12-15-2022

Notebooks Web Application

Vivekananda Marellali
Grand Valley State University

Notebooks Web Application

Vivekananda Marellali

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing and Information Systems

December 2022

GRAND VALLEY STATE UNIVERSITY

The signatures of the individuals below indicate that they have read and approved the project of Vivekananda Marellali in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

___ _____12/20/22____

Project Advisor Date

_____ _____

Graduate Program Director Date

_____

Unit head Date

Abstract

In our day-to-day activities we often need to make quick short notes related to our work, study, or other activities. Later, those notes should be easy to access, modify, delete, share with others, or create PDF documents as and when required. It is ideal if the note content is rich text format such as HTML, as it provides many text formatting options and provide better view and readability. The Notes are better organized if they are grouped into notebooks where each notebook consists of related topic note pages and multiple users can create such notebooks and share with other users in the application as needed. Only the author of the notebook can edit the page contents and share with selected users. Each user should be able to see a listing of notebooks shared to them and a listing of their own authored notebooks separately. If all the pages in a finished notebook can be grouped together and printed to a PDF, it can then be shared over email or other medium outside of the application too. The above use cases provide the motivation for my project to develop a “Notebooks Web Application” with all the mentioned features.

Introduction

It’s a common practice to take notes and sharing to others as needed. Generally, people store such notes in plain text files on their local computers. The disadvantages of local text file notes are that they are not organized, tough to read and share, stored on a local computer, and can’t be accessed elsewhere. So, this project is developed to simplify the notes creation, formatting, improve their accessibility, enable sharing to others and address all issues with text file notes mentioned above. Web is the best place to host such application due to its accessibility and security. Grouping several notes that are related to same topic into notebooks. A database helps to store all the data in a centralized location and finally a sophisticated UI with rich text editor enables formatting of notes and readability. So, I named as “Notebooks Web Application”. Based on the above use cases, I determined that the application needs to have the following features -

- Allow users to Create Notes in HTML format in a Web App.
- Notes to be organized and grouped into "Notebooks" for better access
- Notes Content Format to be of Rich Text HTML for better view and readability
- Provide Rich-Text-Editor with many text formatting options including support to insert videos and images
- User Authentication
- Multiple Users Support
- Allows to Share Notebooks among other users in the application
- Easy way to Access/Modify/Delete/Share and Create-PDF of the Notebooks
- Access Control - Only Author/Owner allowed to Edit

Based on the features identified above, I then determined the below functional requirements for the application to achieve them.

- Sign up a New User to the Notebooks web application.
- Login into the Notebooks web application.
- Create a New Notebook
- Edit and Share a Notebook
- Delete a Notebook
- Share Notebooks to other users in the system.
- Create a New Page in a Notebook
- Edit a Page in a Notebook and add Rich HTML content in it
- Delete a Page in a Notebook
- Sort the order of Pages in a Notebook
- Print all Pages in the notebook to a PDF

The next steps are to identify the architecture of the major components/services of the application to be developed and their interfaces. Then identify the servers and technologies to use which are described in the following sections.

Project Management

As a first step as an application developer, I've filled out a project intake form with the Project advisor and get required approvals. The document serves several purposes - its primary function is to help me and the project advisors to clearly articulate what the project is to do. It also serves as a sort of contract, helping me control the scope of the project and defend against scope. Once we agreed on the parameters of the project, we agreed on a deadline.

Then I spent a few days to determine the components/services of the application and the interfaces between them. As I determined that I would need a Database, UI Framework and APIs components for the application, I researched for the latest and best technologies and chose the below Technical stack for the project

