Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

ARABIC LMS

WHERE AI MEETS EDUCATION

July 2025

Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Arabic Learning Management System (LMS) mobile application

By:

Nour Ehab Ragaa
Nouran Ayman Abd El-Aleem
Noureen Ashraf Aboarab
Yassmen Ashraf Raafat
Yosra Nabil Shafik
Youstina Ehab William

Under Supervision of:

Dr. Sally Saad,
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Mohammad Essam,
Bioinformatics Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Acknowledgement

All praise and gratitude to ALLAH, the Most Merciful, who has granted us the strength and wisdom to complete this work. May He accept our efforts and bless this endeavor.

We are deeply thankful to our parents and families for their unwavering love, sacrifices and encouragement throughout our academic journey. Their endless support has been our foundation and we aspire to repay their kindness in every way possible.

Our heartfelt appreciation goes to our supervisors, Dr. Sally Saad and TA Mohammad Essam, for their invaluable guidance, patience and expertise. Their insightful feedback and dedication were instrumental in shaping this project.

Lastly, we extend our thanks to our dear friends and everyone who offered support, motivation and kindness along the way. Your belief in us has been a constant source of strength.

Abstract

This project presents an AI-powered Learning Management System (LMS) mobile application designed to transform Arabic-language education by integrating advanced NLP capabilities. Recognizing the limitations of conventional LMS platforms—particularly their lack of AI-driven personalization and robust Arabic language support—this system introduces an intelligent framework that automates critical educational tasks while maintaining educational effectiveness.

The system fundamentally enhances educational workflows through two key capabilities: For teachers, it provides a comprehensive content management system allowing uploads of PDFs and other course materials, coupled with AI-powered automated generation of customizable quizzes and exams with adjustable difficulty levels. For students, the platform incorporates an AI-driven recommendation-based learning system that provides adaptive support in response to mistakes. This targeted feedback is supplemented by ongoing performance analysis that identifies weak areas and suggests personalized study resources. Throughout this process, teachers maintain complete oversight with real-time monitoring of student grades and progress, enabling timely intervention without manual reporting burdens.

To achieve these objectives, we experimented with Natural Language Processing (NLP) models, including AraT5, mT5, and AraBART, fine-tuned on multiple Arabic datasets such as Arabic-SQuADv2.0, Arabic-RACE, and a custom EKB dataset. These models enable question generation, answer evaluation, and distractor creation while maintaining linguistic and contextual accuracy in Arabic. Evaluation using standard NLP metrics, including BLEU, METEOR, and F1-score, confirms the system's capability to maintain high linguistic and contextual accuracy in Arabic educational content.

The final results demonstrate a fully functional AI-powered LMS that automates assessment creation, personalized learning experiences, and enhances educational efficiency for Arabic-speaking users. By reducing manual workload for educators and offering adaptive learning support for students, this system represents a significant advancement in AI-driven Arabic education technology.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
BLEU	Bilingual Evaluation Understudy
BERTScore	Bidirectional Encoder Representations from Transformers Score
CBF	Content-Based Filtering
CF	Collaborative Filtering
Colab	Google Colaboratory
EKB	Egyptian Knowledge Bank
EM	Exact Match
F1	F1 Score (harmonic mean of precision and recall)
FAISS	Facebook AI Similarity Search
FAST API	Fast Application Programming Interface
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
LMS	Learning Management System
MCQ	Multiple-Choice Question
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MRC	Machine Reading Comprehension
MSA	Modern Standard Arabic
NLP	Natural Language Processing
NLTK	Natural Language Toolkit

PDF	Portable Document Format
QA	Question Answering
QG	Question Generation
RACE	Reading Comprehension from Examinations
SQuAD	Stanford Question Answering Dataset
T5	Text-to-Text Transfer Transformer
UI	User Interface
UX	User Experience

Chapter 1

Introduction

Our project focuses on developing an AI-based learning management system (LMS) mobile application that supports learning in Arabic. The system caters to two types of end users: teachers and students. The teacher's role includes uploading educational materials, such as PDFs, and using the system's automated question generation feature to create assignments and exams. Teachers can also view their students' exam grades to assess their performance.

For students, the system provides the ability to take exams with questions intelligently tagged based on the curriculum. After completing exams, students can view their grades for each assessment. Additionally, the system recommends learning resources tailored to their weak areas, helping them improve their understanding of the subject matter.

By streamlining exam creation, grading, and performance review, this AI-powered LMS enhances the teaching and learning experience for both educators and students in Arabic-based education.

1.1 Motivation

The global shift toward digital learning has increased demand for efficient and adaptable educational tools. However, many existing learning management systems (LMS) lack tailored support for multilingual contexts, particularly for Arabic-speaking educators and students. These systems often overlook critical needs, such as streamlined exam creation and transparent performance feedback, placing unnecessary burdens on teachers and leaving students without clear insights into their academic progress.

To address these challenges, our project introduces an AI-powered LMS mobile application designed specifically for Arabic-language education. The system simplifies exam generation for teachers through automated question creation while providing immediate grade visibility for both instructors and students. By offering personalized resource recommendations based on exam performance, the system helps students target their weak areas without complex progress-tracking features.

This focused approach reduces administrative overhead for teachers and empowers students with direct feedback, fostering a more efficient and transparent learning environment. Our goal is to bridge the gap in Arabic-language educational technology by prioritizing practicality, clarity, and actionable insights.

1.2 Problem Definition

Current Learning Management Systems (LMSs) widely used in Arabic-speaking regions lack the advanced AI-driven features needed for effective, personalized learning. These platforms require extensive manual effort from educators to create and manage content, limiting adaptive, real-time support and targeted remediation for students. Additionally, while basic Arabic language support is often available, most LMSs fall short in delivering fully localized, culturally relevant experiences. This lack of automation and tailored resources hinders educators’ ability to address diverse student needs efficiently and leaves learners with limited opportunities for individualized, self-paced progress.

1.3 Objective

The aim of the project is to develop a mobile application that will support :

1. **Automatic Question Generation:** Develop an AI-powered feature that allows teachers to automatically generate questions for quizzes, assignments, and exams, with varying levels of difficulty, based on provided materials.

2. **Learning in Arabic:** Design the platform to support learning in Arabic, making it accessible to a diverse audience.

3. **Resource Recommendation:** The system recommends supplementary learning materials based on each student's exam performance, targeting their specific areas for improvement.

4. **Tag Questions Based on Curriculum:** Ensure that generated questions are tagged and categorized according to the curriculum.

5. **Provide Performance Insights for Teachers:** The system allows teachers to access students' exam results, helping them identify class trends and individual strengths or weaknesses. This enables educators to refine their teaching strategies and offer focused support where needed.

6. **Help Students Track Their Performance:** Students can review their exam grades. This immediate feedback helps learners self-assess their understanding, identify areas for improvement, and take ownership of their progress.

1.4 Time Plan

Figure 1.1 shows the overall project time plan. The time plan manifests the main stages of the project and their corresponding time span. The project process starts with the literature review through until building the final product. As seen, some tasks overlap. Finally, the project documentation is an ongoing process.

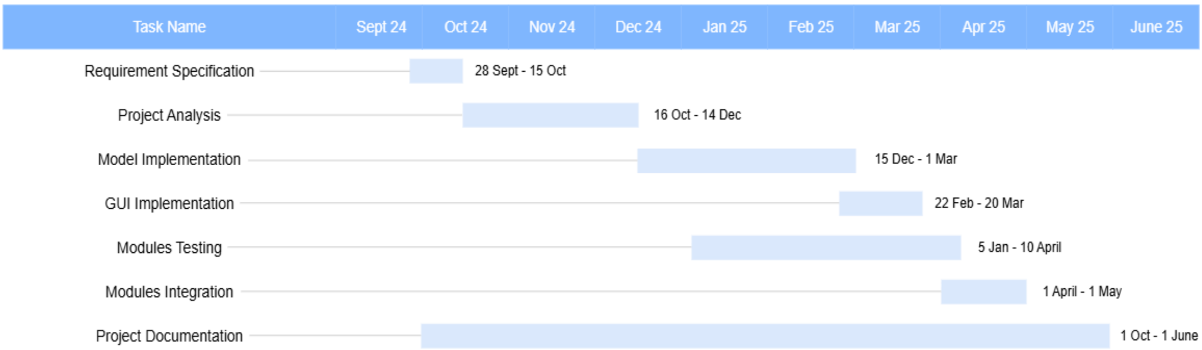


Figure 1.1: Time Plan

1.5 Document Organization

The Document is arranged as follows:

Chapter 2: Discusses the background of the project with respect to its scientific basis and its intended usage field, as well as a brief description of other similar projects.

Chapter 3: Discusses the three-tier system architecture with description to each of its components and the architecture of used deep learning models. The chapter also includes the system's intended users with their characteristics (basic knowledge required for the user to be able to benefit from the project).

Chapter 4: Discusses detailing the outcomes of implementing and testing the proposed LMS. It discusses the evaluation metrics, benchmarks against existing systems, and the performance of key features

Chapter 5: User Manual, providing detailed instructions on using the LMS effectively. It includes step-by-step guidance for both students and educators on navigating the platform.

Chapter 6: Concludes this documentation and suggests intended future improvements to be done in order to maximize the project’s potential.

Chapter 2

Literature Review

2.1 Project Overview

With the widespread adoption of smart devices and online learning platforms, many AI-powered Learning Management Systems (LMS) have been developed to enhance the teaching and learning experience. These platforms are designed to simplify the teaching process, support student engagement, and track performance. Examples of popular LMS platforms include **Moodle, Blackboard, Edmodo, and Classera**.

However, traditional LMS platforms often lack the ability to dynamically adapt to students' unique learning needs. They require educators to manually create and adjust learning paths and assessments, which can be time-consuming and restrictive. To overcome this, AI-powered LMS platforms employ advanced machine learning algorithms and natural language processing techniques to automate these processes.

The core objective of these AI-powered LMS platforms is to enhance learning and teaching experiences by using AI-driven features that automate tasks, provide feedback, and track performance. For **teachers**, AI supports the creation of automated quizzes, real-time grading, and insights into student engagement. For **students**, AI-powered systems offer learning paths, assessments, and targeted resources.

This chapter explores the various approaches used in AI-powered LMS platforms for question generation, topic mapping, and resource recommendation. It also discusses how these systems continuously track student progress using dashboards.

2.2 Scientific Background

Recent advances in **Artificial Intelligence (AI)** and **Natural Language Processing (NLP)** have revolutionized educational technology, enabling systems to perform complex tasks like automated question generation, answer evaluation, and personalized learning path creation. These innovations are particularly impactful in under-resourced languages like Arabic, where the lack of large-scale, high-quality datasets and pretrained models has historically limited development.

Natural Language Processing (NLP) and Question Generation : NLP focuses on enabling machines to understand, interpret, and generate human language. Question Generation (QG) is a core NLP task that involves creating meaningful questions from textual input. Traditional approaches relied on rule-based systems or shallow parsing techniques, which were language-dependent and brittle. Today, state-of-the-art models like AraT5 (an Arabic-specific adaptation of the T5 architecture), AraBART, and mT5 enable generative QG by leveraging transfer learning and massive pretraining.

