

A

Mini Project Report

on

Quizventure – The Ultimate Technical Quiz

Submitted in partial fulfillment of the requirements for the

degree

Second Year Engineering – Computer Science Engineering (Data Science)

by

Yash Bhosale 23107017

Paras Mahajan 23107034

Harshal Gaikwad 23107052

Manas Chaudhari 23107074

Under the guidance of

Ms. Aavani Nair



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING (DATA SCIENCE)

A.P. SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W)-400615

UNIVERSITY OF MUMBAI

Academic year: 2024-25

CERTIFICATE

This to certify that the Mini Project report on **Quizventure – The Ultimate Technical Quiz** has been submitted by Yash Bhosale (23107017), Paras Mahajan (23107034), Harshal Gaikwad (23107052) and Manas Chaudhari (23107074) who are bonafide students of A. P. Shah Institute of Technology, Thane as a partial fulfillment of the requirement for the degree in **Computer Science Engineering (Data Science)**, during the academic year **2024-2025** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Ms. Aavani Nair
Guide

Ms. Anagha Aher
HOD, CSE (Data Science)

Dr. Uttam D. Kolekar
Principal

External Examiner:

1.

Place: A. P. Shah Institute of Technology, Thane

Date:

Internal Examiner:

1.

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Ms. Aavani Nair**. Expressing gratitude towards our HoD, **Ms. Anagha Aher**, and the Department of Computer Science Engineering (Data Science) for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our project coordinator **Ms. Aavani Nair** and **Ms. Richa Singh** who gave us his/her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

TABLE OF CONTENTS

1. Introduction.....	1
1.1.Purpose.....	1
1.2.Problem Statement.....	2
1.3.Objectives.....	2
1.4.Scope.....	4
2. Proposed System.....	5
2.1.Features and Functionality.....	5
3. Project Outcomes.....	6
4. Software Requirements.....	7
5. Project Design.....	9
6. Project Scheduling.....	14
7. Results.....	16
8. Conclusion.....	18
9. Future Work.....	22
References	

Chapter 1

Introduction

The rapid advancements in technology have revolutionized several sectors, including education and corporate training. One area where digital solutions have gained popularity is quiz management. In traditional systems, quizzes are often conducted manually, involving time-consuming processes such as question paper setting, printing, distribution, and evaluation. The advent of digital tools has made it possible to streamline this entire process. However, existing quiz systems often lack comprehensive features such as role-based access control, real-time scoring, and efficient data management.

Quizventure - An Ultimate Technical Quiz App is a desktop-based application designed to facilitate technical quizzes in an organized and user-friendly manner. Built using Java, the app provides a platform for both students and administrators to create, manage, and attempt quizzes. The app incorporates quiz creation, and user management, leveraging the MySQL database for data storage and management.

The app is developed using IDE for Java and MySQL for backend database connectivity, ensuring efficient storage and retrieval of quiz data. It is designed to incorporate both student and teacher (i.e. administrator) roles, to manage the entire quiz-taking and organizing process, respectively. Scope of the project is: Assessment and Evaluation, User Experience, Data Management, etc.

1.1. Purpose:

The purpose of Quizventure is to -provide a robust desktop-based application that allows users to create and participate in quizzes. The app is targeted at educational institutions, quiz organizers, and students who wish to streamline the quiz-taking process. The focus is on delivering a seamless quiz experience without retaining any personal data or quiz history.

1.2. Problem Statement:

The current methods of conducting quizzes, especially in educational and training environments, involve significant manual effort. From creating question papers to distributing them and evaluating responses, the entire process is prone to errors, delays, and inefficiencies. Additionally, existing quiz platforms often fail to address key user needs, such as providing detailed performance analysis, real-time feedback, and efficient data storage.

In many cases, quizzes are conducted without the aid of specialized software, leading to challenges like time-consuming result compilation, limited scalability, and lack of a centralized data management system. Quizventure aims to solve these problems by offering a streamlined, feature-rich quiz platform that provides comprehensive functionality for both quiz administrators and participants.

