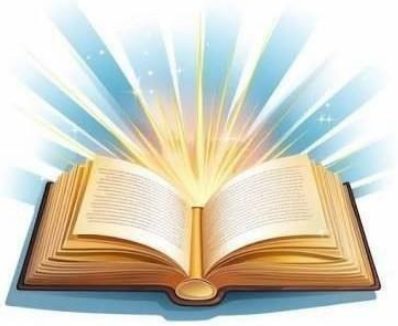
AN-NAJAH NATIONAL UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

Bright path to learning



PREPARED BY:
Reem Abd Al Raheem Hasan
Farah Faisal Ahmad

SUPERVISED BY:
Dr. Shareef Yaseen

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

20 jan 2026

Acknowledgment

First of all, we would like to thank our supervisor, Dr. Shareef Yaseen, for his continuous guidance and valuable feedback during this project. His support and encouragement were essential to the completion of our work. We also extend our thanks to all the faculty members of the Computer Department for the knowledge and skills they have provided us throughout our academic journey. Finally, we are deeply grateful to our families and friends for their patience, motivation, and unwavering belief in us.

Disclamer

This project has been prepared by Reem Hasan and Farah Faisal as a partial requirement for the Bachelor's degree in the Computer Engineering Department. The content of this report has not been modified or corrected by anyone, as it will be evaluated solely by the professors of An-Najah National University. It should be noted that An-Najah National University holds no responsibility for any statements, results, or conclusions presented in this report.

Abstract

This project aims to develop an intelligent mobile and web application that helps students organize their studies in an interactive and efficient way by integrating artificial intelligence with a modern and user-friendly interface. The application analyzes each student's learning style to personalize their educational experience, whether visual, auditory, or kinesthetic.

The system allows users to upload their study materials such as books, PDFs, or images, and automatically generates summaries, quizzes, and flashcards to enhance understanding and retention. It also provides an adaptive study plan that adjusts based on the student's performance and available time, along with motivational quotes and Islamic reminders to maintain focus and positivity throughout the learning journey.

The mobile version of the application will be built using React Native and Expo to ensure a smooth, responsive, and cross-platform user experience, while the web version will be developed using React.js for an intuitive browser-based interface.

The Back-End will be developed using Node.js and Express.js, with MySQL as the main database for efficient and structured data management. The system will also be integrated with the OpenAI API to enable intelligent features such as automatic summarization, question generation, and personalized study planning. Additionally, the project includes an Administrator Dashboard for monitoring user activity, analyzing learning performance, and managing educational content effectively.

This project combines artificial intelligence, smart organization, and motivational design to deliver a complete digital learning experience that empowers students to achieve their academic goals with confidence and efficiency.

Contents

Acknowledgment 2

Disclamer 3

Abstract 4

Table Of Figures 7

Chapter 1: Introduction 9

1.1 General Background 9

1.2 Problem Statement 9

1.3 Objectives of the Project 10

1.4 Significance of the Project 10

1.5 Organization of the Report 11

Chapter 2: Theoretical Background and Previous Work 13

2.1 E-Learning and Digital Education 13

2.2 Artificial Intelligence in Education 14

2.3 Related Work and Existing Platforms 15

2.4 Chat Systems and Real-Time Technologies in Learning Platforms 16

Chapter 3: Methodology 17

3.1 Preface 17

3.1.1 Tools, Methods, and Programming Languages 17

3.1.2 Client Side 17

3.1.3 Database 18

3.2 System Features and Implementation 18

3.2.1 Login and Signup 18

3.2.2 Admin Dashboard and Management Features 24

3.2.3 Teacher Dashboard and Management Features 47

3.2.4 Student Dashboard and Management Features 86

3.3 Standards and Specifications 116

3.3.1 Authentication and Security Standards 116

3.3.2 API Design 116

3.3.3 UI/UX Standards 116

3.3.4 Real-Time Communication 116

3.3.5 Database Standards 117

3.3.6 AI Integration Guidelines 117

3.4 Constraints 117
1. Economic Constraints 117
2. Technological Constraints 117
3. User Experience Constraints 117
4. Social and Ethical Constraints 118
5. Health and Safety 118
6. Scalability and Sustainability 118
3.5 Standards and Specifications 118
1. Software Development Standards 118
2. User Interface Standards 119
3. Security Practices 119
4. Real-Time Communication Standards 119
5. AI Integration Guidelines 119
6. Database Design Standards 120
7. Collaboration and Version Control 120
Chapter 4: Results and Analysis 121
4.1 System Functionality Results 121
4.2 Performance Evaluation 122
4.3 User Feedback and Usability Insights 122
4.4 Key Achievements 122
Chapter 5: Discussion 123
5.1 Problem Resolution 123
5.2 Project Contributions 123
5.3 Implications 123
5.4 Limitations 123
Chapter 6: Conclusions and Recommendation 124
6.1 Conclusions 124
6.2 Recommendations 124
6.3 Future Work 124
Chapter 7: References 125

Table Of Figures

Figure 1 Welcome Page 22
Figure 2 Signup Page 22
Figure 3 Login Page 22
Figure 4 Forgot password process 23
Figure 4.2: invalid email 24
Figure 4.1:invalid password 24
Figure 4.3: weak password 24
Figure 5 Admin overview in web 27
Figure 6 Courses management screen 29
Figure 7 edit Teacher 30
Figure 8 Add New Course 31
Figure 9 Change Course Status 31
Figure 10 Delete Course 31
Figure 11 Pending Courses screen. 33
Figure 12 Manage teachers 35
Figure 13Students management 37
Figure 14Adding Students 39
Figure 15 Payments Screen 41
Figure 16 Platform Reports Screen 43
Figure 17 Platform Reports 44
Figure 18 Admin Settings 46
Figure 19 :Teacher Dashboard 48
Figure 19.1:Teacher Dashboard 49
Figure 19.2:Teacher Dashboard 50
Figure 20: Add Course Screen 52
Figure 21: My Courses Screen 52
Figure 22: add plan Screen 54
Figure 22.1: add plan Screen 55
Figure 23: Live Sessions Screen 57
Figure 23.1: Live Sessions Screen 58
Figure 24:Add Lesson Screen 60
Figure 25: View Lesson Screen 61
Figure 26: Update Lesson Screen 62
Figure 27: Delete Lesson 62
Figure 28: Messaging Screen 64
Figure 29: Student List Screen 65
Figure 30: Add Task Screen 67
Figure 31: view Task Screen 68
Figure 32: Update Task Screen 69
Figure 33: Delete Task Screen 69
Figure 34: Create Test Screen 71
Figure 35: Add Questions Screen 72
Figure 36: Edit Questions Screen 73
Figure 37 : Teacher Test Results Screen 74
Figure 38: Course Content Overview Screen 76

Figure 39 Course Certificates Screen....78
Figure 40 View Certificates Screen....79
Figure 41 Task Submissions Screen....80
Figure 42 Add Grades and feedback....81
Figure 43 Notifications Screen....82
Figure 44 Teacher Profile Screen....84
Figure 44.1: Settings....85
Figure 45 Course Enrollment Screen....87
Figure 46 Payment Interface Screen....87
Figure 47 Learning Style Selection Screen....88
Figure 48 Dashboard for Practical Learning....89
Figure 49 Dashboard for Visual Learning....91
Figure 50 Dashboard for Auditory Learning....91
Figure 51 Student Course Plan....92
Figure 52 Courses Overview....93
Figure 53 Course Payment....94
Figure 54:Student Lessons....95
Figure 55 Student Tasks....96
Figure 56 Edit or delete Task....96
Figure 57 Task Not Submitted....97
Figure 58 grade and feedback task....98
Figure 59 Summary Screen....99
Figure 60 AI Assistant Screen....100
Figure 61 Live Sessions Screen....101
Figure 62 Student Messaging System....102
Figure 63 Quiz Screen....103
Figure 64 View grades....104
Figure 65 Practical Learning AI Screen....105
Figure 66: Practical Learning AI Screen....106
Figure 67 Dynamic Quiz Screen....107
Figure 68 Auditory Listen Screen....108
Figure 69 Auditory Repeat Screen....109
Figure 70 YouTube Learning Screen....110
Figure 71 Mind Map Fun Screen....111
Figure 72 Daily Study Plan Screen....112
Figure 73 Student Certificates Screen....113
Figure 74 Student notifications....114
Figure 75 Student Profile Screen…....115

Chapter One: Introduction

1.1 General Background

In recent years, digital education has expanded rapidly due to technological advancements and the growing need for flexible learning environments. Traditional education methods alone are no longer sufficient, which has increased reliance on e-learning platforms that support online courses, virtual classrooms, assessments, and certifications. Students and teachers now require integrated systems that deliver complete educational experiences beyond basic content sharing.

Modern e-learning platforms must provide more than recorded lessons; they should include live online lectures, assignments, exams, progress tracking, and certification upon course completion. Additionally, instructors need efficient tools to manage courses, evaluate student performance, and communicate with learners. A major challenge in online education is addressing individual differences in learning styles, which can directly affect student engagement and learning outcomes.

Bright Path to Learning is an AI-powered e-learning platform designed to provide a complete and flexible digital learning experience. It integrates online courses, live lectures, assessments, and certification while supporting personalized learning through artificial intelligence. By combining modern technologies with a learner-centered approach, the platform enhances engagement and efficiency in contemporary education.

1.2 Problem Statement

In many online learning environments, students often face a one-size-fits-all approach, where courses, lectures, and assessments do not adapt to individual learning styles. This can lead to difficulties in comprehension, lower engagement, and reduced motivation. Teachers also lack tools to personalize content or provide real-time, data-driven feedback tailored to each student.

Existing platforms may offer course materials or communication features, but few support AI-driven summarization, quiz generation, or adaptive study plans that respond to the student's performance and preferred learning style.

These challenges highlight the need for **Bright Path to Learning**, an intelligent platform that personalizes content, enhances engagement, and provides actionable insights to optimize the learning experience for both students and educators.

1.3 Objectives of the Project

The primary goal of **Bright Path to Learning** is to create a fully functional AI-powered educational platform that provides personalized learning experiences for students while giving teachers and administrators full control over courses and academic activities.

Key objectives include:

- **Student Empowerment:** Students can access online courses with Zoom lectures, assignments, and quizzes. They can choose their preferred learning style, summarize content, generate quizzes, and study with AI-assisted personalized plans.
- **Teacher Support:** Teachers can upload course materials, manage assignments, and monitor student engagement while tailoring lessons to individual learning styles.
- **Administrative Control:** Administrators can manage all users and courses, approve or reject courses submitted by teachers, assign students to courses, track payments, and generate performance and activity reports. They can also reassign teachers to courses or remove courses entirely.
- **AI Integration:** The system leverages AI to provide automatic content summarization, question generation, and adaptive study recommendations, enhancing the learning process for every student.
- **Interactive Communication:** Real-time notifications and messaging facilitate seamless interaction between students, teachers, and administrators.

By achieving these objectives, **Bright Path to Learning** ensures a dynamic, intelligent, and fully manageable educational environment that adapts to the needs of students, teachers, and administrators alike.

1.4 Significance of the Project

In today's digital age, education is rapidly evolving, and traditional teaching methods often fail to meet the diverse needs of students. **Bright Path to Learning** addresses this challenge by offering a fully integrated AI-powered platform that personalizes learning, enhances engagement, and streamlines academic management for students, teachers, and administrators.

The project is significant in several ways:

- **Personalized Student Experience:** Students can select their preferred learning style, access lectures, complete quizzes and assignments, and leverage AI to summarize content and generate practice questions, leading to more effective and motivating learning.
- **Teacher Support:** Teachers can upload materials, manage courses, and monitor student progress while receiving AI-assisted insights to improve lesson quality and engagement.

- **Real-Time Communication and Notifications**: Using technologies like Firebase , the system ensures immediate feedback loops between teachers and students, fostering a more responsive and engaging academic environment.

- **Administrative Control and Efficiency:** Administrators have full oversight of courses, teachers, students, and payments. They can approve courses, assign students, reassign teachers, generate reports, and manage the platform efficiently.

- **Intelligent and Interactive Learning:** AI-driven features, real-time notifications, and seamless communication ensure adaptive learning paths, timely feedback, and active participation across the platform.

- **Market Relevance:** Bright Path to Learning meets the rising demand for personalized, AI-powered education and can grow to support more students, learning styles, and smarter tools.

By combining personalized learning, AI integration, and comprehensive administrative control, **Bright Path to Learning** delivers a modern, flexible, and effective educational solution capable of meeting the evolving demands of digital classrooms.

1.5 Organization of the Report

This report presents a clear and structured overview of the Bright Path to Learning platform, guiding the reader through every stage of its development, from initial research and design to implementation and evaluation.

- **Chapter 1: Introduction**

  Provides an overview of Bright Path to Learning, its goals, significance, and the organization of the report.

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

  Reviews relevant literature, technologies, and existing educational systems, forming the foundation for the platform.

- **Chapter 3: Methodology**

  Describes the development process, including design choices, tools, frameworks, constraints, and ethical considerations.

- **Chapter 4: Results and Analysis**

  Presents outcomes, evaluates system performance, and discusses user experience and functionality.

- **Chapter 5: Discussion**

  Interprets results, examines achievements against objectives, and highlights strengths and limitations.

- **Chapter 6: Conclusions and Recommendations**

  Summarizes findings, reflects on lessons learned, and suggests directions for future enhancements.

- **References and Appendices**:

  Includes cited sources, diagrams, code snippets, and supplementary data supporting the report.

This structured format ensures clarity and a smooth flow, guiding the reader through the project from concept to completion.

Chapter 2: Theoretical Background and Previous Work

2.1 E-Learning and Digital Education

The rapid advancement of technology has transformed education, moving beyond traditional classrooms into flexible, interactive, and learner-centered digital environments. Modern students demand more than static lessons; they require platforms that adapt to their learning style, pace, and engagement needs. This shift has prompted the development of educational systems that are not only content-rich but also intelligent, responsive, and tailored to individual learners.

E-learning platforms have grown significantly, particularly accelerated by global events that necessitated remote learning. Traditional online education often provides recorded lectures, assignments, and discussion boards, yet many platforms still lack personalization, real-time feedback, and interactive engagement. To truly enhance learning outcomes, digital platforms must integrate adaptive methods that respond dynamically to student behavior and preferences.

Bright Path to Learning addresses this need by providing a comprehensive digital learning ecosystem. The platform allows students to select their preferred learning style—visual, auditory, or kinesthetic—and automatically adapts lesson content accordingly. Lessons can be converted into summaries, flashcards, interactive exercises, or quizzes, leveraging AI-powered content generation to enhance understanding and retention.

2.2 Artificial Intelligence in Education

Bright Path to Learning leverages Artificial Intelligence to deliver a **practical and fully personalized learning experience**. Students can input any topic, and the AI instantly generates summaries, creates quizzes, and even searches for relevant instructional videos from platforms like YouTube. This ensures that learners not only understand the content but can also interact with it in a way that matches their preferred learning style—visual, auditory, or kinesthetic.

Beyond content delivery, the platform includes a **smart AI assistant** that answers any student question in real time, providing explanations, guidance, or resources. For practical learning, students can enter a topic, and AI outlines step-by-step procedures to apply the knowledge or complete tasks, bridging the gap between theory and hands-on practice.

By combining **automated summarization, question generation, guided research, and practical step-by-step instruction**, AI in Bright Path to Learning enhances comprehension, engagement, and skill development. It empowers students to learn efficiently while giving teachers actionable insights on student progress and learning patterns.

2.3 Related Work and Existing Platforms

Over the last decade, numerous digital learning platforms have emerged, including Google Classroom, Edmodo, Khan Academy, Coursera, and Moodle. These platforms have significantly improved access to education by enabling remote and blended learning, providing structured course materials, video lectures, quizzes, and discussion forums.

However, most of these systems rely on **static content** and offer **limited interactivity**. While they support quizzes, video playback, or forums, they rarely adapt content dynamically to match individual student preferences, learning pace, or practical application needs. AI features, when present—as in Duolingo or Socratic—tend to focus on narrow tasks like language learning or homework assistance, rather than providing a holistic, flexible learning experience.

Bright Path to Learning distinguishes itself by combining **AI-driven content transformation** with a fully **practical learning approach**. Students can input any topic, and the AI instantly generates summaries, quizzes, step-by-step practical instructions, and even relevant YouTube resources. The platform includes a **smart AI assistant** for real-time question answering and guidance, alongside Firebase-powered messaging for interactive communication.

Additionally, Bright Path provides a **comprehensive administrative layer**, allowing admins to manage students, courses, teachers, approvals, and payments, offering a fully integrated educational ecosystem. Unlike many existing platforms, it seamlessly merges **personalized content, practical application, and real-time interaction**, creating a dynamic and intelligent learning environment tailored to both students and educators.

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

In modern educational platforms, **real-time communication and notifications** are crucial for creating interactive, engaging, and collaborative learning experiences. Traditional e-learning platforms often rely on asynchronous communication, such as emails, comments, or forums, which can introduce delays and reduce student motivation. Students may have questions about lessons, assignments, or practical exercises but must wait for responses, slowing down the learning process. Similarly, teachers may not be immediately aware of student submissions or difficulties, limiting the effectiveness of their support.

To overcome these challenges, **Bright Path to Learning** integrates real-time communication features using **Firebase**, a cloud-hosted platform that simplifies messaging and notifications. With Firebase, students and teachers can **chat instantly**, ask questions, share resources, and clarify doubts as if they were interacting in a physical classroom. This **immediacy enhances engagement**, promotes continuous feedback, and supports a more social and collaborative learning environment.

Moreover, Firebase is used to implement a **real-time notification system**, which automatically alerts users to important events, such as **new lesson uploads, assignment deadlines, test schedules, or AI-generated summaries and exercises**. Students receive immediate updates without having to refresh the app manually, while teachers stay informed about student activity and progress. This ensures that everyone remains **synchronized and informed**, which is particularly important in courses that include **live Zoom sessions, practical exercises, and AI-assisted learning tools**.

The integration of Firebase not only simplifies technical implementation but also provides flexibility for **scalable, cross-platform support**, covering mobile and web users. It allows notifications to be delivered both **inside the app** (in-app messages) and **outside the app** (push notifications), ensuring students and teachers never miss crucial updates. By combining **instant messaging with automated notifications**, Bright Path to Learning transforms the learning experience, fostering **responsiveness, accountability, and continuous interaction**.

In summary, the use of **Firebase real-time technologies** in Bright Path to Learning bridges the gap between remote learners and instructors, enables **immediate support and feedback**, and creates a **dynamic, connected educational ecosystem**. This approach encourages students to actively participate in lessons, complete tasks on time, and engage with AI-powered study tools, while giving teachers the ability to **monitor progress, respond promptly, and guide learning effectively**.

Chapter 3: Methodology

This chapter describes the approach and methods used in the design, development, and deployment of the Bright Path to Learning platform. It covers the planning stages, technology selection, system architecture, and implementation techniques, all tailored to create an interactive, AI-assisted, and user-centered learning environment.

3.1 Preface

3.1.1 Tools, Methods, and Programming Languages

The development of Bright Path to Learning combined modern programming tools and mobile-focused frameworks to create an engaging and intelligent educational platform. The development environment included:

- **Visual Studio Code:** Main IDE for writing and organizing code efficiently.
- **Expo:** Used to run and test the mobile version on both Android and iOS devices seamlessly.
- **React Native:** The core framework for cross-platform mobile development.
- **Firebase:** For real-time chat, notifications, cloud database hosting, and authentication.
- **Postman**: API testing and debugging.
- **Git & GitHub:** Version control and team collaboration.
- **Stripe SDK:** For secure in-app payments and subscription handling.
- **JWT (JSON Web Tokens):** Securing user authentication and role-based access.
- **Assembly AI:** For transcribing audio from lesson videos and practical sessions.
- **AI Tools:** To generate summaries, quizzes, practical guides, and intelligent suggestions for students.

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

3.1.2 Client Side

The client side of **Bright Path to Learning** was developed to provide a smooth and interactive experience across both mobile and web platforms. The design focuses on usability, responsiveness, and clear role separation to ensure that each user interacts only with the features relevant to their role.

The **mobile application** was built using **React Native** with **JavaScript** as the primary programming language and is run and tested using **Expo**, allowing efficient cross-platform development and real-time testing on mobile devices. In addition, **Java and Kotlin** were used for specific Android-related components to enhance performance and system integration. The **web interface**, mainly used by administrators, was developed using JavaScript-based technologies and accessed directly through browsers.

The client side supports role-based dashboards for Admin, Teacher, and Student, interactive course and lesson access, AI-powered learning tools, multiple learning styles (visual, auditory, and practical), and real-time chat using Firebase. This structure ensures a flexible, user-friendly, and adaptive learning experience.

3.1.3 Database

The backend of the Bright Path platform was developed using **Node.js** and **Express.js**, providing a reliable and scalable server-side environment. It manages core functionalities such as user authentication, course and lesson management, assignments, assessments, notifications, and certificates. Secure access is enforced using **JWT-based authentication** with role-based authorization.

The system uses **MySQL** as its primary relational database to store structured data related to users, courses, enrollments, lessons, tasks, tests, submissions, messages, and notifications. Database relationships are defined using foreign keys to ensure data integrity, while indexing is applied to frequently queried fields to maintain efficient performance.

3.2 System Features and Implementation

This section presents the main features of the Bright Path platform and their implementation for each user role: **Admin, Teacher, and Student**. Each role has a dedicated dashboard with role-specific functions. The backend is built with **Node.js and Express.js**, while the frontend uses **React (web) and Expo/React Native (mobile)** for cross-platform support. **JWT authentication** secures user access, and **Firebase** is used for real-time chat between users. Notifications are handled via the backend, and AI features are integrated for lesson summaries and content suggestions.

3.2.1 Login and Signup

Bright Path uses **JWT authentication** to secure login and signup. Each user (Admin, Teacher, Student) is directed to a **role-specific dashboard** after login, with a **welcome page** guiding them to their features. The frontend (React/Expo) handles the interface, while the backend (Node.js/Express) manages authentication and data flow.
The corresponding welcome page is shown in Figure 1

Signup Process

The system allows users to sign up directly as Students or Teachers through a single interface, selecting their role during account creation and providing basic information (name, email, password). Input validation is performed on both the client and server sides to ensure security and data integrity. Admins and account management are handled exclusively through the Admin Dashboard to maintain full control and oversight (see Figure 2 Signup Page) .

Login Process

The Bright Learning login screen allows students and teachers to access their accounts by entering email and password. The backend verifies credentials, returns a JWT token, and the app stores it securely for future API calls. User data is saved for personalized navigation, directing students to course selection and teachers to their dashboard. The interface features intuitive input fields, clear icons, and responsive buttons, ensuring a smooth and engaging login experience. Figure 3 Login Page

Security and Authorization

In BrightPath, security is a top priority. All passwords are hashed using **bcrypt** before being stored in the database, ensuring that sensitive information is never kept in plain text. The backend enforces role-based access control through a custom authentication middleware, allowing each user to access only what they are authorized to. Admins have full control over the platform, managing courses, users, and system settings, while teachers can access and manage their own courses, lessons, and enrolled students. Students are restricted to viewing and interacting only with the courses they are enrolled in, including assignments and materials. This system guarantees a secure, organized, and efficient environment for all users.

Password Recovery

The password recovery feature in BrightPath 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

The BrightPath Forgot Password screen provides a simple and secure way for users to recover access to their accounts. It displays a clear title and a short instruction prompting the user to enter their registered email address. Once the user submits their email, the system communicates with the backend to verify if an account exists for the provided address. If a matching account is found, the backend generates a unique, time-limited verification code and sends it directly to the user's email. Users receive immediate feedback: a confirmation message is shown if the code is successfully sent, guiding them to the next step where they can verify the code and reset their password, while any errors, such as an unregistered email or server issue, are clearly indicated with an appropriate alert. This flow ensures both ease of use and security, keeping the recovery process seamless and reliable.

2. Reset Password Screen

The Reset Password screen in BrightPath is designed to provide a secure and user-friendly process for setting a new password after receiving a verification code via email. The key features and flow are as follows:

- **Clean and Focused Interface:** The screen displays a clear title "Reset Password" along with instructional text that guides the user to enter a new password. The layout is simple, ensuring the user focuses on the task without distractions.

- **Input Fields:**
    - **New Password:** Users enter their desired password in a dedicated field. Basic password validation can be applied to ensure minimum length or strength requirements.
    - **Confirm Password:** Users re-enter the password to confirm it. The system validates that both entries match before submission.

- **Submission Process:**
    - When the user taps the "Save" button, the app sends the email, the verification code received via the previous step, and the new password to the backend API (/reset-password).
    - The backend performs several checks: it verifies the reset code, ensures it has not expired, hashes the new password securely using bcrypt, updates the user's password in the database, and invalidates the used reset code to prevent reuse.

- **User Feedback:**
    - **Success:** If all validations pass and the password is updated successfully, a success message is shown and the user is redirected to the login screen.
    - **Failure:** If there is any issue, such as a mismatch in password confirmation, an invalid or expired code, or server errors, a clear error message is displayed to the user.

- **Navigation Options:** The screen provides a smooth transition back to the login screen, ensuring users can quickly attempt login after resetting their password or abandon the reset process if needed.

This screen ensures a secure, reliable, and intuitive experience for users, giving them confidence that their account recovery process is handled safely and efficiently.

Role-Based Routing

In Bright, once a user successfully logs in, the system inspects the role stored in the JWT token and automatically navigates the user to the appropriate dashboard, ensuring a personalized experience. **Admins** are redirected to the Admin Dashboard to manage platform content and users, **teachers** are taken to their Teacher Dashboard to access their courses, lessons, and student information, and **students** are directed to the Student Dashboard where they can view enrolled courses and track progress.

Technologies Used:

- **Frontend:** React Native for building cross-platform mobile interfaces.
- **Backend:** Node.js with Express for handling authentication, data, and API endpoints.
- **Database:** MySQL for storing user credentials, roles, and related data.
- **Security:** JWT for authentication, bcrypt for password hashing, and AsyncStorage for secure local storage of tokens and user information.