Distractor Generation : For multiple-choice question (MCQ) formats, distractor generation is another key NLP task. It involves creating plausible but incorrect options that challenge students without misleading them unfairly. Unlike English, generating distractors in Arabic is more complex due to semantic ambiguity, root-based word formations, and context sensitivity.

Question Answering :

The Question Answering (QA) component of the system is designed to generate accurate, free-form answers based on a given Arabic context and an associated question. Unlike traditional multiple-choice or extractive QA systems, which select predefined answers or span text directly from the input, our approach uses **generative models** to produce human-like responses that are contextually relevant and grammatically correct.

Resource Recommendation Systems : When a student answers incorrectly, the LMS does more than provide the correct answer. It also activates a resource recommendation engine that suggests videos aligned with the underlying topic. Knowledge graphs and collaborative filtering methods are used to map student behavior to educational resources.

2.3 Related Work

This section discusses the review and analysis of relevant studies conducted to gain insights into advances, challenges, and potential solutions in the development of AI-based learning management platforms. Numerous researchers have explored approaches for automating questions, generating distractors, aligning resources with learning objectives, and providing tailored feedback. These advances have become especially relevant with the growing shift towards mobile and online learning platforms in Arabic-speaking communities.

2.3.1 Resources Recommendation System:

1- Video-Based Educational Content Recommendation:

Timmi et al. [1] and Zrigui et al. [2] established foundational approaches for video-based educational content recommendation. Timmi et al. proposed a hybrid approach combining collaborative filtering and content-based filtering for YouTube video recommendations using user engagement metrics and video metadata, while Zrigui et al. introduced topic modeling techniques for extracting themes from educational videos to align user interests with learning outcomes. Both studies demonstrate the effectiveness of leveraging video metadata and content analysis for personalized educational content delivery.

2- Document Retrieval and Multi-Modal Recommendation Systems:

Advancing beyond single-content-type approaches, Chan et al. [3] developed an orchestrated methodology that integrates content-based filtering with word semantic similarity and page ranking algorithms for learning resource recommendation. This multi-layered approach demonstrates how combining different recommendation techniques can enhance precision across diverse educational materials. Complementing this semantic approach, Challagundla et al. [4] addressed the scalability challenges of large-scale document systems by presenting an end-to-end neural embedding pipeline for PDF document retrieval using distributed FAISS and Sentence Transformer models, enabling efficient retrieval from extensive educational document collections.

Table 2.1: Resource recommendation system

Authors	Methodology	Dataset	Evaluation Approach	Reported Accuracy
Timmi et al. [1] **2024**	Hybrid CBF + CF recommendation model	YouTube video metadata (tags, views, likes)	User feedback and engagement metrics	Not explicitly reported
Zrigui et al. [2] **2023**	Topic modeling (text processing on video content)	Educational video metadata and topic tags	Comparison of topic relevance against user interest	High relevance in topic alignment, accuracy not specified
Chan et al. [3] **2014**	DISCO-style semantic similarity combined with PageRank-style weighting	Static text-based learning resources (~1,294 documents)	Similarity Score (0.5–0.8 average), Coverage (~100%)	~0.5–0.8 average similarity score
Challagundla et al. [4] **2024**	End-to-end neural embedding pipeline for large-scale PDF document retrieval using PyPDF2 for text extraction, FAISS for vector search, and Sentence Transformers for embeddings	Large-scale PDF documents (academic papers, reports, legal documents)	Cosine similarity, precision, recall, F1-score for retrieved sections	Similarity Score ~0.87 (average), precision and recall metrics evaluated

2.3.2 Question Generation:

Lafkiar et al. [5] proposed a Transformer-based QG model for Arabic, utilizing a custom dataset derived from Arabic-SQuAD and ARCD. The model achieved a BLEU-4 score of 20.51 and a METEOR score of 24.04, indicating its efficacy in generating context-relevant questions. Similarly, Alhashedi et al. [6] combined BERT-based transformers with TextRank for Arabic QG tasks using the mMARCO and TyDi QA Arabic datasets, achieving a BLEU-4 of 19.12 and a METEOR score of 23.00. Meanwhile, Nagoudi et al. [7] introduced AraT5, an encoder–decoder model pre-trained on Arabic and Twitter data, yielding a BLEU-4 of 16.99 across 96k QA instances.

Table 2.2: Question Generation

Authors	Methodology	Dataset	Reported Accuracy
Lafkiar et al. [5] **2024**	Transformer-based, sequence-to-sequence model for Arabic question generation	Custom dataset from Arabic-SQuAD and ARCD	BLEU-4: 20.51, METEOR: 24.04
Alhashedi et al. [6] **2022**	Transformer model (BERT-based) with TextRank for extraction	mMARCO and TyDi QA Arabic subsets	BLEU-4: 19.12, METEOR: 23.00
Nagoudi et al. [7] **2022**	AraT5 (T5-based encoder-decoder pre-trained on MSA + Twitter)	ARGENQG (96k QG examples from 5 QA datasets)	BLEU-4: 16.99

2.3.3 Question Answering:

For automated answering and assessment, recent works have leveraged fine-tuning of large-scale Arabic language models. Saja et al. [8] utilized AraT5 fine-tuning on AraSQuAD and GQA, achieving F1 scores of 0.883 and 0.770, respectively. Similarly, Afnan H. Alshehri [9] implemented a fine-tuned Arabic-BERT Large model for the TyDi QA dataset, achieving an F1 score of 71.6 and an Exact Match (EM) of 57.8, and Kholoud et al. [10] applied fine-tuning of AraBERTv0.2-large, reaching F1 and EM scores of 86.49 and 75.14, respectively.

Table 2.3: Question Answering

Authors	Methodology	Dataset	Reported Accuracy
Saja et al. [8] **2024**	Fine-tuning AraT5 model	AraSQuAD	Precision: 0.880, Recall: 0.895, F1: 0.883
Afnan H.[9] **Alshehri 2024**	Fine-tuned Arabic-BERT Large	TyDi QA	F1: 71.6, EM: 57.8
Kholoud et al. [10] **2021**	Fine-tuning AraBERTv0.2-large model	TyDiQA-GoldP	F1: 86.49, EM: 75.14

2.3.4 Similar Applications Comparison:

Current Learning Management Systems demonstrate significant limitations in both automated quiz generation and Arabic language support. While most platforms rely on manual processes with only emerging AI solutions, and existing Arabic support is limited to basic content storage with interface challenges, the proposed application addresses these gaps through integrated automation aligned with Arabic educational standards and comprehensive linguistic support.

Table 2.4: Similar Application Comparison

Feature	Moodle	Classera	Blackboard	Canvas	Google Classroom	Proposed App
Quiz Generation	❌ Requires Manual Setup	❌ static question banks	❌ manually created	✅ External Tool Integration	❌ No Automated Quiz Generation	✅ Aligned with Arabic standards
Automated Resources Recommendation	❌ Limited to instructor-provided content	❌ Manual sharing	❌ Manual sharing	❌ Mainly instructo-sourced	❌ Manual sharing, lacks advanced automation	✅ Based on student weak points
Language Support (Arabic)	✅ Arabic available, less	✅ Arabic available	✅ Offers Arabic	✅ Arabic supported	❌ Limited Arabic	✅ Full Arabic interface,

	cultural adaptation		language support	but not fully localized	support	uses arabic trained models

Chapter 3

Analysis and Design

3.1 System Architecture

Figure 3.1 visualizes the system architecture of the project. The system architecture is composed of three layers. The three-tier architecture is divided into Presentation, Logic and Data layers. Section 3.2 discusses each layer in detail.

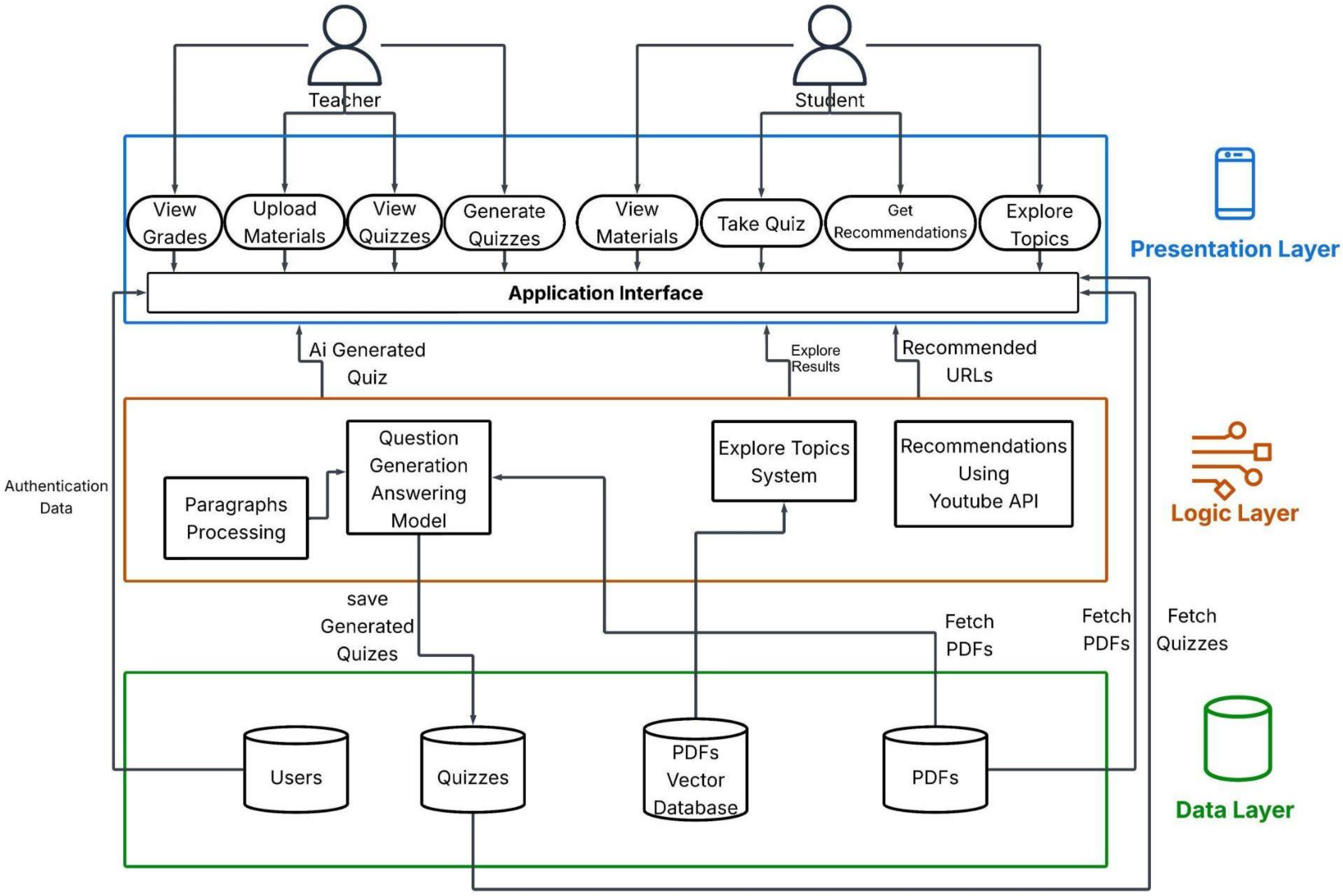


Figure 3.1: Three Tier System Architecture

3.2 Description of Methods and Procedures

1. **Presentation Layer**

    1.1. **Mobile Application (Teachers and Students):** This is the user-facing interface where teachers and students interact with the system. Through the mobile app, users can access personalized resources, track progress, and receive updates.