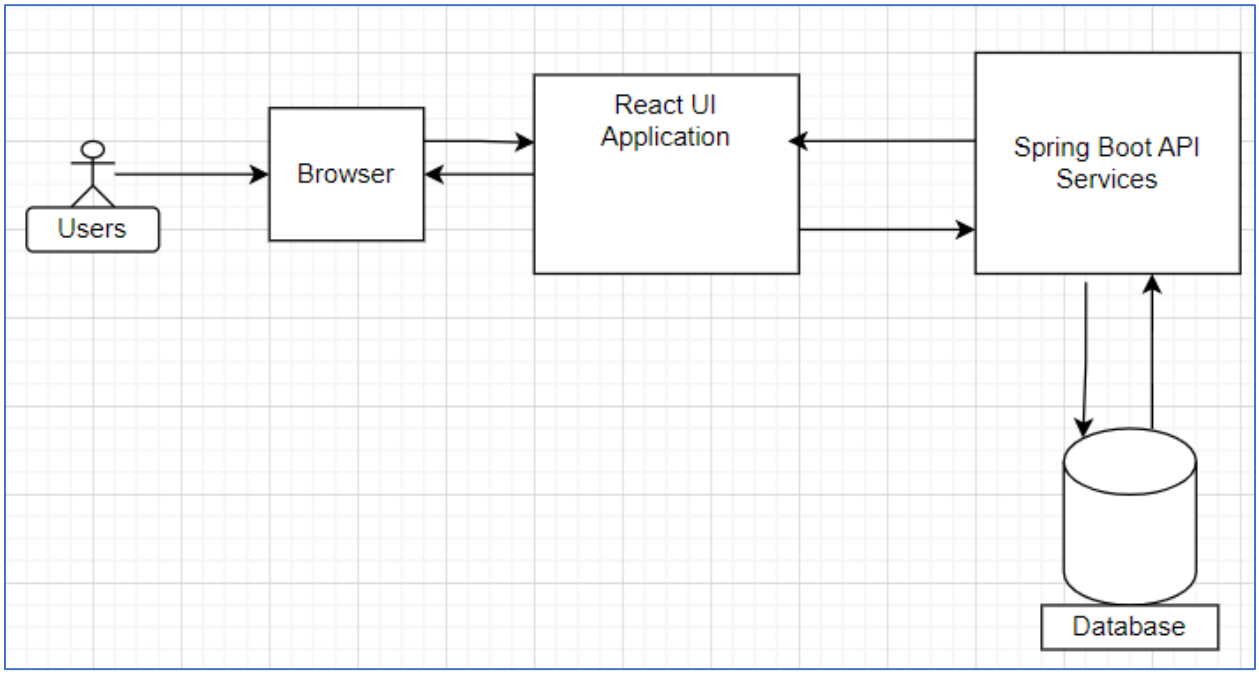
- Database – MongoDB - To store the notebooks and users data
- Middleware – Spring Boot REST Services – For CRUD data operations
- Frontend – React – User Interface for the application

For each of these of components I needed to spend time on the following analysis –

- MongoDB - Identify the collections/ data-fields/datatypes in MongoDB to store the data related to notebooks, note pages and users.
- Spring Boot REST Services – Identify the API Services required for each of the functionality and the individual parameters and return types for them. Here the Request Json and Response Json needed to be determined with some mock data.
- Frontend UI React application - Identify the UI layout, the routing URLs to Component mappings, the React Components and React External library module to perform various functions like calling the APIs, print react component content to PDFs etc.

After identifying the above required details, I moved to the implementation which involved the coding and testing phases which took few weeks of effort. After the first version of the application was finished, I got few improvement ideas which I implemented as well. Due to the complexity of some of the features and technologies involved, the scope of the project was increased, but overall, I was able to accommodate the changes while still staying on-schedule.

Organization



This project required three main components:

1) A front-end React UI Application that users can login to create/edit/share the notebooks. The react application calls the REST API Endpoints and consumes the JSON response data from APIs to render the UI components on the screen.
2) The backend Spring boot Rest Services application that receives the API Requests from frontend and is responsible to query the Mongo DB to fetch and return relevant data back to the UI App.
3) The MongoDB database that stores and returns the data related to the notes.

The Appendix section provides the details of each component.

Reflection

I learned a great deal about React, Spring Boot Rest Services and MongoDB NoSQL database by doing this project.