Figures

Figure 1 Welcome Page

Figure 2 : Signup Page

Figure 3: Login Page

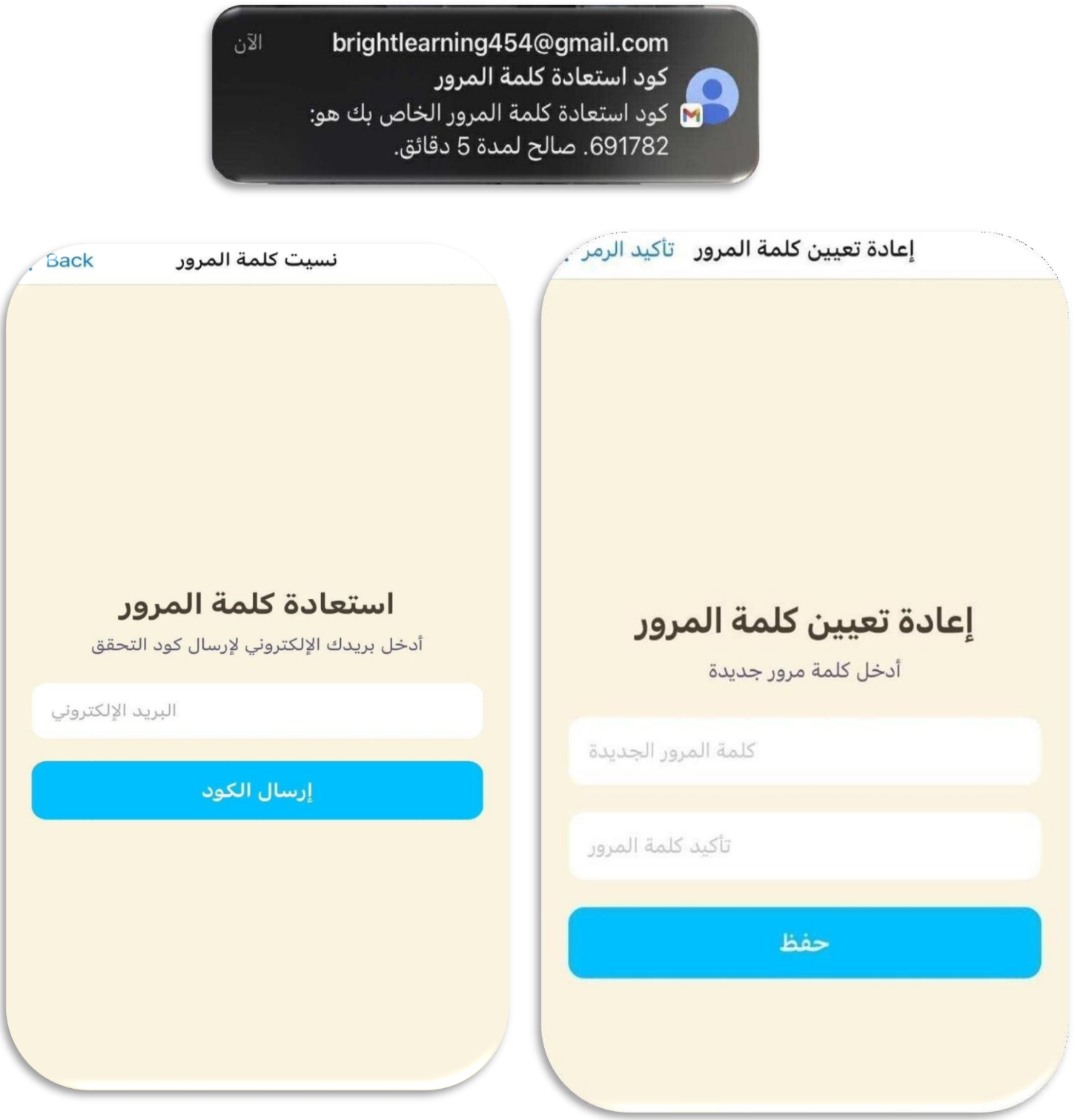


Figure 4 :Forgot password process

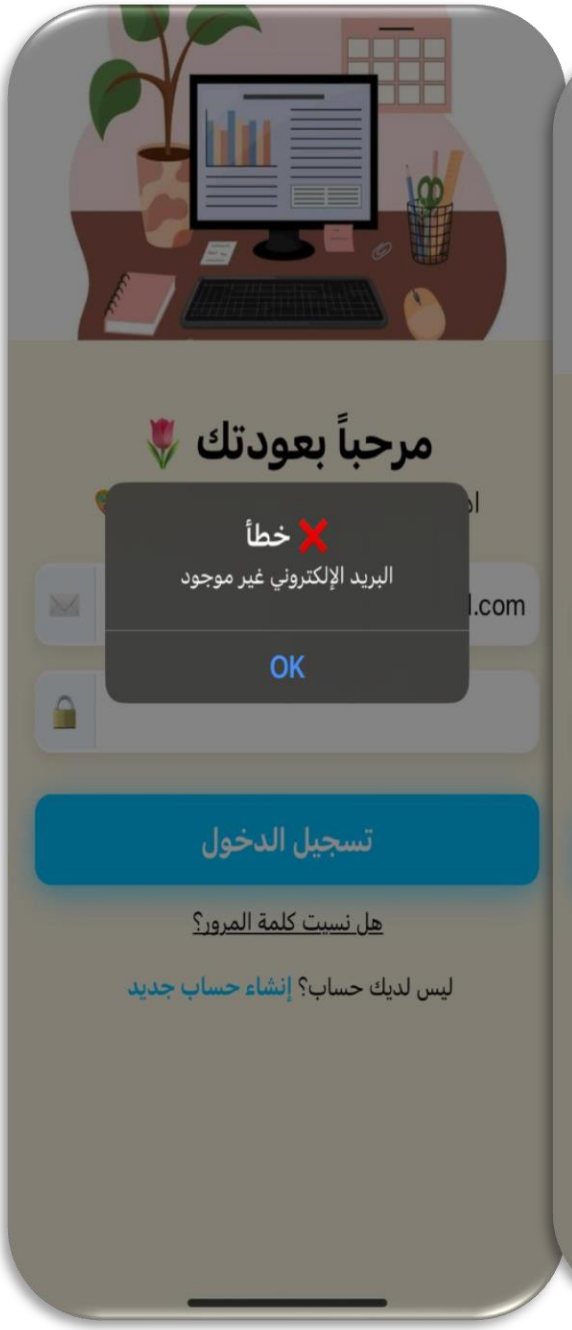


Figure 4.2: invalid email

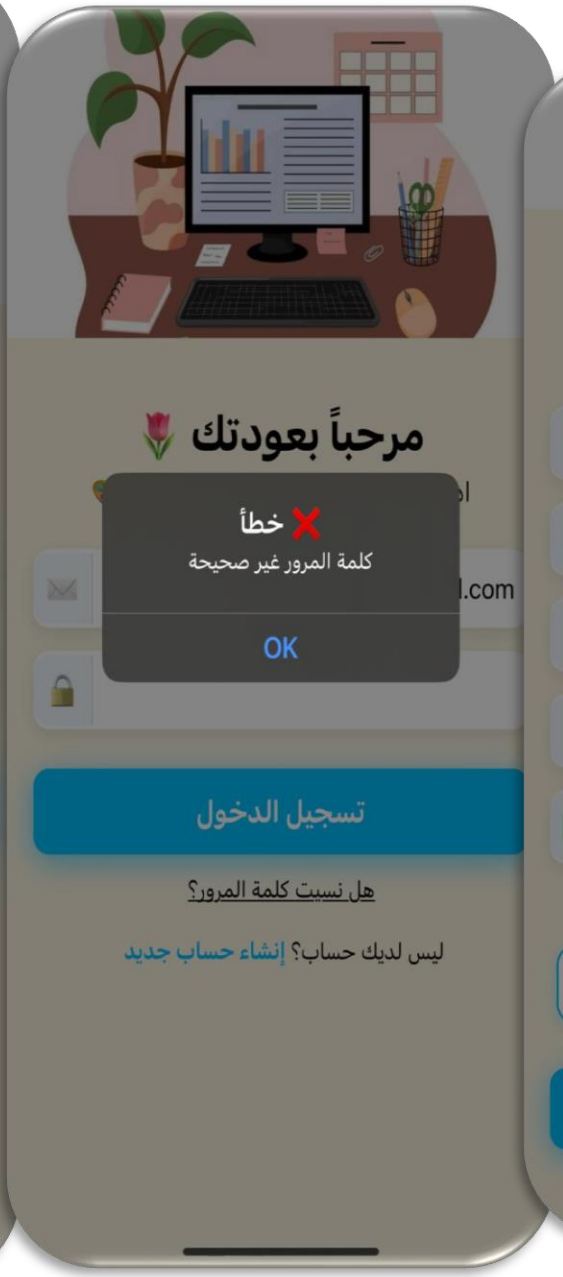


Figure 4.1:invalid password

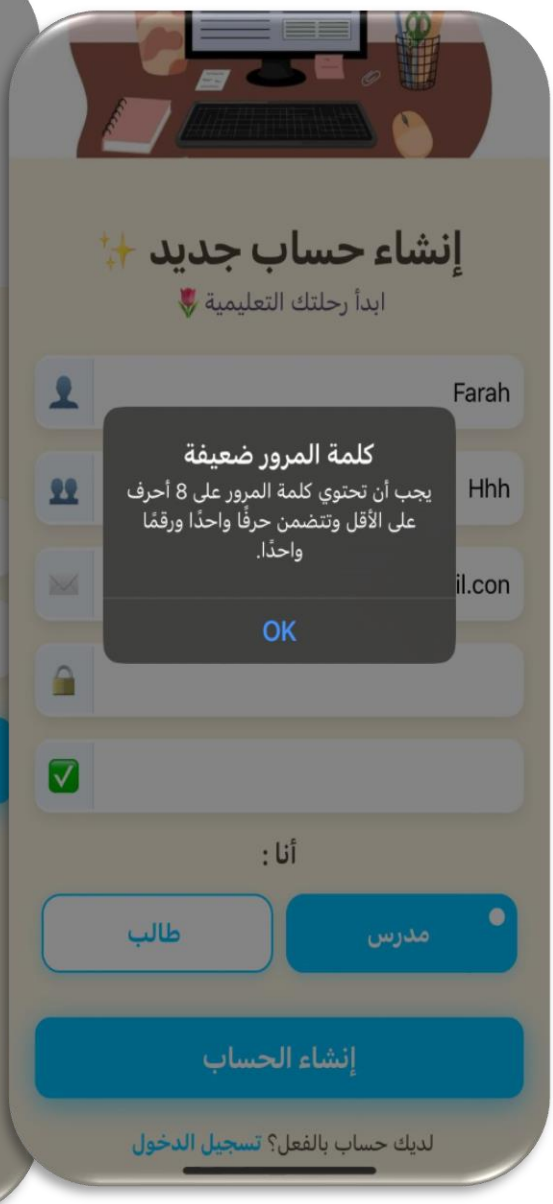


Figure 4.3: weak password

3.2.2 Admin Dashboard and Management Features

Bright implements a **role-based dashboard system** to provide a tailored user experience for each core role: Admin, Teacher, and Student. Each dashboard aligns with the user's responsibilities and access level, ensuring both usability and security.

Admin Dashboard

The **Admin Dashboard** is the central control panel for managing Bright's educational platform. It equips administrators with the tools to monitor platform operations, manage users (teachers and students), curate courses, track payments, and oversee overall activity. The interface is fully **responsive**, adapting seamlessly for both desktop/web and mobile devices. Navigation is provided via a **side drawer** on larger screens and a **bottom navigation bar** on smaller devices.

Overview Screen

This screen provides the administrator with an immediate, high-level summary of the platform's current state and key metrics. It shows the total number of students, teachers, courses, pending courses, and rejected courses in a clear and visual layout.

Key Features

1. **Statistical Summary:**
   The dashboard provides a **real-time overview** of vital metrics, including:
   - Total students
   - Total teachers
   - Total courses
   - Pending course approvals
   - Rejected courses

   Each metric is displayed in visually distinct **gradient cards** for quick comprehension, with relevant icons for clarity.

2. **Quick Actions:**
   Administrators can perform common tasks directly from the dashboard, including:
   - Adding a new student to a course
   - Reviewing pending courses
   - Viewing payment statuses

   These actions open modals or navigate to relevant sections for efficient workflow.

3. **Recent Activity & Notifications:**
   Admins can monitor recent platform activities, such as new teacher registrations, student

enrollments, and course approvals or rejections. Notifications are **real-time**, with unread badges to alert the admin.

4. **Quick Assignment Modal:**
   The dashboard includes a **modal system** that allows admins to quickly assign students to courses. Using **dropdown selectors**, admins can choose a student and a course, then confirm the assignment. The modal dynamically updates the statistics and available options after each action.
5. **Navigation & Responsiveness:**
   - **Side drawer:** Accessible on larger screens for full menu options.
   - **Bottom navigation:** Optimized for mobile users with shortcuts to Home, Students, Teachers, and Logout.
   - **Animated transitions:** Smooth UI interactions using React Native Animatable and Animated API.

Admin Dashboard:

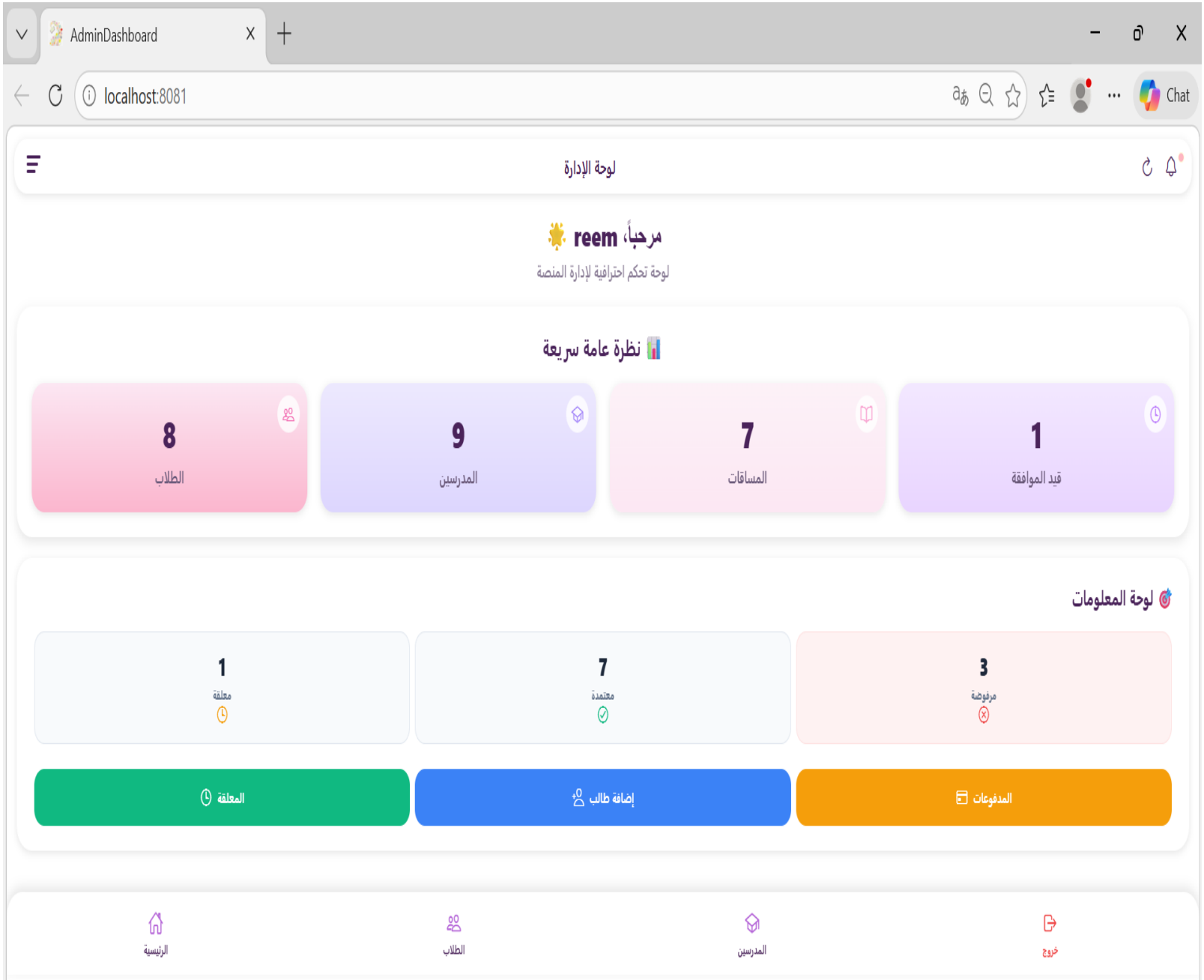


Figure 5: Admin overview in web

2. Course Screen

The Course Management interface serves as a central control panel for administrators to manage all courses within the BrightPath platform. This interface presents courses in a structured card-based layout, where each course displays essential information including the course title, assigned instructor, brief description, price, number of enrolled students, and current approval status. This organized presentation allows administrators to quickly assess the overall state of the course catalog.

At the top of the screen, summary statistics provide real-time insights into the system, such as the total number of courses, the number of approved courses, and overall student enrollment. These indicators support informed decision-making and efficient system monitoring.

The interface enables administrators to add new courses through a dedicated form that validates input data to ensure accuracy and consistency. Additionally, administrators can reassign instructors to courses when needed, allowing flexibility in course management without disrupting existing content or student enrollment.
Approval and status management features allow administrators to review courses before they become visible to students. Courses can be approved or rejected through a controlled process, ensuring content quality and platform standards. Furthermore, the interface includes secure deletion options with confirmation prompts to prevent accidental data loss.

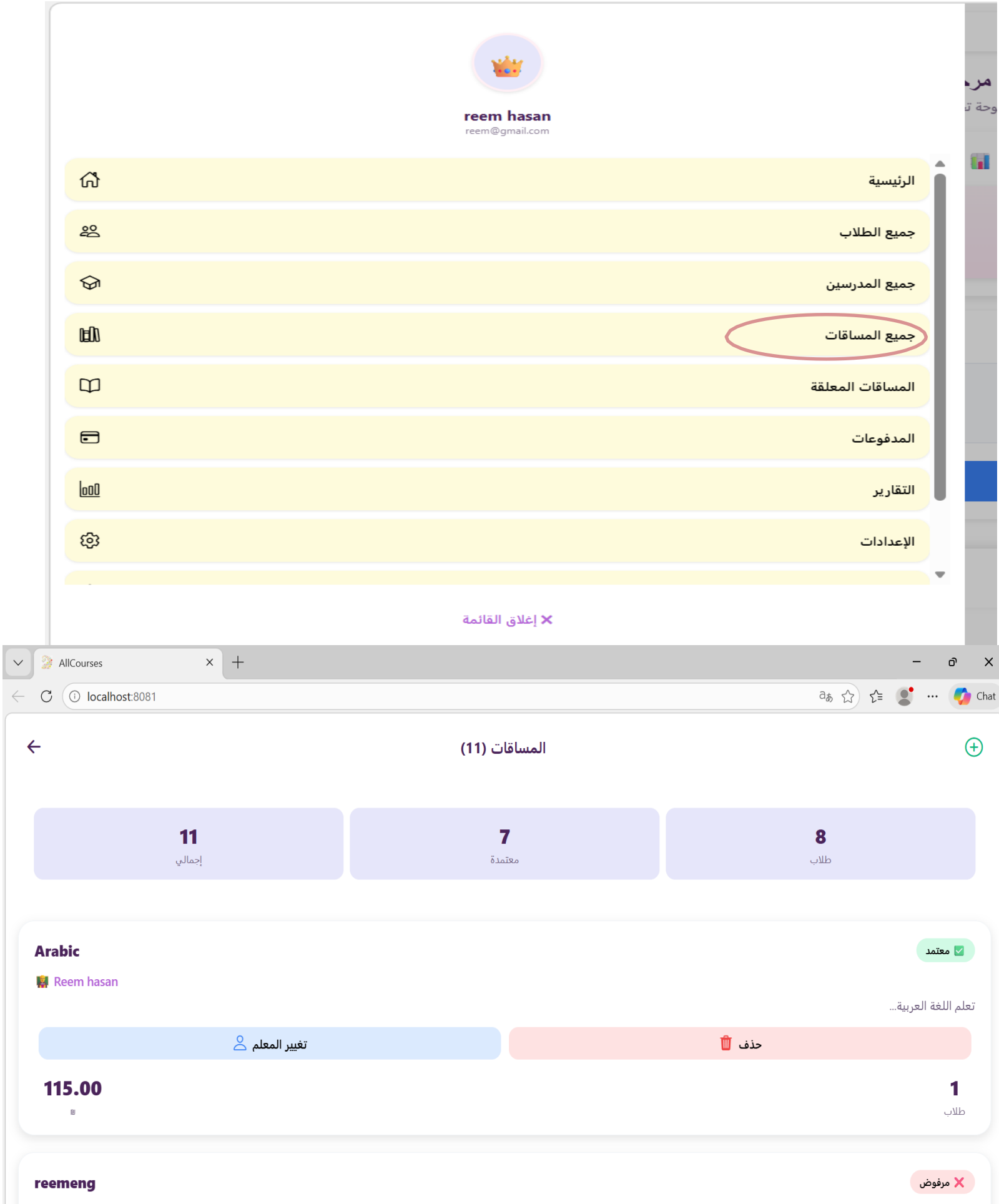


Figure 6: Courses management screen

Figure 7: edit Teacher

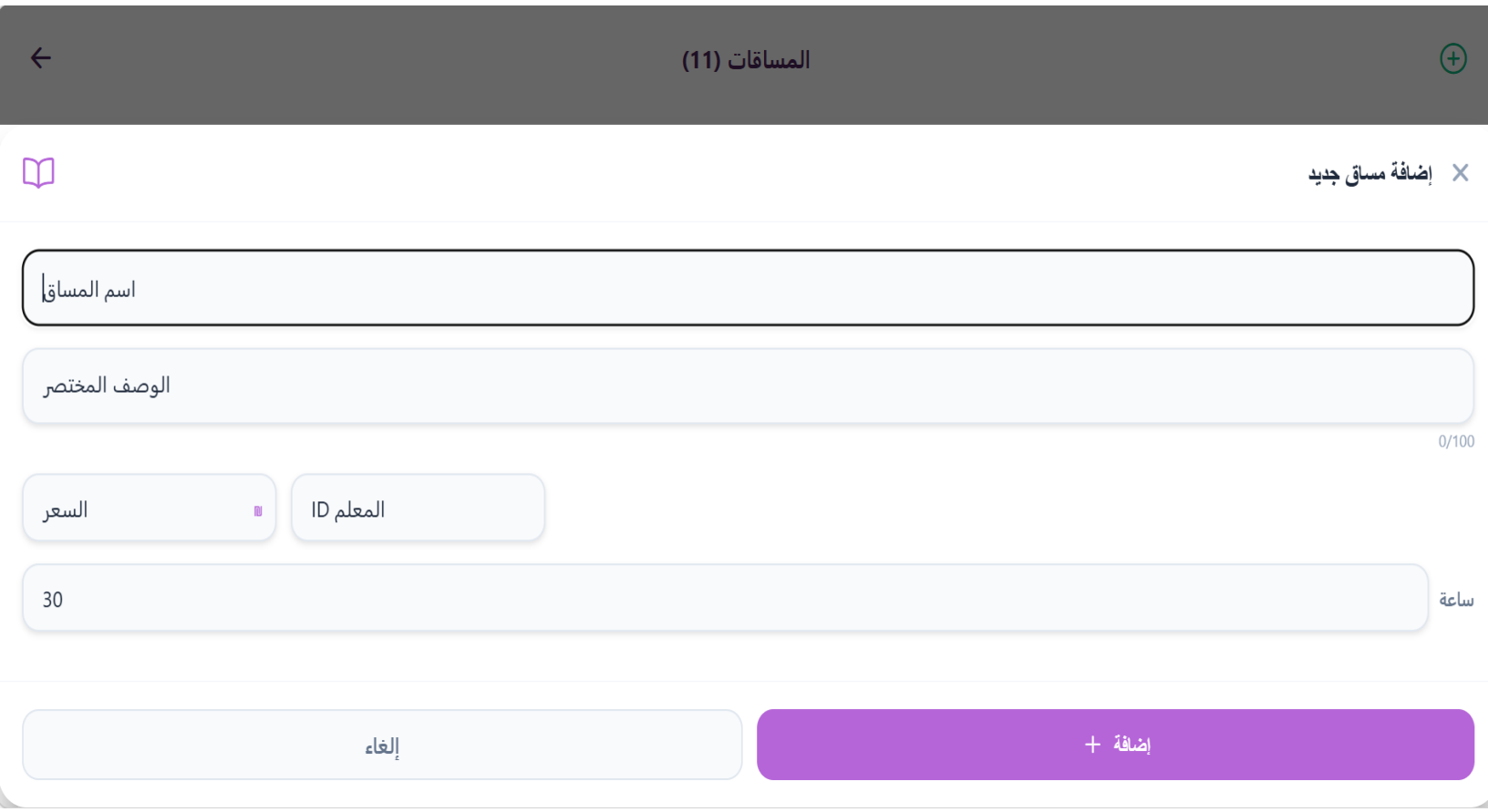


Figure 8 : Add New Course

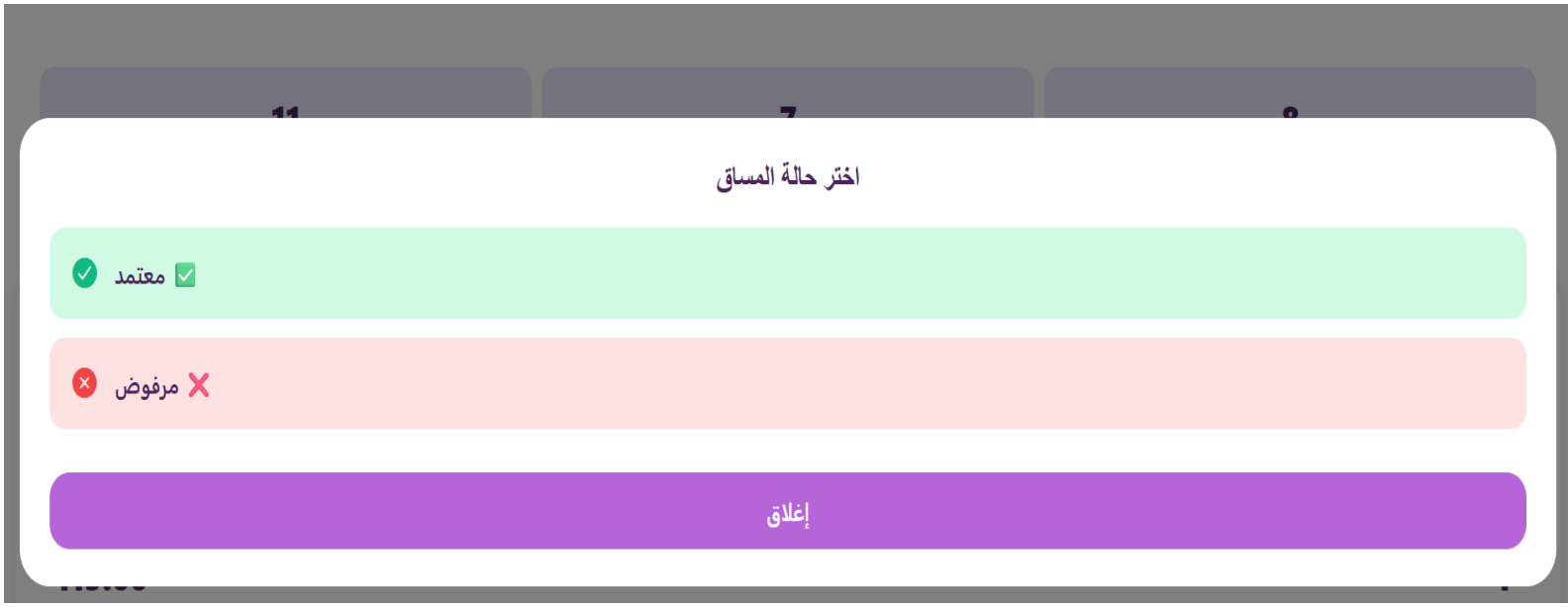


Figure 9 : Change Course Status Modal

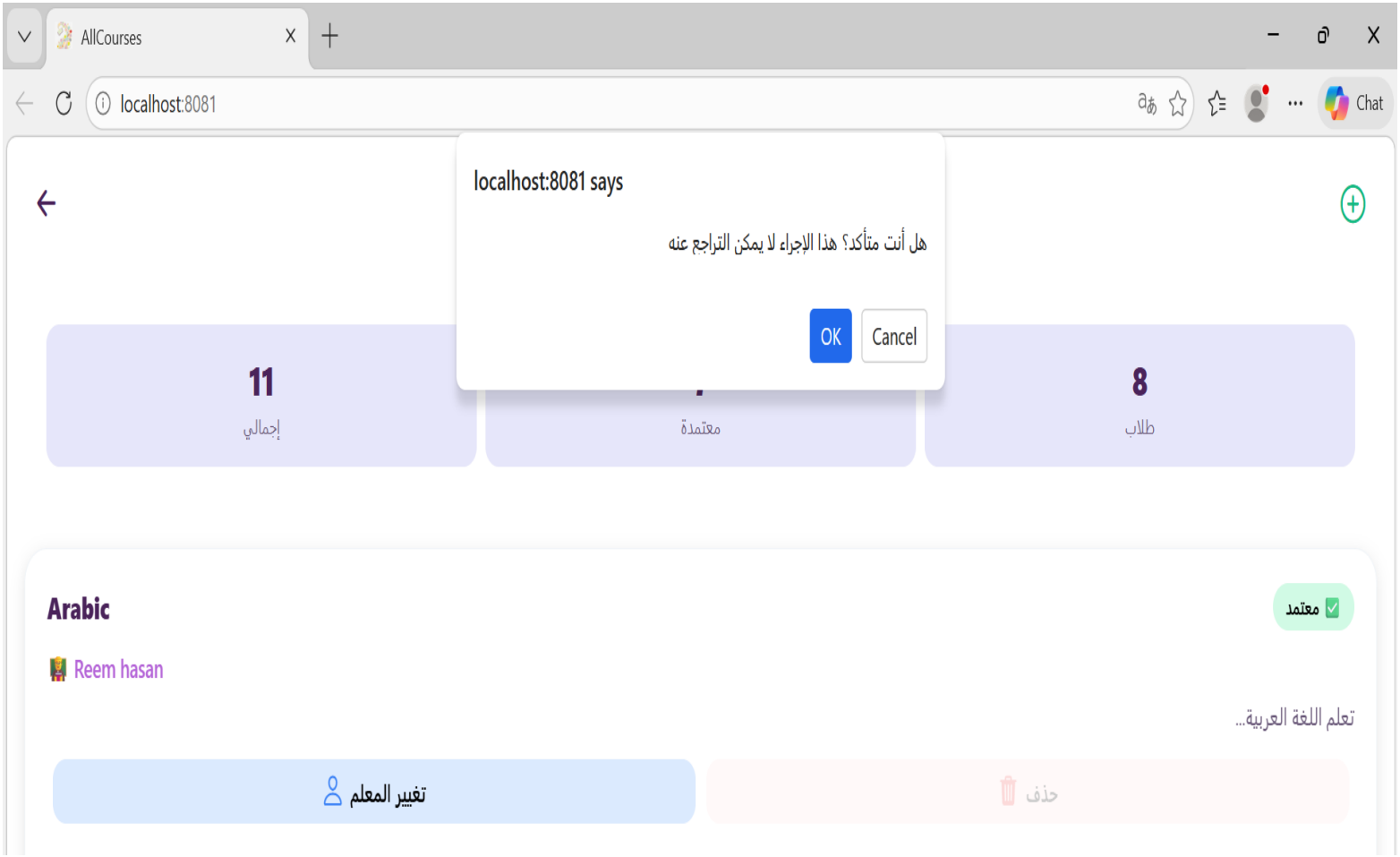


Figure 10 Delete Course.

3. Pending Courses Screen

This screen allows administrators to review all courses that are awaiting approval before they become visible to students. Admins can approve or reject courses directly from this list.

Key Features:

- **Course Listing:** Displays all pending courses with their name, code, assigned teacher, description, and price.
- **Approve / Reject Actions:** Each course has buttons to quickly approve or reject it. Confirmation is required before taking action.
- **Refresh Button:** Updates the list of pending courses.
- **Empty State:** If no courses are pending, a message and icon inform the admin that there are currently no courses to review.
- **Error Handling:** Displays an error message if fetching the pending courses fails, with an option to retry.

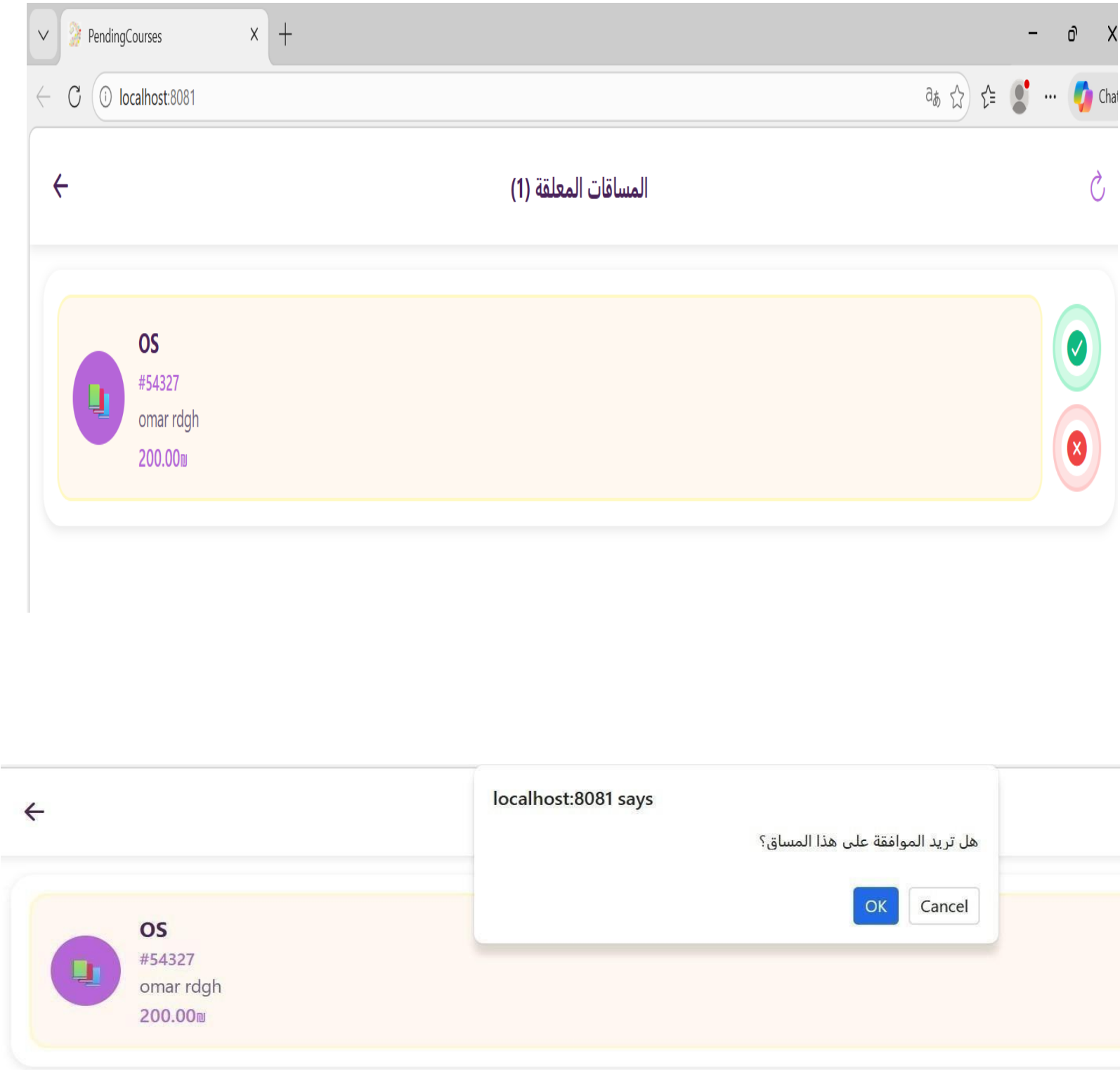


Figure 11: Subject Details screen.

4. All Teachers Screen

The **All Teachers Screen** is designed to provide administrators with full control over teacher accounts in the system. The main goal of this screen is to allow the admin to view, add, and delete teachers, as well as manage their basic information in a simple and user-friendly way.

At the top of the screen, there is a **header** section. This header includes a back button that allows the admin to return to the previous page, a title showing “All Teachers” along with the total number of teachers currently registered, and an add button represented by a + icon. By pressing the add button, the admin can open a modal window to add a new teacher to the system. This layout makes navigation straightforward and keeps the primary actions within easy reach.

