A QUIZ APP

A PROJECT REPORT

Submitted by

HARSHAVARDHINI P (2303811710422060)

In partial fulfillment of requirements for the award of the course

CGB1221-DATABASE MANAGEMENT SYSTEMS

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE- 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **“A QIUZ APP”** is the bonafide work of **HARSHAVARDHINI P (2303811710422060)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Mrs. A. Delphin Carolina Rani, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram – 621112.

S. uma

SIGNATURE

Ms. S. Uma Mageshwari, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram – 621112.

Submitted for the viva-voce examination held on 02.06.2025

S. uma 2/6/25

K. 2/6/25.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on **“A QUIZ APP”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS.**

Signature

Harshavardhini P.

HARSHAVARDHINI P

Place : Samayapuram

Date : 02.06.2025

ACKNOWLEDGEMENT

It is with great pride that I express my gratitude and in-debt to our institution **K.Ramakrishnan College of Technology (Autonomous)‖**, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of my study in college.

I would like to express my sincere thanks to my beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to my project and offering adequate duration in completing my project.

I would like to thank **Dr. N. VASUDEVAN, M. Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E., Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express my deep expression and sincere gratitude to my project supervisor **Ms. S. UMA MAGESHWARI, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render my sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express my special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusion.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The development of a comprehensive, interactive digital platform designed to streamline and enhance the process of quiz administration and participation in an educational setting. By integrating multiple functional modules ranging from secure user authentication and intuitive quiz creation tools to real- time participation interfaces and sophisticated performance analytics—the system addresses the multifaceted needs of both educators and learners. It enables teachers to efficiently design, schedule, and manage quizzes while providing students with an engaging environment to attempt quizzes, receive instant feedback, and monitor their academic progress. Advanced analytical features allow educators to gain deep insights into individual and collective student performance, identifying areas of weakness and informing targeted pedagogical interventions. Furthermore, the incorporation of real-time communication capabilities ensures that users receive immediate notifications and updates, enhancing interactivity and responsiveness. Built upon robust front-end and back-end technologies, including HTML, Python, MySQL, and PostgreSQL, the platform emphasizes data integrity, scalability, and usability. Ultimately, the system aspires to foster a more dynamic, transparent, and data-driven learning experience, supporting continuous academic improvement and promoting a culture of active engagement and accountability within educational institutions.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5: BUILD DATABASE MANAGEMENT SYSTEMS APPLICATIONS FOR SOLVING REAL - TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
Innovative digital solution offers an intuitive platform that simplifies quiz creation and management for educators, ensuring assessments are both efficient and accessible. It enhances the learning experience by enabling students to participate in quizzes effortlessly, receive instant feedback through scores and rankings, and review their performance comprehensively. Additionally, it provides educators with detailed insights into student progress, allowing for the identification of learning gaps and the delivery of precise, constructive feedback. By fostering continuous evaluation and engagement, the platform contributes to a more interactive, motivating, and effective educational environment that promotes sustained academic growth.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO9 -3** **PO10-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO.
	ABSTRACT	Viii
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS	xiii
1	**INTRODUCTION**	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 SQL and Database concepts	1
2	**PROJECT METHODOLOGY**	3
	2.1 Proposed Work	3
	2.2 Block Diagram	3
3	**MODULE DESCRIPTION**	4
	3.1 Use Authentication Module	4
	3.2 Quiz Creation And Management Module	4
	3.3 Quiz Participation Module	4
	3.4 Performance Analytics Module	5
	3.5 Real – time Communication Module	5
4	**CONCLUSION & FUTURE SCOPE**	7
	4.1 Conclusion	7
	4.2 Future Scope	7

5	**APPENDIX A SOURCE CODE**	9
	APPENDIX B SCREENSHOTS	33
	REFERENCES	36

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
2.1	Block Diagram	3

LIST OF ABBREVIATIONS

SQL	-	Structured Query Language
MySQL	-	My Structured Query Language
PostgreSQL	-	Postgre Structured Query Language
EC2	-	Elastic Compute Cloud
RDS	-	Relational Database Service
S3	-	Simple Storage Service
DBMS	-	Database Management Systems
PHP	-	Hypertext Preprocessor
HTML	-	HyperText Markup Language

CHAPTER 1

INTRODUCTION

1.1 Objective

To develop an efficient and user-friendly digital platform for quiz management that simplifies the process of quiz creation, administration, and participation. The platform is designed to enhance student engagement through interactive and intuitive features, making learning more dynamic and enjoyable. For educators, it offers powerful tools to monitor student performance through real-time analytics, enabling them to track progress, identify learning gaps, and make informed instructional decisions. By combining interactivity with insightful data, the platform aims to improve the overall teaching and learning experience, fostering a more effective and personalized educational environment.

1.2 Overview

Focuses on building an interactive digital platform designed to simplify and enhance the quiz-taking experience for both educators and students. It allows educators to easily create, manage, and deploy quizzes, while students can seamlessly participate and receive immediate feedback on their performance.The system also provides detailed analytics and insights to help educators track student progress, identify areas requiring additional attention, and offer targeted feedback. By integrating user-friendly features and real-time performance tracking, the platform aims to foster a more engaging, efficient, and data-driven learning environment.

1.3 SQL and Database Concepts

The project is built upon fundamental SQL and database concepts, which are essential for managing and organizing data efficiently. A database is a structured collection of data that allows for easy access, management, and updating, while SQL (Structured Query Language) is the standard language used to communicate with relational databases. In this project, databases are used to store crucial information such as quiz questions, student details, scores, and performance metrics in an organized manner. MySQL and PostgreSQL serve as the back-end systems, enabling reliable data handling through tables, relationships, and constraints that ensure data integrity and consistency. Using SQL, the platform performs essential operations like data insertion, updating records, retrieving quiz results, and generating analytical reports for teachers. These operations not only allow smooth quiz management but also provide valuable insights into student progress. By applying core database principles such as normalization, indexing, and relational modeling the project ensures secure, scalable, and efficient data management, which is vital for supporting multiple users and maintaining real-time performance.

CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The proposed system is designed to create an efficient, user-friendly platform that simplifies quiz management and enhances digital learning experiences. It will include key modules such as user authentication to manage secure access for teachers and students, and a quiz creation module that allows teachers to design, edit, and schedule quizzes with ease. Students will participate through an interactive interface that provides real-time scores and feedback. To support continuous learning, a performance analytics module will offer detailed reports and insights into individual and class performance. Additionally, real-time communication will be implemented to deliver instant notifications and updates. The system will be built using HTML and Python for the front-end, with MySQL and PostgreSQL ensuring secure and structured back-end data management, making the platform reliable, scalable, and well-suited for educational institutions.

2.2 Block Diagram

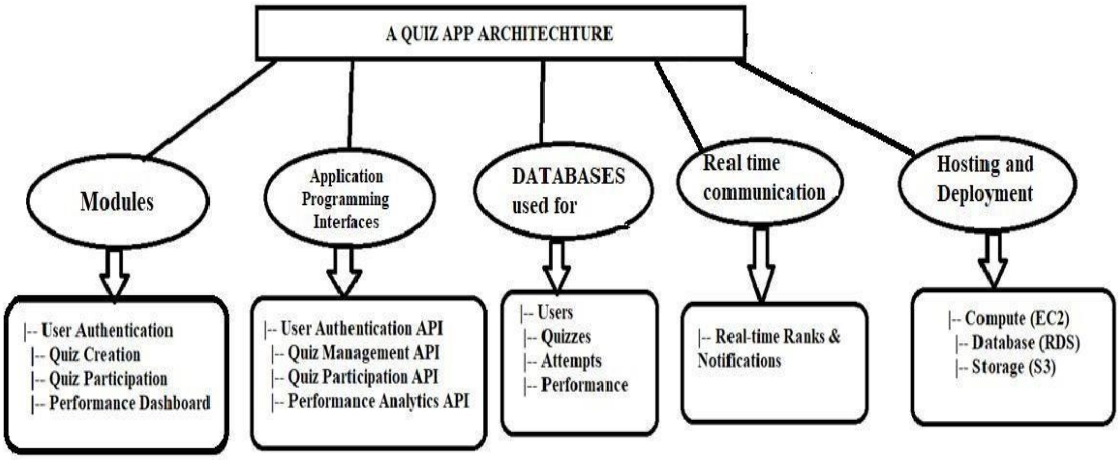


Fig 2.1 Block Diagram

CHAPTER 3
MODULE DESCRIPTION

3.1 User Authentication Module

This module is pivotal in ensuring that only authorized users—teachers and students can access the platform securely. It handles the entire authentication flow, including user registration, where new users can sign up by providing essential details, and login, which verifies user credentials to grant access. Additionally, it manages user sessions to maintain a seamless experience, ensuring that users stay logged in during active use and are logged out automatically after periods of inactivity. By implementing robust session management and authentication protocols, this module safeguards user data and protects the platform from unauthorized access.

3.2 Quiz Creation and Management Module

The quiz creation and management module empowers teachers to efficiently design and control quizzes. It provides an intuitive quiz builder that enables teachers to input questions, multiple-choice answers, and the correct responses. Teachers can also edit quizzes to update or correct content prior to quiz launch, ensuring accuracy and relevance. Moreover, it offers a scheduling feature, allowing teachers to set specific dates and times when quizzes become available to students, streamlining the quiz delivery process. This module plays a crucial role in maintaining the quality and organization of assessments.

3.3 Quiz Participation Module

This module is designed to deliver an engaging and straightforward quiz-taking experience for students. It provides a user-friendly interface where students can access quizzes within defined deadlines and submit their answers in real-time

The module enhances motivation and competitiveness by displaying live rankings based on quiz performance as students complete their attempts. Post-quiz, students gain access to a detailed performance review, where they can see which questions they got right or wrong along with explanations, helping them to understand their mistakes and deepen their learning.

3.4 Performance Analytics Module

The performance analytics module transforms raw quiz data into insightful reports for teachers. It generates individual reports for each student, detailing scores, ranks, and personalized feedback on strengths and weaknesses. At the class level, it provides an aggregated view of performance, highlighting average scores and common problem areas, thus helping teachers identify learning gaps. Additionally, the module includes question analysis, which examines performance trends across individual questions, assisting educators in refining their teaching strategies and improving question design.

3.5 Real-time Communication Module

The Real-time Communication Module plays a crucial role in enhancing the interactivity and responsiveness of the quiz platform by integrating real-time communication technologies, such as WebSockets. This module ensures that updates and interactions occur instantly, without the need for users to refresh their pages or wait for periodic updates. One of its key features is the live updating of quiz rankings, allowing students to see their positions change in real-time as they progress through the quiz. This feature adds a competitive and engaging element to the quiz-taking experience.

In addition, students receive immediate notifications and reminders about upcoming quizzes, helping to boost participation and ensure they don’t miss scheduled assessments. These alerts can be customized and scheduled by teachers, allowing for better communication and time management. For educators, the module provides real-time alerts when students submit their quizzes, enabling them to monitor participation live and act promptly if needed.

CHAPTER 4
CONCLUSION AND FUTURE ENHANCEMENT

4.1 CONCLUSION

The development of this comprehensive quiz platform successfully bridges the gap between digital learning and interactive assessments by providing an intuitive, secure, and efficient environment for both teachers and students. Through modules such as user authentication, quiz management, real-time participation, and performance analytics, the system ensures streamlined quiz creation, seamless participation, and insightful analysis of results. The integration of real-time communication further enhances engagement and responsiveness. Overall, the project not only simplifies the management of quizzes but also fosters a more data-driven, transparent, and effective learning experience, contributing meaningfully to the advancement of modern educational practices.

4.2 FUTURE ENHANCEMENT

In the future, the system can be upgraded with cutting-edge features to enhance its functionality and user engagement. One key improvement could be the integration of AI- powered question banks that dynamically generate quiz questions based on curriculum standards and student performance data. Additionally, automated grading and instant feedback mechanisms would streamline the assessment process, saving valuable time for teachers. A dedicated mobile application could also be developed to ensure seamless access and usability across various devices, making the platform more flexible and convenient for both students and educators.

