

10/24/2025



Study pro

*“An AI-powered e-learning app with
courses, and progress tracking”*

*Cep Semester Project Mobile Application
Development*

Submitted By

Ghazanfar Ali 22SW014

Muhammad Khan 22SW068



Table of Content-----

- 1) Project Overview -----**
 - a) Problem Identification -----
 - b) Proposed Solution -----
 - c) Key Features-----
 - d) Technology Stack-----
- 2) Architecture & System Design -----**
UML Activity Diagram for Study Pro -----
- 3) User Interface Documentation-----**
 - a) Login and Sign-up Screen for User Interface -----
 - b) Home Screen Features -----
 - c) Detailed Dashboard -----
 - d) Course Details & Reviews -----
 - e) Video Playlist-----
 - f) AI Screen-----
 - g) Settings Screen-----
- 4) API Integration-----**
 - a) Chat with Gemini-----
 - b) Chat with PDF-----
 - c) Grammar Check-----
 - d) Quiz Generation-----
- 5) Issues and Bugs Encountered and Resolved during Development-----**
- 6) GitHub repo link-----**



Project Overview

Problem Identification

Today, many students use online learning platforms, but most of them do not give personal help, real-time feedback, or progress tracking. Because of this, students often lose focus and find it hard to understand difficult topics. There is a need for a smart learning app that offers AI help, interactive lessons, and progress tracking to make studying easier and more effective. Such an app can help students learn at their own pace and improve their understanding of each subject. It can also make online learning more engaging and useful for both students and teachers.

Proposed Solution

To solve this problem, we created **StudyPro**, an AI-based learning app that makes studying easy and smart. It offers video lessons, notes, and progress tracking in one place. The app uses **AI features** like chat with Gemini, grammar check, and quiz generation to give students quick help and feedback.

StudyPro keeps students motivated, helps teachers manage courses, and makes online learning more engaging and effective. It also allows students to learn anytime and anywhere through a simple and user-friendly design. The app provides a modern way of learning that saves time and improves understanding for all users.



Project Summary

Study Pro is an online learning platform that connects teachers and students through comprehensive course management and AI-powered features.

Key Features

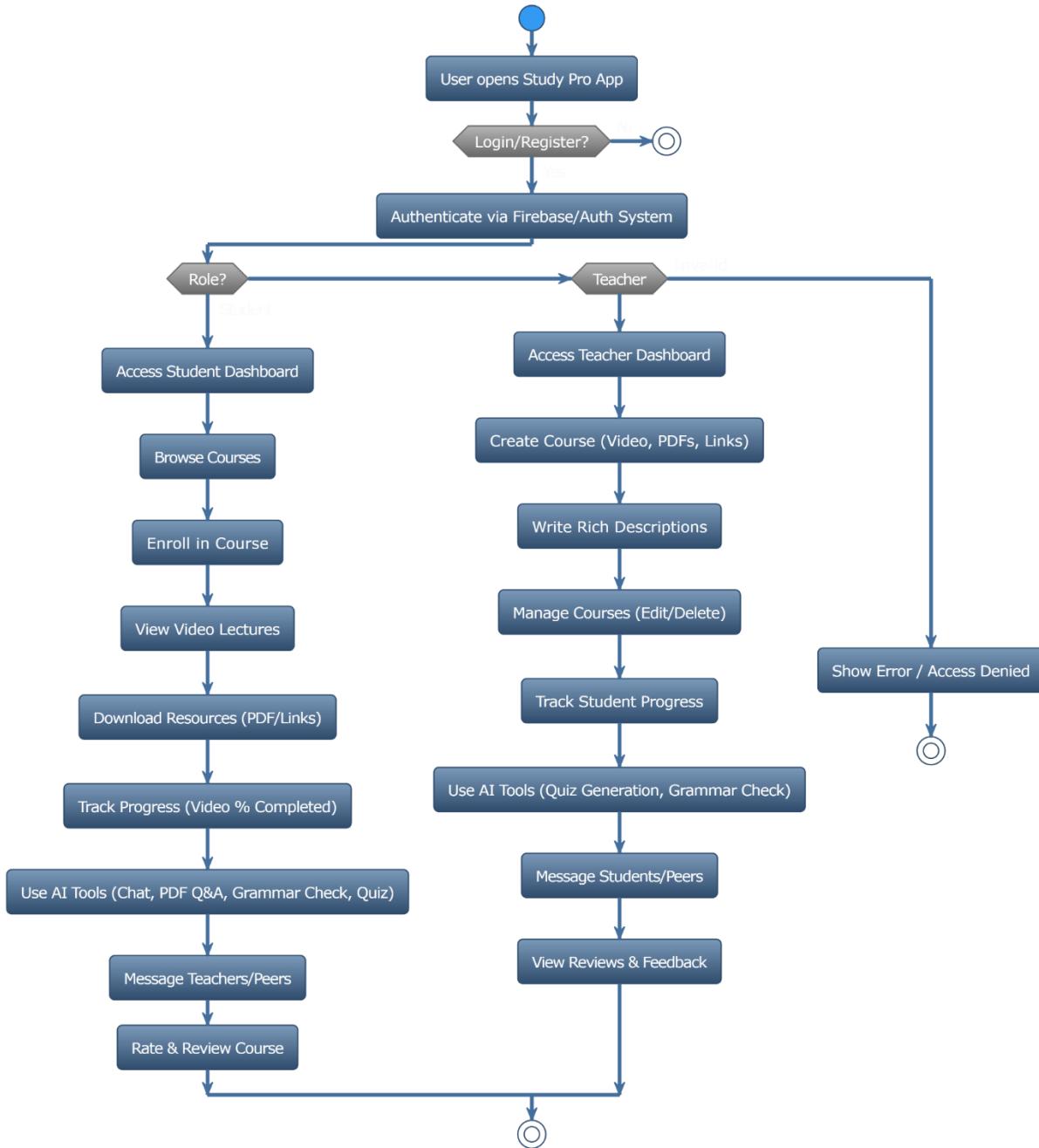
- **Course Management:** Video uploads, PDF resources, external links
- **User Dashboards:** Separate interfaces for students and teachers
- **AI Integration:** Gemini-powered chat, PDF analysis, grammar checking
- **Social Features:** Real-time messaging, reviews, ratings
- **Content Access:** Video streaming with progress tracking

