**Faculty of Engineering Department of**

**Informatics Engineering-Software and**

**Information Systems Engineering**

**Artificial Intelligence Engineering and Data Science**

**AI-Powered Complementary Education System**

Senior Project - Completed the requirements for obtaining a bachelor's degree in Informatics Engineering - Software Engineering and Information Systems

**Prepared By**

Rawad Al-Zoubi Mohammed Hammoud

**Supervised By**

Eng. Anas-Abdulaziz

كلية الهندسة

قسم الهندسة المعلوماتية

هندسة البرمجيات ونظم المعلومات

هندسة الذكاء الصنعي وعلوم البيانات

نظام تعليمي تكميلي مدعوم بالذكاء الاصطناعي

مشروع تخرج 1 - قدم لاستكمال متطلبات الحصول على درجة البكالوريوس في هندسة المعلوماتية

هندسة البرمجيات ونظم المعلومات

هندسة الذكاء الصنعي وعلوم البيانات

إعداد

رواد عاطف الزعبي محمد نجدات حمود

إشراف

المهندس أنس عبد العزيز

# SUPERVISION CERTIFICATION

### Supervisor Certification

|  |
| --- |
| I certify that the preparation of this project entitled:  **"AI-Powered Complementary Education System"**  prepared by **Rawad Al-Zoubi & Mohammed Hammoud**, was made under my supervision at the Faculty of Informatics Engineering in partial fulfillment of the requirements for the Degree of Bachelor of Software and Information System Engineering. |

**Name:** ...........................................................

**Date:** ...........................................................

**Signature:** .....................................................

# **ABSTRACT**

This project aims to develop an educational platform that connects teachers and volunteers with children who are out of school due to social or economic challenges. The platform provides both online and offline access to simplified educational content tailored to students' levels, including recorded lessons, live classes, and interactive exercises. It features a structured progress tracking system, attendance monitoring, and personalized learning recommendations powered by artificial intelligence. The system also includes gamified elements to boost student engagement and emotional support content to maintain motivation. By combining technology, AI, and volunteer efforts, the project seeks to bridge educational gaps and make quality learning more accessible to underserved communities.

# **الملخص**

يهدف هذا المشروع إلى تطوير منصة تعليمية تربط المعلمين والمتطوعين بالأطفال المنقطعين عن الدراسة نتيجة لظروف اجتماعية أو اقتصادية. توفر المنصة محتوىً تعليميًا مبسطًا يتناسب مع مستويات الطلاب، ويشمل دروسًا مسجّلة، وحصصًا مباشرة، وتمارين تفاعلية، مع إمكانية الوصول عبر الإنترنت أو بالحضور المباشر. كما تحتوي المنصة على نظام منظم لتتبع تقدم الطلاب، وتسجيل الحضور، وتقديم توصيات تعليمية مخصصة باستخدام تقنيات الذكاء الاصطناعي. وتشمل أيضًا عناصر تحفيزية تعتمد على (Gamification) لتعزيز التفاعل، بالإضافة إلى محتوى داعم نفسي للحفاظ على دافعية الطلاب. يجمع المشروع بين التكنولوجيا والذكاء الاصطناعي والجهود التطوعية لسد الفجوات التعليمية وتسهيل الوصول إلى التعليم الجيد للفئات المحرومة.

**TABLE OF CONTENT**

**LIST OF FIGURES**

[Figure 1 - Gantt Chart V1 33](#_Toc218505048)

[Figure 2 - Gantt Chart V2 33](#_Toc218505049)

[Figure 3 High Level Use case diagram 36](#_Toc218505050)

[Figure 4 High level Architecture 39](#_Toc218505051)

[Figure 5 sprint 2 usecase diagram - V1 46](#_Toc218505052)

[Figure 6 - Register New User Activity 47](#_Toc218505053)

[Figure 7 - Register New User Sequence 47](#_Toc218505054)

[Figure 8 - login Activity 48](#_Toc218505055)

[Figure 9 - Login Sequence 48](#_Toc218505056)

[Figure 10 - View profile Activity 49](#_Toc218505057)

[Figure 11 - View profile Sequence 49](#_Toc218505058)

[Figure 12 - Edit Profile Activity 50](#_Toc218505059)

[Figure 13 - Edit profile Sequence 50](#_Toc218505060)

[Figure 14 - logout Activity 51](#_Toc218505061)

[Figure 15 - logout Sequence 51](#_Toc218505062)

[Figure 16 - Reset Password Activity 52](#_Toc218505063)

[Figure 17 - Reset Password sequence 53](#_Toc218505064)

[Figure 18 - Sprint 4 use case diagram 55](#_Toc218505065)

[Figure 19 - Upload Video Activity 56](#_Toc218505066)

[Figure 20 - Upload Video Sequence 57](#_Toc218505067)

[Figure 21 - Upload PDF Activity 58](#_Toc218505068)

[Figure 22 - Upload PDF Sequence 58](#_Toc218505069)

[Figure 23 - Upload MCQ Exercises Activity 59](#_Toc218505070)

[Figure 24 - Upload MCQ Exercises Sequence 60](#_Toc218505071)

[Figure 25 - View Uploaded Activity 61](#_Toc218505072)

[Figure 26 - View Uploaded Sequence 61](#_Toc218505073)

[Figure 27 - Update Lesson Activity 62](#_Toc218505074)

[Figure 28 - Update Lesson Sequence 62](#_Toc218505075)

[Figure 29 - Delete Lesson Activity 63](#_Toc218505076)

[Figure 30 - Delete Lesson Sequence 63](#_Toc218505077)

[Figure 31 - Browse Assigned Lesson Activity 64](#_Toc218505078)

[Figure 32 - Browse Assigned Lesson Sequence 64](#_Toc218505079)

[Figure 33 - Select Lesson Activity 65](#_Toc218505080)

[Figure 34 - Select Lesson Sequence 65](#_Toc218505081)

[Figure 35 - Download PDF Activity 66](#_Toc218505082)

[Figure 36 - Download PDF Sequence 66](#_Toc218505083)

[Figure 37 - Solve Exercises Activity 67](#_Toc218505084)

[Figure 38 - Solve Exercises Sequence 67](#_Toc218505085)

[Figure 39 - View Student Progress Activity 68](#_Toc218505086)

[Figure 40 - View Student Progress Sequence 68](#_Toc218505087)

[Figure 41 - Assign Lessons to Student Activity 69](#_Toc218505088)

[Figure 42 - Assign Lessons to Student Sequence 69](#_Toc218505089)

[Figure 43 - Search Lessons by FilterActivity 70](#_Toc218505090)

[Figure 44 - Search Lessons by Filter Sequence 70](#_Toc218505091)

[Figure 45 - sprint 5 use case diagram 72](#_Toc218505092)

[Figure 46 - View quiz Activity 72](#_Toc218505093)

[Figure 47 - View quiz Sequence 73](#_Toc218505094)

[Figure 48 - View Student Quiz Result Activity 74](#_Toc218505095)

[Figure 49 - View Student Quiz Result Sequence 74](#_Toc218505096)

[Figure 50 - Submit Feedback Activity 75](#_Toc218505097)

[Figure 51 - Submit Feedback Sequence 75](#_Toc218505098)

[Figure 52 - View Feedback Activity 76](#_Toc218505099)

[Figure 53 - View Feedback Sequence 76](#_Toc218505100)

[Figure 54 - View Student Answers Activity 77](#_Toc218505101)

[Figure 55 - View Student Answers Sequence 77](#_Toc218505102)

[Figure 56 - Sprint 3 use case diagram 78](#_Toc218505103)

[Figure 57 - Ask AI Homework Helper Activity 79](#_Toc218505104)

[Figure 58 - Ask AI Homework Helper Sequence 80](#_Toc218505105)

[Figure 59 - Sprint 9 Use case diagram 82](#_Toc218505106)

[Figure 60 - Create Individual Live Sessions Activity 84](#_Toc218505107)

[Figure 61 - Create Individual Live Sessions Sequence 84](#_Toc218505108)

[Figure 62 - Edit Live Session Information Activity 85](#_Toc218505109)

[Figure 63 - Edit Live Session Information Sequence 86](#_Toc218505110)

[Figure 64 - Cancel Live Sessions Activity 87](#_Toc218505111)

[Figure 65 - Cancel Live Sessions Sequence 87](#_Toc218505112)

[Figure 66 - Join Live Sessions (as Host) Activity 88](#_Toc218505113)

[Figure 67 - Join Live Sessions (as Host) Sequence 89](#_Toc218505114)

[Figure 68 - View Teacher's Sessions Activity 90](#_Toc218505115)

[Figure 69 - View Teacher's Sessions Sequence 90](#_Toc218505116)

[Figure 70 - View Assigned Students List Sequence 92](#_Toc218505117)

[Figure 71 - View All Live Sessions Activity 93](#_Toc218505118)

[Figure 72 - View All Live Sessions Sequence 93](#_Toc218505119)

[Figure 73 - Assign Students to Sessions Activity 95](#_Toc218505120)

[Figure 74 - Assign Students to Sessions Sequence 95](#_Toc218505121)

[Figure 75 - Unassign Students from Sessions Activity 96](#_Toc218505122)

[Figure 76 - Unassign Students from Sessions Sequence 97](#_Toc218505123)

[Figure 77 - Modify Student Assignments Activity 98](#_Toc218505124)

[Figure 78 - Modify Student Assignments Sequence 98](#_Toc218505125)

[Figure 79 - View Assigned Sessions Activity 99](#_Toc218505126)

[Figure 80 - View Assigned Sessions Sequence 99](#_Toc218505127)

[Figure 81 - Join Live Sessions Activity 100](#_Toc218505128)

[Figure 82 - Join Live Sessions sequence 101](#_Toc218505129)

[Figure 83 - View Session Details Activity 102](#_Toc218505130)

[Figure 84 - View Session Details Sequence 102](#_Toc218505131)

[Figure 85 - Sprint 10 use case diagram 104](#_Toc218505132)

[Figure 86 - Admin Authentication Activity 105](#_Toc218505133)

[Figure 87 - Admin Authentication Sequence 105](#_Toc218505134)

[Figure 88 - View Advisors List Activity 106](#_Toc218505135)

[Figure 89 - Create New Advisor Activity 107](#_Toc218505136)

[Figure 90 - Create New Advisor Sequence 107](#_Toc218505137)

[Figure 91 - Delete Advisor Activity 108](#_Toc218505138)

[Figure 92 - Delete Advisor Sequence 108](#_Toc218505139)

[Figure 93 - Reset Advisor Password Activity 109](#_Toc218505140)

[Figure 94 - Reset Advisor Password Sequence 110](#_Toc218505141)

[Figure 95 - Activate Advisor Activity 111](#_Toc218505142)

[Figure 96 - Activate Advisor Sequence 111](#_Toc218505143)

[Figure 97 - Deactivate Advisor Activity 113](#_Toc218505144)

[Figure 98 - Deactivate Advisor Sequence 113](#_Toc218505145)

[Figure 99 - Sprint 11 use case diagram 115](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Toc218505146)

[Figure 100 - Create Session Template Activity 117](#_Toc218505147)

[Figure 101 - Create Session Template Sequence 117](#_Toc218505148)

[Figure 102 - View Session Templates Activity 118](#_Toc218505149)

[Figure 103 - View Session Templates Sequence 118](#_Toc218505150)

[Figure 104 - Edit Session Template Sequence 120](#_Toc218505151)

[Figure 105 - Delete Session Template Activity 121](#_Toc218505152)

[Figure 106 - Delete Session Template Sequence 122](#_Toc218505153)

[Figure 107 - Manage Templates Status Activity 124](#_Toc218505154)

[Figure 108 - Manage Templates Status Sequence 124](#_Toc218505155)

[Figure 109 - View All Templates (Advisor) Activity 125](#_Toc218505156)

[Figure 110 - View All Templates (Advisor) Sequence 125](#_Toc218505157)

[Figure 111 - Assign Template Activity 126](#_Toc218505158)

[Figure 112 - Assign Template Sequence 127](#_Toc218505159)

[Figure 113 - Create Student Group Activity 128](#_Toc218505160)

[Figure 114 - Create Student Group Sequence 128](#_Toc218505161)

[Figure 115 - Manage Student Group Activity 129](#_Toc218505162)

[Figure 116 - Manage Student Group Sequence 130](#_Toc218505163)

[Figure 117 - View Assigned Templates Activity 131](#_Toc218505164)

[Figure 118 - View Assigned Templates Sequence 131](#_Toc218505165)

[Figure 119 - Sprint 2 Architecture Diagram 148](#_Toc218505166)

[Figure 120 - Sprint 4 Architecture Diagram 148](#_Toc218505167)

[Figure 121 - Sprint 5 Architecture Diagram 149](#_Toc218505168)

[Figure 122 - Sprint 9 Architecture Diagram 149](#_Toc218505169)

[Figure 123 - Sprint9 Class diagram 150](#_Toc218505170)

[Figure 124 - Sprint 10 Architecture Diagram 150](#_Toc218505171)

[Figure 125 - Sprint 10 Class diagram 151](#_Toc218505172)

[Figure 126 - Sprint 11 Architecture Diagram 151](#_Toc218505173)

[Figure 127 - Sprint 11 Class diagram 152](#_Toc218505174)

[Figure 128 - login page 159](#_Toc218505175)

[Figure 129 -Register 159](#_Toc218505176)

[Figure 130 - Forgot Password 160](#_Toc218505177)

[Figure 131 - Home page 160](#_Toc218505178)

[Figure 132 - Advisor dashboard: Student list 161](#_Toc218505179)

[Figure 133 - Advisor dashboard: lesson list 161](#_Toc218505180)

[Figure 134 - Assign lesson 162](#_Toc218505181)

[Figure 135 - Student details 162](#_Toc218505182)

[Figure 136 - Student Performance Page 163](#_Toc218505183)

[Figure 137 - Quiz Result page 163](#_Toc218505184)

[Figure 138 - Advisor Profile page 164](#_Toc218505185)

[Figure 139 - Teacher Dashboard Page 164](#_Toc218505186)

[Figure 140 - Lesson Management page 165](#_Toc218505187)

[Figure 141 - Add new lesson page 165](#_Toc218505188)

[Figure 142 - Update lesson Page 166](#_Toc218505189)

[Figure 143 - Create Quiz page 166](#_Toc218505190)

[Figure 144 - Add Question page 167](#_Toc218505191)

[Figure 145 - Student performance (Teacher) 167](#_Toc218505192)

[Figure 146 - Submit Feedback Page 168](#_Toc218505193)

[Figure 147 - Teacher profile page 168](#_Toc218505194)

[Figure 148 - Student dashboard page 169](#_Toc218505195)

[Figure 149 - View lesson Page 169](#_Toc218505196)

[Figure 150 - Solve quiz page 170](#_Toc218505197)

[Figure 151 - Quiz result Page 170](#_Toc218505198)

[Figure 152 - Quiz result Page 171](#_Toc218505199)

[Figure 153 - Create Live session Page 171](#_Toc218505200)

[Figure 154 - created Sessions 172](#_Toc218505201)

[Figure 155 - Edit Live Session page 172](#_Toc218505202)

[Figure 156 - View live Session info page 173](#_Toc218505203)

[Figure 157 - View Sessions as a Card 173](#_Toc218505204)

[Figure 158 - Create Session Templates page 174](#_Toc218505205)

[Figure 159 - Edit Session Template page 174](#_Toc218505206)

[Figure 160 - Pause Templates page 175](#_Toc218505207)

[Figure 161 - End Templates page 175](#_Toc218505208)

[Figure 162 - View Templates page 176](#_Toc218505209)

[Figure 163 - Join Early page 176](#_Toc218505210)

[Figure 164 - Join Late 177](#_Toc218505211)

[Figure 165 - Join a Cancelled Session page 177](#_Toc218505212)

[Figure 166 - Assign Session to student 178](#_Toc218505213)

[Figure 167 - Assign Session to student page 178](#_Toc218505214)

[Figure 168 - Assign Templates to Students page Manage Assigned Student 179](#_Toc218505215)

[Figure 169 - Manage Assigned Student 179](#_Toc218505216)

[Figure 170 - Create Student Group page 180](#_Toc218505217)

[Figure 171 - View Student Sessions page 180](#_Toc218505218)

[Figure 172 - View Student Templates page 181](#_Toc218505219)

[Figure 173 - Admin Login page 181](#_Toc218505220)

[Figure 174 - Admin Home Page 182](#_Toc218505221)

[Figure 175 - Create Advisor page 182](#_Toc218505222)

[Figure 176 - Delete Advisor Page 183](#_Toc218505223)

[Figure 177 - Reset Advisor Password page 183](#_Toc218505224)

[Figure 178 - Smart HomeWork Helper page 184](#_Toc218505225)

**LIST OF TABLES**

[Table 1-comparative Table 21](#_Toc218507694)

[Table 2 - Indexing Strategies 23](#_Toc218507695)

[Table 3 - Embedding Models 23](#_Toc218507696)

[Table 4 - Vector Database Comparison 24](#_Toc218507697)

[Table 5 Roles and Responsibilities 28](#_Toc218507698)

[Table 6-Human Resources: 31](#_Toc218507699)

[Table 7 - Risk Management 32](#_Toc218507700)

[Table 8-Sprint Table V1 32](#_Toc218507701)

[Table 9 - Sprint Table V2 33](#_Toc218507702)

[Table 10 Requirements Table 45](#_Toc218507703)

[Table 11 Register New User UCS 47](#_Toc218507704)

[Table 12 - Login User UCS 48](#_Toc218507705)

[Table 13 - View profile UCS 50](#_Toc218507706)

[Table 14 - Edit Profile UCS 51](#_Toc218507707)

[Table 15 - Logout UCS 52](#_Toc218507708)

[Table 16 - Reset Password UCS 53](#_Toc218507709)

[Table 17 - Upload Video UCS 57](#_Toc218507710)

[Table 18 - Upload PDF UCS 58](#_Toc218507711)

[Table 19 - Upload MCQ Exercises UCS 60](#_Toc218507712)

[Table 20 - View Uploaded Lesson UCS 61](#_Toc218507713)

[Table 21 - Update Lesson UCS 63](#_Toc218507714)

[Table 22 - Delete Lesson UCS 64](#_Toc218507715)

[Table 23 - Browse Assigned Lesson UCS 65](#_Toc218507716)

[Table 24 - Select Lesson UCS 65](#_Toc218507717)

[Table 25 - Download PDF UCS 66](#_Toc218507718)

[Table 26 - Solve Exercises UCS 67](#_Toc218507719)

[Table 27- View Student Progress UCS 68](#_Toc218507720)

[Table 28 - Assign Lessons to Student UCS 69](#_Toc218507721)

[Table 29 - Search Lessons by Filter UCS 70](#_Toc218507722)

[Table 30 - View Quiz Result UCS 73](#_Toc218507723)

[Table 31 - View Student Quiz Result UCS 74](#_Toc218507724)

[Table 32 - Submit Feedback UCS 76](#_Toc218507725)

[Table 33 - View Feedback UCS 77](#_Toc218507726)

[Table 34 - View Student AnswersUCS 78](#_Toc218507727)

[Table 35 - Ask AI Homework Helper UCS 80](#_Toc218507728)

[Table 36 - Create Individual Live Sessions UCS 84](#_Toc218507729)

[Table 37 - Edit Live Session Information UCS 86](#_Toc218507730)

[Table 38 - Cancel Live Sessions UCS 87](#_Toc218507731)

[Table 39 - Join Live Sessions (as Host) UCS 89](#_Toc218507732)

[Table 40 - View Teacher's Sessions UCS 90](#_Toc218507733)

[Table 41 - View Assigned Students List UCS 92](#_Toc218507734)

[Table 42 - View Assigned Students List Activity 92](#_Toc218507735)

[Table 43 - View All Live Sessions UCS 93](#_Toc218507736)

[Table 44 - Assign Students to Sessions UCS 95](#_Toc218507737)

[Table 45 - Unassign Students from Sessions UCS 97](#_Toc218507738)

[Table 46 - Modify Student Assignments UCS 98](#_Toc218507739)

[Table 47 - View Assigned Sessions UCS 100](#_Toc218507740)

[Table 48 - Join Live Sessions UCS 101](#_Toc218507741)

[Table 49 - View Session Details UCS 102](#_Toc218507742)

[Table 50 - Admin Authentication UCS 105](#_Toc218507743)

[Table 51 - View Advisors List UCS 106](#_Toc218507744)

[Table 52 - Create New Advisor UCS 107](#_Toc218507745)

[Table 53 - Delete Advisor UCS 109](#_Toc218507746)

[Table 54 - Reset Advisor Password 110](#_Toc218507747)

[Table 55 - Activate Advisor UCS 111](#_Toc218507748)

[Table 56 - Deactivate Advisor UCS 113](#_Toc218507749)

[Table 57 - Create Session Template UCS 117](#_Toc218507750)

[Table 58 - View Session Templates UCS 119](#_Toc218507751)

[Table 59 - Edit Session Template UCS 120](#_Toc218507752)

[Table 60 - Edit Session Template Activity 121](#_Toc218507753)

[Table 61 - Delete Session Template UCS 122](#_Toc218507754)

[Table 62 - Pause Template UCS 123](#_Toc218507755)

[Table 63 - Resume Template UCS 124](#_Toc218507756)

[Table 64 - Resume Template UCS 124](#_Toc218507757)

[Table 65 - View All Templates (Advisor) UCS 126](#_Toc218507758)

[Table 66 - Assign Template UCS 127](#_Toc218507759)

[Table 67 - Create Student Group UCS 128](#_Toc218507760)

[Table 68 - Manage Student Group UCS 130](#_Toc218507761)

[Table 69 - View Assigned Templates UCS 131](#_Toc218507762)

[Table 70 - Initial Test Cases 133](#_Toc218507763)

[Table 71 - RTM V0 146](#_Toc218507764)

[Table 72 - RTM V1 156](#_Toc218507765)

[Table 73 - Execute test case 189](#_Toc218507766)

[Table 74 - RTM V2 193](#_Toc218507767)

CHAPTER 1 - INTRODUCTION

## **1.1 Introduction**

This chapter introduces the AI-Powered Complementary Education System, a digital learning platform designed to support out-of-school children by providing accessible, structured, and engaging educational content. The system leverages modern web technologies and applied artificial intelligence to deliver personalized learning assistance, multimodal search capabilities, and interactive lessons.

The chapter outlines the motivation behind developing this platform, explains the educational challenges faced by children disconnected from formal schooling, and highlights the system’s key features—such as AI-based homework assistance, image- and voice-based search (multimodal retrieval), smart quiz generation, and gamified learning experiences.

It also presents the structure of the report and briefly describes the topics covered in the upcoming chapters.

## **1.2 Problem Statement**

Millions of children worldwide remain out of school due to poverty, conflict, displacement, social restrictions, or limited access to technology. These students lack consistent, structured, and motivating educational environments. Existing e-learning platforms are either too advanced, require digital skills, or lack adaptability to students with weak educational backgrounds.

Traditional online education often fails to provide:

* personalized guidance,
* simplified explanations,
* motivational systems,
* flexible learning accessible on low-spec devices,
* effective support for students who struggle to type or formulate academic questions.

Moreover, many students communicate better verbally or visually, making text-only systems insufficient. Without intelligent assistance, multimodal support, or gamified engagement, the learning gap continues to grow.

Therefore, there is a critical need for an inclusive and AI-powered educational platform capable of adapting to each student's needs and offering interactive support through text, images, and voice.

## **1.3 Proposed System**

The proposed system is an AI-powered educational platform designed to support out-of-school children by providing structured lessons, real-time interaction, and intelligent assistance. The platform includes:

### Core Features

* **Live Sessions 2.0:** A redesigned system for scheduling, managing, and attending live classes with improved stability and recurring sessions.
* **Admin System:** Tools for administrators to manage users, roles, lessons, and system operations.
* **Progress Tracking:** Monitoring student advancement through quizzes, attendance, and lesson completion.
* **Gamification System:** Points, badges, levels, and motivation-based feedback to encourage consistent learning.

### AI-Enhanced Features

* **AI Homework Helper:** Provides step-by-step explanations, guidance, and support for assignments.
* **Smart Quiz Generator:** Automatically generates questions and quizzes based on lesson content.
* **Multimodal Search System:**
  + **Image Search:** Students can upload a photo or screenshot to find related explanations or lessons.
  + **Voice Search (S2R Research):** A custom-trained model that retrieves answers based on spoken questions, helping students who cannot type academic terms correctly.

### User Roles

* **Students:** Access structured lessons, attend live classes, perform quizzes, and interact with the AI helper.
* **Teachers:** Upload, organize, and manage lessons; create live sessions; monitor student progress.
* **Supervisors:** Oversee students’ learning journey and ensure educational consistency.
* **Admin:** control the system, Supervisors management.

Through these integrated features, the system aims to provide a complete, accessible, and intelligent learning environment tailored to vulnerable children.

## **1.4 Project Objective**

The primary objective of this project is to develop an AI-powered online platform that enables teachers and volunteers to deliver structured educational content to out-of-school children.

The key goals include:

* Delivering a digital platform that supports remote access to lessons, live classes, and interactive content.
* Enabling students to engage with educational material through text, images, and voice-based search.
* Implementing tools for teachers to manage lessons, monitor students, and organize classes.
* Providing an intelligent Homework Helper that assists children with assignments.
* Adding a Smart Quiz Generator to create assessments automatically.
* Supporting student motivation through a comprehensive gamification system.
* Offering dashboards that display progress, performance, and activity in a clear and measurable way.

## **1.5 Report Organization**

The remainder of the report is organized into the following chapters:

* Chapter 1: Introduction
* Chapter 2: Basic Concepts and Literature Review
* Chapter 3: Project Management
* Chapter 4: System Analysis
* Chapter 5: System Design
* Chapter 6: Practical Implementation

## **1.6 Summary**

This chapter introduced the motivation and vision behind the AI-Powered Complementary Education System. It addressed the educational challenges faced by out-of-school children and highlighted the limitations of traditional learning tools.

To overcome these challenges, the proposed system integrates structured online lessons, interactive live sessions, AI-assisted homework help, smart quiz generation, and multimodal search capabilities.

Through these components, the platform aims to deliver a more accessible, engaging, and intelligent learning experience for students who are most at risk of educational exclusion.

**CHAPTER 2 – LITERATURE REVIEW**

## **2.1 Introduction**

In today’s rapidly evolving digital landscape, technology has become a crucial gateway to education—especially for children who have been forced out of traditional schooling due to conflict, displacement, poverty, or social barriers. These learners often lack access to structured guidance, personalized feedback, and emotional support, which significantly impacts their educational continuity.

The AI-Powered Complementary Education System is designed to address this gap by providing an accessible, interactive, and intelligent learning environment tailored to out-of-school children. The platform combines structured learning materials with modern technologies such as AI-powered assistance, multimodal search, and live teacher-student interaction, ensuring that students receive academic support even in low-resource settings.

This chapter introduces the educational challenges faced by underserved learners, outlines the vision behind the proposed system, and reviews existing digital learning platforms to highlight their strengths and limitations. Through this analysis, we identify the technological and pedagogical gaps that our system aims to address—particularly the lack of personalized feedback, Arabic-language support, motivational tools, and AI-driven assistance designed specifically for vulnerable children.

## **2.2 Literature Review (Software Engineering Section): Comparative Analysis of Existing Platforms**

Several educational platforms aim to support underserved learners through varying levels of accessibility, interactivity, and instructional design. While each has contributed significantly to expanding educational access, they also exhibit certain limitations when addressing the specific needs of out-of-school children. Below is a comparative analysis of five notable platforms, highlighting their strengths, weaknesses, similarities to our proposed system, and what differentiates our solution.

### 2.2.1 Khan Academy

Khan Academy is a globally recognized educational platform offering high-quality video lessons and exercises across a wide range of subjects. One of its main strengths lies in its structured learning paths and performance-based content recommendations. However, it lacks live teacher-student interaction and relies heavily on stable internet connectivity, which limits its use in under-resourced environments. Similar to our platform, it provides progress tracking and personalized content. Nevertheless, our system stands out by incorporating live classes, Arabic language support, and an AI-powered Homework Helper tailored specifically to marginalized learners.

### 2.2.2 Kolibri

Kolibri, developed by Learning Equality, is an offline-first educational platform targeting learners in areas with limited or no internet access. Its primary strength is its ability to operate entirely offline while still providing structured content and learner tracking. Despite this, it lacks interactivity, gamification elements, and AI-driven features. Our platform shares Kolibri’s mission of reaching underserved children but adds a layer of engagement and personalization through interactive features, AI assistance, and emotional support tools.

### 2.2.3 Edraak (K–12 Stream)

Edraak provides Arabic-language educational content aligned with national curricula in the Arab world. It offers professional-quality video lessons and structured learning experiences. However, Edraak does not support adaptive learning paths or real-time student-teacher interaction. Our platform improves upon this by offering personalized learning journeys using AI, real-time class scheduling, and motivational systems designed to increase student involvement.

### 2.2.4 Rumie

Rumie focuses on delivering bite-sized educational content for mobile users, especially in refugee and low-income communities. Its simplicity and mobile-friendly design make it accessible, but its content is limited in depth and does not support interaction or progress tracking. While both platforms aim to serve disadvantaged learners, our solution distinguishes itself by offering full lesson modules, interactive sessions, and AI-driven learning support, which Rumie currently lacks.

### 2.2.5 Nafham

Nafham is a free Arabic educational platform that relies on volunteer-generated content aligned with school curricula. It is accessible and community-driven, but it lacks systems for student-teacher engagement, progress monitoring, and adaptive learning. While it shares our platform’s goal of Arabic-language, volunteer-supported education, our system goes further by integrating technology such as AI, gamification, and live instruction—making it more comprehensive and personalized.

## **2.3 Comparative Table: Feature-by-Feature Analysis**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature / Platform | Khan Academy | Kolibri | Edraak | Rumie | Nafham | Our Platform (AI-Powered CES) |
| **Language Support (Arabic)** | Limited | Partial | Yes | Limited | Yes | Yes (Full Support) |
| **Live Classes / Teacher Interaction** | No | No | No | No | No | Yes |
| **Progress Tracking** | Yes | Basic | Yes | No | No | Yes (Advanced Dashboard) |
| **Gamification System** | Limited | No | No | No | No | Yes (Points, Badges, Streaks) |
| **AI-Powered Assistance** | Partial | No | No | No | No | Yes (Homework Helper) |
| **Multimodal Search (Text, Image, Voice)** | No | No | No | No | No | Yes (Image Search + Voice Search S2R) |
| **Volunteer-Based Teaching** | No | No | No | No | Yes | Yes |
| **Customized Learning Paths** | Yes | No | No | No | No | Will be supported in future releases (\*) |
| **Mobile-Friendly Design** | Yes | Yes | Yes | Yes | Partial | Yes (Low-resource optimized) |
| **Content Types** | Videos, Exercises | Videos, Offline Content | Video Lessons | Bite-sized Content | Curriculum Videos | Videos + PDFs + Interactive Content + AI Support |
| **Target Audience** | General Audience | Underserved Communities | Arabic K-12 | Refugees & Low-Income | Arabic Students | Out-of-school children (primary focus) |
| **(\*)** Customized learning paths are future features planned for later development phases. | | | | | | |

Table 1-comparative Table

# Reference Study for Multimodal Retrieval Augmented Generation (Multimodal-RAG)

### Abstract

Multimodal Retrieval Augmented Generation (MM-RAG) represents a significant evolution of the traditional RAG paradigm, expanding capabilities beyond text to integrate diverse modalities including images, audio, and structured data. This study provides a detailed architectural breakdown and comparative analysis of the core components required to build a robust MM-RAG system. We evaluate key technologies across the pipeline, including specialized embedding models (e.g., CLIP and DINOv2), vector database architectures, and advanced retrieval techniques such as Hybrid Search and Reranking. Furthermore, the report analyzes advanced architectural solutions like HetaRAG (Hybrid Retrieval over Heterogeneous Data Stores) and DRAG (Distilling RAG) for mitigating hallucinations and optimizing efficiency in resource-constrained environments. The analysis concludes with a synthesis of best practices for constructing an optimal, multi-stage MM-RAG architecture.

### I. Introduction and Foundational Concepts

Retrieval Augmented Generation (RAG) is established as a dominant paradigm for mitigating knowledge Hallucinations and ensuring Factual Grounding in Large Language Models (LLMs). Multimodal RAG (MM-RAG) extends this framework by integrating non-textual data—such as Images, Audio, and Video—into the retrieval and generation processes. This integration is critical for enabling complex, real-world applications where contextual understanding relies on rich, heterogeneous data sources.

#### Core Multimodal-RAG Architecture

A robust MM-RAG system operates through three primary, interdependent stages:

* **Knowledge Preparation & Indexing:** Converting diverse raw data (Multimodal Data) into numerical vector representations (Embeddings).
* **Multimodal Retriever:** Locating the most relevant context (Contextual Relevance) using vector search and advanced ranking techniques.
* **Generative Model (MLLM/LLM):** Synthesizing the final response based on the retrieved multimodal context.

### II. MM-RAG Architectural Stages

#### II.1. Knowledge Preparation and Indexing

The effectiveness of MM-RAG is fundamentally determined during this initial offline stage. The process involves specialized handling for each modality to mitigate Information Loss and ensure accurate retrieval.

**Modality Processing Challenges:**

* **Text and Images (PDFs):** Requires OCR to extract text and link it to the original image to enable search across Mixed Content Types. A primary risk is the loss of design information or fine details.
* **Image:** Requires generating Embeddings directly using Multimodal Embedding models or creating Text Summaries/Captions. The challenge lies in preserving fine visual details during textual conversion.
* **Audio:** Raw sound waves are converted into vectors using specialized models (e.g., Wav2Vec). Maintaining Acoustic Features and semantic meaning in the vector space is the primary technical hurdle.

**Indexing Strategies:**

|  |  |  |
| --- | --- | --- |
| Strategy | Mechanism | Advantages / Limitations |
| **Raw Visual Indexing** | Embed images using Multimodal models; pass raw images + text chunks to MLLM. | Preserves original image fidelity but requires computationally expensive MLLM generation. |
| **Text Summary Indexing** | Use MLLM to generate Captions; index and retrieve only these texts. | Simplifies retrieval to traditional text search but suffers from significant Information Loss. |
| **Hybrid Indexing** | Index text summaries while retaining a Reference to the original image for generation. | **Optimal Approach:** Balances efficient text search with the retention of original data fidelity. |

Table 2 - Indexing Strategies

#### II.2. Multimodal Retrieval

The retrieval stage is responsible for accurately locating content relevant to the user's Query. This stage is heavily dependent on the choice of embedding models and database architecture.

**A. Embedding Models: Image and Audio Comparison**

Visual Embedding Models: The choice between contrastive and self-supervised models dictates the system's ability to handle alignment versus detail.

|  |  |  |
| --- | --- | --- |
| Feature | CLIP (e.g., ViT-Base32) | DINOv2 (e.g., ViT-B14) |
| **Training Paradigm** | Contrastive Learning (Vision-Language Alignment) | Self-Supervised Learning (Pure Vision) |
| **Primary Strength** | Alignment / Search | Detail / Accuracy |
| **Dimensionality** | d=512 (Storage efficient) | d=768 (Higher feature capacity) |
| **Zero-Shot Capability** | Excellent: Native Cross-Modal Retrieval | Limited: Lacks native text alignment; requires Functional Mapping. |
| **Fine-Grained Accuracy** | Low (e.g., 15% on iNaturalist-2021) | High (e.g., 70% on iNaturalist-2021) |

Table 3 - Embedding Models

Audio Retrieval Approaches:

* **Wav2Vec 2.0:** Processes raw sound waves via CNN/Transformer to extract acoustic features.
* **Whisper:** An Encoder-Decoder Transformer for Speech-to-Text. While useful for metadata creation, it creates an ASR cascade bottleneck.
* **Speech-to-Retrieval (S2R):** A Dual-Encoder model that bypasses ASR, mapping audio queries directly to a Semantic Vector. This prevents cascading errors and focuses on user intent.

**B. Vector Database Comparison**

|  |  |  |  |
| --- | --- | --- | --- |
| Database | Architecture Focus | Scalability Feature | Key Search Advantage |
| **Milvus** | Distributed, Open-Source | Horizontal Scaling for billions of vectors. | Supports Hybrid Vector–Scalar Queries. |
| **Weaviate** | Open-Source, Cloud-Native | Multi-node Clustering. | Robust Hybrid Search (Dense Vector + Keyword). |
| **Pinecone** | Serverless, Managed Cloud | Auto-scaling; optimized for Low-Latency. | Integrated Inference and Sparse/Dense fusion. |
| **FAISS** | In-Memory Library (Meta) | Limited / Single-node. | Ultra-Low Latency (0.19 ms avg). |

Table 4 - Vector Database Comparison

**C. Reranking as a Critical Second Stage**

Reranking refines the initial retrieval set. A specialized Relevancy Score (RS), where RS ∈ [0, 1], provides a quantitative metric optimized to penalize irrelevant data. This explicitly targets the failures of traditional Cosine Similarity (CLIP-score), significantly reducing Hallucinations.

#### II.3. Multimodal Generation

* **MLLM-Based Generation:** Uses MLLMs to directly process retrieved multimodal inputs (images/text), reducing information loss.
* **Augmented Multimodal Output:** Utilizes pre-existing data (charts, videos) to provide step-by-step visual explanations.

### III. Comparative Analysis of Core Components

#### III.1. The Role of Vision-Language Models (VLMs)

VLMs (e.g., CLIP, Flamingo) enable Cross-Modal Transfer Learning, aligning visual and language features into a Shared Embedding Space. This unification allows developers to leverage textual knowledge to improve inference on image-related tasks or fine-tune on limited data, enhancing robustness against noisy or incomplete modalities.

#### III.2. Practical Applications (Use Cases)

* **E-Commerce:** Visual Product Search allows retrieval based on visual similarity and textual descriptions.
* **Healthcare:** Practitioners can upload medical images with text queries to retrieve clinical info, supporting handwritten notes via OCR.
* **Finance:** Automates Mortgage Workflows by analyzing diverse docs (contracts, forms), potentially reducing response time by 90%.
* **E-Governance:** Rapid analysis of complex survey data (graphs, text) to support executive decision-making.

#### III.3. Challenges

* **Retrieval Failure:** Selecting inputs irrelevant to the query.
* **Context Conflict:** Hallucinations arising from conflicting retrieved information.
* **GIGO Problem:** Information loss during indexing leads to unreliable responses.
* **Bias:** Inheritance of biases from Teacher LLMs.

### IV. Optimal Solution Recommendations for Multimodal-RAG Construction

Based on the architectural analysis, the following "Hybrid Multi-Stage" approach is recommended for optimal system performance.

1. **Knowledge Preparation: Hybrid Multi-Modal Indexing**
   * **Technique:** Text Summary Indexing + Reference + OCR + Structured Data.
   * **Rationale:** Balances search performance (light text indexing) with the necessity of preserving Fine-Grained Details (via raw image references) to minimize Information Loss.
2. **Embedding & Alignment: Tri-Embedding Strategy**
   * **Technique:** Utilize CLIP (for Cross-Modal Alignment) AND DINOv2 (for Fine-Grained Visual Accuracy).
   * **Rationale:** Leverages CLIP’s zero-shot power for language-to-visual search while ensuring DINOv2’s superior discrimination for complex visual retrieval.
3. **Audio Retrieval: Speech-to-Retrieval (S2R)**
   * **Technique:** Use a Dual-Encoder model to map audio to semantic vectors.
   * **Rationale:** Avoids Cascading Errors inherent in ASR (Speech-to-Text) cascades; focuses on spoken intent.
4. **Core Retrieval: HetaRAG Architecture**
   * **Technique:** Fuse results from a Vector Database (Semantic), Full-Text Index (Lexical), and Knowledge Graph (Relational).
   * **Rationale:** Mitigates single-store weaknesses; ensures maximum Recall for complex multi-hop reasoning.
5. **Context Refinement: Multi-Stage Reranking**
   * **Technique:** Implement a secondary filter using an MLLM-based Reranker to calculate the Relevancy Score (RS).
   * **Rationale:** Explicitly penalizes irrelevant content to reduce hallucinations.
6. **Final Generation: MLLM Direct Processing**
   * **Technique:** Direct MLLM processing of raw multimodal data.
   * **Rationale:** Ensures Factual Grounding by allowing the model to self-verify against source images.

**CHAPTER 3 – Project Management and Initialization**

## **3.1 Introduction**

This chapter presents the project management approach adopted for developing the AI-Powered Complementary Education System, a platform designed to support out-of-school children through interactive learning, live instruction, gamification, and AI-driven assistance.

The chapter outlines the key management processes, including defining the project scope, identifying the objectives and major deliverables, structuring the project workflow, and organizing tasks into incremental development cycles using the Scrum methodology.

Additionally, the chapter introduces the essential project management documents—such as the Project Charter, Sprint Plan, and Roles & Responsibilities—that guide the execution and monitoring of the system throughout its lifecycle.

## **3.2 Project Management Documents**

### **3.2.1 Project Charter**

A Project Charter is an official authorization to initiate the project. It outlines the project’s purpose, goals, scope boundaries, key stakeholders, and overall management direction. It also serves as a reference document for maintaining alignment throughout the development lifecycle.

**Project Title:** AI-Powered Complementary Education System

**Project Start Date:** October 18, 2025

**Projected Finish Date:** January 25, 2026

**Project Manager:** Eng. Anas Abdulaziz

**Project Objectives**

The primary objective of this project is to build an intelligent, accessible, and engaging educational platform that supports out-of-school children by combining structured learning with AI-powered assistance. The key objectives include:

* Developing a secure multi-role platform supporting Students, Teachers, Supervisors, and Admins.
* Enabling teachers to upload, manage, and deliver recorded lessons and live sessions.
* Providing students with remote access to lessons, live classes, and assigned educational paths.
* Implementing structured progress tracking for lesson completion, quizzes, and live class attendance.
* Building an AI-powered Homework Helper that guides students step-by-step in solving academic tasks.
* Developing a Multimodal Search System (RAG-based) that supports Text Search, Image Search, and Voice-based Retrieval.
* Implementing a full Live Classes Management System (scheduling, assigning, joining, notifications).
* Adding a complete Gamification System to motivate students through points, badges, and streak tracking.
* Providing a centralized Admin System for managing Supervisors and platform-level configurations.

**Approach (Scrum Methodology)**

The project follows the Scrum framework, adopting an iterative and incremental development approach. Key elements include:

* Dividing the work into multiple sprints, each lasting two weeks, except the R&D sprints (three weeks).
* Conducting sprint planning, daily reviews, backlog refinement, and sprint retrospectives.
* Ensuring continuous improvement and adaptability based on feedback from the supervisor and team.
* Delivering functional increments at the end of each sprint to ensure transparency and measurable progress.

**Roles and Responsibilities**

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibility |
| Eng. Anas Abdulaziz | Project Manager | Oversee development, manage resources, and ensure project objectives are met. |
| Rawad Alzoubi | Fullstack Software Engineer (React + Django) | Develop and integrate frontend and backend components, ensuring seamless user experience. |
| Mohammed Hammoud | AI Engineer | Implement machine learning algorithms for personalized learning recommendations and student performance analysis. |

Table 5 Roles and Responsibilities

### **3.2.2 Statement of Work (SOW)**

#### Project Title:

**AI-Powered Complementary Education System**

#### 1. Project Description and Objectives

This project aims to develop an intelligent, accessible educational platform designed to support children who have dropped out of school. The system provides structured recorded content, interactive exercises, live online classes, progress tracking, and motivational tools.

A major enhancement in this version is the integration of AI-powered features, including an advanced **Homework Helper** and a **Multimodal Search System** capable of retrieving information using text, images, and voice queries. The system targets students, teachers, supervisors, and administrators, offering a comprehensive learning environment supported by modern AI and web technologies.

#### 2. Project Scope

The project scope includes the following key features and systems:

* A secure **role-based system** (Student, Teacher, Supervisor, Admin).
* **Content management system** for uploading and organizing lessons (videos, PDFs, exercises).
* Full **Live Classes Management System**: scheduling, assignment, joining, and notifications.
* Attendance and **progress tracking** for lessons and quizzes.
* **Gamification System**: points, badges, and streak tracking.
* **AI-powered Homework Helper** for guiding students in solving assignments.
* **Multimodal Search System**, featuring:
  + Text Search (existing RAG pipeline)
  + Image Search (CLIP-based retrieval)
  + Voice Search (custom S2R model)
* **Admin panel** for managing Supervisors and system configurations.
* Motivational and emotional support components integrated into the student experience.

#### 3. Project Goals

* Build a user-friendly educational platform accessible across devices.
* Enable teachers to create, upload, and manage educational content easily.
* Provide students with structured learning, live sessions, and interactive engagement.
* Integrate AI tools (Homework Helper + Multimodal Search) to improve accessibility and self-learning.
* Implement reliable systems for progress tracking and attendance monitoring.
* Develop a full gamification layer that maintains student motivation.
* Ensure system security, scalability, and maintainability for future expansions.

#### 4. Deliverables

* Updated Project Plan with phases, sprints, and milestones.
* Updated Software Requirements Specification (SRS).
* A functional education management system including:
  + Secure role-based authentication.
  + Lesson management system (videos, PDFs, exercises).
  + Live classes scheduling, assignment, and joining.
  + Progress & attendance tracking modules.
  + Gamification module (points, badges, streaks).
  + AI-powered Homework Helper.
  + Multimodal Search (Text + Image + Voice).
  + Supervisor management through Admin panel.
* Complete documentation for system design and implementation.
* Final project report, testing results, and demonstration materials.

#### 5. Project Requirements

**Technologies & Tools**

* **Programming Languages:** Python, JavaScript, HTML, Tailwind CSS
* **Frameworks:** Django (Backend), React (Frontend)
* **Database:** MySQL
* **AI / ML Tools:**
  + Embedding Models (SentenceTransformers, CLIP)
  + Whisper for speech processing
  + PyTorch for S2R model training
  + FAISS / ChromaDB for vector search

#### 6. Assumptions

* Project manager provides continuous supervision and approvals.
* Team members are available for weekly sprint reviews and stand-up meetings.
* Required computational resources for AI training (Colab / local GPU) are available.
* Content required for indexing (videos, PDFs, exercises) is accessible.
* The supervisor provides feedback for each sprint increment.

#### 7. Project Resources

### Human Resources:

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibility |
| Eng. Anas Abdulaziz | Project Manager | Oversee development, manage resources, and ensure project objectives are met. |
| Rawad Alzoubi | Fullstack Software Engineer (React + Django) | Develop and integrate frontend and backend components, ensuring seamless user experience. |
| Mohammed Hammoud | AI Engineer | Implement machine learning algorithms for personalized learning recommendations and student performance analysis. |

Table 6-Human Resources:

#### 8. Schedule

|  |  |
| --- | --- |
| Milestone | Date |
| Project Start Date | 18 October 2025 |
| First Review Seminar | 15 November 2025 |
| Second Review Seminar | 27 November 2025 |
| Final Review Seminar | 10 January 2026 |
| Project Completion Date | 25 January 2026 |

### **3.2.3 Risk Management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Risk Type | Description | State | Risk Level | Mitigation Plan |
| **RK01** | Single Developer handling FE+BE | One person managing both frontend and backend may cause delays. | Open | High | Use of artificial intelligence tools (e.g., code generators) to speed up routine tasks. |
| **RK02** | Time Pressure | Risk of missing deadlines or delivering an incomplete product. | Open | Medium | Focus strictly on the **Minimum Viable Product (MVP)** for core features. |
| **RK03** | Small team size (only two members) | Slower task execution and workload overload. | Open | High | Divide tasks clearly based on expertise and use task/time management tools effectively. |
| **RK04** | Lack of experience in project development | Poor planning, possible technical and organizational mistakes. | Open | High | Consult supervisor regularly, study similar projects, and follow established best practices. |
| **RK05** | Incomplete knowledge of proposed technologies | Delays or ineffective implementation of features. | Open | High | Create a parallel learning plan (tutorials/documentation) and prioritize core features first to reduce initial technical risk. |

Table 7 - Risk Management

### **3.2.4 Sprint Table**

**First version**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint | Name | Start Date | End Date | Goals |
| SP1 | Project management | 7/04/2025 | 20/04/2025 | Plan project, define scope, assign roles |
| SP2 | System Setup & Authentication | 21/04/2025 | 4/5/2025 | - Set up the development environment. - Implement user authentication. |
| SP3 | Ai feature | 7/4/2025 | 2/6/2025 | Implement Ai feature |
| SP4 | Content Management | 5/05/2025 | 19/05/2025 | - Develop the lesson upload feature (Video, PDF, Exercises). - Categorize lessons based on subject and level. - Design the content management dashboard for teachers. |
| SP5 | Attendance & Progress Tracking | 20/05/2025 | 4/06/2025 | - Develop the attendance tracking system (automatic/manual). - Implement a student progress tracking dashboard. - Allow teachers to leave feedback on student performance. |
| SP6 | Final Testing | 5/06/2025 | 21/06/2025 | - Perform final testing, bug fixes, and optimizations. |
| SP7 | Final | 17/06/2025 | 30/06/2025 | - Final document and review it. - Final seminar preparation. |

Table 8-Sprint Table V1

**Second version**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint | Name | Start Date | End Date | Goals |
| **SP8** | Project Management | 18/10/2025 | 31/10/2025 | Finalize scope & backlog; Sprint planning; Environment setup |
| **SP9** | Live Sessions | 01/11/2025 | 14/11/2025 | Implement live single-session scheduling; Teacher create/edit sessions; Advisor assigns students; Student join sessions |
| **SP10** | Admin System | 15/11/2025 | 28/11/2025 | Enable Admin to manage Supervisors; Add create/update/activate/deactivate Supervisor accounts |
| **SP11** | Recurring Sessions System | 29/11/2025 | 12/12/2025 | Implement templates for recurring sessions; Group assignment; Auto-generate sessions |
| **SP12** | Image Search (Multimodal – Part 1) | 10/11/2025 | 30/11/2025 | Index platform images using CLIP; Build POST /search-image API; Integrate with vector DB |
| **SP13** | Voice Search (Multimodal – Part 2) | 01/12/2025 | 21/12/2025 | Train S2R model; Build embedding pipeline for audio; Create POST /search-voice API |
| **SP14** | Fix Bugs & Final Optimization | 22/12/2025 | 05/01/2026 | Bug fixing; Refactoring; Prepare system for final testing |

Table 9 - Sprint Table V2

### 3.2.5 Gantt Chart

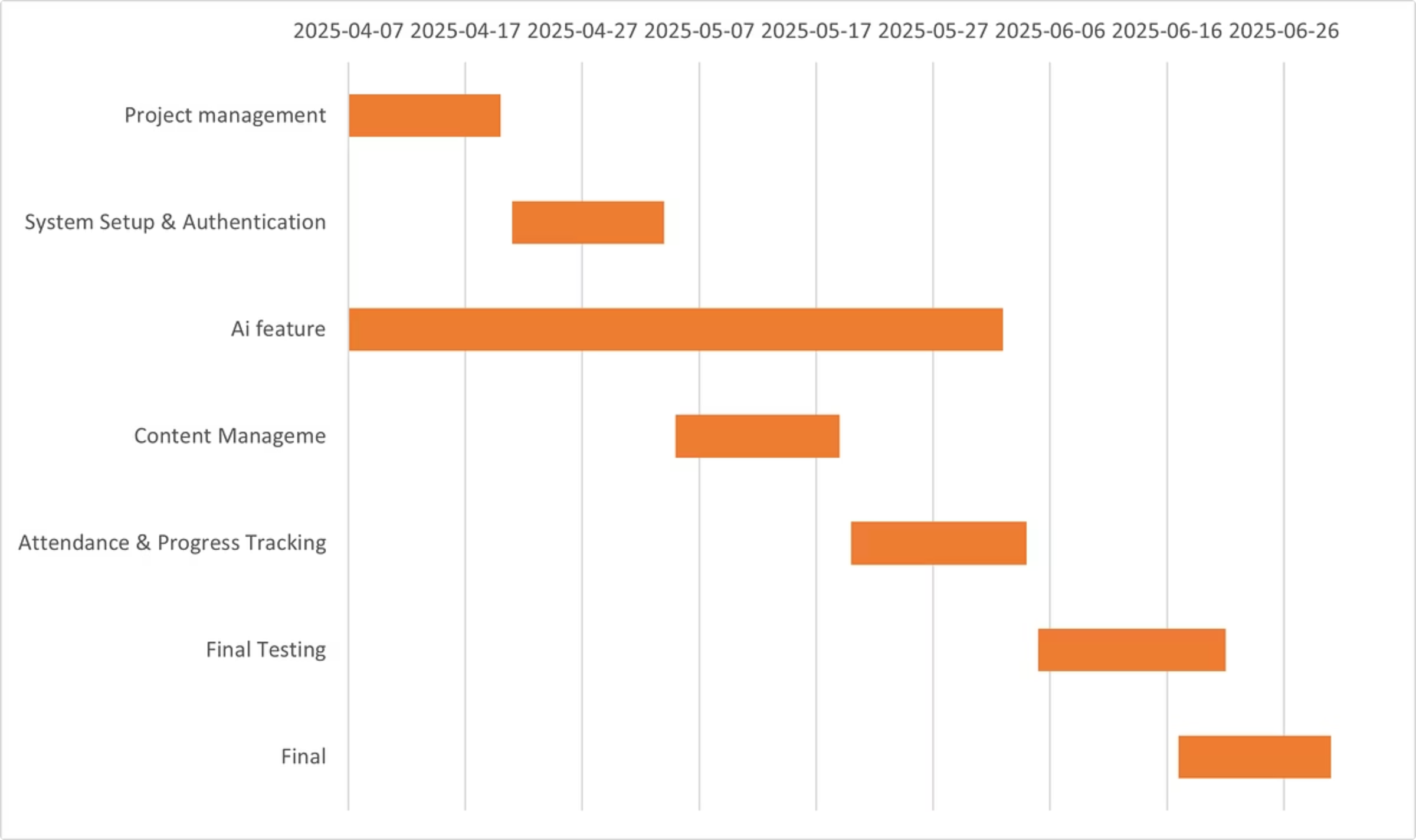


Figure 1 - Gantt Chart V1

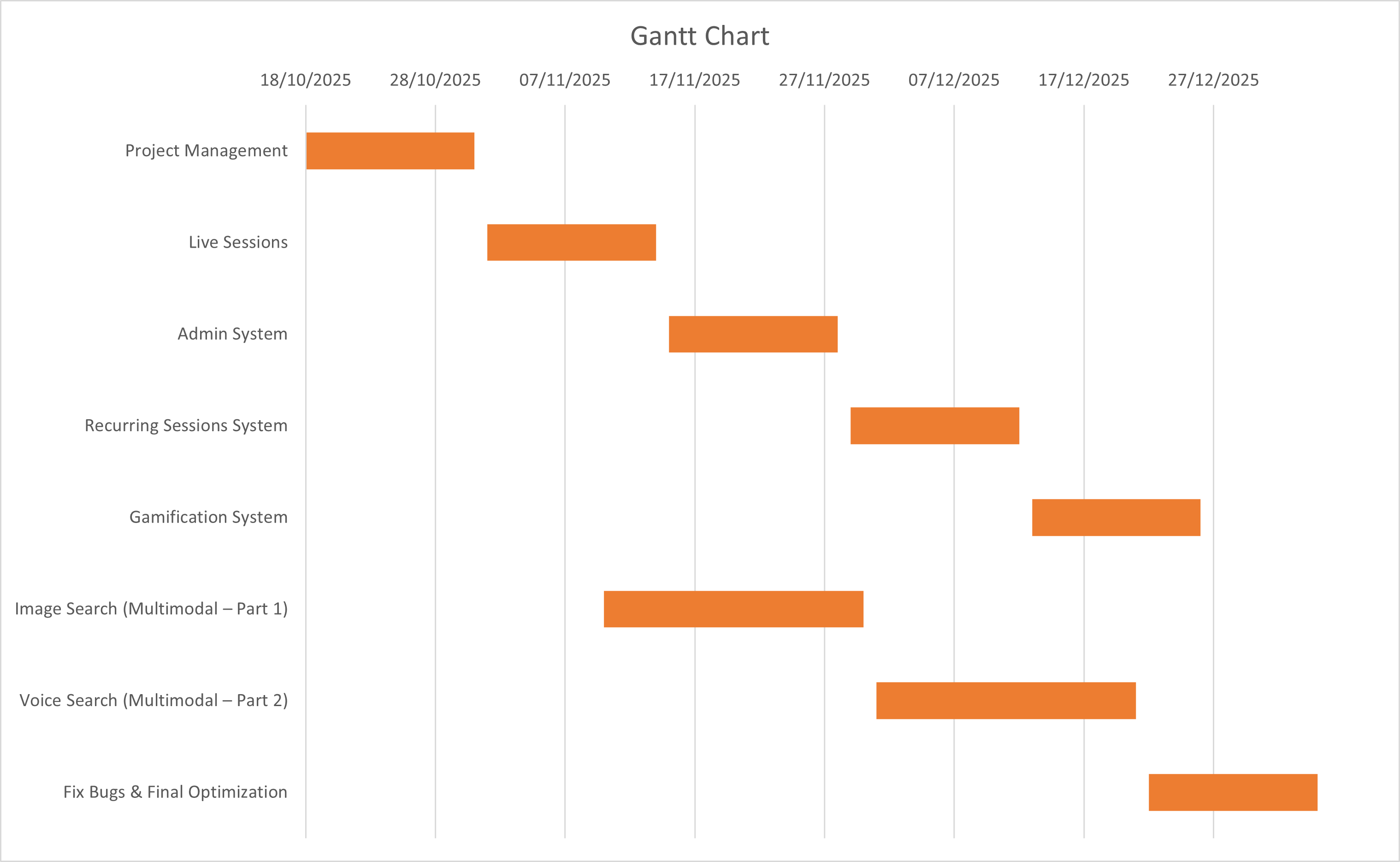


Figure 2 - Gantt Chart V2

## **3.3 System Initial Analysis**

### 3.3.1 Introduction

This section presents the initial analysis and high-level design of the **AI-Powered Complementary Education System**, focusing on how the technical structure aligns with the educational needs of out-of-school children.