Firstly, I got good experience working on the React for the frontend application. I was particularly surprised by how complex the React Routing and Components Mappings are. Each route should render/update only the required components based on Route URL, State and Properties. Designing the React UI Layout was another challenging part as the UI should stay consistent for all the functionalities/screens for better user experience. It was interesting to explore the various UI Icons required for the various functionality links. The react icons module (https://react-icons.github.io/react-icons/) provides vast number of icons to choose from and import into react components. Another major work is to research and choose the required library modules from the vast react node modules available in marketplace. Generally there are more than one module for the same requirement, so we need to explore, read the pros and cons of each of them and choose the best suited for our application. Finally found it tricky when components need to communicate with each other. When an application is made up of many nested components, as this one is, passing down state data from one component to another is extremely fiddly. One typo in one place can result in data not being displayed properly or at all. I spent a lot of coding time chasing the errors due to Re-renders of components due to UI events and routing.

Secondly, I got good hands-on experience designing and implementing Rest Services using spring boot. The Swagger-UI application provided with a dependency in the Spring boot was very useful to design and debug the APIs. I also got good knowledge on the Spring Security and OAuth Token based authentication which was implemented for this application for User

login functionality. It was amazing to work on the Spring data JPA for database persistence and data retrieval.

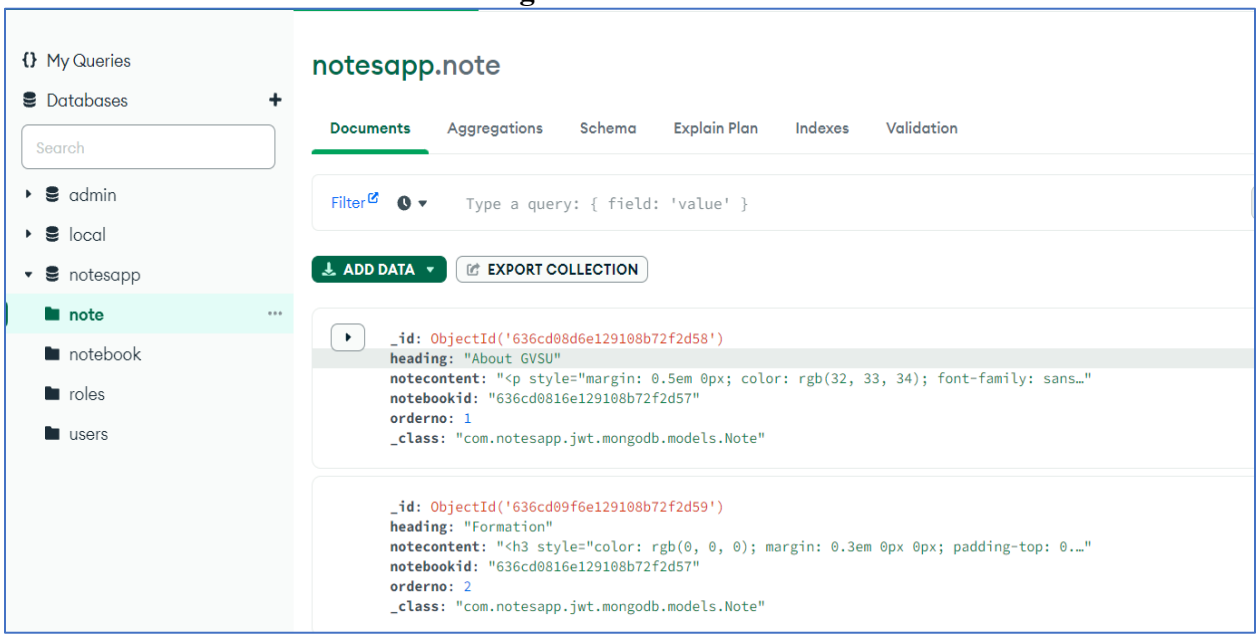
Finally, as I am new to NoSQL database, it was an interesting and a good learning experience working on MongoDB. One of the main benefits of a NoSQL database is the Ease of updating of schemas and fields due to unstructured data. I could leverage this benefit few times as I had redesigned the collections by adding and remove fields, without much effort. I found MongoDB extremely easy to develop when compared to SQL-databases.

