An-Najah National University

Faculty of Engineering & Information Technology

Computer Engineering Department

Spring 2024/2025

Midadium

Presented By:

Teama Ahmad Amer

Leen Imad Batta

Supervised by: Dr. Shareef Yaseen

Presented in partial fulfilment of the requirements for Bachelor degree in Computer Engineering.

Acknowledgment

We would like to extend our sincere gratitude to our supervisor, **Dr. Shareef Yaseen**, whose calm leadership, consistent support, and invaluable guidance were key to the success of this project. His ability to balance academic rigor with encouragement allowed us to grow both technically and intellectually. Throughout the development of this work, Dr. Yaseen's insights helped us refine our vision, overcome challenges, and maintain focus on our objectives with clarity and purpose.

Our appreciation also goes to the **Department of Computer Engineering at An-Najah National University**, whose educational environment fostered the mindset, curiosity, and discipline required to pursue such a project. The tools, knowledge, and support we received during our academic years laid the foundation upon which this work was built.

To our **families**, we offer our deepest thanks. Your patience, love, and belief in us—especially during moments of exhaustion and uncertainty—gave us the emotional strength to persevere. Your role in this achievement is as vital as any line of code written or problem solved.

We are also grateful to our **colleagues and friends**, who contributed thoughtful suggestions, shared feedback, and provided motivation along the way. This project was shaped not only by our efforts but by a spirit of collaboration and shared growth.

Finally, we are proud to have developed **Midadium**—a platform born from the vision of Palestinian students, built with a commitment to accessible, modern, and intelligent education. While this project may represent a single chapter in our academic journey, it reflects our sincere belief in technology as a tool for empowerment, learning, and positive change.

Disclamer

This report was authored and submitted by students **Teama Amer Ahmad Amer**, and **Leen Imad Faris Batta**, from the Computer Engineering Department, Faculty of Engineering, at **An-Najah National University**, in partial fulfillment of the requirements for the Bachelor's degree in Computer Engineering.

The contents of this document represent the collective efforts, academic research, and personal interpretations of the students throughout the course of their graduation project. While the work has undergone basic editorial review, it has not been formally revised or corrected by the department or university, and may contain unintentional errors in language, formatting, or analysis.

The opinions, methodologies, outcomes, and recommendations expressed herein are those of the authors alone, and do not necessarily reflect the official views of the supervisor, department, or university.

Accordingly, **An-Najah National University** assumes no responsibility or liability for the consequences of applying, sharing, or interpreting the information contained in this report for any purpose beyond its intended academic evaluation.

Abstract

This report presents the design and development of **Midadium**, an AI-powered Educational Management Platform built to transform the traditional learning experience in schools into an adaptive, interactive, and student-centred environment. The platform enables teachers to upload structured video lessons and define their instructional goals through guided forms. Students, in turn, can convert the lesson content into personalized formats—including flashcards, interactive games, summaries, and worksheets—based on their individual learning preferences.

A core innovation of Midadium lies in its integration of artificial intelligence. Using Gemini APIs, the platform automatically analyses lesson content, generates dynamic learning resources, and processes student course reviews to offer AI-driven feedback to teachers. Once a course reaches a review threshold, the system synthesizes comments and presents constructive "Do" and "Don't" recommendations to improve content and delivery. This feedback loop enhances the overall learning quality through continuous data-driven refinement.

The platform further includes key educational features such as real-time chat between students and teachers, assignment creation and submission, role-based dashboards (admin, teacher, student), and real-time notification systems powered by Socket.IO. Teachers can manage lessons, assignments, course materials, and student progress through an intuitive interface, while students benefit from flexible content access and engagement tracking.

Midadium was developed as both a mobile application and a web application using **Flutter and Dart**, with a backend powered by **Node.js** and **MongoDB**. Real-time chat functionality is enabled through **Firebase**, while **Socket.IO** powers the platform's notification system. Over a three-month development cycle, the system was implemented with a modular architecture emphasizing scalability, usability, and seamless interaction. The platform demonstrates strong potential for enhancing digital education by offering personalized, intelligent, and accessible learning experiences tailored to diverse classroom needs.

Table of Contents

Table Of Figures

Chapter One: Introduction

1.1 General Background

In the modern era, education is undergoing a fundamental transformation driven by rapid technological advancements and evolving learner expectations. Traditional teaching methods, while foundational, often fall short in addressing the diverse learning styles, pacing preferences, and engagement needs of today's students. With the global rise in digital literacy and the increased reliance on online platforms, there is a growing demand for educational solutions that are not only accessible but also interactive, adaptive, and personalized.

Artificial Intelligence (AI) and educational technology (EdTech) have emerged as powerful tools in bridging the gap between conventional classroom teaching and the dynamic needs of digital learners. Schools and institutions are seeking platforms that not only facilitate content delivery but also support active learning, feedback, and continuous improvement. This shift has led to the development of intelligent platforms that can analyze learner behavior, personalize content, and foster more meaningful interactions between students and educators.

Midadium is born in response to this evolving educational landscape. It is an AI-powered educational management platform designed to enhance the teaching and learning experience through intelligent content conversion, real-time communication, and performance tracking. By leveraging modern technologies and focusing on flexibility and engagement, Midadium aims to redefine how educational content is created, delivered, and consumed within school environments.

1.2 Problem Statement

In traditional school environments, lesson delivery and student engagement often follow a one-size-fits-all model, which limits flexibility, personalization, and adaptability in the learning process. Students with diverse learning preferences may struggle to keep pace or fully grasp content delivered in a fixed format, while teachers lack tools to effectively tailor their lessons to meet individual student needs. Moreover, the absence of real-time feedback, adaptive content formats, and interactive engagement methods leads to reduced student motivation and weaker knowledge retention.

Additionally, many existing educational platforms focus solely on content hosting or communication, offering little to no support for transforming lesson material into multiple learning styles, nor do they empower teachers with intelligent feedback tools to improve their performance.

These challenges highlight the urgent need for a more dynamic, AI-powered educational platform that can address these gaps by enabling content adaptation, encouraging interactive learning, and providing real-time insights for both teachers and students.

1.3 Objectives of the Project

The primary objective of this project is to develop **Midadium**, an AI-driven Educational Management Platform that modernizes the teaching and learning experience in schools by introducing personalized, interactive, and intelligent tools for both students and teachers.

More specifically, the project aims to:

- **Empower Teachers**: Enable educators to upload lesson content (e.g., videos) and structure their material using guided forms that define clear learning objectives and key points.
- **Enhance Student Engagement**: Allow students to interact with lesson content by converting it into different learning formats such as flashcards, interactive games, summaries, or worksheets—based on their individual preferences and learning styles.
- **Incorporate AI Assistance**: Leverage AI models to analyze student reviews and generate constructive feedback for teachers, helping them improve the quality and delivery of their lessons.
- **Facilitate Communication**: Introduce a real-time chat system and notification mechanism using technologies like Firebase and Socket.IO to promote timely interaction and collaboration between teachers and students.
- **Enable Academic Monitoring**: Provide dedicated dashboards for teachers and students to track academic progress, assignment submissions, performance analytics, and overall engagement.
- **Support Administrative Oversight**: Implement user role-based access for admins, teachers, and students to manage courses, users, and academic activities efficiently.

By achieving these objectives, the project intends to bridge the gap between traditional education methods and the capabilities of modern educational technologies.

1.4 Significance of the Project

In an era where technology continues to reshape every aspect of our lives, education stands as one of the most critical fields in need of innovation. Traditional teaching methods often struggle to address the diverse learning needs of students, leading to disengagement, inconsistent performance, and limited scalability. **Midadium** responds to this challenge by integrating artificial intelligence and modern software technologies to deliver an adaptive, interactive, and personalized learning experience.

The significance of this project lies in several key areas:

- **Personalized Learning**: By enabling students to choose how they interact with lesson content—whether through summaries, flashcards, games, or worksheets—Midadium supports different learning preferences and cognitive styles, leading to improved retention and motivation.
- **Teacher Support and Optimization**: The platform provides teachers with structured lesson planning tools and AI-powered feedback generated from student reviews. This not

only enhances the quality of teaching but also allows educators to reflect and improve on their instructional methods.

- **Real-Time Communication and Notifications**: Using technologies like Firebase and Socket.IO, the system ensures immediate feedback loops between teachers and students, fostering a more responsive and engaging academic environment.
- **Data-Driven Insights**: With integrated analytics and progress tracking, educators can monitor student performance in real-time, identify learning gaps, and provide timely interventions.
- **Administrative Efficiency**: The multi-role system (admin, teacher, student) simplifies educational management, enabling scalable deployment across institutions without overwhelming administrative staff.
- **Market Relevance**: With the global EdTech market projected to exceed $400 billion by 2030, Midadium addresses a real and growing need in both public and private educational sectors—particularly in regions seeking affordable, adaptable, and AI-enhanced platforms.

In summary, this project not only delivers a practical solution for modern classrooms but also contributes to the broader goal of democratizing quality education through intelligent technology.

1.5 Organization of the Report

This report is structured to provide a comprehensive and coherent understanding of the development and implementation of the **Midadium** educational platform. Each chapter is designed to guide the reader through the various stages of the project, from background research to practical execution and evaluation.

- **Chapter 1: Introduction**

  Offers an overview of the project, outlines its objectives, highlights its significance, and presents the overall structure of the report.

- **Chapter 2: Theoretical Background and Previous Work**

  Explores existing literature, systems, and technologies relevant to educational platforms and AI-assisted learning, providing the foundation on which Midadium is built.

- **Chapter 3: Methodology**

  Details the technical and procedural approach used in the development of the platform, including tools, frameworks, design decisions, constraints, and ethical considerations.

- **Chapter 4: Results and Analysis**

Presents the key outcomes of the development process, analyzes system performance, and discusses user feedback and system functionality.

- **Chapter 5: Discussion**

  Interprets the results, evaluates their impact in relation to the original objectives, and identifies the strengths and limitations of the platform.

- **Chapter 6: Conclusions and Recommendations**

  Summarizes the findings, highlights lessons learned, and offers suggestions for future development and enhancements.

- **References and Appendices**:

  Includes cited works and supporting materials such as diagrams, code snippets, and extended data that contribute to the report.

This structured format ensures clarity and facilitates a logical flow, allowing readers to follow the progression of the project from concept to conclusion.

Chapter 2: Theoretical Background and Previous Work

2.1 E-Learning and Digital Education

The rapid evolution of digital technologies has fundamentally transformed the landscape of education, giving rise to what is now widely referred to as **e-learning**. E-learning represents the use of electronic technologies to access educational curriculum and learning resources outside the boundaries of a traditional classroom. It offers learners flexibility, accessibility, and a wide range of content formats, making education more inclusive and dynamic.

Over the past decade, e-learning has gained significant traction across schools, universities, and training institutions. The shift has been accelerated by global events such as the COVID-19 pandemic, which forced educational institutions to adopt online learning models at an unprecedented scale. As a result, both educators and learners have become increasingly reliant on digital platforms to facilitate instruction, assessment, communication, and collaboration.

Modern e-learning platforms are often equipped with features such as video lectures, online assessments, discussion forums, assignment submission systems, and progress tracking dashboards. These tools aim to replicate and enhance the classroom experience in a digital environment. However, despite these advancements, many e-learning systems still fall short in terms of interactivity, personalization, and student engagement.

A growing body of research in educational technology highlights the need for more **learner-centered approaches** that adapt to individual needs, learning styles, and performance patterns. This demand has paved the way for the integration of **Artificial Intelligence (AI)** into e-learning platforms, enabling smarter systems that can offer real-time feedback, generate personalized content, and foster a more engaging educational experience.

Midadium responds to this transformation by providing an AI-driven educational platform where content is not only delivered but also **transformed** based on learner preferences. By allowing students to convert lesson materials into summaries, interactive games, or worksheets, Midadium enhances engagement and accommodates diverse learning strategies—marking a significant evolution in digital education.

2.2 Artificial Intelligence in Education (AI-EdTech)

Artificial Intelligence (AI) is revolutionizing nearly every industry, and education is no exception. **AI in education**—often referred to as **AI-EdTech**—introduces intelligent systems that can analyze, adapt, and respond to educational content and learner behavior in ways that traditional systems cannot. This transformation aims not only to automate administrative tasks but also to create deeply personalized learning experiences.

One of the core strengths of AI in education is its ability to **understand and react to individual learning needs**. By collecting data on student performance, engagement, preferences, and progress, AI-powered platforms can dynamically tailor content delivery, provide immediate feedback, and suggest optimal learning paths. For instance, a student struggling with a concept can be automatically redirected to simpler explanations, practice quizzes, or visual aids, while advanced learners can be challenged with more complex material.

AI also enhances **assessment and evaluation** by automating grading, detecting plagiarism, and even analyzing open-ended responses using Natural Language Processing (NLP). Furthermore, AI facilitates intelligent tutoring systems, virtual teaching assistants, and content generation tools that reduce the workload on teachers and increase the availability of support for students.

In Midadium, AI plays a pivotal role in shaping how students interact with educational content. Through the use of advanced models like Google's Gemini API, the platform allows lessons to be **automatically transformed** into various formats—such as summaries, flashcards, games, and worksheets—based on student selection. Additionally, AI is utilized in **course review analysis**, where multiple student reviews are aggregated and processed to generate automated "Do’s and Don'ts" for teachers, enabling continuous teaching improvement through data-driven insights.

This fusion of AI and education not only streamlines learning but also **redefines the teacher-student dynamic**, shifting towards a more collaborative, efficient, and personalized learning ecosystem. AI-EdTech is not about replacing educators, but empowering them—and their students—with intelligent tools that elevate the overall quality of education.

2.3 Related Work and Existing Platforms

Over the past decade, the global landscape of education has witnessed a surge in the development of digital learning platforms. These systems have redefined traditional education by leveraging web and mobile technologies to provide scalable, accessible, and interactive learning experiences. Some of the most well-known platforms include **Google Classroom**, **Edmodo**, **Khan Academy**, **Coursera**, and **Moodle**—each offering distinct approaches to content delivery, student engagement, and course management.

While these platforms successfully support remote and blended learning, they tend to rely heavily on **static content formats** and limited interactivity beyond quizzes, video playback, or forums. Most do not incorporate adaptive learning mechanisms or intelligent content transformation tools. For example, while platforms like **Khan Academy** offer a wealth of educational videos and exercises, the content remains fixed and cannot be dynamically adapted based on a student's learning preference or pace.

More recent platforms have begun experimenting with AI-enhanced features. Tools like **Duolingo** integrate AI to tailor language lessons and optimize practice sessions based on learner behavior. **Socratic by Google** uses AI to help students solve homework questions by analyzing images and providing explanations. However, these applications often focus on narrow subjects or predefined experiences rather than offering a fully flexible and intelligent content generation pipeline.

In comparison, **Midadium** distinguishes itself through a **transformative use of AI and real-time technologies**. Rather than providing static lesson materials, it empowers students to choose how they want to engage with a given lesson—whether as a **summary**, **interactive game**, **worksheet**, or **flashcard deck**—with the transformation being handled instantly by AI models like **Gemini**. Moreover, Midadium incorporates a real-time **chat system** using Firebase, and **Socket.IO-based notification system**, enhancing interaction and responsiveness across the platform.

Another key differentiator is the use of AI for **course quality evaluation**, where aggregated student feedback is analyzed to generate constructive teaching suggestions. This level of **student-centered customization**, coupled with a tri-role system (admin, teacher, student), makes Midadium a **comprehensive, intelligent learning environment** that addresses both content personalization and communication in ways many existing platforms have yet to achieve

2.4 Chat Systems and Real-Time Technologies in Learning Platforms.

Real-time communication technologies have become an essential component of modern digital learning environments. They play a pivotal role in enhancing interactivity, fostering engagement, and supporting collaborative learning. Among the most widely used real-time technologies are **chat systems**, **live notifications**, and **collaborative tools**, which provide immediate feedback, discussion, and support between students, teachers, and peers.

In traditional e-learning platforms, communication is often limited to asynchronous methods—such as forums, emails, or comments on lessons—which can result in delays and reduced learner motivation. To address this gap, many platforms have started integrating **real-time messaging systems**. These systems allow instant question-asking, clarification, and group discussions, simulating a more natural classroom-like interaction in virtual environments.

Technologies such as **Firebase Realtime Database**, **WebSockets**, and **Socket.IO** have become popular choices for implementing real-time features. Firebase, in particular, offers reliable cloud-hosted real-time databases and messaging infrastructure, making it suitable for chat systems, presence detection, and data synchronization. Socket.IO, on the other hand, is a powerful JavaScript library used for enabling real-time, bi-directional communication between client and server. It supports live notifications, activity tracking, and instant updates—critical features in an educational setting.

In the context of **Midadium**, real-time technologies are deeply integrated into the platform's core. A dedicated **chat system built using Firebase** allows students and teachers to communicate instantly, encouraging continuous feedback and learner support. In parallel, **Socket.IO** is used to implement a dynamic **notification system**, which alerts users in real-time about key events such as new lesson uploads, assignment submissions, or important announcements. This ensures that both teachers and students remain up-to-date and engaged without the need for constant manual checking.

The integration of real-time communication not only enhances responsiveness but also contributes to building a **more social and emotionally intelligent learning environment**. It bridges the gap between remote learners and instructors, facilitates collaboration, and ultimately leads to improved learning outcomes.

Chapter 3: Methodology

This chapter outlines the methodologies employed during the design, development, and implementation of the Midadium educational platform. It presents an overview of the planning process, selected technologies, architectural design, and implementation strategies. The methodology was driven by the need to build an engaging, intelligent, and scalable system that meets the demands of modern digital education.

3.1 Preface

3.1.1 Tools, Methods, and Programming Languages

The development of Midadium was guided by a deliberate selection of modern tools, frameworks, and programming languages to ensure robustness, performance, and user satisfaction. The team utilized:

- **Visual Studio Code**: Primary development environment.
- **Git & GitHub**: Version control and collaboration.
- **Postman**: API testing and debugging.
- **Figma/Canva**: UI/UX design.
- **MongoDB Atlas**: Cloud database hosting.
- **Firebase Console**: For chat system integration.
- **JWT (JSON Web Tokens)**: Used for secure authentication and authorization throughout the backend.
- **FFMPEG**: For Extracting audio from uploaded videos by teachers in lessons,
- **Assembly AI**: For audio transcription, it transcripts the uploaded audios extracted from uploaded videos.
- **Gemini API (Google AI)**: For generating intelligent lesson formats and review summaries.
- **Socket.IO**: Real-time notification system between students and teachers.

This combination of tools supported rapid development, clean architecture, and smooth team collaboration.

3.1.2 Client Side

• **Design Approach**: Midadium's client side was developed with a focus on accessibility, responsiveness, and user engagement across different devices (web and mobile).

• **Framework**: The frontend was built using **Flutter**, allowing cross-platform development for both mobile and web apps from a single codebase.

• **Programming Language**: **Dart** was used as the primary programming language, supported by custom widgets and modern Flutter libraries to deliver smooth and beautiful interfaces.

• **Features**:

- Role-based interfaces for Admin, Teacher, and Student.
- Custom dashboards for lesson access, review, and submission.
- Real-time chat using Firebase Firestore.

3.1.3 Database

The backend utilizes **MongoDB**, a flexible NoSQL database, for storing structured and semi-structured data related to users, courses, lessons, assignments, notifications, reviews, and reports.

- **Collections**:
  - `users`: Stores information about students, teachers, and admins.
  - `courses`: Contains metadata about available courses, assignments, and enrollments.
  - `lessons`: Includes lesson details, transcripts, AI-generated content, and files.
  - `assignments`, `submissions`: Linked to students and teachers.
  - `notifications`, `reviews`, `reports`: Supports platform interaction and feedback.
- **Data Structure**: Embedded documents and references were used as appropriate to optimize for fast reads and efficient updates.
- **Indexing**: Applied to high-query fields like `user._id`, `course._id`, and `createdAt` to ensure performance.

3.2 System Features and Implementation

This section outlines the key system features of **Midadium** and describes their implementation across different user roles: **Admin**, **Teacher**, and **Student**. Each role has a dedicated dashboard with specific functionalities. The system is designed following the **MERN stack** (MongoDB, Express.js, React/Flutter, Node.js), with **JWT-based authentication**, **Socket.IO for real-time notifications**, and **Firebase for chat services**. AI capabilities are integrated using **Google Gemini APIs**.

3.2.1 Login and Signup

The login and signup system is essential for secure access control. Midadium uses a **JWT-based authentication mechanism**, ensuring that only authenticated users can access the platform based on their assigned roles (**Admin**, **Teacher**, or **Student**). Each role is routed to its respective dashboard upon login and the user will be welcomed with a welcome page as in Figure 1 Welcome Page

Signup Process

- Users can sign up as **students** by providing their name, email, and password see Figure 2 Signup Page
- **Teachers** and **Admins** are added only by the **Admin** for security and moderation.
- Input validation is performed both on the **client side** (using Flutter) and **server side** (using Express.js with validation middleware).

Login Process

- The login screen accepts email and password Figure 3 Login Page
- On successful login, the backend verifies credentials, generates a **JWT token**, and sends it to the frontend.
- The frontend stores the token securely using **flutter_secure_storage**.
- The token is then used in all subsequent API calls in the `Authorization` header as `Bearer <token>`

Security and Authorization

- Passwords are hashed using **bcrypt.js** before storing in the MongoDB database.
- Each route in the backend is protected using `authMiddleware.js` to ensure role-based access:
    - `Admin` can access admin routes and platform management areas.
    - `Teacher` can access their own courses, lessons, and student data.
    - `Student` can only access their enrolled courses and assignments.

Password Recovery

The password recovery feature in Midadium provides a secure and user-friendly way for users (students, teachers, or admins) who have forgotten their password to regain access to their accounts. This process involves two main screens Figure 4 Forgot password process:

1. Forgot Password Screen

This screen is the initial step for users who cannot recall their password. Its primary function is to collect the user's registered email address to initiate the password reset process.

- **Key Features & User Flow:**
    - **Interface:** Presents a clean and focused interface, displaying the platform logo, a clear title like "Forgot Password?" and an instructional message (e.g., "Enter your email address below, and we'll send you a code to reset your password.").
    - **Email Input:** A single TextFormField allows the user to enter the email address associated with their Midadium account. Input validation ensures a correctly formatted email address is entered.
    - **Submit Action:** A prominent "Send Reset Code button triggers the process.

        - Upon submission, the frontend sends the entered email address to a dedicated backend API endpoint (/api/auth/forgot-password).
        - The backend verifies if an account exists with the provided email.
        - If an account exists, the backend generates a unique, time-sensitive reset code (a 6-digit number), stores it (often hashed) along with an expiry timestamp (10 minutes) in the user's database record, and sends this code to the user's email address.
    - **User Feedback:**
        - **Success:** If the email is found and the code is sent successfully, the user is shown a confirmation message (e.g., "A password reset code has been sent to your email address. Please check your inbox and spam folder."). They are then typically navigated to the "Reset Password Screen."
        - **Failure:** If the email address is not found in the system, or if there's an error sending the email, an appropriate error message is displayed (e.g., "No account found with that email address," or "Could not send reset email. Please try again later.").
    - **Navigation:** Provides a link or button to navigate back to the "Login Screen" if the user remembers their password or wants to abandon the reset process.

2. Reset Password Screen

This screen allows users to set a new password after they have received a reset code via email (from the "Forgot Password" process).

- **Key Features & User Flow:**
    - **Interface:** Displays a title "Reset Password" and shows the email address for which the reset is being performed (passed as an argument from the previous screen).
    - **Input Fields:**
        - **Reset Code Input:** A TextFormField where the user enters the reset code they received in their email. Validation ensures the code is entered and might check for a specific length (e.g., 6 digits).
        - **New Password Input:** A TextFormField for the user to enter their desired new password. Includes standard password requirements (e.g., minimum length) and an icon to toggle password visibility.
        - **Confirm New Password Input:** A TextFormField for the user to re-enter their new password to ensure accuracy. Validation checks if this matches the "New Password" field. Also includes a visibility toggle.
    - **Submit Action:** A "Reset Password" button submits the form.
        - The frontend sends the email, the entered reset code, and the new password to a backend API endpoint (/api/auth/reset-password).
        - The backend verifies:
            - If the reset code matches the one stored for the user.
            - If the reset code has not expired.
        - If valid, the backend hashes the new password, updates the user's password in the database, and invalidates/clears the reset code and its expiry.
    - **User Feedback:**

        - **Success:** If the code is valid and the password is updated, a success message is shown ("Password reset successfully! You can now log in with your new password."). The user is navigated to the "Login Screen."
        - **Failure:** If the reset code is invalid, expired, or if there's any other error, an appropriate error message is displayed ("Invalid or expired reset code," "Passwords do not match," or "Failed to reset password.").
    - **Navigation:** a link back to the Login Screen.

Role-Based Routing

Upon login, the system checks the role embedded in the JWT and navigates accordingly:

- **Admin ➤ Admin Dashboard**
- **Teacher ➤ Teacher Dashboard**
- **Student ➤ Student Dashboard**

Technologies Used

- **Frontend:** Flutter (for web and mobile)
- **Backend:** Node.js with Express
- **Database:** MongoDB (for storing user credentials and roles)
- **Security:** JWT, bcrypt.js, flutter_secure_storage

3.2.2 Admin Dashboard

The Midadium platform adopts a **role-based dashboard system** to provide tailored functionality and user experience for each of the core user roles: **Admin**, **Teacher**, and **Student**. Each dashboard is designed to align with the specific responsibilities and access level of the respective role, ensuring security, usability, and a focused interaction model.

Admin Dashboard

The Admin Dashboard serves as the central control panel for managing the ***Midadium*** Educational Management Platform. It provides administrators with the tools and insights necessary to oversee platform operations, manage users (teachers and students), curate content (subjects and courses), and monitor overall activity. The dashboard is designed to be responsive, adapting its layout for optimal use on both desktop/web and mobile devices. Navigation is facilitated by a side rail on wider screens and a bottom navigation bar on smaller screens.

Overview Screen

This screen provides administrator with an immediate, high-level summary of the platform's current state and key metrics Figure 5 Admin overview in web and mobile.

- Key Features:
    - **Statistical Summary:** Displays vital statistics such as the total number of registered teachers, total students, total courses available, and the total number of active student enrollments across all courses. These are typically presented in visually distinct cards for quick comprehension.
    - **Quick Actions:** Offers direct shortcuts to frequently performed administrative tasks, such as "Add New Subject," "Add New Teacher," and "Add New Student." These actions navigate the admin to the relevant management sections or open appropriate creation dialogs.
    - **Recent Activity Feed**: Presents a clist of recent significant actions performed on the platform, such as new teacher registrations, course approvals/rejections, student enrollments, or new content submissions. This allows admins to stay informed about ongoing activities without navigating to specific sections.

2. Subjects Screen

This screen allows administrators to define, organize, and manage the academic subjects offered on the Midadium platform. Courses are now categorized under these subjects Figure 6 Subject management screen.

- **Key Features:**
    - **Subject Listing**: Displays all created subjects, typically showing the subject name and a brief description.
    - **Add New Subject:** A Floating Action Button (FAB) or an "Add" button allows admins to open a dialog to create a new subject by providing its name and an optional description.
    - **Edit Subject:** Each listed subject has an option (an edit icon) to modify its name and description Figure 15 edit Subject.
    - **Delete Subject:** An option to delete a subject, typically with a confirmation dialog. Deletion is usually restricted if courses are currently assigned to that subject, prompting the admin to reassign or delete those courses first.
    - **Navigation to Subject Detail:** Tapping on a subject in the list navigates the admin to the Subject Detail Screen, where they can view and manage courses specifically within that subject.

3. Subject Detail Screen

provides a focused view of a single subject, allowing administrators to manage the courses offered within that subject and edit the subject's own details Figure 11 Subject Details screen.

- **Key Features:**
    - **Subject Information:** Displays the name and description of the selected subject. An "Edit" action allows modification of these details (reusing the subject creation/edit dialog).
    - **Course Listing (Filtered by Subject):** Lists all courses that belong to the currently viewed subject. This list is the primary way courses are now managed within a subject context.
    - **Course Status Filters:** Provides filters (e.g., "All," "Pending," "Approved," "Rejected") to refine the list of courses shown for the current subject.
    - **Add New Course (to this Subject):** A prominent "Add Course" button allows the admin to create a new course that will automatically be associated with the current subject. The creation dialog will include fields for course name, description, assigning a teacher (from a list of available teachers), setting a price, and providing syllabus/resources.
    - **Course Actions: Each listed course has actions available:**
        - **Navigate to Course Detail**: Tapping a course opens the Course Detail Screen for more granular management Figure 12 Course details Screen**.**
        - **Edit course Details.** Figure 13 edit course
        - **Approve/Reject**: Buttons to quickly approve or reject courses directly from the list (if they are pending or to change their status).
        - **Delete:** Option to delete a course (with confirmation), subject to enrollment checks.

4. Teachers Screen

manage teacher accounts, view their details, and oversee their assigned courses Figure 7 Manage teachers**.**

- **Key Features:**
    - **Teacher Listing**: Displays a list of all users with the 'teacher' role, showing their username and email. Includes a search bar to find specific teachers.
    - **Add New Teacher:** A FAB "Add" button opens a dialog to create a new teacher account (username, email, password).
    - **View/Edit Teacher Details:** Tapping a teacher navigates to the Teacher Detail Screen.

- **Assign Courses to teachers:** in the teacher details screen the admin can assign courses to this teacher.
- **Delete Teacher:** Each teacher in the list has a delete option. This triggers a confirmation process, especially if the teacher has assigned courses. The admin is presented with options:
  - **Cancel deletion.**
  - **Delete teacher only (orphaning their courses, which will have their teacher field set to null).**
  - **Delete teacher AND their associated courses.**

5. Teacher Detail Screen

provide a comprehensive view of a single teacher, including their profile information and the courses they are assigned to teach Figure 16 Teacher Detail Screen.

- **Key Features:**
  - **Teacher Profile:** Displays the teacher's username and email. An "Edit Profile" button allows the admin to update these details**.**
  - **Assigned Courses List:** Lists all courses currently assigned to this teacher. Each course item shows its name and subject.
  - **Navigate to Course Detail:** Tapping on an assigned course navigates to the Course Detail Screen.
  - **Assign Course to Teacher:** A button opens a dialog listing all *approved* courses that are *not currently assigned* to this teacher, allowing the admin to assign them Figure 14 Assign Course to Teacher.
  - **(Implicit) Unassign Course:** This is typically handled by editing a course (via Course Detail Screen) and changing its assigned teacher or by deleting a course assigned to this teacher.

6. Students Screen

manage student accounts and view a list of all students on the platform Figure 8Students management.

- **Key Features:**
  - **Student Listing:** Displays a list of all users with the 'student' role, showing their username and email. Includes a search bar.
  - **Add New Student**: A FAB "Add" button opens a dialog to create a new student account (username, email, password). The "Grade" field has been removed from this process.

- **View/Edit Student Details**: Tapping a student navigates to the Student Detail Screen Figure 22 Adding and editing Students.
- **Delete Student:** Each student in the list has a delete option with confirmation. Deleting a student also removes their active enrollments and any pending enrollment requests.

7. Student Detail Screen (Admin View)

For administrators to view a specific student's profile and manage their course enrollments directly Figure 17 Student details and management.

- **Key Features:**
  - **Student Profile**: Displays the student's username and email. An "Edit Profile" button allows the admin to update these details.
  - **Active Enrollments List:** Lists all courses the student is currently and actively enrolled in (fetched from the Enrollment collection). Each item shows the course name, subject, and enrollment date.
  - **Navigate to Course Detail:** Tapping an enrolled course navigates to the Course Detail Screen.
  - **Manually Unenroll:** An option on each enrolled course to immediately unenroll the student from that course (admin override).
  - **Manually Enroll:** A button opens a dialog listing all *approved* courses the student is *not currently enrolled in*, allowing the admin to enroll them directly, bypassing the student request/payment flow**.**

8. Reports Screen

provide administrators with visual summaries and insights into platform data and performance Figure 9 Platform Reports Screen and Figure 10 Platform Reports**.**

- **Key Features:**
  - Reported problem sections that navigated to the reported problems in the platform but users where the admin will navigate to a screen that contains a list of problems reported Figure 18 Problem Report Screen where he can filter the by type or status Figure 19 Problem reports Filtering and see the user's description of the problem and try to resolve them Figure 20 Report Details.
  - **Course Status Overview**: A pie chart showing the distribution of courses by status (Pending, Approved, Rejected), along with total course count.
  - **Courses per Subject:** A bar chart showing the number of courses under each subject.

- **Courses per Teacher:** A bar chart showing the number of courses managed by each teacher (top N teachers).
- **Student Engagement: a**verage course completion rates.

10. Settings Screen

To allow the logged-in administrator to manage their own profile settings Figure 21 Admin Settings.

- **Key Features:**
  - **Admin Profile Information:** Displays the admin's username and email.
  - **Edit Profile:** Option to update their username and email.
  - **Change Password:** A dedicated button to navigate to a screen where the admin can change their own account password (requires current password and new password confirmation).

Problem Report System

A built-in reporting system allows users (teachers or students) to submit **problem reports** regarding platform issues or errors.

- Admins can view all reports in a dedicated panel.
- After investigating, they can mark a report as resolved.
- Upon resolution, a **real-time notification** is sent to the user confirming the fix.

This promotes a sense of responsiveness and trust in the platform.

3.2.3 Teacher Dashboard

The **Teacher Dashboard** in *Midadium* is a personalized control center tailored for educators, providing them with comprehensive tools to manage their courses, students, and instructional content efficiently. It is designed for ease of use, task centralization, and performance tracking.

Overview Page

The first landing page for teachers is the **Overview**, which presents essential information in an intuitive and visually appealing layout Figure 53 Teacher Overview:

- **Key Statistics Cards**: Display the total number of:
  - Enrolled students
  - Assigned courses
  - Uploaded lessons

    - Pending or unread notifications
- **Quick Tools**: Provide instant access to frequently used actions such as:
    - Uploading a new lesson or assignment
    - Viewing submissions
- **Recent Activities**: Highlights the teacher's latest actions, such as content uploads, course updates, or student interactions.

This overview empowers the teacher with a bird's-eye view of their teaching performance and classroom activity.

My Students Page

This page displays students dynamically Figure 55 Students, based on **course enrollment**:

- If a student is enrolled in a course, and that course is assigned to a specific teacher, the student will automatically appear in the teacher's **My Students** page.
- Teachers can view detailed student profiles and track engagement or progress in specific courses Figure 56 Student Details.

This ensures that teachers only see students relevant to their assigned courses, keeping the experience personalized and focused.

My Courses Page

In this screen Figure 57 My Courses.Teachers can:

- **Submit new courses** for review by the admin.
- Once a course is submitted, its state is marked as **pending approval**.
- Upon approval, the course becomes active and fully manageable by the teacher.

Clicking on a course redirects the teacher to the **Course Detail Page**.

Course Detail Page

This page provides complete control over course content and interaction Figure 59 Course Details and Figure 60: Course Details Cont.

- **Course Info**: Title, description, subject, status, and student count.
- **Enrolled Students**: List of students registered in the course.
- **Lessons Management**:
    - Upload new lessons (video-based with auto-transcription Figure 61: Add Lesson.
    - Edit or delete existing lessons Figure 64: Edit Lesson.
    - View uploaded lesson Figure 65: Video Viewer
- **Assignments Management**:
    - Add, edit, or delete assignments Figure 62: Add Assignment and Figure 63: Edit Assignment.

    - View submissions for each assignment Figure 66: View Submission.
    - **Review and Rate Submissions**: Teachers can provide written reviews and rate students using a 5-star system Figure 67: Review Submission.
- **Course Chat**: A dedicated real-time chatroom for course participants, enabling interactive discussions and Q&A .
- **AI-Powered Review Analyzer**: Uses Gemini AI to summarize course reviews submitted by students and provides constructive recommendations to the teacher.

Performance Reports

The **Reports** page offers visual insights into teaching activity and lesson delivery Figure 68: Performance Report

- Total number of lessons uploaded for the current week.
- The most active day of the week.
- Average lesson uploads per day.

These statistics are visualized using dynamic bar charts, helping teachers monitor and optimize their performance over time.

Settings Page

The **Settings** section Figure 69: Settings allows teachers to:

- View and update their profile Figure 70: Profile.
- Change their account password Figure 71: Change Password.
- Enable or disable real-time **notifications** (powered by Socket.IO).

3.2.4 Student Dashboard

The **Student Dashboard** in *Midadium* is designed to enhance learner engagement and autonomy, enabling students to access educational content, interact with teachers and peers, and take advantage of intelligent features that support self-paced learning and progress tracking.

Home Page

Upon login, students are greeted with a personalized **Home Page** that serves the primary dashboard for students, it provides a personalized engaging overview if their learning process Figure 23 Student Home Screen.

The Student Home Screen is structured into several sections:

- **Personalized greeting and header** with a welcoming greeting message with the student's name, and also this header contains a quick access button for **notifications.**
- **Notifications**: Real-time updates regarding course approvals, lesson uploads, assignment reviews, and announcements.
- **Integrated search Bar:** below the greeting area, allows students to search for content across the platform:
    - **Real time filtering:** as the student types either the subjects if he searches for a subject will be displayed or if for a course he enrolled in the My enrolled courses list is dynamically filtered to show course matching his query.
    - **Search Submission:** On submitting on the search and usually if no courses are found for what he is searching for, the student can type whatever they like, and after submitting the search which will take him to another screen Figure 41 AI Enhanced search. where the ai will take his query and give him back all the related courses to whatever he wants to Figure 42 AI Search results with AI insights

- **Explore Subjects slider**: A horizontally scrollable slider that features all the available subjects in the platform and each subject has its own catalogue screen where a student can see the courses and their prices.
- **Recommended For You slider**: it features a personalized recommended courses for students that have already enrolments, the ai takes the list of his courses and the available courses and recommend up to 5 courses.

  If the student is new and have no enrolled courses yet, the recommendations will feature the most popular courses in the platform based on enrolment counts/ratings.

- **My enrolled courses list**: this section lists all the courses the student is enrolled in, for quick access for each course screen.
- **Chatbot button**: navigates to the platform ai assistant where u can ask him anything about the platform and it will help you indeed.

ChatBot page

From the home screen there is a robot icon this icon will take the user to the chatbot page where the platform ai assistant lays, in this screen the student can chat with ai and ask whatever he like about the platform Figure 51 Chatbot and Figure 52 Chatbot.

we used gemeni ai to make this work where we give it list of our courses and subjects and teachers and using this data it will only answer questions related to **Midadium**

Notification page

On the top of the home screen there is an icon that navigates to this page that displays Real-time updates regarding course approvals, lesson uploads, assignment reviews, and announcements Figure 24 Student Notification Screen.

Course Catalog Page

Each subject from the subject slider in the home screen that lays under the explosre subjects section, has a course catalog screen where students can browse and discover all available and approved courses on the Midadium platform Figure 26 Subject's Available courses screen (Course catalog).

- Key features and Functionality:

Course Listing: Displays a grid or list of available courses. Each course is typically presented as a CourseCatalogCard.

Filtering (by Subject): If navigated from a subject slider, the catalog is pre-filtered to show only courses belonging to that specific subject. The AppBar title dynamically reflects the selected subject or "All Available Courses."

Search Integration: A prominent StudentSearchBar is placed below the AppBar, allowing students to type queries to filter the displayed courses in real-time (client-side filtering of the fetched list, or potentially a backend-supported search within the catalog).

Course Card Details: Each CourseCatalogCard displays essential information like the course name, subject (if not already filtered by it), price, and a representative icon/image.

Navigation to Preview: Tapping on any course card navigates the student to the Student Course Preview Screen for that specific course, where they can see more details and choose to enroll.

Pull-to-Refresh: Allows students to refresh the list of available courses.

Course Preview Page

This screen provides detailed public-facing information about a specific course before a student decides to enroll or purchase it. It's designed to convert interest into enrolment Figure 25 Course preview screen (Before Enrollment).

- Key Features and Functionality:

Dynamic Header: Features a large, immersive header image (ideally specific to the course's subject, with Top-left back button.
Content Card: A white content card with rounded top corners slides up from below the header image.
Course Information: Displays the course name, teacher's name, info chips (lesson count,estimated duration), and a detailed "About Course" description.

