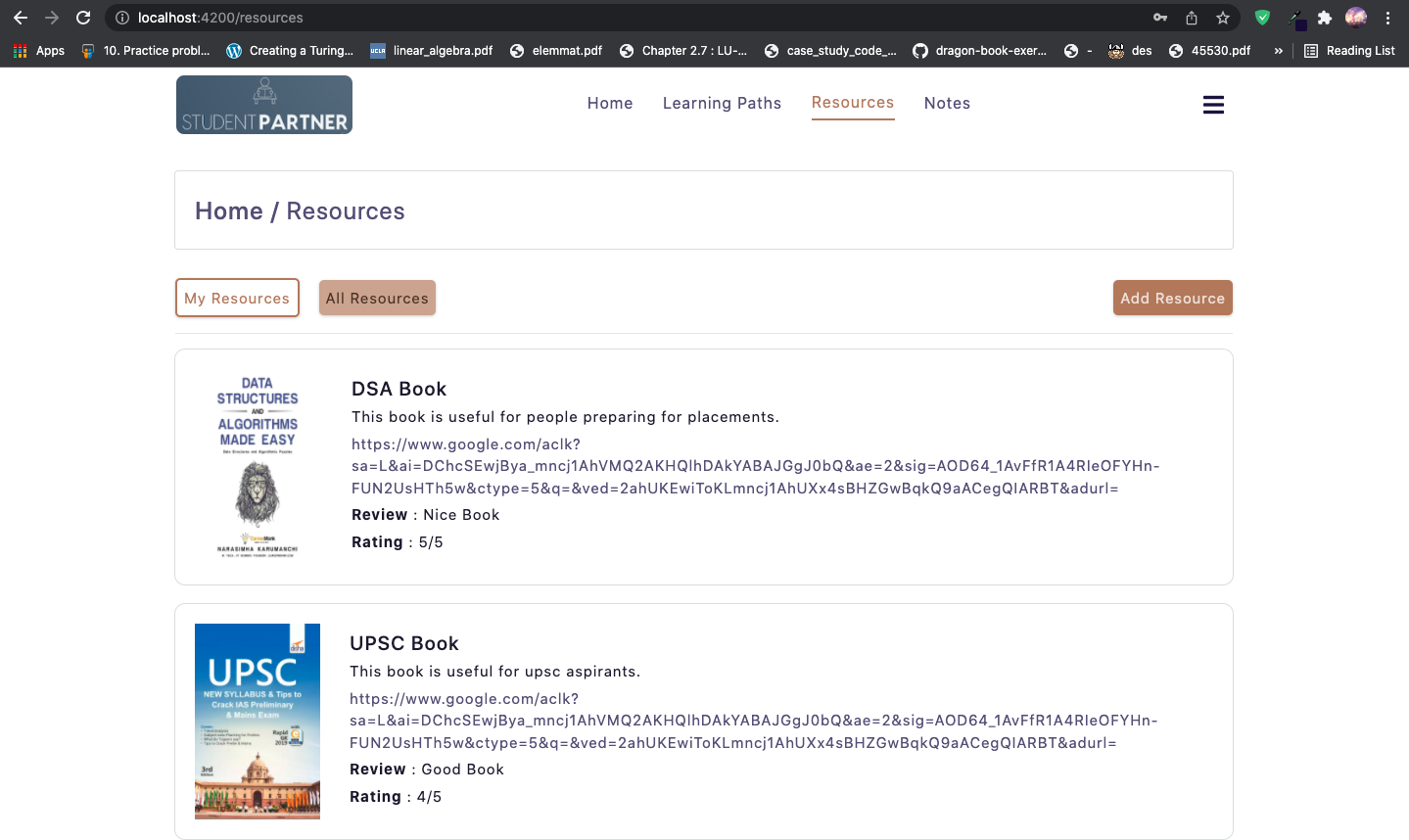
**Learning Path detailed page:**

****

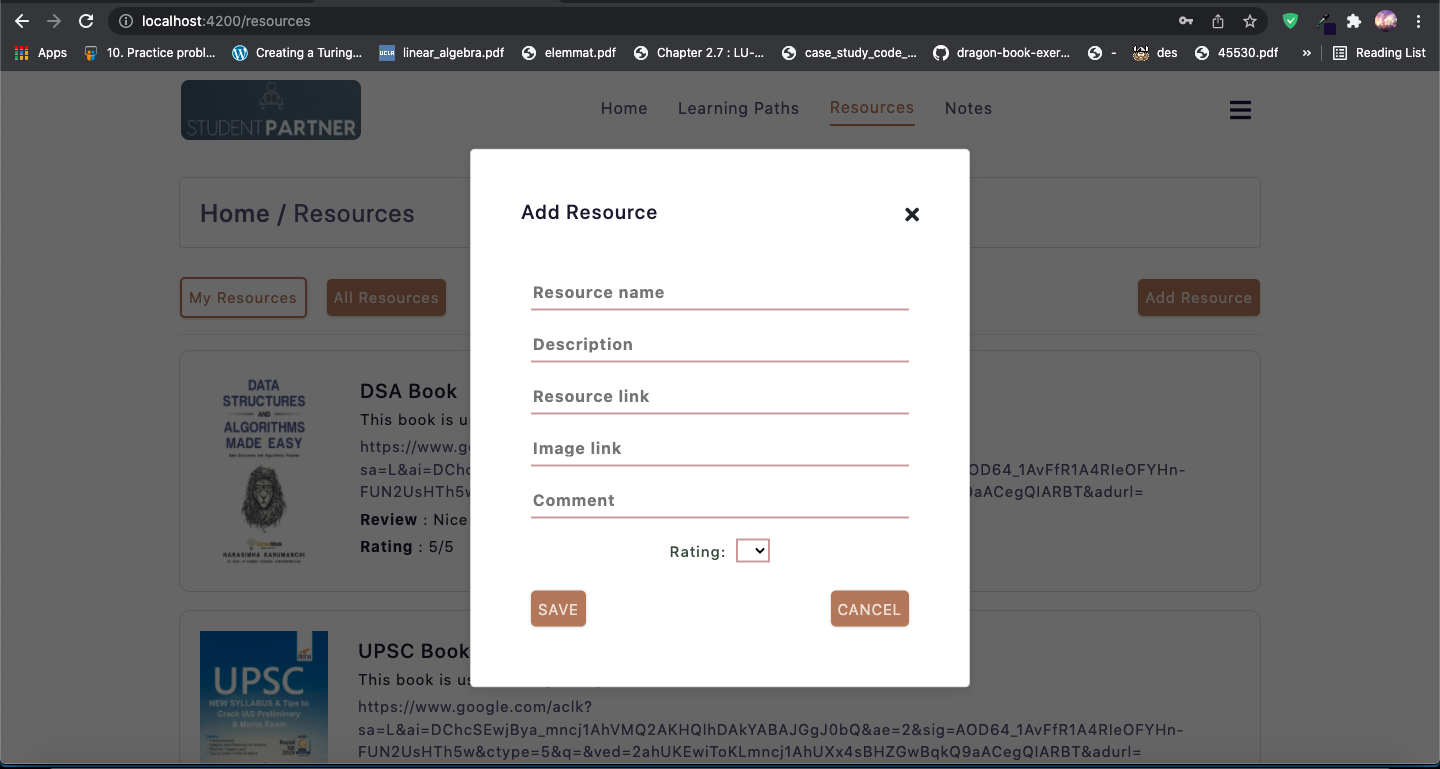
**Create Learning Path page:**

****

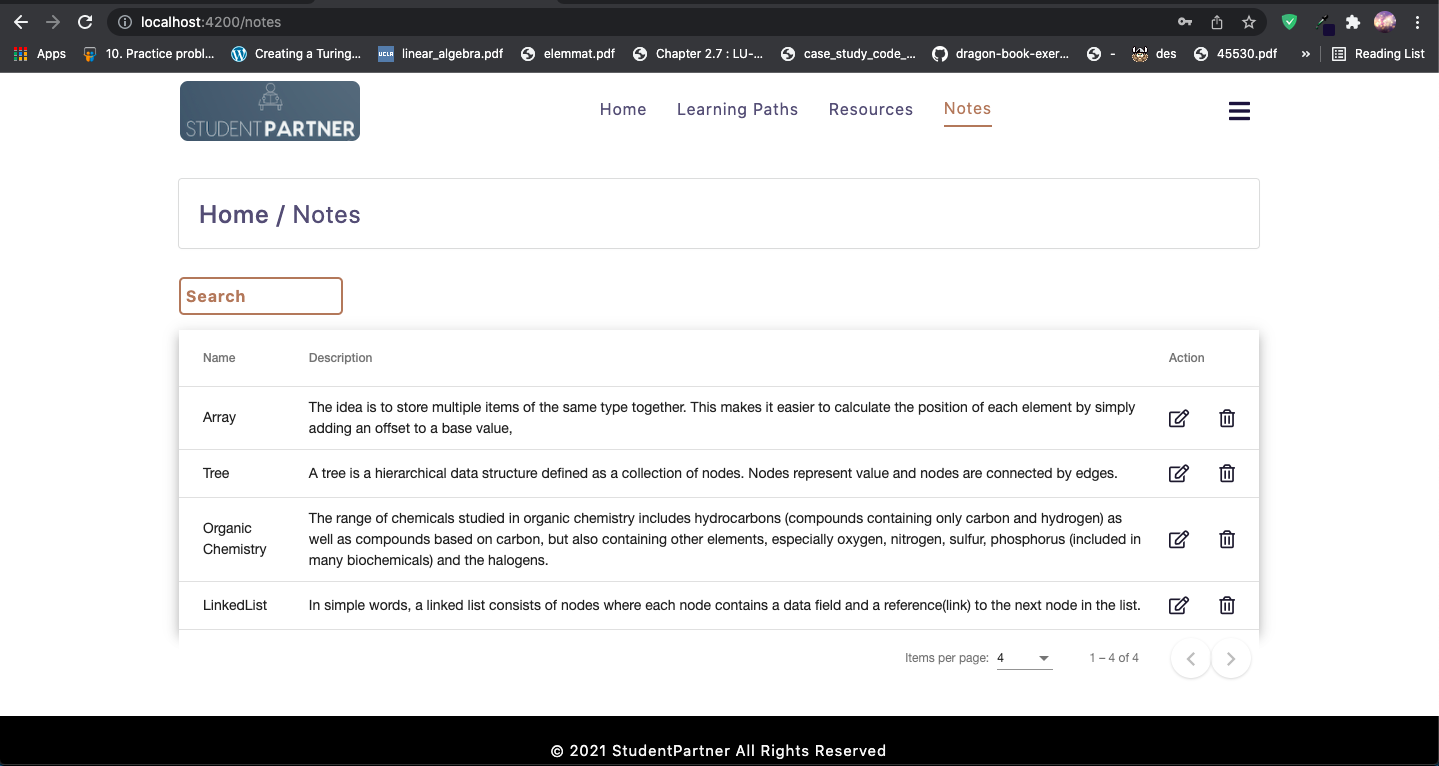
**User Resources page:**

****

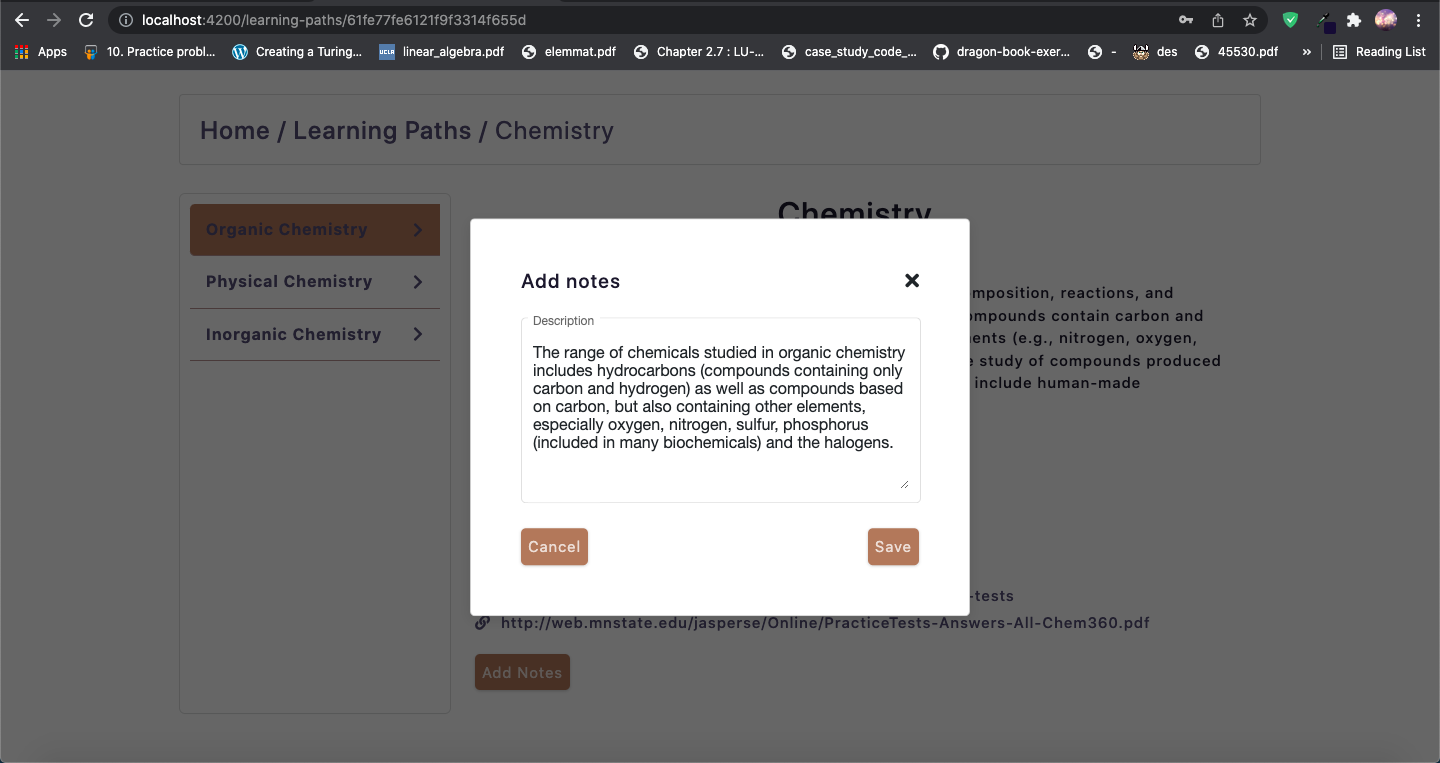
**Create Resource page:**

****

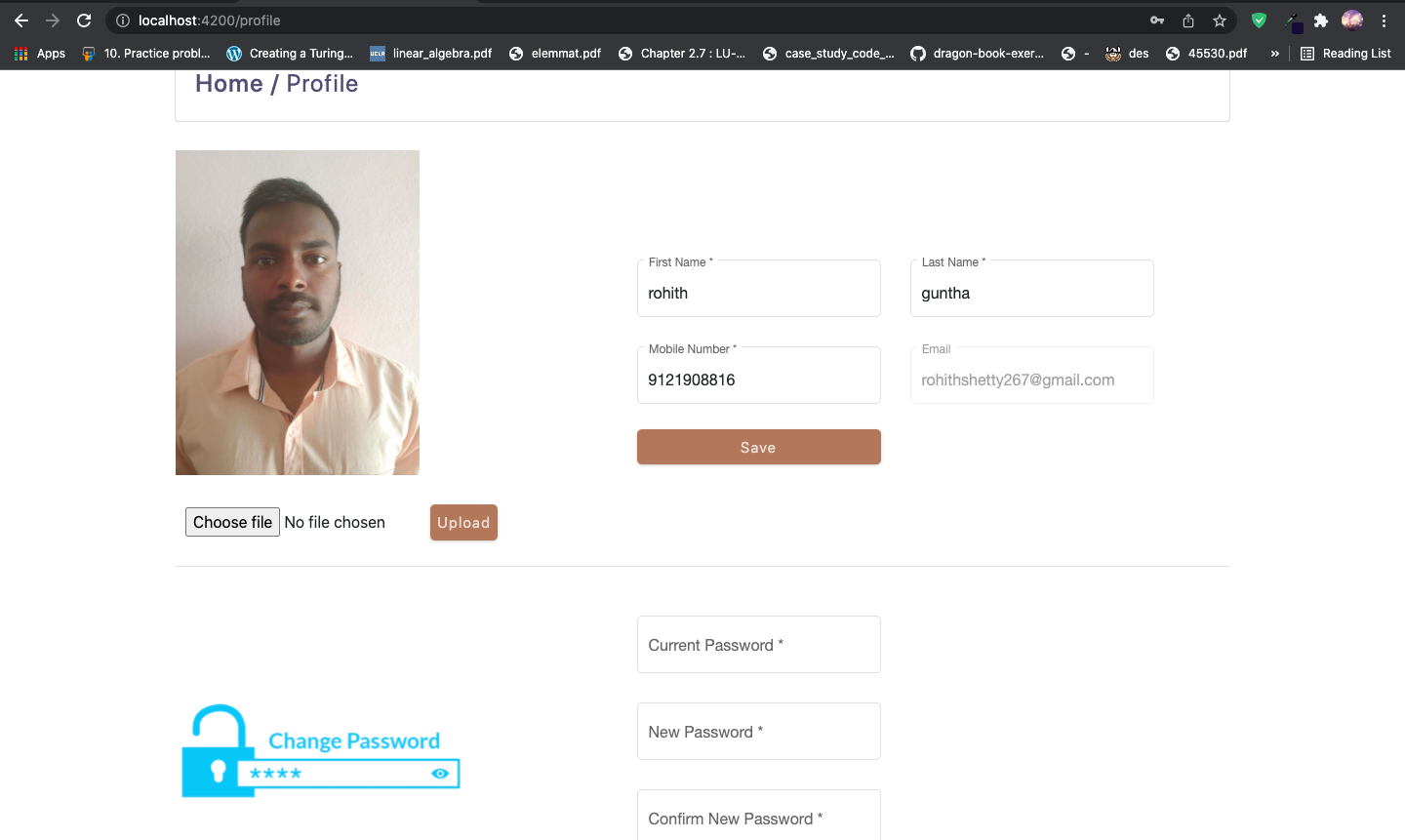
