

NATIONAL TECHNICAL UNIVERSITY OF UKRAINE

“IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE”

Faculty of Informatics and Computer Engineering

Department of Computer Engineering

Admitted to the defense:

Chairman of the Department

Sergii STIRENKO
(signature)

“—” 2025

Graduation project

for obtaining a bachelor's degree

in the educational and professional program

“Software Engineering for Computer Systems”

specialty 121 "Software Engineering"

on the topic: Web Service for Creating Tests (QuizMaster)

The work was performed by: 4 year student, IM-14 group
(group code)

Mehmet KULUBECIOGLU

(Full Name)



(signature)

Project supervisor Professor Kornienko Bogdan Yaroslavovich

(position, academic degree, academic status, surname and initials)

(signature)

Advisor (normative control) Associate Professor Valerii Pavlov

(section name) (position, academic degree, academic status, surname and initials)

(signature)

Reviewer Professor Inna Stetsenko

(position, academic degree, academic status, surname and initials)

(signature)

I certify that in this graduation project there are no borrowings from the works of other authors without proper references.

Mehmet KULUBECIOGLU

(signature)

Kyiv – 2025

**NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
“IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE”**

Faculty of Informatics and Computer Engineering

Department of Computer Engineering

Level of higher education - first (bachelor)

Education and Professional Program

"Computer Systems Software Engineering

specialty 121 "Software Engineering"

APPROVED

**Chairman of the
Department**

_____ Sergii STIRENKO

«____» _____ 20__ p.

TASK

for a student's bachelor's degree project

Mehmet KULUBECIOGLU

1. Project topic Web Service for Creating Tests (QuizMaster), Project supervisor Professor Kornienko Bogdan Yaroslavovich,
approved by the order of the University of _____, 2025 № _____
2. Deadline for submission of the completed project by the student May 20, 2025.
3. Output data to the project: Technical documentation, theoretical background, system architecture, web design layout with partial responsive adjustments, screenshots of the working QuizMaster prototype.
4. The content of the explanatory note (the list of issues to be developed):
Description of the problem area related to online test creation,

analysis of existing online testing and assessment systems,
design of a user-friendly interface for creating and managing exams using
HTML, CSS, and JavaScript,
implementation of a PHP-based server-side logic with MySQL database
integration,
development of session tracking and result management features,
and testing of system performance and accessibility with partial mobile
compatibility.

5. List of graphic material (with exact designation of obligatory drawings):
Architecture diagram of the QuizMaster system,
component structure of the PHP-based application,
question management module layout,
desktop view of the QuizMaster interface with adjusted mobile
screenshots,
user flow diagram from login to exam submission and result viewing.
6. Project consultant, indicating the sections of the project that are included
in them

Section	Advisor	Signature, date	
		Task issued	Task accepted

7. Date of issue of the task

Timetable

Nº	Naming the stages of the diploma project	Timeframe for completing the phases	Notes
1	<i>Approval of the subject of the project</i>	<i>01.02.2025 – 05.02.2025</i>	
2	<i>Study and analysis of the task</i>	<i>05.02.2025 – 25.02.2025</i>	
3	<i>Development of architecture and general system structures</i>	<i>26.02.2025 – 10.03.2025</i>	
4	<i>Development of separate structures subsystems</i>	<i>11.03.2025 – 25.03.2025</i>	
5	<i>Implementation of the web application</i>	<i>26.03.2025 – 05.04.2025</i>	
6	<i>Writing an explanatory note</i>	<i>06.04.2025 – 20.04.2025</i>	
7	<i>Project testing and adjustments</i>	<i>21.04.2025 – 5.04.2025</i>	
8	<i>Preliminary protection</i>	<i>05.05.2025</i>	
9	<i>Protection</i>	<i>20.05.2025</i>	

Graduate student

Mehmet KULUBECIOGLU

Project supervisor

Kornienko Bogdan Yaroslavovich

Annotation

This diploma project aims to develop a web-based system designed to streamline the creation and management of online tests. The main goal of the application, named QuizMaster, is to enable educators to create multiple-choice exams, assign durations, monitor student performance, and review results through a centralized dashboard, enhancing the efficiency of the assessment process.

The software component of this system is implemented using HTML5, CSS3, and JavaScript for the front-end, providing a structured and partially responsive interface, while the back-end is built with PHP and integrated with a MySQL database to handle user authentication, session management, and data storage. The application focuses on real-time session tracking and automated result analysis to support a robust educational experience.

In this diploma project, various technical modules were developed, including: a question management interface for teachers, an exam interface for students with optional audio support, a results tracking system, and a secure login mechanism. Additionally, diagrams such as the system architecture, use case diagram, and user flow diagrams were prepared to visualize the technical structure and interactions within the QuizMaster application.

Анотація

Цей дипломний проект спрямований на розробку вебзастосунку, призначеного для спрощення створення та управління онлайн-тестами. Основна мета застосунку, названого QuizMaster, — надати педагогам можливість створювати тести з кількома варіантами відповідей, встановлювати тривалість, відстежувати продуктивність студентів та переглядати результати через централізовану панель управління, що підвищує ефективність процесу оцінювання.

Програмний компонент цієї системи реалізований за допомогою HTML5, CSS3 та JavaScript для фронтенду, що забезпечує структурований та частково адаптивний інтерфейс, тоді як бекенд побудований на PHP із інтеграцією бази даних MySQL для обробки автентифікації користувачів, управління сесіями та зберігання даних. Застосунок зосереджений на реальному відстеженні сесій та автоматичному аналізі результатів, щоб підтримувати надійний навчальний досвід.

У цьому дипломному проекті було розроблено різні технічні модулі, зокрема: інтерфейс управління питаннями для викладачів, екзаменаційний інтерфейс для студентів із необов'язковою підтримкою аудіо, система відстеження результатів та безпечний механізм входу. Крім того, були підготовлені діаграми, такі як архітектура системи, діаграма варіантів використання (use case diagram) та діаграми користувацького потоку, для візуалізації технічної структури та взаємодій у межах застосунку QuizMaster.

Information	Format	Value	Name	number of sheets	copy number	Addition
			General documentation			
			Redesigned			
	A4	IAЛЦ.466538.002 TT	Web Service for Creating Tests (QuizMaster) Technical task	3		
	A4	IAЛЦ.466538.003 TPN	Web Service for Creating Tests (QuizMaster) Technical project notice	1		
	A4	IAЛЦ.466538.004 TOC	Web Service for Creating Tests (QuizMaster) Table of Contents	5		
	A4	IAЛЦ.466538.005 EN	Web Service for Creating Tests (QuizMaster) Explanatory note	97		
	A1	IAЛЦ.466538.006 D1	Web Service for Creating Tests (QuizMaster) User flow diagram	1		
	A1	IAЛЦ.466538.007 D2	Web Service for Creating Tests (QuizMaster) Entity-Relationship Diagram (ERD)	1		
	A1	IAЛЦ.466538.008 D3	Web Service for Creating Tests (QuizMaster) File Structure Diagram	1		
	A1	IAЛЦ.466538.009 D4	Web Service for Creating Tests (QuizMaster) System architecture diagram	1		

CHG	Sheet	№ document	Sign.	Date	IAЛЦ.466538.001 AD			
Developed	Mehmet KULUBECIOGLU				Web Service for Creating Tests (QuizMaster)			
Checked	Kornienko Bogdan Yaroslavovich				Type	sheet	Sheets	
Reviewer	Professor Inna Stetsenko					1	1	
Norm. control	Simonenko V. P				NTUU “KPI”, FIOT IM-14			
Approved	Stirenko S. G.							

*Technical task for the
diploma project*

CONTENTS

1. NAME AND AREA OF APPLICATION
2. REASONS FOR DEVELOPMENT
3. PURPOSE OF THE DEVELOPMENT
4. THE SOURCES OF DEVELOPMENT
5. TECHNICAL REQUIREMENTS
 - 5.1 REQUIREMENTS FOR THE PRODUCT TO BE DEVELOPED
 - 5.2 SOFTWARE REQUIREMENTS
 - 5.3 REQUIREMENTS FOR THE HARDWARE
6. STAGES OF DEVELOPMENT

C H G	Sheets	No document.	Sign.	Date	IAЛЦ.467100.002 TT			
					Web Service for Creating Tests (QuizMaster)	Type	Sheets	Sheet
Developed	Mehmet KULUBECI OGLU						3	1
Checked	Kornienko Bogdan Yaroslavovich							
Norm. control	Simonenko V. P.							
Approved	Stirenko S. G.					NTUU «KPI», FIOT IM-14		

1. NAME AND AREA OF APPLICATION

This technical task pertains to the development of a web-based service named "QuizMaster" for creating and managing online tests. Scope of application: an alternative to the current manual methods of exam creation and assessment in higher educational institutions, specifically targeting the course "Software Engineering" at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute."

2. REASONS FOR DEVELOPMENT

The basis for the development is the assignment for the undergraduate graduation project under the educational and professional program "Software Engineering for Computer Systems," specialty 121 "Software Engineering," approved by the Department of Computer Engineering of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute."

3. PURPOSE OF THE DEVELOPMENT

The purpose of this project is to develop a web-based platform, QuizMaster, that enables educators to create, manage, and evaluate multiple-choice exams, monitor student performance, and review results through a centralized dashboard, while providing students with an accessible and efficient exam-taking experience.

4. THE SOURCES OF DEVELOPMENT

The sources of development include scientific and technical literature on web development, online assessment systems, and database management, publications in periodicals, guides on web development frameworks and tools (e.g., PHP, MySQL, HTML, CSS, JavaScript), and online resources such as PHP and MySQL documentation, W3Schools, Stack Overflow, and Mozilla Developer Network (MDN).

5. TECHNICAL REQUIREMENTS

5.1 Requirements for the Product to be Developed

- Functional completeness: The system must allow teachers to create and manage exams, and students to take exams and view scores.
- User-friendliness: The interface must be intuitive for both teachers and students, with clear navigation and minimal technical requirements.
- Technical correctness: The system must accurately calculate scores, track sessions, and store data securely.
- Partial mobile compatibility: The system should be accessible on mobile devices with basic responsiveness.

Ch	Sheets	No document	Sig	Date	IAЛП.467100.002 TT	Sheet
G						2

5.2 Software Requirements

- Operating Systems: Windows 10, Windows 11, or any OS supporting modern web browsers.
- Web Server: XAMPP (Apache + MySQL) for local development.
- Browser: Google Chrome, Mozilla Firefox, Microsoft Edge (latest versions).
- Programming Languages: HTML5, CSS3, JavaScript, PHP 8.x.
- Database: MySQL 8.0 or higher.
- Development Tools: Visual Studio Code, phpMyAdmin for database management.

5.3 Requirements for the Hardware

- Computer based on Intel Core i3 or higher processor.
- At least 4 GB of RAM.
- Free hard disk space of at least 500 MB.
- Stable internet connection for development and testing.

6. STAGES OF DEVELOPMENT

Stage	Description	Date
6.1	Study of literature and existing systems	05.02.2025
6.2	Creation and arrangement of the technical task	10.02.2025
6.3	Analysis of the structure of the QuizMaster system	26.02.2025
6.4	Development of system modules (e.g., admin panel, student interface)	11.03.2025
6.5	Implementation, debugging, and bug fixing	26.03.2025
6.6	Preparation of diploma project documentation	06.04.2025

					IAЛЦ.467100.002 TT	Sheet 3

TECHNICAL PROJECT NOTICE

Web Service for Creating Tests (QuizMaster)

- Project Overview:

This project involves the development of a web-based platform named "QuizMaster" for creating and managing online tests, targeting the course "Software Engineering" at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute."

- Objective:

QuizMaster aims to enable educators to create, manage, and evaluate multiple-choice exams, monitor student performance, and review results through a centralized dashboard, while providing students with an accessible exam-taking experience.

- Scope:

The system is developed using HTML5, CSS3, JavaScript, PHP, and MySQL, with features including session tracking, result management, and partial mobile compatibility.

Project Details:

- **Student:** Mehmet Kulubecioğlu
- **Supervisor:** Professor Kornienko Bogdan Yaroslavovich
- **Deadline:** June 20, 2025

C H G	Sheets	No document.	Sign.	Date	IAЛЦ.467100.003 ТРН			
Developed		Mehmet KULUBECI OGLU			Web Service for Creating Tests (QuizMaster)	Type	Sheets	Sheet
Checked		Kornienko Bogdan Yaroslavovich					1	1
Norm. control		Simonenko V. P.				NTUU «KPI», FIOT IM-14		
Approved		Stirenko S.						

1. INTRODUCTION.....	1
2. REVIEW OF SIMILAR SYSTEMS.....	3
2.1 Google Forms.....	3
Advantages:.....	4
Limitations:.....	4
Comparison with QuizMaster:.....	5
2.2 Kahoot!.....	5
Advantages:.....	5
Limitations:.....	6
Comparison with QuizMaster:.....	6
2.3 Quizizz.....	6
Advantages:.....	7
Limitations:.....	7
Comparison with QuizMaster:.....	7
2.4 Testmoz.....	8
Advantages:.....	8
Limitations:.....	8
Comparison with QuizMaster:.....	8
2.5 Comparative Analysis Table.....	9
2.6 Conclusion of This Section.....	10
3. TOOLS AND TECHNOLOGIES USED.....	11
3.1 Programming and Scripting Languages.....	11
3.1.1 HTML5 (HyperText Markup Language).....	11
3.1.2 CSS3 (Cascading Style Sheets).....	12
3.1.3 JavaScript (JS).....	12
3.1.4 PHP (Hypertext Preprocessor).....	13
3.1.5 MySQL (Relational Database Management System).....	14

C H G	Sheets	№ document.	Sign.	Date	IAJЦ.466538.004 ТОС		
Developed	Mehmet KULUBECI OGLU				Web Service for Creating Tests (QuizMaster)	Type	Sheets
Checked	Kornienko Bogdan Yaroslavovich					5	1
Norm. control	Simonenko V. P.						
Approved	Stirenko S. G.					NTUU «KPI», FIOT IM-14	

3.2 Development Tools and Environment.....	15
3.2.1 Visual Studio Code (VS Code).....	15
3.2.2 XAMPP (Apache, MySQL, PHP, Perl Stack).....	15
3.3 Supporting Tools and Utilities.....	16
3.4 Summary.....	17
4. REQUIREMENTS ANALYSIS.....	18
4.1 Functional Requirements.....	18
4.1.1 Teacher/Instructor Requirements.....	19
4.1.3 Admin/System Requirements.....	20
4.2 Non-Functional Requirements.....	20
4.3 User Roles.....	22
4.4 User Stories.....	22
4.5 Use Case Diagram (Optional in Report).....	23
4.6 Summary.....	24
5. DATABASE DESIGN.....	25
5.1 Overview.....	25
5.2 Core Tables.....	25
5.3 Table Structures.....	27
5.6 Summary.....	32
Users Table.....	34
Questions Table.....	34
Scores Table.....	34
Relationships.....	35
5.8 Summary.....	35
5.10 Summary.....	38

G.	Sheets	No document	Sig n.	Date	Sheet
					2

ІАЛЦ.467100.004 ТОС

6. SYSTEM ARCHITECTURE.....	39
6.1 Overview.....	39
6.2 Client-Side Components.....	39
6.2.1. Teacher/Admin Interface.....	39
6.2.2. Student Interface.....	40
6.3 Server-Side Logic.....	40
6.4 Data and Session Flow.....	42
6.4.1 Teacher Workflow:.....	42
6.4.2 Student Workflow:.....	42
6.5 File Structure.....	43
6.6 Architecture Highlights.....	44
6.6.1 System Architecture Diagram.....	44
6.7 Summary.....	45
7. INTERFACE AND UI DESIGN.....	46
7.1 Design Principles.....	46
7.2 Admin Panel (Teacher Interface).....	46
7.3 Student Interface.....	51
7.4 Styling and Assets.....	54
8. IMPLEMENTATION.....	55
8.1 Admin Panel.....	55
8.2 Student Interface.....	56
Start Exam Page (exam.php).....	56
Exam Interface.....	57
Exam Submission (submit_exam.php).....	58
Session Management.....	59
Key Features:.....	59
8.3 Exam System Logic.....	60
8.3.1. Starting the Exam (exam.php).....	60
8.3.3. Session Tracking with exam_session_id.....	61
8.3.4. Duration Tracking.....	62
8.4 Summary.....	62

G. H	She ets	No document	Sig n.	Da te	IAЛІЦ.467100.004 ТОС	Sh eet
						3

9. TESTING AND RESULTS.....	63
9.1 Overview.....	63
9.2 Test Environment.....	64
9.3 Admin Panel Test Cases.....	65
9.4 Student Exam Test Cases.....	67
9.5 Database Verification.....	69
9.6 Sample Result Entry.....	70
9.7 Summary.....	71
10. SECURITY CONSIDERATIONS.....	72
10.1 Introduction.....	72
10.2 Authentication and Session Management.....	72
10.3 Exam Session Tracking (Students).....	73
10.4 Input Validation and Sanitization.....	73
10.5 SQL Injection Prevention.....	74
10.6 Password Security.....	74
10.7 File Upload Security.....	75
10.8 Privacy Considerations.....	75
10.9 Security Limitations and Future Enhancements.....	75
10.10 Summary.....	76
11. CHALLENGES ENCOUNTERED.....	77
11.1 Technical Challenges.....	77
11.1.1. Database Session Tracking.....	77
11.1.2. Audio File Uploading.....	77
11.1.3. Countdown Timer Synchronization.....	78
11.2 Design and Usability Challenges.....	78
11.2.1. User Interface Consistency.....	78
11.2.2. Mobile Responsiveness.....	78
11.3 Project Management Challenges.....	79
11.3.1. Time Management.....	79
11.3.2. File & Code Organization.....	79
11.4 Summary.....	79
12. CONCLUSION.....	80
Key Achievements:.....	80
Final Reflection:.....	81

G. H.	She ets	No document	Sig n.	Da te	IAЛІЦ.467100.004 ТОС	Sh eet
						4

13. REFERENCES.....	82
13.1 Online Resources.....	82
13.2 Tools Used.....	82
13.3 Project Inspiration & Coursework.....	82
14. APPENDICES.....	83
14.1 Code Snippets.....	83
14.1.1. Starting an Exam Session.....	83
14.1.2. Calculating and Storing Exam Score.....	84
14.1.3. Question Deletion Logic.....	84
14.1.4. Server-Side Duration Calculation.....	85
14.2 Screenshots.....	86
14.3 ER & UML Diagrams and Figures.....	95
14.3.1 SQL Table Schemas.....	95
14.3.2 Use Case Diagram.....	98
14.3.3 Entity-Relationship Diagram (ERD).....	99
14.3.4 File Directory Structure.....	100
14.3.5 System Architecture Diagram.....	101

Sheet				
5				
G. She ets	No document	Sig n.	Date	ІАЛЦ.467100.004 ТОС

1. INTRODUCTION

In today's digital era, the need for effective and accessible assessment tools is greater than ever. As education increasingly moves towards online and hybrid models, the demand for platforms that allow instructors to create, manage, and evaluate tests has grown significantly. Manual methods of preparing exams are not only time-consuming but also prone to human error and inefficiencies. This graduation project aims to address this need by designing and developing a **web-based service for creating and managing tests**, allowing educators to streamline the testing process in a secure and user-friendly environment.

The proposed system, named **QuizMaster**, is a full-stack web application that enables instructors to create multiple-choice exams, assign durations, monitor student performance, and review results—all from a centralized dashboard. Unlike traditional tools, QuizMaster focuses on **real-time session tracking**, **exam grouping**, and **automated result analysis**, offering a robust experience both for educators and examinees.

The project is built using commonly available technologies such as **HTML**, **CSS**, **JavaScript**, **PHP**, and **MySQL**, ensuring wide compatibility and maintainability. It also considers critical aspects such as **session management**, **exam timing**, **user authority roles**, and **data security**.

This report will provide a comprehensive look at the development of the QuizMaster system. It will begin with an overview of similar platforms in use today, followed by an in-depth explanation of the tools and frameworks utilized. The document will then delve into the design, architecture, and implementation of the system, accompanied by database diagrams, code snippets, and screenshots. Lastly, the report will present the testing procedures, results, challenges encountered during development, and a conclusion outlining future improvement opportunities.