Tabbed Interface (Overview, Reviews, Teacher):

Overview: Shows the info chips and course description.
Reviews: Fetches and displays user-submitted reviews for the course, including ratings (star display) and comments. Loads reviews on demand when the tab is selected.
Teacher: Displays information about the course instructor (name, email) and lists other approved courses taught by that same teacher, allowing students to explore more from a preferred instructor. Loads teacher details on demand.
Enrollment Action: A prominent, fixed "Enroll Now" button is displayed at the bottom of the screen.
The button text dynamically shows the course "Enroll Now (Free)" for free courses.
Tapping this button initiates the enrollment process by calling StudentService.requestEnrollment(). If the course is paid, this service call returns a clientSecret from Stripe, and the app navigates to the StudentPaymentScreen.
If the course is free, the student is enrolled directly, and a success message is shown. with a dialog that contains two buttons one to take him home if he wants and the other takes him directly to the enrolled course screen.

Student Payment Page

This page Provides a secure and user-friendly interface for students to complete payments for paid courses or course packages from roadmaps using Stripe Figure 27 Course Payment Process.

Key Features and Functionality:

Course/Package Summary: Displays a summary of the item being purchased (course name, package description, total price).

Payment Method Selection: Shows icons for various payment methods (Visa, Mastercard, Apple Pay, PayPal), with "Card" being the primary implemented method. The selected method is visually highlighted.

Stripe CardField (or CardFormField): Integrates Stripe's secure UI element for collecting card number, expiry date, and CVC. Cardholder name is collected via a separate TextFormField.

- **Input Validation:** Uses a Form and validators for the cardholder name and relies on Stripe's CardField for its internal validation.

Payment Processing:

- On tapping "Pay," it calls Stripe.instance.confirmPayment() using the clientSecret (obtained from the backend during enrollment request/package initiation) and the card details from the form.
- Handles different PaymentIntentsStatus (Succeeded, Canceled, Failed).

Success/Error Feedback: Shows appropriate messages (PaymentSuccessDialog) based on the payment outcome.

Navigation: On successful payment Figure 28 Payment success on stripe dashboard, the student can go to the course he purchased or go to the home screen.

Student Course Detail Page (Enrolled)

This is the main learning interface for a course the student is *already enrolled in*. It provides access to lessons, assignments, reviews, and other course-related information Figure 29 Course Details.

- Key Features and Functionality:

Layout: Similar to StudentCoursePreviewScreen with a header image (specific to the course subject) and an overlapping content card.

And it also has a **chat button** where each course has its own chatroom where the teacher and all the enrolled students can chat in and discuss about the course or its lessons or assignments!

Tabbed Interface:

Overview: Course description, info chips (lesson countduration), and *integrated teacher information* (name, email, list of their other approved courses).

Lessons:

Displays an overall course progress bar for completed lessons Figure 30 Lessons tab.

Lists all lessons for the course using LessonListItemCard.

Each lesson card shows its number, title, a status icon (playable, completed, processing), and a "Mark as Complete/Incomplete" toggle button.

Tapping a playable lesson card navigates to the StudentLessonPlayerScreen.

The "Mark as Complete" button updates the lesson's progress status both optimistically in the UI and via a backend call (StudentService.updateLessonProgressStatus) Figure 36 Course progress Tracking

Reviews: Allows viewing all reviews for the course and provides an interface for the student to write, edit, or delete their *own* review (using _showReviewDialog and relevant service calls). A "Write a Review" button/bar appears if they haven't reviewed yet Figure 38 Course reviews.

Assignments: Displays a list of all assignments for the course. For each assignment, it shows the title, due date, and the student's submission status (Not Submitted, Submitted, Overdue, Graded - with grade if available). Tapping an assignment navigates to StudentAssignmentDetailScreen Figure 40 Assignments tab

Favorite Button: An icon button in the header area allows students to add/remove the course from their favorites list, with state managed via StudentService functions: add Course to Favorites and removeCourseFromFavorites.

Report Problem: A button to open a dialog for reporting issues with the course Figure 37 Report a problem to Admin.

Pull-to-Refresh: Implemented to allow students to refresh all course data, including lesson progress, reviews, and assignments.

Assignment Submission Page

This screen serves as the dedicated interface for students to view the requirements of a specific assignment for a course they are enrolled in, and to submit their work (text-based content and/or file uploads). It also allows them to view or edit their previous submissions if permitted Figure 39 Assignment Submission Screen.

Access: Navigated to when a student taps on an individual assignment from the "Assignments" tab within the StudentCourseDetailScreen.

Key Features and Functionality:

- Assignment Information Display:

Clearly presents the assignment's title, detailed description, and the dueDate (formatted for readability, with an "Overdue" indicator if applicable).

This information is retrieved from the Assignment model.

- Submission Form:

Text Input: Provides a multi-line TextFormField for students to type in text-based answers or comments related to their submission.

File Upload: Includes a "Select File" / "Replace File" button that uses the file_picker package to allow students to choose a file from their devices.

Displays the name of the currently selected/submitted file.

Allows removal of a selected/existing file.

Frontend validation for file size (e.g., max 10MB) is implemented as an initial check.

- Submission Handling:

A "Submit Assignment" (or "Update Submission" if editing) button triggers the submission process.

The screen validates that either text content or a file is provided.

It calls the appropriate StudentService method (submitAssignment or updateSubmission).

The service method sends the text content and/or file to the backend using an HTTP POST or PUT (MultipartRequest for files).

The backend saves the submission (text and file path) to the SubmissionModel in MongoDB.

Viewing/Editing Existing Submissions:

If the student has already submitted work for this assignment (initialSubmission is passed to the screen), the form fields are pre-populated with their previous content and submitted file name.

The main action button changes to "Update Submission."

A "Delete Submission" button appears (typically in the AppBar or near the submission area), allowing the student to remove their submission after confirmation.

Loading and Feedback: Displays loading indicators during file picking and submission processing. Provides success or error messages to the user via SnackBars.

Lesson Player Page

The primary screen for students to consume individual video lessons and interact with AI-powered learning tools Figure 31 Lesson Screen And the available Ai tools for generation.

Key Features and Functionality:

Video Playback: Integrates the chewie package for a rich video player experience with standard controls (play/pause, seek, volume, fullscreen).

Lesson Information: Displays the lesson title, learning objectives, and keywords directly below the video player for context.

AI Learning Tools Grid: Presents buttons for available AI formats: "Summary," "Flashcards," "Quiz Game," "Worksheet."

Tapping a button triggers _generateAIFormatAndNavigate.

Navigation to AI Format Screens: After the AI successfully generates the content, the screen navigates to the respective dedicated screen (Summary Screen, Flashcards Screen, Quiz Game Screen, WorkSheet Screen.) to display it.

Loading & Error States: Shows loading indicators while AI content is being generated and displays errors if generation fails.

AI Format Screens

(SummaryScreen, FlashcardsScreen, QuizGameScreen, WorksheetScreen)

Each screen provides a dedicated, optimized UI for interacting with a specific type of AI-generated content

.**Key Features and Functionality:**

Themed AppBar: Consistent gradient AppBar displaying the lesson title and the type of format. with a button to **translate** the form into one the four languages: (Arabic, French, Deutsch, Español)

Content Display:

Summary: Displays text, potentially with Markdown rendering, and a "Copy to Clipboard" action Figure 32 Summary AI format with options to copy and AI translation.

Flashcards: Uses CardSwiper for a flippable card interface with "Term" and "Definition" sides. Includes controls for next/previous and flipping. Implements saving of "known" and "still learning" card progress to the backend via StudentService.updateFlashcardProgress Figure 33 Flashcards AI format.

Quiz Game: Displays multiple-choice questions one at a time. Allows students to select answers and submit the quiz. Shows a results view with score. Submits quiz attempts and scores to the backend via StudentService.submitQuizAttempt Figure 34 Quiz Game AI format With AI Feedback for Wrong Answers .

Worksheet: Displays open-ended questions with guidelines. Provides TextFormFields for answers. Includes a progress bar for answered questions. **Submits answers for AI evaluation** (or just saves them) via StudentService.submitAndEvaluateWorksheet or submitWorksheet. **Displays AI feedback and scores per question after evaluation** Figure 35 Worksheet Format with AI evaluation (score out of 5 + feedback)**.**

Language Translation: Each of these screens includes a PopupMenuButton in the AppBar allowing students to select a language.

On selection, it calls StudentService.translateGeneratedContent(), sending the original English content of that format and the target language to ai so it translates it.

The screen then re-renders with the translated content. The original English content is preserved in the state for subsequent translations.

Course Chat Screen

provide a real-time communication channel for each course, allowing enrolled students and the course teacher to interact, ask questions, share insights, and build a learning community Figure 43 Course Chatroom.

Access: Typically accessed via a "Chat" tab within the StudentCourseDetailScreen for an enrolled course.

Key Features and Functionality:

Real-time Messaging: Leverages Firebase Firestore to send and receive messages instantly. New messages appear automatically for all participants in the chatroom.

Message Display: Messages are displayed in chat bubbles, distinguishing between the current user's messages (e.g., aligned to the right, different color) and messages from other participants. Each bubble shows the sender's name, the message text, and a timestamp (e.g., "5 min ago").

Message Input: A TextField at the bottom of the screen allows users to type and send messages. A send button is enabled when text is present.

Scrolling: The message list is scrollable, with new messages appearing at the bottom and the view automatically scrolling (or providing an easy way to scroll) to the latest message.

User Identification: Displays the sender's name for each message. The system uses the student's or teacher's authenticated identity.

My Courses Page

This section displays all courses in which the student is actively enrolled.and it also includes a search bar so he can search for what he wants Figure 44 Student My Courses Screen.

My Favorite courses Page

This page lists all the favorite courses the student selected earlier Figure 45 My favorite Courses Screen.

Learning Roadmaps Section:

This section can be accessed from the student dashboard navigation bar in which it a dedicated "Roadmaps" section Figure 46 Learning Roadmaps Powered by AI.

The first thing the student will see on these sections is to option: either a button to generate a new roadmap or a button for already saved roadmaps:

AI Roadmap Generator Page:

This page allows students to define a learning goal and optionally their current knowledge level, and then request the AI (Gemini) to generate a personalized learning roadmap using courses available on the platform.

Key Features and Functionality:

- **Input Fields:**
    - A TextField for the student to describe their "Learning Goal" (e.g., "Become a proficient Flutter developer," "Master data science fundamentals").
    - Optional: ChoiceChips or a DropdownButton for the student to indicate their "Current Knowledge Level" (e.g., Beginner, Intermediate, Advanced, Not Sure).
    -

Roadmap Display Page

present the AI-generated or previously saved learning roadmap to the student in a visually appealing, organized, and "catchy" timeline format Figure 47 Roadmap Result with info and suggested courses.

- **Access:**
    - Navigated to from StudentRoadmapGeneratorScreen after a new roadmap is created.

- Navigated to from StudentSavedRoadmapsScreen when a student views one of their saved roadmaps.

- **Key Features and Functionality:**
  - **Roadmap Visualization (Timeline Design):**
    - Displays the learning goal and (if provided) the initial knowledge level at the top.
    - Each phase/step of the roadmap is presented as a distinct visual block or card, arranged sequentially to create a timeline effect (e.g., "Space Mission" design).
    - Each step card shows:
      - Phase number and phaseTitle.
      - estimatedDuration (e.g., using a chip with a timer icon).
      - An "info" icon that, when tapped, reveals the AI's justification for that phase in a dialog or an expandable section.
      - A list of **Recommended Courses** for that phase. Course names are displayed.
      - Each recommended course is tappable, navigating to its Student Course Preview Screen.
      - **Course Selection (Dropdown/Chooser):** If an AI phase recommends multiple courses, a dropdown or a "Choose Course" button allows the student to select *one* primary course for that phase if they wish to focus (this selection is stored in _selectedCourses state).
  - **Save Roadmap:** An IconButton (bookmark icon) in the AppBar allows students to save a newly generated roadmap to their profile. This button is disabled or changes appearance if the roadmap is already saved. Calls StudentService: saveLearningRoadmap().
  - **Purchase Roadmap Package:** A prominent button (FloatingActionButton) allows the student to initiate the purchase of all *selected* (or all recommended, un-enrolled, paid) courses in the roadmap as a single package Figure 48 Roadmap course package purchase.
    - This calculates the total price (with potential package discounts applied on the backend).
    - Calls StudentService.initiateRoadmapPackagePurchase().

- Navigates to StudentPaymentScreen with a single clientSecret for the package total.

Saved Roadmaps Page

Allows students to view, manage, and revisit all the learning roadmaps they have previously generated and saved Figure 49 Saved Roadmaps Screen.

- **Access:** from the main student dashboard.
- **Key Features and Functionality:**
  - **List of Saved Roadmaps:** Fetches and displays all roadmaps saved by the student using StudentService.getSavedRoadmaps().
  - Each list item shows key information like the learningGoal and the date it was saved.
  - **View Roadmap:** Tapping a saved roadmap navigates to the Student Roadmap Display Screen, passing the savedRoadmapId (and potentially the basic roadmap data for initial display while full details load).
  - **Delete Roadmap:** Each list item has a delete icon/button that calls the Student Service : deleteSavedRoadmap() after confirmation.
  - **Pull-to-Refresh:** Allows refreshing the list.

My Profile Page

This page holds the student info like his email and username , in the top left there is an edit button if the student wants to edit his info and there is button for changing password that takes him to a dedicated page to do so and it also has a log out button Figure 50 Profile screen and change password screen.

AI-Enhanced Learning Tools

One of the core features of *Midadium* is the intelligent learning experience, powered by **Gemini AI**, which allows students to:

- **Transform lesson content** into alternate formats such as:
  - Flashcards
  - Summaries
  - Interactive games
  - Worksheets
- **Request format generation** per lesson, ensuring diverse and adaptive learning styles are accommodated.

Assignment Submission

For each enrolled course:

- Students can access available **assignments**, download instructions, and submit their responses directly from the dashboard.
- The platform supports **resubmission** and file updates.
- Submissions are reviewed and rated by the teacher, and feedback is made available through the same interface.

Chat System

Each course includes a dedicated **chatroom**:

- Enables students to communicate with the course teacher and other enrolled peers.
- Supports real-time discussions and collaborative learning.
- Built using **Firebase Realtime Database** for fast, responsive interactions.

Problem Reporting

Students have access to a built-in **problem reporting system** that allows them to:

- Describe bugs, issues, or user experience problems.
- Select relevant categories and attach supporting screenshots if needed.
- Admins receive and respond to these reports, with status updates and replies visible to the student.
- Notifications inform students when issues are resolved.

3.2.5 Course Management and Enrollment

Course management and enrollment form the core of the Midadium educational platform, facilitating the connection between teachers and students through a structured and controlled process. This system ensures quality content delivery while allowing for administrative oversight.

Course Creation and Approval Workflow

Teachers have the ability to create new courses through their dedicated dashboard. When a course is submitted, it enters a **pending** state. This ensures that courses are not visible to students until reviewed and approved by an admin.

Once the admin approves a course, it becomes active and available for student enrollment. This step guarantees content quality and consistency across the platform.

- Teachers fill in essential details such as course name, description, subject, grade level, and price.
- Submitted courses remain invisible to users until approved.
- Teachers can track the approval status of each course.

Admins are responsible for managing the lifecycle of every course. They can review submissions, approve or reject courses, assign or reassign teachers, and edit or delete any course from the system as needed.

Student Enrollment Process

1. **Course Discovery & Preview:**
    - Students first discover courses through various means: browsing subjects, viewing the course catalog, exploring recommendations (popular, AI-personalized), or seeing courses listed by a specific teacher.
    - Upon selecting a course they are interested in but not yet enrolled in, they are navigated to a **Course Preview Screen**. This screen provides public-facing details such as the course name, description, teacher, subject, lesson count, estimated duration, price, and existing student reviews.
2. **Initiating Enrollment:**
    - On the Course Preview Screen, an "Enroll Now" button is displayed.
    - If the course has is free the button will indicate that "Enroll Now (Free)".
    - Tapping this button triggers an API call from the student's app to the backend (POST /api/student/courses/:courseId/request-enrollment).
3. **Backend Enrollment Logic:**

- The backend receives the request and verifies the student's authentication (via JWT) and the validity of the courseId.
- It checks if the student is *already enrolled* in the course to prevent duplicates.
- It checks if the course is approved for enrollment.
- **For Free Courses:**
  - If the course price is zero (or negligible), the backend directly creates an Enrollment record in the MongoDB database, linking the studentId and courseId.
  - It then automatically adds the student's ID to the members array in the corresponding Firebase Firestore chatrooms/{courseId} document, granting them access to the course chat.
  - A success message is sent back to the frontend, indicating immediate enrollment.
- **For Paid Courses:**
  - If the course has a price, the backend interacts with the **Stripe API** to create a PaymentIntent for the specified course amount.
  - Crucial metadata (including studentId, courseId, courseName, teacherId, and purchaseType: 'single_course') is attached to this Payment Intent.
  - The backend returns the clientSecret of this Payment Intent to the Flutter app, along with an indication that payment is required.

4. **Frontend Payment (for Paid Courses):**
   - The Flutter app receives the clientSecret and navigates the student to the dedicated StudentPaymentScreen.
   - This screen uses the flutter_stripe package to display a secure payment form (e.g., CardField).
   - The student enters their payment details.
   - The app uses Stripe.instance.confirmPayment() with the clientSecret to process the payment directly with Stripe.
5. **Payment Confirmation & Webhook Processing (for Paid Courses):**
   - If the Stripe payment is successful, the StudentPaymentScreen shows a success message and typically navigates the user back.
   - **Crucially, Stripe sends a payment_intent.succeeded webhook event to the Midadium backend** (to the /api/webhooks/stripe endpoint).

- The backend's webhook handler verifies the event's signature.
- It extracts the studentId, courseId, teacherId, and purchaseType from the Payment Intent's metadata.
- It then creates the Enrollment record in MongoDB for the student and the paid course.
- Finally, it adds the student's ID to the members array in the Firestore chatrooms/{courseId} document.

6. **Access Granted:**
    - Once the Enrollment record exists (either directly for free courses or after webhook processing for paid courses), the student is officially enrolled.
    - They can now access the course content (lessons, assignments, AI tools) and participate in the course chatroom. The frontend UI (e.g., "My Courses" list, course detail screen) will reflect this new enrollment status upon data refresh.

This process ensures a secure and clear path for students to join courses, whether they are free or require payment, and seamlessly integrates them into the course's learning environment and community features.

Course Detail Page

After enrollment, both teachers and students gain access to the course detail page, which acts as the central hub for course interaction and content management.

For teachers:

- View all enrolled students.
- Upload and manage lessons and assignments.
- Access a real-time chatroom for course communication.
- Use AI tools to analyze student feedback and generate review summaries.

For students:

- View the about course section and the teacher info and other courses assigned to him
- View structured lessons and multimedia content with progress tracking for each course by the number of finished lessons.
- Complete and submit assignments.
- Access interactive AI-powered tools for learning enhancement.
- Communicate with teachers and peers through the course chatrooms.
- Apply reveiws and edit/delete them.

Security and Validation

All course and enrollment functionalities are protected using secure role-based authentication powered by JWT. MongoDB ensures reliable and scalable data storage, while the backend enforces proper validation to protect against unauthorized access.

This structured approach to course management ensures fairness, transparency, and high educational standards across the platform.

3.2.6 Lesson and Assignment Management

Lesson and assignment management are central to Midadium's role as an educational platform, enabling teachers to deliver structured content and assess student progress. The platform provides intuitive tools for uploading multimedia lessons, assigning tasks, and handling student submissions.

Lesson Management by Teachers

Each approved course allows its assigned teacher to upload lessons directly from the **Course Details** screen. Teachers can:

- Upload video files via a dedicated UI using multipart form data.
- Add structured metadata, including title, objectives, keywords, and description.
- Use integrated **Gemini AI** to automatically:
    - Generate transcripts from video using speech-to-text (via AssemblyAI).
    - Summarize the content into flashcards, worksheets, quizzes, and games.
- Edit or delete any lesson after uploading.

This process ensures that learning materials are rich, interactive, and accessible. All lessons are stored securely on the server and linked to the corresponding course and teacher.

Assignment Management by Teachers

Teachers can also create and manage assignments from the **Course Details** view:

- Upload assignment files (PDF, DOC, or structured form-based assignments).
- Specify a title, description, and due date.
- Edit or delete assignments as needed.

Assignments appear on the student's dashboard under the relevant course. Each assignment includes instructions and a submission link.

Student Assignment Submission

Students can view and submit assignments directly through their dashboard. Features include:

- File upload interface for submitting documents, images, or code files.
- Real-time validation to ensure format and size constraints.
- Option to update or delete a submission before the due date.

All submissions are stored in the database and associated with the assignment and student. The teacher is notified in real time using **Socket.IO**.

Review and Feedback

Teachers can access all student submissions through the **"View Submissions"** panel:

- Open and download submission files.
- Add a textual review or feedback.
- Provide a star rating or numeric score.
- Submit the feedback, which is instantly available to the student.

This feedback loop ensures accountability and allows students to continuously improve based on instructor insights.

AI-Powered Enhancements

Each lesson uploaded is enriched with smart features:

- **AI Format Conversion**: Students can choose to transform lesson content into:
  - Flashcards
  - Summaries
  - Games
  - Worksheets.
- **Quiz Attempts and Evaluation**: Worksheets and quizzes submitted can optionally be evaluated using AI for feedback.
- **Enhanced search using ai**: in the student dashboard the student can search for whatever he likes other than subjects/courses and the ai will suggest him related courses that can be purchased for this purpose.
- **Recommendations**: an AI-driven component where Google Gemini analyzes the student's enrolled course profiles (summaries) and a catalog of other available courses to suggest complementary skills or deeper dives. And for New students' popular courses depending on reviews/enrolment counts are recommended.

3.2.7 Real-Time Chat and Notifications

To foster continuous communication and engagement within the Midadium platform, two core features were implemented: a **real-time chat system** and a **real-time notification system**. These components enhance the interactivity between teachers and students and provide immediate feedback on system actions.

A. Real-Time Chat System

The platform includes a dedicated chatroom for each course. Once a student is enrolled in a course and the course is approved, they can access its chatroom from the **Course Details** page.

Features of the chat system include:

- **One-to-many communication:** Teachers and all enrolled students can communicate in a shared space.
- **Firebase Integration:** Chat is built using **Firebase Realtime Database**, which provides real-time message updates and synchronization.
- **Message history:** Messages are saved and persisted for each course, so new users joining a course can review past discussions.
- **Typing indicators and timestamps:** Enhances usability and clarity during conversation.
- **Role-based display:** Different styling and tags are applied to messages based on user role (e.g., Teacher, Student).
- **Firebase Firestore:** Used as the backend for storing and synchronizing chat messages. Each course has a dedicated "chatroom" document (ID matching the course ID), which contains a subcollection of "messages."
- **Firebase Authentication** (via Custom Tokens): Students and teachers are authenticated with Firebase using custom tokens minted by the application's backend (Node.js) after their primary JWT login. This allows Firestore Security Rules to enforce access control based on the user's ID and their membership in the course's chatroom (stored in the members array of the chatroom document).
- **Firestore Security Rules:** Ensure that only enrolled students and the teacher for a specific course can read and write messages in that course's chatroom. The senderId in each message must match the authenticated user's ID.
- **StreamBuilder (Flutter):** Used on the frontend to listen to real-time updates from the Firestore messages subcollection and rebuild the UI instantly when new messages arrive.

This system promotes active discussion, question clarification, and community learning within each course.

B. Real-Time Notifications

Midadium employs a notification system to alert users (students or teachers) of important actions across the platform. This system is built using **Socket.IO** for real-time delivery and a MongoDB-based notification model for persistence.

Key Events Triggering Notifications:

- **When a teacher uploads a lesson or assignment**, enrolled students receive a notification.
- **When a student submits an assignment**, the teacher receives a notification.
- **When an admin responds to a student's report**, a notification is sent to the reporting student.
- **When a teacher's course is approved or rejected**, a notification is sent to the teacher.

Notification Features:

- Delivered instantly using **WebSockets** via Socket.IO.
- Persisted in the database so users can view their history of alerts.
- Include metadata such as:
    - Title
    - Message body
    - Link to relevant page (e.g., lesson, assignment)
    - Read/unread status
- Notifications can be marked as read or deleted by the user.

3.2.8 AI-Powered Features

One of Midadium's most innovative aspects is the seamless integration of artificial intelligence to enhance learning and feedback. These AI-powered features were built using **Google Gemini API**, and other api's, allowing dynamic content transformation and review summarization across the platform.

1. **Automated Lesson Transcription:**

   How It Works:

- Video lessons uploaded by teachers are processed to extract audio.
- This Audio is converted into accurate text transcripts using a sophisticated Speech-to-Text AI service **AssemblyAI**.

Benefits:

Provides a readable version of the lesson, crucial for accessibility, searching within content, and as the foundation for other AI tools.

2. **AI-Generated Content Formats (from Lesson Transcripts):**

   Once a lesson transcript is available, students can instantly generate various learning materials:

   a. **Summaries:** The AI condenses lengthy lesson transcripts into concise, easy-to-digest summaries, highlighting key concepts and learning objectives. This aids in quick review and grasping core ideas.
   b. **Flashcards:** The AI identifies important terms, definitions, and question-answer pairs from the lesson content and automatically structures them into digital flashcards. This supports active recall and memorization.
   c. **Interactive Quizzes (Games):** Based on the lesson material, the AI formulates multiple-choice questions with options and correct answers, providing students with an engaging way to self-assess their understanding and identify areas needing further review.
   d. **Worksheets:** The AI generates open-ended or short-answer questions designed to encourage critical thinking and deeper application of the lesson's concepts. Guidelines are often provided to assist students.

3. **AI-Powered Worksheet Answer Evaluation :**

   a. **How it Works:** When students submit their answers to AI-generated worksheets, the platform can use an LLM to evaluate these open-ended responses. The AI compares the student's answer against the lesson transcript and objectives.
   b. **Benefit:** Provides students with immediate, conceptual feedback and a score for their written answers, offering insights beyond simple right/wrong grading and helping them understand their grasp of the material.

4. **Ai-Powered Quiz Feedback:**
   a. How it works: When students submit their answers to AI-generated quizez,and the result comes the platform can use an LLM to track the wrong answer and then. The Ai will gi the student a feedback in what he did bad and will suggest him what to review and study to avoid wrong answer

   b. **Benefit:** Provides students with immediate, conceptual feedback and for their answers, offering insights beyond simple right/wrong grading and helping them understand their grasp of the material.

5. **AI-Enhanced Course Search & Recommendations (Personalized Learning Paths):**
   a. **Search:** LLMs can be used to understand the semantic meaning of student search queries, going beyond simple keyword matching to provide more relevant course results.
   b. **Recommendations:**

i. **Content-Based:** AI analyzes the student's current enrollments (subjects, course content) and a catalog of available courses.

6. **Roadmap Generation:**
Trigger: Student visits the AI Roadmap tab and types a learning goal (e.g., "I want to learn Python").

Action: The system matches the request with existing content and available courses.

The Gemini API structures a step-by-step roadmap with course recommendations, goals, and outcomes.

a. Students can input a learning goal, and the AI (Gemini) constructs a personalized, multi-phase learning roadmap, suggesting a sequence of relevant courses from the platform's catalog, complete with justifications and estimated durations for each phase.

Benefit: Helps students discover the most suitable courses for their needs, guides their learning journey effectively, and can suggest complementary skills or advanced topics based on their profile.

7. **AI Course Review Analysis (Teacher Feature)**

To help teachers evaluate student satisfaction and improve course quality, Midadium offers a smart feature called **"Analyze Course Reviews."**

Trigger: On the course detail page, the teacher clicks "Analyze Reviews."

Action:

- The platform collects all written reviews submitted by students for that course.
- These reviews are aggregated and analyzed by the **Gemini API**.
- The AI returns:
  - A general sentiment analysis.
  - Common praise points.
  - Areas of improvement.
  - A final recommendation summary.

Benefits:

- Helps teachers respond to student needs more effectively.
- Automates what would otherwise be a manual and subjective review process.
- Provides measurable insights for continuous improvement.

3.2.9 Assignment Management and Submissions

Midadium offers a streamlined and structured workflow for managing assignments, allowing teachers to create, edit, and monitor assignments, and enabling students to submit their work and receive feedback—all within the platform.

A. Assignment Creation (Teacher)

Teachers can create assignments directly from the **Course Details** page. Each assignment includes:

- **Title** and **description**
- **Due date**
- **Associated course**
- **Optional attachments** (e.g., images, PDF instructions)

Once created, assignments are listed under the relevant course and become visible to all enrolled students.

B. Student Submission

Students can view all active assignments from their enrolled courses. For each assignment:

- A **"Submit"** button allows students to upload their solution (text or file).
- After submission, the file is stored securely, and the student receives a confirmation.
- Students can also **update or delete** their submissions before the deadline.

C. Teacher Review and Feedback

Teachers can access all submissions per assignment and perform the following actions:

- View or download the submission
- Assign a **rating (out of 5)** for performance
- Provide **written feedback or review**
- Submissions are stored with **timestamp** for submission tracking

Feedback is instantly visible to the student and is accompanied by a **notification**.

D. Real-Time Notifications

Midadium leverages **Socket.IO** to notify both teachers and students of relevant assignment activities:

- Student receives a notification when an assignment is added or reviewed.
- Teacher receives a notification when a student submits a new assignment.

E. Benefits

- Centralized tracking of assignments and deadlines
- Transparent grading and timely feedback
- Encourages student accountability and teacher responsiveness

3.2.10 Reports and Analytics

Midadium integrates comprehensive reporting and analytics tools to support teachers in monitoring their performance and understanding student engagement. These tools are essential for evaluating teaching effectiveness and guiding instructional decisions.

A. Performance Reports

Within the teacher dashboard, a dedicated Reports section provides visual insights into weekly activity. Teachers can track:

- The total number of lessons uploaded during the current week
- The most active day of the week based on lesson uploads
- The average number of lessons uploaded per day

These metrics are displayed through interactive bar charts, enabling teachers to quickly assess their teaching pace and consistency.

The backend supports this functionality through a weekly statistics API that dynamically retrieves and aggregates relevant data, ensuring real-time updates.

B. Student Activity Monitoring

While primarily focused on teacher performance, the platform also provides visibility into student activity, including:

- Which students have viewed specific lessons
- Submission status for assignments
- Overall patterns in engagement across different courses

These insights allow teachers to identify inactive students early and intervene as needed.

C. AI-Based Review Summaries

Midadium leverages Gemini AI to analyze student reviews submitted for courses. Once a minimum number of reviews are collected, the system processes them to produce a summary containing:

- Key strengths of the course as perceived by students
- Common areas for improvement
- Consolidated feedback trends for reflection

This AI-driven feature supports teachers by offering objective analysis of qualitative feedback, helping improve future course delivery.

D. Significance

The reporting and analytics system offers several benefits:

- Enables data-informed decisions for content development
- Encourages reflective teaching practices
- Highlights trends in student engagement
- Supports early identification of students who may require additional support

The integration of visual analytics and artificial intelligence ensures that feedback is not only collected but also meaningfully interpreted.

3.3 Standards and Specifications

The development of the Midadium educational platform adhered to a set of technical standards, software architecture conventions, and best practices to ensure reliability, scalability, and maintainability. While no formal international standards such as ISO or IEEE were mandated, the system was built upon widely recognized design principles and software development guidelines to ensure quality and compliance.

3.3.1 Authentication and Security Standards

- **JWT (JSON Web Token):** The platform uses JWT to manage secure user authentication and session control. JWT ensures stateless, encrypted communication between the client and server, following RESTful security practices.
- **Password Hashing:** User credentials are hashed using industry-standard encryption techniques before being stored in the database, protecting user data against unauthorized access.
- **HTTPS Protocol:** All communication between the client and server is conducted over HTTPS, safeguarding data transmission.

3.3.2 API Design

- **RESTful API Architecture:** The backend follows RESTful conventions for structuring endpoints, enabling predictable, scalable, and consistent communication between the frontend and backend services.
- **Versioning and Modularity:** APIs are modular and can be extended or versioned in the future, in line with common software engineering standards.

3.3.3 UI/UX Standards

- **Cross-Platform Compatibility:** The UI is built using Flutter and adheres to Material Design Guidelines, ensuring accessibility and usability across devices and screen sizes.
- **Accessibility Practices:** Font contrast, responsive layouts, and logical navigation structures have been considered to enhance user experience and accessibility.

3.3.4 Real-Time Communication

- **Socket.IO Protocol:** For real-time notifications and chat features, Socket.IO was implemented following its event-driven communication model, ensuring smooth bi-directional data flow and low-latency user interaction.

3.3.5 Database Standards

- **NoSQL (MongoDB) Schema Design Best Practices:** Collections and documents follow structured, indexed, and normalized formats where appropriate, allowing for scalable data handling, fast queries, and logical relationships between entities (e.g., courses, users, lessons).

3.3.6 AI Integration Guidelines

- **Google Gemini APIs:** Integration of AI-powered features such as lesson format generation and course review summarization follows Google's AI usage standards for ethical and controlled deployment of generative models.

3.4 Constraints

Throughout the development of Midadium, several practical and theoretical constraints influenced the design, implementation, and deployment of the platform. Addressing these constraints was essential to ensure a functional, scalable, and ethical educational system.

1. Economic Constraints

- As a student-led project, Midadium was built with **limited financial resources**.
- Free and open-source technologies such as **Node.js, MongoDB Atlas (free tier), Firebase (free tier), and Gemini API (with limited usage)** were selected to minimize operational costs.
- Hosting, storage, and AI usage were optimized to avoid exceeding free quotas while still delivering rich functionality.

2. Technological Constraints

- The system needed to function across both **web and mobile** platforms, which imposed design and performance challenges.
- Some advanced features like **AI content generation** depended on third-party APIs, which introduced **rate limits, latency, and dependency risks**.
- Real-time features such as **chat and notifications** required proper synchronization using **Socket.IO and Firebase**, which demanded careful backend coordination.

3. User Experience Constraints

- The platform targets a diverse user base: **administrators, teachers, and students**, including those with **limited technical skills**.

- Thus, Midadium had to prioritize **ease of use, clarity, and accessibility** without compromising on advanced features.

4. Social and Ethical Constraints

- Since the platform collects and processes student feedback, chat messages, and course content, it was crucial to:
    - Respect **privacy and data security**.
    - Ensure **AI-generated content does not promote bias or misinformation**.
    - Provide moderation tools for reporting problems and resolving disputes fairly.

5. Health and Safety

- The system was built to **minimize digital fatigue** by encouraging interactive and gamified learning formats.
- AI tools such as **flashcards and visual formats** were designed to promote retention and cognitive ease.

6. Scalability and Sustainability

- The architecture (Node.js + MongoDB + Firebase + Gemini API) supports **future upgrades** and **horizontal scaling**.
- Codebase and database structure were designed to accommodate new features such as **quizzes, exams, live sessions**, and **advanced analytics**.

3.5 Standards and Specifications

In developing Midadium, a set of software development standards, best practices, and design principles were followed to ensure functionality, maintainability, security, and user-centered design. While the project did not rely on formal engineering standards such as IEEE or ISO due to its software-based nature, it adhered to widely accepted technical and ethical guidelines relevant to modern web and mobile platforms.

1. Software Development Standards

- **RESTful API Design**: All backend endpoints followed RESTful conventions to ensure consistency, scalability, and ease of integration.
- **Model-View-Controller (MVC) Architecture**: The backend code was structured using MVC principles to enhance modularity, separation of concerns, and maintainability.
- **JWT Authentication**: Secure user login and session management were implemented using **JSON Web Tokens (JWT)** to protect user identities and prevent unauthorized access.

2. User Interface Standards

- **Responsive Design**: The frontend was built using **Flutter**, ensuring that the interface is responsive across devices and screen sizes.
- **Accessibility Guidelines**: UI/UX designs took into account accessibility principles, such as contrast, button sizes, and readable fonts, to serve a broad user base including students with visual limitations.

3. Security Practices

- **Input Validation and Sanitization**: All forms and user inputs are validated and sanitized to prevent attacks such as XSS and injection.
- **Role-Based Access Control (RBAC)**: Permissions are strictly defined for each user type (admin, teacher, student) to prevent unauthorized operations.
- **HTTPS Communication**: The system is designed to be hosted over secure HTTPS protocols.

4. Real-Time Communication Standards

- **Socket.IO** was used for real-time messaging and notifications, following event-driven programming standards.
- **Firebase Realtime Database** was used for chat systems, following structured and indexed data access practices for real-time synchronization.

5. AI Integration Guidelines

- The AI features were implemented using **Google Gemini API**, following safe prompting practices and response filtering to ensure age-appropriate, unbiased, and relevant output.
- AI responses were reviewed and stored with contextual references to maintain transparency.

6. Database Design Standards

- **MongoDB Schemas** were defined with validation rules to enforce data consistency and reduce the risk of runtime errors.
- Data relations (e.g., courses ↔ lessons, teachers ↔ students) were structured to reflect real-world educational hierarchies and operations.

7. Collaboration and Version Control

- All project code was managed using **Git** and hosted on **GitHub**, adhering to commit message conventions, branching models (feature branches, main), and pull request review standards.

Chapter 4: Results and Analysis

This chapter presents the results of Midadium's implementation, evaluating system performance, feature access, and user feedback across all roles. It also highlights how well the platform met its goals and areas for improvement.

4.1 System Functionality Results

1. User Authentication and Role-Based Access

- The login and signup processes function as expected, securely routing users to role-specific dashboards.
- JWT token-based authentication ensures secure and persistent sessions.
- Only authorized roles (Admin, Teacher, Student) access their respective features, and no cross-role leakage was observed during testing.

2. Course Management and Enrollment

- Admins successfully created and managed subjects and courses.
- Teachers could submit new courses and manage their lessons and assignments after approval.
- Students were able to browse available courses, submit enrollment requests, and track approval status.

3. Lesson and Assignment Handling

- Teachers were able to upload video-based lessons and create assignments seamlessly.
- Students could view lessons, interact with content, and submit assignments within deadlines.
- Assignment submissions were stored and visible to both teachers and students, supporting revisions and feedback exchange.

4. Real-Time Features

- The chat system, powered by Firebase, enabled real-time interaction between teachers and students in each course.
- Socket.IO-based notifications delivered real-time alerts for actions such as lesson uploads, assignment feedback, and system responses to reports.

5. AI Integration

- Students used AI tools (via Gemini API) to transform lessons into alternative formats (flashcards, games, summaries, worksheets).
- Teachers utilized the "Analyze Reviews" tool to get sentiment analysis and improvement suggestions based on course feedback

4.2 Performance Evaluation

Feature Tested	Result	Response Time	Notes
Login/Signup	100% successful	< 2 sec	No delay or errors
Course Submission by Teacher	Functional & Approved by Admin	< 4 sec	Approval required
Lesson Upload (Video + AI)	Functional	5–10 sec	Includes transcription and AI processing
Assignment Submission	Functional	< 3 sec	Multiple file formats tested
Real-Time Notifications	Instant Delivery	< 1 sec	Socket.IO setup confirmed
Chat Messaging	Real-time, no lag	< 0.5 sec	Firebase sync successful
AI Format Generation (Student)	Functional, accurate formats	~5–8 sec	Based on lesson length

4.3 User Feedback and Usability Insights

After user testing with students and teachers:

- **Students appreciated**:
  - The flexibility of AI-generated lesson formats.
  - The clarity of the dashboard and ease of access to lessons and assignments.
  - Real-time notifications and chat support.
- **Teachers valued**:
  - The ability to track student progress visually.
  - Automated review analysis for improvement.
  - The structured process for course management.
- **Admins reported**:
  - Efficient problem-report tracking and resolution flow.
  - Comprehensive activity logs for audit purposes.

4.4 Key Achievements

- **99% feature implementation** as per planned scope.
- Seamless integration between backend (Node.js, MongoDB) and frontend (Flutter).
- Stable and real-time communication and alerts across modules.
- AI features provided value-added functionality for both learning and evaluation.

Chapter 5: Discussion

The development of the Midadium platform successfully addressed the initial objectives of creating an AI-powered educational system that supports students, teachers, and administrators through dedicated dashboards and intelligent tools.

5.1 Problem Resolution

The core challenge—designing a system that personalizes education, simplifies communication, and automates administrative tasks—was effectively resolved. Midadium integrates role-based access, real-time chat, smart notifications, and AI tools to enhance the learning process.

5.2 Project Contributions

Key contributions of the system include:

- AI-generated learning formats (e.g., flashcards, summaries, games).
- Smart course review analysis for teachers.
- Real-time feedback and progress tracking.
- A fully integrated web/mobile experience using Flutter and MERN stack.