The system is designed to provide:

* Accessible digital learning content
* Interactive live teacher–student communication
* AI-assisted homework support
* Motivational gamification
* Multimodal intelligent search

This chapter outlines the system’s primary functional requirements, major components, and early UML representations that illustrate the platform’s architecture and user interactions.

### **3.3.2 High-Level Analysis**

The platform is developed as an inclusive, modular, and intelligent learning environment offering structured lessons, teacher-led live sessions, and AI-powered assistance.

By integrating:

* Live 1:1 and recurring sessions
* Gamification (points, badges, streaks)
* Supervisor-guided learning
* AI Homework Helper
* Image-based and voice-based multimodal search

the system enhances engagement and supports continuous learning for disadvantaged students. Its modular architecture allows easy expansion of features and parallel development of new AI components.

#### 3.3.2.1 Actors and Interactions Overview

This section defines the system’s primary actors and summarizes how each interacts with the platform. These actors form the foundation for the use-case diagrams and system flow models.

**Primary Actors**

**1. Student**

* Accesses simplified educational content (videos, PDFs)
* Joins live 1:1 sessions and recurring sessions
* Uses the **AI Homework Helper**
* Uses **image search** and **voice search** to find answers
* Completes quizzes and receives scores
* Tracks progress via a personalized dashboard
* Earns points, badges, and streaks through the gamification system

**2. Teacher**

* Uploads and organizes lessons
* Creates and manages live sessions (1:1 sessions and Recurring session templates)
* Reviews student assignments and provides feedback
* Uses AI-generated quizzes (future extension)
* Monitors lesson engagement

**3. Educational Supervisor**

* Assigns students to live sessions
* Manages student groups for recurring sessions
* Monitors student performance and attendance
* Communicates with students through messaging
* Oversees student progress and engagement

**4. Admin**

* Manages supervisor accounts (create, edit, activate/deactivate)
* Monitors high-level platform health
* Does not engage in educational tasks

### **3.3.3 Use Case Diagram**

Use-case diagrams illustrate how users interact with a system by outlining the system’s expected functions and the goals users aim to achieve, helping to define and analyze system requirements.

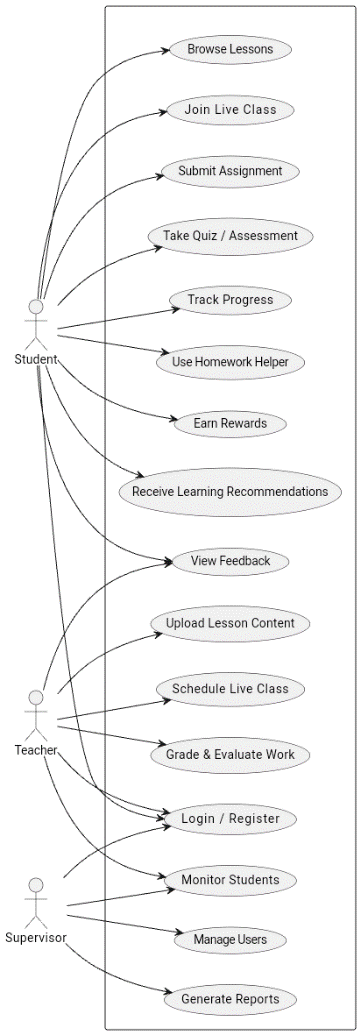
****

Figure 3 High Level Use case diagram

### **3.3.4 System Requirements**

#### 3.3.4.1 High-Level Requirements

**1. User Management**

* Secure authentication using JWT
* Role-based access (Student / Teacher / Supervisor / Admin)
* Personalized dashboards for each role

**2. Educational Content Management**

* Teachers upload content (videos, PDFs, images, exercises)
* Content categorized by subject and educational level
* Students can view lesson materials anytime

**3. Live Sessions (1:1 and Recurring)**

* Integration with **Jitsi** for hosting live sessions
* Teachers create and manage live sessions
* Supervisors assign students to sessions (1:1 or groups)
* Students join sessions from their dashboards
* Automatic notifications for upcoming sessions
* Attendance tracking (manual + automatic when possible)

**4. Assessments & Performance Tracking**

* Quizzes available after lessons
* Teachers can add questions manually
* System records scores and generates basic performance summaries
* Supervisors review student progress

**5. Dashboards (Teacher – Supervisor – Student)**

* Teachers: manage content, sessions, assignments
* Supervisors: monitor student engagement, schedule sessions
* Students: check schedule, lessons, rewards, progress

**6. Gamification System**

* Points for completing lessons and attending sessions
* Badges for achievements
* Daily learning streak tracker
* (Leaderboard is optional for future expansion)

**7. AI & Multimodal Intelligence**

* **AI Homework Helper:** Provides guided explanations for queries
* **Image Search (this semester):** CLIP-based visual search
* **Voice Search (this semester):** Custom-trained S2R model
* (Personalized AI learning recommendations are future scope)

### **3.3.5 High-Level System Design**

The system is built using a **3-Tier Architecture**, ensuring a clear separation between the user interface, core logic, and data storage. This structure enhances scalability, security, and maintainability—making it ideal for an educational platform developed by a small team and expected to grow over time.

#### 1. Presentation Layer

This layer includes all user-facing components, providing intuitive interfaces for students, teachers, supervisors, and admins. It manages:

* Lesson viewing (videos, PDFs, interactive content)
* Live session interfaces
* Student dashboards
* Teacher content management pages
* Supervisor monitoring dashboards
* Authentication and session handling

The goal of this layer is to ensure a simple, responsive, and accessible experience across devices.

#### 2. Application Layer (Business Logic Layer)

This is the core of the system, responsible for processing all operations. It includes key modules such as:

* Lesson & content management
* Assignments and quizzes
* Progress tracking engine
* Live session scheduler
* **AI-powered Homework Helper**
* Recommendation algorithms
* **Gamification** (points, badges, achievements)
* Notifications and messaging services

This layer enforces system rules, manages workflows, and coordinates interactions between the interface and the database.

#### 3. Data Access Layer

This layer ensures secure communication with the system database. It handles:

* User accounts and roles
* Lessons and uploaded content
* Student progress data
* Assessment scores
* Attendance logs
* AI analytics records
* Rewards and gamification history

It guarantees efficient, secure, and well-structured data storage.

The modular nature of the 3-tier architecture allows future expansions—such as full live-class integration, offline support, and extended analytics—without major restructuring of the system.

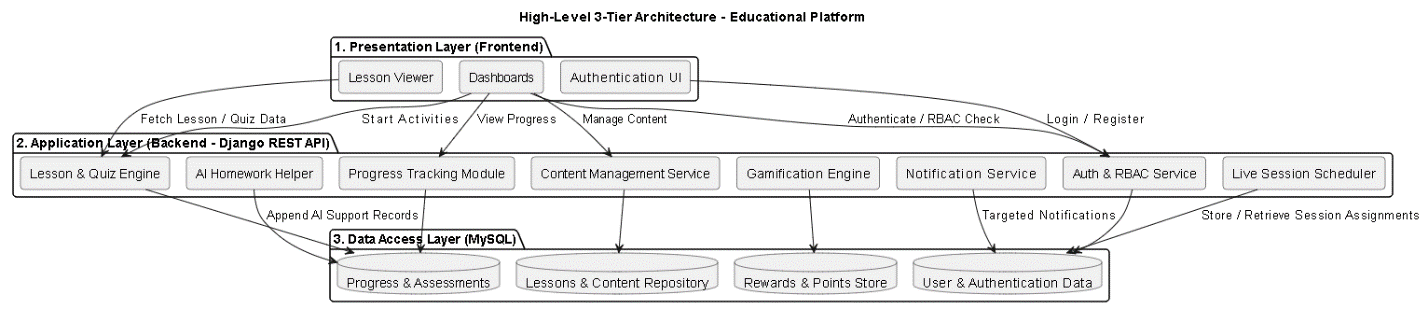


Figure 4 High level Architecture

## **3.4 Summary**

The updated system architecture provides a structured and scalable foundation that aligns closely with the use case model and the core needs of students, teachers, and supervisors. Each **module** within the system directly corresponds to key user activities—such as accessing lessons, managing content, scheduling classes, submitting assignments, monitoring progress, and receiving **AI-powered guidance**.

By adopting a **3-Tier Architecture**, the platform ensures a clean separation between the presentation layer, business logic, and data management. This separation enhances maintainability, supports future expansions, and allows new features—such as advanced analytics, extended AI support, and live-class integration—to be added without disrupting existing functionality.

Overall, the architecture ensures that the platform remains reliable, extensible, and capable of supporting the long-term educational needs of out-of-school children.

**Chapter 4 - System Analysis**

### **4.1 Introduction**

This chapter presents a comprehensive analysis of the **AI-Powered Complementary Education System**. It outlines the system’s core functionality and technical structure, designed to bridge the educational gap for out-of-school children. The chapter defines both the functional and non-functional requirements and identifies the primary actors: **Students, Teachers, Supervisors, and Admins**. It details the system's modular design, which integrates a foundational **Web-based Learning Management System (LMS)** with an advanced **Dual-System AI Engine**. The goal is to lay a solid foundation for the system's design and implementation using a modern **3-Tier architecture** that supports scalability, security, and multimodal intelligence.

### **4.2 Purpose**

The purpose of this system is to develop an accessible, inclusive, and intelligent educational platform. It serves a dual purpose:

* **Foundational Access:** Allowing teachers and volunteers to deliver structured lessons (live and recorded) to children who have dropped out of school, ensuring continuity of education.
* **Intelligent Empowerment:** Providing students with self-study tools powered by AI (**Smart Homework Helper**) that enable them to search for information using **Voice, Text, and Images**, overcoming literacy and digital barriers.

### **4.3 Project Scope**

The scope of the AI-Powered Complementary Education System extends beyond standard e-learning functionalities to establish a holistic, intelligent, and secure educational environment.

Fundamentally, the project is built upon a robust **Core Platform** that manages the complete user lifecycle, implementing strict **Role-Based Access Control (RBAC)** to differentiate between Students, Teachers, Supervisors, and Administrators. This foundation ensures secure access and comprehensive content management capabilities, allowing educators to organize and distribute diverse educational materials efficiently.

Building on this foundation, the system incorporates advanced **Educational and Interactive Services** designed to replicate the classroom experience digitally. This includes a fully integrated **Live Session Management System** that leverages Jitsi Meet for secure, real-time instruction, coupled with an automated **Recurring Sessions engine**. This automation handles the complex logic of scheduling weekly classes, thereby reducing the administrative burden on teachers and ensuring consistent learning opportunities for students.

To maintain platform integrity, the scope also encompasses **Administrative and Security Services**, featuring automated protocols for user deactivation and session invalidation, ensuring that access rights are strictly enforced at all times.

### **4.4 Requirements Elicitation**

The requirements for this platform were gathered through a comparative analysis of existing educational platforms and an iterative **Agile Scrum** methodology.

* **Baseline Requirements:** Initial analysis focused on the essential needs for remote learning: simple user interfaces, reliable video streaming, and robust user management (addressed in early Sprints).
* **Advanced Requirements:** Further analysis identified critical gaps in accessibility for disadvantaged children, specifically the difficulty in searching through "black box" content like videos and scanned PDFs. This led to the definition of requirements for the **AI Engine** (addressed in later Sprints), ensuring the system is not just functional, but also intelligent and accessible to students with limited digital skills

### **4.5 Requirements Tables**

|  |  |  |  |
| --- | --- | --- | --- |
| REQ ID | Title | Category | Type |
| REQ-2.1 | User Registration | Authentication | Functional |
| REQ-2.2 | Login | Authentication | Functional |
| REQ-2.3 | User Profile Viewing | User Management | Functional |
| REQ-2.4 | User Profile Editing | User Management | Functional |
| REQ-2.5 | Logout | Authentication | Functional |
| REQ-2.6 | Reset password | User Management | Functional |
| REQ-2.7 | Form Validation on Frontend | Frontend Validation | Non-Functional |
| REQ-2.8 | Input Error Messages on Frontend | Frontend Validation | Non-Functional |
| REQ-2.9 | Secure Password Storage (Hashing) | Security | Technical |
| REQ-2.10 | Session Expiry (Token Expiration) | Security | Technical |
| REQ-4.1 | Upload Video Lesson | Content Management | Functional |
| REQ-4.2 | Upload PDF Lesson | Content Management | Functional |
| REQ-4.3 | Upload Exercises (MCQ) | Content Management | Functional |
| REQ-4.4 | View My Lessons | Content Management | Functional |
| REQ-4.5 | Update Lesson | Content Management | Functional |
| REQ-4.6 | Delete Lesson | Content Management | Functional |
| REQ-4.7 | Browse Assigned Lessons | Student Interaction | Functional |
| REQ-4.8 | Select Lesson | Student Interaction | Functional |
| REQ-4.9 | Download PDF Lessons | Student Interaction | Functional |
| REQ-4.10 | Solve Exercises | Student Interaction | Functional |
| REQ-4.11 | View Student Progress | Advisor Tools | Functional |
| REQ-4.12 | Assign Lessons to Student | Advisor Tools | Functional |
| REQ-4.13 | Search Lessons by Filter | Advisor Tools | Functional |
| REQ-5.1 | Track Manual Attendance | Attendance Management | Functional |
| REQ-5.2 | Track Automatic Attendance | Attendance Management | Functional |
| REQ-5.3 | View Quiz Results (Student) | Progress Tracking | Functional |
| REQ-5.4 | View Students’ Quiz Performance | Progress Tracking | Functional |
| REQ-5.5 | View All Students' Quiz Results | Progress Reporting | Functional |
| REQ-5.6 | Submit Feedback by Teacher | Feedback System | Functional |
| REQ-5.7 | View Feedback (Student) | Feedback System | Functional |
| REQ-5.8 | View Student Answers | Evaluation | Functional |
| REQ-LS-01 | Create Live Session | Live Sessions | Functional |
| REQ-LS-02 | Edit Live Session | Live Sessions | Functional |
| REQ-LS-03 | Cancel Live Session | Live Sessions | Functional |
| REQ-LS-04 | Join Live Session (Host) | Live Sessions | Functional |
| REQ-LS-05 | View Teacher's Sessions | Live Sessions | Functional |
| REQ-LS-06 | View Assigned Students List | Live Sessions | Functional |
| REQ-LS-07 | View All Live Sessions | Live Sessions | Functional |
| REQ-LS-08 | Assign Students to Sessions | Live Sessions | Functional |
| REQ-LS-09 | Unassign Students from Sessions | Live Sessions | Functional |
| REQ-LS-10 | Modify Student Assignments | Live Sessions | Functional |
| REQ-LS-11 | View Assigned Sessions | Live Sessions | Functional |
| REQ-LS-12 | Join Live Session (Student) | Live Sessions | Functional |
| REQ-LS-13 | View Session Details | Live Sessions | Functional |
| REQ-ADS-01 | Admin Login | Admin System | Functional |
| REQ-ADS-02 | View Advisors List | Admin System | Functional |
| REQ-ADS-03 | Create New Advisor Account | Admin System | Functional |
| REQ-ADS-04 | Delete Advisor Account | Admin System | Functional |
| REQ-ADS-05 | Reset Advisor Password | Admin System | Functional |
| REQ-ADS-06 | Activate Advisor Account | Admin System | Functional |
| REQ-ADS-07 | Deactivate Advisor Account | Admin System | Functional |
| REQ-ADS-08 | Invalidate Active Sessions | Admin System | Functional |
| REQ-ADS-09 | View System Statistics | Admin System | Functional |
| REQ-RS-01 | Create Session Templates | Recurring Sessions | Functional |
| REQ-RS-02 | View Session Templates | Recurring Sessions | Functional |
| REQ-RS-03 | Edit Session Templates | Recurring Sessions | Functional |
| REQ-RS-04 | Delete Session Templates | Recurring Sessions | Functional |
| REQ-RS-05 | Pause Template Status | Recurring Sessions | Functional |
| REQ-RS-06 | Resume Paused Templates | Recurring Sessions | Functional |
| REQ-RS-07 | End Template Permanently | Recurring Sessions | Functional |
| REQ-RS-08 | View Generated Sessions | Recurring Sessions | Functional |
| REQ-RS-09 | View Template Statistics | Recurring Sessions | Functional |
| REQ-RS-10 | View All Session Templates (Advisor) | Advisor Tools | Functional |
| REQ-RS-11 | Assign Templates to Students/Groups | Advisor Tools | Functional |
| REQ-RS-12 | Create Student Groups | Advisor Tools | Functional |
| REQ-RS-13 | Manage Student Groups | Advisor Tools | Functional |
| REQ-RS-14 | View Available Students | Advisor Tools | Functional |
| REQ-RS-15 | View Assigned Templates (Student) | Student Interaction | Functional |
| REQ-RS-16 | Track Upcoming Sessions | Student Interaction | Functional |
| REQ-RS-17 | View Template Information | Student Interaction | Functional |
| REQ-AI-01 | Upload Image for Search (CLIP) | AI Engine | Functional |
| REQ-AI-02 | Voice Search (Whisper ASR) | AI Engine | Functional |
| REQ-AI-03 | Search Inside Videos (Metadata) | AI Engine | Functional |
| REQ-AI-04 | Smart Homework Helper (RAG) | AI Engine | Functional |

Table 10 Requirements Table

# Release 1: 4.6 Sprint 2 Analysis

In this section, we present the analytical study for Sprint 2, which focuses on implementing the User Management and Authentication System for the Educational Support Platform. This sprint lays the foundation for secure and personalized access to the platform by handling user registration, login, account activation, profile access, and logout functionality.

The analysis includes identifying and modeling the primary functional and non-functional requirements using Unified Modeling Language (UML) tools. This includes use case diagrams, sequence diagrams, and activity diagrams to illustrate the system’s behavior and how users interact with the backend APIs. This structured methodology ensures that the implemented components are aligned with the platform’s goals of usability, security, and scalability.

### Sprint 2 Requirements Covered

The following requirements will be implemented and validated during Sprint 2:

* **REQ-2.1** – The system shall allow users (guests) to register an account using their email, password, full name, and role (Student or Teacher).
* **REQ-2.2** – The system shall allow authenticated users to log in using a JWT-based token system, with account activity verification.
* **REQ-2.3** – The system shall allow users to view their profile data after successful authentication.
* **REQ-2.4** – The system shall allow users to edit their profile information, such as full name, age, region, or specialization.
* **REQ-2.5** – The system shall allow users to securely log out, terminating the local session on the frontend.
* **REQ-2.6** – The frontend shall validate form inputs before submission to the backend.
* **REQ-2.7** – The frontend shall display clear and user-friendly error messages for invalid input or API errors.
* **REQ-2.8** – The system shall securely store passwords using a hashing algorithm.
* **REQ-2.9** – The system shall enforce session expiration through JWT token expiration settings.

### 4.6.1 Sprint 2 - Requirements Modeling

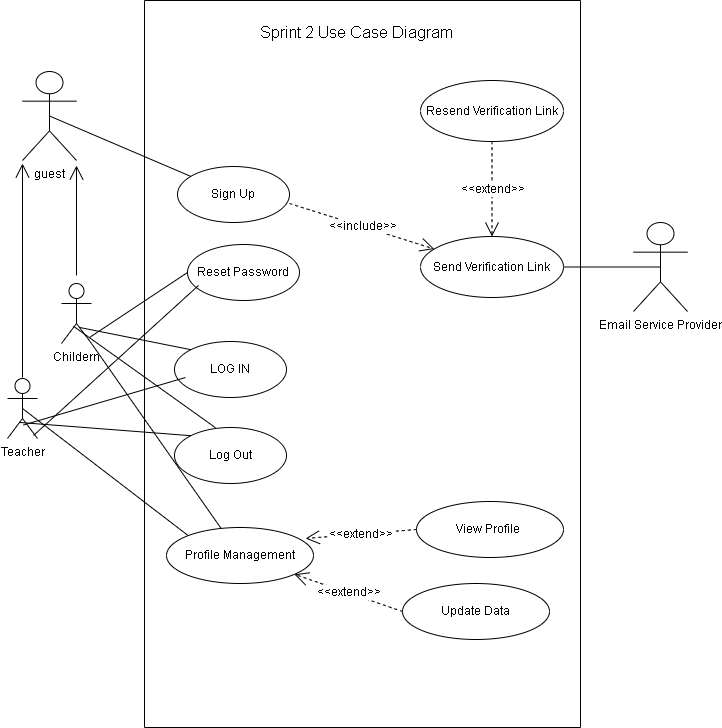


Figure 5 sprint 2 usecase diagram - V1

#### UC-2.1 – Register New User

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-2.1 |
| **Use Case Name** | Register New User |
| **Actor** | Guest |
| **Pre-conditions** | User is not authenticated or registered. |
| **Main Scenario** | 1. User fills out the registration form (name, email, password, role).  2. System validates input.  3. System creates user with is\_active = false.  4. System sends email verification link.  5. System displays success message. |
| **Alternative Scenario** | - Email already exists → show 'email in use' error.  - User re-enters a different email. |
| **Exceptional Scenario** | - Email sending fails → show 'verification failed' message. |
| **Post-condition** | User account created but inactive until email is verified. |

Table 11 Register New User UCS

#### 

Figure 6 - Register New User Activity

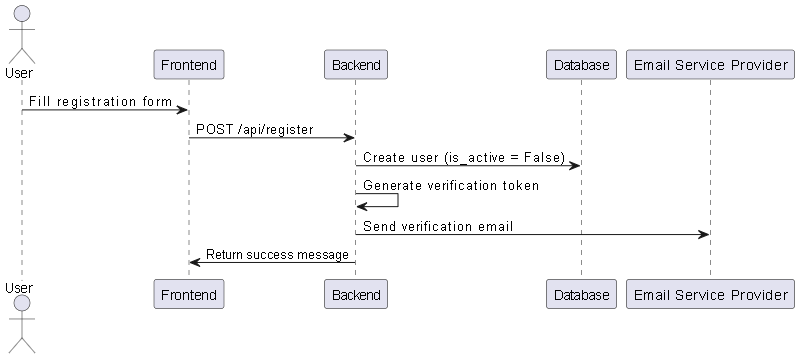


Figure 7 - Register New User Sequence

#### UC-2.2 – Login User

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-2.2 |
| **Use Case Name** | Login User |
| **Actor** | Student / Teacher |
| **Pre-conditions** | User has an active account. |
| **Main Scenario** | 1. User submits login form.  2. System checks credentials.  3. If correct and active → issue JWT token.  4. Redirect based on role. |
| **Alternative Scenario** | - Incorrect password → show error message. |
| **Exceptional Scenario** | - Account not active → show activation required error. |
| **Post-condition** | User logged in with token issued. |

Table 12 - Login User UCS

#### 

Figure 8 - login Activity

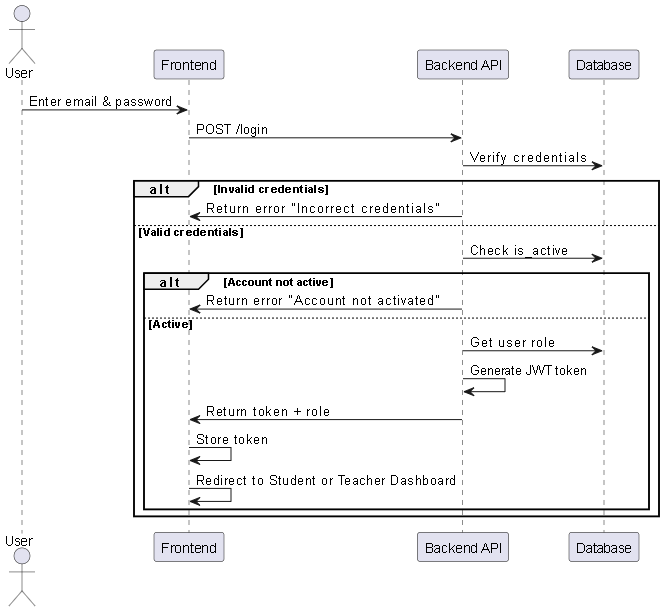


Figure 9 - Login Sequence

#### UC-2.3 – View Profile

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-2.3 |
| **Use Case Name** | View Profile |
| **Actor** | Student / Teacher |
| **Pre-conditions** | User is authenticated. |
| **Main Scenario** | 1. User requests profile data.  2. System validates token.  3. System fetches user data.  4. System returns profile info to frontend. |
| **Alternative Scenario** | - Only basic info available → show name and email. |
| **Exceptional Scenario** | - Token expired → redirect to login. |
| **Post-condition** | User profile is displayed. |

Table 13 - View profile UCS

#### 

Figure 10 - View profile Activity

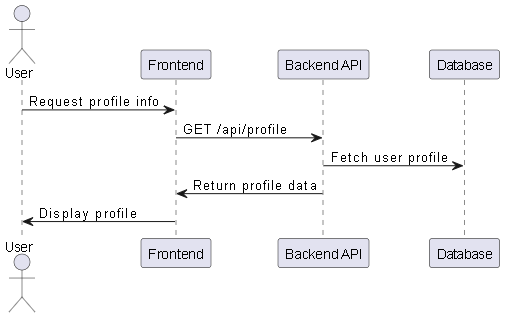


Figure 11 - View profile Sequence

#### UC-2.4 – Edit Profile

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-2.4 |
| **Use Case Name** | Edit Profile |
| **Actor** | Student / Teacher |
| **Pre-conditions** | User is authenticated. |
| **Main Scenario** | 1. User edits and submits updated info.  2. System validates token and new input.  3. System updates user info in DB.  4. System shows success message. |
| **Alternative Scenario** | - No changes submitted → show 'no updates' warning. |
| **Exceptional Scenario** | - Invalid token → redirect to login. |
| **Post-condition** | Profile changes saved to database. |

Table 14 - Edit Profile UCS

#### 

Figure 12 - Edit Profile Activity

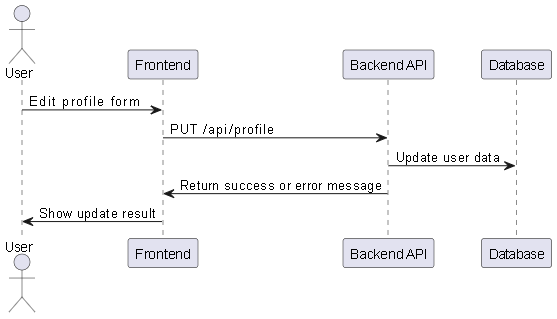


Figure 13 - Edit profile Sequence

#### UC-2.5 – Logout

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-2.5 |
| **Use Case Name** | Logout |
| **Actor** | Student / Teacher |
| **Pre-conditions** | User is logged in. |
| **Main Scenario** | 1. User clicks logout button.  2. System removes token from storage.  3. User redirected to login page. |
| **Alternative Scenario** | - User closes tab without clicking logout. |
| **Exceptional Scenario** | - Token missing → force redirect to login. |
| **Post-condition** | User session terminated. |

Table 15 - Logout UCS

#### 

Figure 14 - logout Activity

#### 

Figure 15 - logout Sequence

#### UC-2.6 – Reset Password

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-2.6 |
| **Use Case Name** | Reset Password |
| **Actor** | Student / Teacher |
| **Pre-conditions** | User has a registered email address. |
| **Main Scenario** | 1. User clicks “Forgot Password” on login page.  2. System displays a form to enter email.  3. User submits their registered email.  4. System sends a password reset link to the email.  5. User clicks the link from the email.  6. System displays form to enter and confirm new password.  7. User submits new password.  8. System updates the password and redirects user to login page with success message. |
| **Post-condition** | User password is updated and access is restored. |

Table 16 - Reset Password UCS

# 

Figure 16 - Reset Password Activity

# e

Figure 17 - Reset Password sequence

# 4.7 Sprint 4 Analysis

In this section, we present the analytical study for Sprint 4, which focuses on implementing the Content Management and Lesson Assignment features of the Educational Support Platform. This sprint introduces the ability for teachers to upload and manage various types of lesson materials (videos, PDFs, and exercises), while empowering the advisor to assign appropriate lessons to students based on their educational level or needs.

This sprint also refines the student experience by allowing them to browse only the lessons assigned to them, select a lesson of interest, and then either download the lesson content or solve associated exercises. Additionally, the advisor can filter and search lessons to streamline the assignment process.

The analysis includes identifying and modeling the key functional and non-functional requirements using Unified Modeling Language (UML) tools. These include use case diagrams, activity diagrams, and sequence diagrams to illustrate how each actor interacts with the system and how internal components coordinate to deliver these features. This structured design approach ensures that lesson workflows are role-driven, file uploads are validated and secure, and that content delivery is intuitive and student-centric, laying the foundation for a scalable, content-rich learning platform.

### **Sprint 4 Requirements Covered**

The following requirements will be implemented and validated during Sprint 4:

* **REQ-4.1** – The system shall allow teachers to upload video lessons.
* **REQ-4.2** – The system shall allow teachers to upload PDF lesson materials.
* **REQ-4.3** – The system shall allow teachers to upload exercises in multiple choice question (MCQ) format.
* **REQ-4.4** – The system shall allow teachers to view and manage their uploaded lessons.
* **REQ-4.5** – The system shall allow teachers to update existing lessons.
* **REQ-4.6** – The system shall allow teachers to delete lessons.
* **REQ-4.7** – The system shall allow students to browse the lessons assigned to them by an advisor.
* **REQ-4.8** – The system shall allow students to select a lesson from the assigned list.
* **REQ-4.9** – The system shall allow students to download PDF lesson materials.
* **REQ-4.10** – The system shall allow students to solve MCQ-based exercises linked to the lesson.
* **REQ-4.11** – The system shall allow advisors to view student progress through lesson interaction and quiz results.
* **REQ-4.12** – The system shall allow advisors to assign specific lessons to individual students.
* **REQ-4.13** – The system shall allow advisors to search for lessons using filter criteria such as subject or level.

### 4.7.1 Sprint 4 - Requirements Modeling

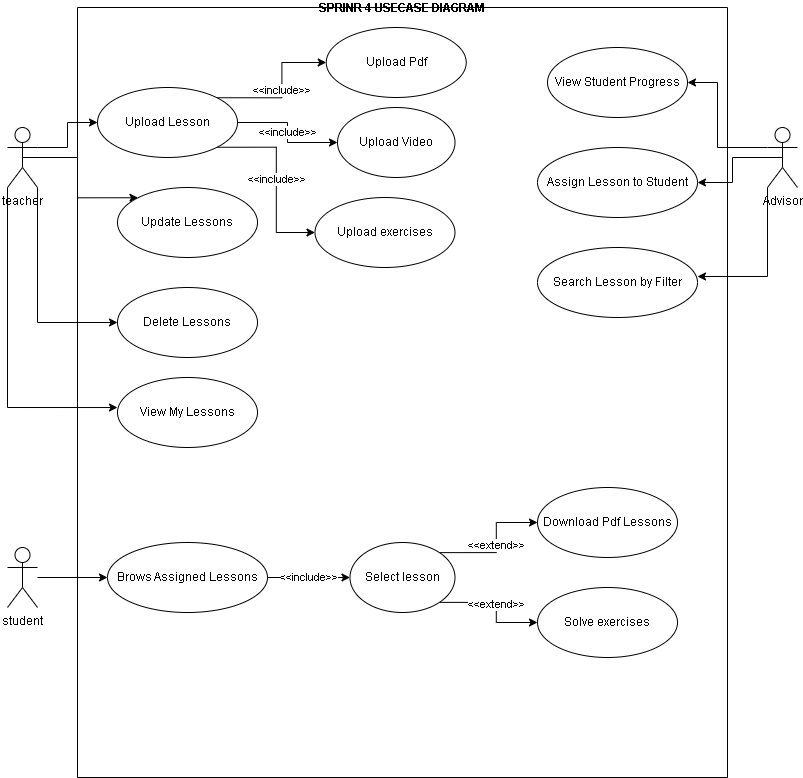


Figure 18 - Sprint 4 use case diagram

#### UC-4.1 – Upload Video Lesson

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.1 |
| **Use Case Name** | Upload Video Lesson |
| **Actor** | Teacher |
| **Pre-conditions** | User is authenticated and has a teacher role. |
| **Main Scenario** | 1. Teacher selects video file.  2. System validates file format and size.  3. System uploads the video and stores metadata.  4. Success message displayed. |
| **Alternative Scenario** | - Upload canceled by teacher. |
| **Exceptional Scenario** | - File format invalid → show error.  - Upload fails → show 'Upload error'. |
| **Post-condition** | Video lesson is stored and associated with the teacher. |

Table 17 - Upload Video UCS

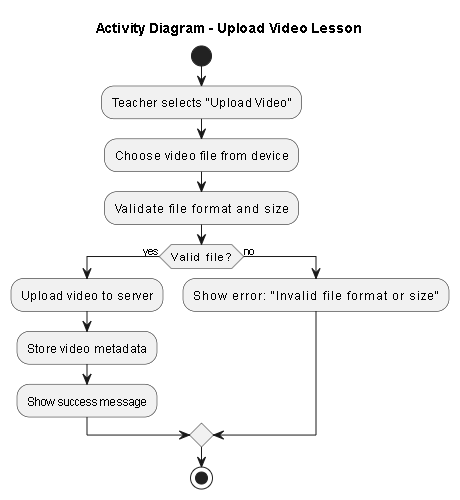


Figure 19 - Upload Video Activity

#### 

Figure 20 - Upload Video Sequence

#### UC-4.2 – Upload PDF Lesson

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.2 |
| **Use Case Name** | Upload PDF Lesson |
| **Actor** | Teacher |
| **Pre-conditions** | User is authenticated and has a Teacher role. |
| **Main Scenario** | 1. Teacher selects PDF file.  2. System validates and uploads the file.  3. Success confirmation is shown. |
| **Alternative Scenario** | - Teacher decides not to upload. |
| **Exceptional Scenario** | - PDF validation fails → show error. |
| **Post-condition** | PDF lesson saved and categorized. |

Table 18 - Upload PDF UCS

#### 

Figure 21 - Upload PDF Activity

#### 

Figure 22 - Upload PDF Sequence

#### UC-4.3 – Upload MCQ Exercises

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.3 |
| **Use Case Name** | Upload MCQ Exercises |
| **Actor** | Teacher |
| **Pre-conditions** | User is authenticated and has a Teacher role. |
| **Main Scenario** | 1. Teacher selects “Upload Exercises”.  2. Teacher enters the MCQ question and answer choices.  3. Teacher marks one or more correct answers.  4. Teacher submits the form.  5. System validates the format.  6. System stores the question and links it to the appropriate lesson. |
| **Alternative Scenario** | - Teacher cancels before submitting the exercise.  - Teacher leaves some required fields empty and is prompted to fill them. |
| **Exceptional Scenario** | - Invalid format or missing answers → show error.  - System/database error during save. |
| **Post-condition** | MCQ exercise is saved and associated with the corresponding lesson. |

Table 19 - Upload MCQ Exercises UCS

#### 

Figure 23 - Upload MCQ Exercises Activity

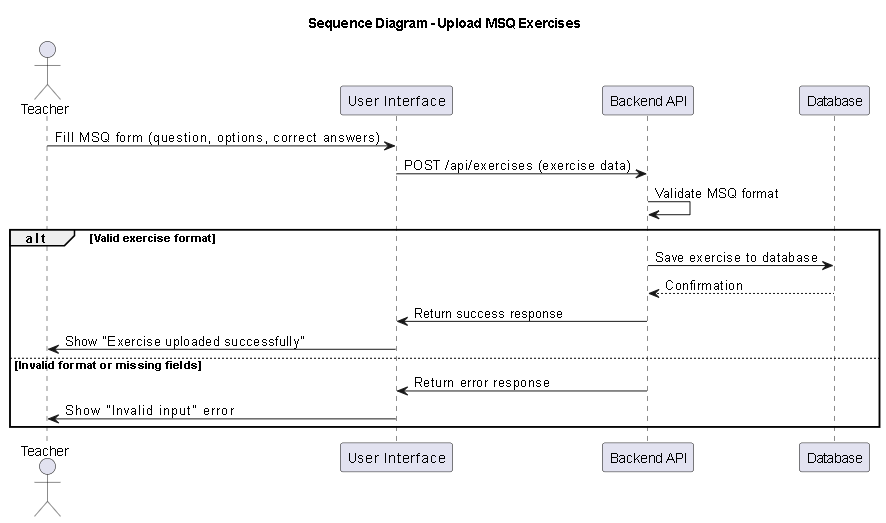


Figure 24 - Upload MCQ Exercises Sequence

#### UC-4.4 – View Uploaded Lessons

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.4 |
| **Use Case Name** | View Uploaded Lessons |
| **Actor** | Teacher |
| **Pre-conditions** | User is logged in as Teacher. |
| **Main Scenario** | 1. Teacher opens lesson dashboard.  2. System fetches and displays list. |
| **Alternative Scenario** | - List is empty → show 'No lessons uploaded'. |
| **Exceptional Scenario** | - Database error → show fallback message. |
| **Post-condition** | Lessons displayed in dashboard. |

Table 20 - View Uploaded Lesson UCS

#### 

Figure 25 - View Uploaded Activity

#### 

Figure 26 - View Uploaded Sequence

#### UC-4.5 – Update Lesson

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.5 |
| **Use Case Name** | Update Lesson |
| **Actor** | Teacher |
| **Pre-conditions** | Lesson exists and belongs to the teacher. |
| **Main Scenario** | 1. Teacher edits lesson content or metadata.  2. System saves changes. |
| **Alternative Scenario** | - Edit canceled before saving. |
| **Exceptional Scenario** | - Update fails due to validation or server error. |
| **Post-condition** | Lesson data updated. |

Table 21 - Update Lesson UCS

#### 

Figure 27 - Update Lesson Activity

#### 

Figure 28 - Update Lesson Sequence

#### UC-4.6 – Delete Lesson

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.6 |
| **Use Case Name** | Delete Lesson |
| **Actor** | Teacher |
| **Pre-conditions** | Lesson exists and is accessible to the teacher. |
| **Main Scenario** | 1. Teacher clicks delete.  2. System confirms and removes lesson. |
| **Alternative Scenario** | - Deletion canceled at confirmation. |
| **Exceptional Scenario** | - Lesson not found or permission denied. |
| **Post-condition** | Lesson removed from the system. |

Table 22 - Delete Lesson UCS

#### 

Figure 29 - Delete Lesson Activity

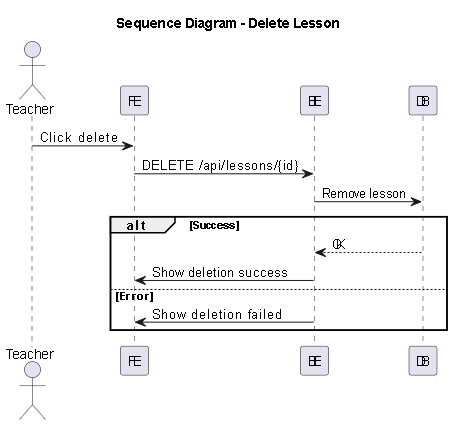


Figure 30 - Delete Lesson Sequence

#### UC-4.7 – Browse Assigned Lessons

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.7 |
| **Use Case Name** | Browse Assigned Lessons |
| **Actor** | Student |
| **Pre-conditions** | Student is authenticated. |
| **Main Scenario** | 1. Student navigates to lessons page.  2. System shows assigned lessons. |
| **Post-condition** | Student can see assigned content. |

Table 23 - Browse Assigned Lesson UCS

#### 

Figure 31 - Browse Assigned Lesson Activity

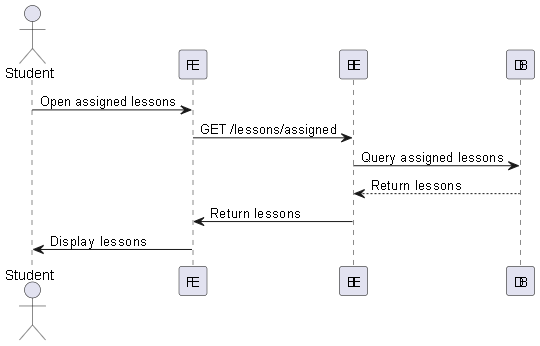


Figure 32 - Browse Assigned Lesson Sequence

#### UC-4.8 – Select Lesson

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.8 |
| **Use Case Name** | Select Lesson |
| **Actor** | Student |
| **Pre-conditions** | Student has lessons assigned. |
| **Main Scenario** | 1. Student clicks on a lesson.  2. Lesson details are displayed. |
| **Post-condition** | Lesson is ready for interaction. |

Table 24 - Select Lesson UCS

#### 

Figure 33 - Select Lesson Activity

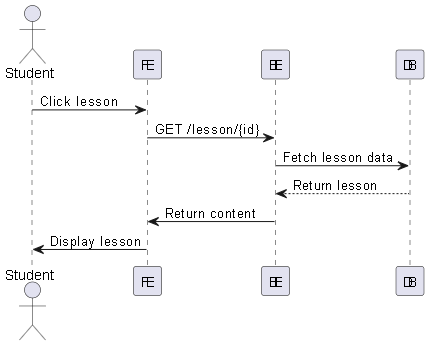


Figure 34 - Select Lesson Sequence

#### UC-4.9 – Download PDF Lessons

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.9 |
| **Use Case Name** | Download PDF Lessons |
| **Actor** | Student |
| **Pre-conditions** | Lesson contains PDF file. |
| **Main Scenario** | 1. Student clicks download.  2. PDF file is streamed or downloaded. |
| **Alternative Scenario** | - No PDF available. |
| **Exceptional Scenario** | - File not found. |
| **Post-condition** | PDF is downloaded. |

Table 25 - Download PDF UCS

#### 

Figure 35 - Download PDF Activity

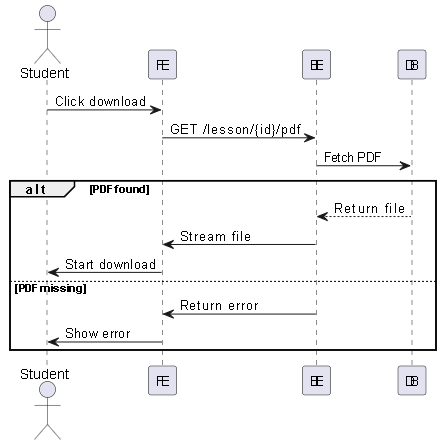


Figure 36 - Download PDF Sequence

#### UC-4.10 – Solve Exercises

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.10 |
| **Use Case Name** | Solve Exercises |
| **Actor** | Student |
| **Pre-conditions** | Lesson includes exercises. |
| **Main Scenario** | 1. Student starts exercise session.  2. Student submits answers.  3. System stores and evaluates. |
| **Post-condition** | Answers are recorded. |

Table 26 - Solve Exercises UCS

#### 

Figure 37 - Solve Exercises Activity

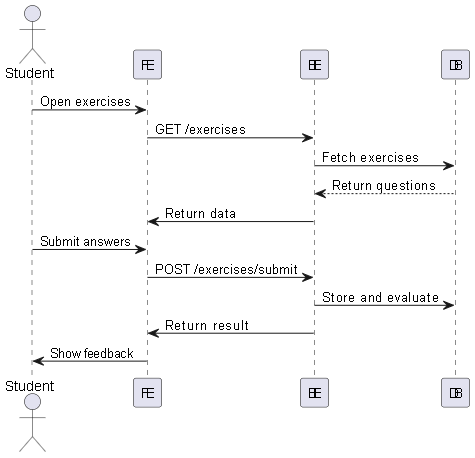


Figure 38 - Solve Exercises Sequence

#### UC-4.11 – View Student Progress

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.11 |
| **Use Case Name** | View Student Progress |
| **Actor** | Advisor |
| **Pre-conditions** | Advisor is authenticated. |
| **Main Scenario** | 1. Advisor selects student profile.  2. System displays progress chart or metrics. |
| **Post-condition** | Advisor can assess learning. |

Table 27- View Student Progress UCS

#### 

Figure 39 - View Student Progress Activity

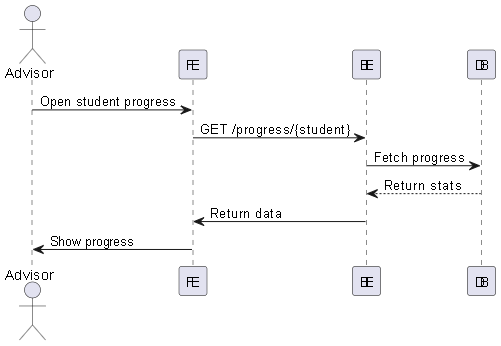


Figure 40 - View Student Progress Sequence

#### UC-4.12 – Assign Lessons to Student

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.12 |
| **Use Case Name** | Assign Lessons to Student |
| **Actor** | Advisor |
| **Pre-conditions** | Advisor is logged in. |
| **Main Scenario** | 1. Advisor selects student.  2. Chooses lessons from list.  3. System links lessons to student. |
| **Post-condition** | Lessons assigned to student. |

Table 28 - Assign Lessons to Student UCS

#### 

Figure 41 - Assign Lessons to Student Activity

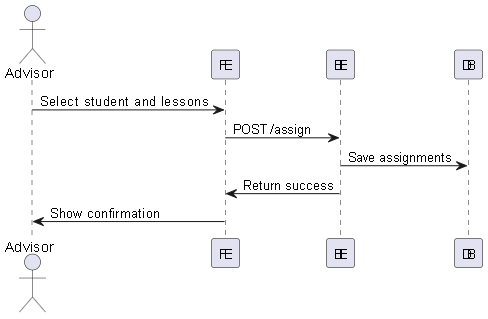


Figure 42 - Assign Lessons to Student Sequence

#### UC-4.13 – Search Lessons by Filter

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-4.13 |
| **Use Case Name** | Search Lessons by Filter |
| **Actor** | Advisor |
| **Pre-conditions** | Advisor is authenticated. |
| **Main Scenario** | 1. Advisor inputs search filters.  2. System shows filtered lessons. |
| **Post-condition** | Advisor views matching results. |

Table 29 - Search Lessons by Filter UCS

# 

Figure 43 - Search Lessons by FilterActivity

# 

Figure 44 - Search Lessons by Filter Sequence

# 4.8 Sprint 5 Analysis

In this section, we present the analytical study for Sprint 5, which centers on implementing the Progress Tracking and Feedback capabilities of the Educational Support Platform. This sprint focuses on enabling students to monitor their academic progress, and allowing teachers and advisors to evaluate student performance based on quiz results.

During this sprint, students gained the ability to view their quiz results, while teachers can review student answers, track performance, and submit individualized feedback. Advisors are granted access to the performance data of all students across the platform, allowing for broader oversight. The feedback submitted by teachers becomes visible to students to promote learning and continuous improvement.

Although attendance tracking was originally part of this sprint, its implementation was postponed as it depends on the live class feature, which has yet to be developed.

The analysis incorporates Unified Modeling Language (UML) diagrams such as use case diagrams, activity diagrams, and sequence diagrams to document user interactions, system behavior, and backend communications. This structured design supports the platform’s goals of usability, clarity, and targeted educational feedback, ensuring students receive appropriate support at each step of their learning journey.

### **Sprint 5 Requirements Covered**

The following requirements will be implemented and validated during Sprint 5:

* **REQ-5.3** – The system shall allow students to view their quiz results.
* **REQ-5.4** – The system shall allow teachers to view quiz performance of their students.
* **REQ-5.5** – The system shall allow advisors to view quiz results of all students.
* **REQ-5.6** – The system shall allow teachers to submit feedback on student performance.
* **REQ-5.7** – The system shall allow students to view feedback given by their teachers.
* **REQ-5.8** – The system shall allow teachers to review student answers to quizzes.

### 4.8.1 Sprint 5 - Requirements Modeling

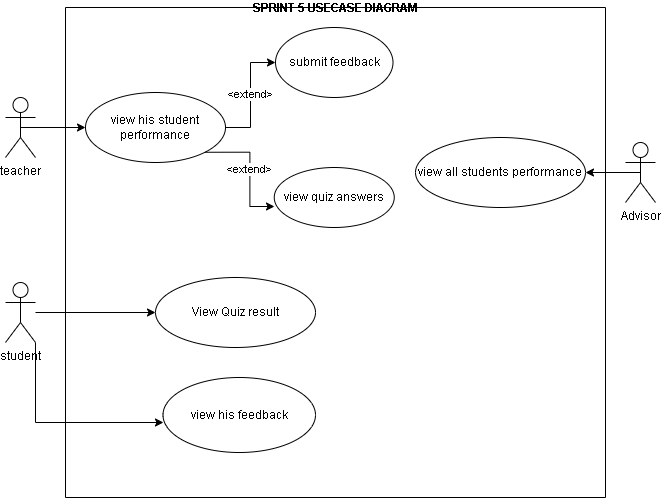


Figure 45 - sprint 5 use case diagram

#### UC-5.4 – View Quiz Results of My Students

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-5.4 |
| **Use Case Name** | View Quiz Results of My Students |
| **Actor** | Teacher |
| **Preconditions** | Teacher must be authenticated and have assigned students. |
| **Main Scenario** | 1. Teacher opens "Student Performance" dashboard.  2. System retrieves quiz results of teacher’s students.  3. Results are displayed per student and per quiz. |
| **Alternative Scenario** | - No students assigned → show “No data to display.” |
| **Exceptional Scenario** | - Backend error → show “Error fetching quiz results.” |
| **Postconditions** | Teacher can review quiz performance of their students. |

Table 30 - View Quiz Result UCS

#### 

Figure 46 - View quiz Activity

#### 

Figure 47 - View quiz Sequence

#### UC-5.5 – View Student Quiz Result

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-5.5 |
| **Use Case Name** | View Student Quiz Result |
| **Actor** | Advisor |
| **Preconditions** | Advisor is authenticated and the student has completed at least one quiz in an assigned lesson. |
| **Main Scenario** | 1. Advisor navigates to the list of students.  2. Advisor clicks on "View Details" for a student.  3. Advisor selects "Performance" tab.  4. System displays lessons assigned to that student.  5. Advisor selects a specific lesson.  6. System retrieves and displays the student’s quiz result for that lesson. |
| **Alternative Scenario** | - Student has no assigned lessons with quizzes → show message: “No quiz data available for this student.” |
| **Exceptional Scenario** | - System error while loading quiz results → show: “Unable to retrieve performance data. Please try again later.” |
| **Postconditions** | Advisor is able to review the student’s quiz performance for the selected lesson. |

Table 31 - View Student Quiz Result UCS

#### 

Figure 48 - View Student Quiz Result Activity

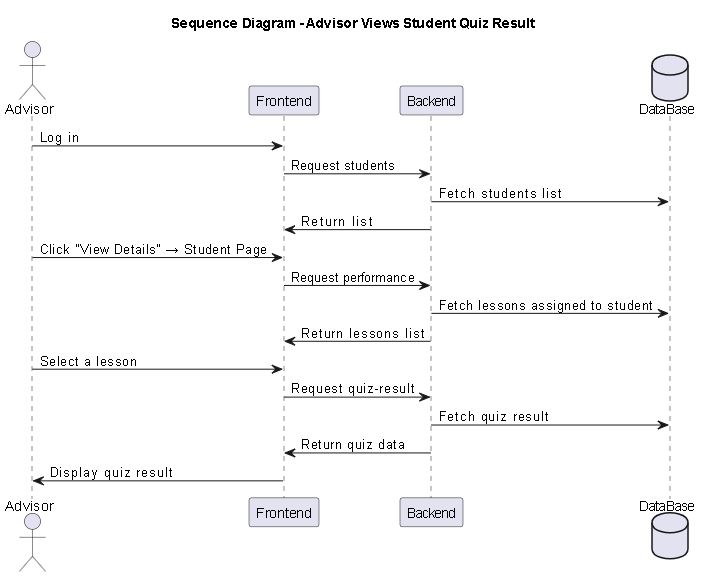


Figure 49 - View Student Quiz Result Sequence

#### UC-5.6 – Submit Feedback

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-5.6 |
| **Use Case Name** | Submit Feedback |
| **Actor** | Teacher |
| **Preconditions** | Teacher is authenticated and student has completed quiz. |
| **Main Scenario** | 1. Teacher opens a student’s quiz result.  2. Clicks "Provide Feedback".  3. Writes and submits feedback.  4. System saves and confirms submission. |
| **Alternative Scenario** | - Feedback left empty → show warning “Please enter feedback text.” |
| **Exceptional Scenario** | - Saving fails → show “Error submitting feedback.” |
| **Postconditions** | Feedback is stored and linked to the student’s quiz. |

Table 32 - Submit Feedback UCS

#### 

Figure 50 - Submit Feedback Activity

#### 

Figure 51 - Submit Feedback Sequence

#### UC-5.7 – View Feedback

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-5.7 |
| **Use Case Name** | View Feedback |
| **Actor** | Student |
| **Preconditions** | Student is authenticated and has received feedback. |
| **Main Scenario** | 1. Student opens "Dashboard" page.  2. System retrieves feedback entries.  3. Feedback is displayed under the corresponding quiz. |
| **Alternative Scenario** | - No feedback exists → show message “You don't have any feedback yet.” |
| **Exceptional Scenario** | - Server error → show “Unable to load feedback.” |
| **Postconditions** | Student can read the feedback left by their teacher. |

Table 33 - View Feedback UCS

#### 

Figure 52 - View Feedback Activity

#### 

Figure 53 - View Feedback Sequence

#### UC-5.8 – View Student Answers

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-5.8 |
| **Use Case Name** | View Student Answers |
| **Actor** | Teacher |
| **Preconditions** | Teacher is authenticated and quiz attempt exists. |
| **Main Scenario** | 1. Teacher selects a quiz and student.  2. System retrieves that student's submitted answers.  3. Each question is shown with selected answers and correctness. |
| **Alternative Scenario** | - Quiz not attempted → show “Student hasn’t completed this quiz.” |
| **Exceptional Scenario** | - Retrieval error → show “Error loading answers.” |
| **Postconditions** | Teacher can review the answers and use them for analysis or feedback. |

Table 34 - View Student AnswersUCS

# 

Figure 54 - View Student Answers Activity

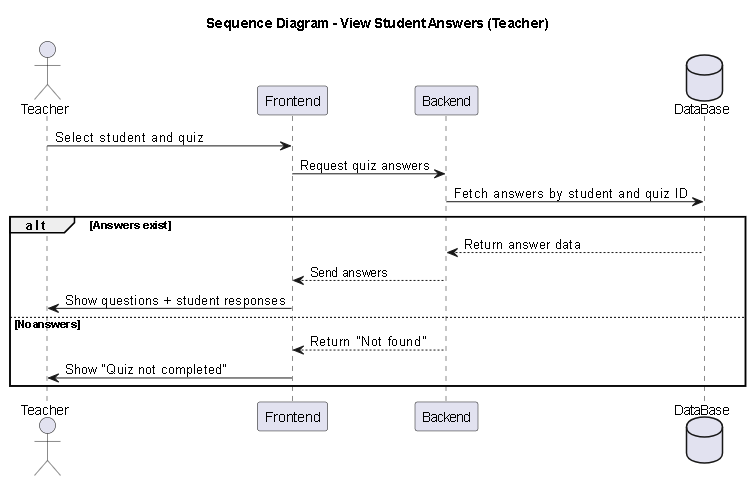


Figure 55 - View Student Answers Sequence

# 4.9 Sprint 3 Analysis

In this section, we present the analytical study for Sprint 3, which focuses on implementing the AI Homework Helper feature in the Educational Support Platform. This sprint introduces an intelligent assistant designed to support students by providing simplified explanations and guidance when solving homework questions. The feature enhances learning accessibility, particularly for students who need additional support outside class time.

The analysis includes identifying and modeling the functional and non-functional requirements of the AI component using Unified Modeling Language (UML) tools. This includes use case diagrams, activity diagrams, and sequence diagrams that illustrate how the student interacts with the AI assistant, how requests are processed, and how responses are delivered in real time. This structured methodology ensures that the AI system is seamlessly integrated into the platform, aligns with educational goals, and remains efficient, intuitive, and scalable.

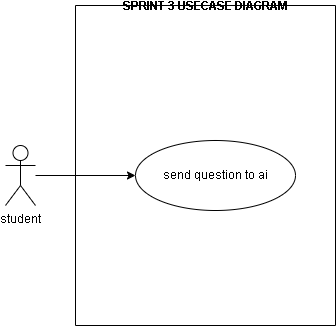
****

Figure 56 - Sprint 3 use case diagram

#### UC-3.1 – Ask AI Homework Helper

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-3.1 |
| **Use Case Name** | Ask AI Homework Helper |
| **Actor** | Student |
| **Preconditions** | Student must be authenticated and have access to the dashboard. |
| **Main Scenario** | 1. Student logs in and accesses the dashboard.  2. Student clicks on the "AI Helper" Icon.  3. System routes the user to the AI Helper page.  4. Student enters a homework-related question.  5. System processes the input using an AI model and generates a simplified explanation.  6. Response is displayed to the student. |
| **Alternative Scenario** | - Student leaves input field empty → system prompts for input. |
| **Exceptional Scenario** | - AI model fails or times out → system displays error message like “AI service unavailable, please try again later.” |
| **Postconditions** | Student receives a useful and simplified explanation related to their homework question. |

Table 35 - Ask AI Homework Helper UCS

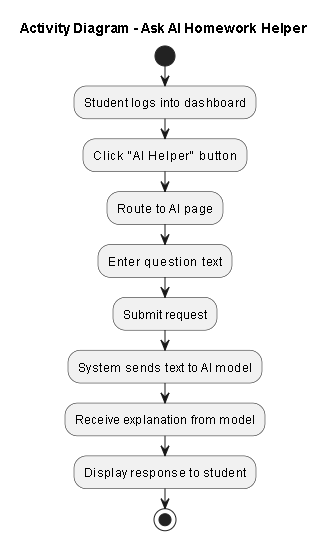


Figure 57 - Ask AI Homework Helper Activity

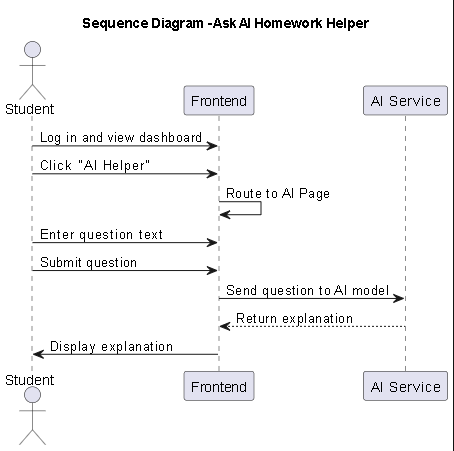


Figure 58 - Ask AI Homework Helper Sequence

Release 2:

# 4.10 Sprint 9: Live Sessions Analysis

In this section, we present the analytical study for Sprint 2, which focuses on implementing the Live Session Management System. This sprint is critical for enabling real-time interaction between teachers and students. The analysis covers the entire session lifecycle: creating sessions, assigning students by advisors, managing schedules, and finally, the secure execution of the session using Jitsi Meet.