Another promising direction is the incorporation of adaptive learning techniques that tailor quiz content to each student’s learning pace and proficiency level, fostering a personalized educational experience. The system could further be enhanced with multilingual support, voice-assisted quizzes for better accessibility, and gamification elements like badges and leaderboards to maintain high student motivation. Advanced analytics and predictive tools could provide deeper insights into student trends and performance, enabling teachers to implement more effective intervention strategies and continuously improve the quality of learning.

APPENDIX A

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Quiz App</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            background-color: #f4f4f4;
            padding: 20px;
        }
        .container {
            max-width: 800px;
            margin: auto;
            background: white;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        h1, h2 {
            color: #333;
        }
        .module {
            margin: 20px 0;
            padding: 10px;
            background: #ddd;
            border-radius: 5px;
        }
        button {
            padding: 10px 15px;
            margin-top: 10px;
            font-size: 16px;
            color: white;
            background-color: #007BFF;
            border: none;
            border-radius: 5px;
```

```
         cursor: pointer;
       }
       button:hover {
         background-color: #0056b3;
       }
       .form-group {
         margin-bottom: 15px;
       }
       input, select {
         padding: 8px;
         width: 100%;
         border: 1px solid #ccc;
         border-radius: 5px;
       }
     </style>
     <script>
       function login() {
         var username = document.getElementById("username").value;
         var password = document.getElementById("password").value;
         var role = document.getElementById("role").value;

         if (username && password) {
           if (role === "teacher") {
             window.location.href = "file:///C:/xampp/htdocs/quiz_app/teacher.html";
           } else {
             window.location.href = "file:///C:/xampp/htdocs/quiz_app/student.html";
           }
         } else {
           alert("Please enter username and password.");
         }
       }
     </script>
   </head>
   <body>
     <div class="container">
       <h1>Quiz App</h1>

       <div class="login-form module">
         <h2>Login</h2>
<form onsubmit="event.preventDefault(); login();">
           <div class="form-group">
 <label for="username">Username:</label>
```

```
                    <input type="text" id="username" name="username" required>
                </div>
                <div class="form-group">
                    <label for="password">Password:</label>
                    <input type="password" id="password" name="password" required>
                </div>
                <div class="form-group">
                    <label for="role">Role:</label>
                    <select id="role" name="role">
                        <option value="teacher">Teacher</option>
                        <option value="student">Student</option>
                    </select>
                </div>
                <button type="submit">Login</button>
            </form>
        </div>
    </div>
</body>
</html>
```

INDEX.PHP

```
<?php
session_start();

// DB connection
$conn = new mysqli("localhost", "root", "", "quiz_app");
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Get form inputs
$username = $_POST['username'];
$password = $_POST['password'];
$role = $_POST['role'];

// Validate user
$sql = "SELECT * FROM users WHERE username = ? AND password = ? AND role = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("sss", $username, $password, $role);
$stmt->execute();
$result = $stmt->get_result
```

```
if ($result->num_rows === 1) {
    $_SESSION['username'] = $username;
    $_SESSION['role'] = $role;

    if ($role === "teacher") {
        header("Location: teacher.php");
    } else {
        header("Location: student.php");
    }
    exit();
} else {
    echo "<script>alert('Invalid credentials!'); window.location.href='index.html';</script>";
}
?>
```

STUDENT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Dashboard</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            background-color: #f4f4f4;
            padding: 20px;
        }
        .container {
            max-width: 800px;
            margin: auto;
            background: white;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
        .quiz-container {
            margin-top: 20px;
        }
        .quiz {
            margin-bottom: 20px;
            padding: 15px;
            background-color: #eee;
            border-radius: 8px;
        }
        button {
            padding: 10px 15px;
            margin-top: 10px;
            font-size: 16px;
            color: white;
            background-color: #007BFF;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
        .option-btn {
            display: block;
            margin: 10px auto;
            width: 80%;
            background-color: #007BFF;
        }
        .option-btn:disabled {
            cursor: default;
            opacity: 0.7;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Student Dashboard - Take a Quiz</h1>
        <div id="quiz-container" class="quiz-container">
            <p>Loading quizzes...</p>
        </div>
        <a href="file:///C:/xampp/htdocs/quiz_app/result.html"><button>Go to Result
Dashboard</button></a>
    </div>


    <script>
```

```
let correctCount = 0;
let totalQuestions = 0;
let answered = [];

function loadQuiz() {
    let quizzes = JSON.parse(localStorage.getItem("quizzes")) || [];
    totalQuestions = quizzes.length;
    let quizContainer = document.getElementById("quiz-container");
    quizContainer.innerHTML = "";

    if (quizzes.length === 0) {
        quizContainer.innerHTML = "<p>No quizzes available.</p>";
        return;
    }

    quizzes.forEach((quiz, index) => {
        let quizDiv = document.createElement("div");
        quizDiv.className = "quiz";
        quizDiv.innerHTML = `
            <h3>${quiz.question}</h3>
            ${quiz.options.map(option => `
                <button class="option-btn" onclick="checkAnswer(this, '${option}',
'${quiz.correctAnswer}', ${index})">${option}</button>
            `).join('')}
            <hr>
        `;
        quizContainer.appendChild(quizDiv);
    });
}

function checkAnswer(button, selected, correct, index) {
    if (answered[index]) return;
    answered[index] = true;

    let buttons = button.parentElement.querySelectorAll(".option-btn");
    buttons.forEach(btn => btn.disabled = true);

    if (selected === correct) {
        button.style.backgroundColor = "green";
        correctCount++;
    } else {
        button.style.backgroundColor="red";
```

```
                buttons.forEach(btn => {
                    if (btn.textContent === correct) {
                        btn.style.backgroundColor = "green";
                     }
                 });
            }

            if (answered.filter(Boolean).length === totalQuestions) {
                setTimeout(() => {
                    const name = prompt("Enter your name to view your result:");
                    if (name) {
                        const newResult = {
                            student_name: name,
                            correct: correctCount,
                            incorrect: totalQuestions - correctCount
                        };

                        localStorage.setItem("latestResult",  JSON.stringify(newResult));

                        let allResults = JSON.parse(localStorage.getItem("allResults")) || [];
                        allResults.push(newResult);
                        localStorage.setItem("allResults",  JSON.stringify(allResults));

                        window.location.href = "result.html";
                    }
                }, 500);
            }
        }

        window.onload = loadQuiz;
    </script>
</body>
</html>
```