These features distinguish Midadium from many existing e-learning platforms by offering tailored support and automation based on user role.

5.3 Implications

The results demonstrate that AI-enhanced content improves engagement and comprehension. Real-time features like chat and notifications support collaboration, while the structured dashboards enable role clarity and focus.

5.4 Limitations

Some limitations identified include:

- Limited scalability testing under high traffic.
- Lack of full offline access for mobile users.
- Manual admin approval still needed for some processes.

Chapter 6: Conclusions and Recommendation

6.1 Conclusions

This project successfully introduced **Midadium**, an AI-powered educational platform tailored for schools. Through its modular design and role-based dashboards for Admins, Teachers, and Students, the platform achieved its core goals of enhancing content delivery, enabling intelligent content transformation, and streamlining communication and management in a digital learning environment.

Key accomplishments include:

- A secure authentication system with role-based access.
- AI features like lesson transformation and review analysis using Google Gemini API.
- Real-time communication via Firebase and Socket.IO.
- Dynamic course management, student tracking, and performance analytics.

Midadium not only fulfills the technical and functional requirements, but it also sets the foundation for more adaptive and personalized learning experiences in the future.

6.2 Recommendations

To improve Midadium and expand its potential:

- **Enhance mobile capabilities**, including offline access and push notifications.
- **Automate more admin tasks**, such as intelligent course approval suggestions.
- **Incorporate AI-driven recommendations** for student learning paths.
- **Integrate gamification elements** to increase student motivation.
- **Continue usability testing** with real students and teachers to refine the user experience.

6.3 Future Work

Building on the current system, future development could explore:

- A mobile-native app with AI voice interaction.
- Integration with national or regional education systems.
- Advanced analytics dashboards with machine learning insights.
- AI teaching assistants or tutors trained on the curriculum

Figures

Figure 1 Welcome Page

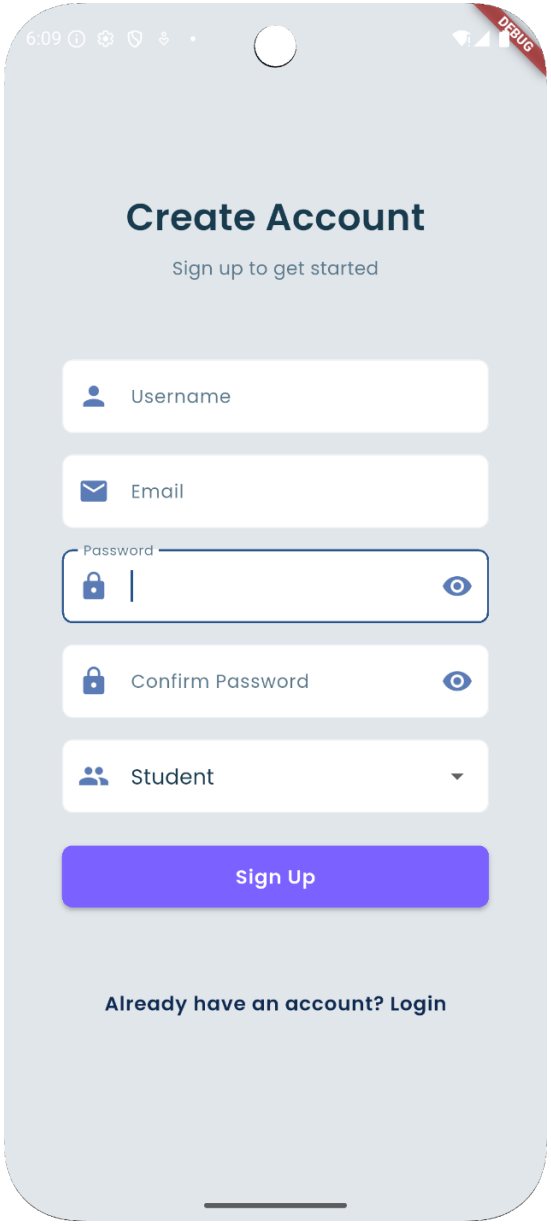


Figure 2 Signup Page

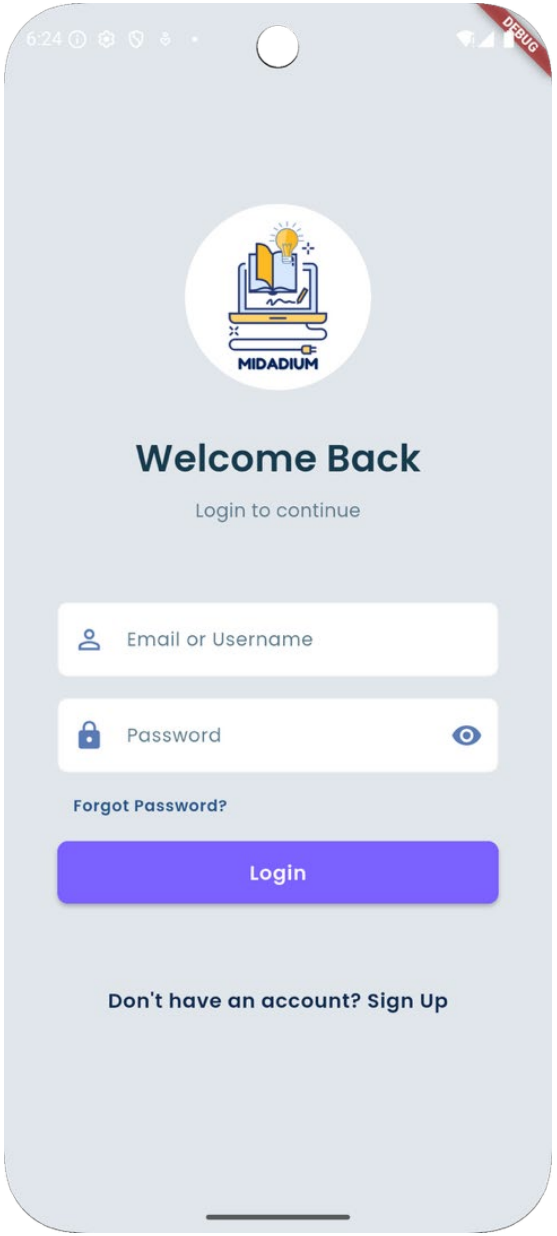


Figure 3 Login Page

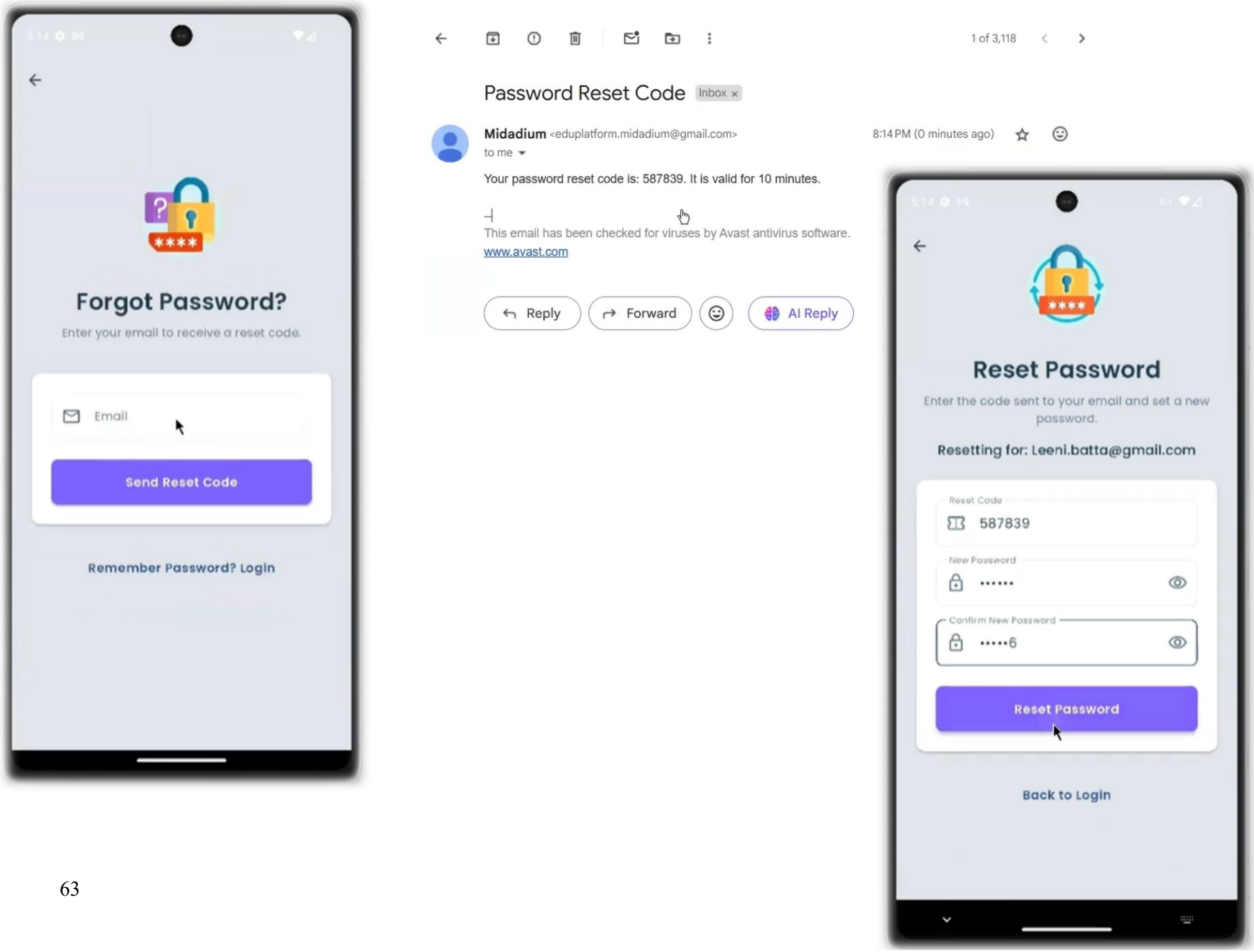


Figure 4 Forgot password process

Admin Dashboard:

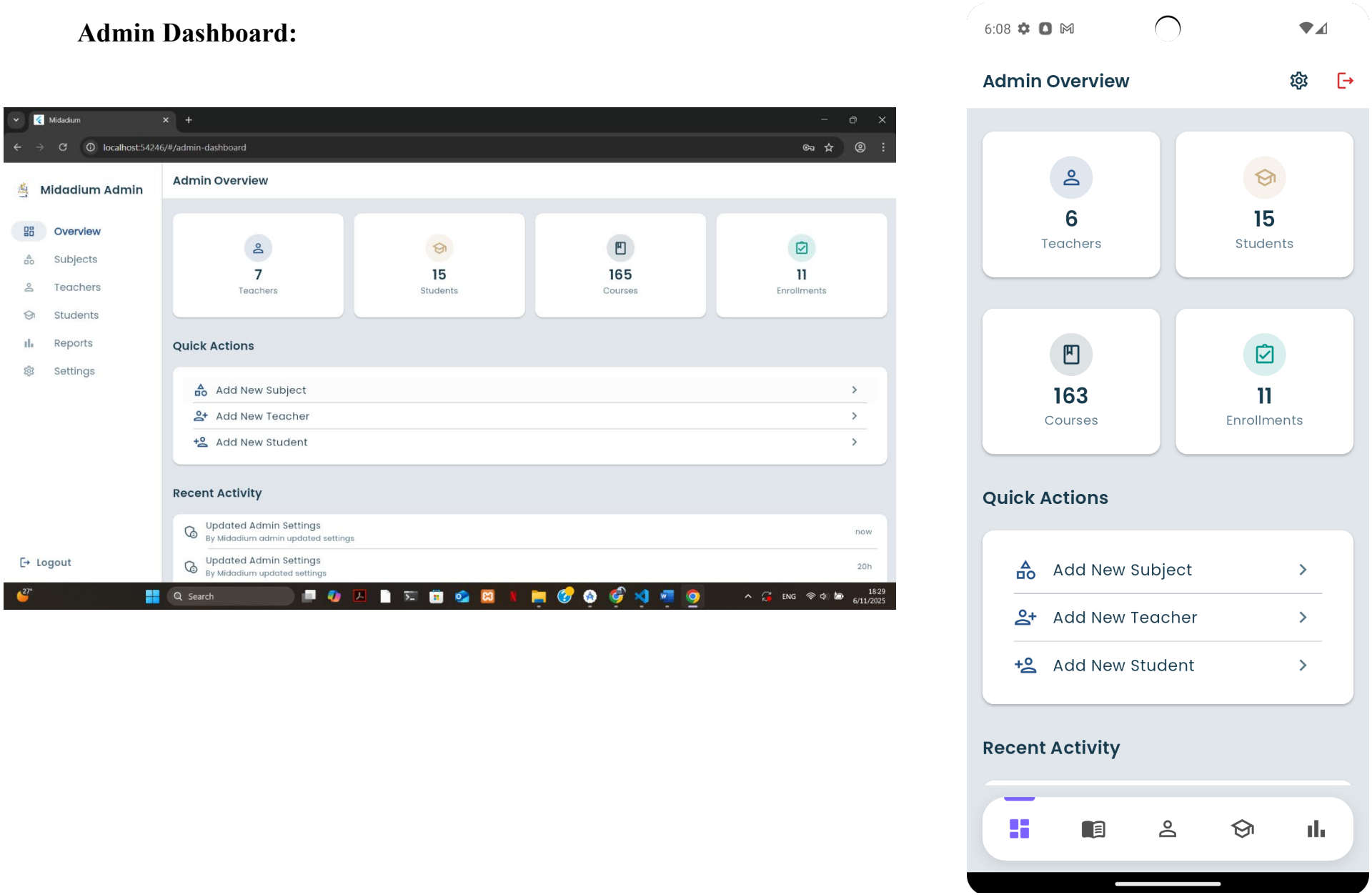
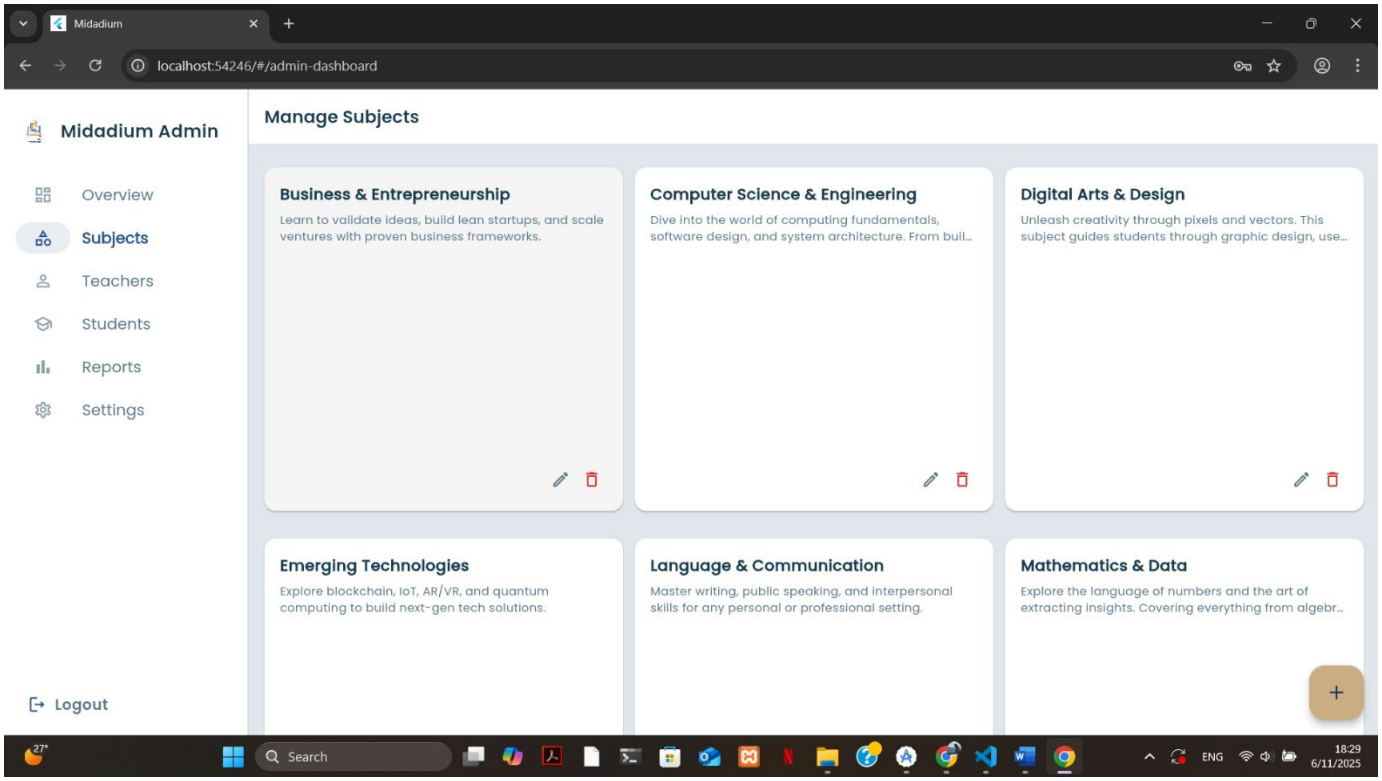


Figure 5 Admin overview in web and mobile



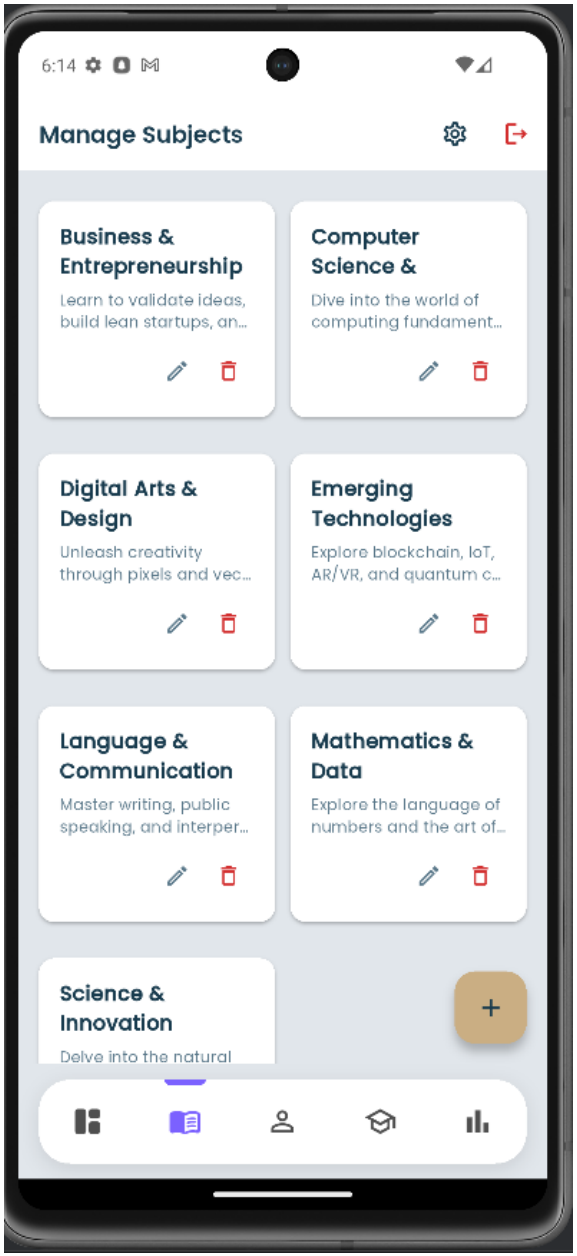
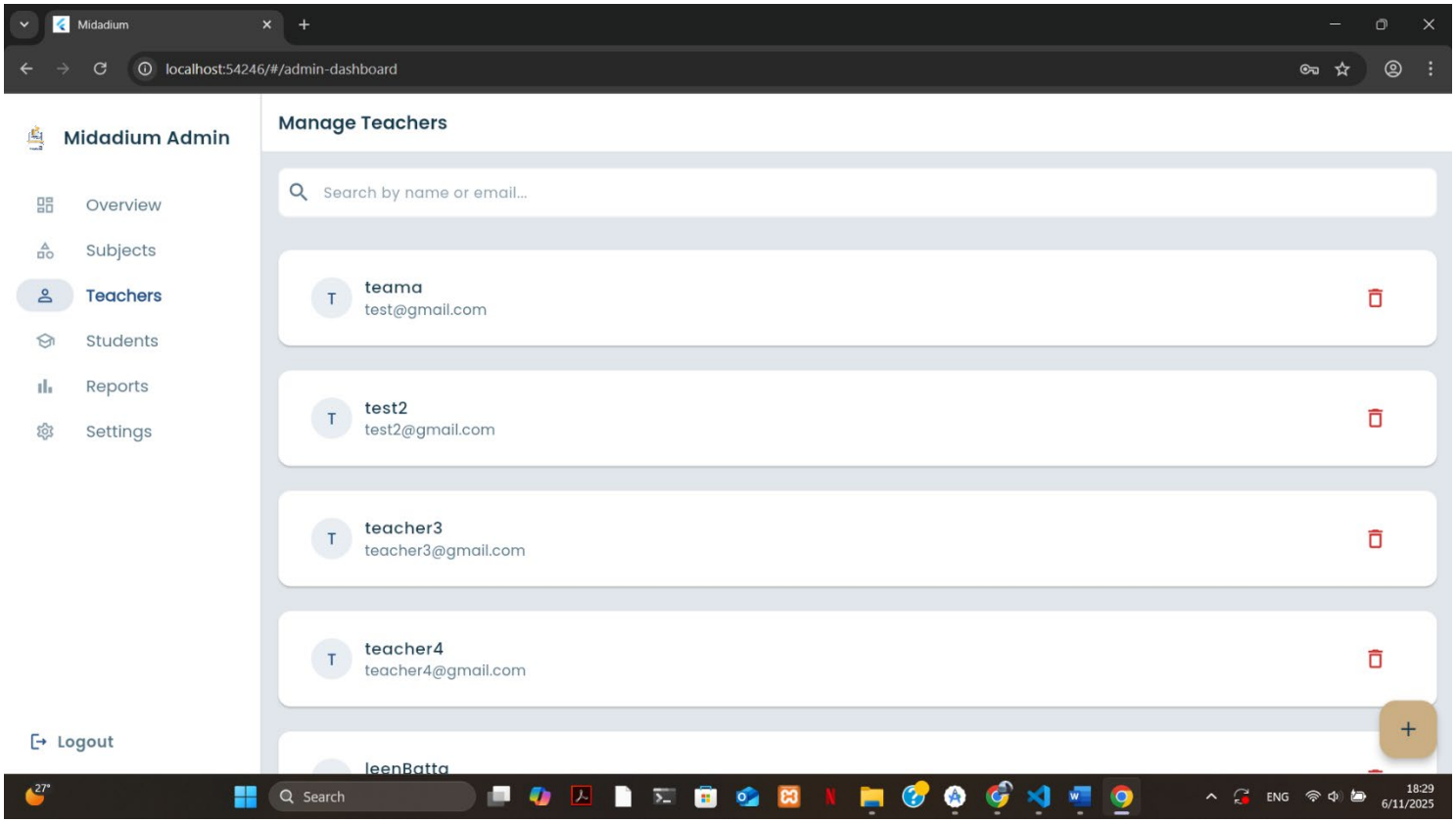


Figure 6 Subject management screen



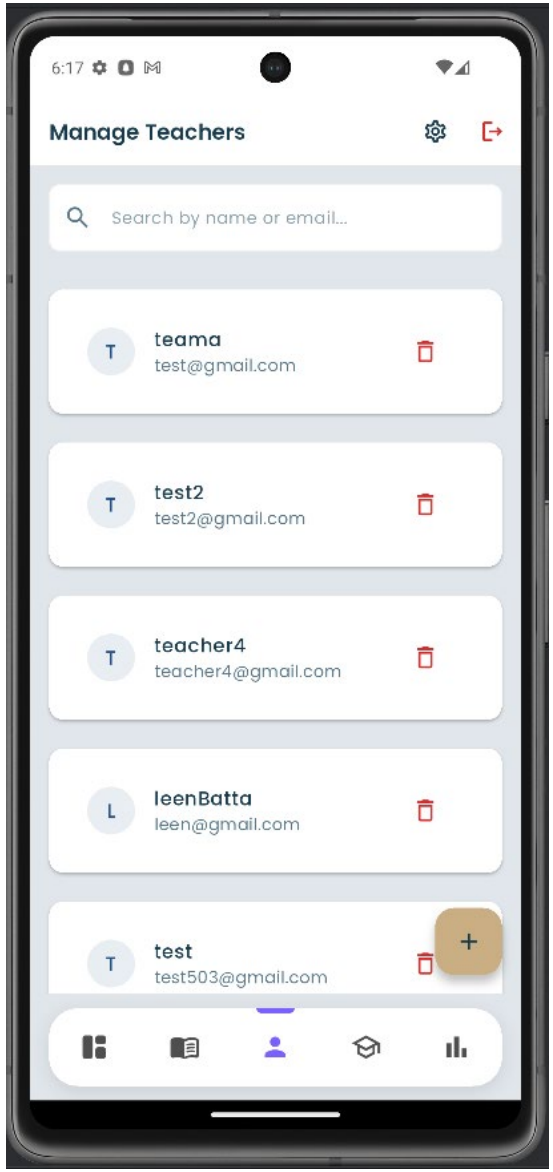
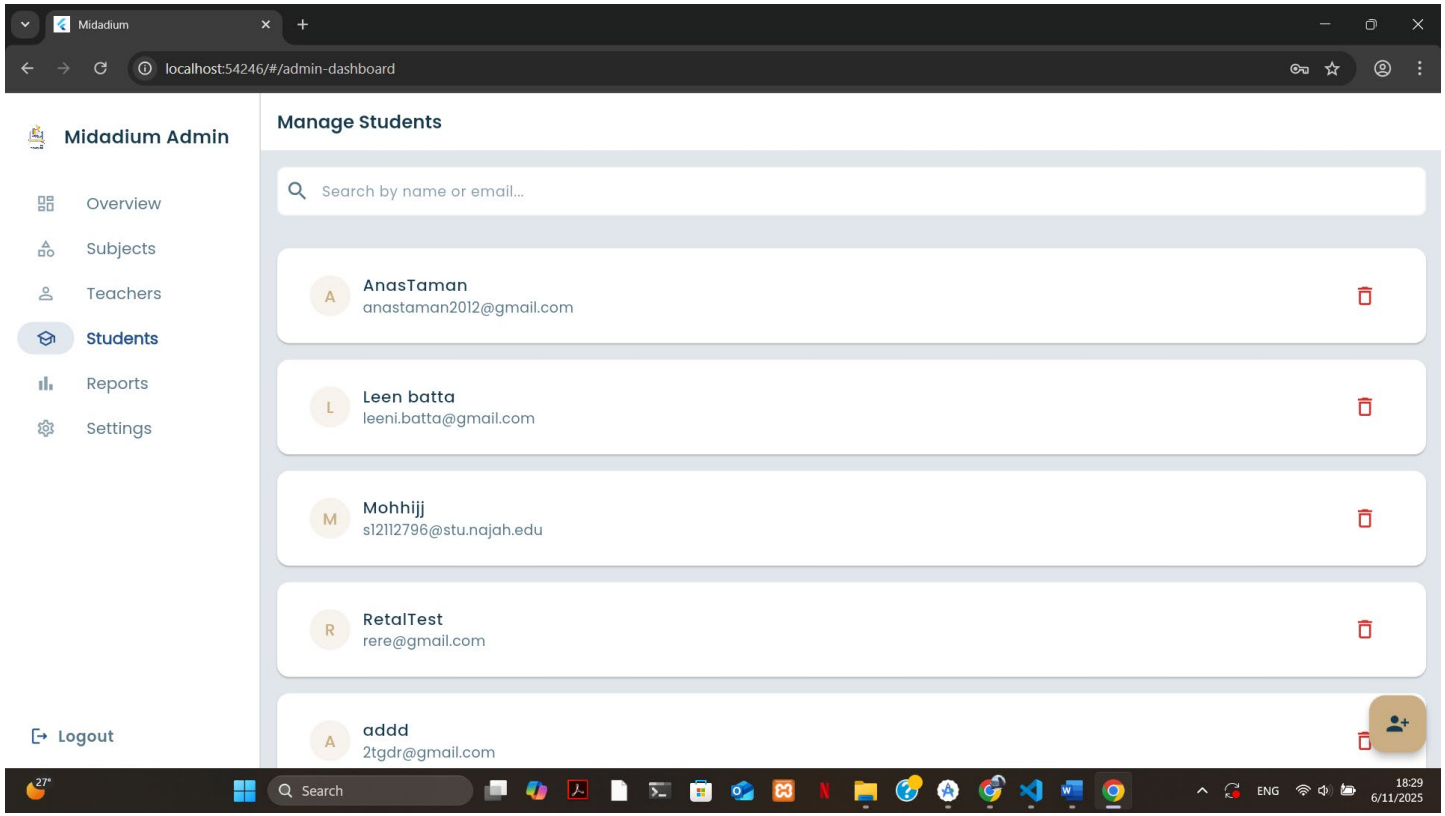


Figure 7 Manage teachers



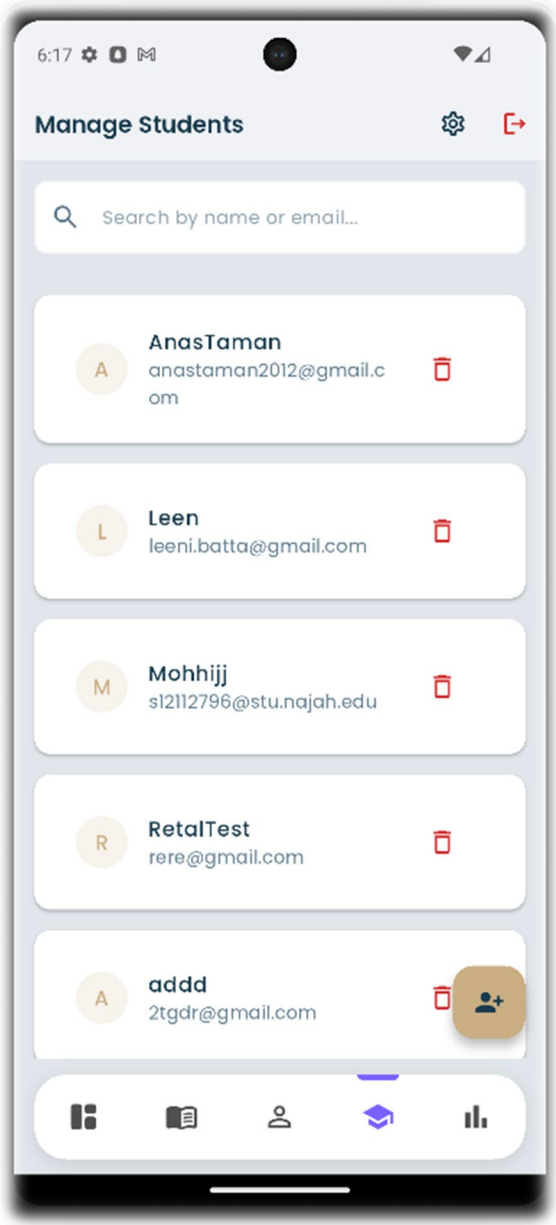


Figure 8Students management

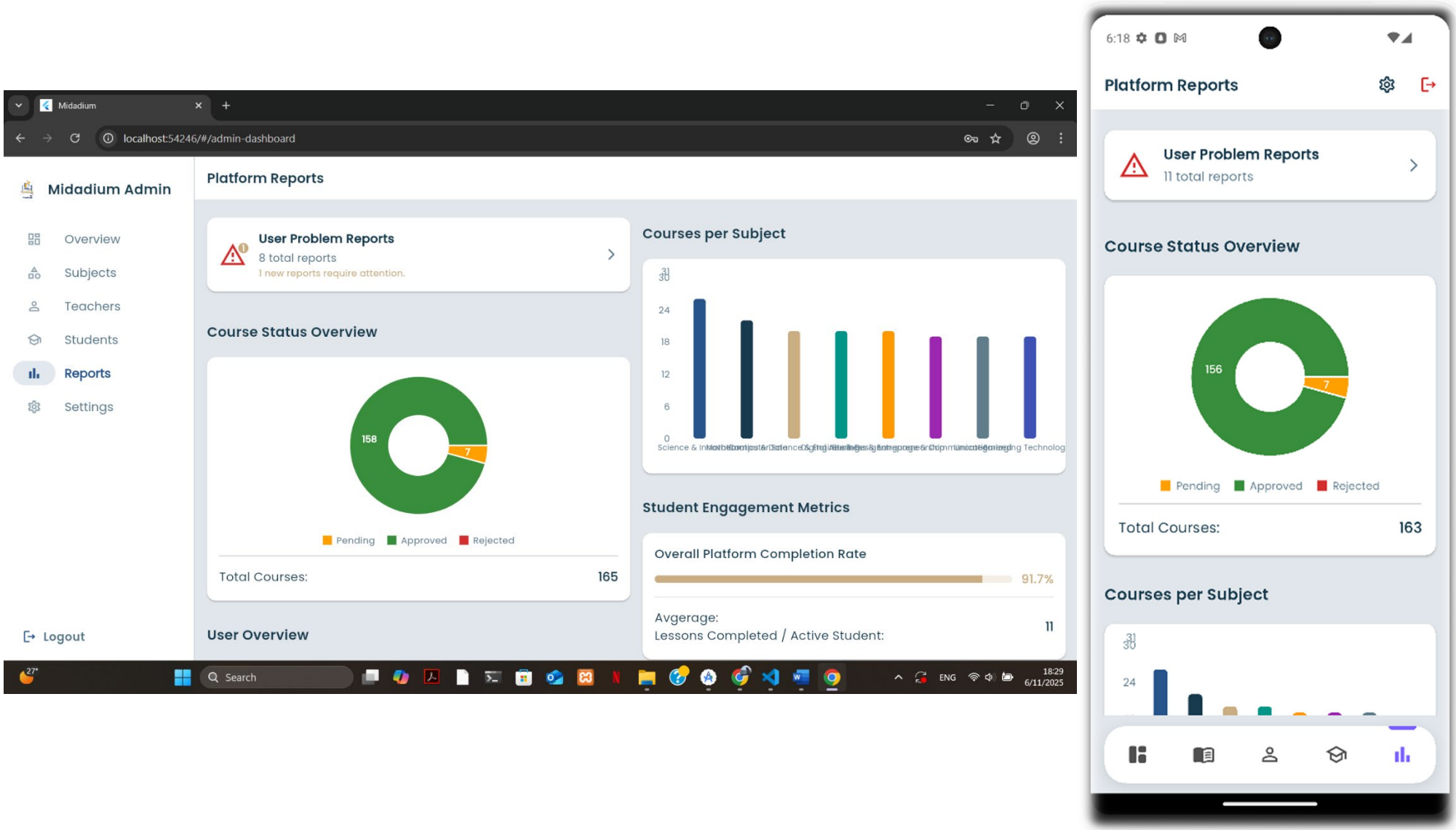


Figure 9 Platform Reports Screen

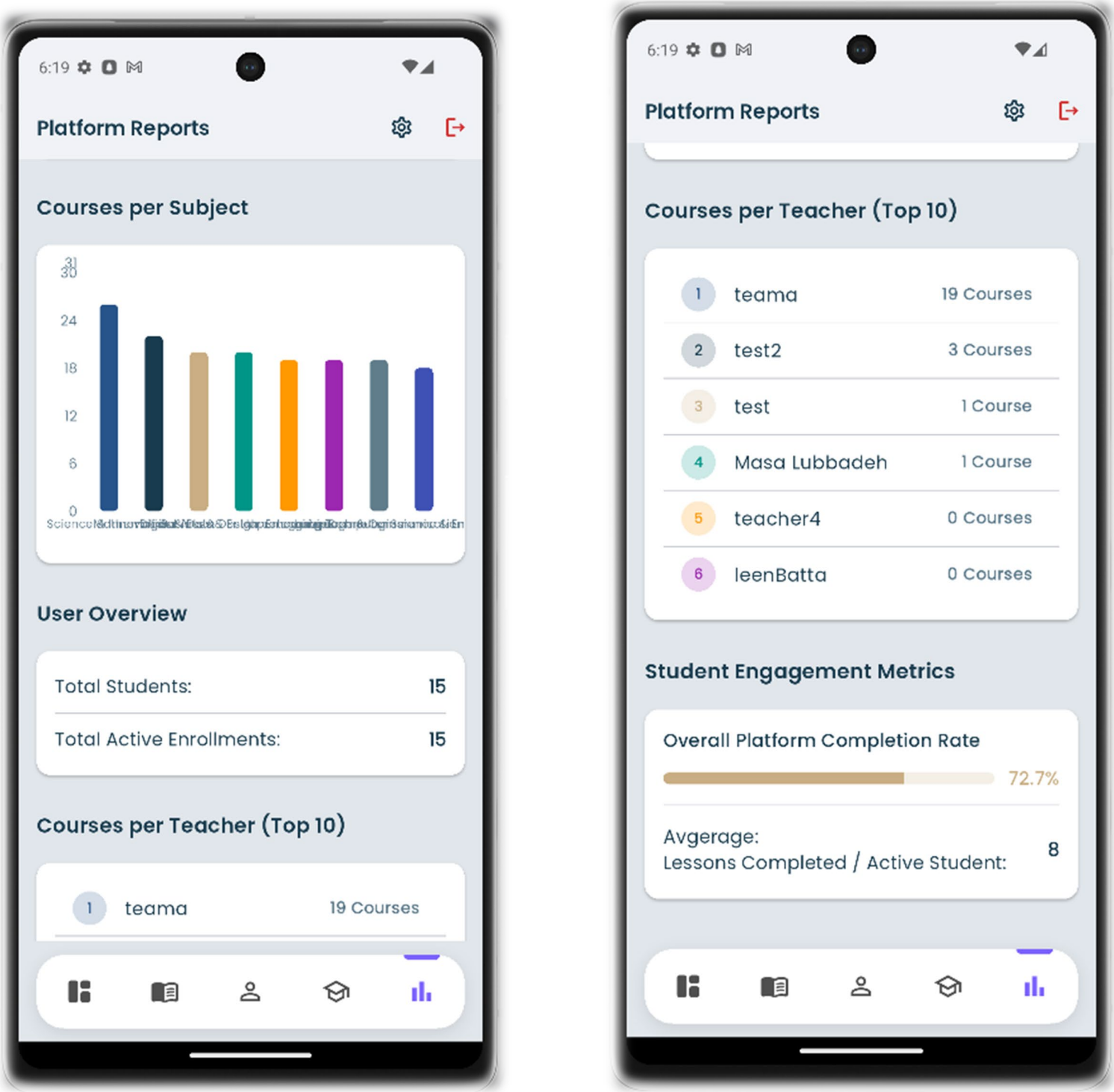


Figure 10 Platform Reports

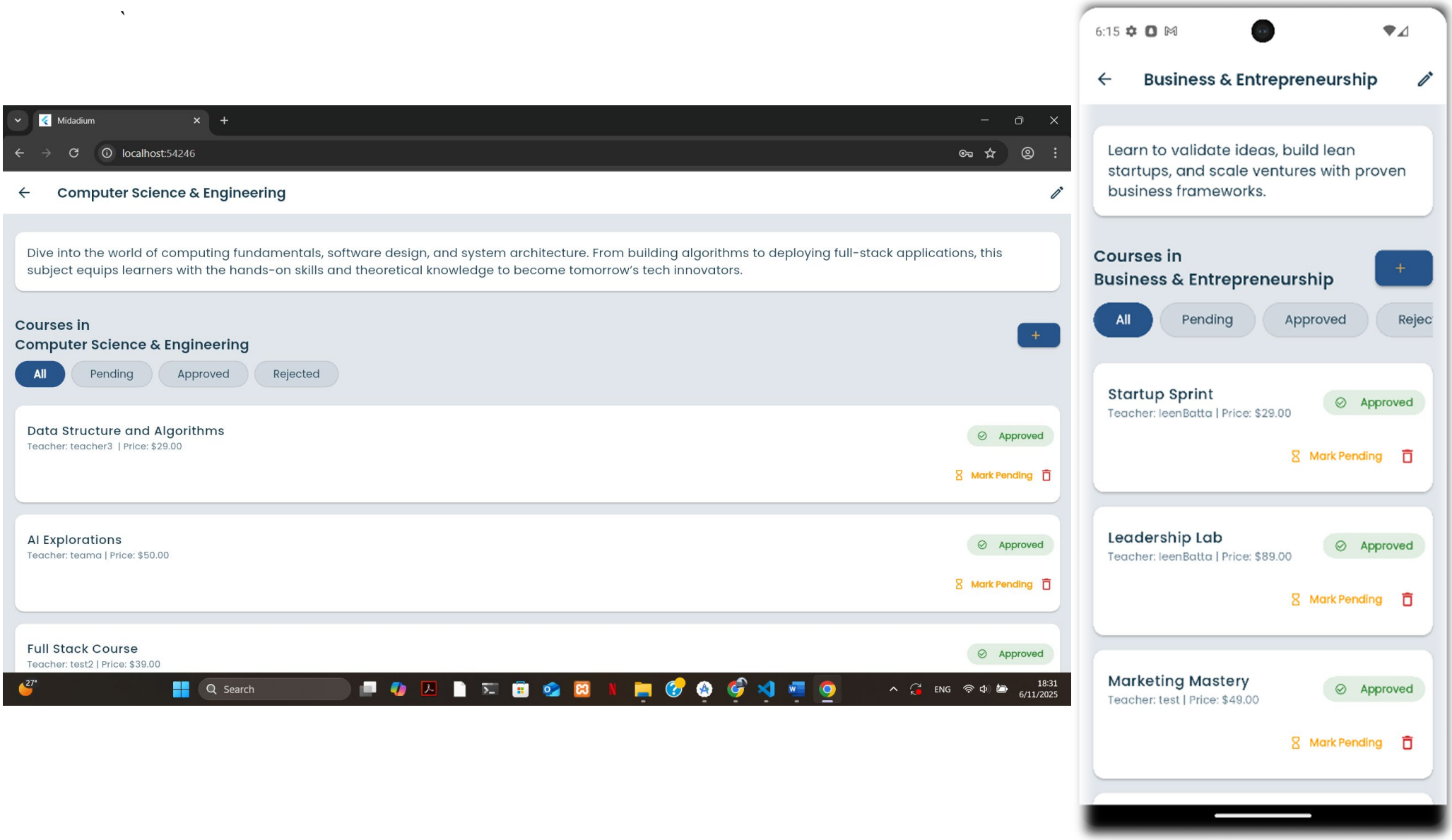
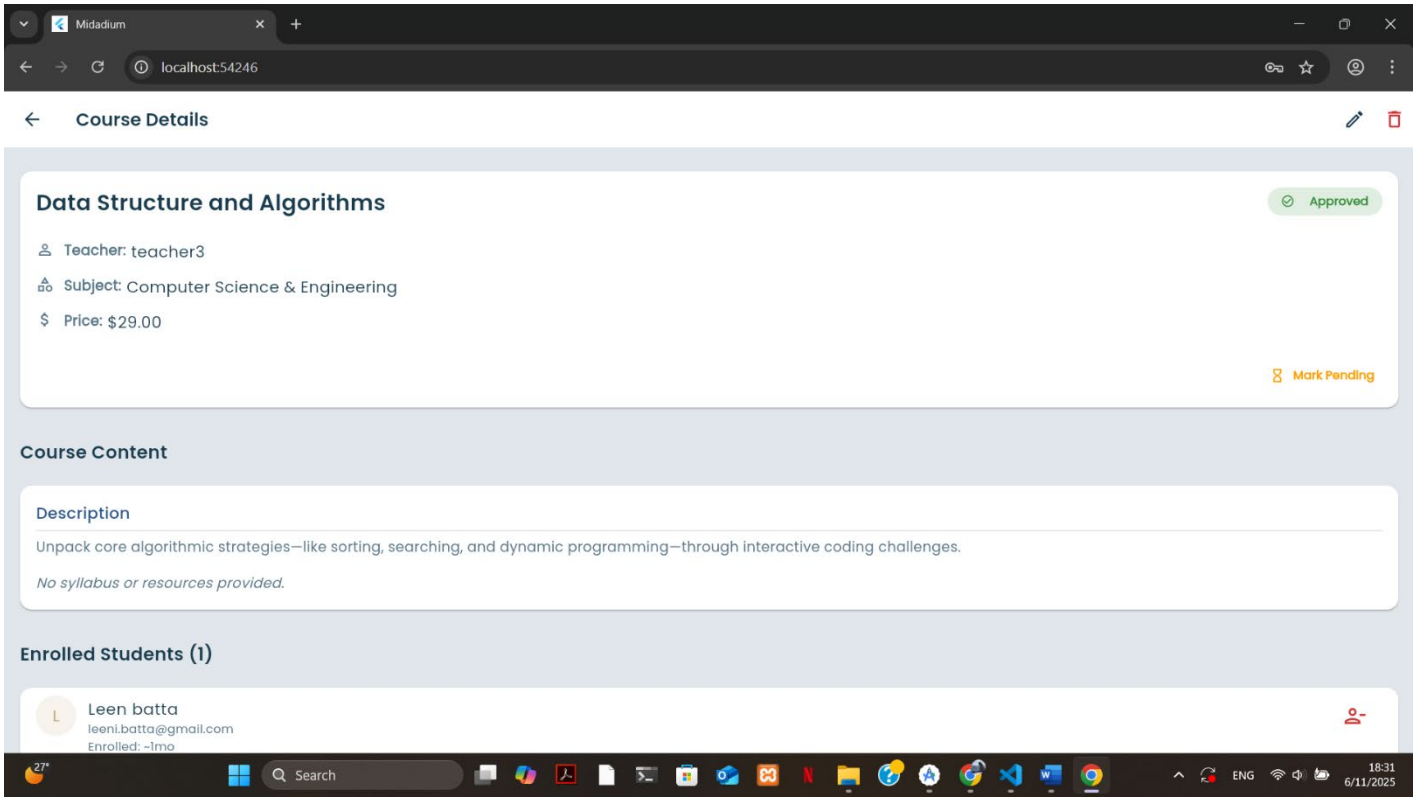


