

## Cloud Document Manager - Project Summary

The Cloud Document Manager is a web-based application developed during an internship to offer secure document storage, management, and sharing using cloud technologies. The platform uses Firebase for backend services and React for frontend development. Key features include secure user authentication, a responsive drag-and-drop upload interface, role-based access control, and mobile-friendly design. The system is integrated with Firebase Authentication, Firestore, and Storage to ensure reliable data handling and secure storage. Although current limitations due to Firebase's paid tier affect the upload feature, the core application is complete and ready for deployment.

### Technologies Used:

- Frontend: React, Material-UI, React Router
- Backend: Firebase Authentication, Firestore, Storage, Firebase Security Rules

### Achievements:

- Full-stack cloud integration using Firebase
- Secure, role-based document management
- Responsive, modern UI with Material-UI

## Interview Questions and Answers

1. Q: What is the Cloud Document Manager project?

A: It is a web application developed to securely manage, store, and share documents using cloud technologies like Firebase and React.

2. Q: What technologies did you use in this project?

A: React for frontend development, Material-UI for design, and Firebase for backend services including Authentication, Firestore, and Storage.

3. Q: How does Firebase Authentication work in your application?

A: It enables secure login and registration. Users are authenticated using email/password, and their session is managed securely by Firebase.

4. Q: How are documents uploaded and stored?

A: Documents are uploaded using a drag-and-drop interface and stored in Firebase Storage. Metadata is stored in Firestore for search and organization.

5. Q: What are Firebase Security Rules and how are they used?

A: Firebase Security Rules control access to Firestore and Storage. They ensure that only authorized users can read/write specific data.

6. Q: What limitation did you face with Firebase?

A: Firebase Storage now requires a paid subscription beyond the free trial. This affects the ability to upload documents unless a billing account is active and funded.

7. Q: What could be a potential improvement or future enhancement?

A: Integrating AWS S3 for storage to manage costs or implementing features like version control and real-time collaboration.

8. Q: How did you ensure the application is mobile-friendly?

A: By using Material-UI and responsive design practices within React to ensure compatibility across screen sizes.

9. Q: What routing mechanism did you use in the frontend?

A: React Router was used to manage navigation between different views and components.

10. Q: What makes this project stand out?

A: The seamless integration of Firebase services with a React frontend to create a secure, user-friendly cloud document management system.