    1.2. **Authentication Service:** This module handles user authentication and ensures secure access. It verifies the identity of teachers and students and controls access to other parts of the system. Once authenticated, users are allowed to access specific content based on their roles (teacher or student).

    1.3. **Progress Management:** This component allows users to view and monitor learning progress, acting as a gateway for viewing updated performance data provided by the logic layer.

    1.4. **Material Management:** This module allows users to access, view, and interact with educational materials.

2. **Logic Layer**

    2.1. **Resources Recommendation System:** This model delivers personalized learning support by recommending YouTube educational videos and relevant PDF excerpts when students answer questions incorrectly. The system uses YouTube API integration for supplemental video content and employs vector-based semantic search with FAISS-indexed curriculum materials and Arabic Sentence Transformers to retrieve topically relevant textual resources, ensuring comprehensive remediation tailored to individual learning needs.

2.2. **Quiz Generation System** :Automatically creates complete multiple-choice assessments from uploaded course materials using Arabic NLP models for question generation, answer extraction, and distractor creation. This system enables teachers to transform PDF content into customized quizzes with adjustable difficulty levels, eliminating manual assessment creation while ensuring linguistically accurate and contextually appropriate Arabic educational content.

2.3. **Progress Tracking System**: Delivers real-time grade management by providing teachers with comprehensive class analytics (average, highest, lowest, and individual student grades) and giving students immediate feedback upon quiz completion. This system enables efficient performance monitoring and supports data-driven educational decision-making while maintaining detailed records for analysis and reporting.

3. **Data Layer**

3.1. **Users Info Database:** : Stores comprehensive user profiles including personal information (name, email, phone), authentication credentials, role assignments, and profile images. This database serves as the central user management system, enabling personalized access control and user identification across all platform features.

3.2. Assessment and Performance Tracking: The quiz_attempts table maintains detailed records of student quiz performance, including attempt dates, scores, time taken, and complete response data in JSON format. This enables the system to track learning progress, identify knowledge gaps, and provide personalized feedback based on individual performance history.

3.3. Educational Content Management:

- The qa_documents table stores question-answer pairs with contextual information and timestamps, serving as the foundation for quiz generation and knowledge assessment.
- The youtube_videos table maintains a curated collection of educational video content organized by topic and video ID, enabling multimedia learning support.
- The pdfs table manages uploaded educational materials and documents with metadata tracking, providing downloadable resources for comprehensive learning.

3.4. Quizzes: The quizzes table contains quiz templates and structures, including question-paragraph pairings stored in JSON format, linked to specific teachers. This table enables dynamic quiz creation and customization based on educational objectives and teacher preferences.

3.3 System Users

A. Intended Users:

1. **Teachers**:
   - Use the system to create assessments, monitor student performance, and access tailored teaching resources.
   - Rely on recommendations and insights to enhance teaching strategies and manage student learning.
2. **Students**:
   - Use the system to access personalized study materials, complete assignments, and track academic progress.
   - Benefit from targeted feedback and tailored practice opportunities to improve performance.

B. User Characteristics:

1. **Teachers**:
   - Basic familiarity with mobile apps and digital tools.
   - Ability to interpret data and use insights to guide teaching.
   - Skills to manage resources and assign personalized content.
2. **Students**:
   - Familiar with mobile apps for learning purposes.
   - Capable of interacting with quizzes, assignments, and recommendations.
   - Open to adapting study habits based on feedback and tracking progress.

3.4 Design Diagrams

3.4.1 Use Case Diagram

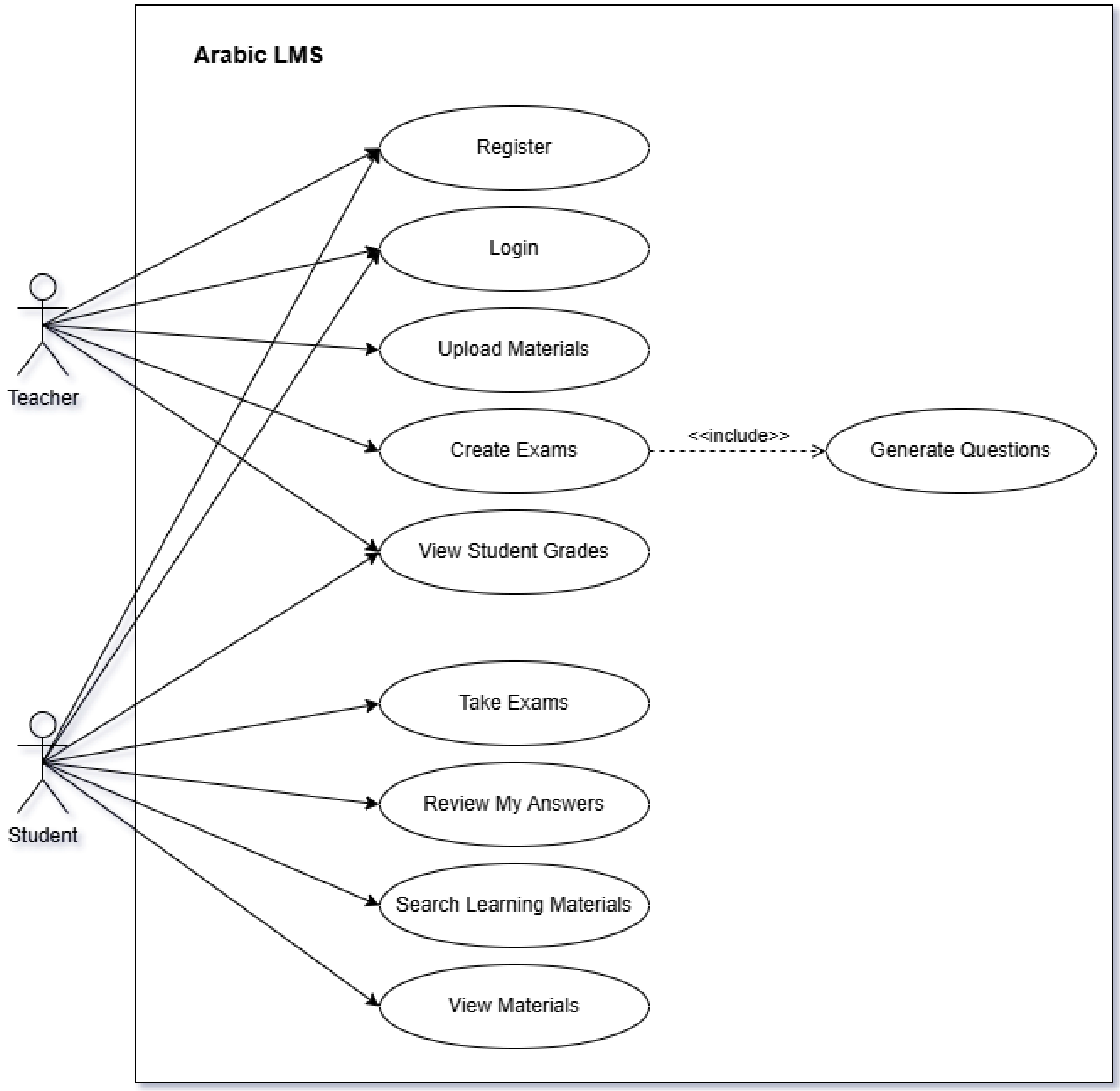


Figure 3.2: Use Case Diagram

3.4.1.1 Description of use cases

- Register
  New users (students or teachers) can create an account by selecting their role and entering the required details. Once registered, they gain access to the LMS platform.
- Login
  Registered users (teachers or students) log in with their credentials to access the LMS platform. After successful login, they are directed to a dashboard based on their role.

- Upload Materials
  Teachers can upload learning files PDFs, to the system. These materials are saved and organized, so students can access them anytime under the right subject or topic.

- Generate Questions
  Teachers can use AI models to automatically create questions from uploaded materials. These questions can be saved and reviewed, and later used in exam creation.

- Create Exams
  Teachers compile exams by selecting questions (from AI-generated or manually added ones), setting the duration and title, and publishing them for students to attempt.

- View Student Grades
  After students complete exams, the system grades them. Teachers can see overall performance and common weak points. Students can log in to view their scores and get feedback on each question.

- Take Exams
  Students choose an available exam and submit their answers through the app. Their responses are

saved and scored automatically or when the exam ends.

- Review My Answers

  Students get feedback after finishing an exam, especially for incorrect answers. The feedback can include explanations or links to helpful resources like videos.

- Search Learning Materials

  Students can search by keyword or topic name within the application. The system then displays all learning materials, such as sections from uploaded PDFs.

- View Materials

  Students can access all learning content uploaded by teachers. They can browse, open, and read files at their own pace for better understanding and review.