The **teacher listing** occupies the main part of the screen. Each teacher is displayed as a card containing a small circular avatar with the teacher's initials, the full name, and the email address. Pressing on a teacher card will display a pop-up with detailed information including the teacher's full name, email, and ID. This allows the admin to quickly check teacher details without leaving the screen. Next to each teacher card, there is a delete button with a clear visual design. Pressing this button triggers a confirmation dialog to prevent accidental deletion. If the admin confirms, the teacher is deleted from the system using an authorized admin token, and the list automatically refreshes to reflect the changes.

To ensure the data is always up to date, the screen includes a **refresh functionality**. The admin can tap the refresh icon to reload the teacher list from the server. In case there is an error while fetching the data, the screen displays a red error message along with a retry button, allowing the admin to attempt the operation again.

If no teachers are registered, the screen shows a specially designed empty state. This empty state features a school icon, a message indicating that no teachers are available, and a button to add the first teacher. This approach guides the admin on what action to take next and maintains a clean, organized interface even when the data is empty.

Finally, the **add teacher modal** is a separate dialog where the admin can input the first name, last name, email, and password of a new teacher. Each input field has a clear placeholder to guide the admin, and the add button checks that all fields are filled before sending the data to the server. While the teacher is being added, a spinner is displayed to indicate loading. If the operation succeeds, a success alert appears and the teacher list updates automatically; if it fails, an error message is shown. This workflow ensures that adding new teachers is fast, secure, and visually intuitive.

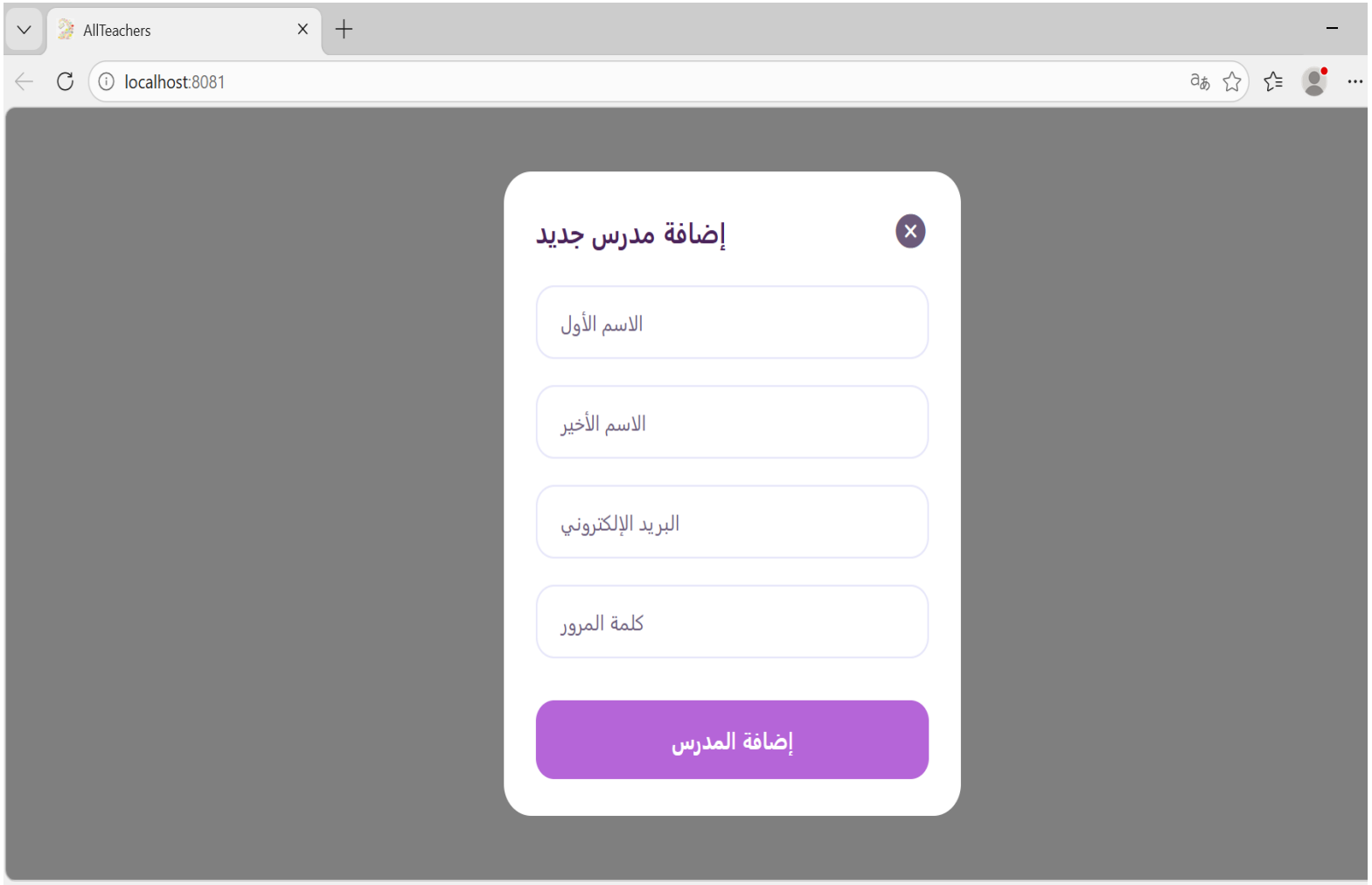


Figure 12: Manage teachers

5. Students Screen

The **Students Screen** is designed for administrators to manage student accounts on the platform. It provides a clear overview of all students, allows adding new students, viewing details, and deleting accounts.

Viewing Students

When the screen loads, a list of all users with the `student` role is displayed. Each student item shows their **full name** and **email address**, and can be tapped to see more details or edit their profile. A **search bar** at the top allows admins to quickly find a specific student by name or email.

Adding a New Student

Administrators can add a new student using the **floating "Add" button** in the top-right corner. Clicking this button opens a modal where the admin can input:

- First Name
- Last Name
- Email
- Password

The "Grade" field is removed in this process for simplicity. After filling in all required fields and pressing **Add Student**, the system sends a request to the server to create the new student. Upon successful addition, a confirmation alert appears, the modal closes, and the student list refreshes automatically.

Deleting a Student

Each student card has a **delete button** displayed prominently. When pressed, a confirmation dialog appears to prevent accidental deletion. Deleting a student also removes:

- Any **active enrollments**
- Any **pending enrollment requests**

This ensures that all related data is cleaned up and the platform remains consistent.

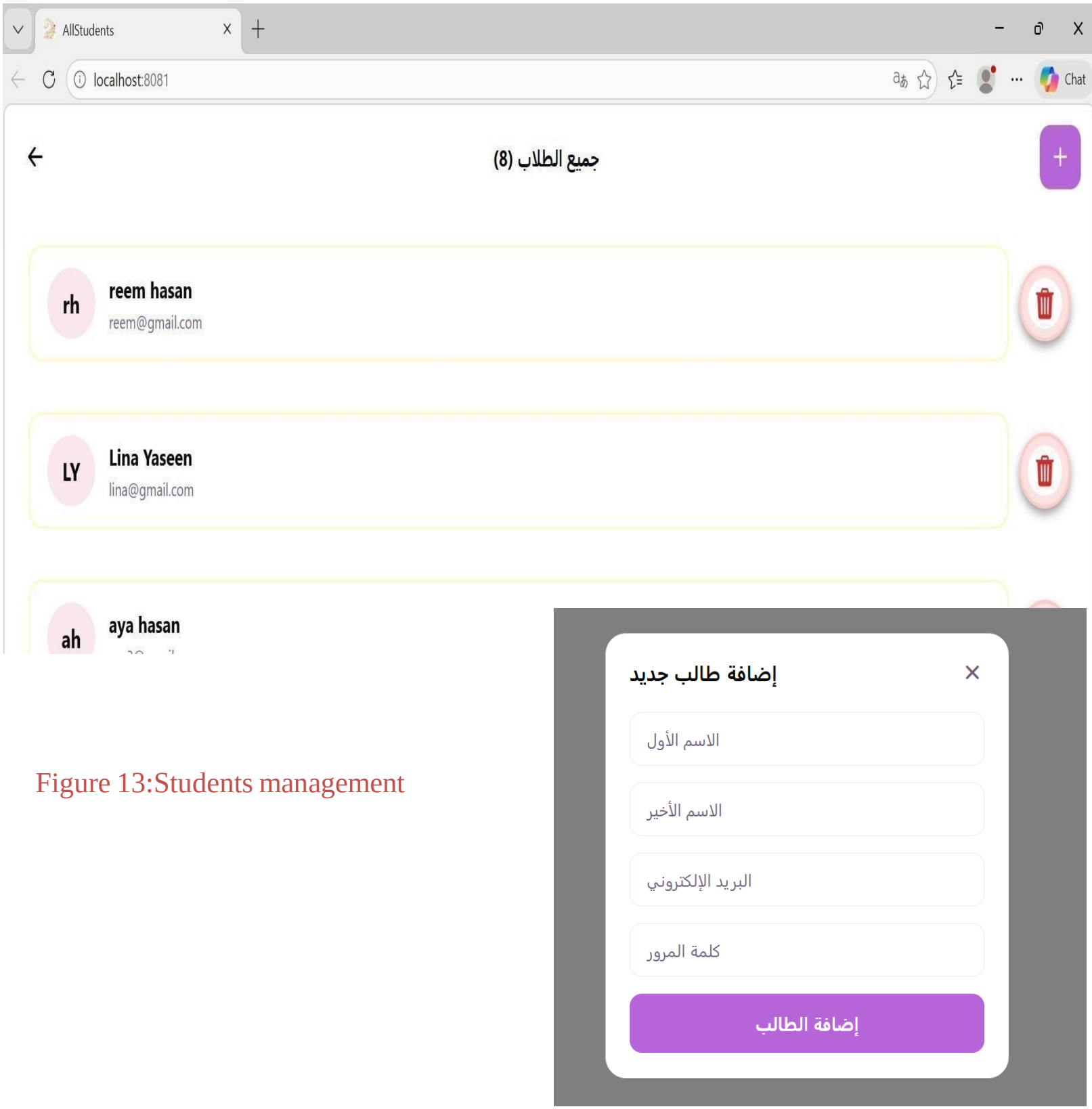


Figure 13:Students management

6. Student Assigning Students to Courses (Admin Dashboard)

The **Admin Dashboard** provides administrators with direct control over student enrollments through a quick and efficient assignment feature. From this interface, the admin can manually add any student to any existing course without relying on the student's own enrollment request or the standard registration flow.

Through the **Quick Actions** section, the administrator can open a dedicated modal titled *"Add Student to Course"*. This modal presents two dropdown menus: one for selecting a student and another for selecting a course. The student list displays registered students on the platform, while the course list includes available and approved courses. This design allows the admin to quickly locate and assign users without navigating away from the dashboard.

Once both a student and a course are selected, the admin confirms the action, and the system immediately enrolls the student in the chosen course. The assignment is processed securely through an authenticated backend request using the admin's token, ensuring that only authorized administrators can perform this operation. After a successful assignment, the system updates the platform statistics automatically, reflecting the change in total enrollments and courses.

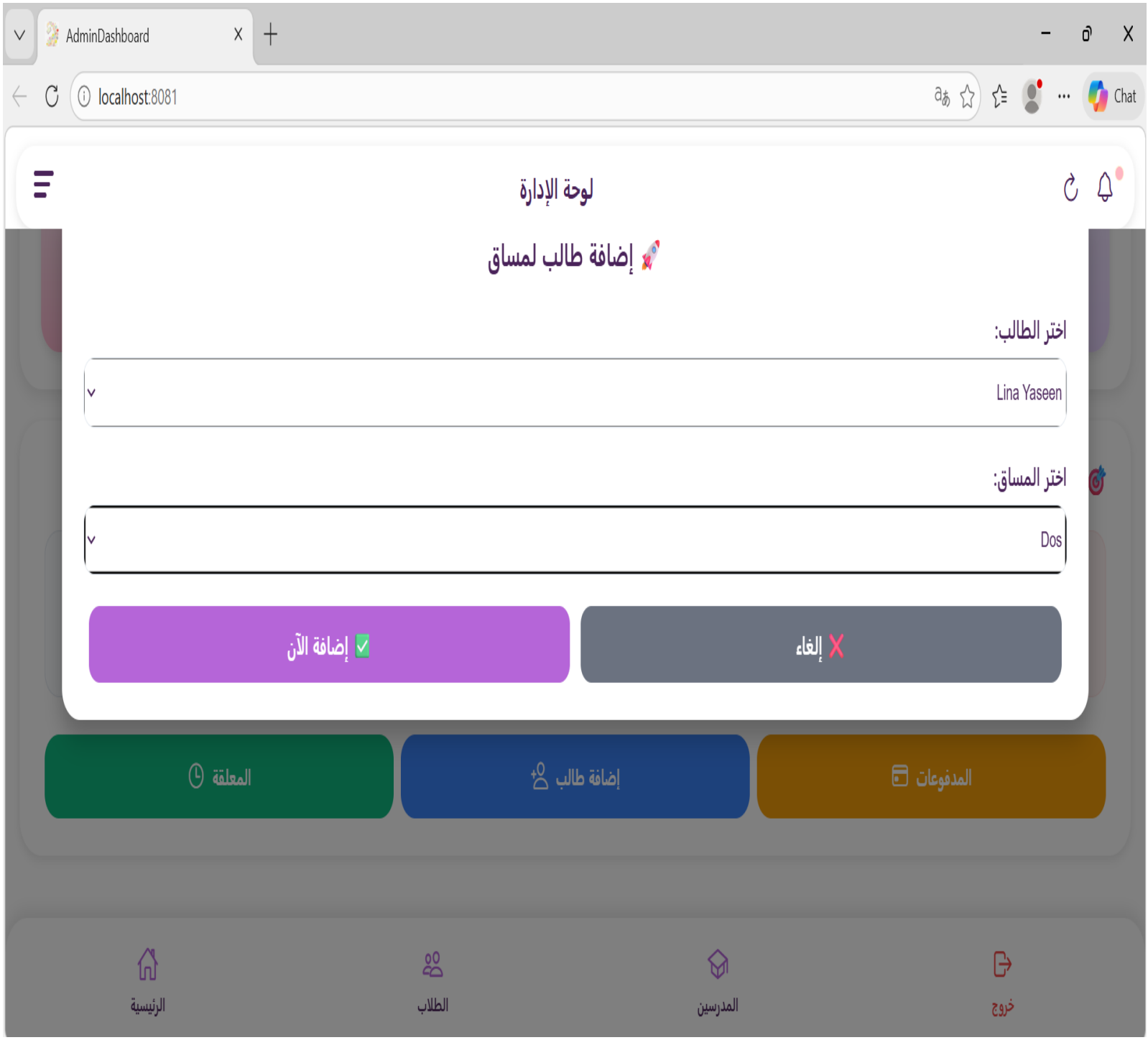


Figure 14 :Adding Students

7. Payments Screen

The Payments Screen allows administrators to view and manage all financial transactions on the platform in a centralized and organized manner. It provides a clear overview of student payments and helps ensure transparency and accuracy in financial records.

At the top of the screen, the admin can see the total number of payments and refresh the data when needed. A search bar is available to filter payments by student name, course name, payment date, or transaction ID, making it easy to locate specific records quickly.

The screen also displays a total revenue summary, which shows the accumulated amount of completed payments. This gives administrators an immediate insight into the platform's financial performance.

Each payment is shown in a structured card that includes the student name, course name, payment amount, and payment date. The payment status is clearly indicated as Completed, Pending, or Rejected using visual badges for quick identification.

Administrators can delete payment records when necessary, with a confirmation prompt to prevent accidental actions. The screen also supports data refreshing and clear empty states to improve usability.

Overall, the Payments Screen provides an efficient and user-friendly interface for monitoring and managing payments within the system.

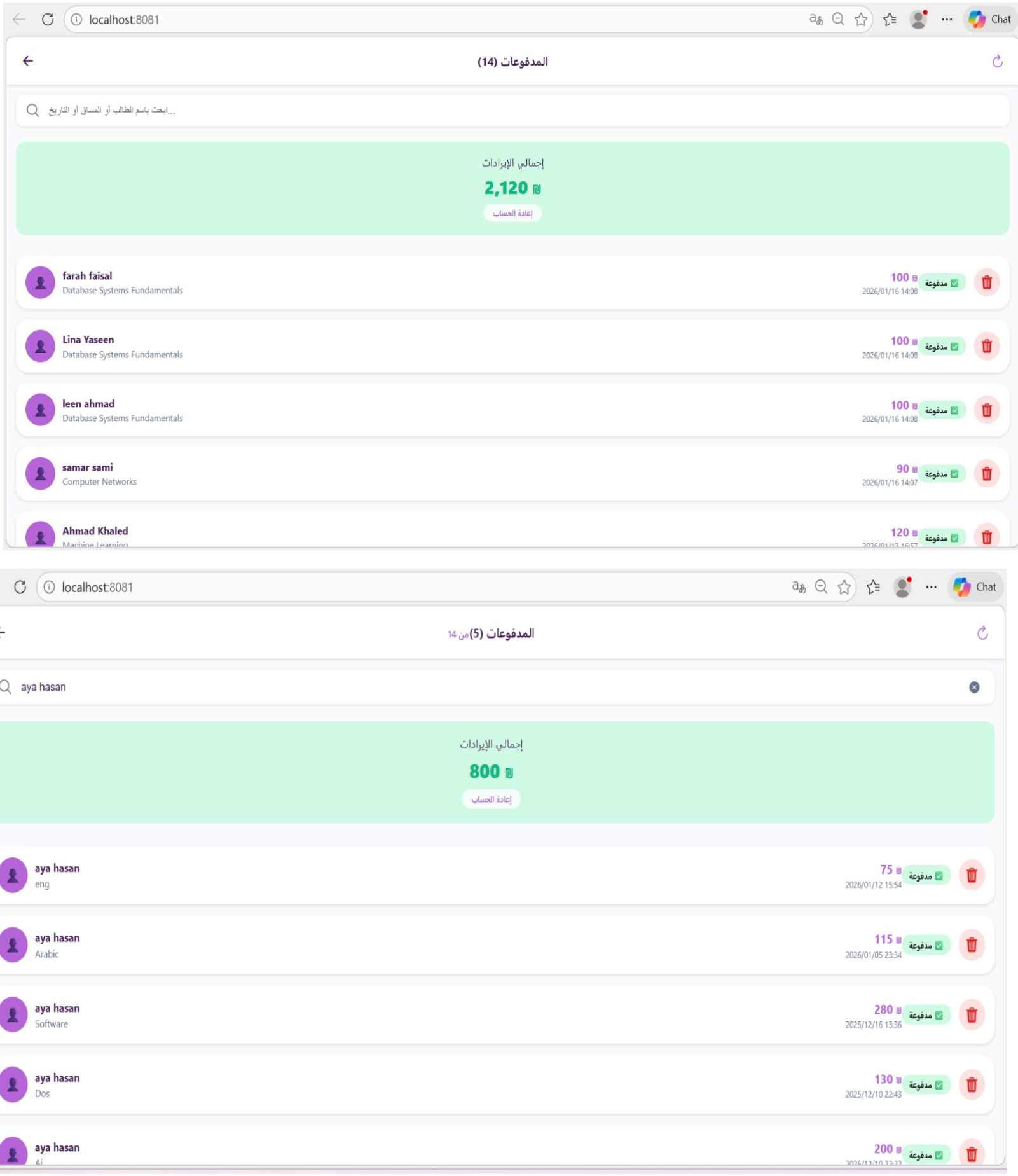


Figure 15 : Payments Screen.

8. Reports Screen

The **Reports Screen** provides administrators with a comprehensive overview of the platform's performance through visual analytics and key statistics. It is designed to support data-driven decision making by presenting financial, academic, and user-related insights in a clear and organized layout.

At the top of the screen, key performance indicators (KPIs) are displayed, including the total number of students, teachers, courses, and enrollments. These summary cards allow administrators to quickly assess the overall status of the system at a glance.

The screen includes detailed **revenue analytics**, showing total revenue, today's revenue, and monthly revenue. A line chart visualizes the overall revenue distribution, making trends easier to understand. Administrators can also apply a **date range filter** to view revenue for a specific period, with the filtered amount highlighted separately for clarity.

In addition, the Reports Screen presents insights into course and teaching performance. A bar chart shows the distribution of courses by status (approved, pending, rejected), while a ranked list highlights the **top five teachers** based on student enrollment. This helps identify high-performing instructors and course engagement levels.

The screen also includes **enrollment statistics**, displaying total enrollments and daily enrollments, along with a summarized revenue section for quick reference. Error handling, loading indicators, and refresh functionality are integrated to ensure a smooth and reliable user experience.

Overall, the Reports Screen acts as a centralized dashboard that combines analytics, charts, and statistics to
give administrators a clear and actionable overview of the platform's performance.

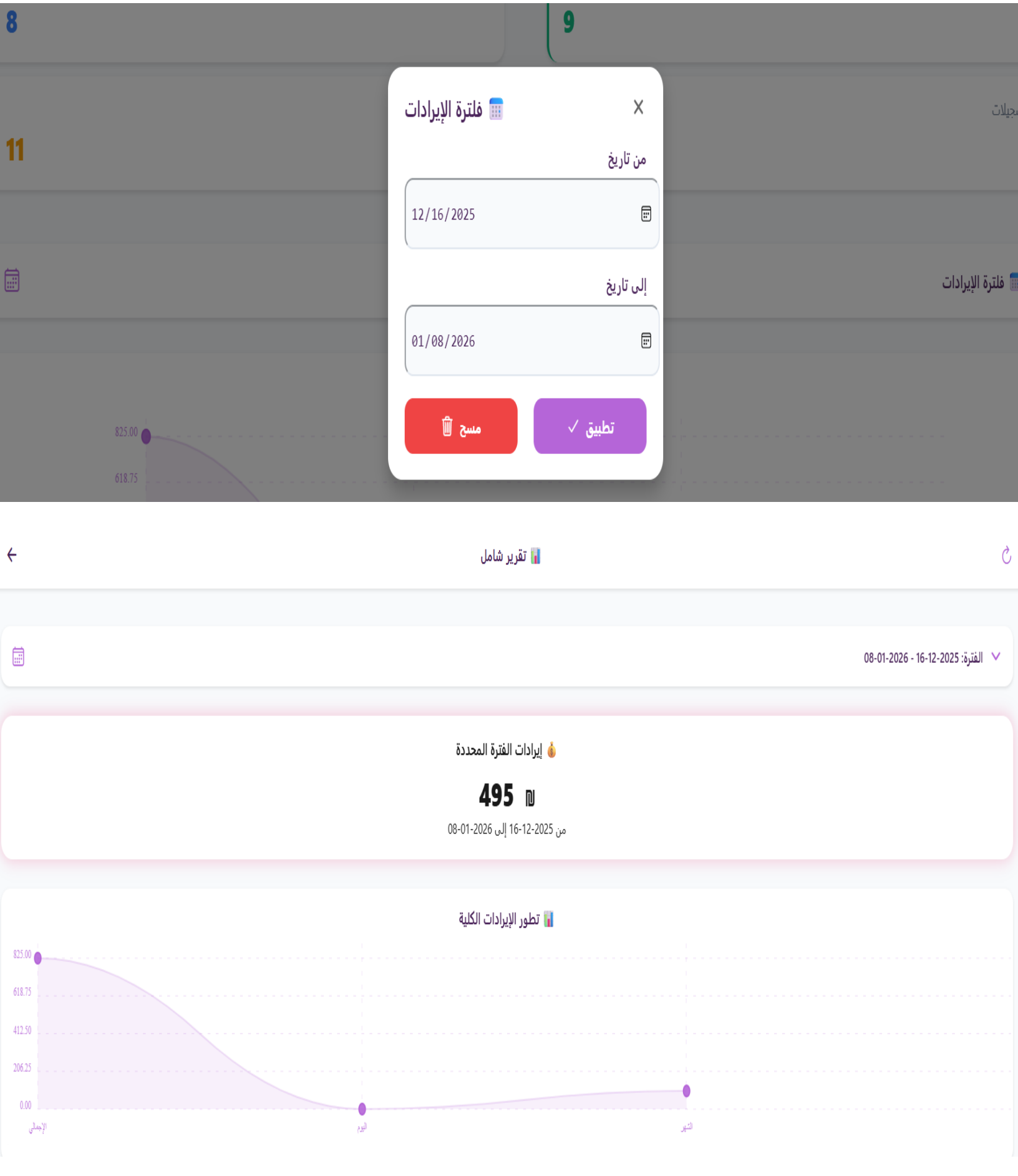


Figure 16 :Platform Reports Screen

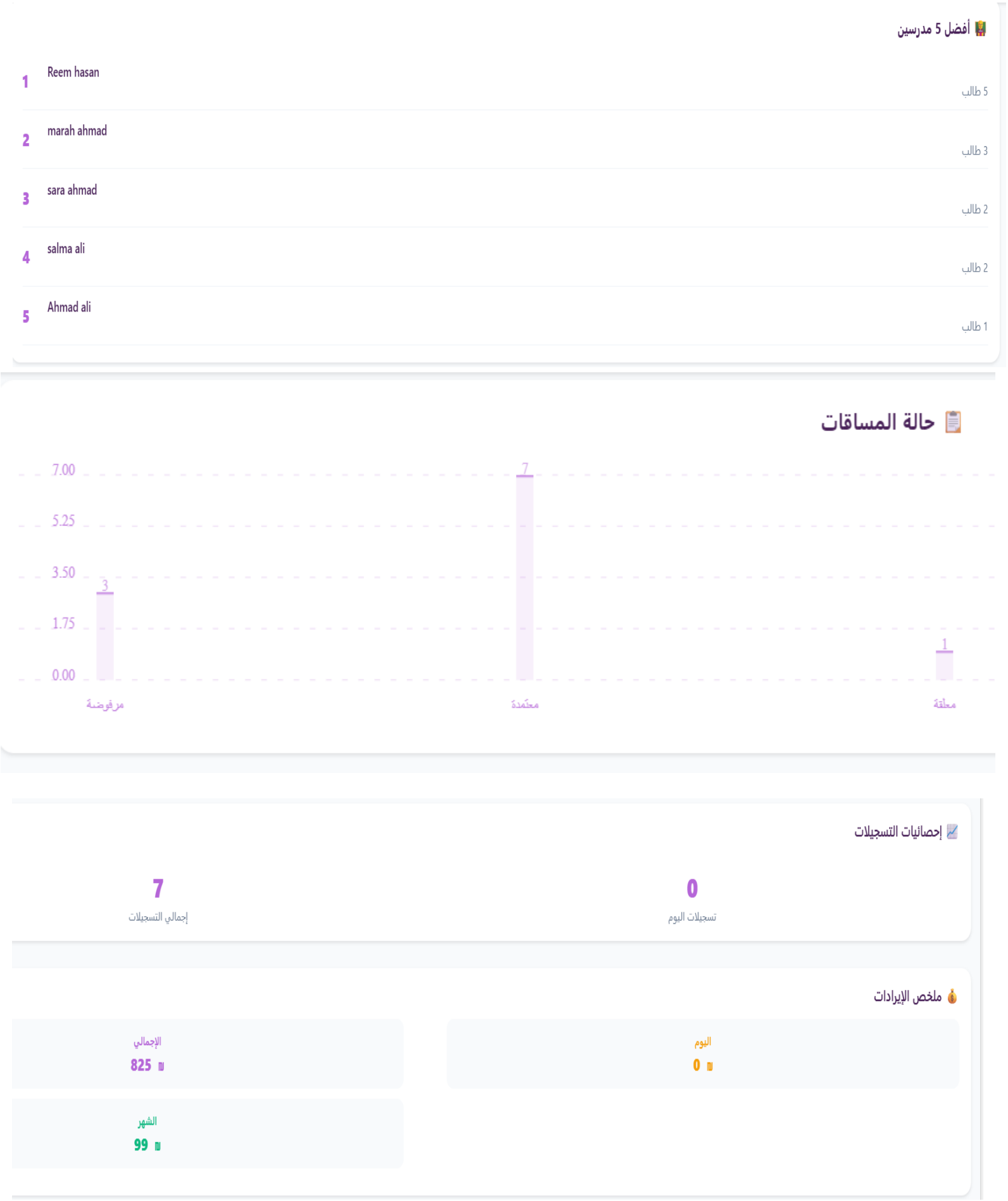


Figure 17 : Platform Reports

9. Settings Screen

To allow the logged-in administrator to manage their own profile settings

The **Admin Settings Screen** allows administrators to manage their personal information and account security. The screen retrieves the admin's profile data securely using a JWT-authenticated request and displays a loading indicator while the data is being fetched.

Administrators can update their personal details such as name and email, with input validation to ensure data accuracy. Any changes are saved through secure backend requests, and toast notifications provide immediate feedback on the success or failure of the operation.

The screen also includes a password management section where administrators can change their password by entering the current password and a new one. Password validation is enforced, and successful updates are confirmed through toast notifications, ensuring a secure and user-friendly experience.

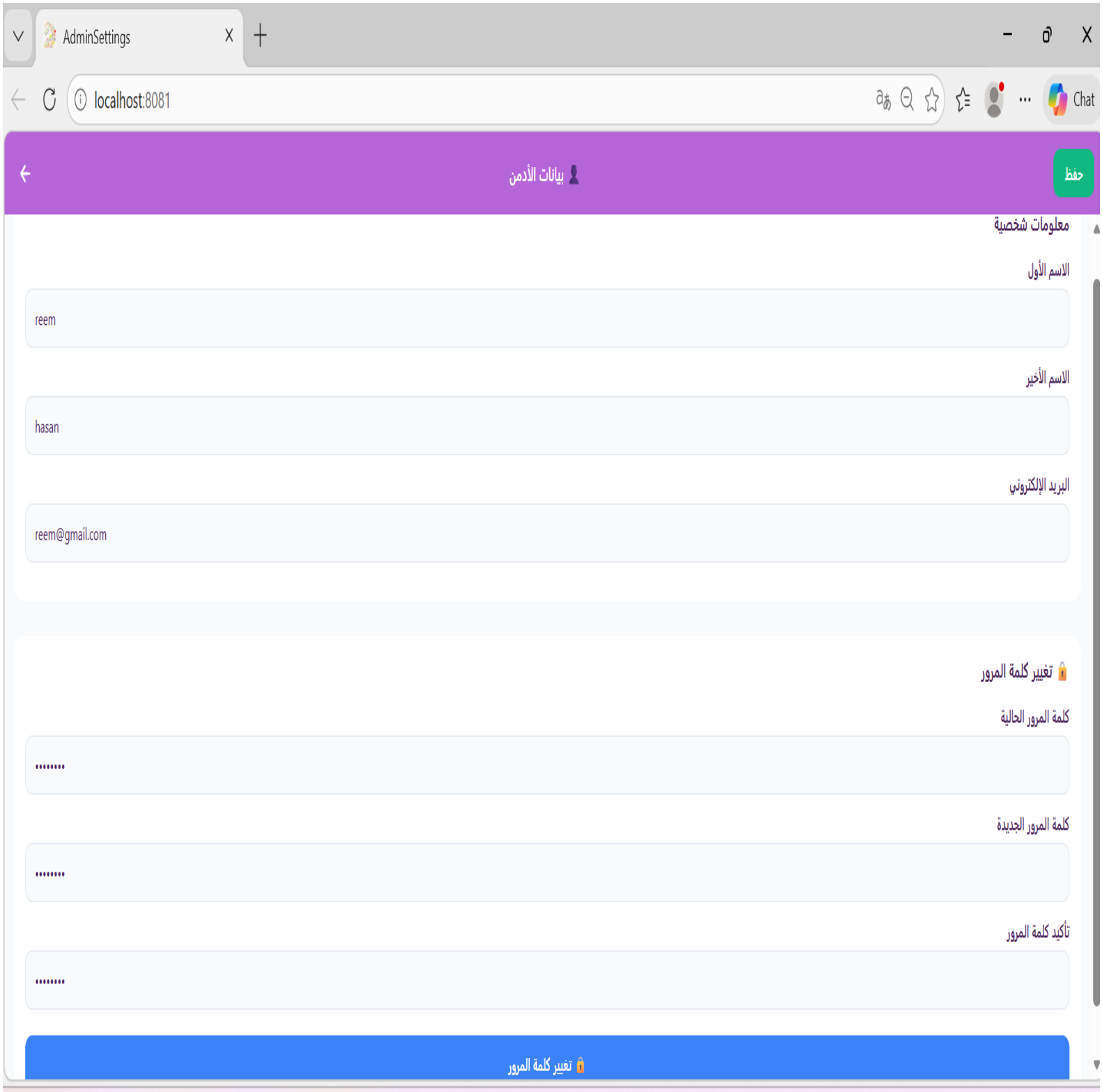


Figure 18: Admin Settings.