Figure 11 Subject Details screen



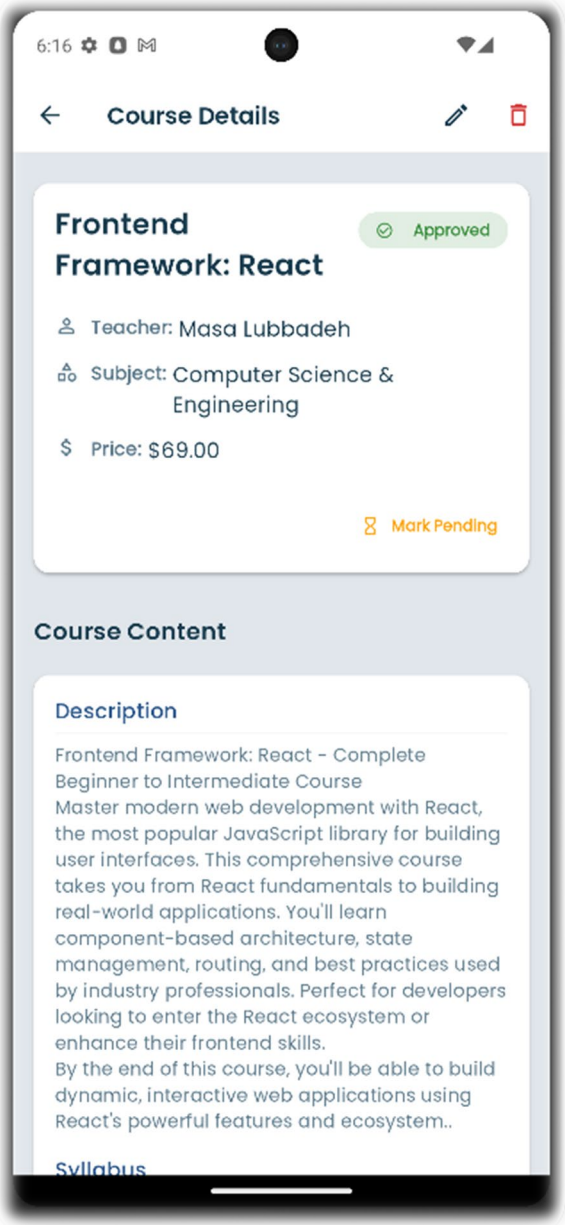


Figure 12 Course details Screen

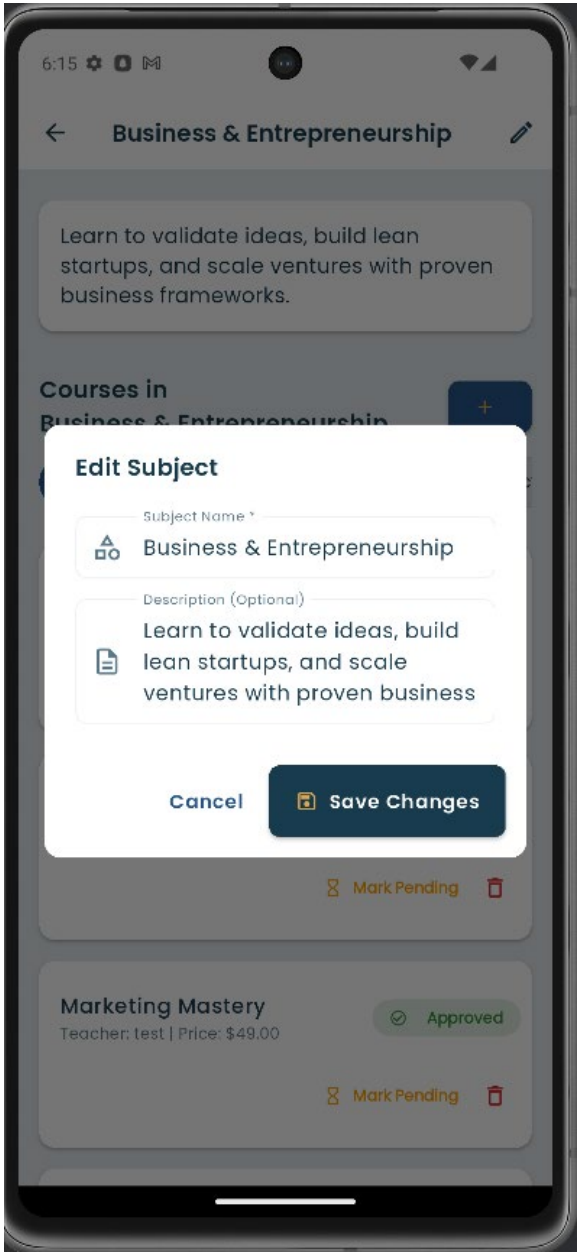


Figure 15 edit Subject

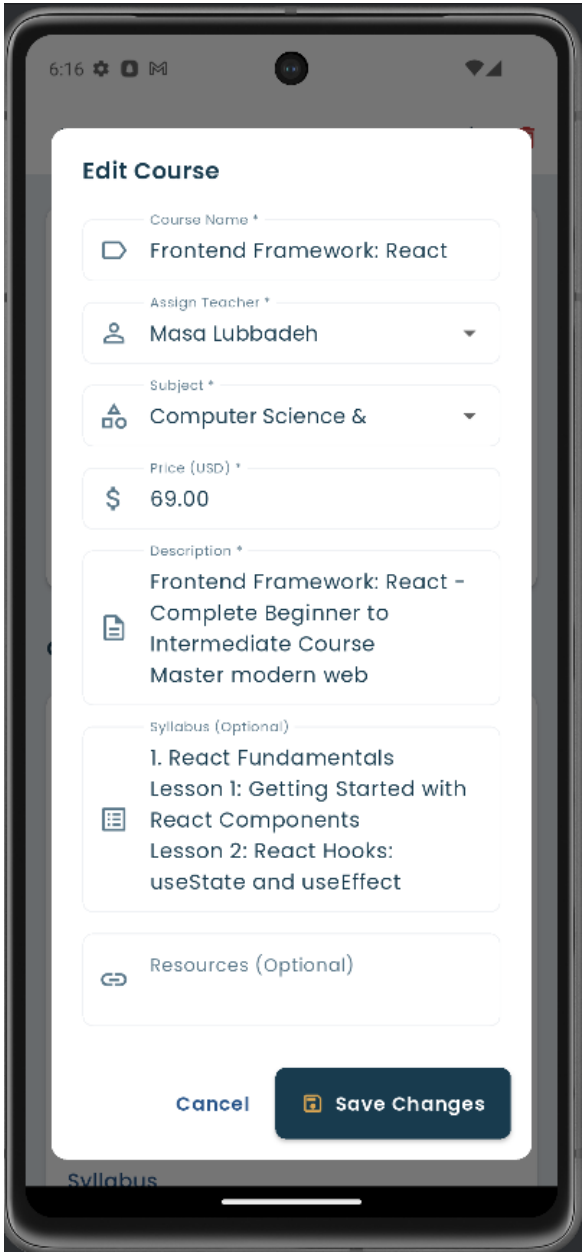


Figure 13 edit course

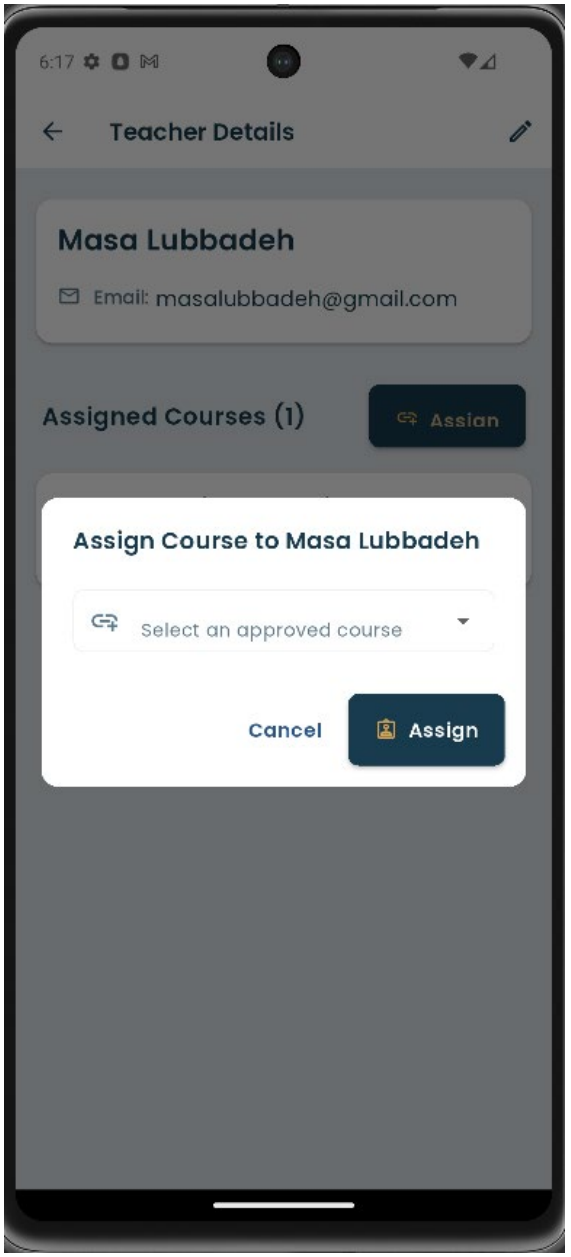
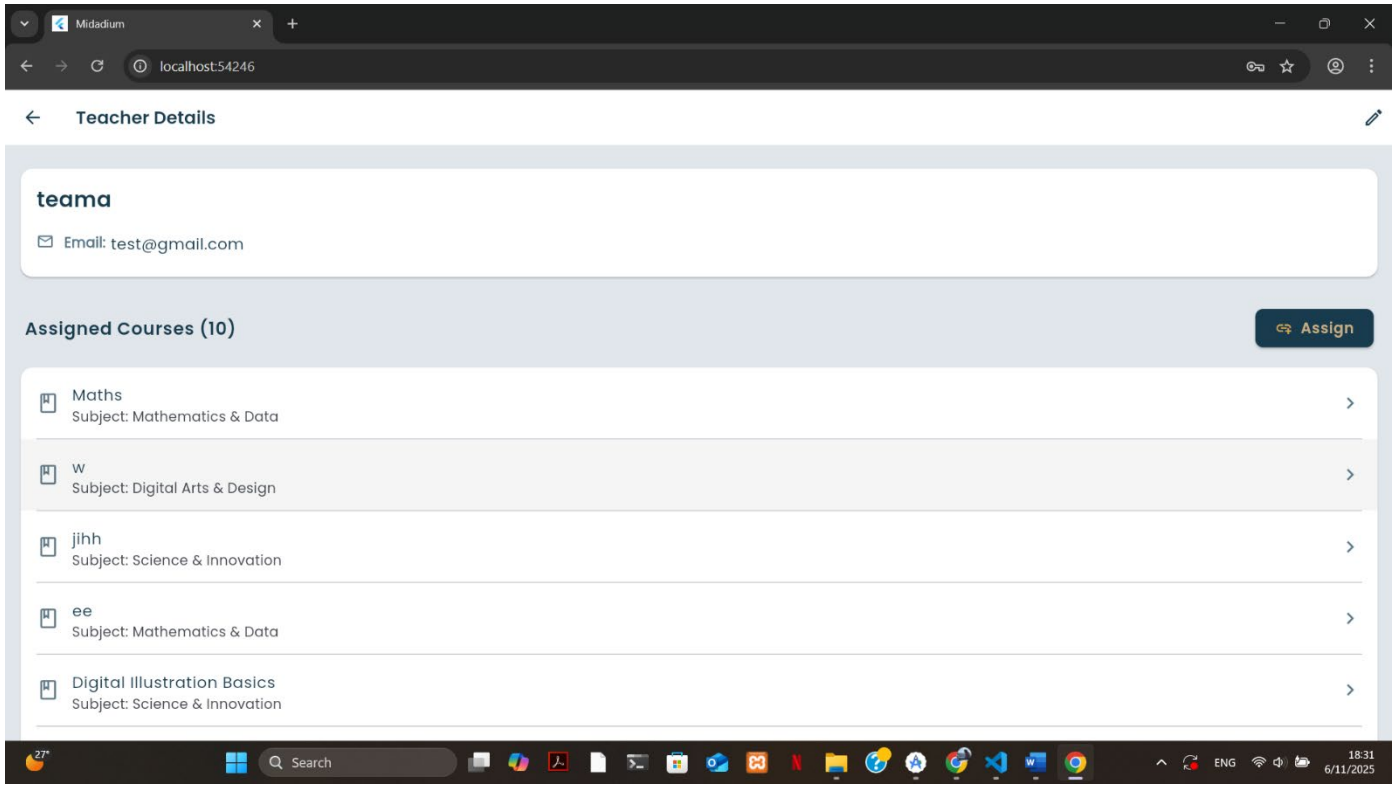


Figure 14 Assign Course to Teacher



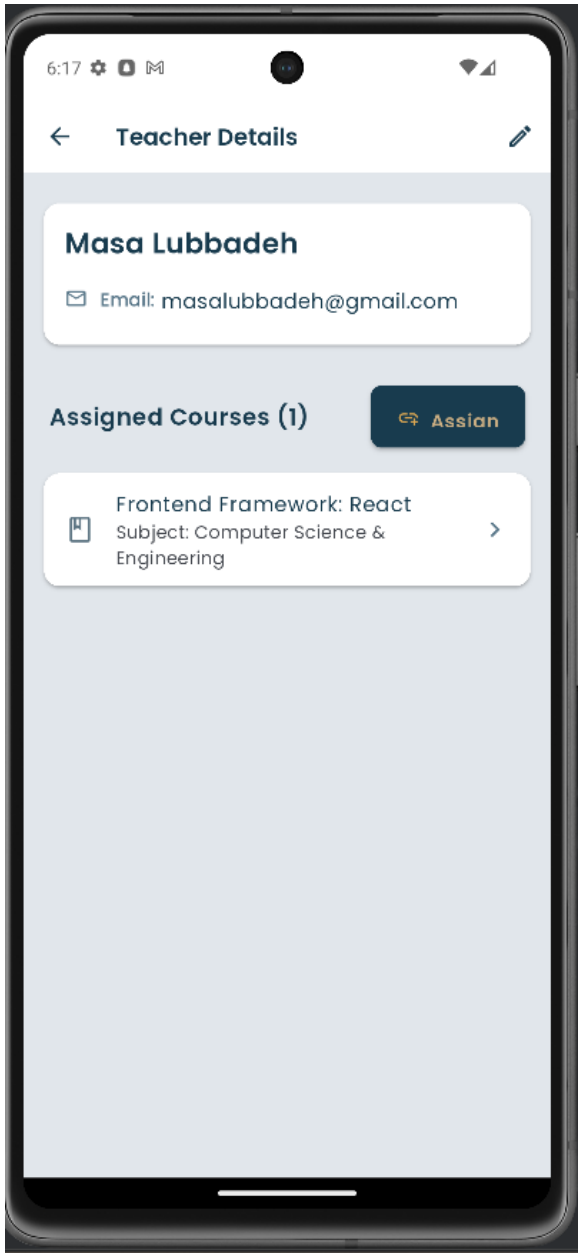
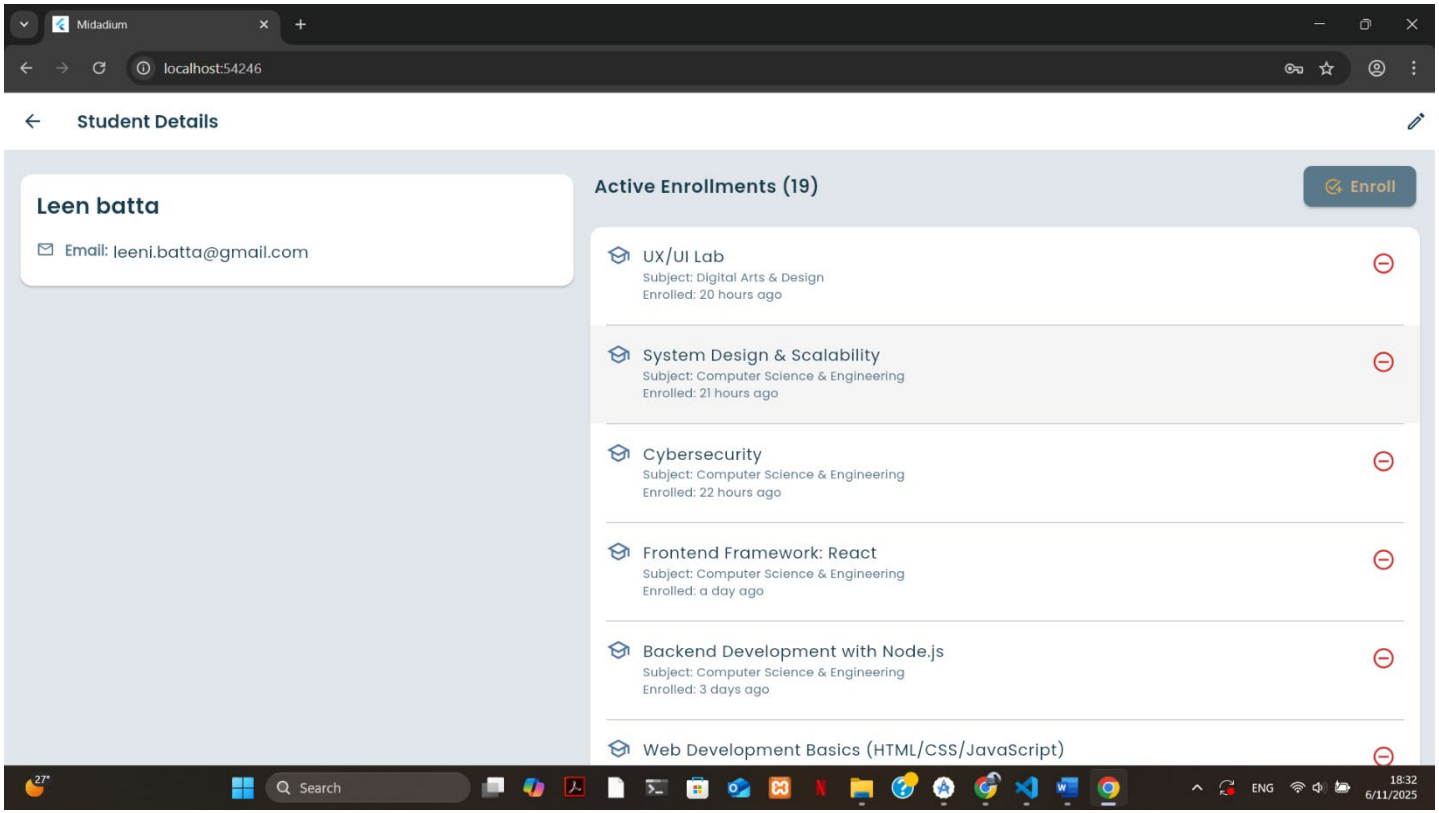


Figure 16 Teacher Detail Screen



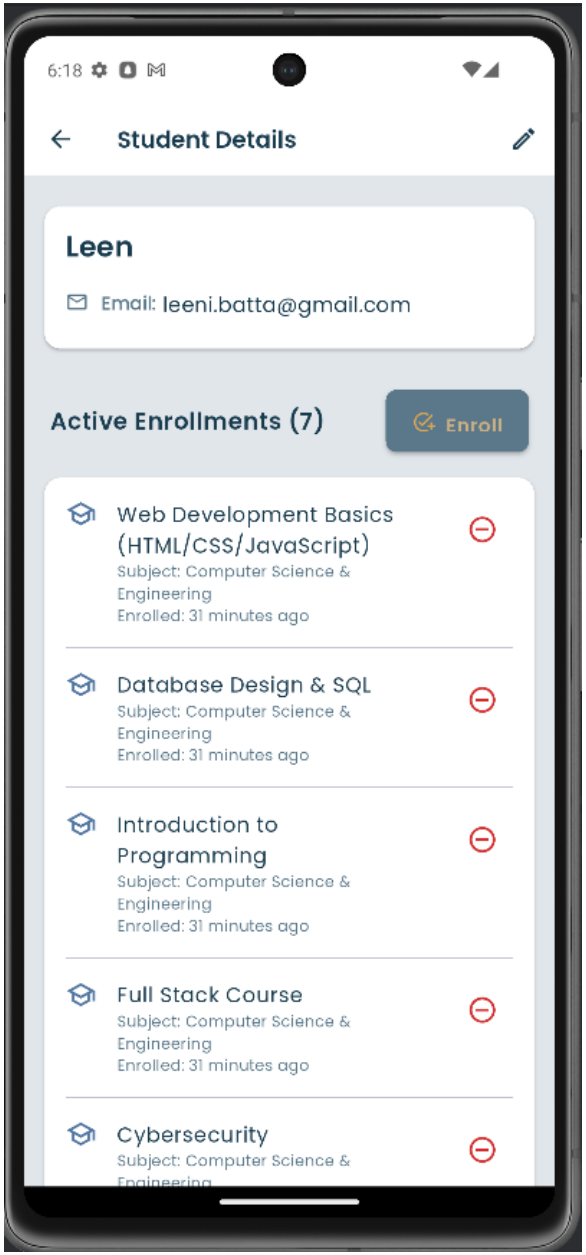
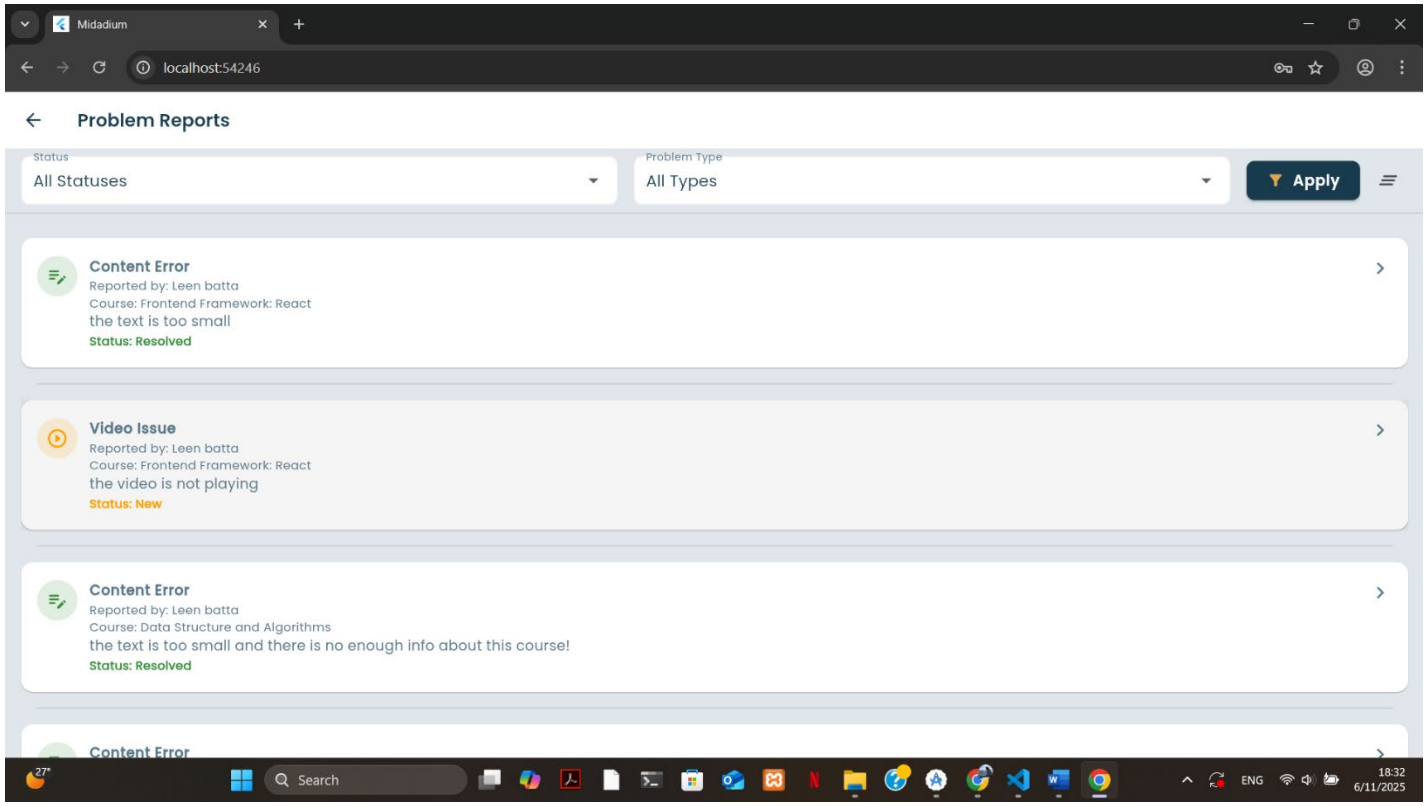


Figure 17 Student details and management



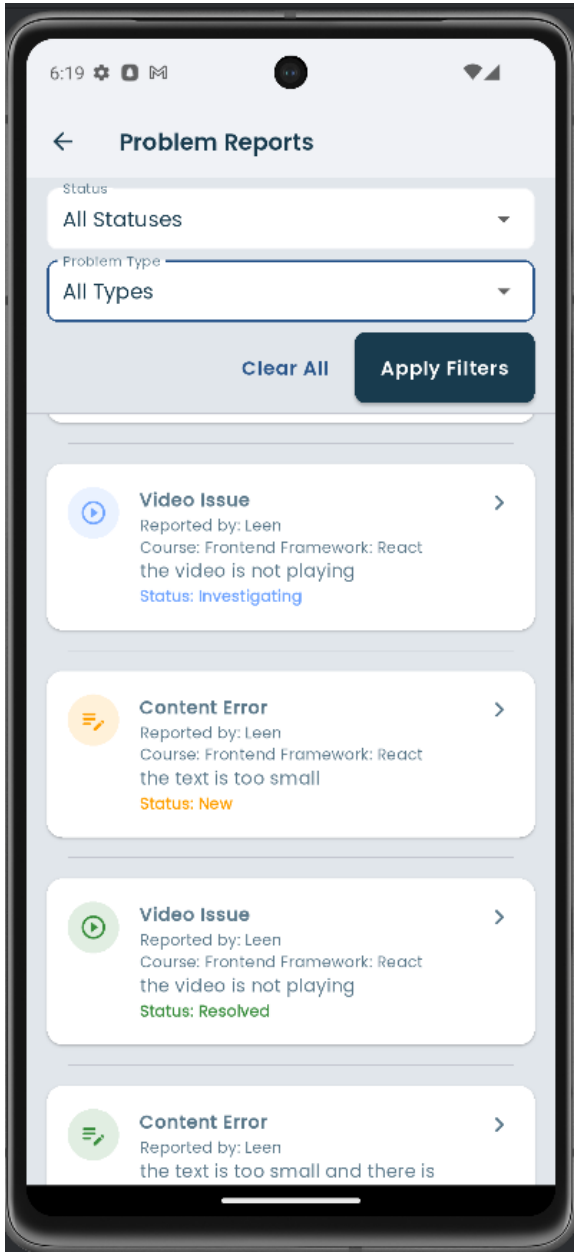


Figure 18 Problem Report Screen

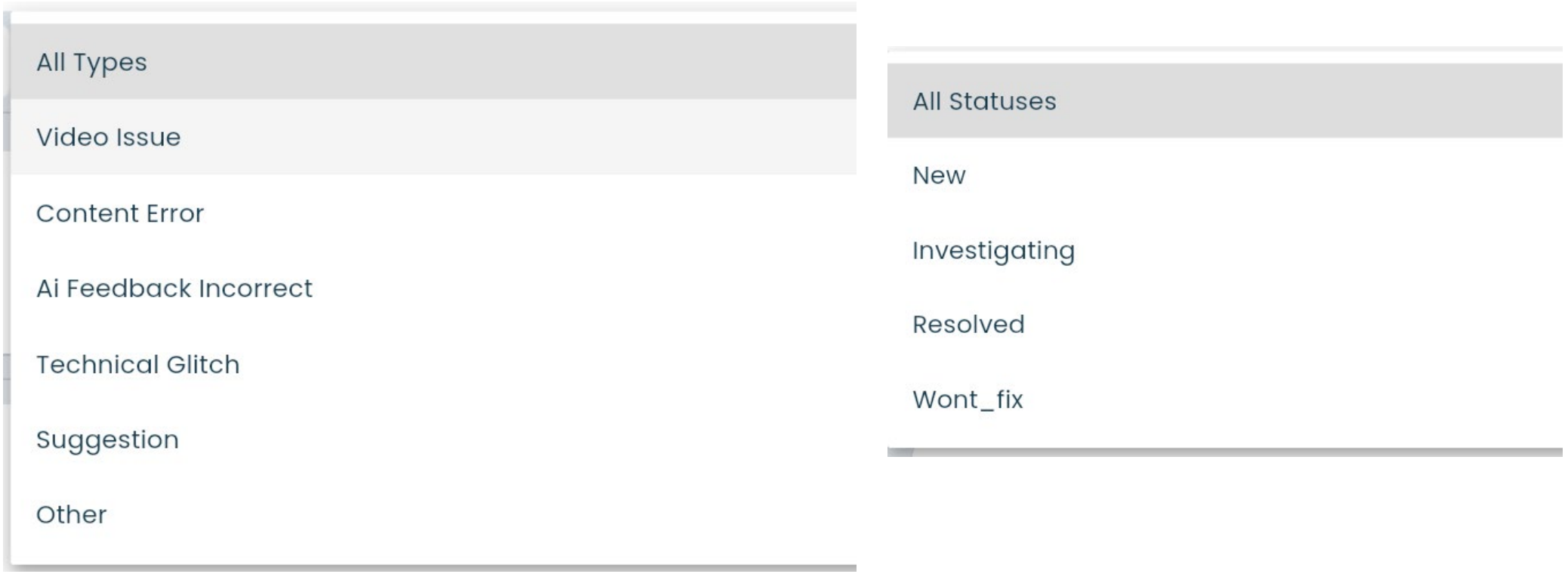


Figure 19 Problem reports Filtering

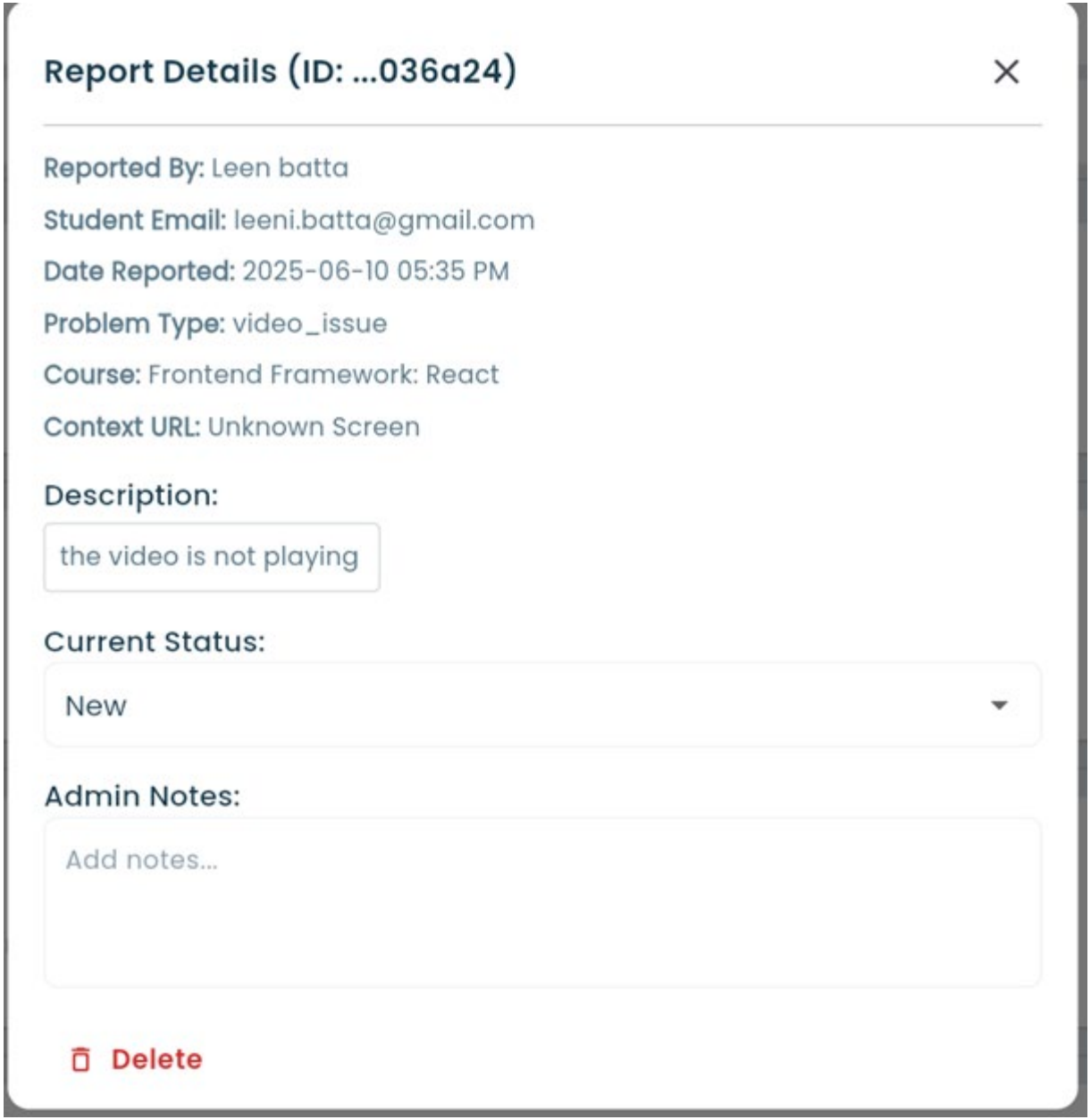
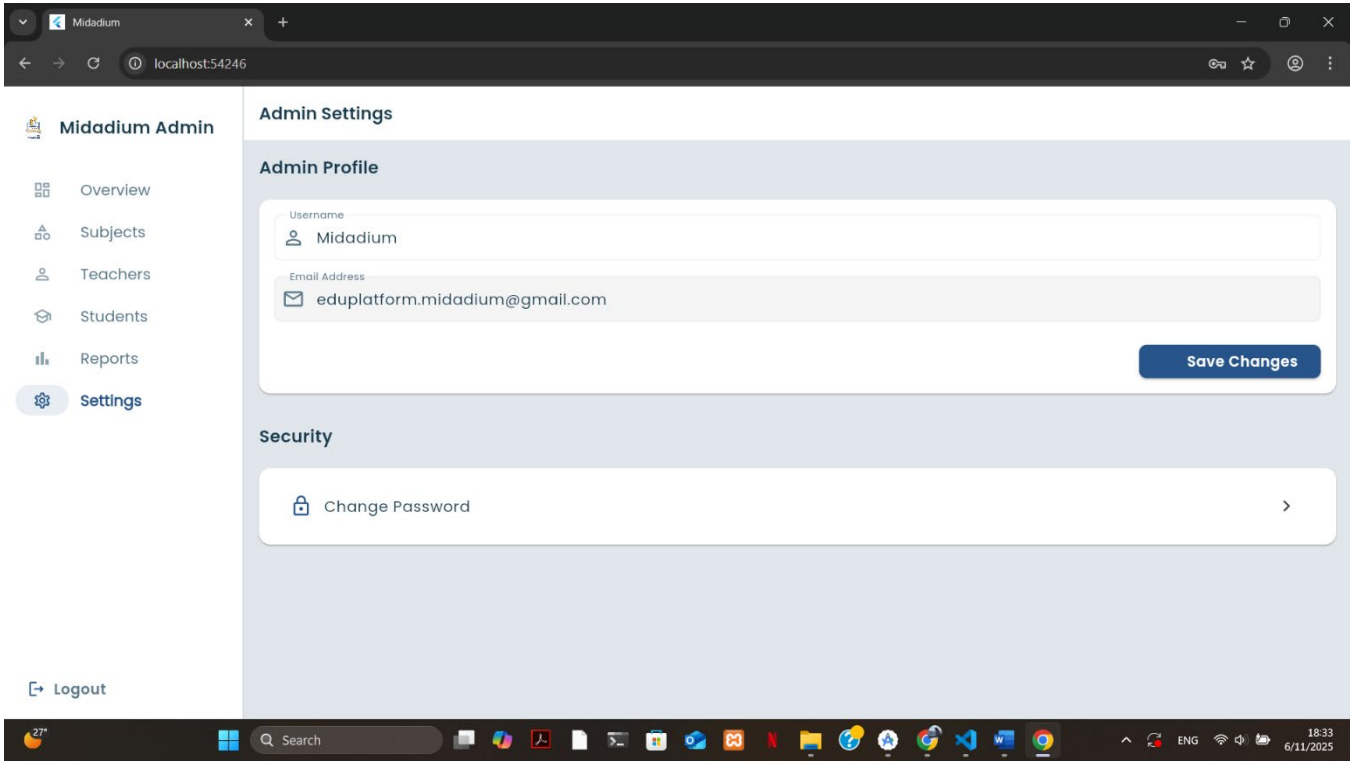
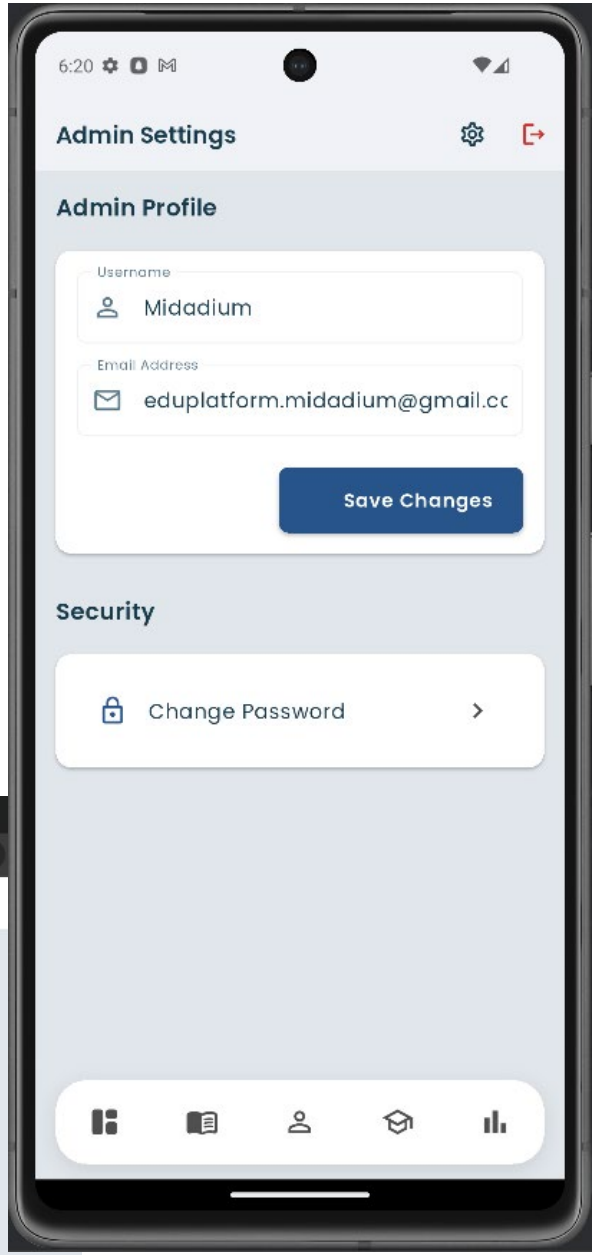