STUDENT.PHP

```
<?php
$host = "localhost";
$db = "quiz_db";
$user = "root";
$pass = "";
```

```
$conn = new mysqli($host, $user, $pass, $db);
if ($conn->connect_error) die("Connection failed: " . $conn->connect_error);

$result = $conn->query("SELECT * FROM quizzes");
$quizzes = [];

while ($row = $result->fetch_assoc()) {
    $quizzes[] = [
        'question' => $row['question'],
        'options' => [$row['option1'], $row['option2'], $row['option3'], $row['option4']],
        'correctAnswer' => $row['correct_answer']
    ];
}

$conn->close();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Student Dashboard</title>
    <style>
        body { font-family: Arial, sans-serif; text-align: center; background: #f4f4f4; padding: 20px; }
        .container { max-width: 800px; margin: auto; background: white; padding: 20px; border-radius: 10px; box-shadow: 0 0 10px rgba(0,0,0,0.1); }
        .quiz-container { margin-top: 20px; }
        .quiz { margin-bottom: 20px; padding: 15px; background-color: #eee; border-radius: 8px; }
        button { padding: 10px 15px; margin-top: 10px; font-size: 16px; color: white; background-color: #007BFF; border: none; border-radius: 5px; cursor: pointer; }
        button:hover { background-color: #0056b3; }
        .option-btn { display: block; margin: 10px auto; width: 80%; background-color: #007BFF; }
        .option-btn:disabled { cursor: default; opacity: 0.7; }
    </style>
</head>
<body>
<div class="container">
    <h1>Student Dashboard - Take a Quiz</h1>
    <div id="quiz-container" class="quiz-container">
        <p>Loading quizzes...</p>
    </div>
```

```
</div>

<script>
    const quizzes = <?php echo json_encode($quizzes); ?>;
    let correctCount = 0;
    let totalQuestions = quizzes.length;
    let answered = [];

    function loadQuiz() {
        const quizContainer = document.getElementById("quiz-container");
        quizContainer.innerHTML = "";

        if (quizzes.length === 0) {
            quizContainer.innerHTML = "<p>No quizzes available.</p>";
            return;
        }

        quizzes.forEach((quiz, index) => {
            let quizDiv = document.createElement("div");
            quizDiv.className = "quiz";
            quizDiv.innerHTML = `
                <h3>${quiz.question}</h3>
                ${quiz.options.map(option => `
                    <button class="option-btn" onclick="checkAnswer(this, '${option}', '${quiz.correctAnswer}', ${index})">${option}</button>
                `).join('')}
                <hr>
            `;
            quizContainer.appendChild(quizDiv);
        });
    }

    function checkAnswer(button, selected, correct, index) {
        if (answered[index]) return;
        answered[index] = true;

        const buttons = button.parentElement.querySelectorAll(".option-btn");
        buttons.forEach(btn => btn.disabled = true);

        if (selected === correct) {
            button.style.backgroundColor = "green";
            correctCount++;
        } else {
            button.style.backgroundColor = "red";
            buttons.forEach(btn => {
                if (btn.textContent === correct) {
                    btn.style.backgroundColor = "green";
                }
            });
        }

        if (answered.filter(Boolean).length === totalQuestions) {
            setTimeout(() => {
                const name = prompt("Enter your name to view your result:");
                if (name) {
                    const form = new FormData();
                    form.append("student_name", name);
                    form.append("correct", correctCount);
                    form.append("incorrect", totalQuestions - correctCount);

                    fetch("save_result.php", {
                        method: "POST",
                        body: form
                    })
                    .then(res => res.text())
                    .then(data => {
                        alert("Your result has been submitted.");
                        window.location.href = "result.php";
                    });
                }
            }, 500);
        }
    }

    window.onload = loadQuiz;
</script>
</body>
</html>
```

TEACHER.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Teacher Dashboard</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            padding: 20px;
            text-align: center;
        }
        .container {
            max-width: 900px;
            margin: auto;
            background: white;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
        h1, h2 {
            color: #333;
        }
        button {
            padding: 10px 15px;
            margin-top: 10px;
            font-size: 16px;
            color: white;
            background-color: #007BFF;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
        .quiz-container {
            margin: 20px 0;
            padding: 15px;
            background: #eee;
            border-radius: 5px;
            text-align: left;
        }
        .form-group {
            margin-bottom: 10px;
        }
        input {
            padding: 8px;
            width: 95%;
            border: 1px solid #ccc;
            border-radius: 5px;
            margin: 5px 0;
        }
        .results-section {
            margin-top: 40px;
        }
        table {
            width: 100%;
            margin-top: 15px;
            border-collapse: collapse;
        }
        th, td {
            padding: 10px;
            border: 1px solid #999;
        }
        th {
            background-color: #007BFF;
            color: white;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Teacher Dashboard</h1>
        <h2>Create Up to 5 Quizzes</h2>
        <button onclick="addQuiz()">Add Quiz</button>
        <div id="quiz-section"></div>
        <button onclick="saveQuizzes()">Save Quizzes</button>

         <div class="results-section">
             <h2>Student Results</h2>
        <button onclick="loadResults()">Show Results</button>
        <div id="resultsDisplay"></div>
    </div>
</div>

