

Project Report On



Public Drive

Submitted in partial fulfillment for the award of
Post Graduate Diploma in Advanced Computing from
C-DAC ACTS (Pune)

Guided by

Mr.

Presented By

Ayush Chandra – 240840120043

Aditya Nikhate - 240840120009

Rashi Rajesh Patil – 240840120132

Pranjal Nehete - 240840120116

Raj Pandya – 240840120129

Centre of Development of Advanced Computing (C-DAC), Pune



CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Ayush Chandra - 240840120043

Aditya Nikhate- 240840120009

Rashi Rajesh Patil - 240840120132

Pranjal Gopal Nehete- 240840120116

Raj Pandya- 240840120129

have successfully completed their project titled

“Public Drive”

Under the Guidance of Mr.

Project Guide

HOD



ACKNOWLEDGEMENT

This project “Public Drive” was a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS).

We all are very glad to mention the name of Mr. for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

Our most heartfelt thank goes to Ms Swati mam (Course Coordinator, PGDAC) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS, Pune.

Ayush Chandra -240840120043

Aditya Nikhate - 240840120009

Rashi Rajesh Patil - 240840120132

Pranjal Gopal Nehete – 240840120116

Raj Pandya - 240840120129

TABLE OF CONTENTS

1. Introduction
2. Software Requirement and specification
3. Tools and technologies used
4. Project Flow Diagram
5. ER Diagram
6. Advantages
7. Screenshots
8. Future Scope
9. Conclusion
10. References

1. Introduction:

PublicDrive is a distributed enterprise web application designed to enable users to securely upload, share, edit, and download files while managing their access through role-based permissions. Built using React.js for the frontend and Spring Boot with JWT authentication for backend security, the application ensures safe and seamless file handling. Users can upload files in various formats, share them with specific individuals or groups, modify supported files within the platform, and manage their own activity timeline. Additional features include starring/un-starring files, creating groups, and managing group members. The backend relies on MySQL for database management and a server-based file system for storage. Role-based access control restricts user actions based on permissions, while audit logs enhance security by tracking user activity. The project is aimed at providing hands-on experience in developing enterprise applications using modern technology trends like Spring Boot.

2. Software/Hardware Requirement

Server:

Processor: Intel Core i5 or equivalent AMD processor.

RAM: Minimum 8GB RAM.

Storage: SSD storage for improved performance.

Network: Ethernet or Wi-Fi connectivity.

Operating System: Linux distribution (Ubuntu, CentOS) preferred for server deployment.

Client Devices:

Processor: Dual-core processor or higher.

RAM: Minimum 4GB RAM.

Storage: Sufficient storage for caching and local data.

Network: Ethernet or Wi-Fi connectivity.

Browser: Compatible with latest versions of popular browsers like Google Chrome, Mozilla Firefox, and Safari.

