AIR UNIVERSITY

Semester Project

Members:

221055 - Abdul Basit, 221073 - M Moiz

Subject: Web Technologies

Project Report: Docify Web Application

Team Members:

- Moiz
- Basit

Technologies Used:

- **Frontend:** React, Vite
- **Backend:** Node.js, Express
- **Database:** MongoDB (local setup)

Github repository: click here

Introduction

Our web application, Docify, is designed as a document management platform similar to Google Docs, offering users the ability to create, edit, and manage documents. However, unlike Google Docs, real-time collaboration is not a feature of this application. We have used React and Vite for the frontend, while the backend is powered by Node.js and Express, with MongoDB used for storing user data and documents locally.

Features

1. **Login and Signup System:**
   The application provides a secure authentication system with user login and signup functionality. Passwords are hashed before storing them in the MongoDB database to ensure user security.
2. **Document Management:**
   Users can create, update, and delete documents. The documents are stored in the MongoDB database, allowing for persistent storage across sessions.
3. **User Interface:**
   The frontend, developed using React and Vite, offers a simple, user-friendly

interface for managing documents. Users can navigate through their documents, edit them, and save changes.

Challenges Faced

1. **Integration Between Frontend and Backend:**
   One of the primary challenges was establishing smooth communication between the frontend and the backend. As the project involves multiple components (frontend, backend, database), ensuring consistent data flow and handling API requests/responses correctly was a key hurdle. We overcame this challenge by thoroughly testing the API endpoints and ensuring proper error handling.
2. **Database Management:**
   Using MongoDB locally meant we had to manage the database setup, especially handling persistent storage of documents. Ensuring the database was properly initialized and configured for document storage was tricky, and we faced occasional issues with document retrieval during the initial testing phase. This was resolved by fine-tuning the database connection and schema design.
3. **Authentication System:**
   Implementing a secure login and signup feature posed some challenges, especially around password hashing and session management. We used bcrypt for hashing passwords, but initially faced issues with session persistence after user login. We solved this by improving our session management and storing JWT tokens securely.
4. **UI/UX Design:**
   Designing a user interface that is both functional and aesthetically pleasing was challenging. We wanted to make the platform simple to navigate while ensuring it had all the necessary features. After several iterations, we finalized the layout and flow of the application, ensuring it was easy for users to create and manage their documents.
5. **Performance Optimization:**
   Initially, the app faced some performance issues when handling larger documents, especially when loading them from the database. We addressed this by implementing lazy loading and optimizing the document storage structure in MongoDB, which helped improve performance.

Conclusion

Despite the challenges we faced, the Docify web application was successfully developed and deployed with the desired functionality. The use of React and Vite for frontend development allowed us to create a responsive and fast user interface, while Node.js and Express provided a robust backend to manage user authentication and document storage. MongoDB served as an effective local database for storing user data and documents.

This project not only improved our technical skills but also taught us valuable lessons in handling full-stack development, from frontend to backend and database management. The experience gained in debugging, troubleshooting, and optimizing the application will be invaluable for future projects.

Future Enhancements

- **Real-time Collaboration:** Although real-time collaboration is not part of the current project scope, this feature could be a potential enhancement for future versions.
- **Cloud Database Integration:** Moving the MongoDB database to the cloud (such as MongoDB Atlas) could improve scalability and ease of access.
- **File Export Options:** Adding the ability for users to export documents in different formats (PDF, DOCX) could further enhance the user experience.