CH	She	No document	Sig	Date	Sheet
G.	sts		n.		2

IAIЦ.466538.005 EN

2. REVIEW OF SIMILAR SYSTEMS

In the rapidly evolving landscape of digital education, the development of tools that facilitate online learning, assessment, and evaluation has become increasingly important. Among the most significant components of this transformation is the ability to conduct exams and assessments through web-based platforms. These systems are not only designed to simplify the test creation and grading process for educators but also to improve the accessibility and efficiency of testing for students. Over the years, several web-based platforms have emerged, offering various features for educators to create and administer online tests. This section provides an in-depth review of some of the most prominent systems: **Google Forms**, **Kahoot**, **Quizizz**, and **Testmoz**. Each platform will be analyzed in terms of functionality, strengths, weaknesses, and how it compares to the proposed system, **QuizMaster**.

2.1 Google Forms

Google Forms is one of the most widely used tools for creating forms, surveys, and quizzes. As part of the Google Workspace suite, it offers a seamless experience for users already integrated into the Google ecosystem. Its interface is intuitive and user-friendly, making it accessible to individuals with little to no technical background.

Ch	Sheets	No document	Sig n.	Date	IAЛЦ.466538.005 EN	Sheet 3
----	--------	-------------	--------	------	--------------------	---------

Advantages:

- **Ease of use:** With a drag-and-drop interface, users can create quizzes within minutes.
- **Integration with other Google services:** Automatically saves data to Google Sheets, allowing for real-time collaboration and result analysis.
- **Support for multiple question types:** Allows for multiple-choice, checkboxes, dropdowns, and paragraph responses.
- **Basic automatic grading:** Especially for multiple-choice questions.
- **Free to use:** No cost involved, making it highly accessible.

Limitations:

- **No user authentication:** Anyone with the link can access and attempt the test; there is no secure login system.
- **No control over test duration per user or session:** Time-bound exams are not supported natively.
- **Limited analytics:** Only basic result summaries are available.
- **No session grouping:** Multiple attempts by the same user are treated separately, with no session tracking.
- **Not built specifically for exams:** Designed more for surveys and feedback collection rather than secure testing.

Ch	She	No document	Sig	Date	IAЛЦ.466538.005 EN	Sheet 4
----	-----	-------------	-----	------	--------------------	---------

Comparison with QuizMaster:

While Google Forms excels in simplicity and accessibility, it falls short in providing the features required for structured, secure online testing environments. In contrast, **QuizMaster** includes user roles, session IDs, exam timing, and per-question analysis—features that are crucial for formal education settings. QuizMaster also ensures controlled access and better data integrity through its authentication mechanisms.

2.2 Kahoot!

Kahoot! has revolutionized classroom engagement by introducing a game-based learning model. It allows instructors to create quizzes that are played live with real-time feedback and competition between students.

Advantages:

- **Gamified learning:** Turns quizzes into fun, interactive games.
- **Real-time feedback:** Leaderboards and instant scoring increase motivation.
- **Rich media integration:** Questions can include images and videos to enhance the learning experience.
- **Cross-platform support:** Available via web and mobile applications

Ch	She	No document	Sig	Date	IAЛЦ.466538.005 EN	Sheet
G.	sts		n.	Da		5

Limitations:

- **Not ideal for formal exams:** Designed more for formative assessments rather than summative evaluation.
- **No built-in login system for security:** Players join with PINs; identity validation is minimal.
- **No detailed analytics per user/session:** Results are aggregated, making individual tracking difficult.
- **Requires synchronous participation:** All participants must be present at the same time.

Comparison with QuizMaster:

Kahoot is a powerful engagement tool, but it lacks the structural and security capabilities needed for exam settings. **QuizMaster**, on the other hand, supports asynchronous testing, structured result storage, and detailed logging of user responses. While Kahoot focuses on engagement, QuizMaster is focused on reliability, session control, and formal evaluation processes.

2.3 Quizizz

Quizizz is a platform similar to Kahoot but with the added flexibility of asynchronous testing. It combines game elements with educational assessment and offers some analytical features.

G. H.	Sne sts	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sh eet 6

Advantages:

- **Asynchronous mode:** Allows students to take quizzes on their own time.
- **Interactive design:** Gamification remains a core feature.
- **Question bank:** Reuse and search from a massive question repository.
- **Basic performance tracking:** Provides class-wide and individual reports.

Limitations:

- **Customization is limited:** Backend functionality and layout cannot be altered.
- **Some features require paid plans:** Advanced reporting and branding are locked behind subscriptions.
- **Less control over user roles:** Role-based access is not deeply implemented.
- **Not open-source or self-hosted:** Institutions cannot deploy it on their own servers.

Comparison with QuizMaster:

While Quizizz is strong in its usability and asynchronous testing support, **QuizMaster** allows full control of both frontend and backend, including database-level modifications and deployment on private servers. This makes QuizMaster better suited for institutions that require ownership over their data and infrastructure.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sh eet
						7

2.4 Testmoz

Testmoz is a lightweight platform that allows users to quickly generate and share tests without creating an account. Its minimal interface and simple design appeal to users who need fast results with minimal configuration.

Advantages:

- **No login required for test takers:** Quick access.
- **Simple and clean interface:** Ideal for creating short tests quickly.
- **Supports automatic grading:** Especially for multiple-choice questions.

Limitations:

- **Lack of analytics and tracking:** No support for session grouping or individual performance history.
- **No user roles or permissions:** Everyone has the same level of access.
- **Limited security:** Anyone with the link can take or retake the test.
- **Minimal customization options:** Static structure with little flexibility.

Comparison with QuizMaster:

While Testmoz excels in its simplicity, it lacks scalability and customization. **QuizMaster** not only provides a user authentication system but also tracks session history, calculates scores securely, and allows future feature expansion. This makes it more suitable for academic and enterprise-level environments.

G.	Sheets	No document	Sig. n.	Date	IAЛЦ.466538.005 EN	Sheet
8						

2.5 Comparative Analysis Table

Table 2.1 – Comparative Analysis Table

Feature/Platform	Google Forms	Kahoot	Quizizz	Testmoz	QuizMaster
User Authentication	-	-	+	-	+
Session-Based Tracking	-	-	-	-	+
Role Management	-	-	Partial	-	+
Per-Question Duration	-	-	-	-	+
Multimedia Support	+	+	+	-	+
Asynchronous Testing	+	-	+	+	+
Detailed Analytics	-	Partial	+	-	+
Self-hosted Capability	-	-	-	-	+
Custom Backend Logic	-	-	-	-	+ (PHP/MySQL)
Data Security & Integrity	-	-	Partial	-	+

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet 9
----	--------	-------------	------	------	--------------------	---------

This table provides a comparative analysis of QuizMaster against other platforms (Google Forms, Kahoot, Quizizz, Testmoz), evaluating features such as user authentication, session-based tracking, role management, per-question duration, multimedia support, asynchronous testing, detailed analytics, self-hosted capability, custom backend logic, and data security & integrity.

2.6 Conclusion of This Section

The comparison clearly highlights that while existing platforms each have their own strengths, they are limited when it comes to handling **formal, structured online examinations** with full administrative control. Most of these systems are designed for **engagement, speed, or convenience**, but not for educational institutions that require robust user management, data integrity, session tracking, and customization.

QuizMaster was designed with these precise needs in mind. It integrates the core strengths of modern quiz systems—such as accessibility and UI clarity—with the architectural and security foundations needed for a real-world academic environment. Its self-hosted nature and database-driven structure offer unmatched flexibility, while its session-based exam tracking and per-question time controls make it a standout solution for formal testing scenarios.

CH G.	She ets	No document	Sig n.	Da te	ІАЛІЦ.466538.005 EN	Shee t 10
----------	------------	----------------	-----------	----------	---------------------	-----------------

3. TOOLS AND TECHNOLOGIES USED

This section provides a comprehensive overview of the programming languages, frameworks, development tools, and system components used in the design and development of the **QuizMaster** web service. The selection of these technologies was guided by key criteria such as platform compatibility, ease of development, system scalability, security, and suitability for educational environments.

3.1 Programming and Scripting Languages

3.1.1 HTML5 (HyperText Markup Language)

HTML5 forms the structural foundation of the QuizMaster system's frontend. It defines the basic layout of the web pages, forms, buttons, tables, and input fields. HTML5 is chosen over previous versions due to its support for semantic tags, responsive behavior, and enhanced multimedia integration.

Usage in QuizMaster:

- Creating form structures for login, quiz creation, and answer submission
- Designing static and dynamic content blocks
- Ensuring cross-browser compatibility with clean markup
- Structuring UI components like navigation bars, input groups, headers, and footers

G.	Sheets	No document	Sig.	Date	IAJЦ.466538.005 EN	Sheet
						11

3.1.2 CSS3 (Cascading Style Sheets)

CSS is used to enhance the visual presentation of HTML elements and to define the overall aesthetics of the system. In the QuizMaster project, the CSS file used was externally sourced and consisted of over 30,000 lines, likely aggregated from multiple unrelated projects. However, only relevant parts of this stylesheet were actively used within the application.

Usage in QuizMaster:

- Styling forms, buttons, menus, and typography
- Enhancing user experience through consistent color schemes and layout spacing
- Applying responsive design for mobile compatibility
- Customizing form validation and error indicators

3.1.3 JavaScript (JS)

JavaScript is the primary scripting language used on the client side to handle dynamic content and user interactions. Its implementation in QuizMaster focuses on time-sensitive operations and interface interactivity.

CH G.	Sheets	No document	Sig n.	Date	IAЛІЦ.466538.005 EN	Sheet 12

Usage in QuizMaster:

- Implementing countdown timers for timed exams
- Providing confirmation alerts before form submission
- Dynamically enabling/disabling form elements based on user actions
- Enhancing form behavior and input validation

By utilizing pure JavaScript rather than heavy libraries like jQuery or frameworks like React, the system remains lightweight and easy to maintain.

3.1.4 PHP (Hypertext Preprocessor)

PHP is the core server-side scripting language used in the QuizMaster system. It manages the business logic, database interactions, user authentication, and session management.

Usage in QuizMaster:

- Processing user login and role-based access (admin vs. student)
- Handling form submissions (quiz creation, answer saving)
- Executing CRUD (Create, Read, Update, Delete) operations on database entries
- Managing exam sessions and calculating scores
- Loading dynamic content from the database into the frontend

PHP was selected for its wide support, ease of integration with MySQL, and suitability for small to medium-scale educational web systems.

Sheet
ІАЛЦ.466538.005 EN 13

3.1.5 MySQL (Relational Database Management System)

MySQL serves as the data backbone of the QuizMaster platform. It is used to store and manage all application data, including user accounts, quiz questions, responses, scores, and session identifiers.

Key Database Tables:

- **users:** Stores registered teacher accounts with access roles
- **questions:** Contains all test questions along with correct answers, options, and durations
- **scores:** Tracks student exam performance and durations
- **exam_results:** Logs each answer attempt with correctness and timestamp
- **exam_session_id:** Uniquely identifies test attempts for grouping purposes

Database Design Highlights:

- Normalized table structures for optimized query performance
- Use of foreign keys and session tokens to ensure relational integrity
- Data sanitization and prepared statements in PHP to prevent SQL injection

Sheet
14
G. Sheets No document Sig n. Date IAЛЦ.466538.005 EN

3.2 Development Tools and Environment

3.2.1 Visual Studio Code (VS Code)

Visual Studio Code is a modern, lightweight code editor used throughout the development of QuizMaster. It provides syntax highlighting, integrated terminal, version control tools, and a rich ecosystem of extensions tailored to PHP and web development.

Why VS Code?

- Seamless integration with Git for version control
- Extensions for PHP, MySQL, HTML, and JavaScript support
- Built-in debugging tools and live preview features
- Multi-platform compatibility (Windows/Linux/macOS)

VS Code significantly improved development productivity and code management during the project lifecycle.

3.2.2 XAMPP (Apache, MySQL, PHP, Perl Stack)

XAMPP is a popular open-source local development environment that includes all the necessary server components for running PHP applications with MySQL support. It was used for developing and testing the QuizMaster system on a local machine before deployment.

Sheet
ІАЛЦ.466538.005 EN
G. Sheets No document Sig. n. Date 15

Components Used in XAMPP:

- **Apache**: Web server hosting the PHP files
- **MySQL**: Database service used by QuizMaster
- **phpMyAdmin**: GUI tool for database design and query execution

Benefits:

- Easy installation and quick setup
- Local testing environment with no need for internet connection
- Direct access to error logs and system feedback during development

3.3 Supporting Tools and Utilities

In addition to core technologies, several supporting tools were employed to assist development, testing, and documentation.

Sheet	
ІАЛЦ.466538.005 EN	
G. Sheets No document Sig n. Date	16

Table 3.1 – Supporting Tools and Their Uses

Tool	Purpose
phpMyAdmin	Visual management of MySQL databases, table creation, data insertion
Web Browsers	UI/UX testing across Chrome, Firefox, and Edge
Paint & Snipping Tool	Capturing screenshots for system documentation
Google Fonts	Integration of custom fonts for improved readability
W3Schools/ MDN	Online references for syntax validation and documentation

3.4 Summary

The technology stack chosen for the QuizMaster system prioritizes **simplicity, performance, scalability, and open-source availability**. PHP and MySQL offer robust backend capabilities, while HTML, CSS, and JavaScript deliver a responsive and accessible user interface. The development environment—comprising VS Code and XAMPP—ensures streamlined project execution from design to deployment.

By leveraging these widely supported and well-documented technologies, the system maintains a balance between customizability and reliability. The overall architecture reflects a deliberate design philosophy: to build a secure, session-based, multi-user quiz system that is easy to maintain and extend in the future.

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet 17
----	--------	-------------	------	------	--------------------	----------

4. REQUIREMENTS ANALYSIS

(Functional and Non-Functional Requirements of the QuizMaster System)

A successful software system must be built on a solid foundation of well-defined requirements. The **QuizMaster** platform was developed with a clear set of goals that address the needs of educational institutions, particularly instructors and students involved in online testing environments.

This section outlines both **functional** and **non-functional** requirements, along with **user roles**, **user stories**, and **use case scenarios**, all of which guided the design and implementation of the system.

4.1 Functional Requirements

Functional requirements describe **what the system should do**—the core features and capabilities that directly support user tasks. These were determined through observation of existing systems, analysis of institutional needs, and interviews with instructors.

Sheet	
ІАЛЦ.466538.005 EN	
G. Sheets No document Sig n. Date	18

4.1.1 Teacher/Instructor Requirements

- The system shall allow a teacher to log in securely with a username and password.
- The system shall allow a teacher to create new exams with multiple-choice questions.
- The system shall allow setting a specific duration for each test.
- The system shall allow setting the correct answer for each question.
- The system shall allow uploading optional audio files for listening-based questions.
- The system shall allow editing or deleting existing questions.
- The system shall allow assigning questions to a particular test.
- The system shall display a list of all created tests and their associated questions.
- The system shall display a report of student results, including scores and completion times.
- The system shall group results based on a unique session ID to separate exam attempts.
- The system shall allow teachers to view exam statistics and detailed logs.

4.1.2 Student/User Requirements

- The system shall allow a student to enter their name and email to take an exam.
- The system shall prevent students from taking the same test more than once in the same session.

G. No	Sheets	No document	Sig. n.	Date	IAЛЦ.466538.005 EN	Sheet 19

- The system shall present questions one by one or all at once, depending on the design.
- The system shall start a countdown timer when the test begins.
- The system shall automatically record the student's answers and time taken.
- The system shall calculate the final score and store it in the database.
- The system shall display a summary of the student's performance upon completion.
- The system shall ensure that once submitted, an exam cannot be modified.

4.1.3 Admin/System Requirements

- The system shall authenticate users and manage access rights.
- The system shall ensure that only authorized teachers can create or modify exams.
- The system shall manage session-based data grouping via **exam_session_id**.
- The system shall track all user interactions with timestamps.
- The system shall store data in a secure and consistent manner.

4.2 Non-Functional Requirements

These requirements describe **how the system should behave**, ensuring quality, performance, and reliability.

Sheet
G. Sheets No document Sig n. Date IAЛЦ.466538.005 EN 20

Table 4.1 – Non-Functional Requirements

Category	Requirement
Usability	The interface shall be clean and intuitive, even for non-technical users.
Performance	The system shall load pages and execute actions within 2 seconds under normal conditions.
Scalability	The system shall support multiple concurrent users without performance degradation.
Availability	The system shall be operational at least 99% of the time in a deployed environment.
Portability	The system shall work on modern web browsers (Chrome, Firefox, Edge).
Security	Passwords shall be stored encrypted; SQL injection and XSS shall be prevented.
Maintainability	The code shall be modular and well-documented to support future updates.
Backup	All exam results shall be backed up regularly or exportable from the database.

G. Ch	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 21
----------	------------	----------------	-----------	----------	--------------------	-------------

4.3 User Roles

QuizMaster distinguishes two primary user roles:

Table 4.2 – User Roles and Descriptions

Role	Description
Teacher	Has full access to the system backend. Can create/edit exams, manage questions, and view results.
Student	Has access only to the exam interface. Can take tests and view personal results post-submission.

There is no separate “admin” interface—teacher accounts act as system administrators

4.4 User Stories

User stories are informal descriptions of system features written from the perspective of the end-user.

- As a teacher, I want to create a multiple-choice test, so that my students can take it online.
- As a teacher, I want to set a timer for the test, so that students finish within a given time.
- As a teacher, I want to see all student scores and compare their

Sheet
ІАЛЦ.466538.005 EN 22

- performance across sessions.
- As a student, I want to take a test and see my result immediately after submission.
- As a student, I want to know how much time I have left while solving the test.
- As a teacher, I want to organize results by session to better analyze classroom progress.

4.5 Use Case Diagram (Optional in Report)

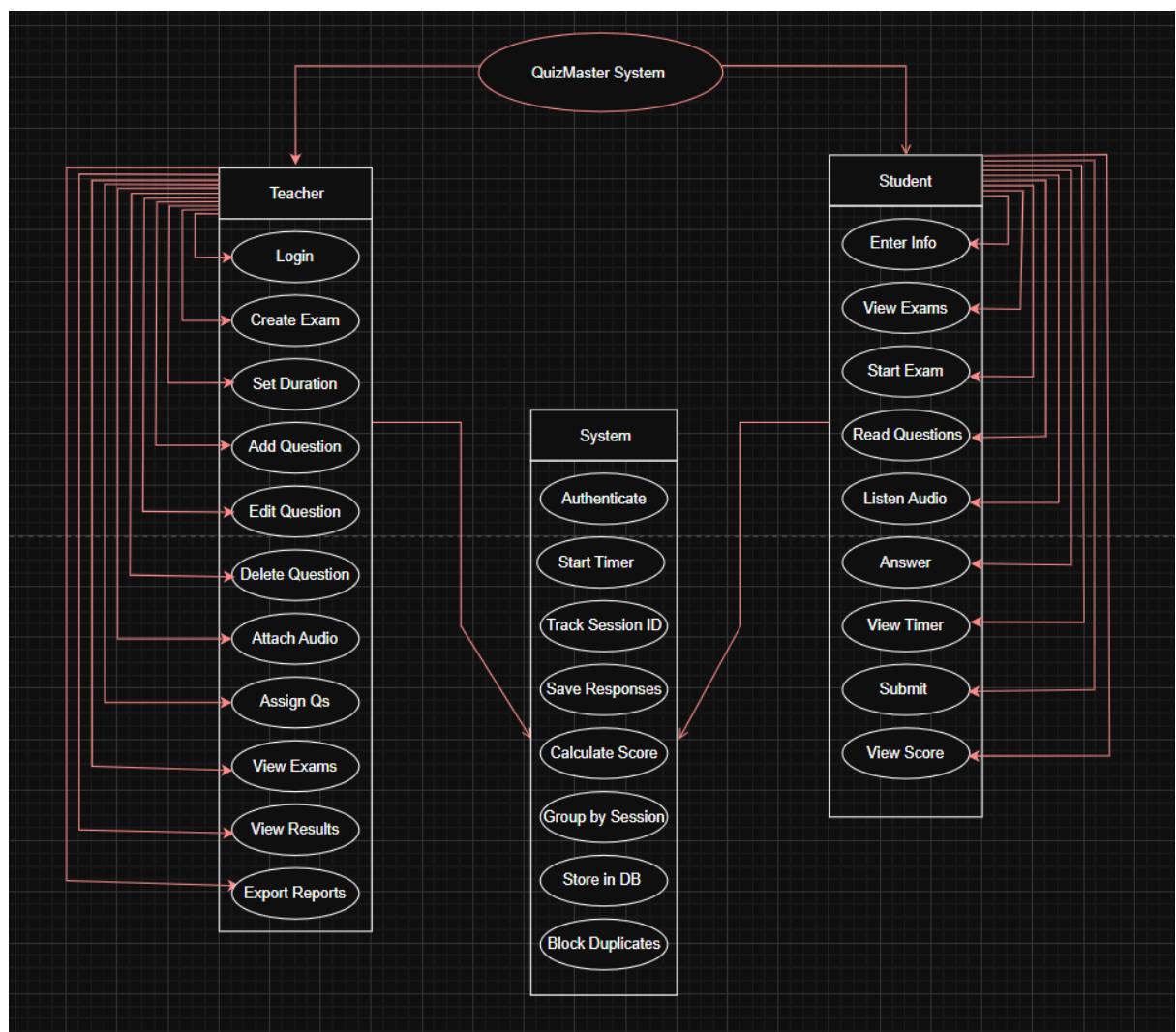


Figure 4.1 – Use Case Diagram of the QuizMaster System

							Sheet
C.H G.	She ets	No document	Sig n.	Date		ІАЛЦ.466538.005 EN	23

4.6 Summary

The detailed requirements defined in this phase laid a solid foundation for the development of the QuizMaster system. By explicitly distinguishing between functional and non-functional requirements, and by assigning clear responsibilities to each user role (Teacher, Student, and System), the system was designed to deliver a seamless experience that aligns with real-world educational workflows.