Technology Stack

- **Frontend:** Flutter
- **Backend:** Firebase Auth, Fire Store
- **Database:** NoSQL Database
- **AI Integration:** Google Gemini API



UML Activity Diagram for Study Pro





Login and Sign-up Screen for User

Email

Password

Login

Forgot password

Or

Don't have an account? [Sign up](#)

Username

Email

Password

Country

Select Role

Student Teacher

Sign up

Already have an account? [Login](#)

Forgot Password

Reset Your Password

Enter your email and we will send you a password reset link.

Enter your email

Send Reset Link

Email

Password

Login

Forgot password

Or

Don't have an account? [Sign up](#)

Username

Email

Password

Country

Select Role

Student Teacher

Sign up

Already have an account? [Login](#)

Forgot Password

Reset Your Password

Enter your email and we will send you a password reset link.

Enter your email

Send Reset Link



After a successful authentication, the user is redirected to the **Home Screen**. The Home Screen serves as the central hub of the application and provides the following features:

1. Course Categories

- a. The Home Screen displays a list of course categories available within the platform.
- b. Users can browse and select a course category of their interest.

2. Student Abstract Dashboard

- a. Alongside course categories, the Home Screen provides an abstract (high-level) view of the student dashboard.
- b. This section gives a quick overview of the learner's current progress.