Conclusions

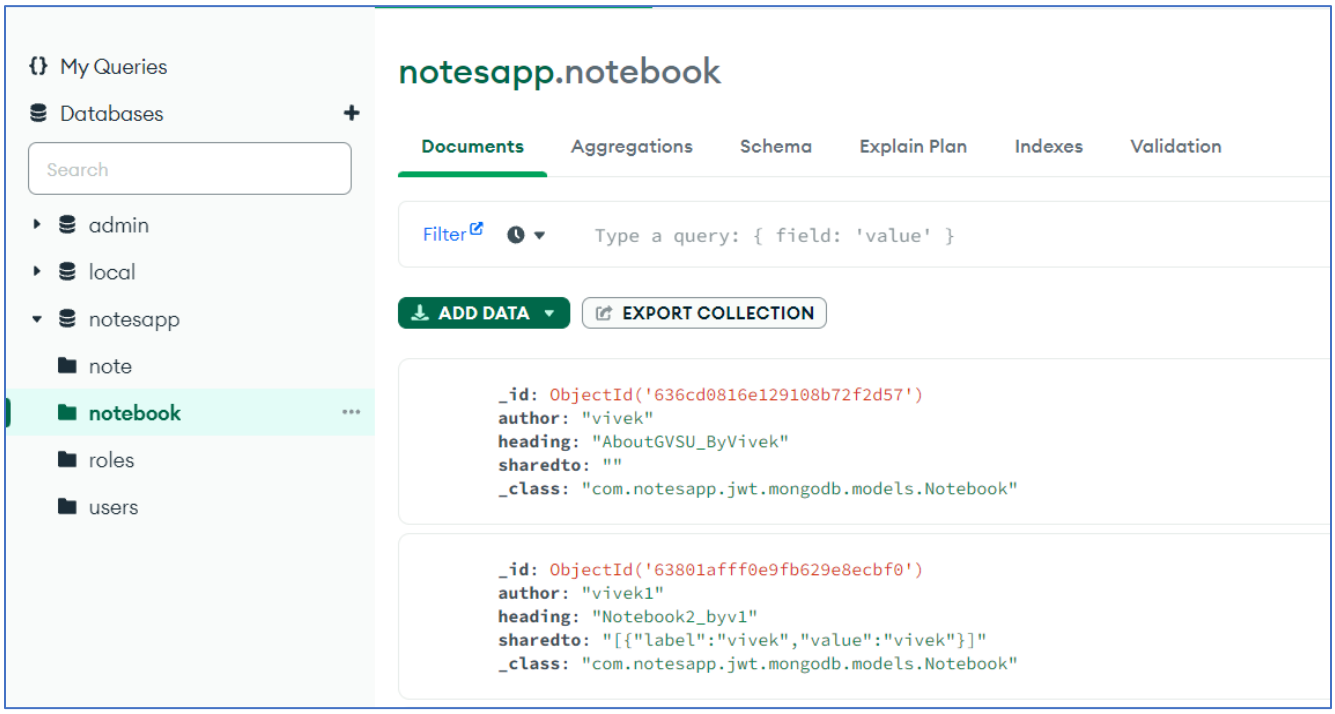
The final version of “Notebooks Web Application” was completed on planned schedule and I received positive feedback when I showed a demo to my Project advisor. I also prepared presentation slides and video presentation for the same. Although there are no open issues I noticed in the application, I see that there is a scope to improve the performance of the application by leveraging caching of data on the client side and reducing the API Calls. For directions for future work, I can think of few additional features that can be added in the future versions of this app such as adding search feature to search text within the notebook pages and return the matching notebooks, adding support for storing versions of PDFs in the system for future access, creating of index page in the notebook with page numbers etc.