<script>
    let quizCount = 0;
    const maxQuizzes = 5;

    function addQuiz() {
        if (quizCount >= maxQuizzes) {
            alert("You can only create up to 5 quizzes.");
            return;
        }

        const quizDiv = document.createElement("div");
        quizDiv.classList.add("quiz-container");

        quizDiv.innerHTML = `
            <h3>Quiz ${quizCount + 1}</h3>
            <div class="form-group">
                <label>Question:</label>
                <input type="text" class="question" required>
            </div>
            <div class="form-group">
                <label>Options:</label>
                <input type="text" class="option1" placeholder="Option 1" required>
                <input type="text" class="option2" placeholder="Option 2" required>
                <input type="text" class="option3" placeholder="Option 3" required>
                <input type="text" class="option4" placeholder="Option 4" required>
            </div>
            <div class="form-group">
                <label>Correct Answer:</label>
                <input type="text" class="correct-answer" required>
            </div>
        `;
        document.getElementById("quiz-section").appendChild(quizDiv);
        quizCount++;
    }

    function saveQuizzes() {
        const quizzes = [];
    const containers = document.querySelectorAll(".quiz-container");

    for (let i = 0; i < containers.length; i++) {
        const div = containers[i];
        const question = div.querySelector(".question").value.trim();
        const option1 = div.querySelector(".option1").value.trim();
        const option2 = div.querySelector(".option2").value.trim();
        const option3 = div.querySelector(".option3").value.trim();
        const option4 = div.querySelector(".option4").value.trim();
        const correctAnswer = div.querySelector(".correct-answer").value.trim();

        if (!question || !option1 || !option2 || !option3 || !option4 || !correctAnswer) {
            alert(`Please fill all fields in Quiz ${i + 1}`);
            return;
        }

        quizzes.push({
            question,
            options: [option1, option2, option3, option4],
            correctAnswer
        });
    }

    if (quizzes.length > 0) {
        localStorage.setItem("quizzes", JSON.stringify(quizzes));
        alert("Quizzes saved successfully!");
    }
}

function loadResults() {
    const results = JSON.parse(localStorage.getItem("allResults")) || [];
    const container = document.getElementById("resultsDisplay");

    if (results.length === 0) {
        container.innerHTML = "<p>No results available.</p>";
        return;
    }

    let html = `
        <table>
            <tr>
                <th>Student Name</th>
```

```
                        <th>Correct</th>
                        <th>Incorrect</th>
                    </tr>
            `;

            results.forEach(res => {
                html += `
                    <tr>
                        <td>${res.student_name}</td>
                        <td>${res.correct}</td>
                        <td>${res.incorrect}</td>
                    </tr>
                `;
            });

            html += `</table>`;
            container.innerHTML = html;
        }
    </script>
</body>
</html>
```

TEACHER.PHP

```
<?php
session_start();
if (!isset($_SESSION['username']) || $_SESSION['role'] !== 'teacher') {
    header("Location: index.html");
    exit();
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Teacher Dashboard</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            padding: 20px;
            text-align: center;
}
.container {
   max-width: 900px;
   margin: auto;
   background: white;
   padding: 20px;
   border-radius: 10px;
   box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
h1, h2 {
   color: #333;
}
button {
   padding: 10px 15px;
   margin-top: 10px;
   font-size: 16px;
   color: white;
   background-color: #007BFF;
   border: none;
   border-radius: 5px;
   cursor: pointer;
}
button:hover {
   background-color: #0056b3;
}
.quiz-container {
   margin: 20px 0;
   padding: 15px;
   background: #eee;
   border-radius: 5px;
   text-align: left;
}
.form-group {
   margin-bottom: 10px;
}
input {
   padding: 8px;
   width: 95%;
   border: 1px solid #ccc;
   border-radius: 5px;
   margin: 5px 0;
}
```

```
        .results-section {
            margin-top: 40px;
        }
        table {
            width: 100%;
            margin-top: 15px;
            border-collapse: collapse;
        }
        th, td {
            padding: 10px;
            border: 1px solid #999;
        }
        th {
            background-color: #007BFF;
            color: white;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome, <?php echo $_SESSION['username']; ?> (Teacher)</h1>
        <h2>Create Up to 5 Quizzes</h2>
        <button onclick="addQuiz()">Add Quiz</button>
        <div id="quiz-section"></div>
        <button onclick="saveQuizzes()">Save Quizzes</button>

         <div class="results-section">
             <h2>Student Results</h2>
            <button onclick="loadResults()">Show Results</button>
            <div id="resultsDisplay"></div>
        </div>
    </div>

    <script>
        let quizCount = 0;
        const maxQuizzes = 5;

        function addQuiz() {
            if (quizCount >= maxQuizzes) {
                alert("You can only create up to 5 quizzes.");
                return;
            }
    const quizDiv = document.createElement("div");
    quizDiv.classList.add("quiz-container");

    quizDiv.innerHTML = `
        <h3>Quiz ${quizCount + 1}</h3>
        <div class="form-group">
            <label>Question:</label>
            <input type="text" class="question" required>
        </div>
        <div class="form-group">
            <label>Options:</label>
            <input type="text" class="option1" placeholder="Option 1" required>
            <input type="text" class="option2" placeholder="Option 2" required>
            <input type="text" class="option3" placeholder="Option 3" required>
            <input type="text" class="option4" placeholder="Option 4" required>
        </div>
        <div class="form-group">
            <label>Correct Answer:</label>
            <input type="text" class="correct-answer" required>
        </div>
    `;
    document.getElementById("quiz-section").appendChild(quizDiv);
    quizCount++;
}

function saveQuizzes() {
    const quizzes = [];
    const containers = document.querySelectorAll(".quiz-container");

    for (let i = 0; i < containers.length; i++) {
        const div = containers[i];
        const question = div.querySelector(".question").value.trim();
        const option1 = div.querySelector(".option1").value.trim();
        const option2 = div.querySelector(".option2").value.trim();
        const option3 = div.querySelector(".option3").value.trim();
        const option4 = div.querySelector(".option4").value.trim();
        const correctAnswer = div.querySelector(".correct-answer").value.trim();

        if (!question || !option1 || !option2 || !option3 || !option4 || !correctAnswer) {
            alert(`Please fill all fields in Quiz ${i + 1}`);
            return;
        }

        quizzes.push({
            question,
            options: [option1, option2, option3, option4],
            correctAnswer
        });
    }

    // For now, just store in localStorage. Replace with AJAX to PHP later.
    localStorage.setItem("quizzes", JSON.stringify(quizzes));
    alert("Quizzes saved successfully!");
}

function loadResults() {
    const results = JSON.parse(localStorage.getItem("allResults")) || [];
    const container = document.getElementById("resultsDisplay");

    if (results.length === 0) {
        container.innerHTML = "<p>No results available.</p>";
        return;
    }

    let html = `
        <table>
            <tr>
                <th>Student Name</th>
                <th>Correct</th>
                <th>Incorrect</th>
            </tr>
    `;

    results.forEach(res => {
        html += `
            <tr>
                <td>${res.student_name}</td>
                <td>${res.correct}</td>
                <td>${res.incorrect}</td>
            </tr>
        `;
    });
```