3. Detailed Dashboard

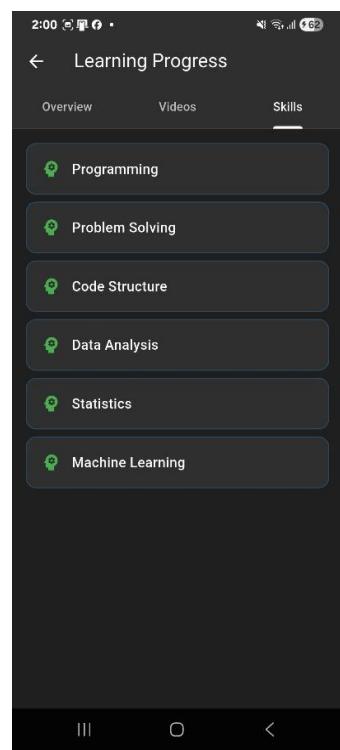
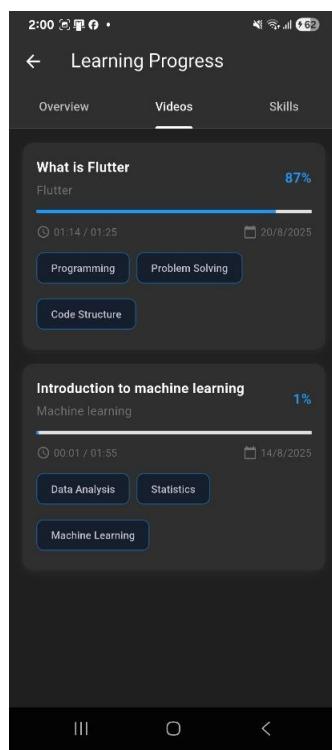
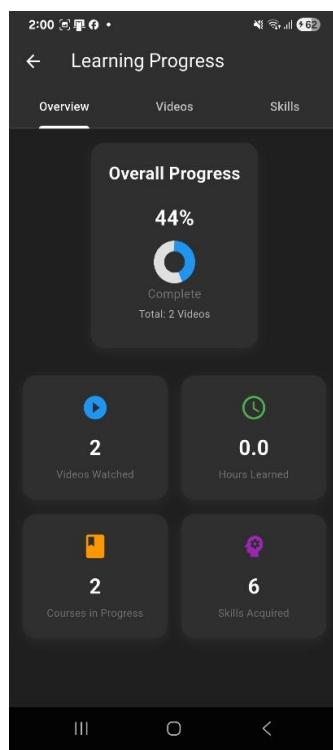
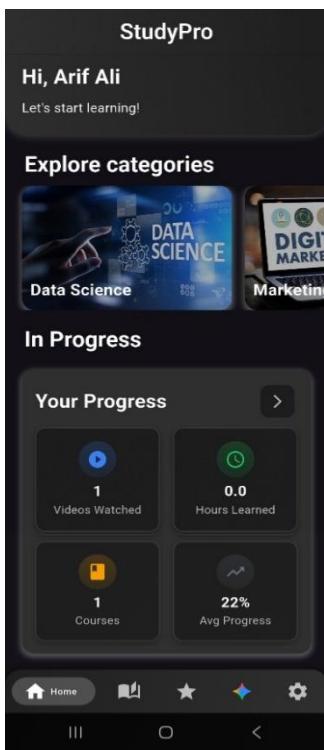
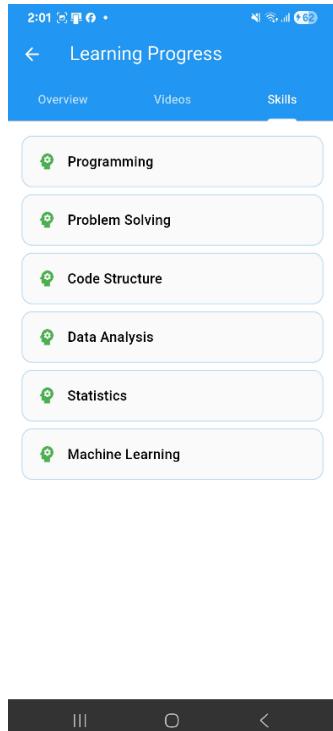
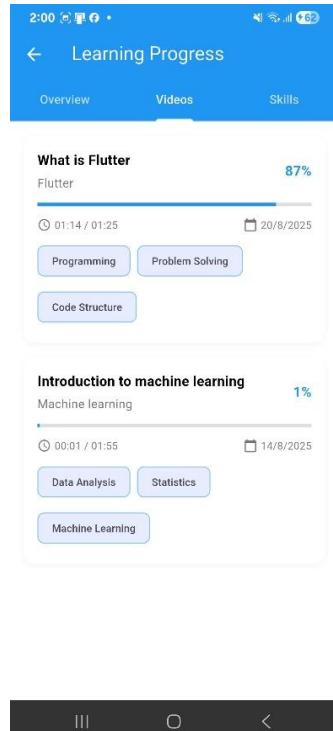
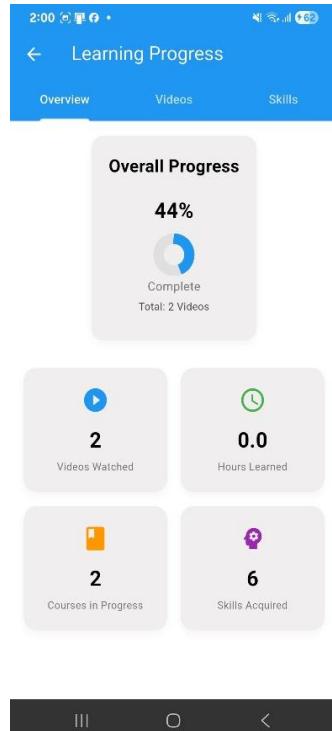
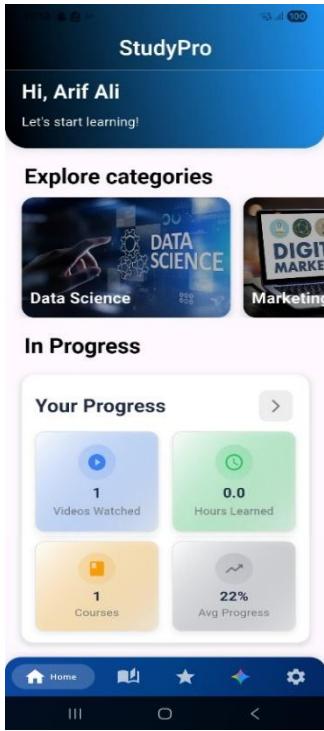
When the student navigates into the detailed dashboard, they can view:

- i. **Study Progress:** The number of videos watched out of the total available.
- ii. **Skill Acquisition:** A breakdown of skills that the learner has acquired by completing specific course modules or videos.

This structured flow ensures that learners not only have seamless access to available learning resources but can also **track their progress and achievements** in real time, thereby enhancing engagement and motivation throughout the learning journey.



Mobile Application Development





Course Search & Discovery: Students can easily search and filter courses by keywords or categories.

Course Details & Reviews: Each course includes detailed information such as title, description, instructor details, and learner reviews to help students evaluate course quality before enrolment.

Video Playlist: Courses are structured into a playlist of videos, enabling students to navigate lessons in sequence or revisit specific topics.

Video Player: An integrated player allows learners to watch videos directly within the app, with support for playback controls and progress tracking.