Appendix 1: Database Mongodb Collections

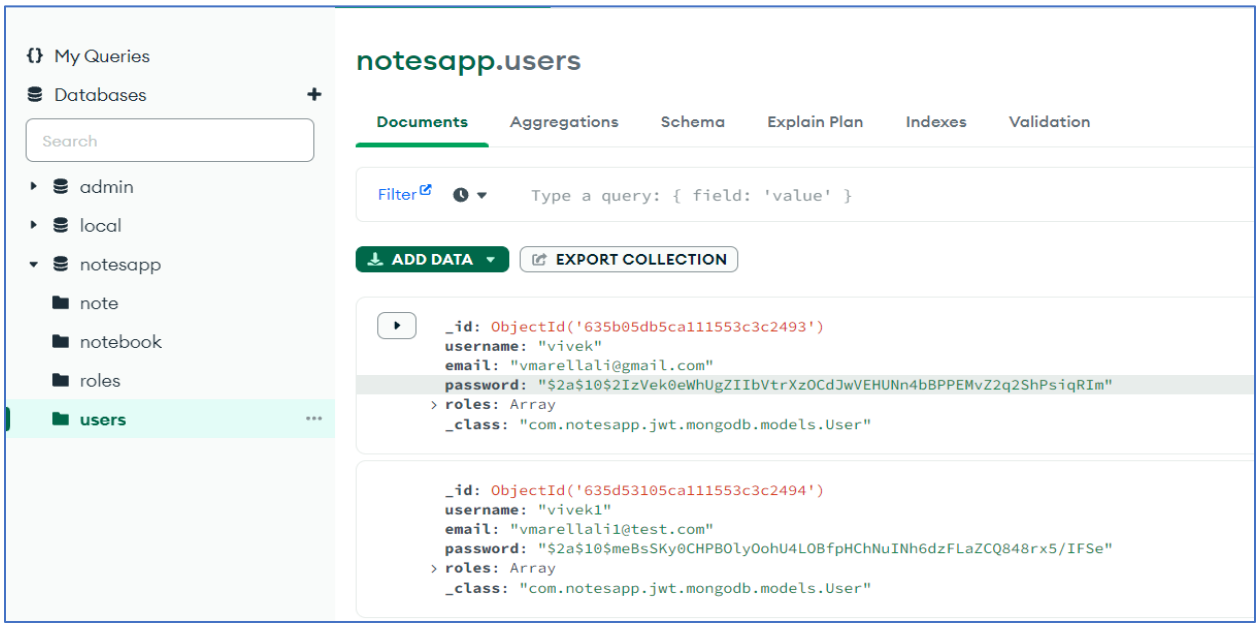
- **Collection Name: Note**
  - Stores the data related to "Pages"



- **Collection Name: Notebook**
  - Stores the data related to "Notebooks"