```
            html += `</table>`;
            container.innerHTML = html;
        }
    </script>
</body>
</html>
```

RESULT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Your Quiz Result</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            padding: 30px;
            background-color: #f4f4f4;
        }
        .container {
            background: white;
            padding: 20px;
            max-width: 600px;
            margin: auto;
            border-radius: 10px;
            box-shadow: 0 0 10px gray;
        }
        canvas {
            margin-top: 30px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Your Quiz Result</h1>
        <div id="info"></div>
        <canvas id="resultChart" width="400" height="200"></canvas>
    </div>
```

```
  <script>
    const result = JSON.parse(localStorage.getItem("latestResult"));

    if (!result) {
      document.getElementById("info").innerHTML = "<p>No result found. Please take
a quiz first.</p>";
    } else {
      const totalQuestions = result.correct + result.incorrect;
      const scorePercent = ((result.correct / totalQuestions) * 100).toFixed(2);

      document.getElementById("info").innerHTML =
`<h3>${result.student_name}</h3>
        <p>Correct: ${result.correct}</p>
        <p>Incorrect: ${result.incorrect}</p>
        <p><strong>Score: ${scorePercent}%</strong></p>`;

      new Chart(document.getElementById("resultChart"), {
        type: 'bar',
        data: {
          labels: ["Correct", "Incorrect", "Score (%)"],
          datasets: [{
            label: "Quiz Performance",
            data: [result.correct, result.incorrect, scorePercent],
            backgroundColor: ['#28a745', '#dc3545', '#007bff']
          }]
        },
        options: {
          scales: {
            y: {
              beginAtZero: true,
              precision: 0
            }
          },
          plugins: {
            title: {
              display: true,
              text: "Your Quiz Performance"
            },
            legend: {
              display: false
            }
          }
```

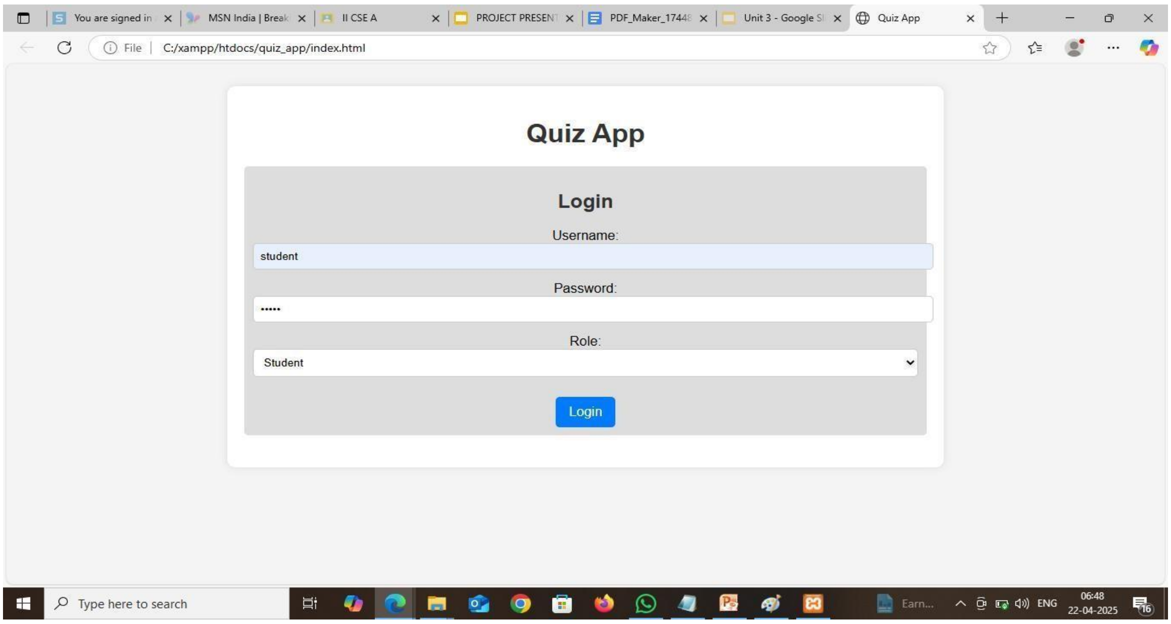
```
                }
            });
        }
    </script>
</body>
</html>
```

RESULT.PHP

```
<?php
$host = "localhost";
$db = "quiz_db";
$user = "root";
$pass = "";

$conn = new mysqli($host, $user, $pass, $db);
if ($conn->connect_error) die("Connection failed: " . $conn->connect_error);

// Fetch the most recent result
$sql = "SELECT student_name, correct, incorrect FROM results ORDER BY id DESC
LIMIT 1";
$result = $conn->query($sql);
$data = $result->fetch_assoc();
$conn->close();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Your Quiz Result</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            padding: 30px;
            background-color: #f4f4f4;
        }
        .container {
            background: white;
            padding: 20px;
            max-width: 600px;
            margin: auto;
            border-radius: 10px;
            box-shadow: 0 0 10px gray;
        }
        canvas {
            margin-top: 30px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Your Quiz Result</h1>
        <?php if ($data): ?>
            <h3><?php echo htmlspecialchars($data['student_name']); ?></h3>
            <p>Correct: <?php echo $data['correct']; ?></p>
            <p>Incorrect: <?php echo $data['incorrect']; ?></p>
            <canvas id="resultChart" width="400" height="200"></canvas>
        <?php else: ?>
            <p>No results found. Please take a quiz first.</p>
        <?php endif; ?>
    </div>

    <?php if ($data): ?>
    <script>
        new Chart(document.getElementById("resultChart"), {
            type: 'bar',
            data: {
                labels: ["Correct", "Incorrect"],
                datasets: [{
                    label: "Quiz Score",
                    data: [<?php echo $data['correct']; ?>, <?php echo $data['incorrect']; ?>],
                    backgroundColor: ['#28a745', '#dc3545']
                }]
            },
            options: {
                scales: {
                    y: {
                        beginAtZero: true,
                        precision: 0
                    }
},
                    plugins: {
                        title: {
                            display: true,
                            text: "Your Quiz Performance"
                        },
                        legend: {
                            display: false
                        }
                    }
                }
            });
        </script>
        <?php endif; ?>
    </body>
    </html>
```