The analysis models these interactions using UML diagrams, ensuring that strict validation rules (such as time constraints and assignment checks) are enforced before a student can access the video conference. This ensures a secure and organized educational environment.

### **Sprint 9 Requirements Covered**

The following 13 functional requirements will be implemented and validated during Sprint 9:

* **REQ-LS-01** – The system shall allow Teachers to create a new live session by defining the title, subject, date, time, and duration.
* **REQ-LS-02** – The system shall allow Teachers to update the details of an existing session (e.g., rescheduling) before it starts.
* **REQ-LS-03** – The system shall allow Teachers to cancel or delete a scheduled session, automatically notifying relevant users.
* **REQ-LS-04** – The system shall allow Teachers to view their personal schedule of upcoming and past sessions.
* **REQ-LS-05** – The system shall allow Advisors to view all scheduled sessions across the platform for monitoring purposes.
* **REQ-LS-06** – The system shall allow Advisors to assign specific students or groups to a scheduled live session.
* **REQ-LS-07** – The system shall allow Students to view their personalized schedule of assigned sessions.
* **REQ-LS-08** – The system shall allow Users (Students/Teachers) to view full Session Details, including instructor name and description.
* **REQ-LS-09** – The system shall allow Teachers to Start the session, generating a secure Jitsi Meet host link.
* **REQ-LS-10** – The system shall allow Students to Join the session, generating a secure Jitsi Meet participant link.
* **REQ-LS-11** – The system shall enforce Time Validation, preventing students from joining before the scheduled start time.
* **REQ-LS-12** – The system shall enforce Access Validation, ensuring only assigned students can generate a joining link.
* **REQ-LS-13** – The system shall automatically Track Attendance by recording the timestamp when a student joins the session.

### **4.10.1 Sprint 9 - Requirements Modeling**

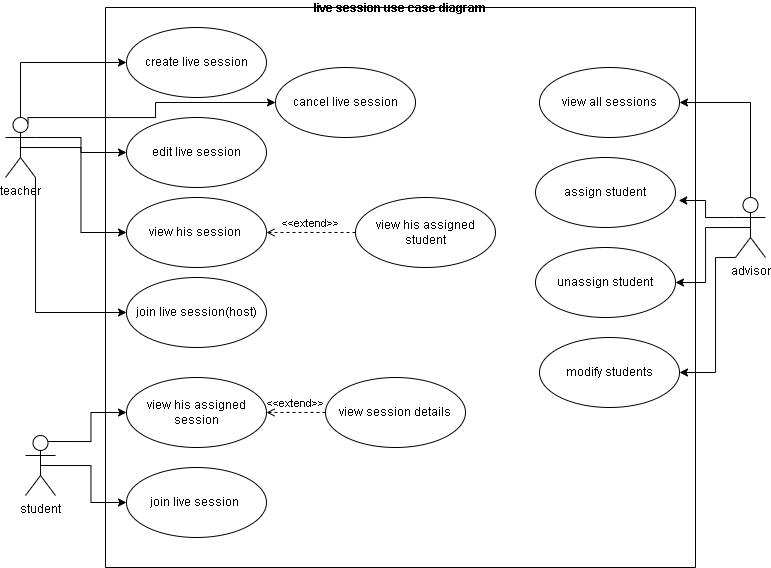


Figure 59 - Sprint 9 Use case diagram

### **Use Case Specifications**

#### UC-LS-01 – Create Individual Live Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-01 |
| **Use Case Name** | Create Individual Live Sessions |
| **Actor** | Teacher |
| **Pre-conditions** | User is authenticated as teacher. |
| **Main Scenario** | 1. Teacher clicks "Create Session" button.  2. System displays session creation form.  3. Teacher fills required fields (title, subject, level, datetime, duration).  4. Teacher submits form.  5. System validates input data.  6. System creates session with PENDING status.  7. System generates unique Jitsi room name.  8. System saves session to database.  9. System displays success message. |
| **Alternative Scenario** | - Invalid data → System shows validation errors.  - Past date selected → System shows "cannot schedule in past" error. |
| **Exceptional Scenario** | - Network error → System shows retry option.  - Server error → System shows error message.  - Database error → System logs error and shows user-friendly message. |
| **Post-condition** | New session created and visible in teacher's session list. |

Table 36 - Create Individual Live Sessions UCS

#### 

Figure 60 - Create Individual Live Sessions Activity

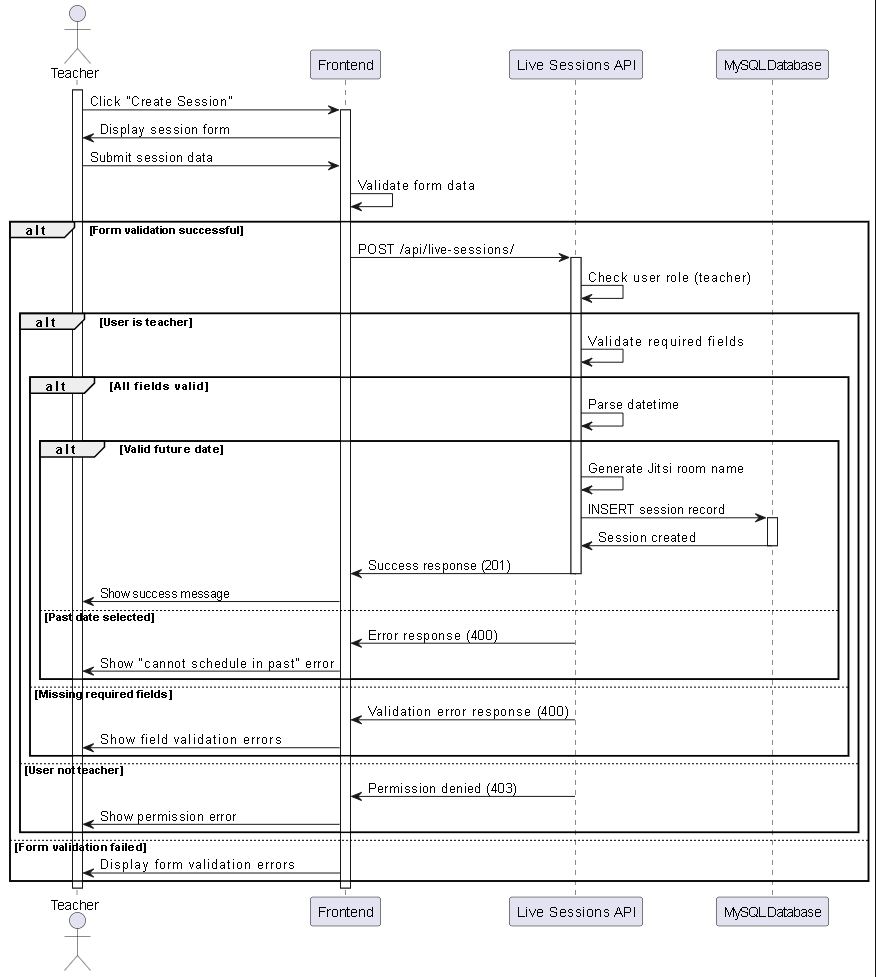


Figure 61 - Create Individual Live Sessions Sequence

#### UC-LS-02 – Edit Live Session Information

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-02 |
| **Use Case Name** | Edit Live Session Information |
| **Actor** | Teacher |
| **Pre-conditions** | Teacher owns session, session can be modified. |
| **Main Scenario** | 1. Teacher selects session to edit.  2. Teacher clicks edit button.  3. System opens edit form with current data.  4. Teacher modifies session details.  5. Teacher submits changes.  6. System validates and updates session.  7. System shows success message. |
| **Alternative Scenario** | - Session cannot be modified → System shows "cannot edit" message.  - No changes made → System shows "no updates" warning. |
| **Exceptional Scenario** | - Permission denied → System shows access error.  - Session not found → System redirects to session list. |
| **Post-condition** | Session details updated in database. |

Table 37 - Edit Live Session Information UCS

#### 

Figure 62 - Edit Live Session Information Activity



Figure 63 - Edit Live Session Information Sequence

#### UC-LS-03 – Cancel Live Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-03 |
| **Use Case Name** | Cancel Live Sessions |
| **Actor** | Teacher |
| **Pre-conditions** | Teacher owns session, session is PENDING or ASSIGNED. |
| **Main Scenario** | 1. Teacher selects session to cancel.  2. Teacher clicks cancel button.  3. System shows confirmation dialog.  4. Teacher confirms cancellation.  5. System updates session status to CANCELLED.  6. System shows success message. |
| **Alternative Scenario** | - Teacher cancels confirmation → No action taken. |
| **Exceptional Scenario** | - Session already started → System shows "cannot cancel active session" error.  - Session not found → System shows error message. |
| **Post-condition** | Session status changed to CANCELLED. |

Table 38 - Cancel Live Sessions UCS

#### 

Figure 64 - Cancel Live Sessions Activity

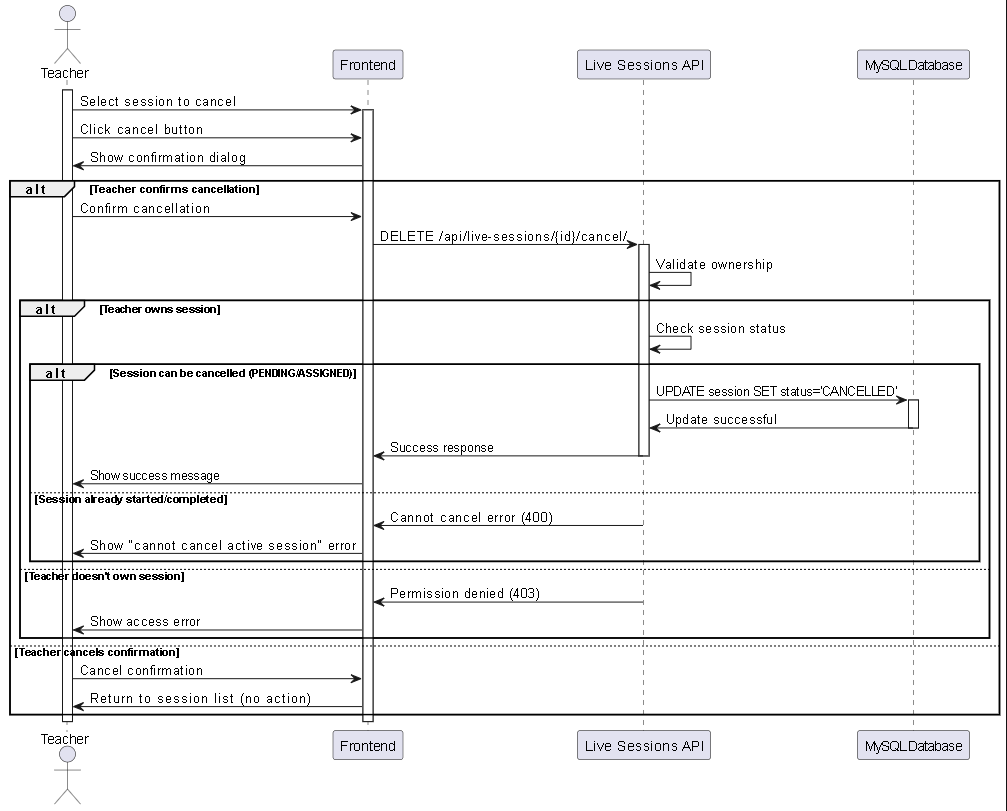


Figure 65 - Cancel Live Sessions Sequence

#### UC-LS-04 – Join Live Sessions (as Host)

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-04 |
| **Use Case Name** | Join Live Sessions (as Host) |
| **Actor** | Teacher |
| **Pre-conditions** | Teacher owns session, session not cancelled. |
| **Main Scenario** | 1. Teacher views session list.  2. Teacher clicks "Join Session" button.  3. System validates join permissions and timing.  4. System generates Jitsi meeting URL.  5. System opens meeting in popup window.  6. Teacher joins video conference. |
| **Alternative Scenario** | - Too early to join → System shows "available in X minutes" message.  - Session ended → System shows "session has ended" message. |
| **Exceptional Scenario** | - Popup blocked → System shows URL for manual access.  - Network error → System shows troubleshooting guide. |
| **Post-condition** | Teacher successfully joins live session. |

Table 39 - Join Live Sessions (as Host) UCS

#### 

Figure 66 - Join Live Sessions (as Host) Activity

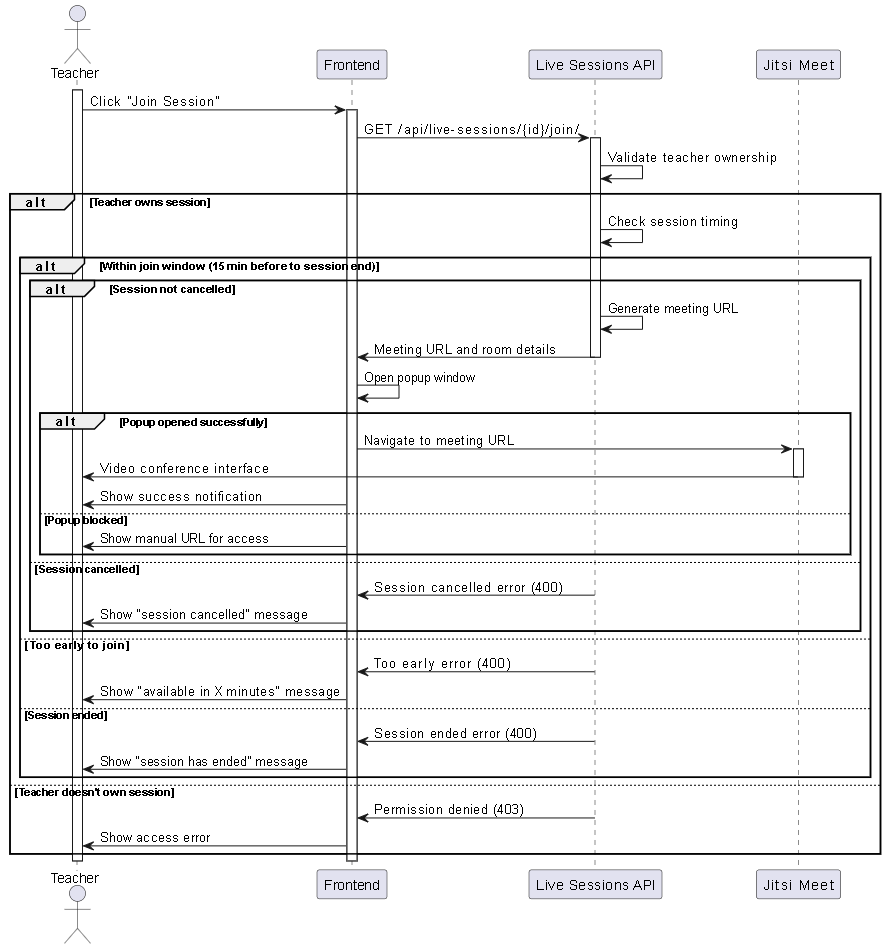


Figure 67 - Join Live Sessions (as Host) Sequence

#### UC-LS-05 – View Teacher's Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-05 |
| **Use Case Name** | View Teacher's Sessions |
| **Actor** | Teacher |
| **Pre-conditions** | User is authenticated as teacher. |
| **Main Scenario** | 1. Teacher navigates to sessions page.  2. System loads teacher's sessions.  3. System displays sessions with status filters.  4. Teacher can switch between tabs (active, completed, cancelled).  5. Teacher can change view mode (list/cards). |
| **Alternative Scenario** | - No sessions exist → System shows "create your first session" message. |
| **Exceptional Scenario** | - Loading error → System shows retry option. |
| **Post-condition** | Teacher sees their session overview. |

Table 40 - View Teacher's Sessions UCS

#### 

Figure 68 - View Teacher's Sessions Activity

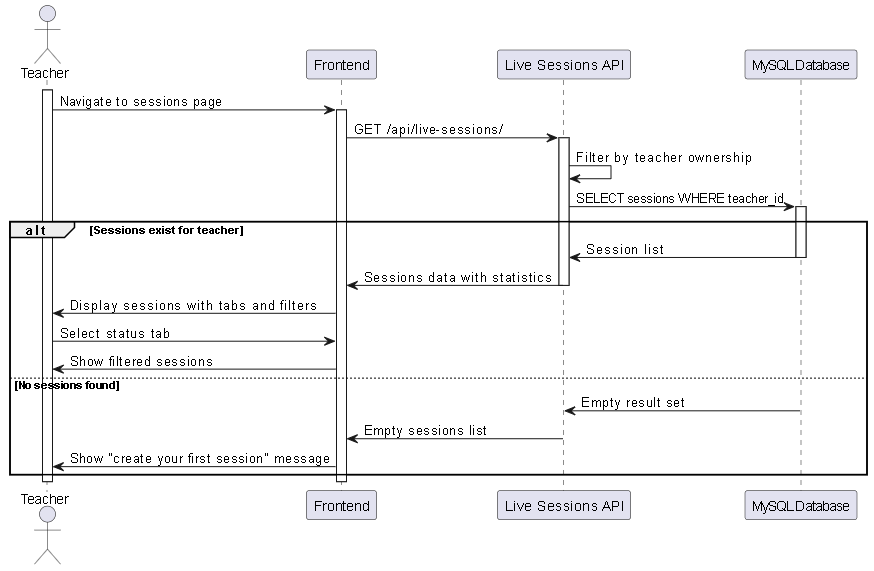


Figure 69 - View Teacher's Sessions Sequence

#### UC-LS-06 – View Assigned Students List

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-06 |
| **Use Case Name** | View Assigned Students List |
| **Actor** | Teacher |
| **Pre-conditions** | Teacher owns session, session has assignments. |
| **Main Scenario** | 1. Teacher selects session.  2. Teacher clicks "View Students" or assignment count.  3. System displays list of assigned students.  4. System shows student details and assignment info.  5. Teacher can see join times and attendance. |
| **Alternative Scenario** | - No students assigned → System shows "no assignments" message. |
| **Exceptional Scenario** | - Loading error → System shows retry option. |
| **Post-condition** | Teacher sees student assignment details. |

Table 41 - View Assigned Students List UCS

#### 

Table 42 - View Assigned Students List Activity



Figure 70 - View Assigned Students List Sequence

#### UC-LS-07 – View All Live Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-07 |
| **Use Case Name** | View All Live Sessions |
| **Actor** | Advisor |
| **Pre-conditions** | User is authenticated as advisor. |
| **Main Scenario** | 1. Advisor navigates to live sessions page.  2. System loads all sessions in system.  3. System displays sessions with teacher and assignment info.  4. Advisor can filter by status.  5. Advisor can switch between list and card view. |
| **Alternative Scenario** | - No sessions exist → System shows "no sessions available" message. |
| **Exceptional Scenario** | - Loading error → System shows retry option. |
| **Post-condition** | Advisor sees complete session overview. |

Table 43 - View All Live Sessions UCS

#### 

Figure 71 - View All Live Sessions Activity

#### 

Figure 72 - View All Live Sessions Sequence

#### UC-LS-08 – Assign Students to Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-08 |
| **Use Case Name** | Assign Students to Sessions |
| **Actor** | Advisor |
| **Pre-conditions** | Session exists, students available for assignment. |
| **Main Scenario** | 1. Advisor selects session to assign.  2. Advisor clicks "Assign Students" button.  3. System opens student selection modal.  4. Advisor selects students from list.  5. Advisor adds optional assignment message.  6. Advisor confirms assignment.  7. System creates assignments and updates session status.  8. System shows success message with count. |
| **Alternative Scenario** | - No students selected → System shows "select students" error.  - Student already assigned → System skips duplicate assignment. |
| **Exceptional Scenario** | - Student not found → System continues with other students.  - Assignment failed → System shows error details. |
| **Post-condition** | Students assigned to session; session status updated to ASSIGNED. |

Table 44 - Assign Students to Sessions UCS

#### 

Figure 73 - Assign Students to Sessions Activity

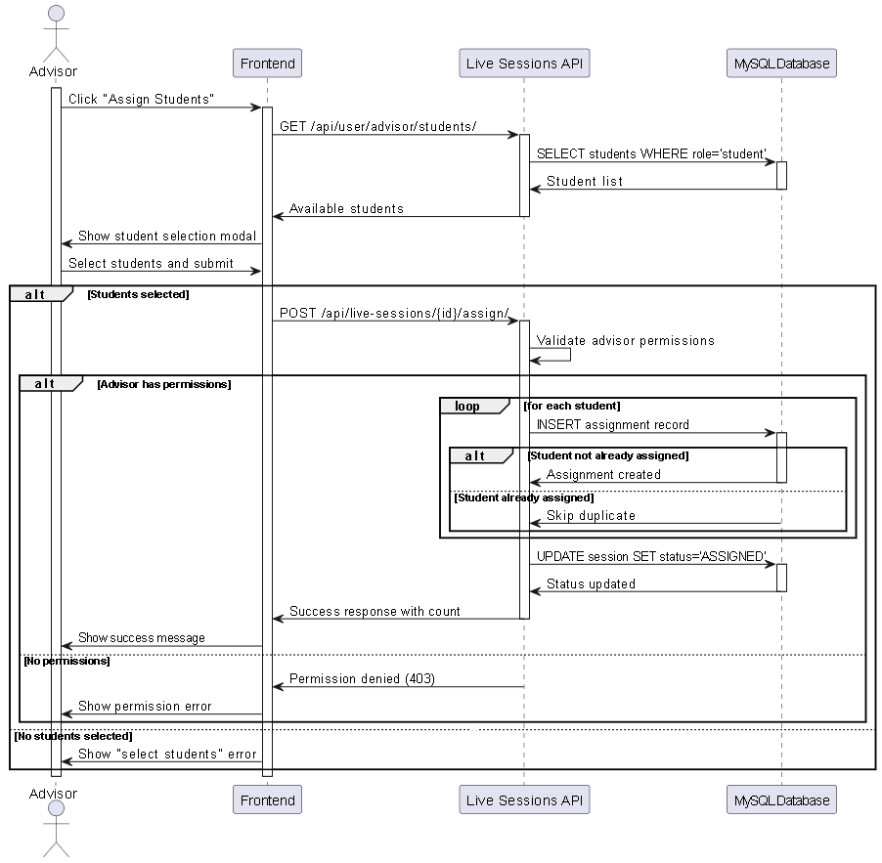


Figure 74 - Assign Students to Sessions Sequence

#### UC-LS-09 – Unassign Students from Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-09 |
| **Use Case Name** | Unassign Students from Sessions |
| **Actor** | Advisor |
| **Pre-conditions** | Session has assigned students. |
| **Main Scenario** | 1. Advisor selects session with assignments.  2. Advisor clicks "Manage Students" button.  3. System shows list of assigned students.  4. Advisor selects students to unassign.  5. Advisor confirms unassignment.  6. System removes assignments.  7. System shows confirmation message. |
| **Alternative Scenario** | - No students selected → System shows "select students" error. |
| **Exceptional Scenario** | - Unassignment failed → System shows error message. |
| **Post-condition** | Student assignments removed from session. |

Table 45 - Unassign Students from Sessions UCS

#### 

Figure 75 - Unassign Students from Sessions Activity

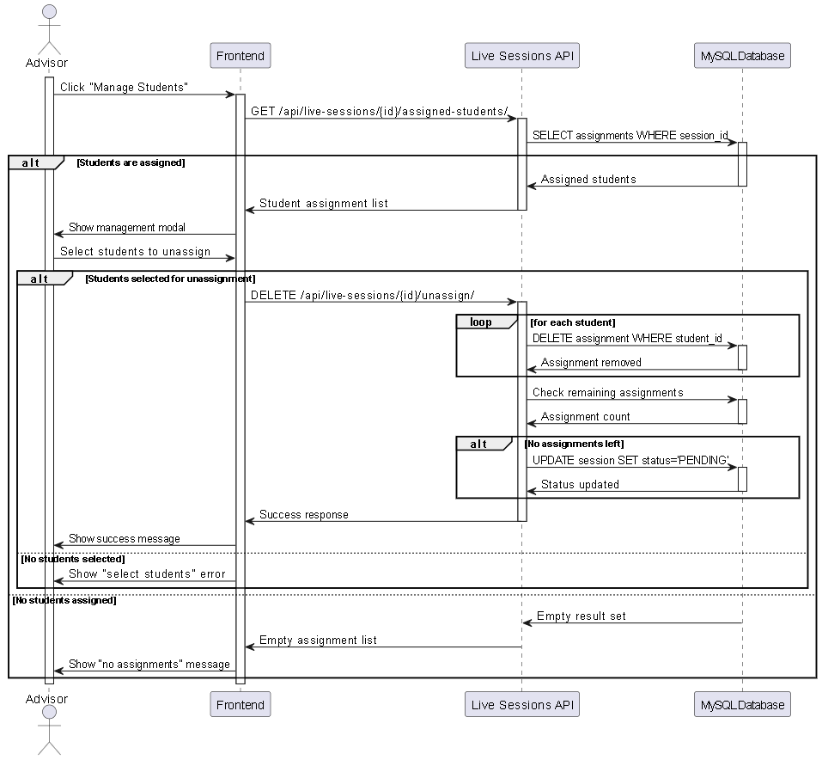


Figure 76 - Unassign Students from Sessions Sequence

#### UC-LS-10 – Modify Student Assignments

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-10 |
| **Use Case Name** | Modify Student Assignments |
| **Actor** | Advisor |
| **Pre-conditions** | Session exists with or without assignments. |
| **Main Scenario** | 1. Advisor opens assignment management.  2. System shows current assignments and available students.  3. Advisor adds new students or removes existing ones.  4. Advisor updates assignment messages if needed.  5. System processes all changes.  6. System shows summary of modifications. |
| **Alternative Scenario** | - No changes made → System shows "no modifications" message. |
| **Exceptional Scenario** | - Partial failure → System shows which operations succeeded/failed. |
| **Post-condition** | Assignment list updated according to advisor's changes. |

Table 46 - Modify Student Assignments UCS

#### 

Figure 77 - Modify Student Assignments Activity

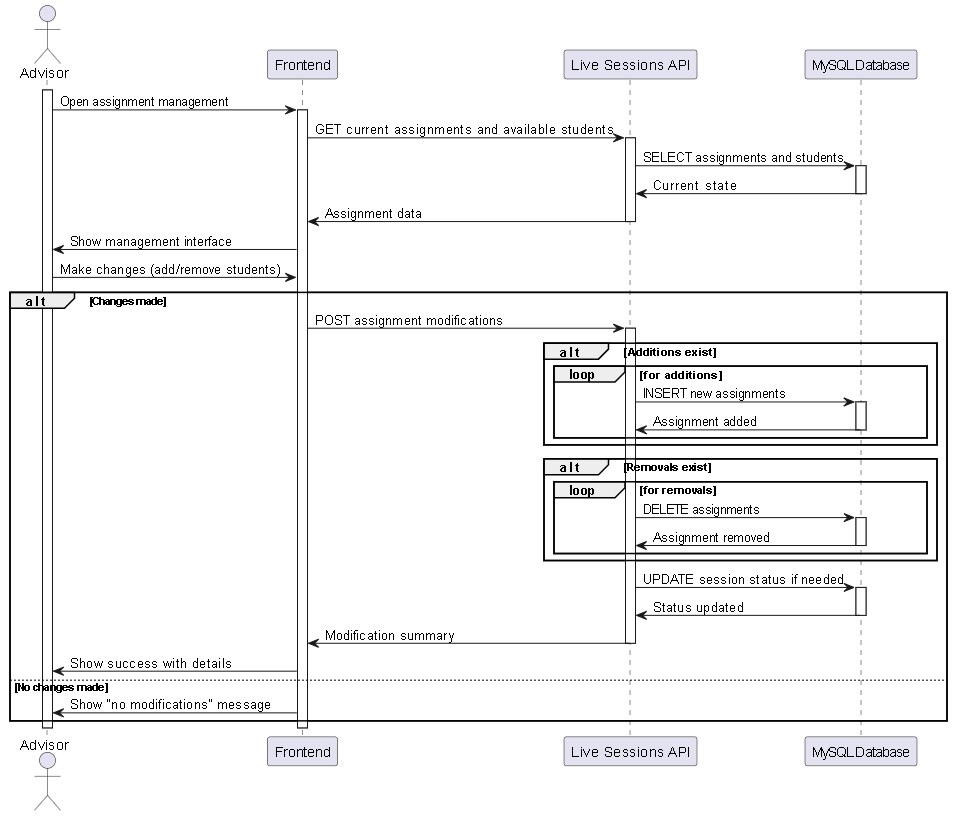


Figure 78 - Modify Student Assignments Sequence

#### UC-LS-11 – View Assigned Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-11 |
| **Use Case Name** | View Assigned Sessions |
| **Actor** | Student |
| **Pre-conditions** | Student is authenticated. |
| **Main Scenario** | 1. Student navigates to live sessions page.  2. System loads sessions assigned to student.  3. System displays sessions with teacher info and schedule.  4. System shows join status and timing for each session.  5. System auto-refreshes every 30 seconds. |
| **Alternative Scenario** | - No assigned sessions → System shows "no sessions assigned" message. |
| **Exceptional Scenario** | - Loading error → System shows retry option. |
| **Post-condition** | Student sees their assigned sessions. |

Table 47 - View Assigned Sessions UCS

#### 

Figure 79 - View Assigned Sessions Activity

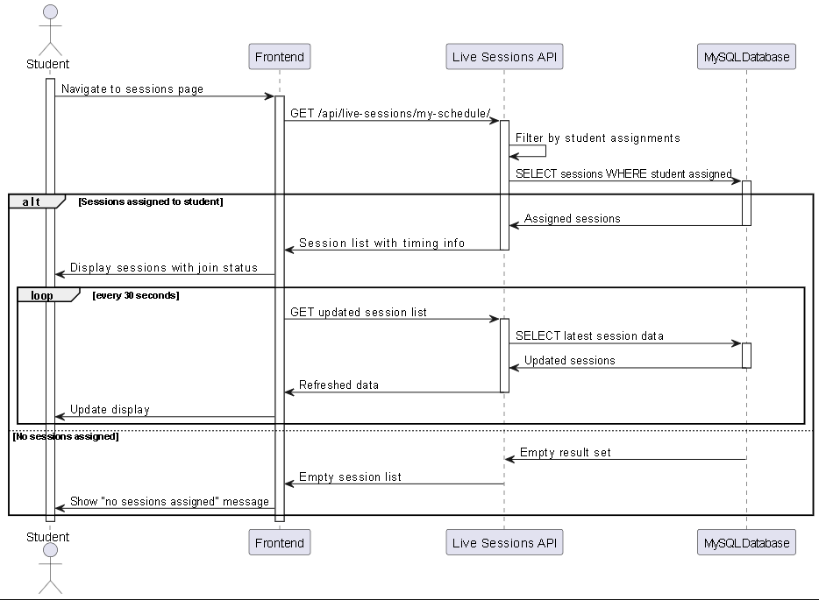


Figure 80 - View Assigned Sessions Sequence

#### UC-LS-12 – Join Live Sessions

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-12 |
| **Use Case Name** | Join Live Sessions |
| **Actor** | Student |
| **Pre-conditions** | Student is assigned to session, session is active. |
| **Main Scenario** | 1. Student views assigned sessions.  2. Student clicks "Join Now" on active session.  3. System validates assignment and timing.  4. System generates meeting URL and opens popup.  5. Student joins video conference.  6. System records join time. |
| **Alternative Scenario** | - Session not started → System shows "starts in X minutes" message.  - Late join period ended → System shows "late join closed" message. |
| **Exceptional Scenario** | - Not assigned → System shows "not assigned" error.  - Technical issues → System shows troubleshooting guide. |
| **Post-condition** | Student successfully joins live session. |

Table 48 - Join Live Sessions UCS

#### 

Figure 81 - Join Live Sessions Activity

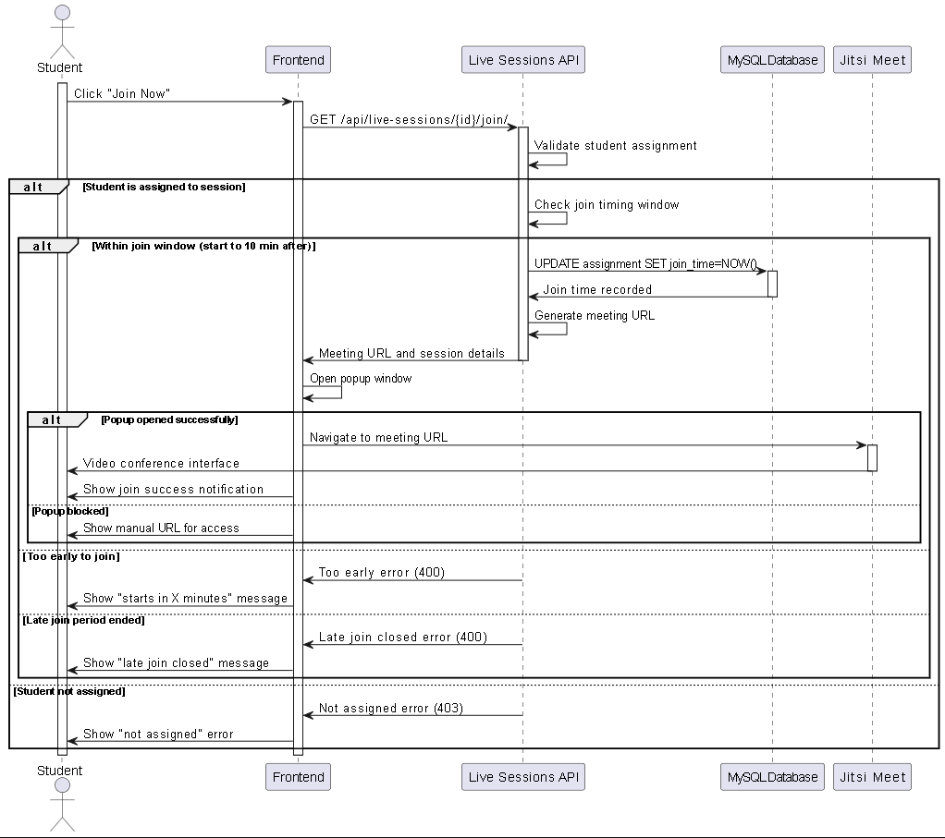


Figure 82 - Join Live Sessions sequence

#### UC-LS-13 – View Session Details

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-LS-13 |
| **Use Case Name** | View Session Details |
| **Actor** | Student |
| **Pre-conditions** | Student has access to session. |
| **Main Scenario** | 1. Student selects session from list.  2. Student clicks "Details" button.  3. System displays detailed session information.  4. System shows teacher info, schedule, description.  5. System displays join instructions and timing rules. |
| **Alternative Scenario** | - Session details unavailable → System shows basic info only. |
| **Exceptional Scenario** | - Session not found → System shows error message. |
| **Post-condition** | Student views complete session information. |

Table 49 - View Session Details UCS

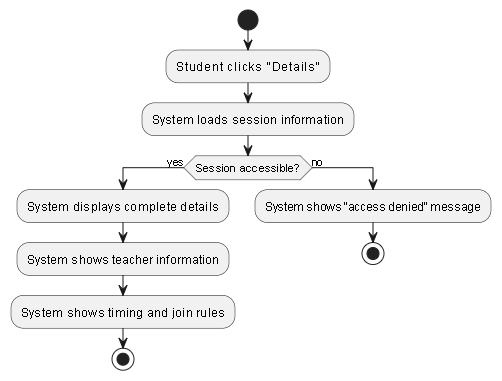


Figure 83 - View Session Details Activity

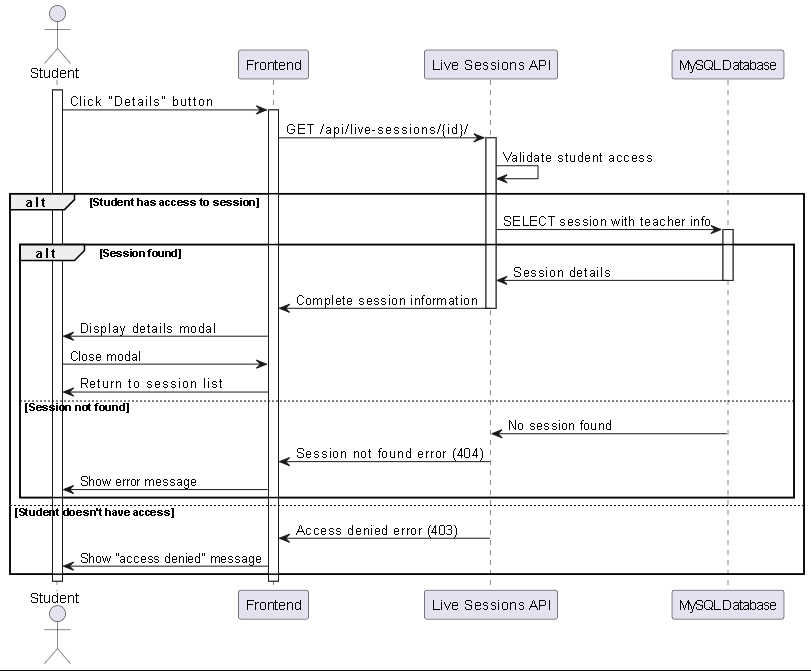


Figure 84 - View Session Details Sequence

# 4.11 Sprint 10: Admin System Analysis

In this section, we present the analytical study for Sprint 3, which focuses on the Administrative Control System. This module is crucial for maintaining the platform's security and integrity. The analysis details the capabilities of the "Superuser" (Admin) to manage the lifecycle of Advisors (Supervisors).

Key workflows modeled in this sprint include the secure creation of advisor accounts, password resets, and most importantly, the Account Activation/Deactivation protocol. Activity diagrams were used to define the security logic: specifically, how the system must immediately invalidate all active sessions when an advisor is deactivated, ensuring real-time access control.

### **Sprint 10 Requirements Covered**

The following 7 functional requirements will be implemented and validated during Sprint 10:

* **REQ-ADS-01** – The system shall allow Admins to securely log in to access the dedicated administrative dashboard.
* **REQ-ADS-02** – The system shall allow Admins to view a complete list of all registered advisors, displaying their current status (Active/Inactive).
* **REQ-ADS-03** – The system shall allow Admins to create new advisor accounts by inputting their personal and credential details.
* **REQ-ADS-04** – The system shall allow Admins to permanently delete advisor accounts that are no longer needed.
* **REQ-ADS-05** – The system shall allow Admins to reset an advisor's password in case of credential loss.
* **REQ-ADS-06** – The system shall allow Admins to Activate a deactivated advisor account to restore their access.
* **REQ-ADS-07** – The system shall allow Admins to Deactivate an advisor account, immediately preventing future logins.

### **4.10.1 Sprint 10 - Requirements Modeling**

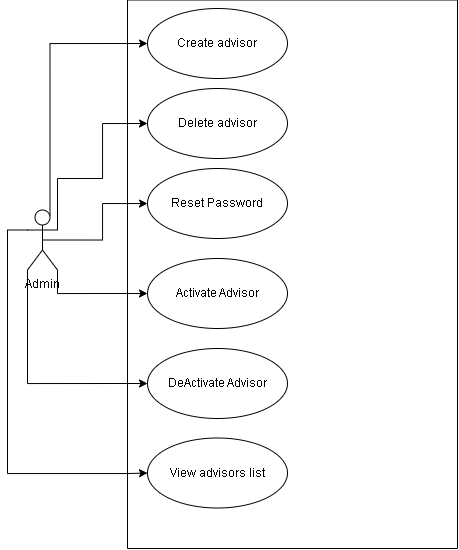


Figure 85 - Sprint 10 use case diagram

### **4.11.2 Use Case Specifications**

#### REQ-ADS-01 – Admin Authentication

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | REQ-ADS-01 |
| **Use Case Name** | Admin Authentication |
| **Actor** | Admin |
| **Pre-conditions** | User has superuser credentials. |
| **Main Scenario** | 1. Admin navigates to admin login page.  2. Admin enters username and password.  3. System validates credentials.  4. System checks superuser status.  5. System grants access to admin interface.  6. System displays admin dashboard. |
| **Alternative Scenario** | - Invalid credentials → Show "Invalid login" error.  - Non-superuser → Show "Access denied" error. |
| **Exceptional Scenario** | - Account deactivated → Show "Account deactivated" message. |
| **Post-condition** | Admin is authenticated and has access to admin features. |

Table 50 - Admin Authentication UCS

#### 

Figure 86 - Admin Authentication Activity

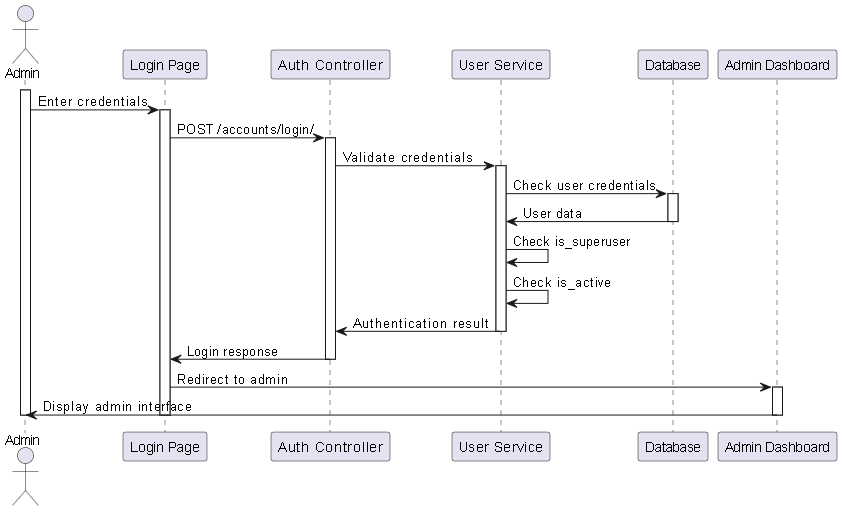


Figure 87 - Admin Authentication Sequence

#### REQ-ADS-02 – View Advisors List

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | REQ-ADS-02 |
| **Use Case Name** | View Advisors List |
| **Actor** | Admin |
| **Pre-conditions** | Admin is authenticated. |
| **Main Scenario** | 1. Admin navigates to advisors’ management page.  2. System retrieves all advisor accounts.  3. System displays advisors list with statistics.  4. Admin can see total, active, and inactive advisors.  5. Admin can view individual advisor details. |
| **Alternative Scenario** | - No advisors exist → Show "No advisors found" message. |
| **Exceptional Scenario** | - Database error → Show "Unable to load advisors" message. |
| **Post-condition** | Admin sees comprehensive advisors overview. |

Table 51 - View Advisors List UCS

#### 

Figure 88 - View Advisors List Activity

#### REQ-ADS-03 – Create New Advisor

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | REQ-ADS-03 |
| **Use Case Name** | Create New Advisor |
| **Actor** | Admin |
| **Pre-conditions** | Admin is authenticated. |
| **Main Scenario** | 1. Admin clicks "Add New Advisor" button.  2. System displays advisor creation form.  3. Admin fills advisor details (name, email, password).  4. System validates input data.  5. System checks email uniqueness.  6. System creates advisor account with advisor role.  7. System displays success message. |
| **Alternative Scenario** | - Email already exists → Show "Email already in use" error.  - Invalid data → Show validation errors. |
| **Exceptional Scenario** | - Creation fails → Show "Failed to create advisor" message. |
| **Post-condition** | New advisor account is created and active. |

Table 52 - Create New Advisor UCS

#### 

Figure 89 - Create New Advisor Activity

#### 

Figure 90 - Create New Advisor Sequence

#### REQ-ADS-04 – Delete Advisor

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | REQ-ADS-04 |
| **Use Case Name** | Delete Advisor |
| **Actor** | Admin |
| **Pre-conditions** | Admin is authenticated, advisor exists. |
| **Main Scenario** | 1. Admin selects advisor to delete.  2. Admin clicks "Delete" button.  3. System displays deletion confirmation dialog.  4. Admin confirms deletion.  5. System permanently removes advisor account.  6. System displays success message. |
| **Alternative Scenario** | - Admin cancels deletion → Returns to advisors list. |
| **Exceptional Scenario** | - Deletion fails → Show "Failed to delete advisor" message. |
| **Post-condition** | Advisor account is permanently removed. |

Table 53 - Delete Advisor UCS

#### 

Figure 91 - Delete Advisor Activity

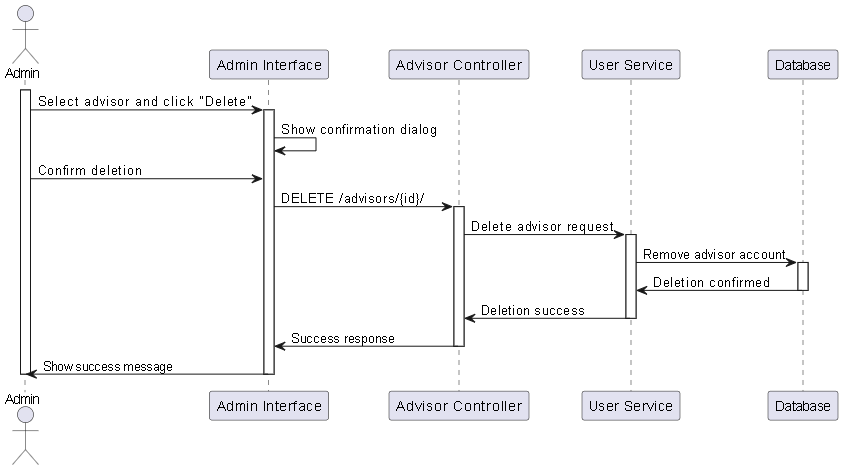


Figure 92 - Delete Advisor Sequence

#### REQ-ADS-05 – Reset Advisor Password

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | REQ-ADS-05 |
| **Use Case Name** | Reset Advisor Password |
| **Actor** | Admin |
| **Pre-conditions** | Admin is authenticated, advisor exists. |
| **Main Scenario** | 1. Admin selects advisor for password reset.  2. Admin clicks "Reset Password" button.  3. System displays password reset form.  4. Admin enters new password and confirmation.  5. System validates password strength.  6. System updates advisor password.  7. System displays success message. |
| **Alternative Scenario** | - Passwords don't match → Show "Passwords don't match" error.  - Weak password → Show password requirements.  - Admin can use "Generate Password" feature. |
| **Exceptional Scenario** | - Reset fails → Show "Failed to reset password" message. |
| **Post-condition** | Advisor password is updated. |

Table 54 - Reset Advisor Password

#### 

Figure 93 - Reset Advisor Password Activity

#### 

Figure 94 - Reset Advisor Password Sequence

#### REQ-ADS-06 – Activate Advisor

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | REQ-ADS-06 |
| **Use Case Name** | Activate Advisor |
| **Actor** | Admin |
| **Pre-conditions** | Admin is authenticated, advisor exists, advisor is currently inactive. |
| **Main Scenario** | 1. Admin selects inactive advisor from list.  2. Admin clicks "Activate" button.  3. System updates advisor is\_active status to True.  4. System enables login access for advisor.  5. System displays "Advisor activated successfully" confirmation.  6. System updates advisor list with green active indicator. |
| **Alternative Scenario** | - Advisor already active → Show "Advisor is already active" message. |
| **Exceptional Scenario** | - Activation fails → Show "Failed to activate advisor" message. |
| **Post-condition** | Advisor account is active and can log in. |

Table 55 - Activate Advisor UCS

#### 

Figure 95 - Activate Advisor Activity

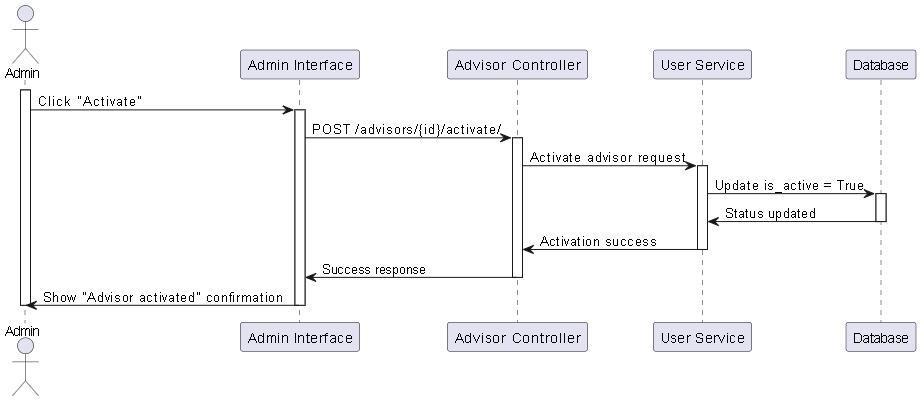


Figure 96 - Activate Advisor Sequence

#### REQ-ADS-07 – Deactivate Advisor

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | REQ-ADS-07 |
| **Use Case Name** | Deactivate Advisor |
| **Actor** | Admin |
| **Pre-conditions** | Admin is authenticated, advisor exists, advisor is currently active. |
| **Main Scenario** | 1. Admin selects active advisor from list.  2. Admin clicks "Deactivate" button.  3. System displays deactivation confirmation dialog.  4. Admin confirms deactivation.  5. System updates advisor is\_active status to False.  6. System prevents future login attempts for advisor.  7. System invalidates existing advisor sessions.  8. System displays "Advisor deactivated successfully" confirmation.  9. System updates advisor list with red inactive indicator. |
| **Alternative Scenario** | - Advisor already inactive → Show "Advisor is already inactive" message.  - Admin cancels deactivation → Return to advisor list. |
| **Exceptional Scenario** | - Deactivation fails → Show "Failed to deactivate advisor" message. |
| **Post-condition** | Advisor account is inactive and cannot log in. |

Table 56 - Deactivate Advisor UCS

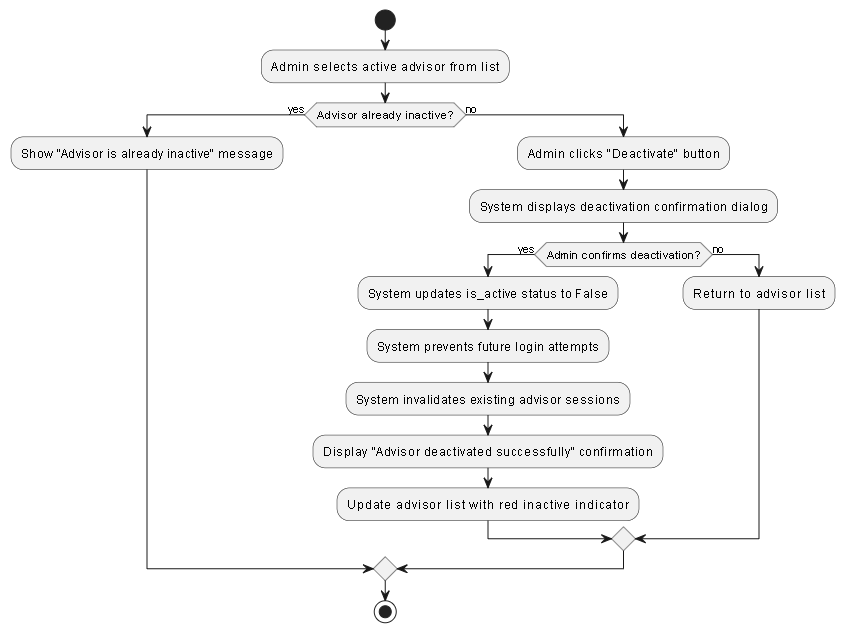


Figure 97 - Deactivate Advisor Activity

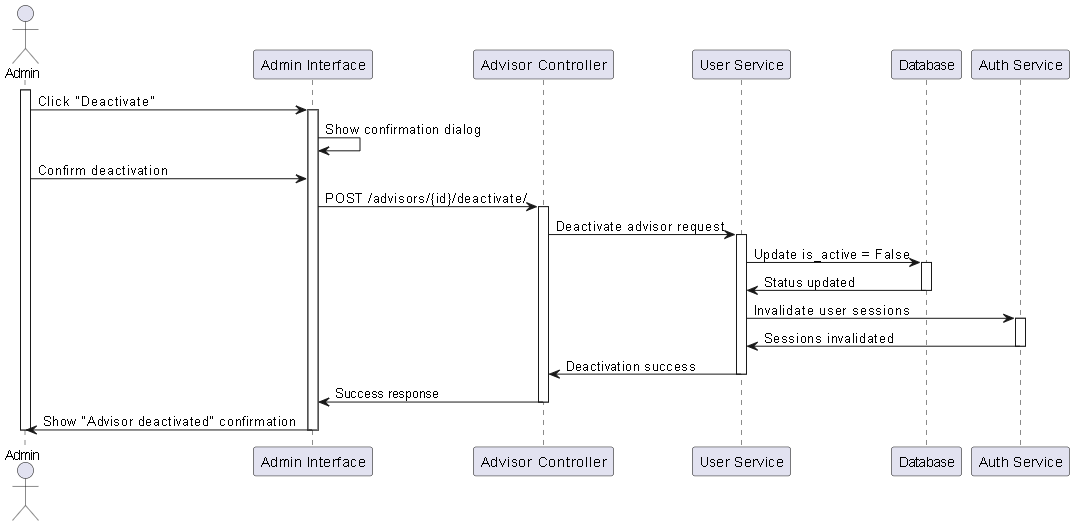


Figure 98 - Deactivate Advisor Sequence

# 4.12 Sprint 11: Recurring Sessions Analysis

**Analysis** This section outlines the analysis for **Sprint 4**, which introduces the **Recurring Sessions Automation Engine**. The goal of this sprint is to reduce the manual workload on teachers by automating the scheduling process. The analysis introduces a new architectural concept: separating the **Session Template** (the rule) from the **Generated Session** (the instance). The domain model was updated to include Session Template and Student Group entities. The logic defines how a teacher can create a "Monday Math Class" template once, and the system automatically generates the specific sessions for the coming months. This ensures consistent educational delivery without repetitive administrative tasks.

**Sprint 4 Requirements Covered** The following **17 functional requirements** will be implemented and validated during Sprint 11:

* **REQ-RS-01** – The system shall allow **Teachers** to create **Session Templates**, defining recurrence patterns (e.g., Weekly on Mondays).
* **REQ-RS-02** – The system shall allow **Teachers** to view and manage their list of active templates.
* **REQ-RS-03** – The system shall allow **Teachers** to edit template details, with options to apply changes to future sessions only.
* **REQ-RS-04** – The system shall allow **Teachers** to delete templates, stopping future generation.
* **REQ-RS-05** – The system shall allow **Teachers** to **Pause** a template (e.g., during holidays) without deleting it.
* **REQ-RS-06** – The system shall allow **Teachers** to **Resume** a paused template.
* **REQ-RS-07** – The system shall allow **Teachers** to permanently **End** a template's lifecycle.
* **REQ-RS-08** – The system shall automatically generate individual sessions based on active templates via a background service.
* **REQ-RS-09** – The system shall provide statistics on how many sessions were generated from each template.
* **REQ-RS-10** – The system shall allow **Advisors** to view all session templates in the system.
* **REQ-RS-11** – The system shall allow **Advisors** to assign templates to **Student Groups** rather than just individuals.
* **REQ-RS-12** – The system shall allow **Advisors** to create and name new Student Groups.
* **REQ-RS-13** – The system shall allow **Advisors** to manage group memberships (add/remove students).
* **REQ-RS-14** – The system shall allow **Advisors** to view available students for group assignment.
* **REQ-RS-15** – The system shall allow **Students** to view their recurring schedule generated from templates.
* **REQ-RS-16** – The system shall allow **Students** to track upcoming recurring sessions.
* **REQ-RS-17** – The system shall allow **Students** to view the details of the parent template for their assigned sessions.