Resource Downloads: Additional learning resources (e.g., notes, slides, PDFs, or assignments) can be downloaded for offline access.

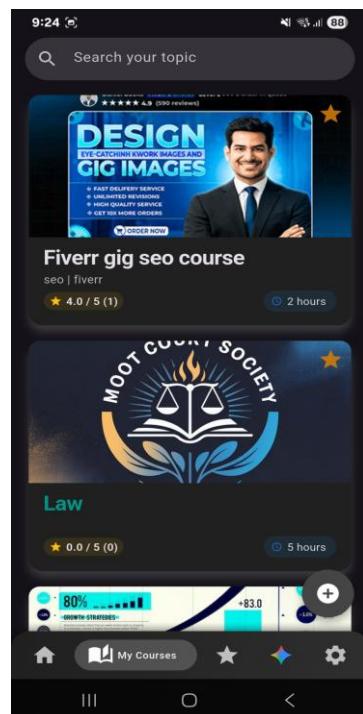
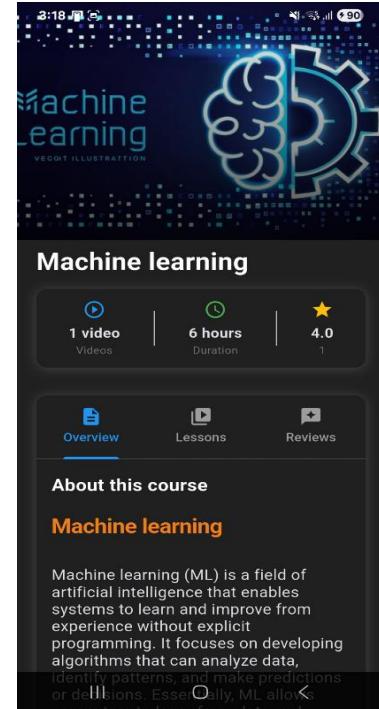
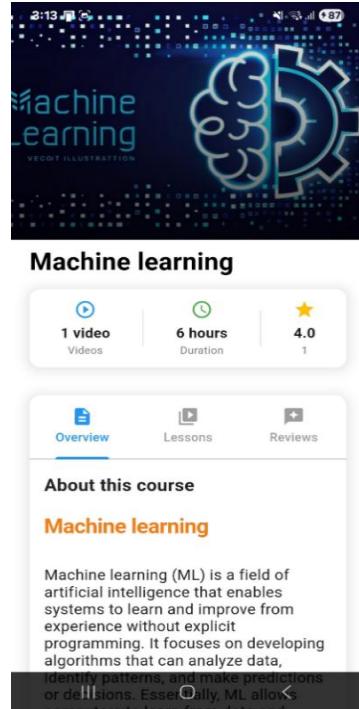
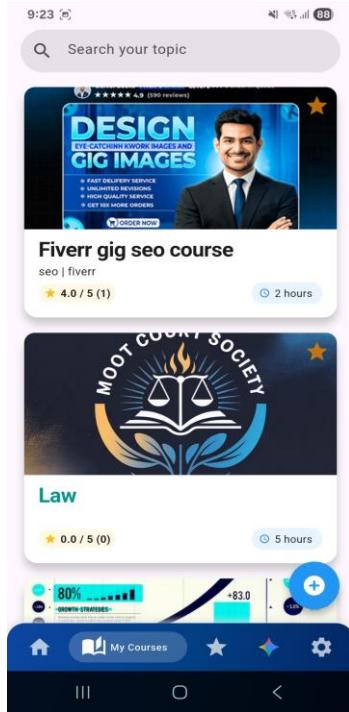
External Links: Students can access external references or suggested reading materials through embedded links.

Progress Tracking:

- The system tracks each student's learning activity by monitoring how much of each video has been watched (based on video length).
- A cumulative report is reflected in the **Student Dashboard**, showing the total minutes watched versus the overall course duration. This helps students track their learning pace and motivates them to complete the course.