3.4.2 Sequence Diagram

3.4.2.1 Question Generation and Answer Evaluation

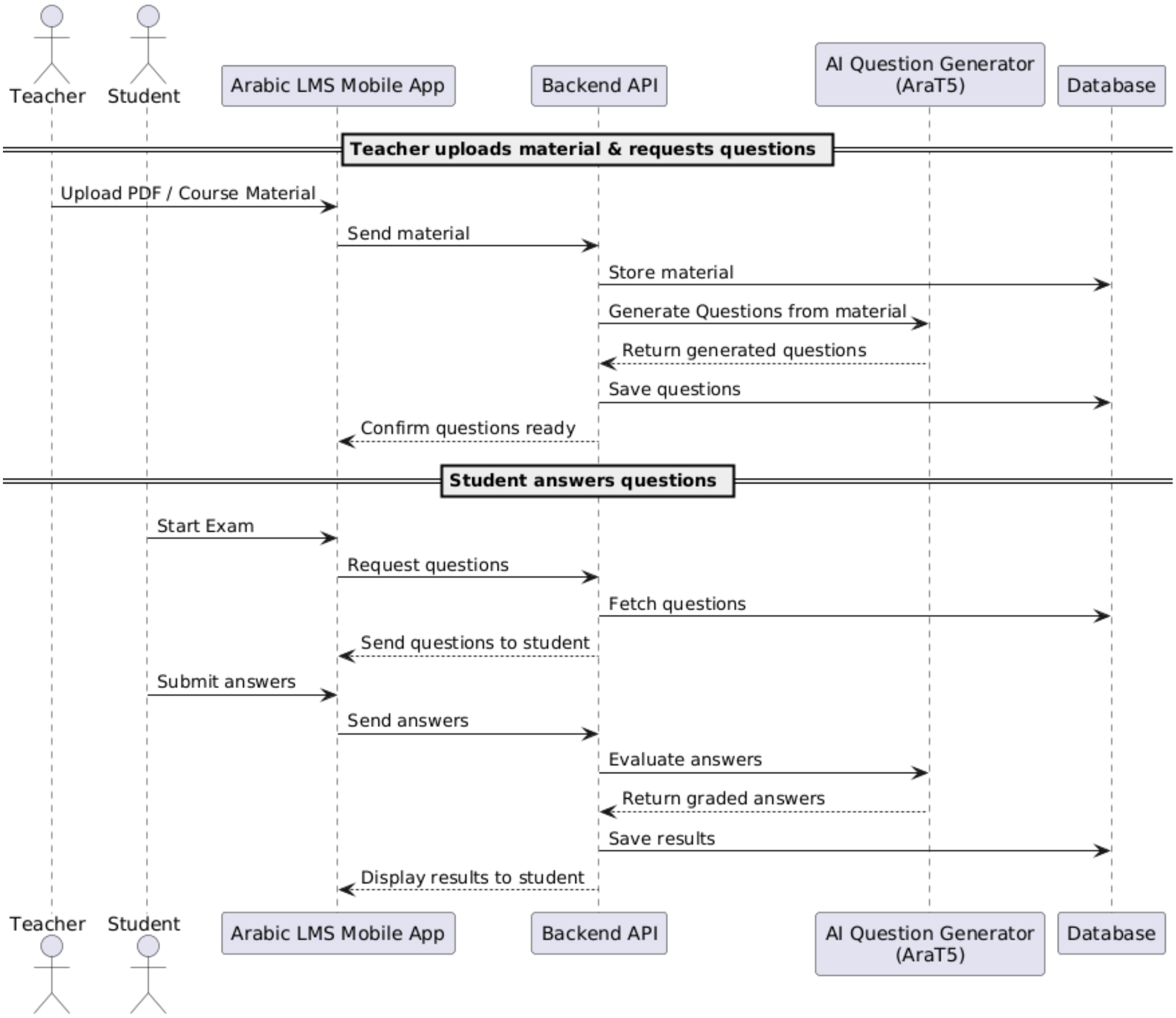


Figure 3.3: Question Generation and Answer Evaluation Sequence Diagram

3.4.3.2 Resource Recommendation System

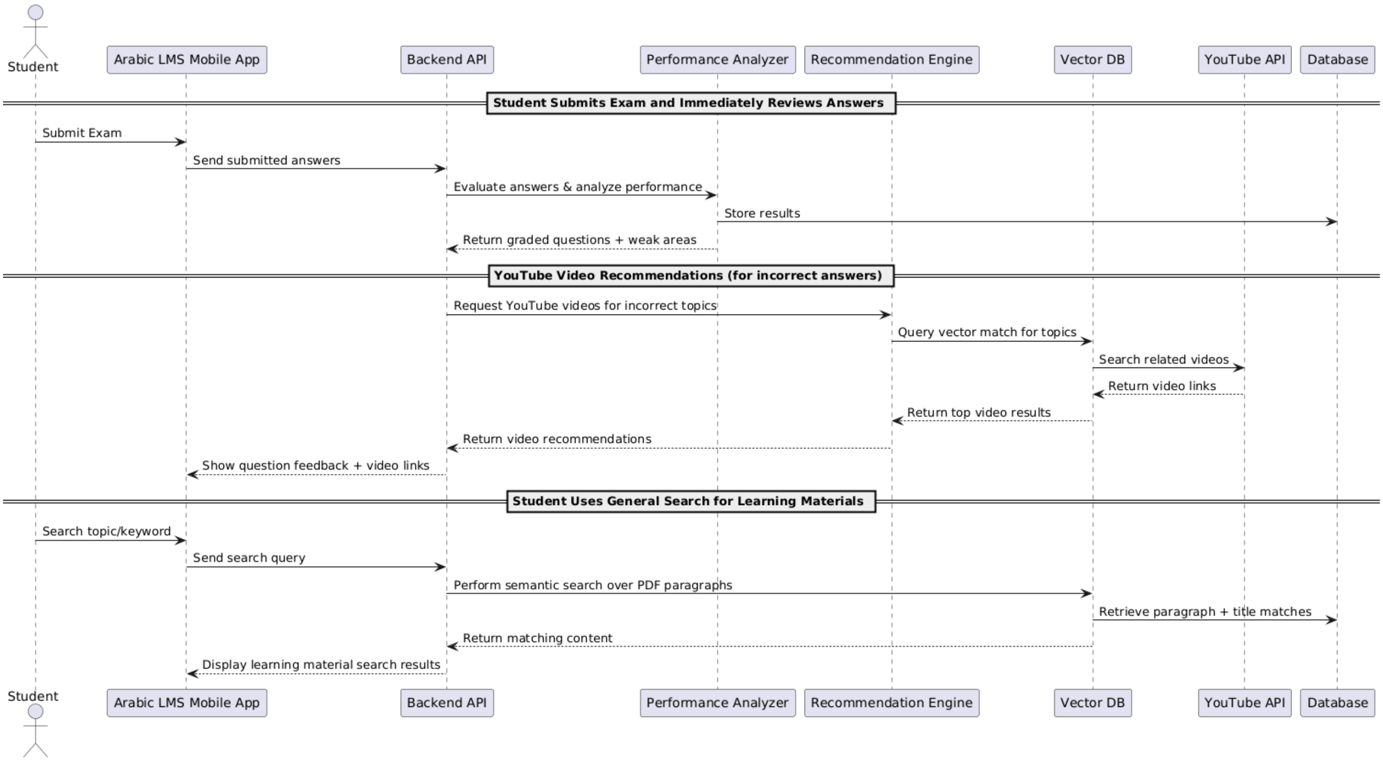


Figure 3.4: Resource Recommendation System Sequence Diagram

3.4.3 Database Diagram

Figure 3.5: Database Diagram

Chapter 4

Implementation and Testing

4.1 Dataset Description

Three datasets are used to train and evaluate different models in this project. Each dataset is explained below, outlining the data format, structure, and specific applications within the system architecture. The datasets are: **Custom EKB dataset**, **Arabic-RACE dataset**, and **Arabic-SQuADv2.0**.

4.1.1 Custom EKB Dataset

The Custom Egyptian Knowledge Bank (EKB) History Dataset is a specialized benchmark for Arabic machine reading comprehension, designed to evaluate models on domain-specific historical knowledge. The dataset was constructed by processing secondary-stage history lessons from the Egyptian Knowledge Bank, where each lesson was segmented into 947 paragraphs serving as contextual bases. For each paragraph, 6.26 questions on average were generated in Modern Standard Arabic, with 5,934 multiple-choice questions generated through a structured pipeline: GPT-4o was prompt-engineered to produce "Wh-" type questions (who, when, where) and answers, followed by the automated generation of semantically plausible distractors. Each question includes one correct answer and three distractors designed to test nuanced historical understanding. The dataset emphasizes curriculum-aligned historical knowledge, covering topics prescribed for Egyptian secondary education. To ensure robust evaluation, the data is split into training

(80%), validation (10%), and test (10%) sets, maintaining proportional topic representation across splits. Unlike general-purpose MRC datasets, this resource targets educational AI applications, enabling precise assessment of models' ability to comprehend and reason about Arabic historical texts.

4.1.2 Arabic-SQuADv2.0

The Arabic-SQuADv2.0 dataset is a large-scale, high-quality benchmark for Arabic machine reading comprehension (MRC), derived from the English SQuADv2.0 dataset. It comprises 96,051 triplets (question-context-answer pairs), created by translating SQuADv2.0's training split into Modern Standard Arabic using state-of-the-art neural machine translation (Microsoft Azure). Crucially, it preserves SQuADv2.0's key innovation: 45% of questions are unanswerable, while 55% are answerable. The dataset is partitioned into training (76,840 triplets, 80%), validation (9,605, 10%), and test (9,606, 10%) sets, with consistent answerable/unanswerable ratios across splits. Unlike prior Arabic adaptations (e.g., AQAD), it ensures context-answer alignment by translating full passages rather than relying on error-prone cross-lingual article matching. Format-compatible with SQuADv2.0, it enables standardized evaluation using Exact Match (EM) and F1-score metrics.

4.1.3 Arabic-RACE Dataset

The Arabic-RACE dataset is a linguistically adapted version of the RACE benchmark (Reading Comprehension from Examinations) for Arabic machine reading comprehension research. Derived from the original English RACE dataset (Lai et al., 2017), it comprises

27,933 passages and 97,867 multiple-choice questions translated into Modern Standard Arabic using the Large Dataset Translator tool. This translation pipeline preserves the dataset's core structure: passages (averaging 300-500 words) and their corresponding question-answer quadruplets (one correct option among four choices). The dataset retains RACE's signature reasoning-intensive focus, with >55% of questions requiring inference, contextual analysis, or logical deduction – capabilities now testable for Arabic NLP systems

4.2 Phases Description

This section presents the scientific and technical approaches used throughout the development of the Arabic Learning Management System (LMS), with a focus on the AI models and NLP techniques applied for question generation, question answering, and distractor creation.

4.2.1 Model Architecture

The system relies primarily on **AraT5-base-1024**, a Transformer-based encoder-decoder model specifically pre-trained for Arabic. AraT5 is based on Google's Text-to-Text Transfer Transformer (T5) framework, treating every NLP task as a text generation problem. This unified architecture supports:

- **Question Generation**: Input a paragraph and generate relevant questions.
- **Question Answering**: Input a paragraph and a question, generate a free-form answer.

- **Distractor Generation**: Generate multiple distractor choices for MCQs.

The model consists of:

- Approximately 220 million parameters.
- A SentencePiece tokenizer adapted for Arabic morphology and script.
- Encoder: Processes input contexts and instructions.
- Decoder: Generates text outputs based on the task.