Quizventure aims to resolve these issues by offering a platform where users can easily create, manage, and attempt quizzes, all while ensuring seamless data storage in a MySQL database.

1.3. Objectives:

The primary objectives of Quizventure Pro are to streamline the creation, management, and participation in technical quizzes. This involves developing a user-friendly desktop application that offers robust tools for quiz organizers, including quiz creation, question customization, and quiz settings. The objectives are stated as:

To develop a desktop application that facilitates the creation, management, and participation in technical quizzes: The primary goal of this project is to design and build a user-friendly desktop application that serves as a platform for technical quizzes. This includes providing tools for quiz organizers to create quizzes, add questions, and customize quiz settings. Simultaneously, participants (students or users) should be able to easily access the quizzes, participate in them, and view results. The application will streamline the quiz process for both creators and participants, ensuring a smooth and efficient experience.

To store and manage user data and quiz records using multiple databases: The system will manage large amounts of data, including user information (such as profiles, login credentials, and roles) and quiz records (such as questions, answers, and results). To ensure this data is securely and efficiently stored, the application will use multiple databases. These databases will store and retrieve information, facilitating smooth operation while maintaining the integrity and privacy of the data. This approach allows for the segregation of data based on its type, enhancing security and data retrieval speed.

To store quiz data using MySQL databases, ensuring data retrieval and management are efficient: MySQL, a reliable and widely used relational database management system, will be used for storing quiz-related data. This includes storing quiz questions, possible answers, correct answers, and user responses. By utilizing MySQL, the application will benefit from efficient data querying, secure storage, and optimized data retrieval. The choice of MySQL will also ensure that the application can handle multiple quizzes and participants concurrently, with minimal latency in retrieving data.

To build a responsive and interactive user interface by making use of Java Swing Components: The front end of the application will be designed using Java Swing, a powerful toolkit for building graphical user interfaces (GUIs). The focus will be on creating a responsive and intuitive interface where users can easily navigate between different sections, such as quiz creation, participation, and results. The use of Swing components will allow for the development of buttons, menus, panels, and other interactive elements that respond to user inputs, providing a seamless and engaging experience for both quiz creators and participants.

To offer role-based access for students (quiz takers) and teachers (quiz creators): The application will include a role-based access control system to ensure that different users have appropriate permissions. For instance, teachers or quiz organizers will have the ability to create quizzes, modify questions, and set quiz parameters. On the other hand, students or participants will only have access to quizzes they are authorized to take, and they will be restricted from altering quiz content. This role-based structure will ensure that the system remains secure and that each user interacts with the application in a way that aligns with their designated role.

To allow quiz organizers to set time limits and organize questions into sections for efficient quiz management: One of the advanced features of the application will be the ability for quiz organizers to manage quizzes more effectively by setting time limits for individual quizzes or sections within a quiz. Additionally, questions can be organized into sections, which could be based on difficulty levels, topics, or question types (e.g., multiple-choice, true/false). This functionality will help ensure that the quiz-taking experience is structured and that participants are challenged appropriately, while also giving organizers more control over how quizzes are presented and managed.

1.4. Scope:

The scope of Quizventure Pro covers a wide range of functionalities aimed at enhancing educational and training experience. For educational institutions, the platform enables student profile management and personalized learning, the scope are stated below:

Educational Institutions: The platform allows students to register their profiles, which helps institutions efficiently handle student information and personalize learning experiences.

Training Programs: Administrators can create and customize quizzes, making it possible to develop varied assessments tailored to specific training needs, such as employee skill evaluations or professional certifications.

Data Management: The app uses a MySQL database for storing user and quiz data securely. This ensures that data retrieval is efficient and reliable, benefiting administrators who need to manage and analyse large volumes of information.

User Experience: The application provides a user-friendly desktop interface, which enhances accessibility and ease of use for both students and administrators. This intuitive design facilitates smooth interaction with the platform.

Assessment and Evaluation: Quizventure supports various testing and assessment needs in educational and corporate environments, making it a valuable tool for measuring knowledge and skills effectively across different sectors.