Special emphasis was placed on usability, performance, and data integrity to ensure the platform meets both pedagogical and technical expectations. These requirements guided critical decisions during the architectural planning stage and directly influenced the design of the database schema, the structure of the user interface, and the implementation of core business logic.

Furthermore, the defined use cases not only facilitated the creation of a structured UML Use Case Diagram but also ensured that every action within the system is purpose-driven and traceable. The output of this phase now serves as a blueprint for all subsequent development steps detailed in the upcoming chapters.

Sheet
IAЛЦ.466538.005 EN 24

5. DATABASE DESIGN

5.1 Overview

The database structure of the **QuizMaster** system is designed to provide a reliable, scalable, and normalized foundation for all exam-related operations. The focus is on minimizing redundancy and clearly organizing data related to questions, answers, teachers, scores, and exam sessions.

The system uses a **relational MySQL database**, and all tables follow 3NF normalization to support performance and data consistency. Students do not log in — instead, they participate in exams by entering basic information at the exam start screen. Only **teachers** can log in to the system using the **users** table.

5.2 Core Tables

The QuizMaster system includes the following **four main tables**:

Sheet				
IAJLЦ.466538.005 EN				
G. She ets	No document	Sig n.	Da te	25

Table 5.1 – Core Tables and Descriptions

Table Name	Description
users	Stores teacher accounts who log in and create/manage exams
questions	Stores all exam questions, their options, correct answer, and optional audio files
scores	Stores the total score and session data for each student who completed an exam
exam_results	Stores detailed results of each question answered in a session (per student per question)

Note: Tables like **admin** or **moderator** are defined in roles but not actively used in the system.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 26
---------	------------	----------------	-----------	----------	--------------------	-------------

5.3 Table Structures

Table 5.2 – Users Table

Field	Type	Description
user_id	INT (PK)	Unique identifier
user_name	VARCHAR	Username for login
user_password	VARCHAR	Encrypted password
user_authority	ENUM	Only user role is used actively
created_at	TIMESTAMP	Account creation date

This table defines the structure of the users table in the QuizMaster system, which stores information about teachers who manage exams, including their unique ID, username, password, authority level, and account creation date.

G. Ch	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 27
----------	------------	----------------	-----------	----------	--------------------	-------------

Table 5.3 – Questions Table

Field	Type	Description
id	INT (PK)	Unique question ID
type	ENUM	Question type: reading, listening
question_text	TEXT	The text of the question
option_a - option_d	VARCHAR	Answer options
correct_option	CHAR(1)	Correct choice (A, B, C, or D)
audio_file	VARCHAR	Path to audio file for listening questions
created_at	TIMESTAMP	Creation date
status	BOOLEAN	Active status (1 = visible)
exam_duration_seconds	INT	Duration for this question set

This table outlines the structure of the questions table in the QuizMaster system, detailing the fields for storing question data such as unique ID, type, text, answer options, correct answer, audio file path, creation date, status, and exam duration.

G. No	Sheets No	document n.	Sig Date	IAЛЦ.466538.005 EN	Sheet 28
----------	--------------	----------------	-------------	--------------------	-------------

Table 5.4 – Scores Table

Field	Type	Description
id	INT (PK)	Unique score ID
user_name	VARCHAR	Name entered by the student
user_email	VARCHAR	Email entered by the student
score	INT	Total score received
exam_duration	VARCHAR	Time spent on exam (readable format)
exam_session_id	VARCHAR	Unique session ID (links to exam_results)

No login required by students name and email are collected once per session.

G. Ch	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 29
----------	------------	----------------	-----------	----------	--------------------	-------------

Table 5.5 – Exam Results Table

Field	Type	Description
id	INT (PK)	Unique result ID
user_name	VARCHAR	Student name
user_email	VARCHAR	Student email
question_id	INT (FK)	Refers to question answered
user_answer	CHAR(1)	Answer chosen
is_correct	BOOLEAN	1 = correct, 0 = incorrect
date	TIMESTAMP	Timestamp of submission
exam_session_id	VARCHAR	Links this result to a session in scores

This table defines the structure of the `exam_results` table in the QuizMaster system, which stores detailed results of each question answered by students, including unique result ID, student details, question ID, user answer, correctness, submission timestamp, and session ID linking to the scores table.

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet 30
----	--------	-------------	------	------	--------------------	-------------

Table 5.6 – Entity Relationships

Relationship	Description
users (teacher) - creates - questions	Each teacher in the users table creates and manages multiple questions.
questions - appear in - exam_results	Each question can be answered multiple times by different students in various sessions.
exam_results - grouped by - exam_session_id	A unique session ID is used to group all answers submitted during one test attempt.
scores → summarizes → session results	The scores table provides the overall result (total score & duration) for each session using exam_session_id .

This table outlines the relationships between entities in the QuizMaster system, detailing how teachers create questions, questions appear in exam results, exam results are grouped by session ID, and session results are summarized in the scores table for overall performance tracking.

G. Sheets		No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet
						31

5.5 Normalization Strategy

All tables follow **Third Normal Form (3NF)**:

Table 5.7 – Normalization Strategy and Application

Rule	How it is applied in QuizMaster
1NF: Atomic values	No repeating groups or arrays
2NF: Full functional dependency	Every non-key field depends on the full key
3NF: No transitive dependency	Only direct dependencies to primary key

This table outlines the normalization strategy applied to the QuizMaster system, detailing the rules (1NF, 2NF, 3NF) and how they are implemented to ensure all tables follow Third Normal Form (3NF).

5.6 Summary

The database architecture of **QuizMaster** is minimal, efficient, and optimized for exam session management. The tables are closely tied to each other via `exam_session_id` and the system avoids redundant data through normalization. Teachers manage the system via login, while students provide minimal input and have no persistent user accounts, which simplifies user handling and increases privacy and speed.

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet
32						

5.7 ER Diagram

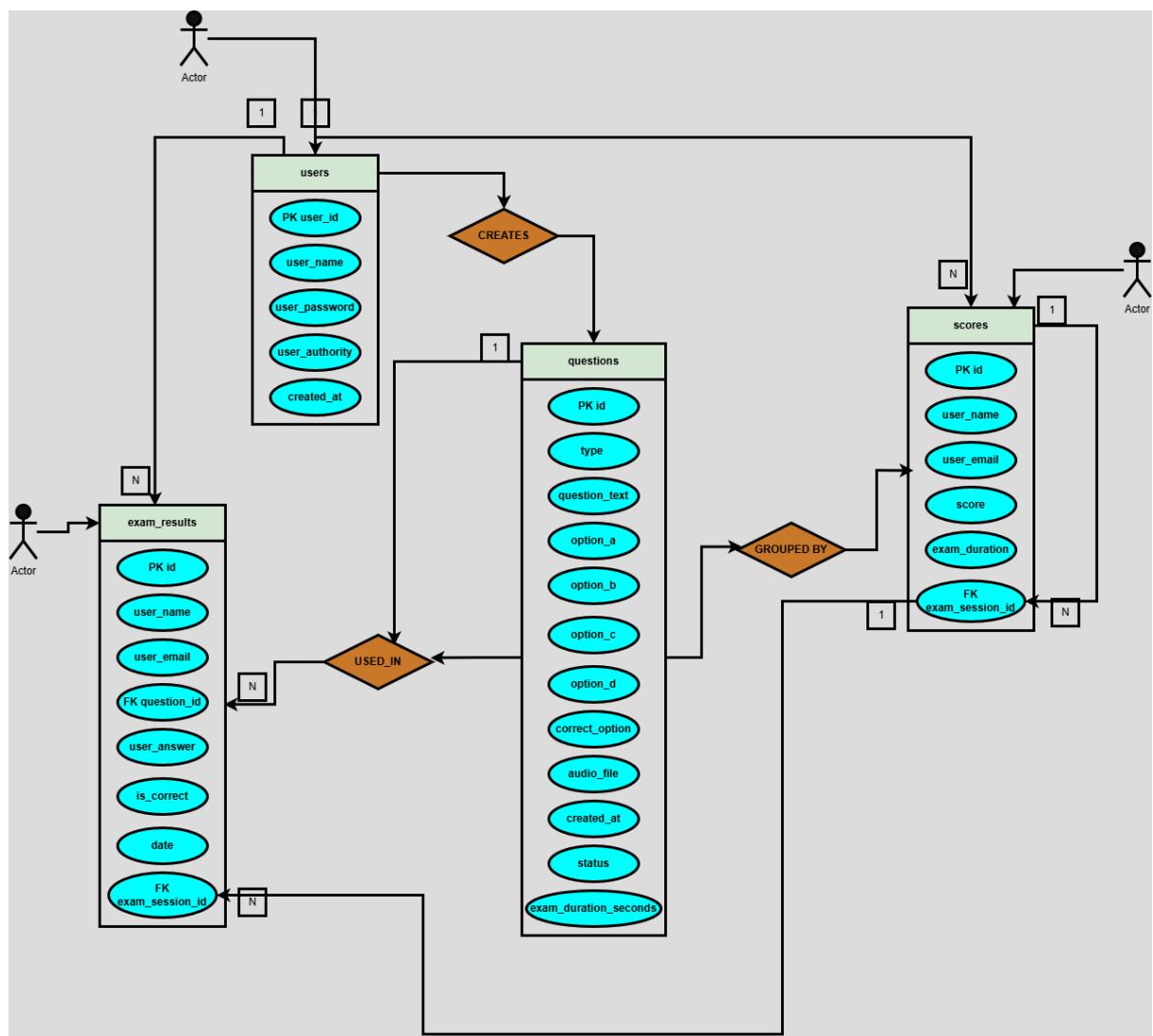


Figure 5.1 – Entity-Relationship Diagram (ERD) of QuizMaster

						Sheet
G. Ch	She ets	No document	Sig n.	Date	IAЛЦ.466538.005 EN	33

Entity Relationship Diagram Explanation

The Entity Relationship (ER) diagram for the **QuizMaster System** is designed to illustrate how different data entities interact with each other within the system. This model represents the core database structure and serves as a reference for understanding how quizzes are created, managed, and evaluated.

Users Table

- Represents all authorized individuals using the system (mainly teachers).
- Attributes include: **user_id** (Primary Key), **user_name**, **user_password**, **user_authority**, and **created_at**.
- Each user can create multiple questions and access results.

Questions Table

- Stores all questions added by users.
- Each question includes its type, text, multiple choices, correct option, optional audio file, creation time, status, and allotted time in seconds.
- **user_id** (FK - not directly shown here) implies that each question is **created by a single user**, establishing a **1:N relationship** between users and questions.

Scores Table

- Logs the final score for each exam session by a user.
- Attributes: **id** (PK), **user_name**, **user_email**, **score**, **exam_duration**, and **exam_session_id** (FK).
- A user may have **multiple score entries**, especially across different sessions.

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet
						34

Exam_Results Table

- Stores detailed logs of each question answered during an exam session.
- Attributes include question ID (FK), user answer, correctness flag, session ID (FK), and timestamp.
- Related to both **questions** and **scores** through **many-to-one (N:1)** relationships.

Relationships

- **CREATES:** A user can create many questions (1:N).
- **USED_IN:** Each question can be used in many exam results (1:N).
- **GROUPED_BY:** Questions are grouped into sessions, and each session has multiple result records.
- **Scores ↔ Exam_Results:** Each score summary (1) is related to many detailed result entries (N) via the shared **exam_session_id**.

5.8 Summary

This ER diagram efficiently captures the logical structure of the QuizMaster application. It ensures data normalization, clear referential integrity, and optimized relationships for querying both individual results and cumulative statistics. This structure lays a solid foundation for implementing the database schema and driving the backend logic of the system

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet
						35

5.9 Table Creation Scripts (SQL)

The following SQL scripts define the structure of each database table used in the **QuizMaster** system. These scripts ensure that the database is created with proper **primary keys**, **foreign keys**, and **data types**, following the relational design principles discussed earlier.

users Table

```
CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `user_name` varchar(255) NOT NULL,
  `user_password` varchar(255) NOT NULL,
  `user_authority` enum('admin','user','moderator') NOT NULL DEFAULT 'user',
  `created_at` timestamp NULL DEFAULT current_timestamp()
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Figure 5.2 – SQL Table Creation: Users Table

questions Table

```
CREATE TABLE `questions` (
  `id` int(11) NOT NULL,
  `type` enum('reading','listening') NOT NULL,
  `question_text` text NOT NULL,
  `option_a` varchar(255) NOT NULL,
  `option_b` varchar(255) NOT NULL,
  `option_c` varchar(255) NOT NULL,
  `option_d` varchar(255) NOT NULL,
  `correct_option` enum('A','B','C','D') NOT NULL,
  `audio_file` varchar(255) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT current_timestamp(),
  `durum` int(11) NOT NULL DEFAULT 1,
  `exam_duration_seconds` int(11) DEFAULT 3600
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Figure 5.3 – SQL Table Creation: Questions Table

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet
						36

scores Table

```
CREATE TABLE `scores` (
    `id` int(11) NOT NULL,
    `user_name` varchar(255) NOT NULL,
    `user_email` varchar(255) NOT NULL,
    `score` int(11) NOT NULL,
    `exam_duration` varchar(50) NOT NULL,
    `exam_session_id` varchar(255) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Figure 5.4 – SQL Table Creation: Scores Table

- user_name and user_email are not linked via foreign key to users because students do not log in. Data is entered manually at exam start.

exam_results Table

```
CREATE TABLE `exam_results` (
    `id` int(11) NOT NULL,
    `user_name` varchar(255) NOT NULL,
    `user_email` varchar(255) NOT NULL,
    `question_id` int(11) NOT NULL,
    `user_answer` char(1) NOT NULL,
    `is_correct` tinyint(1) NOT NULL,
    `date` datetime NOT NULL DEFAULT current_timestamp(),
    `exam_session_id` varchar(100) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Figure 5.5 – SQL Table Creation: exam_results Table

- question_id links each answer to the corresponding question in the questions table.

Sheet
37
G. Sheets No document Sig Date ІАЛЦ.466538.005 EN

5.10 Summary

These SQL scripts ensure that all tables are properly structured, with relationships defined where appropriate. The system uses minimal student data, with focus on teachers managing content and exams. All sessions are tracked using the **exam_session_id** key, ensuring logical grouping of results and scores.

Sheet					
IAЛЦ.466538.005 EN					
38					
<table border="1"><tr><td>G.</td><td>Sheets</td><td>No document</td><td>Sig n.</td><td>Date</td></tr></table>	G.	Sheets	No document	Sig n.	Date
G.	Sheets	No document	Sig n.	Date	

6. SYSTEM ARCHITECTURE

6.1 Overview

The **QuizMaster** system follows a **client-server web architecture**, developed using the **PHP + MySQL** stack and deployed locally through **XAMPP**. The system is designed to separate teacher and student workflows, allowing teachers to manage exams and students to take them without requiring authentication.

The system components are divided into:

- **Frontend (Client):** HTML/CSS/JavaScript-based interfaces.
- **Backend (Server):** PHP scripts that handle data processing and logic.
- **Database:** MySQL database named quiz.

6.2 Client-Side Components

The client-side presents two distinct interfaces:

6.2.1. Teacher/Admin Interface

- Built using classic HTML forms and styled with multiple CSS libraries (located in **/assets/css/**).

Sheet
ІАЛЦ.466538.005 EN
39

- Admin UI components include:
 - Question creation form (**sinav-ekle.php**)
 - Question editing (**sinav-duzenle.php**)
 - Result pages (**sinav-sonuclari.php, sinav-detay.php**)
 - Login page (**login.php**)

6.2.2. Student Interface

- Students access the exam via `exam.php`.
- No login is required — only **name** and **email** are entered.
- A dynamic multiple-choice interface is presented.
- Answers are submitted to `submit_exam.php`.

6.3 Server-Side Logic

The core server logic is written in **PHP**. Main files include:

Sheet
IAЛЦ.466538.005 EN
40

Table 6.1 – Server-Side Logic and PHP Files

File	Purpose
baglan.php	Database connection script
exam.php	Loads questions, handles timer and student data
submit_exam.php	Calculates score, stores results
sinav-ekle.php	Form to add new question
sinav-duzenle.php	Edit existing questions
sinav-sonuclari.php	Shows exam score summaries
sinav-detay.php	Shows detailed answers per session
login.php	Authenticates teachers
islem.php	Contains logic for DB operations like INSERT, UPDATE, DELETE

G.	Sheets	No document	Sig.	Date	IAJLЦ.466538.005 EN	Sheet
						41

6.4 Data and Session Flow

6.4.1 Teacher Workflow:

1. Teacher logs in via login.php (session starts)
2. Can create, edit, and view questions
3. Views student results via result pages

6.4.2 Student Workflow:

1. Student accesses exam.php and enters name/email
2. Exam is loaded (one or multiple questions)
3. A **session ID** is created per attempt
4. Upon submission, submit_exam.php:
 - Evaluates answers
 - Calculates score
 - Stores entries in exam_results and scores tables

Note: exam_session_id is crucial it groups all answers and final score per student per exam.