**Notes page:**

****

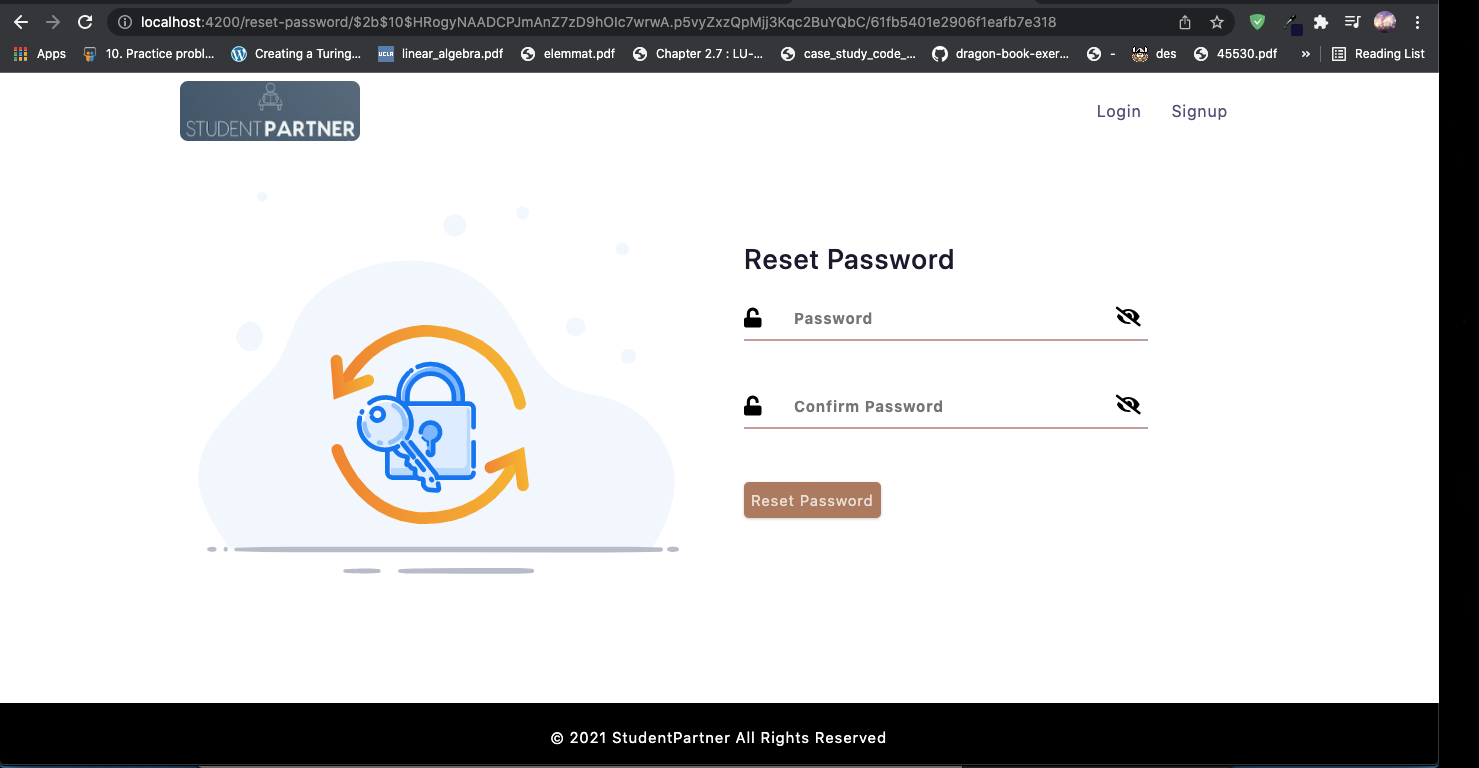
**Add notes page:**

****

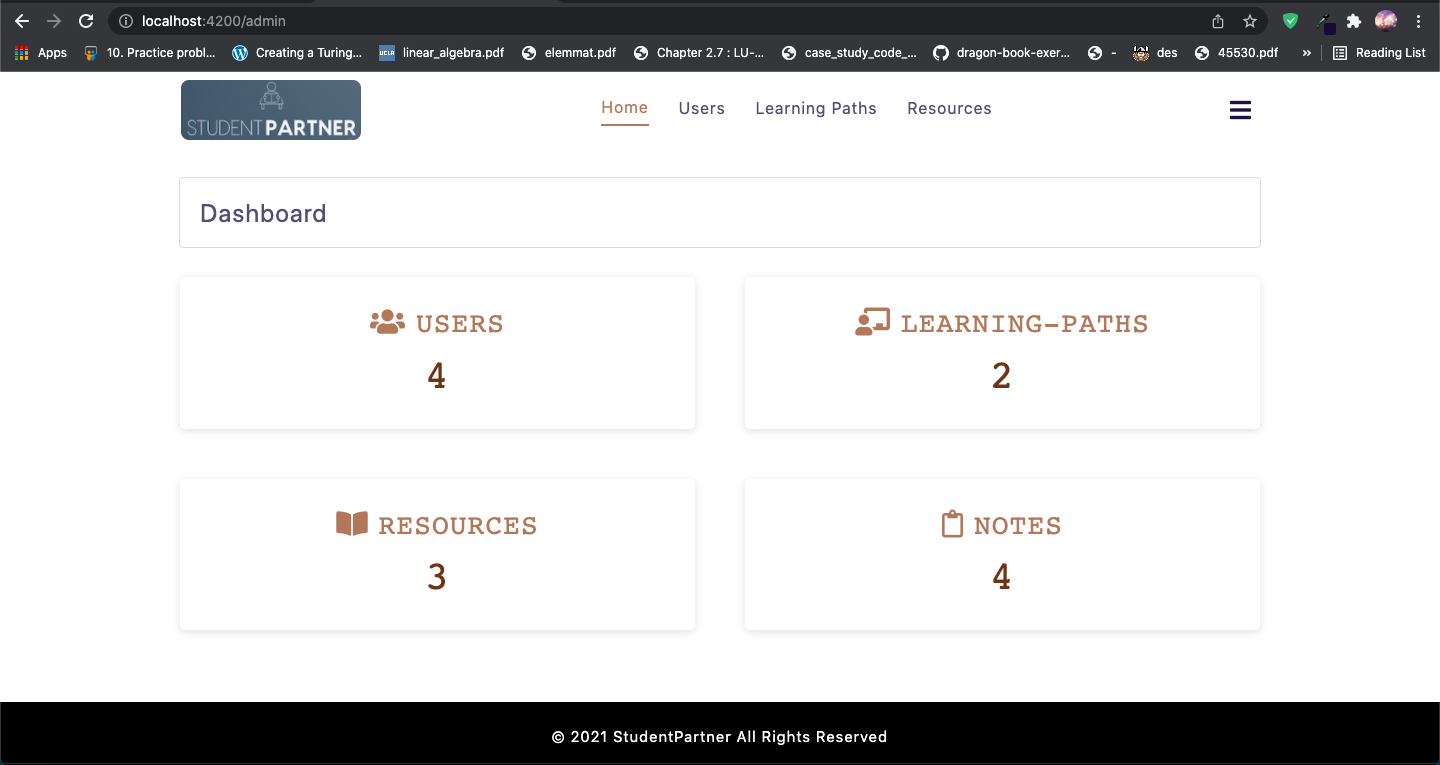
**Profile page:**

****

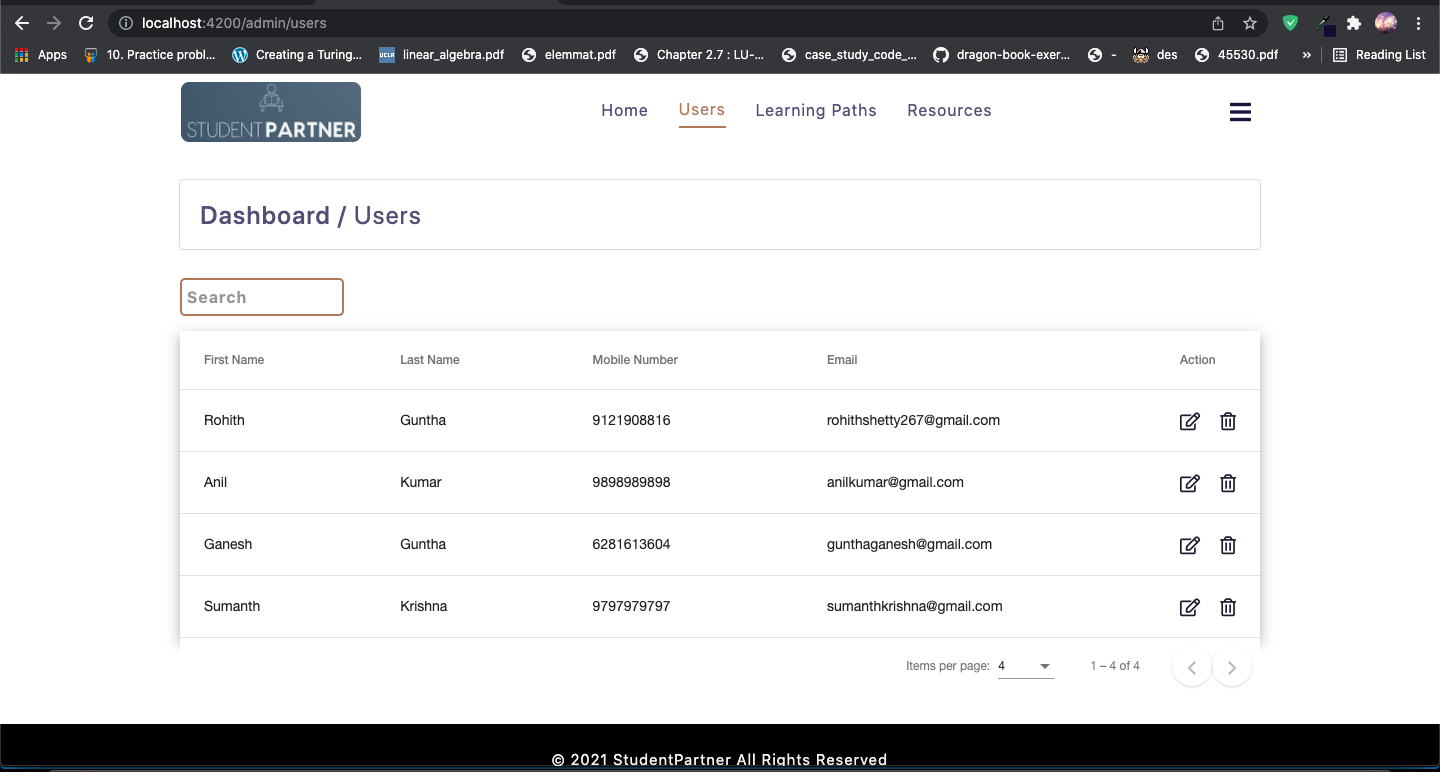
**Reset password page:**

****

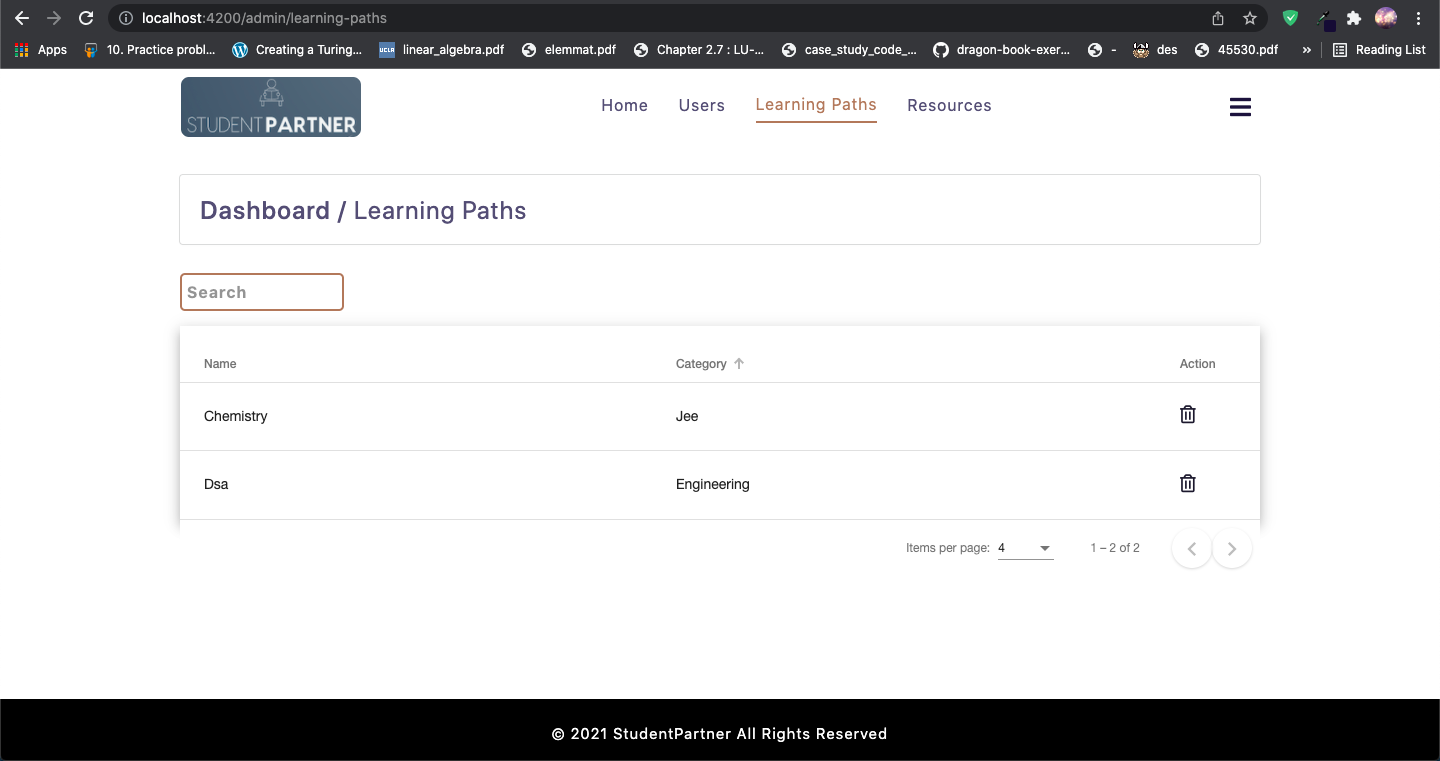
**Admin dashboard:**

****

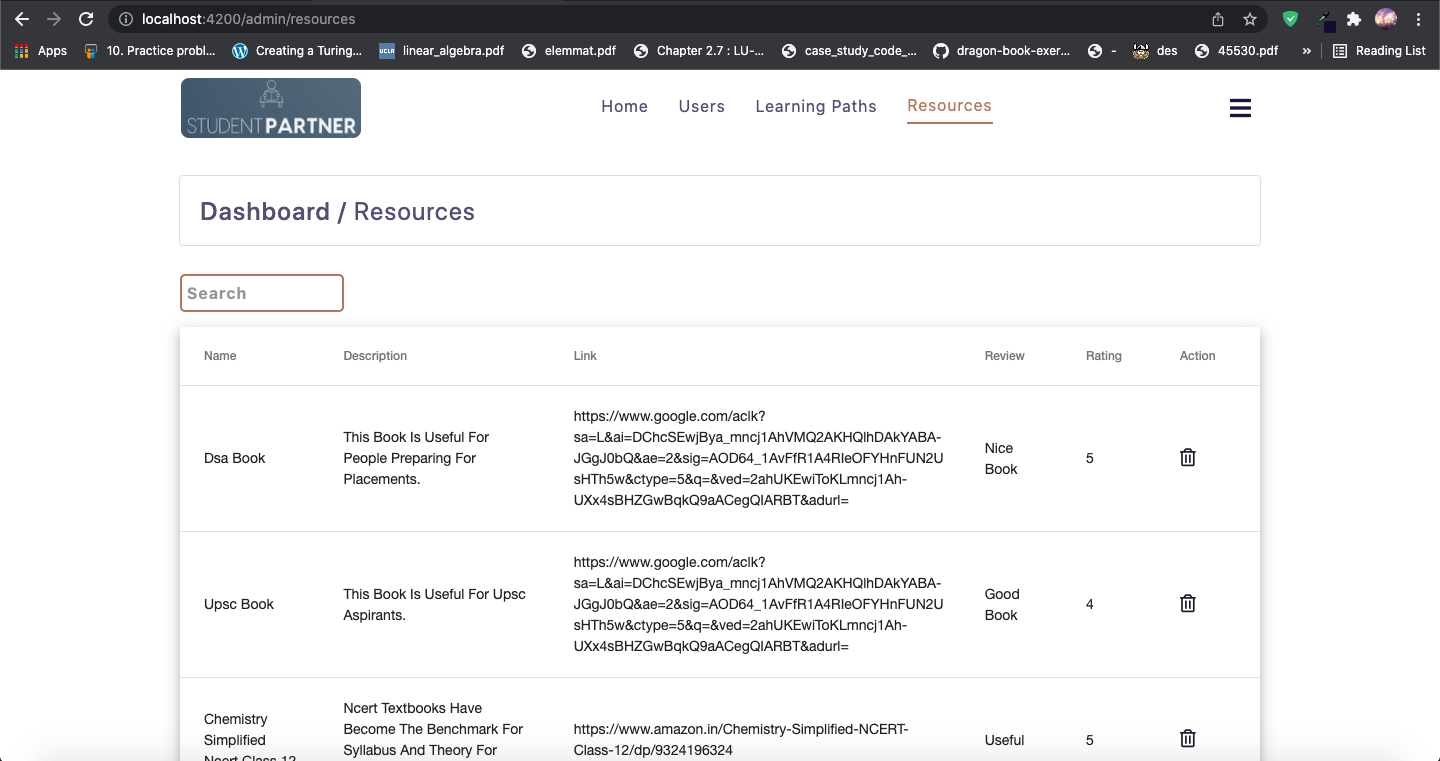