3. Tools and technologies used

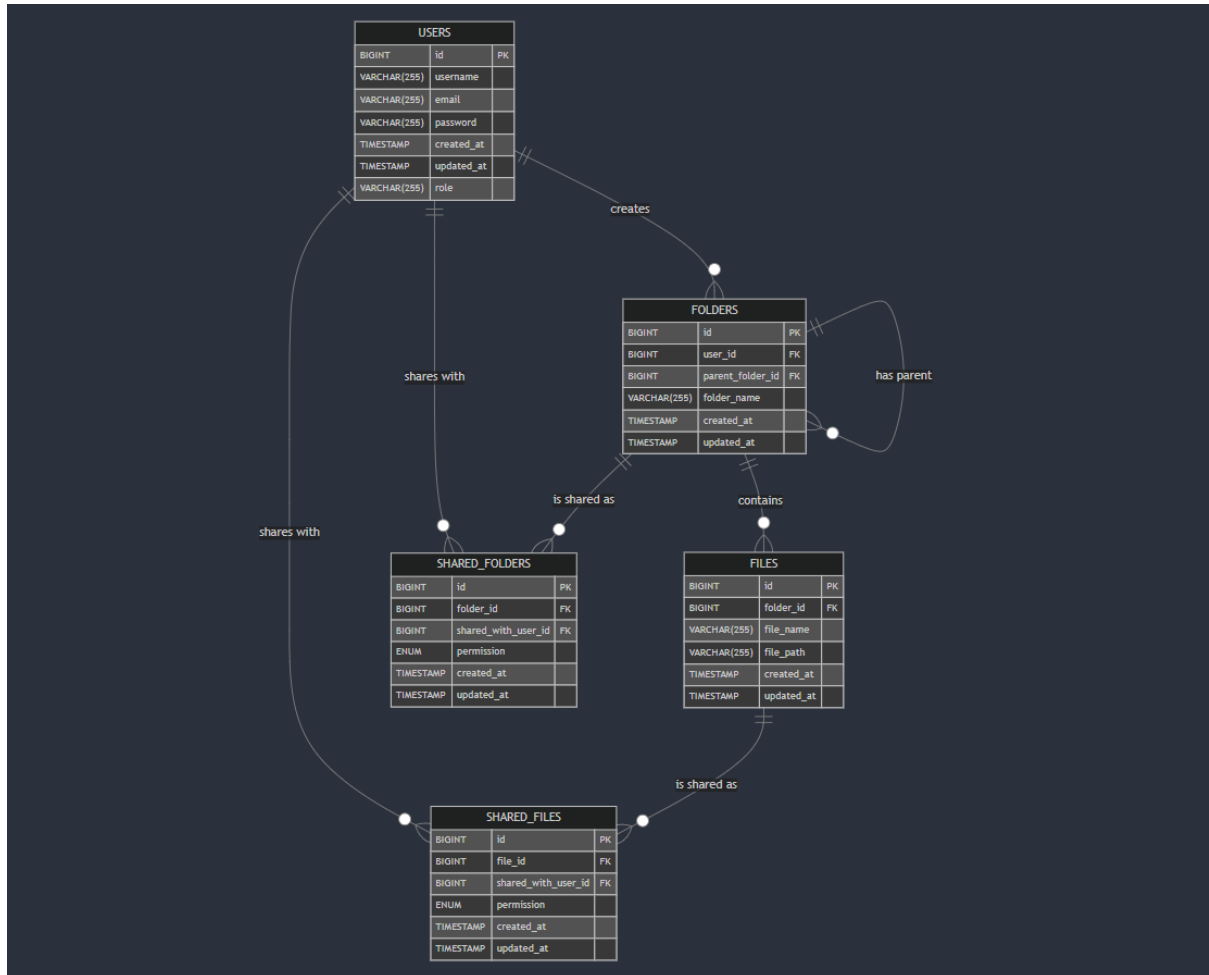
- SpringBoot
- SpringDataJPA
- RESTful Web
- Node JS
- Express JS
- SpringWeb
- MYSQL Database
- JWT
- Git
- Spring Security
- React JS
- HTML and CSS
- Axios
- Material UI

1. Spring Boot: Utilized to develop the backend of the application, providing a robust framework for building Java-based web applications with ease.
2. Spring Data JPA: Implemented for data access, allowing seamless interaction with the MySQL database to store and retrieve sports data efficiently.
3. RESTful Web Services: In the context of an e-commerce web application like Book Charm, RESTful web services play a crucial role in facilitating communication between the frontend and backend components. These services adhere to the principles of Representational State Transfer (REST), which emphasizes a stateless, standardized approach for building web services
4. Node JS: While the core of PublicDrive is powered by Spring Boot for secure file management and API services, Node.js can be optionally employed for specialized background tasks. For instance, Node.js might be used to handle asynchronous operations such as processing file metadata, generating file thumbnails, or managing real-time notifications. This modular approach allows for leveraging Node.js's non-blocking architecture for tasks that require rapid, concurrent processing, complementing the robust, secure operations handled by Spring Boot.
5. Express JS: Express.js is a web application framework for Node.js that simplifies the creation of robust, scalable APIs and web applications by providing a set of middleware

