



ShareNITT

A easy role based file sharing system for community needs

Our Team

Prem Ranjan 106121094

Purav Shah 106121098

Rogeshwaran 106121106

Overview

ShareNITT is a digital file sharing system that can provide role based file access and sharing functionalities within a community (here based on NITT).

The following project emphasizes the use of different databases available in the market such as

- 1) MongoDB
- 2) MySQL
- 3) Firebase

Goals

1. Creating a community focused file sharing system that allows role based file sharing. (Here its **Student** and **Professor**)
2. A possible application is for sharing assignments and exam question papers with a select group of students in a secure way.
3. Establish and integrate various databases for different functionalities of the software
4. Providing an intuitive and easy user experience by functional front end.

Technical Specifications

- **React**

Router-Dom For Secure routing

React-Bootstrap For front-end UI

- **Docker** Dockerized application for easy hosting and handling without local dependency setup such as SQL.

- **Node** Primary Server

- **Mongodb** File Storage and access

- **SQL** File access and permissions
- **Firebase** Authentication and user detail storage backend.
- **Axios** A promise-based HTTP library that lets developers make requests to either their own or a third-party server to fetch data

User Flow

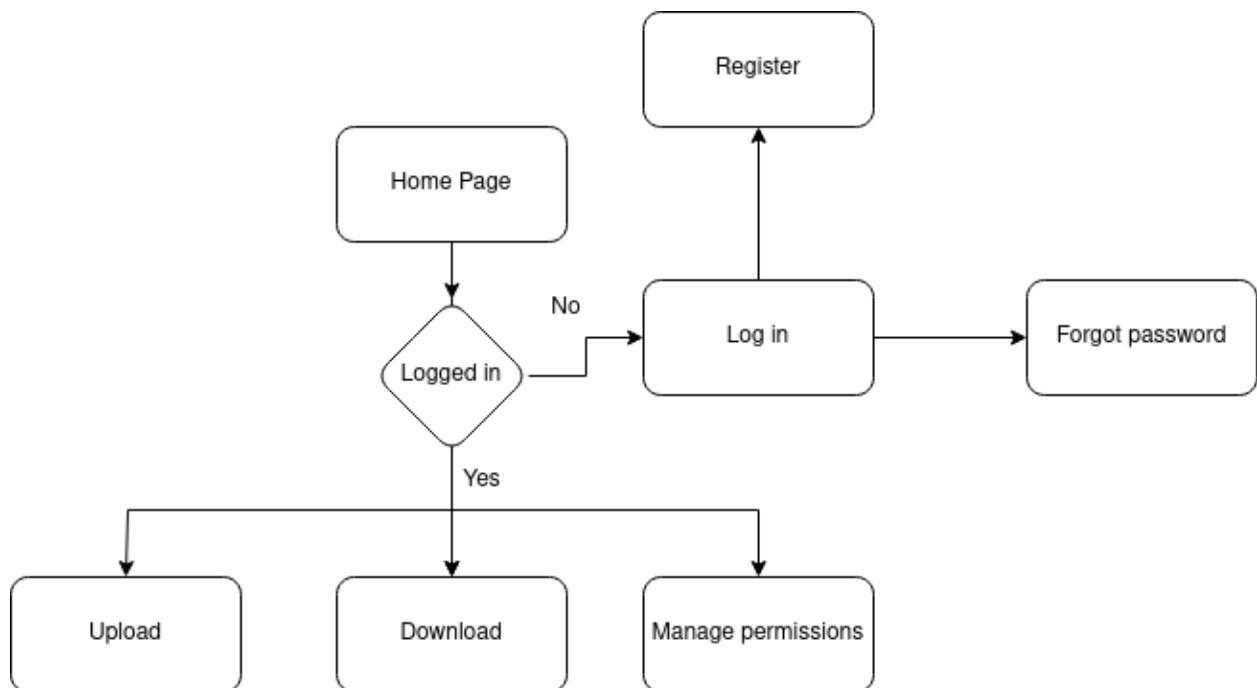
When a user is on the home page, they need to login to use any of the features. This is done to ensure the usage of the platform can be monitored if needed.

Once a user is logged in, if they are a student, they can either upload or download a file.