Mobile Application Development

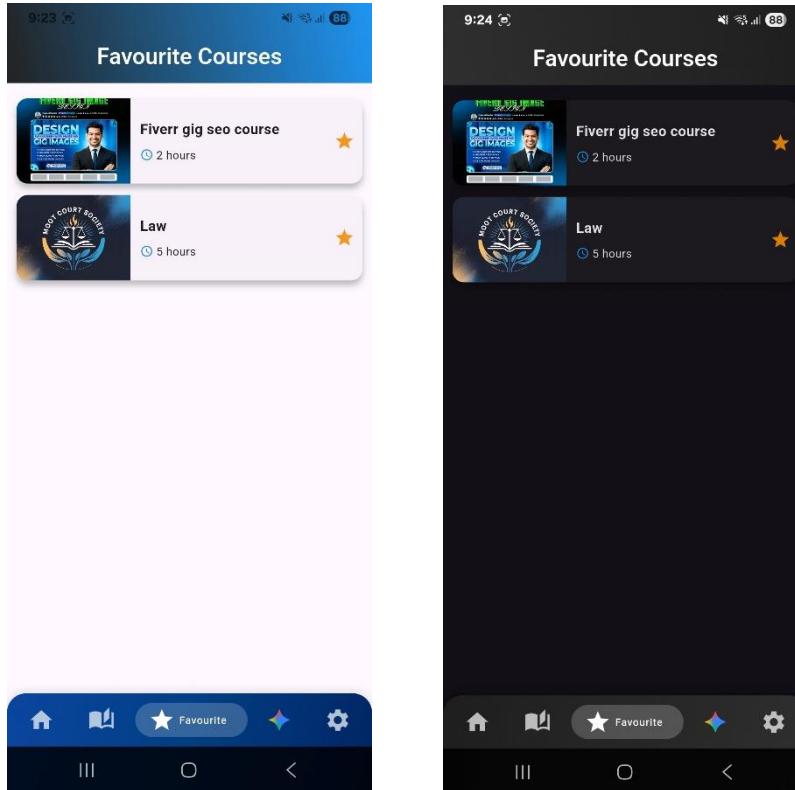




Favorite Course Screen

The Favorite Courses Screen allows students to save and easily access the courses they like or plan to enroll in later. When a student taps the star icon (★) on any course card, that course is added to their Favorites list and displayed in this dedicated screen.

It provides quick access to Favorite courses without needing to search again, improving user experience and engagement





AI Screen

The **AI Screen** integrates the **Gemini API** to provide intelligent, AI-powered learning support for students. It enhances the study experience by enabling learners to interact with AI in multiple ways. Key features include:

- **Chat with Gemini**

Students can have natural conversations with Gemini by sending text or images. The AI responds with context-aware answers, explanations, or clarifications, supporting personalized learning.

- **Chat with PDF**

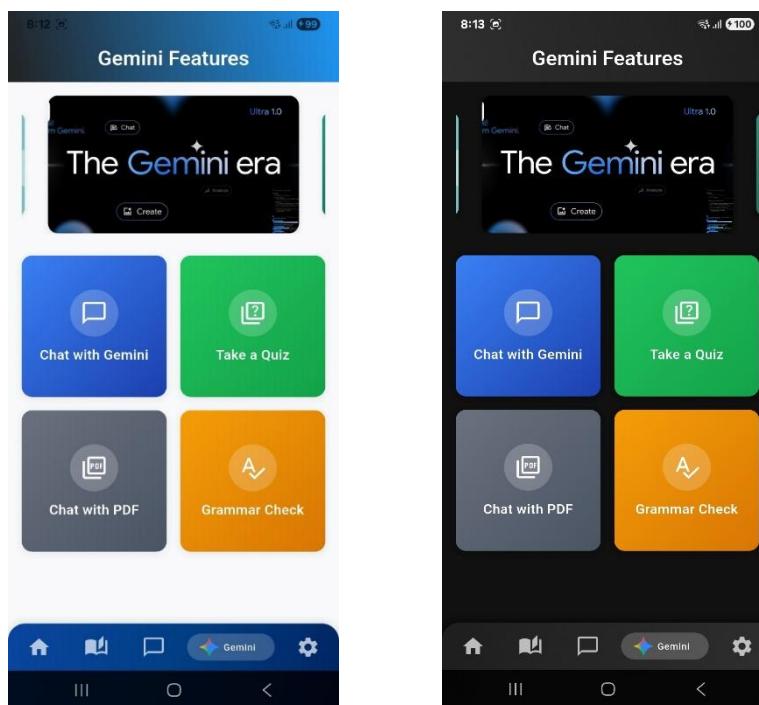
Students can upload or open a **PDF file** and interact with Gemini to ask questions about its content. This helps in quickly understanding complex documents, summarizing material, and extracting key points.

- **Grammar Check**

Students can enter text and use Gemini to **identify and correct grammatical errors**. The feature provides suggestions and improvements for writing assignments, notes, or discussions.