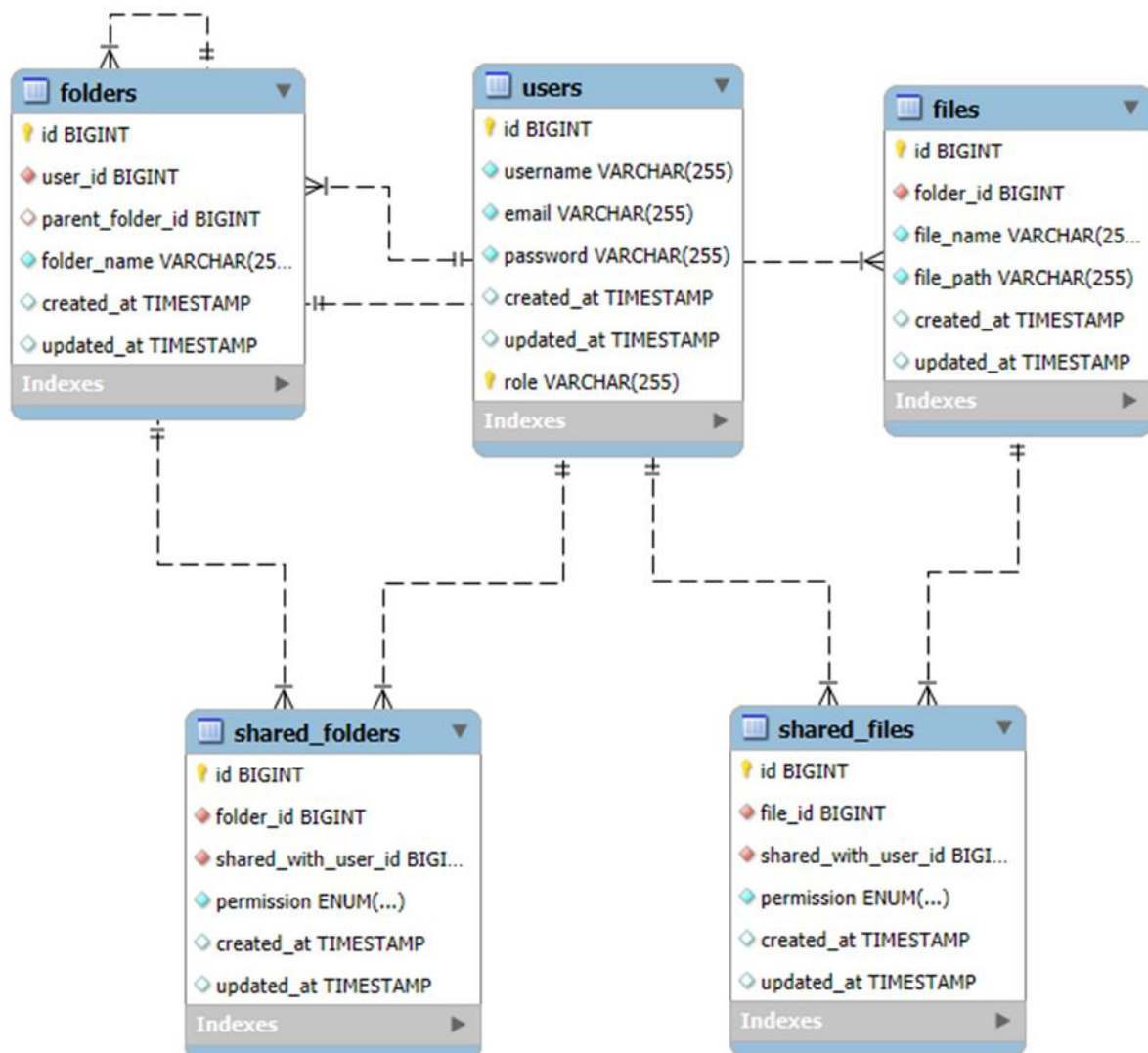
and routing mechanisms. It streamlines the process of handling HTTP requests, making it efficient for building server-side components in a Node.js application.

6. Spring Web: Used for handling web requests and responses, managing controllers, and serving static resources to the frontend.
7. MySQL : Chosen as the relational database management system, MySQL is used to store critical data on the cloud for PublicDrive. This includes user details, file metadata (such as file names, types, paths, upload dates), and audit logs. Its reliability and ACID compliance ensure that all data remains consistent and secure, which is essential for a robust file management system.
8. JWT (JSON Web Tokens):
JWT is implemented in PublicDrive for secure user authentication and authorization. This token-based mechanism ensures that only authenticated users can log in, manage their files, and share documents. Additionally, JWT supports role-based access control (RBAC), ensuring that users only perform actions permitted by their roles, thereby maintaining the overall security of the application.
9. Axios: In PublicDrive, Axios is used as the client-side HTTP library within the React.js frontend. It simplifies making asynchronous HTTP requests to the Spring Boot backend, facilitating smooth communication between the two layers. Axios handles tasks such as fetching file metadata, managing file uploads/downloads, and processing user actions, ensuring that data retrieval and interactions are efficient and responsive.
10. React: Employed to build the frontend of the application, offering a component-based architecture for creating dynamic and interactive user interfaces.
11. CSS: Used for styling the frontend components with utility-first CSS classes, allowing for rapid prototyping and customization of the user interface.
12. Material UI: Leveraged to enhance the visual appeal and user experience of the application by incorporating pre-designed React components following Google's Material Design principles.
13. Git: Implemented as a version control system to track changes in the source code, enabling collaboration among developers, and facilitating code management and deployment workflows.

4. Project Flow Diagram



5. Project E-R(Entity relationship) Diagram



6. Advantages

Frontend: React.js

- **Efficient Rendering:**
 - React's Virtual DOM ensures efficient updates and fast rendering, leading to a smooth user experience.
- **Component-Based Architecture:**
 - Encourages reusability and easier maintenance by breaking the UI into modular components.
- **Rich Ecosystem:**
 - A vast library of tools, extensions, and community support simplifies development and troubleshooting.
- **Responsive Design:**
 - Facilitates building dynamic and responsive interfaces, ideal for a file management system.