### **4.12.1 Sprint 11 - Requirements Modeling**

Figure 99 - Sprint 11 use case diagram

# 4.12.2 Use Case Specifications

#### UC-RS-01 – Create Session Template

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-01 |
| **Use Case Name** | Create Session Template |
| **Actor** | Teacher |
| **Pre-conditions** | Teacher is authenticated; Teacher is on Templates Page. |
| **Main Scenario** | 1. Teacher clicks "Create Template" button.  2. System displays Create Template.  3. Teacher inputs title, subject, schedule pattern (e.g., Weekly), time, and dates.  4. Teacher submits form.  5. System validates inputs.  6. System creates template with ACTIVE status.  7. System triggers session generator service.  8. System displays success message. |
| **Alternative Scenario** | - Invalid Data: System shows validation errors (e.g., End Date before Start Date). |
| **Exceptional Scenario** | - Server Error: System shows "Failed to create template". |
| **Post-condition** | Template created and stored in database. |

Table 57 - Create Session Template UCS

#### 

Figure 100 - Create Session Template Activity

#### 

Figure 101 - Create Session Template Sequence

#### UC-RS-02 – View Session Templates

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-02 |
| **Use Case Name** | View Session Templates |
| **Actor** | Teacher |
| **Pre-conditions** | Teacher is authenticated. |
| **Main Scenario** | 1. Teacher navigates to Templates Page.  2. System requests templates list.  3. System retrieves templates owned by the teacher.  4. System displays TemplateCard for each template.  5. Teacher sees status, schedule summary, and actions. |
| **Alternative Scenario** | - No Templates: System displays "No templates found" empty state. |
| **Exceptional Scenario** | - Network Error: System shows retry button. |
| **Post-condition** | Teacher views list of templates. |

Table 58 - View Session Templates UCS

#### 

Figure 102 - View Session Templates Activity

#### 

Figure 103 - View Session Templates Sequence

#### UC-RS-03 – Edit Session Template

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-03 |
| **Use Case Name** | Edit Session Template |
| **Actor** | Teacher |
| **Pre-conditions** | Template exists and belongs to Teacher. |
| **Main Scenario** | 1. Teacher clicks "Edit" on Template Card.  2. System opens Edit Template Modal with pre-filled data.  3. Teacher modifies fields (e.g., changes Time).  4. Teacher submits changes.  5. System asks: "Apply to future sessions?".  6. Teacher confirms.  7. System updates template and regenerates future sessions. |
| **Alternative Scenario** | - Cancel: Teacher cancels edit → No changes saved. |
| **Exceptional Scenario** | - Concurrent Edit: System detects conflict → Shows warning. |
| **Post-condition** | Template updated; Future sessions reflect changes. |

Table 59 - Edit Session Template UCS

#### 

Table 60 - Edit Session Template Activity

#### 

Figure 104 - Edit Session Template Sequence

#### UC-RS-04 – Delete Session Template

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-04 |
| **Use Case Name** | Delete Session Template |
| **Actor** | Teacher |
| **Pre-conditions** | Template exists. |
| **Main Scenario** | 1. Teacher clicks "Delete" icon.  2. System shows Confirmation Dialog.  3. Teacher confirms deletion.  4. System deletes the Template and associated future sessions.  5. System removes template from list. |
| **Alternative Scenario** | - Deny: Teacher clicks Cancel → No action. |
| **Exceptional Scenario** | - Constraint Error: Cannot delete if sessions have attendance → System shows error. |
| **Post-condition** | Template removed from database. |

Table 61 - Delete Session Template UCS

#### 

Figure 105 - Delete Session Template Activity



Figure 106 - Delete Session Template Sequence

#### UC-RS-05 – Pause Template

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-05 |
| **Use Case Name** | Pause Template |
| **Actor** | Teacher |
| **Pre-conditions** | Template exists; Template status is ACTIVE; Teacher owns the template. |
| **Main Scenario** | 1. Teacher clicks "Pause" button on an active template.  2. System displays Confirm Pause TemplateModal.  3. Teacher confirms the action.  4. System updates template status to PAUSED in database.  5. System stops automatic generation for this template.  6. System displays success message and updates list UI. |
| **Alternative Scenario** | - Teacher Cancels: Teacher clicks "Cancel" in the confirmation modal → System closes modal; No changes are made. |
| **Exceptional Scenario** | - Network Error: Request fails due to connection issue → System shows "Network Error, please try again".  - Database Error: Status update fails → System logs error and informs teacher. |
| **Post-condition** | Template status is PAUSED; No new sessions will be generated until resumed. |

Table 62 - Pause Template UCS

#### UC-RS-06 – Resume Template

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-06 |
| **Use Case Name** | Resume Template |
| **Actor** | Teacher |
| **Pre-conditions** | Template exists; Template status is PAUSED; Teacher owns the template. |
| **Main Scenario** | 1. Teacher clicks "Resume" button on a paused template.  2. System sends activation request.  3. System updates template status to ACTIVE in database.  4. System triggers logic to calculate next generation date.  5. System displays success message.  6. System updates template card to show "Active" status. |
| **Alternative Scenario** | - Schedule Conflict: If resuming creates an immediate session conflict (rare) → System warns teacher but proceeds. |
| **Exceptional Scenario** | - Server Error: fails to update status → System shows error message.  - Template Expired: If end date has passed while paused → System changes status to ENDED instead of ACTIVE. |
| **Post-condition** | Template status is ACTIVE; Session generation resumes based on schedule. |

Table 63 - Resume Template UCS

#### UC-RS-07 – End Template Permanently

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-07 |
| **Use Case Name** | End Template Permanently |
| **Actor** | Teacher |
| **Pre-conditions** | Template is ACTIVE or PAUSED; Teacher owns the template. |
| **Main Scenario** | 1. Teacher clicks "End Template" (or Delete icon).  2. System displays Confirm EndTemplateModal with warning "This action cannot be undone".  3. Teacher confirms the action.  4. System updates status to ENDED in database.  5. System cancels any future GeneratedSession that hasn't started yet.  6. Template becomes read-only in the UI. |
| **Alternative Scenario** | - Teacher Cancels: Teacher clicks "Cancel" in confirmation modal → Action aborted.  - Has Active Sessions: If sessions are currently running → System allows ending but keeps current session active. |
| **Exceptional Scenario** | - Data Integrity Error: Failure to cleanup future sessions → System shows error and rolls back transaction. |
| **Post-condition** | Template status is ENDED; Template cannot be reactivated; Future generation is permanently stopped. |

Table 64 - Resume Template UCS

#### 

Figure 107 - Manage Templates Status Activity

#### 

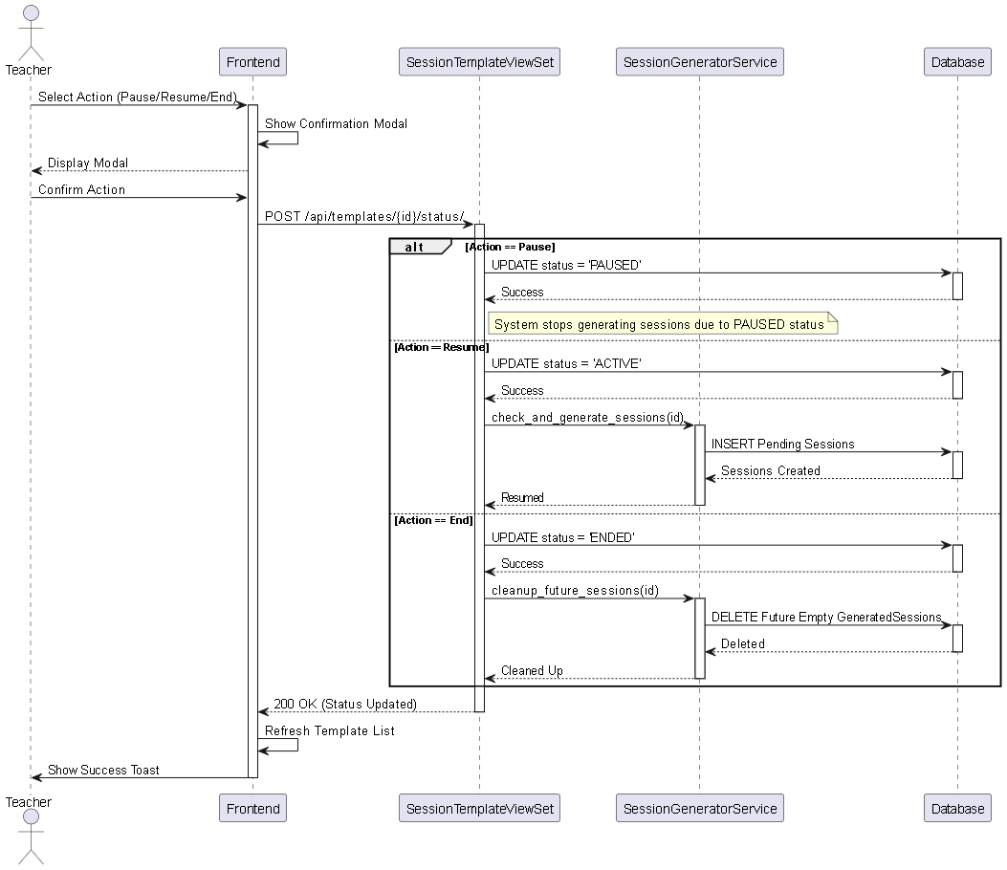


Figure 108 - Manage Templates Status Sequence

#### UC-RS-09 – View All Templates (Advisor)

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-09 |
| **Use Case Name** | View All Templates (Advisor) |
| **Actor** | Advisor |
| **Pre-conditions** | Advisor authenticated. |
| **Main Scenario** | 1. Advisor navigates to Templates Page.  2. System checks role = Advisor.  3. System fetches ALL templates.  4. System displays templates with Teacher Name info. |
| **Post-condition** | Advisor views all system templates. |

Table 65 - View All Templates (Advisor) UCS

#### 

Figure 109 - View All Templates (Advisor) Activity

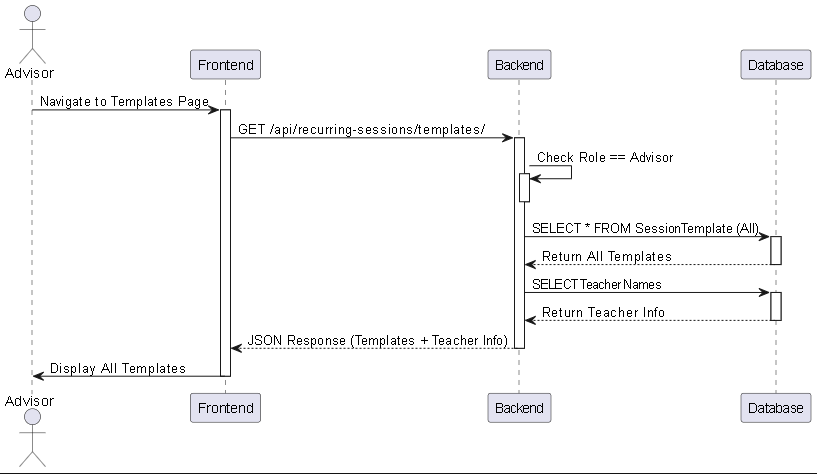


Figure 110 - View All Templates (Advisor) Sequence

#### UC-RS-10 – Assign Template

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-10 |
| **Use Case Name** | Assign Template |
| **Actor** | Advisor |
| **Pre-conditions** | Template exists. |
| **Main Scenario** | 1. Advisor clicks "Assign" on template.  2. System opens AssignTemplateModal.  3. Advisor selects "Groups" or "Students" tab.  4. Advisor selects targets (e.g., "Group A").  5. Advisor clicks Submit.  6. System creates TemplateGroupAssignment. |
| **Alternative Scenario** | - Already Assigned: System shows error "Target already assigned". |

Table 66 - Assign Template UCS

#### 

Figure 111 - Assign Template Activity

#### 

Figure 112 - Assign Template Sequence

#### UC-RS-11 – Create Student Group

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-11 |
| **Use Case Name** | Create Student Group |
| **Actor** | Advisor |
| **Pre-conditions** | Advisor authenticated. |
| **Main Scenario** | 1. Advisor navigates to StudentGroupsPage.  2. Click "Create Group".  3. Open CreateGroupModal.  4. Enter Group Name.  5. Select Students from list.  6. Submit.  7. System saves group. |
| **Post-condition** | New Student Group exists. |

Table 67 - Create Student Group UCS

#### 

Figure 113 - Create Student Group Activity

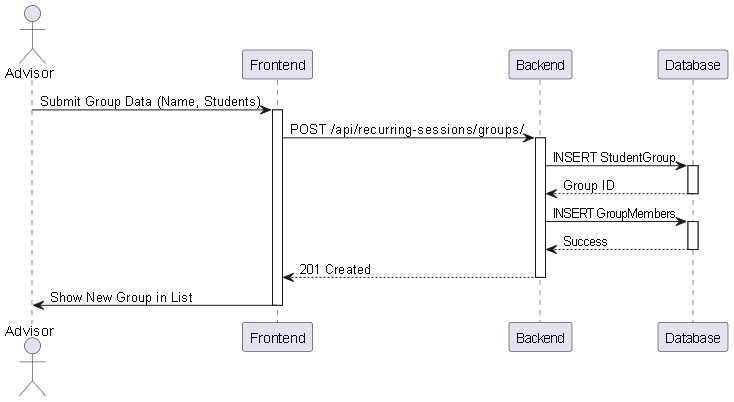


Figure 114 - Create Student Group Sequence

#### UC-RS-12 – Manage Student Group

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-12 |
| **Use Case Name** | Manage Student Group |
| **Actor** | Advisor |
| **Pre-conditions** | Group exists. |
| **Main Scenario** | 1. Advisor clicks "Edit" on GroupCard.  2. System opens EditGroupModal.  3. Advisor adds/removes students.  4. Submit changes.  5. System updates StudentGroup members. |
| **Post-condition** | Group membership updated. |

Table 68 - Manage Student Group UCS

#### 

Figure 115 - Manage Student Group Activity

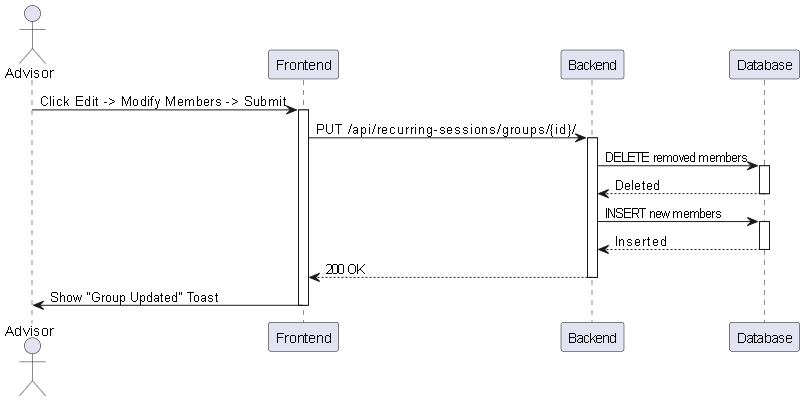


Figure 116 - Manage Student Group Sequence

#### UC-RS-14 – View Assigned Templates

|  |  |
| --- | --- |
| Field | Description |
| **Use Case ID** | UC-RS-14 |
| **Use Case Name** | View Assigned Templates |
| **Actor** | Student |
| **Pre-conditions** | Student is assigned to a template. |
| **Main Scenario** | 1. Student opens MyRecurringSessionsPage.  2. System returns templates.  3. Student views list. |
| **Alternative Scenario** | - No Assignments: Show empty state. |
| **Post-condition** | Student aware of schedule. |

Table 69 - View Assigned Templates UCS

#### 

Figure 117 - View Assigned Templates Activity

#### 

Figure 118 - View Assigned Templates Sequence

# 4.13 Initial Test Cases

### Table 70 - Initial Test Cases

### 1. Authentication & User Management

#### TC-AUTH-001 – Successful Registration

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AUTH-001 |
| **Associated Use Case** | User Registration (REQ-2.1) |
| **Title** | Successful Registration for a New User |
| **Objective** | To verify that a new user with valid and unique data can successfully create an account. |
| **Prerequisites** | 1. User is guest/not logged in.  2. Email does not exist in DB.  3. Access to registration page. |
| **Test Data** | Name: Test Teacher, Email: new.teacher.2025@example.com, Role: Teacher |
| **Steps** | 1. Navigate to registration page.  2. Enter First/Last Name.  3. Enter Email.  4. Enter Password & Confirm.  5. Select "Teacher" role.  6. Check Terms of Service.  7. Click "Register". |
| **Pass Criteria** | User account created in DB; User redirected to login or verification page. |
| **Fail Criteria** | Error if email exists; Validation error for weak/mismatching password. |

#### TC-AUTH-002 – Successful Login

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AUTH-002 |
| **Associated Use Case** | Login (REQ-2.2) |
| **Title** | Successful Login for an Existing User |
| **Objective** | To verify that a registered user with valid credentials can successfully log in. |
| **Prerequisites** | User account exists and is active. |
| **Test Data** | Valid Email, Valid Password |
| **Steps** | 1. Navigate to Login page.  2. Enter Email and Password.  3. Click "Login". |
| **Pass Criteria** | User authenticated and redirected to role-specific dashboard. |
| **Fail Criteria** | "Invalid credentials" or "Account not activated" error shown. |

#### TC-USER-001 – Edit Profile

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-USER-001 |
| **Associated Use Case** | User Profile (REQ-2.3, REQ-2.4) |
| **Title** | View and Successfully Edit User Profile Information |
| **Objective** | To verify that an authenticated user can view and update their profile. |
| **Prerequisites** | User is logged in. |
| **Test Data** | Change Country to "SYRIA" |
| **Steps** | 1. Navigate to "Profile" page.  2. Click "Edit Profile".  3. Change "Country" field.  4. Click "Save". |
| **Pass Criteria** | Profile updated; New value ("SYRIA") displayed. |
| **Fail Criteria** | System fails to save or reverts to old info. |

#### TC-AUTH-003 – Logout

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AUTH-003 |
| **Associated Use Case** | Logout (REQ-2.5) |
| **Title** | Successful Logout |
| **Objective** | Verify logged-in user can terminate session. |
| **Steps** | 1. Click "Logout" button. |
| **Pass Criteria** | Session terminated; Redirected to login page. |
| **Fail Criteria** | User remains logged in. |

#### TC-AUTH-004 – Password Reset

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AUTH-004 |
| **Associated Use Case** | Reset Password (REQ-2.6) |
| **Title** | Successful Password Reset Request |
| **Objective** | Verify user can initiate password reset. |
| **Test Data** | Valid registered email |
| **Steps** | 1. Click "Forgot password?".  2. Enter email.  3. Click "Send Reset Link". |
| **Pass Criteria** | Success message shown; Email sent. |
| **Fail Criteria** | Error displayed or email not sent. |

### 2. Content Management (Teacher)

#### TC-CNT-001 – Upload Lesson

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-CNT-001 |
| **Associated Use Case** | Upload Video/PDF (REQ-4.1, REQ-4.2) |
| **Title** | Teacher Successfully Uploads a New Lesson |
| **Objective** | Verify Teacher can create lesson and upload content. |
| **Test Data** | Title: Intro to Arabic Grammar, Content: .mp4/.pdf file |
| **Steps** | 1. Go to "Lesson Management".  2. Select "Add New Lesson".  3. Enter title and type.  4. Upload file and click "Create". |
| **Pass Criteria** | Lesson created; Appears in "Recent Lessons". |
| **Fail Criteria** | Upload fails; Lesson not created. |

#### TC-CNT-002 – Create MCQ

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-CNT-002 |
| **Associated Use Case** | Upload Exercises (REQ-4.3) |
| **Title** | Teacher Creates a New MCQ Exercise |
| **Objective** | Verify teacher can add MCQ quiz to lesson. |
| **Test Data** | Q: Capital of Syria? A: Damascus. |
| **Steps** | 1. Navigate to lesson "Quizzes" tab.  2. Click "Create New Quiz".  3. Add Question/Answers.  4. Click "Save". |
| **Pass Criteria** | Quiz created and linked to lesson. |
| **Fail Criteria** | System fails to save. |

#### TC-CNT-003 – Manage Lesson

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-CNT-003 |
| **Associated Use Case** | Manage Lesson (REQ-4.4, 4.5, 4.6) |
| **Title** | Teacher Manages an Existing Lesson |
| **Objective** | Verify view, update, and delete functionality. |
| **Steps** | 1. (View) Confirm list visibility.  2. (Update) Edit title and save.  3. (Delete) Delete lesson and confirm. |
| **Pass Criteria** | List visible; Title updated; Lesson deleted. |
| **Fail Criteria** | Operations fail. |

### 3. Student & Advisor Interaction

#### TC-STU-001 – Browse Assigned Lessons

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-STU-001 |
| **Associated Use Case** | Browse/Select Lessons (REQ-4.7, 4.8) |
| **Title** | Student Views and Opens an Assigned Lesson |
| **Objective** | Verify student sees only assigned lessons and can open them. |
| **Prerequisites** | Student logged in; Lesson assigned. |
| **Steps** | 1. Locate "My Assigned Lessons".  2. Click on a lesson. |
| **Pass Criteria** | Only assigned lessons displayed; Content opens on click. |
| **Fail Criteria** | Unassigned lessons visible; Cannot open. |

#### TC-ADV-001 – Assign Lesson

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-ADV-001 |
| **Associated Use Case** | Assign Lessons (REQ-4.12) |
| **Title** | Advisor Assigns a Lesson to a Specific Student |
| **Objective** | Verify advisor can assign content. |
| **Steps** | 1. View student list.  2. Click "Assign Lessons".  3. Select lesson and confirm. |
| **Pass Criteria** | Lesson linked; Appears on student dashboard. |
| **Fail Criteria** | Assignment fails. |

#### TC-STU-002 – Interact with Content

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-STU-002 |
| **Associated Use Case** | Download/Solve (REQ-4.9, 4.10) |
| **Title** | Student Interacts with Lesson Content |
| **Objective** | Verify download and quiz submission. |
| **Steps** | 1. Click "Download" on PDF.  2. Navigate to quiz.  3. Submit answers. |
| **Pass Criteria** | PDF downloads; Quiz submitted with score shown. |
| **Fail Criteria** | Broken link; Submission fails. |

#### TC-ADV-002 – Filter Lessons

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-ADV-002 |
| **Associated Use Case** | Search Lessons (REQ-4.13) |
| **Title** | Advisor Filters Lessons for Easier Management |
| **Test Data** | Filter: Grade 3 |
| **Steps** | 1. Navigate to "Lessons".  2. Select "Grade 3" filter. |
| **Pass Criteria** | List shows only Grade 3 lessons. |
| **Fail Criteria** | Filter dysfunctional. |

### 4. Progress, Feedback & Evaluation

#### TC-EVAL-001 – View Results

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-EVAL-001 |
| **Associated Use Case** | View Quiz Results (REQ-5.3, 5.4, 5.5) |
| **Title** | All Roles View Student Quiz Results |
| **Objective** | Verify visibility of results based on permissions. |
| **Steps** | 1. (Student) View "My Progress".  2. (Teacher) View "Student Performance".  3. (Advisor) View student details. |
| **Pass Criteria** | Correct detail level shown for each role. |
| **Fail Criteria** | Data incorrect or unauthorized access allowed. |

#### TC-EVAL-002 – Submit Feedback

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-EVAL-002 |
| **Associated Use Case** | Feedback (REQ-5.6, 5.7) |
| **Title** | Teacher Submits Feedback and Student Views It |
| **Objective** | Verify feedback loop. |
| **Steps** | 1. (Teacher) Click "Provide Feedback" on quiz.  2. Send text.  3. (Student) Refresh dashboard. |
| **Pass Criteria** | Feedback saved and visible to student. |
| **Fail Criteria** | Save fails or not visible. |

### 5. Artificial Intelligence (Basic)

#### TC-AI-001 – Homework Helper

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AI-001 |
| **Associated Use Case** | AI Homework Helper (REQ-AI-01) |
| **Title** | Student Uses the AI Homework Helper |
| **Objective** | Verify AI provides relevant responses. |
| **Test Data** | Q: "Explain second-degree equation" |
| **Steps** | 1. Navigate to AI Helper.  2. Type question.  3. Click "Ask". |
| **Pass Criteria** | Relevant, explanatory response provided. |
| **Fail Criteria** | Irrelevant, incorrect, or hallucinated response. |

### 6. Non-Functional Requirements

#### TC-NF-001 – Frontend Validation

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-NF-001 |
| **Associated Use Case** | Form Validation (REQ-2.7, 2.8) |
| **Title** | Verify Frontend Input Validation |
| **Test Data** | Email: invalid-email |
| **Steps** | 1. Enter invalid email format.  2. Click out of field. |
| **Pass Criteria** | Inline error appears immediately (no reload). |
| **Fail Criteria** | No validation or server-side only error. |

#### TC-SEC-001 – Secure Hashing

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-SEC-001 |
| **Associated Use Case** | Secure Storage (REQ-2.9) |
| **Title** | Verify Passwords are Hashed in the Database |
| **Objective** | Verify protection of user credentials. |
| **Steps** | 1. Access DB via admin tool.  2. Inspect user password column. |
| **Pass Criteria** | Password is a hash/unreadable string. |
| **Fail Criteria** | Plaintext password visible. |

### 7. Live Sessions

#### TC-LS-001 – Create Live Session

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-LS-001 |
| **Associated Use Case** | Create Live Session (REQ-LS-01) |
| **Title** | Teacher Successfully Schedules a Live Session |
| **Objective** | Verify creation of pending sessions. |
| **Test Data** | Title: Math Rev, Date: Tomorrow, Time: 10:00 |
| **Steps** | 1. Click "Create Session".  2. Enter details.  3. Click "Schedule". |
| **Pass Criteria** | Session created; Status "PENDING". |
| **Fail Criteria** | Creation prevented erroneously. |

#### TC-LS-002 – Join Live Session

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-LS-002 |
| **Associated Use Case** | Join Live Session (REQ-LS-12) |
| **Title** | Student Joins Active Session (Jitsi Integration) |
| **Prerequisites** | Session is "Active" (time is now). |
| **Steps** | 1. Navigate to "My Sessions".  2. Click "Join Now". |
| **Pass Criteria** | Jitsi popup opens; Attendance recorded. |
| **Fail Criteria** | Button disabled or Jitsi fails. |

#### TC-LS-003 – Validate Timing

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-LS-003 |
| **Associated Use Case** | Validate Timing (REQ-LS-04) |
| **Title** | Prevent Early Joining of Future Sessions |
| **Objective** | Verify access control based on time. |
| **Steps** | 1. Locate future session.  2. Attempt to join. |
| **Pass Criteria** | "Join" button disabled or access denied. |
| **Fail Criteria** | Student enters room early. |

#### TC-LS-004 – View Teacher Schedule

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-LS-004 |
| **Associated Use Case** | View Schedule (REQ-LS-05) |
| **Title** | Teacher Views/Filters Session List |
| **Steps** | 1. Go to "My Sessions".  2. Filter by "Completed". |
| **Pass Criteria** | List updates to show finished sessions. |
| **Fail Criteria** | List empty or incorrect. |

### 8. Admin System

#### TC-ADM-001 – Create Advisor

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-ADM-001 |
| **Associated Use Case** | Create Advisor (REQ-ADS-03) |
| **Title** | Admin Creates a New Advisor Account |
| **Objective** | Verify admin creation workflow. |
| **Steps** | 1. Go to "Advisors Management".  2. Click "Add Advisor".  3. Fill details and Create.  4. Verify login. |
| **Pass Criteria** | Account created; Login successful. |
| **Fail Criteria** | Permission or duplicate error. |

#### TC-ADM-002 – Deactivate Advisor

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-ADM-002 |
| **Associated Use Case** | Deactivate Advisor (REQ-ADS-07) |
| **Title** | Admin Deactivates an Advisor Account |
| **Objective** | Verify login prevention upon deactivation. |
| **Steps** | 1. Locate Advisor.  2. Toggle "Inactive".  3. Confirm. |
| **Pass Criteria** | Status "Inactive"; Login denied. |
| **Fail Criteria** | Advisor can still log in. |

#### TC-ADM-003 – Invalidate Active Sessions

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-ADM-003 |
| **Associated Use Case** | Invalidate Sessions (REQ-ADS-08) |
| **Title** | Security: Immediate Session Termination |
| **Objective** | Verify token invalidation. |
| **Prerequisites** | Advisor logged in on Browser B. |
| **Steps** | 1. (Admin) Deactivate account.  2. (Advisor) Refresh page on Browser B. |
| **Pass Criteria** | Immediate logout / 401 Unauthorized. |
| **Fail Criteria** | Session remains active. |

#### TC-ADM-004 – Reset Advisor Password

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-ADM-004 |
| **Associated Use Case** | Reset Password (REQ-ADS-05) |
| **Title** | Admin Resets Advisor Password |
| **Steps** | 1. Select Advisor.  2. Reset Password.  3. Save new password. |
| **Pass Criteria** | Login with new password successful. |
| **Fail Criteria** | Update fails. |

### 9. Recurring Sessions

#### TC-RS-001 – Create Session Template

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-RS-001 |
| **Associated Use Case** | Create Template (UC-RS-01) |
| **Title** | Create a Weekly Recurring Template |
| **Test Data** | Pattern: Weekly, Monday, 09:00 AM |
| **Steps** | 1. Click "Create Template".  2. Fill details and pattern.  3. Save. |
| **Pass Criteria** | Template saved as "Active". |
| **Fail Criteria** | Save fails. |

#### TC-RS-002 – View Generated Sessions

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-RS-002 |
| **Associated Use Case** | View Generated (UC-RS-08) |
| **Title** | Verify Automatic Session Generation |
| **Objective** | Verify instances created from template. |
| **Steps** | 1. Trigger generation.  2. Check Calendar. |
| **Pass Criteria** | Future Monday sessions appear. |
| **Fail Criteria** | No sessions generated. |

#### TC-RS-003 – Pause Template

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-RS-003 |
| **Associated Use Case** | Pause Template (UC-RS-05) |
| **Title** | Teacher Pauses a Recurring Template |
| **Objective** | Verify generation stop. |
| **Steps** | 1. Click "Pause".  2. Confirm. |
| **Pass Criteria** | Status "Paused"; Generation stops. |
| **Fail Criteria** | Status remains Active. |

#### TC-RS-004 – Edit Session Template

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-RS-004 |
| **Associated Use Case** | Edit Template (UC-RS-03) |
| **Title** | Edit Template and Propagate Changes |
| **Test Data** | Change time to 11:00 AM |
| **Steps** | 1. Edit template.  2. Change time.  3. Check "Apply to future".  4. Save. |
| **Pass Criteria** | Future sessions moved to 11:00 AM. |
| **Fail Criteria** | Sessions remain at old time. |

#### TC-RS-005 – Assign Template to Group

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-RS-005 |
| **Associated Use Case** | Assign Template (UC-RS-11) |
| **Title** | Advisor Assigns Template to Student Group |
| **Objective** | Verify bulk assignment. |
| **Steps** | 1. Find template.  2. Assign to "Grade 5 A".  3. Submit. |
| **Pass Criteria** | All group members see sessions in schedule. |
| **Fail Criteria** | Schedule not visible to group. |

### 10. AI Engine & Smart Helper

#### TC-AI-001 – Visual Search

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AI-001 |
| **Associated Use Case** | Visual Search / CLIP (REQ-AI-01) |
| **Title** | Student Searches using an Image |
| **Test Data** | Image of Right-angled Triangle |
| **Steps** | 1. Open Smart Helper.  2. Upload image.  3. Search. |
| **Pass Criteria** | System identifies shape/content; Returns Geometry lessons. |
| **Fail Criteria** | Unrelated results or failure to process. |

#### TC-AI-002 – Voice Search

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AI-002 |
| **Associated Use Case** | Voice Search / Whisper (REQ-AI-02) |
| **Title** | Student Searches using Voice Command |
| **Test Data** | Input: "Explain the Pythagorean theorem" |
| **Steps** | 1. Click Mic icon.  2. Speak phrase.  3. Stop recording. |
| **Pass Criteria** | Text transcribed correctly; Relevant results shown. |
| **Fail Criteria** | Bad transcription or no results. |

#### TC-AI-003 – Video Semantic Search

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AI-003 |
| **Associated Use Case** | Video Search (REQ-AI-03) |
| **Title** | Search Inside Video (Timestamp Jump) |
| **Objective** | Verify timestamp deep-linking. |
| **Steps** | 1. Search "How does photosynthesis happen?".  2. Click video result. |
| **Pass Criteria** | Video plays from exact explanation timestamp (e.g., 03:45). |
| **Fail Criteria** | Video starts at 00:00. |

#### TC-AI-004 – Smart RAG Answer

|  |  |
| --- | --- |
| Field | Value |
| **Test ID** | TC-AI-004 |
| **Associated Use Case** | RAG (REQ-AI-04) |
| **Title** | AI Generates a Textual Answer |
| **Objective** | Verify generated summary. |
| **Test Data** | Q: "Three states of matter" |
| **Steps** | 1. Enter question.  2. Observe "AI Answer". |
| **Pass Criteria** | Summary displayed: "Solid, Liquid, Gas..." with citations. |
| **Fail Criteria** | Generic error or hallucination. |

# 4.1 Requirements Traceability Matrix (RTM) V0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Title | Analysis Section | Design Section | Code Implementation | Test Case ID |
| **REQ-2.1** | User Registration | [4.6 Sprint 2 Analysis](#_UC-2.1_–_Register) |  |  | [TC-AUTH-001](#_Table_67_-) |
| **REQ-2.2** | Login | [4.6 Sprint 2 Analysis](#_UC-2.2_–_Login) |  |  | [TC-AUTH-002](#_Table_67_-) |
| **REQ-2.3** | User Profile Viewing | [4.6 Sprint 2 Analysis](#_UC-2.3_–_View) |  |  | [TC-USER-001](#_Table_67_-) |
| **REQ-2.4** | User Profile Editing | [4.6 Sprint 2 Analysis](#_UC-2.4_–_Edit) |  |  | [TC-USER-001](#_Table_67_-) |
| **REQ-2.5** | Logout | [4.6 Sprint 2 Analysis](#_UC-2.5_–_Logout) |  |  | [TC-AUTH-003](#_Table_67_-) |
| **REQ-2.6** | Reset Password | [4.6 Sprint 2 Analysis](#_UC-2.6_–_Reset) |  |  | [TC-AUTH-004](#_Table_67_-) |
| **REQ-4.1** | Upload Video Lesson | [4.7 Sprint 4 Analysis](#_UC-4.1_–_Upload) |  |  | [TC-CNT-001](#_Table_67_-) |
| **REQ-4.2** | Upload PDF Lesson | [4.7 Sprint 4 Analysis](#_UC-4.2_–_Upload) |  |  | [TC-CNT-001](#_Table_67_-) |
| **REQ-4.3** | Upload Exercises (MCQ) | [4.7 Sprint 4 Analysis](#_UC-4.3_–_Upload) |  |  | [TC-CNT-002](#_Table_67_-) |
| **REQ-4.4** | View My Lessons | [4.7 Sprint 4 Analysis](#_UC-4.4_–_View) |  |  | [TC-CNT-003](#_Table_67_-) |
| **REQ-4.5** | Update Lesson | [4.7 Sprint 4 Analysis](#_UC-4.5_–_Update) |  |  | [TC-CNT-003](#_Table_67_-) |
| **REQ-4.6** | Delete Lesson | [4.7 Sprint 4 Analysis](#_UC-4.6_–_Delete) |  |  | [TC-CNT-003](#_Table_67_-) |
| **REQ-4.7** | Browse Assigned Lessons | [4.7 Sprint 4 Analysis](#_UC-4.7_–_Browse) |  |  | [TC-STU-001](#_Table_67_-) |
| **REQ-4.8** | Select Lesson | [4.7 Sprint 4 Analysis](#_UC-4.8_–_Select) |  |  | [TC-STU-001](#_Table_67_-) |
| **REQ-4.9** | Download PDF Lessons | [4.7 Sprint 4 Analysis](#_UC-4.9_–_Download) |  |  | [TC-STU-002](#_Table_67_-) |
| **REQ-4.10** | Solve Exercises | [4.7 Sprint 4 Analysis](#_UC-4.10_–_Solve) |  |  | [TC-STU-002](#_Table_67_-) |
| **REQ-4.11** | View Student Progress | [4.7 Sprint 4 Analysis](#_UC-4.11_–_View) |  |  | [TC-EVAL-001](#_Table_67_-) |
| **REQ-4.12** | Assign Lessons to Student | [4.7 Sprint 4 Analysis](#_UC-4.12_–_Assign) |  |  | [TC-ADV-001](#_Table_67_-) |
| **REQ-4.13** | Search Lessons by Filter | [4.7 Sprint 4 Analysis](#_UC-4.13_–_Search) |  |  | [TC-ADV-002](#_Table_67_-) |
| **REQ-5.4** | View Quiz Performance (Teacher) | [4.8 Sprint 5 Analysis](#_UC-5.4_–_View) |  |  | [TC-EVAL-001](#_Table_67_-) |
| **REQ-5.5** | View All Quiz Results (Advisor) | [4.8 Sprint 5 Analysis](#_UC-5.5_–_View) |  |  | [TC-EVAL-001](#_Table_67_-) |
| **REQ-5.6** | Submit Feedback | [4.8 Sprint 5 Analysis](#_UC-5.6_–_Submit) |  |  | [TC-EVAL-002](#_Table_67_-) |
| **REQ-5.7** | View Feedback | [4.8 Sprint 5 Analysis](#_UC-5.7_–_View) |  |  | [TC-EVAL-002](#_Table_67_-) |
| **REQ-LS-01** | Create Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-01_–_Create) |  |  | [TC-LS-001](#_Table_67_-) |
| **REQ-LS-02** | Edit Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-02_–_Edit) |  |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-03** | Cancel Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-03_–_Cancel) |  |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-04** | Join Live Session (Host) | [4.10 Sprint 9 Analysis](#_UC-LS-04_–_Join) |  |  | [TC-LS-002](#_Table_67_-) |
| **REQ-LS-05** | View Teacher's Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-05_–_View) |  |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-06** | View Assigned Students | [4.10 Sprint 9 Analysis](#_UC-LS-06_–_View) |  |  | [TC-LS-001](#_Table_67_-) |
| **REQ-LS-07** | View All Live Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-07_–_View) |  |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-08** | Assign Students to Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-08_–_Assign) |  |  | [TC-LS-003](#_Table_67_-) |
| **REQ-LS-09** | Unassign Students | [4.10 Sprint 9 Analysis](#_UC-LS-09_–_Unassign) |  |  | [TC-LS-003](#_Table_67_-) |
| **REQ-LS-10** | Modify Student Assignments | [4.10 Sprint 9 Analysis](#_UC-LS-10_–_Modify) |  |  | [TC-LS-003](#_Table_67_-) |
| **REQ-LS-11** | View Assigned Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-11_–_View) |  |  | [TC-LS-002](#_Table_67_-) |
| **REQ-LS-12** | Join Live Session (Student) | [4.10 Sprint 9 Analysis](#_UC-LS-12_–_Join) |  |  | [TC-LS-002](#_Table_67_-) |
| **REQ-LS-13** | View Session Details | [4.10 Sprint 9 Analysis](#_UC-LS-13_–_View) |  |  | [TC-LS-003](#_Table_67_-) |
| **REQ-ADS-01** | Admin Login | [4.11 Sprint 10 Analysis](#_REQ-ADS-01_–_Admin) |  |  | [TC-ADM-001](#_Table_67_-) |
| **REQ-ADS-02** | View Advisors List | [4.11 Sprint 10 Analysis](#_REQ-ADS-02_–_View) |  |  | [TC-ADM-001](#_Table_67_-) |
| **REQ-ADS-03** | Create New Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-03_–_Create) |  |  | [TC-ADM-001](#_Table_67_-) |
| **REQ-ADS-04** | Delete Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-04_–_Delete) |  |  | [TC-ADM-002](#_Table_67_-) |
| **REQ-ADS-05** | Reset Advisor Password | [4.11 Sprint 10 Analysis](#_REQ-ADS-05_–_Reset) |  |  | [TC-ADM-004](#_Table_67_-) |
| **REQ-ADS-06** | Activate Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-06_–_Activate) |  |  | [TC-ADM-002](#_Table_67_-) |
| **REQ-ADS-07** | Deactivate Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-07_–_Deactivate) |  |  | [TC-ADM-002](#_Table_67_-) |
| **REQ-RS-01** | Create Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-01_–_Create) |  |  | [TC-RS-001](#_Table_67_-) |
| **REQ-RS-02** | View Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-02_–_View) |  |  | [TC-RS-001](#_Table_67_-) |
| **REQ-RS-03** | Edit Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-03_–_Edit) |  |  | [TC-RS-004](#_Table_67_-) |
| **REQ-RS-04** | Delete Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-04_–_Delete) |  |  | [TC-RS-001](#_Table_67_-) |
| **REQ-RS-05** | Pause Template Status | [4.12 Sprint 11 Analysis](#_UC-RS-05_–_Pause) |  |  | [TC-RS-003](#_Table_67_-) |
| **REQ-RS-06** | Resume Paused Templates | [4.12 Sprint 11 Analysis](#_UC-RS-06_–_Resume) |  |  | [TC-RS-003](#_Table_67_-) |
| **REQ-RS-07** | End Template Permanently | [4.12 Sprint 11 Analysis](#_UC-RS-07_–_End) |  |  | [TC-RS-003](#_Table_67_-) |
| **REQ-RS-10** | View All Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-09_–_View) |  |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-11** | Assign Templates | [4.12 Sprint 11 Analysis](#_UC-RS-10_–_Assign) |  |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-12** | Create Student Groups | [4.12 Sprint 11 Analysis](#_UC-RS-11_–_Create) |  |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-13** | Manage Student Groups | [4.12 Sprint 11 Analysis](#_UC-RS-12_–_Manage) |  |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-15** | View Assigned Templates | [4.12 Sprint 11 Analysis](#_UC-RS-14_–_View) |  |  | [TC-RS-005](#_Table_67_-) |
| **REQ-AI-01** | Upload Image for Search | 4.9 AI Analysis |  |  | [TC-AI-001](#_Table_67_-) |
| **REQ-AI-02** | Voice Search (Whisper) | 4.9 AI Analysis |  |  | [TC-AI-002](#_Table_67_-) |
| **REQ-AI-03** | Search Inside Videos | 4.9 AI Analysis |  |  | [TC-AI-003](#_Table_67_-) |
| **REQ-AI-04** | Smart Homework Helper | 4.9 AI Analysis |  |  | [TC-AI-004](#_Table_67_-) |

Table 71 - RTM V0

**CHAPTER 5 – System Design**

### 5.1 Introduction

The system adopts a **3-Tier Architecture** embedded within a traditional Client-Server model. This architectural approach provides a clear separation of concerns, enhances maintainability, and enables scalable development. It ensures efficient data flow and interaction between components, supporting real-time live sessions, automated scheduling, and administrative control while maintaining modularity across each layer of the system.

### 5.2 Architecture Layers

#### 5.2.1 Client Side

This is the layer where users interact with the platform through a web-based interface built using **React.js**. It handles user-facing operations such as registration, login, profile updates, managing live session schedules, creating recurring session templates, and administrative monitoring. It communicates with the server via RESTful API calls to fetch schedules and manage educational content.

#### 5.2.2 Server Side

* **Presentation Layer:** Acts as the entry point for client requests through REST APIs powered by **Django REST Framework**. It handles request routing for authentication, session management, and admin tasks, passing calls to the appropriate business logic services.
* **Business Logic Layer:** Contains the core functional modules of the system. This includes:
  + **User Management:** Authentication and token generation (JWT).
  + **Live Session Management:** Logic for scheduling classes and integrating with Jitsi Meet.
  + **Recurring Automation:** Services that process Session Templates to automatically generate future sessions based on defined patterns.
  + **Admin Services:** Security protocols for user deactivation and session invalidation.

These responsibilities are structured into independent service components to improve clarity and testability.

* **Data Access Layer:** Responsible for communication with the database using **Django's ORM**. It provides data abstraction and handles all data operations such as retrieving, updating, and storing user profiles, session logs, templates, and student group assignments.

#### 5.2.3 Database

The system uses **MySQL** as its persistent storage solution. It stores core entities such as users, roles, and account status, as well as complex operational data including Live Sessions, Session Templates, Generated Sessions, and Student Groups. Data is accessed through the Data Access Layer using Django’s ORM, ensuring secure, efficient, and structured data management.

### Release 1: 5.3 Sprint2 Design

### 5.3.1 System Architecture & Component Diagram

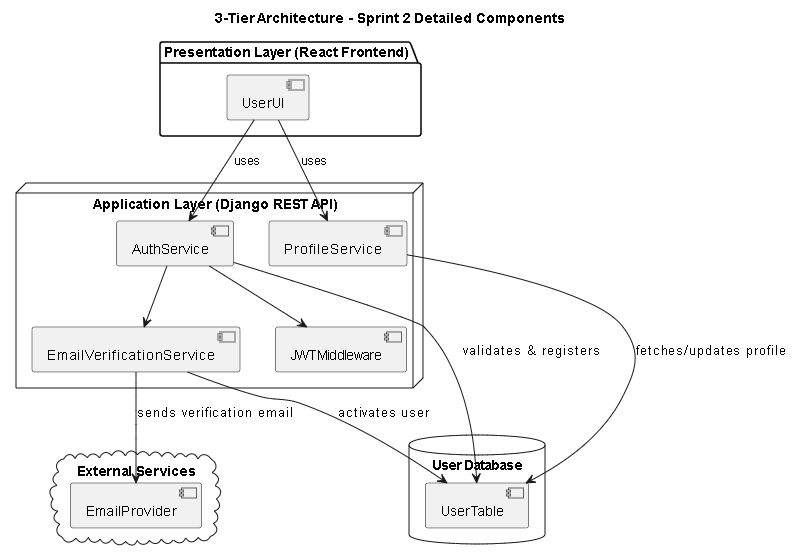


Figure 119 - Sprint 2 Architecture Diagram

### 5.4 Sprint4 Design

### 5.4.1 System Architecture & Component Diagram



Figure 120 - Sprint 4 Architecture Diagram

### 5.5 Sprint5 Design

### 5.3.1 System Architecture & Component Diagram

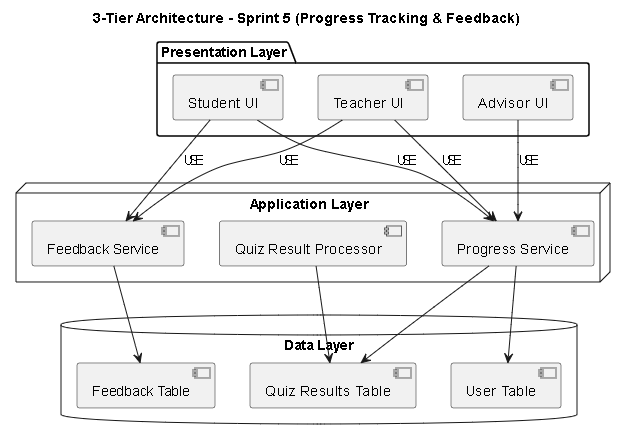


Figure 121 - Sprint 5 Architecture Diagram

### Release 2: 5.6 Sprint9 Design

### 5.6.1 System Architecture & Component Diagram

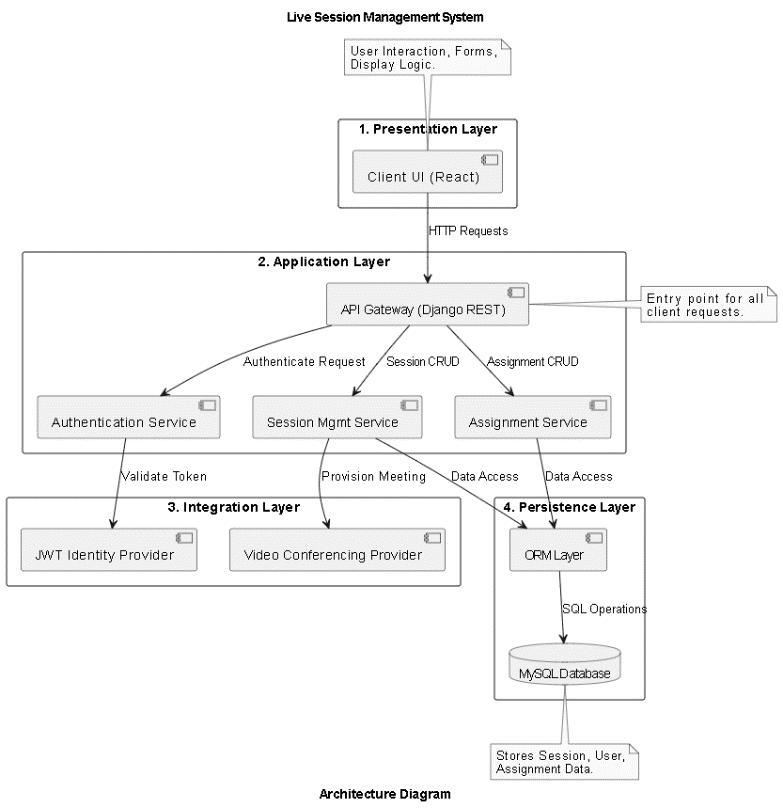
****

Figure 122 - Sprint 9 Architecture Diagram

5.6.2 Class Diagram

### 

Figure 123 - Sprint9 Class diagram

### 5.7 Sprint10 Design

### 5.7.1 System Architecture & Component Diagram

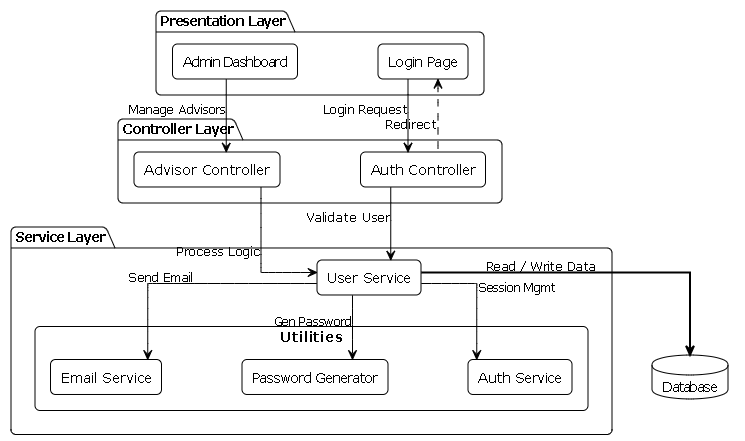


Figure 124 - Sprint 10 Architecture Diagram

### 5.7.2 Class Diagram

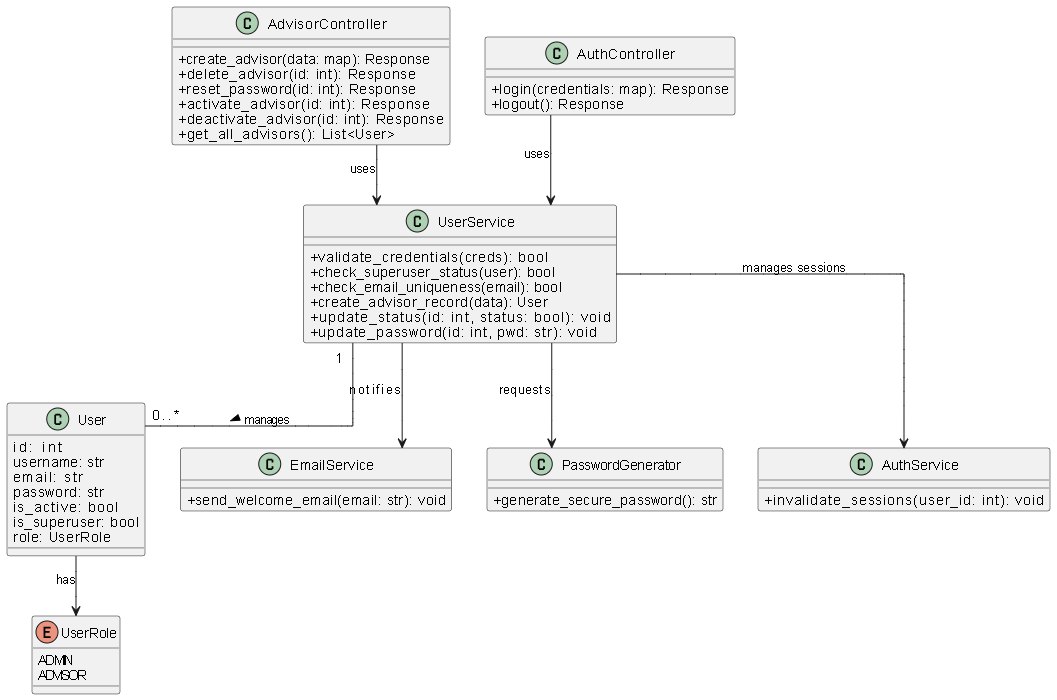


Figure 125 - Sprint 10 Class diagram

### 5.8 Sprint11 Design

### 5.8.1 System Architecture & Component Diagram

### 

Figure 126 - Sprint 11 Architecture Diagram

### 5.8.2 Class Diagram

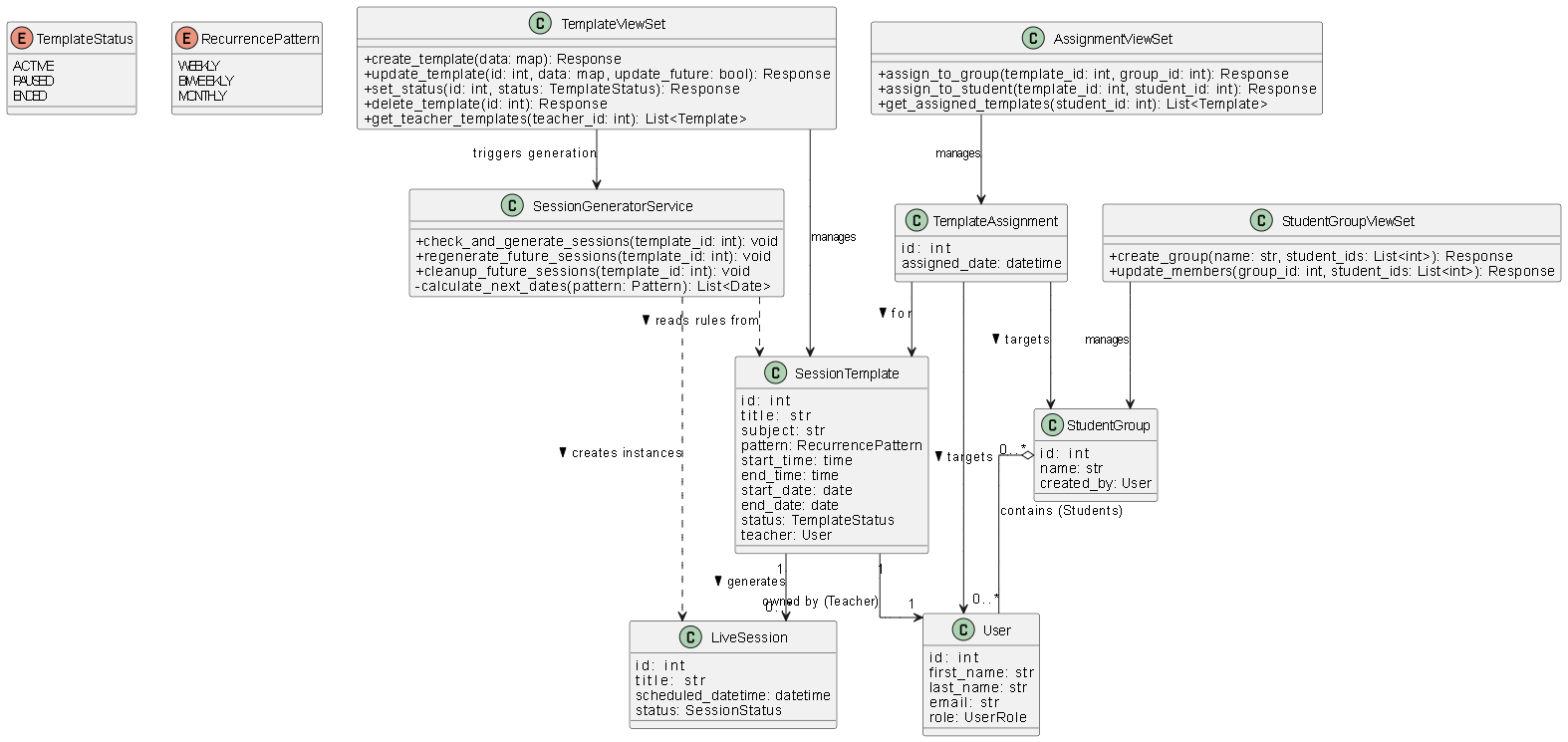


Figure 127 - Sprint 11 Class diagram

# 5.9 Requirements Traceability Matrix (RTM) V1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Title | Analysis Section | Design Section | Code Implementation | Test Case ID |
| **REQ-2.1** | User Registration | [4.6 Sprint 2 Analysis](#_UC-2.1_–_Register) | [Sprint 2 Design](#_Release_1:_5.3) |  | [TC-AUTH-001](#_Table_67_-) |
| **REQ-2.2** | Login | [4.6 Sprint 2 Analysis](#_UC-2.2_–_Login) | [Sprint 2 Design](#_Release_1:_5.3) |  | [TC-AUTH-002](#_Table_67_-) |
| **REQ-2.3** | User Profile Viewing | [4.6 Sprint 2 Analysis](#_UC-2.3_–_View) | [Sprint 2 Design](#_Release_1:_5.3) |  | [TC-USER-001](#_Table_67_-) |
| **REQ-2.4** | User Profile Editing | [4.6 Sprint 2 Analysis](#_UC-2.4_–_Edit) | [Sprint 2 Design](#_Release_1:_5.3) |  | [TC-USER-001](#_Table_67_-) |
| **REQ-2.5** | Logout | [4.6 Sprint 2 Analysis](#_UC-2.5_–_Logout) | [Sprint 2 Design](#_Release_1:_5.3) |  | [TC-AUTH-003](#_Table_67_-) |
| **REQ-2.6** | Reset Password | [4.6 Sprint 2 Analysis](#_UC-2.6_–_Reset) | [Sprint 2 Design](#_Release_1:_5.3) |  | [TC-AUTH-004](#_Table_67_-) |
| **REQ-4.1** | Upload Video Lesson | [4.7 Sprint 4 Analysis](#_UC-4.1_–_Upload) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-CNT-001](#_Table_67_-) |
| **REQ-4.2** | Upload PDF Lesson | [4.7 Sprint 4 Analysis](#_UC-4.2_–_Upload) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-CNT-001](#_Table_67_-) |
| **REQ-4.3** | Upload Exercises (MCQ) | [4.7 Sprint 4 Analysis](#_UC-4.3_–_Upload) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-CNT-002](#_Table_67_-) |
| **REQ-4.4** | View My Lessons | [4.7 Sprint 4 Analysis](#_UC-4.4_–_View) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-CNT-003](#_Table_67_-) |
| **REQ-4.5** | Update Lesson | [4.7 Sprint 4 Analysis](#_UC-4.5_–_Update) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-CNT-003](#_Table_67_-) |
| **REQ-4.6** | Delete Lesson | [4.7 Sprint 4 Analysis](#_UC-4.6_–_Delete) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-CNT-003](#_Table_67_-) |
| **REQ-4.7** | Browse Assigned Lessons | [4.7 Sprint 4 Analysis](#_UC-4.7_–_Browse) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-STU-001](#_Table_67_-) |
| **REQ-4.8** | Select Lesson | [4.7 Sprint 4 Analysis](#_UC-4.8_–_Select) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-STU-001](#_Table_67_-) |
| **REQ-4.9** | Download PDF Lessons | [4.7 Sprint 4 Analysis](#_UC-4.9_–_Download) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-STU-002](#_Table_67_-) |
| **REQ-4.10** | Solve Exercises | [4.7 Sprint 4 Analysis](#_UC-4.10_–_Solve) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-STU-002](#_Table_67_-) |
| **REQ-4.11** | View Student Progress | [4.7 Sprint 4 Analysis](#_UC-4.11_–_View) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-EVAL-001](#_Table_67_-) |
| **REQ-4.12** | Assign Lessons to Student | [4.7 Sprint 4 Analysis](#_UC-4.12_–_Assign) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-ADV-001](#_Table_67_-) |
| **REQ-4.13** | Search Lessons by Filter | [4.7 Sprint 4 Analysis](#_UC-4.13_–_Search) | [Sprint 4 Design](#_5.4_Sprint4_Design) |  | [TC-ADV-002](#_Table_67_-) |
| **REQ-5.4** | View Quiz Performance (Teacher) | [4.8 Sprint 5 Analysis](#_UC-5.4_–_View) | [Sprint 5 Design](#_5.5_Sprint5_Design) |  | [TC-EVAL-001](#_Table_67_-) |
| **REQ-5.5** | View All Quiz Results (Advisor) | [4.8 Sprint 5 Analysis](#_UC-5.5_–_View) | [Sprint 5 Design](#_5.5_Sprint5_Design) |  | [TC-EVAL-001](#_Table_67_-) |
| **REQ-5.6** | Submit Feedback | [4.8 Sprint 5 Analysis](#_UC-5.6_–_Submit) | [Sprint 5 Design](#_5.5_Sprint5_Design) |  | [TC-EVAL-002](#_Table_67_-) |
| **REQ-5.7** | View Feedback | [4.8 Sprint 5 Analysis](#_UC-5.7_–_View) | [Sprint 5 Design](#_5.5_Sprint5_Design) |  | [TC-EVAL-002](#_Table_67_-) |
| **REQ-LS-01** | Create Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-01_–_Create) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-001](#_Table_67_-) |
| **REQ-LS-02** | Edit Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-02_–_Edit) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-03** | Cancel Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-03_–_Cancel) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-04** | Join Live Session (Host) | [4.10 Sprint 9 Analysis](#_UC-LS-04_–_Join) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-002](#_Table_67_-) |
| **REQ-LS-05** | View Teacher's Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-05_–_View) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-06** | View Assigned Students | [4.10 Sprint 9 Analysis](#_UC-LS-06_–_View) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-001](#_Table_67_-) |
| **REQ-LS-07** | View All Live Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-07_–_View) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-004](#_Table_67_-) |
| **REQ-LS-08** | Assign Students to Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-08_–_Assign) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-003](#_Table_67_-) |
| **REQ-LS-09** | Unassign Students | [4.10 Sprint 9 Analysis](#_UC-LS-09_–_Unassign) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-003](#_Table_67_-) |
| **REQ-LS-10** | Modify Student Assignments | [4.10 Sprint 9 Analysis](#_UC-LS-10_–_Modify) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-003](#_Table_67_-) |
| **REQ-LS-11** | View Assigned Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-11_–_View) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-002](#_Table_67_-) |
| **REQ-LS-12** | Join Live Session (Student) | [4.10 Sprint 9 Analysis](#_UC-LS-12_–_Join) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-002](#_Table_67_-) |
| **REQ-LS-13** | View Session Details | [4.10 Sprint 9 Analysis](#_UC-LS-13_–_View) | [Sprint 9 Design](#_Release_2:_5.6) |  | [TC-LS-003](#_Table_67_-) |
| **REQ-ADS-01** | Admin Login | [4.11 Sprint 10 Analysis](#_REQ-ADS-01_–_Admin) | [Sprint 10 Design](#_5.7_Sprint10_Design) |  | [TC-ADM-001](#_Table_67_-) |
| **REQ-ADS-02** | View Advisors List | [4.11 Sprint 10 Analysis](#_REQ-ADS-02_–_View) | [Sprint 10 Design](#_5.7_Sprint10_Design) |  | [TC-ADM-001](#_Table_67_-) |
| **REQ-ADS-03** | Create New Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-03_–_Create) | [Sprint 10 Design](#_5.7_Sprint10_Design) |  | [TC-ADM-001](#_Table_67_-) |
| **REQ-ADS-04** | Delete Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-04_–_Delete) | [Sprint 10 Design](#_5.7_Sprint10_Design) |  | [TC-ADM-002](#_Table_67_-) |
| **REQ-ADS-05** | Reset Advisor Password | [4.11 Sprint 10 Analysis](#_REQ-ADS-05_–_Reset) | [Sprint 10 Design](#_5.7_Sprint10_Design) |  | [TC-ADM-004](#_Table_67_-) |
| **REQ-ADS-06** | Activate Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-06_–_Activate) | [Sprint 10 Design](#_5.7_Sprint10_Design) |  | [TC-ADM-002](#_Table_67_-) |
| **REQ-ADS-07** | Deactivate Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-07_–_Deactivate) | [Sprint 10 Design](#_5.7_Sprint10_Design) |  | [TC-ADM-002](#_Table_67_-) |
| **REQ-RS-01** | Create Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-01_–_Create) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-001](#_Table_67_-) |
| **REQ-RS-02** | View Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-02_–_View) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-001](#_Table_67_-) |
| **REQ-RS-03** | Edit Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-03_–_Edit) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-004](#_Table_67_-) |
| **REQ-RS-04** | Delete Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-04_–_Delete) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-001](#_Table_67_-) |
| **REQ-RS-05** | Pause Template Status | [4.12 Sprint 11 Analysis](#_UC-RS-05_–_Pause) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-003](#_Table_67_-) |
| **REQ-RS-06** | Resume Paused Templates | [4.12 Sprint 11 Analysis](#_UC-RS-06_–_Resume) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-003](#_Table_67_-) |
| **REQ-RS-07** | End Template Permanently | [4.12 Sprint 11 Analysis](#_UC-RS-07_–_End) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-003](#_Table_67_-) |
| **REQ-RS-10** | View All Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-09_–_View) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-11** | Assign Templates (Group) | [4.12 Sprint 11 Analysis](#_UC-RS-10_–_Assign) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-12** | Create Student Groups | [4.12 Sprint 11 Analysis](#_UC-RS-11_–_Create) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-13** | Manage Student Groups | [4.12 Sprint 11 Analysis](#_UC-RS-12_–_Manage) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-005](#_Table_67_-) |
| **REQ-RS-15** | View Assigned Templates | [4.12 Sprint 11 Analysis](#_UC-RS-14_–_View) | [Sprint 11 Design](#_5.8_Sprint11_Design) |  | [TC-RS-005](#_Table_67_-) |
| **REQ-AI-01** | Upload Image for Search | 4.9 AI Analysis | 5.11 AI Arch |  | [TC-AI-001](#_Table_67_-) |
| **REQ-AI-02** | Voice Search (Whisper) | 4.9 AI Analysis | 5.11 AI Arch |  | [TC-AI-002](#_Table_67_-) |
| **REQ-AI-03** | Search Inside Videos | 4.9 AI Analysis | 5.11 AI Arch |  | [TC-AI-003](#_Table_67_-) |
| **REQ-AI-04** | Smart Homework Helper | 4.9 AI Analysis | 5.11 AI Arch |  | [TC-AI-004](#_Table_67_-) |