Figure 20 Report Details





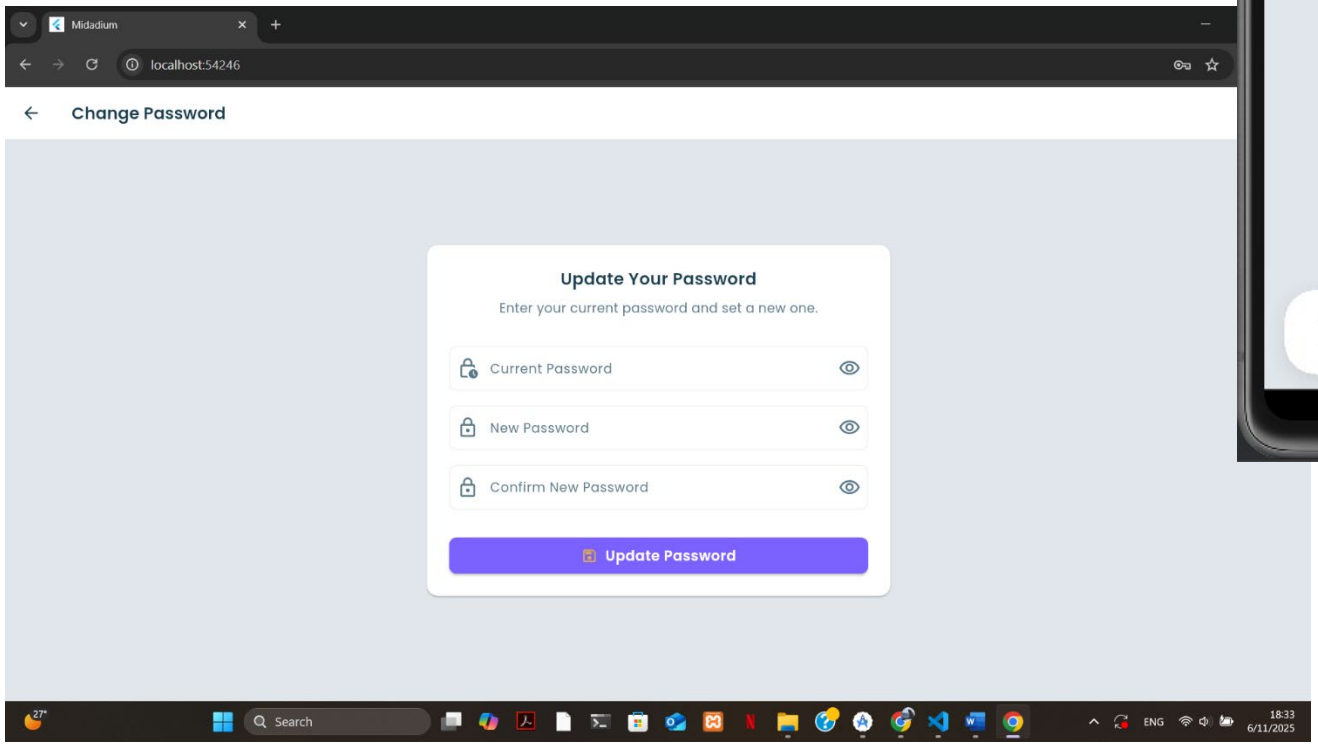


Figure 21 Admin Settings

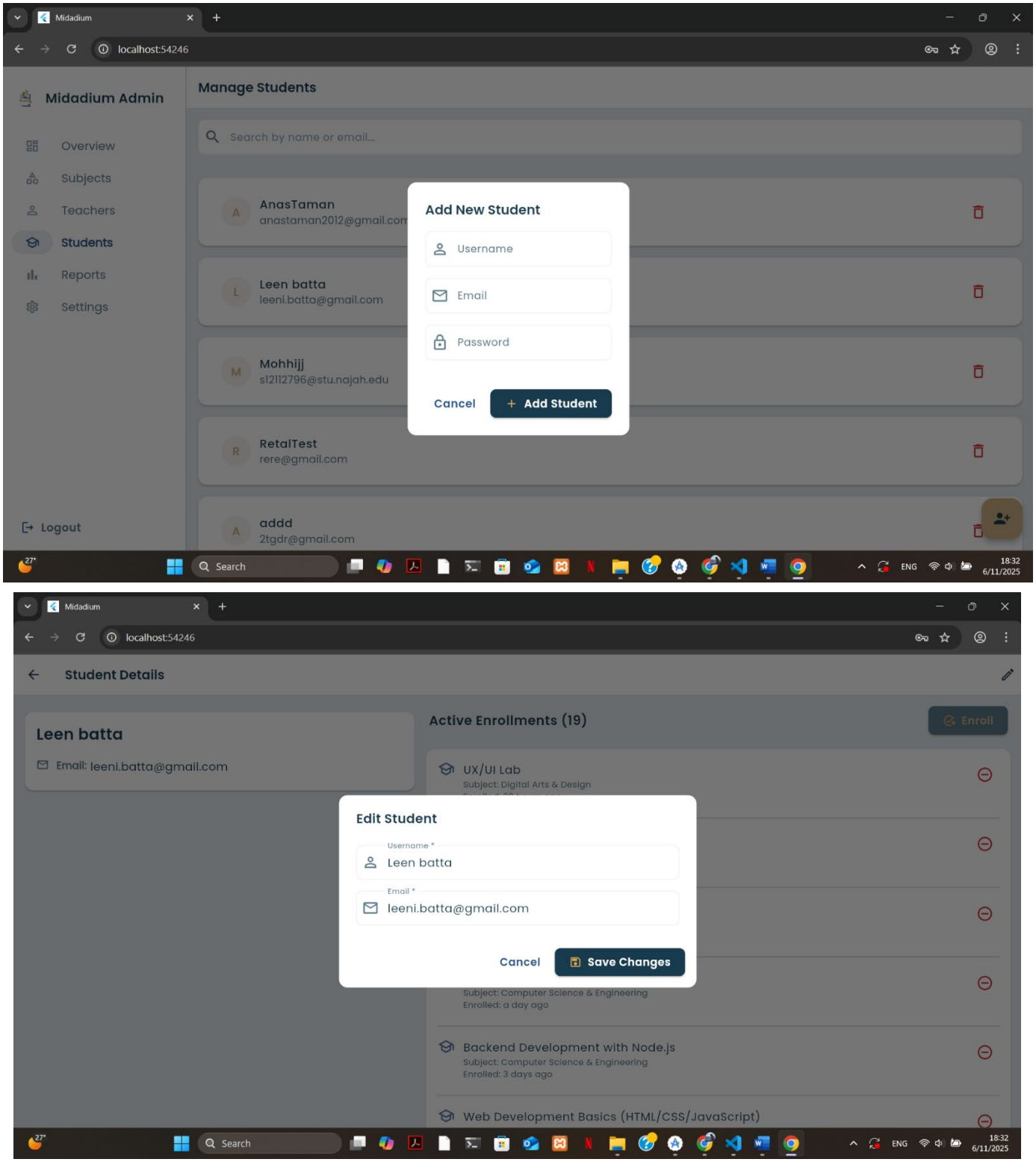


Figure 22 Adding and editing Students

Student Dashboard :

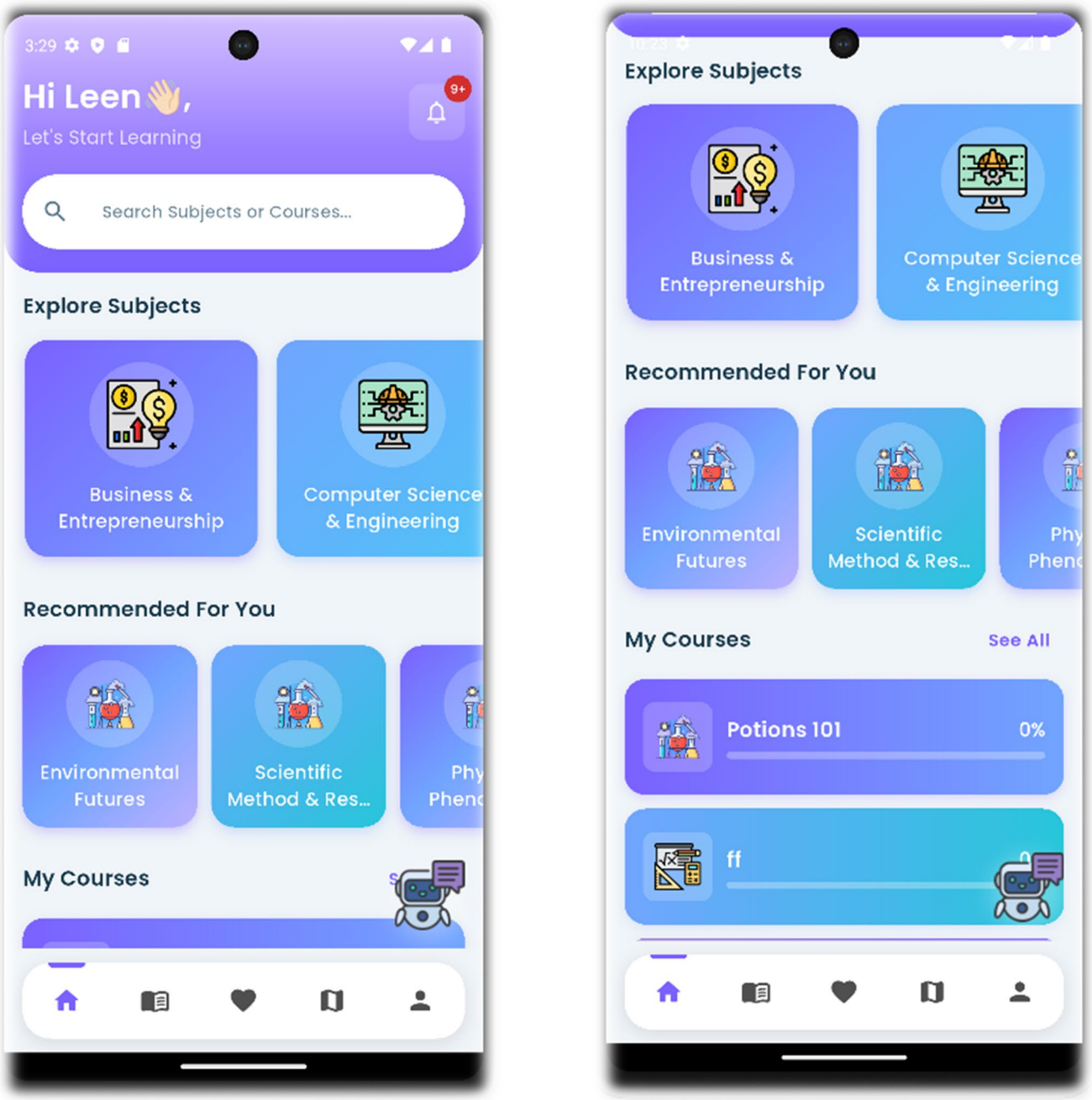


Figure 23 Student Home Screen

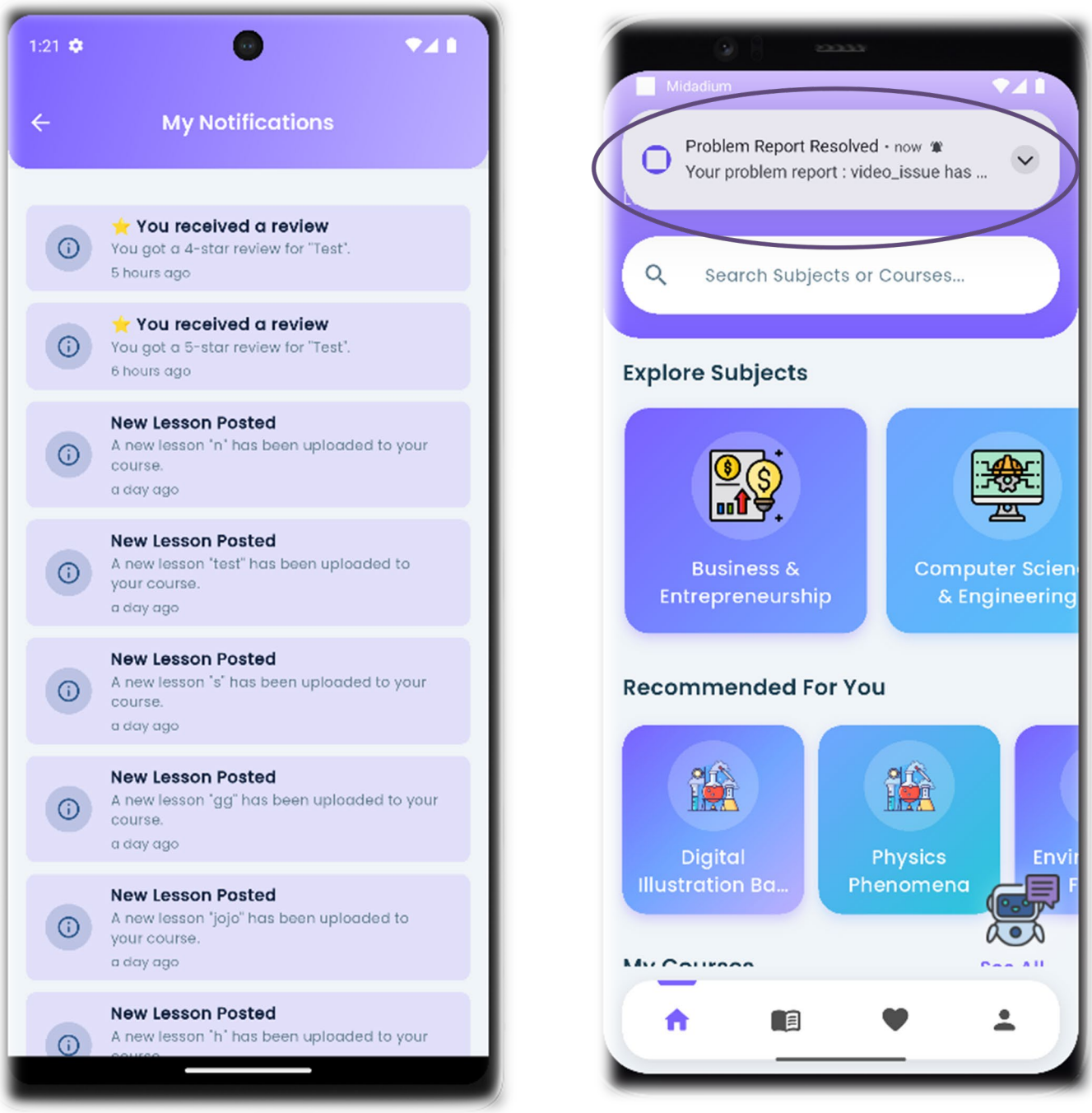


Figure 24 Student Notification Screen

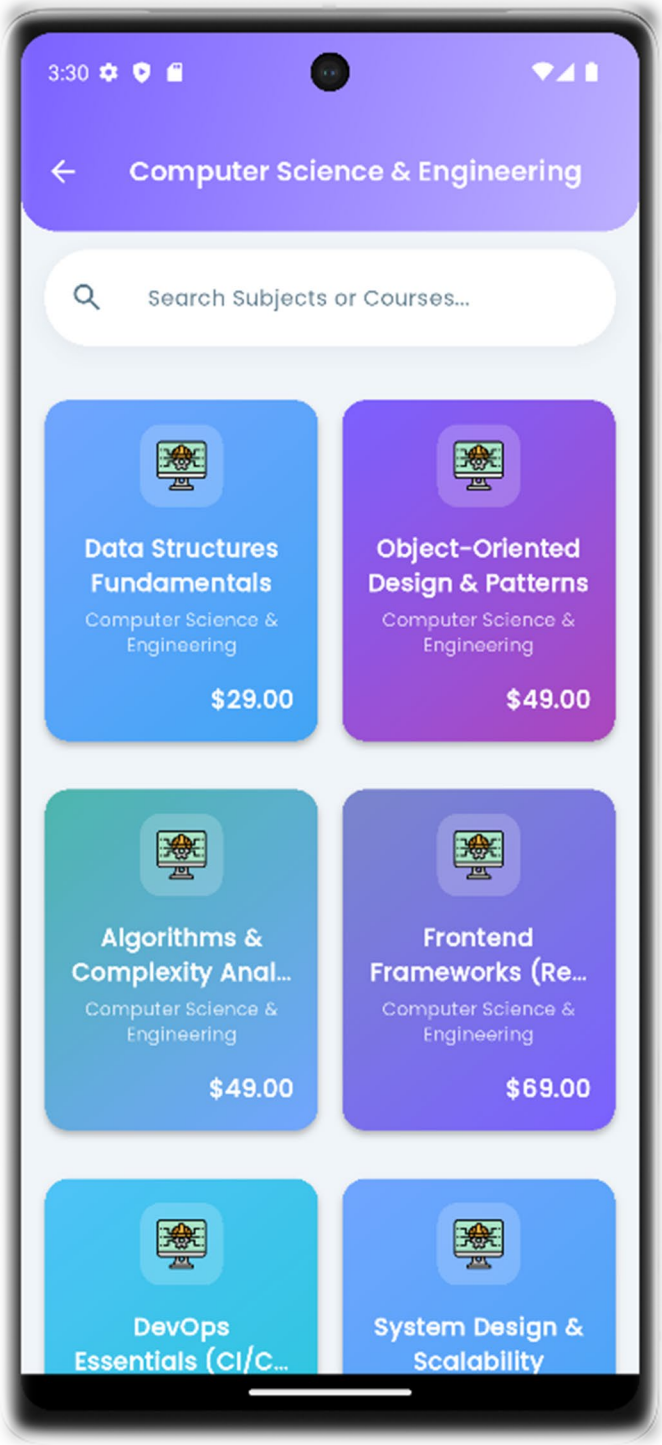


Figure 26 Subject's Available courses screen (Course catalog)

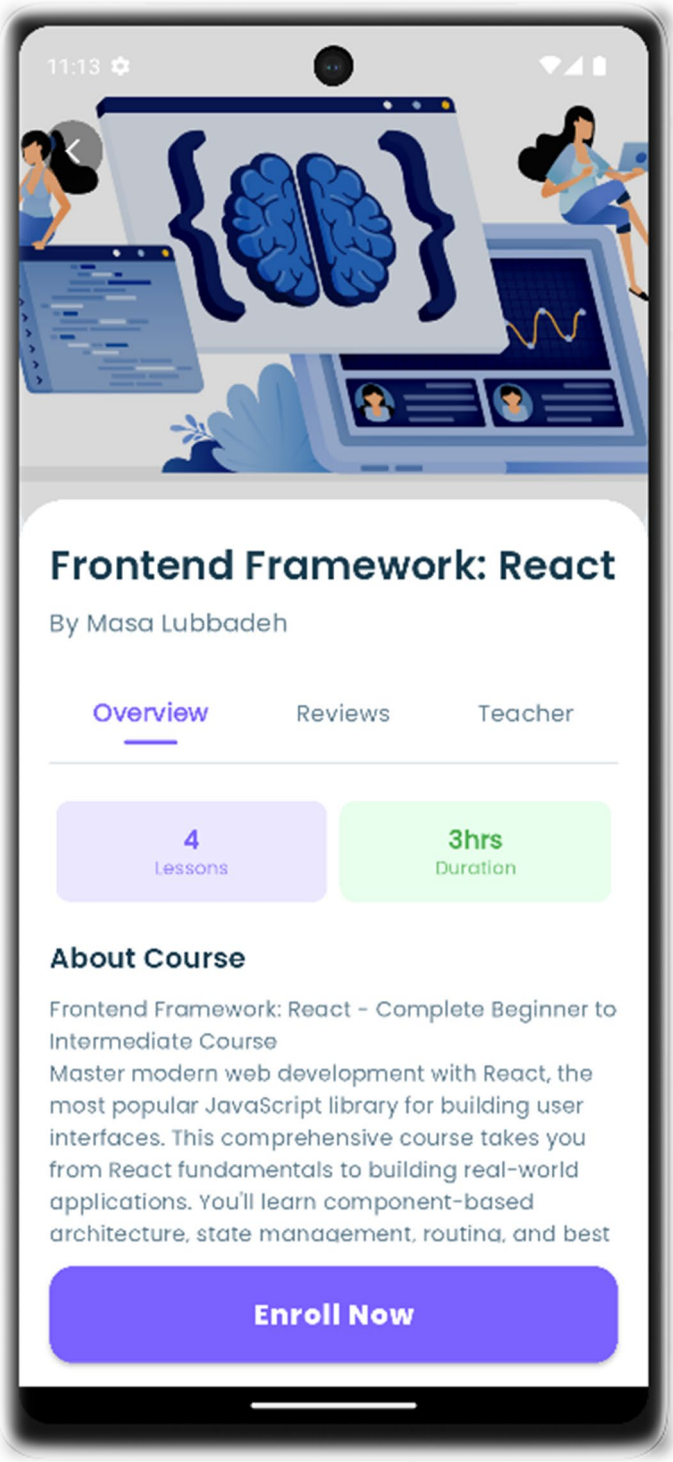


Figure 25 Course preview screen (Before Enrollment)

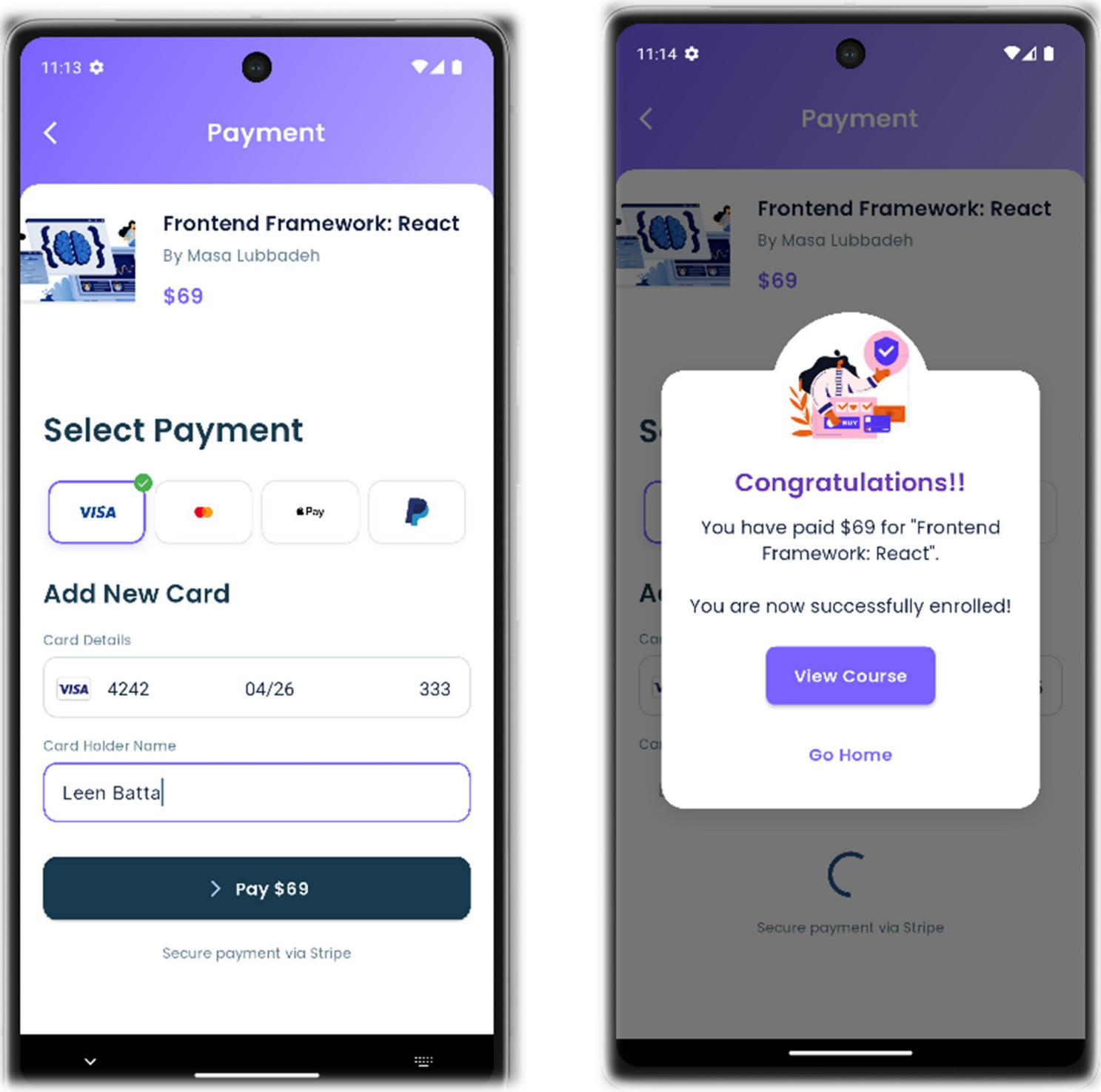
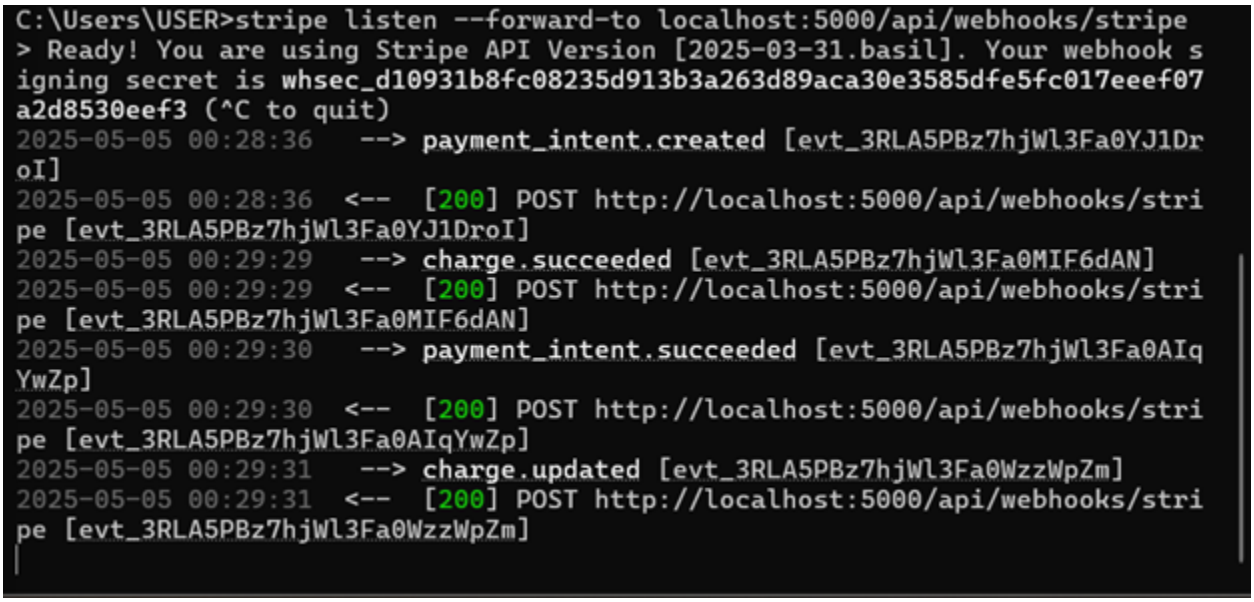