Backend: Spring Boot with JWT Authentication

- **Rapid Development:**
 - Spring Boot simplifies configuration and setup, enabling you to quickly build robust RESTful APIs.
- **Integrated Security:**
 - JWT provides stateless, secure authentication, ensuring that only authorized users can access and perform file operations.
- **Scalability and Robustness:**
 - Designed for enterprise-level applications, Spring Boot can handle increased loads while maintaining performance.
- **Modular Design:**
 - Promotes a clean separation of concerns, making it easier to implement features like role-based access control (RBAC) and audit logging.

Database: MySQL

- **Reliability and Performance:**
 - MySQL is a proven relational database that efficiently manages structured data, such as user credentials, file metadata, and audit logs.
- **ACID Compliance:**

- Ensures data integrity and consistency, which is critical for secure file management systems.
- Wide Adoption and Support:
 - Extensive community and documentation make it easier to troubleshoot and optimize database operations.

File Storage: Server-Based File System

- Simplicity:
 - Using a server-based file system for file storage is straightforward to implement, making it easier to manage files directly.
- Security:
 - Controlled access to files can be managed alongside the application's security measures, ensuring secure file storage.
- Direct Integration:
 - Seamlessly integrates with the backend to provide fast read/write operations required for file handling.

Overall Tech Stack Benefits

- End-to-End Security:
 - The integration of JWT and RBAC across the tech stack ensures that every layer, from user interface to data storage, maintains high security standards.
- Performance & Scalability:
 - Each component is designed to handle large volumes of data and concurrent users, ensuring the system can scale as needed.
- Ease of Maintenance & Future Enhancements:
 - With a modular architecture, it's simpler to add new features or upgrade parts of the system without significant overhauls.
- Community and Ecosystem Support:
 - All components of the stack are well-documented and widely adopted, making it easier to find resources, libraries, and community assistance when needed.

These advantages combine to create a robust, scalable, and secure file management system that leverages modern enterprise technologies effectively.