Table 72 - RTM V1

**Chapter 6 - Implementation & Testing**

### 6.1 Introduction

This chapter presents the overall implementation and testing strategy for the AI-Powered Complementary Education System. It covers how each major component—User Management, Content Delivery, Live & Recurring Session Scheduling, Administrative Control, and the Advanced AI Engine—will be developed, integrated, and validated.

The goal is to demonstrate how the functional and non-functional requirements defined earlier are translated into a cohesive, working system. Furthermore, it describes the tools and methodologies used to ensure correctness, performance, security, and maintainability across the entire project lifecycle, moving from standard LMS features to advanced multimodal intelligence.

### 6.2 Technologies Used

* **Django** A high-level Python web framework used to build the core backend. It manages project configuration, URL routing, middleware, and settings. Django’s “batteries-included” philosophy accelerates development and enforces best practices for scalable, maintainable applications, serving as the foundation for both the LMS and the AI integration layers.
* **Django REST Framework (DRF)** An extension of Django that facilitates the rapid creation of RESTful APIs. DRF provides serializers, viewsets, routers, and authentication/permission classes, enabling secure and modular implementation of all backend endpoints. This includes handling complex logic for Live Session generation, Recurring Template automation, and serving the results from the AI inference engine.
* **MySQL** A robust, open-source relational database used for persistent storage of users, content, schedules, assessments, and analytics data. Its advanced feature set (ACID compliance, JSON fields, indexing) supports complex queries and large-scale data operations, ensuring data integrity for student groups and session logs.
* **React.js** The chosen frontend library for building a dynamic, component-driven user interface. React renders registration, login, profile, lesson views, scheduling calendars, and the Smart Homework Helper interface. Its virtual DOM and ecosystem (React Router, Context API) ensure responsive, maintainable client-side code that provides a seamless experience across devices.
* **Jitsi Meet (Integration)** An open-source video conferencing solution integrated into the platform to power the Live Sessions System. It allows the platform to securely generate unique meeting rooms, handle participant access control, and facilitate real-time audio/video interaction between teachers and students without requiring external software installation.
* **Background Task Processing (Celery & Redis)** Used to handle asynchronous operations and automation, specifically for the Recurring Sessions System. This technology stack manages the background cron jobs that scan active session templates and automatically generate future session instances, ensuring the system remains performant without blocking user requests.
* **AI & Machine Learning Stack (PyTorch, CLIP, Whisper)** The core engines powering the Smart Homework Helper. PyTorch serves as the deep learning framework, running the CLIP model for visual semantic search (image-to-text matching) and the Whisper model for high-accuracy automatic speech recognition (Voice Search), enabling multimodal accessibility for students.
* **FAISS (Facebook AI Similarity Search)** A library for efficient similarity search and clustering of dense vectors. It is used as the Vector Database backend to index educational content (video frames, PDF text). This allows the system to perform high-speed retrieval of relevant study materials based on the vector embeddings generated by the AI model