**Users page:**

****

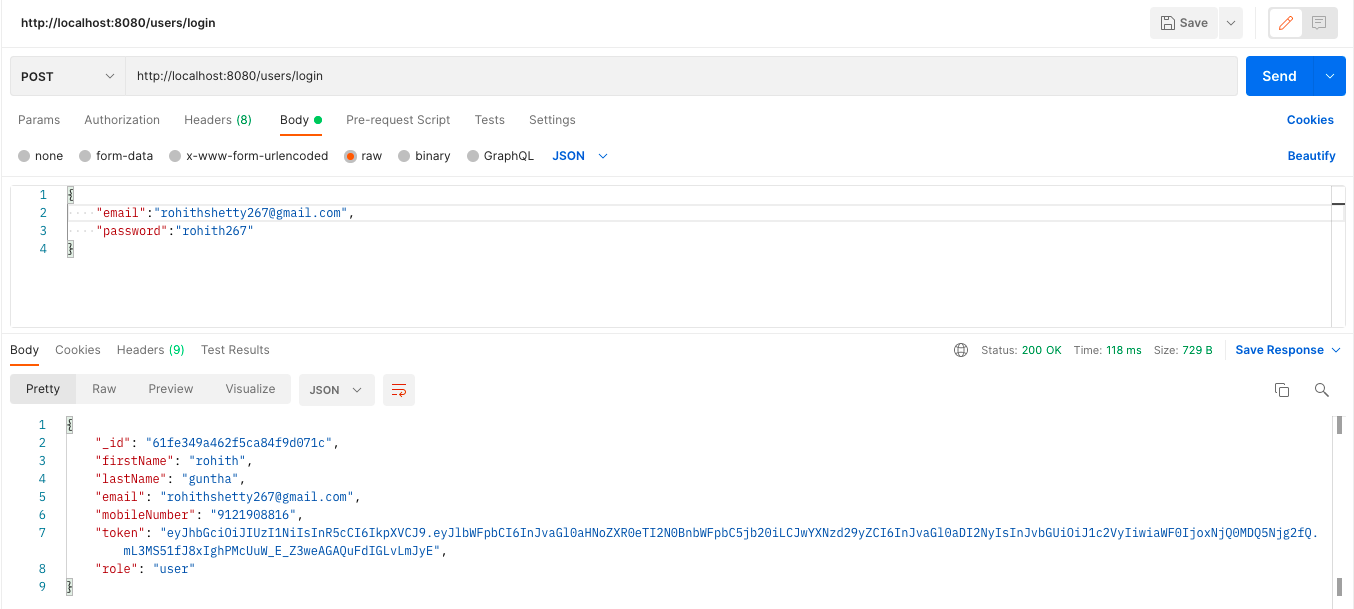
**Admin Learning Paths page:**

****

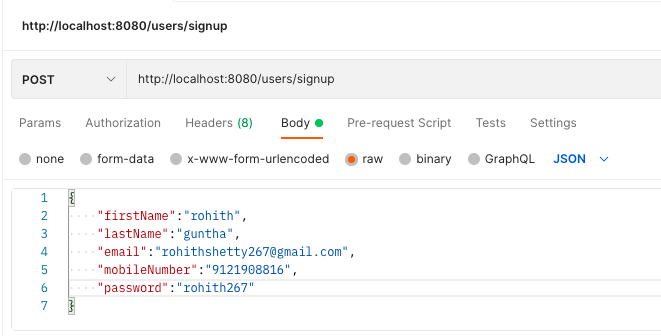
**Admin Resources page:**

****

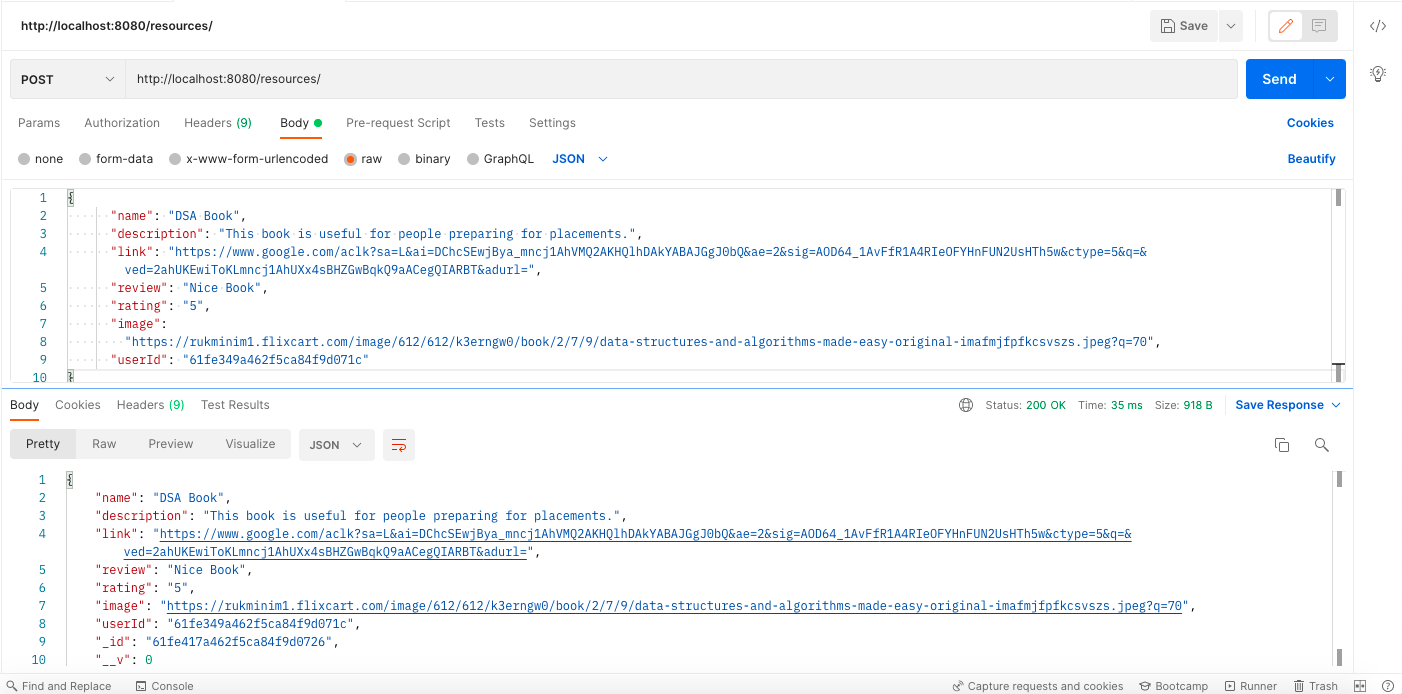
**10. Api testing using Postman**

**Login:**

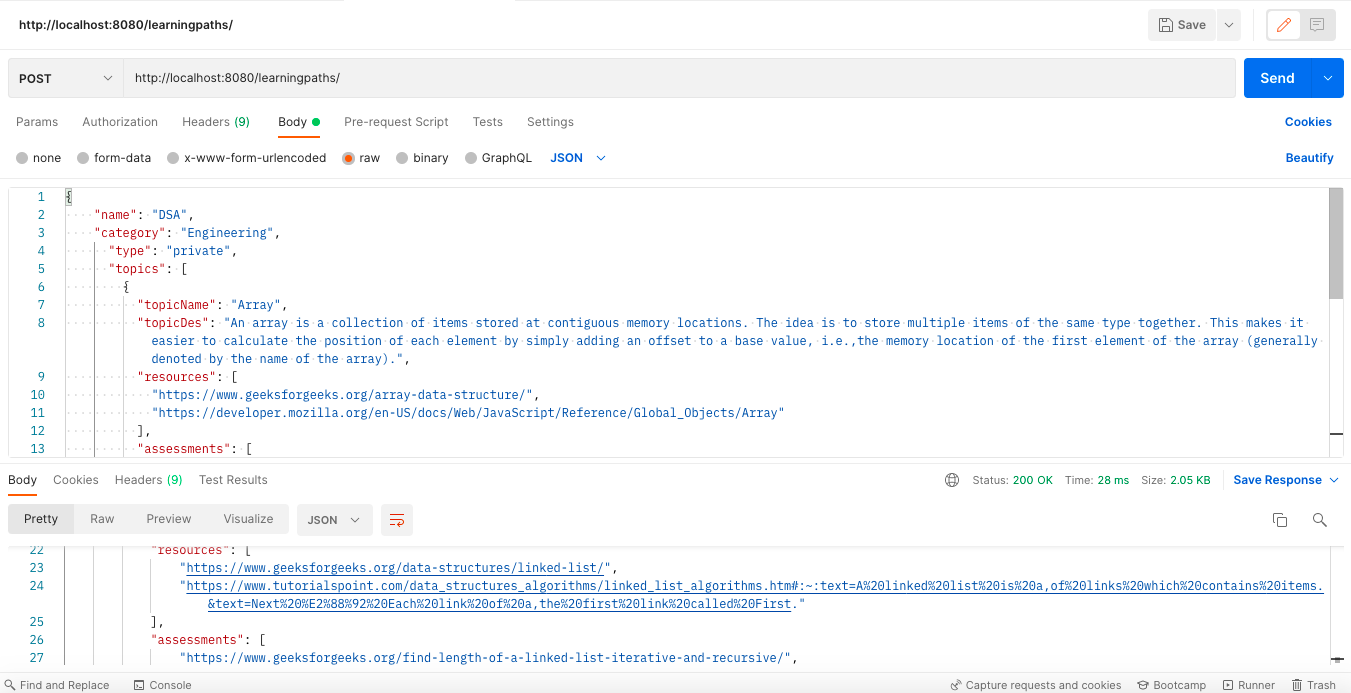
**Signup:**

****

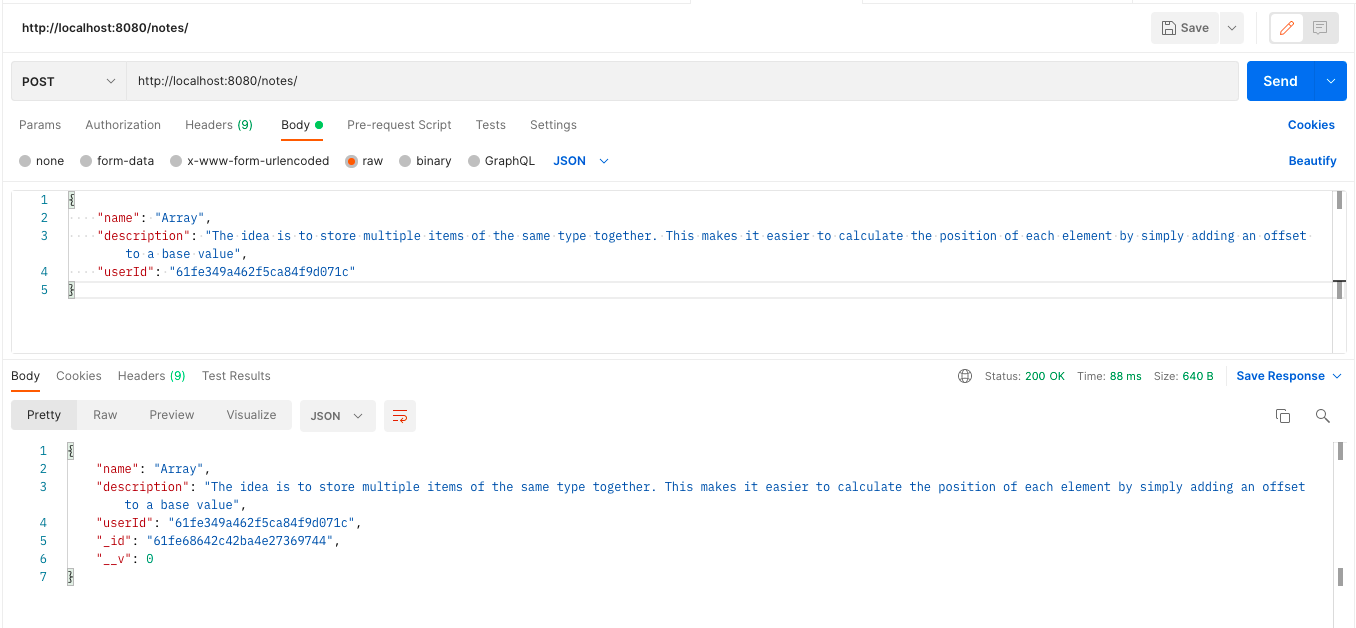