7. Screenshots

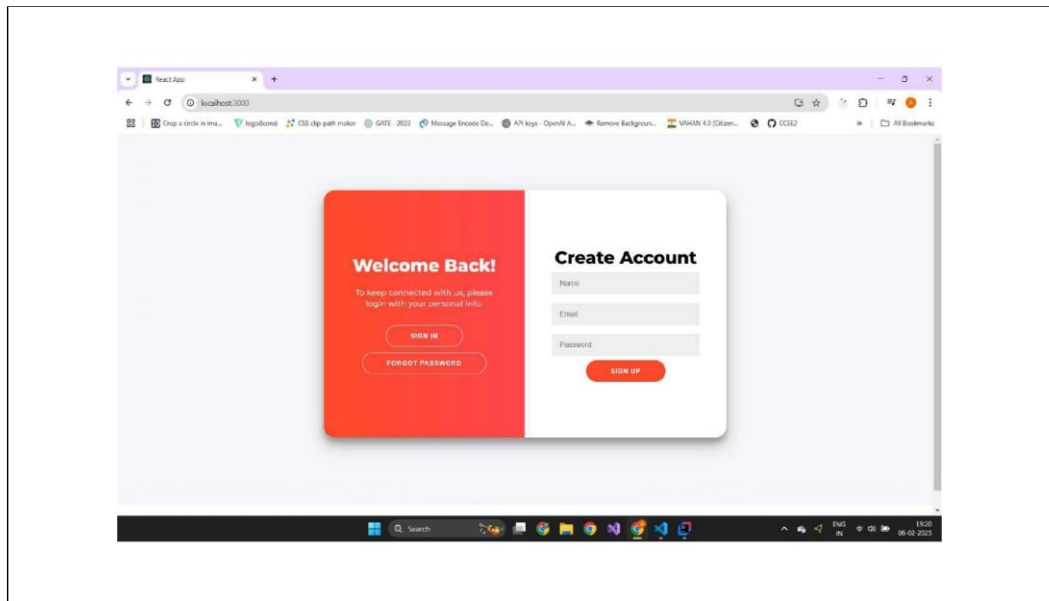


Fig1 _ User Registration page

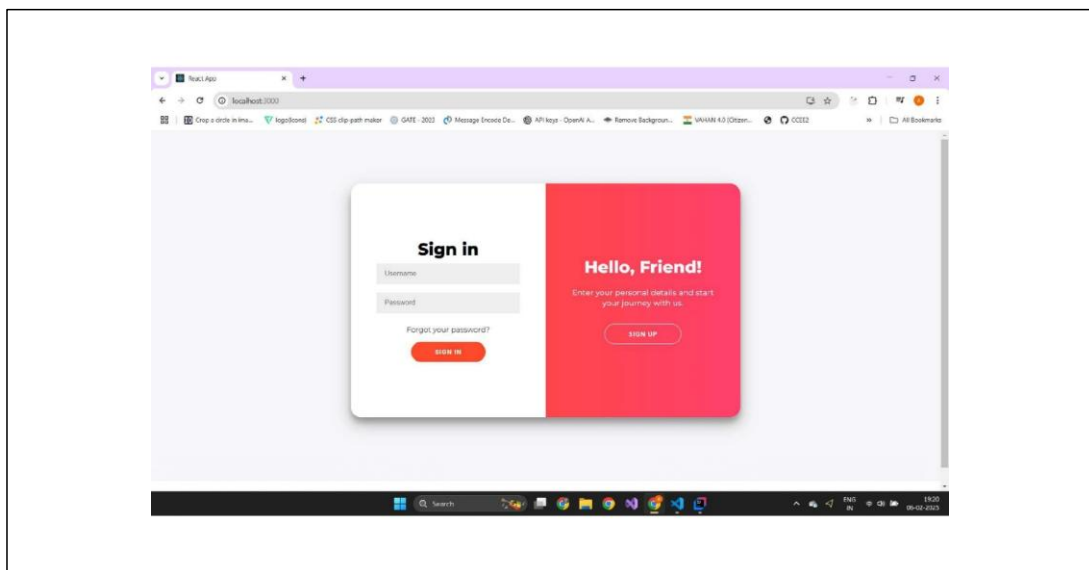


Fig2_Login Page

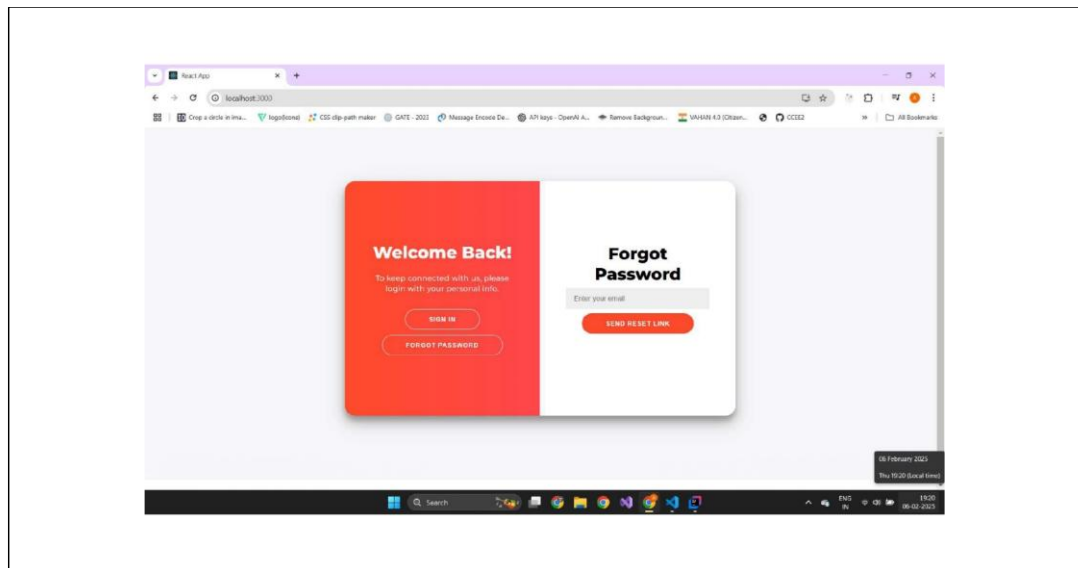
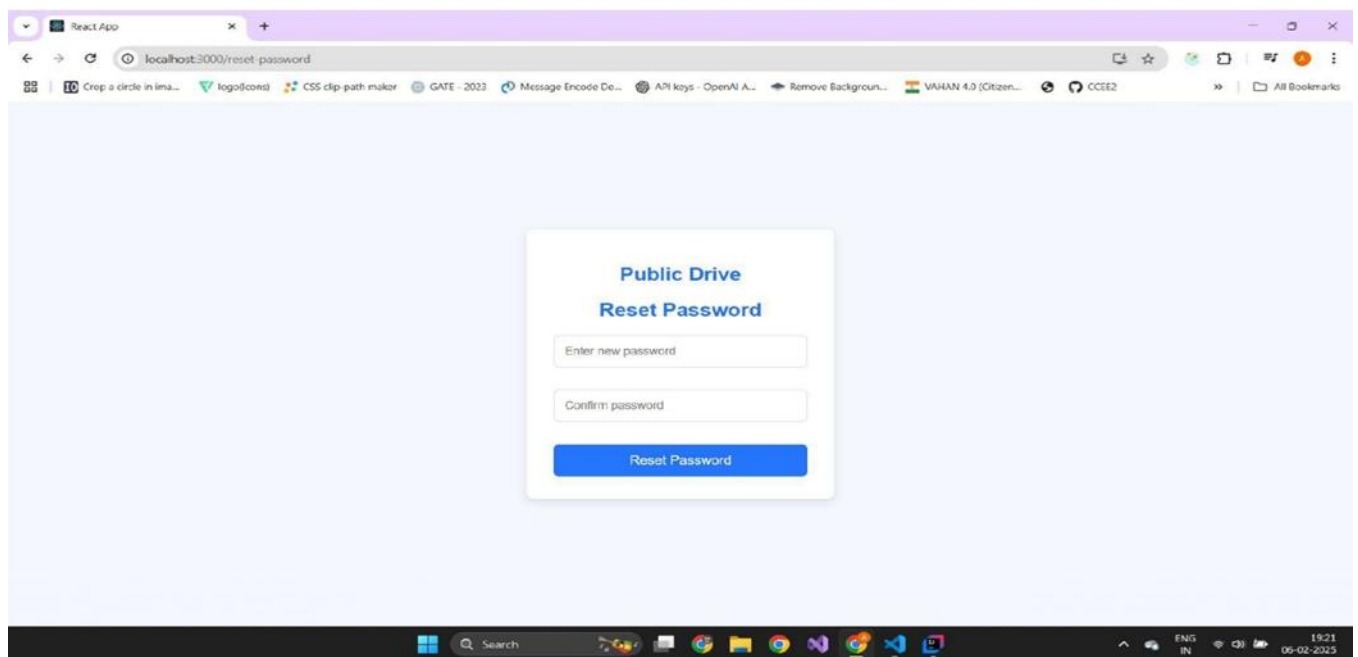