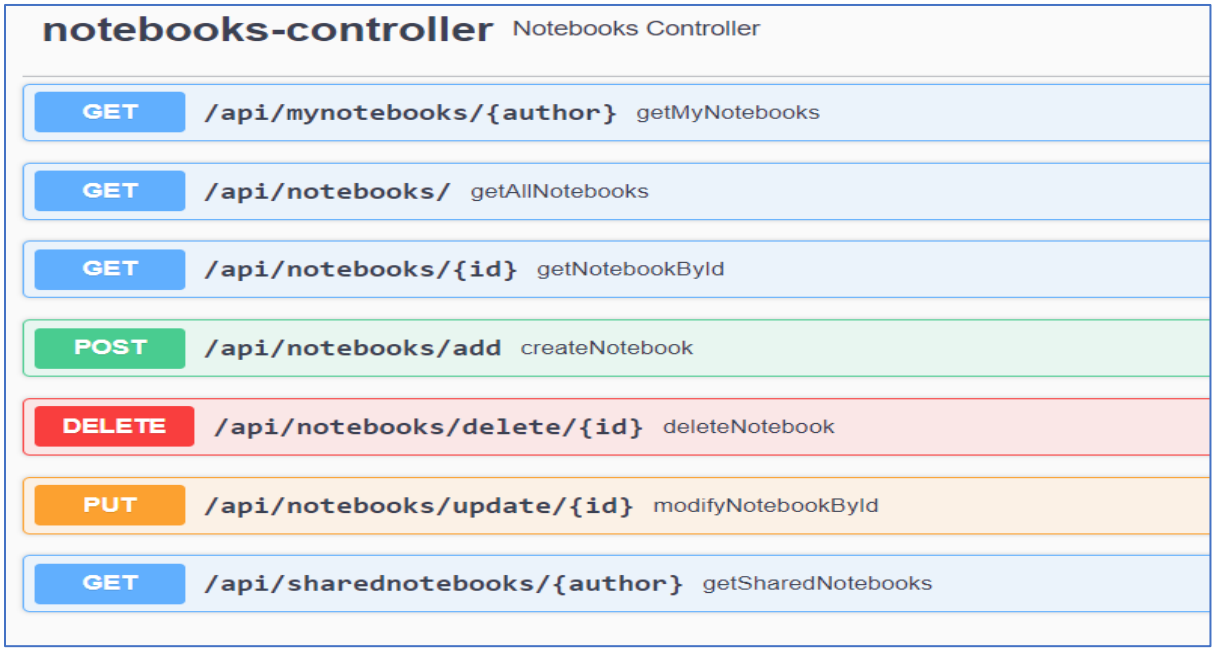


- **Collection Name: Users**
  – Stores the data related to "Users"

