RIPHAH
INTERNATIONAL
UNIVERSITY

Name	Muhammad Mujahid
Sap-Id	54526
Course	Mobile App Development
Course Instructor	Sir Muhammad Usman
Project Name	Coaching App
Semester	05
Section	SE5-1

Project Report

Introduction:

The CouchingApp project is a mobile e-learning platform (internal name: courses_app) designed to deliver educational video content to users. It is built using the Flutter framework, ensuring a consistent and high-quality cross-platform user experience.

The application's architecture is comprehensive and scalable, relying on the Firebase suite to provide robust, modern services for:

- Secure user management.
- Real-time storage of course metadata.
- Optimized delivery of large video assets.

Application Architecture and Workflow:

The application is structured to provide a seamless content consumption experience, facilitated by three primary Firebase services and a core Flutter plugin.

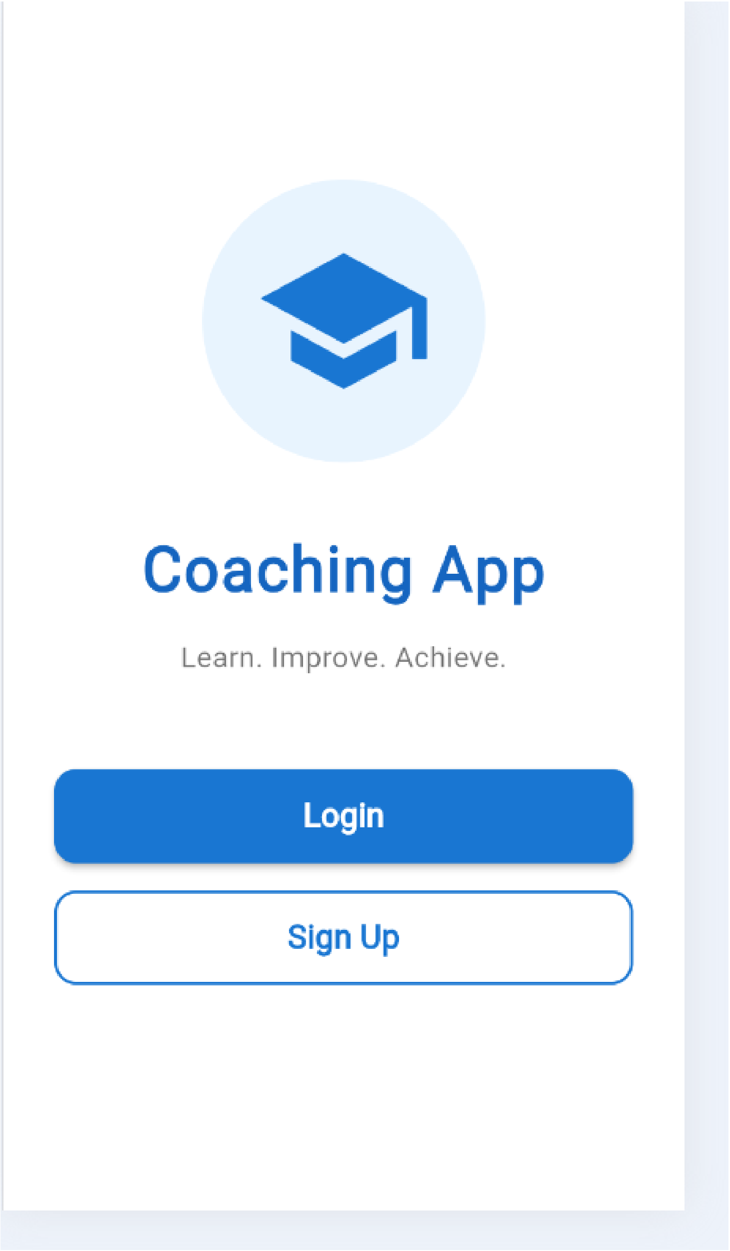
1. **Authentication (firebase_auth):** The user must log in to access materials, ensuring a personalized and secure experience.
2. **Course Metadata (cloud_firestore):** Course details (titles, descriptions, module lists) are stored in this database, allowing for flexible and real-time display of the course catalog.
3. **Video & Asset Storage (firebase_storage):** Large course video files are securely housed and served from this service, which is optimized for high-volume media content.
4. **Content Delivery (video_player):** When a video is selected, the app fetches the secure streaming URL from Firebase Storage. The video_player plugin then streams the content directly within the app, providing a native playback experience.

The overall workflow involves users logging in, browsing courses from Firestore, and consuming videos streamed via Firebase Storage.

Role of a Specific Screen (video_playback_screen.dart):

While the application features multiple screens (login, course list), the Video Playback Screen is functionally central, handling the complex logic of content delivery.

- **Player Initialization:** This screen receives a unique video identifier, communicates with Firebase Storage to retrieve the secure streaming URL, and then initializes the player.
- **Controller Management:** It instantiates and manages the VideoPlayerController lifecycle, ensuring the player is initialized correctly and properly disposed of when the user navigates away to prevent memory leaks.
- **Display and Controls:** The screen's primary function is to render the video feed and provide the standard user controls (play, pause, seek, volume), ensuring a familiar video experience.
- **Contextual Information:** It also displays metadata fetched from Cloud Firestore alongside the player, such as the video title and its position within the course module.