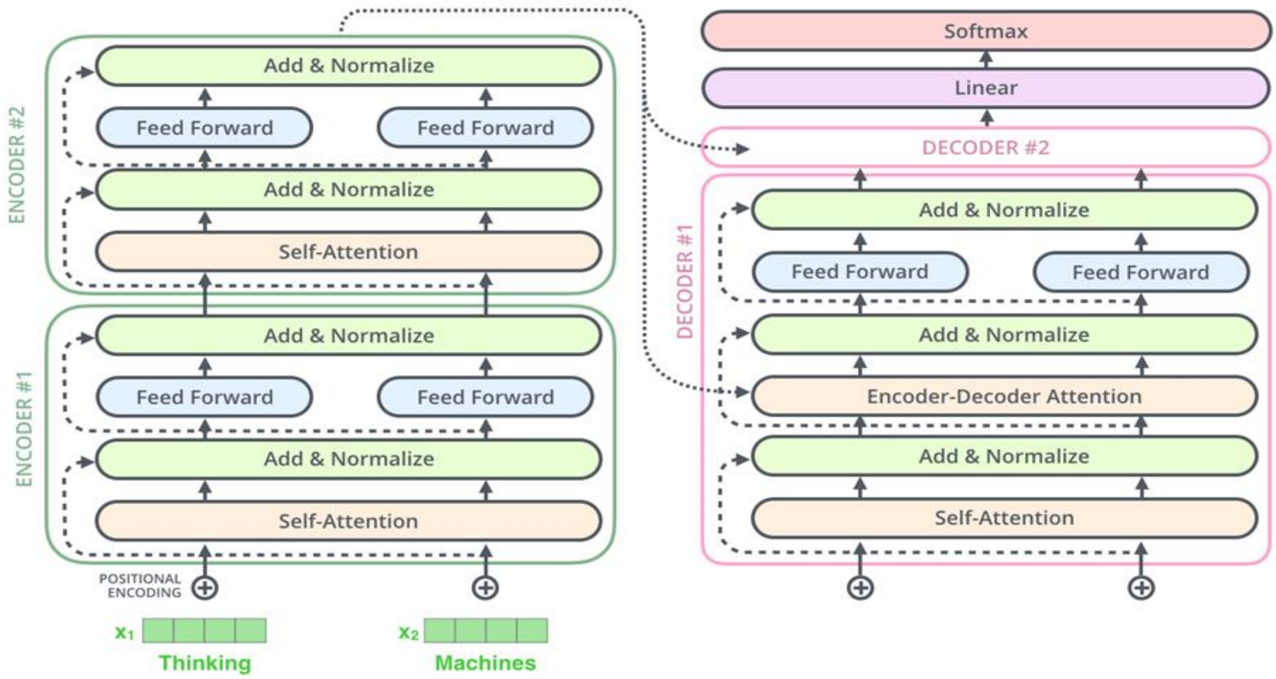


Figure 4.1: T5 Architecture

4.2.2 Dataset Preprocessing and Training

Three primary datasets were utilized:

- **Arabic-SQuADv2.0:** A translated version of SQuAD into Arabic, consisting of over 48,000 QA pairs. Used for fine-tuning both QA and QG tasks.
- **Custom EKB Dataset:** Scraped from Egyptian Knowledge Bank, containing domain-specific content with 5,000 QA pairs. Used to enhance relevance to curriculum-based materials.
- **Arabic-RACE Dataset:** Originally in English, this dataset was translated using a custom pipeline. Contains over 97,000 QA pairs with distractors for MCQ-focused tasks.

4.2.3 Task-Specific Model Fine-Tuning

a. Question Generation

- **Goal**: Generate meaningful and grammatically correct questions from Arabic educational paragraphs.
- **Input**: A paragraph or passage of Arabic text.
- **Output**: A relevant question targeting a key concept or fact in the paragraph.
- **Datasets Used**:
  - Arabic-SQuADv2.0
  - Custom EKB QA Pairs
- **Preprocessing**:
  - Removed overly short or vague answers.
  - Balanced question types (factual, definitional, inferential).

b. Question Answering

- **Goal**: Automatically answer questions based on a given context in Arabic.
- **Input**: A paragraph and a question.
- **Output**: A concise answer generated based on the context.
- **Datasets Used**:
  - Arabic-SQuADv2.0
  - Custom EKB QA Pairs
- **Approach**:
  - Fine-tuned the model using a generative approach.
  - Included multiple answer styles (single word, short phrases, complete sentences).

c. Distractor Generation

- **Goal:** Generate multiple incorrect but plausible options for MCQs.
- **Input**: A question and its correct answer with context.
- **Output**: 3 similar in theme but incorrect.
- **Datasets Used**:
  - Arabic-RACE Dataset.
- **Translation Strategy**:
  - Used "large_dataset_translator" tool for dataset translation.
  - Applied post-editing to fix awkward phrasing from automatic translation.

4.2.4 Resource Recommendation System

a. **YouTube API Integration:**

- **Goal:** Recommends supplemental videos based on
    - Triggers on incorrect exam answers
    - Topic relevance

b. **Vector-Based Topic Exploration:**

- Semantic search across curriculum materials:
    - FAISS-indexed dataset (EKB)
    - Sentence Transformers for Arabic query understanding
    - Returns PDF excerpts by topic relevance

4.3 Technologies & UI

- **Frontend & UI Development**

  Flutter: An open-source UI development toolkit by Google that allows for building natively compiled applications for mobile, web, and desktop from a single codebase.

  Dart: The programming language used with Flutter. Dart supports reactive programming and provides fast compilation and expressive syntax ideal for mobile UI development.

  Visual Studio Code: Used as the primary integrated development environment (IDE) for developing and testing the mobile application on Android devices and emulators.

- **Machine Learning & Backend**

Kaggle: An online platform for data science and machine learning that offers datasets, kernels (notebooks), and collaborative tools. Served as an environment for early model prototyping and sharing ideas among team members.

Google Colab: A cloud-based Jupyter notebook environment used for training and testing machine learning models, providing GPU acceleration and easy collaboration.

FAST API: A modern, fast (high-performance) web framework used for building APIs to integrate the machine learning backend with the Flutter frontend.

Python: The core programming language used for developing machine learning models, due to its extensive libraries and support for scientific computing.

In the project, we leverage certain frameworks and libraries of Python, such as:

- **Pandas**: A data manipulation and analysis library that provides flexible data structures such as DataFrames, used extensively to preprocess and explore datasets and transform features before feeding them into models.
- **NumPy**: Is the foundational library for numerical computing in Python, used to perform fast mathematical operations on arrays and matrices.
- **Sentence Transformers**: Sentence Transformers is a library built on top of Hugging Face's Transformers and PyTorch, used for generating dense vector representations of sentences.

- **Keras**: Keras is a high-level neural networks API that runs on top of TensorFlow (or other backend engines). It provides an easy-to-use interface for building and training neural networks.
- **PyTorch**: An open-source deep learning framework widely used for academic and industrial research, utilized for building custom deep neural network architectures.
- **scikit-learn**: Used for classical machine learning models, evaluation metrics, and feature selection.
- **NLTK (Natural Language Toolkit)**: Used for basic NLP preprocessing tasks such as tokenization, stemming, and stopword removal.

4.4 Experimental Results

- **Question Generation**

The results of fine-tuning multiple transformer-based models (AraT5, AraBART, and mT5) for the task of Arabic Question Generation. The goal of these experiments is to evaluate the models' performance using various configurations and compare their effectiveness based on standardized NLP metrics.

Each experiment utilized the custom EKB dataset, a domain-specific Arabic educational corpus. The dataset was split into 80% for training and 20% for evaluation to ensure reliable performance measurement. Key hyperparameters varied across runs include learning rate, number of epochs, gradient accumulation steps, and optimizer settings. Evaluation was conducted using metrics such as BERTScore, BLEU-4, METEOR, and F1-Score to assess the fluency, relevance, and semantic accuracy of the generated questions.

Table 4.1: QG AraT5 Model Performance on EKB Dataset

Parameters	**BERTScore**	**BLEU-4**	**F1-Score**	**METEOR**
Learning_rate: 3e-5 Per_device_train_batch_size: 4 Gradient_accumulation_steps: 4 Num_train_epochs: 10 Optimizer: AdamW Scheduler: OneCycleLR	0.86851	0.21769	0.50950	0.45444
Learning_rate: 3e-5 Per_device_train_batch_size: 4 Num_train_epochs: 10	0.86463	0.20189	0.50357	0.45663
Learning_rate: 3e-5 Per_device_train_batch_size: 4	0.86878	0.22200	0.52109	0.45977

Gradient_accumulation_steps: 8 Num_train_epochs: 15 Optimizer: AdamW Scheduler: OneCycleLR				
Learning_rate: 3e-5 Per_device_train_batch_size: 4 Gradient_accumulation_steps: 8 Num_train_epochs: 30 Optimizer: AdamW Scheduler: OneCycleLR	**0.87333**	**0.21653**	**0.53226**	**0.47012**
Learning_rate: 5e-5 Per_device_train_batch_size: 4 Gradient_accumulation_steps: 4 Num_train_epochs: 50 Optimizer: AdamW Scheduler: OneCycleLR	0.86526	0.19895	0.49481	0.43609
Learning_rate: 3e-5 Per_device_train_batch_size: 4 Gradient_accumulation_steps: 8 Num_train_epochs: 50 Optimizer: AdamW Scheduler: OneCycleLR	**0.87458**	**0.23949**	**0.52661**	**0.45823**

Table 4.2: QG AraBART Model Performance on EKB Dataset

Parameters	**BERTScore**	**BLEU-4**	**F1-Score**	**METEOR**
learning_rate=3e-5 per_device_train_batch_size=4 num_train_epochs=10	0.85905	0.21653	0.46965	0.38785
Learning_rate: 3e-5 Per_device_train_batch_size: 4 Gradient_accumulation_steps: 8 Num_train_epochs: 15 Optimizer: AdamW Scheduler: OneCycleLR	0.85561	0.15670	0.47414	0.37753

Learning_rate: 3e-5 Per_device_train_batch_size: 4 Gradient_accumulation_steps: 8 Num_train_epochs: 30 Optimizer: AdamW Scheduler: OneCycleLR	0.84893	0.15874	0.44985	0.35821
Learning_rate: 3e-5 Per_device_train_batch_size: 4 GradieEKB Custom Datant_accumulation_steps: 4 Num_train_epochs: 15 Optimizer: AdamW Scheduler: OneCycleLR	0.85000	0.14528	0.44881	0.36643

Table 4.3: QG mt5 Model Performance on EKB Dataset

Parameters	**BERTScore**	**BLEU-4**	**F1-Score**	**METEOR**
learning_rate=3e-5 num_train_epochs=12 Per_device_train_batch_size: 2	0.83607	0.17535	0.46840	0.39472

The experimental results clearly demonstrate the effectiveness of the **AraT5 model** for Arabic question generation across key evaluation metrics, including BLEU-4, METEOR, F1-score, and BERTScore. Among all tested configurations, AraT5 trained with 8 gradient accumulation steps over 50 epochs achieved the highest **BLEU-4 score (0.23949)**, while training over 30 epochs led to the highest **METEOR score (0.47012)**. These outcomes highlight AraT5's strong ability to generate fluent, semantically relevant Arabic questions.

Overall performance was notably improved by adjusting gradient accumulation and increasing training epochs, allowing the model to generalize better over the dataset without requiring a larger batch size.

While AraBART, mT5 also delivered acceptable results, it consistently underperformed compared to AraT5 in key metrics.

These findings validate the selection of AraT5 as the most suitable model for deployment in the LMS system. They also provide practical insights into effective hyperparameter tuning for future Arabic-language educational NLP applications.

- **Question Answering**

This section presents the results of fine-tuning Arabic transformer models for the Question Answering (QA) task using both Arabic-SQuADv2.0 and the domain-specific EKB dataset.
The EKB dataset, used throughout these experiments, contains longer and more detailed answers, which naturally impacts evaluation scores compared to shorter answer benchmarks. Despite this, the models demonstrate reliable performance under practical conditions relevant to educational applications.