3.2.2 Teacher Dashboard

The Teacher Dashboard consists of several sections that allow the teacher to manage courses, students, and educational content efficiently. It is organized into a main dashboard screen and multiple sub-pages, each serving a specific purpose.

The main dashboard screen provides a general overview of the teacher's activity. It displays summary information such as the number of assigned courses, enrolled students, and recent notifications related to messages, assignments, or exams. Quick access buttons are also available to help the teacher perform common actions easily.

The My Courses page displays all courses assigned to the teacher. From this page, the teacher can view the list of courses and navigate to the details of each course. Selecting a course opens the Course Details page, which contains course information, a list of enrolled students, uploaded lessons, and related assignments. The dashboard allows the teacher to upload new lessons, add assignments or exams, edit existing content, and view student submissions. Teachers can review submissions, provide feedback, and evaluate student performance.

The My Students page shows only the students who are enrolled in the teacher's courses. This page allows the teacher to view student information and monitor their participation within specific courses.
The dashboard also includes a messaging and notification system that helps the teacher stay updated with student interactions and course-related activities.

Finally, the Settings page enables the teacher to manage account-related options such as updating personal information, changing the password, and adjusting notification preferences .

Figure 19 :Teacher Dashboard

Figure 19.1:Teacher Dashboard

Figure 19.2:Teacher Dashboard

Course Management

The Courses Management section in the Teacher Dashboard allows educators to efficiently create, view, and manage their courses. This section mainly consists of two screens: My Courses and Add Course.

My Courses Screen

This screen displays all courses associated with the current teacher. Each course is represented by a card showing key information such as course name, course code, course description, course duration, course price, and approval status.

Course Statuses:

Pending: The course has been submitted and is awaiting admin approval.
Approved: The course is accepted and active.
Rejected: The course was not approved by the admin.

The screen also includes a fixed button for adding a new course, allowing teachers to quickly create new courses.

Note: From the dashboard slider, clicking the “Add Course” quick action also navigates to this Add Course screen.

Add Course Screen

This screen enables teachers to add new courses using a structured form that includes course code, course name, course description, course duration (in hours), and course price. The form validates required fields before submission. Once submitted, the course is registered with a Pending status until reviewed by the admin.

Figure 20: Add Course Screen

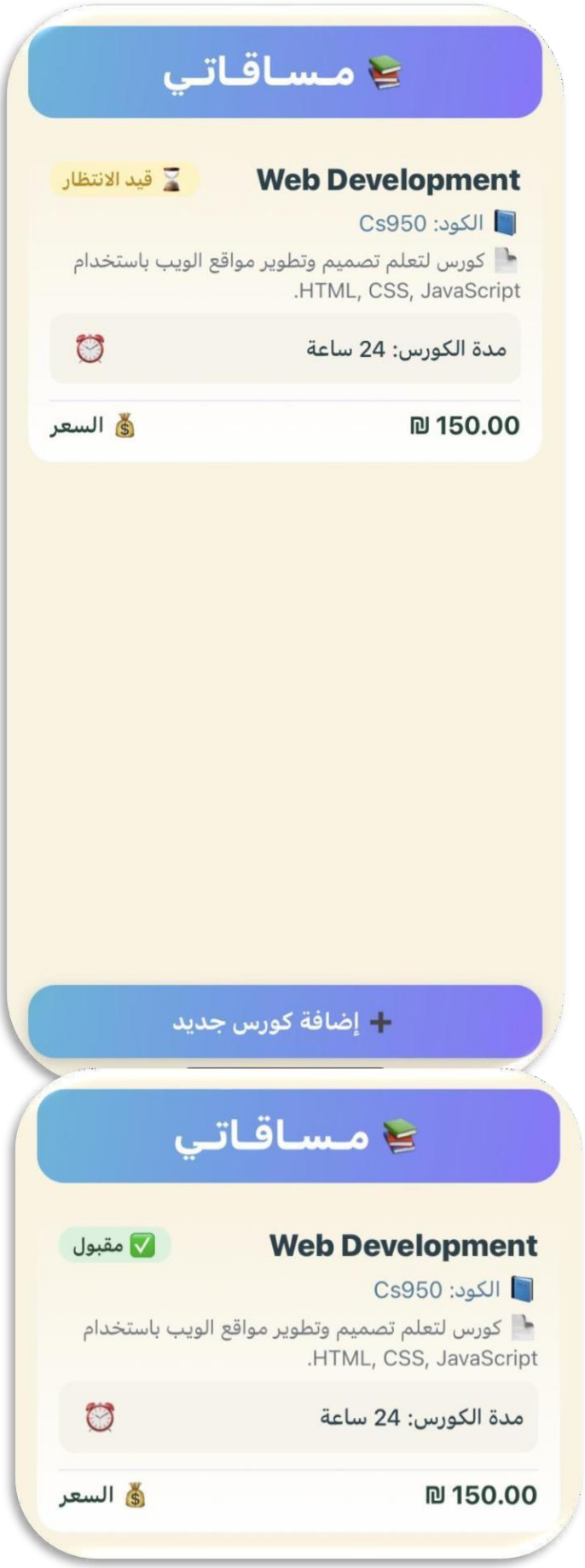


Figure 21: My Courses Screen

Course Plan Screen

The Course Plan screen provides teachers with a comprehensive interface to plan, organize, and manage their courses on a weekly basis. The total number of weeks is automatically calculated based on the course duration. For example, the app divides the total course hours by 3 (hours per week) to determine the number of weeks. Each week can be viewed as a separate section where teachers can add or edit:

Week Title: The main topic or focus of the week.

Objectives: Specific learning goals or outcomes expected for that week.

Content: Lessons, materials, or activities planned for the week.

Evaluation: Methods or criteria for assessing student performance.

Teachers can open a modal for any week to input or update this information. If a week already has data, it is prefilled for easier editing. The updates are saved to the server immediately, ensuring that the course plan stays current.

The screen also provides a clear overview of all weeks, allowing teachers to track which weeks have content and which need planning. This ensures structured teaching, efficient management, and easy tracking of the course’s progress over time.

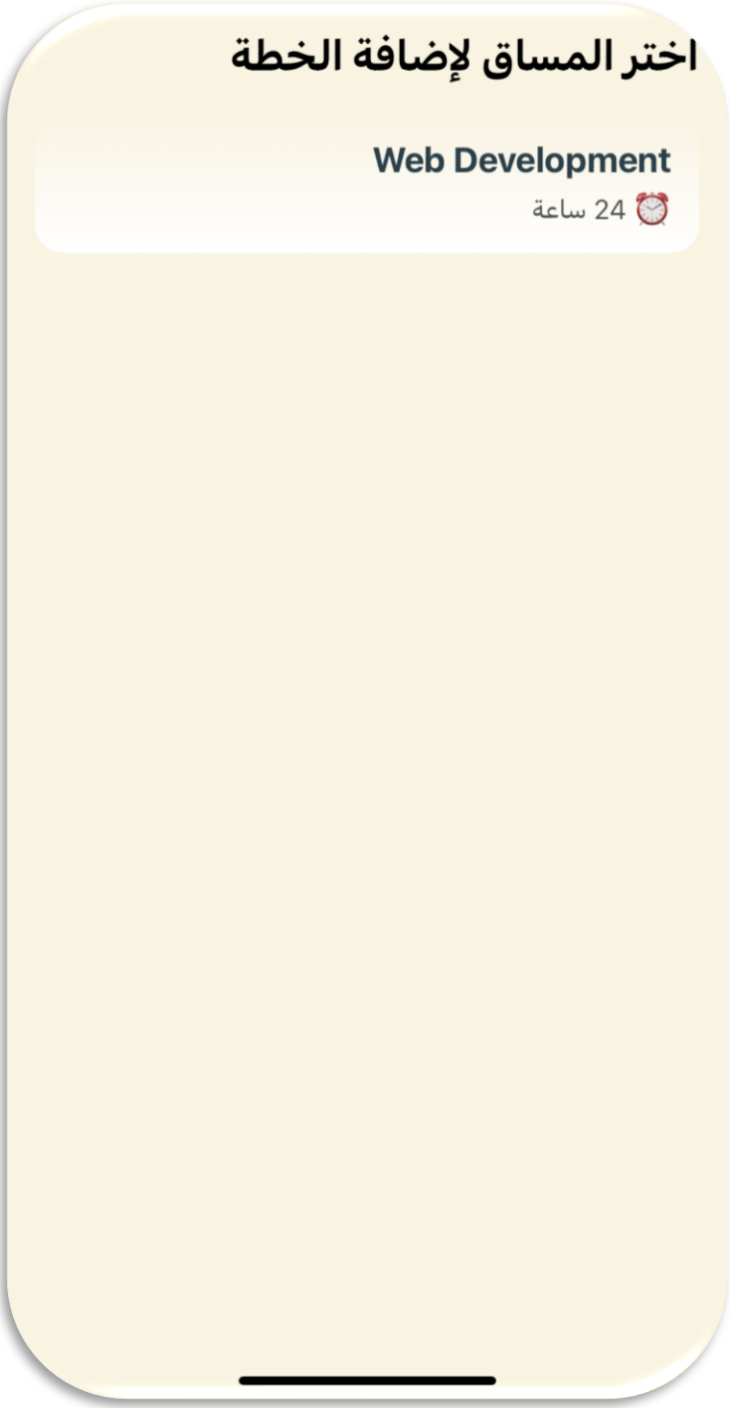


Figure 22: add plan Screen

Figure 22.1: add plan Screen

Live Sessions Screen

The screen displays current, past, and upcoming sessions in an organized manner using cards that show the course name, session title, start and end times, and password if available. The status of each session is displayed clearly to indicate whether it is live, ended, or upcoming.

Teachers can copy the session link directly and open the Zoom application automatically if installed, or use the link in a browser.

The screen allows creating a new session by selecting a course from the approved courses and entering the session title, Zoom link, password, and start and end times. The times are automatically adjusted to match the correct timezone.

When a new session is created, notifications are sent to all students enrolled in the course using Firebase Firestore.

The screen supports automatic updating of the session list at regular intervals and also allows manual refresh using pull-to-refresh. Sessions can be organized and displayed by the current week, upcoming weeks, and past weeks to facilitate tracking and reporting.

The screen provides teachers with an easy-to-use interface for managing live sessions fully, with continuous updates, automatic notifications to students, easy access to session links, Zoom integration, and the ability to organize sessions by weeks for better reporting.

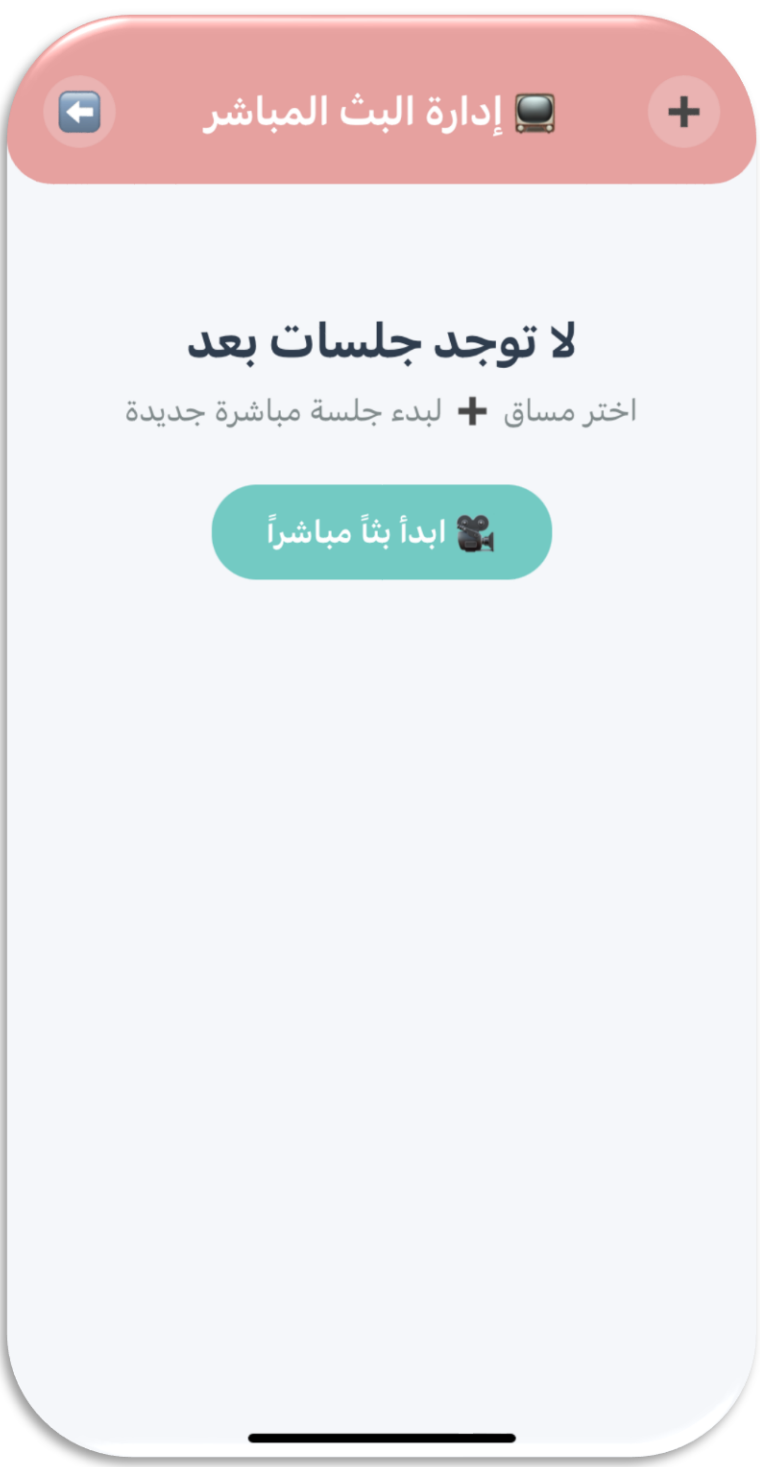


Figure 23: Live Sessions Screen

Figure 23.1: Live Sessions Screen

Lesson Management

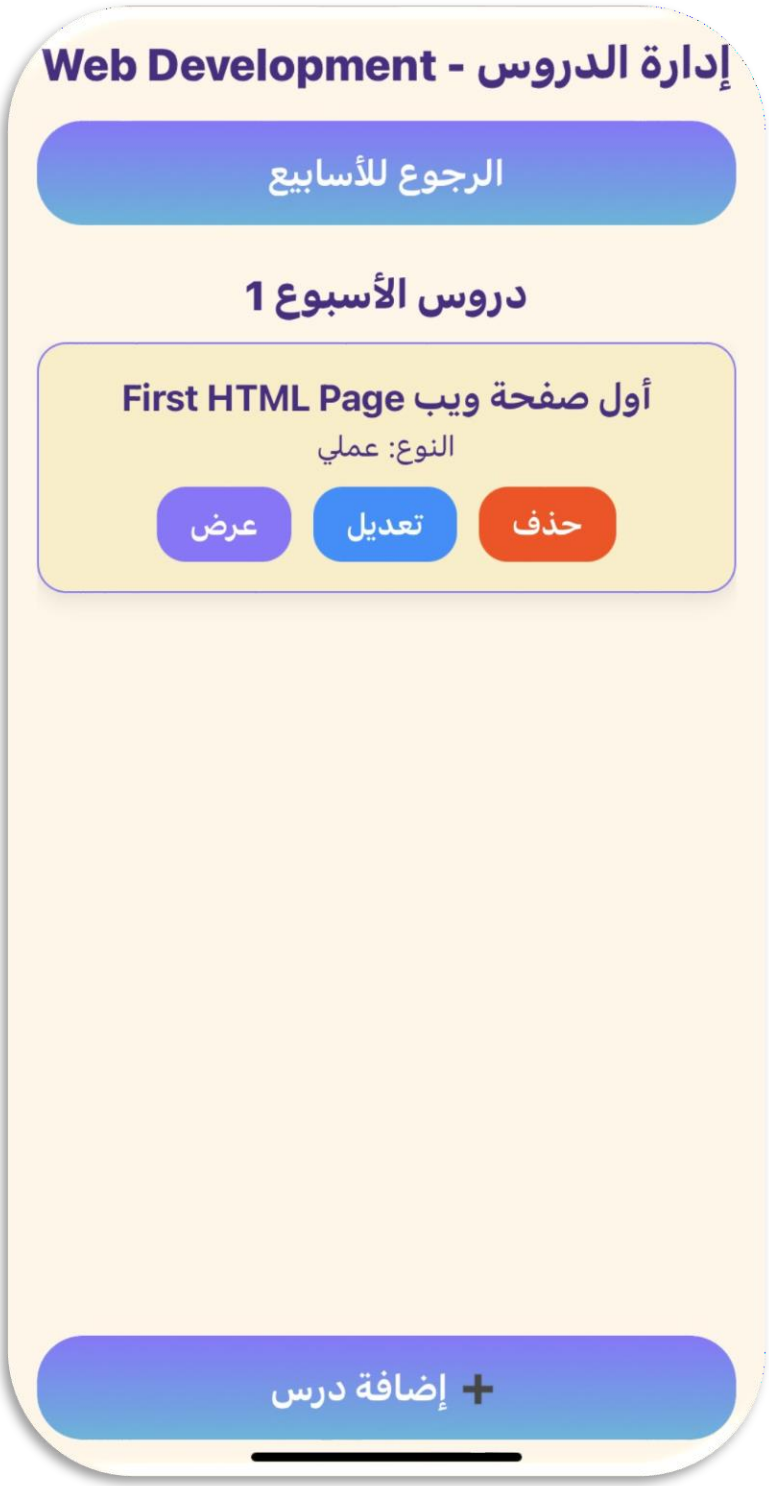
The LessonsScreen displays all lessons associated with a specific course, allowing the teacher to select a week. The number of weeks is calculated based on the course duration. When a week is selected, the lessons for that week are fetched from the server and displayed as cards showing the lesson title and type, with options to view details, edit, or delete the lesson. The teacher can return to the week selection screen or add a new lesson directly from this screen.

The UploadLessonScreen allows uploading a new lesson or editing an existing one. Teachers can select the lesson type, either audio, visual, or practical, enter the lesson title, and upload a file, whether it is audio, video, or other types. The screen supports previewing the files before uploading and allows playing videos and audio directly to verify the content. After entering the information and selecting the file, the data is sent to the server via Axios using FormData to handle file uploads.

The LessonDetailsScreen displays the full details of a lesson, including its title, type, and attached files. It supports displaying images, videos, and audio, with audio playback enabled even if the device is in silent mode. Files that cannot be displayed directly can be opened in a browser. All screens use a consistent and visually appealing design with clear and easy-to-use buttons to ensure a smooth user experience for teachers.

In summary, these screens provide teachers with a complete interface to manage lessons, from selecting the week, viewing lessons, uploading or editing lessons, to viewing details and playing attached files easily and securely

Figure 24:Add Lesson Screen



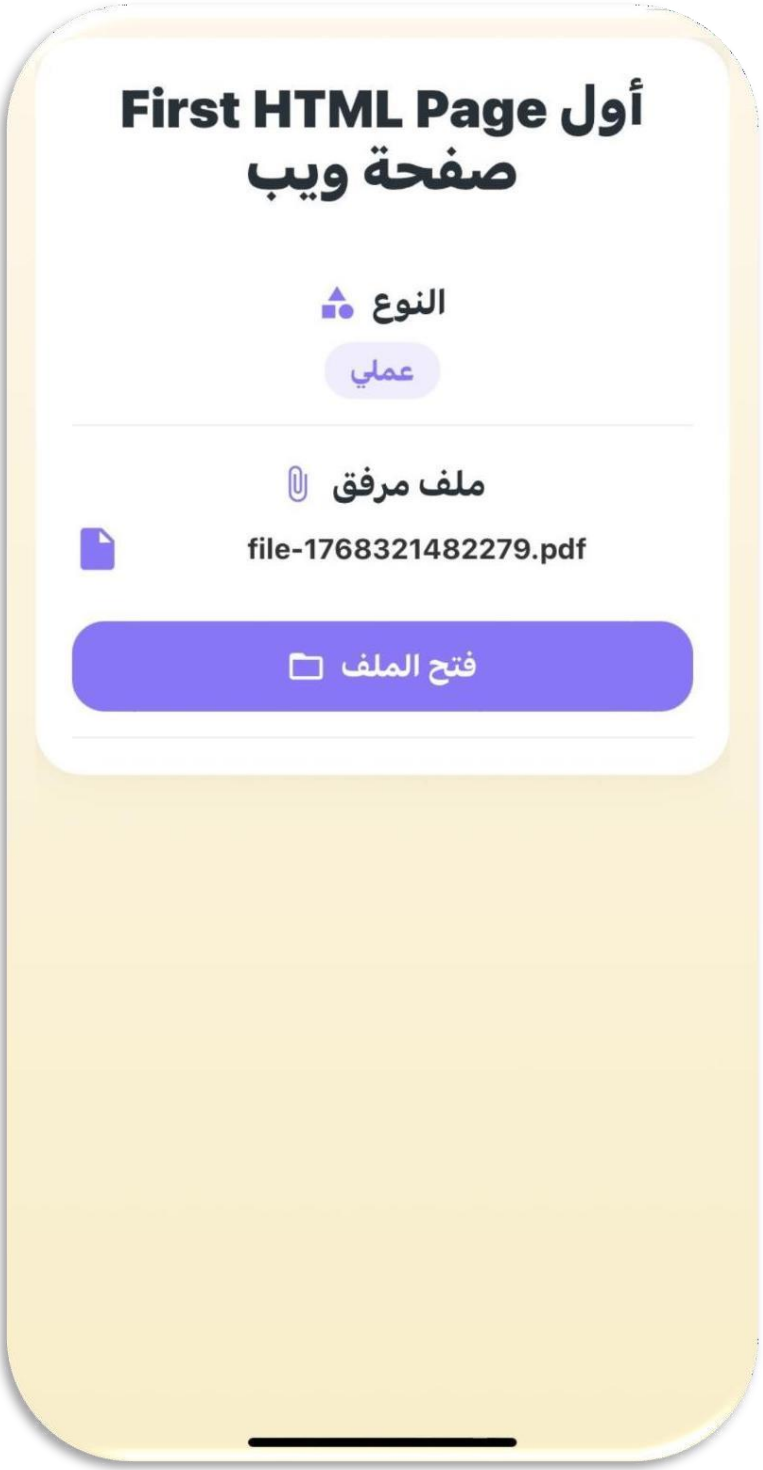


Figure 25: View Lesson Screen

Figure 26: Update Lesson Screen

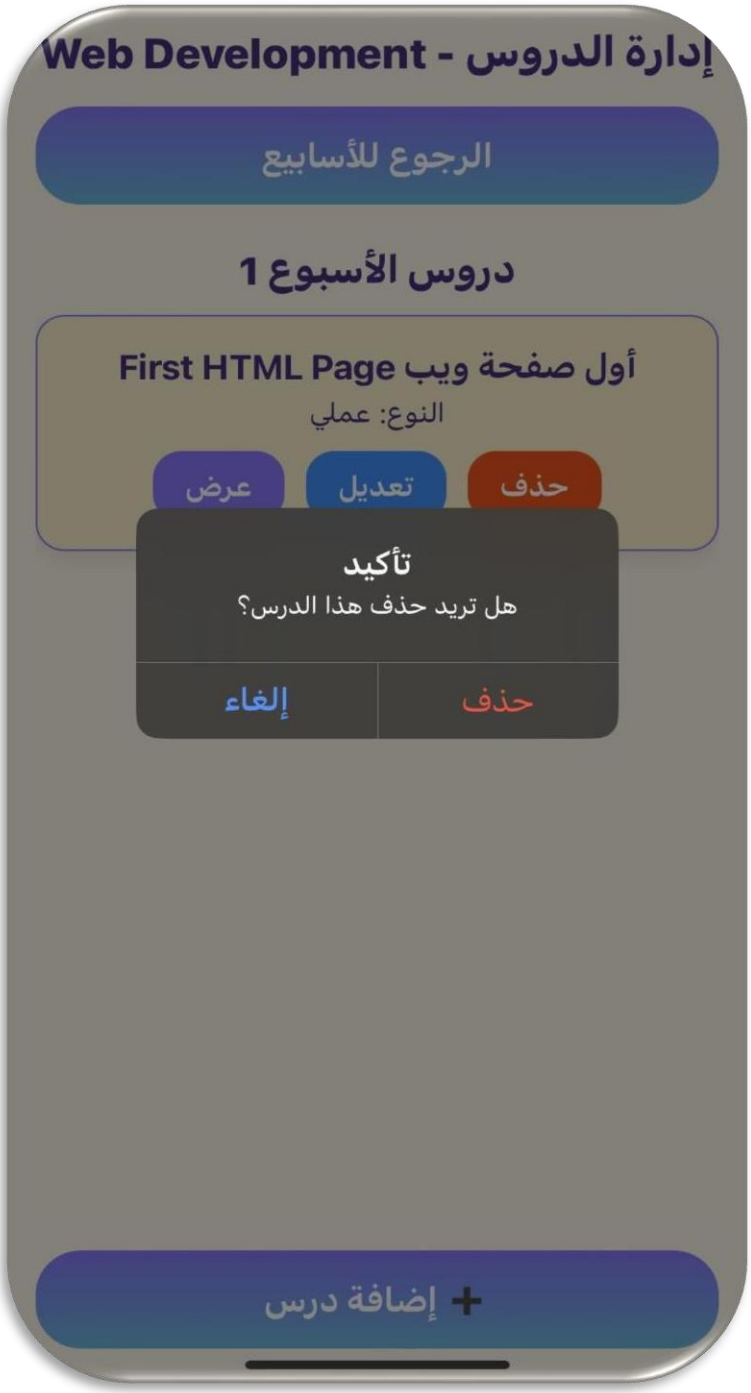


Figure 27: Delete Lesson

Messaging Screens

Students Screen:
This screen allows the teacher to view all students assigned to them. It displays each student's full name and the latest message received. The teacher can search students by first or last name. The student list updates in real-time using Firebase Firestore to fetch the latest messages.

Teacher Messages Screen:
This screen provides real-time messaging between the teacher and a selected student. Messages are fetched and displayed instantly using Firebase onSnapshot. The teacher can send new messages, which are immediately added to the list and stored in Firestore. Notifications are also generated for the student when a new message is sent. Messages from the teacher and student are visually differentiated for clarity.

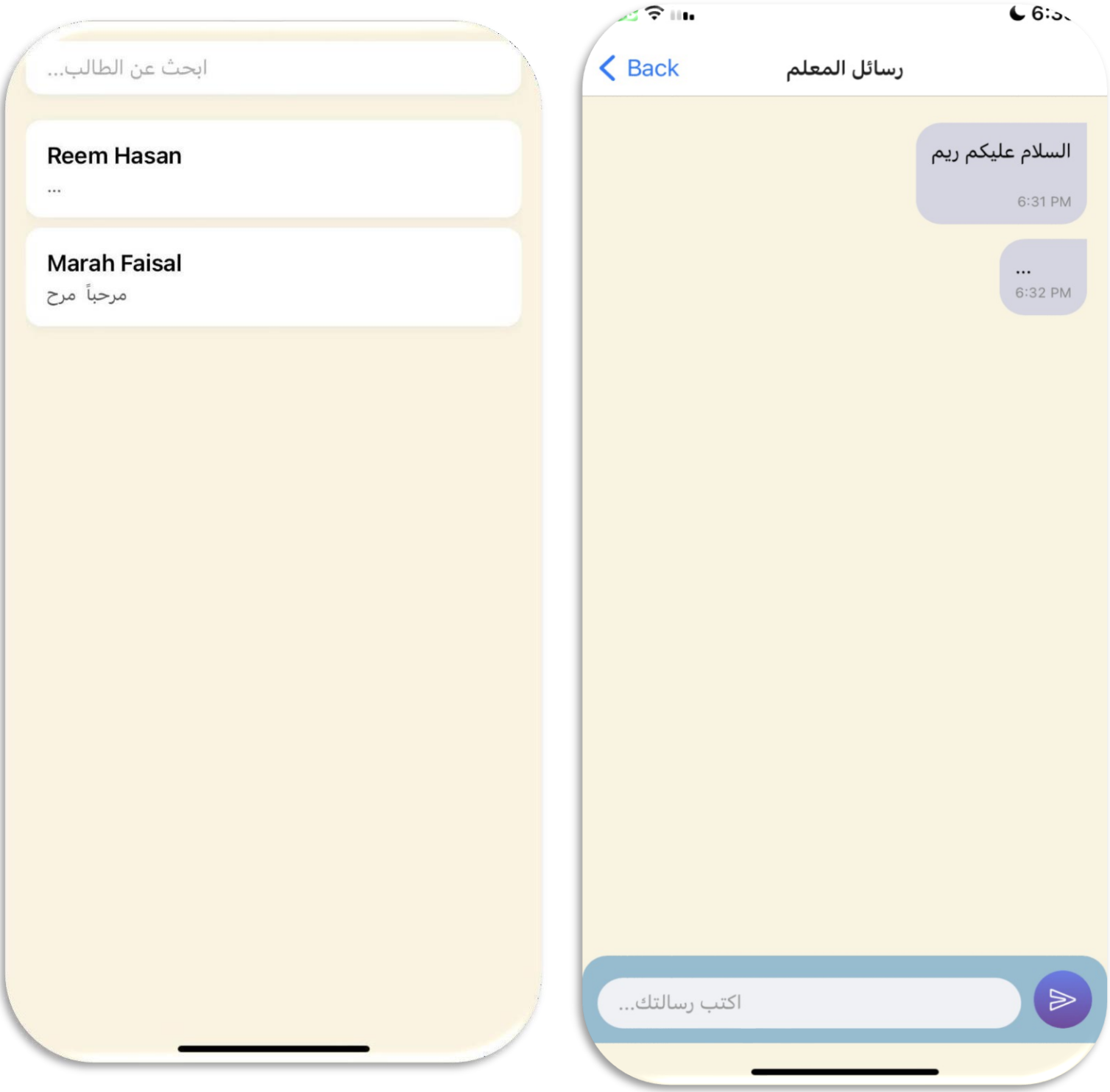


Figure 28: Messaging Screen

Student List Screen

This screen displays all students assigned to a specific teacher, organized by their respective courses. Each course is presented as a collapsible card, showing the course name, the number of students enrolled, and an expandable list of students. When a course card is tapped, it expands to reveal the students under that course, displaying each student's full name and email. The screen fetches data from the server on load and provides a loading indicator while retrieving information. Smooth animations enhance the user experience, making the list visually appealing and interactive.