**Add resource:**

****

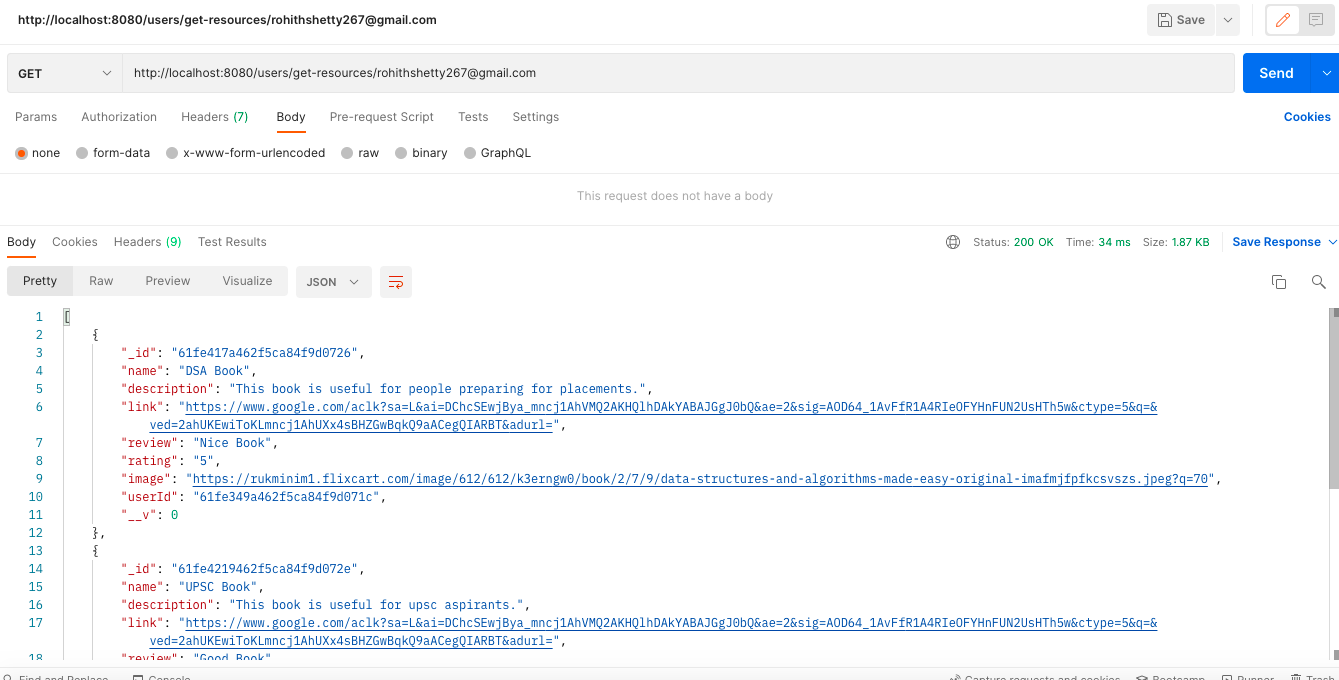
**Add learning path:**

****

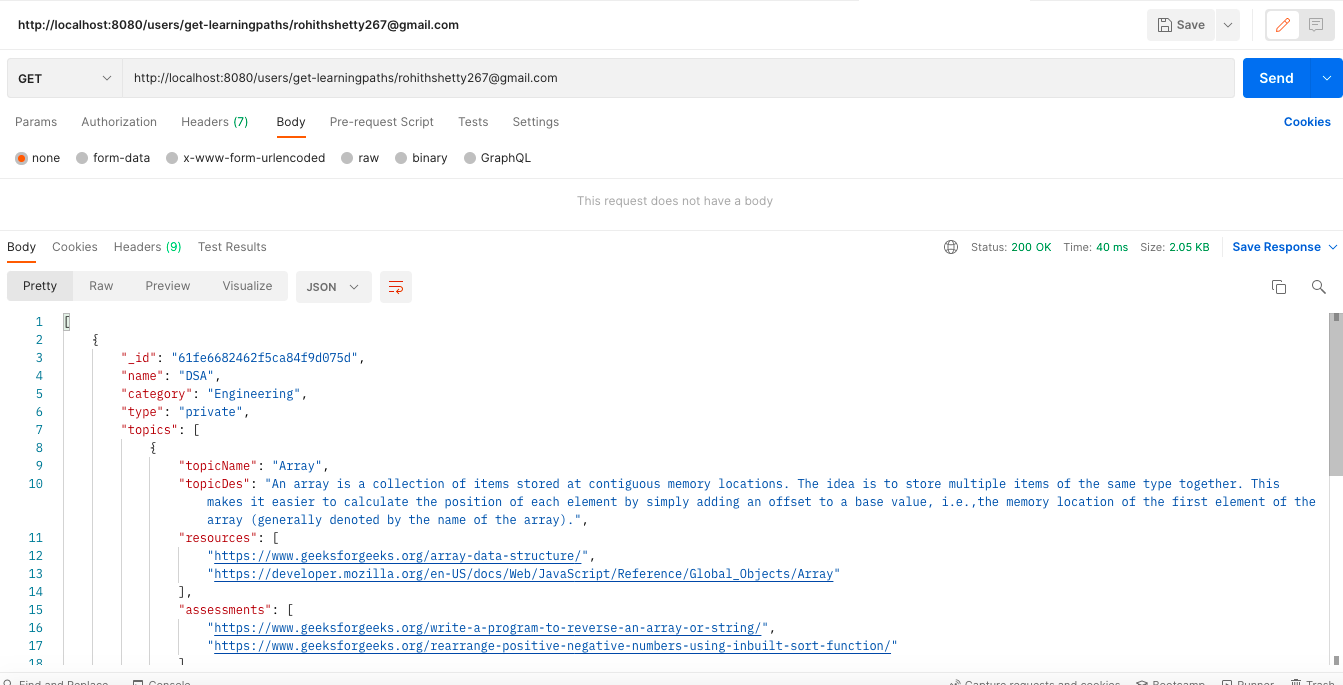
**Add notes:**

****

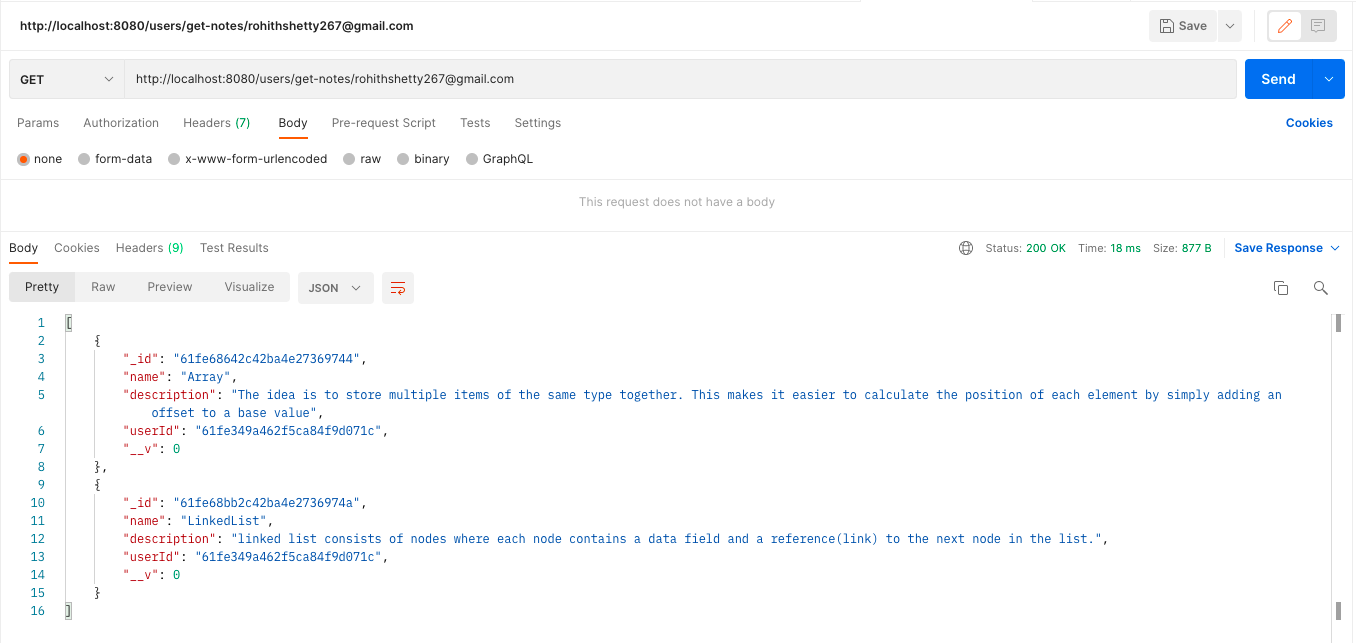
**Get user resources:**

****

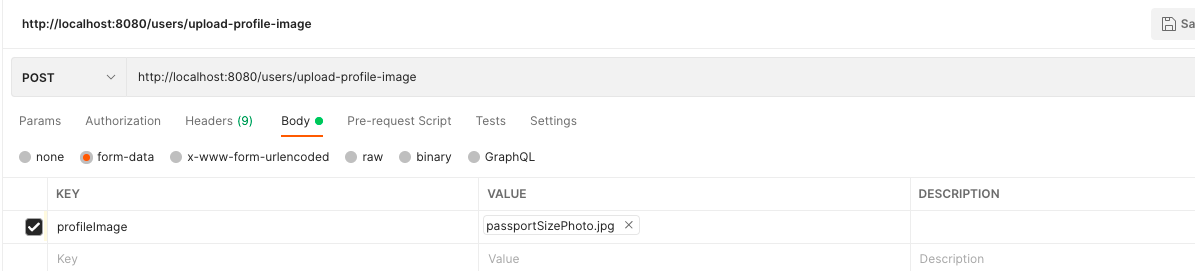
**Get user learning paths:**

****

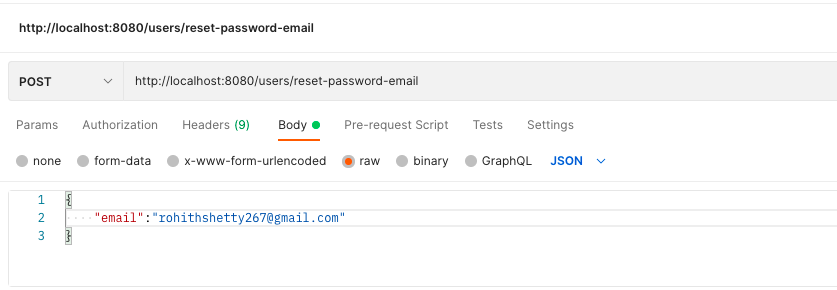