Sheet
ІАЛЦ.466538.005 EN 42

6.5 File Structure

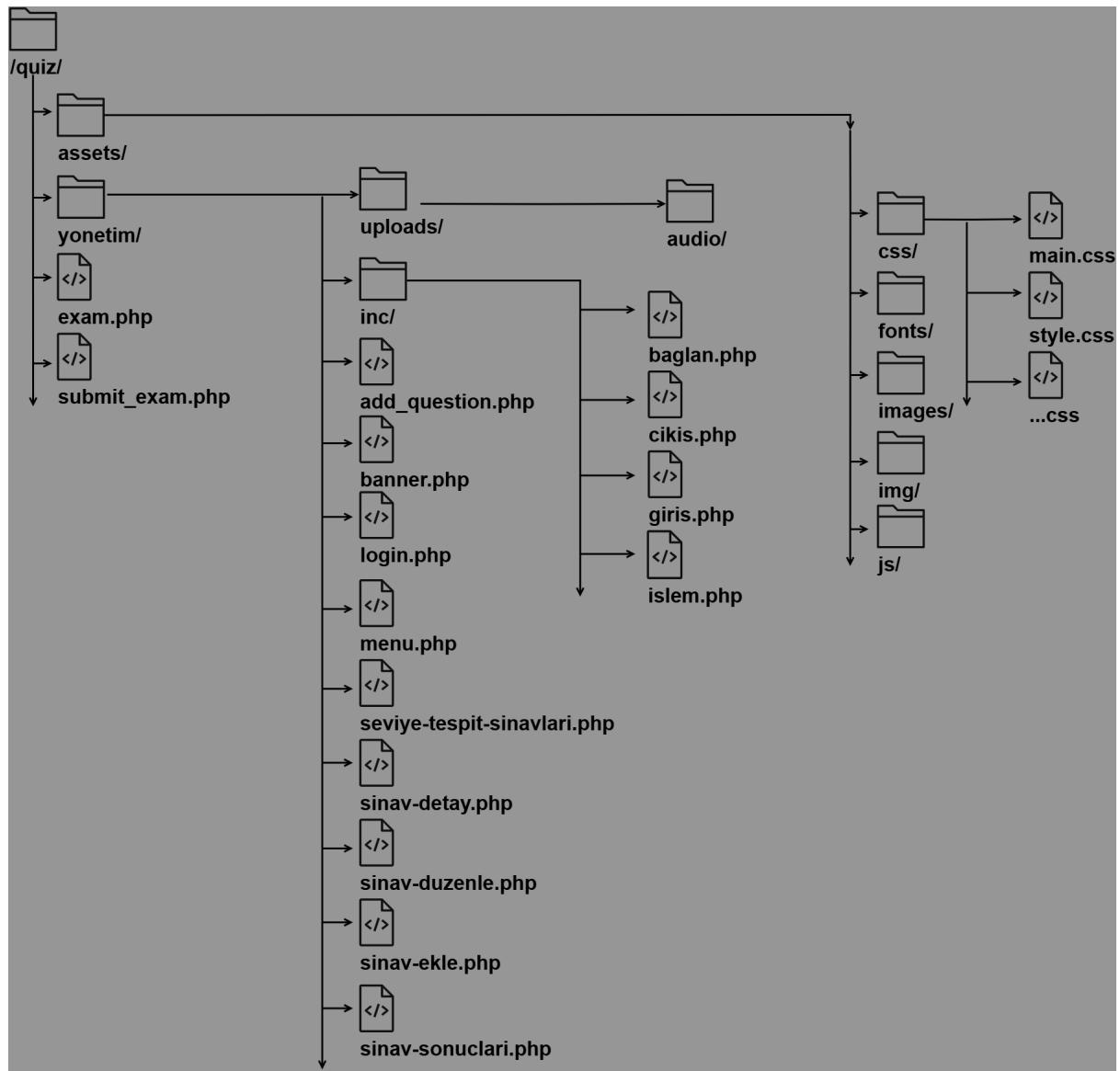


Figure 6.1 – File Structure

The file structure of the QuizMaster project is modular and separated into functional folders for frontend assets, admin scripts, and student interfaces. The above diagram visualizes the actual folder hierarchy of the project directory.

G. No	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet
						43

6.6 Architecture Highlights

- Uses MySQL for persistent storage.
- All student interactions are stateless (no login), managed via session ID.
- Admins are authenticated via session variables.
- CSS files are modular but large (style.css includes unused legacy code).
- All exam timing is managed client-side (JavaScript countdown).
- Form submissions are secured via POST method and validated server-side.

6.6.1 System Architecture Diagram

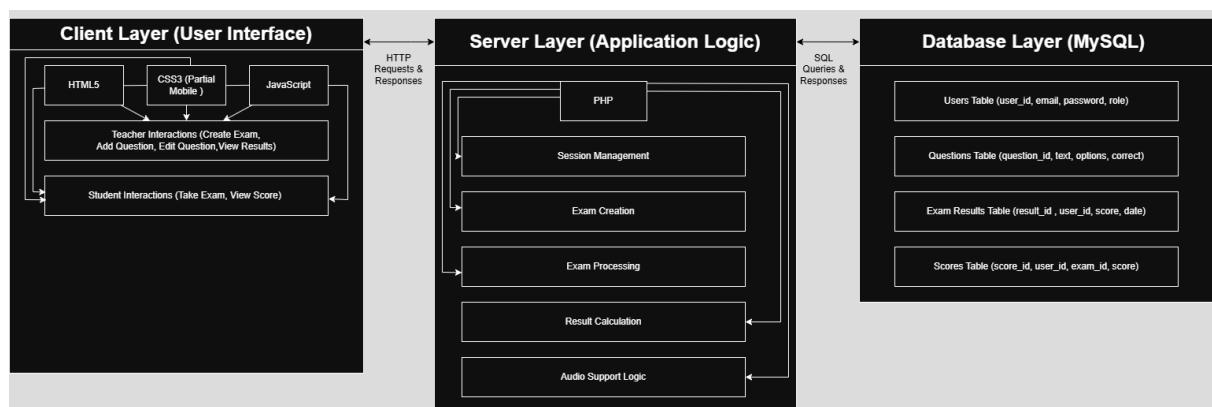


Figure 6.2 – System Architecture Diagram of QuizMaster

This diagram illustrates the architecture of the QuizMaster system, showcasing the client layer (built with HTML5, CSS3, and JavaScript, with partial mobile compatibility), the server layer (powered by PHP for session management, exam creation, exam processing, result calculation, and audio support), and the MySQL database layer (storing users, questions, exam results, and scores).

Sheet
G. Sheets No document n. Sig Date ІАЛЦ.466538.005 EN 44

6.7 Summary

The QuizMaster architecture is practical, modular, and minimal. It enables real-time exam creation and evaluation without relying on heavy frameworks. With a focus on teacher-only authentication, stateless student handling, and clean PHP-MySQL logic, the system delivers a reliable test experience under a simple yet effective architectural model.

Sheet
ІАЛЦ.466538.005 EN
45

G.	Sheets	No document	Sig n.	Date
----	--------	-------------	--------	------

7. INTERFACE AND UI DESIGN

7.1 Design Principles

The interface of the QuizMaster system was built upon the following key principles:

- **Simplicity:** A clean and distraction-free layout ensures that users can focus solely on tasks like creating, taking, or evaluating exams.
- **Responsiveness:** All pages are styled with responsive CSS to ensure compatibility with desktops, laptops, and tablets.
- **Separation of Roles:** The system visually and functionally separates the teacher (admin panel) and the student (exam interface).
- **Consistency:** Consistent color schemes, typography, button styles, and layout patterns are applied across all pages to support usability and brand identity.

7.2 Admin Panel (Teacher Interface)

Teachers log in via the secure login panel and access a centralized dashboard. Key features of this interface include:

Login Screen: Username and password fields are validated; incorrect attempts are blocked.

Sheet
46
G. Sheets No document Sig n. Date ІАЛЦ.466538.005 EN

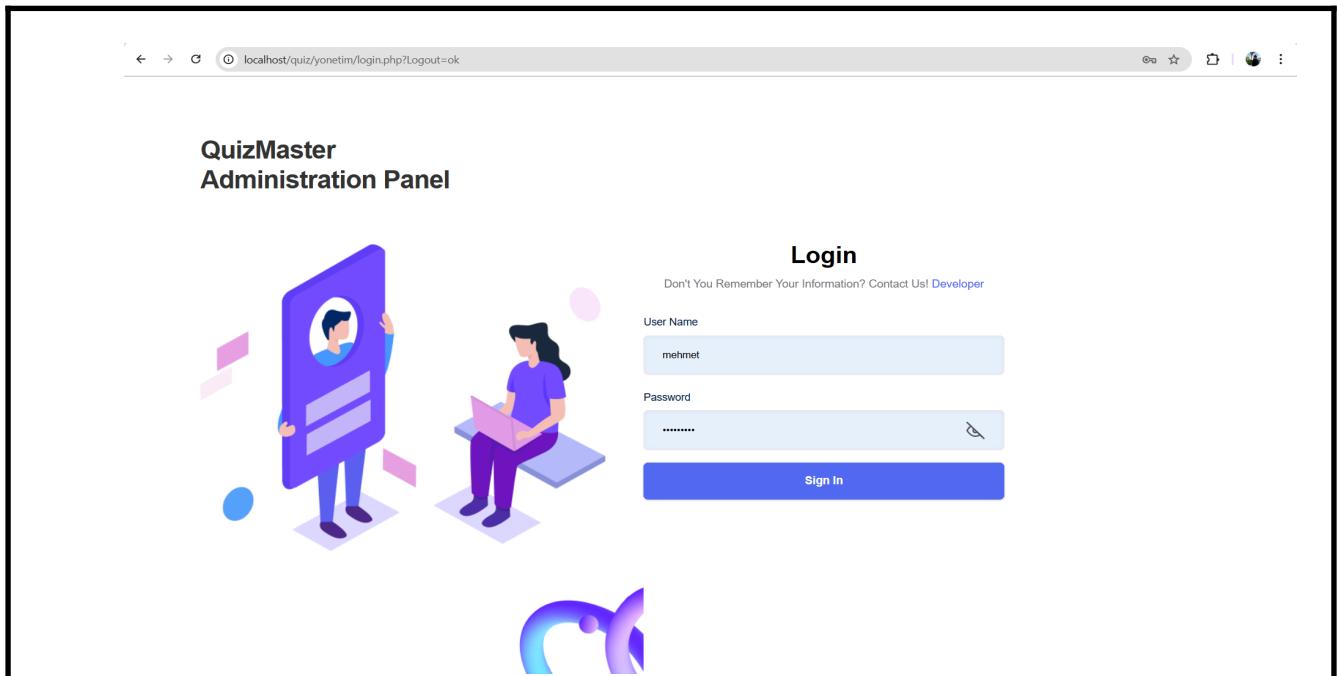


Figure 7.1 – Login Screen of the QuizMaster Admin Panel

Exam Management Dashboard: After login, teachers can view existing questions, their status, creation time, and access edit/delete actions.

Question	Date	Status	Actions
What does HTML stand for?	2025-03-30 17:04:06	✓	⋮
Which language is used for styling web pages?	2025-03-30 17:04:41	✓	⋮
what is he name ?	2025-03-27 18:52:30	✓	⋮
who?	2025-03-26 17:13:58	✓	⋮

Figure 7.2 – Question Management Dashboard

					Sheet
G. H.	She ets	No document	Sig n.	Da te	47
					IAЛЦ.466538.005 EN

Add/Edit Question Page: Teachers can add new questions by specifying the question type (Reading/Listening), question text, four options (A–D), the correct answer, and optionally upload an audio file (for listening questions).

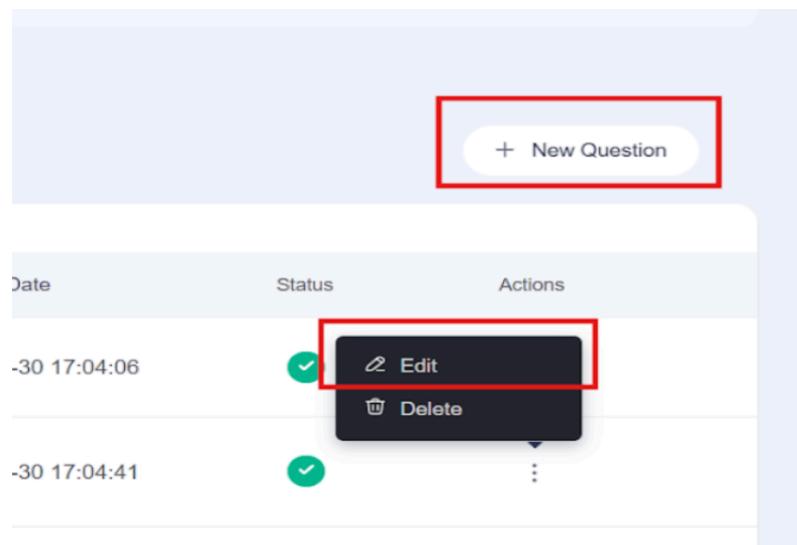


Figure 7.3 – Question Creation and Editing Interface

Figure 7.4 – Add Question Interface

					IAЛЦ.466538.005 EN	Sheet
G.	She ets	No document	Sig n.	Date		48

The screenshot shows a web-based application for managing exam questions. At the top, the URL is `localhost/quiz/yonetim/sinav-duzenle.php?id=67`. The main title is "Update Question". Under "Question Type", a dropdown menu is set to "Listening" (highlighted with a red box). The "Question Text" field contains the word "listening". Below it, under "Options", there are four entries: A: Mehmet, B: Ali, C: Ece, and D: Eda. The "Correct Option" dropdown is set to "A". In the "Audio File for Listening Question" section, a button labeled "Dosya Seç" (Select File) is highlighted with a red box; its tooltip indicates the file path "mynameismehmet.mp3". At the bottom are "Update Question" and "Back" buttons.

Figure 7.5 – Edit Question Interface

Exam Results Page: Lists all exam sessions taken by students, showing name, email, total score, and a "Details" button.

Results		
Mail	Score	Actions
cakyagmur@gmail.com	60	Details
a@gmail.com	80	Details
emoklbc@icloud.com	90	Details

Figure 7.6 – Student Exam Results Listing

G.	Sheets	No document	Sig n.	Date	IAЛЦ.466538.005 EN	Sheet 49
----	--------	-------------	--------	------	--------------------	----------

Exam Detail View: Clicking on "Details" displays each question from a session, the student's answer, the correct answer, and whether it was correct or not. Duration and score summary are included.

The screenshot shows a web-based application titled 'QuizMaster' with a chameleon logo. The user is identified as 'Administrator'. On the left, there's a sidebar with 'MENU' options: 'Level Determination Exam', 'Exam Results', and 'USER Exit'. Below the menu is an 'Online Test' section featuring an illustration of a person sitting at a desk with a laptop. The main content area is titled 'Level Assessment Exam' and shows 'Test Review' details. It includes a table with columns: No., Type, Question, Answer Given, Correct Answer, and Result. The table shows two questions: Question 1 asks about JavaScript variable declaration, with 'B' given and 'B' correct, marked as 'Correct'. Question 2 asks about Python print output, with 'B' given and 'A' correct, marked as 'Wrong'. Summary statistics at the top right indicate a score of 90 and a duration of 0 hours 1 minute 13 seconds.

Figure 7.7 – Detailed View of a Completed Exam

					IAЛЦ.466538.005 EN	Sheet
G. Sheets		No document	Sig n.	Date		50

The screenshot shows a completed exam interface with the following details:

- URL:** localhost/quiz/yonetim/sinav-detay.php?id=memoklbc@icloud.com
- Questions:**
 - Q8: Which symbol is used to start a single-line comment in Python? (Answer: #, Status: Wrong)
 - Q9: What is the output of the following Python code? print(2 + 3 * 4) (Answer: 14, Status: Wrong)
 - Q10: Who? (Answer: A, Status: Wrong)
- Summary:**
 - Exam Date: 2025-03-30 18:42
 - Score: 100
 - Duration: 0 Hour 3 Minute 22 Seconds
- Previous Questions:**
 - Q1: Which language is used for styling web pages? (Answer: CSS, Status: Correct)
 - Q2: What is his name? (Answer: John, Status: Correct)
 - Q3: Which symbol is used to start a single-line comment in Python? (Answer: #, Status: Correct)

Figure 7.8 – Detailed View of a Completed Exam

7.3 Student Interface

Students do not register or log in; instead, they simply provide their name and email at the beginning of each exam session.

- **Start Exam Page:** Collects basic identity information and then launches the exam.

					IAJLЦ.466538.005 EN	Sheet
G. Ch	She ets	No document	Sig n.	Date		51

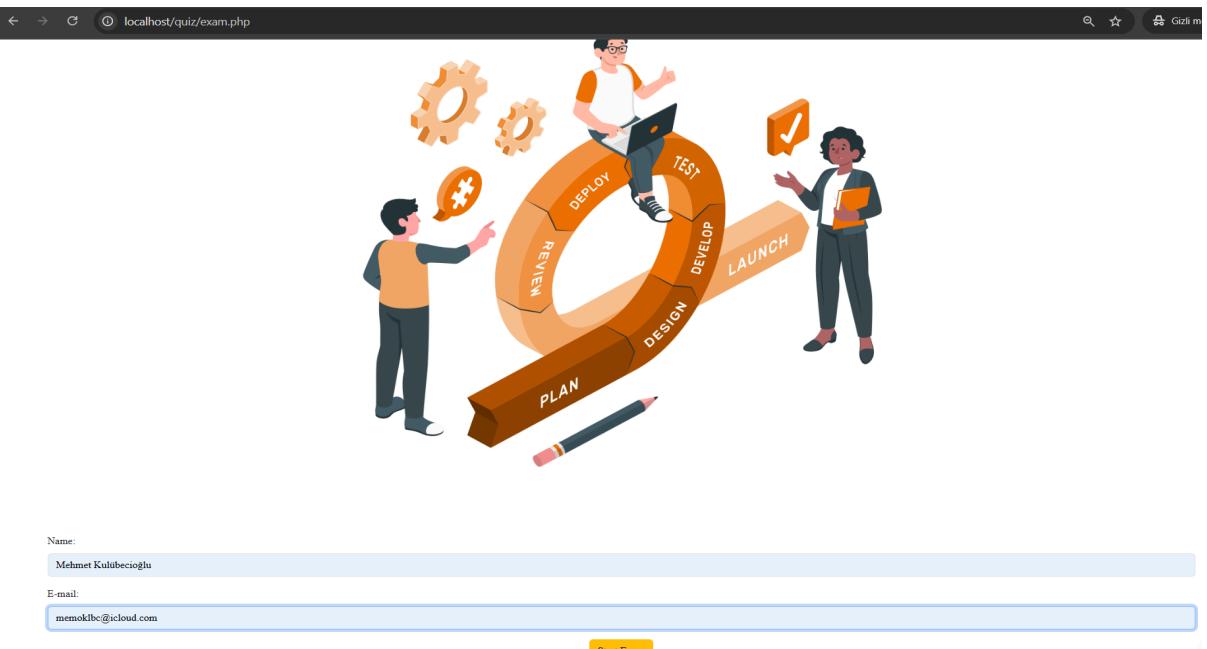


Figure 7.9 – Exam Entry Screen for Students

- **Exam Page:**

Figure 7.10 – Question View and Audio Playback in Exam

G.	Sheets	No document	Sig n.	Date	IAЛЦ.466538.005 EN	Sheet 52
----	--------	-------------	--------	------	--------------------	----------

- Questions are displayed one by one.
- If the question type is "Listening", an audio file is embedded with playback controls.
- Navigation buttons allow users to move forward or back between questions.
- A real-time timer appears in the corner.