Figure 27 Course Payment Process



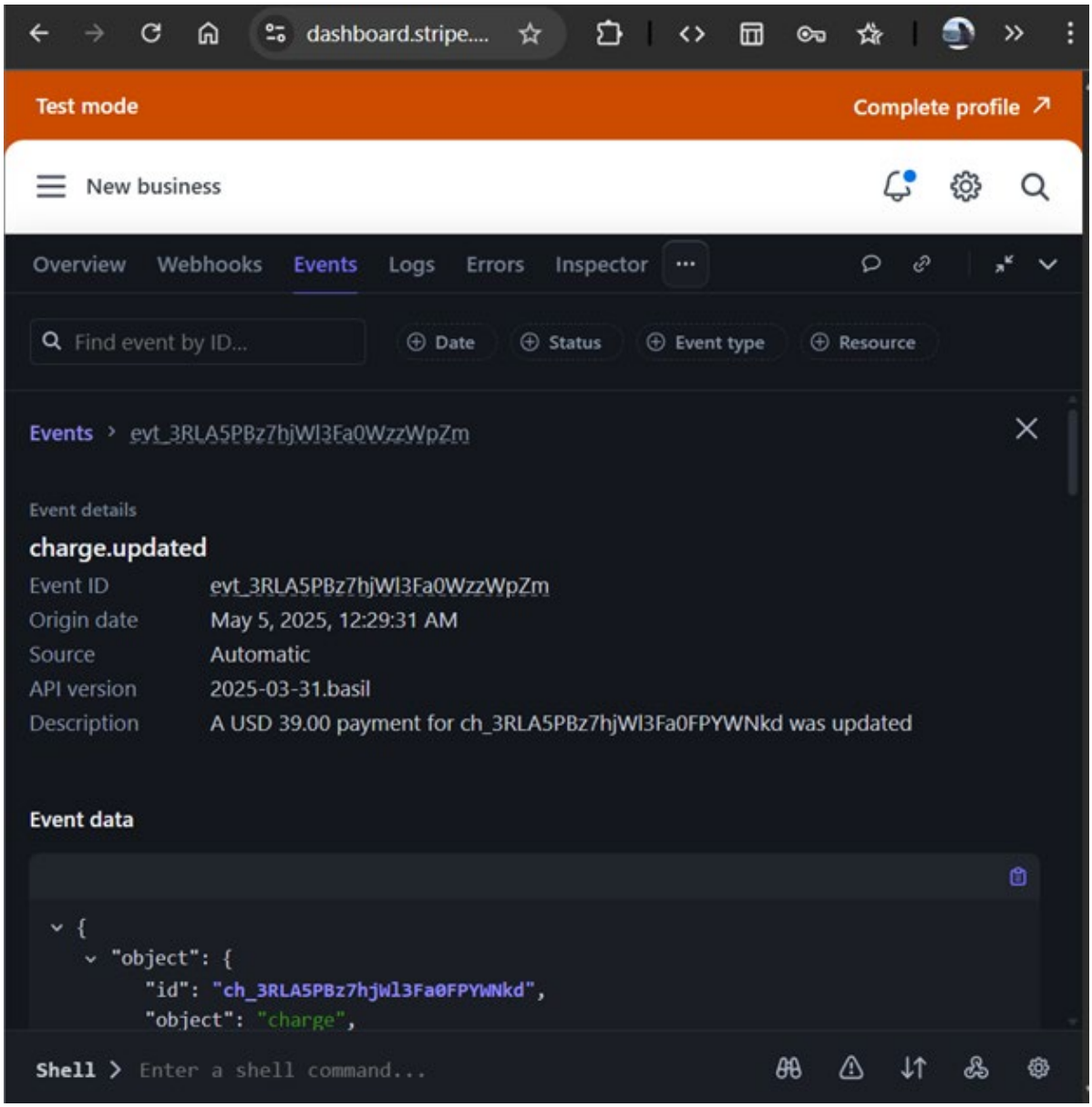


Figure 28 Payment success on stripe dashboard

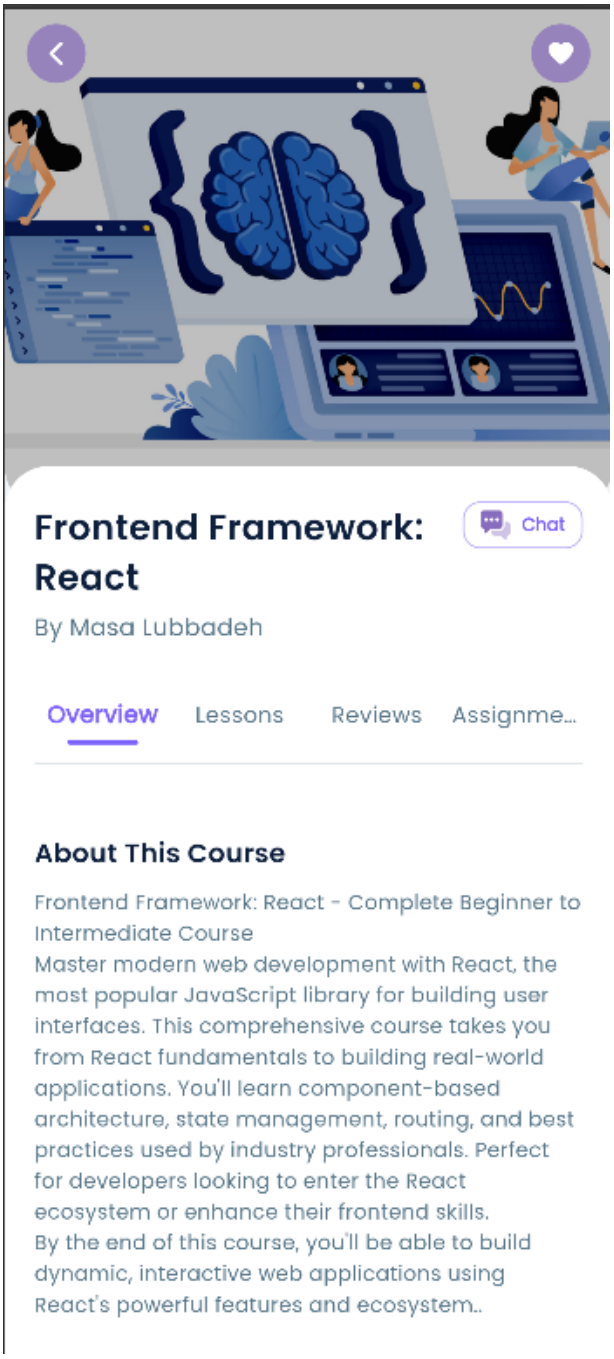


Figure 29 Course Details

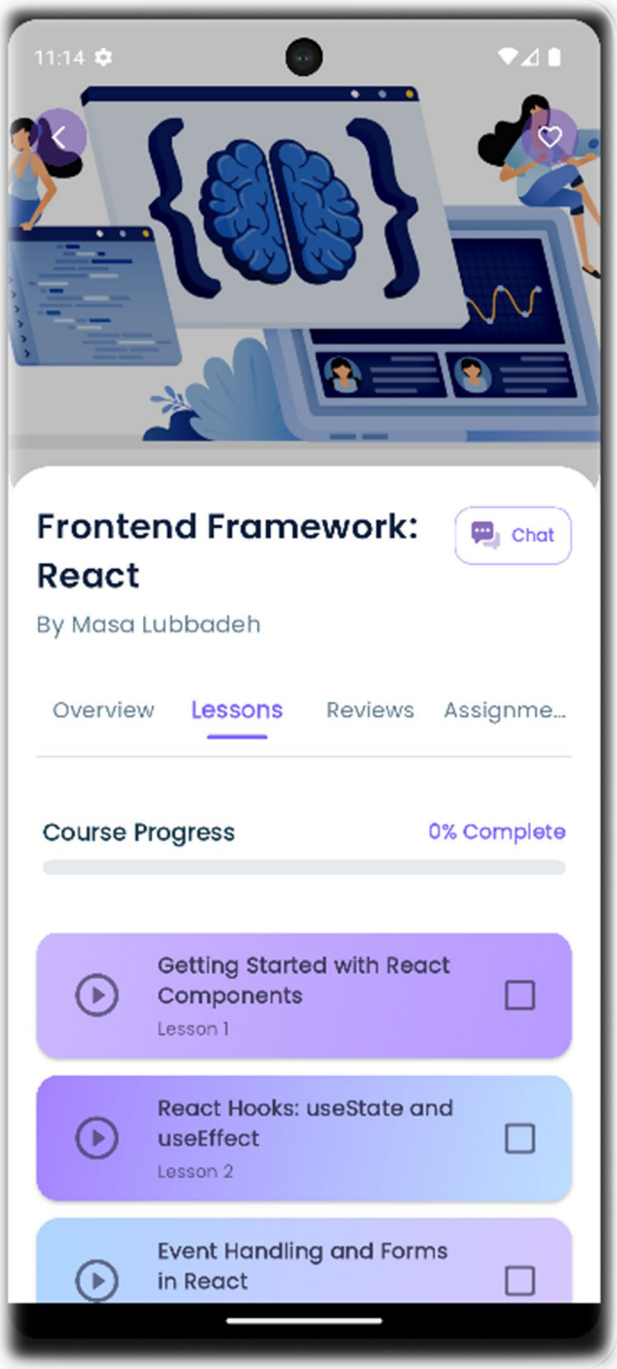


Figure 30 Lessons tab

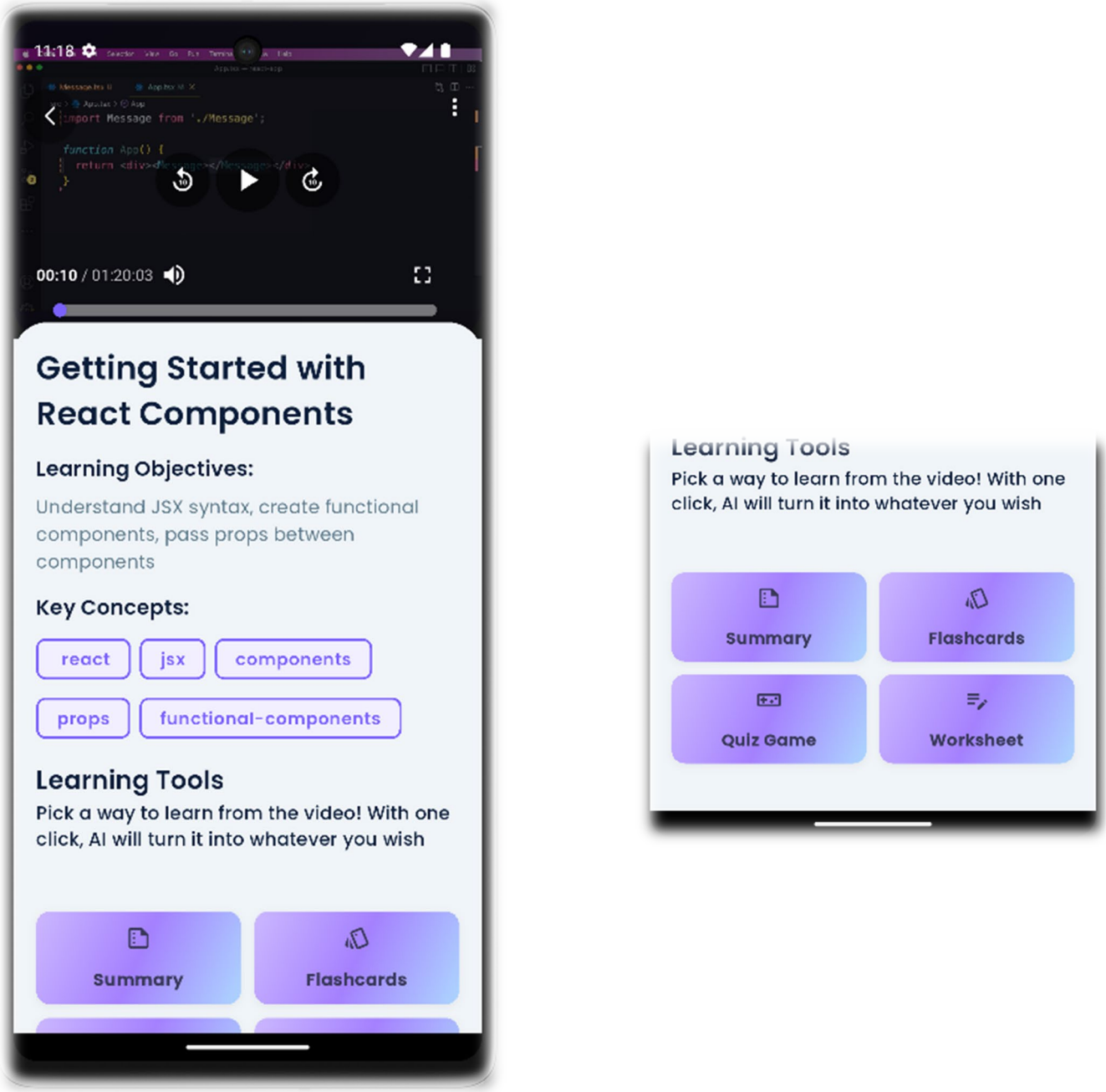


Figure 31 Lesson Screen And the available Ai tools for generation

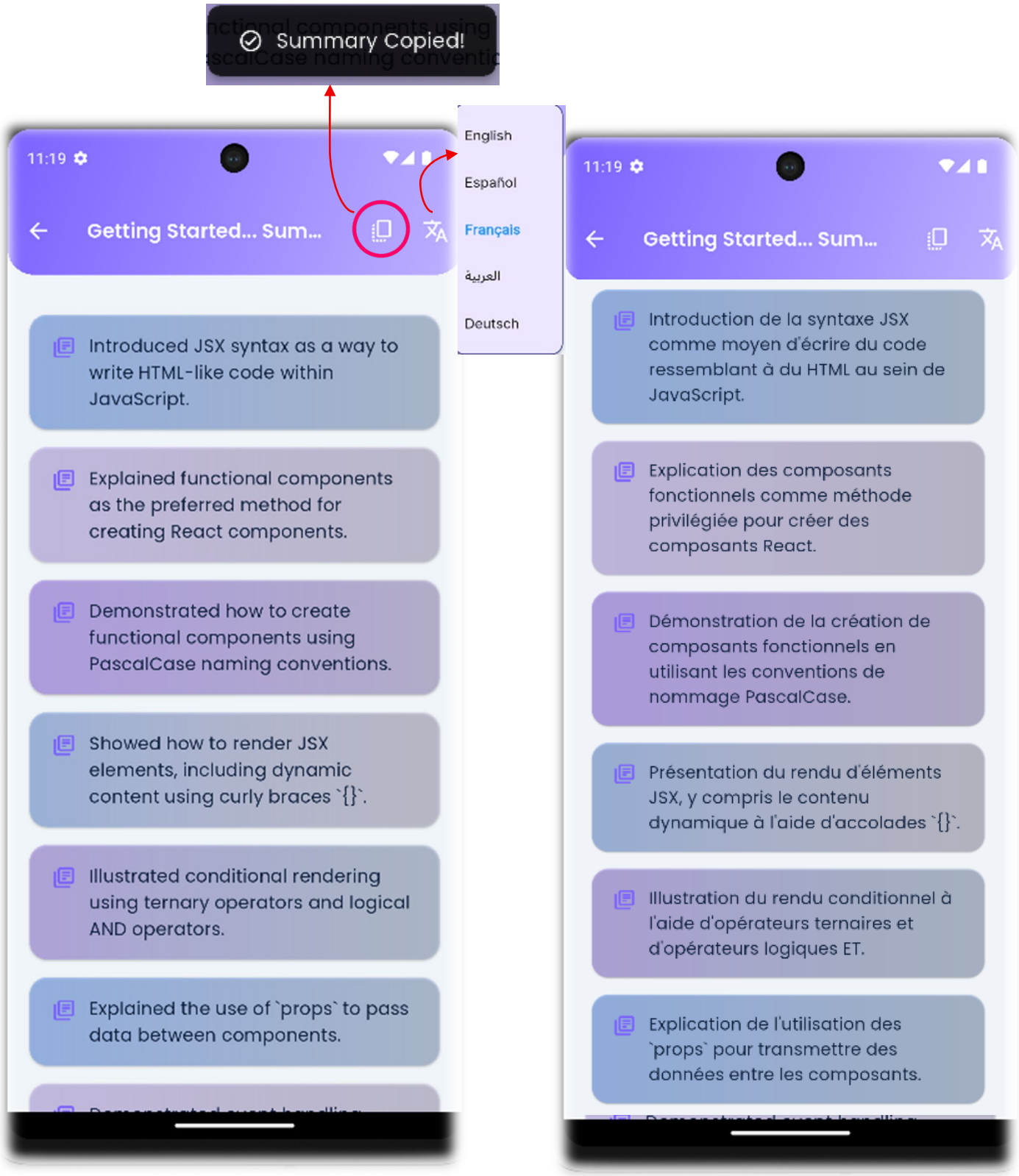


Figure 32 Summary AI format with options to copy and AI translation

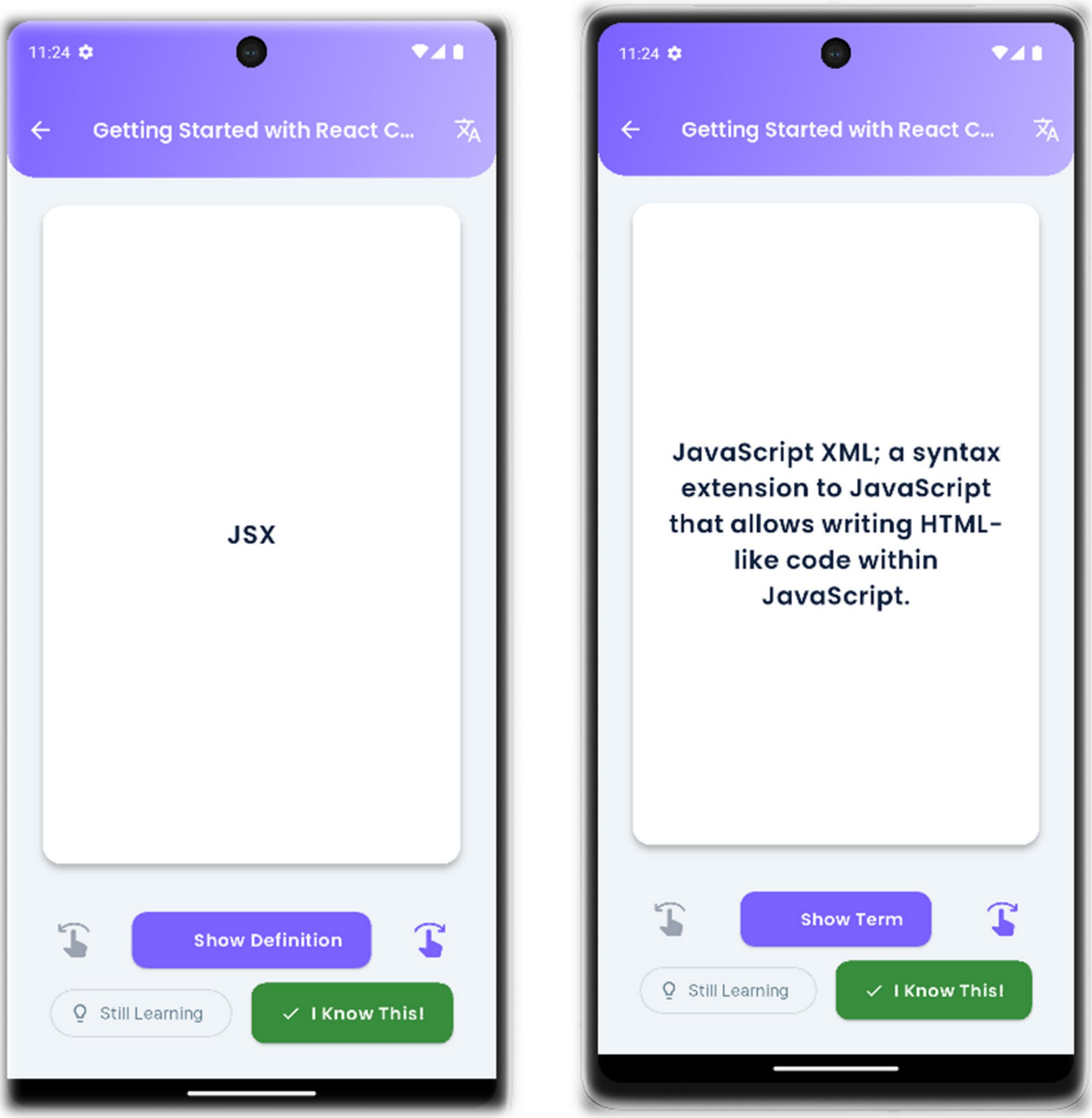


Figure 33 Flashcards AI format

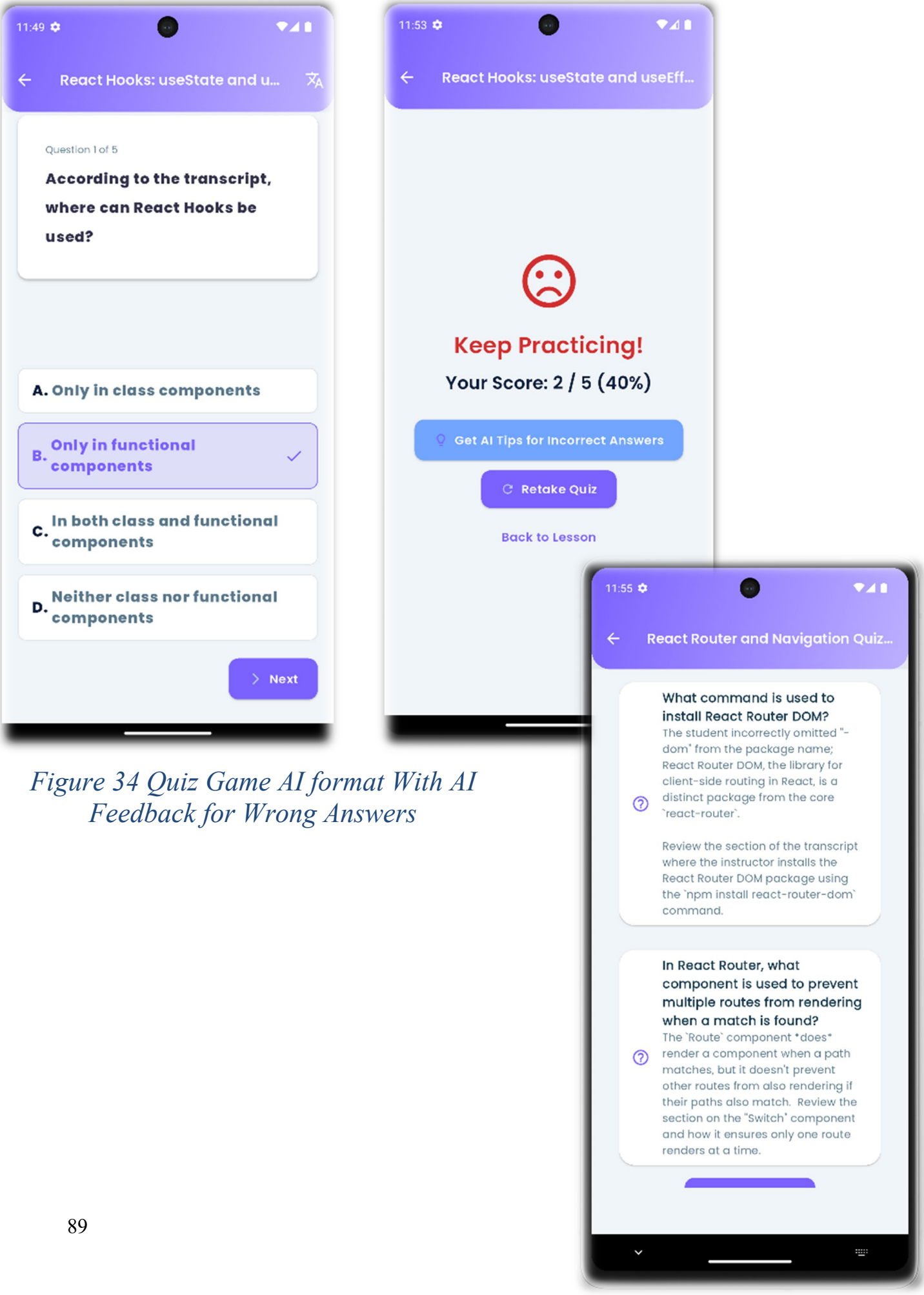


Figure 34 Quiz Game AI format With AI Feedback for Wrong Answers

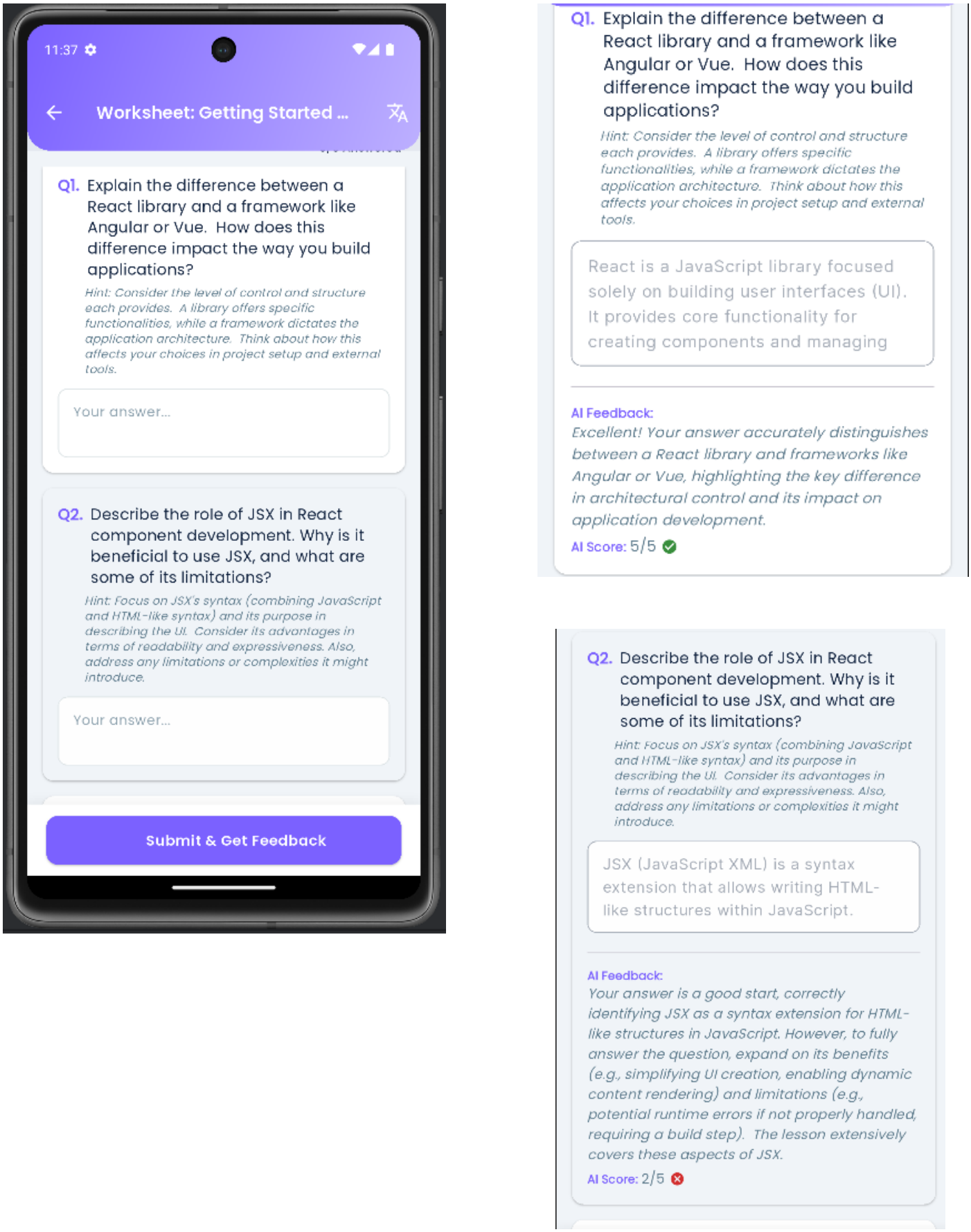


Figure 35 Worksheet Format with AI evaluation (score out of 5 + feedback)