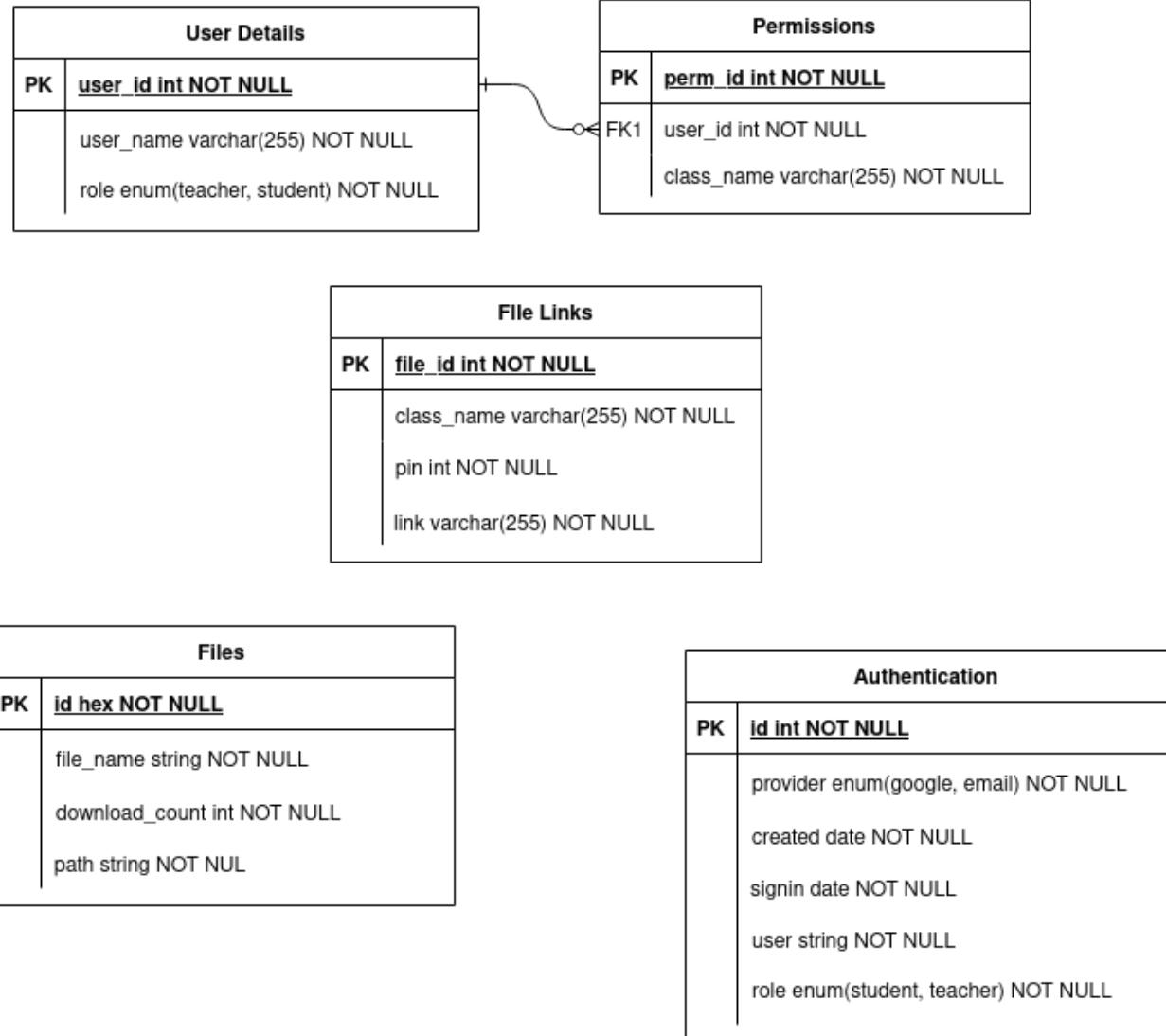
If they are a professor, they can upload, download or create groups. Groups are a set of students who can access the files shared with them.

Although, for making the process secure, along with being a member of the group, one also needs a numeric pin to access the file. If they don't have the pin, even though they are members of the group, they can't access the file.

One possible use of this in college is for sharing assignments or question papers (regular CT, end semester exams or even Compensation) in a secure way so as to not let the question papers be visible to any unintended person.