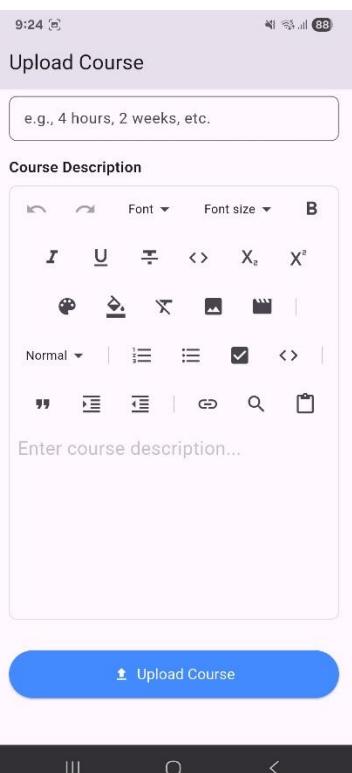
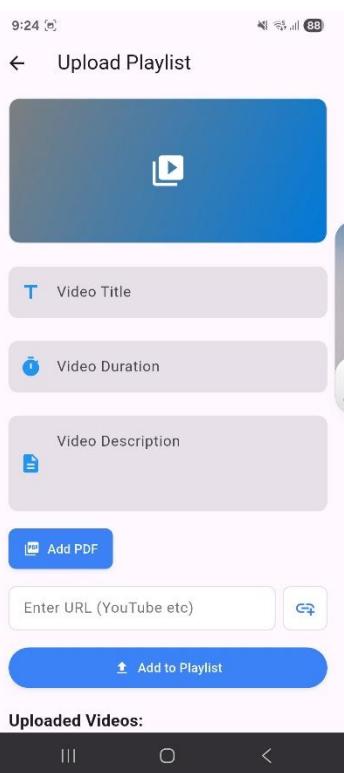
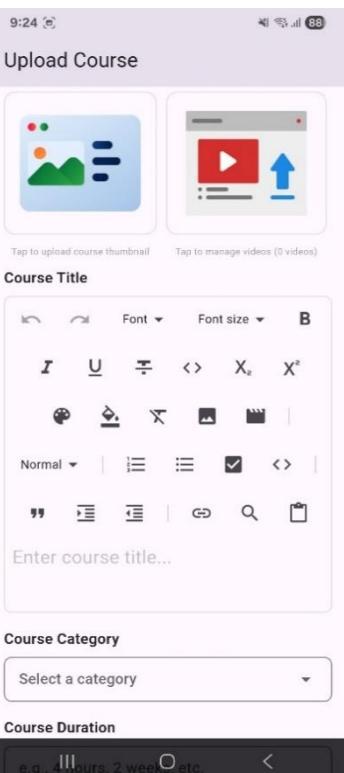
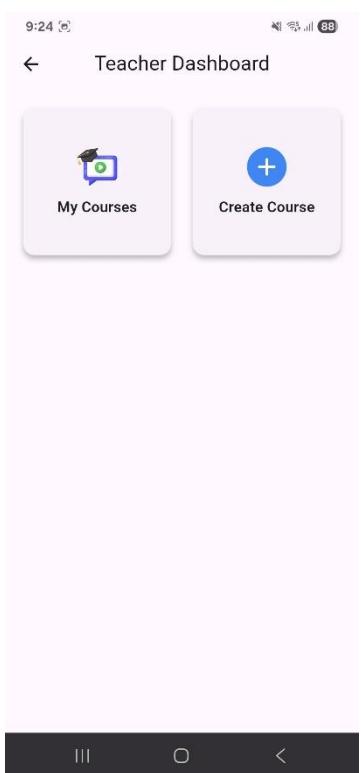
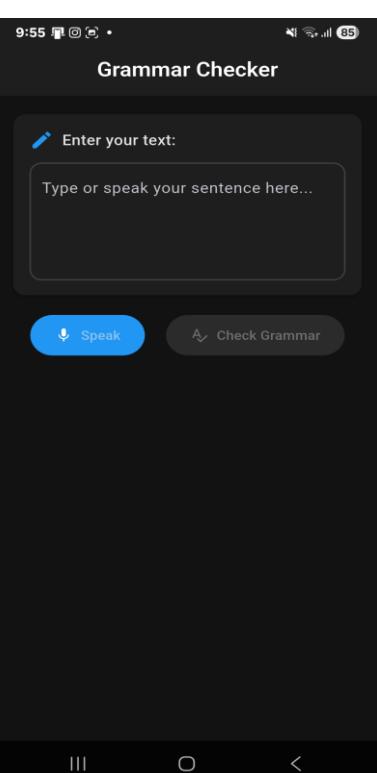
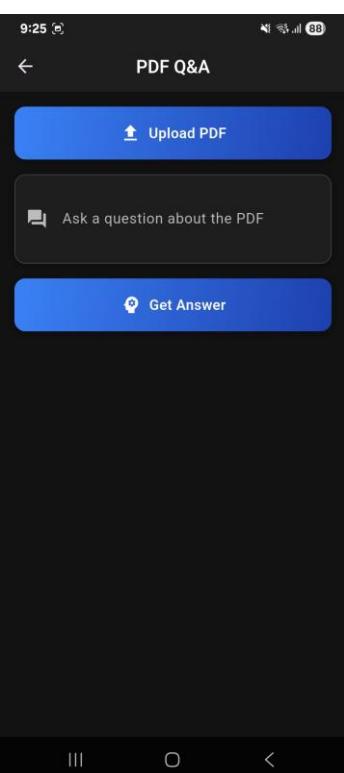
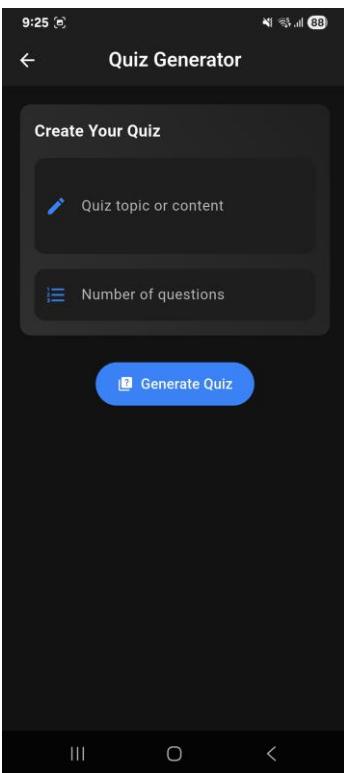
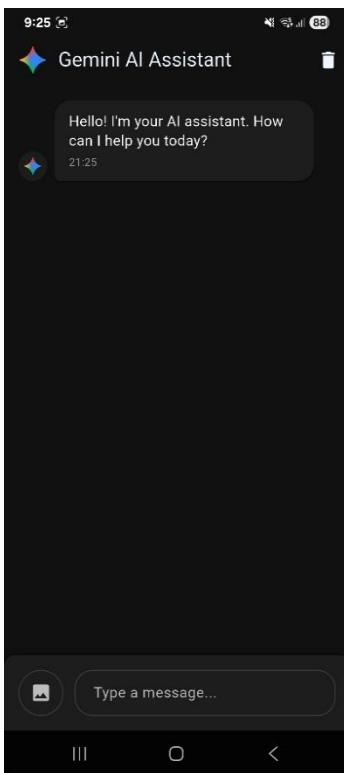
- **Quiz Generation**