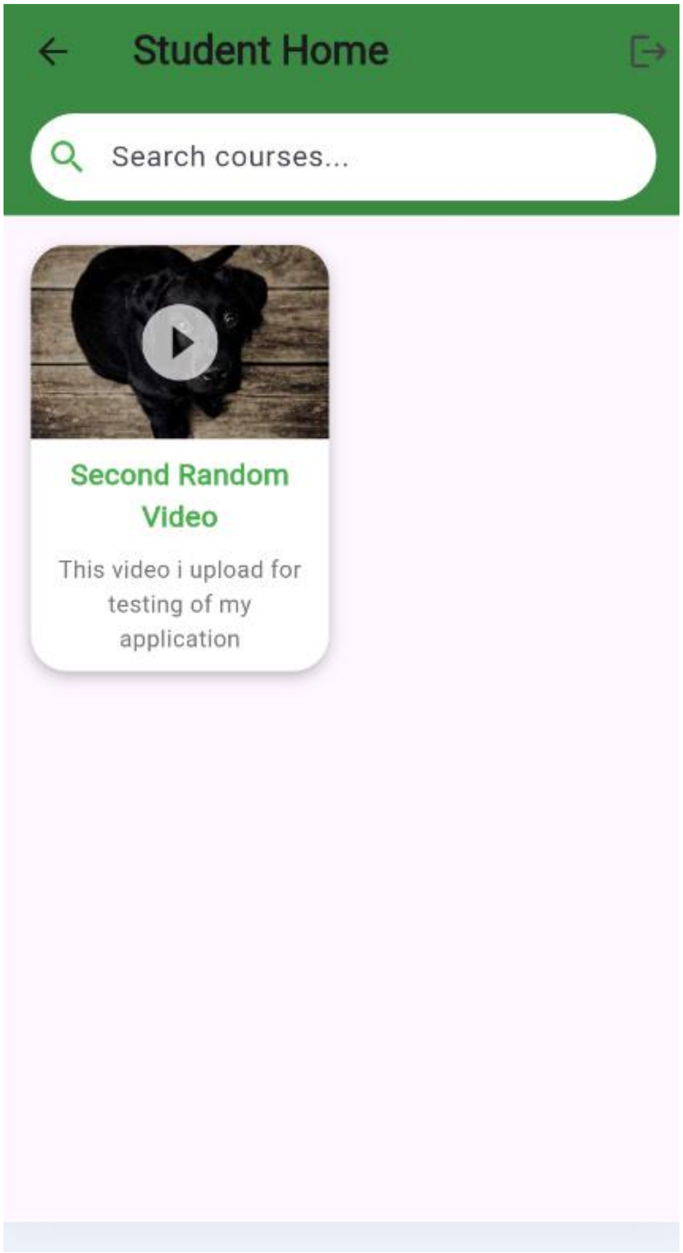


Course List Screen (courses_list_screen.dart):

This screen serves as the primary navigational hub and the real-time display of all available educational content.

- **Real-time Data Fetching:** It uses a StreamBuilder targeting the "courses" collection in Cloud Firestore. This ensures that the course catalog is instantly updated if new courses are added to the backend.

- **Data Model Mapping:** It takes the raw JSON data from Firestore snapshots and maps it to a Dart Course model object for easy handling within the UI.
- **User Interface (UI):** It constructs the visual elements necessary for browsing, such as:
    - Search and Filter bars (if implemented) for discoverability.
    - ListView.builder or GridView to efficiently render a potentially large list of course cards.
    - Course Cards that display key metadata (title, instructor, duration, progress status).
- Navigation & Interaction: Tapping on a course card triggers navigation to the detailed Course Details Screen (or directly to the video_playback_screen.dart), passing the selected course's unique ID as an argument.



Student Home

Search courses...

Second Random Video

This video i upload for testing of my application

Development Exposure:

My involvement in this project was deep and focused primarily on resolving a complex cascade of native Android build-time errors, positioning me as a build system troubleshooter and configuration expert.

- Gradle Configuration & Modernization: I diagnosed and corrected build issues in native files, fixing failures related to plugin resolution and class references.

- Dependency Versioning: A significant effort was dedicated to resolving version conflicts by systematically upgrading the Android Gradle Plugin (AGP), Kotlin Gradle Plugin (KGP), and the compileSdk to ensure full compatibility with modern Flutter and video playback dependencies.
- Inter-Screen Data Flow: Managed the complex routing and data passing required to ensure selected content from the Course List screen was correctly initialized by the Video Playback screen.
- Native Android Troubleshooting: My primary contribution was navigating the interface between the Flutter framework and the underlying Android build system, resolving errors that prevented the application from starting the compilation process.

Summary:

The CouchingApp project is a robust e-learning platform built on Flutter and a scalable Firebase backend (Auth, Firestore, Storage). The success of the application relies on the seamless data flow between the Course List, Video Playback, and Profile screens. My role was instrumental in unblocking development by systematically resolving critical native Android build issues related to Gradle and dependency conflicts. This effort has resulted in a stable, correctly configured build system, making the application fully functional and ready for further development and deployment.