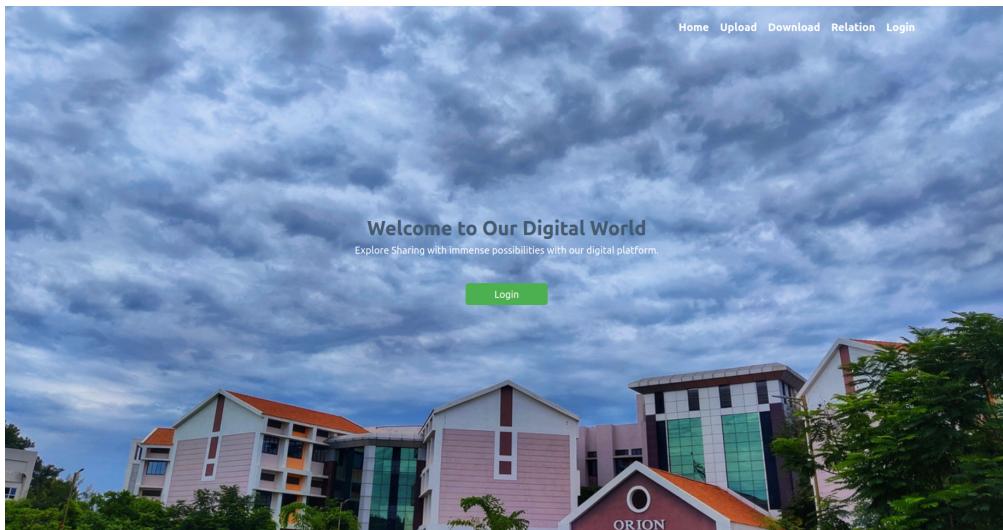
ER-Diagram



Routes

I. Home

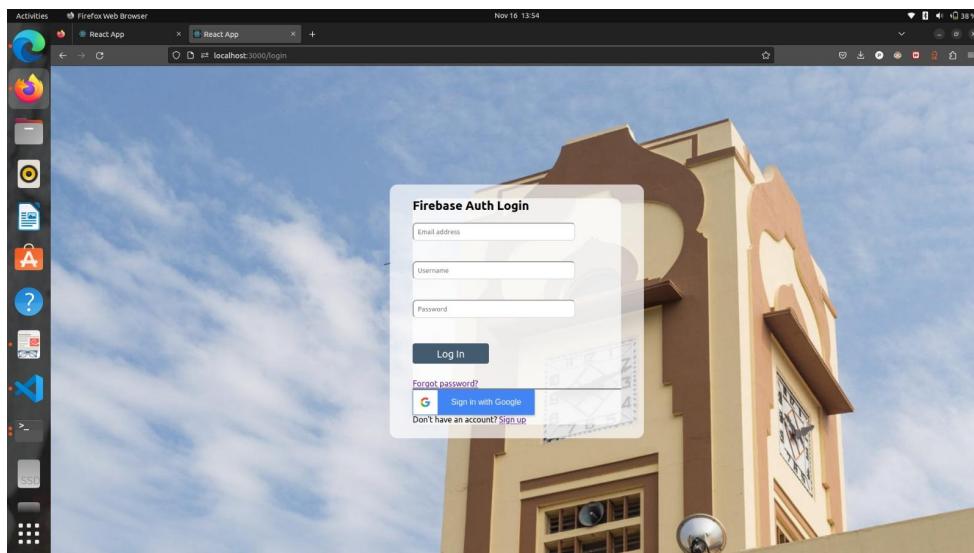
The home route consists of primary options such as login and various other options that are available for a user irrespective of roles.



II. Login

The login route takes to the login page which provides necessary fields of entry to login as a user.

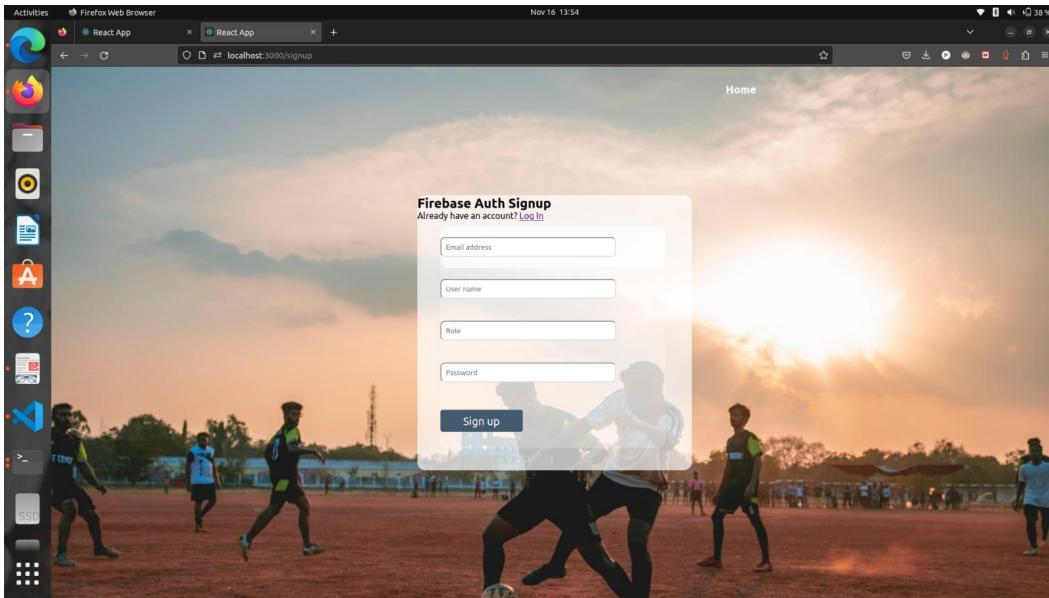
- Fetches the user data from firebase and stores it in local storage.
- Navigates to different pages according to roles.