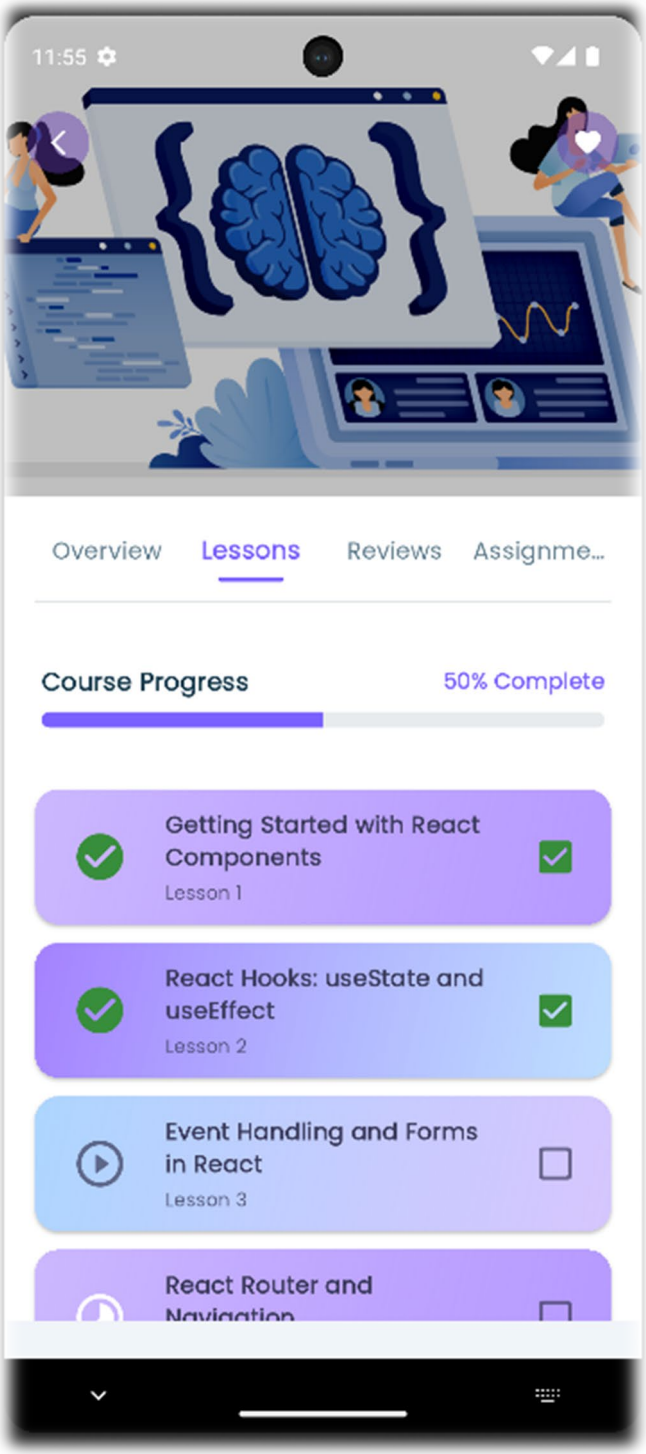


Figure 36 Course progress Tracking

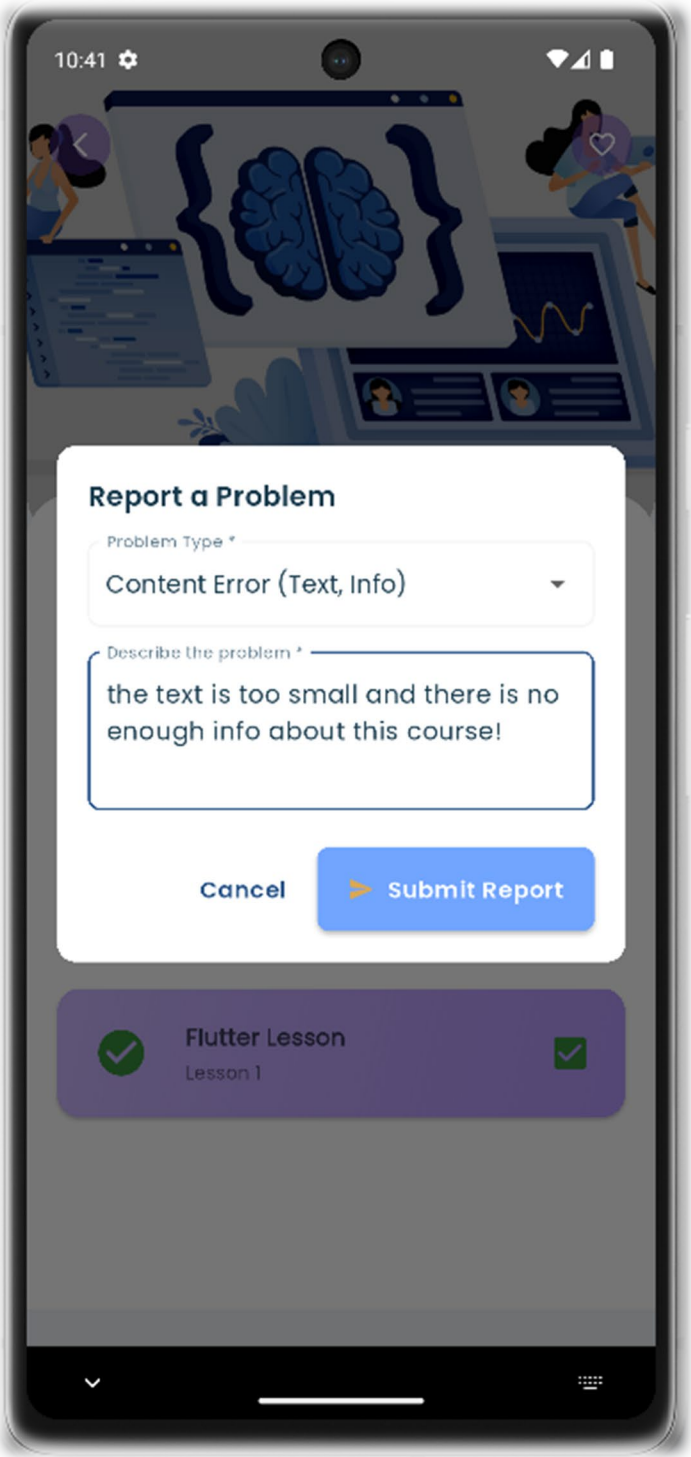


Figure 37 Report a problem to Admin

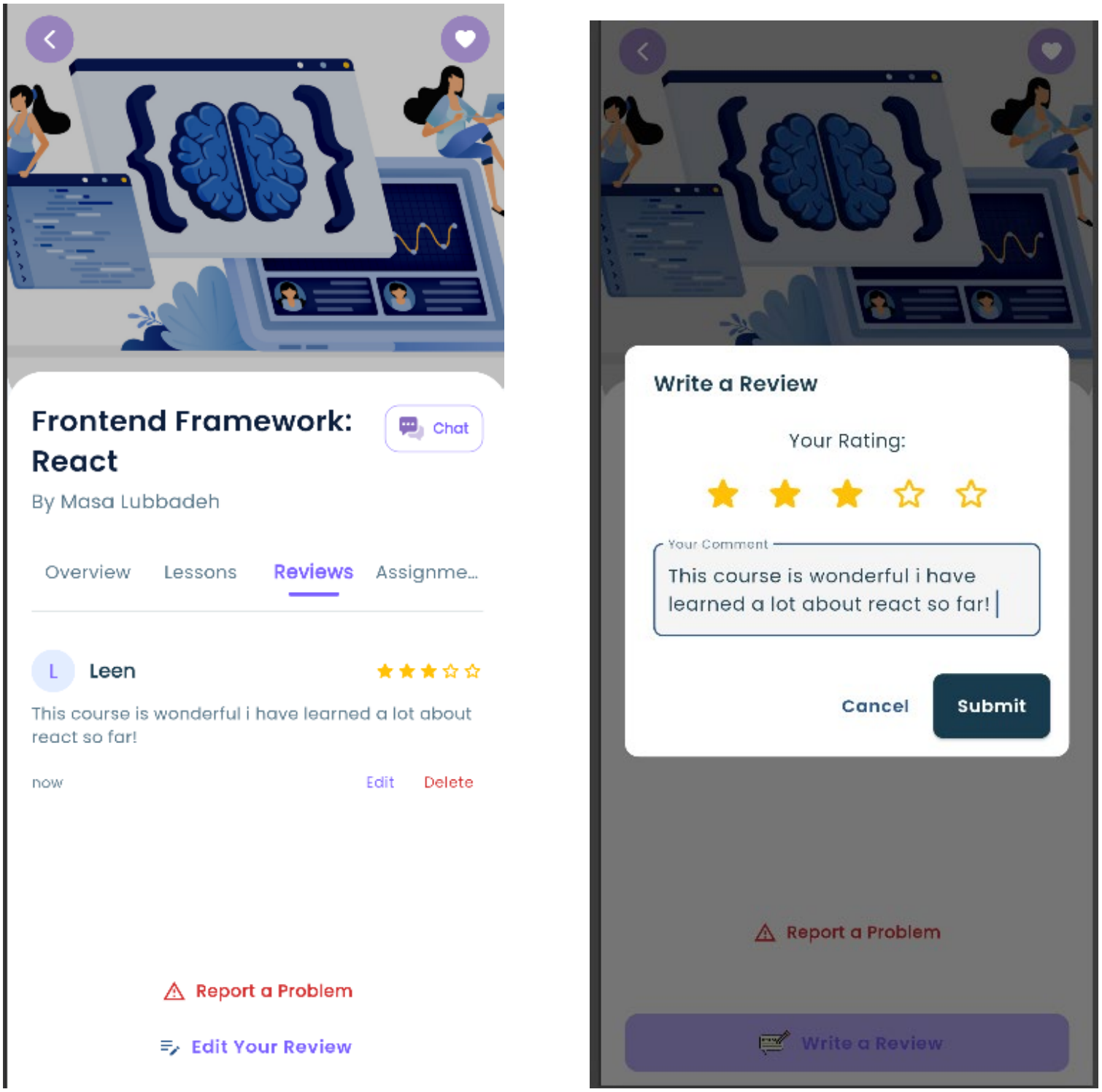


Figure 38 Course reviews

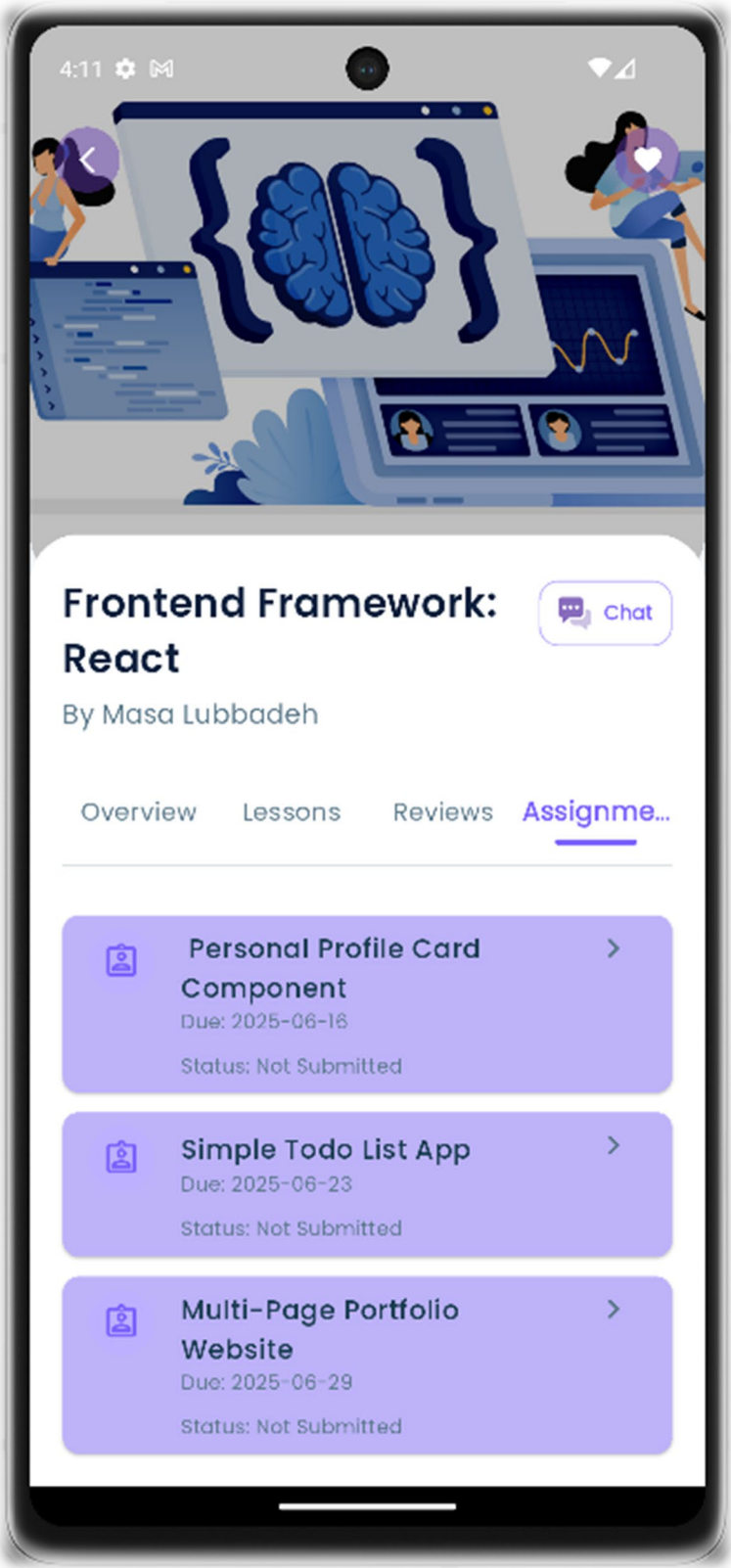


Figure 40 Assignments tab

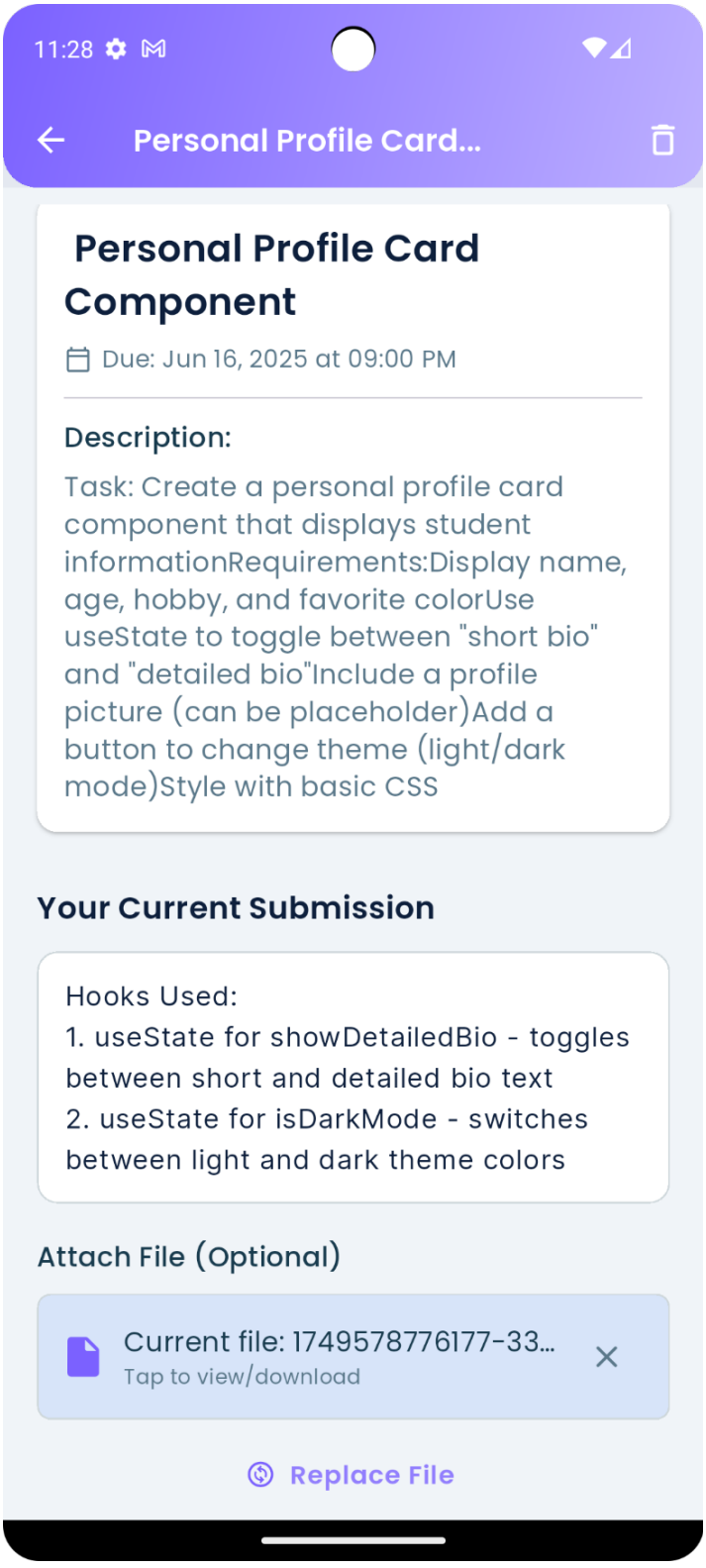


Figure 39 Assignment Submission Screen

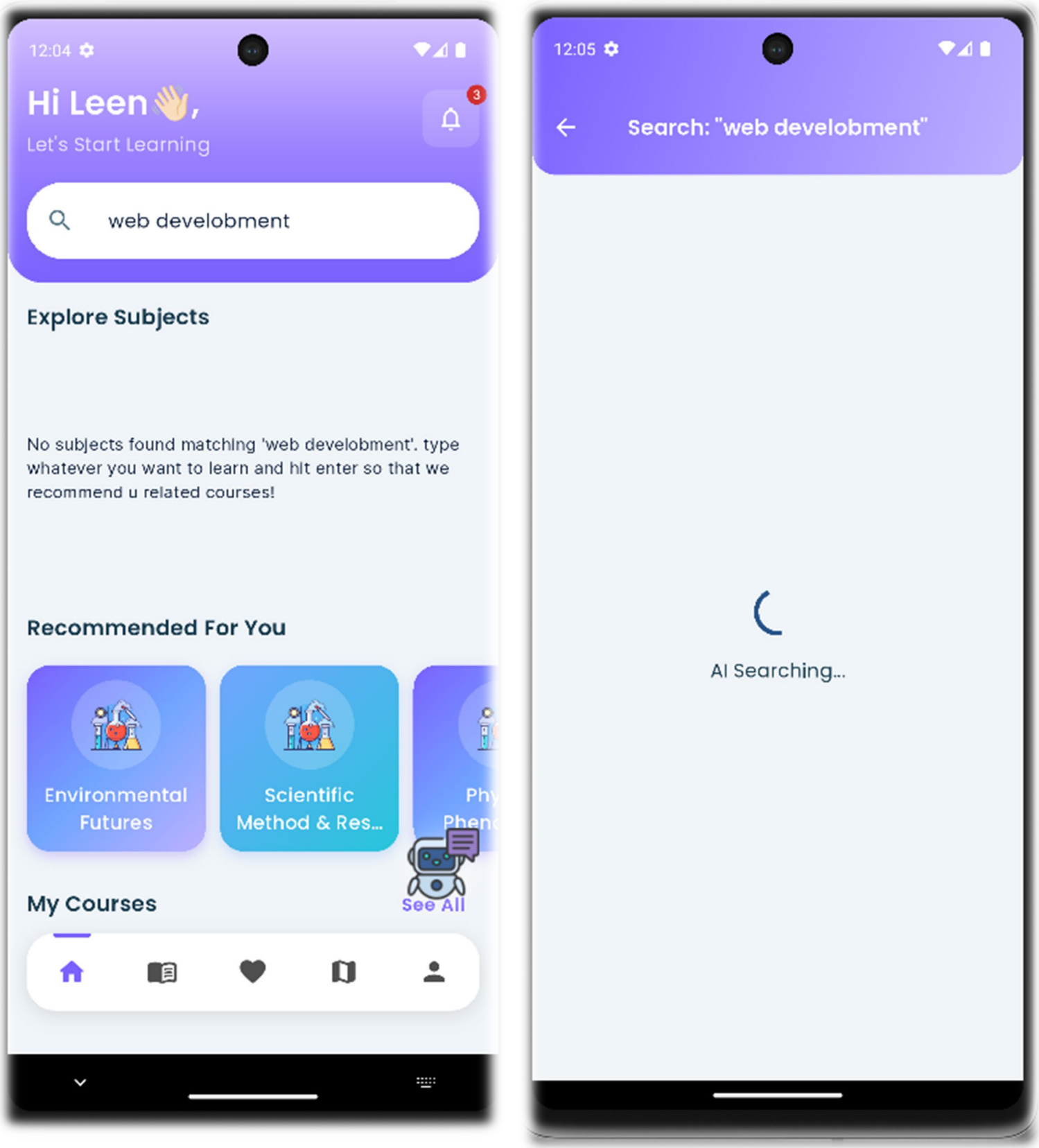


Figure 41 AI Enhanced search

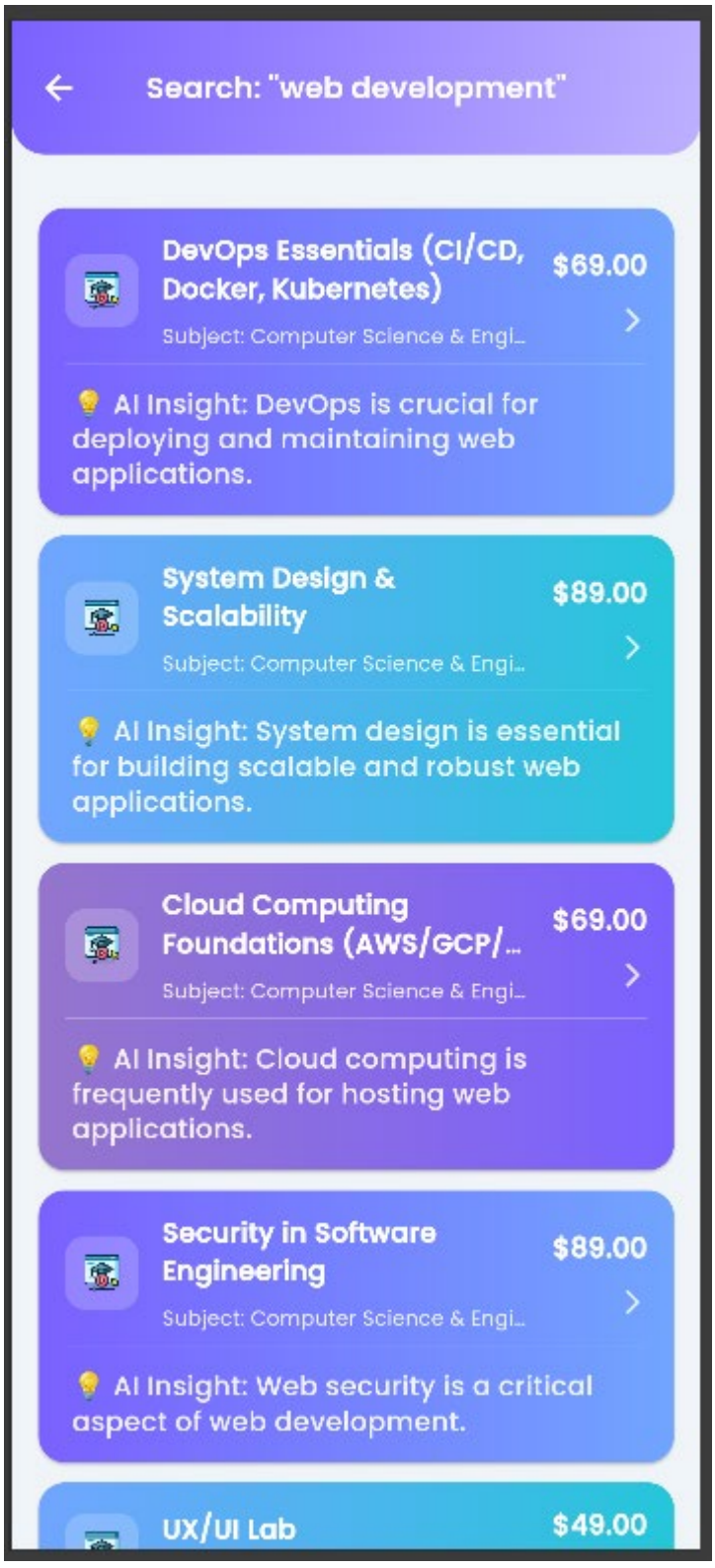


Figure 42 AI Search results with AI insights

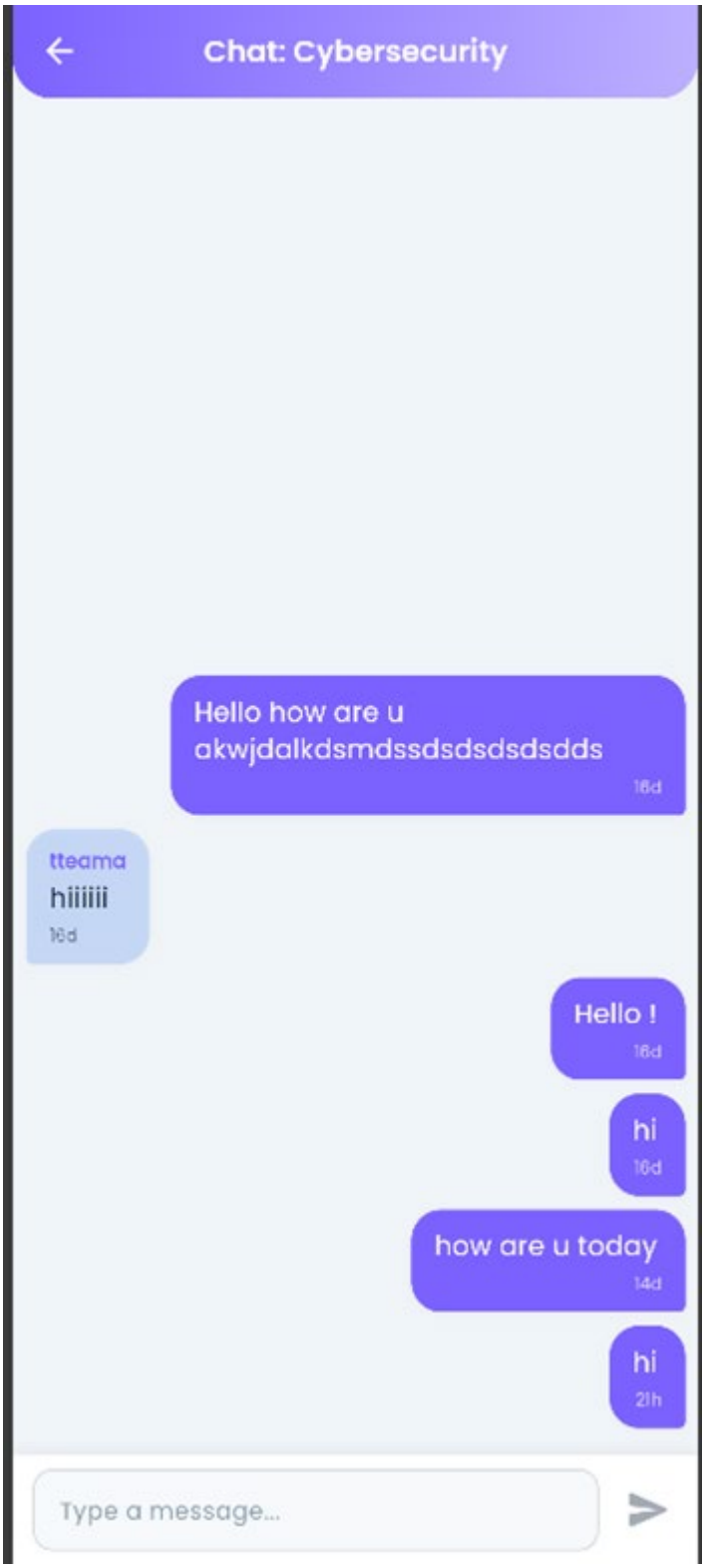


Figure 43 Course Chatroom

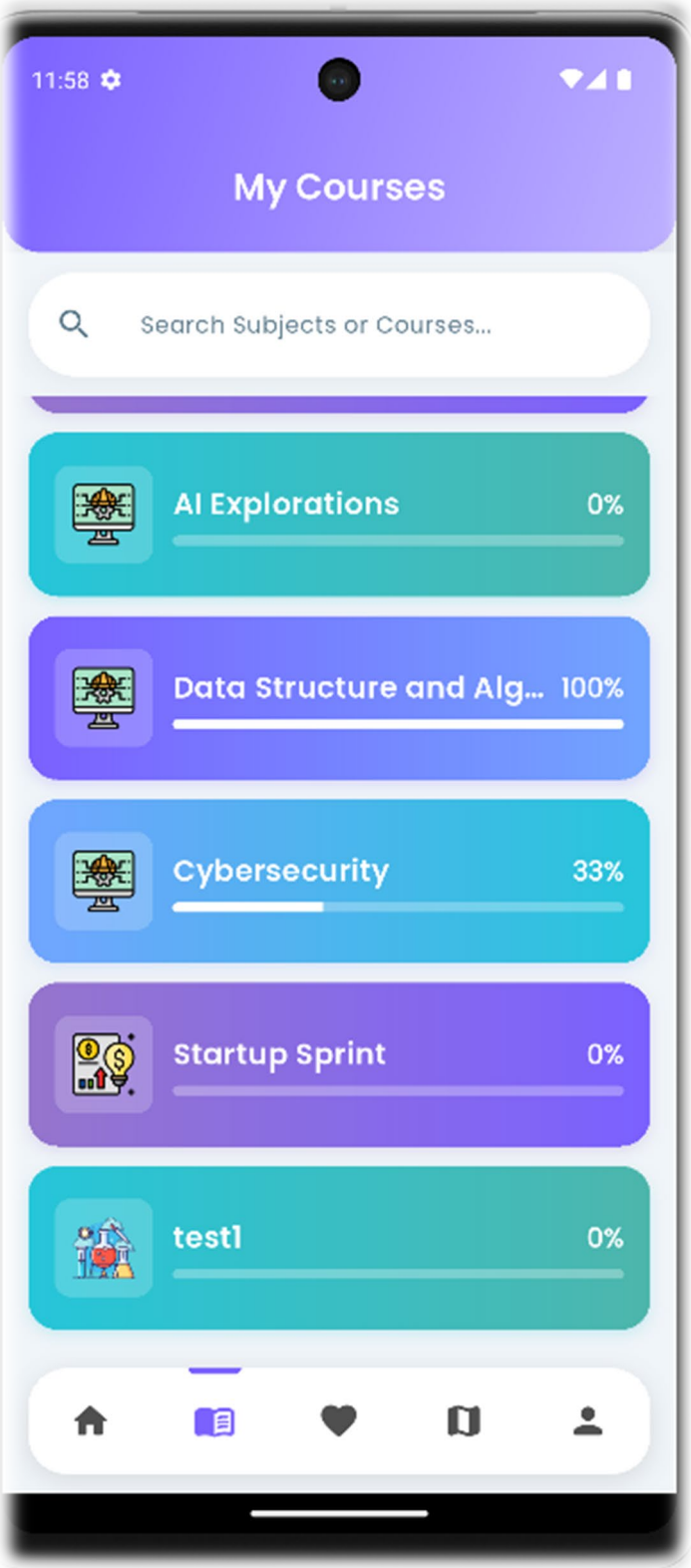


Figure 44 Student My Courses Screen

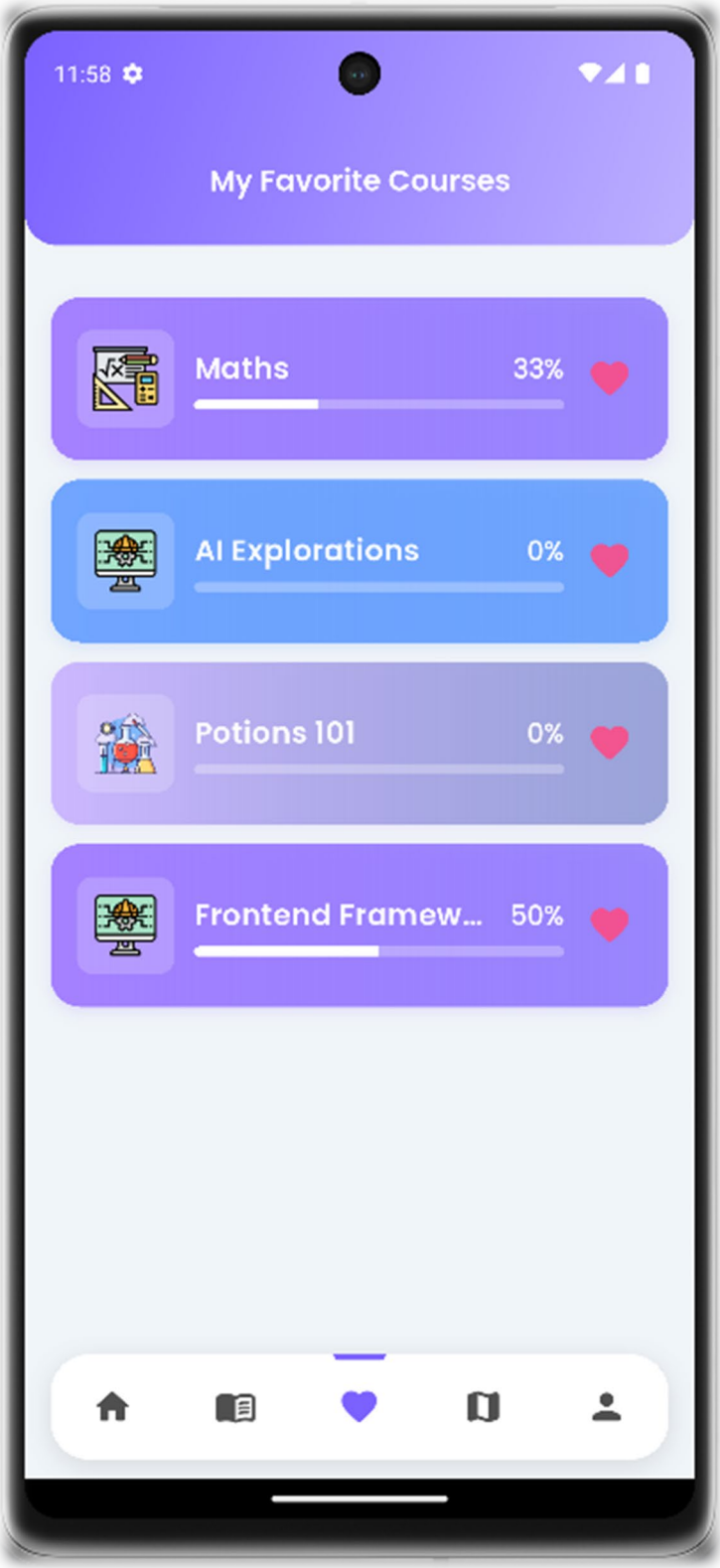


Figure 45 My favorite Courses Screen

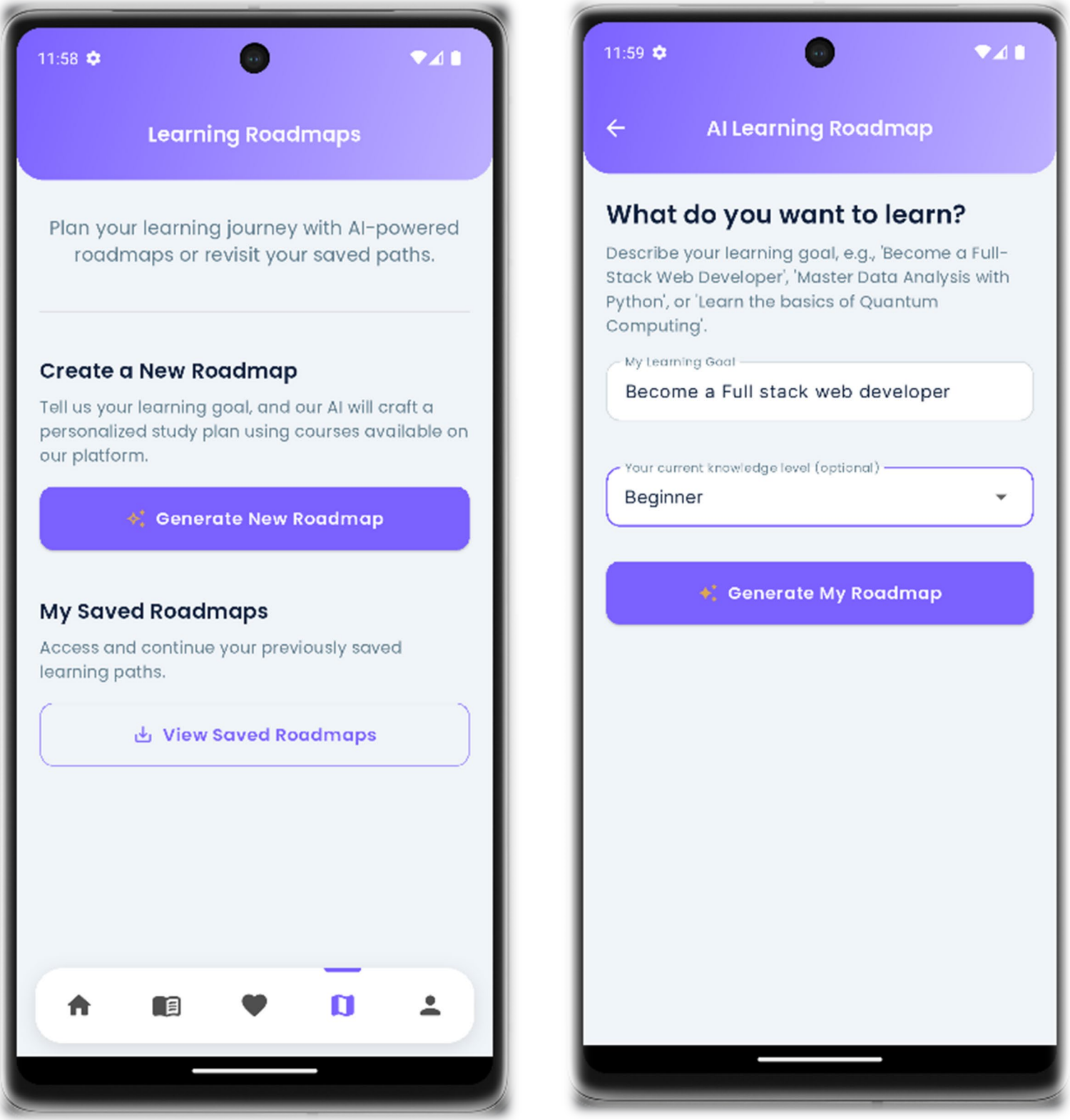


Figure 46 Learning Roadmaps Powered by AI

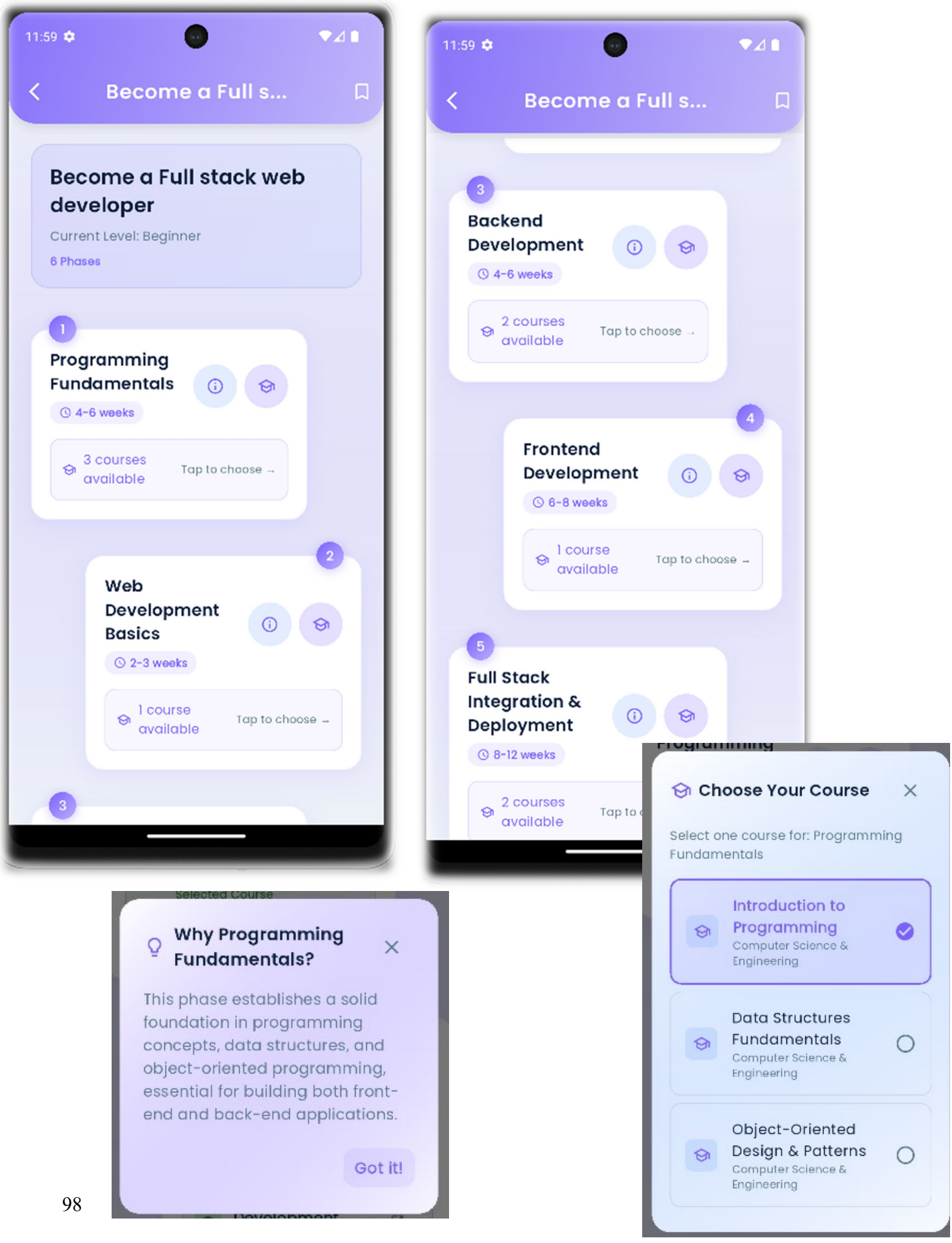


Figure 47 Roadmap Result with info and suggested courses

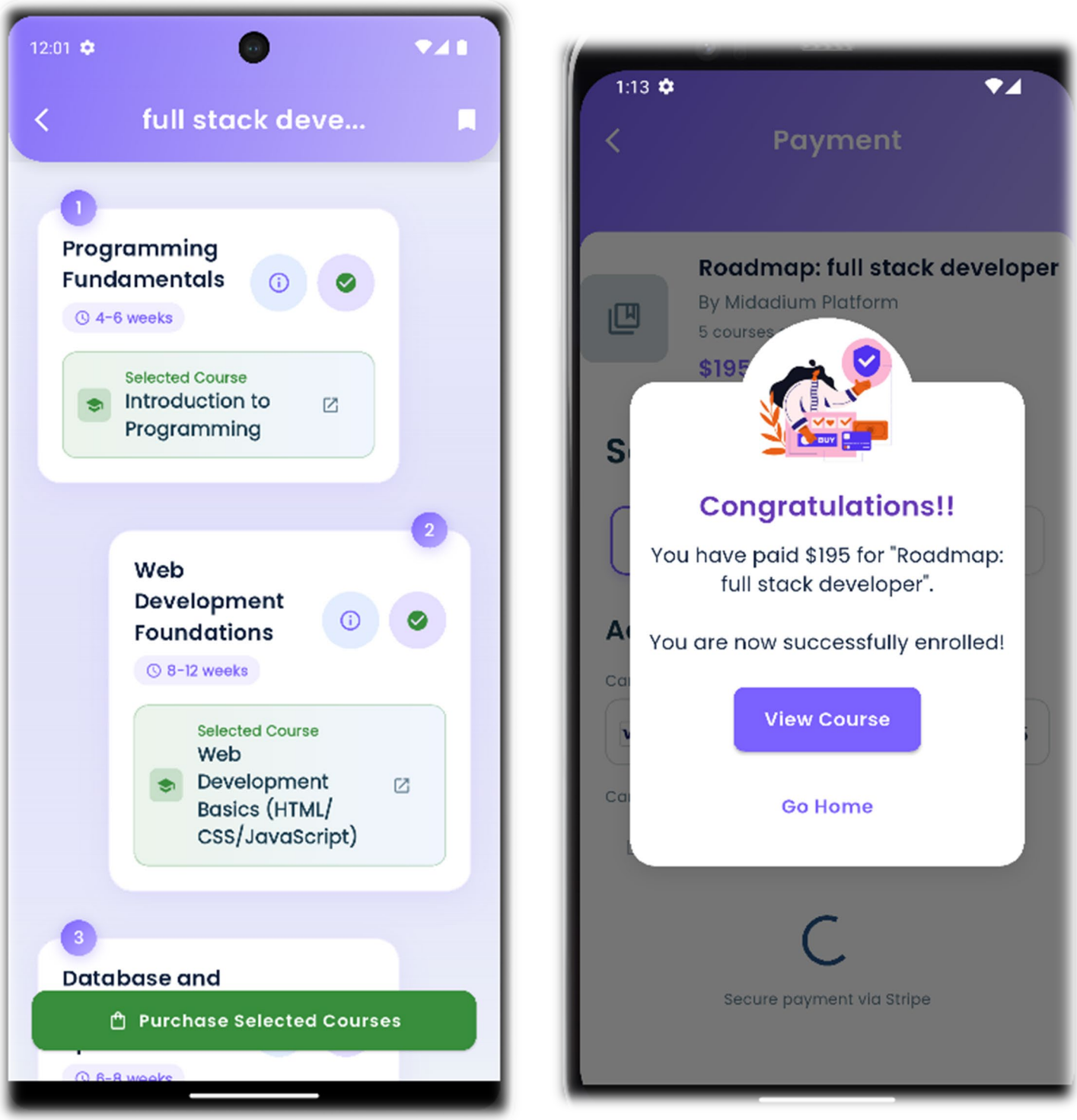


Figure 48 Roadmap course package purchase

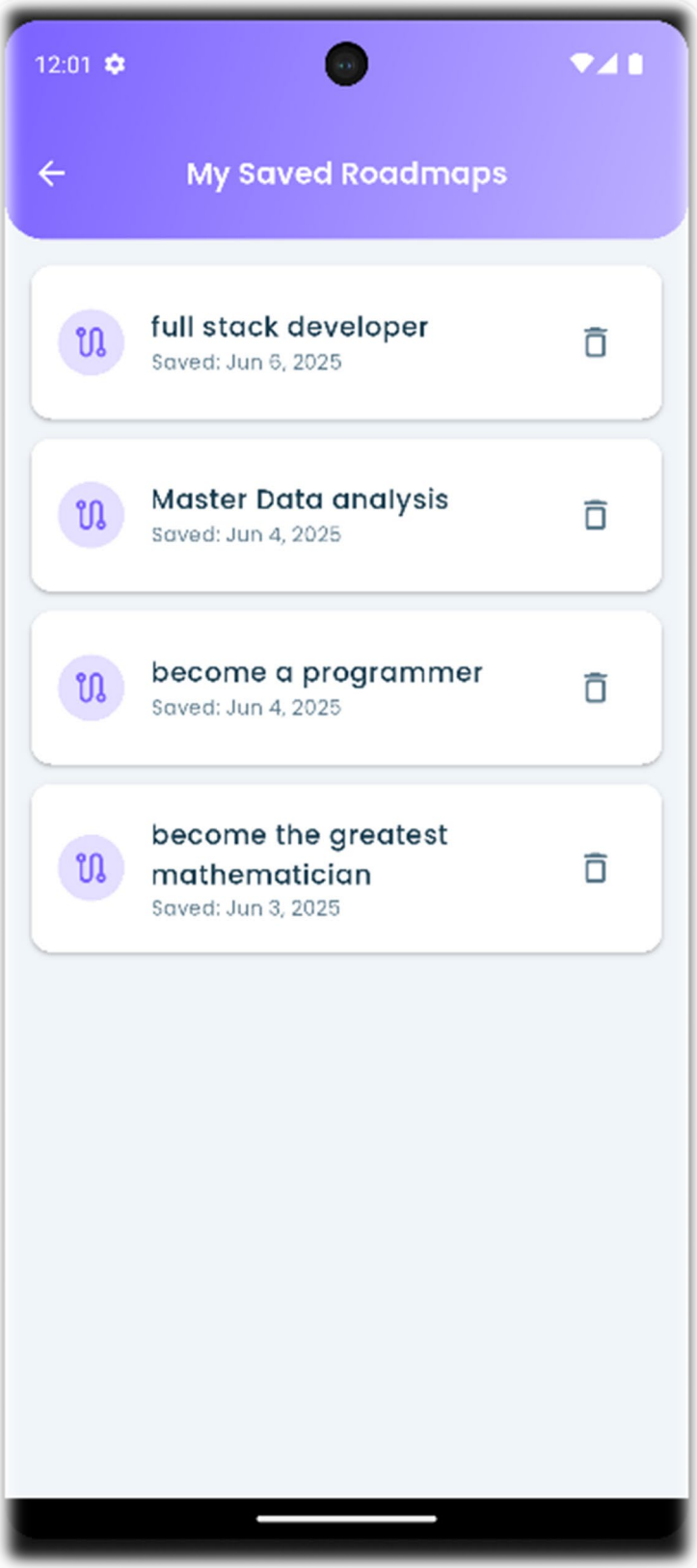


Figure 49 Saved Roadmaps Screen

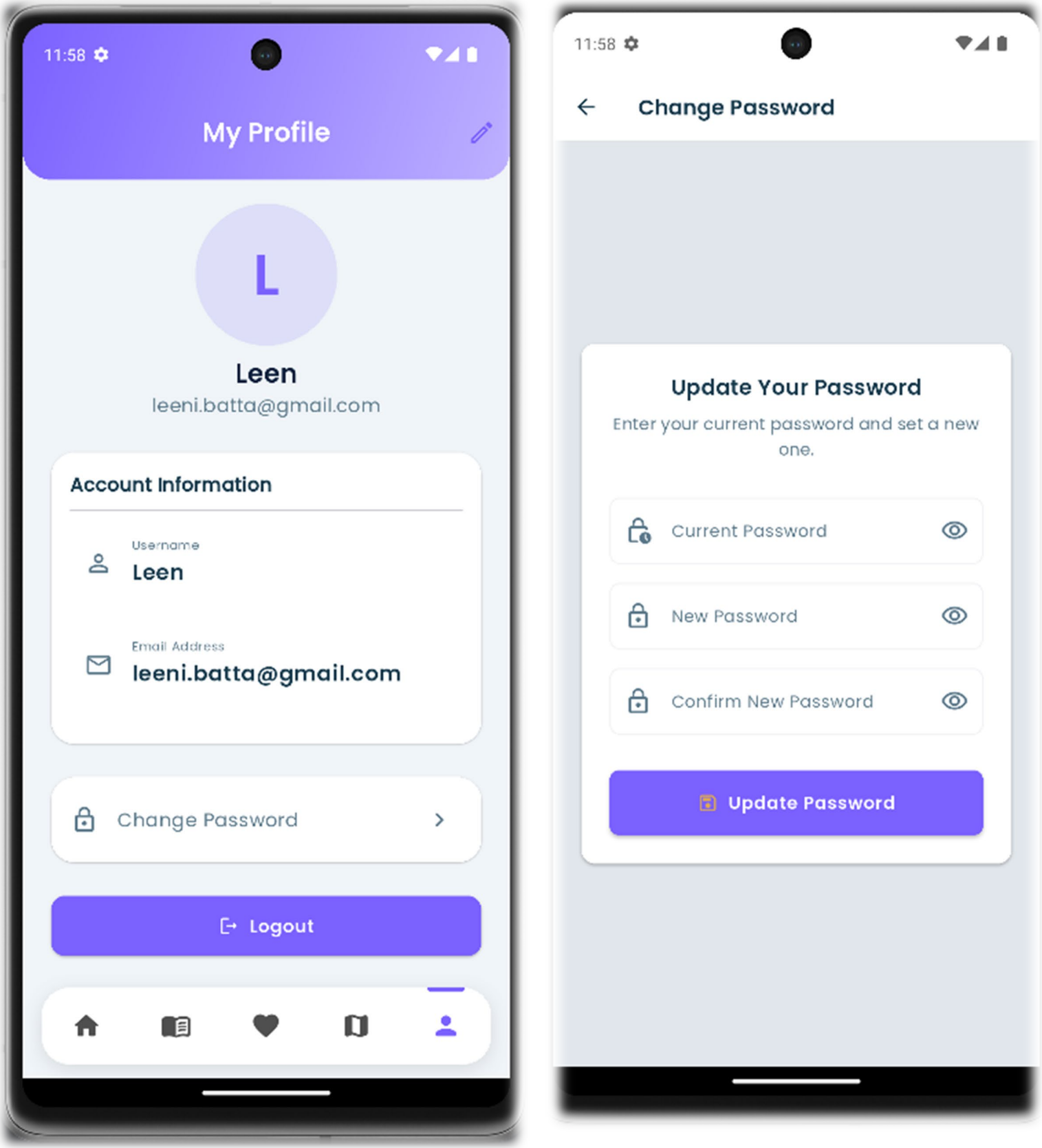


Figure 50 Profile screen and change password screen

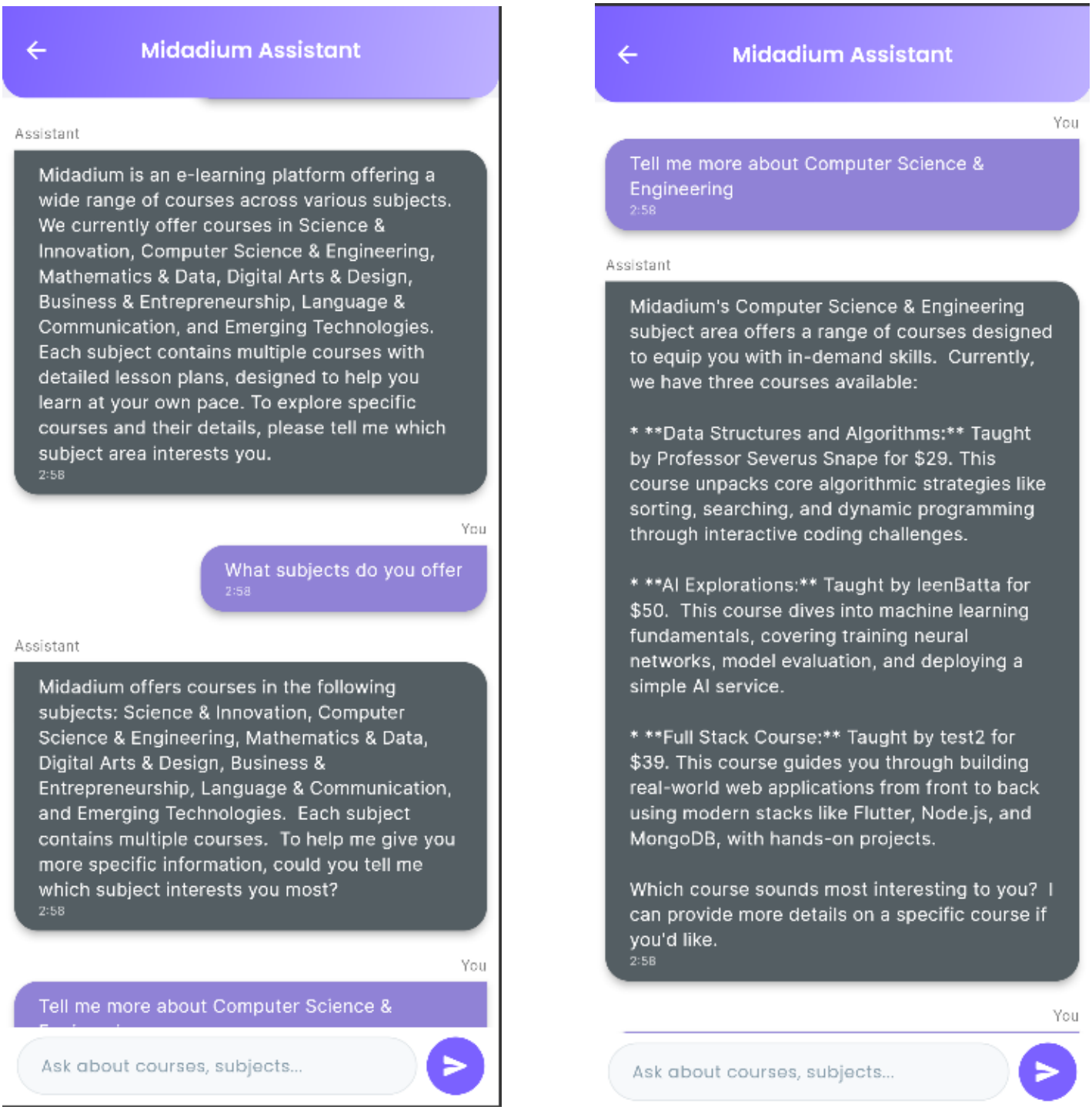


Figure 51 Chatbot

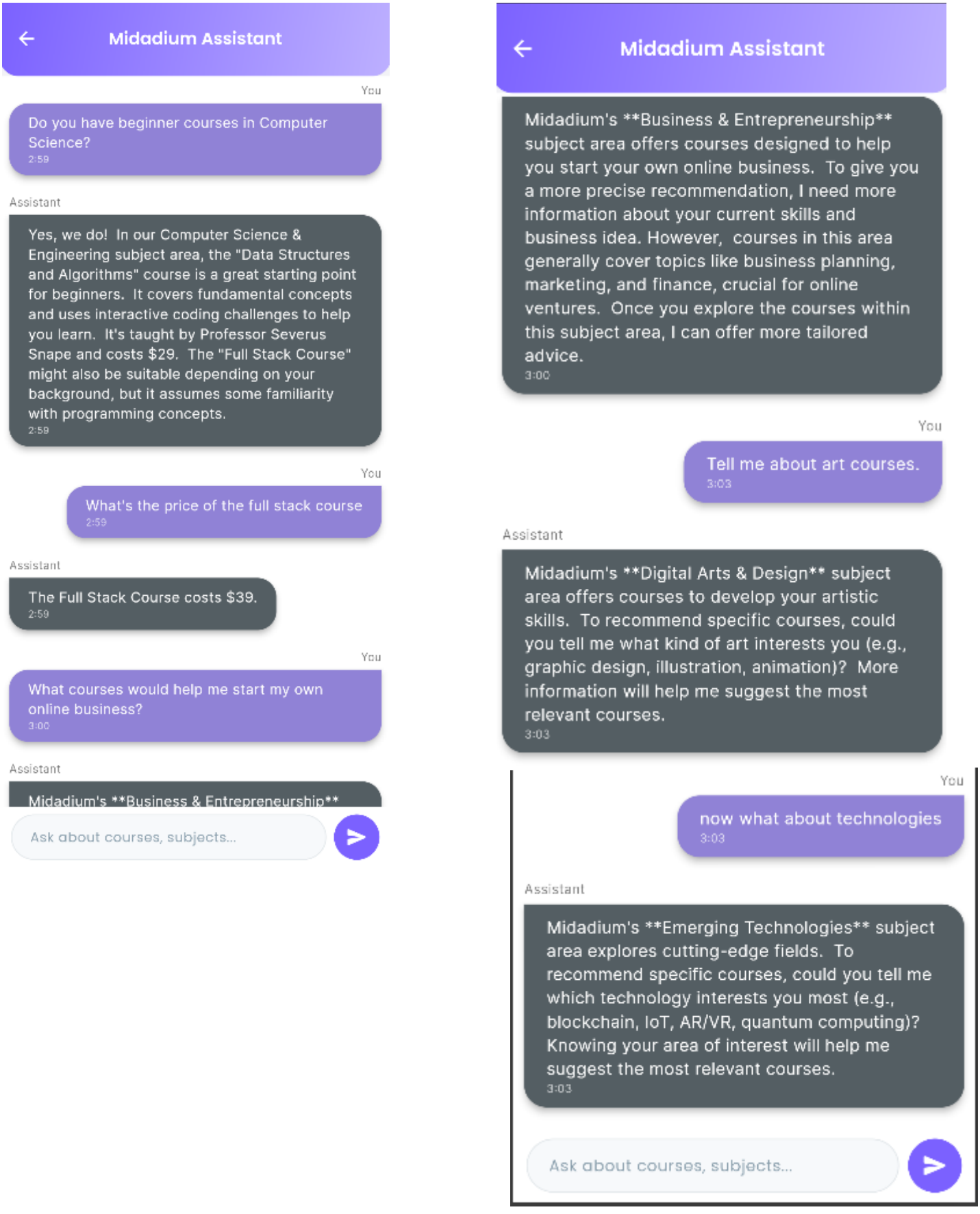


Figure 52 Chatbot

Teacher Dashboard

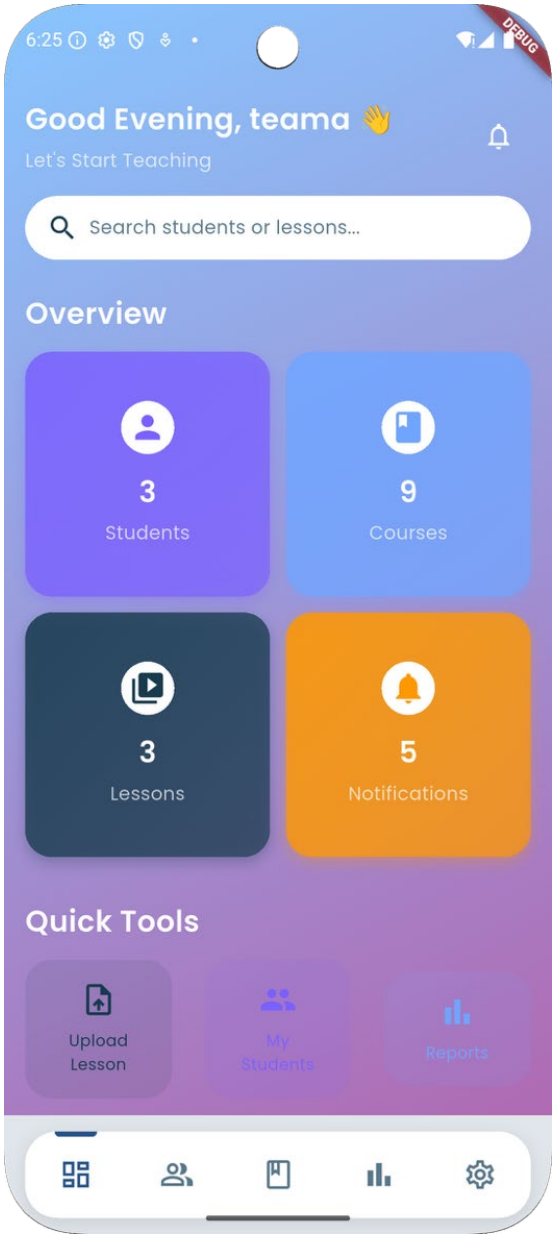


Figure 53 Teacher Overview

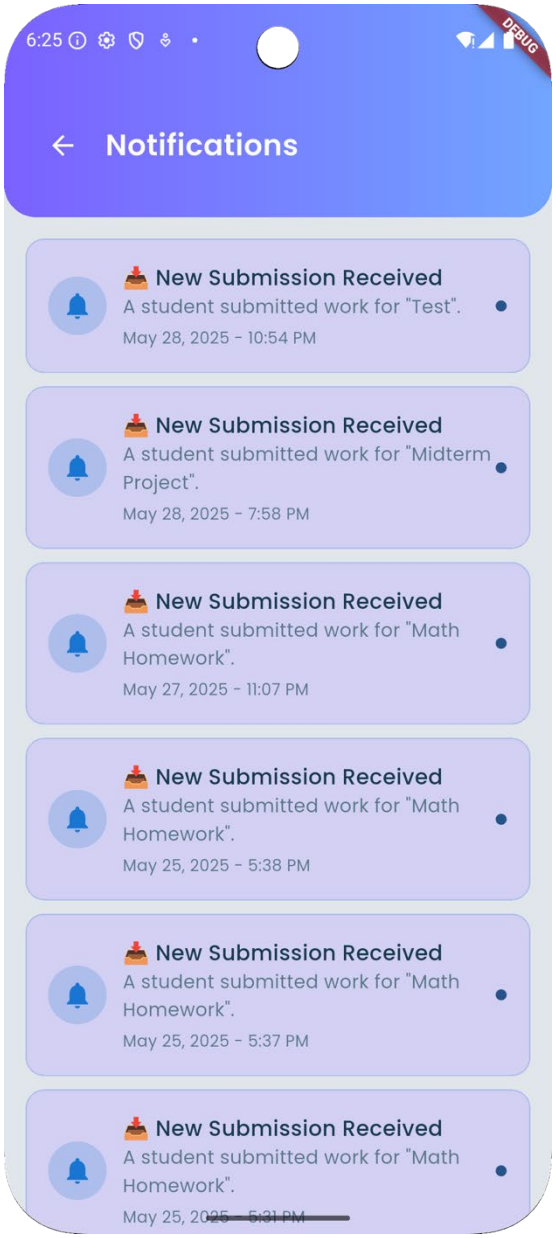


Figure 54 Teacher Notifications

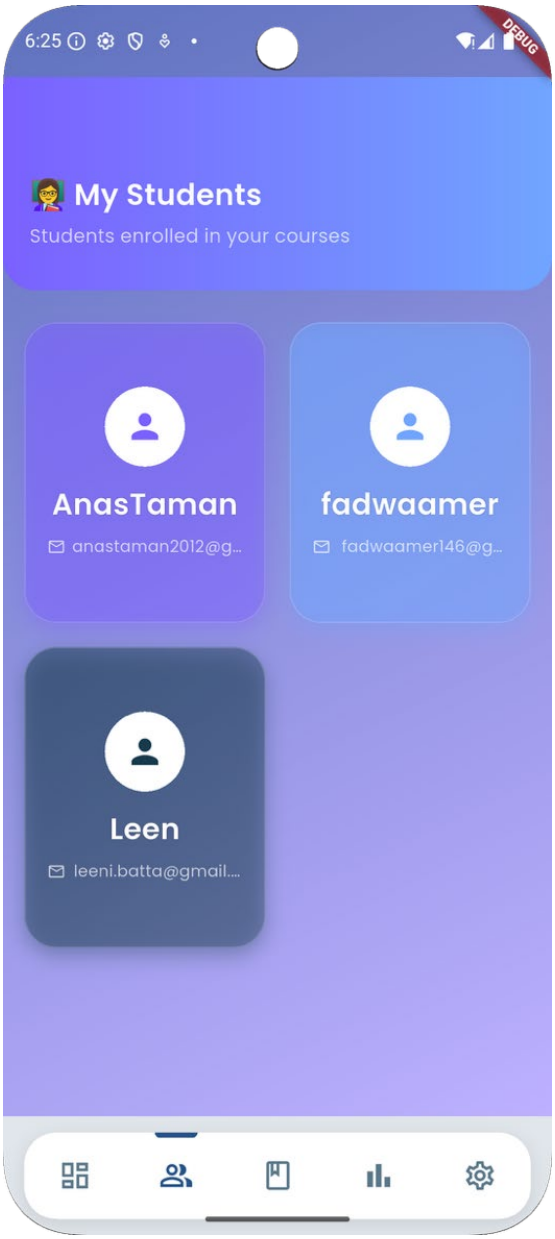


Figure 55 Students

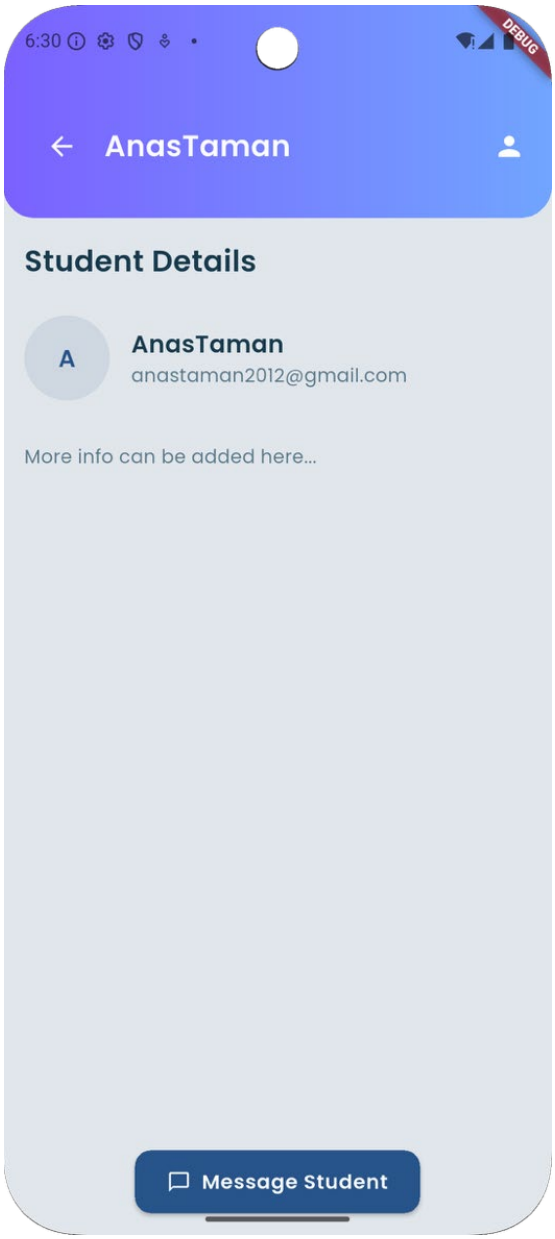


Figure 56 Student Details

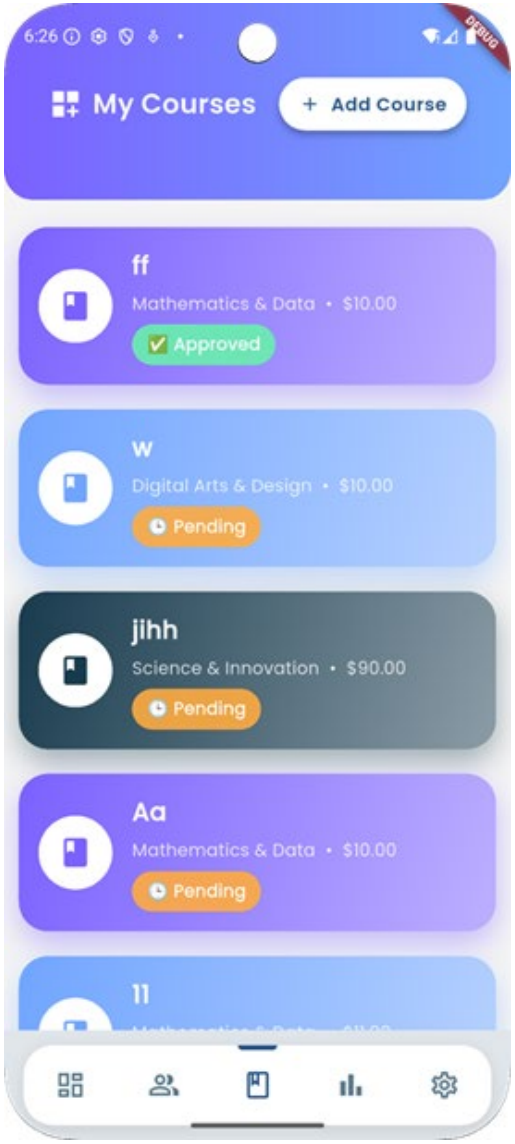