**Get user notes:**

****

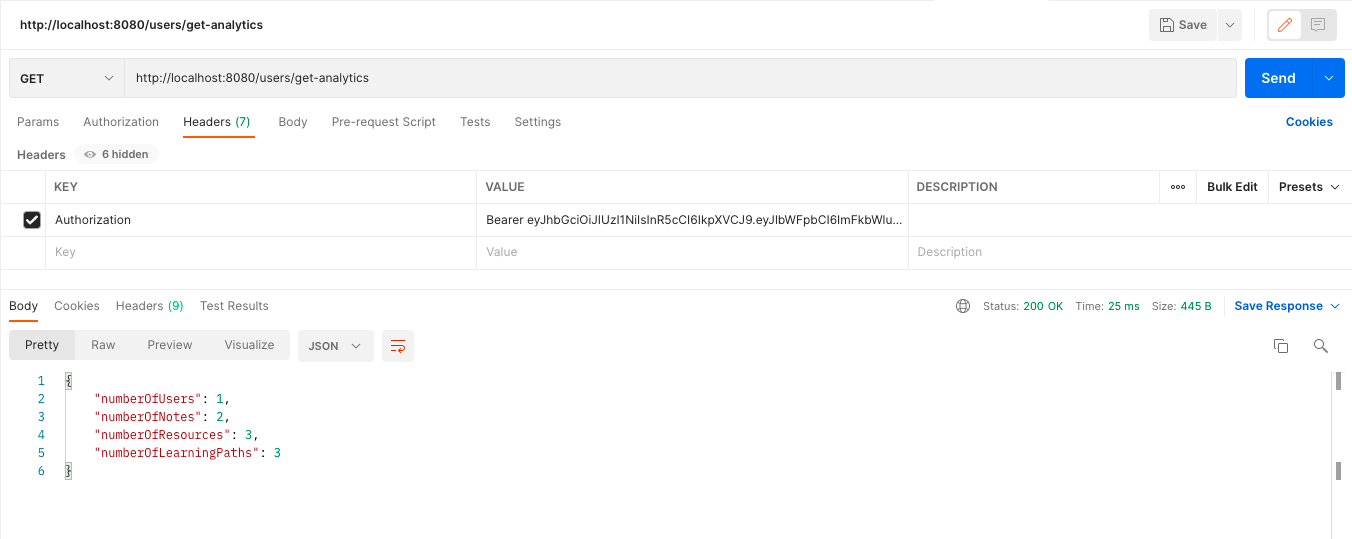
**Upload profile image:**

****

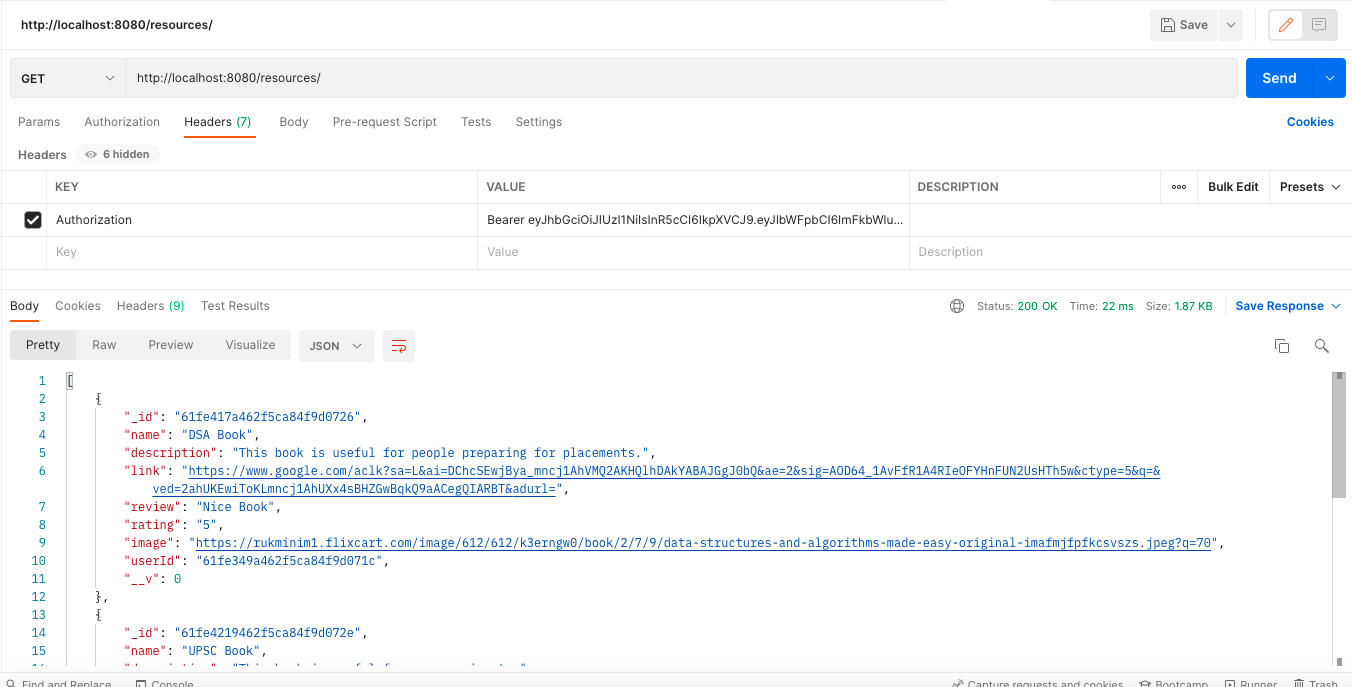
**Reset password:**

****

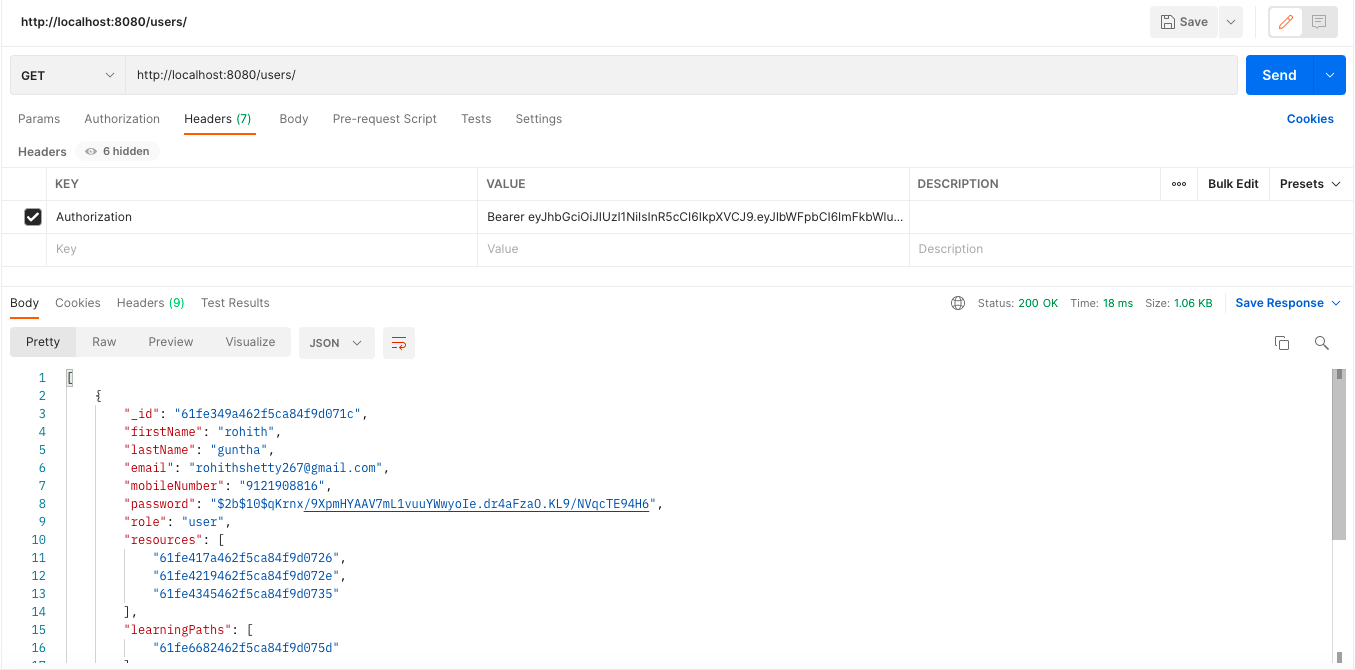
**Get analytics:**

****

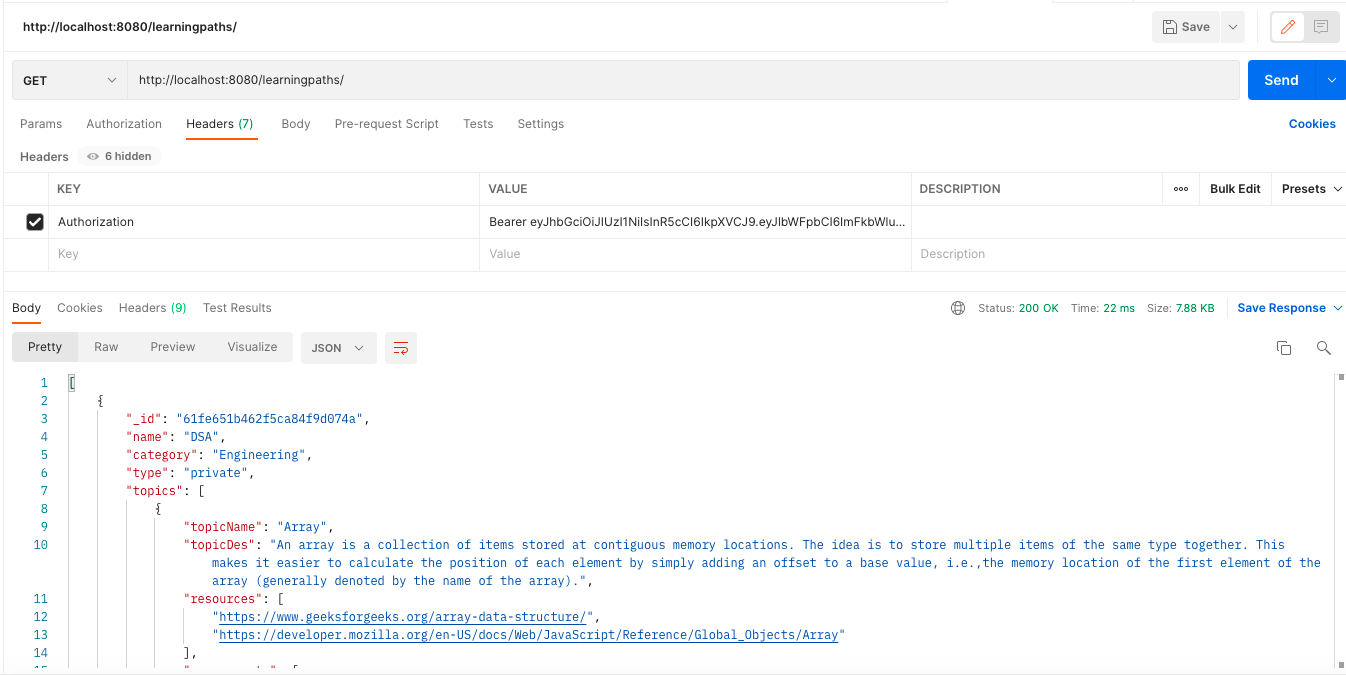
**Get admin resources:**

****

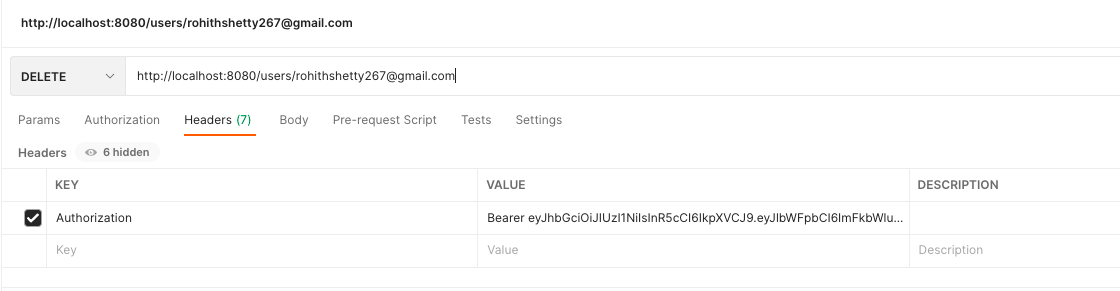