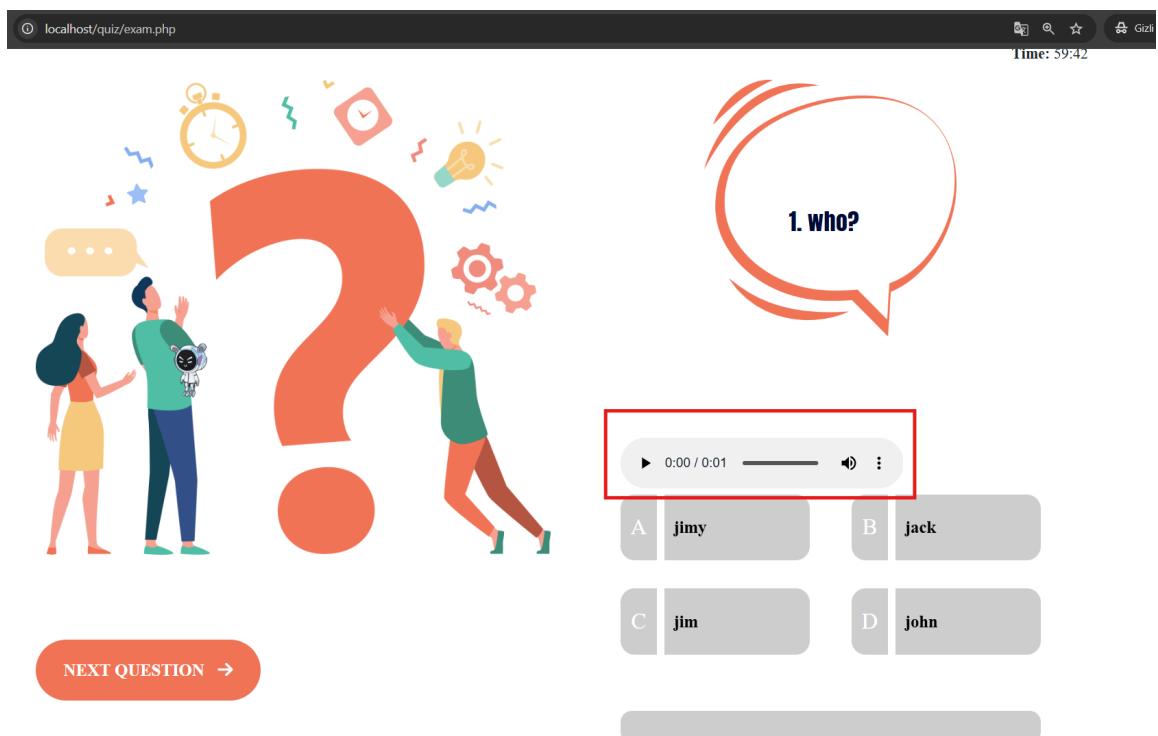


Figure 7.11 – Question View and Audio Playback in Exam

- **Submission Screen:**

- After finishing the exam, students see a thank-you message with their score.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 53
---------	------------	----------------	-----------	----------	--------------------	-------------

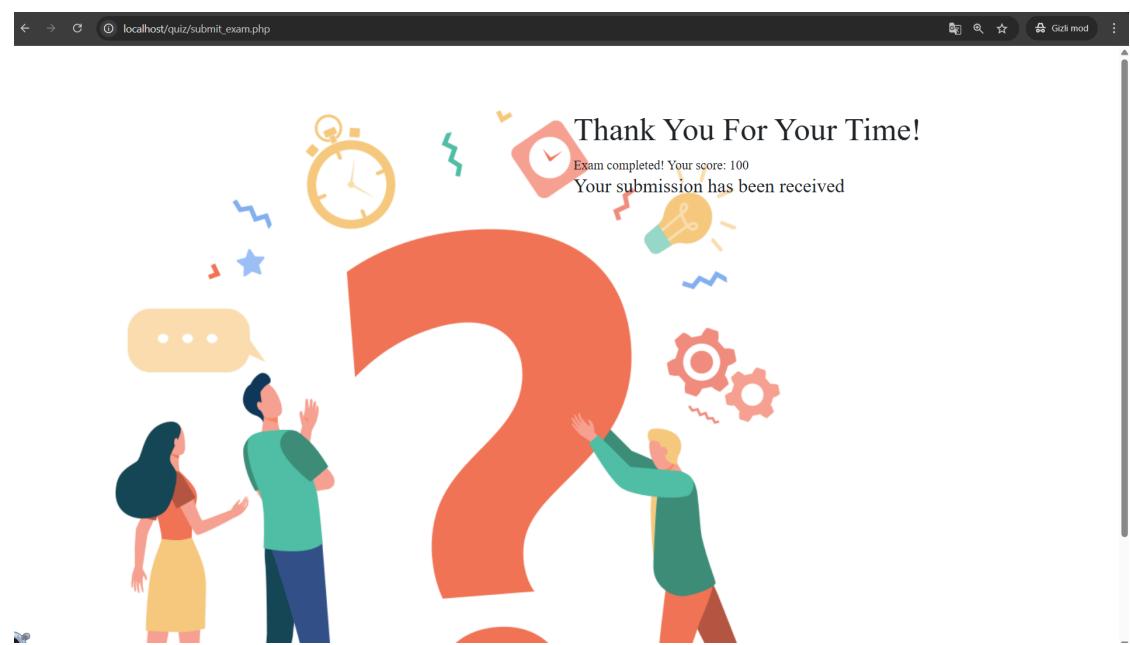


Figure 7.12 - Exam Submission Confirmation

7.4 Styling and Assets

- **CSS:** Custom CSS files (main.css, style.css) are used to ensure a professional, modern design. Color palette focuses on clean blues, whites, and subtle gradients.
- **Illustrations:** SVG/PNG assets are used to add visual engagement without affecting loading performance.
- **Directory Structure:** The project files are organized under /quiz/ with separate folders for assets (css/, images/, audio/) and logic files.

Sheet		
ГН Sheets No document Sig. Date	IAЛЦ.466538.005 EN	54

8. IMPLEMENTATION

This section describes how the QuizMaster system was developed. It covers the technical implementation of the backend logic, UI functionality, and data flow. The development is divided into three main components: the admin panel (for teachers), the student interface (for exam takers), and the overall exam logic.

8.1 Admin Panel

The admin panel is designed for authorized users (teachers) to manage exam questions and view student results. Only users with accounts stored in the users table can access this panel.

- **login.php:** Provides a secure login screen that verifies credentials against the users table.
- **menu.php:** Sidebar navigation for switching between question management and results view.
- **seviye-tespit-sinavlari.php:** Displays all created questions. Each question can be edited or deleted.
- **sinav-ekle.php:** Allows admins to add new questions. Fields include question type (reading/listening), text, multiple choices, correct answer, and optional audio file.
- **sinav-duzenle.php:** Enables admins to update existing questions.
- **sinav-sonuclari.php:** Lists exam results per student. Teachers can view individual performance.
- **sinav-detay.php:** Shows detailed results for each exam session, including question-wise correctness and duration.

G. H.	She ets	No document	Sig n.	Da te	IAJIC.466538.005 EN	Sheet
						55

Key Features:

- Clean and responsive UI.
- Session-based access control for security.
- Questions are stored in the questions table.
- Scores and session info are stored in the scores table.

8.2 Student Interface

The student interface is designed to be simple, intuitive, and accessible — even for first-time users. Unlike teachers, students do not log in; they simply provide their name and email at the start of an exam session. The system uses this information along with a generated exam_session_id to track results.

Start Exam Page (exam.php)

- Students are greeted with a welcome message and a form requesting basic identity information (name and email).
- Once submitted, the exam session begins immediately and a unique session ID is created.
- Questions are fetched from the database and displayed one at a time.

G. Sheets	No document	Sig n.	Date	IAЛЦ.466538.005 EN	Sheet 56

<p>Name: Mehmet Kulübecioğlu</p> <p>E-mail: memoklbc@icloud.com</p> <p style="text-align: center;">Start Exam</p>				
<p>Figure 8.1 – Exam Start Page</p>				
<h3>Exam Interface</h3> <p>The exam experience is minimal and focused:</p> <ul style="list-style-type: none"> ● Questions: Displayed one at a time with multiple-choice options (A–D). ● Listening Support: If the question type is "listening", an audio player appears at the top of the card. ● Timer: A countdown timer is displayed at the top right corner. The duration is fetched from the <code>exam_duration_seconds</code> field in the database. ● Navigation: Students can move between questions using Next and Previous buttons. ● Validation: Students cannot proceed without selecting an option. 				
G. Sheets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN Sheet 57

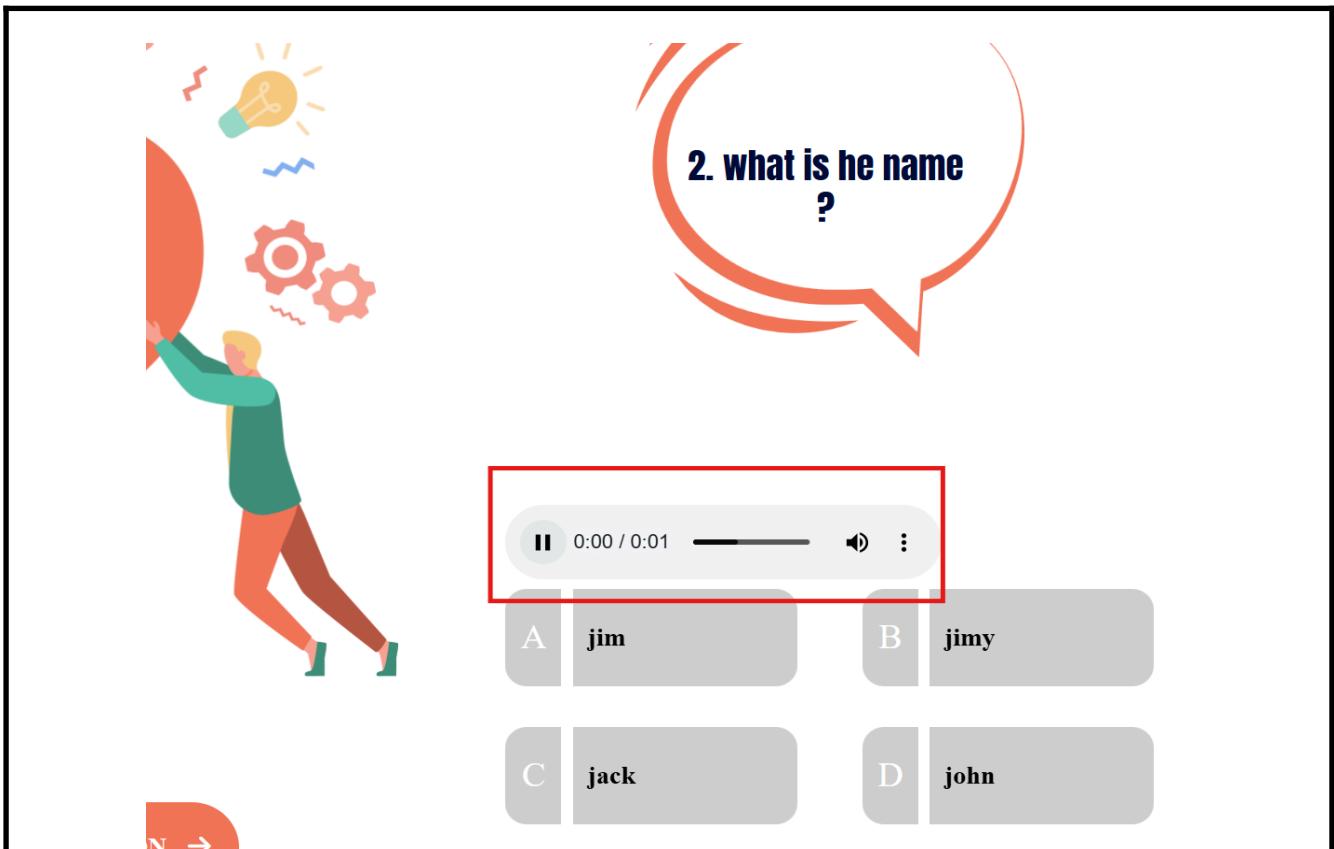


Figure 8.2– Audio-Embedded Listening Question

Exam Submission (`submit_exam.php`)

- Upon finishing the exam, answers are submitted via a POST request.
- The script evaluates the number of correct answers, calculates the final score, and stores all data in:
 - `exam_results`: One entry per question
 - `scores`: Total score and session summary
- A confirmation message is displayed with a "Thank you" note and student score.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet
						58

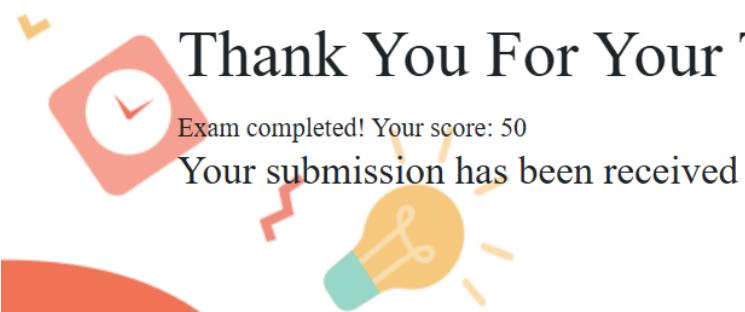


Figure 8.3 – Exam Completion and Submission Confirmation

Session Management

- A random string is generated as exam_session_id upon exam start.
- All answer records and final scores are grouped by this session ID.
- This approach eliminates the need for persistent student accounts and provides a lightweight, privacy-friendly solution.

Key Features:

- No login required
- Mobile-friendly layout
- Audio support for listening questions
- Real-time countdown
- Seamless submission and result storage

Sheet
G. Sheets No document Sig. n. Date ІАЛЦ.466538.005 EN 59

8.3 Exam System Logic

The core exam functionality of the QuizMaster system is managed through two main files: `exam.php` and `submit_exam.php`. These scripts handle the loading of questions, collection of student answers, session management, and score calculation.

8.3.1. Starting the Exam (`exam.php`)

- When the student enters their name and email, a unique `exam_session_id` is generated using random functions and stored in PHP session variables.
- The list of questions is fetched from the questions table using SQL queries, and displayed one at a time using a basic loop or pagination.
- The `exam_duration_seconds` field is retrieved from the database to control the countdown timer (managed via JavaScript).
- Students select their answer (A–D) and proceed to the next question.

8.3.2. Answer Collection and Evaluation (`submit_exam.php`)

- After the last question, all answers are submitted via a form (usually as arrays or separate POST values).
- The script checks each selected answer against the `correct_option` value from the questions table.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet
						60

- For each answer:
 - A new row is inserted into the exam_results table, storing the question ID, student's selected answer, correctness flag (is_correct), and session ID.
- The total number of correct answers is counted and stored as the score in the scores table, along with the student's name, email, and exam duration.

8.3.3. Session Tracking with exam_session_id

- Each exam attempt is assigned a unique exam_session_id.
- All entries in both exam_results and scores reference this ID.
- This allows grouping and filtering results per session without requiring student login or persistent accounts.

	<input type="button" value="← T →"/>	<input type="button" value="id"/>	<input type="button" value="user_name"/>	<input type="button" value="user_email"/>	<input type="button" value="question_id"/>	<input type="button" value="user_answer"/>	<input type="button" value="is_correct"/>	<input type="button" value="date"/>	<input type="button" value="exam_session_id"/>
<input type="checkbox"/>	<input type="button" value="Düzenle"/>	<input type="button" value="Kopyala"/>	<input type="button" value="Sil"/>	1092 Mehmet Kulübecioğlu memokibc@icloud.com	62	B	1	2025-03-30 19:00:19	session_67e96a8e433aa1.70152760
<input type="checkbox"/>	<input type="button" value="Düzenle"/>	<input type="button" value="Kopyala"/>	<input type="button" value="Sil"/>	1093 Mehmet Kulübecioğlu memokibc@icloud.com	64	A	1	2025-03-30 19:00:20	session_67e96a8e433aa1.70152760
<input type="checkbox"/>	<input type="button" value="Düzenle"/>	<input type="button" value="Kopyala"/>	<input type="button" value="Sil"/>	1094 Mehmet Kulübecioğlu memokibc@icloud.com	65	C	0	2025-03-30 19:00:21	session_67e96a8e433aa1.70152760

Figure 8.4 – Exam Results with Session ID Tracking

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 61
---------	------------	----------------	-----------	----------	--------------------	-------------

8.3.4. Duration Tracking

- The timer begins client-side (JavaScript) when the exam starts.
- The total time spent is calculated and stored as a string (for example, 05:00) in the exam_duration field of the scoreboard.
- The server does not enforce duration limits but relies on front-end countdown for UX control.

8.4 Summary

The exam logic in QuizMaster is streamlined and efficient. Using simple PHP session variables and minimal JavaScript, the system can:

- Load dynamic questions from a database
- Track answers and session progress
- Store individual responses and calculate results
- Group sessions by a unique session ID

This design ensures that the system remains lightweight, secure, and easy to maintain, while fulfilling its core educational purpose.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 62

9. TESTING AND RESULTS

9.1 Overview

After the implementation phase, the QuizMaster system was subjected to functional testing to ensure stability, usability, and correctness of operations. The tests focused on the following aspects:

- Validating teacher login and session control
- Ensuring question creation, editing, and deletion work as expected
- Testing the full student exam workflow (start, answer, submit)
- Verifying score calculations and answer correctness
- Checking database record consistency (exam_results and scores)
- Observing session management through exam_session_id

Each test case was executed manually using the browser-based interface on a local server via XAMPP.

Sheet
ІАЛЦ.466538.005 EN
63

9.2 Test Environment

Table 9.1 – Test Environment Components

Item	Description
Local Server	XAMPP (Apache + MySQL)
Browser	Google Chrome
Database	MySQL 8.0 (quiz database)
Operating System	Windows 11
PHP Version	PHP 8.x
Test Accounts	1 teacher (admin), multiple test users
Device	Laptop (15.6", 1920x1080 resolution)

This table outlines the key components and specifications of the test environment used for the QuizMaster system, including server, browser, database, and device details.

Sheet
ІАЛЦ.466538.005 EN

9.3 Admin Panel Test Cases

Table 9.2 – Admin Panel Test Cases

Test Case	Description	Expected Result	Status
TC-01	Teacher login with correct credentials	Redirects to dashboard	Passed
TC-02	Invalid login attempt	Displays error message	Passed
TC-03	Add question (Reading)	Saves to DB, visible in list	Passed
TC-04	Add question (Listening with audio)	Audio uploads correctly	Passed
TC-05	Edit/Delete question	Changes reflected in UI and DB	Passed

This table presents the test cases for the admin panel of the QuizMaster system, detailing the description, expected results, and status of each test.

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet
						65

localhost/quiz/yonetim/sinav-ekle.php

Add Question

Question Type:

Reading

Question Text:

Which company developed the Java programming language?

Options:

A: Google

B: IBM

C: Microsoft

D: Sun Microsystems

Correct Option:

D

Audio File for Listening Question:

Dosya Seç Dosya seçilmedi

Add Question **Back**

Figure 9.1 – Sample Reading Question Added via Teacher Panel

QuizMaster
Administrator

MENU

- Level Determination Exam
- Exam Results

USER

- Exit

Online Test

Level Determination Exams

Question	Date	Status	Actions
What does HTML stand for?	2025-03-30 17:04:06	<input checked="" type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Which language is used for styling web pages?	2025-03-30 17:04:41	<input checked="" type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
what is he name ?	2025-03-27 18:52:30	<input checked="" type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

+ New Question

Figure 9.2 – Question Management Dashboard with Edit/Delete Options

					IAЛЦ.466538.005 EN	Sheet
G.	Sheets	No document	Sig n.	Date		66

9.4 Student Exam Test Cases

Table 9.3 – Student Exam Test Cases

Test Case	Description	Expected Result	Status
TC-06	Student enters name/email and starts exam	Session created, first question shown	Passed
TC-07	Question with audio	Audio file is playable	Passed
TC-08	Answer submission without selecting option	Alert shown, cannot proceed	Passed
TC-09	Timer countdown	Stops at 0 and triggers auto-submit (optional)	Manual-only
TC-10	Exam submission	Answers stored, score calculated	Passed

This table details the test cases for the student exam interface of the QuizMaster system, including descriptions, expected results, and the status of each test.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 67
---------	------------	----------------	-----------	----------	--------------------	-------------