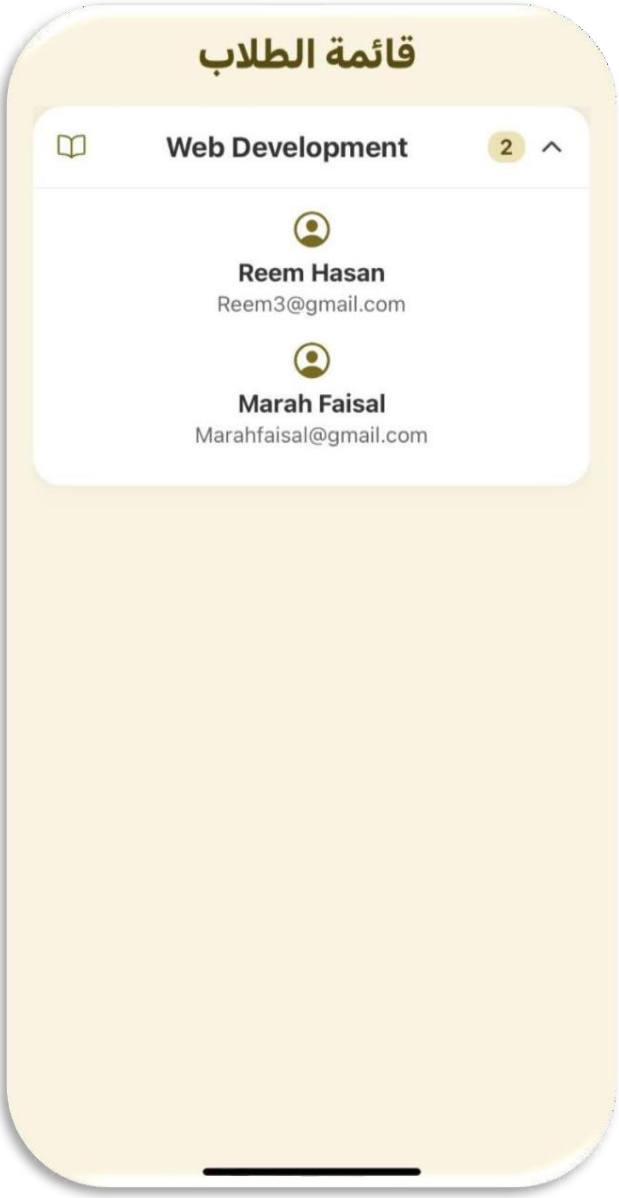


Figure 29: Student List Screen

Tasks Management

This module provides a complete workflow for managing course tasks. Teachers can view, add, edit, and check the details of each task efficiently. The Tasks Screen allows teachers to select a specific week within the course duration and displays all tasks for that week. Each task card shows the task title, due date, and provides actions to view, edit, or delete the task. The screen dynamically fetches tasks from the server, handles empty states, and provides smooth navigation between weeks.

The Add Task Screen enables teachers to create new tasks or edit existing ones. Teachers can input the task title, description, type, attach files, and set start and due dates. The screen also supports sending notifications to all enrolled students whenever a new task is added. Validation ensures all necessary fields are completed before submission. For tasks with attached media, the screen handles files, images, and videos properly, allowing previews when applicable.

The Task Details Screen provides a detailed view of any selected task. It displays the title, description, type, schedule, and any attached files or media. Attached images and videos can be previewed directly within the screen, while other file types can be opened externally. The design organizes all task information clearly, making it easy for teachers and students to review content.

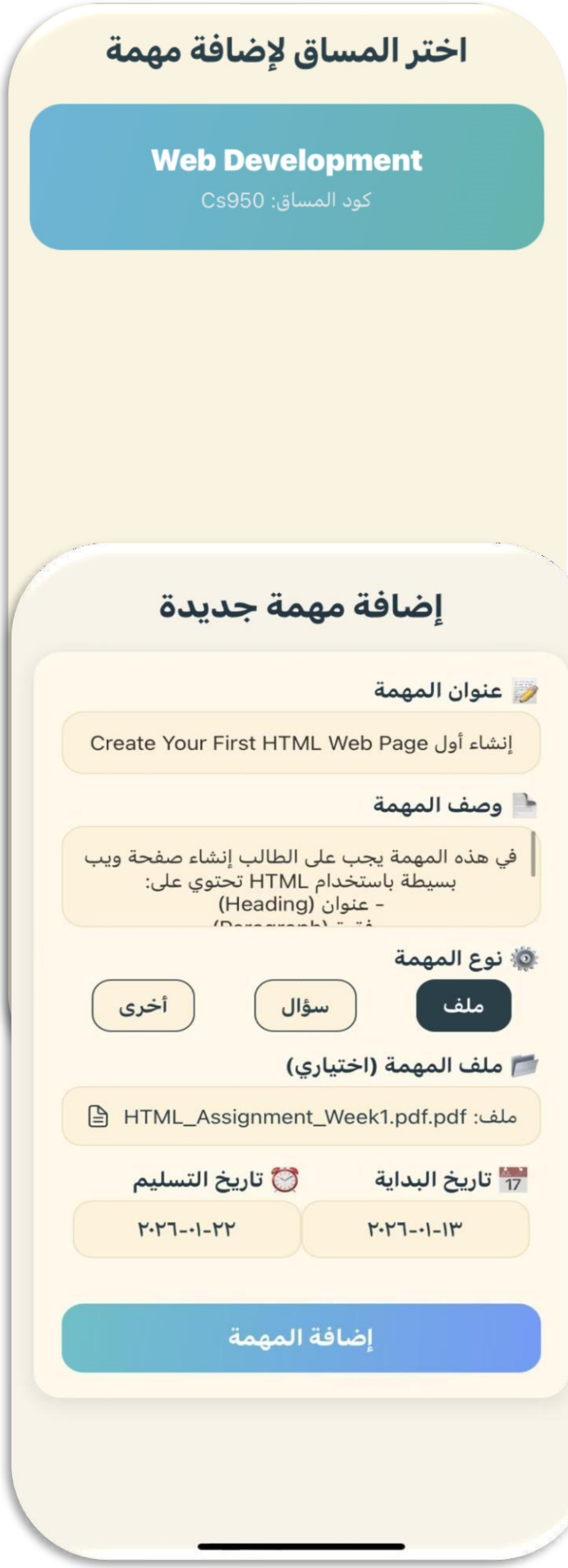


Figure 30: Add Task Screen

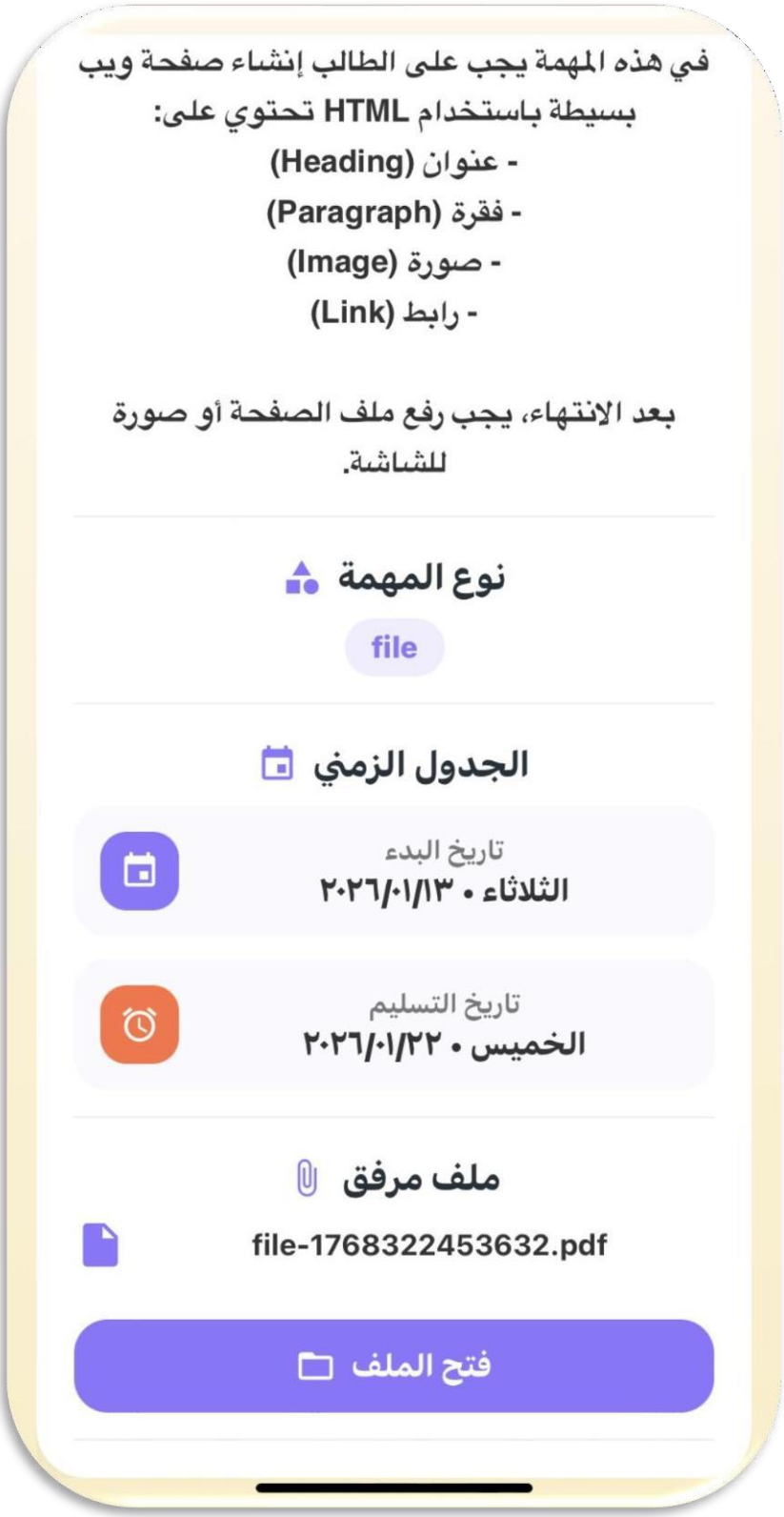


Figure 31: view Task Screen

Figure 32: Update Task Screen

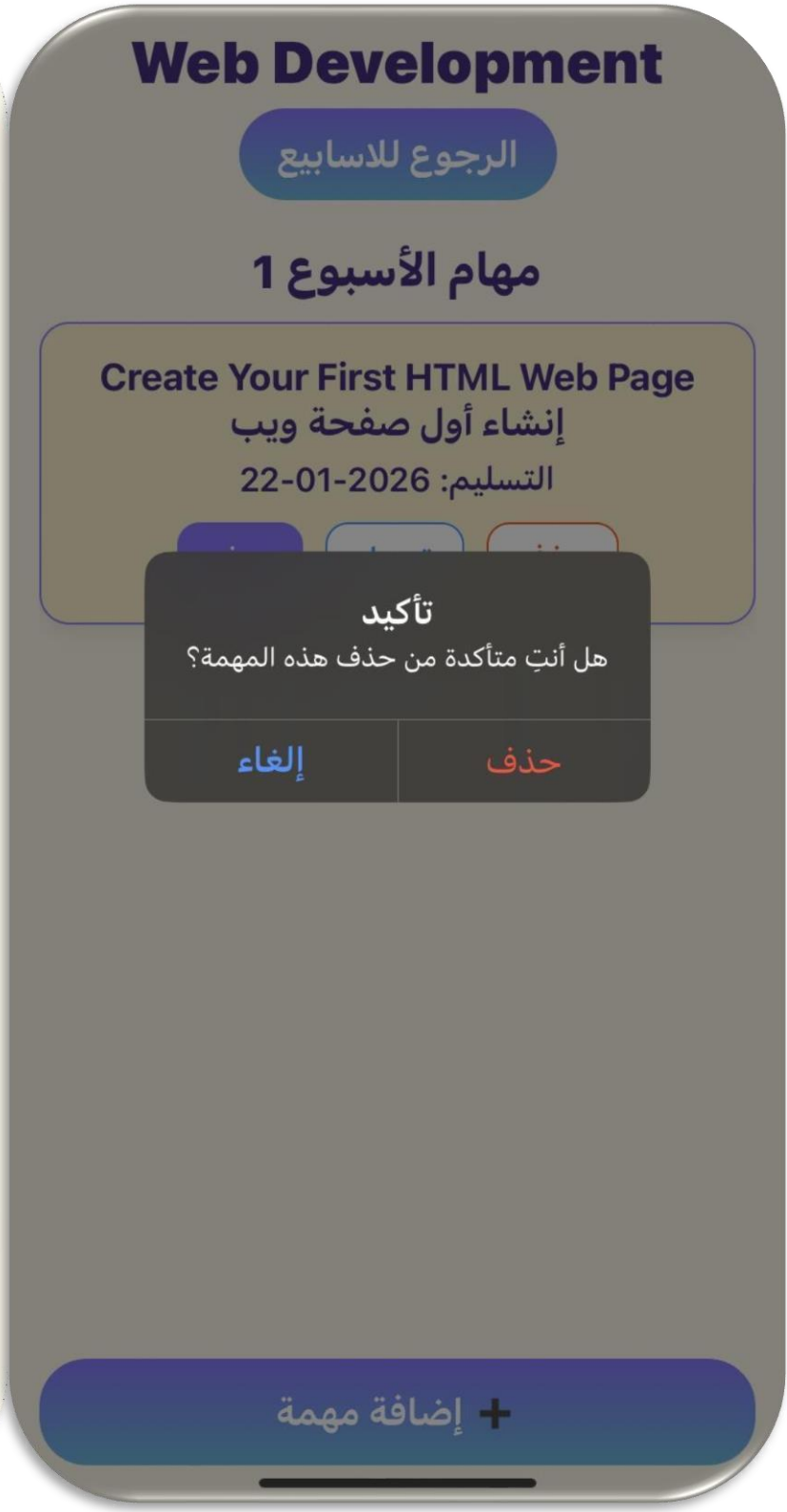


Figure 33: Delete Task Screen

Tests and Questions Management Screens

This module provides a complete and organized workflow for managing course assessments. It allows the teacher to create, manage, and monitor tests, as well as design and control their questions in a structured way. The tests are organized by weeks based on the course duration, which helps the teacher distribute assessments evenly throughout the course timeline.

The Tests Screen enables the teacher to view all weeks of the course and select a specific week to manage its tests. For each test, the teacher can add a title, description, start date, and end date. The teacher can also edit or delete existing tests when needed. Once a new test is created, students enrolled in the course are notified automatically. The screen also adapts its behavior based on the test status, allowing the teacher to add or edit questions while the test is active, and view test results after the test has ended.

The Questions Screen allows the teacher to fully manage the content of each test. The teacher can add different types of questions, including text-based questions, multiple-choice questions, and file-based questions. For multiple-choice questions, the teacher can define several options and select the correct answer. Each question can be assigned a specific mark value, ensuring flexible grading. The teacher can also edit or delete questions at any time. This screen ensures that all test questions are clearly organized and easy to manage, providing a smooth experience for building complete and well-structured assessments.

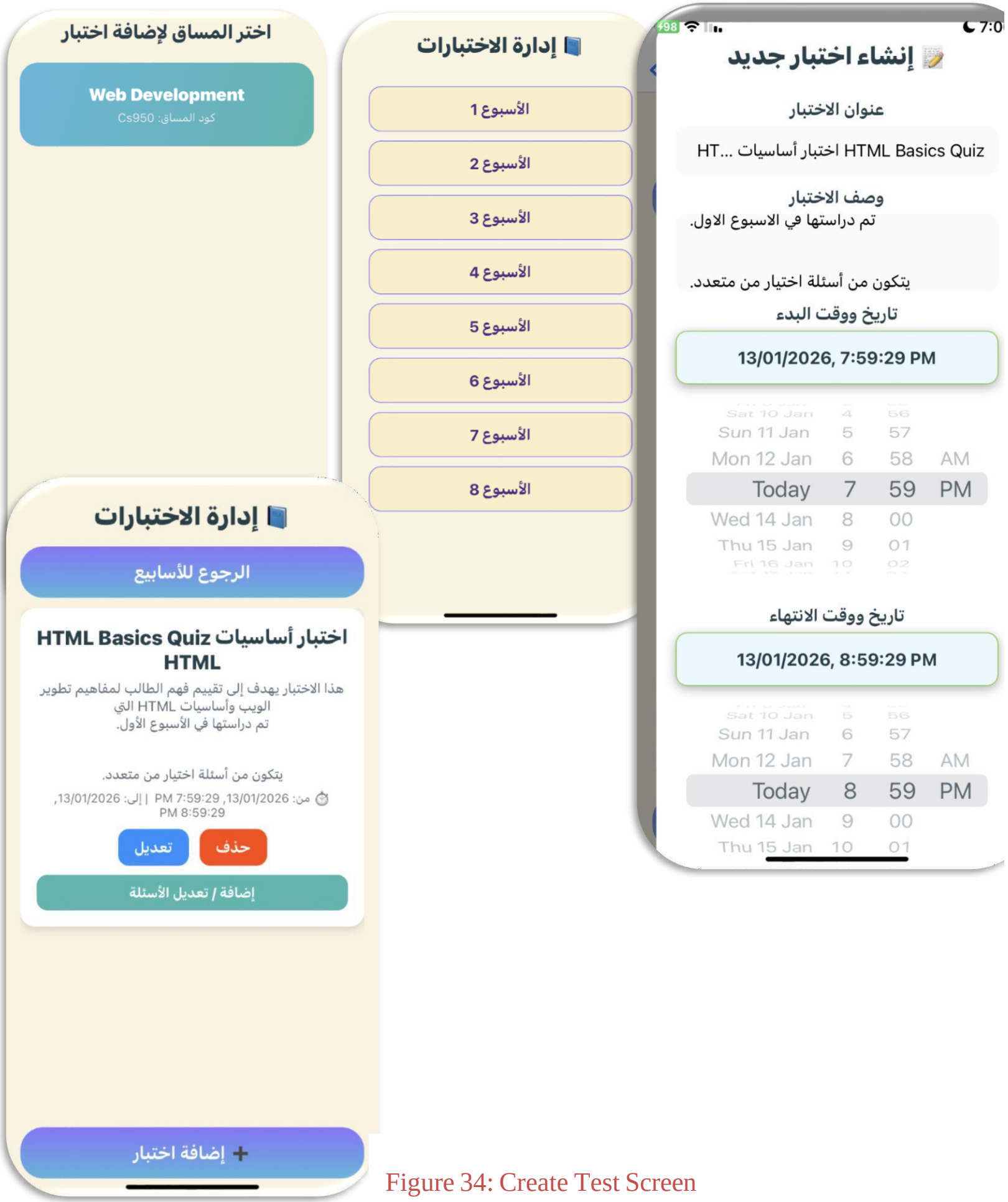


Figure 34: Create Test Screen

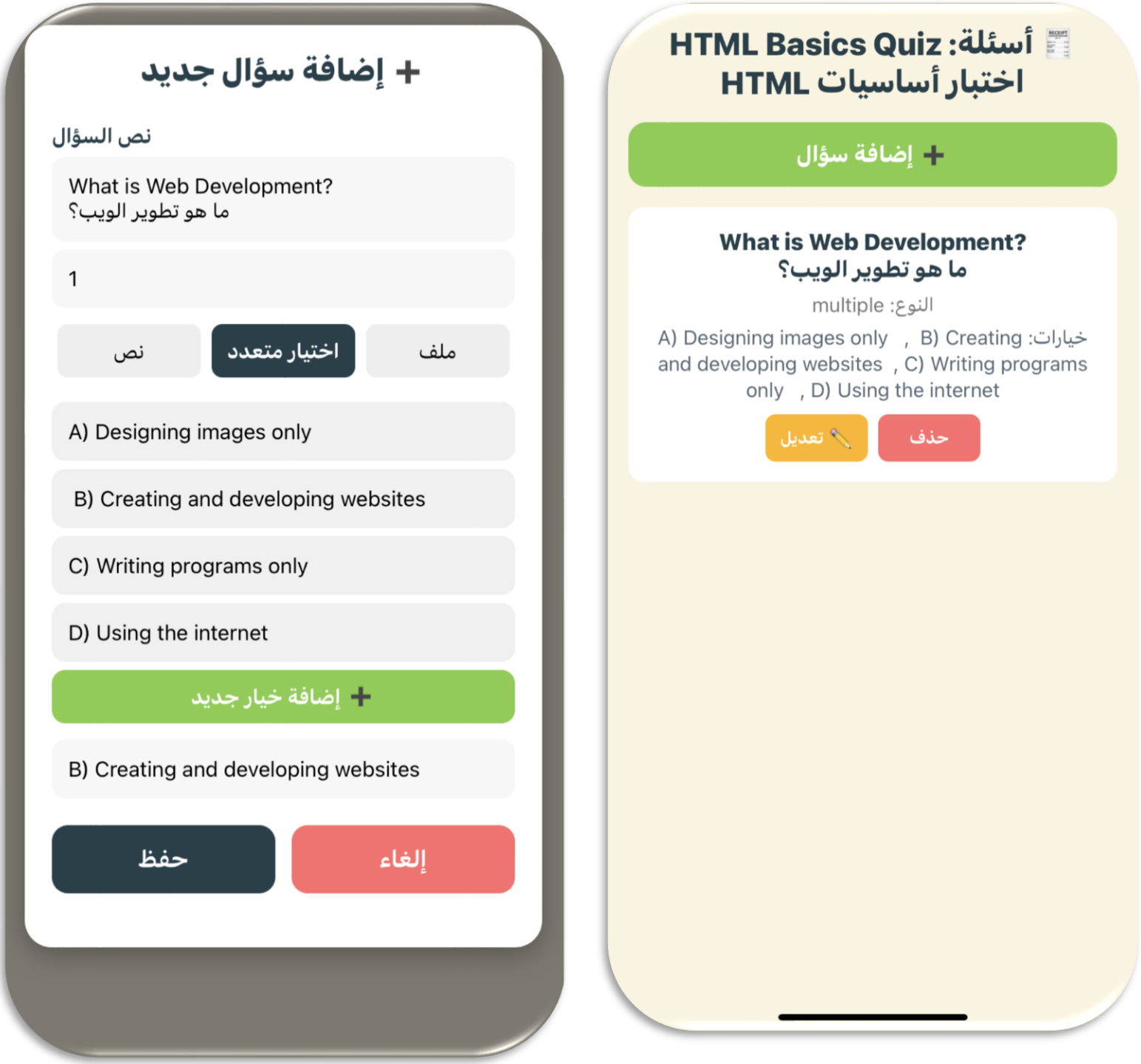


Figure 35: Add Questions Screen

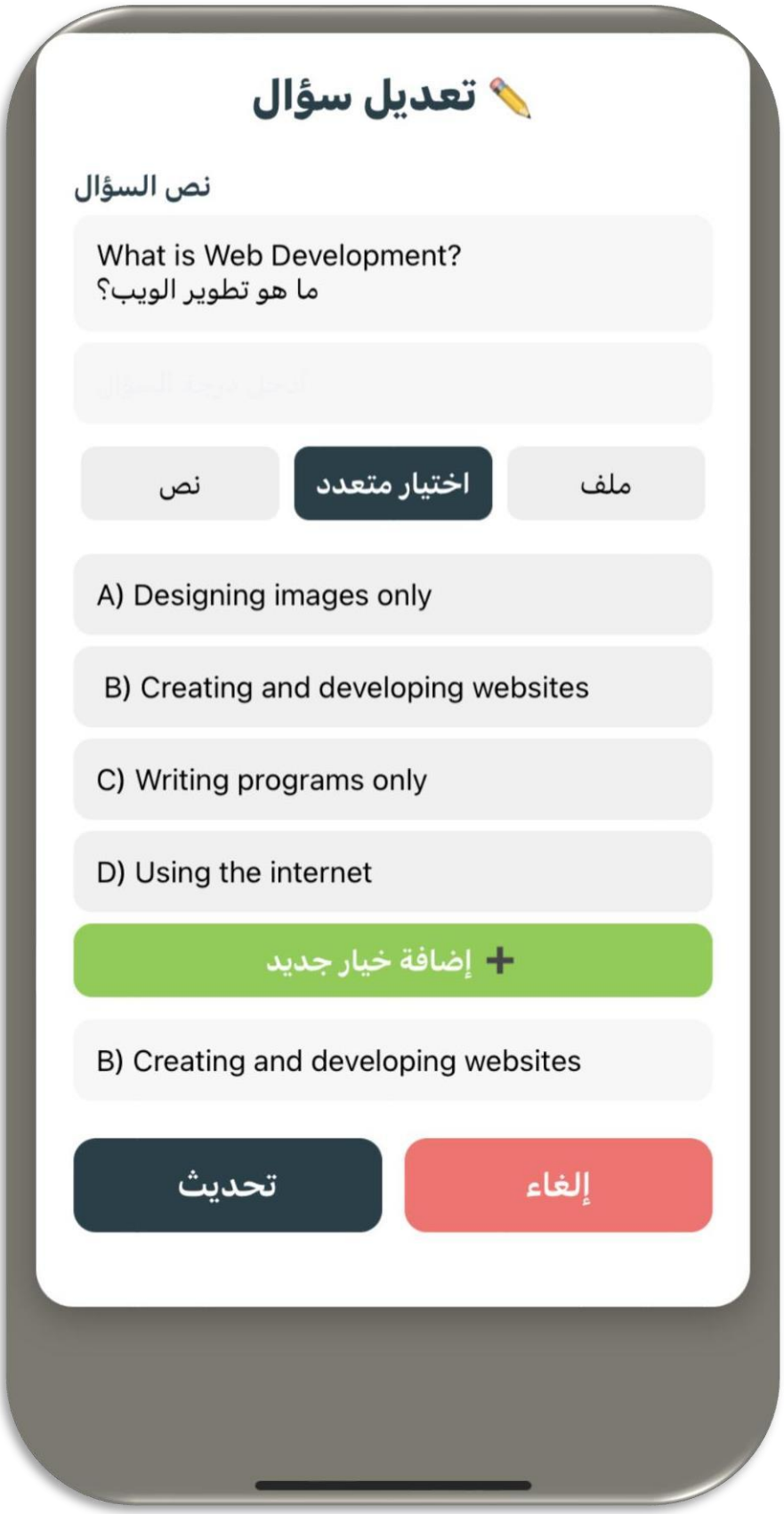


Figure 36: Edit Questions Screen

Teacher Test Results Screen

The Teacher Test Results Screen allows teachers to view and manage students' test submissions efficiently. Teachers can expand each student to see all their answers, review multiple-choice, text, or file uploads, enter and save grades manually for text/file answers, and view total marks compared to the maximum possible. Student-submitted files can be opened directly.

Figure 37 : Teacher Test Results Screen

Course Content Overview

This screen provides the teacher with a comprehensive overview of all course-related activities in an organized and hierarchical structure. The screen displays all approved courses assigned to the teacher, where each course can be expanded to show its weekly content. Courses are presented as interactive cards, allowing the teacher to easily navigate between different course materials.

Within each course, the content is divided into weeks based on the course duration. Each week can be expanded individually to display its related educational components. For every week, the teacher can view lessons, tasks, and tests associated with that specific period. Lessons and tasks are interactive, allowing the teacher to open detailed views for each item, while tests are displayed for reference and tracking purposes.