Chapter 2

Proposed System

The Quizventure system is designed to address the challenges faced by traditional quiz management methods. By incorporating modern technologies such as Java for the frontend and MySQL for the backend, the system ensures efficient, secure, and scalable quiz management.

2.1. Features and Functionality

The Quizventure system is designed to streamline the process of quiz creation, management, and participation through a range of innovative features. It supports multiple user roles, ensuring that both students and teachers have access appropriate to their needs, with administrative capabilities reserved for teachers, the features are stated below:

User Roles: The application supports multiple user roles such as students and teachers. The use of multiple user roles helps us cater the individual based on his/her hierarchical level in an institute, and shift the command of administrative accesses strictly to the teachers.

Quiz Creation and Management: Teachers can create common management questions according to the user requirement. The quizzes mainly have to be MCQ formatted, as they are meant for a quick academic evaluation of the candidate, and not for a more prolonged and conclusive analysis of his/her calibre.

MySQL Databases Integration: User information (like his/her First and Last Name, Roll Numbers, Email IDs, etc.) and quiz data (namely, the quiz questions, the options and the correct answers) are stored in a MySQL database for easy access and management in a tabular form.

Responsive User Interface: The application is built using Java, ensuring a responsive interface that is easy to navigate for all users.

Real-time Scoring: Once a student completes a quiz, the system automatically calculates the score and provides instant results.

Chapter 3

Project Outcomes

Taking into consideration the identified problems, we have kept the app extremely uncomplicated and intuitive in handling. Minimising the hassles in operation and taking care of the fact that the overall user experience is not hindered.

The final Outcomes of the project are as expected:

Admin Registration: To be able to gain the access to create and manage quizzes the individual would have to register on the app as an Admin. Once they have entered all the necessary information asked for, they can begin their quiz making journey!

Quiz Creation: After being identified as an Admin on the platform, he/she can design and create quizzes, including setting questions, multiple choice options and correct answers. Admins would also have the liberty to save and manage quizzes, including editing or deleting them.

Student Registration: Just like the admin user model, students are also required to register on the platform by providing necessary details (e.g., Name, email, and roll number) for their identification. The "student registration" process is a recurring action, requiring students to register each time they wish to attempt a quiz on the platform.

Quiz Access: Registered students can view and select available quizzes created by admins. Students can start attempting quizzes, answering questions and submitting responses.

Final Scoring: The student, (along with the teachers) can see the quiz taker's score once the quiz is over. Apart from this fact, the quiz taking candidate can also view his/her score gradual incrementing right by the side of the quiz dashboard!

Chapter 4

Software Requirements

To develop and run the Quizventure app, several software tools are required for both the development environment and deployment. Below is a list of the key software components and their purpose:

Java Development Kit (JDK)

Version: JDK 11 or higher

Purpose: The Java Development Kit is required for writing, compiling, and running Java applications. It provides the necessary libraries and tools to build the frontend for the Quizventure app. JDK includes the Java Runtime Environment (JRE), which is essential for executing the Java application on various platforms.

Integrated Development Environment (IDE): NetBeans

Purpose: An IDE like NetBeans is essential for coding and debugging Java programs. It offers features such as syntax highlighting, code completion, and integrated debugging, which streamline the development process.

JavaFX/Swing

Purpose: JavaFX or Swing are used for building the graphical user interface (GUI) for the app. This includes windows, buttons, input fields, and other visual elements that make the app interactive.

MySQL Server

Version: MySQL 8.0 or higher

Purpose: MySQL serves as the backend database for the Quizventure app. It stores user information, group details, and expense records. MySQL offers a relational database system that supports complex queries, transactions, and secure data handling.

JDBC (Java Database Connectivity)

Purpose: JDBC is an API that enables Java applications to connect to relational databases such as MySQL. It is required to execute SQL queries from the frontend and retrieve data, ensuring seamless communication between the app and the database.

Chapter 5

Project Design

The Project Design for Quizventure focuses on creating an engaging and efficient user experience through a well-structured interface. Leveraging Java's Swing framework, we aim to deliver a responsive and intuitive platform for both students and administrators. The design emphasizes clarity and usability, ensuring that users can effortlessly navigate through features such as quiz creation, participation, and result viewing. By prioritizing simplicity and accessibility, we strive to enhance the overall functionality and user satisfaction of the application.