Fig 3 _Forgot Password



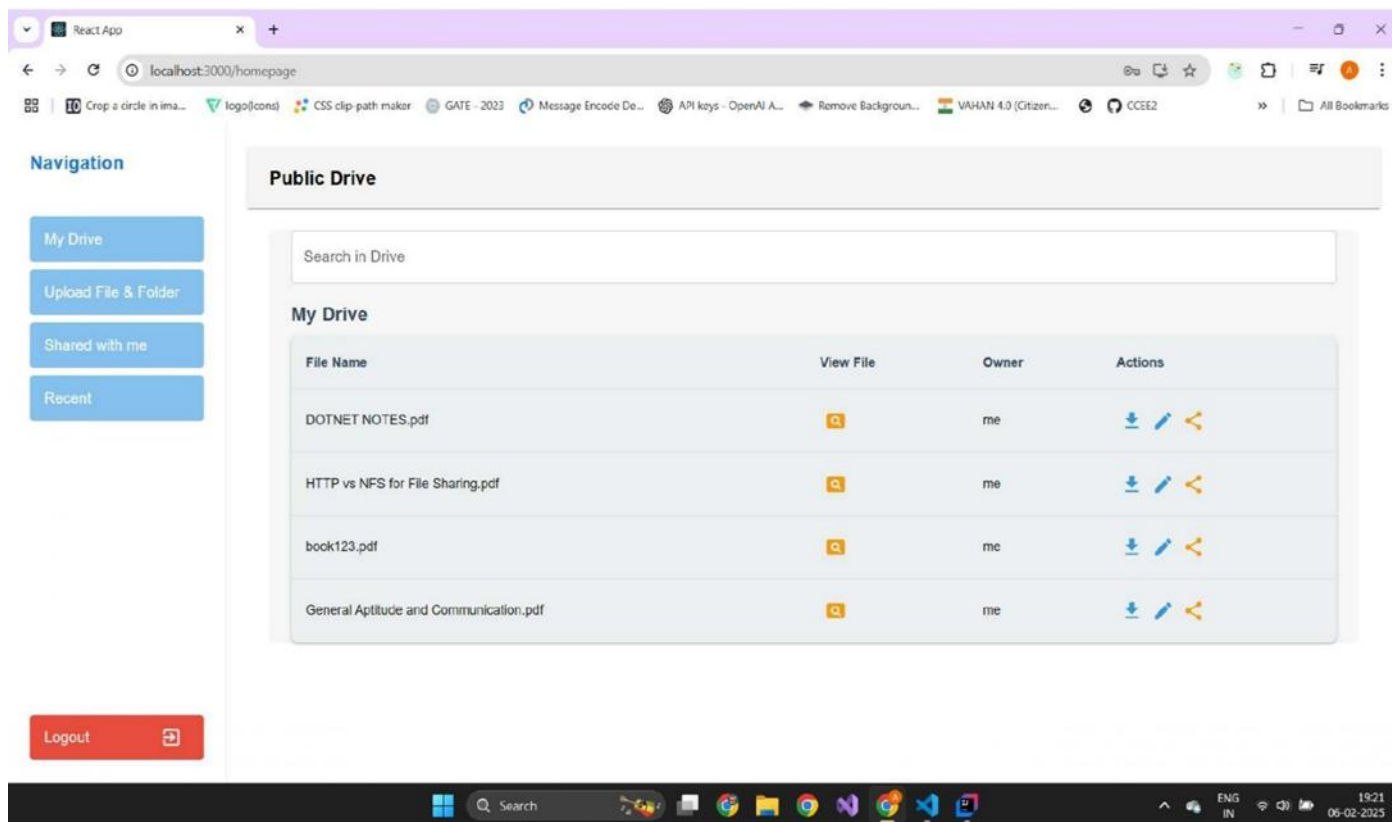
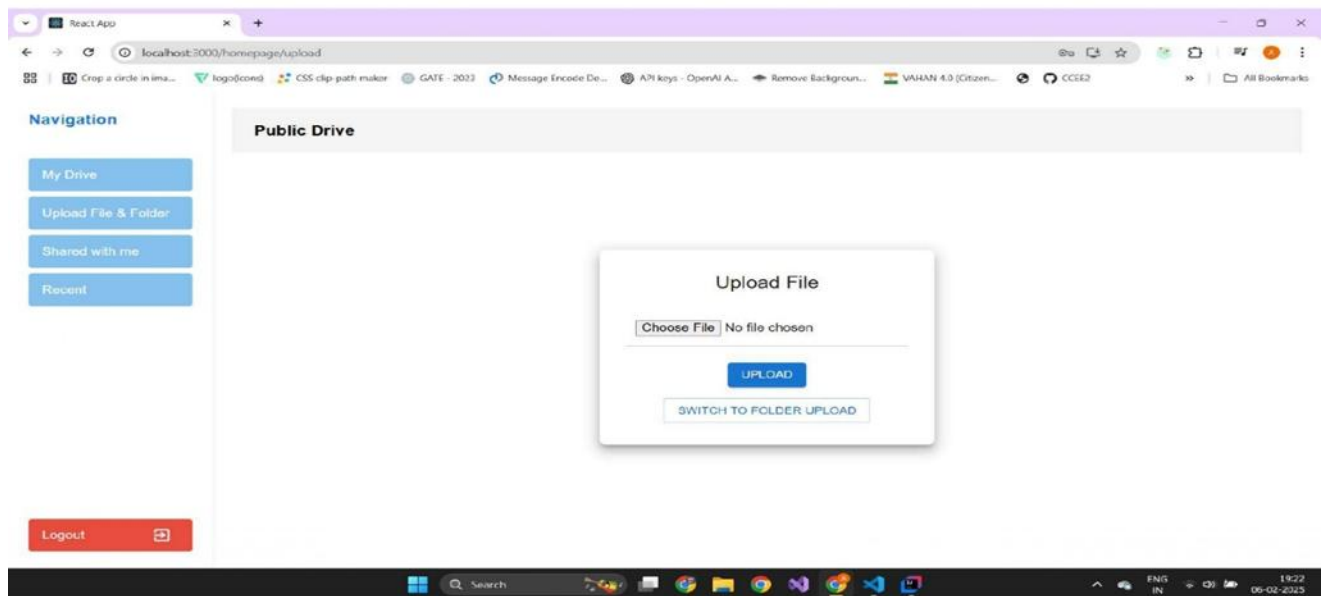