### 6.3 System Interfaces

**Login interface**

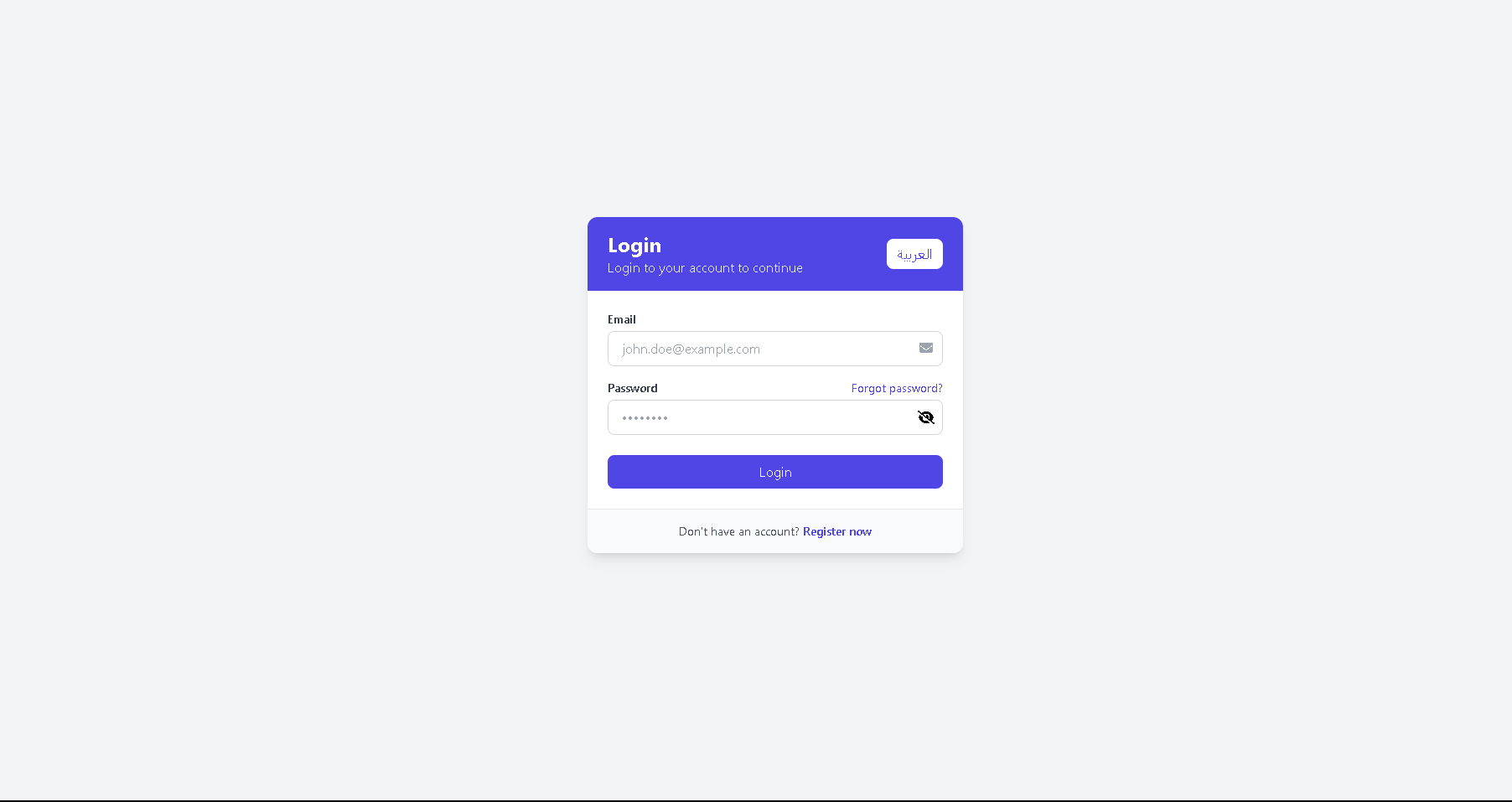
****

Figure 128 - login page

**Register interface**

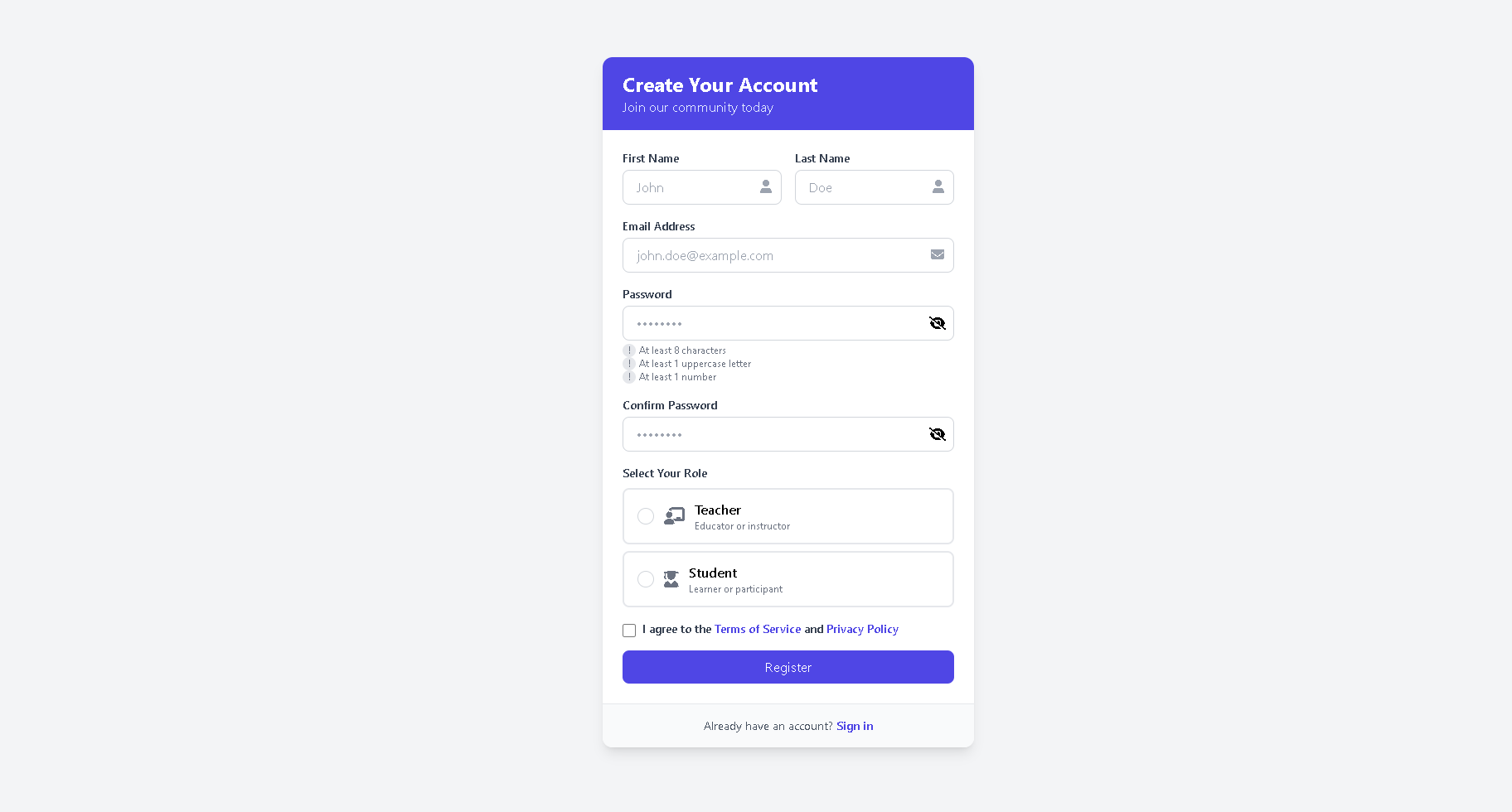


Figure 129 -Register

**Forgot Password interface**

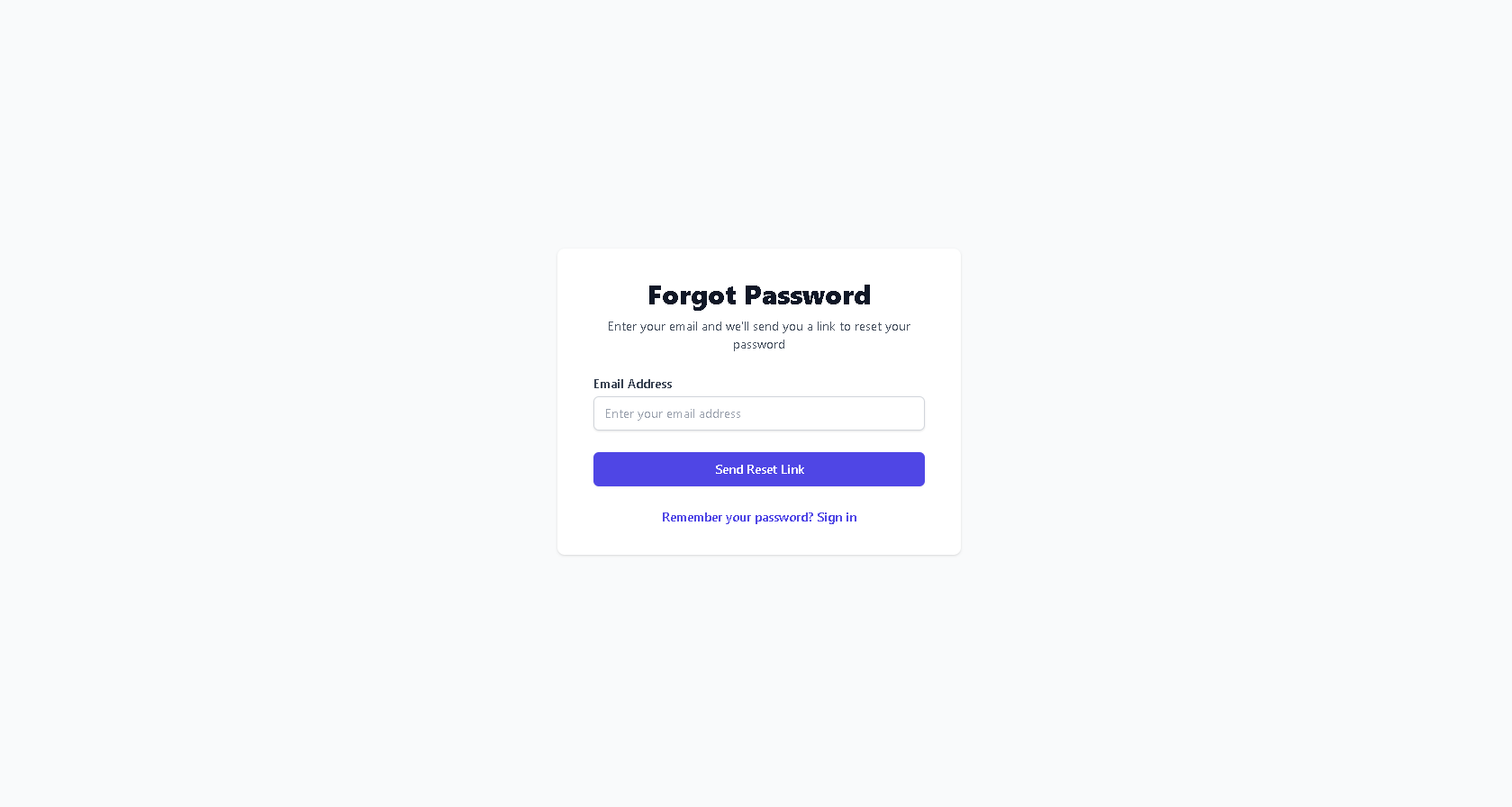
****

Figure 130 - Forgot Password

**Home page interface**

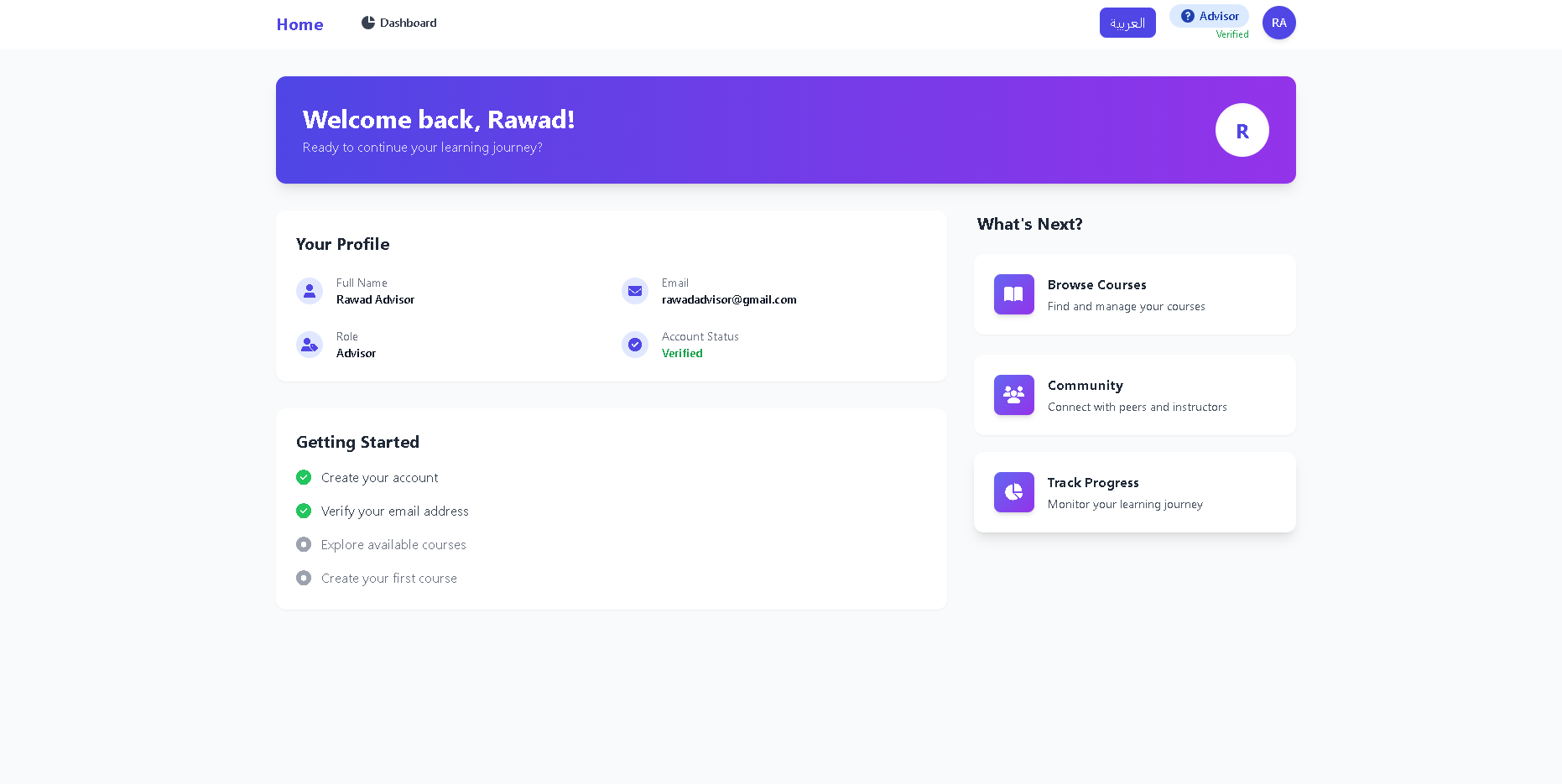


Figure 131 - Home page

**Advisor dashboard interface: Student list**

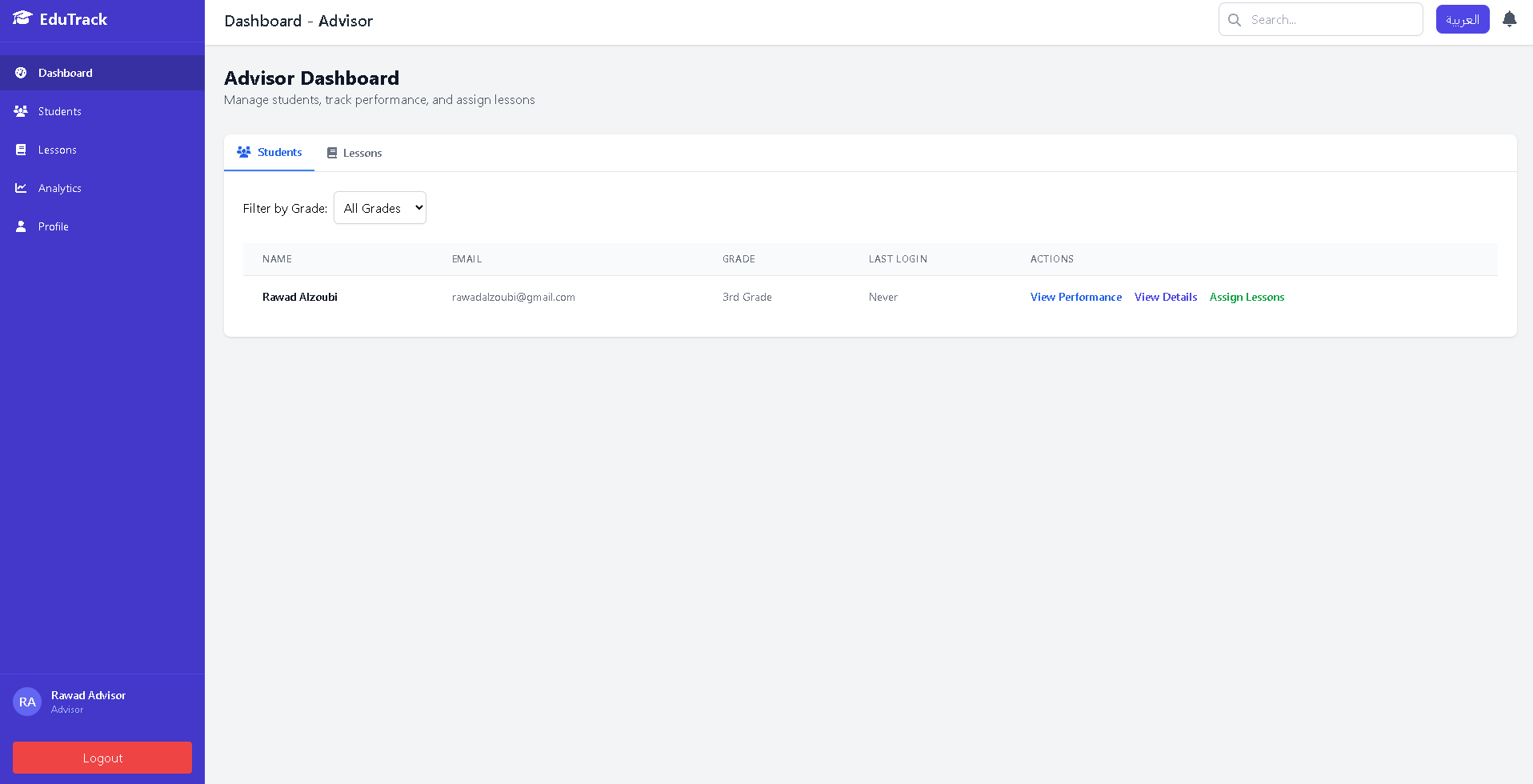
****

Figure 132 - Advisor dashboard: Student list

**Advisor dashboard interface: lesson list**

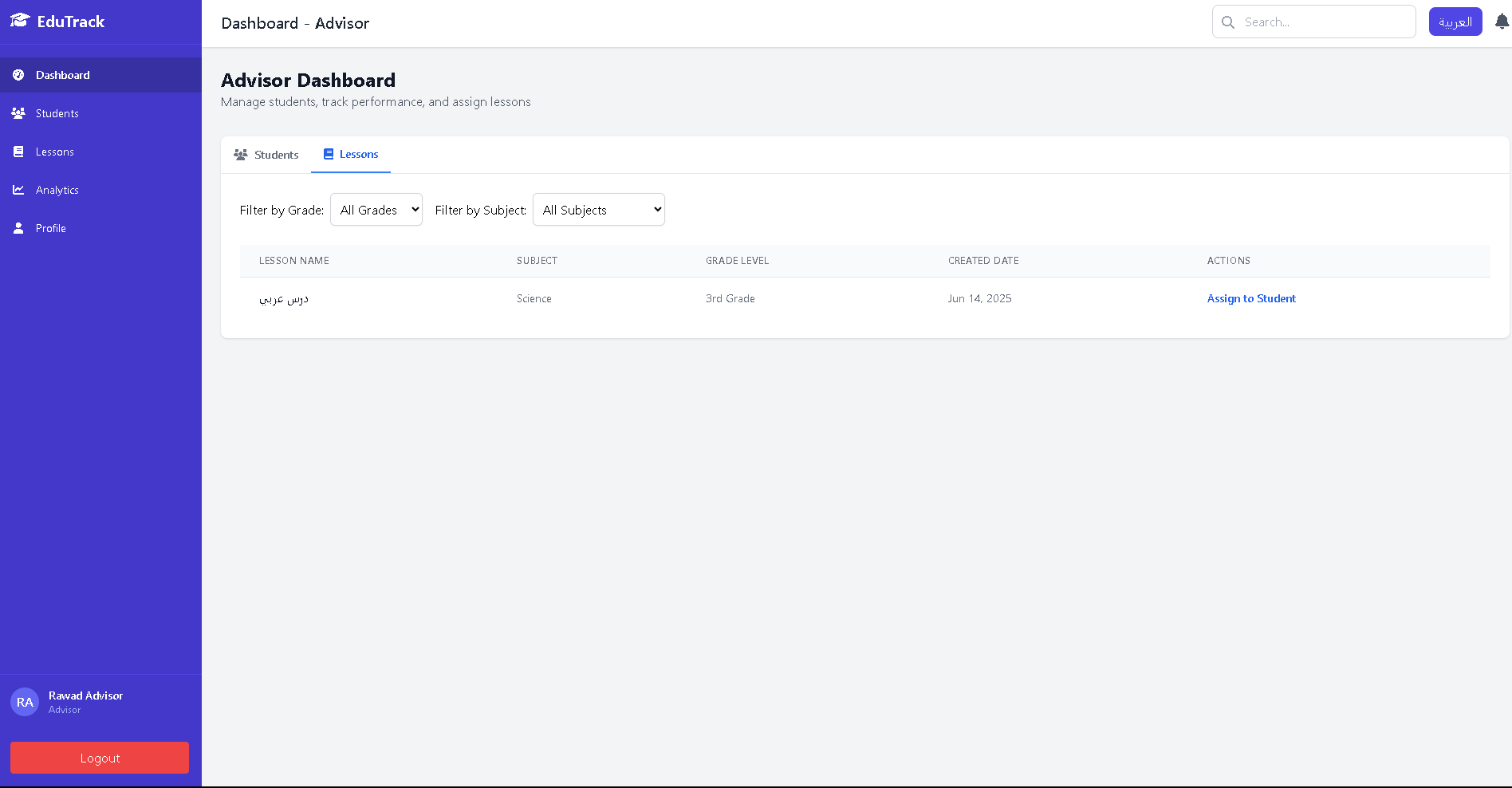
****

Figure 133 - Advisor dashboard: lesson list

**Assign lesson interface**

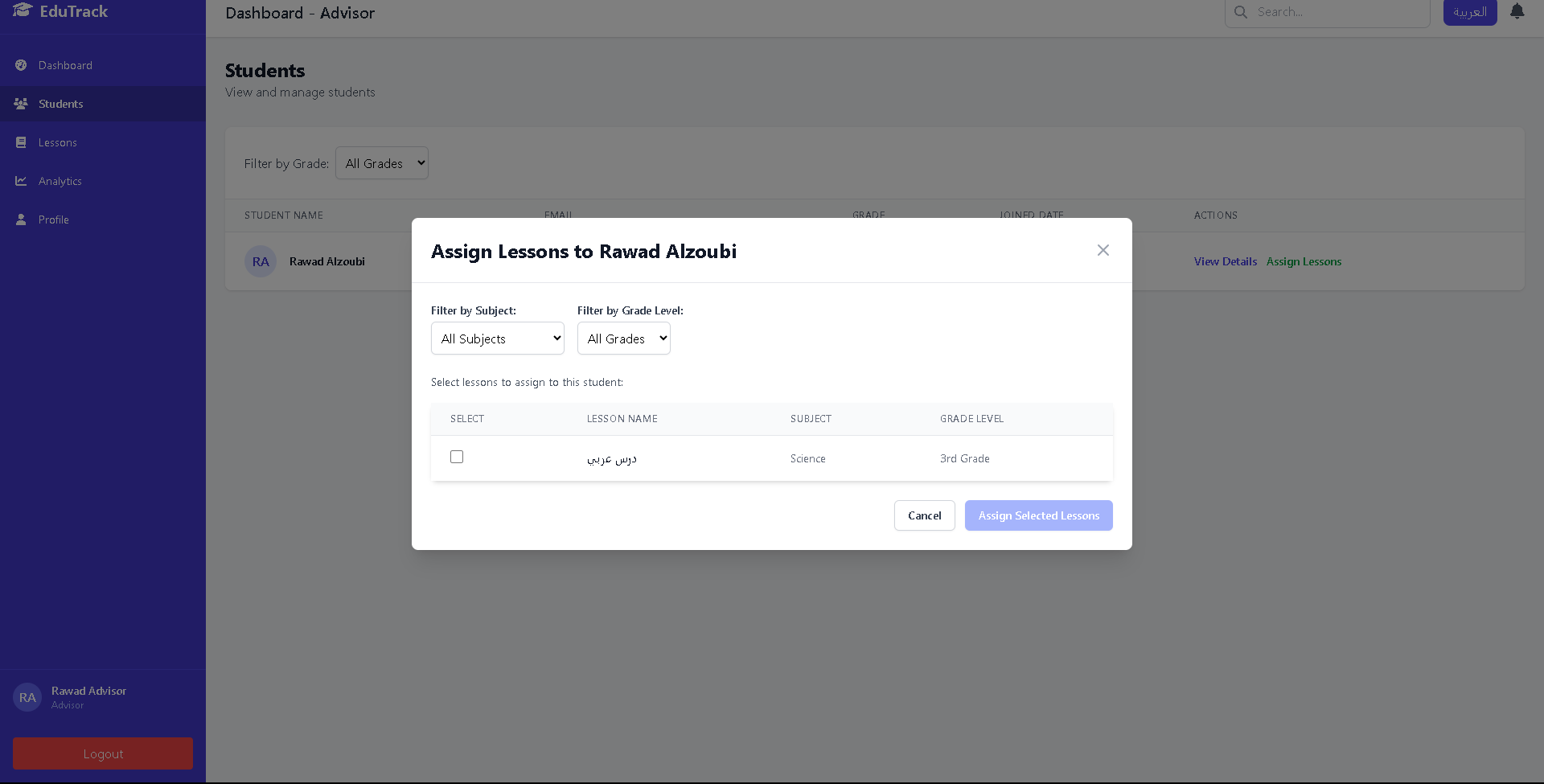
****

Figure 134 - Assign lesson

**Student details interface**

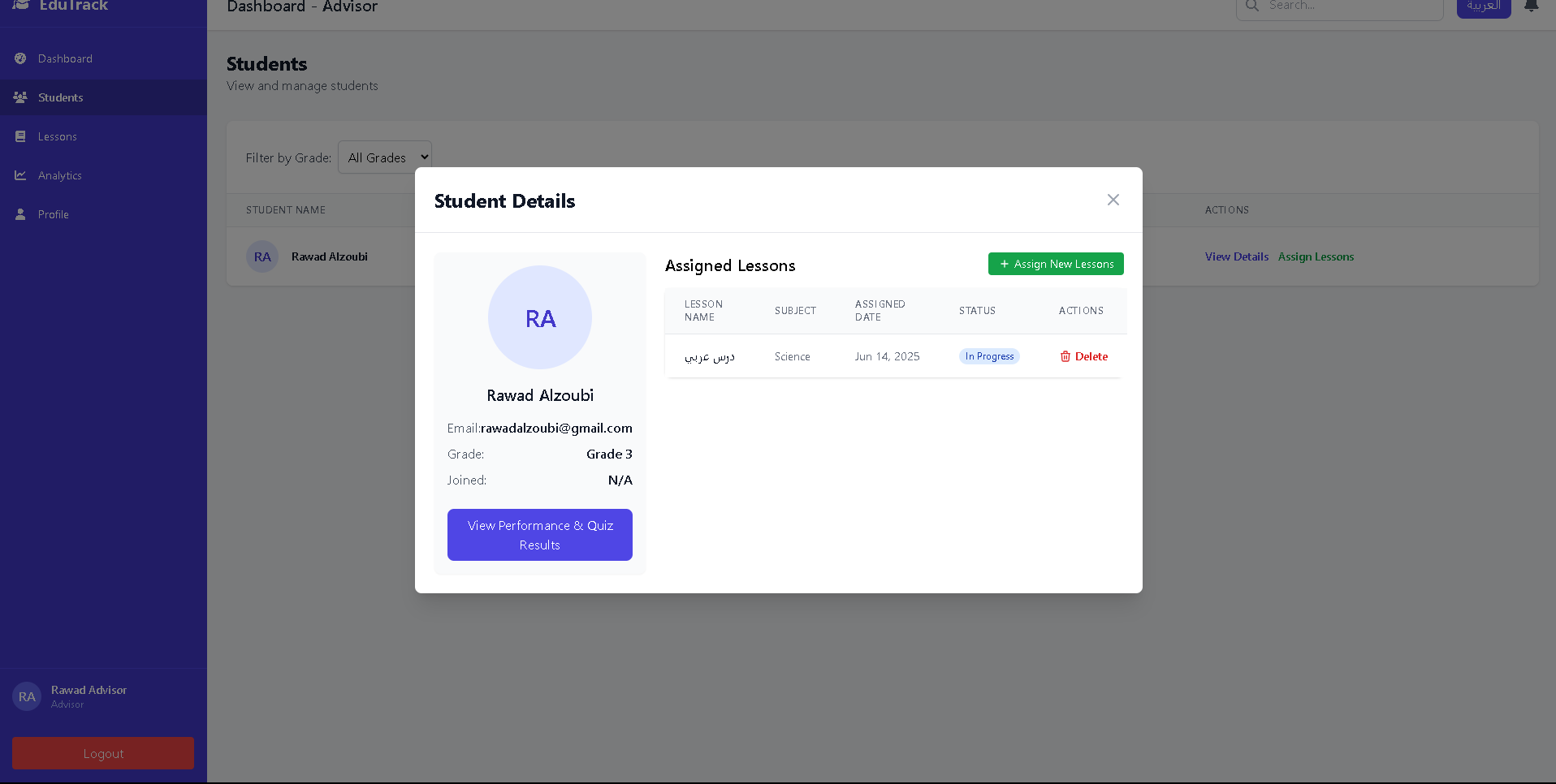
****

Figure 135 - Student details

**Student Performance interface**

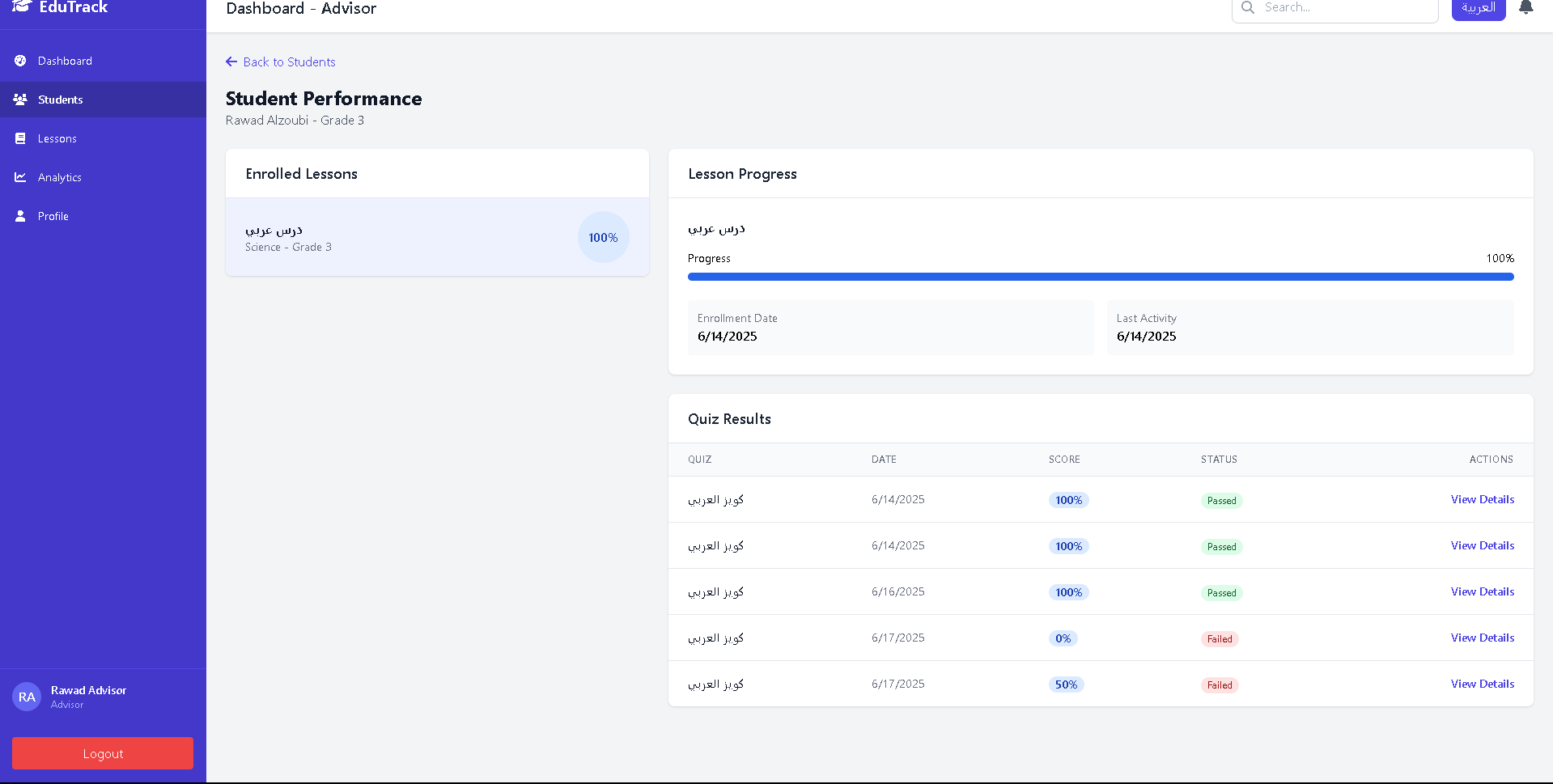
****

Figure 136 - Student Performance Page

**Quiz Result interface**

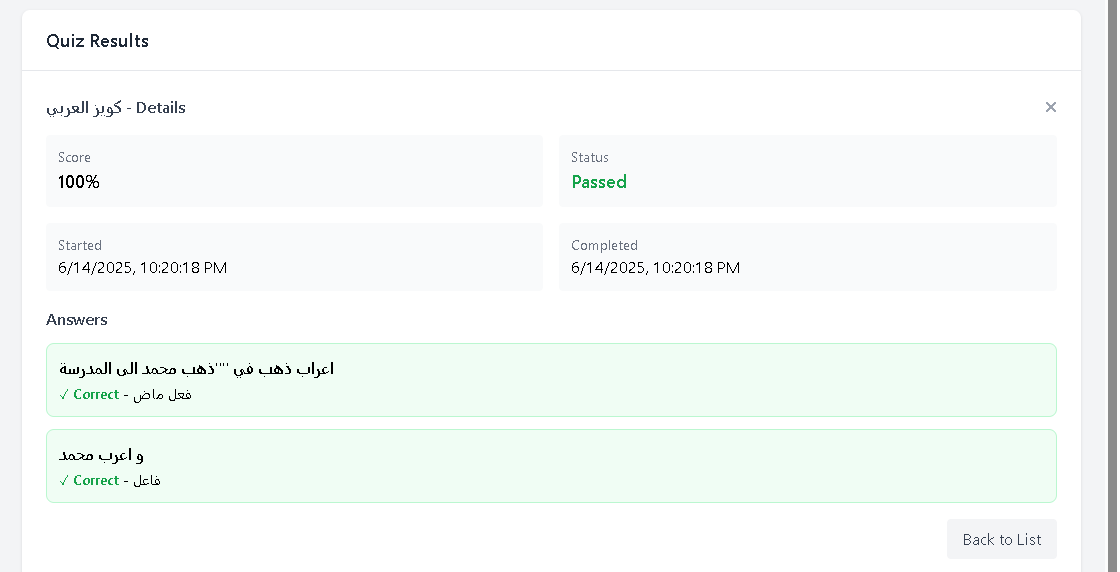
****

Figure 137 - Quiz Result page

**Advisor Profile interface**

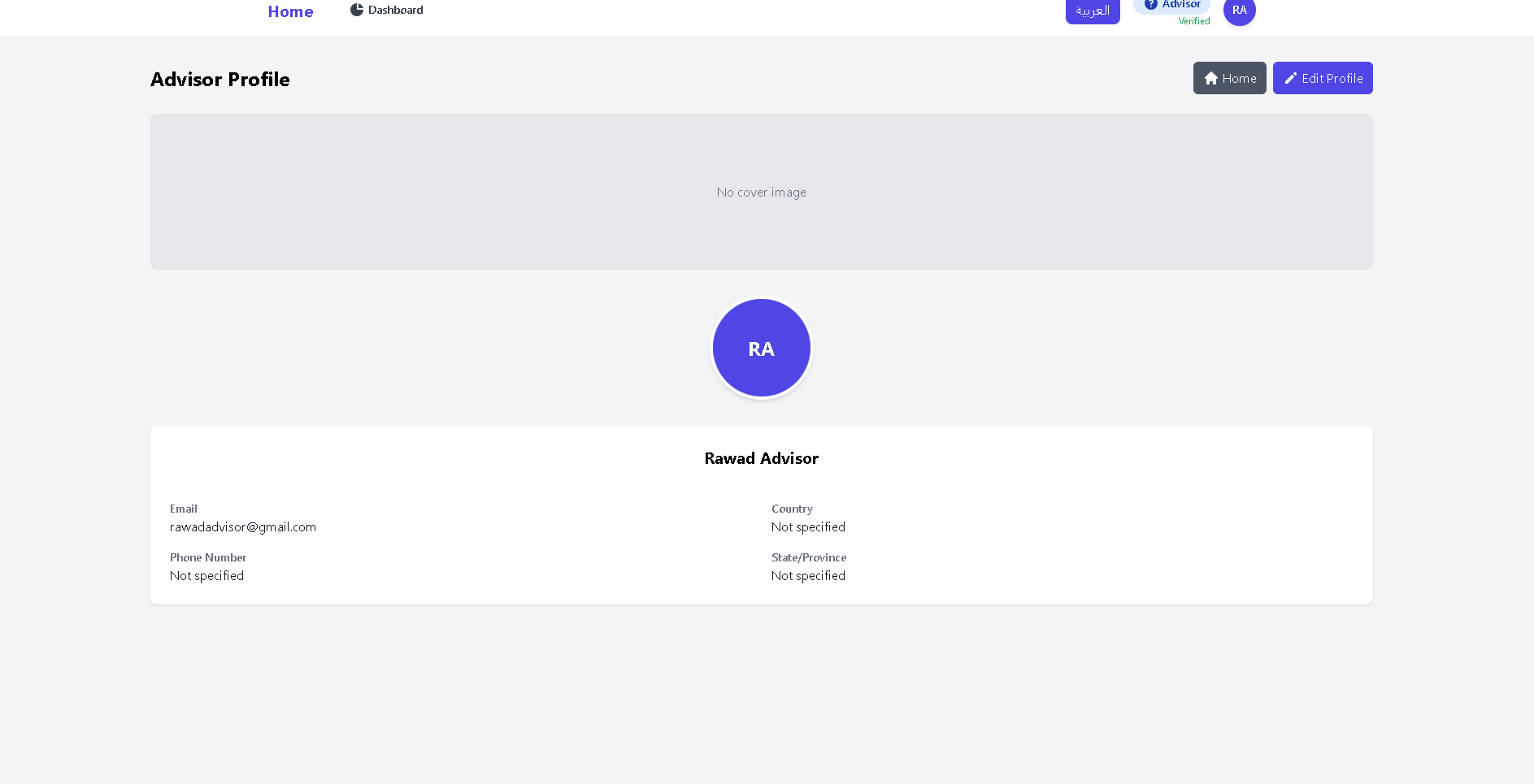
****

Figure 138 - Advisor Profile page

**Teacher dashboard interface**

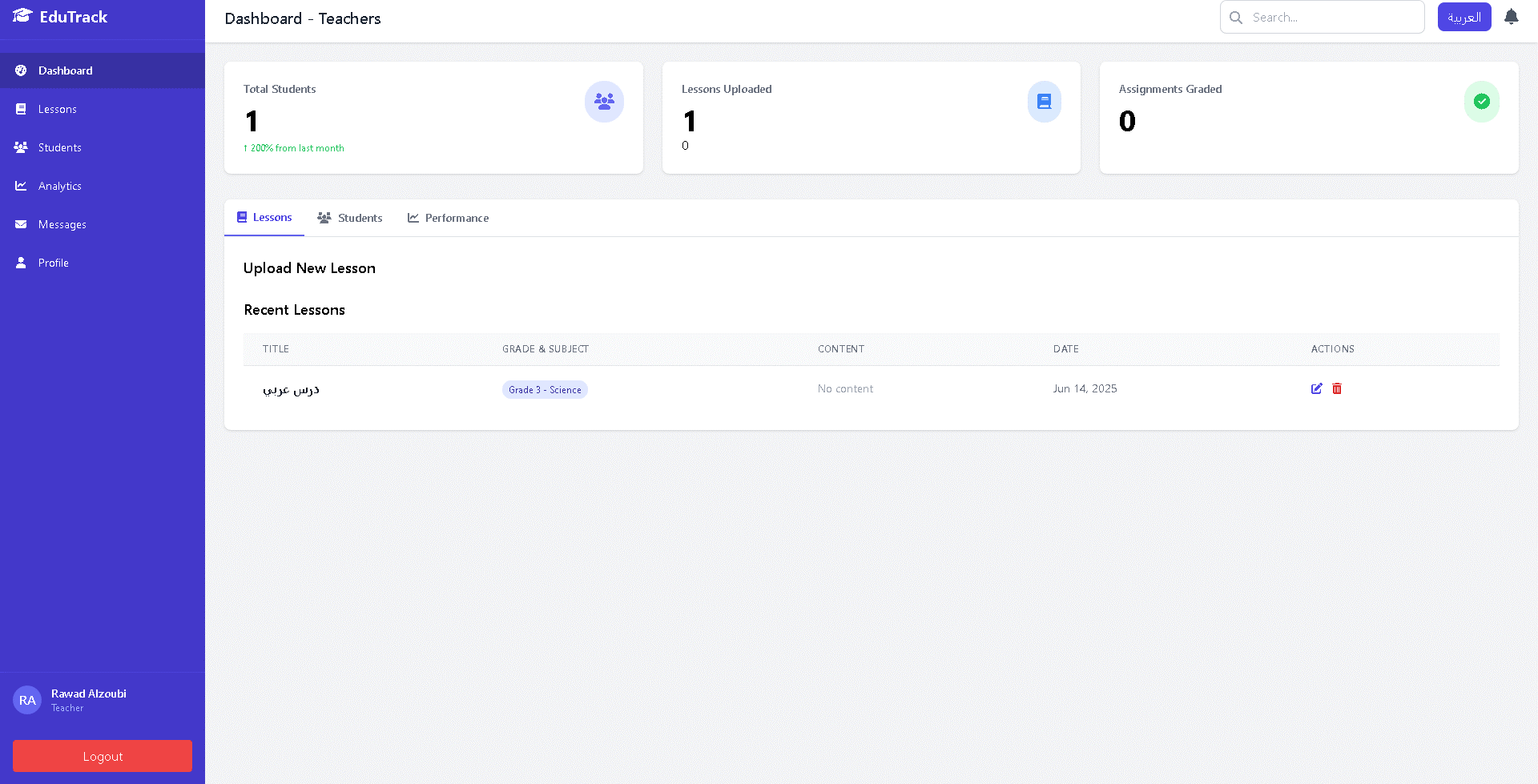


Figure 139 - Teacher Dashboard Page

**Lesson Management interface**

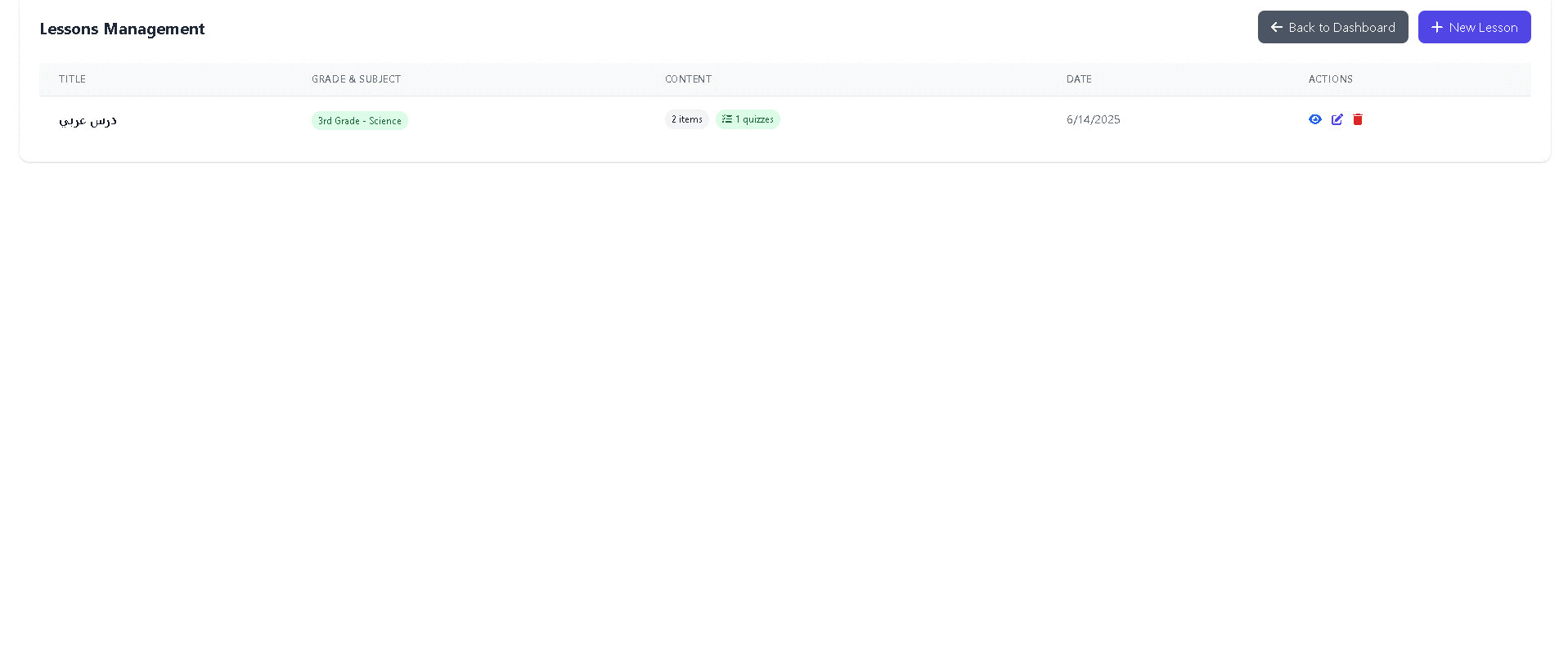


Figure 140 - Lesson Management page

**Add new lesson interface**

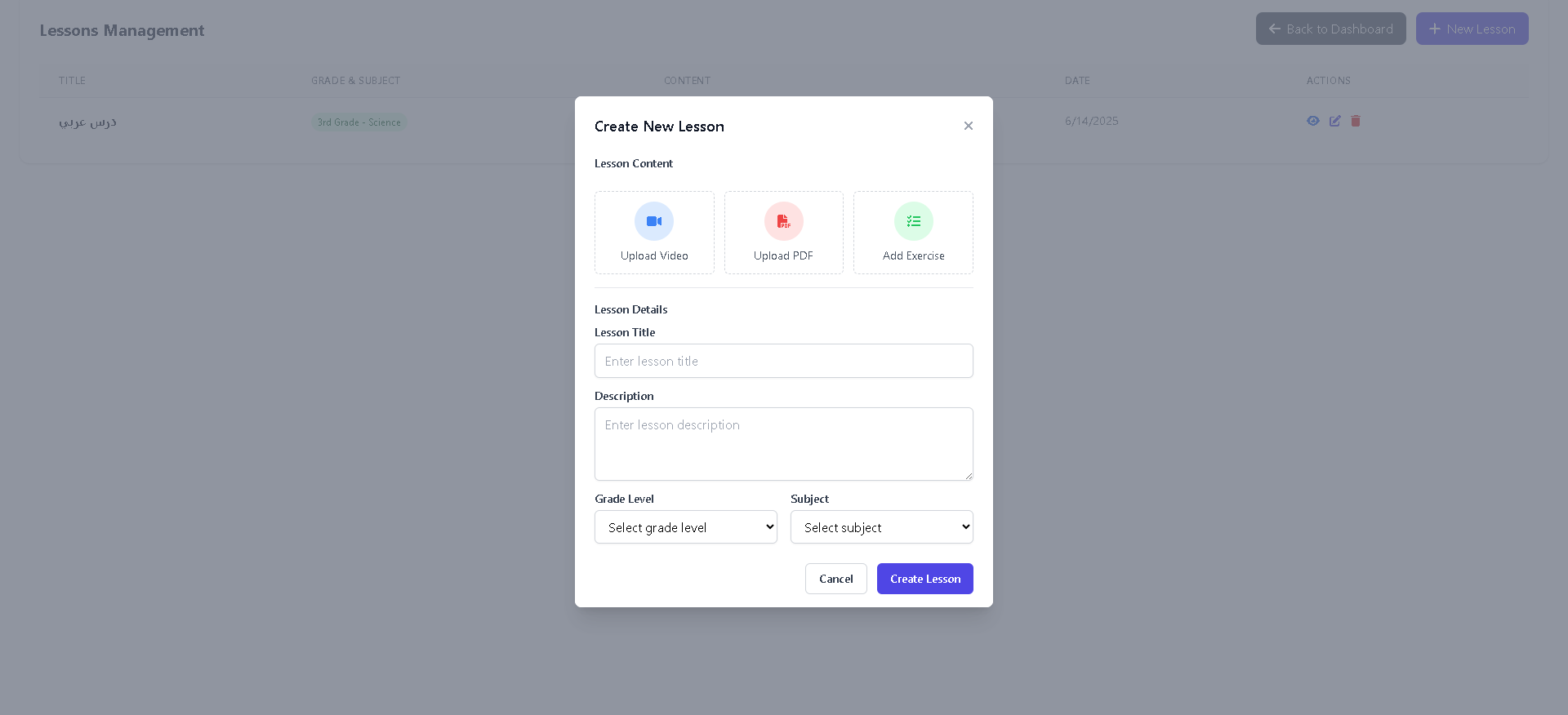
****

Figure 141 - Add new lesson page

**Update lesson interface**

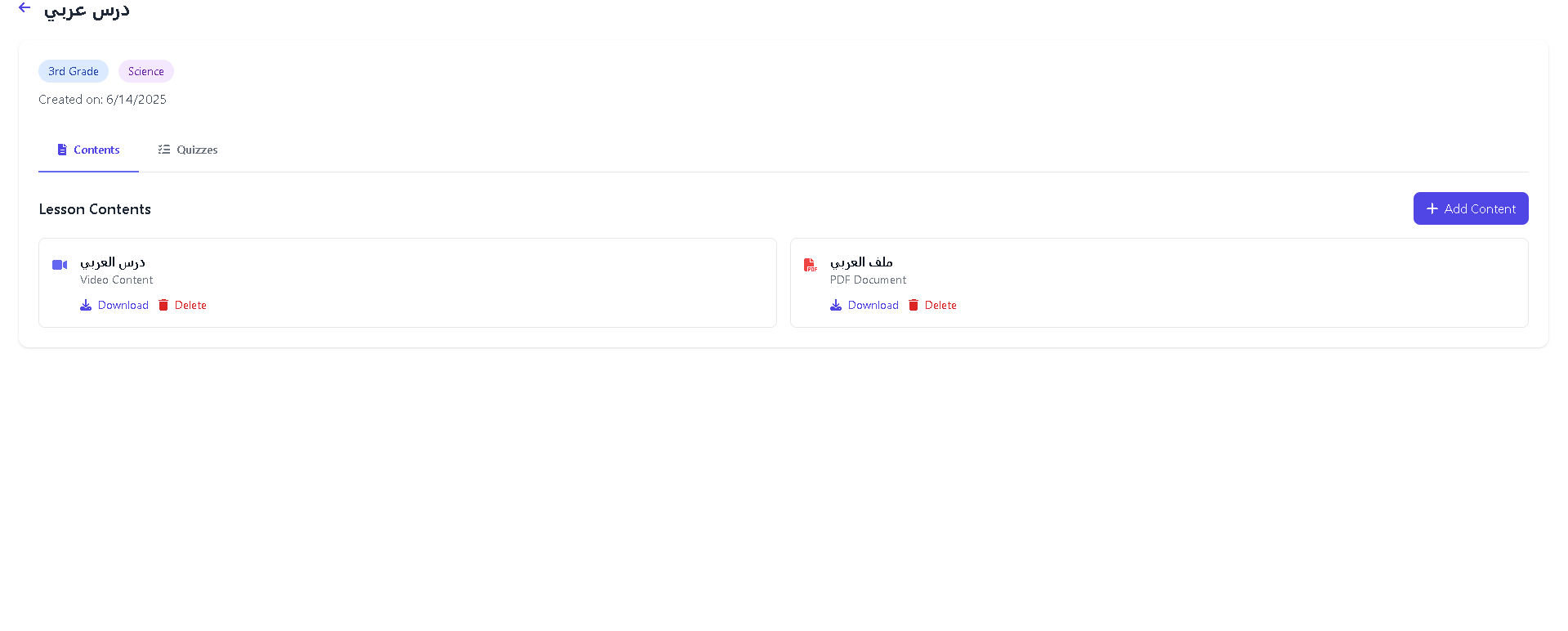
****

Figure 142 - Update lesson Page

**Create Quiz interface**

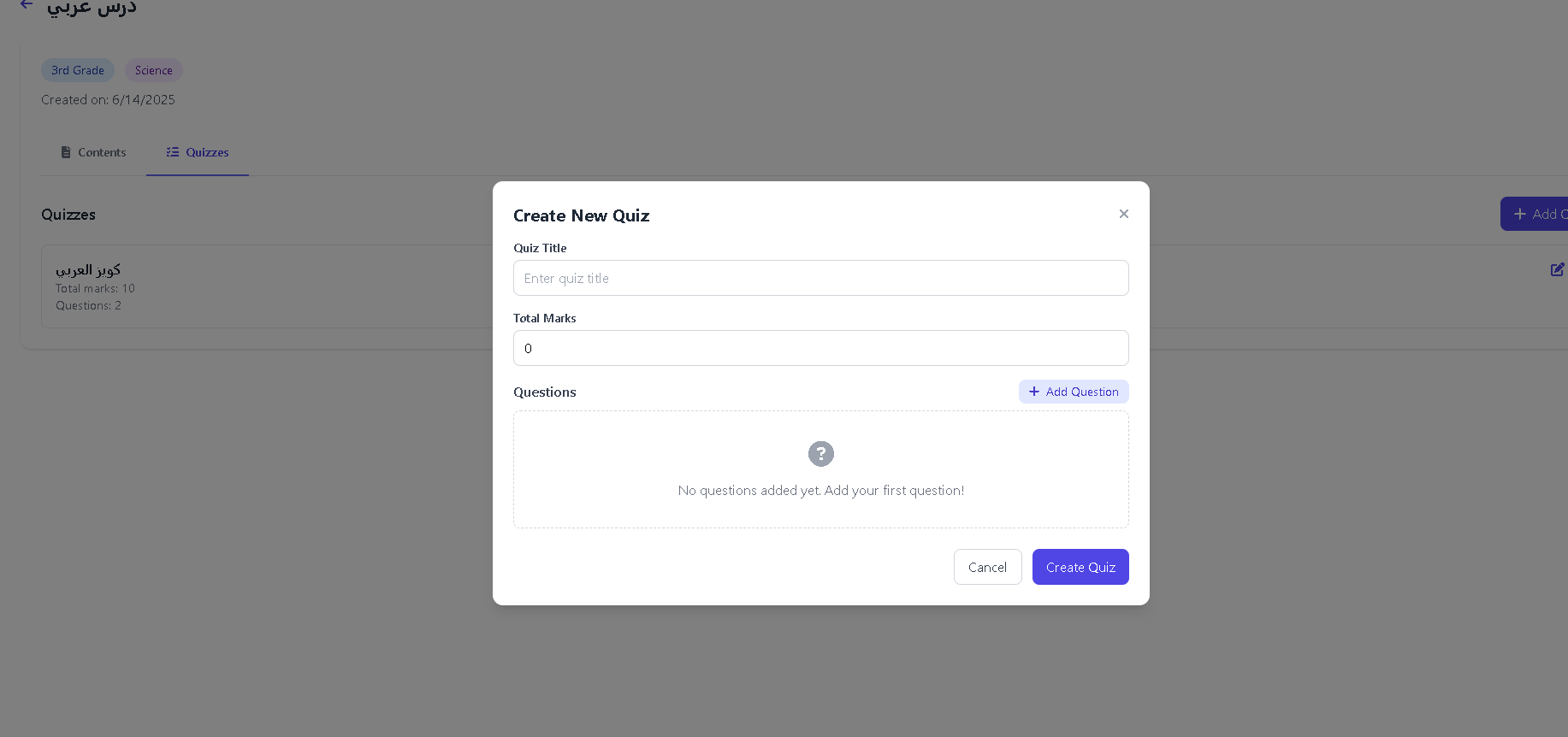
****

Figure 143 - Create Quiz page

**Add Question interface**

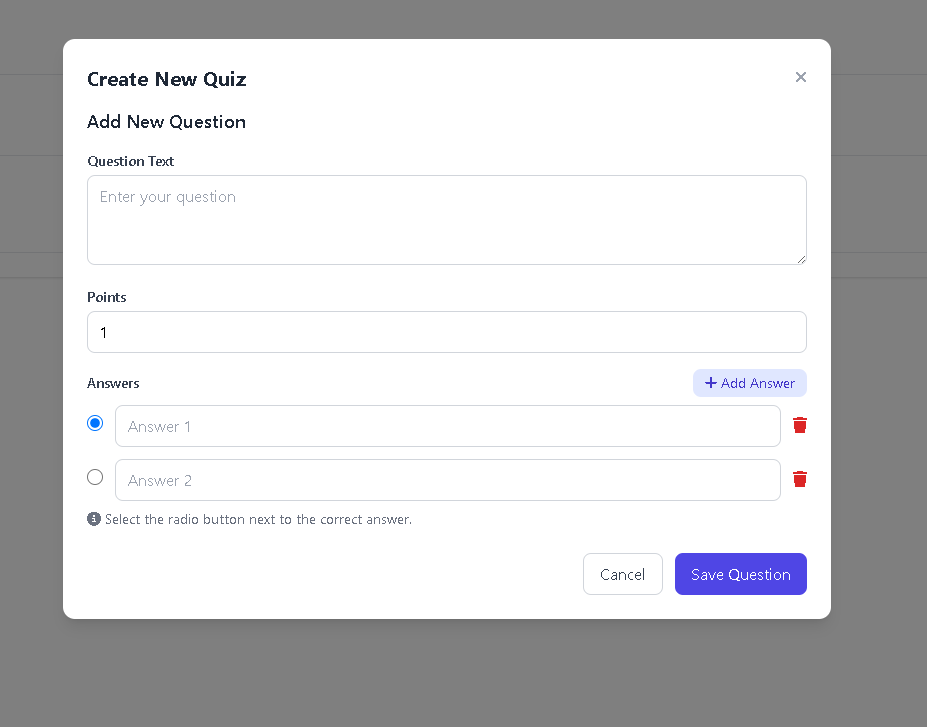
****

Figure 144 - Add Question page

**Student performance interface (Teacher)**

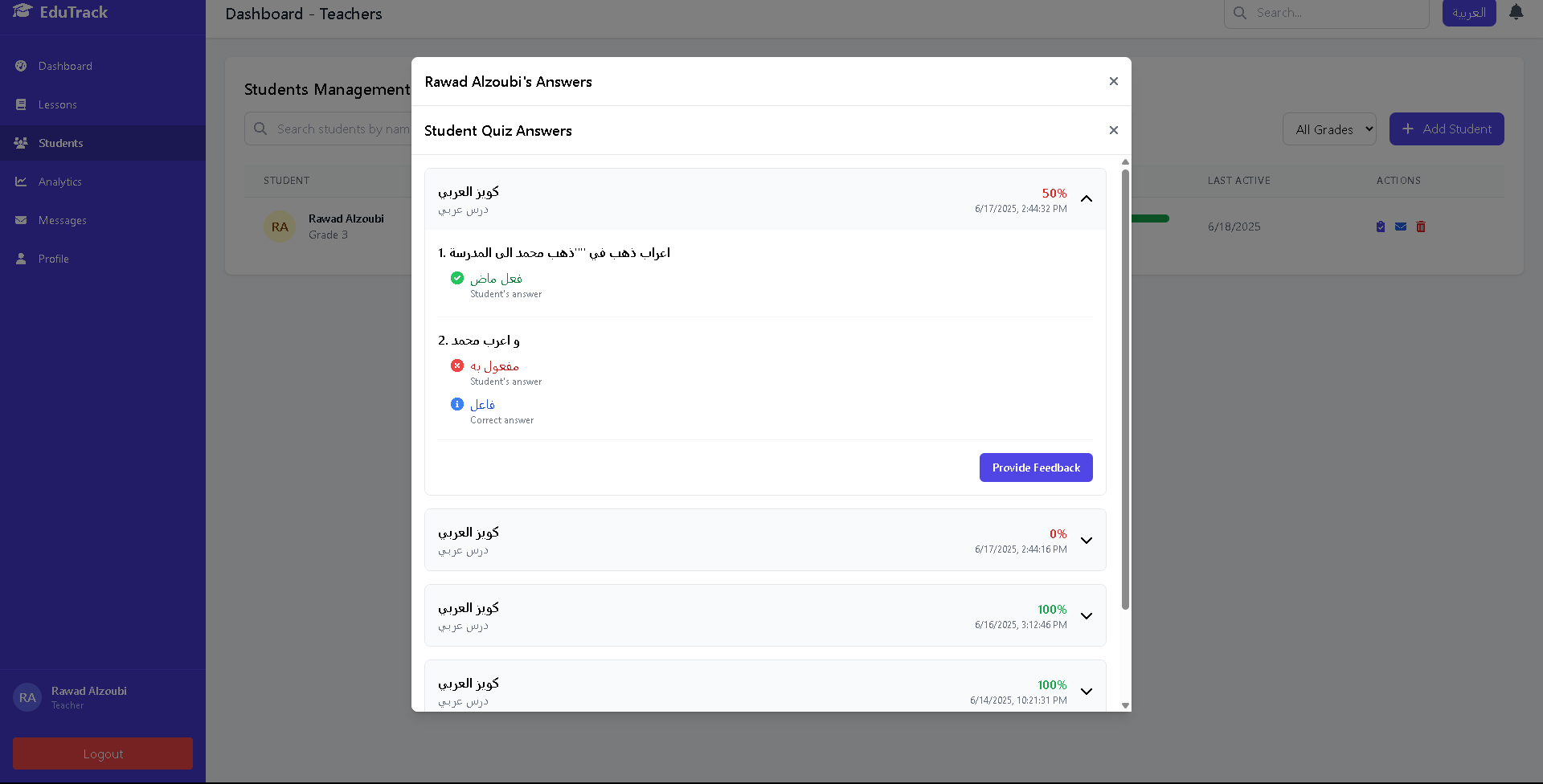
****

Figure 145 - Student performance (Teacher)