5.1. Database Design

The backend of the Quizventure system is managed by a MySQL database, which stores user information, quiz data, and results. The database is structured to ensure efficient data retrieval and storage, with relational links established between user profiles, quizzes, and performance scores. The table structure includes entities for users (students and administrators), quizzes, and results, with appropriate relationships to ensure data integrity.

The database is designed to handle large volumes of data, ensuring scalability for institutions with a large number of users and quizzes. Indexing and optimization techniques are employed to improve query performance, enabling fast access to data even as the dataset grows.

Field	Type	Description
first	varchar (100)	First name of the administrator (i.e. teacher's name)
last	varchar (100)	Last name of the admin
username	varchar (150)	The username that is set by the admin for login
pass	varchar (150)	The password that is set by the admin for login

Table 5.2.1: Table for storing the admin login credentials

The table outlines the key fields required for an administrator's login information. It includes the first name and last name of the admin, stored as varchar(100) to allow for up to 100 characters each. The username, used for login, is stored as varchar(150) to accommodate a larger input size. Additionally, the password is stored as varchar(150), ensuring secure and unique access for each administrator. These fields collectively ensure proper identification and secure login credentials for administrators within the system.

Field	Type	Description
first_name	varchar (255)	First name of the student (i.e. student's name)
last_name	varchar (255)	Last name of the quiz taker
roll_no	varchar (20)	Roll number of the quiz taker
email	varchar (255)	Email address of the quiz taker

Table 5.2.2: Table for storing the quiz taker's credentials

This table outlines the essential fields for storing student information. The first_name and last_name fields, both varchar(255), capture the first and last names of the quiz takers, respectively. The roll_no field, defined as varchar(20), records the unique roll number of each quiz taker. Finally, the email field, also varchar(255), stores the email address of the quiz taker, ensuring effective communication and identification within the system. These fields collectively provide a comprehensive profile for each student.

Field	Type	Description
id	int	Unique identifier for each question.
name	varchar (500)	Name of the Student
opt1	varchar (500)	First option
opt2	varchar (500)	Second option
opt3	varchar (500)	Third option
opt4	varchar (500)	Fourth option
answer	varchar (500)	Correct option (i.e. the answer)

Table 5.2.3: Table for storing questions and their answers

This table outlines the key fields necessary for quiz questions. Each question is uniquely identified by an id field, which is an integer. The name field, a varchar(500), stores the name of the student. The table also includes four option fields (opt1, opt2, opt3, opt4), each defined as varchar(500), to provide possible answers for the quiz question. Finally, the answer field, also a varchar(500), records the correct option. These fields collectively ensure a structured and comprehensive way to manage quiz questions and their respective answers.

Field	Type	Description
first_name	varchar (255)	First name of the quiz taker (i.e. student's name)
last_name	varchar (255)	Last name of the quiz taker
roll_no	varchar (20)	Roll number of the quiz taker
email	varchar (255)	Email address of the quiz taker
marks_obt	int	Marks obtained by the quiz taker
total_mks	int	Total of all marks for the quiz

Table 5.2.4: Table for storing the final result

This table is designed to capture essential information about quiz takers. It includes fields for the first name and last name of the student (first_name and last_name), each allowing up to 255 characters. The roll_no field records the quiz taker's roll number, limited to 20 characters. Their email address is stored in the email field, also up to 255 characters. Additionally, the table tracks the marks_obt, which is the marks obtained by the quiz taker, and total_mks, which represents the total marks possible for the quiz. These fields collectively provide a comprehensive profile of each quiz participant and their performance.

Chapter 6

Project Scheduling

A schedule outlining planned start and finish dates, durations, and allocated resources for each task, ensuring tasks are completed on time and within budget for effective task and time management.