Table 4.4: QA AraT5 Model Performance on Arabic-SQuADv2.0 Dataset

Parameters	**precision**	**Recall**	**F1-Score**
learning_rate=3e-5 num_train_epochs=12 gradient_accumulation_steps=4, per_device_train_batch_size=4 train_dataset=20k rows	**0.9037**	**0.9121**	**0.8890**
learning_rate=3e-5 num_train_epochs=15 gradient_accumulation_steps=4, per_device_train_batch_size=4 train_dataset=10k rows	0.8006	0.8647	0.8038

Table 4.5: QA AraT5 Model Performance on EKB Dataset

Parameters	precision	Recall	F1-Score
learning_rate=3e-5 num_train_epochs=15 gradient_accumulation_steps=4, per_device_train_batch_size=4	0.7634	0.7778	0.7387

Table 4.6: QA mt5 Model Performance on EKB Dataset

Parameters	precision	Recall	F1-Score
learning_rate=3e-5 num_train_epochs=20 gradient_accumulation_steps=4, per_device_train_batch_size=4	0.6434	0.7402	0.6688

Table 4.7: QA AraBART Model Performance on EKB Dataset

Parameters	precision	Recall	F1-Score
learning_rate=3e-5 num_train_epochs=20 gradient_accumulation_steps=4, per_device_train_batch_size=4	0.7137	0.7521	0.7056

As shown in **Table 4.4**, the **Arabic-SQuADv2.0** dataset on **AraT5** achieved significantly higher scores, with an **F1-Score of 0.8890**, highlighting the benefits of clean, well-structured benchmark data. On the **EKB dataset**, despite the longer and more complex answer spans, **AraT5** maintained strong performance with an **F1-Score of 0.7387** as shown in **Table 4.5**, demonstrating its adaptability to real-world educational content and robustness in handling extended answers.

These experiments confirm **AraT5** as the most effective model for Arabic QA within the LMS system, balancing accuracy with generalization.

- **Distractor Generation**

This section presents the results of fine-tuning AraT5 for the Distractor Generation task, experimenting with both the Arabic-RACE and the custom EKB datasets. Unlike the other tasks where multiple models were evaluated, distractor generation was exclusively performed using the AraT5 model due to its superior performance in preliminary Arabic text generation tasks. The evaluation was conducted on two distinct datasets: the Arabic-RACE dataset, which is a translated version of the original RACE dataset adapted for Arabic language processing, and our custom EKB dataset specifically curated for Arabic educational content. The model performance was assessed using three standard evaluation metrics: F1-score, BLEU, and METEOR to measure the quality and accuracy of generated distractors.

Table 4.8: Distractor Generation AraT5 Model Performance

Dataset	F1	BLEU	METEOR
Arabic-Race	**0.6667**	**0.4659**	**0.6543**
EKB Dataset	0.5556	0.2765	0.5355

As shown in **Table 4.8**, the **AraT5** model achieved notably better performance on the **Arabic-RACE** dataset across all evaluation metrics, with **F1-score of 0.6667**, **BLEU score of 0.4659**, and **METEOR score of 0.6543**, compared to the EKB dataset. This performance difference can be attributed to the Arabic-RACE dataset's larger size and more diverse question types, which provided richer training examples for distractor generation patterns. The consistently higher scores across all metrics for Arabic-RACE demonstrate the importance of dataset quality and diversity in training effective distractor generation models for Arabic educational content.

To complement the automated metrics and provide a more comprehensive assessment of distractor effectiveness, a human evaluation survey was conducted with Arabic-speaking experts. Participants evaluated generated distractors using a 5-point Likert scale across three key criteria: (1) plausibility - "Could each distractor reasonably be the answer?", (2) relevance - "Do they fit the question?", and (3) distinctiveness - "Are distractors meaningfully different?". The survey results showed that Arabic-RACE-trained distractors consistently outperformed EKB-trained distractors across all criteria, with average ratings of [**3.72 vs 3.72** for **plausibility**, **4.4 vs 3.92** for **relevance**, and **3.28 vs 3.2** for **distinctiveness**]. This human evaluation confirms that the superior automated metric performance translates to genuinely higher quality distractors that are more suitable for practical educational use.

- **Resource Recommendation System:**

The resource recommendation system was implemented and tested in two distinct setups:
The first setup used the YouTube API, with 105 queries issued to retrieve relevant video resources. The second setup leveraged a FAISS Vector Database built from an EKB Custom Dataset, with 951 queries issued to measure retrieval quality.

In both setups, the system utilized various multilingual Sentence Transformer models, and performance was evaluated using the average similarity score across all queries. This approach allowed for a direct and comparative assessment of how effectively each model captured semantic relationships and recommended relevant resources.

- **Using YouTube API:**

Table 4.9: Resource Recommendation using YouTube API

Model	Accuracy
Multilingual-E5-large-instruct	87.18%
Multilingual-E5-large	83.50%
paraphrase-multilingual-mpnet-base-v2	62.63%
paraphrase-multilingual-MiniLM-L12-v2	55.55%
distiluse-base-multilingual-cased-v2	40.18%

- **Using FAISS Vector Database:**

Table 4.10: Resource Recommendation on EKB Custom Dataset using FAISS Vector Database

Model	Accuracy
Multilingual-E5-large-instruct	87.46%
Multilingual-E5-large	84.59%
paraphrase-multilingual-MiniLM-L12-v2	72.43%
paraphrase-multilingual-mpnet-base-v2	67.34%
distiluse-base-multilingual-cased-v2	42.70%

The results reveal that the Multilingual-E5-large-instruct model achieved the best performance across both the YouTube API and the FAISS Vector Database, yielding an average similarity score of (87.18%) and (87.46%) respectively.

The model significantly outperformed other embeddings such as Multilingual-E5-large, paraphrase-multilingual-mpnet-base-v2, and distiluse-base-multilingual-cased-v2, making it the preferred choice for Arabic resource recommendation. These findings underscore the efficacy of advanced multilingual embeddings in providing accurate, context-aware recommendations, aligning well with the requirements of an Arabic-focused learning management system.

Chapter 5

User Manual

This chapter is a walkthrough of the whole application.

5.1 Welcome Screens

- **Splash Screen**
  Displays the app logo, the slogan and loads the application.

Figure 5.1: Splash Screen

- **OnBoarding Screens**

  When you open the app for the first time, you will go through the following screens:
  A brief introduction to the app’s features. Swipe up or click the button to navigate through them.

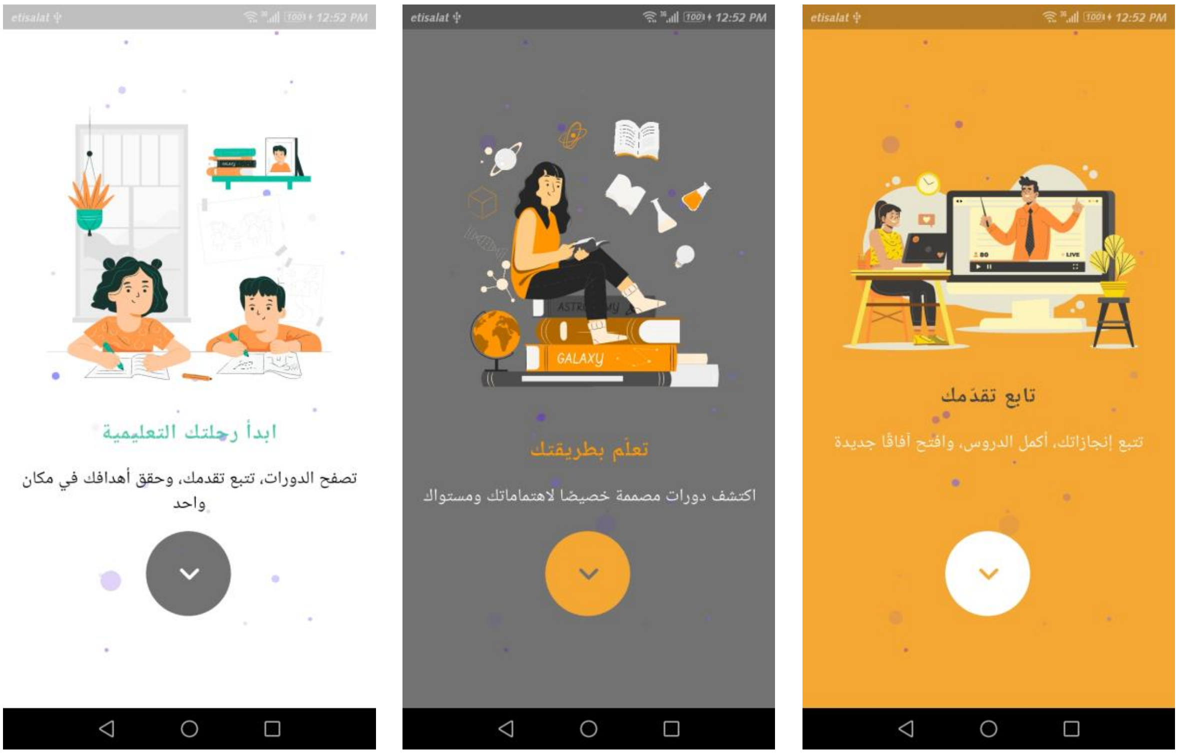


Figure 5.2: OnBoarding Screens

- **Start Screen**

  The start screen allows the user to choose between signing up to create a new account or logging in if they already have an account. The user can navigate to the Sign-up or Sign-in screens from this screen.

Figure 5.3: Start Screen

5.2 Authentication Screens

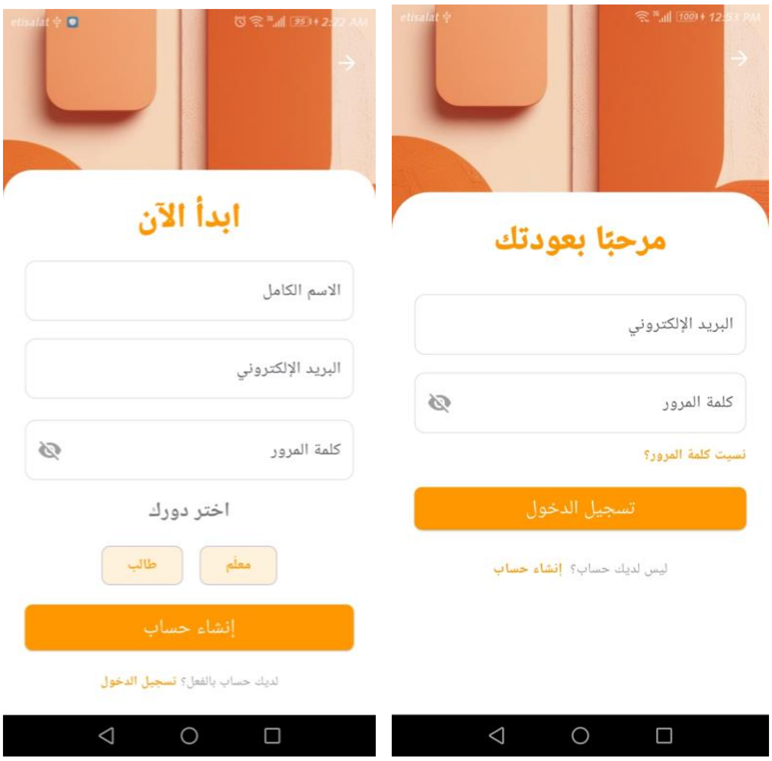


Figure 5.4: Authentication Screens

- **Sign Up Screen**

  This screen allows new users to create an account by entering their full name, email, password and role, the user is then navigated to the home screen.