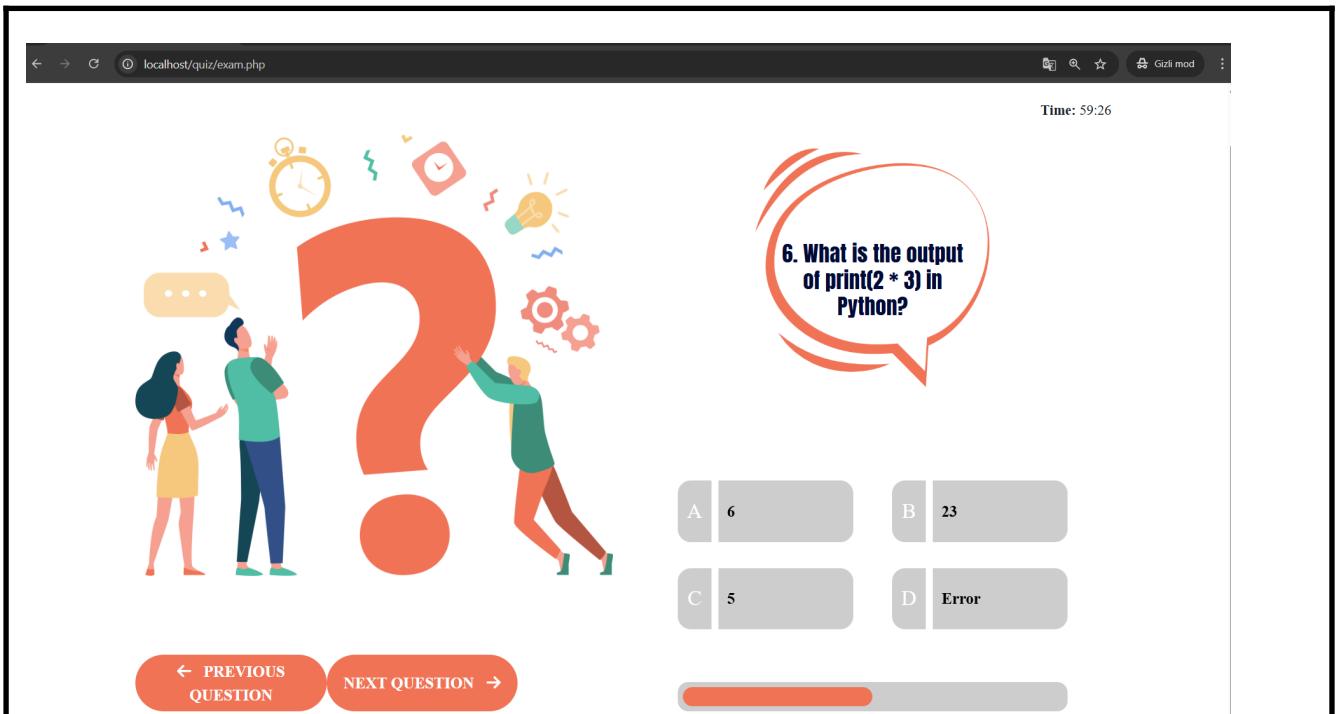


Figure 9.3 – Exam Interface with Navigation and Timer

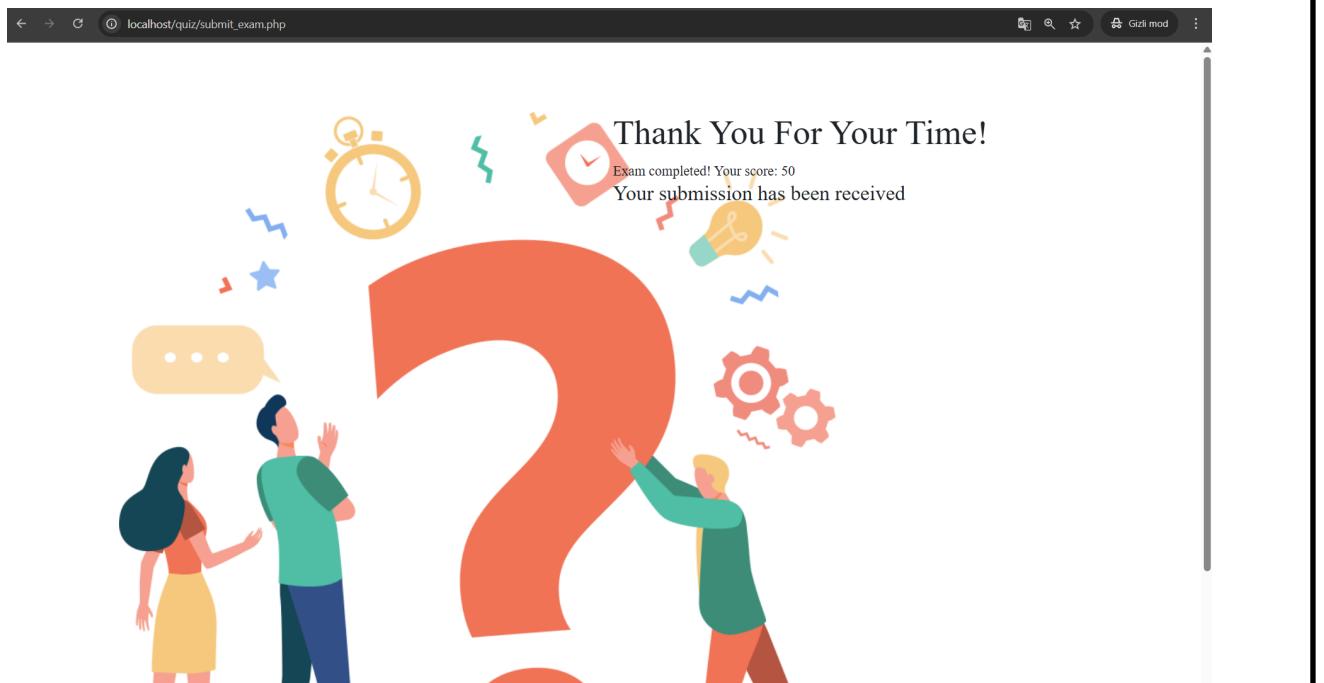


Figure 9.4 – Final Submission Screen with Score Display

					ІАЛЦ.466538.005 EN	Sheet
G. Ch	She ets	No document	Sig n.	Date		68

9.5 Database Verification

After test completion, the exam_results and scores tables were inspected:

- Each exam_session_id had matching entries in both tables
- is_correct flags matched with actual correct answers
- Scores matched the number of correct answers
- Duration was correctly recorded

The screenshot shows the phpMyAdmin interface with the following details:

- SQL query: `SELECT * FROM `scores``
- Buttons: Profil çıkart [Satır içi düzenle] [Düzenle] [SQL'i açıka] [PHP kodu oluştur] [Yenile]
- Filter: Tümünü göster (checkbox), Satır sayısı: 500, Satırları süz: Bu tabloda ara, Anahtara göre sırala: Yok
- Table Headers: id, user_name, user_email, score, exam_duration, exam_session_id
- Table Data:

	Düzenle	Kopyala	Sil	score	exam_duration
122	Mehmet Kulübecioğlu	memoklbc@icloud.com		30	0 Hour 0 Minute 21 Seconds
123	Mehmet Kulübecioğlu	memoklbc@icloud.com		100	0 Hour 3 Minute 22 Seconds
124	Mehmet Kulübecioğlu	memoklbc@icloud.com		70	0 Hour 0 Minute 33 Seconds
125	yağmur	6ocakyagmur@gmail.com		60	0 Hour 1 Minute 30 Seconds
126	eda	eda@gmail.com		80	0 Hour 2 Minute 20 Seconds
127	Mehmet Kulübecioğlu	memoklbc@icloud.com		90	0 Hour 1 Minute 13 Seconds
128	newstudent	newsrudent@gmail.com		50	0 Hour 1 Minute 5 Seconds
129	Mehmet Kulübecioğlu	memoklbc@icloud.com		100	0 Hour 2 Minute 48 Seconds
130	yağmur	6ocakyagmur@gmail.com		50	0 Hour 2 Minute 0 Seconds
- Actions: Tümünü işaretle, Seçilileri: Düzenle, Kopyala, Sil, Dışa aktar

Figure 9.5 – Scores Table in phpMyAdmin

G.	Sheets	No document	Sig. n.	Date	IAЛЦ.466538.005 EN	Sheet
						69

	Fazladan seçenekler	<input type="checkbox"/>	Düzenle	<input type="checkbox"/>	Kopyala	<input checked="" type="radio"/>	Sıralama	id	user_name	user_email	question_id	user_answer	is_correct	date	exam_session_id
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1092	Mehmet Kulubecioğlu	memokbc@icloud.com	62	B	1	2025-03-30 19:00:18	session_67e96a8e433aa1.70152760
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1093	Mehmet Kulubecioğlu	memokbc@icloud.com	64	A	1	2025-03-30 19:00:20	session_67e96a8e433aa1.70152760
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1094	Mehmet Kulubecioğlu	memokbc@icloud.com	65	C	0	2025-03-30 19:00:21	session_67e96a8e433aa1.70152760
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1095	Mehmet Kulubecioğlu	memokbc@icloud.com	61	D	0	2025-03-30 19:00:22	session_67e96a8e433aa1.70152760
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1096	Mehmet Kulubecioğlu	memokbc@icloud.com	19	D	0	2025-03-30 19:00:23	session_67e96a8e433aa1.70152760
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1097	yağmur	6ocakyagmur@gmail.com	58	B	1	2025-03-30 21:04:06	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1098	yağmur	6ocakyagmur@gmail.com	64	A	1	2025-03-30 21:04:07	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1099	yağmur	6ocakyagmur@gmail.com	65	B	1	2025-03-30 21:04:08	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1100	yağmur	6ocakyagmur@gmail.com	19	B	0	2025-03-30 21:04:09	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1101	yağmur	6ocakyagmur@gmail.com	33	A	1	2025-03-30 21:04:10	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1102	yağmur	6ocakyagmur@gmail.com	61	A	0	2025-03-30 21:04:11	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1103	yağmur	6ocakyagmur@gmail.com	60	C	1	2025-03-30 21:04:12	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1104	yağmur	6ocakyagmur@gmail.com	63	D	0	2025-03-30 21:04:13	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1105	yağmur	6ocakyagmur@gmail.com	59	C	1	2025-03-30 21:04:14	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1106	yağmur	6ocakyagmur@gmail.com	62	D	0	2025-03-30 21:04:15	session_67e987965d8408.20011526
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1107	eda	eda@gmail.com	65	B	1	2025-03-30 21:08:42	session_67e988aa192ce5.50850132
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1108	eda	eda@gmail.com	33	A	1	2025-03-30 21:08:43	session_67e988aa192ce5.50850132
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1109	eda	eda@gmail.com	58	B	1	2025-03-30 21:08:44	session_67e988aa192ce5.50850132
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1110	eda	eda@gmail.com	64	A	1	2025-03-30 21:08:45	session_67e988aa192ce5.50850132
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1111	eda	eda@gmail.com	19	C	1	2025-03-30 21:08:46	session_67e988aa192ce5.50850132
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1112	eda	eda@gmail.com	61	A	0	2025-03-30 21:08:47	session_67e988aa192ce5.50850132
<input type="checkbox"/>		Düzenle	<input type="checkbox"/>		Kopyala	<input checked="" type="radio"/>	1	1113	eda	eda@gmail.com	60	C	1	2025-03-30 21:08:48	session_67e988aa192ce5.50850132

Figure 9.6 – Exam Results Table in phpMyAdmin

9.6 Sample Result Entry

Table 9.4 – Sample Result Entry from Scores Table

user_name	user_email	score	exam_duration	exam_session_id
Mehmet Kulübecioğlu	memoklbc@icloud.com	70	0 Hour 0 Minute 33 Seconds	session_67e9 6a8e433aa1.7 0152760

Each score was supported by 10 detailed entries in exam_results, linked by the same session ID.

					Sheet	
G. She ets	No document	Sig n.	Date		IAЛЦ.466538.005 EN	70

9.7 Summary

The system passed all core functionality tests. Teachers were able to securely manage exams, and students completed exams successfully with accurate score tracking. The session ID mechanism allowed proper grouping of exam data, and all database operations performed reliably without data loss or inconsistency.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet
						71

10. SECURITY CONSIDERATIONS

10.1 Introduction

Security is a fundamental component in the development of any web-based application, especially in systems like QuizMaster that handle sensitive data such as exam content, student performance, and login credentials. Although QuizMaster is designed as a lightweight educational platform, several security mechanisms were integrated to ensure safe operation, protect user privacy, and maintain data integrity.

10.2 Authentication and Session Management

- Only **registered teachers** are allowed to log into the system. Login credentials are checked against the users table using a secure form on login.php.
- Upon successful login, a PHP session variable (e.g., \$_SESSION ['user_id']) is set to identify the authenticated user.
- All sensitive admin pages (e.g., sinav-ekle.php, sinav-duzenle.php, sinav-sonuclari.php) include **session control checks**. If the user is not logged in, they are redirected to the login screen.
- Logging out is handled via cikis.php, which securely **destroys the session** using session_destroy() and redirects the user.

This session management ensures that unauthorized users cannot access protected features even if they know the URL.

Sheet
G. Sheets No document Sig. Date IAЛЦ.466538.005 EN 72

10.3 Exam Session Tracking (Students)

- Students do not have user accounts or login credentials.
- When a student begins an exam, a unique, randomly generated exam_session_id is created in exam.php and stored in the session.
- This ID is used to track:
 - All answers recorded in exam_results
 - Final score saved in scores

This session ID acts as a secure identifier, enabling the grouping of exam data while maintaining student anonymity.

10.4 Input Validation and Sanitization

- All input fields in forms are **validated on the client side** using JavaScript (e.g., required fields, email format checks).
- On the server side (PHP), inputs are sanitized using:
 - trim() to remove whitespace
 - htmlspecialchars() to prevent HTML injection
 - mysqli_real_escape_string() for safe SQL execution
- Students cannot proceed to the next question in an exam without selecting an answer, avoiding incomplete submissions.

This two-layer validation system protects the database from injection attacks and ensures data quality.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet
						73

10.5 SQL Injection Prevention

- SQL queries are carefully written with **input sanitization** and query structure controls.
- While prepared statements are not yet used in all parts of the system, current queries are protected through controlled input fields and escaping.
- Future versions of QuizMaster will migrate to **PDO** or **MySQLi prepared statements** for stronger protection.

10.6 Password Security

- Teacher credentials are stored in the users table.
- Currently, passwords are stored as plain text in the database — which is functional but insecure.
- Future improvements include:
 - Using `password_hash()` and `password_verify()` functions in PHP
 - Enforcing password strength (e.g., minimum length, symbols, etc.)
 - Adding password reset functionality

Storing plain text passwords is discouraged; hashing is highly recommended for production systems.

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet
						74

10.7 File Upload Security

- Teachers can upload **audio files** for listening-type questions via sinav-ekle.php.
- The system restricts file uploads to valid types like .mp3, .wav, and .ogg.
- Uploaded files are saved into the /audio/ directory and renamed to prevent filename collisions or path traversal attacks.
- No executable or script files are permitted.

This protects the server from malicious uploads while maintaining flexibility for media-based questions.

10.8 Privacy Considerations

- **No student accounts** are created, which enhances privacy and reduces data exposure risks.
- Student input is limited to **name** and **email**, collected only at the time of the exam.
- All exam results are linked to the **exam_session_id** — not a user ID — meaning historical tracking is session-based, not user-based.

This approach simplifies data storage and helps meet basic data minimization principles of privacy-focused design.

10.9 Security Limitations and Future Enhancements

Despite the basic protections in place, QuizMaster can be further improved with the following:

G. H.	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 75

Table 10.1 – Security Limitations and Future Enhancements

Area	Current Status	Future Improvement
Password storage	Plain text	Use password_hash() and password_verify()
CSRF protection	Not implemented	Add CSRF tokens in form submissions
Login brute force	No rate limiting	Implement login attempt limits
Audit logging	Not available	Track question edits and admin actions
HTTPS support	Local only	Deploy over SSL for production use

10.10 Summary

QuizMaster incorporates several foundational security techniques to support secure exam delivery, session tracking, and data management. While the current implementation provides basic protection for an educational project, adopting stronger encryption, validation, and authentication strategies would be essential for deploying the system in a real-world, multi-user environment.

With proper session management, input validation, file controls, and a privacy-first design, QuizMaster ensures a secure experience for both students and teachers.

G. ^{RI}	Sne ets	No document	Sig n.	Da te	IAIЛЦ.466538.005 EN	Sheet 76
------------------	------------	----------------	-----------	----------	---------------------	-------------

11. CHALLENGES ENCOUNTERED

11.1 Technical Challenges

11.1.1. Database Session Tracking

- **Problem:** Linking student answers with their score without a login system was initially confusing.
- **Solution:** Introduced a unique exam_session_id generated during the exam start. This ID became the glue between exam_results and scores.
- **Outcome:** Allowed seamless grouping of exam data per student per attempt without requiring accounts.

11.1.2. Audio File Uploading

- **Problem:** Integrating file upload into the question creation flow was technically tricky, especially handling file validation and ensuring correct playback in exams.
- **Solution:** Used standard PHP file handling methods to restrict file types and store them in a designated audio/ folder.
- **Outcome:** Teachers can now add listening questions easily with audio support.

Sheet
G. Sheets No document Sig. n. Date ІАЛЦ.466538.005 EN 77

11.1.3. Countdown Timer Synchronization

- **Problem:** Accurately tracking the time spent by a student without server-side enforcement.
- **Solution:** Implemented JavaScript timers to control time display and calculated duration client-side.
- **Outcome:** Worked well in most cases, although client-side timers are inherently less secure than server-based ones.

11.2 Design and Usability Challenges

11.2.1. User Interface Consistency

- **Problem:** The initial UI included mixed styles and unused CSS due to using a prebuilt 30,000-line stylesheet.
- **Solution:** Selected only the necessary CSS components and reorganized form layouts for clarity.
- **Outcome:** A clean, consistent admin and student interface optimized for usability.

11.2.2. Mobile Responsiveness

- **Problem:** Some pages (e.g., exam screen) didn't render correctly on mobile devices due to hardcoded widths.
- **Solution:** Applied CSS media queries and fluid layouts.
- **Outcome:** Improved the mobile experience for students taking exams on tablets.

G. H	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 78

11.3 Project Management Challenges

11.3.1. Time Management

- **Problem:** Balancing the demands of coursework, personal life, and this graduation project was difficult.
- **Solution:** Created a structured weekly plan and focused on completing one module at a time.
- **Outcome:** Helped stay organized and meet deadlines without sacrificing quality.

11.3.2. File & Code Organization

- **Problem:** As the number of PHP files grew, managing dependencies and logic flow became difficult.
- **Solution:** Organized files into logical sections (admin, exam, assets) and created helper files like baglan.php.
- **Outcome:** Easier maintenance and better readability in later stages of development.

11.4 Summary

The development of QuizMaster was both technically and logically challenging. From database design to user interaction, each step presented unique obstacles. However, with creative problem-solving and structured planning, these challenges were overcome effectively. This experience significantly enhanced the developer's understanding of full-stack development, security, and user-centered design.

Sheet
ІАЛЦ.466538.005 EN 79

12. CONCLUSION

The development of the **QuizMaster** system successfully met its original objective: to create a functional, secure, and user-friendly web service for creating and conducting online exams.

Throughout the development process, a variety of tools and technologies were integrated, including **PHP**, **MySQL**, **HTML/CSS**, and **JavaScript**. These were supported by a well-organized file structure, modular backend scripts, and a normalized database schema. The system was developed using the **XAMPP** local environment and tested extensively under real-use scenarios.

Key Achievements:

- Teachers can log in, manage questions, and view exam results through a secure and intuitive admin panel.
- Students can take exams easily without logging in, simply by entering their name and email.
- Questions support both **reading** and **listening** types, with optional audio file integration.
- Answers and scores are stored in a relational database, grouped by a unique session ID for each attempt.
- The UI is mobile-responsive, with clean separation between the teacher and student interfaces.

The system's architecture emphasized **simplicity and functionality**, while maintaining essential **security practices** such as input validation, session control, and restricted file uploads.

Sheet	
IAЛЦ.466538.005 EN	
G. Sheets No document Sig n. Date	80

While the system currently functions well in its educational context, several areas have been identified for future improvement:

- Use of **hashed passwords** for teachers
- Implementation of **CSRF protection**
- Addition of **audit logging** for sensitive operations
- Deployment over **HTTPS** for production

Final Reflection:

QuizMaster proved to be a valuable learning experience, especially in handling real-world concerns like database integrity, UI consistency, and user privacy. It demonstrated how a web application can be built from scratch using open-source tools while following structured design principles.

This project not only fulfills the academic requirements of a graduation thesis but also lays the foundation for a scalable and secure web-based exam platform in future deployments.

G. Ch	She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN	Sheet 81
----------	------------	----------------	-----------	----------	--------------------	-------------

13. REFERENCES

A good academic project should always acknowledge the tools, libraries, and resources used during development. Below is a list of key references that supported the creation of the **QuizMaster** system.

13.1 Online Resources

1. PHP Documentation – <https://www.php.net/docs.php>
2. MySQL Documentation – <https://dev.mysql.com/doc/>
3. W3Schools (HTML/CSS/JS Reference) – <https://www.w3schools.com>
4. Stack Overflow – <https://stackoverflow.com>
5. Mozilla Developer Network (MDN) – <https://developer.mozilla.org>

13.2 Tools Used

- **XAMPP** – Local development environment
- **Visual Studio Code** – Code editor
- **phpMyAdmin** – Database management tool
- **draw.io** – Diagram creation (UML, ER diagrams)
- **Canva / Word** – Documentation visuals and layout

13.3 Project Inspiration & Coursework

- Lecture notes and guidance from [Instructor's Name / Course Title]
- Previous small-scale exam systems and UI samples studied as part of academic research
- Discussions and feedback from peers during project development

G.	Sheets	No document	Sig.	Date	IAЛЦ.466538.005 EN	Sheet
						82

14. APPENDICES

14.1 Code Snippets

Below are selected code snippets taken directly from the source files of the QuizMaster system. These represent essential logic blocks such as session management, answer evaluation, database interaction, and security handling.