This screen helps the teacher monitor the academic flow of each course, ensuring that lessons, tasks, and tests are properly distributed across the weeks. The expandable design improves clarity and usability, making it easier for the teacher to manage course content and follow student progress throughout the semester

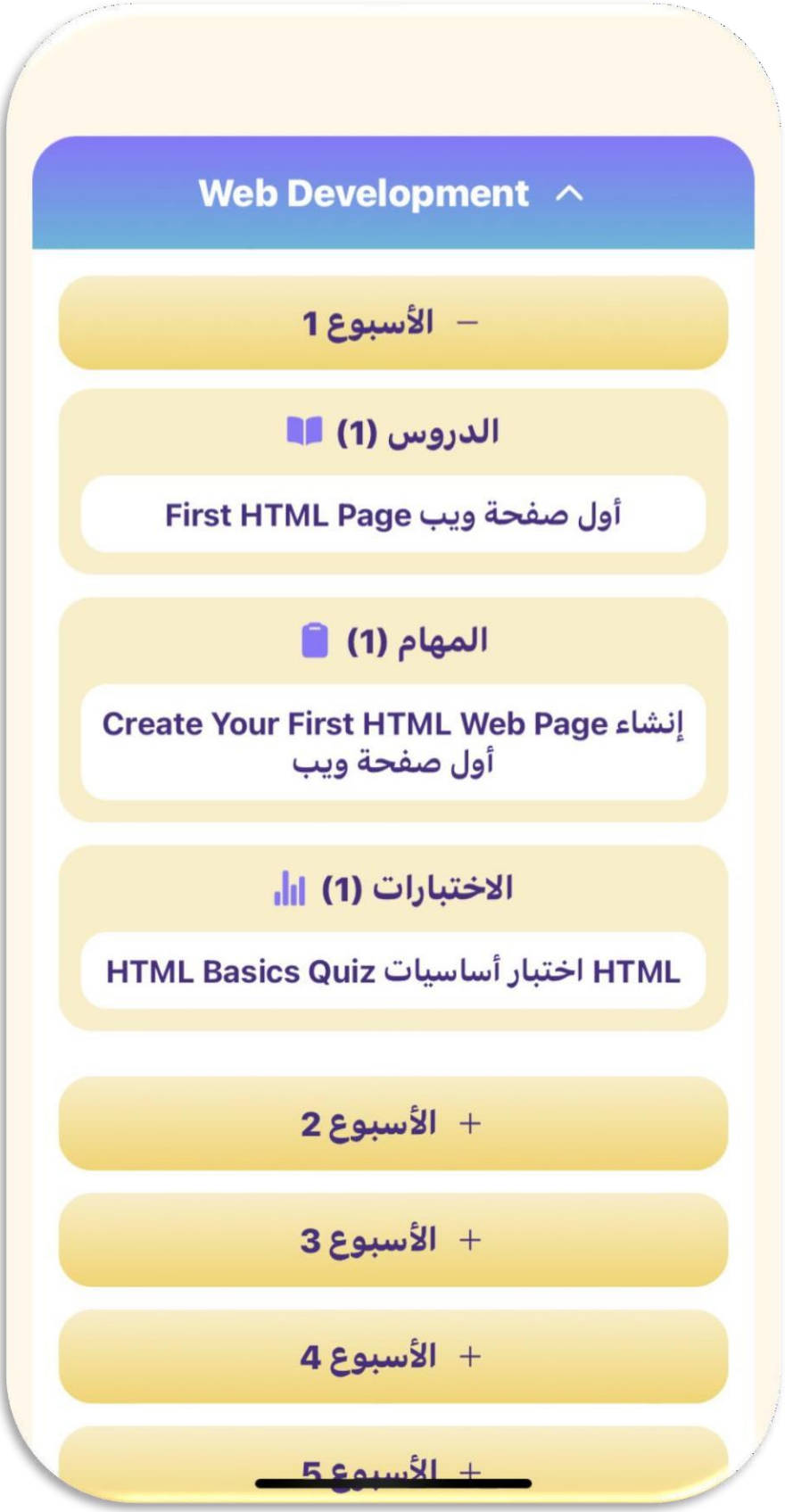


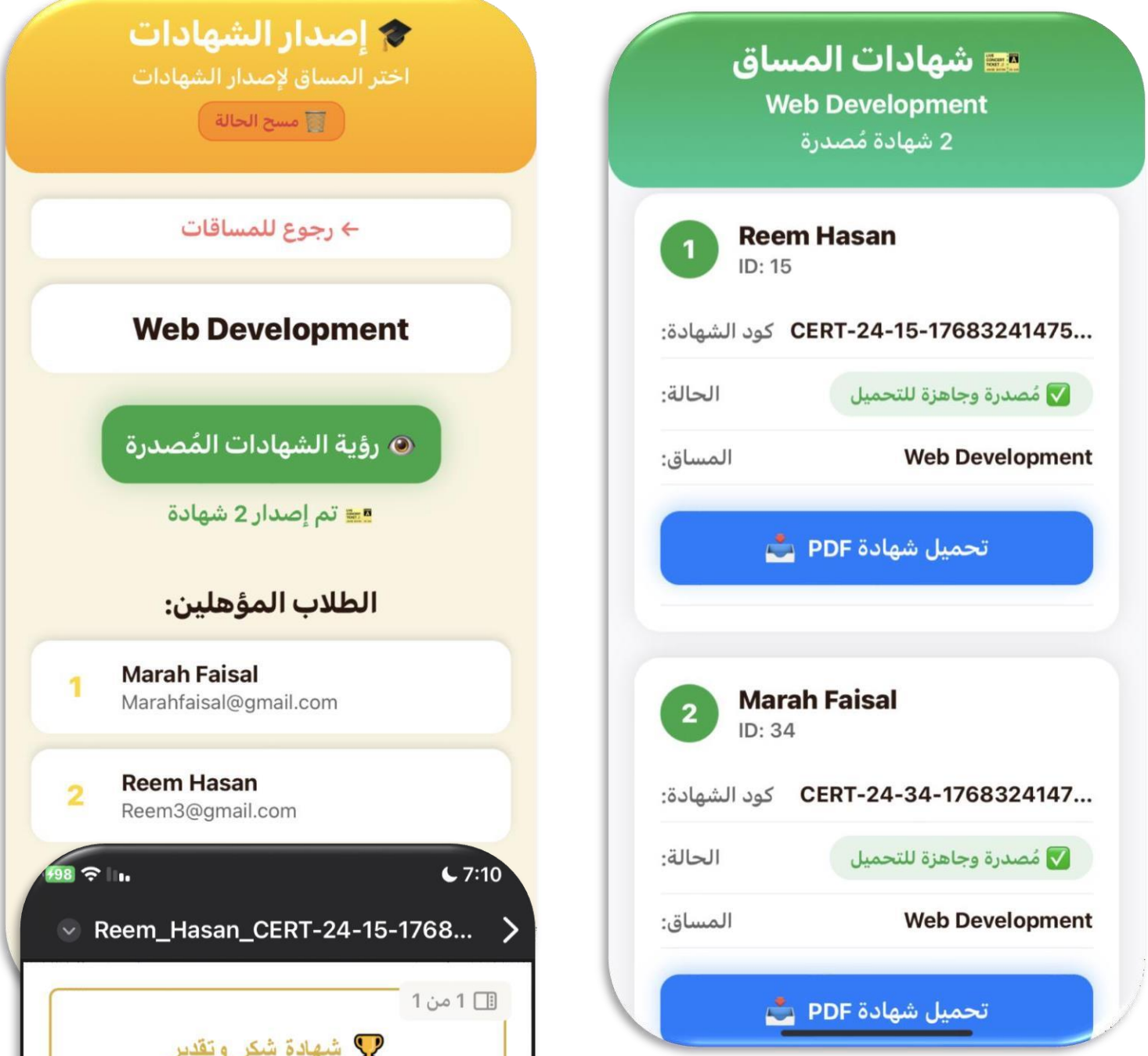
Figure 38: Course Content Overview Screen

Course Certificates Management

The Certificates Management screen allows the teacher to fully manage the process of issuing course certificates. Upon accessing the screen, all approved courses assigned to the teacher are displayed, along with a clear indication of whether certificates have already been issued for each course. The teacher can select a course to view the enrolled students or initiate the certificate issuance process if it has not been completed before.
When the teacher confirms the issuance, certificates are generated collectively for all eligible students in the selected course, and the issuance status is stored locally to prevent duplicate issuance for the same course. Additionally, a real-time notification is sent to each student to inform them that their course certificate has been successfully issued. Once the process is completed, the teacher can navigate to the course certificates screen.

The Course Certificates screen displays all certificates issued for a specific course. Each certificate includes clear student information such as the student's name, student ID, certificate code, and certificate status. This screen enables the teacher to download each student's certificate individually in PDF format, with the option to share the file immediately after downloading. Various download states are handled gracefully, including loading and error cases, with informative messages displayed to the user. Together, these screens provide an organized and efficient workflow for issuing, managing, and accessing course certificates in a clear and user-friendly manner

Figure 39: Course Certificates Screen



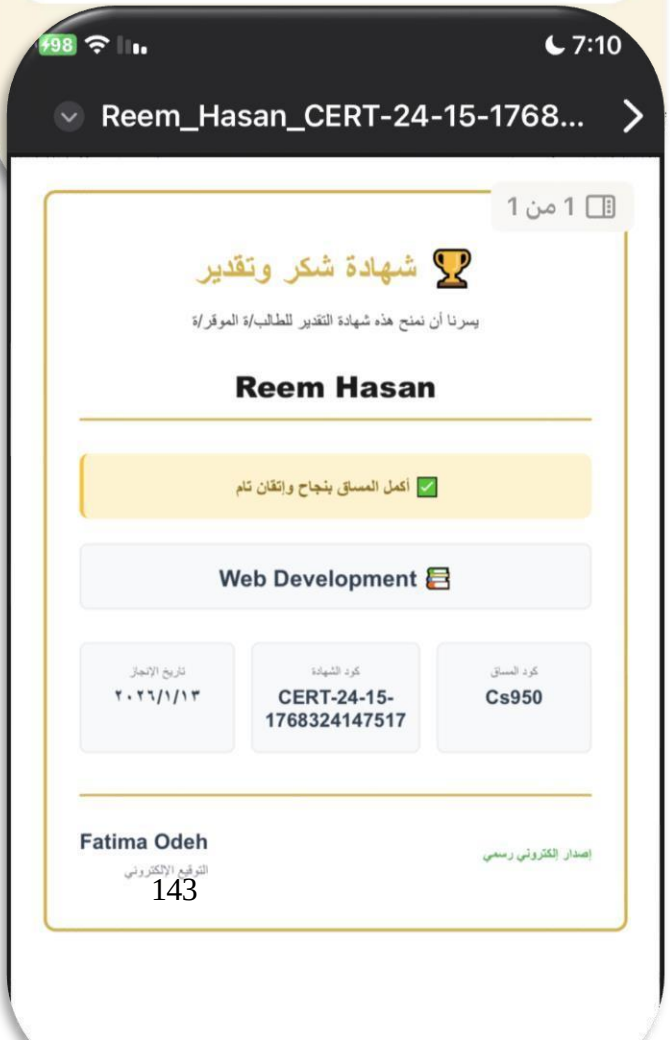


Figure 40: View Certificates Screen

Task Submissions Screen

This screen allows the teacher to manage and review students' submissions for course tasks. The teacher starts by viewing a list of assigned courses, then selects a specific course to display its available weeks. For each week, the teacher can view the tasks published during that period. Upon selecting a task, all related student submissions are displayed, including submission date, student name, and submission status. The teacher can open attached files or review written answers directly within the application. The screen also enables the teacher to evaluate submissions by assigning grades and providing feedback through a dedicated grading interface. All updates are saved immediately, ensuring accurate tracking of student performance and efficient assessment management.

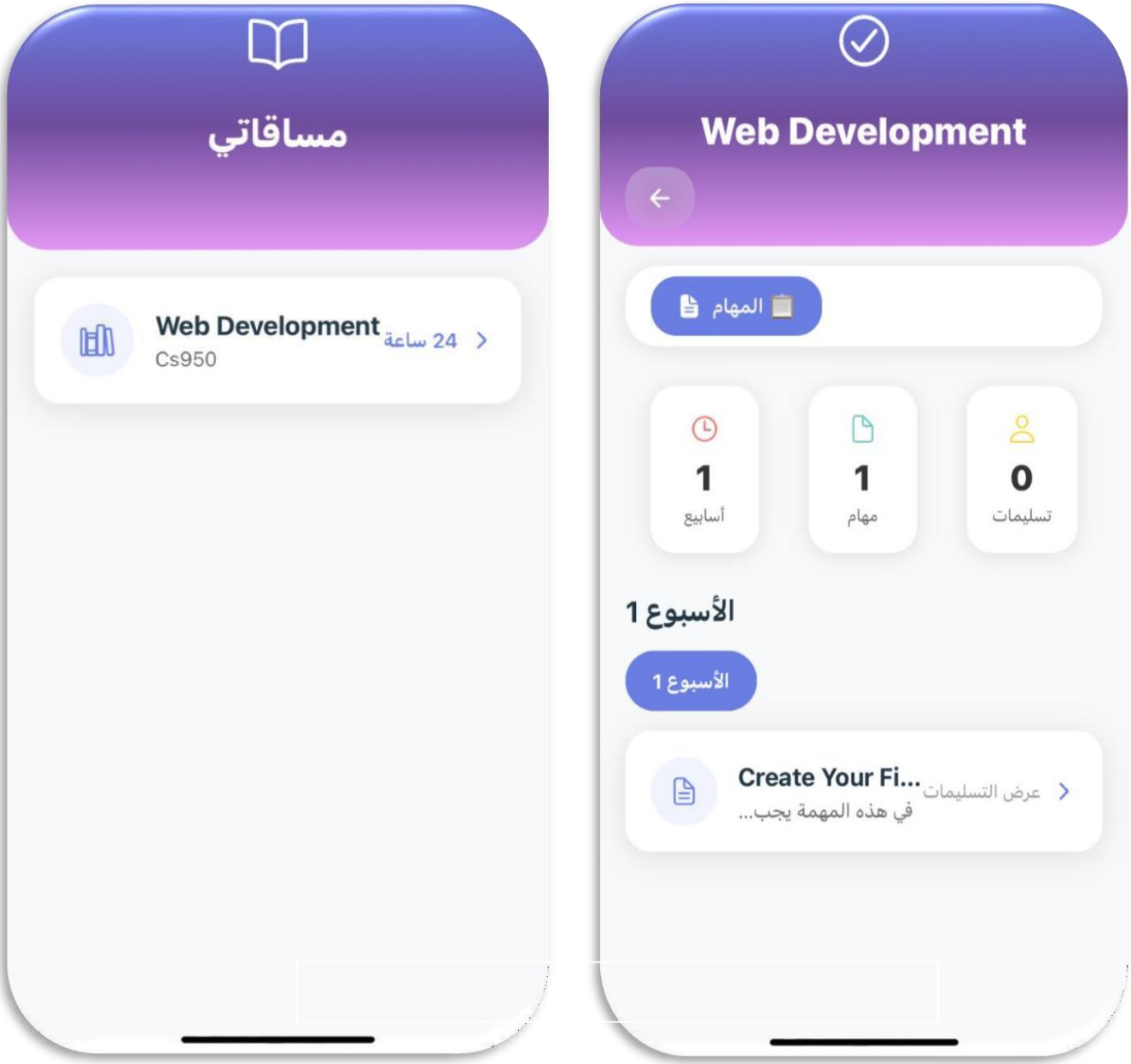
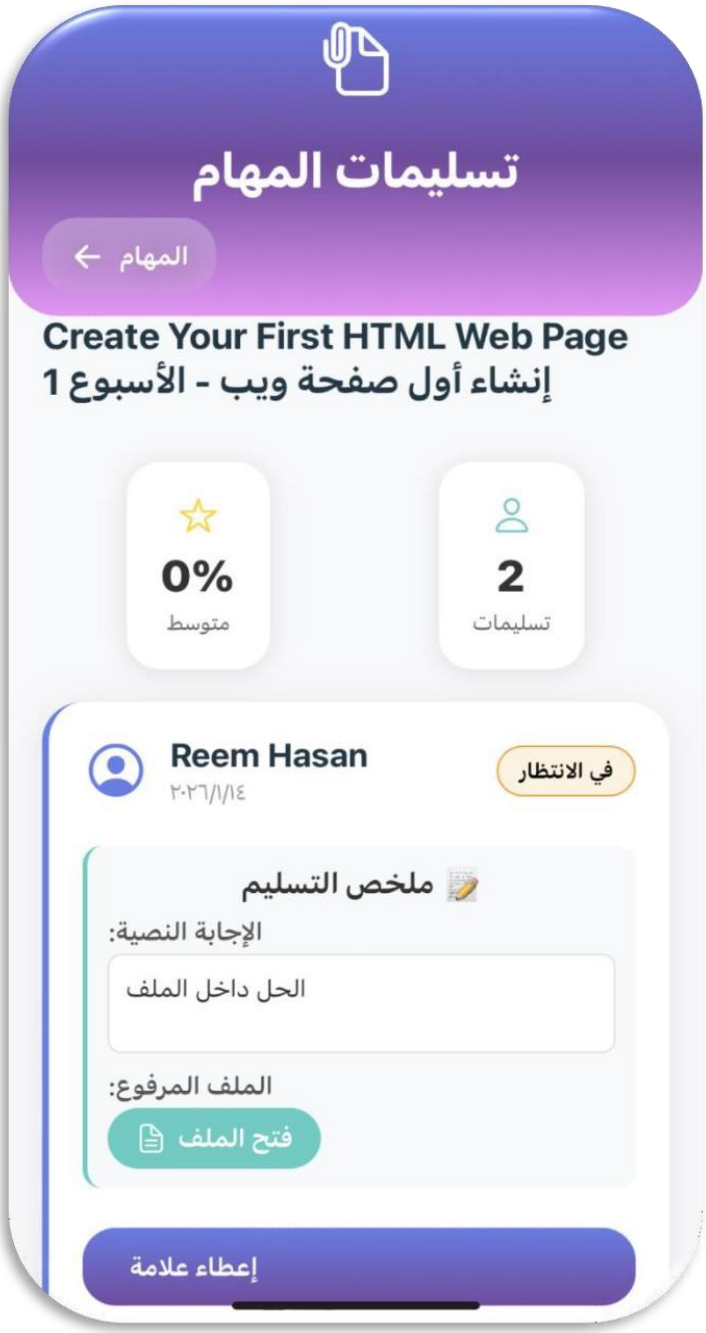


Figure 41: Task Submissions Screen



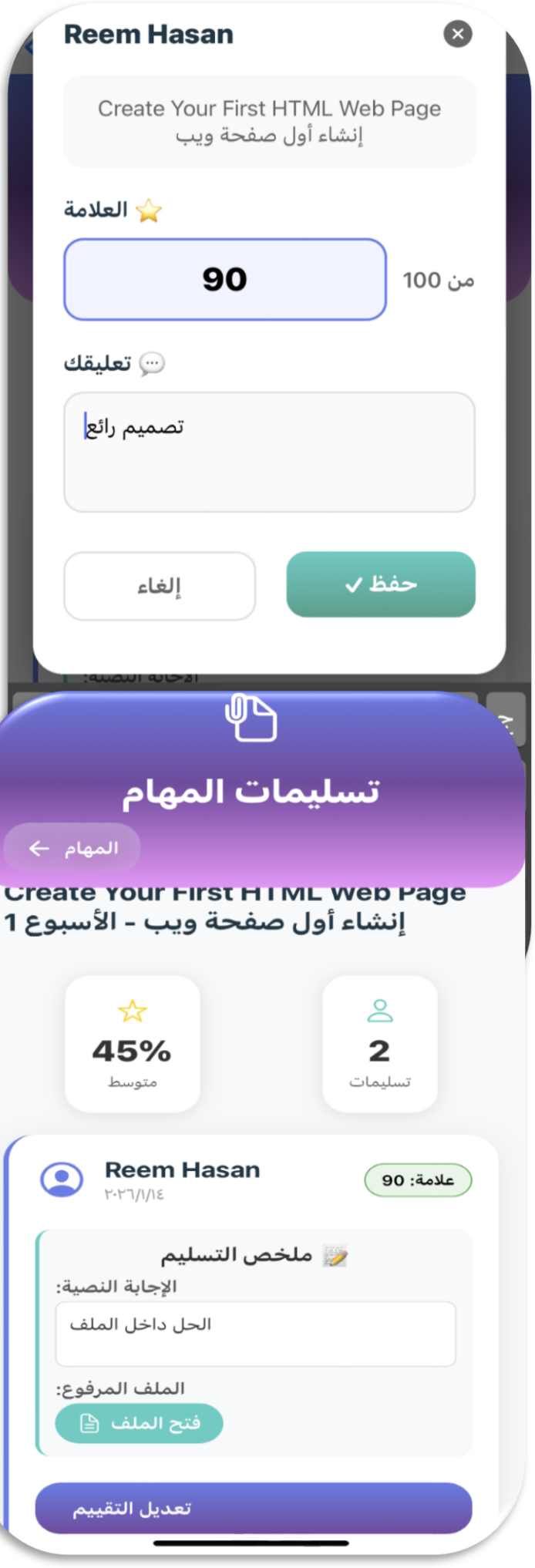


Figure 42: Add Grades and feedback

Figure 43: Notifications Screen

Teacher Profile Screen

The Profile Screen allows the teacher to view and manage their personal account information in a dedicated interface. It displays the teacher's full name, email address, and profile picture. The teacher can update the profile picture by selecting a new image from the device gallery, and the selected image is saved locally to ensure it remains available after reopening the application.

The screen also provides the ability to edit the email address. When the teacher updates the email, the new value is validated and sent to the server, then stored locally to maintain consistency across sessions. This ensures that the teacher's contact information is always up to date within the system.

For security purposes, the Profile Screen includes a password change feature. The teacher can enter the current password and set a new one through a secure modal interface with proper validation. In addition, the screen allows the teacher to log out of the application safely, clearing stored session data and redirecting to the login screen

Figure 44: Teacher Profile Screen

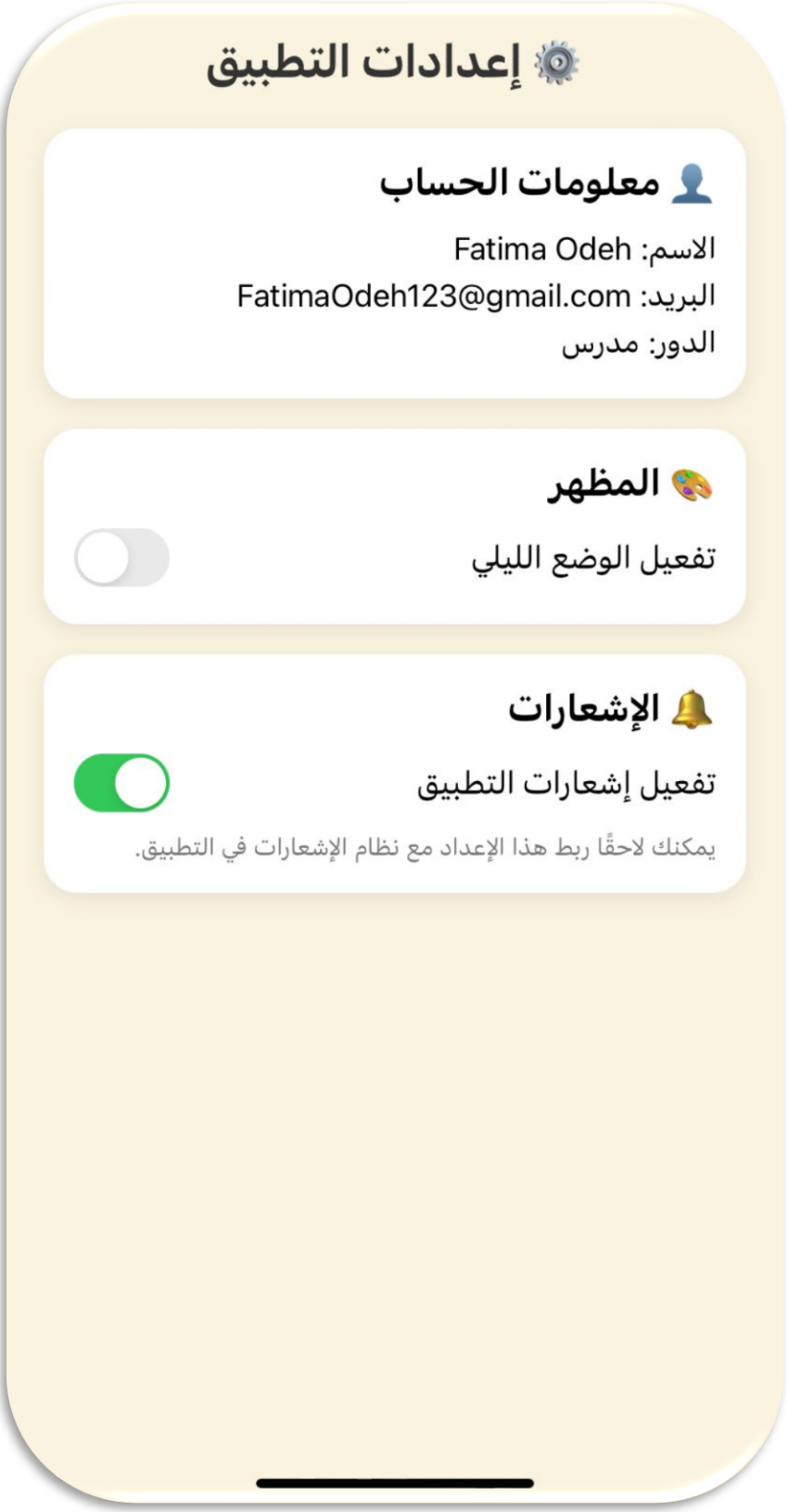


Figure 44.1: Settings

3.2.3 Student Dashboard

Course Enrollment and Payment Interface

The Course Selection screen allows students to browse all available courses in an organized and user-friendly manner. Upon accessing the screen, only approved courses are displayed, including the course name, instructor name, price, and a brief description. The screen enables students to switch between available courses and the courses they are already enrolled in, providing clear and structured course management.

The screen verifies the payment status of each course, allowing students to access paid courses directly while prompting payment for unpaid ones. If a course requires payment, the student is redirected to the Payment screen to complete the enrollment process. Paid courses are visually marked to clearly indicate enrollment status.

The Payment Screen is responsible for securely completing the course payment process by displaying the course title and required amount and allowing the student to enter card details to finalize the payment. The application uses Stripe as a secure payment gateway, where the payment process is implemented using the Payment Intent model to ensure that all sensitive payment operations are handled on the backend server rather than the mobile application. Stripe applies multiple security layers through different types of keys: the Publishable Key, which is used in the React Native application to initialize Stripe and collect card information, and the Secret Key, which is stored on the backend and used to create payment intents and communicate with Stripe's API. For each transaction, a temporary Client Secret is generated and sent to the application to securely confirm the payment. Card details are collected using Stripe's official CardField component, which transmits the data directly to Stripe and ensures compliance with PCI-DSS security standards. Once the payment is successfully completed, the system updates the student's enrollment status in the database and automatically redirects the student to the course page to begin learning. In addition, Stripe Webhooks are used to verify successful payments and update the database even if the application is closed or the network connection is interrupted, ensuring a smooth, secure, and reliable payment experience.

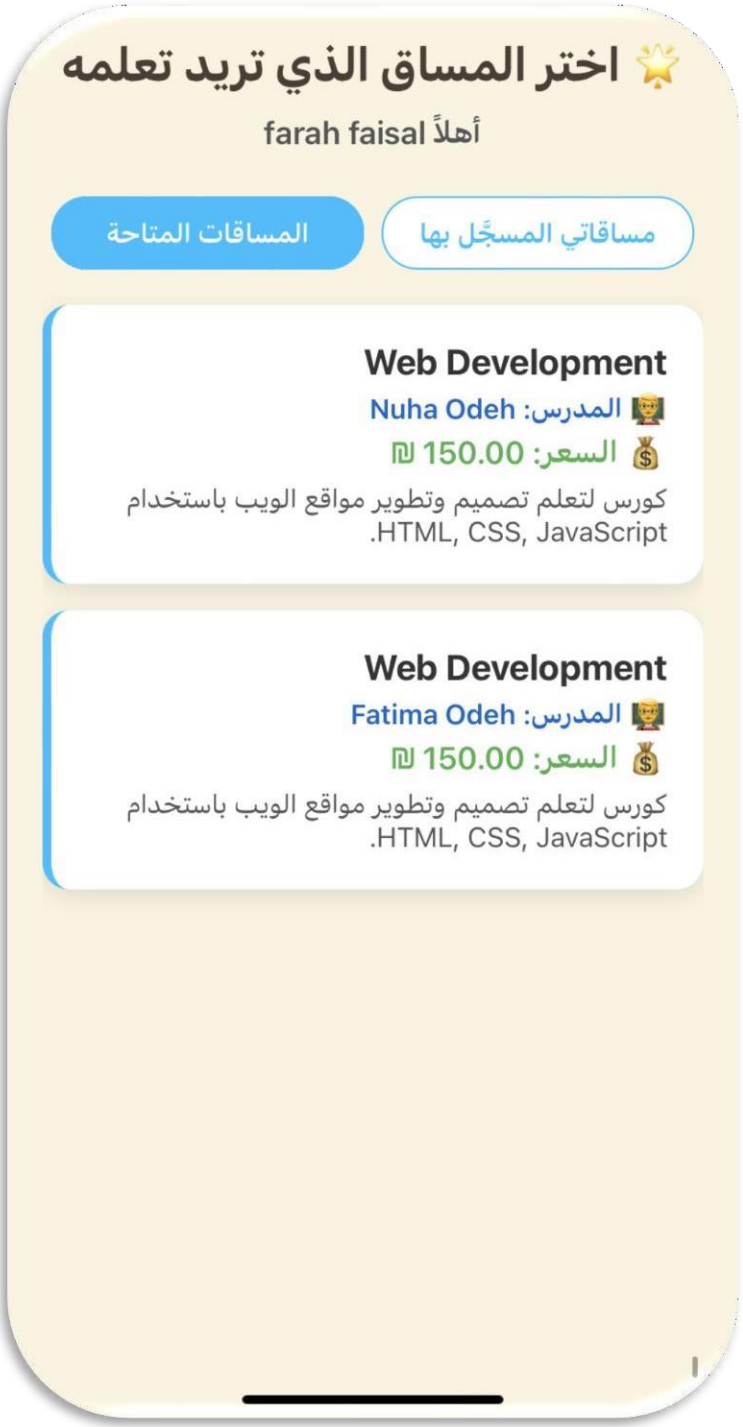


Figure 45 : Course Enrollment Screen

Figure 46: Payment Interface Screen

Learning Style Selection Screen

This screen allows the student to choose a preferred learning style before accessing the course dashboard in order to personalize the learning experience. The student's name and course name are displayed for clarity. The student can select one learning style from visual, auditory, or practical, each explained with a brief description. The system requires a selection before continuing and displays a warning if no option is chosen. After confirmation, the selected learning style is passed to the dashboard along with the student and course information, with the option to change it later from the settings.

Figure 47: Learning Style Selection Screen

Dashboard Screen

The Dashboard screen is the main interface of the educational app, providing the student with a personalized and interactive experience. It displays a welcome message, a daily motivational quote, and learning cards tailored to the student's selected learning style (visual, auditory, or practical). The screen includes a sidebar menu with navigation to courses, summaries, tasks, exams, messages, certificates, live sessions, statistics, and settings.

Notifications are dynamically fetched from Firebase and can be viewed, marked as read, or deleted using swipe gestures. A modal shows "Motivation of the Day" with an option to refresh for a new quote. The bottom navigation bar allows quick access to home, profile, statistics, and settings. Animations are used throughout to enhance user experience.

Figure 48:Dashboard for Practical Learning

Figure 48.1:Dashboard for Practical Learning

Figure 49: Dashboard for Visual Learning

Figure 50: Dashboard for Auditory Learning

Student Course Plan Screen

This page is used to present the course study plan to the student, allowing them to view the overall structure and content of the course throughout the semester. The displayed plan is based on the data added by the teacher and includes a weekly breakdown of the course with learning objectives, content, and evaluation methods. This page helps students understand course requirements and track their academic progress in a clear and organized manner

Figure 51 : Student Course Plan

Courses Overview

This page is used to display the courses that the student has enrolled in and paid for, allowing them to view all their registered courses in one place. The page retrieves the student's courses and presents essential information such as the course name and teacher name. It also enables the student to select a specific course in order to choose or associate an appropriate learning style, which will later be used to personalize the learning experience. The main purpose of this page is to help students manage their enrolled courses and organize their learning process in a simple and clear way

Figure 52: Courses Overview

Course Payment

The Choose Course Screen allows students to view available courses with their basic information such as course name, teacher, description, and price. When a course is selected, the system checks the payment status; free or previously paid courses allow direct access, while unpaid paid courses redirect the student to the payment screen. This screen simplifies course selection and payment management in a clear and user-friendly way.

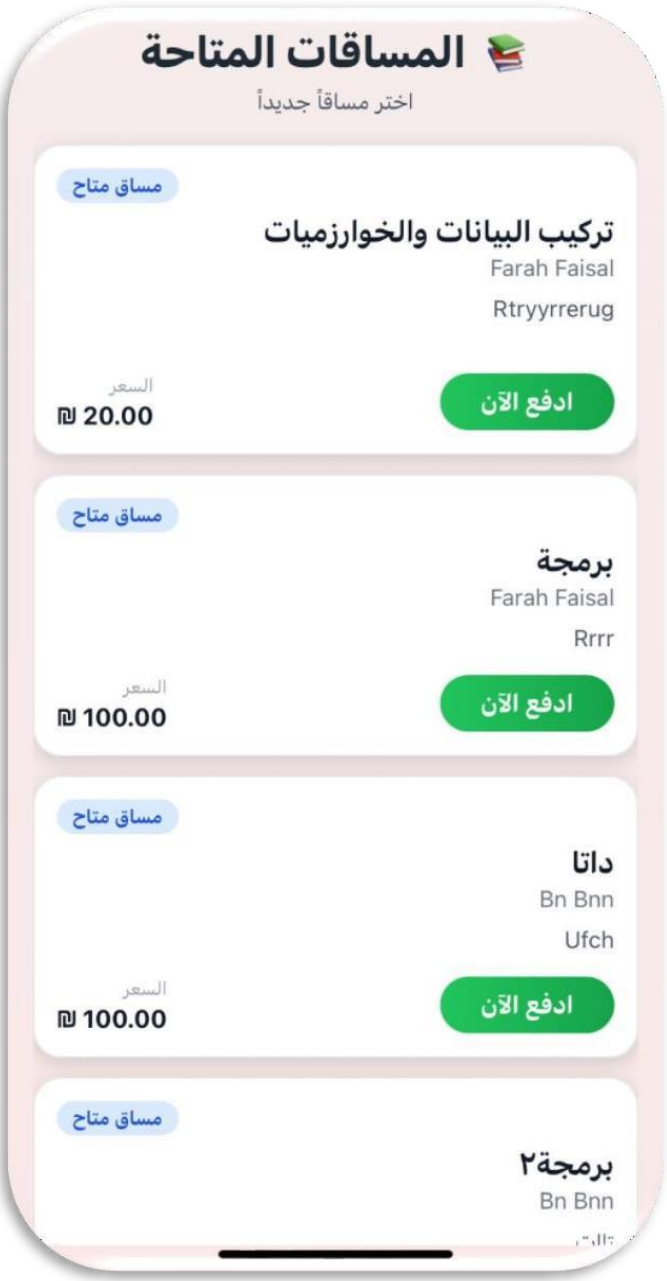


Figure 53: Course Payment

Student Lessons

The Student Lessons Screen allows students to select a week to view lessons according to their chosen learning style. Once a week is selected, the available lessons are displayed, including images, videos, or audio files. Students can view content directly or open files via external links. This screen provides a flexible and user-friendly learning experience, allowing smooth navigation between weeks and lessons.