By simply entering a **prompt or topic**, students can generate quizzes created by Gemini. This allows learners to test their knowledge interactively and prepare for assessments.





Mobile Application Development



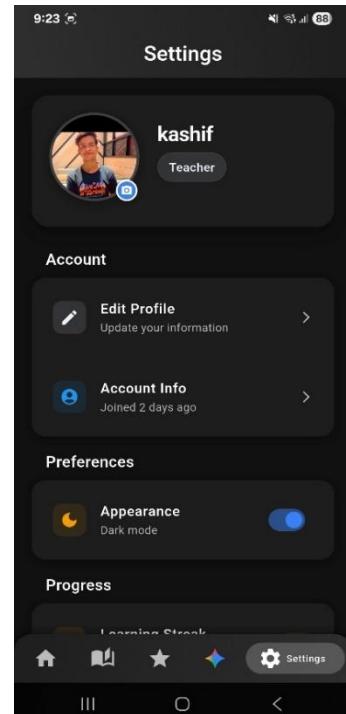
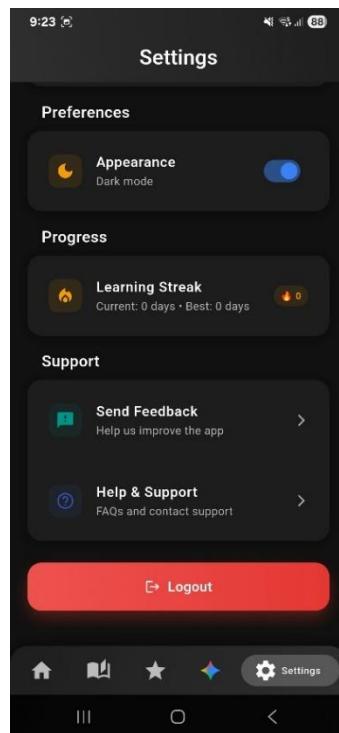
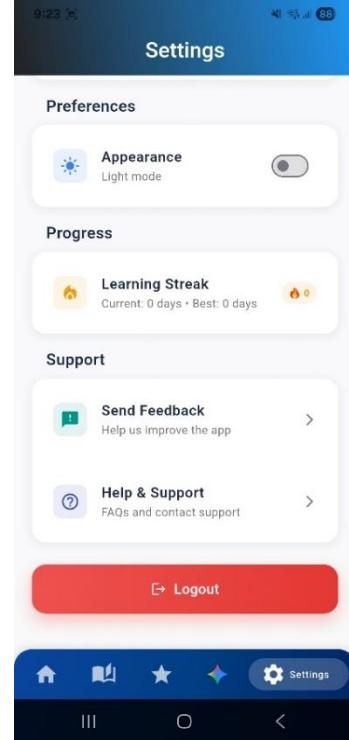
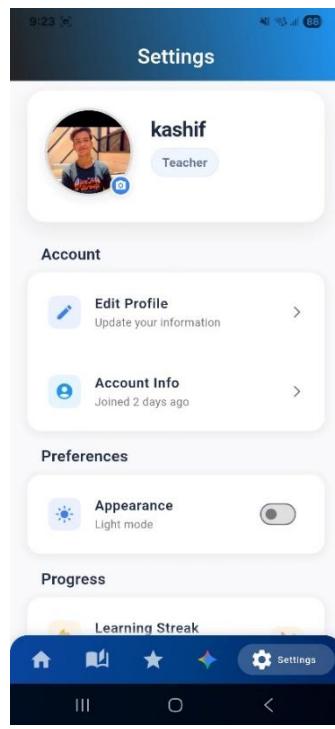


Settings Screen

- The **Settings Screen** provides users with control over their personal preferences and account management. It allows students to customize their learning experience and manage their profile efficiently. Key features include:
- **Edit Profile**
Students can update personal details such as name, email, and other profile information.
- **Upload Profile Picture**
Users can personalize their account by uploading or changing their profile photo.
- **Theme Preferences**
Toggle between **Dark Mode** and **Light Mode** to adjust the app's appearance based on personal preference.
- **Help & Support**
Direct access to assistance for technical issues, FAQs, or reaching out to the support team.
- **Review the App**
Students can rate and provide feedback on the app to improve the overall learning experience.
- **Learning Streak**
A feature that displays the user's daily/weekly learning streak, motivating students to maintain consistency.
- **Logout**
Securely sign out from the app.