14.1.1. Starting an Exam Session

File: exam.php

Purpose: This code initializes a student's exam session by collecting their name and email and generating a unique session ID.

```
7 // Kullanıcı daha önce isim ve e-posta girip girdiyse oturumu başlatıyoruz (form gönderildiye)
8 if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['name']) && isset($_POST['email'])) {
9     $_SESSION['user_name'] = $_POST['name']; // Kullanıcı adını oturuma kaydet
10    $_SESSION['user_email'] = $_POST['email']; // E-posta oturuma kaydet
11
12    // start_time sadece ilk kez atanır (birden fazla tekrar başlatılmasın diye kontrol)
13    if (!isset($_SESSION['start_time'])) {
14        $_SESSION['start_time'] = date('Y-m-d H:i:s'); // Sınavın başladığı zaman
15    }
16
17    // exam_session_id de sadece sınav başlarken atanmalı
18    if (!isset($_SESSION['exam_session_id'])) {
19        $_SESSION['exam_session_id'] = uniqid('session_', true); // benzersiz ID üret
20    }
21 }
```

Figure 14.1 – Session Initialization in exam.php

Sheet				
G. She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN

14.1.2. Calculating and Storing Exam Score

File: submit_exam.php

Purpose: This block calculates how many answers were correct and prepares the data for storage in the database.

```
45  foreach ($answers as $question_id => $user_answer) {
46      $answerTime = clone $baseTime; // Zamanı klonla
47      $answerTime->modify("+{$i} seconds"); // Her cevap arası 1 saniye ekle (simülasyon)
48      $formattedDate = $answerTime->format('Y-m-d H:i:s'); // Zamanı biçimlendir
49      $i++;
50
51      $stmt = $db->prepare("SELECT correct_option FROM questions WHERE id = ?");
52      $stmt->execute([$question_id]);
53      $correct_answer = $stmt->fetchColumn(); // Doğru cevabı al
54
55      if ($correct_answer === $user_answer) {
56          $score += 10; // Doğruysa 10 puan ekle puan burada hesaplanıyor
57      }
58
59      // exam_results tablosuna cevabı kaydet
60      $stmt = $db->prepare("INSERT INTO exam_results (
61          user_name, user_email, question_id, user_answer, is_correct, date, exam_session_id
62      ) VALUES (
63          :user_name, :user_email, :question_id, :user_answer, :is_correct, :date, :exam_session_id
64      )");
65      $stmt->execute([
66          ':user_name' => $user_name,
67          ':user_email' => $user_email,
68          ':question_id' => $question_id,
69          ':user_answer' => $user_answer,
70          ':is_correct' => ($correct_answer === $user_answer) ? 1 : 0,
71          ':date' => $formattedDate,
72          ':exam_session_id' => $_SESSION['exam_session_id']
73      ]);
74  }
75
76 }
```

Figure 14.2 – Answer Evaluation and Score Storage Logic in submit_exam.php

14.1.3. Question Deletion Logic

File: inc/islem.php

Purpose: This function handles the deletion of questions from the admin panel and redirects with status.

G.	Sheets	No document	Sig n.	Date	IAЛЦ.466538.005 EN	Sheet
						84

```

6 if (isset($_GET['sinav-sil'])) {
7     $id = intval($_GET['sinav-sil']);
8
9     // Veritabanında id'nin var olup olmadığını kontrol et
10    $kontrol = $db->prepare("SELECT * FROM questions WHERE id = ?");
11    $kontrol->execute([$id]);
12
13    if ($kontrol->rowCount() > 0) {
14        // ID mevcut, silme işlemi yapılabilir
15        $sil = $db->prepare("DELETE FROM questions WHERE id = ?");
16        $sil->execute([$id]);
17
18        // Silme işleminin başarılı olup olmadığını kontrol et
19        if ($sil->rowCount() > 0) {
20            header("Location: ../../yonetim/seviye-tespit-sinavlari.php?durum=basarili");
21            exit;
22        } else {
23            header("Location: ../../yonetim/seviye-tespit-sinavlari.php?durum=hatali");
24            exit;
25        }
26    } else {
27        // ID bulunamadı
28        header("Location: ../../yonetim/seviye-tespit-sinavlari.php?durum=bulunamadi");
29        exit;
30    }
}

```

Figure 14.3 – Question Deletion Operation in islem.php

14.1.4. Server-Side Duration Calculation

File: submit_exam.php

Purpose: Calculates the duration of the exam session from start to end using server time.

```

try {
    $start = new DateTime($start_time); // Başlangıç zamanını DateTime nesnesine çevir
    $end = new DateTime($end_time); // Bitiş zamanını DateTime nesnesine çevir

    // ++ Sınav süresi kontrolü (sunucu tarafından)
    $maxDurationSeconds = 3600; // 1 saat sınırı (3600 saniye)
    $elapsedSeconds = $end->getTimestamp() - $start->getTimestamp(); // Geçen süreyi hesapla

    if ($elapsedSeconds > $maxDurationSeconds) {
        die("⚠ Exam time exceeded! Answers not saved."); // Süre aşılırsa işlem sonlandırılır
    }
}

```

Figure 14.4 – Exam Duration Calculation in submit_exam.php

G. H	She ets	No document	Sig n.	Da te	IAЛIЦ.466538.005 EN	Sheet 85
---------	------------	----------------	-----------	----------	---------------------	-------------

14.2 Screenshots

Below are the actual user interface screenshots captured during the usage and testing of the QuizMaster system. These images demonstrate key functionalities across both the teacher (admin) and student interfaces.

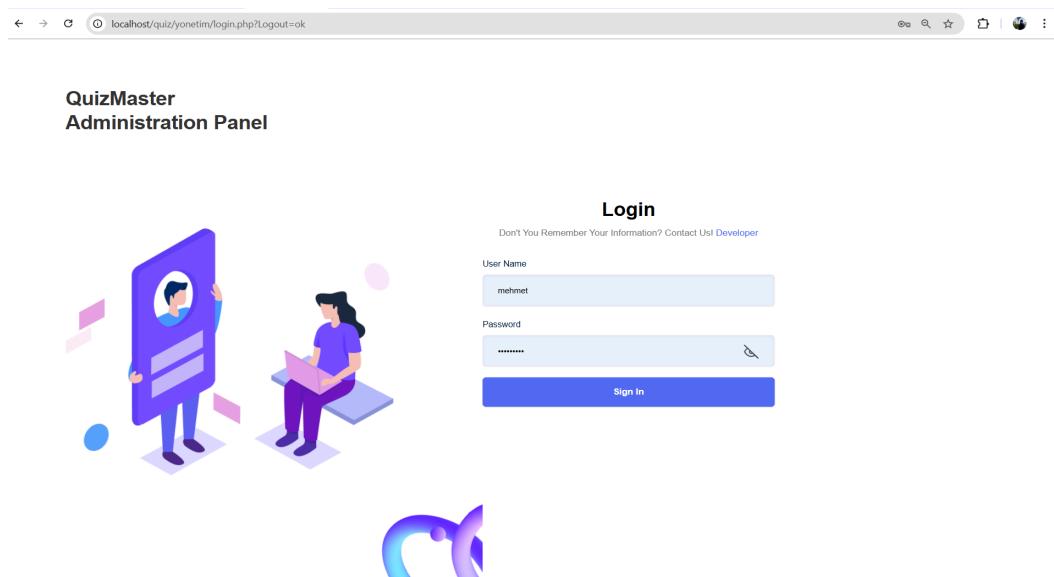


Figure 14.5 – Login Screen

CH	Sheets	No document	Sig n.	Date	IAЛЦ.466538.005 EN	Sheet
						86

The login interface for teachers, validating credentials and preventing unauthorized access.

Add Question

Question Type:

Reading

Question Text:

What does HTML stand for?

Options:

A: Hyper Trainer Marking Language

B: Hyper Text Markup Language

C: High Text Machine Language

D: Hyper Tool Multi Language

Correct Option:

B

Audio File for Listening Question:

Dosya Seç Dosya seçilmedi

Add Question Back

Figure 14.6 – Add Question Interface

Teachers can input question type, text, answer options, and correct option. Audio upload is available for listening-type questions.

Sheet				
G. She ets	No document	Sig n.	Da te	IAЛЦ.466538.005 EN 87

localhost/quiz/yonetim/sinav-duzenle.php?id=58

Update Question

Question Type:

Reading

Question Text:

What does HTML stand for?

Options:

A: Hyper Trainer Marking Language

B: Hyper Text Markup Language

C: High Text Machine Language

D: Hyper Tool Multi Language

Correct Option:

B

Audio File for Listening Question:

Dosya Seç Dosya seçilmemi

Update Question **Back**

Figure 14.7 – Update Question Interface

Allows editing an existing question, including changing its type (Reading/Listening) and options.

Sheet
G. Sheets No document Sig n. Date IAЛЦ.466538.005 EN 88

Name:
Mehmet Kulibecioğlu

E-mail:
memoklbc@icloud.com

Start Exam

Figure 14.8 – Student Identity Form

Students must provide their name and email before starting the exam session.



Figure 14.9 – Listening Question Display

Sheet
IAЛЦ.466538.005 EN
G. Sheets No document Sig n. Date

An example of a listening-type question with an embedded audio player.

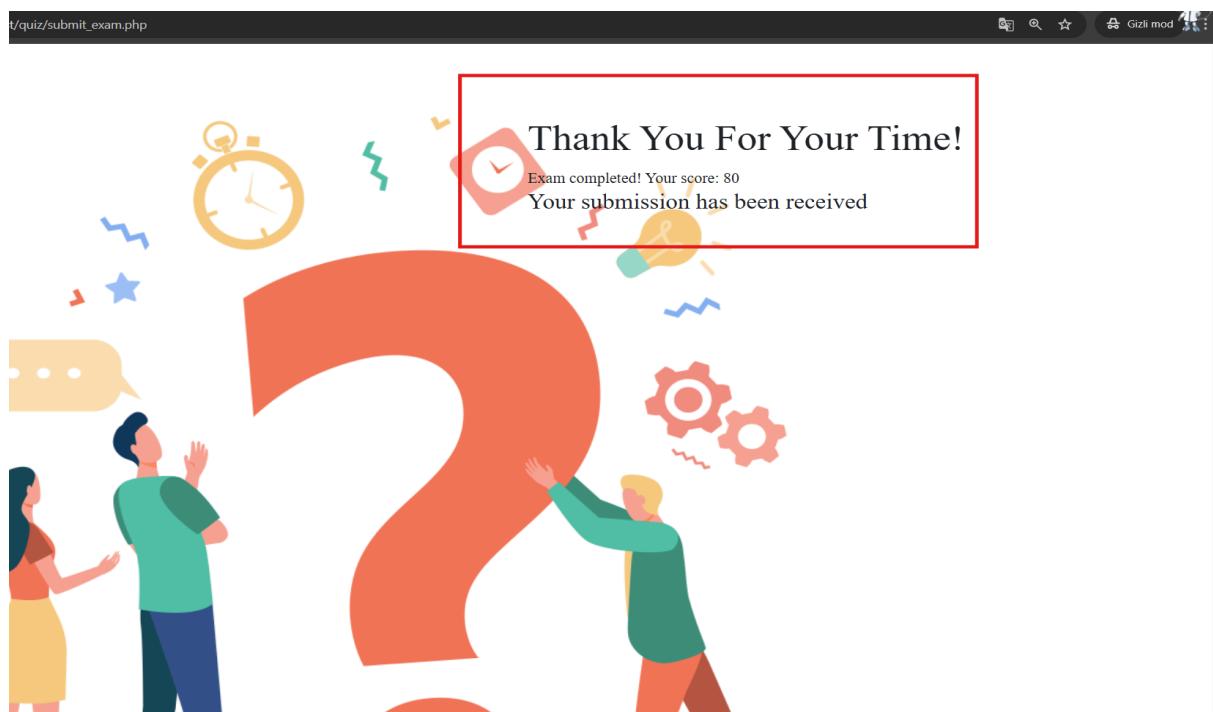


Figure 14.10 – Exam Completion Message

A thank-you screen confirming exam submission and showing the score.

Gösterilen satır 0 - 24 (toplam 100, Sorgu 0,0005 saniye sürdü.)								
SELECT * FROM `exam_results`								
<input type="checkbox"/> Profil çıkart [Satır içi düzenle] [Düzenle] [SQL'i açıka] [PHP kodu oluştur] [Yenile]								
1	>	>>	<input type="checkbox"/> Tümünü göster	Satır sayısı:	25	Satırları süz:	Bu tabloda ara	Anahtara göre sırala: Yok
Fazladan seçenekler								
←	→	↑	↓	id	user_name	user_email	question_id	user_answer
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Düzenle	Kopyala	<input type="checkbox"/> Sil	1067	Mehmet Kulübecioğlu memoklbc@icloud.com
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Düzenle	Kopyala	<input type="checkbox"/> Sil	1068	Mehmet Kulübecioğlu memoklbc@icloud.com
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Düzenle	Kopyala	<input type="checkbox"/> Sil	1069	Mehmet Kulübecioğlu memoklbc@icloud.com
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Düzenle	Kopyala	<input type="checkbox"/> Sil	1070	Mehmet Kulübecioğlu memoklbc@icloud.com

Figure 14.11 – Database View of exam_results Table

					IAЛЦ.466538.005 EN	Sheet
G. ^H	She ets	No document	Sig n.	Da te		90

Illustrates how student answers and session IDs are stored in the database.

Question	Date	Status	Actions
What does HTML stand for?	2025-03-30 17:04:06		Edit Delete
Which language is used for styling web pages?	2025-03-30 17:04:41		Edit Delete
what is he name ?	2025-03-27 18:52:30		Edit Delete
Which language is used for styling	2025-04-02 10:21:31		Edit Delete

Figure 14.12 – Question Management Dashboard with Edit/Delete Options

Teachers can view, edit, and delete previously added questions.

Figure 14.13 – Question Display with Navigation and Timer

					ІАЛЦ.466538.005 EN	Sheet
G. Sheets	No document	Sig n.	Date			91

Students view each question with a countdown timer and navigation buttons.

The screenshot shows a web-based quiz interface. At the top, the URL is localhost/quiz/yonetim/sinav-detay.php?id=6ocakyagmur@gmail.com. Below the header, there is a navigation bar with a magnifying glass icon and a star icon. The main content area displays a series of questions:

- Question 10: What does HTML stand for? (Icon: book) - Options: C (Correct), B (Wrong). Status: Wrong.
- Section header: Exam Date: 2025-04-02 11:03
- Section header: Score: 50 Duration: 0 Hour 2 Minute 0 Seconds
- Question 1: Which company developed the Java programming language? (Icon: book) - Options: B (Correct), B (Correct). Status: Correct.
- Question 2: what is he name ? (Icon: headphones) - Options: A (Correct), A (Correct). Status: Correct.
- Question 3: Which of the following is a programming language? (Icon: book) - Options: C (Correct), C (Correct). Status: Correct.
- Question 4: Which language is used for styling web pages? (Icon: book) - Options: C (Correct), C (Correct). Status: Correct.
- Question 5: What does HTML stand for? (Icon: book) - Options: A (Correct), B (Wrong). Status: Wrong.

Figure 14.14 – Scores Table View

Displays all completed sessions with user, score, and duration information in the scores table.

Sheet
IAЛЦ.466538.005 EN G. She ets No document Sig n. Da te 92

Figure 14.15 – Exam Results Table View

Detailed answer records in the exam_results table, including correctness and timestamps.

```

7 // Kullanıcı daha önce isim ve e-posta girip girdiyse oturumu başlatıyoruz (form gönderildiyse)
8 if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['name']) && isset($_POST['email'])) {
9     $_SESSION['user_name'] = $_POST['name']; // Kullanıcı adını oturuma kaydet
10    $_SESSION['user_email'] = $_POST['email']; // E-posta oturuma kaydet
11
12    // start_time sadece ilk kez atanır (birden fazla tekrar başlatılmasın diye kontrol)
13    if (!isset($_SESSION['start_time'])) {
14        $_SESSION['start_time'] = date('Y-m-d H:i:s'); // Sınavın başladığı zaman
15    }
16
17    // exam_session_id de sadece sınav başlarken atanmalı
18    if (!isset($_SESSION['exam_session_id'])) {
19        $_SESSION['exam_session_id'] = uniqid('session_', true); // benzersiz ID üret
20    }

```

Figure 14.16 – Session Initialization Code (exam.php)

PHP block that starts a session and generates a unique session ID.

					IAЛIЦ.466538.005 EN	Sheet
G. Ch	She ets	No document	Sig n.	Da te		93

```

foreach ($answers as $question_id => $user_answer) {
    $answerTime = clone $baseTime; // Zamanı klonla
    $answertime->modify("+1 seconds"); // Her cevap arası 1 saniye ekle (simülasyon)
    $formattedDate = $answerTime->format('Y-m-d H:i:s'); // Zamanı biçimlendir
    $i++;

    $stmt = $db->prepare("SELECT correct_option FROM questions WHERE id = ?");
    $stmt->execute([$question_id]);
    $correct_answer = $stmt->fetchColumn(); // Doğru cevabı al

    if ($correct_answer === $user_answer) {
        $score += 10; // Doğruysa 10 puan ekle puan burada hesaplanıyor
    }

    // exam_results tablosuna cevabı kaydet
    $stmt = $db->prepare("INSERT INTO exam_results (
        user_name, user_email, question_id, user_answer, is_correct, date, exam_session_id
    ) VALUES (
        :user_name, :user_email, :question_id, :user_answer, :is_correct, :date, :exam_session_id
)");
    $stmt->execute([
        ':user_name' => $user_name,
        ':user_email' => $user_email,
        ':question_id' => $question_id,
        ':user_answer' => $user_answer,
        ':is_correct' => ($correct_answer === $user_answer) ? 1 : 0,
        ':date' => $formattedDate,
        ':exam_session_id' => $_SESSION['exam_session_id']
    ]);
}

```

Figure 14.17 – Score Calculation and Storage Logic (submit_exam.php)

Code responsible for validating answers and storing them in the database.

```

6  if (isset($_GET['sinav-sil'])) {
7      $id = intval($_GET['sinav-sil']);
8
9      // Veritabanında id'nin var olup olmadığını kontrol et
10     $kontrol = $db->prepare("SELECT * FROM questions WHERE id = ?");
11     $kontrol->execute([$id]);
12
13     if ($kontrol->rowCount() > 0) {
14         // ID mevcut, silme işlemi yapılabılır
15         $sil = $db->prepare("DELETE FROM questions WHERE id = ?");
16         $sil->execute([$id]);
17
18         // Silme işleminin başarılı olup olmadığını kontrol et
19         if ($sil->rowCount() > 0) {
20             header("Location: ../../yonetim/seviye-teşpit-sinavlari.php?durum=basarili");
21             exit;
22         } else {
23             header("Location: ../../yonetim/seviye-teşpit-sinavlari.php?durum=hatali");
24             exit;
25         }
26     } else {
27         // ID bulunamadı
28         header("Location: ../../yonetim/seviye-teşpit-sinavlari.php?durum=bulunamadi");
29         exit;
30     }
31 } else {
32     // 'sinav-sil' parametresi URL'de bulunamadı
33     header("Location: ../../yonetim/seviye-teşpit-sinavlari.php?durum=bulunamadi");
34     exit;
}

```

Figure 14.18 – Question Deletion Logic (islem.php)

G.	Sne ets	No document	Sig n.	Da te	IAJLIL.466538.005 EN	Sheet
						94

Code that allows teachers to delete questions from the admin interface.

```
29 // ++ Sınav süresi kontrolü (sunucu tarafında)
30 $maxDurationSeconds = 3600; // 1 saat sınırı (3600 saniye)
31 $elapsedSeconds = $end->getTimestamp() - $start->getTimestamp(); // Geçen süreyi hesapla
32
33 if ($elapsedSeconds > $maxDurationSeconds) {
34     die("⚠ Exam time exceeded! Answers not saved."); // Süre aşıldıkça işlem sonlandırılır
35 }
36 }
```

Figure 14.19 – Exam Duration Check Logic

Enforces maximum time limit for exams on the server side.