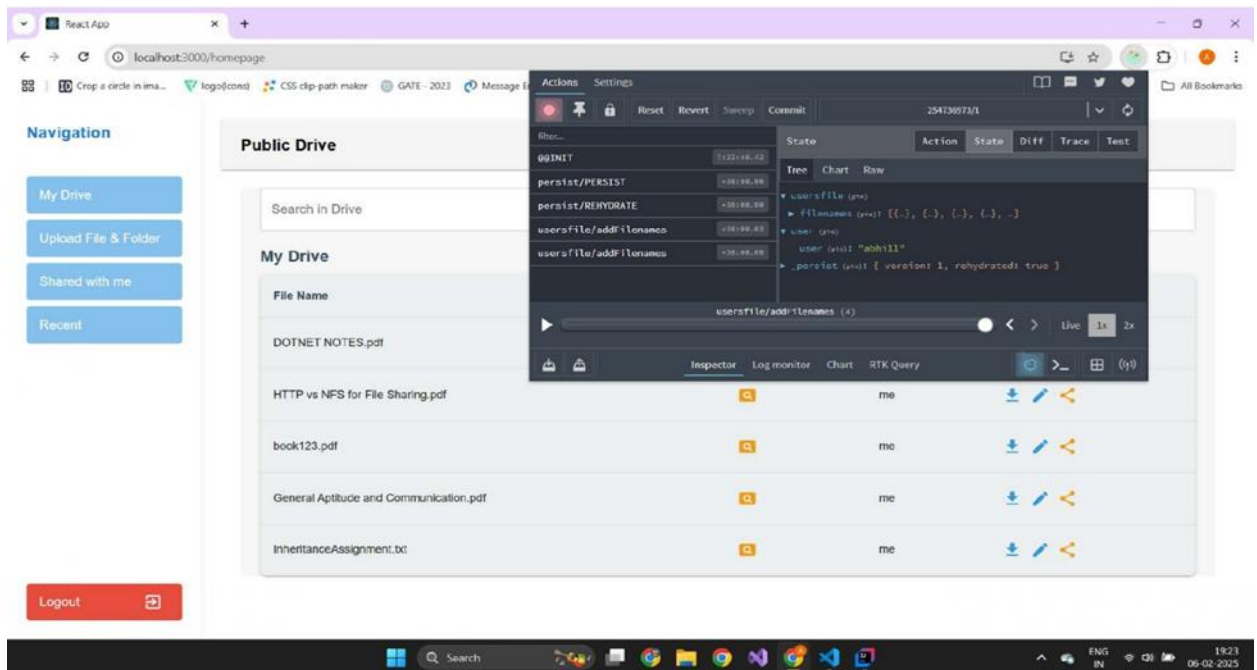
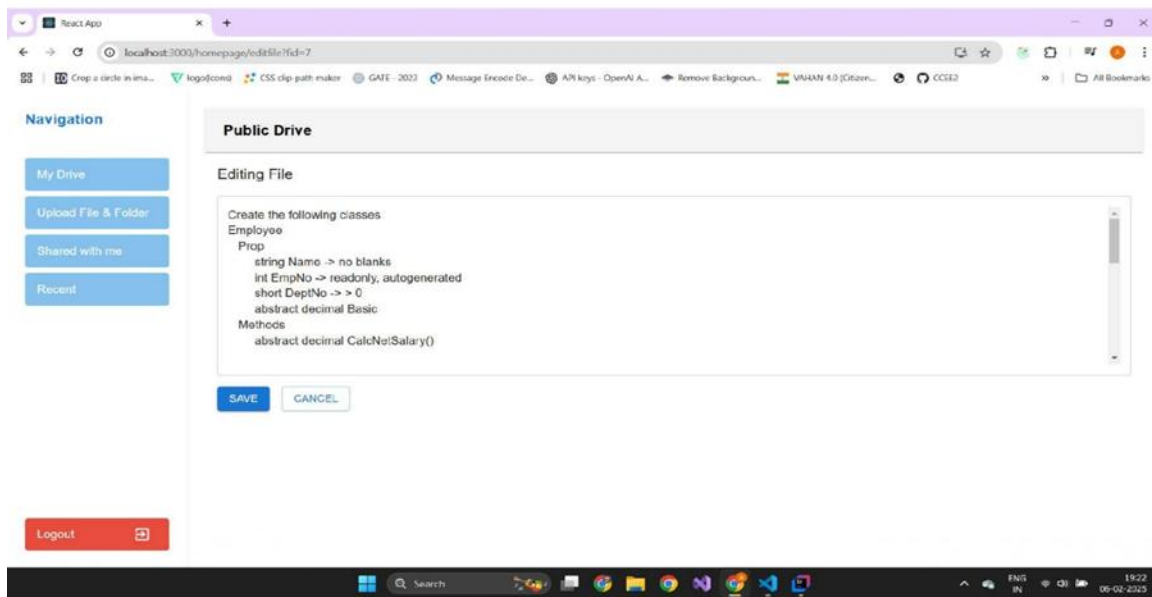