Mobile Application Development





Issues and Bugs Encountered and Resolved during Development

1. Dependency Version Conflicts

Issue: These two packages often cause **dependency version conflicts**, because they may depend on **different versions** of common libraries (like quill_delta, flutter_html, or intl).

```
flutter_quill: ^11.4.1
flutter_quill_extensions: ^11.0.0
```

Fix:

Make sure both packages use compatible versions (check pub.dev for matched releases).

Try running:

```
flutter pub upgrade --major-versions
flutter pub get
```

2. Responsiveness and Hardcoded Sizes

Issue:

Initially, hardcoded height and width values were used across the app, causing poor responsiveness on different screen sizes.

Fix Implemented:

Created SizeConfig class to dynamically scale dimensions based on device size using MediaQuery.

Error in Terminal (Before Fix):

```
RenderFlex overflowed by 45 pixels on the right.
LayoutConstraint error: BoxConstraints forces an infinite width.
```



Fix

Replaced hardcoded width and height values with responsive scaling functions using the SizeConfig class.

Now, all UI elements automatically adjust based on the device screen size for better responsiveness.

Used:

SizeConfig().scaleWidth(value, context)

SizeConfig().scaleHeight(value, context)

File → size_config.dart

```
1 import 'package:flutter/widgets.dart';
2
3 class SizeConfig {
4
5     double scaleHeight(double size, BuildContext context) {
6         double baseHeight = 812;
7         double screenHeight = MediaQuery.of(context).size.height;
8         return screenHeight * (size / baseHeight);
9     }
10
11    double scaleWidth(double size, BuildContext context) {
12        double baseWidth = 375;
13        double screenWidth = MediaQuery.of(context).size.width;
14        return screenWidth * (size / baseWidth);
15    }
16}
17
```



3. Null Safety Errors (→ File: all_course_screen.dart)

```
        return ListView.builder(
            shrinkWrap: true,
            physics: const NeverScrollableScrollPhysics(),
            itemCount: filteredCourses.length,
            itemBuilder: (context, index) {
                final course = filteredCourses[index];
                final courseData = course.data() as Map<String, dynamic>;
                final courseId = course['courseId'];

                final stats = reviewStats[courseId] ;
```

Fix:

safely handle null values using:

```
        }

        return ListView.builder(
            shrinkWrap: true,
            physics: const NeverScrollableScrollPhysics(),
            itemCount: filteredCourses.length,
            itemBuilder: (context, index) {
                final course = filteredCourses[index];
                final courseData = course.data() as Map<String, dynamic>;
                final courseId = course['courseId'];

                final stats = reviewStats[courseId] ?? {
                    'totalReviews': 0,
                    'averageRating': 0.0,
                    'ratingDistribution': {1: 0, 2: 0, 3: 0, 4: 0, 5: 0},
                };
```



4. Gemini version issue (→ File name : gemini_services.dart)

Issue:

Using an **old Gemini version** might also cause null-safety issues since older APIs don't support Dart null safety properly.

```
1 import 'package:google_genenerative_ai/google_genenerative_ai.dart';
2 import 'package:studypro/views/Gemini_AI/constants.dart';
3
4 class GeminiService {
5
6
7     Future<String> callGeminiAPI(String prompt) async {
8         try {
9             final model = GenerativeModel(
10                 model: 'gemini-1.5-flash',
11                 apiKey: geminiApiKey,
12             );
13             final content = [Content.text(prompt)];
14             final response = await model.generateContent(content);
15             return response.text ?? 'No response from Gemini API';
16         } catch (e) {
17             throw Exception('Failed to call Gemini API: $e');
18         }
19     }
20 }
```

Fix

updated Gemini code uses 2.5 Flash

```
Future<String> callGeminiAPI(String prompt) async {
    try {
        final model = GenerativeModel(
            model: 'gemini-2.5-flash',
            apiKey: geminiApiKey,
        );
        final content = [Content.text(prompt)];
        final response = await model.generateContent(content);
        return response.text ?? 'No response from Gemini API';
    } catch (e) {
        throw Exception('Failed to call Gemini API: $e');
    }
}
```



5. Network / API Issues

Issue:

Some times our app makes API calls and you forgot to add internet permission, it will fail with errors like:

SocketException: Failed host lookup

Fix (Android):

Add this line in your AndroidManifest.xml file, **before the <application> tag**:

Path:

android/app/src/main/AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
```

GitHub Repo link: <https://github.com/ghazanfar-ali-tech/StudyPro>

THE END