During the project timeline, the group members undertook various tasks to ensure the successful completion of the Mini Project. In the first two weeks of July, Yash Bhosale and Paras Mahajan focused on group formation and topic finalization, identifying the project's scope and objectives. Following this, from the last week of July to the first week of August, Paras Mahajan and Harshal Gaikwad identified the key functionalities needed for the Mini Project.

From the second week to the last week of August, a collaborative effort involving Paras Mahajan, Harshal Gaikwad, and Manas Chaudhari was made to discuss the project topic, utilizing a paper prototype to visualize ideas. Concurrently, Yash Bhosale, Paras Mahajan, and Manas Chaudhari worked on designing the Graphical User Interface (GUI) during the last week of August to the first week of September, focusing on creating a user-friendly layout.

In the first two weeks of September, the team, including Yash Bhosale, Paras Mahajan, Harshal Gaikwad, and Manas Chaudhari, prepared for Presentation I. Harshal Gaikwad then took the lead from the second week to the last week of September, concentrating on database design, which was crucial for the project's functionality.

By the last week of September, Harshal Gaikwad completed the database connectivity for all modules, ensuring seamless integration. In the first week of October, Paras Mahajan, Harshal Gaikwad, and Manas Chaudhari worked together to integrate all project modules and began report writing, while Yash Bhosale joined them for Presentation II in the same week. This structured approach allowed the team to efficiently collaborate and advance their project systematically.

A Gantt chart's visual timeline allows you to

Smartsheet Tip see details about each task as well as project dependencies.

INSTITUTE & DEPARTMENT NAME: AP SHAH INSTITUTE OF TECHNOLOGY (DSE-Data Science)

DATE: 10-8-24

DATE: 10-8-24

[illegible]

Chapter 7

Results

Quizventure – The Ultimate Technical Quiz successfully achieves its goal of providing an efficient platform for creating, managing, and participating in technical quizzes. Below is a detailed summary of the results obtained during the development and testing phases.

7.1. Performance and Responsiveness

Data Handling: The application efficiently handles the storage and retrieval of large volumes of quiz data. MySQL queries are optimized to ensure fast response times even with multiple quizzes and participants. The JDBC connection between the Java frontend and the MySQL backend performs consistently during various operations, such as quiz creation and scoring.

User Interface: The user interface, built using Java Swing, is intuitive and easy to navigate. Users can seamlessly switch between different screens (Login, Quiz Creation, Quiz Attempt, Results), and actions like creating quizzes or submitting answers are processed quickly with minimal delay.

7.2. Testing Results

Functional Testing: The app was rigorously tested to ensure all features work as intended.

Test cases included:

- Registering multiple users and assigning different roles (student, admin).
- Creating quizzes with varying difficulty levels and question types.
- Students attempting quizzes and receiving immediate feedback.
- Administrators managing multiple quizzes and viewing results.

All test cases passed successfully, and the app handled user roles, quiz data, and scoring without errors.

Performance Testing: The system was tested with a larger number of users and quizzes to assess scalability. The app maintained stable performance, with optimized query execution and no noticeable lag, even with a high volume of quiz attempts and data retrieval operations.

7.3. Challenges and Resolutions

Database Integrity: Ensuring data consistency and accuracy was a key challenge during development, particularly with managing relationships between users, quizzes, and results. To resolve this, the database schema was designed with relational integrity constraints such as foreign keys and validation checks. This ensured no orphaned records or inconsistencies in quiz results.

Error Handling: Error-handling mechanisms were implemented to manage scenarios such as duplicate quiz entries, invalid user data, and system errors during quiz creation or submission. These mechanisms provided clear and helpful error messages to users while maintaining the app's stability.

7.4. UI Design

The user interface (UI) for Quizventure is designed using Java's Swing framework, which provides a responsive and intuitive environment for both students and administrators. The UI includes clear navigation options that allow users to easily access different features of the application, such as quiz creation, quiz participation, and result viewing.

For students, the interface is designed to be simple and easy to use, displaying quizzes in a clean format with large, readable text and prominent buttons for submitting responses. Administrators are provided with a more detailed interface, enabling them to create and manage quizzes with ease. The design prioritizes ease of use, with minimal clutter and clearly labelled buttons and forms.