**Get admin users:**

****

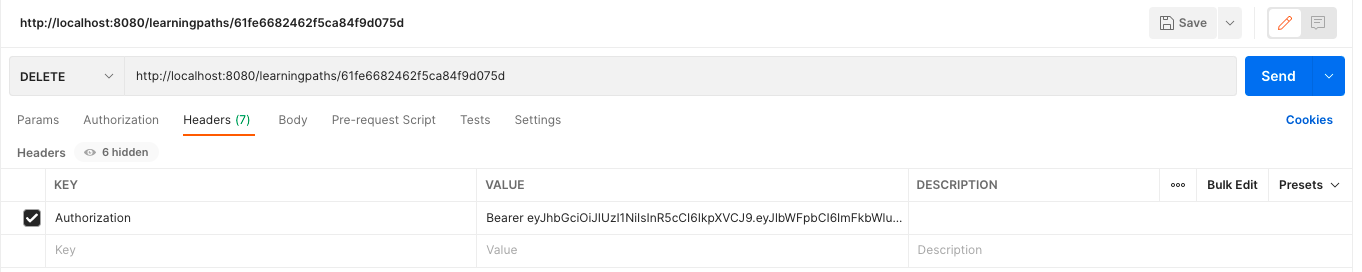
**Get admin learning paths:**

****

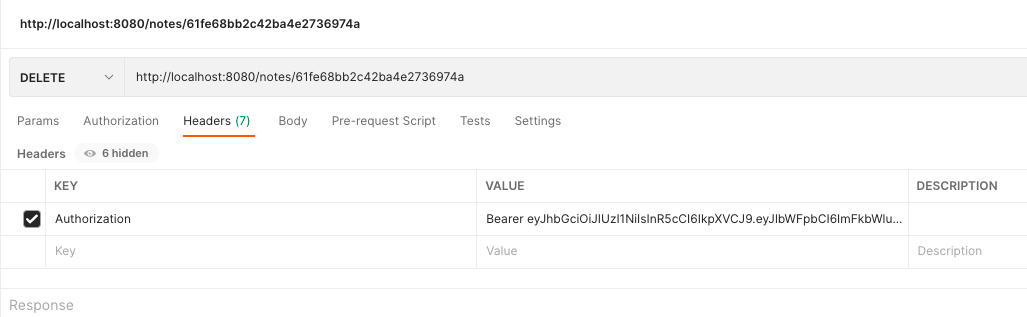
**Delete user:**

****

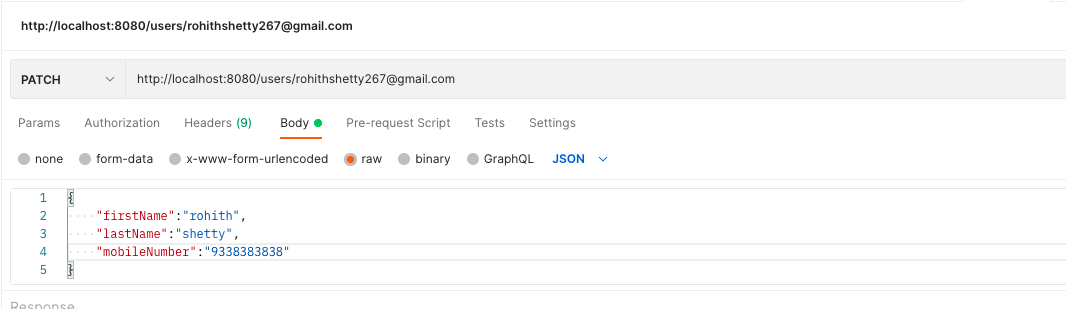
**Delete learning path:**

****

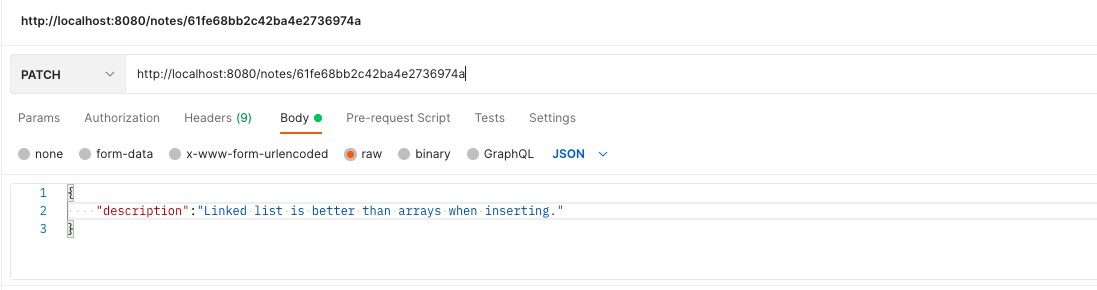
**Delete notes:**

****

**Edit user:**

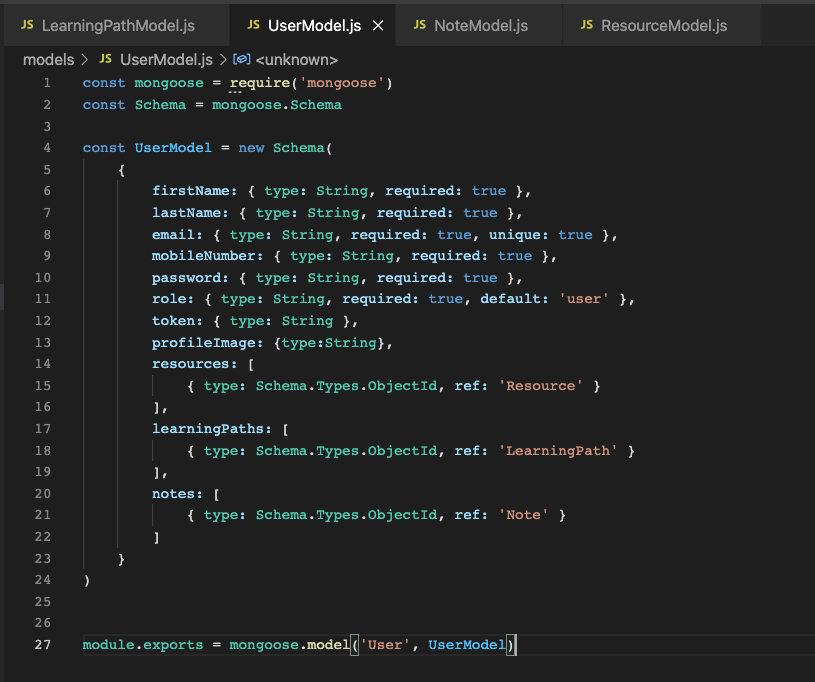