III. Sign Up

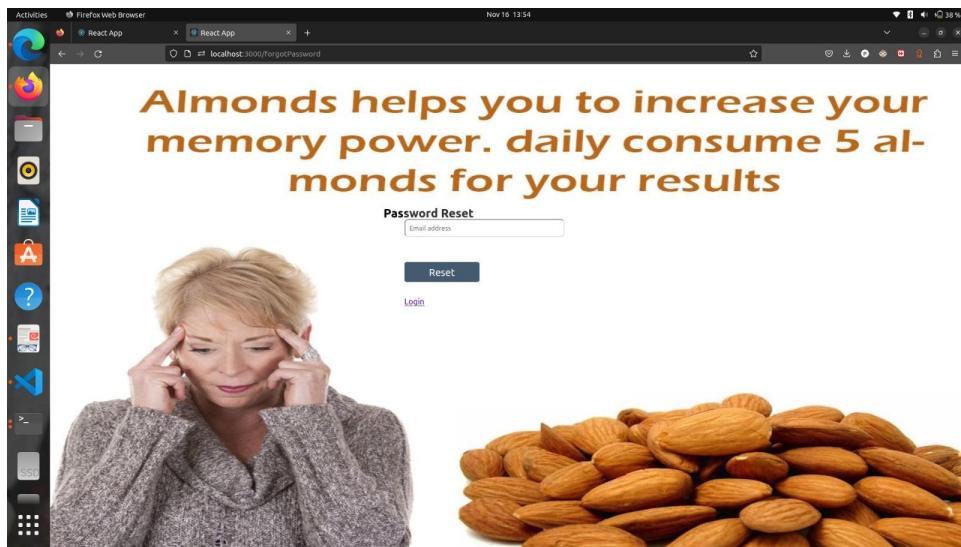
The SignUp route takes the user to the SignUp page which provides necessary fields of entry to SignUp as a user.

- Stores the user details into the firebase realtime database.



IV. Forgot Password

The Forgot Password route takes the user to the forgot password section where the user can provide the email to get a mail for password reset.

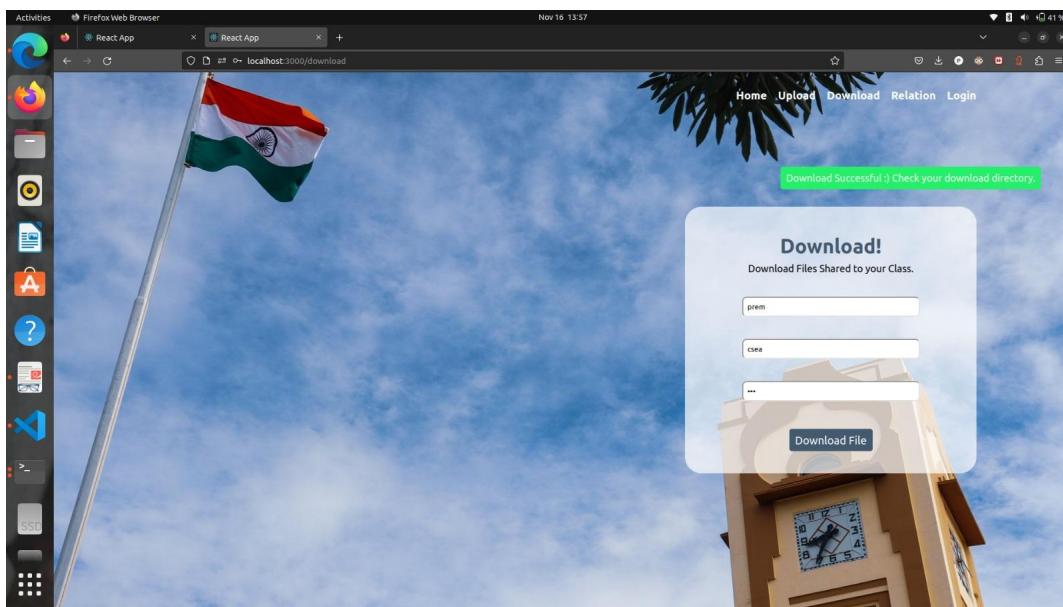
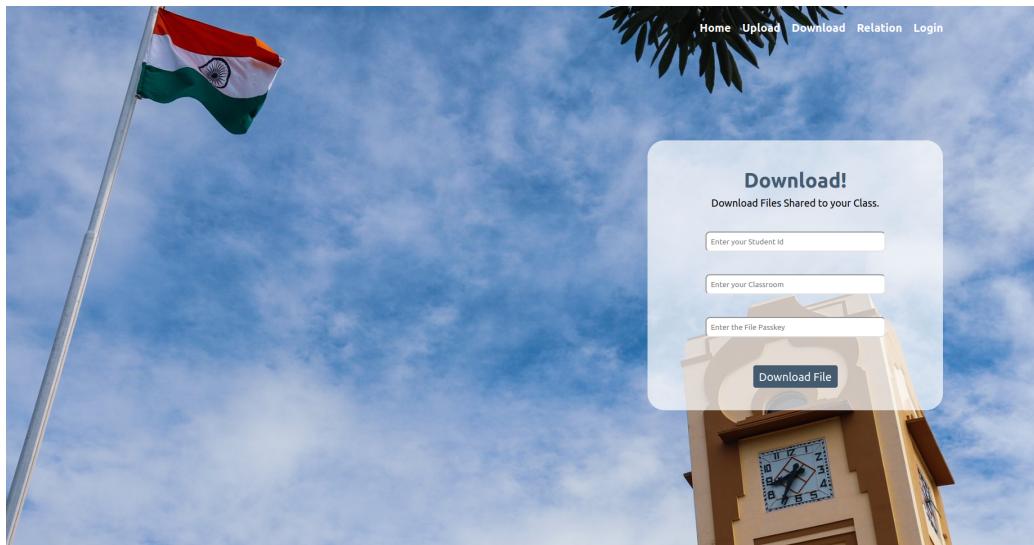