Figure 54 :Student Lessons

StudentTasksScreen

The Student Tasks Screen allows students to view weekly tasks for each course. Students can select a week to see the associated tasks and access details such as title, description, due date, and status.

The screen supports multiple submission options: uploading a file, writing a text answer, or both. After submission, a notification is sent to the teacher. Students can edit or delete submissions before the deadline and view grades and feedback after evaluation .

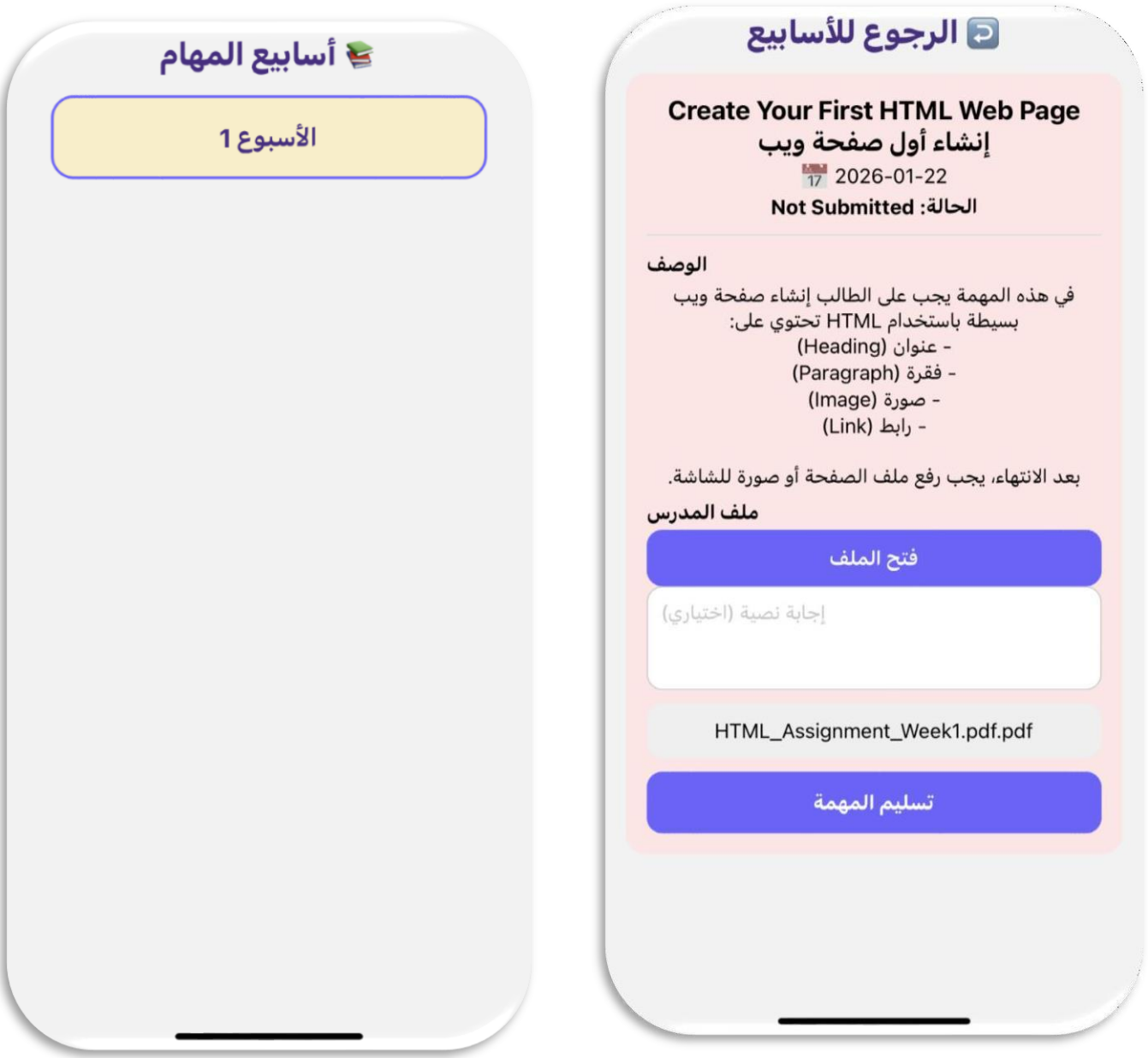


Figure 55: Student Tasks

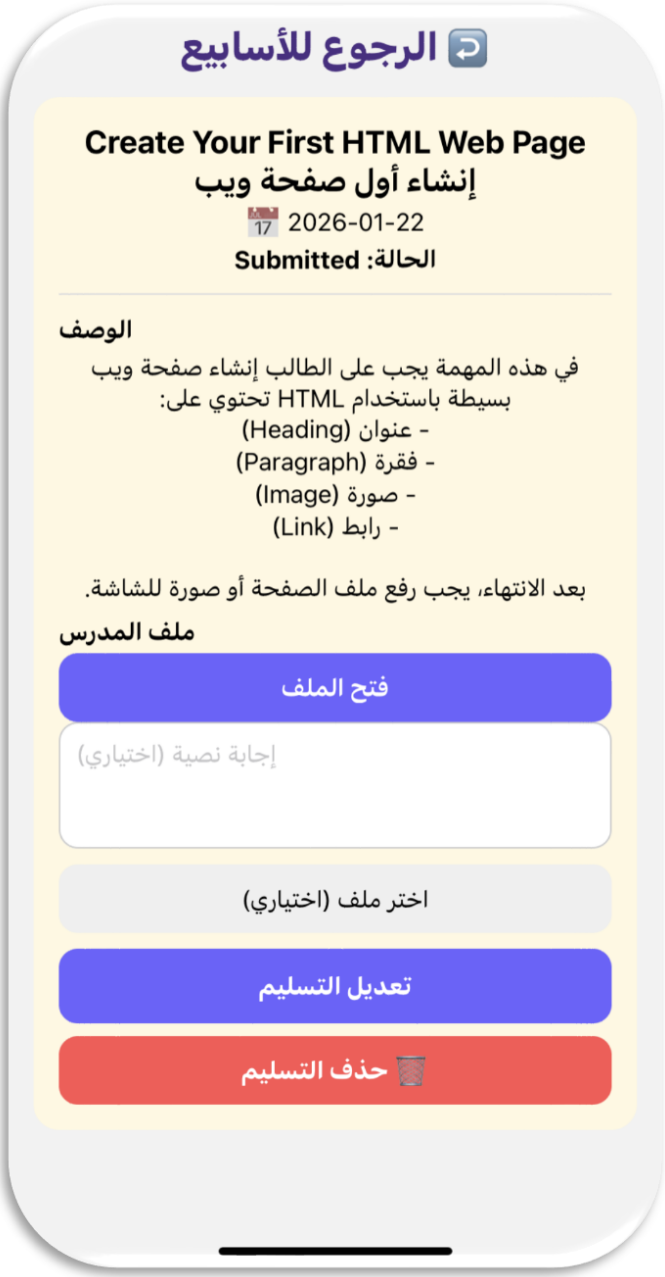


Figure 56 : Edit or delete Task

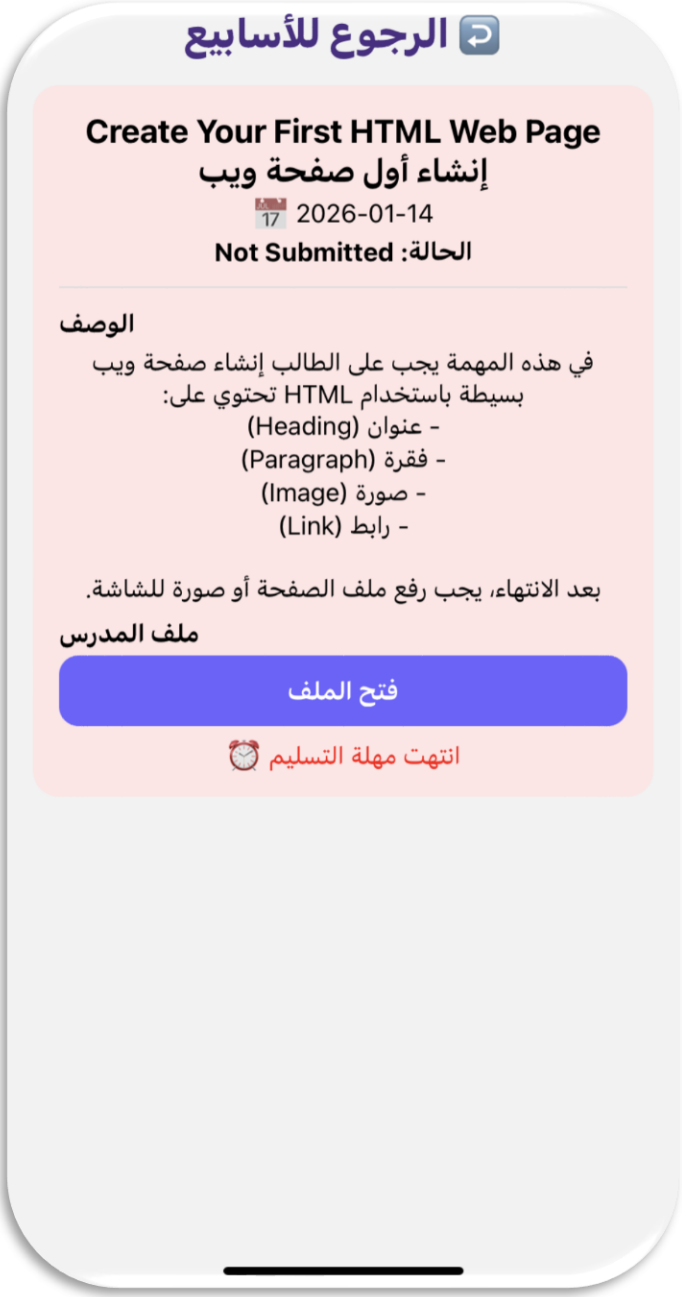


Figure 57: Task Not Submitted

Figure 58 : grade and feedback task

File Summary Screen

The SummaryScreen allows users to select text files in TXT or DOCX formats and automatically extract and summarize their content using **Hugging Face AI models**. After selecting a file, the raw text is extracted and split into smaller chunks to avoid exceeding the input limits of the models. The text language is automatically detected to choose the appropriate model, using moussaKam/arabic-summarization for Arabic texts and facebook/bart-large-cnn for English texts. Each chunk is then sent to the model via the Hugging Face API to generate its summary, and all summaries are combined into a final text displayed in a user-friendly scrollable interface, showing the processing status. This provides a smooth and reliable experience for quickly generating summaries of large texts directly on the device.

Figure 59: Summary Screen

AI Assistant Screen

The AI Assistant Screen provides an interactive interface for users to communicate with an AI by typing questions or messages. Submitted messages are sent through the OpenRouter API, which accesses the **GPT-3.5-turbo model from OpenAI** to generate real-time responses. Messages appear in colored chat bubbles to distinguish between user and AI, with a loading indicator shown while processing. The screen also supports automatic scrolling to the latest messages and easy chat management, including clearing the conversation, delivering a fast, responsive, and intelligent user experience.

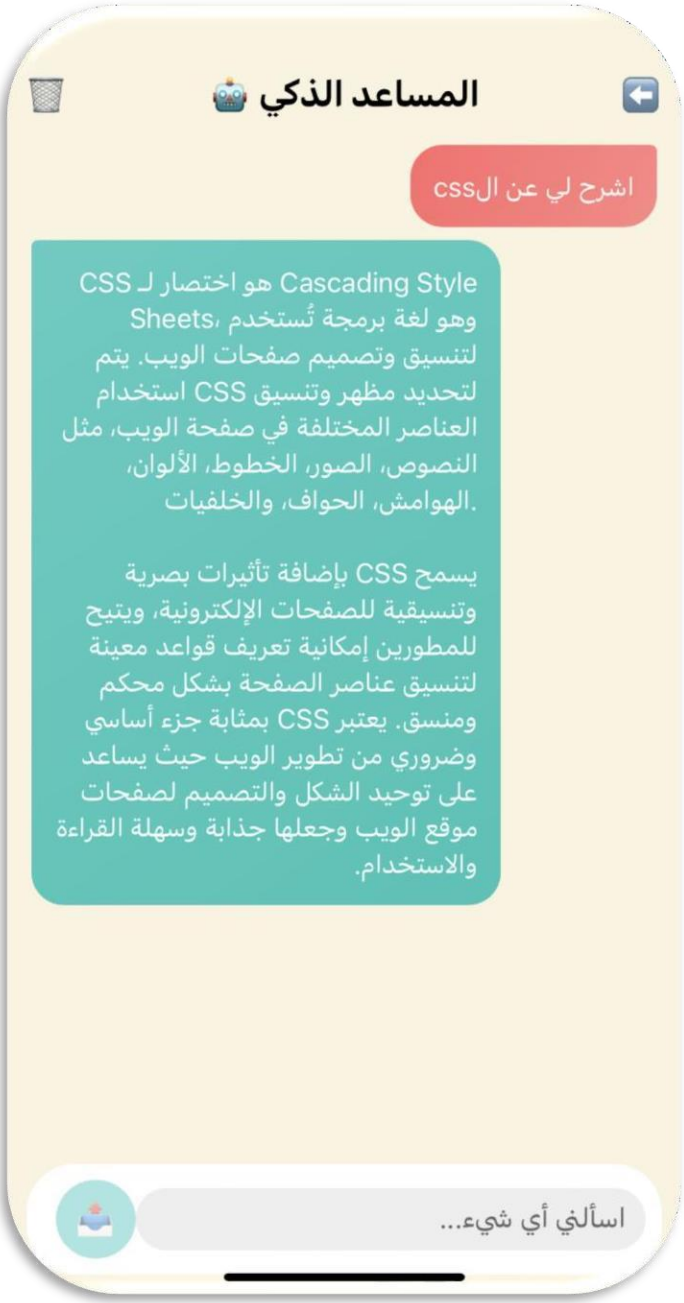


Figure 60: AI Assistant Screen

Live Sessions Screen

The Live Sessions Screen displays all live and upcoming online sessions for a specific course. Students can view session details, see the number of currently live sessions, refresh the list, and join live classes directly using the meeting link.

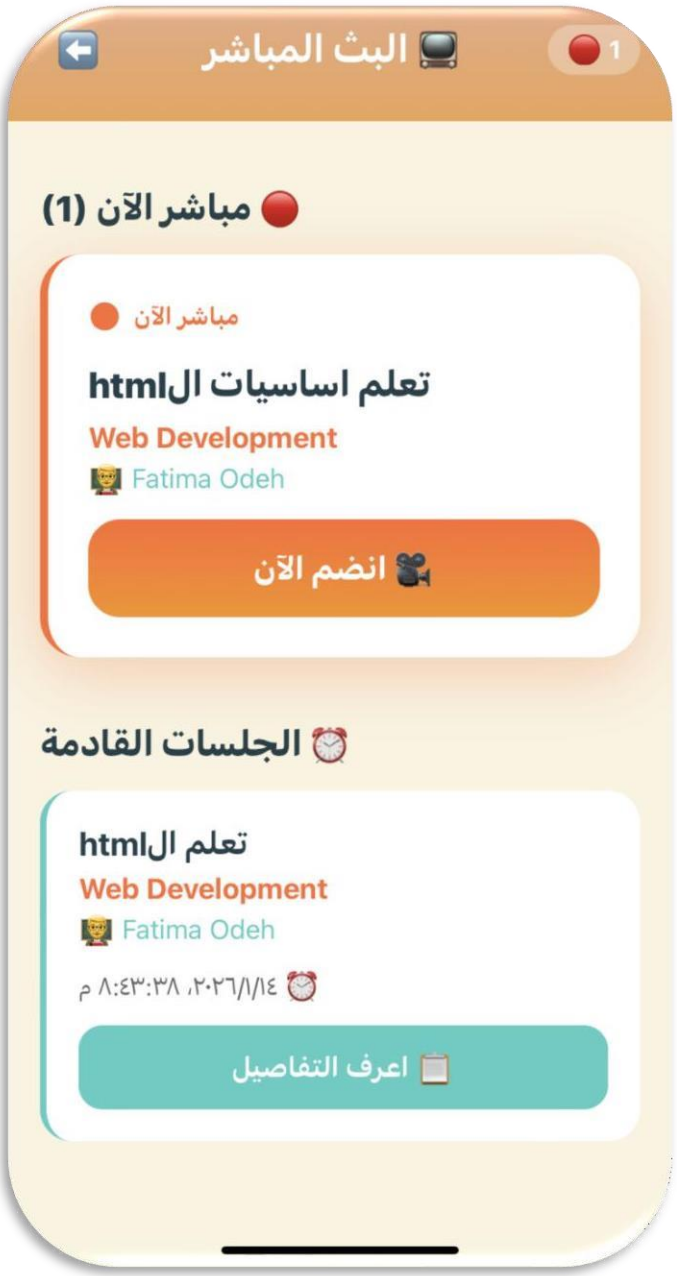


Figure 61: Live Sessions Screen

Student Messaging System

The Student Messaging System allows real-time communication between students and teachers. It includes a conversations screen that displays all available chats with teachers, showing the last message and supporting search functionality. A messages screen enables sending and receiving messages instantly using Firebase Firestore, with clear separation between sent and received messages, timestamps, and automatic notifications sent to teachers upon message delivery.

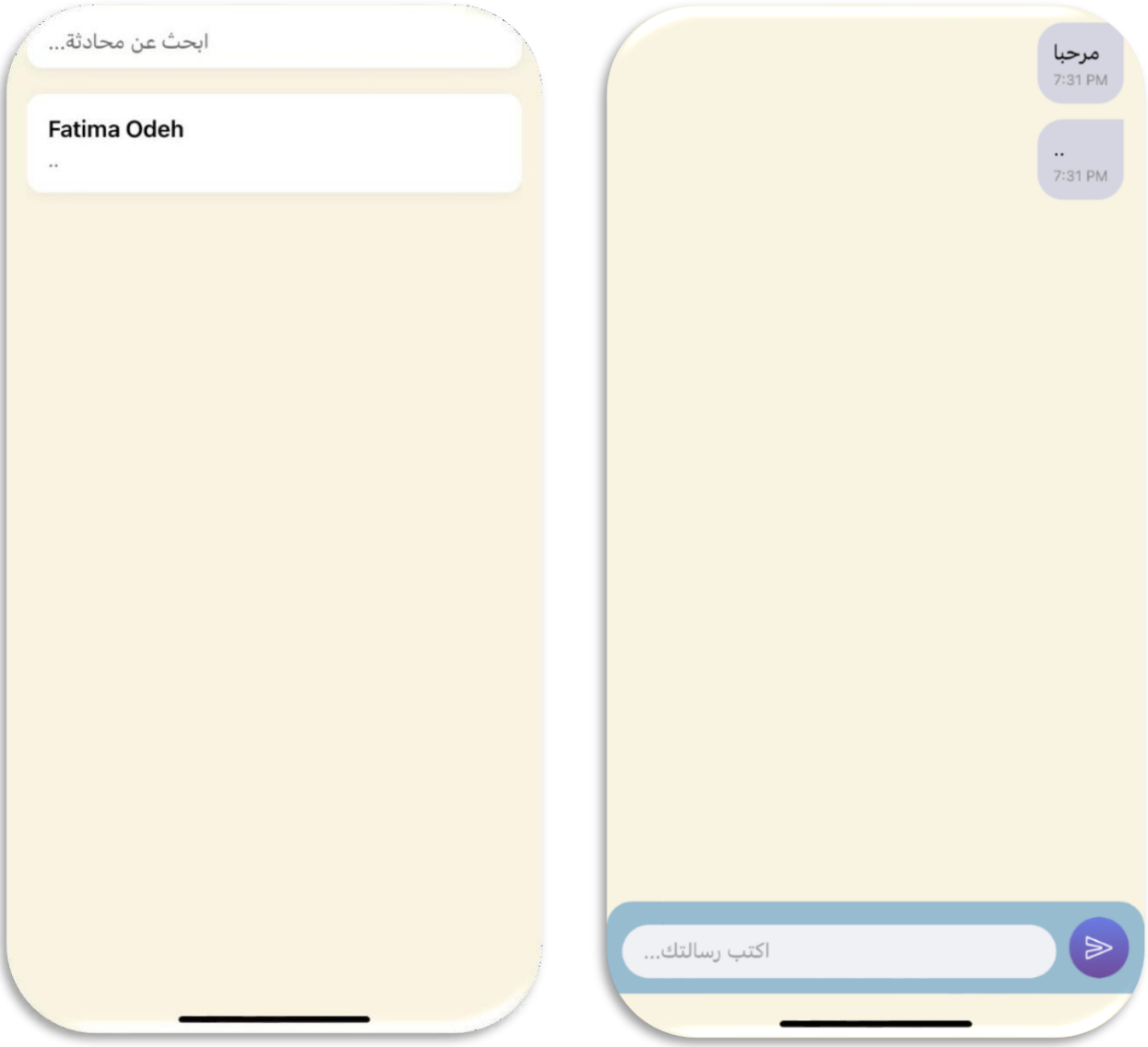


Figure 62: Student Messaging System

Student Tests & Submissions Screen

This screen allows students to view all available tests for a specific course, including start and end times and submission status. Students can answer different question types such as text, multiple-choice, and file upload questions. The system prevents access before the test starts and blocks submissions after the deadline. After submission, answers are saved, a confirmation message is shown, and a notification is automatically sent to the course teacher. Once grading is completed, students can view their grades and total score.

Figure 63: Quiz Screen

Figure 64: View grades

Practical Learning AI Screen

This screen is designed to support practical learning style by generating complete hands-on activities using **OpenAI's** artificial intelligence. The user enters a topic, and the system automatically creates a structured practical lesson that includes a clear title, required materials, detailed step-by-step instructions, a challenge to assess understanding, and an estimated completion time. This approach helps learners apply theoretical concepts through real-world practice, enhancing engagement and comprehension

Figure 65: Practical Learning AI Screen

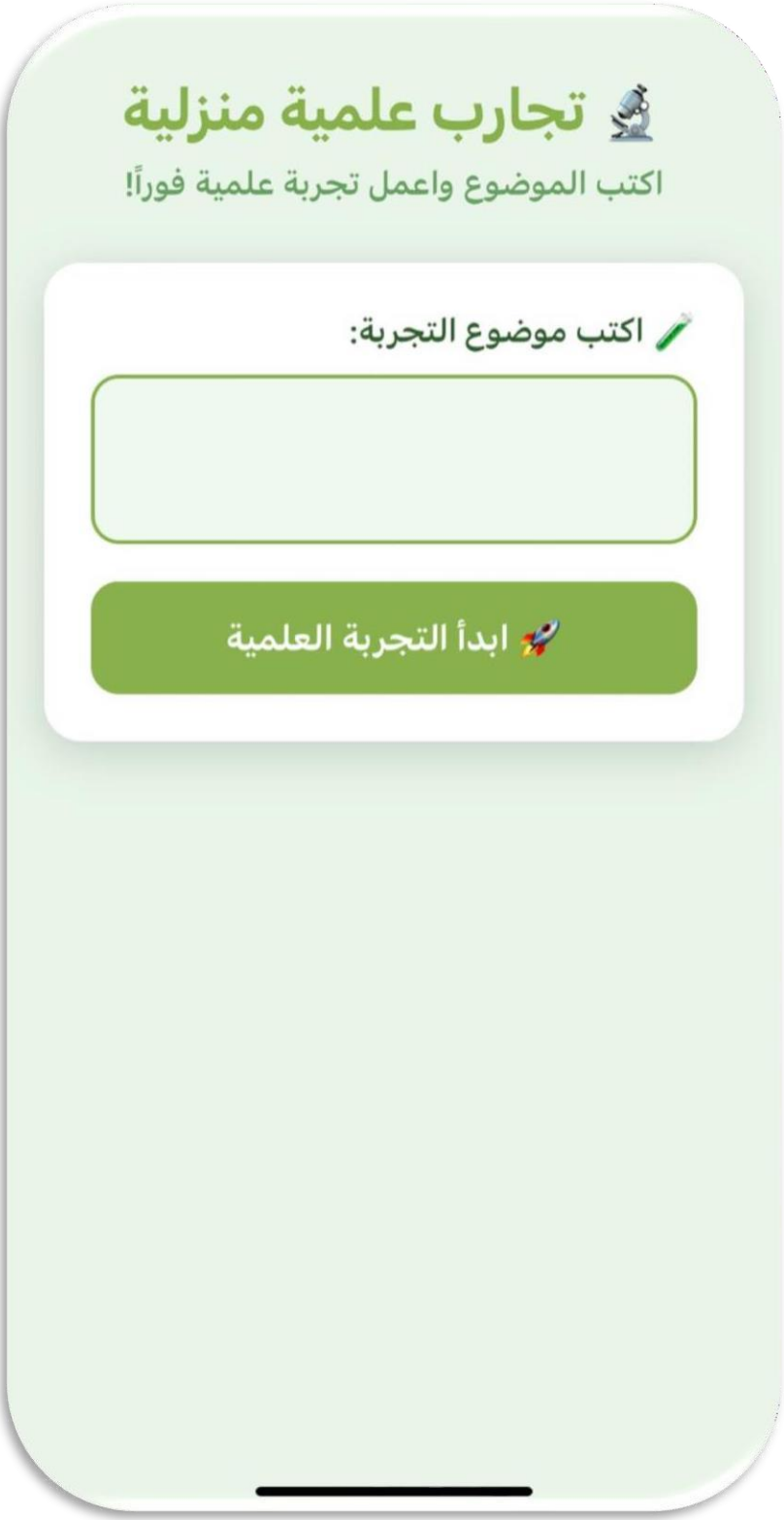


Figure 66: Practical Learning AI Screen

Dynamic Quiz Screen

The Dynamic Quiz Screen generates interactive multiple-choice quizzes using artificial intelligence. The user enters a specific topic, and the system automatically creates five quiz questions with four options each using **OpenAI's GPT-3.5-turbo model**. After answering the questions, the system evaluates the answers, calculates the final score, and displays feedback with a grade indicator. This screen enhances active learning, self-assessment, and engagement through instant feedback.

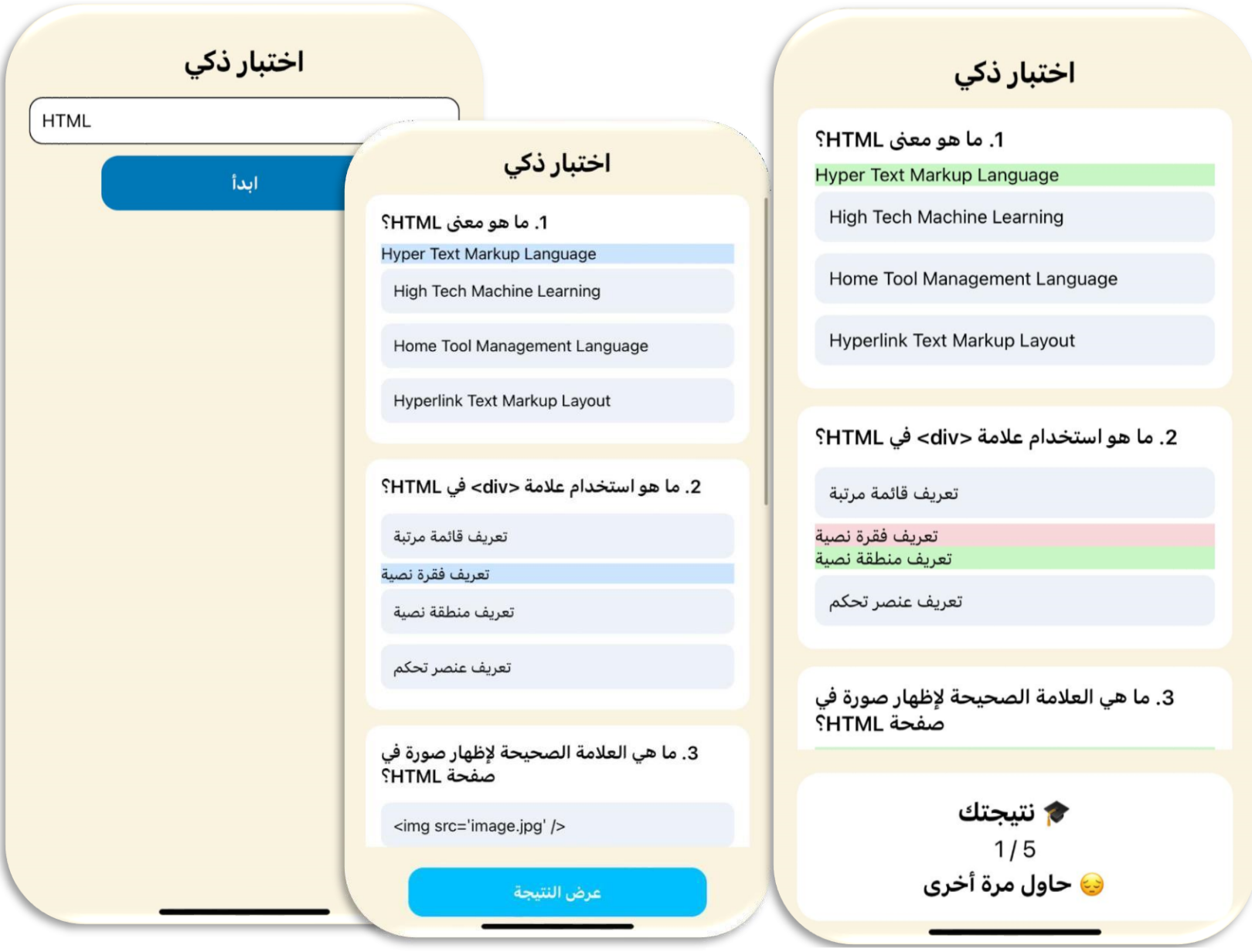


Figure 67: Dynamic Quiz Screen

Auditory Listen Screen

The Auditory Listen Screen provides an interactive audio-based learning experience powered by **OpenAI's** artificial intelligence. Users enter a topic, and the AI generates a clear explanation in text form. This explanation can then be converted to speech using text-to-speech technology, implemented for both mobile devices via **Expo Speech**. This allows users to listen instead of reading, supporting auditory learners and enhancing accessibility and understanding across platforms.

Figure 68: Auditory Listen Screen

Auditory Repeat Screen

The Auditory Repeat Screen allows users to record their own voice in order to reinforce learning through repetition. Users can start and stop audio recordings, save them locally, play them back, or delete them when needed. This feature helps improve memorization, pronunciation, and active learning by encouraging learners to repeat information aloud.