****

**Edit notes:**

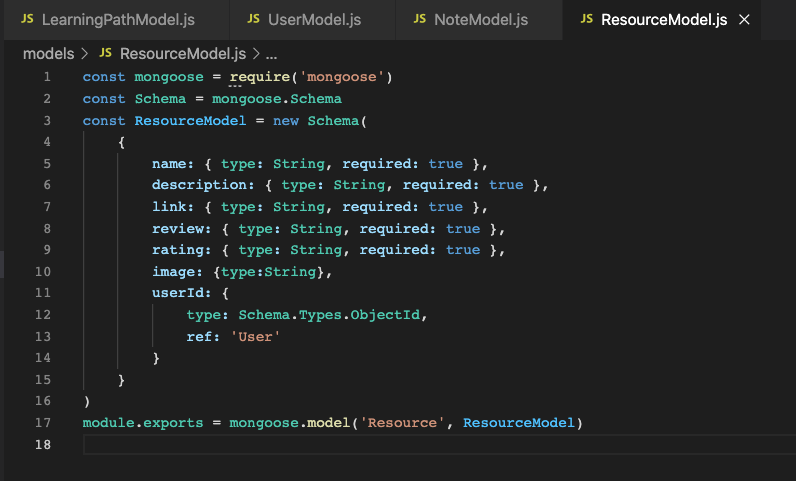
****

**11. Backend walk through**

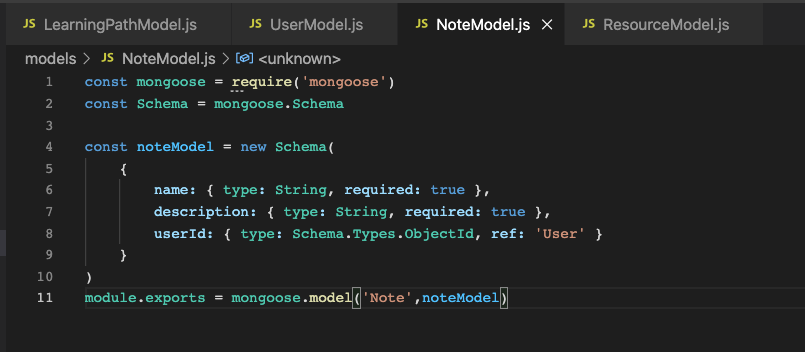
**User model:**

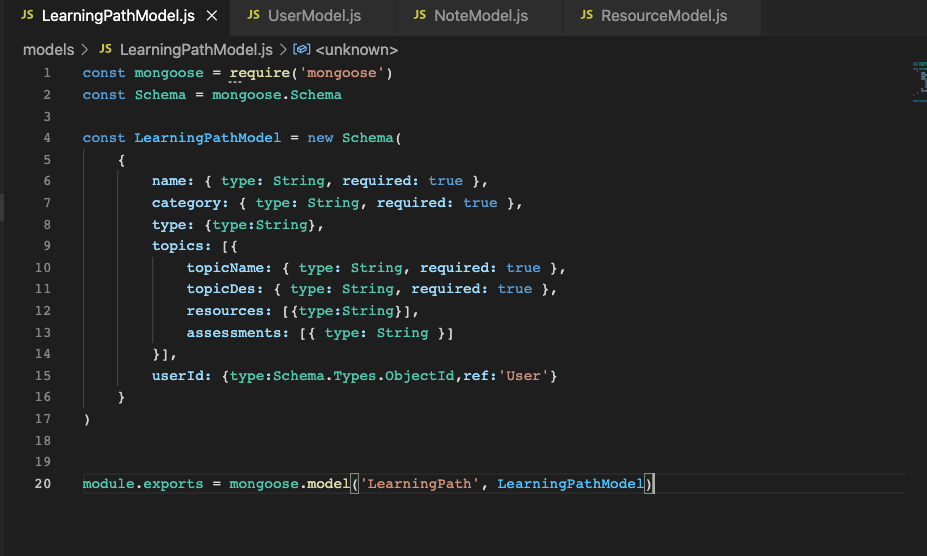
****

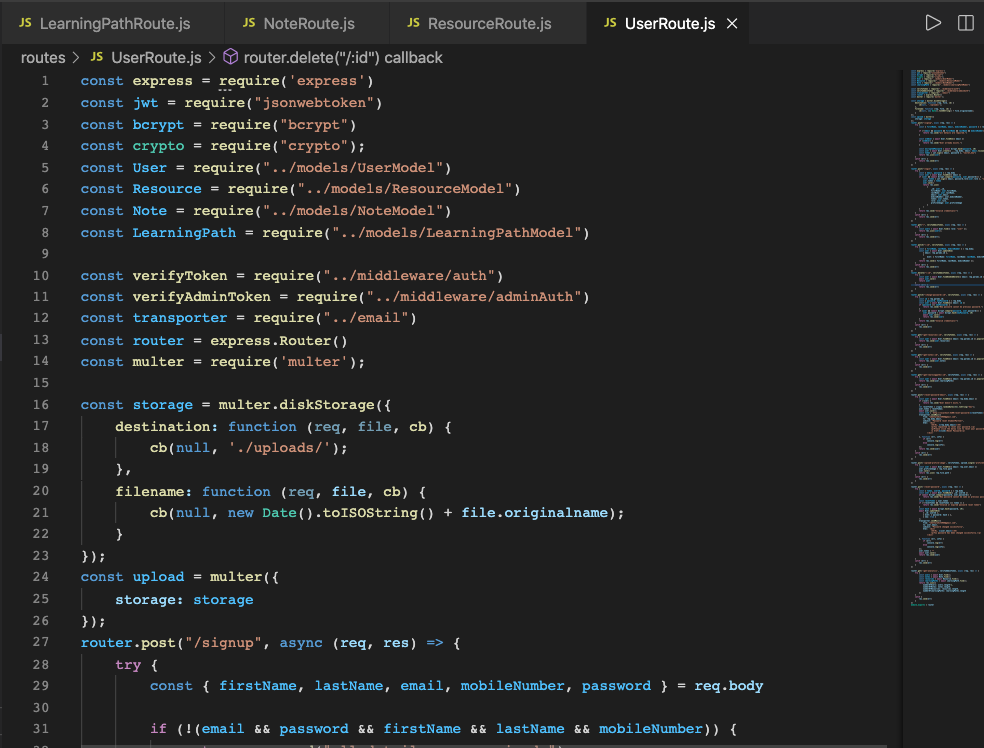
**Resource model:**

****

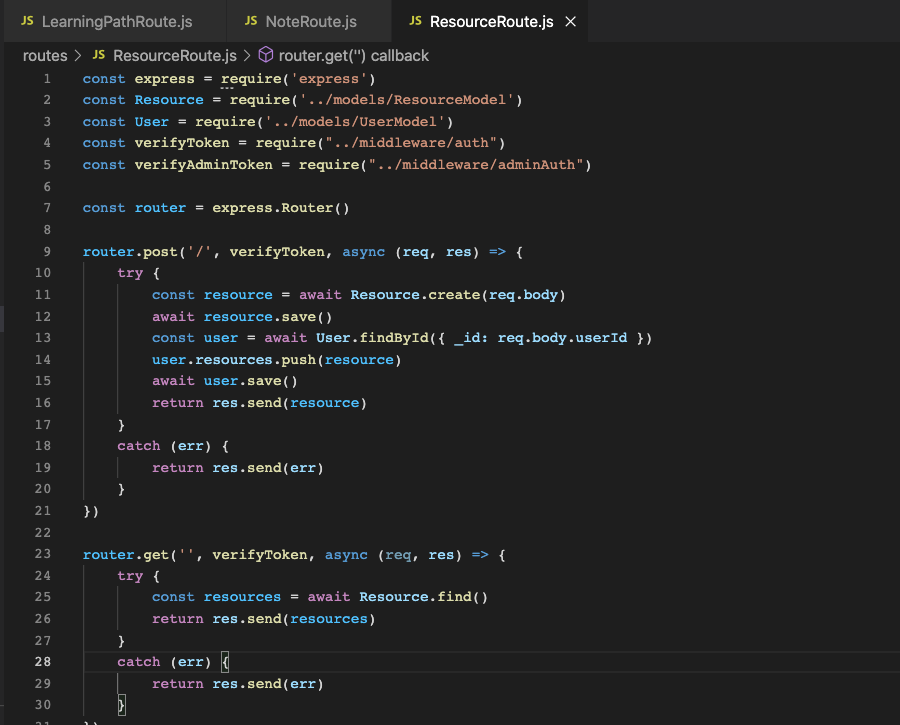
**Notes model:**

****

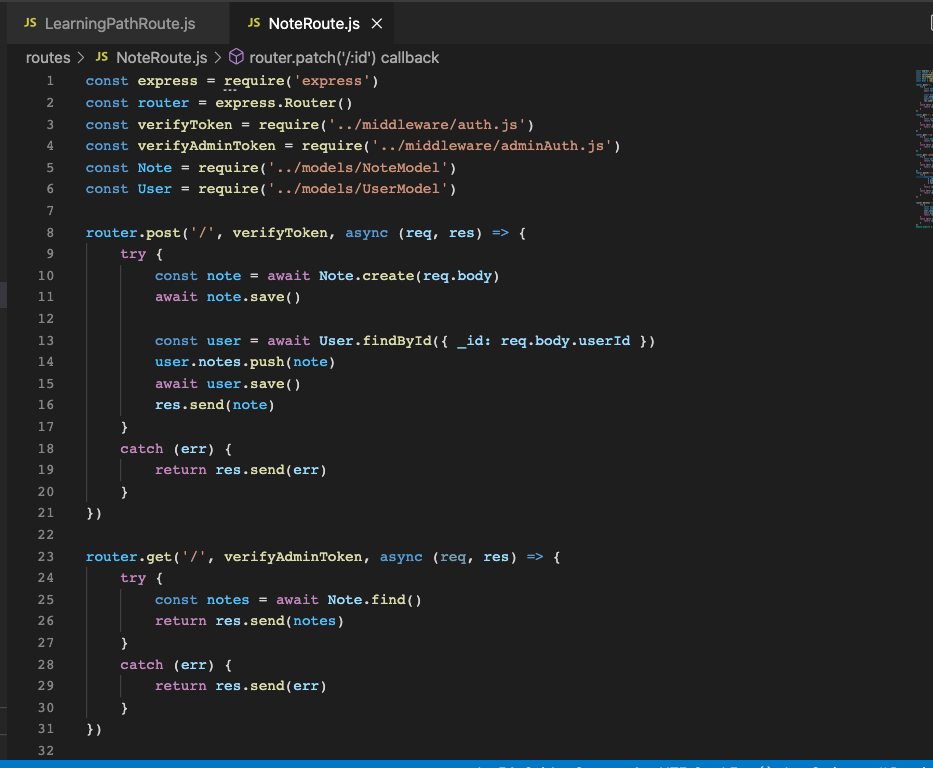