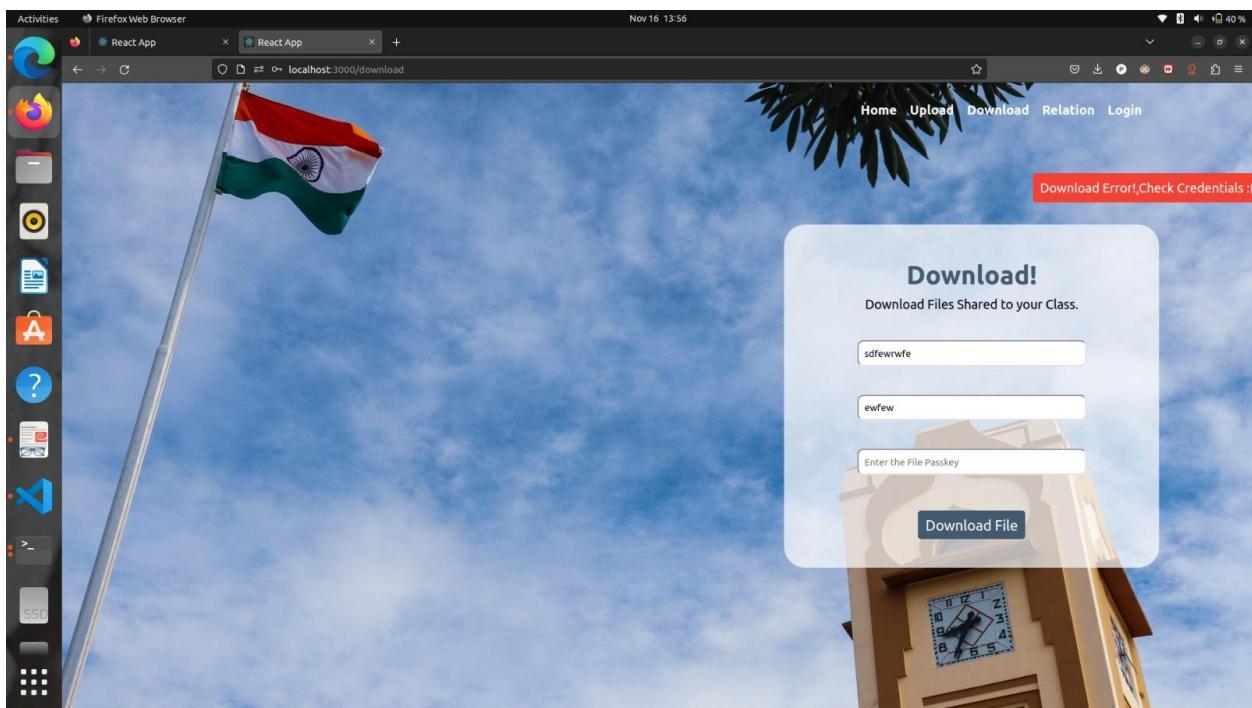
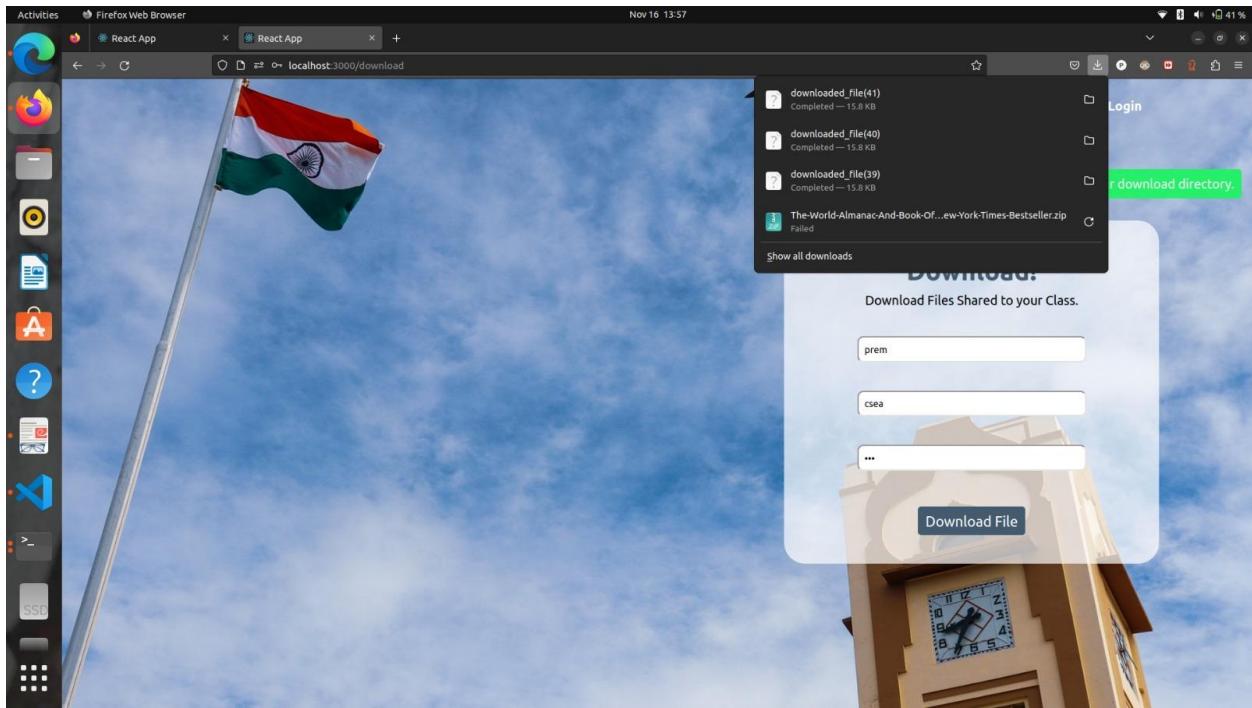
V. Download (Protected route - Anyone can access only after they are logged in)