- **Sign In Screen**

  This screen allows existing users to authenticate and access their account. After successful login, the user is navigated to the home screen.

5.3 Teacher Screens

- **Home Screen**
  - Navigation options
    - Profile
    - Upload Material
    - Generate Quiz
    - View Generated Quizzes
    - View Students Grades

Figure 5.5: Teacher Home Screen

Figure 5.6: Teacher Upload Screen

- **Upload Material**
  - Tap Upload Material from the home screen.
  - Enter the Title, Description and Tags of the material.
  - Attach a PDF file.
  - Tap Upload to save the material.

- **Generate Quiz**
  - Tap Generate Quiz from the home screen.
  - Choose a PDF Source:
    - Upload from Device
    - Use an Already Uploaded PDF
  - Enter Quiz Details:
    - Insert Quiz Title
  - Select Number of Questions (3–20).
  - Tap Generate to create your quiz.

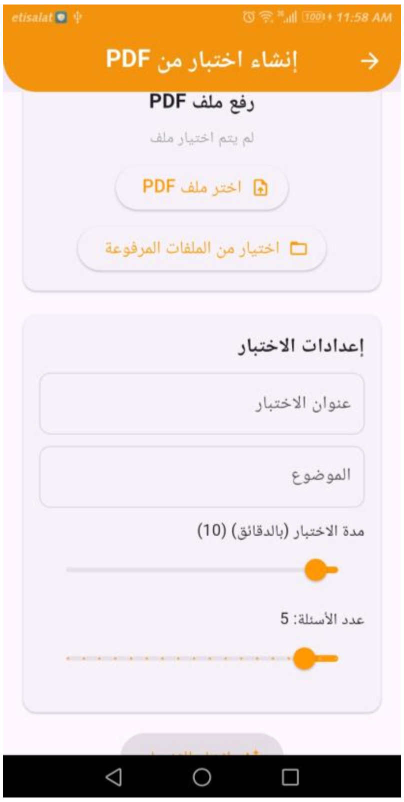


Figure 5.7.1: Teacher Generate Quiz Screen

Figure 5.7.2: Teacher Chooses from Existed PDFs Screen

- **View Generated Quizzes**
    - Displays a list of all created quizzes.
    - Tap on a quiz to view.
    - Tap on Basket to Delete.

Figure 5.8: Teacher View Quiz Screen

- **View Students Grades**

  Teachers can review student performance for each quiz.

  - Select a Quiz from the dropdown menu (lists all generated quizzes).
  - The app displays:
    - List of Students who took the quiz, along with their names and scores.
    - Performance Summary at the top:
      - Highest Score (Top performer)
      - Average Score (Class average)
      - Lowest Score (Weakest performer)

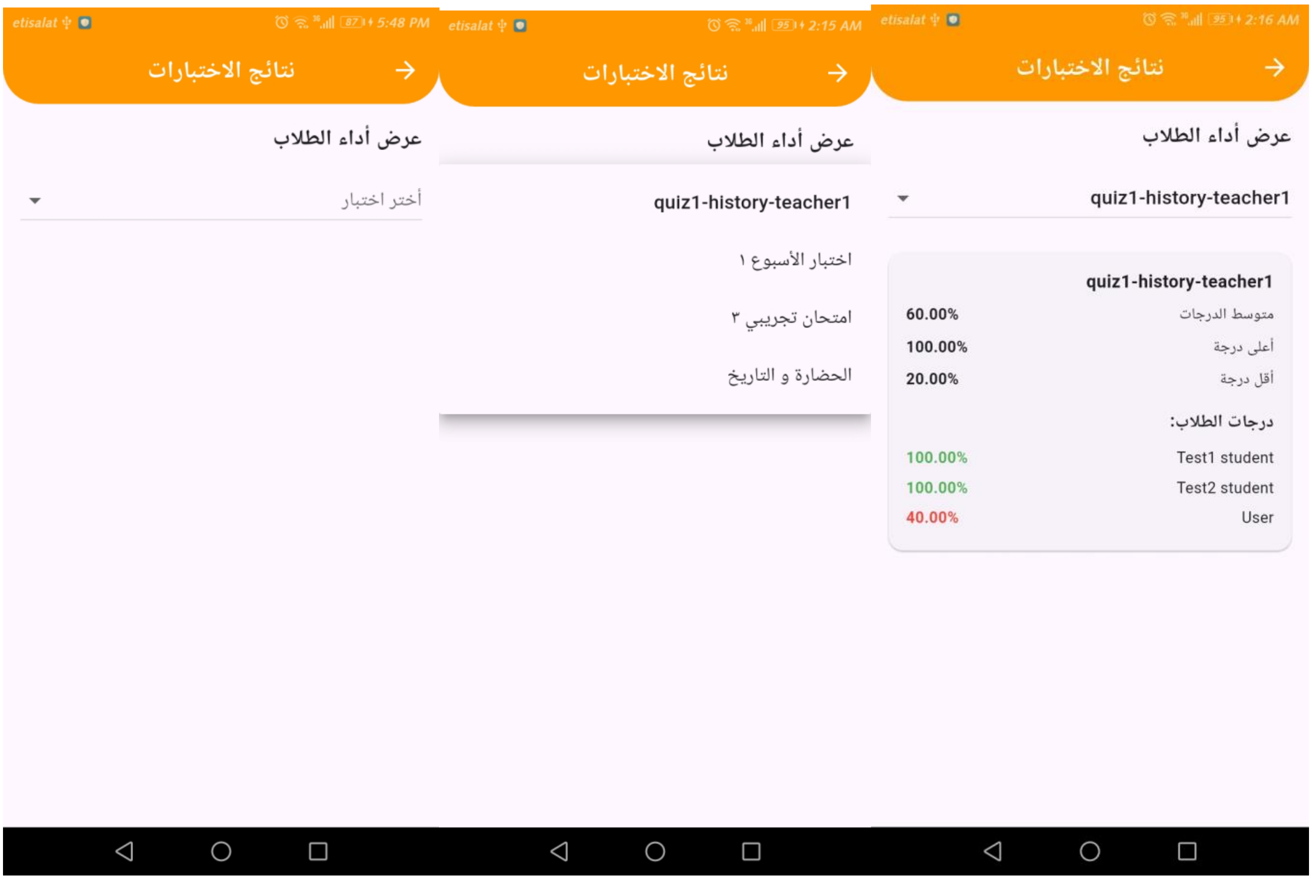


Figure 5.9: Teacher View Grades Screens

- **Profile Screen**
    - View and edit your profile information.
    - Change Language.
    - Log out.

Figure 5.10: Teacher Profile Screens

5.4 Student Screens

- **Home Screen**

    - Key Sections:
        - Student Progress.
        - Quiz Results – View past quiz performance.
        - Download Materials – Download any material pdf.
        - Upcoming Tasks – Pending quizzes.
    - Quick Navigation:
        - Profile – Access account settings.
        - Main Subjects – Browse course materials.
        - Home Page.

Figure 5.11: Student Home Screen

Figure 5.12: Student Subjects Screen

- **Access Study Materials**
    - Tap on any subject to view its materials.

- **Subject Details Screen**
  - Organized into 2 main sections:
    - Educational Materials
      - PDF resources for lessons
    - Exams
  - Tap on Download Button to Download lesson PDF.
  - Tap on Exam/Quiz to Begin.

Figure 5.13: Student Subject Screen

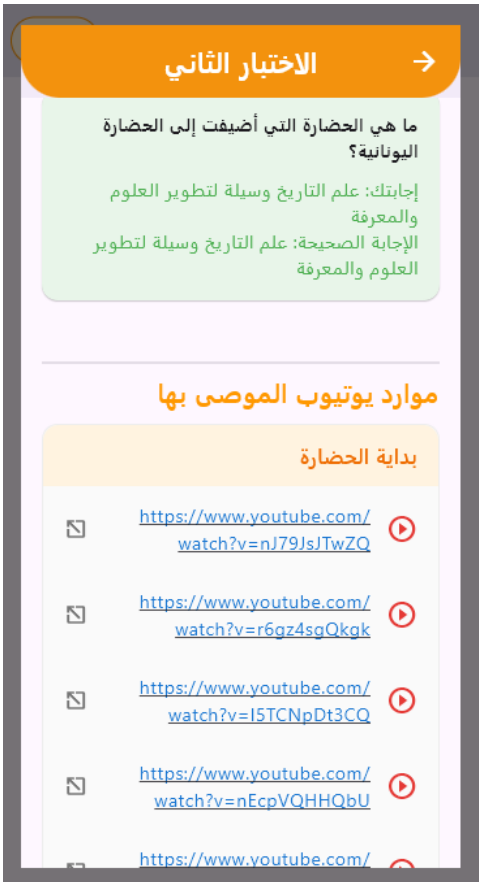
- **Take Quiz**
    - Tap Take Quiz from the home screen.
    - Select a quiz from the list.
    - Read each question and select the correct answer.
    - Tap Submit when finished to see your score.

Figure 5.14: Student Take Quiz Screens

    - Get Youtube videos URLs recommendations if score <= 90%.

- **Quizzes Results**

  Students can check their performance on past quizzes:



  - Tap "Quiz Results" from the home screen.
  - A list of all attempted quizzes appears, showing:
    - Quiz Title
    - Grade
  - Tap on a quiz to view detailed results:
    - Your Score
    - Correct & Incorrect Answers
  - Close the window to return to the grades list.