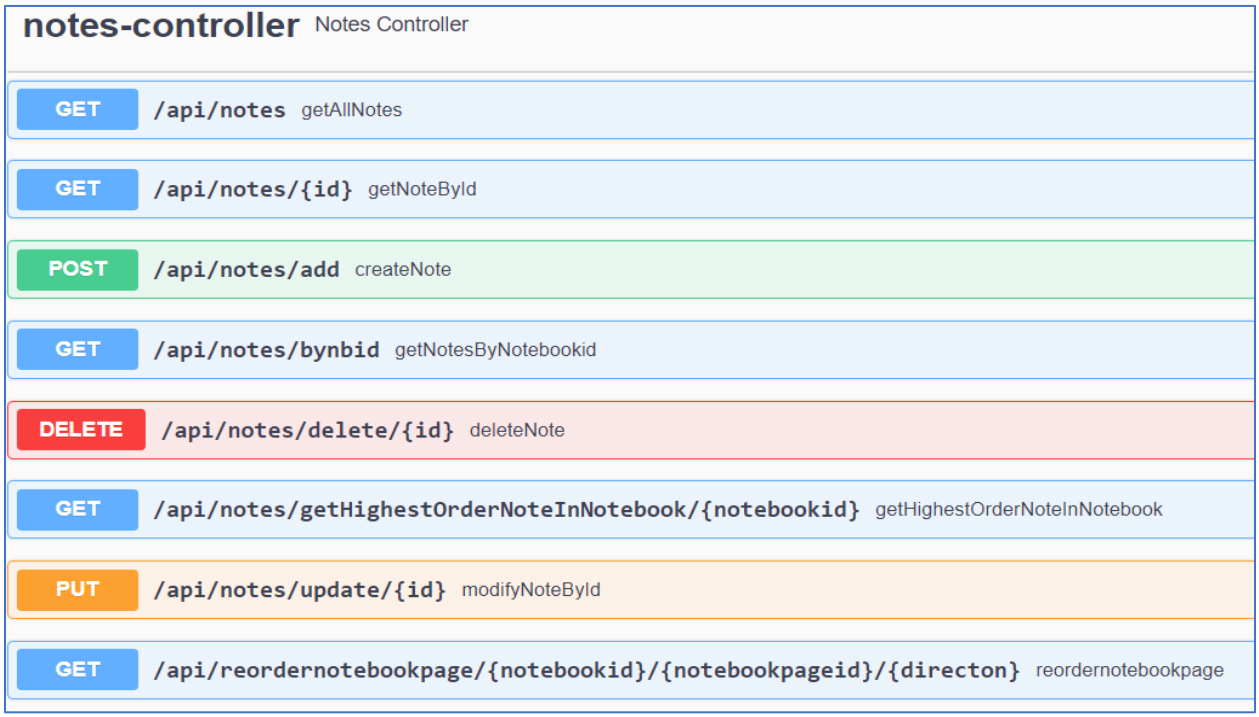


Appendix2 - Spring Boot Rest Services

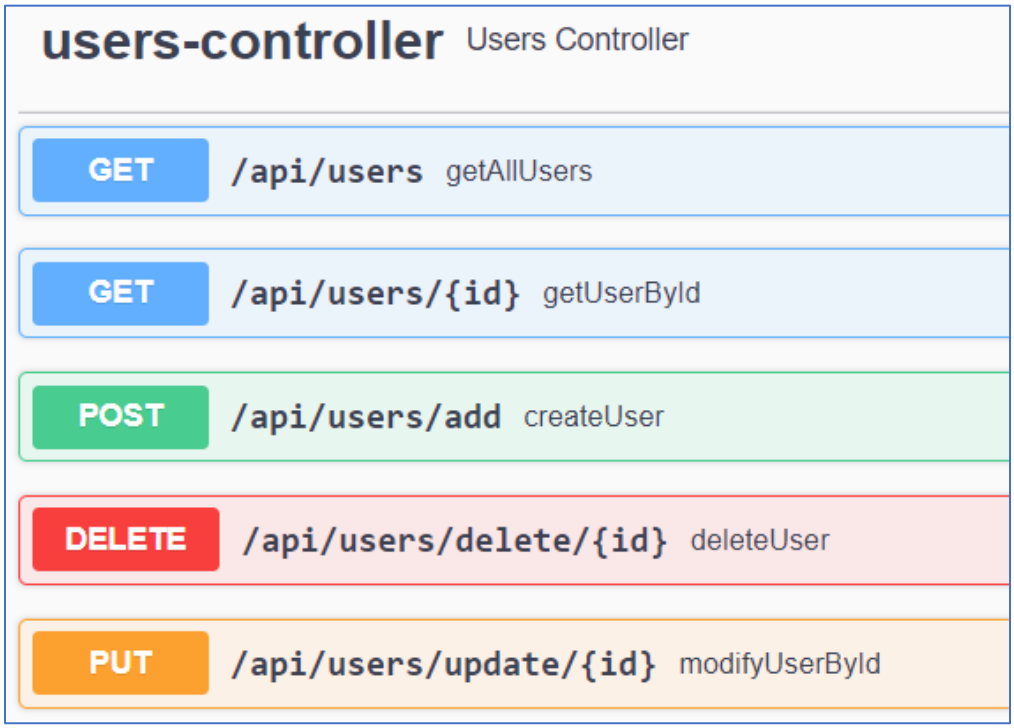
- **Notebooks Controller APIs**



Notes Controller APIs



Users Controller APIs



Appendix3 React UI Components List

- contentarea.component.js
- leftnavarea.component.js
- addnotebook.component.js
- addnotebookpage.component.js
- deletenotebook.component.js
- deletenotebookpage.component.js
- displayallnotepages.component.js
- displaynote.component.js
- displaynotebookpages.component.js
- displaynotebooks.component.js
- editnotebook.component.js
- editnotebookpage.component.js
- headerarea.component.js
- login.component.js
- profile.component.js
- register.component.js
- reordernotebookpage.component.js

React External Library Modules List

REACT MODULE NAME	PURPOSE
React-router-dom	Mapping of Route URLs to Components for rendering
bootstrap@4.6.2	Styling of components
axios	Calling the REST API and receiving data from them.
React-validation validator	Form Validations

React-icons	**Rendering various Icons**
Jodit-react	**Rich Text Editor**
React-select	**Multiselect Dropdown**
React-to-print	**Print Components to PDF**