Fig5_ Dashboard: Showing the list of uploaded files and navigation options



File Upload Interface: Demonstrating file selection and progress.



8. FUTURE SCOPE:

- **Cloud Integration:**
 - Transition file storage from a server-based file system to cloud storage for improved scalability and reliability.
- **Enhanced Collaboration Tools:**
 - Integrate real-time collaborative editing features.
- **Mobile Application:**
 - Develop a companion mobile app for on-the-go file management.
- **Advanced Analytics:**
 - Implement data analytics to monitor file usage trends and optimize performance.
- **Improved Security Measures:**
 - Integrate multi-factor authentication (MFA) and enhanced encryption standards.

9. Conclusion

PublicDrive serves as a robust, secure, and user-friendly application for managing file operations in an enterprise environment. By leveraging modern technologies like React.js and Spring Boot, along with JWT for security and MySQL for data management, the application meets the growing demands of secure file handling. The system's modular design, comprehensive feature set, and emphasis on security make it a strong candidate for real-world deployment. The future enhancements outlined can further expand its capabilities and user base.

10. References

1. <https://spring.io/projects/spring-boot>
2. <https://spring.io/projects/spring-data-jpa>
3. <https://restfulapi.net/>
4. <https://www.mysql.com/>
5. <https://spring.io/projects/spring-web>
6. <https://reactjs.org/>
7. <https://nodejs.org/>