STUDENT REGISTRATION

FIRST NAME:


LAST NAME:

ROLL NO:

EMAIL:

Figure 7.1.1: Student Registration Page

This page collects information from the students (namely, First and Last Name, roll number, and Email IDs).



ADD THE QUESTION

QUESTION ID :

QUESTION :

OPTION 1 :

OPTION 2 :

OPTION 3 :

OPTION 4 :

ANSWER :

Figure 7.1.2: Page to Add Questions

This page takes input from the quiz maker in the form of question, options, and correct answer.

UPDATE A QUESTION

QUESTION ID : 3 **SEARCH**

QUESTION : What is Pycharm?

OPTION 1 : A C# IDE

OPTION 2 : A Java IDE

OPTION 3 : A Eclipse IDE

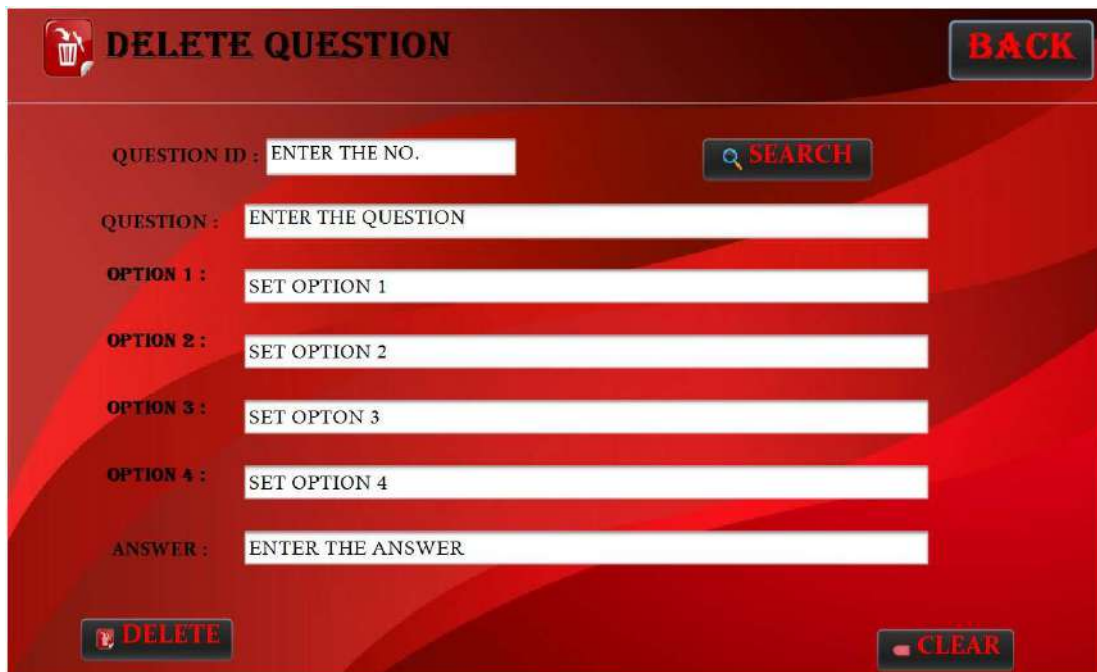
OPTION 4 : A Python IDE

ANSWER : A Python IDE

CLEAR **UPDATE**

Figure 7.1.3: Page to Update Questions

This page helps the admin update any question(s) or answer(s) that the admin previously might have set incorrectly.



DELETE QUESTION BACK

QUESTION ID : SEARCH

QUESTION :

OPTION 1 :

OPTION 2 :

OPTION 3 :

OPTION 4 :

ANSWER :

DELETE CLEAR

Figure 7.1.4: Page to Delete Questions

This page lets the admin delete any unwanted question(s) that he/she may not need anymore.