**Submit Feedback interface**

****

Figure 146 - Submit Feedback Page

**Teacher profile interface**

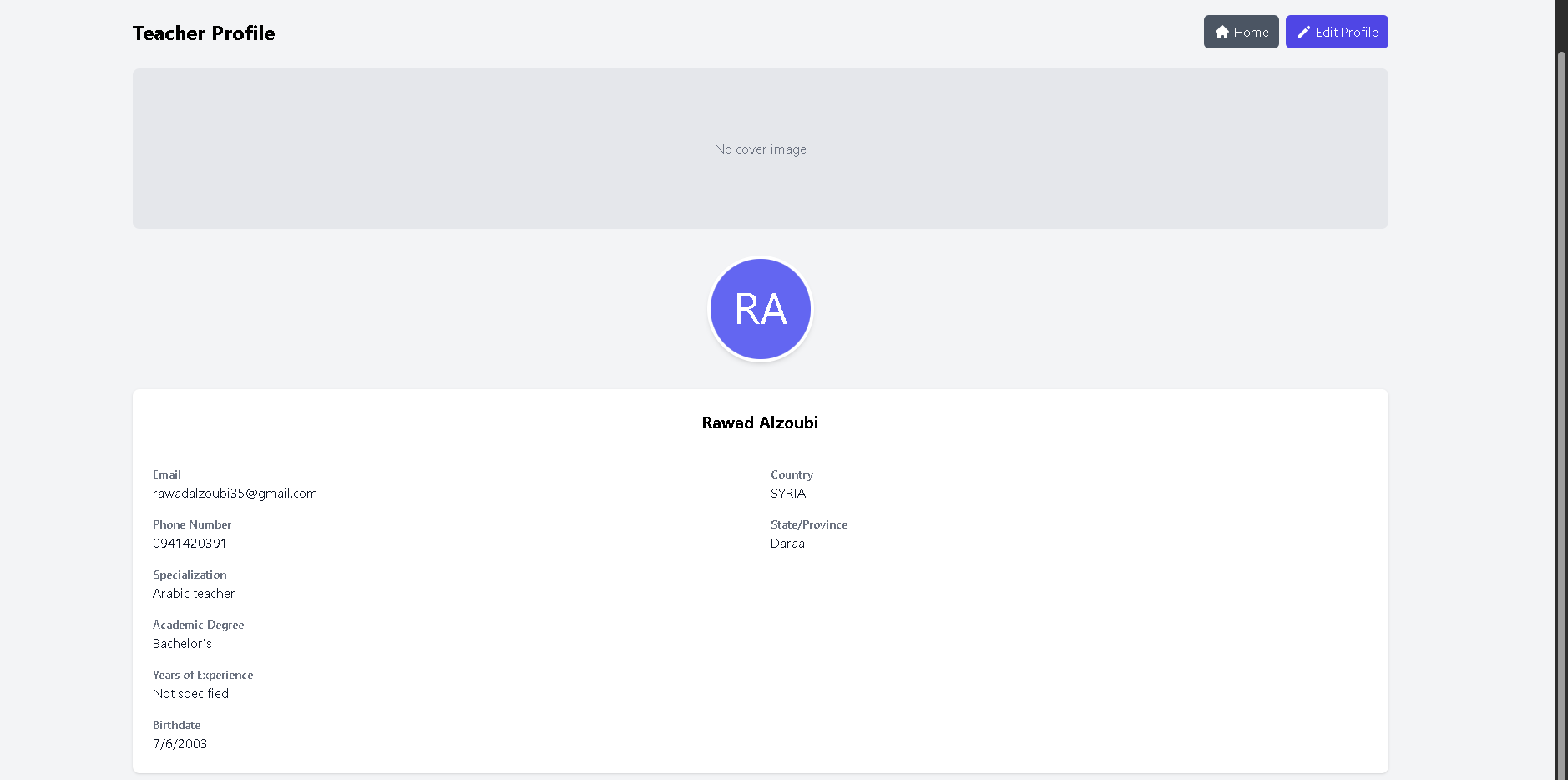
****

Figure 147 - Teacher profile page

**Student dashboard**

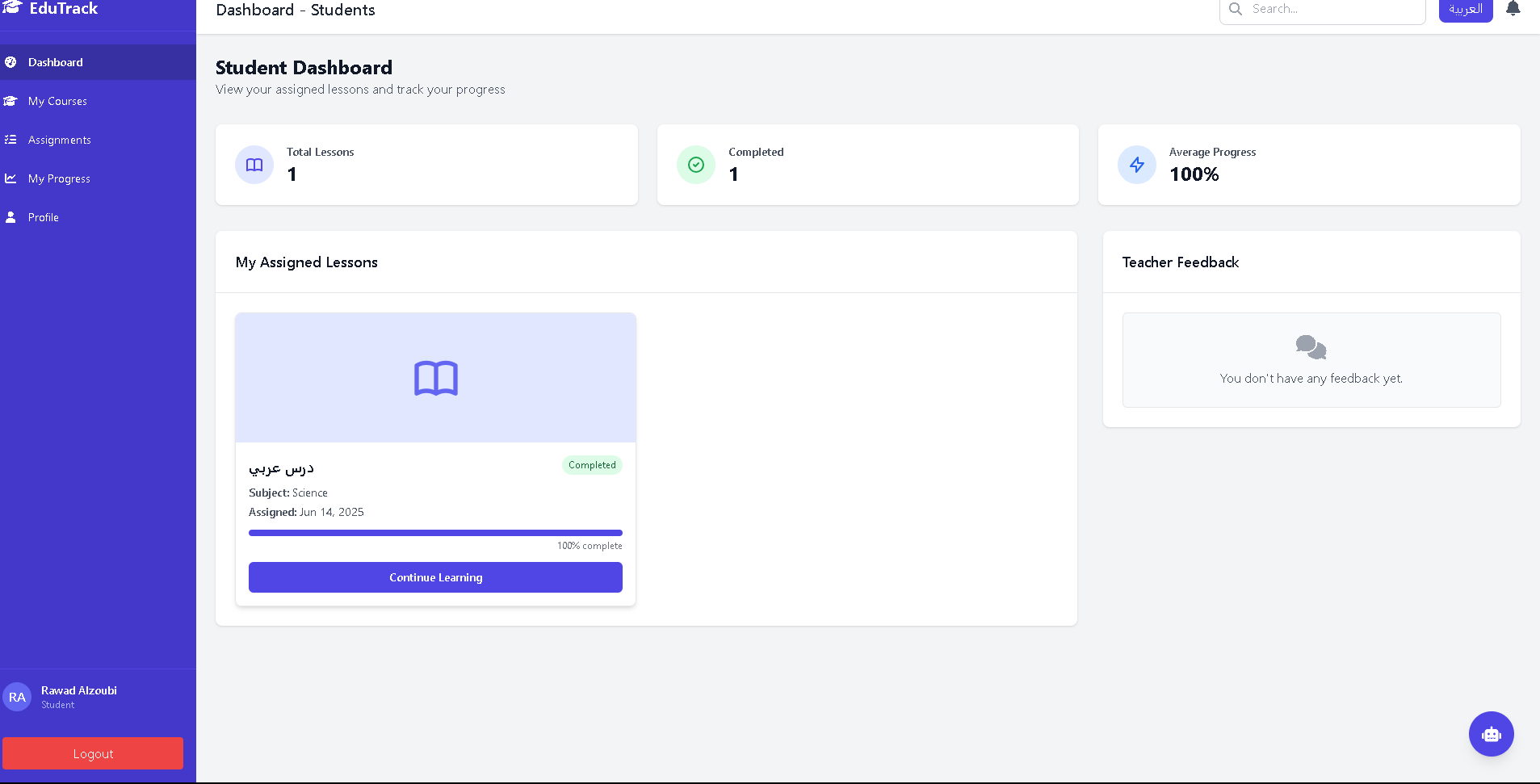
****

Figure 148 - Student dashboard page

**View lesson interface**

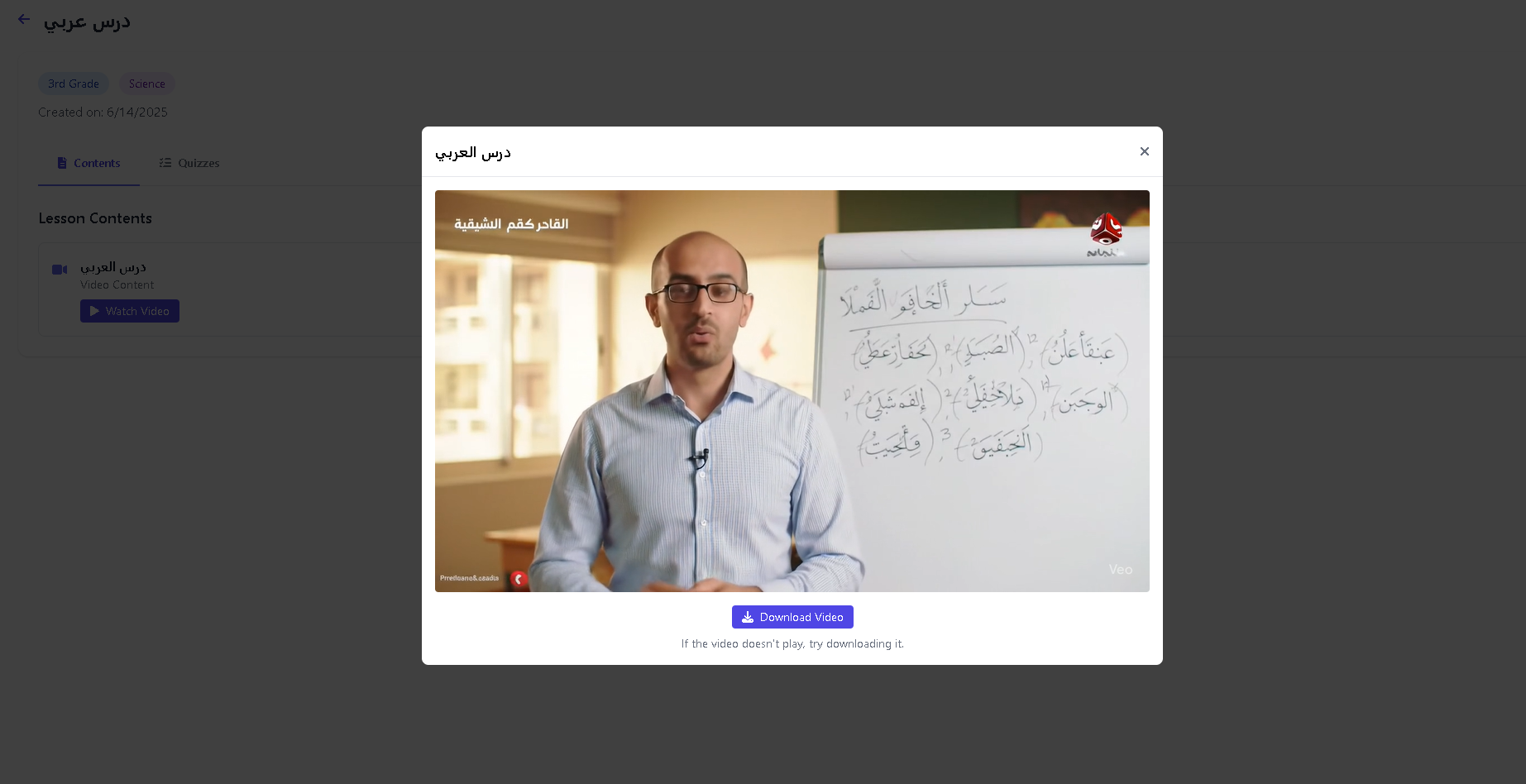
****

Figure 149 - View lesson Page

**Solve quiz interface**

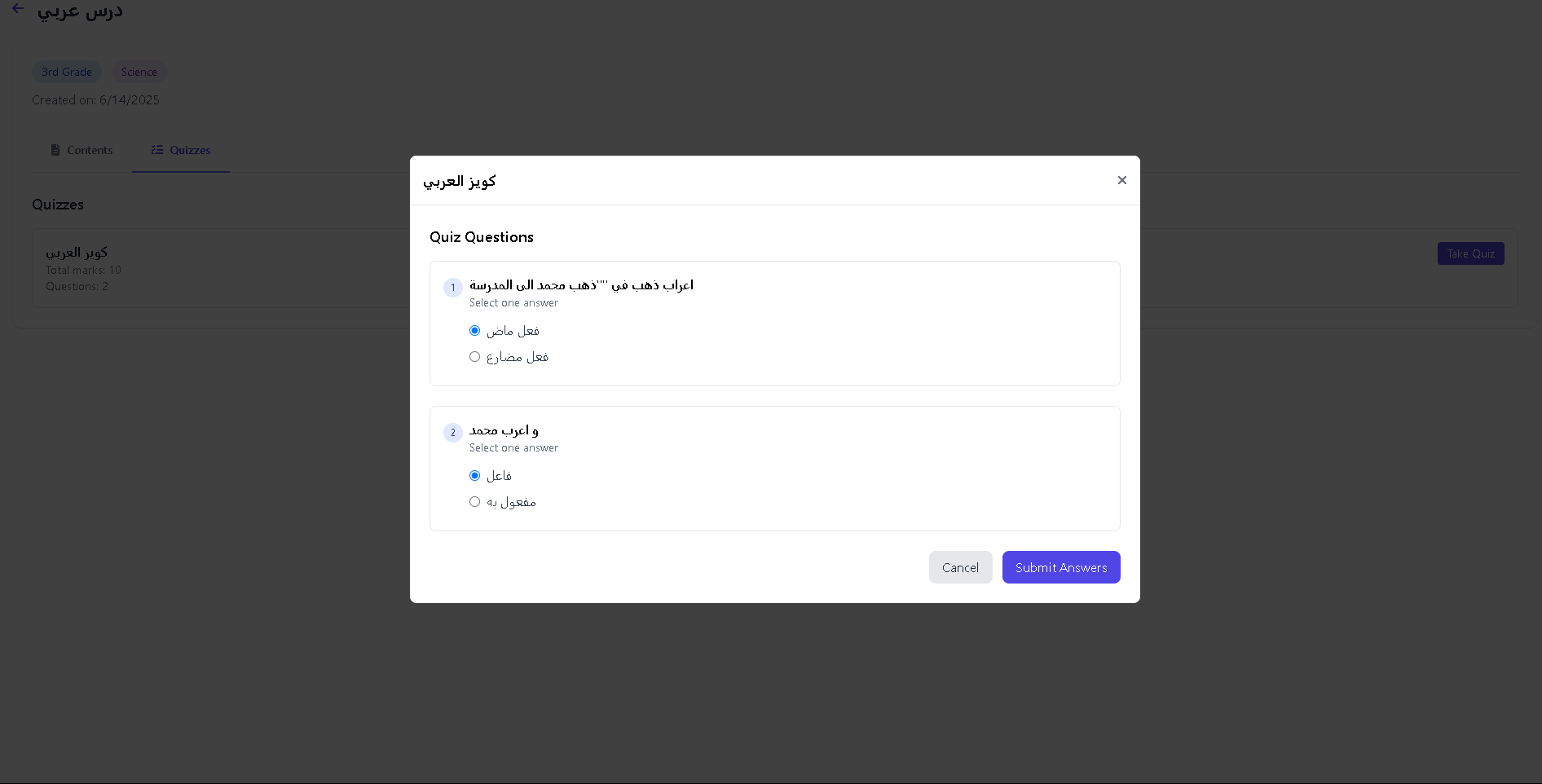
****

Figure 150 - Solve quiz page

**Quiz result interface**

****

Figure 151 - Quiz result Page

**Student profile**

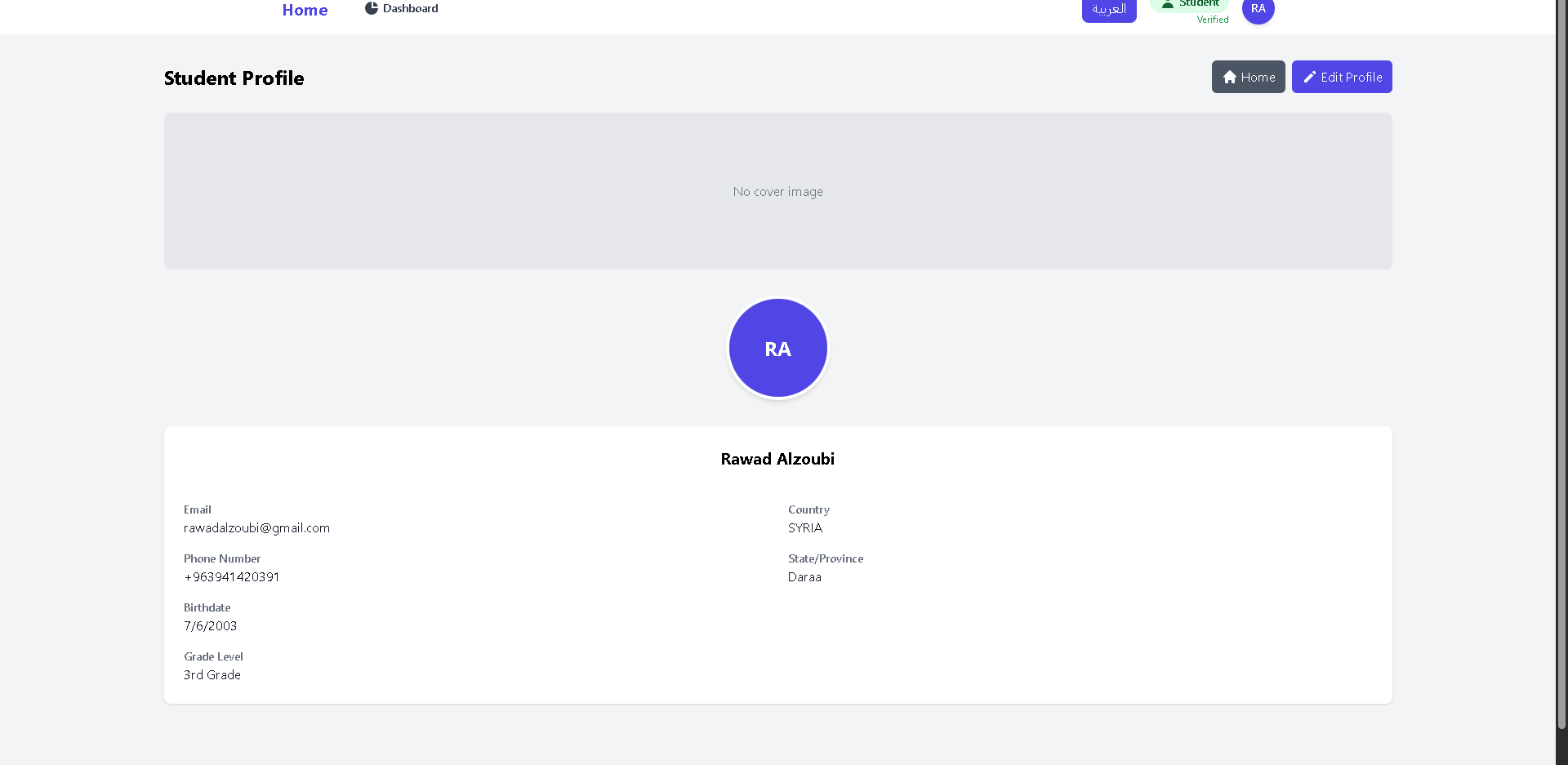
****

Figure 152 - Quiz result Page

**Create Live session**

****

Figure 153 - Create Live session Page

**Created sessions**

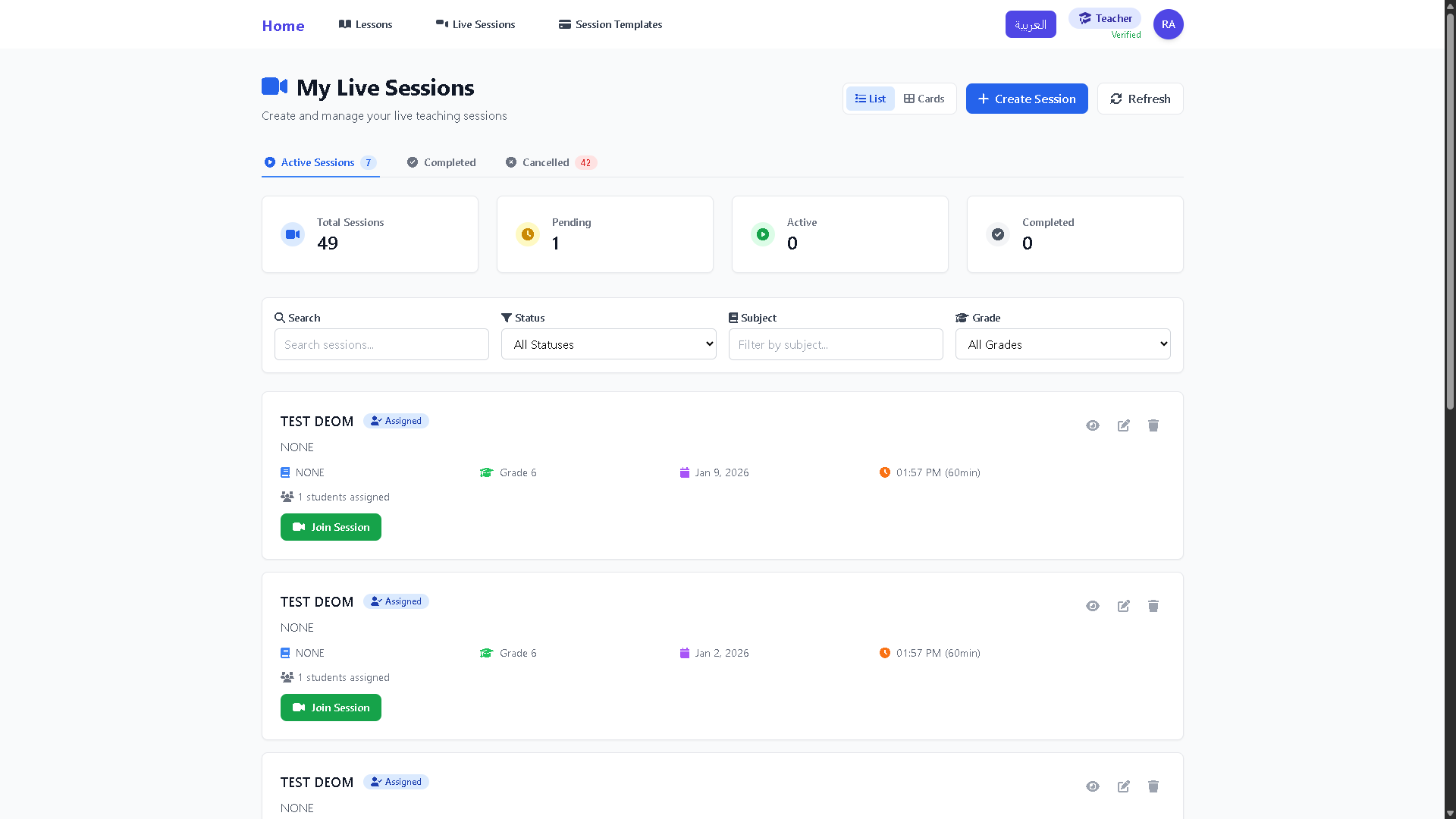
****

Figure 154 - created Sessions

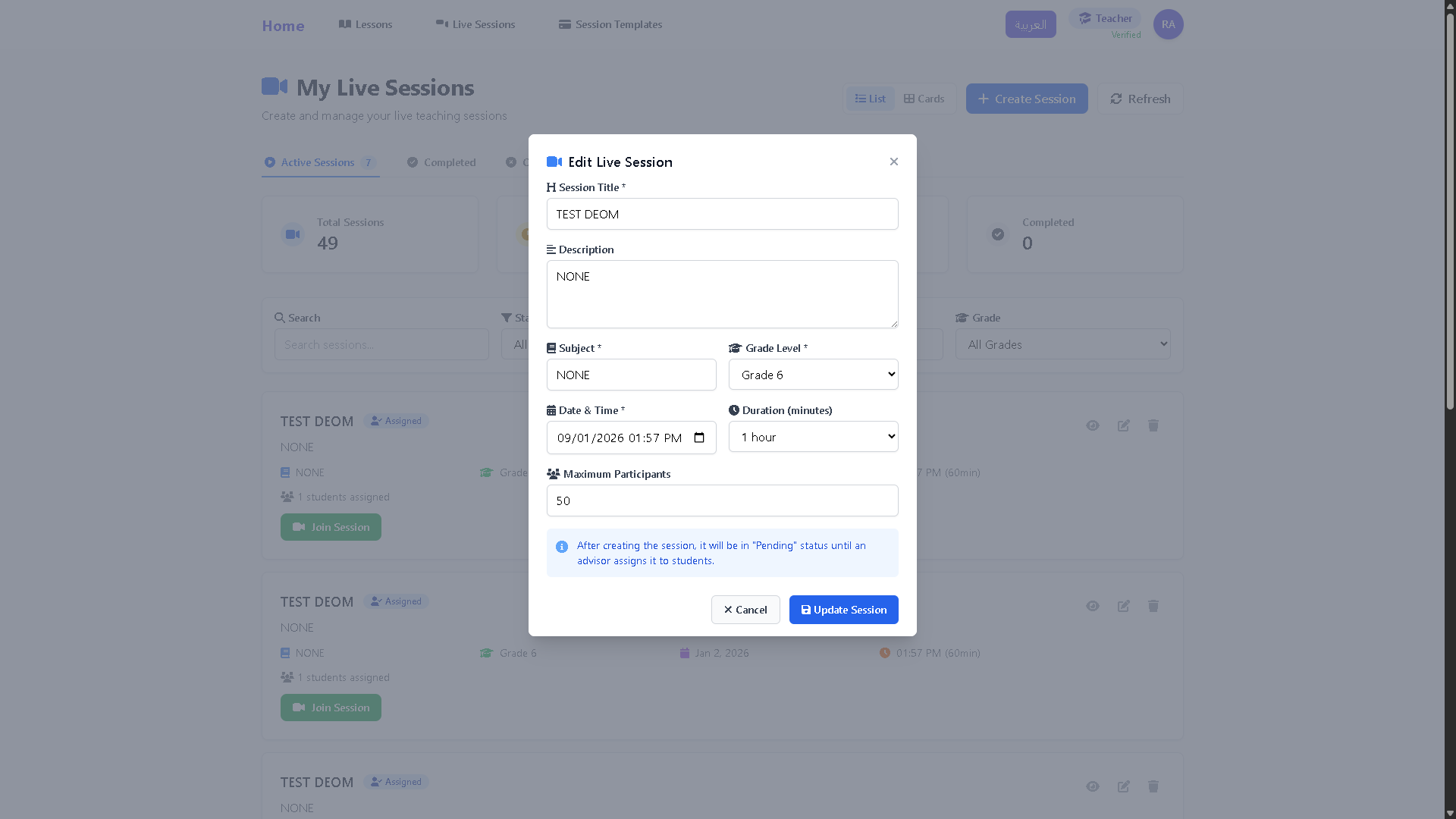
**Edit Live Session  
**

Figure 155 - Edit Live Session page

**View live Session info**

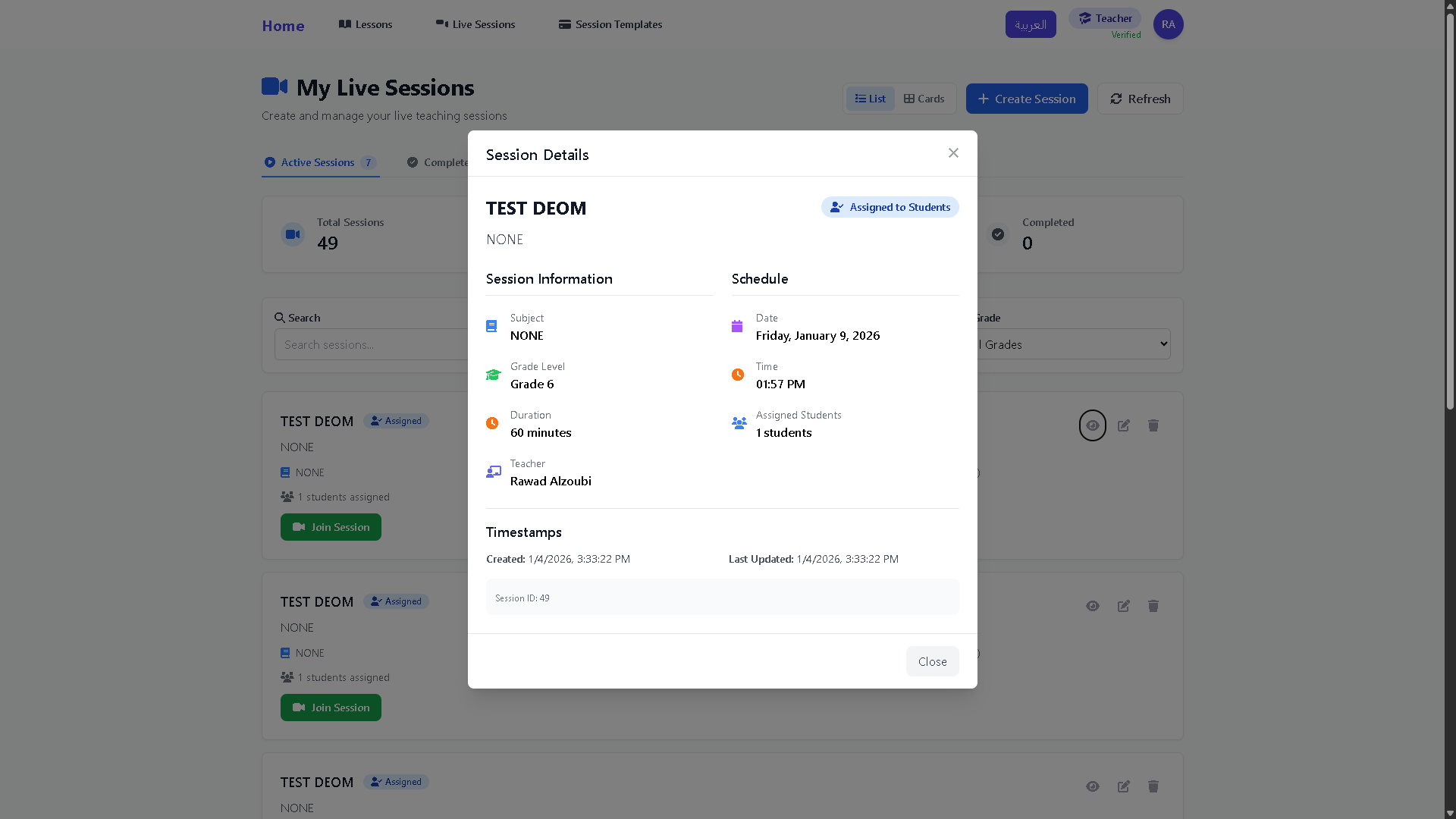
****

Figure 156 - View live Session info page

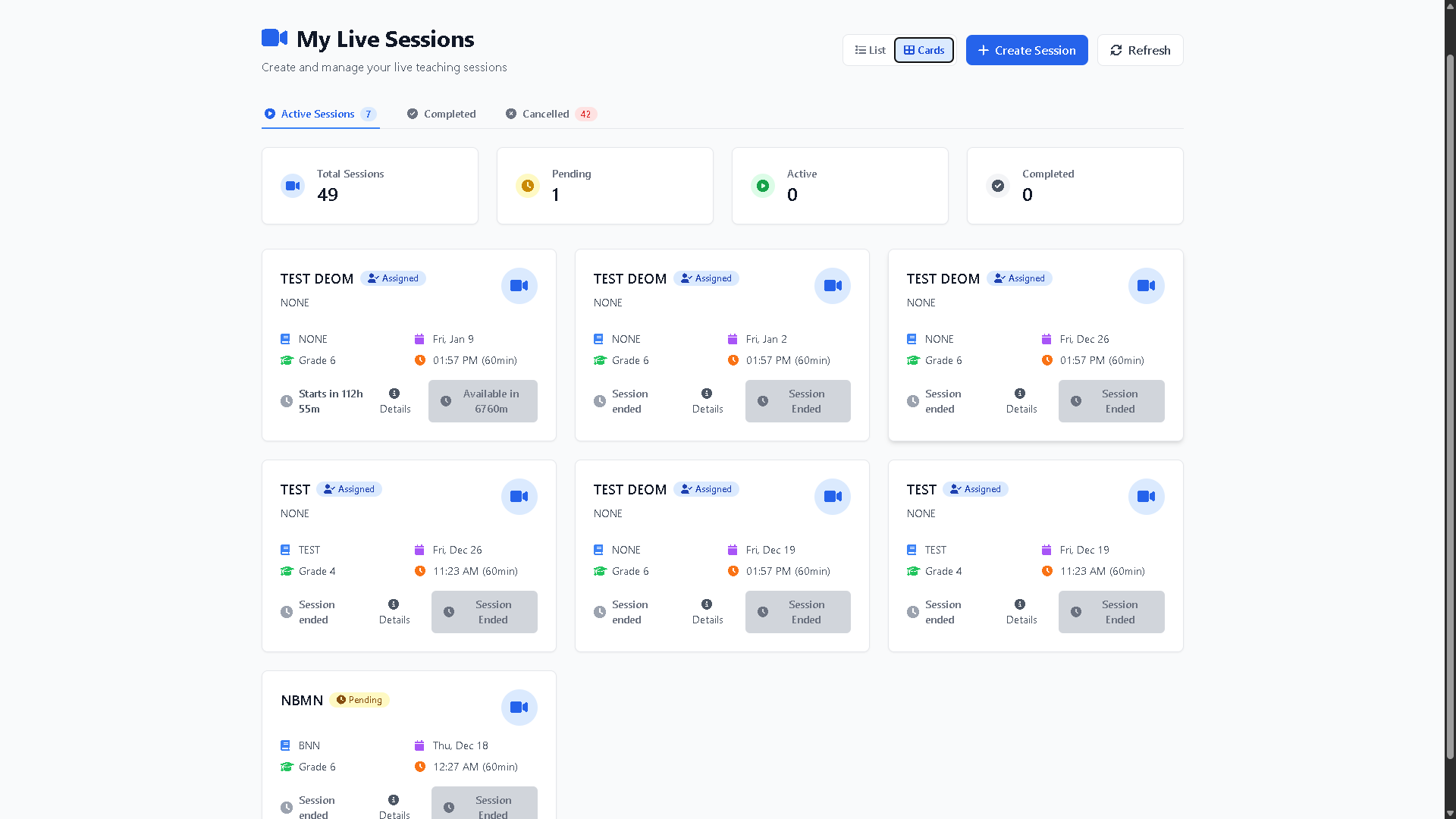
**View Sessions as a Card  
**

Figure 157 - View Sessions as a Card

**Create Session Templates**

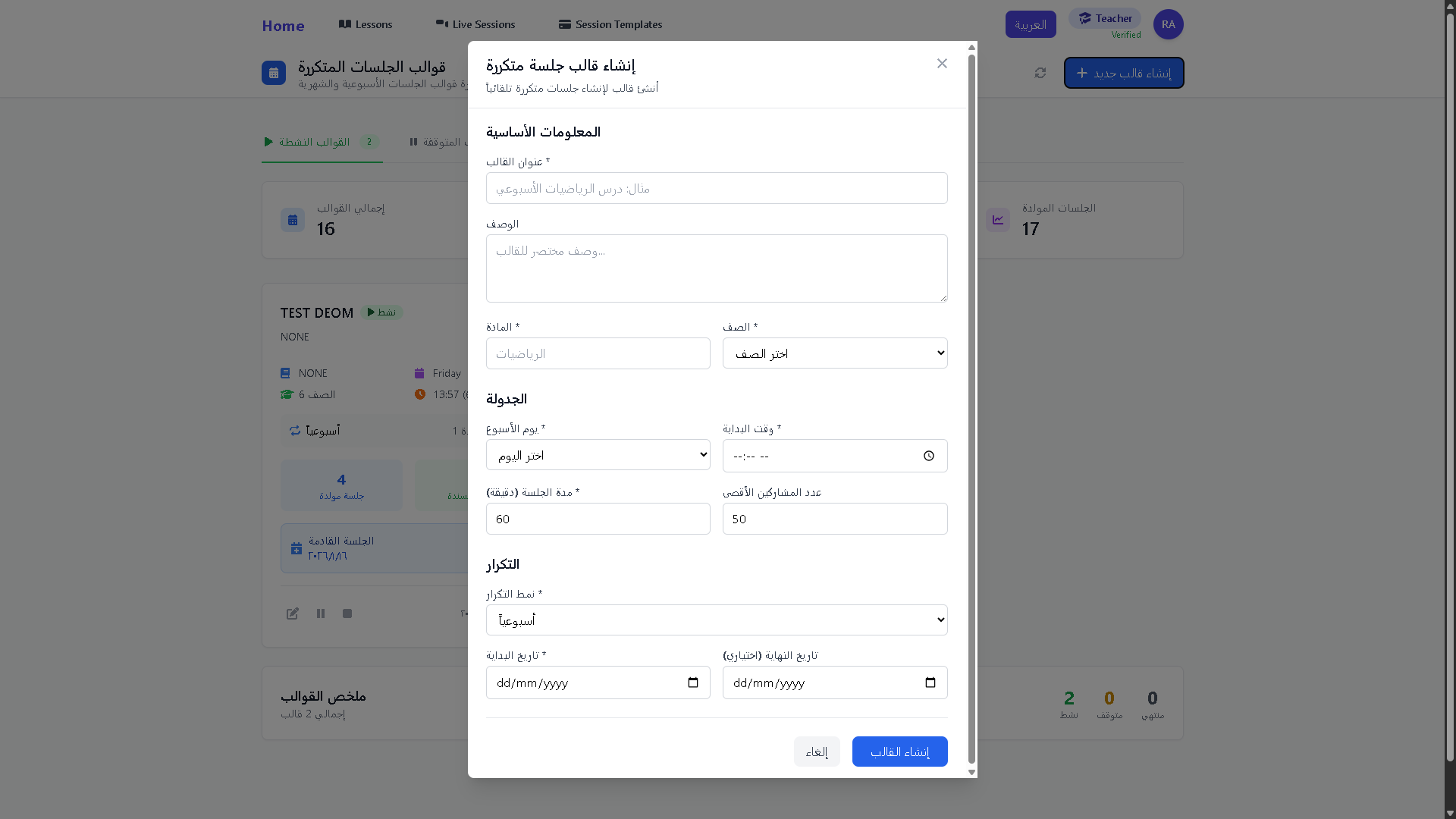
****

Figure 158 - Create Session Templates page

**Edit Session Template**

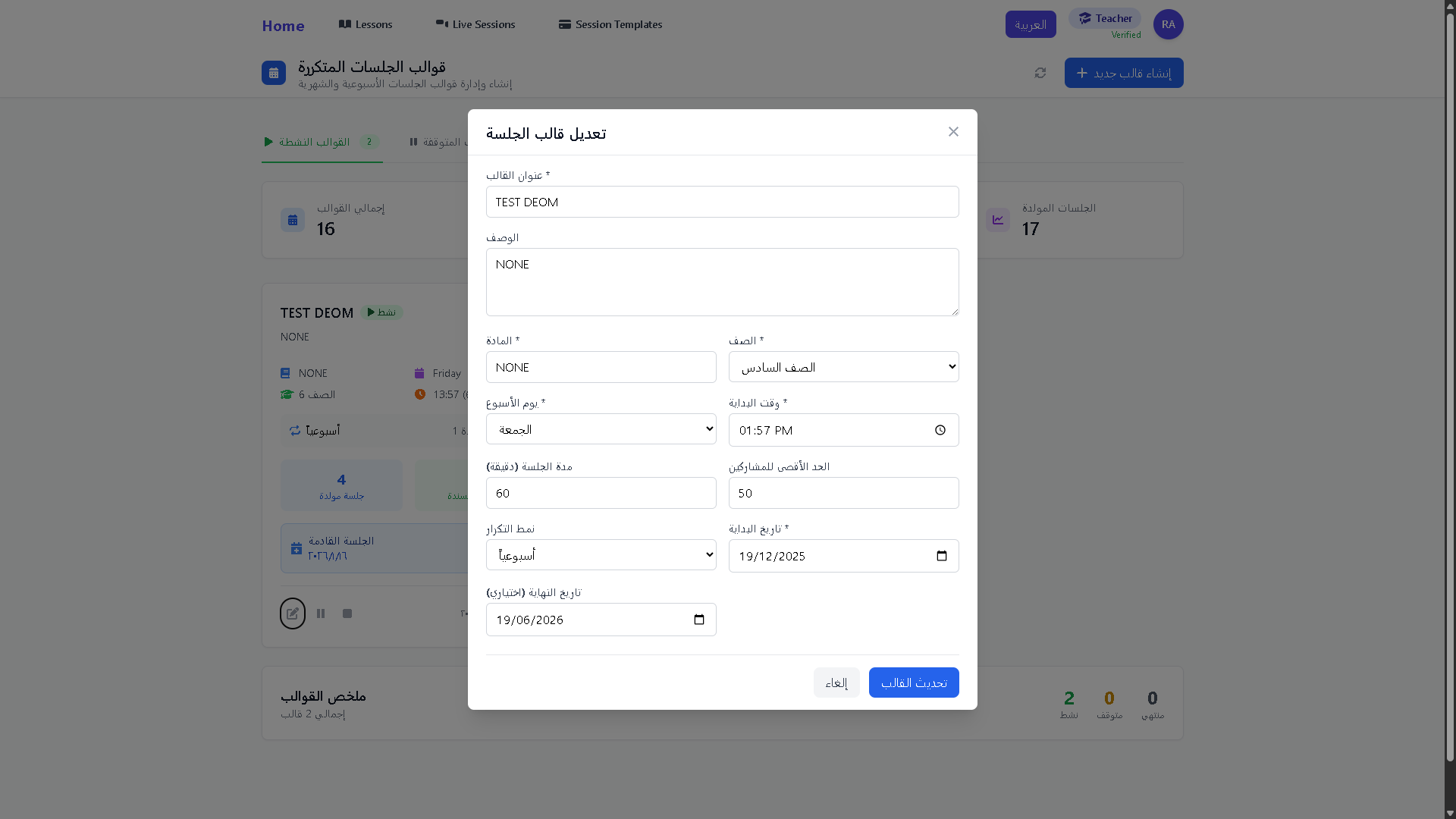
****

Figure 159 - Edit Session Template page

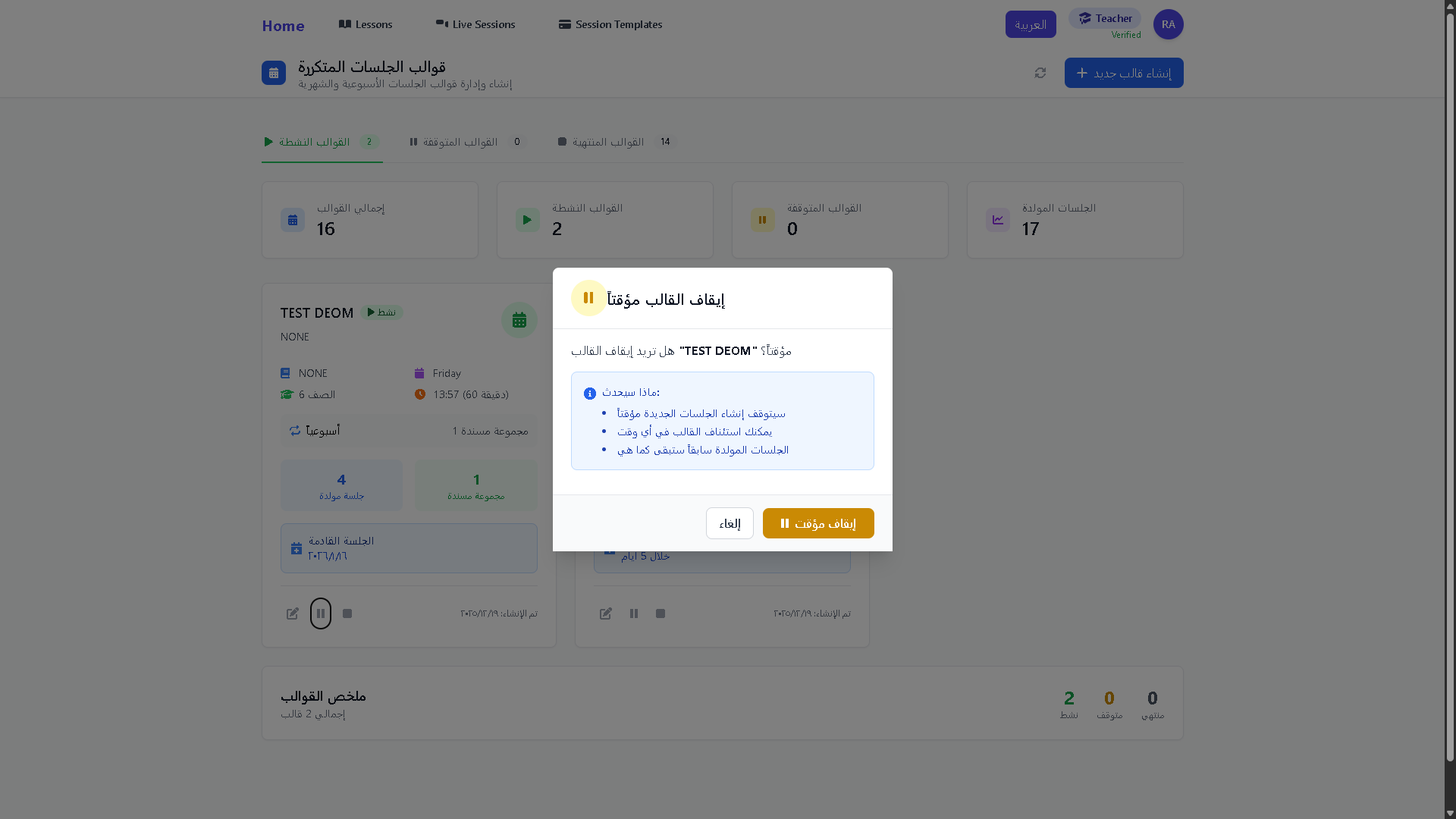
**Pause Templates  
**

Figure 160 - Pause Templates page

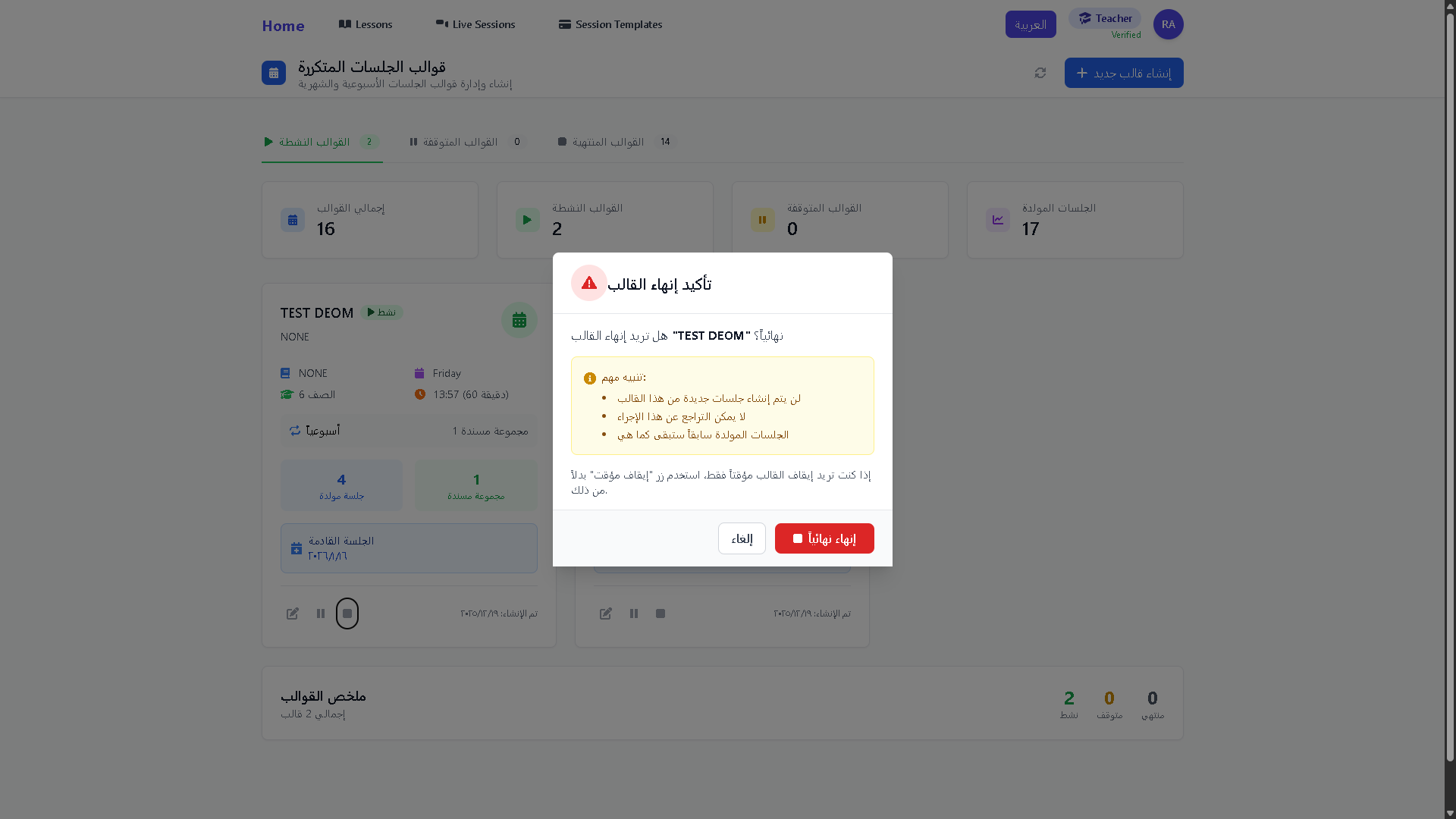
**End Templates  
**

Figure 161 - End Templates page

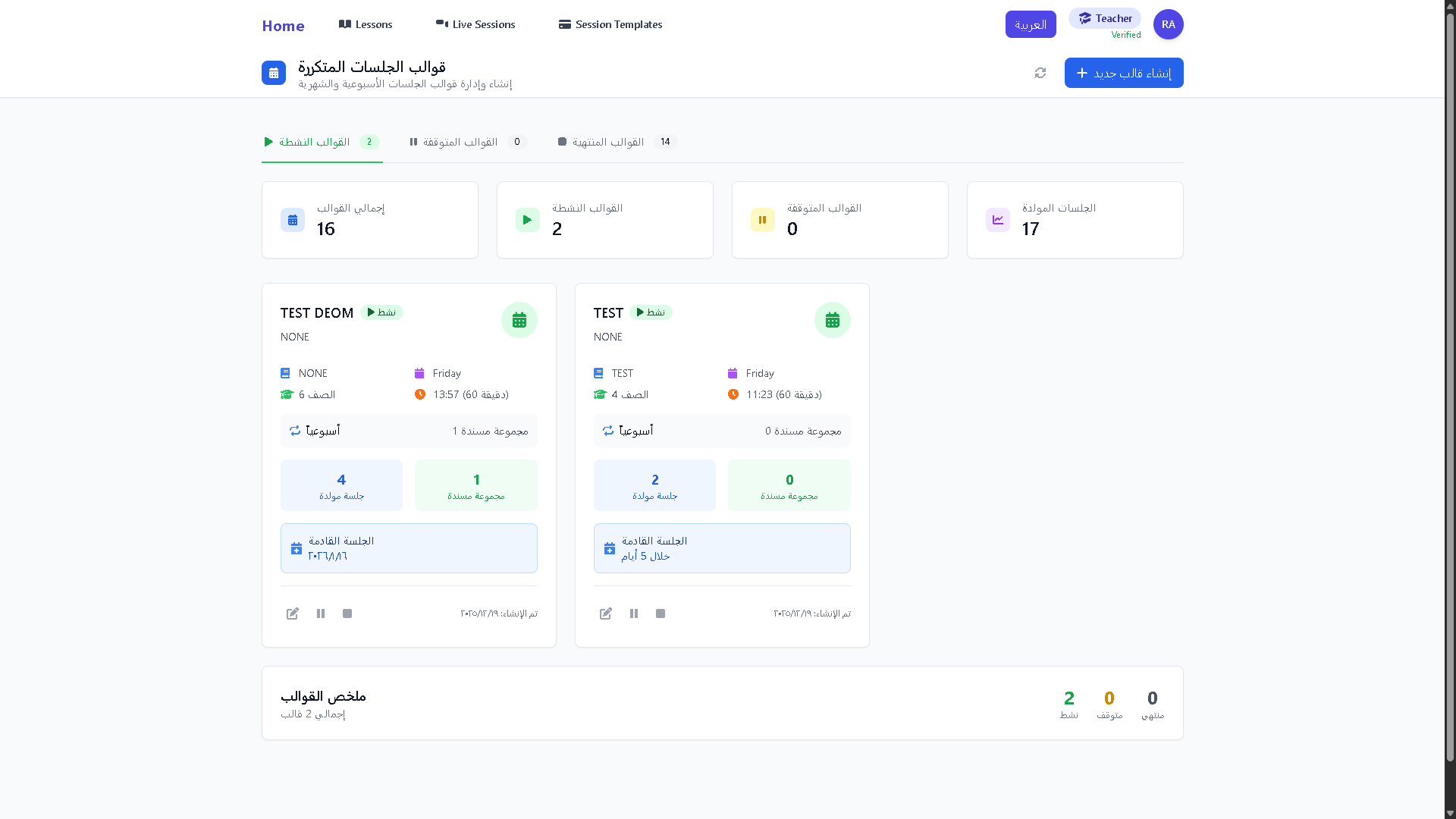
**View Templates  
**

Figure 162 - View Templates page

**Join Early**

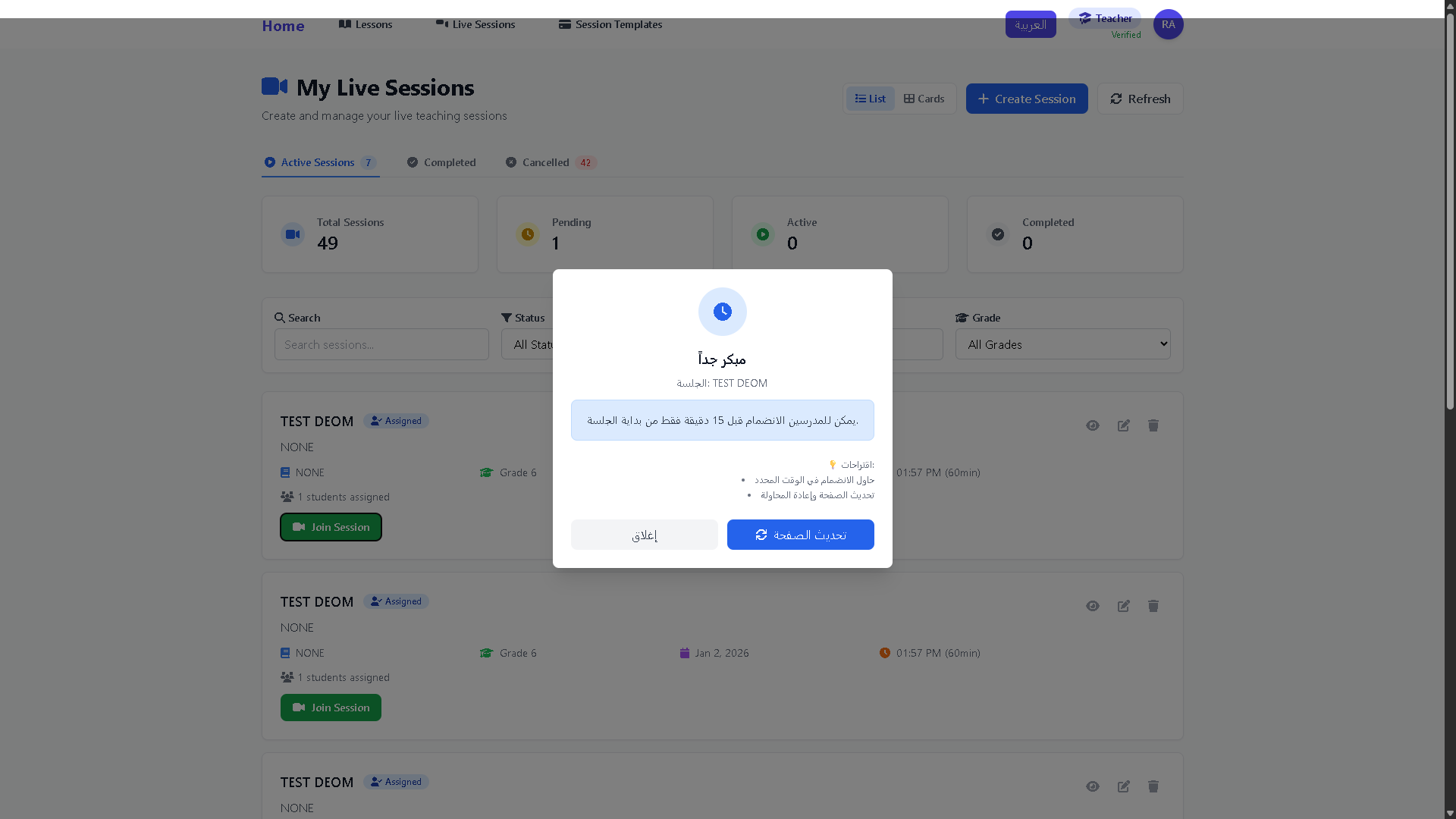
****

Figure 163 - Join Early page

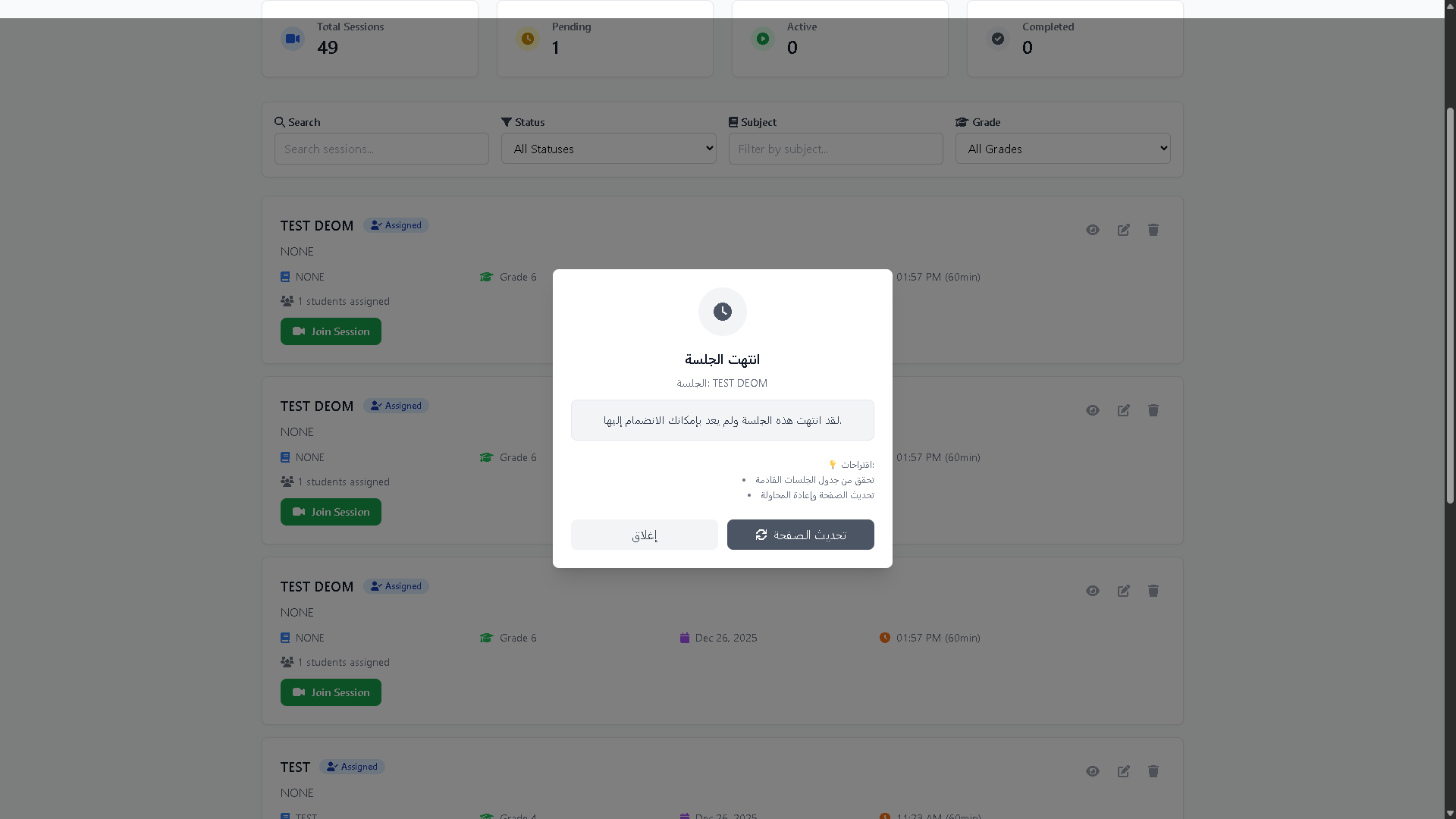
**Join Late  
**

Figure 164 - Join Late

**Join a Cancelled Session**

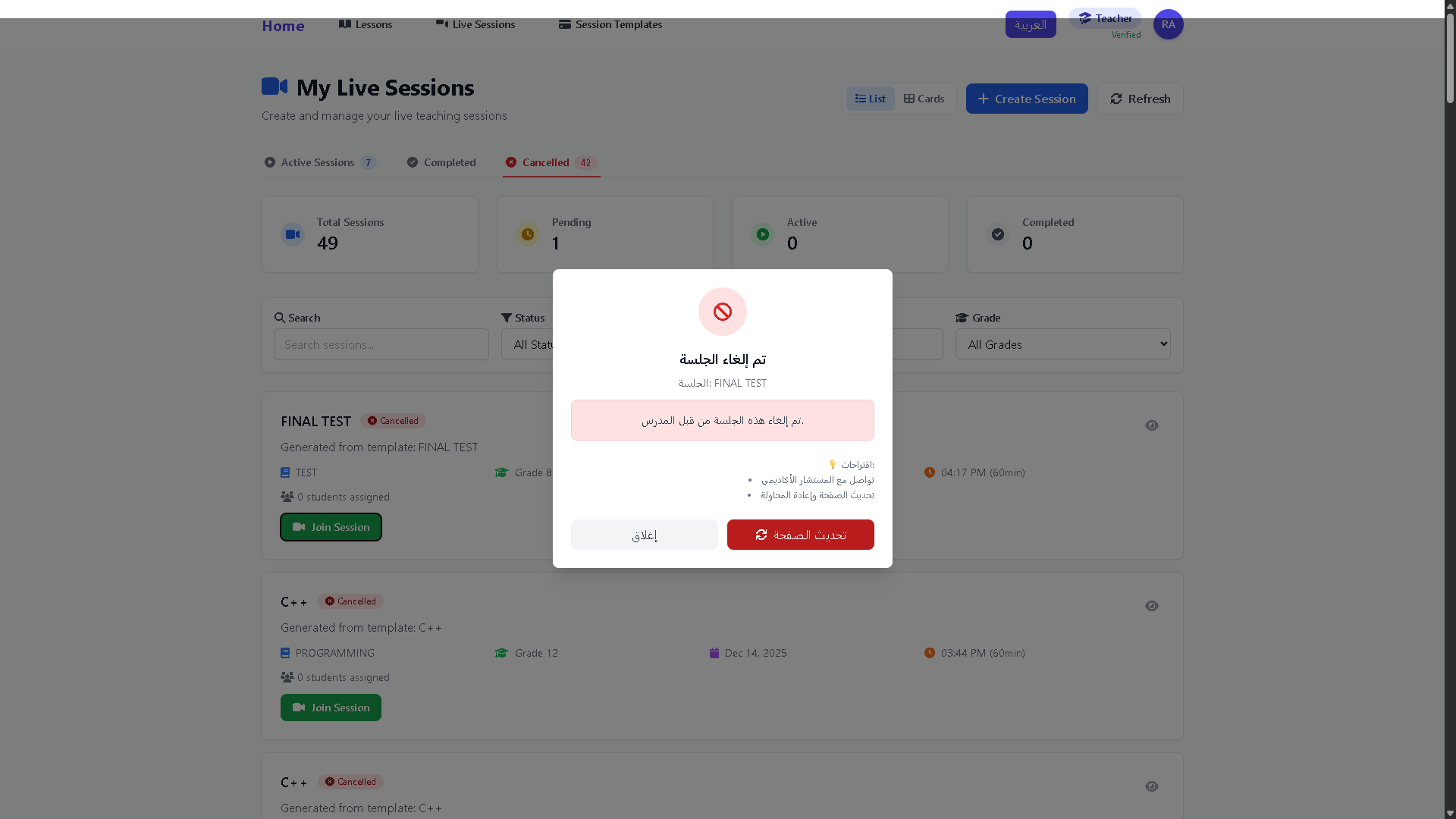
****

Figure 165 - Join a Cancelled Session page

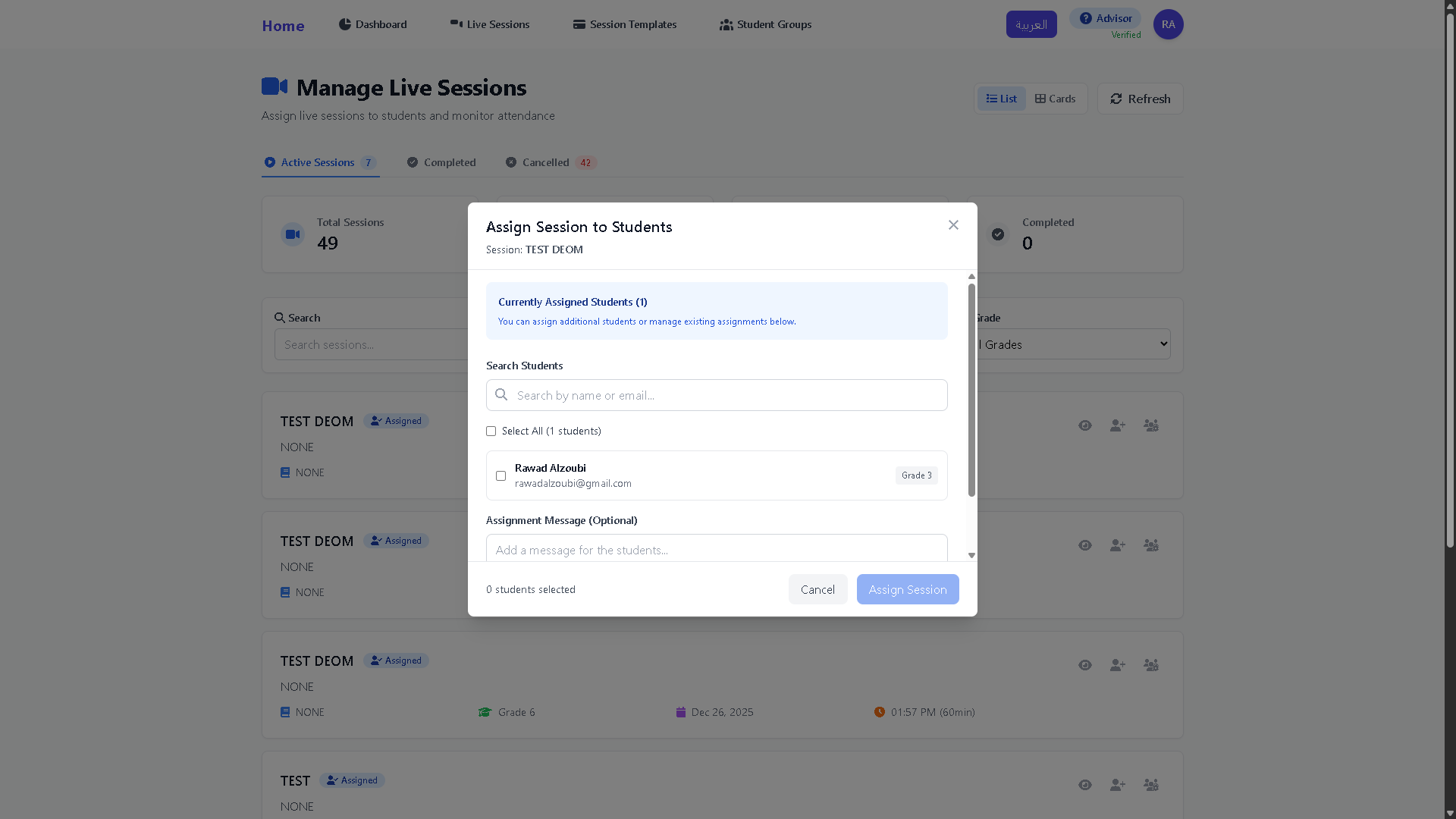
**Assign Session to student**

Figure 166 - Assign Session to student

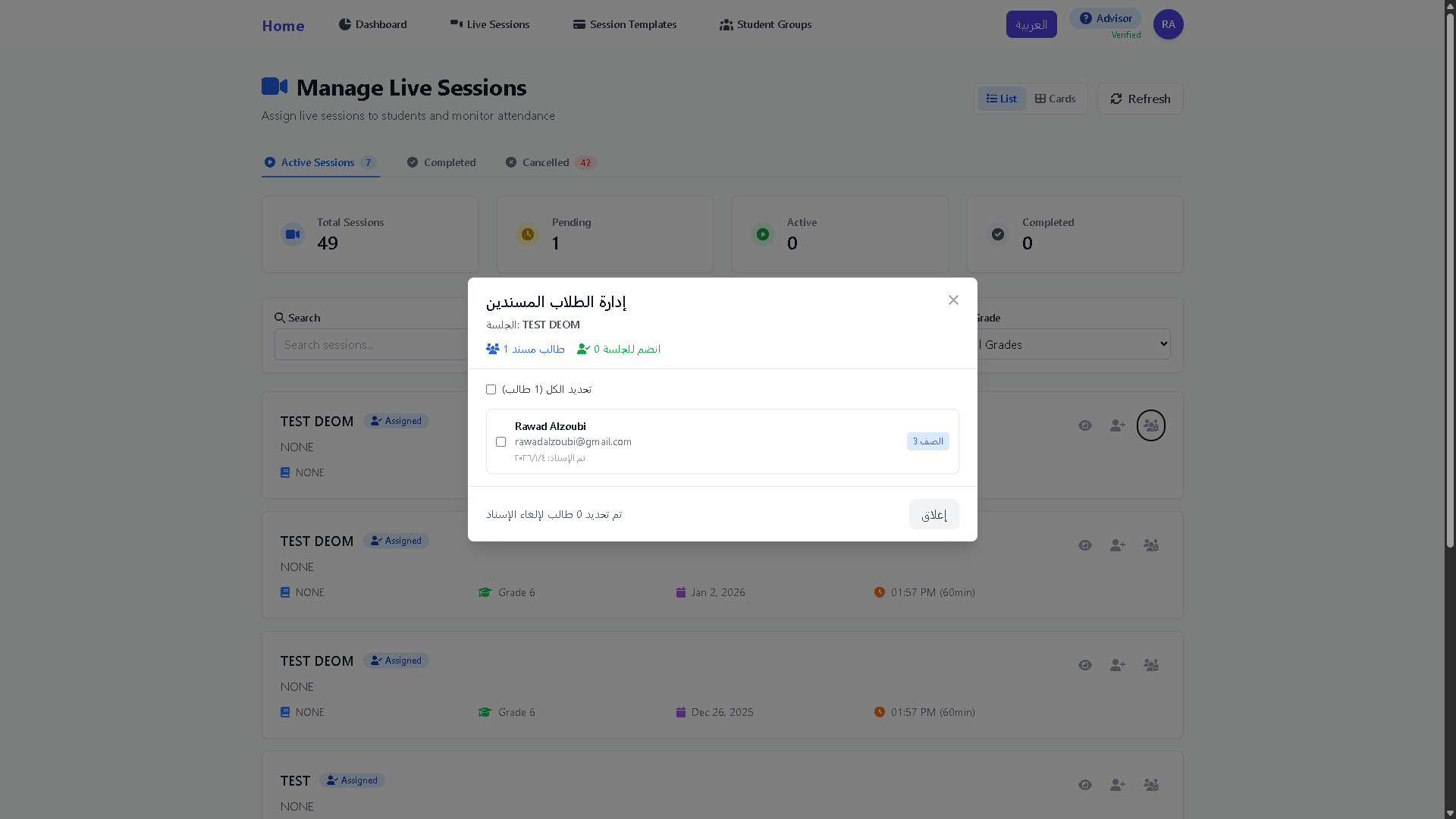
**Managed Assigned Student  
**

Figure 167 - Assign Session to student page

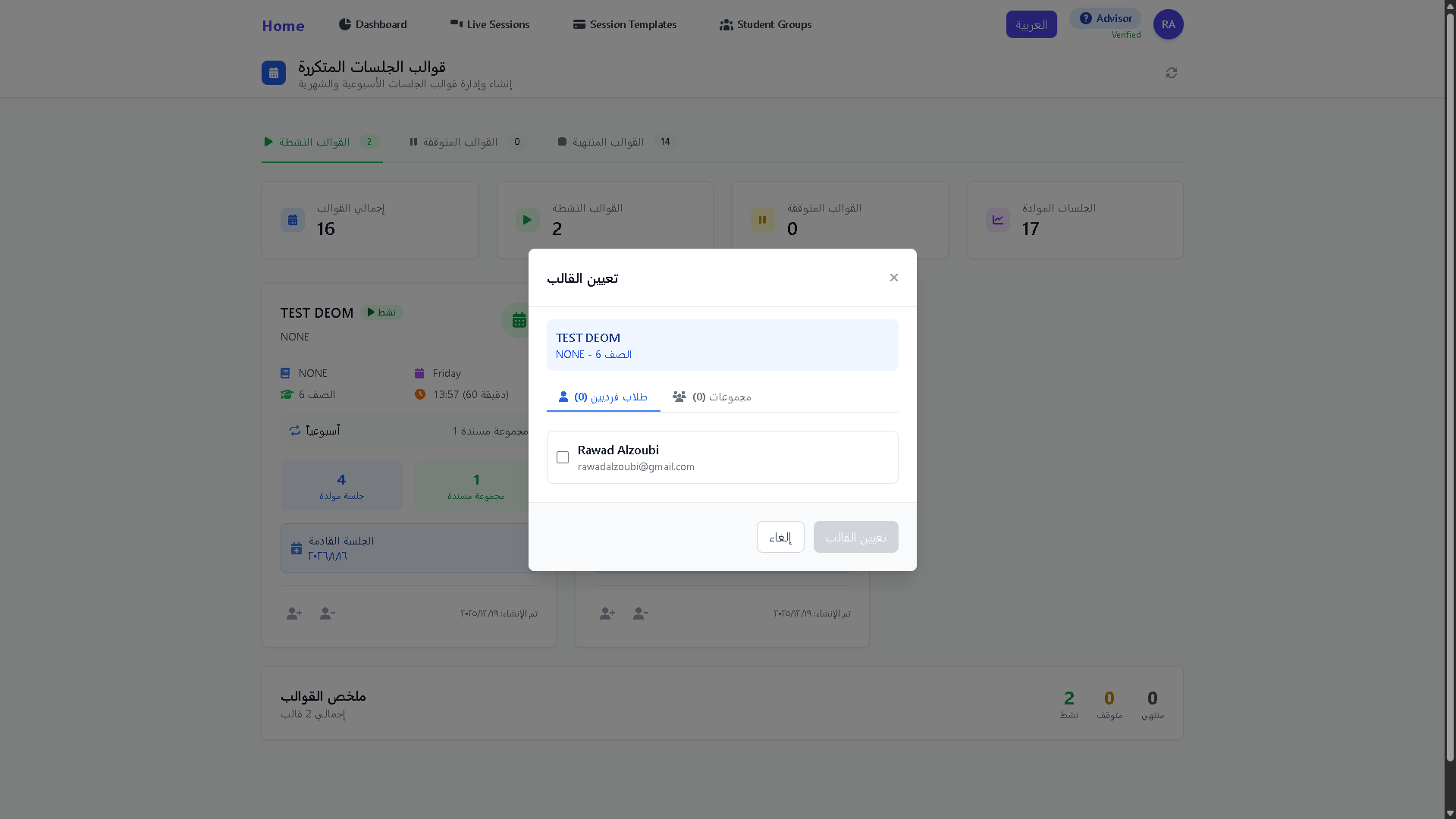
**Assign Templates to Students**

Figure 168 - Assign Templates to Students page  
Manage Assigned Student  


Figure 169 - Manage Assigned Student

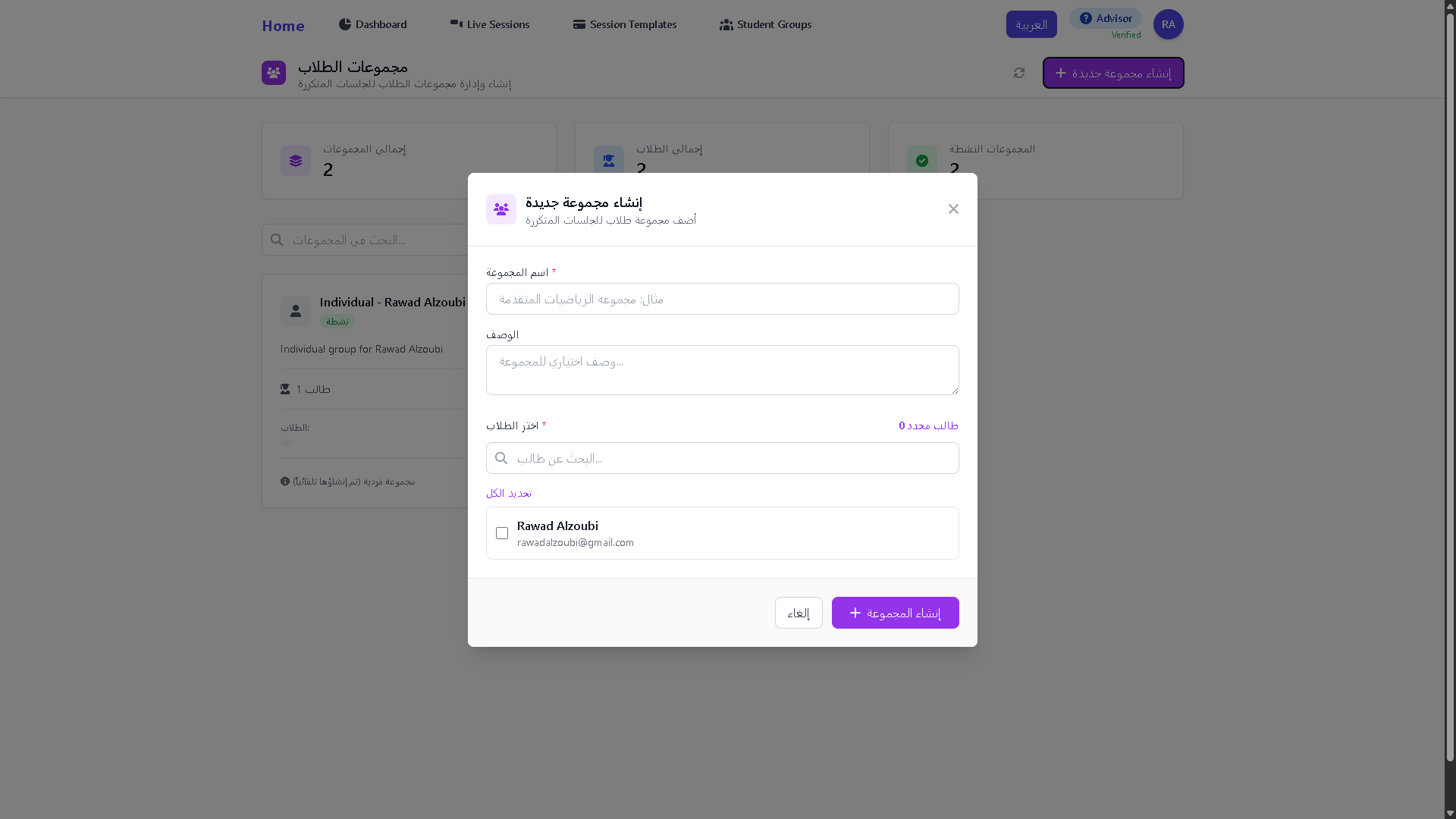
**Create Student Group  
**

Figure 170 - Create Student Group page

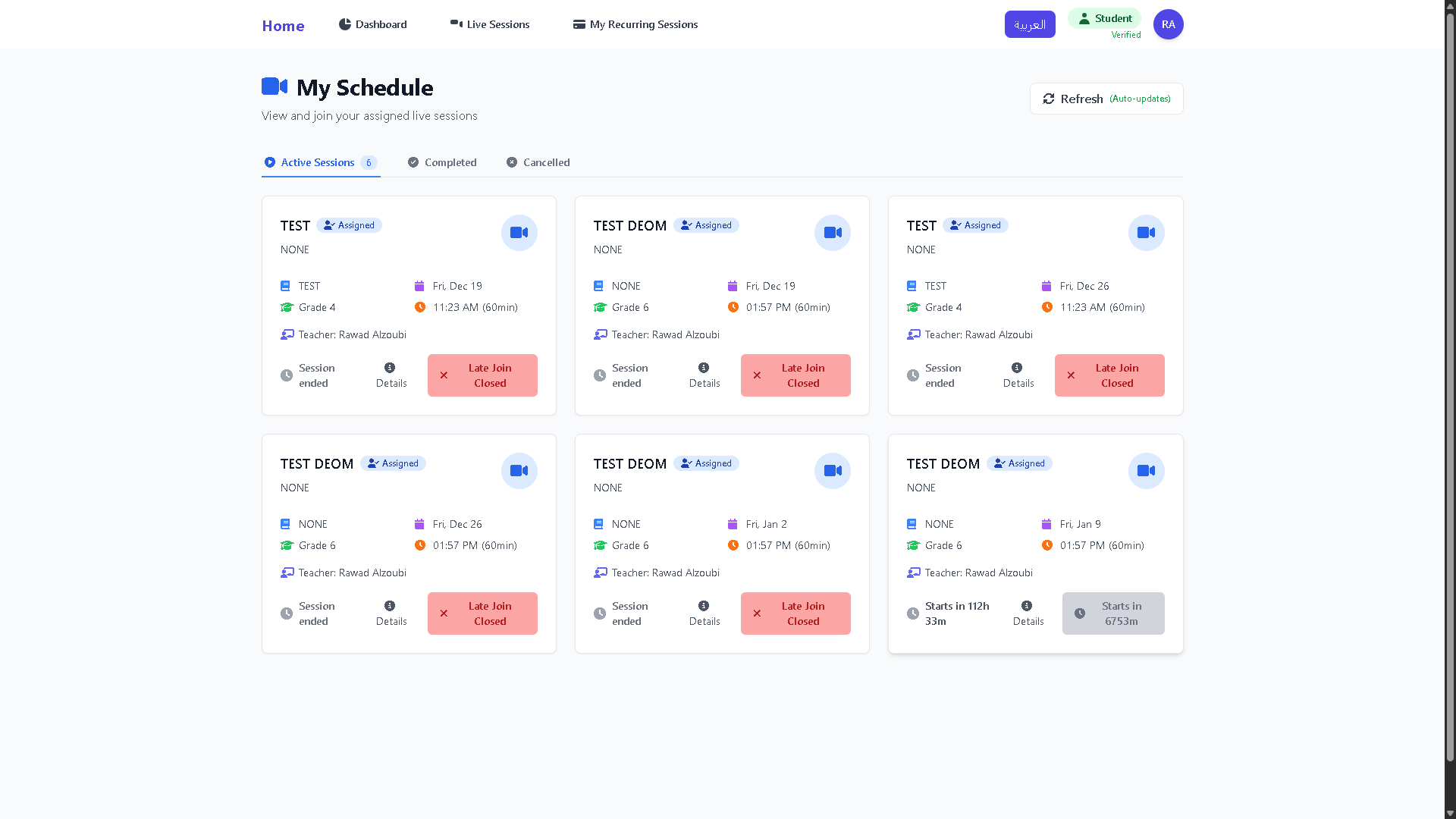
**View Student Sessions   
**

Figure 171 - View Student Sessions page

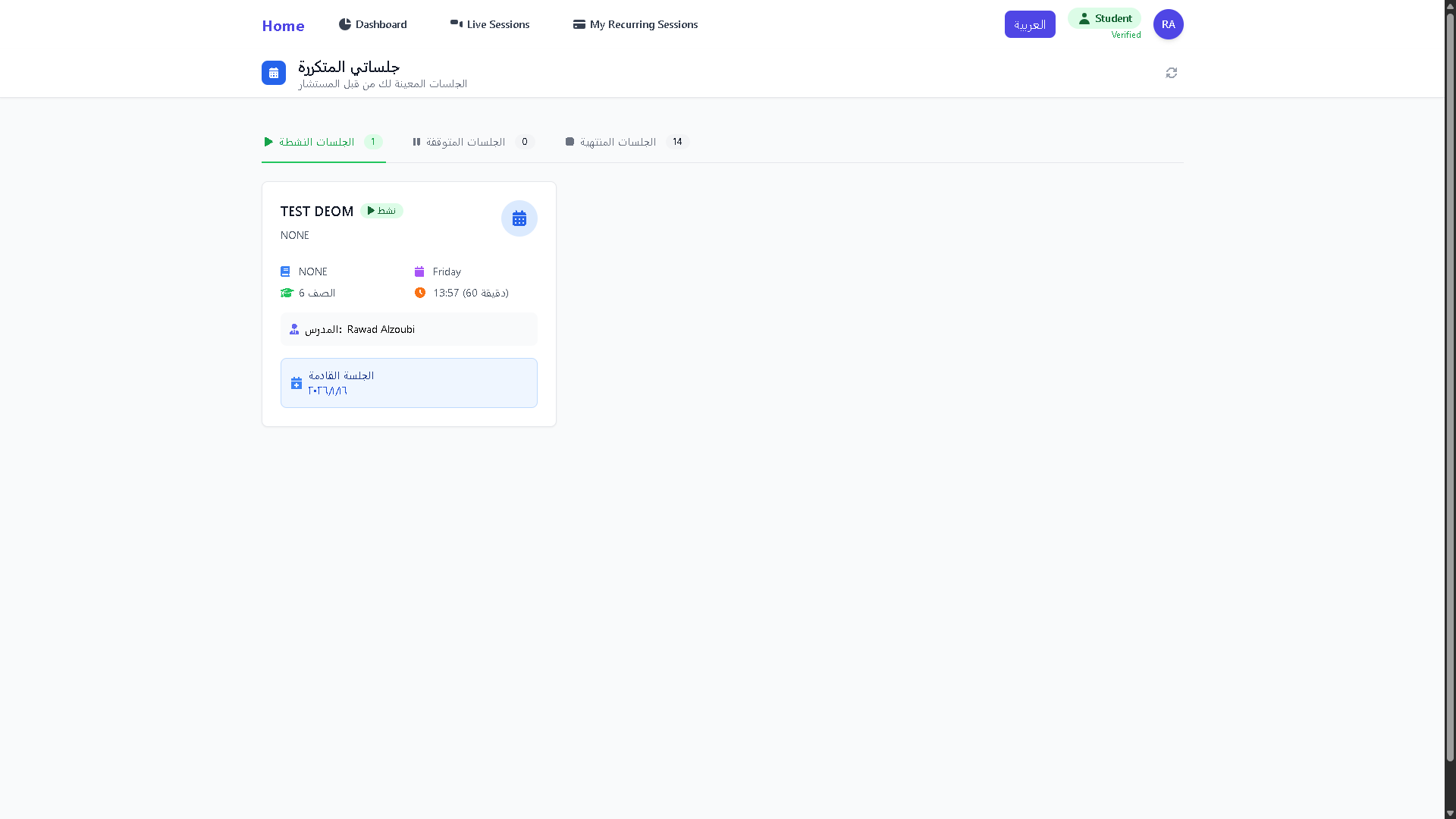
**View Student Templates  
**

Figure 172 - View Student Templates page

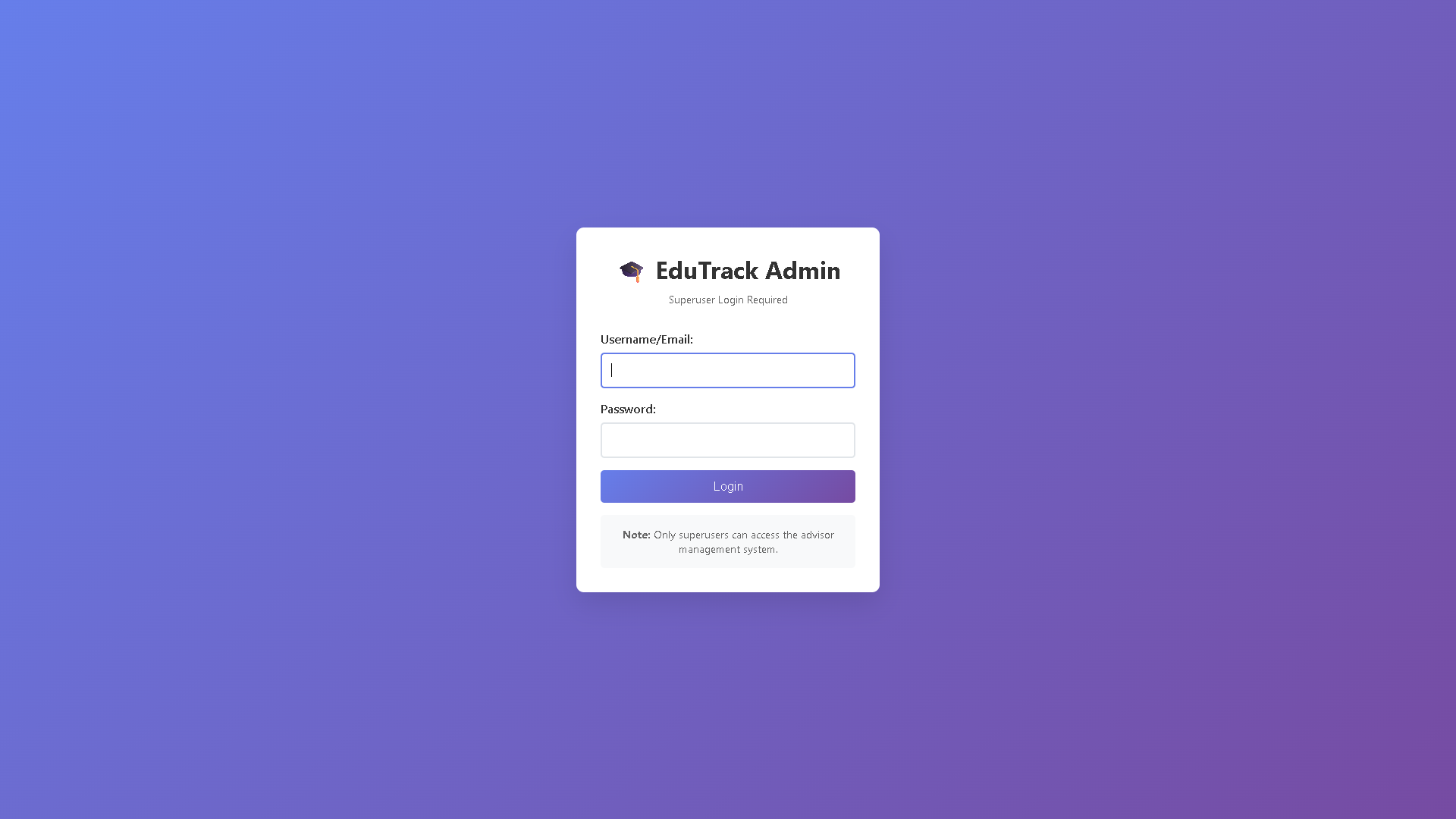
**Admin Login  
**

Figure 173 - Admin Login page

**Admin Home Page**

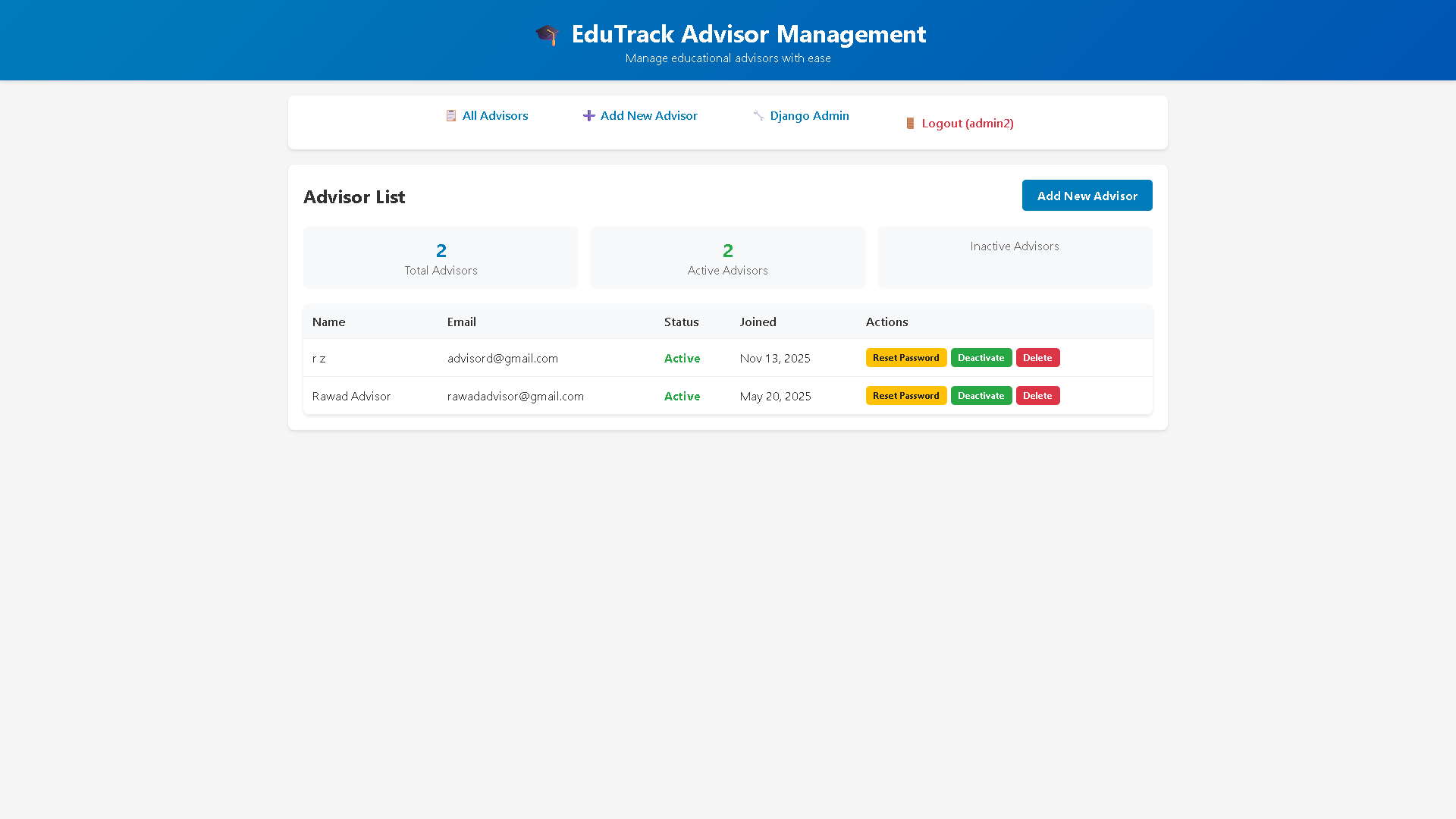
****

Figure 174 - Admin Home Page

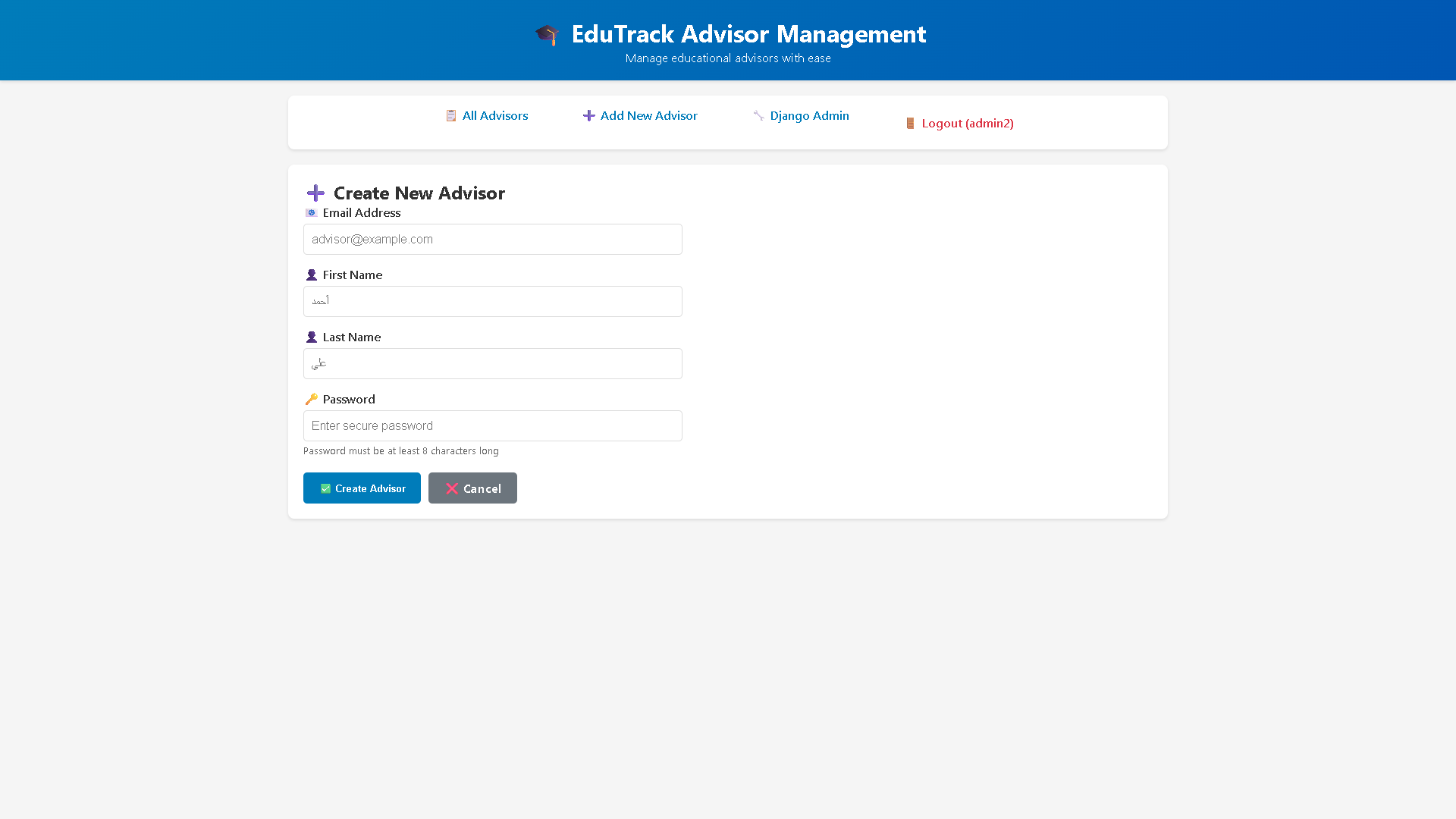
**Create Advisor**

Figure 175 - Create Advisor page

**Delete Advisor**

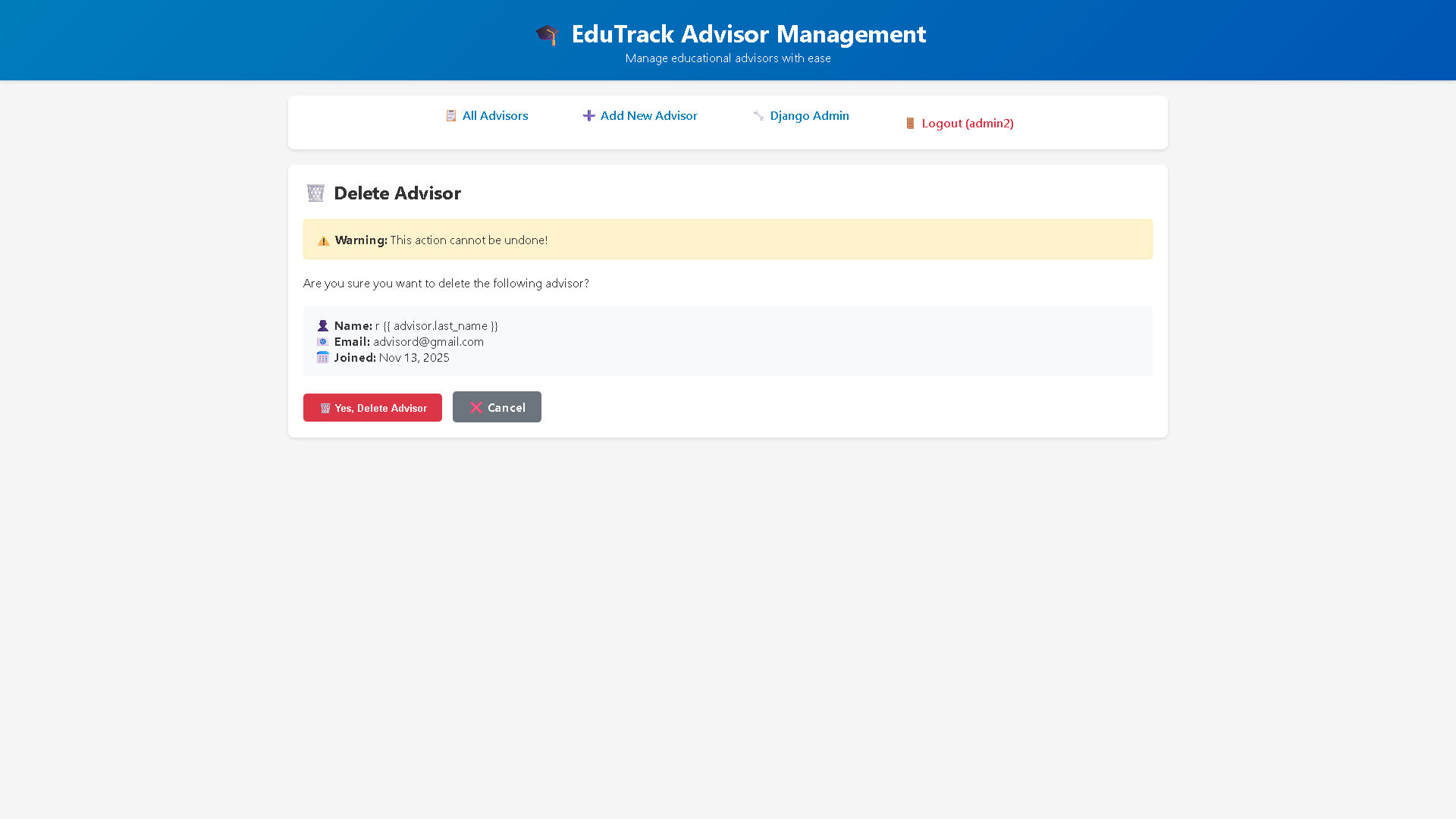
****

Figure 176 - Delete Advisor Page

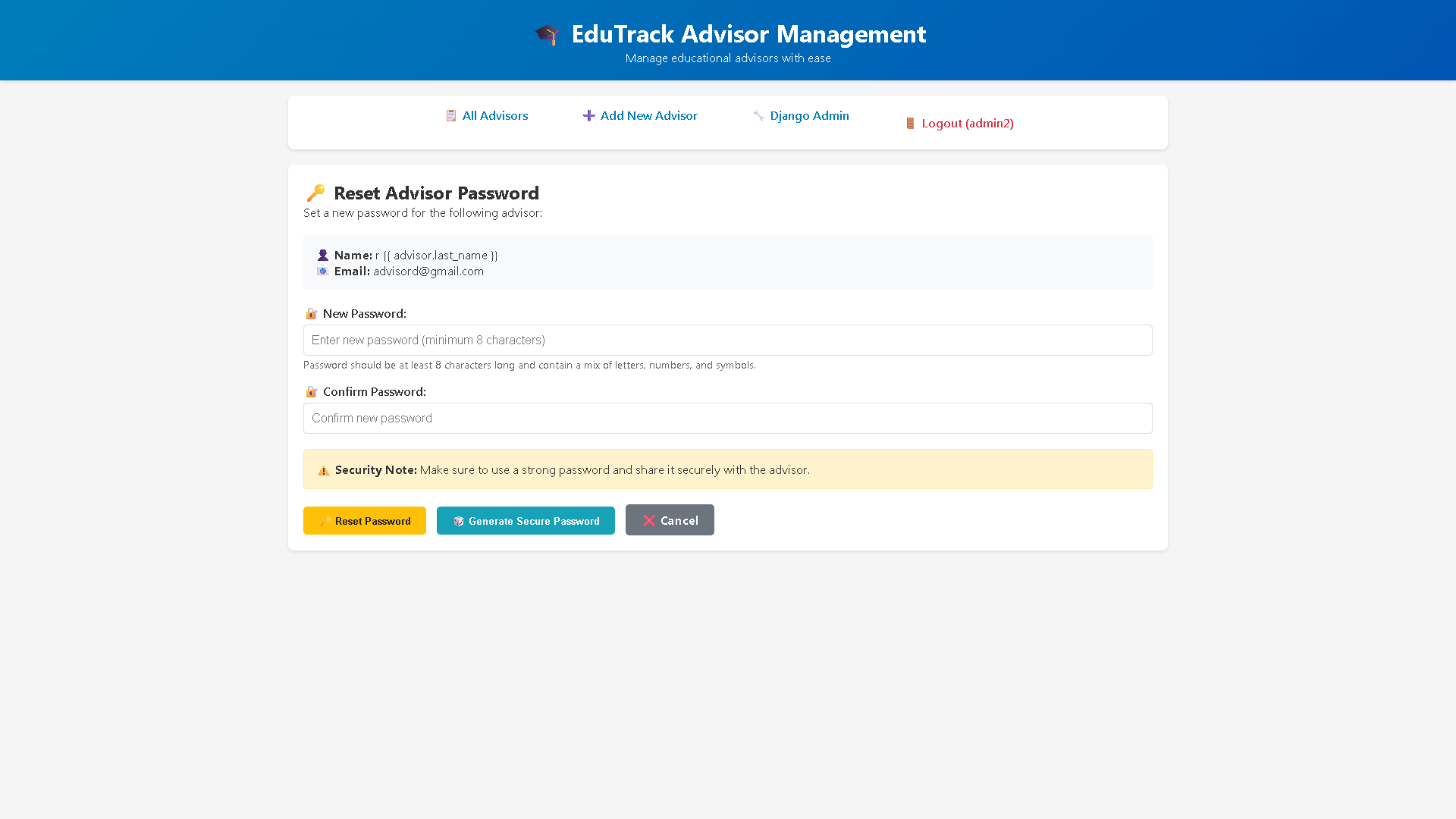
**Reset Advisor Password**

Figure 177 - Reset Advisor Password page

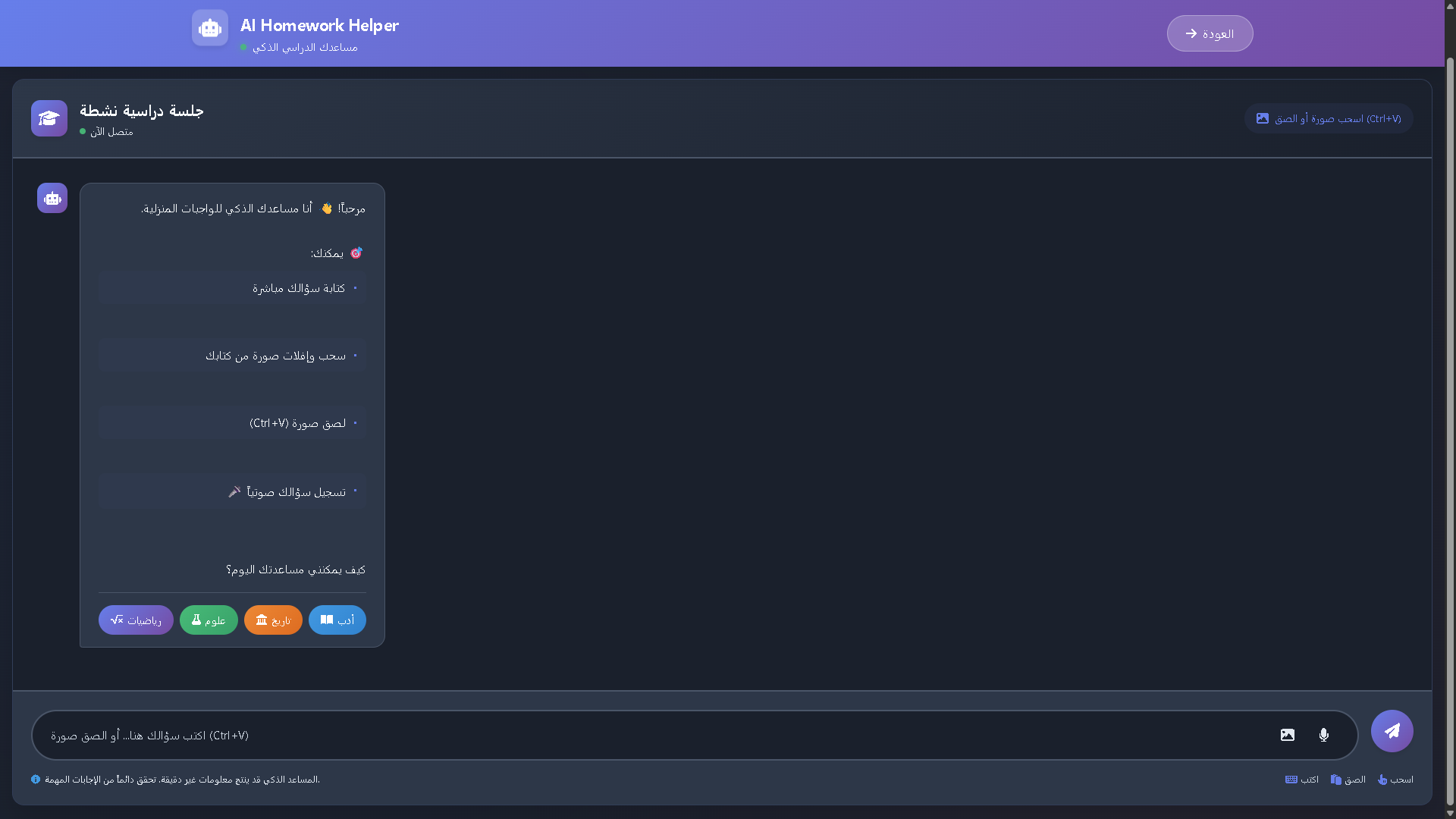
**Smart HomeWork Helper**

Figure 178 - Smart HomeWork Helper page

### 6.4 Execute Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (Category) | (ID) | (Test Title) | البيانات المستخدمة (Tested Data) | (Expected Result) | (Actual Result) | (Status) |
| 1. Authentication | **TC-AUTH-001** | Successful Registration | Name: Rawad, Email: rawad.student@example.com, Role: Student | Account created, verification email sent. | Account created in DB, email received. | **Pass** |
|  | **TC-AUTH-002** | Successful Login | Email: teacher@example.com, Pass: ValidPass | User logged in, redirected to dashboard. | Login successful, redirected to Teacher Dashboard. | **Pass** |
|  | **TC-AUTH-003** | Successful Logout | User: Mohammed Hammoud | Session terminated, redirected to login. | Token cleared, redirected to login page. | **Pass** |
|  | **TC-AUTH-004** | Password Reset | Email: rawad.student@example.com | Reset link sent to email. | Success message shown, link arrived. | **Pass** |
|  | **TC-USER-001** | Edit Profile | Field: Country, New Value: SYRIA | Profile updated, new info displayed. | Saved successfully, "SYRIA" is visible. | **Pass** |
| 2. Content Mgmt | **TC-CNT-001** | Upload Video Lesson | Title: Intro to Science, File: science.mp4 | Lesson created, appears in list. | Lesson uploaded, visible in "Recent Lessons". | **Pass** |
|  | **TC-CNT-002** | Create MCQ Exercise | Q: "What is H2O?", Ans: "Water" | MCQ added to lesson quiz. | Question saved and linked correctly. | **Pass** |
|  | **TC-CNT-003** | Delete Lesson | Action: Delete "Old Lesson" | Lesson permanently deleted. | Removed from DB and UI. | **Pass** |
| 3. Interaction | **TC-STU-001** | Browse Assigned Lessons | Student: Rawad Alzoubi | Only assigned lessons visible. | Only "Intro to Science" was visible. | **Pass** |
|  | **TC-STU-002** | Solve Exercises | Quiz: Science Quiz | System grades and shows score. | Submitted, Score "10/10" displayed. | **Pass** |
|  | **TC-ADV-001** | Assign Lesson | Student: Rawad, Lesson: Math 101 | Lesson appears in student dash. | Assignment successful, visible to Rawad. | **Pass** |
|  | **TC-ADV-002** | Filter Lessons | Filter: "Grade 3" | List shows only Grade 3. | List filtered correctly. | **Pass** |
| 4. Feedback | **TC-EVAL-001** | View Quiz Results | Student: Rawad | View answers vs correct key. | Details page showed correct breakdown. | **Pass** |
|  | **TC-EVAL-002** | Submit Feedback | Text: "Great job!" | Feedback visible to student. | Feedback saved, notification sent. | **Pass** |
|  | **TC-EVAL-003** | View Student Answers | Quiz Attempt #45 | Teacher sees specific answers. | Teacher viewed attempt details. | **Pass** |
| 5. AI Engine | **TC-AI-001** | Visual Search (CLIP) | Input: Image of Triangle | Returns Geometry lessons. | Uploaded image, got 3 relevant PDFs. | **Pass** |
|  | **TC-AI-002** | Voice Search (Whisper) | Audio: "Explain Python loops" | Text transcribed, results shown. | Transcribed correctly, showed coding lessons. | **Pass** |
|  | **TC-AI-003** | Video Semantic Search | Query: "Photosynthesis" | Player jumps to timestamp. | Video played exactly from 03:45. | **Pass** |
|  | **TC-AI-004** | Smart RAG Answer | Q: "States of matter" | Summary: "Solid, Liquid, Gas". | Text summary appeared at top of results. | **Pass** |
| 6. Technical/NF | **TC-NF-001** | Frontend Validation | Email: "invalid-email" | Inline error appears. | Red error "Invalid Format" shown. | **Pass** |
|  | **TC-SEC-001** | Secure Hashing | DB Check User ID 5 | Password is hash string. | Field contained PBKDF2 hash. | **Pass** |
| 7. Live Sessions | **TC-LS-001** | Create Live Session | Title: Math Live, Date: Tomorrow | Session created (PENDING). | Saved, appeared in "Upcoming". | **Pass** |
|  | **TC-LS-002** | Join Live Session | Action: Click "Join" | Jitsi popup opens. | Window opened, join time logged. | **Pass** |
|  | **TC-LS-003** | Validate Timing | Action: Join future session | Button disabled / Access denied. | Button disabled, tooltip shown. | **Pass** |
|  | **TC-LS-004** | Filter Schedule | Filter: "Completed" | Shows only past sessions. | List updated correctly. | **Pass** |
| 8. Admin System | **TC-ADM-001** | Create Advisor | Name: Advisor Ali | Account created, Login OK. | Created and logged in successfully. | **Pass** |
|  | **TC-ADM-002** | Deactivate Advisor | Action: Toggle "Inactive" | Status Inactive, Login denied. | Login attempt returned 403. | **Pass** |
|  | **TC-ADM-003** | Invalidate Session | Action: Deactivate active user | User logged out immediately. | Redirected to login instantly. | **Pass** |
|  | **TC-ADM-004** | Reset Password | New Pass: "Secure@123" | Login with new pass OK. | Old pass rejected, new worked. | **Pass** |
| 9. Recurring | **TC-RS-001** | Create Template | Pattern: Weekly, Mon 9am | Template saved as "Active". | Status Active, template stored. | **Pass** |
|  | **TC-RS-002** | View Generated | Action: Trigger Generator | Future sessions appear. | Calendar populated for next month. | **Pass** |
|  | **TC-RS-003** | Pause Template | Action: Click Pause | Status "Paused", Stop gen. | Status Paused, generation stopped. | **Pass** |
|  | **TC-RS-004** | Edit Template | Change time to 11:00 AM | Future sessions moved. | All upcoming sessions updated. | **Pass** |
|  | **TC-RS-005** | Assign to Group | Group: "Grade 5 A" | Schedule visible to group. | Students confirmed visibility. | **Pass** |

Table 73 - Execute test case



Figure 179 - Test Cases

### 6.5 Requirements Traceability Matrix V2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Title | Analysis Section | Design Section | Code Implementation | Test Case ID |
| REQ-2.1 | User Registration | [4.6 Sprint 2 Analysis](#_UC-2.1_–_Register) | [Sprint 2 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_1:_5.3) | [Register.jsx / UserViewSet](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AUTH-001](#_Table_67_-) |
| REQ-2.2 | Login | [4.6 Sprint 2 Analysis](#_UC-2.2_–_Login) | [Sprint 2 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_1:_5.3) | [Login.jsx / TokenObtainPairView](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AUTH-002](#_Table_67_-) |
| REQ-2.3 | User Profile Viewing | [4.6 Sprint 2 Analysis](#_UC-2.3_–_View) | [Sprint 2 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_1:_5.3) | [UserProfile.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-USER-001](#_Table_67_-) |
| REQ-2.4 | User Profile Editing | [4.6 Sprint 2 Analysis](#_UC-2.4_–_Edit) | [Sprint 2 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_1:_5.3) | [EditProfileModal.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-USER-001](#_Table_67_-) |
| REQ-2.5 | Logout | [4.6 Sprint 2 Analysis](#_UC-2.5_–_Logout) | [Sprint 2 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_1:_5.3) | [Navbar.jsx (Logout Function)](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AUTH-003](#_Table_67_-) |
| REQ-2.6 | Reset Password | [4.6 Sprint 2 Analysis](#_UC-2.6_–_Reset) | [Sprint 2 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_1:_5.3) | [ForgotPassword.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AUTH-004](#_Table_67_-) |
| REQ-4.1 | Upload Video Lesson | [4.7 Sprint 4 Analysis](#_UC-4.1_–_Upload) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [UploadLesson.jsx / LessonViewSet](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-CNT-001](#_Table_67_-) |
| REQ-4.2 | Upload PDF Lesson | [4.7 Sprint 4 Analysis](#_UC-4.2_–_Upload) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [UploadLesson.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-CNT-001](#_Table_67_-) |
| REQ-4.3 | Upload Exercises (MCQ) | [4.7 Sprint 4 Analysis](#_UC-4.3_–_Upload) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [CreateQuiz.jsx / QuestionViewSet](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-CNT-002](#_Table_67_-) |
| REQ-4.4 | View My Lessons | [4.7 Sprint 4 Analysis](#_UC-4.4_–_View) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [TeacherLessons.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-CNT-003](#_Table_67_-) |
| REQ-4.5 | Update Lesson | [4.7 Sprint 4 Analysis](#_UC-4.5_–_Update) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [EditLessonModal.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-CNT-003](#_Table_67_-) |
| REQ-4.6 | Delete Lesson | [4.7 Sprint 4 Analysis](#_UC-4.6_–_Delete) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [LessonService.js (delete)](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-CNT-003](#_Table_67_-) |
| REQ-4.7 | Browse Assigned Lessons | [4.7 Sprint 4 Analysis](#_UC-4.7_–_Browse) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [StudentDashboard.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-STU-001](#_Table_67_-) |
| REQ-4.8 | Select Lesson | [4.7 Sprint 4 Analysis](#_UC-4.8_–_Select) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [LessonDetails.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-STU-001](#_Table_67_-) |
| REQ-4.9 | Download PDF Lessons | [4.7 Sprint 4 Analysis](#_UC-4.9_–_Download) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [FileDownloader.js](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-STU-002](#_Table_67_-) |
| REQ-4.10 | Solve Exercises | [4.7 Sprint 4 Analysis](#_UC-4.10_–_Solve) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [QuizPlayer.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-STU-002](#_Table_67_-) |
| REQ-4.11 | View Student Progress | [4.7 Sprint 4 Analysis](#_UC-4.11_–_View) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [ProgressTracker.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-EVAL-001](#_Table_67_-) |
| REQ-4.12 | Assign Lessons to Student | [4.7 Sprint 4 Analysis](#_UC-4.12_–_Assign) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [AssignContentModal.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADV-001](#_Table_67_-) |
| REQ-4.13 | Search Lessons by Filter | [4.7 Sprint 4 Analysis](#_UC-4.13_–_Search) | [Sprint 4 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.4_Sprint4_Design) | [LessonFilter.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADV-002](#_Table_67_-) |
| REQ-5.4 | View Quiz Performance (Teacher) | [4.8 Sprint 5 Analysis](#_UC-5.4_–_View) | [Sprint 5 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.5_Sprint5_Design) | [StudentPerformance.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-EVAL-001](#_Table_67_-) |
| REQ-5.5 | View All Quiz Results (Advisor) | [4.8 Sprint 5 Analysis](#_UC-5.5_–_View) | [Sprint 5 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.5_Sprint5_Design) | [AdvisorStats.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-EVAL-001](#_Table_67_-) |
| REQ-5.6 | Submit Feedback | [4.8 Sprint 5 Analysis](#_UC-5.6_–_Submit) | [Sprint 5 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.5_Sprint5_Design) | [FeedbackModal.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-EVAL-002](#_Table_67_-) |
| REQ-5.7 | View Feedback | [4.8 Sprint 5 Analysis](#_UC-5.7_–_View) | [Sprint 5 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.5_Sprint5_Design) | [StudentNotifications.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-EVAL-002](#_Table_67_-) |
| REQ-LS-01 | Create Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-01_–_Create) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [CreateSession.jsx / LiveSessionViewSet](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-001](#_Table_67_-) |
| REQ-LS-02 | Edit Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-02_–_Edit) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [EditSession.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-004](#_Table_67_-) |
| REQ-LS-03 | Cancel Live Session | [4.10 Sprint 9 Analysis](#_UC-LS-03_–_Cancel) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [SessionService.js](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-004](#_Table_67_-) |
| REQ-LS-04 | Join Live Session (Host) | [4.10 Sprint 9 Analysis](#_UC-LS-04_–_Join) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [JitsiComponent.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-002](#_Table_67_-) |
| REQ-LS-05 | View Teacher's Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-05_–_View) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [TeacherSchedule.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-004](#_Table_67_-) |
| REQ-LS-06 | View Assigned Students | [4.10 Sprint 9 Analysis](#_UC-LS-06_–_View) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [SessionAttendees.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-001](#_Table_67_-) |
| REQ-LS-07 | View All Live Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-07_–_View) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [AllSessionsList.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-004](#_Table_67_-) |
| REQ-LS-08 | Assign Students to Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-08_–_Assign) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [AssignSession.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-003](#_Table_67_-) |
| REQ-LS-09 | Unassign Students | [4.10 Sprint 9 Analysis](#_UC-LS-09_–_Unassign) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [AssignSession.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-003](#_Table_67_-) |
| REQ-LS-10 | Modify Student Assignments | [4.10 Sprint 9 Analysis](#_UC-LS-10_–_Modify) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [AssignSession.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-003](#_Table_67_-) |
| REQ-LS-11 | View Assigned Sessions | [4.10 Sprint 9 Analysis](#_UC-LS-11_–_View) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [MyLiveSessions.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-002](#_Table_67_-) |
| REQ-LS-12 | Join Live Session (Student) | [4.10 Sprint 9 Analysis](#_UC-LS-12_–_Join) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [JitsiComponent.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-002](#_Table_67_-) |
| REQ-LS-13 | View Session Details | [4.10 Sprint 9 Analysis](#_UC-LS-13_–_View) | [Sprint 9 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_Release_2:_5.6) | [SessionDetails.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-LS-003](#_Table_67_-) |
| REQ-ADS-01 | Admin Login | [4.11 Sprint 10 Analysis](#_REQ-ADS-01_–_Admin) | [Sprint 10 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.7_Sprint10_Design) | [AdminLogin.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADM-001](#_Table_67_-) |
| REQ-ADS-02 | View Advisors List | [4.11 Sprint 10 Analysis](#_REQ-ADS-02_–_View) | [Sprint 10 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.7_Sprint10_Design) | [AdvisorsList.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADM-001](#_Table_67_-) |
| REQ-ADS-03 | Create New Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-03_–_Create) | [Sprint 10 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.7_Sprint10_Design) | [CreateAdvisor.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADM-001](#_Table_67_-) |
| REQ-ADS-04 | Delete Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-04_–_Delete) | [Sprint 10 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.7_Sprint10_Design) | [AdminService.js](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADM-002](#_Table_67_-) |
| REQ-ADS-05 | Reset Advisor Password | [4.11 Sprint 10 Analysis](#_REQ-ADS-05_–_Reset) | [Sprint 10 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.7_Sprint10_Design) | [ResetUserPass.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADM-004](#_Table_67_-) |
| REQ-ADS-06 | Activate Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-06_–_Activate) | [Sprint 10 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.7_Sprint10_Design) | [UserStatusToggle.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADM-002](#_Table_67_-) |
| REQ-ADS-07 | Deactivate Advisor | [4.11 Sprint 10 Analysis](#_REQ-ADS-07_–_Deactivate) | [Sprint 10 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.7_Sprint10_Design) | [UserStatusToggle.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-ADM-002](#_Table_67_-) |
| REQ-RS-01 | Create Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-01_–_Create) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [CreateTemplate.jsx / TemplateViewSet](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-001](#_Table_67_-) |
| REQ-RS-02 | View Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-02_–_View) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [TemplatesList.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-001](#_Table_67_-) |
| REQ-RS-03 | Edit Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-03_–_Edit) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [EditTemplate.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-004](#_Table_67_-) |
| REQ-RS-04 | Delete Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-04_–_Delete) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [TemplateService.js](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-001](#_Table_67_-) |
| REQ-RS-05 | Pause Template Status | [4.12 Sprint 11 Analysis](#_UC-RS-05_–_Pause) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [TemplateActions.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-003](#_Table_67_-) |
| REQ-RS-06 | Resume Paused Templates | [4.12 Sprint 11 Analysis](#_UC-RS-06_–_Resume) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [TemplateActions.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-003](#_Table_67_-) |
| REQ-RS-07 | End Template Permanently | [4.12 Sprint 11 Analysis](#_UC-RS-07_–_End) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [TemplateActions.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-003](#_Table_67_-) |
| REQ-RS-10 | View All Session Templates | [4.12 Sprint 11 Analysis](#_UC-RS-09_–_View) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [AllTemplates.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-005](#_Table_67_-) |
| REQ-RS-11 | Assign Templates (Group) | [4.12 Sprint 11 Analysis](#_UC-RS-10_–_Assign) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [AssignGroupTemplate.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-005](#_Table_67_-) |
| REQ-RS-12 | Create Student Groups | [4.12 Sprint 11 Analysis](#_UC-RS-11_–_Create) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [CreateGroup.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-005](#_Table_67_-) |
| REQ-RS-13 | Manage Student Groups | [4.12 Sprint 11 Analysis](#_UC-RS-12_–_Manage) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [ManageGroups.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-005](#_Table_67_-) |
| REQ-RS-15 | View Assigned Templates | [4.12 Sprint 11 Analysis](#_UC-RS-14_–_View) | [Sprint 11 Design](file:///C:\Users\Rawad\Desktop\GRADUATION%20PROJECT\SENOIR-1-PROJECT\AI-powerd%20complemntary%20edu%20system.docx#_5.8_Sprint11_Design) | [MyRecurringSchedule.jsx](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-RS-005](#_Table_67_-) |
| REQ-AI-01 | Upload Image for Search | 4.9 AI Analysis | 5.11 AI Arch | [HomeworkHelperPage.jsx / CLIPService.py](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AI-001](#_Table_67_-) |
| REQ-AI-02 | Voice Search (Whisper) | 4.9 AI Analysis | 5.11 AI Arch | [HomeworkHelperPage.jsx / WhisperService.py](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AI-002](#_Table_67_-) |
| REQ-AI-03 | Search Inside Videos | 4.9 AI Analysis | 5.11 AI Arch | [HomeworkHelperPage.jsx / FAISS\_Index](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AI-003](#_Table_67_-) |
| REQ-AI-04 | Smart Homework Helper | 4.9 AI Analysis | 5.11 AI Arch | [HomeworkHelperPage.jsx / RAGPipeline.py](https://github.com/rawadalzoubi/AI-powerd-complemntary-edu-system) | [TC-AI-004](#_Table_67_-) |

Table 74 - RTM V2

### 

**Chapter 7 – AI Section**

# Design and Implementation of a Multimodal RAG Educational Assistant using Agile Methodology

## Executive Summary

This report details the development of a Multimodal Retrieval-Augmented Generation (RAG) system designed to serve as an intelligent educational assistant for students who have dropped out of formal schooling. The project addresses the critical need for interactive, accessible, and self-paced learning tools. By leveraging Agile methodology across two distinct sprints, the project successfully integrates text processing, image understanding via textual bridging, and automatic speech recognition (ASR). The system utilizes a cost-efficient architecture comprising Google Gemini 1.5 Flash, Qdrant Vector Database, and FastEmbed, deployed locally to function on standard hardware without GPU acceleration. The result is a robust, scalable system that democratizes access to advanced AI tutoring.