APPENDIX B

1. INDEX



Quiz App

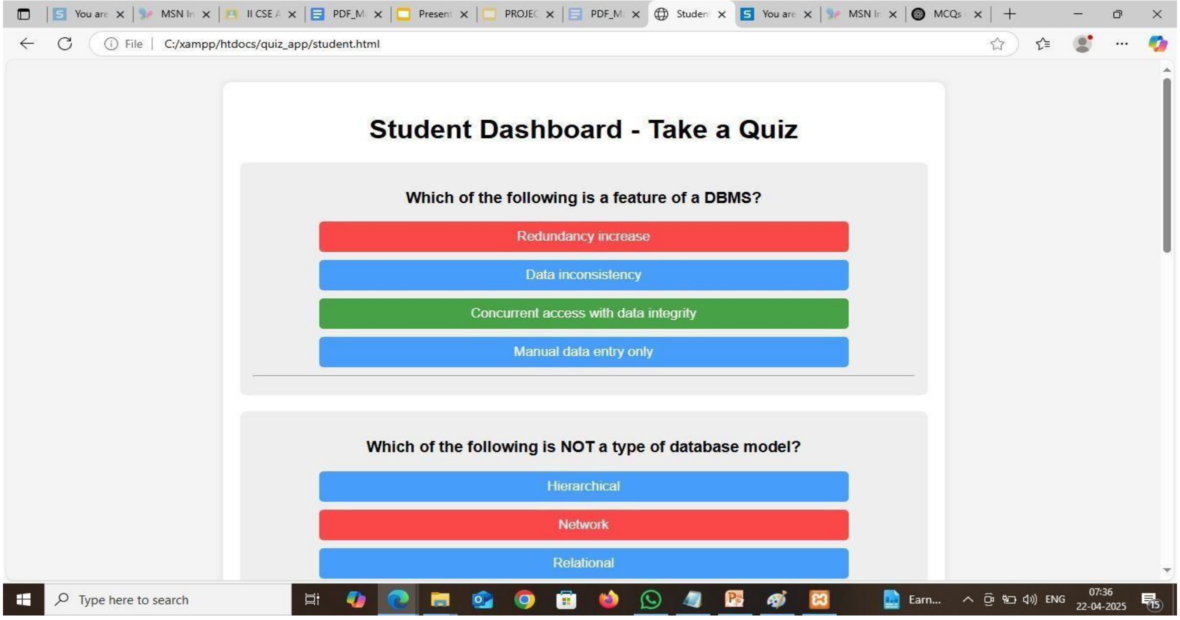
Login

Username: student

Password: •••••

Role: Student

2. STUDENT DASHBOARD



Student Dashboard - Take a Quiz

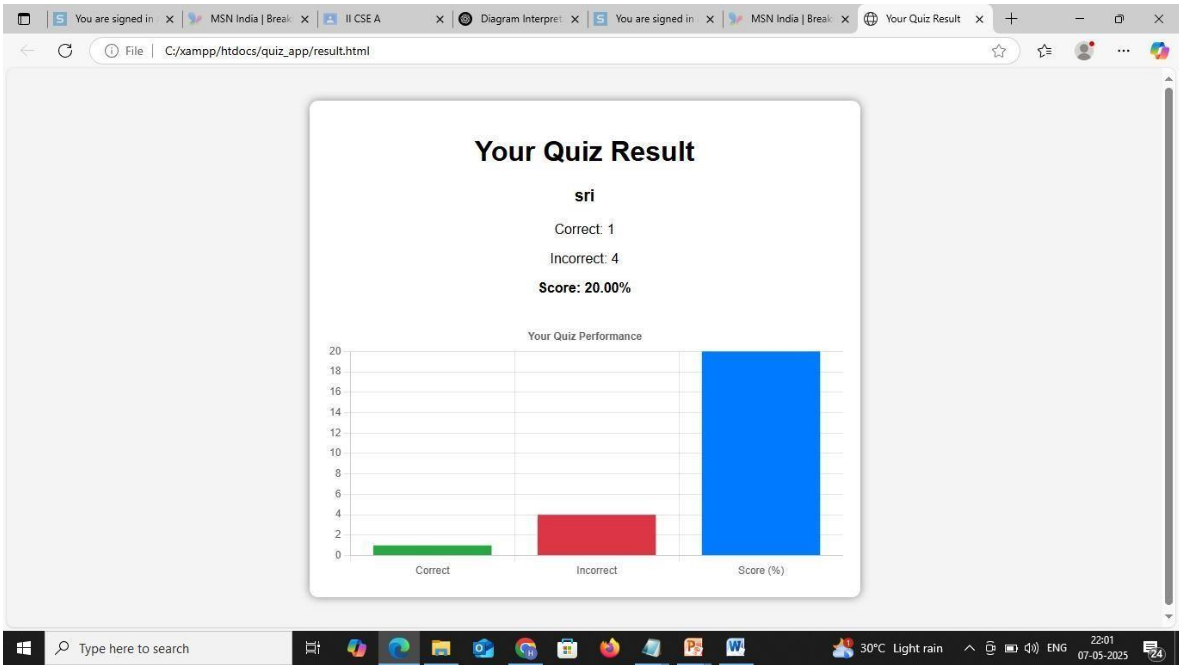
Which of the following is a feature of a DBMS?

- Redundancy increase
- Data inconsistency
- Concurrent access with data integrity
- Manual data entry only

Which of the following is NOT a type of database model?