14.3 ER & UML Diagrams and Figures

This section presents the structural and behavioral diagrams used during the development of the **QuizMaster** system. These visuals are vital for understanding the system's entities, relationships, database structure, and file organization.

14.3.1 SQL Table Schemas

Description:

This subsection includes the SQL statements used to create core database tables. These scripts define the fields, data types, and constraints of each table.

Sheet
95
G. Sheets No document Sig n. Date IAJLЦ.466538.005 EN

SELECT * FROM `users`									
<input type="checkbox"/> Profil çıkart [Satır içi düzenle] [Düzenle] [SQL'i açıkla] [PHP kodu oluştur] [Yenile]									
<input type="checkbox"/> Tümünü göster Satır sayısı: 25		Satırları süz: Bu tabloda ara			Anahtara göre sırala: Yok				
Fazladan seçenekler									
← T →	▼ user_id	user_name	user_password	user_authority	created_at				
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1 mbkakadem	a7c24ca43a6f34097888d3579a04b535	admin	2025-03-08 20:33:45		
<input type="checkbox"/>	Düzenle	Kopyala	Sil	2 creator	c5d0e53ec04641a7d83e95597f8b71fa	user	2025-03-08 20:35:04		
<input type="checkbox"/>	Düzenle	Kopyala	Sil	3 mehmet	5232523ab4dce3a8cb6191ead18068f3	user	2025-03-24 10:32:49		

Figure 14.20 – SQL Table Creation: Users Table

SELECT * FROM `questions`														
<input type="checkbox"/> Profil çıkart [Satır içi düzenle] [Düzenle] [SQL'i açıkla] [PHP kodu oluştur] [Yenile]														
<input type="checkbox"/> Tümünü göster Satır sayısı: 25		Satırları süz: Bu tabloda ara			Anahtara göre sırala: Yok									
Fazladan seçenekler														
← T →	▼ id	type	question_text	option_a	option_b	option_c	option_d	correct_option	audio_file	created_at	durum	exam_duration_seconds		
<input type="checkbox"/>	Düzenle	Kopyala	Sil	58 reading	What does HTML stand for?	Hyper Trainer Marking Language	Hyper Text Markup Language	High Text Machine Language	Hyper Tool Multi Language	NULL	2025-03-30 17:04:06	1	3600	
<input type="checkbox"/>	Düzenle	Kopyala	Sil	59 reading	Which language is used for styling web pages?	HTML	jQuery	CSS	XML	C	NULL	2025-03-30 17:04:41	1	3600
<input type="checkbox"/>	Düzenle	Kopyala	Sil	33 listening	what is he name ?	jim	jimy	jack	john	A	uploads/audio/high-jim-85869.mp3	2025-03-27 18:52:30	1	3600
<input type="checkbox"/>	Düzenle	Kopyala	Sil	67 reading	Which language is used for styling web pages?	HTML	XML	jQuery	CSS	D	NULL	2025-04-02 10:21:31	1	3600
<input type="checkbox"/>	Düzenle	Kopyala	Sil	19 listening	who?	jimy	jack	jim	john	C	uploads/audio/high-jim-85869.mp3	2025-03-26 17:13:58	1	3600
<input type="checkbox"/>	Düzenle	Kopyala	Sil	60 reading	Which of the following is a programming language?	HTML	CSS	Python	SQL	C	NULL	2025-03-30 17:05:16	1	3600
<input type="checkbox"/>	Düzenle	Kopyala	Sil	61 reading	Which symbol is used to start a single-line comment...	//	<!--	#	/*	C	NULL	2025-03-30 17:05:51	1	3600

Figure 14.21 – SQL Table Creation: Questions Table

					IAЛЦ.466538.005 EN	Sheet
G. H G.	She ets	No document	Sig n.	Da te		96

SELECT * FROM `scores`

Profil çikart [Satır içi düzenle] [Düzenle] [SQL'i açıklala] [PHP kodu oluştur] [Yenile]

Tümünü göster | Satır sayısı: 25 | Satırları süz: Bu tabloda ara | Anahtara göre sırala: Yok

Fazladan seçenekler

	← T →							
<input type="checkbox"/>	Düzenle	Kopyala	Sil	122	Mehmet Kulübecioğlu	memoklbc@icloud.com	30	0 Hour 0 Minute 21 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	123	Mehmet Kulübecioğlu	memoklbc@icloud.com	100	0 Hour 3 Minute 22 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	124	Mehmet Kulübecioğlu	memoklbc@icloud.com	70	0 Hour 0 Minute 33 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	125	yağmur	6ocakyagmur@gmail.com	60	0 Hour 1 Minute 30 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	126	eda	eda@gmail.com	80	0 Hour 2 Minute 20 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	127	Mehmet Kulübecioğlu	memoklbc@icloud.com	90	0 Hour 1 Minute 13 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	128	newstudent	newsrudent@gmail.com	50	0 Hour 1 Minute 5 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	129	Mehmet Kulübecioğlu	memoklbc@icloud.com	100	0 Hour 2 Minute 48 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	130	yağmur	6ocakyagmur@gmail.com	50	0 Hour 2 Minute 0 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	131	yağmur	6ocakyagmur@gmail.com	80	0 Hour 1 Minute 45 Seconds
<input type="checkbox"/>	Düzenle	Kopyala	Sil	132	Emre	emre@gmail.com	50	0 Hour 0 Minute 30 Seconds

Figure 14.22 – SQL Table Creation: Scores Table

SELECT * FROM `exam_results`

Profil çikart [Satır içi düzenle] [Düzenle] [SQL'i açıklala] [PHP kodu oluştur] [Yenile]

<< < 4 > >> | Tümünü göster | Satır sayısı: 25 | Satırları süz: Bu tabloda ara | Anahtara göre sırala: Yok

Fazladan seçenekler

	← T →							
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1142	Mehmet Kulübecioğlu	memoklbc@icloud.com	63	B
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1143	Mehmet Kulübecioğlu	memoklbc@icloud.com	61	C
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1144	Mehmet Kulübecioğlu	memoklbc@icloud.com	64	A
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1145	Mehmet Kulübecioğlu	memoklbc@icloud.com	59	C
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1146	Mehmet Kulübecioğlu	memoklbc@icloud.com	19	C
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1147	yağmur	6ocakyagmur@gmail.com	63	B
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1148	yağmur	6ocakyagmur@gmail.com	33	A
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1149	yağmur	6ocakyagmur@gmail.com	60	C
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1150	yağmur	6ocakyagmur@gmail.com	59	C
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1151	yağmur	6ocakyagmur@gmail.com	58	A
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1152	yağmur	6ocakyagmur@gmail.com	67	C
<input type="checkbox"/>	Düzenle	Kopyala	Sil	1153	yağmur	6ocakyagmur@gmail.com	19	B

Figure 14.23 – SQL Table Creation: exam_results Table

							Sheet
G.	Sheets	No document	Sig.	Date		IAЛП.466538.005 EN	
							97

14.3.2 Use Case Diagram

Description:

The Use Case Diagram outlines the primary interactions between users (teachers and students) and the system. It visualizes key functionalities such as login, exam management, and exam participation.

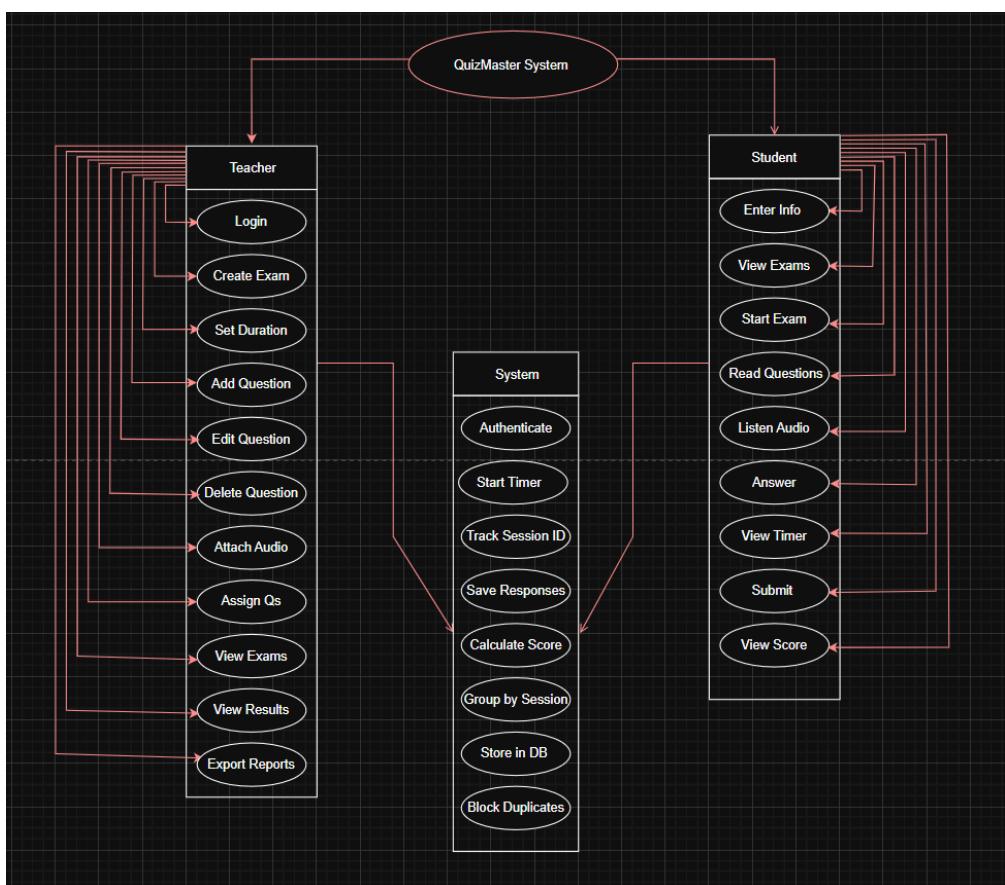


Figure 14.24 – Use Case Diagram of the QuizMaster System

C H G	Sheets	No document.	Sign.	Date	IAЛЦ.466538.006 D1		
Developed	Mehmet KULUBECI OGLU				Type	Sheets	Sheet
Checked	Kornienko Bogdan Yaroslavovich					1	1
Norm. control	Simonenko V. P.				Web Service for Creating Tests (QuizMaster)		
Approved	Stirenko S. G.				NTUU «KPI», FIOT IM-14		

14.3.3 Entity-Relationship Diagram (ERD)

Description:

The ER Diagram provides a high-level view of the database structure, showcasing tables such as users, questions, scores, and exam_results. It also illustrates the relationships and cardinalities between entities.

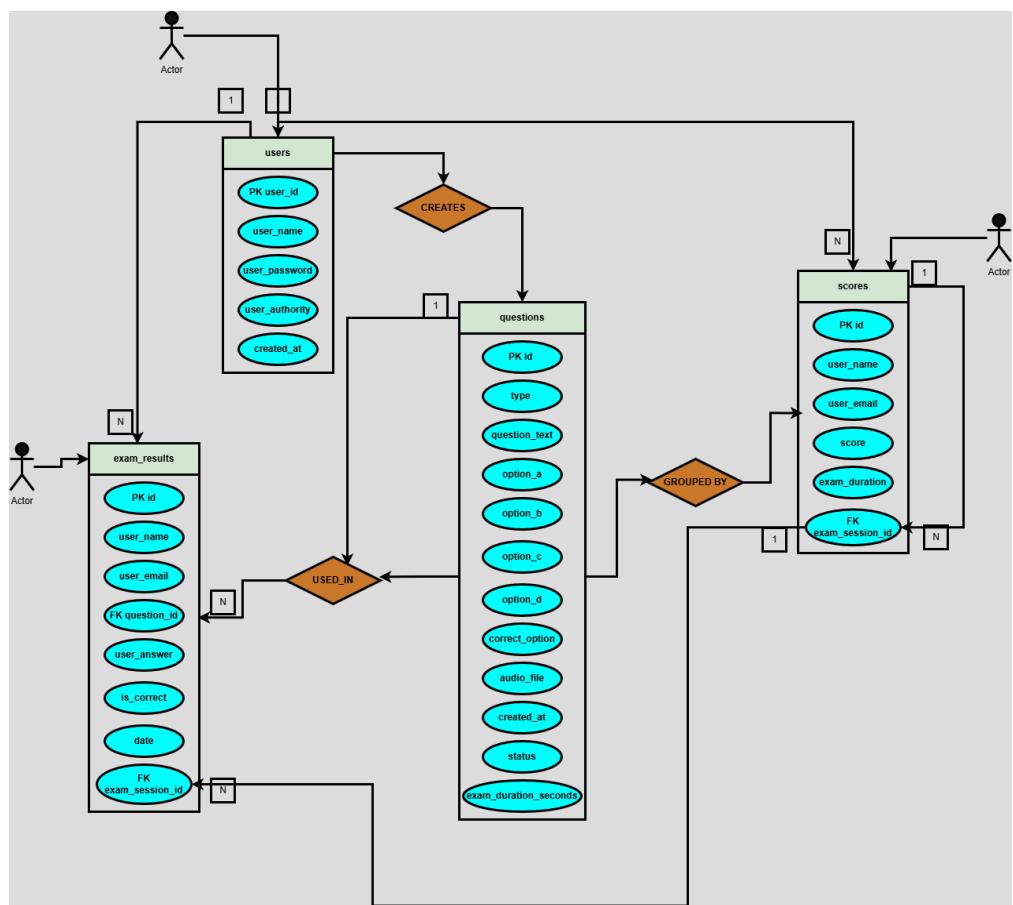


Figure 14.25 – Entity-Relationship Diagram (ERD) of QuizMaster

C H G	Sheets	No document.	Sign.	Date	IAЛЦ.466538.007 D2		
Developed	Mehmet KULUBECI OGLU				Web Service for Creating Tests (QuizMaster)		
Checked	Kornienko Bogdan Yaroslavovich				Type	Sheets	
Norm. control	Simonenko V. P.				1	1	
Approved	Stirenko S.				NTUU «KPI», FIOT IM-14		

14.3.4 File Directory Structure

Description:

The file structure diagram shows the folder and file hierarchy of the project. It includes directories such as assets, uploads, and yonetim, and highlights major PHP and CSS files.

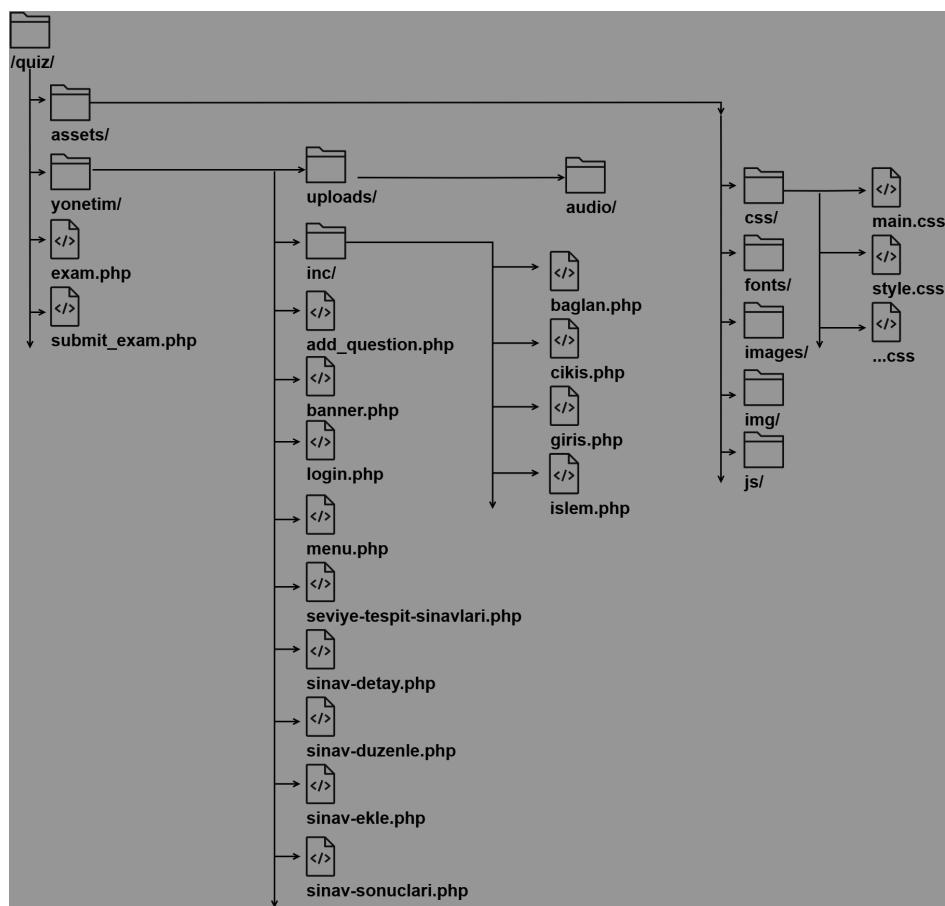


Figure 14.26 – File Structure of the QuizMaster System

C H G	Sheets	№ document.	Sign.	Date	IAЛЦ.466538.008 D3			
					Type	Sheets	Sheet	
Developed	Mehmet KULUBECI OGLU							
Checked	Kornienko Bogdan Yaroslavovich					1	1	
Norm. control	Simonenko V. P.							
Approved	Stirenko S. G.							
Web Service for Creating Tests (QuizMaster)					NTUU «KPI», FIOT IM-14			

14.3.5 System Architecture Diagram

Description: This diagram illustrates the architecture of the QuizMaster system, showing the client layer (built with HTML5, CSS3, and JavaScript), the server layer (powered by PHP), and the database layer (MySQL), along with their interactions.

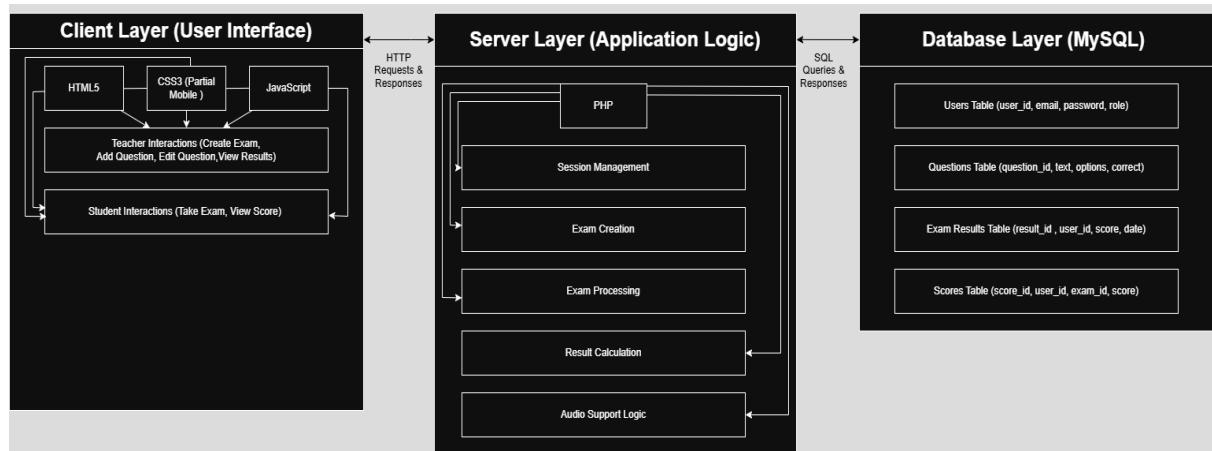


Figure 14.27 – System Architecture Diagram of QuizMaster

C H G	Sheets	№ document.	Sign.	Date	IAЛЦ.466538.009 D4			
					Type	Sheets	Sheet	
Developed	Mehmet KULUBECI OGLU							
Checked	Kornienko Bogdan Yaroslavovich					1	1	
Norm. control	Simonenko V. P.				Web Service for Creating Tests (QuizMaster)			
Approved	Stirenko S.				NTUU «KPI», FIOT IM-14			