## **Chapter 1: Introduction**

### 1.1 Project Background and Motivation

The global education sector is witnessing a digital transformation; however, a significant "Digital Divide" remains. Students who have dropped out of school often lack access to structured learning environments and rely heavily on static digital repositories, primarily in PDF format. While these documents contain valuable knowledge, they are passive. A student cannot "ask" a PDF a question, nor can they easily interpret complex scientific diagrams without guidance.

This project proposes a solution that bridges the gap between static content and active learning. By building an AI agent capable of "seeing" diagrams, "reading" text, and "hearing" questions, we provide a virtual tutor experience that mimics the interactivity of a classroom.

### 1.2 Problem Statement

The project addresses four specific technical and educational challenges:

* **Unstructured Knowledge Silos:** Educational content is trapped in non-machine-readable formats (PDFs), making granular information retrieval difficult.
* **The Visual Information Gap:** Traditional search engines (Keyword Search) utilize Inverted Indices (TF-IDF), which completely ignore visual data. A search for "Mitosis" will not return a diagram of cell division unless the specific keyword is in the metadata.
* **Accessibility for Low-Literacy Users:** Many drop-out students may struggle with writing complex academic queries. Text-only interfaces create a barrier to entry.
* **Resource Constraints:** State-of-the-art AI usually requires expensive NVIDIA GPUs (e.g., A100s). Educational NGOs and students typically possess standard laptops (CPU-only), requiring a highly optimized lightweight architecture.

### 1.3 Project Objectives

The primary goal is to engineer a Low-Resource Multimodal RAG Agent.

**Detailed Sub-objectives:**

* **Ingestion:** Develop a pipeline to parse PDF text and extract images with their page coordinates.
* **Semantic Understanding:** Move beyond keyword matching to "Semantic Search" using high-dimensional vector embeddings.
* **Visual Interpretation:** Implement a cost-effective method to make images searchable (Image Captioning vs. Visual Embeddings).
* **Voice Integration:** Integrate a speech-to-text layer supporting Arabic dialects.
* **Performance:** Achieve a sub-5-second latency on a standard i5 processor with 8GB RAM.

### 1.4 Methodology: The Agile Approach

The project follows the Scrum Framework within Agile. Development was broken down into two time-boxed iterations (Sprints):

* **Sprint 1 (Foundations):** Focus on the "Backend Intelligence" (Text/Image processing and Database setup).
* **Sprint 2 (User Experience):** Focus on "Accessibility" (Voice processing and UI refinement).

## **Chapter 2: Theoretical Framework and Comparative Analysis**

### 2.1 The Evolution of Generative AI

Generative AI has evolved from Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs) to the Transformer Architecture introduced by Google in 2017. Transformers utilize "Self-Attention" mechanisms to weigh the importance of different words in a sequence, allowing for parallel processing and deeper contextual understanding.

#### 2.1.1 Comparison of Large Language Models (LLMs)

For this project, we evaluated three leading models. The decision matrix below justifies the choice of Gemini 1.5 Flash.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | GPT-4o (OpenAI) | Claude 3.5 Sonnet | Gemini 1.5 Flash (Selected) |
| **Context Window** | 128k Tokens | 200k Tokens | **1 Million Tokens** (Best for large books) |
| **Multimodal Native** | Yes | Yes | **Yes** (Highly optimized for video/image) |
| **Cost (Input)** | $5.00 / 1M tokens | $3.00 / 1M tokens | \*\*$0.35 / 1M tokens\*\* (Most affordable) |
| **Speed** | Moderate | Moderate | **Extremely Fast** (Low latency) |
| **Free Tier** | Limited | Limited | **Generous Free Tier** via AI Studio |

**Conclusion:** Gemini 1.5 Flash was selected because it offers the largest context window (allowing us to feed entire chapters if needed) and the lowest cost/latency ratio, which is critical for a student-focused application.

### 2.2 Retrieval-Augmented Generation (RAG) vs. Fine-Tuning

A critical architectural decision was choosing between RAG and Fine-Tuning.

|  |  |  |
| --- | --- | --- |
| Aspect | Fine-Tuning | RAG (Selected) |
| **Knowledge Update** | Static (Requires re-training to add new books) | Dynamic (Just add PDF to database) |
| **Hallucinations** | High risk (Model may invent facts) | Low risk (Model is grounded in retrieved context) |
| **Computational Cost** | High (Requires GPUs for training) | Low (Only inference cost) |
| **Data Privacy** | Data becomes part of the model | Data remains in local database |

**Justification:** RAG was chosen because educational curriculums change. RAG allows us to update the "Knowledge Base" instantly without retraining the AI model.

### 2.3 Vector Databases and Search Algorithms

To perform semantic search, we need a vector database. We compared the top options available in the market.

|  |  |  |  |
| --- | --- | --- | --- |
| Database | Deployment Type | Cost | Pros for this Project |
| **Pinecone** | Cloud Only | Expensive (Tiered) | Easy to use, but no local mode. |
| **Milvus** | Local/Cloud | Free (Open Source) | Powerful, but complex setup (requires Docker/K8s). |
| **ChromaDB** | Local | Free | Good for Python, but slower performance on large datasets. |
| **Qdrant** | Local/Cloud | Free (Open Source) | **Written in Rust (Fast), Supports "Disk Storage" mode.** |

**Decision:** Qdrant was selected. Its ability to run in "Local Mode" (storing data on the hard drive rather than requiring massive RAM) makes it perfect for the target hardware (standard laptops).

### 2.4 Multimodal Processing Strategies

There are two main ways to handle images in RAG:

1. **Multimodal Embeddings (e.g., CLIP):** Converting images and text into a shared vector space.
   * Drawback: CLIP is good at general concepts ("A dog") but bad at OCR and specific technical details ("Figure 3.1: Circuit Diagram").
2. **Image Captioning (Textual Bridging):** Using a Vision-Language Model (VLM) to describe the image in text, then embedding that text.
   * Advantage: The description captures detailed information (labels, numbers, relationships) which allows for precise text-based search.

**Strategy:** We adopted Textual Bridging. Every image is passed to Gemini to generate a dense technical description, which is then indexed.

## **Chapter 3: System Analysis and Sprint Planning**

### 3.1 User Personas

* **Persona A:** The Returning Student. Struggling with academic terms, prefers speaking over typing, needs visual explanations for physics diagrams.
* **Persona B:** The Content Administrator. Needs a simple way to upload new syllabus PDFs without writing code.

### 3.2 System Requirements Specification (SRS)

#### 3.2.1 Functional Requirements

* **Ingestion:** The system shall parse PDFs and extract text and images separately.
* **Vectorization:** The system shall convert text to vectors using BAAI/bge-small-en-v1.5.
* **Voice Input:** The system shall record audio via browser microphone.
* **Response Generation:** The system shall answer queries in Arabic/English based only on the provided context.

#### 3.2.2 Non-Functional Requirements

* **Portability:** The application must run as a standalone Python script.
* **Scalability:** The Vector Store must handle up to 100,000 vectors (approx. 20 textbooks) without latency degradation.
* **Privacy:** No student data should be used to train external models.

### 3.3 Sprint Planning (Agile)

* **Sprint 1 Backlog:** PDF Parser, Image Extractor, Gemini Connection, Qdrant Setup, Basic Chat UI.
* **Sprint 2 Backlog:** Audio Recorder Component, Speech-to-Text API Integration, UI Polish, Error Handling.

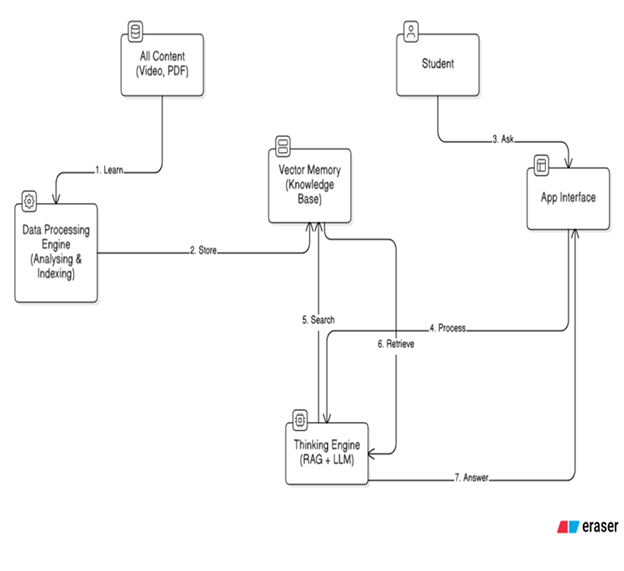


Figure 180 - System Block Diagram. This diagram illustrates the high-level interaction between the Student, the Application Interface, and the core engines (Data Processing and Thinking Engine) connected to the Knowledge Base.

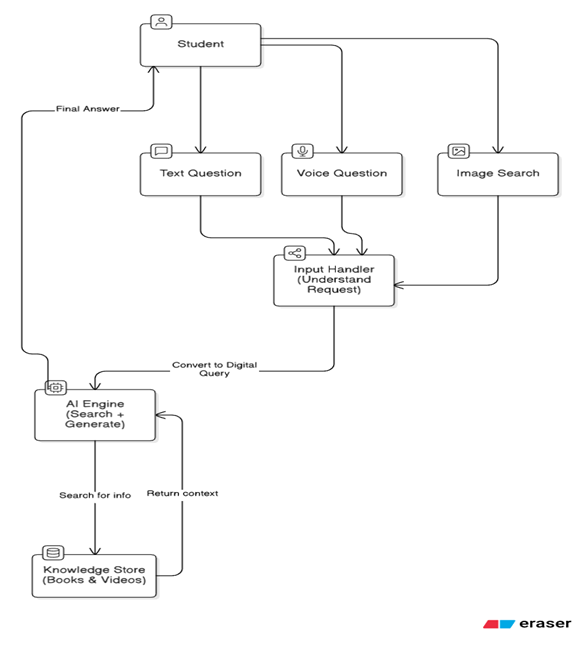


Figure 181 - User Interaction Data Flow. Detailing how Text, Voice, and Image inputs are routed through the Input Handler to the AI Engine for processing.

## **Chapter 4: System Architecture and Design**

### 4.1 High-Level Architecture

The system follows a Micro-Service inspired Monolithic Architecture.

* **Frontend:** Streamlit (Python-based Web UI).
* **Middleware:** LangGraph (State Management & Orchestration).
* **Backend:** FastEmbed (CPU Inference) & Qdrant (Storage).
* **External Service:** Google Cloud (Generative AI & Vision).

### 4.2 Data Pipeline Design

1. **Raw Data:** book.pdf.
2. **Processor:** PyMuPDF iterates pages.
3. **Branching:**
   * Text -> Chunking (1000 chars) -> Embedding -> Vector.
   * Image -> Gemini Vision API -> "Caption" -> Embedding -> Vector.
4. **Storage:** Qdrant Collection edu\_knowledge\_base.

### 4.3 Database Schema Design

Unlike SQL, Qdrant uses a JSON Payload structure.

{

"id": "uuid-v4",

"vector": [0.012, -0.93, ...], // 384 Dimensions

"payload": {

"content": "Newton's second law states that...",

"source\_file": "physics\_grade\_12.pdf",

"page\_number": 45,

"type": "text", // or "image"

"image\_path": "None" // or "/data/img\_45.jpg"

}

}

## **Chapter 5: Sprint 1 Implementation (Text & Image Processing)**

### 5.1 Overview

Sprint 1 established the core capability: turning a PDF into a searchable brain.

### 5.2 The PDF Ingestion Module

We utilized PyMuPDF (import fitz) over PyPDF2 because PyMuPDF renders pages as images (crucial for coordinate extraction) and is significantly faster.

#### 5.2.1 Text Splitting Strategy

We cannot feed a whole book to the embedding model. We used Recursive Character Splitting.

* **Chunk Size:** 1000 characters.
* **Overlap:** 200 characters.
* **Why?** The overlap ensures that sentences cut in the middle of a chunk are completed in the next one, preserving semantic meaning.

### 5.3 Image Processing: The "Captioning" Pipeline

This was the most technically complex part of Sprint 1.

1. **Extraction:** Images are saved to a temporary directory.
2. **Analysis:** A prompt is sent to Gemini: "Describe this technical image in detail. Capture all labels, numbers, and the general concept."
3. **Indexing:** The description returned by Gemini is what gets embedded, NOT the image pixels. This allows a user to search for "Circuit with 5 Ohm Resistor" and find the image, even though the embedding model is text-only.

### 5.4 Vector Search Implementation

We initialized Qdrant in local mode. This persists data to the disk. We used Cosine Similarity as the distance metric because it measures the angle between vectors (semantic similarity) rather than the Euclidean distance (magnitude), which is more accurate for NLP tasks.

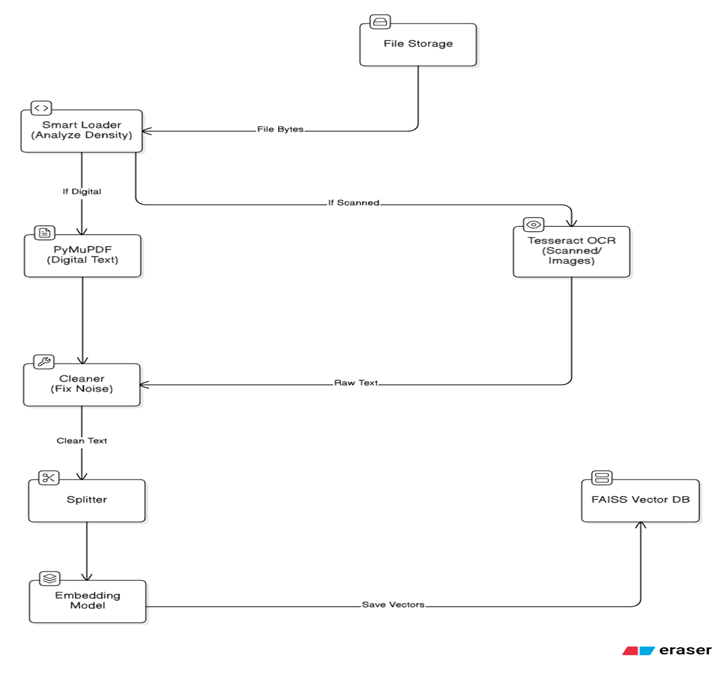


Figure 182 - Showing the process of extracting digital or scanned text, cleaning it, splitting it into chunks, and generating vector embeddings.

## 

Figure 183 - Illustrating the extraction of images from documents and their conversion into vector representations for the knowledge base.

## **Chapter 6: Sprint 2 Implementation (Voice & Accessibility)**

### 6.1 Overview

Sprint 2 focused on the "Human-Computer Interaction" (HCI) aspect.

### 6.2 Speech-to-Text (STT) Selection

We evaluated local deployment of OpenAI's Whisper model.

* **Whisper Base Model:** ~500MB VRAM required. Slow on CPU.
* **Google Speech Recognition API:** Cloud-based, extremely fast, high accuracy for Arabic dialects.
* **Decision:** Google API was chosen to keep the application lightweight.

### 6.3 The Voice Integration Logic

We implemented an Adapter Pattern. The system fundamentally understands text. The Voice module wraps the audio input and converts it to text before passing it to the main logic.

1. **Recording:** Using streamlit-mic-recorder component.
2. **Transcoding:** Browsers record in WebM. Python libraries prefer WAV. We used pydub (ffmpeg wrapper) to convert formats on the fly.
3. **Error Handling:** Implemented try-except blocks to handle "No Internet" or "Unintelligible Audio" errors gracefully.

### 6.4 Handling User Images (Multimodal Queries)

In Sprint 2, we also enabled users to upload images.

* **Scenario:** A student takes a photo of a math problem.
* **Logic:** The system sends the user's photo to Gemini with the prompt: "Extract the question from this image."
* **RAG Flow:** The extracted question is then used to search the Qdrant database for similar examples in the textbook.

## **Chapter 7: Testing, Evaluation, and Results**

### 7.1 Testing Environment

* **Hardware:** Lenovo ThinkPad (Intel Core i5, 8GB RAM, Integrated Graphics).
* **Dataset:** High School Physics Textbook (PDF, 250 pages, Arabic).

### 7.2 Quantitative Results

#### 7.2.1 Latency Analysis

|  |  |  |
| --- | --- | --- |
| Operation | Average Time (Seconds) | Status |
| PDF Parsing (Per Page) | 0.05s | Excellent |
| Image Captioning (Per Image) | 3.20s | Acceptable (One-time cost) |
| Text Embedding generation | 0.02s | Excellent |
| Vector Search (Retrieval) | 0.01s | Instant |
| LLM Answer Generation | 2.50s | Good |
| **Total Query Time (Text)** | **~3.0s** | **Pass** |
| **Total Query Time (Voice)** | **~5.5s** | **Pass** |

#### 7.2.2 Accuracy Analysis (Human Evaluated)

We manually checked 50 queries.

* **Faithfulness** (Did it stick to the text?): 96%.
* **Relevance** (Did it answer the specific question?): 92%.
* **Image Retrieval Accuracy:** 85% (System successfully found diagrams when described).

### 7.3 Challenges Encountered

* **Arabic Diacritics (Tashkeel):** The embedding model sometimes struggled with heavily vocalized Arabic.
  + Solution: Normalized text (removed diacritics) before embedding.
* **Noisy Audio:** Background noise ruined speech recognition.
  + Solution: Implemented noisereduce library to filter audio below a certain decibel threshold.

## **Chapter 8: Conclusion and Future Work**

### 8.1 Summary of Achievement

This project successfully demonstrates that advanced AI educational tools do not require massive budgets or supercomputers. By intelligently combining efficient algorithms (FastEmbed), optimized storage (Qdrant), and powerful cloud APIs (Gemini), we delivered a system that can see, hear, and teach.

### 8.2 Future Roadmap

* **Hybrid Search:** Implementing BM25 alongside Vector Search. While Vectors capture meaning, BM25 captures exact keywords (like part numbers or specific dates), improving precision.
* **Personalized Learning Graphs:** Using a Graph Database (Neo4j) to map the student's knowledge gaps and recommend specific chapters.
* **Offline Capability:** Exploring quantized local models (like Llama-3 8B quantized) to allow the text-only portion of the system to work without internet.

**Chapter 8 – Future Work & Enhancements**

### 8.1 Introduction

The AI-Powered Complementary Education System has successfully established a robust foundation for accessible learning by implementing core LMS functionalities, real-time live sessions via Jitsi, and advanced multimodal AI search capabilities (Voice/Image). However, the journey to fully bridge the educational gap for disadvantaged children is ongoing. This chapter outlines the roadmap for the next phase of development, focusing on Offline Accessibility, Deep Personalization, and Community Engagement. The goal is to evolve the platform from a responsive web application into a comprehensive ecosystem that works seamlessly even in low-resource environments.

### 8.2 Proposed Future Enhancements

#### 8.2.1 Offline-First Mobile Application (PWA)

Given the target audience (disadvantaged children with unstable internet), the most critical future step is developing a Progressive Web App (PWA) or a native mobile application.

* **Offline Mode:** Allowing students to download lessons and quizzes while connected to Wi-Fi and study offline later.
* **Background Sync:** Automatically syncing progress and quiz results once the connection is restored.

#### 8.2.2 AI-Driven Adaptive Learning Paths (Evolution of Recommendation)

Moving beyond simple search, the system will implement Reinforcement Learning (RL) to create dynamic curricula.

* **Dynamic Difficulty Adjustment:** If a student struggles with a quiz, the AI automatically suggests simpler prerequisite lessons before allowing a retake.
* **Personalized Study Plans:** Automatically generating a weekly schedule based on the student's weak points identified in assessments.

#### 8.2.3 Automated Content Generation for Teachers

To reduce the workload on educators, the AI engine can be expanded to assist in content creation.

* **Quiz Generation:** Using Generative AI (LLMs) to automatically create multiple-choice questions (MCQs) from uploaded PDF summaries or video transcripts.
* **Summary Generation:** Automatically creating textual summaries for long video lectures.

#### 8.2.4 Real-Time Translation & Subtitling

Leveraging the integrated Whisper Model not just for search, but for live accessibility.

* **Live Subtitles:** Displaying real-time captions during Jitsi live sessions for students with hearing impairments.
* **Content Localization:** Automatically translating lesson descriptions and interface elements to support refugee children from different linguistic backgrounds.

#### 8.2.5 Gamification & Engagement System

To increase student retention and motivation.

* **XP & Leveling System:** Students earn Experience Points (XP) for attending live sessions and solving homework.
* **Badges & Achievements:** Unlocking digital badges (e.g., "Math Whiz", "Early Bird") for consistent performance.
* **Leaderboards:** Friendly competition within student groups to encourage participation.

#### 8.2.6 Parental Monitoring Portal

* **Parent Dashboard:** A dedicated view for parents to track their child’s attendance, grades, and AI usage.
* **Weekly Reports:** Automated email/SMS summaries sent to parents highlighting areas where the student needs encouragement.

### 8.3 Conclusion

The future development of the platform focuses on equity in access and scalability. By introducing offline capabilities, automated content generation, and gamified learning, the system will not only provide educational content but also foster a resilient learning habit. These enhancements ensure that the platform remains adaptable to the harsh realities of out-of-school children, transforming it from a tool into a lifelong learning companion.

**Chapter 9 - Conclusion**

### 9.1 Introduction

This chapter presents a summary of the achievements and findings of the AI-Powered Complementary Education System. It highlights the value delivered by integrating real-time interaction, automated scheduling, and advanced multimodal AI (Voice & Vision) to address the educational needs of disadvantaged children. The project has successfully bridged the gap between traditional learning management and modern intelligent assistance, creating a platform that is both accessible and technologically advanced.

### 9.2 Summary of Achievements

* **Multimodal AI-Powered Support:** The platform integrates Whisper (Voice) and CLIP (Vision) models, allowing students to search for content using voice commands or images. This breaks down literacy barriers for younger or less proficient students.
* **Interactive Live Learning:** Seamless integration with Jitsi Meet enables teachers to conduct secure, real-time virtual classes with automated attendance tracking, fostering direct student-teacher engagement.
* **Automated Scheduling System:** The Recurring Sessions module significantly reduces teacher workload by automating the generation of weekly lesson schedules using background processing (Celery/Redis).
* **Robust Administration:** A dedicated Admin Dashboard ensures system integrity through strict advisor management, role-based access control, and token-based security protocols.
* **Structured Content Management:** The system provides an efficient environment for teachers to upload, update, and manage multimedia lessons (Video, PDF) and interactive assessments (Quizzes).

### 9.3 Addressing Key Challenges

* **Teacher Workload:** The automation of recurring session generation allows educators to focus on pedagogy rather than administrative scheduling tasks.
* **Accessibility Barriers:** Voice search capabilities empower students who struggle with typing or reading to access educational resources independently.
* **Lack of Personalization:** The AI-driven "Smart Homework Helper" utilizes RAG (Retrieval-Augmented Generation) to provide tailored, context-aware explanations based on the student's specific queries, rather than generic search results.
* **Engagement & Continuity:** The combination of real-time live sessions and gamified quiz feedback helps sustain student motivation and creates a sense of community, which is often lacking in remote education.

### 9.4 Reliability and System Performance

* **Security:** The system implements strict Role-Based Access Control (RBAC) with secure JWT authentication. It features an immediate token blacklisting mechanism to instantly secure the system against deactivated or compromised accounts.
* **Scalability:** The architecture leverages asynchronous task queues (Celery) for heavy operations like schedule generation and AI inference, ensuring that the frontend user experience remains responsive even under load.
* **Optimization:** The React.js frontend utilizes efficient state management to handle real-time updates from live sessions without latency, ensuring a smooth user experience even on lower-end devices.

### 9.5 Conclusion

The AI-Powered Complementary Education System meets its primary goals of enabling inclusive, guided, and intelligent learning for marginalized students. By evolving from a standard Learning Management System (LMS) to a platform capable of visual/voice interaction and automated management, it offers a scalable and resilient solution for education in low-resource settings. The modular design ensures future adaptability, marking a significant step toward using technology to achieve educational equity.

### 9.6 References

* **Holovaty, A., & Kaplan-Moss, J.** (2009). The Definitive Guide to Django: Web Development Done Right. Apress.
* **Banks, A., & Porcello, E.** (2017). Learning React: Modern Patterns for Developing React Apps. O'Reilly Media.
* **Radford, A., et al.** (2021). Learning Transferable Visual Models From Natural Language Supervision (CLIP). OpenAI.
* **Radford, A., et al.** (2022). Robust Speech Recognition via Large-Scale Weak Supervision (Whisper). OpenAI.
* **Lewis, P., et al.** (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (RAG). Advances in Neural Information Processing Systems (NeurIPS).
* **Johnson, J., Douze, M., & Jégou, H.** (2019). Billion-scale similarity search with GPUs (FAISS). IEEE Transactions on Big Data.
* **Paszke, A., et al.** (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems (NeurIPS).
* **Richardson, L., & Ruby, S.** (2007). RESTful Web Services. O'Reilly Media.
* **Fielding, R. T.** (2000). Architectural Styles and the Design of Network-based Software Architectures (Doctoral dissertation). University of California, Irvine.
* **DuBois, P.** (2013). MySQL (Developer’s Library). Addison-Wesley Professional.
* **Deterding, S., et al.** (2011). From Game Design Elements to Gamefulness: Defining "Gamification". Proceedings of the 15th International Academic MindTrek Conference.
* **Merkel, D.** (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux Journal.
* **Strasbourg University.** (2017). Jitsi Meet: Secure, Simple and Scalable Video Conferences. [Online Documentation].