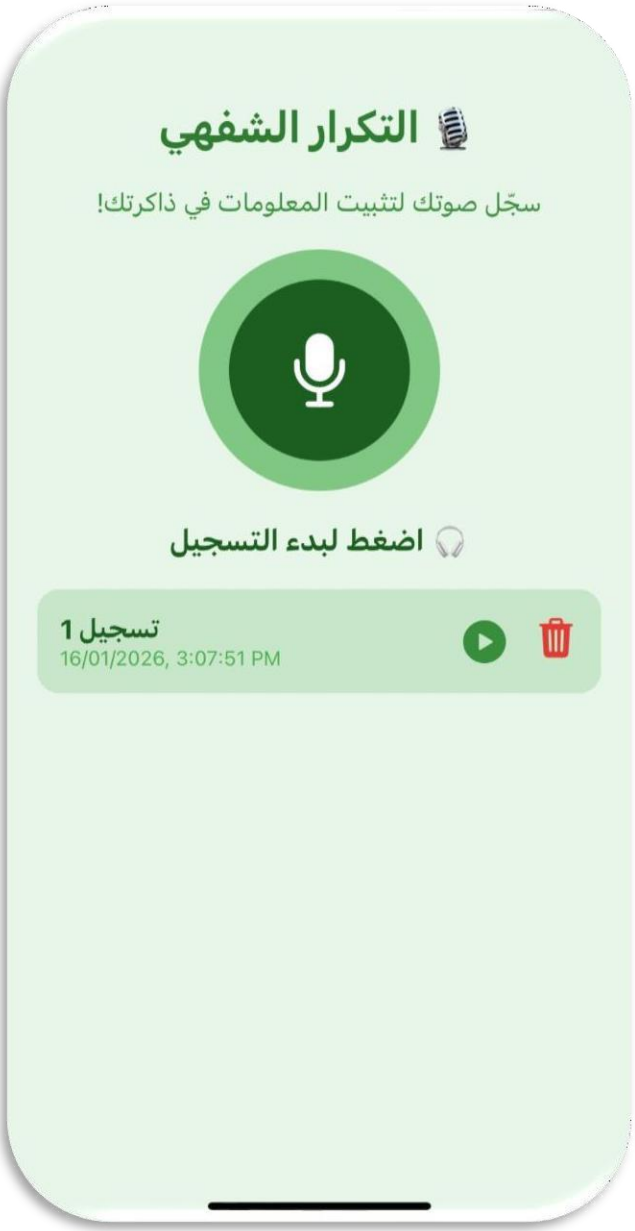


Figure 69: Auditory Repeat Screen

YouTube Learning Screen

The YouTube Learning Screen allows users to search for educational videos directly from YouTube by entering a topic name. The screen fetches relevant videos using the **YouTube Data API** and displays them with an embedded video player, enabling users to watch educational content without leaving the application. This feature supports visual learning and helps students understand topics through real video explanations

Figure 70: YouTube Learning Screen

Mind Map Fun Screen

The Mind Map Fun Screen allows users to create interactive mind maps to organize and visualize their ideas. Users can draw freehand, add shapes (circles, rectangles), insert text, erase elements, and save or load their maps. The screen uses touch gestures for drawing and moving elements, providing a flexible and creative learning tool that supports visual and kinesthetic learning styles.

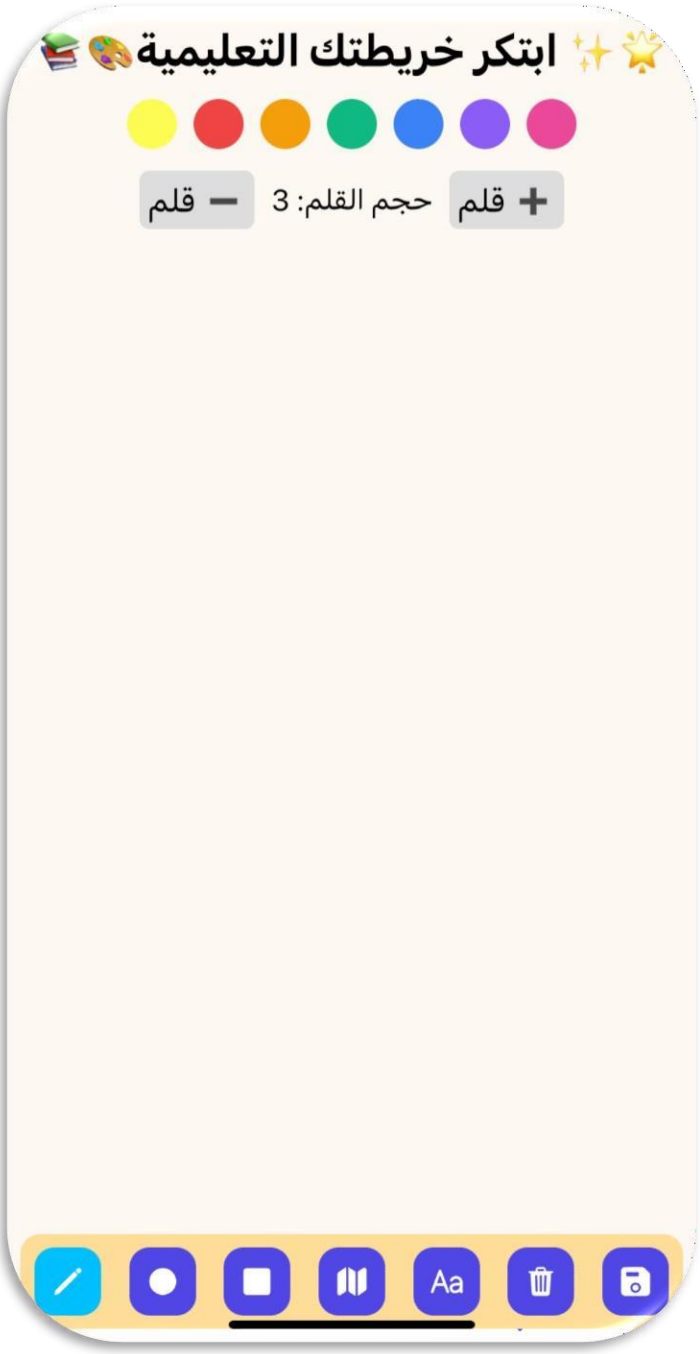


Figure 71: Mind Map Fun Screen

Daily Study Plan Screen

The Daily Study Plan Screen allows students to organize and manage their daily study tasks efficiently. Users can add subjects, tasks, and scheduled times for each day of the week, mark tasks as done, delete tasks, and filter them by status (all, completed, pending). The screen shows progress statistics, supports notifications for new tasks

Figure 72: Daily Study Plan Screen

Student Certificates Screen

The Student Certificates Screen allows students to view, download, and share their course certificates in PDF format. It fetches certificates from the server, shows details like course name, issue date, and certificate code, and supports downloading and sharing via device storage.

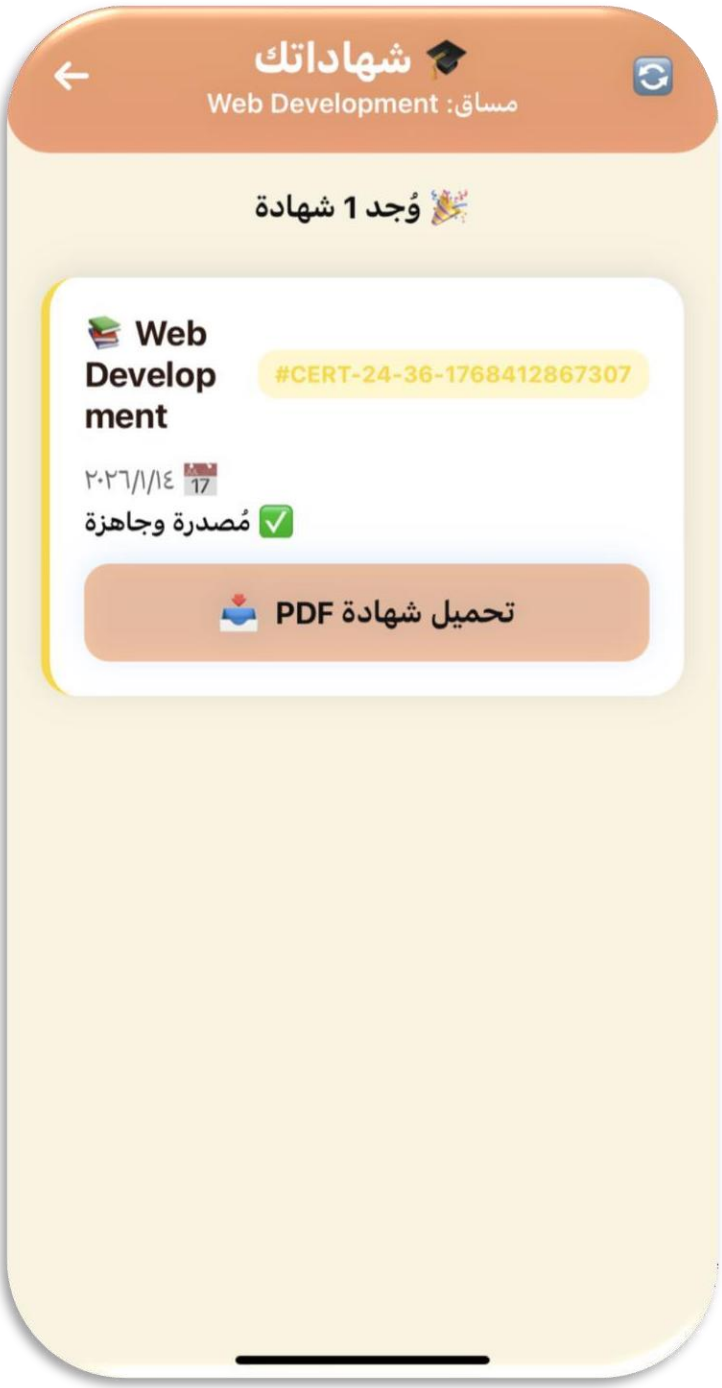


Figure 73: Student Certificates Screen

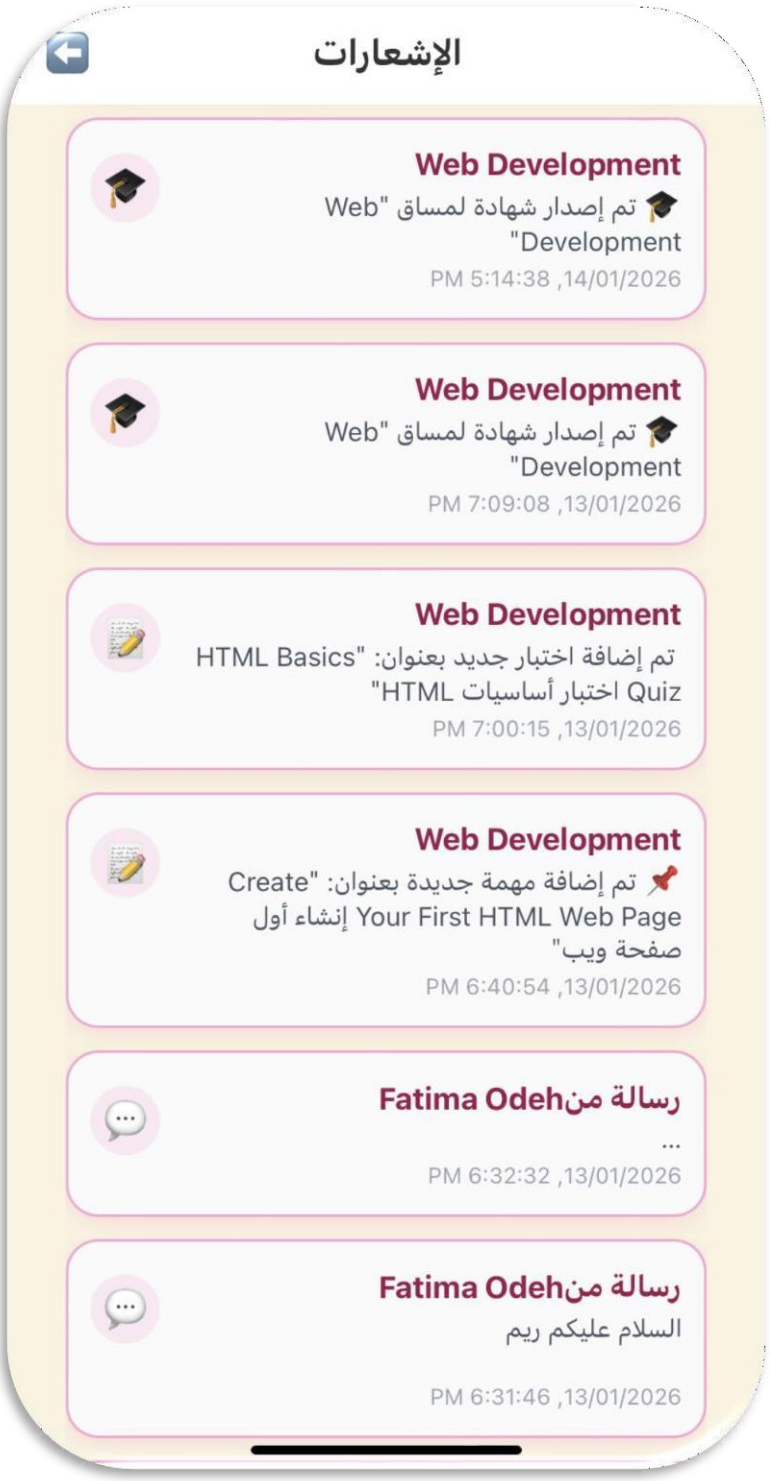


Figure 74: Student notifications

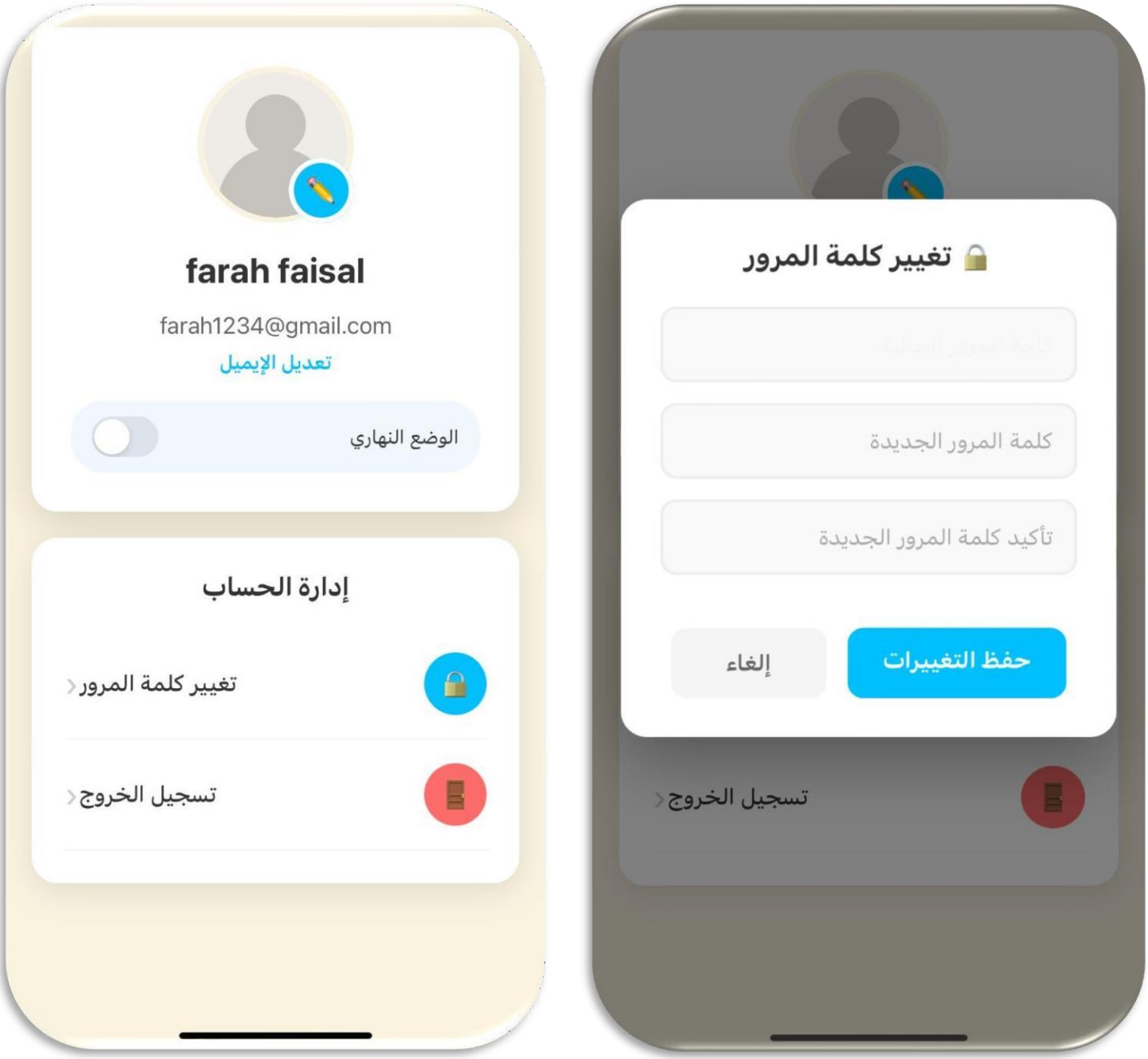


Figure 75 :Student Profile Screen

3.3 Standards and Specifications

The Bright Path platform is developed using common software development best practices to ensure reliability and maintainability. While no formal standards such as ISO or IEEE were required, the system follows well-known design principles and structured development guidelines to support a stable and scalable educational platform.

3.3.1 Authentication and Security Standard

In the Bright Path platform, user authentication is handled using JSON Web Tokens (JWT), where a secure token is generated upon successful login and stored on the client side to authorize access to protected system features. User passwords are never stored in plain text; instead, they are securely hashed using bcrypt before being saved in the database. The platform also includes a secure password recovery mechanism that allows users to reset their passwords through time-limited verification codes sent via email. These measures ensure controlled access, data protection, and a secure authentication process across the system.

3.3.2 API Design

The project's APIs are designed in a REST-like manner, structuring operations such as user registration, login, password management, and course/task management through clearly defined endpoints. The design ensures consistent communication between frontend and backend using JSON, allows future scalability for new features or API versions, and integrates secure authentication (JWT) to protect sensitive data.

3.3.3 UI/UX Standards

The app features a clean and visually appealing interface with consistent use of colors, typography, and layouts to enhance readability and user engagement. Built with React Native, it ensures **cross-platform compatibility** across iOS and Android devices. Accessibility has been considered through clear color contrasts, readable fonts, and intuitive navigation, providing a smooth and user-friendly experience for all users.

3.3.4 Real-Time Communication

Real-time messaging and notifications are implemented using **Firebase**, enabling instant bi-directional communication between users. This ensures low-latency updates, seamless chat interactions, and timely delivery of notifications, enhancing the overall responsiveness and interactivity of the application.

3.3.5 Database Standards

The application uses **MySQL**, following relational database design best practices. Tables are normalized where appropriate, with clearly defined primary and foreign keys to maintain data integrity and logical relationships between entities such as users, courses, lessons, and tasks. Indexed fields are used to ensure fast query performance and scalable data handling.

3.3.6 AI Integration Guidelines

The project integrates OpenAI to provide AI-powered features such as lesson summarization, question generation, practical exercises, and real-time chat assistance. Users can search topics, get relevant information, and access related videos instantly, enhancing interactive and personalized learning.

3.4 Constraints

Development was influenced by technical and practical constraints, including performance, MySQL data management, real-time Firebase communication, AI content integration, and user-friendly UI/UX, all essential for a functional and scalable learning platform.

1. Economic Constraints

The project was developed with limited budget, relying on free/open-source tools (Node.js, MySQL, Firebase, OpenAI) and optimized usage to minimize costs while maintaining full functionality.

2. Technological Constraints

The system needed seamless performance across web and mobile platforms while supporting real-time messaging, notifications, and AI-powered features. Reliance on third-party APIs introduced challenges such as rate limits, latency, and careful backend coordination to ensure smooth, responsive, and reliable user experiences

3. User Experience Constraints

The platform serves administrators, teachers, and students with varying technical skills, requiring intuitive, accessible, and user-friendly interfaces while still supporting advanced learning and AI features.

4. Social and Ethical Constraints

The platform handles sensitive user data, including student messages, feedback, and course content, which requires strict adherence to privacy and ethical standards. All data is managed securely, with controlled access for students, teachers, and administrators. AI-generated content, such as summaries, questions, and practical exercises, is carefully monitored to prevent bias, misinformation, or inappropriate outputs. Additionally, the system includes moderation and reporting tools, ensuring that any violations or issues are addressed promptly and fairly, maintaining a safe and trustworthy learning environment for all users.

5. Health and Safety

The platform is designed to promote a healthy digital learning environment by minimizing cognitive overload and screen fatigue. Interactive elements, gamified exercises, and AI-powered tools—such as flashcards, visualizations, and real-time chat—encourage active engagement, reinforce knowledge retention, and provide a stimulating yet comfortable learning experience for students of all ages.

6. Scalability and Sustainability

The platform is built to grow with its users. Using **React Native**, **Firebase**, **MySQL**, and **OpenAI**, it seamlessly handles real-time messaging, AI-assisted learning, and structured course data. Its flexible architecture allows new features—like interactive quizzes, live sessions, and detailed analytics—to be integrated effortlessly, ensuring long-term performance and adaptability.

3.5 Standards and Specifications

The development of the platform followed best practices and modern design principles to ensure **robust functionality, maintainable code, data security, and anzintuitive user experience**. While no formal standards like IEEE or ISO were strictly applied, the project adhered to widely recognized technical and ethical guidelines relevant to **mobile apps, real-time communication, AI integration, and responsive UI design**.

1. Software Development Standards

All The system was developed using structured backend design principles to ensure security and maintainability. REST-based APIs were used to organize communication between the mobile application and the server, while JWT authentication was implemented to securely manage user access and protect sensitive data.

2. User Interface Standards

The user interface was developed using React Native to provide a consistent and responsive experience across different mobile devices. The design focuses on clarity and ease of use by applying suitable color contrast, readable typography, and well-sized interactive elements to support users with varying levels of technical experience.

3. Security Practices

The Bright platform applies essential security measures to protect user data and system functionality. User authentication is managed using JSON Web Tokens (JWT), ensuring that only authorized users can access protected features. Passwords are securely hashed using bcrypt before storage, preventing exposure of sensitive credentials. Role-based access control is enforced to limit system operations according to user roles (admin, teacher, and student). In addition, file uploads and sensitive actions are validated at the backend level, and password recovery is handled through time-limited verification codes sent via email, ensuring secure and controlled access to the platform.

4. Real-Time Communication Standards

Bright relies on Firebase to support real-time communication features, including instant messaging and notifications between users. Firebase enables immediate data synchronization across devices, ensuring that messages and notifications are delivered without noticeable delay. The system follows structured data organization and real-time update practices to maintain consistency, reliability, and smooth user interaction across both web and mobile platforms.

5. AI Integration Guidelines

Bright integrates artificial intelligence features using the OpenAI API to enhance the learning experience. AI is utilized for generating content summaries, creating practice questions, suggesting practical applications, and enabling interactive chat-based assistance. The system is designed to provide relevant and educational responses while supporting topic-based searches and learning resources, ensuring that AI usage remains focused on improving understanding and engagement in a safe and controlled manner.

6. Database Design Standards

The Bright platform adopts a structured relational database design using MySQL to ensure data accuracy, integrity, and reliability. Tables are carefully organized with primary and foreign keys to represent clear relationships between core entities such as users, courses, lessons, tasks, and assessments. This relational structure reflects real educational workflows, supports efficient querying, and maintains consistency across the system while allowing the platform to scale as new features and data are introduced.

7. Collaboration and Version Control

The Bright project uses Git for version control with a structured branching model and clear commit practices. Code is hosted on GitHub, ensuring team collaboration, review standards, and traceable project history.

Chapter 4: Results and Analysis

This chapter presents Bright's implementation results, assessing system performance, feature usage, and user feedback across students, teachers, and admins, while highlighting successes and areas for improvement.

4.1 System Functionality Results

1. User Authentication and Role-Based Access

The Bright platform provides secure user authentication with JWT-based sessions, ensuring that only authorized roles—Admin, Teacher, or Student—can access their respective features. Signup, login, and password recovery workflows, including verification codes, function reliably, while device registration and notification settings enable safe real-time communication. Role-based access control prevents cross-role data leakage, maintaining the integrity and security of the system.

2. Course Management and Enrollment

Admins efficiently create and manage courses, while teachers submit new courses, organize lessons, and assign tasks after approval. Students can browse courses, submit enrollment requests, and monitor their approval status, ensuring a smooth course management and registration workflow.

3. Content and Assignment Management

In Bright, teachers can seamlessly upload video lessons, create tasks, and manage course content, ensuring students receive structured and interactive learning materials. Students can access lessons, submit assignments on time, and track their progress. The system preserves submissions for both teachers and students, supporting revisions, feedback, and real-time communication, thereby maintaining a smooth workflow and fostering effective learning outcomes.

4. Real-Time Features

The platform leverages Firebase to enable real-time messaging and notifications. Students and teachers can communicate instantly, receive immediate updates on new lessons, assignment feedback, or AI-generated summaries, creating an interactive learning environment.

5. AI Integration

The system integrates AI tools to enhance learning. Students can generate summaries, quizzes, and study plans tailored to their preferred learning style (visual, auditory, practical). Teachers can utilize AI-assisted suggestions to improve lessons, generate practice exercises, and guide students more effectively.

4.2 Performance Evaluation

Feature Tested	Result	Response Time	Notes
Login/Signup	100% successful	< 2 sec	No delay or errors
Course Submission(Teacher)	Functional & Approved by Admin	< 4 sec	Approval required
Lesson Upload (Video + AI)	Functional	5–10 sec	Includes transcription and AI processing via OpenAI
Assignment Submission	Functional	< 3 sec	Supports multiple file formats, submissions visible to teacher
Study Plan Generation	Functional	4–6 sec	Personalized plans based on student learning style
Real-Time Chat (Firebase)	Functional, instant	< 0.5 sec	Students & teachers can message in courses, questions answered quickly
Notifications (Firebase)	Instant Delivery	< 1 sec	Firebase sync successful
Payment Processing (Stripe)	Functional	~2–4 sec	Secure payments for course enrollment
AI Format Generation (Student)	Functional,accurate formats	~3–6 sec	AI processing via OpenAI

4.3 User Feedback and Usability Insights

After User testing showed that students enjoyed personalized learning with AI-powered summaries, quizzes, and study plans, along with easy access to live sessions, lessons, assignments, and real-time chat. Teachers valued efficient course and assignment management, instant insights into student performance, and streamlined test creation. Administrators highlighted smooth course approvals, user and payment management, and clear activity tracking, confirming the platform's usability, responsiveness, and intelligent support for all roles.

4.4 Key Achievements

The platform successfully delivered nearly all planned features, integrating a robust Node.js backend with a React Native frontend for seamless performance. Real-time chat, notifications, and live session management operate smoothly across modules, while AI-powered tools—like summaries, quizzes, and personalized study plans—enhance both learning and assessment, making Bright Path a fully responsive, intelligent, and user-centered educational environment.

Chapter 5: Discussion

Bright Path successfully delivers an AI-assisted educational platform that supports students, teachers, and administrators. It combines interactive lessons, personalized learning paths, live sessions, tasks, assessments, and certificate management in one seamless system. Students enjoy tailored learning experiences, while teachers gain efficient tools for course creation, evaluation, and monitoring, enhancing engagement and overall learning outcomes.

5.1 Problem Resolution

BrightPath successfully addressed the core challenge of modern education: allowing students to choose their preferred learning style while providing intelligent, tailored interfaces for each style. The platform integrates AI-driven tools—such as automatic summarization, interactive quizzes, and performance analytics—within each learning path to enhance the learning experience without enforcing a single approach. This approach effectively resolves the diversity of learning styles while simplifying course management and communication for teachers.

5.2 Project Contributions

The Bright Path platform offers a truly personalized learning experience by allowing students to choose their preferred learning style, with AI-powered tools adapting content accordingly. Key contributions include AI-generated learning aids such as flashcards, summaries, and interactive exercises, intelligent course review analysis for teachers, and real-time progress tracking with feedback. The platform provides a fully integrated web and mobile experience, seamlessly supporting students, teachers, and administrators through tailored dashboards and role-specific features. These capabilities set Bright Path apart by combining user choice, AI assistance, and automated educational support.

5.3 Implications

Bright Path transforms learning by letting students choose their own learning style while AI adapts the content to fit it. Personalized dashboards for students, teachers, and admins provide instant insights, track progress, and streamline interactions—all in one intuitive platform.

5.4 Limitations

The platform is robust, but it has not yet been tested under very high user traffic, which may affect performance. Additionally, some mobile features are not fully accessible offline.

Chapter 6: Conclusions and Recommendation

1.1 Conclusions

Bright Path has successfully established a cutting-edge educational platform that transforms the learning and teaching experience for students, educators, and administrators alike. With its intuitive, well-structured, and interactive dashboards, the platform empowers all users to manage their responsibilities seamlessly, track progress in real time, and engage with course content and communication tools in a highly efficient and user-friendly environment.
Key accomplishments include:

- Secure authentication and role-based access: Ensures that Students, Teachers, and Administrators can access only the features relevant to their roles.
- Comprehensive course and content management: Teachers can create, update, and organize courses, lessons, tasks, and assessments efficiently.
- Real-time communication and notifications: Students and teachers can exchange messages instantly, receive updates on assignments, tests, and live sessions.
- Task and assessment tracking: Enables students to submit tasks and tests, while teachers can review submissions, provide feedback, and monitor academic progress.
- Live session management: Facilitates scheduling, joining, and monitoring of live classes with easy access links.
- Certificate issuance and management: Allows teachers to issue course completion certificates and students to view, download, and share them.
- Interactive dashboards for all users: Provides a clear overview of courses, progress, tasks, and notifications, ensuring smooth and organized workflows.

1.2 Recommendations

- Improve mobile performance with offline access and faster loading.
- Expand analytics and reporting for better tracking of progress.
- Add motivational features like badges and challenges.
- Simplify admin tasks through automation.
- Continue user testing to enhance usability.

1.3 Future Work

- Native mobile app with offline access.
- Integration with school or regional systems for centralized data.
- Virtual support tools for guidance and learning assistance.
- Enhanced collaboration for group projects and peer interaction.

References

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

[2] React. (n.d.). React documentation. Meta. Retrieved from https://react.dev

[3] React Native. (n.d.). React Native documentation. Meta. Retrieved from https://reactnative.dev

[4] Expo. (n.d.). Expo documentation. Retrieved from https://docs.expo.dev

[5] MySQL. (n.d.). MySQL documentation. Oracle. Retrieved from https://dev.mysql.com/doc

[6] Axios. (n.d.). Axios HTTP client documentation. Retrieved from https://axios-http.com

[7] Express.js. (n.d.). Express documentation. Retrieved from https://expressjs.com

[8] Firebase Authentication. (n.d.). Firebase Authentication documentation. Google. Retrieved from https://firebase.google.com/docs/auth

[9] OpenAI. (2024). OpenAI API documentation. Retrieved from https://platform.openai.com/docs

[10] REST API. (n.d.). RESTful API design. Retrieved from https://restfulapi.net

[11] JSON Web Token (JWT). (n.d.). JWT Introduction and usage. Retrieved from https://jwt.io/introduction

[12] OpenRouter. (n.d.). OpenRouter API documentation. Retrieved from https://openrouter.ai/docs

[13] Hugging Face. (n.d.). Hugging Face documentation. Retrieved from https://huggingface.co/docs

[14] [1] YouTube. (n.d.). YouTube Data API v3 documentation. Retrieved https://developers.google.com/youtube/v3

[15] react-native-youtube-iframe. (n.d.). React Native YouTube Player documentation. Retrieved from https://github.com/LonelyCpp/react-native-youtube-iframe

[16] Stripe. (n.d.). Accept a payment. Retrieved from https://stripe.com/docs/payments/accept-a-payment