The users are prompted to fill a form that consists of

- Student ID
- File passkey
- Classroom

Upon entering the data and submitting we provide the file in downloads.



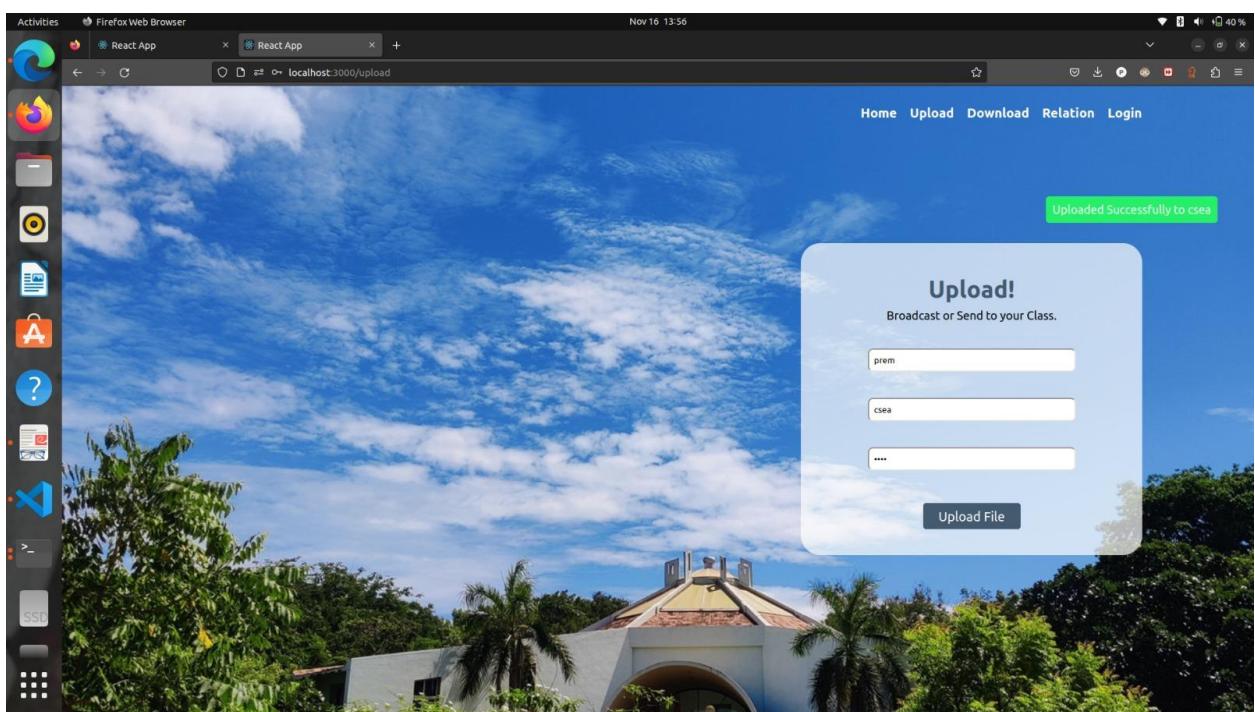
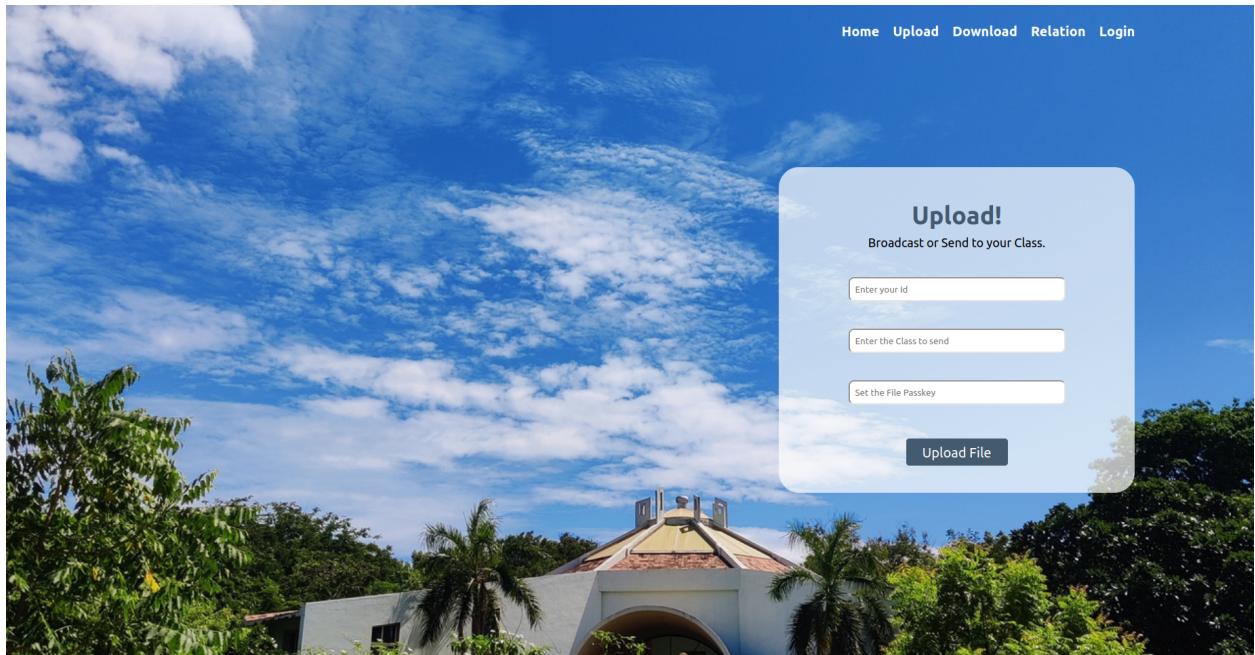


VI. Upload (Protected route - Anyone can access only after they are logged in)

The users are prompted to fill a form that consists of

- Student ID
- File passkey
- Classroom

Upon entering the data and submitting it, the file is uploaded in mongoDB.