- Hierarchical
- Network
- Relational

3. RESULT



Your Quiz Result

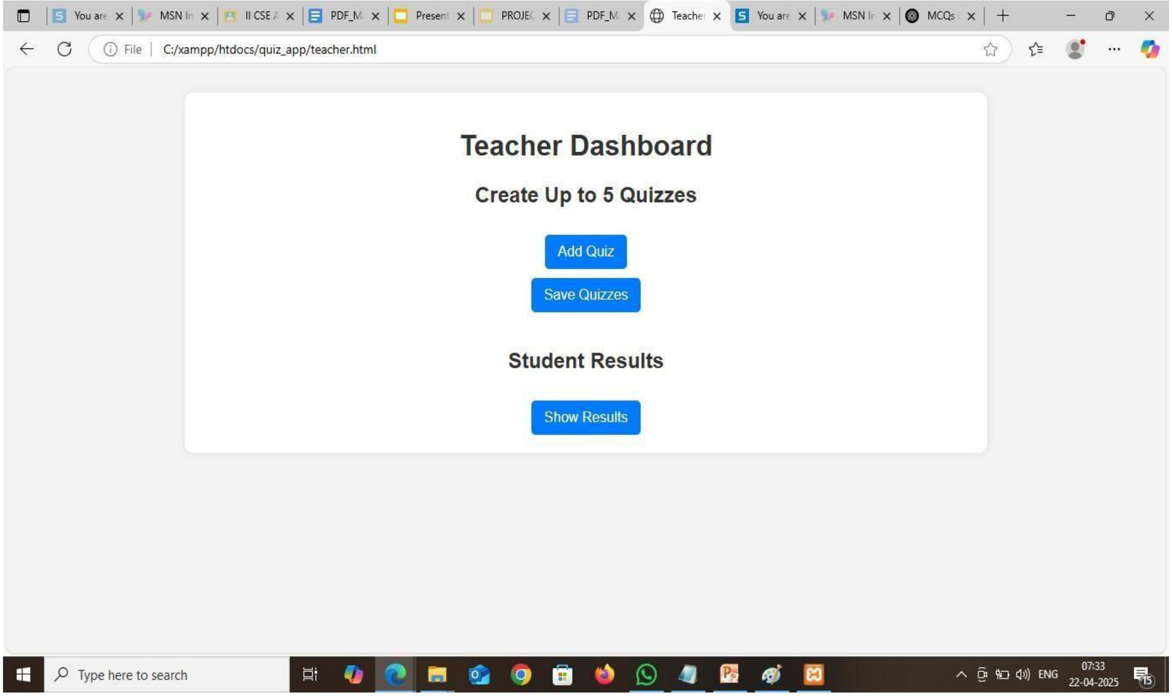
sri

Correct: 1

Incorrect: 4

Score: 20.00%

4. TEACHER DASHBOARD

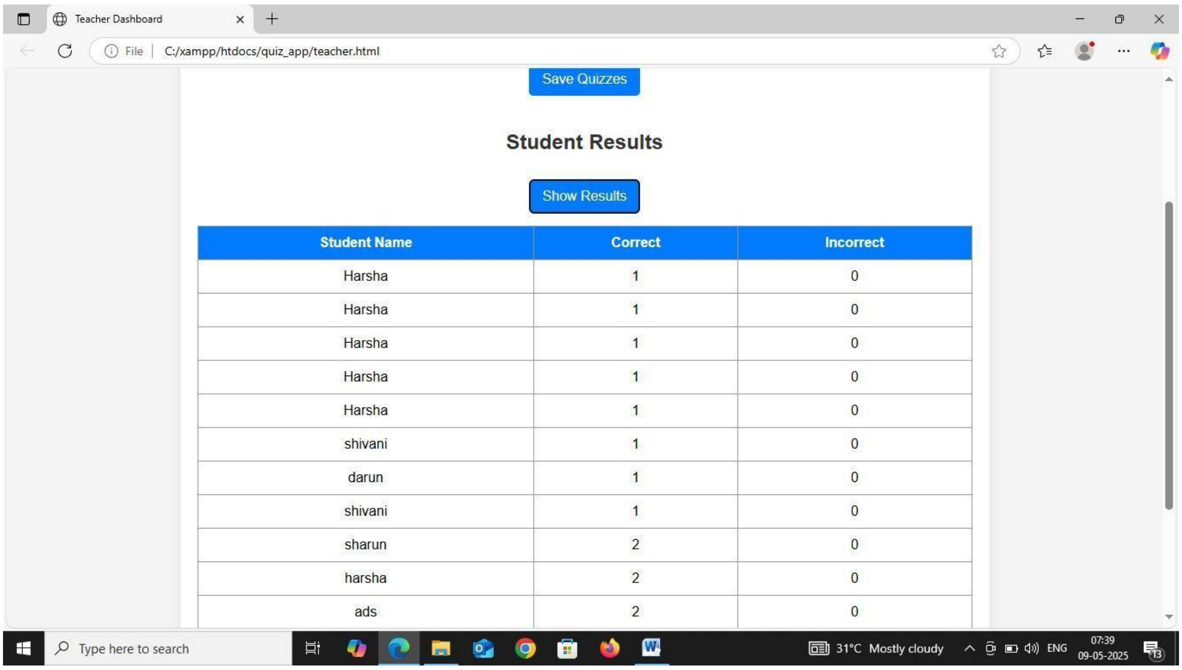


Teacher Dashboard

Create Up to 5 Quizzes

Student Results

5. TEACHER (ABLE TO SEE THE RESULT DASHBOARD



Student Results

Student Name	Correct	Incorrect
Harsha	1	0
Harsha	1	0
Harsha	1	0
Harsha	1	0
Harsha	1	0
shivani	1	0
darun	1	0
shivani	1	0
sharun	2	0
harsha	2	0
ads	2	0

REFERENCES

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts*
    a. (7th ed.). McGraw-Hill Education.
2. Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley.
3. Beighley, L., & Morrison, M. (2017). *Head First SQL*. O'Reilly Media.
4. Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
5. Mozilla Developer Network (MDN). (2024). *HTML and Web APIs Documentation*.
    a. Available at: https://developer.mozilla.org
6. PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*.
    a. Available at: https://www.postgresql.org/docs/
7. MySQL Documentation Team. (2024). *MySQL 8.0 Reference Manual*. Available at: https://dev.mysql.com/doc/
8. Socket.IO. (2024). *Real-time WebSocket Communication Library*. Available at: https://socket.io/docs/