ALL QUESTIONS GET BACK

QUES.NO	QUESTION	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1.	What does GUI stand for?	Graphical User Interface	Graphical User Interaction	Graphical Unified Interface	Graphical User Integration	Graphical User Interface
2.	Which language is known as the 'mother'?	C	Java	Python	Assembly	Assembly
3.	Which one is an operating system?	Python	Java	Linux	HTML	Linux
4.	Which of the following is a database ms.	MySQL	PHP	HTML	CSS	MySQL
5.	Which of these is not a programming la.	Python	Java	HTML	C++	HTML
6.	Which language is primarily used for we.	Python	HTML	Gr++	Java	HTML
7.	What does IoT stand for?	Structured Query Language	Simple Query Language	Standard Query Language	System Query Language	Structured Query Language
8.	What does IoT stand for?	Internet of Things	Internet of Technology	Integration of Technology	Implementation of Technology	Internet of Things
9.	What is the function of a GPU in a com.	Process graphical data	Store data	Manage internet connections	Control peripheral devices	Process graphical data
10.	Which SQL statement is used to updat.	UPDATE	SAVE	MODIFY	CHANGE	UPDATE

Figure 7.1.5: Page to Display All Questions

This page will be visible to the admin after he/she is done setting all the questions, and wants to review them.

Chapter 8

Conclusion

Quizventure – The Ultimate Technical Quiz successfully fulfils its primary objective of providing a robust platform for creating, managing, and participating in technical quizzes. Through a Java-based frontend and MySQL backend, the application offers an efficient, user-friendly solution for conducting quizzes, ensuring secure data storage, real-time scoring, and smooth user interactions.

8.1. Project Success

The app's key functionalities, including user registration, quiz creation, quiz participation, and real-time scoring, were implemented successfully. Users can seamlessly interact with the app, participate in quizzes, and view their scores instantly, while administrators can manage quizzes and monitor user performance. The clear separation between the frontend and backend, supported by MySQL's efficient data handling, ensures secure storage of quiz data and user information.

The project was tested across various use cases, including quizzes of different lengths and complexities, demonstrating its reliability and effectiveness in real-world scenarios. The system's architecture, with role-based access control and database design, provides a strong foundation for scalability and future enhancements.

8.2. Challenges and Learning Outcomes

During development, some challenges were encountered, particularly in managing user roles and ensuring data consistency between quiz creation and quiz-taking functionalities. These issues were resolved by using relational database design principles and implementing error-handling mechanisms. Working with JDBC for Java-to-database communication offered valuable experience in managing efficient data transactions and ensuring smooth interactions between the application and the database.

Chapter 9

Future Work

9.1. Future Work

While the app currently handles basic quiz creation and participation, there is significant potential for expanding its functionality in future versions. Some possible enhancements include:

Advanced Question Types: Adding support for different types of questions, such as drag-and-drop, matching, or multimedia-based questions.

Online Quiz Mode: Enabling live, real-time quizzes where participants can compete against each other.

Mobile and Web Versions: Developing mobile and web versions of the app to increase accessibility across various platforms.

Detailed Performance Analytics: Offering insights and reports for both students and administrators, helping users analyse quiz performance trends.

Timed Quizzes: Introducing time-bound quizzes for better simulation of real-world testing conditions.

9.2. Final Thoughts

Overall, the **Quizventure** project has been a valuable learning experience, combining knowledge of Java programming, database management, and user interface design. The successful implementation of the core features reflects the effectiveness of the chosen technologies and the strength of the system's design. Looking forward, the app provides a solid foundation for further development, both in terms of functionality and usability, offering numerous opportunities for expansion.

References

- [1] MySQL command references: (<https://www.w3schools.com/sql/>)
- [2] GeeksforGeek, “SQL database references” (<https://www.geeksforgeek.org/>)
- [3] MySQL Documentation, "MySQL 8.0 Reference Manual" (<https://dev.mysql.com/doc/refman/8.0/en/>)
- [4] Oracle, "JDBC Guide – Java Database Connectivity" (<https://docs.oracle.com/javase/tutorial/jdbc/>)
- [5] GeeksforGeeks, “Introduction to JavaFX” (<https://www.geeksforgeeks.org/javafx-tutorial/>)
- [6] Stack Overflow, “Various Java and MySQL Development Queries” (<https://stackoverflow.com/>)