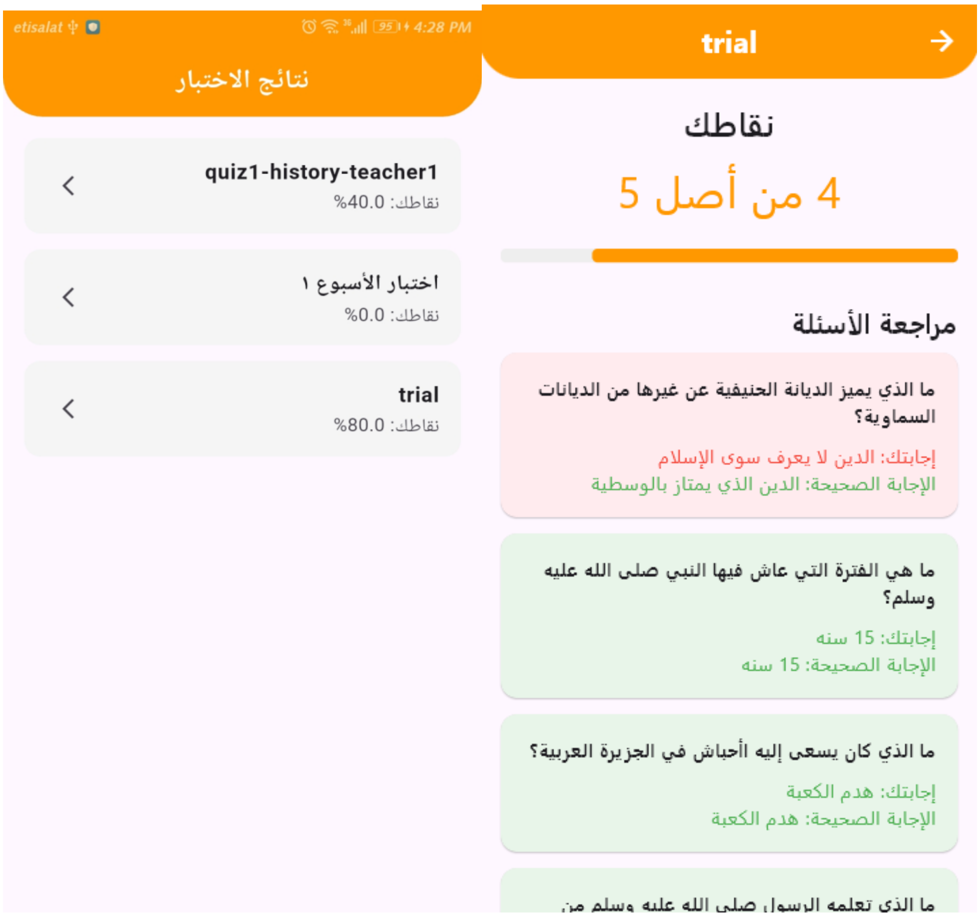


Figure 5.15: Student View Quizzes Results Screens

- **Explore Screen**
  - Tap Explore from the home screen.
  - Search for a specific topic using the search bar.
  - Tap on a title to view its content.
  - Scroll to read all content or return to the topic list.

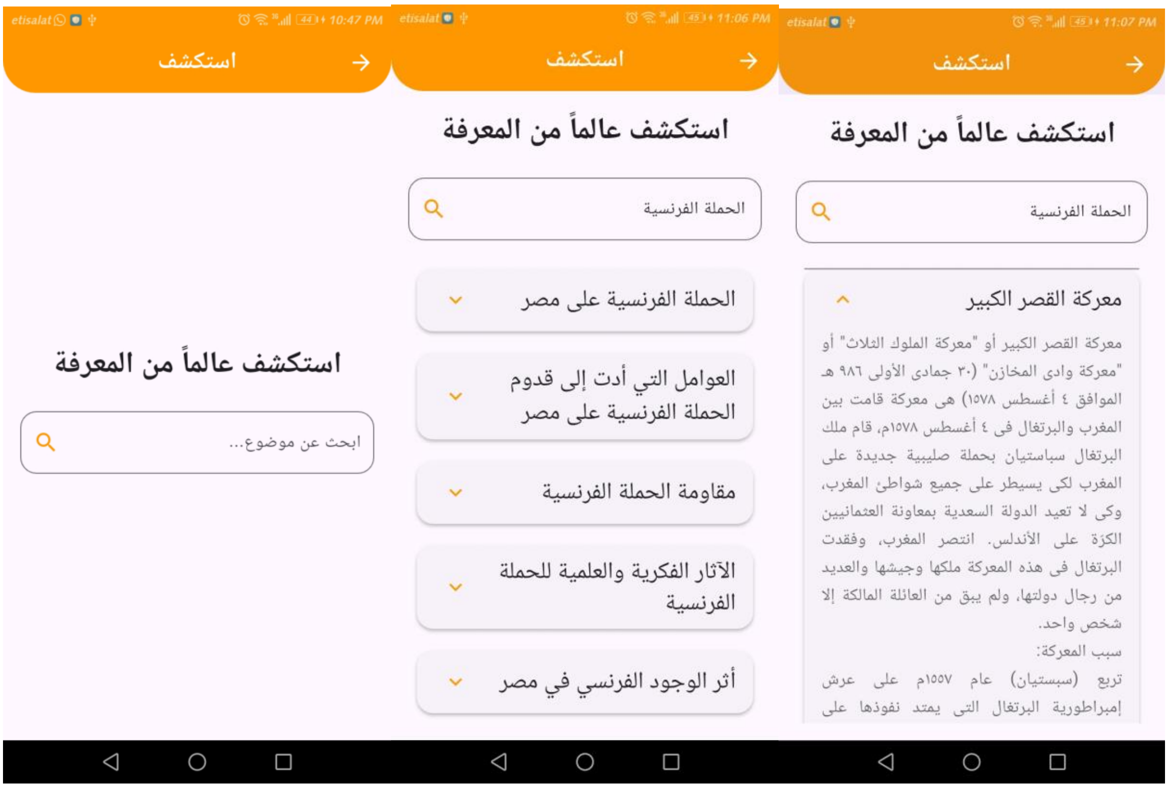


Figure 5.16: Student Explore Screens

- **Profile Screen**
  - View and edit personal details.
  - Change Language

Figure 5.17: Student Profile Screens

  - log out

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This project has successfully developed an AI-powered Learning Management System (LMS) mobile application tailored for Arabic-language education. The system addresses key limitations of traditional LMS platforms by integrating advanced Natural Language Processing (NLP) techniques to enable automated question generation, distractor creation, curriculum tagging, personalized feedback, and performance-based resource recommendations.

State-of-the-art transformer models—including AraT5, mT5, and AraBART—were fine-tuned on diverse Arabic datasets such as Arabic-SQuADv2.0, Arabic-RACE, and a custom EKB dataset. Among these, AraT5 demonstrated the most consistent and superior performance in Arabic educational question-related tasks, based on quantitative evaluations using BLEU, METEOR, and F1-score. These metrics reflect its ability to generate syntactically fluent, semantically accurate, and pedagogically meaningful questions and answers.

For teachers, the system significantly reduces the manual burden of assessment creation and offers detailed analytics for tracking student progress. For students, it delivers a personalized and adaptive learning journey, automatically recommending targeted resources based on individual weaknesses and mistakes.

The final implementation demonstrates a robust, intelligent LMS that automates key educational processes while maintaining instructional integrity, making it a promising step forward in AI-driven Arabic education technology.

6.2 Future Work

Future versions of the system will involve deeper fine-tuning of the NLP models using larger and more diverse Arabic-language datasets. This improvement aims to enhance the linguistic and contextual accuracy of the generated questions, answers, and feedback, making the learning experience more precise and effective.

While the current implementation focuses on history-related content, upcoming iterations of the system will expand support to include additional academic subjects such as STEM subjects, literature, and Social Sciences. This will make the platform applicable across a wider range of school curricula and learning needs.

Planned updates also include the development of advanced learning analytics dashboards for teachers and school administrators.

References

[1] Timmi M, Laaouina L, Jeghal A, El Garouani S, Yahyaouy A. Educational Video Recommender System. International Journal of Information and Education Technology. 2024;14(3):362-71.

[2] Hazar MJ, Abid Muslam Abid Ali A, Zrigui S, Maraoui M, Mabrouk M, Zrigui M. Educational videos recommendation system based on topic modeling. InInternational Conference on Computational Collective Intelligence 2023 (pp. 363-376). Springer, Cham.

[3] Chan, N.N., Roussanaly, A., Boyer, A. (2014). Learning Resource Recommendation: An Orchestration of Content-Based Filtering, Word Semantic Similarity and Page Ranking. In: Rensing, C., de Freitas, S., Ley, T., Muñoz-Merino, P.J. (eds) Open Learning and Teaching in Educational Communities. EC-TEL 2014. Lecture Notes in Computer Science, vol 8719. Springer, Cham. https://doi.org/10.1007/978-3-319-11200-8_23

[4] Challagundla, Bhavith Chandra & Challagundla, Shravani & Gogireddy, Yugandhar & Prasad, Preethika. (2024). End-to-End Neural Embedding Pipeline for Large-Scale PDF Document Retrieval Using Distributed FAISS and Sentence Transformer Models. 10.13140/RG.2.2.12625.34409.

[5] Lafkiar, S., En Nahnahi, N. An end-to-end transformer-based model for Arabic question generation. Multimed Tools Appl (2024). https://doi.org/10.1007/s11042-024-19958-3

[6] Alhashedi, S., Mohd Suaib, N., & Bakri, A. (2022). Arabic automatic question generation using transformer model (EasyChair Preprint No. 8588). EasyChair. https://easychair.org/publications/preprint/8588

[7] Nagoudi, E. M. B., Elmadany, A., & Abdul-Mageed, M. (2021). AraT5: Text-to-text transformers for Arabic language generation. arXiv preprint arXiv:2109.12068.

[8] Nakhleh S, Mustafa AM, Najadat H. AraT5GQA: Arabic Question Answering model using automatic generated dataset. In2024 15th International Conference on Information and Communication Systems (ICICS) 2024 Aug 13 (pp. 1-5). IEEE.

[9] Alshehri, Afnan. (2024). AraQA-BERT: Towards an Arabic Question Answering System using Pre-trained BERT Models. WSEAS TRANSACTIONS ON INFORMATION SCIENCE AND APPLICATIONS. 21. 361-373. 10.37394/23209.2024.21.34.

[10] Alsubhi K, Jamal A, Alhothali A. Pre-trained transformer-based approach for arabic question answering: A comparative study. arXiv preprint arXiv:2111.05671. 2021 Nov 10.

[11] Al Hashedi SS, Suaib NM, Bakri A. Arabic automatic question generation using transformer model. InAIP Conference Proceedings 2024 Jun 7 (Vol. 2991, No. 1). AIP Publishing.

[12] Lafkiar S, En Nahnahi N. An end-to-end transformer-based model for Arabic question generation. Multimedia Tools and Applications. 2024 Aug 1:1-5.

[13] Tami M, Ashqar HI, Elhenawy M. Automated question generation for science tests in arabic language using nlp techniques. InInternational Conference on Intelligent Systems, Blockchain, and Communication Technologies 2024 Jul 13 (pp. 274-285). Cham: Springer Nature Switzerland.

[14] S. Qin, R. Menezes and M. Silaghi, "A Recommender System for Youtube Based on its Network of Reviewers," 2010 IEEE Second International Conference on Social Computing, Minneapolis, MN, USA, 2010, pp. 323-328, doi: 10.1109/SocialCom.2010.53.

[15] M. Abbas, M. U. Riaz, A. Rauf, M. T. Khan and S. Khalid, "Context-aware Youtube recommender system," 2017 International Conference on Information and Communication Technologies (ICICT), Karachi, Pakistan, 2017, pp. 161-164, doi: 10.1109/ICICT.2017.8320183.

[16] Sammoudi, M., Habaybeh, A., Ashqar, H.I., Elhenawy, M. (2025). Question-Answering (QA) Model for a Personalized Learning Assistant for Arabic Language. In: Abdelgawad, A., Jamil, A., Hameed, A.A. (eds) Intelligent Systems, Blockchain, and Communication Technologies. ISBCom 2024.