**Learning path model:**

**User route:**

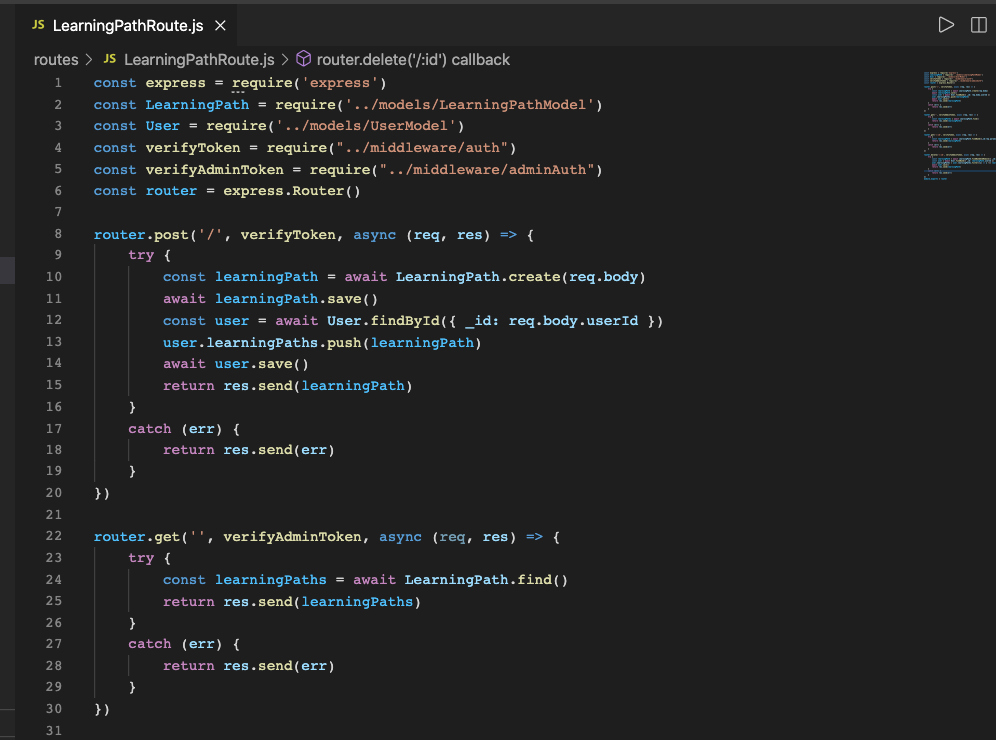
**Resource route:**

****

**Notes route:**

****

**Learning path route:**

****

**12. REFERENCES:**

* **<https://material.angular.io/>**
* **<https://angular.io/docs>**
* **<https://docs.mongodb.com/>**
* **<https://mongoosejs.com/docs/api.html>**