Figure 57 My Courses

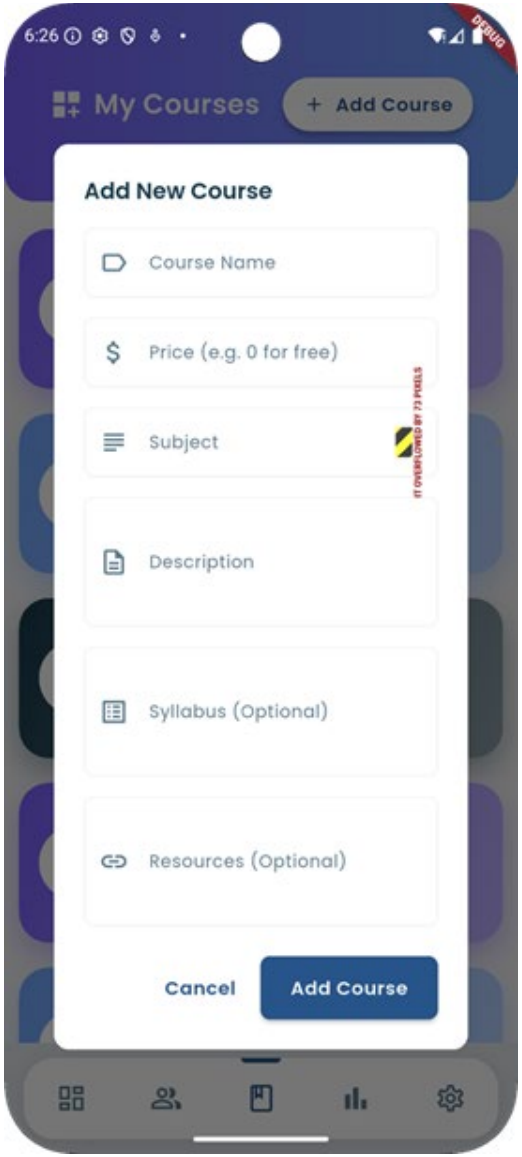


Figure 58 Add New Course

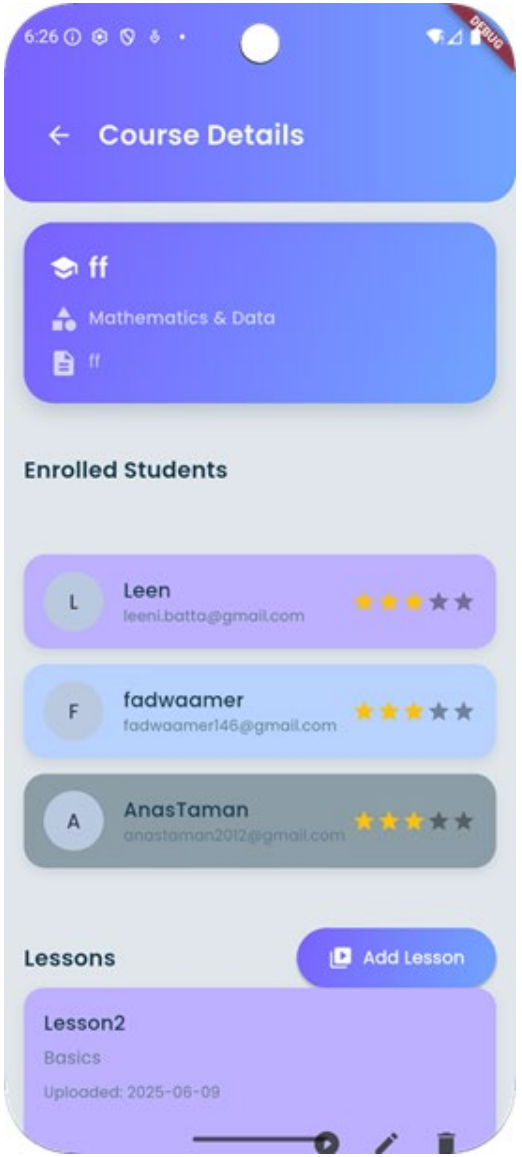


Figure 59 Course Details

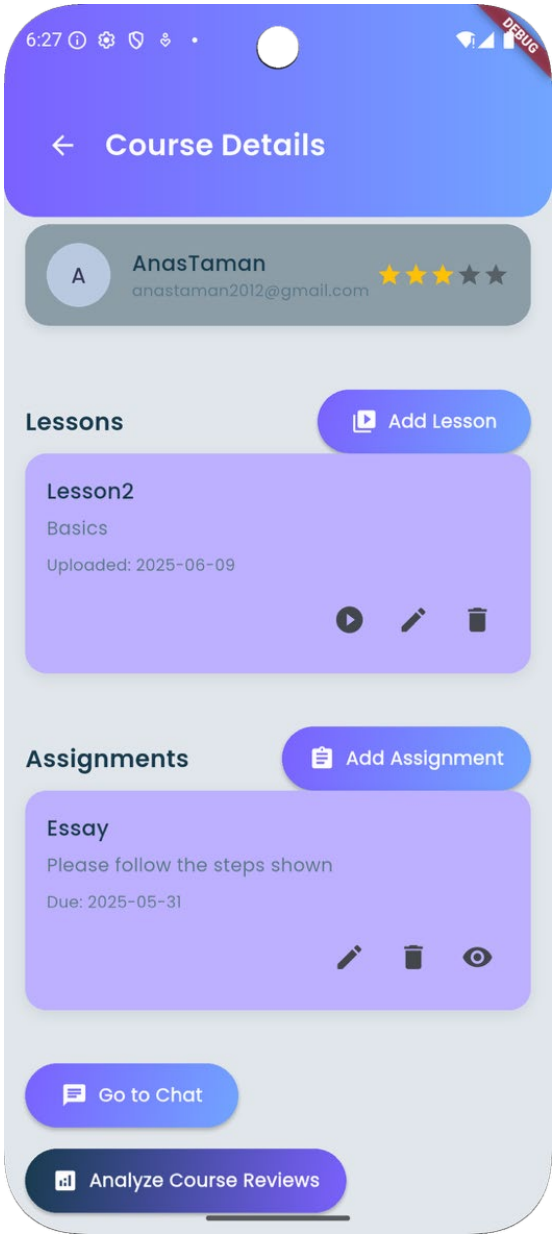


Figure 60: Course Details Cont

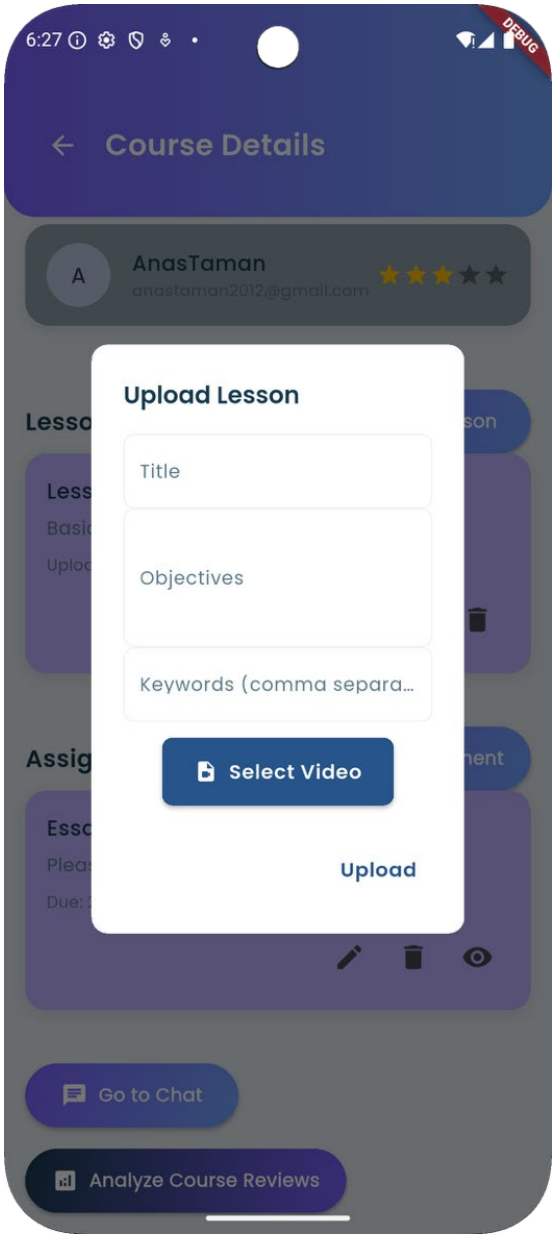


Figure 61: Add Lesson

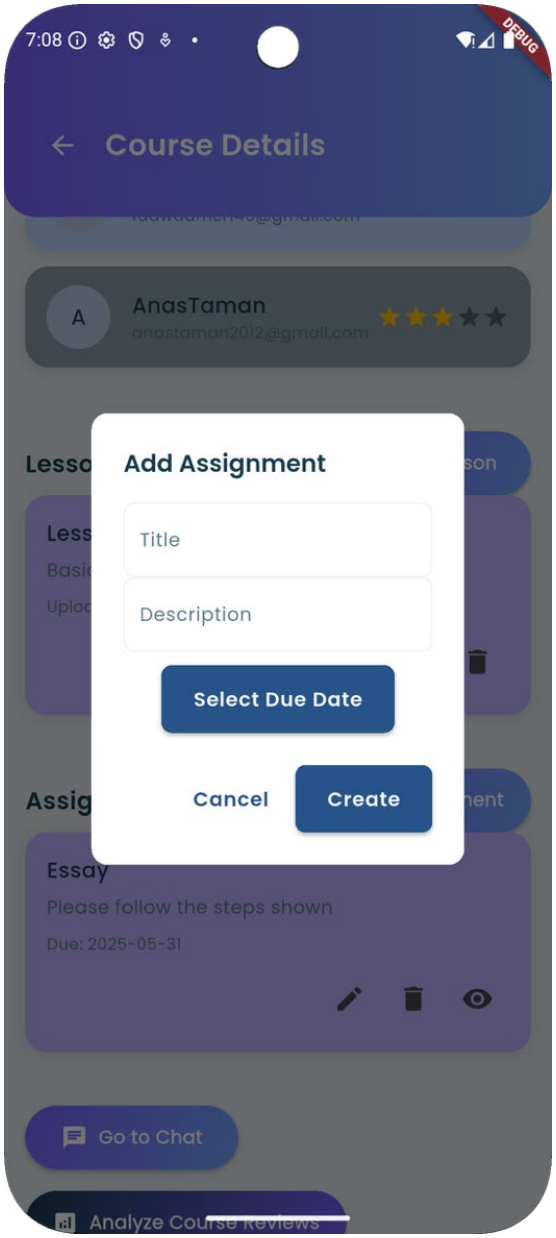


Figure 62: Add Assignment

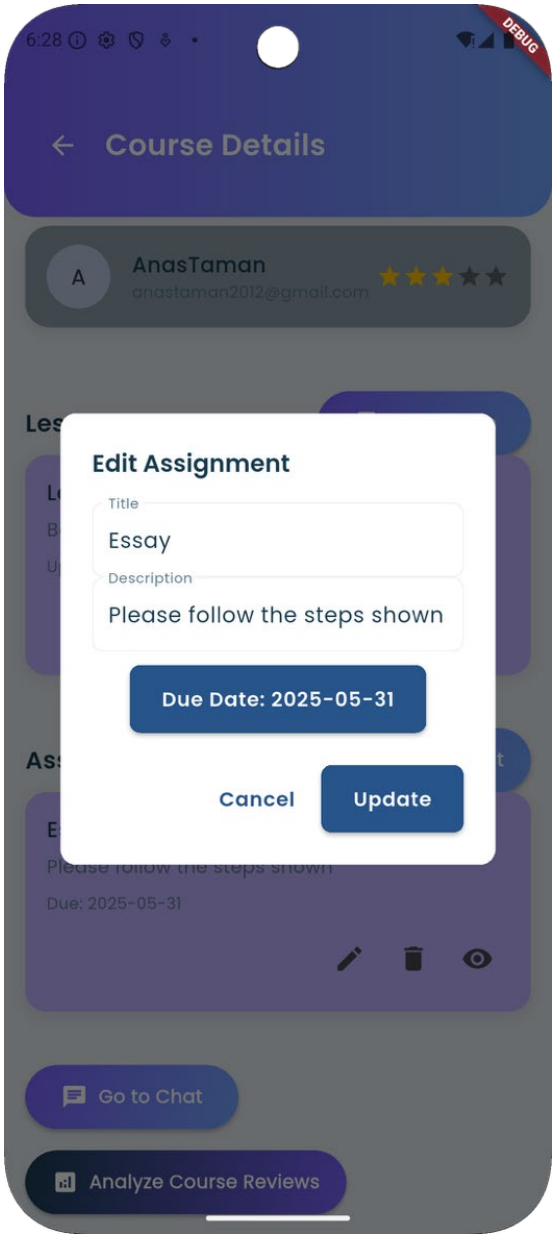


Figure 63: Edit Assignment

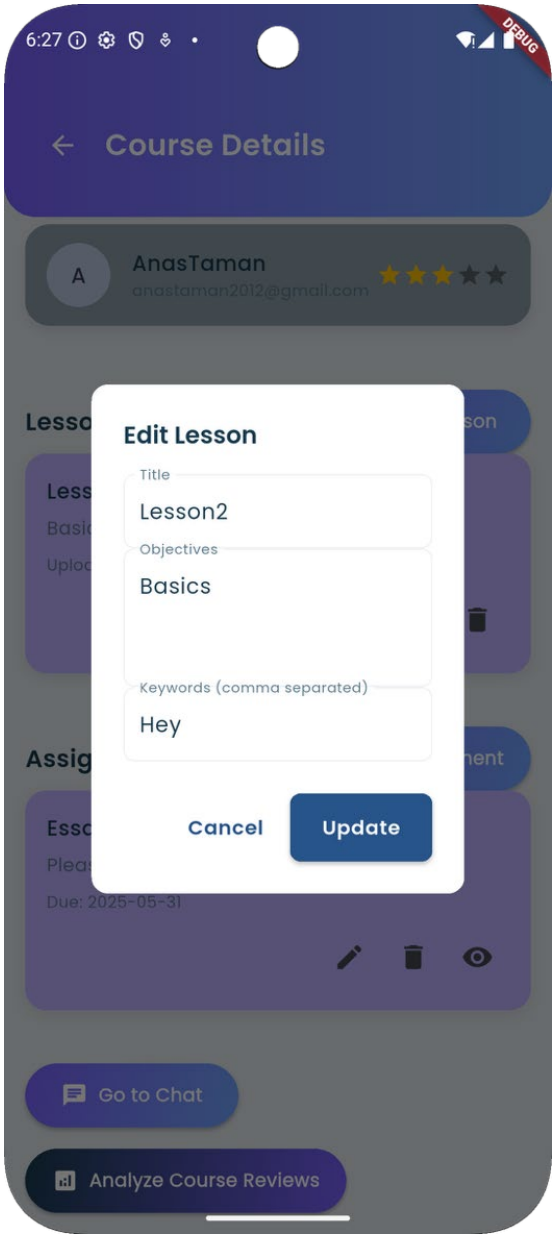


Figure 64: Edit Lesson

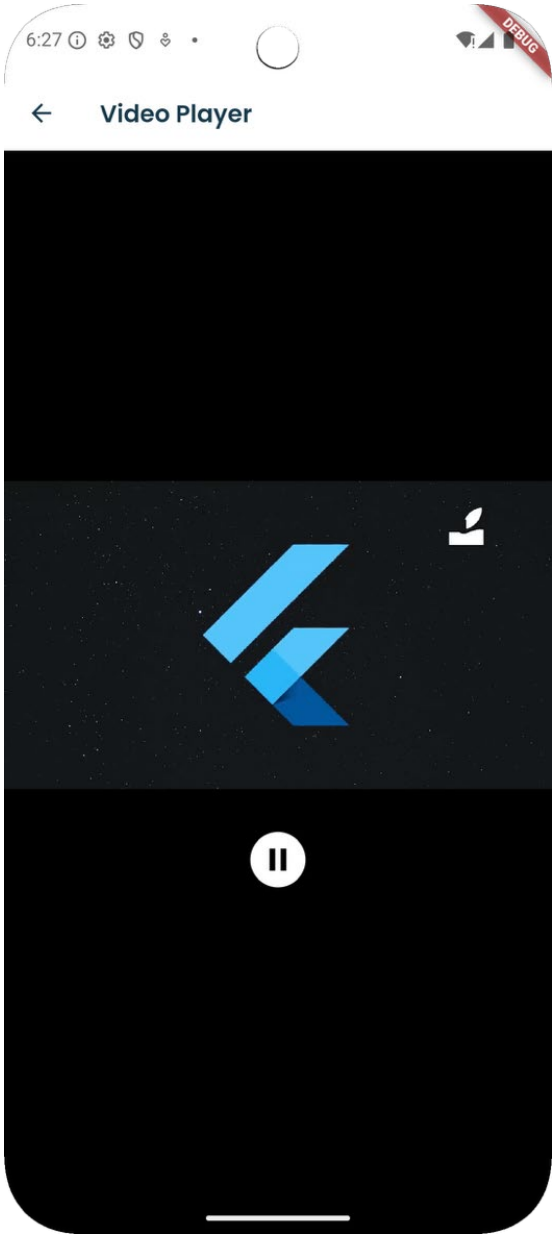


Figure 65: Video Viewer

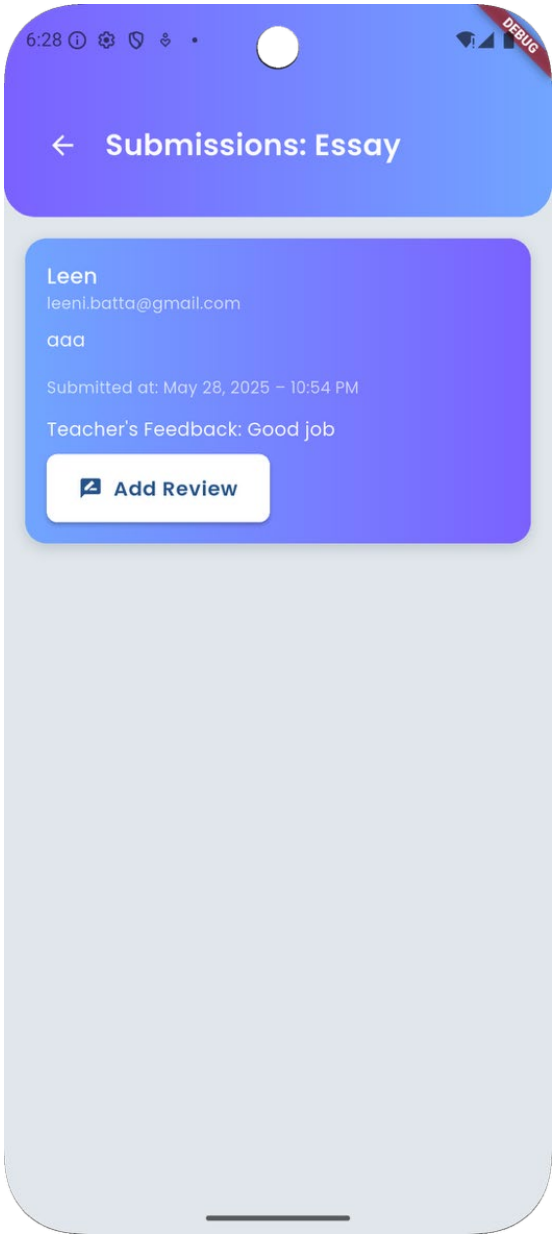


Figure 66: View Submission

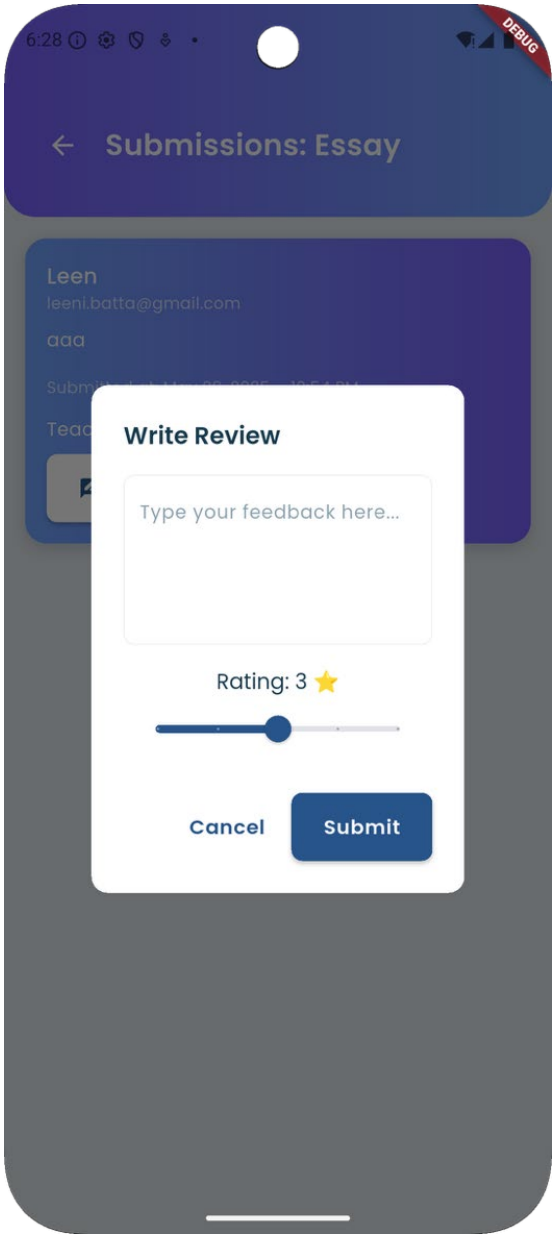


Figure 67: Review Submission

Figure 68: Performance Report

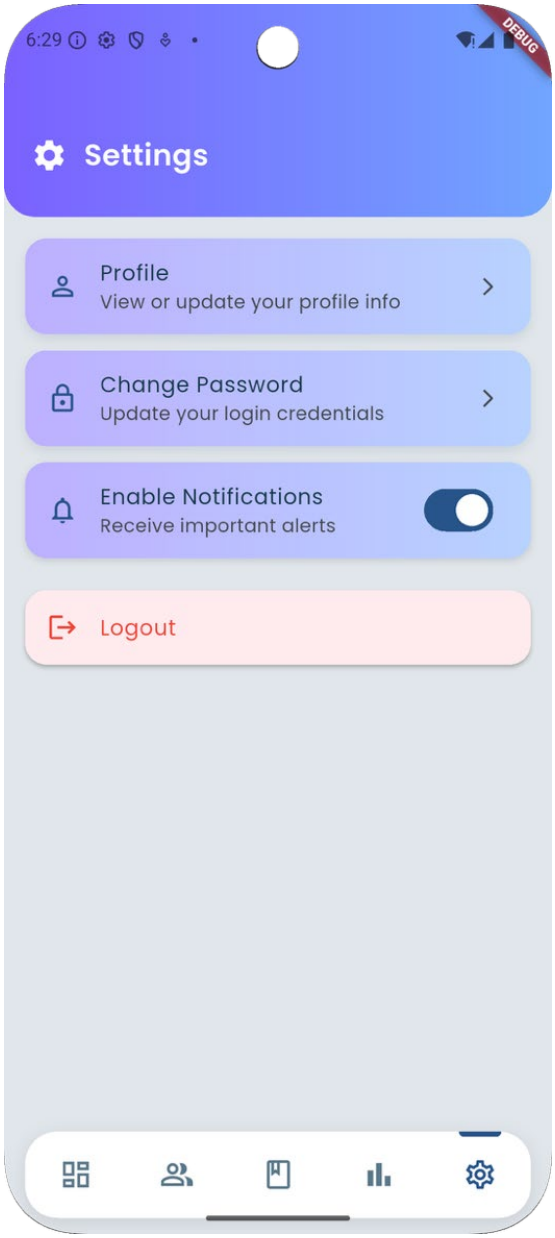


Figure 69: Settings

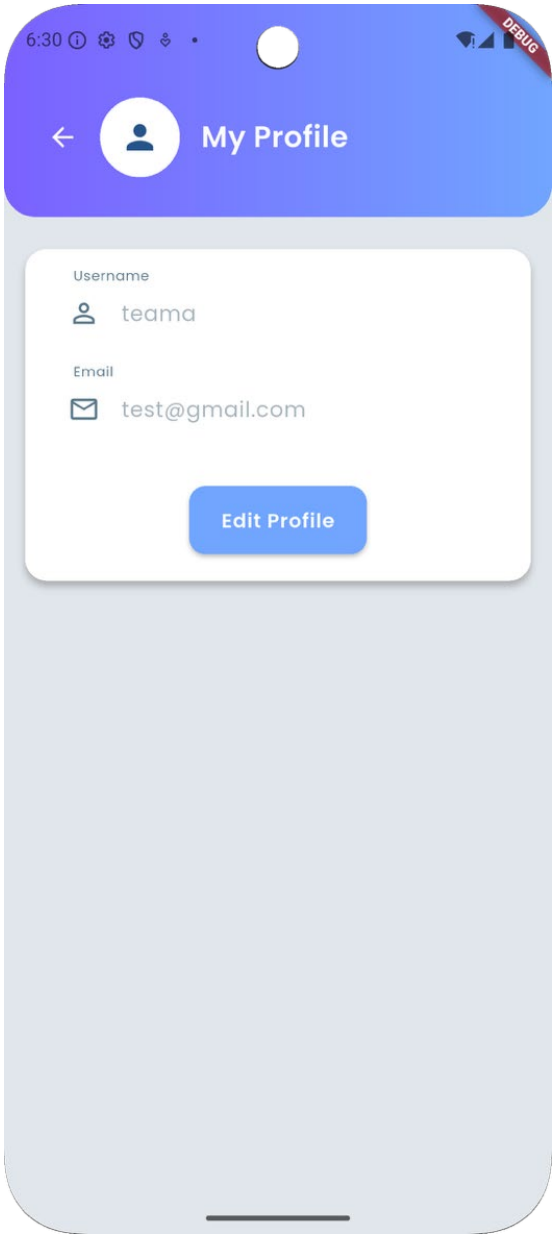


Figure 70: Profile

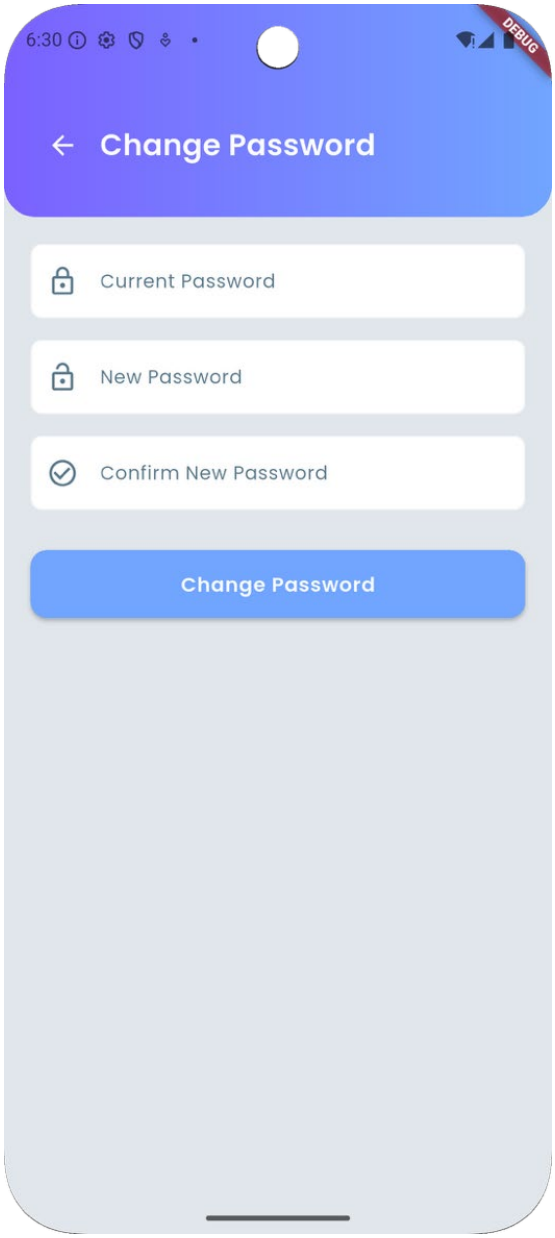


Figure 71: Change Password

References

[1] Firebase. (n.d.). *Firebase documentation*. Google. Retrieved from https://firebase.google.com/docs

[2]Flutter. (n.d.). *Build apps for any screen*. Google. Retrieved from https://flutter.dev

[3]Gemini API. (2024). *Generative AI API reference*. Google AI. Retrieved from https://ai.google.dev

[4]AssemblyAI. (n.d.). *Speech-to-text API documentation*. Retrieved from https://docs.assemblyai.com

[5]MongoDB. (n.d.). *MongoDB manual*. Retrieved from https://www.mongodb.com/docs/manual/

[6]Node.js. (n.d.). *Node.js documentation*. Retrieved from https://nodejs.org/en/docs

[7]Socket.IO. (n.d.). *Real-time communication documentation*. Retrieved from https://socket.io/docs/