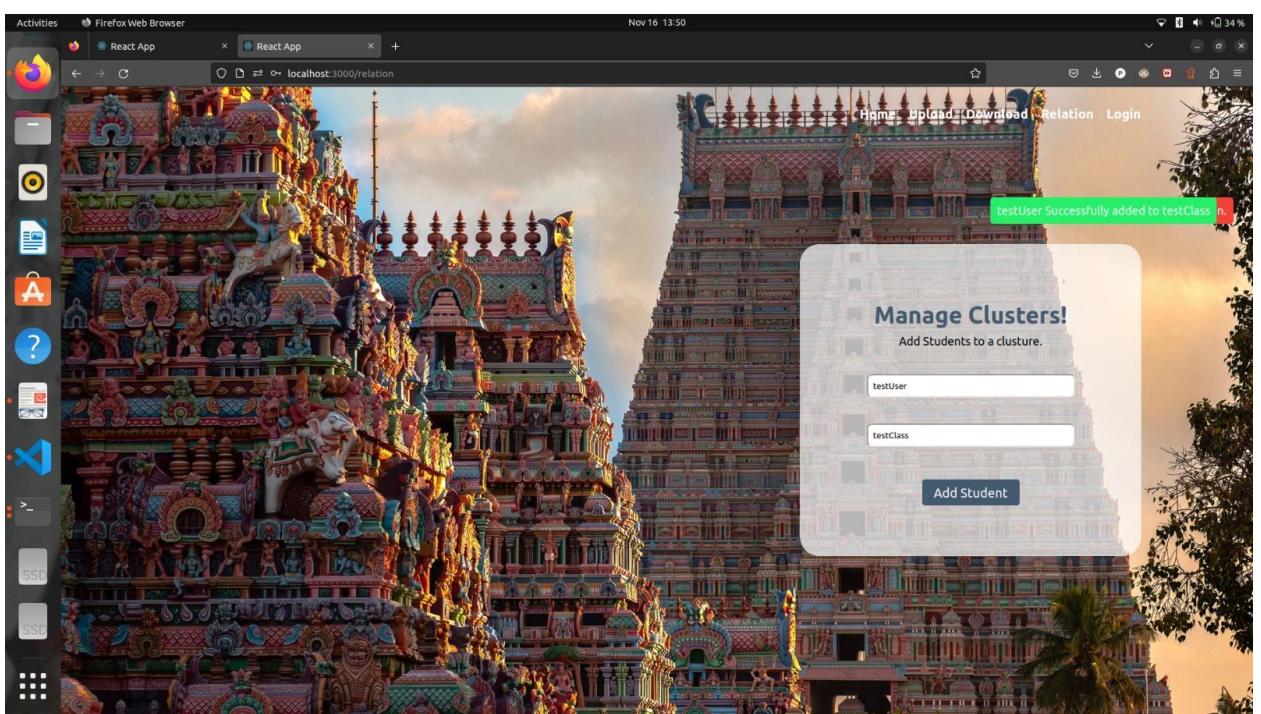
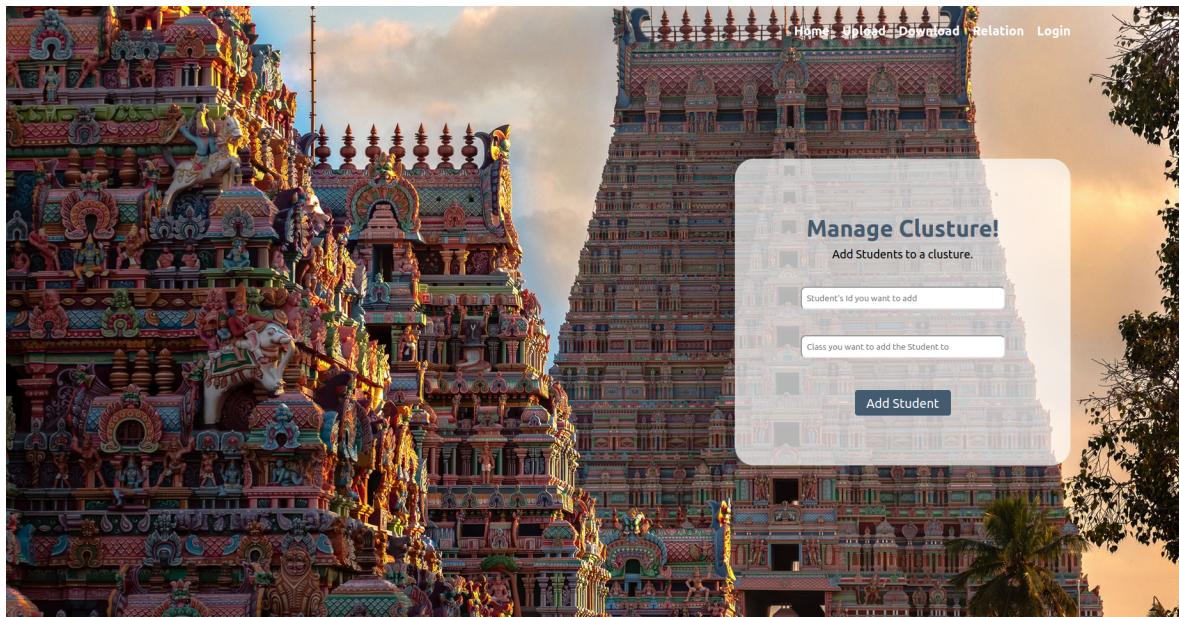


VII. Relation (Protected Route - Can be accessed only by a Professor)

The users are prompted to fill a form that consists of

- Student ID
- Classroom (cluster)

Upon submission the specific Student is added to the specific class through mapping in mySQL



App Testing

The Testing was conducted through microsoft lighthouse

