Saylani Mass IT Training

Collaborative Study Notes App
Hackathon Task

An app where students can create and share study notes in real time, ideal for group study sessions or collaborative learning. Users can log in, create notes, edit them, and see changes made by other users instantly. Each note will display who last edited it, promoting accountability and teamwork.

Key Features:

- **User Authentication**: Allow users to sign up, log in, and log out via Firebase Authentication.
- **Profile Picture:** Allow user to upload profile picture and store it in Firebase storage.
- **Note Creation** & Management:
- **CRUD:** Create, edit, and delete notes.
- **Organize notes** into subject categories.
- **Real-Time Collaboration**: Notes are shared in real time using Firestore, so multiple users can view and edit the same note simultaneously.
- **Comments & Feedback:** Users can comment on notes to ask questions or give feedback.
- **Search and Filter**: Basic search functionality to filter notes by subject or keyword.

Technology Stack

- Frontend: React.js, Redux for state management.
- Backend: Firebase Authentication, Firebase storage for saving images and Firestore for database.
- Styling: TailwindCSS or Material-UI for a clean and responsive design.

Firebase Collection Schema

Collections:

1. Users
2. Notes
3. Comments

Saylani Mass IT Training

1. Users Collection

Stores user information and authentication details.

Collection Name: users

Document ID: userId (auto-generated by Firebase Authentication)

Fields:

- userId (string): Unique identifier from Firebase Authentication.
- displayName (string): User's display name.
- email (string): User's email address.
- profileImage (string): URL for the user's profile image.
- createdAt (timestamp): Date and time when the user account was created.

2. Notes Collection

Stores individual notes and metadata, including real-time collaboration info.

Collection Name: notes

Document ID: noteId (auto-generated or user-defined for each note)

Fields:

- noteId (string): Unique identifier for the note.
- title (string): Title of the note.
- content (string): Main content/body of the note.
- subject (string): Subject or category the note belongs to (e.g., Math, Science).
- createdBy (string): userId of the creator.
- lastEditedBy (string): userId of the last editor.
- lastEditedAt (timestamp): Timestamp of the last edit.
- createdAt (timestamp): Timestamp of when the note was created.
- collaborators (array of strings): Array of userIds of users who have access to edit/view the note.

Saylani Mass IT Training

3. Comments Collection

Each note can have its own comments stored in a sub-collection of notes.
Collection Name: comments (stored as a sub-collection under each note document)
Document ID: commentId (auto-generated for each comment)
Fields:

- commentId (string): Unique identifier for the comment.
- noteId (string): ID of the note this comment belongs to (redundant but useful for querying).
- userId (string): userId of the comment author.
- text (string): Content of the comment.
- createdAt (timestamp): Timestamp when the comment was created.

Submit Here