**ONLINE EXAMINATION SYSTEM USING JSP**

**A Course End Project Report**

**WEB TECHNOLOGIES LABORATORY (****A8605)**

In partial fulfillment of the requirements for the

award of the degree of

**BATCHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**A. AKSHAYA SREE 22881A05K6**

**A.RAJ KUMAR 22881A05K7**

**B. SATHWIK SAI 22881A05K8**

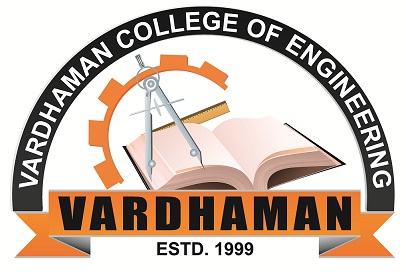
**B. VARSHA REDDY 22881A05K9**

Under the guidance of

**DR. GANESH DESHMUKH**

Assistant Professor

Department of Computer Science and Engineering



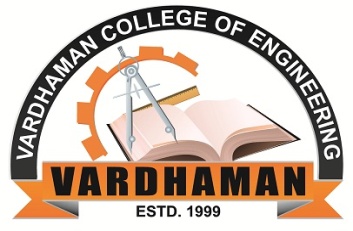
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VARDHAMAN COLLEGE OF ENGINEERING**

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India.

**NOVEMBER,2024**

****

**VARDHAMAN COLLEGE OF ENGINEERING**

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the Course End Project titled **“ONLINE EXAMINATION SYSTEM USING JSP”** is carried out by us towards **A8605–** **WEB TECHNOLOGIES LABORATORY** course and submitted to the **Department of Computer Science and Engineering**, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in the **Department of Computer Science and Engineering** during the Academic year 2024-25.

**Instructor: Head of the Department:**

**Dr. Ganesh Deshmukh, Dr. Ramesh Karnati,**

Assistant Professor, Head of the Department,

Dept of Computer Science and Dept. of Computer Science and

Engineering, Engineering,

Vardhaman College of Engineering, Vardhaman College of Engineering,

Hyderabad. Hyderabad.

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of the task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Mr. Dr. Ganesh Deshmukh**, Assistant Professor, Department of Computer Science and Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the design part of a potential project in time.

We are particularly thankful to **Dr. Ramesh Karnati**, Associate Professor & Head, Department of Computer Science and Engineering for his guidance, intense support, and encouragement, which helped us to mold our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R.Ravindra**, for having provided all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr. Teegala Vijender Reddy**, Chairman, and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.

A. AKSHAYA SREE 22881A05K6

A.RAJ KUMAR 22881A05K7

B. SATHWIK SAI 22881A05K8

B. VARSHA REDDY 22881A05K9

**Abstract**

The Online Examination System using JSP (Java Server Pages) is a web-based application designed to automate the process of conducting and managing exams. It allows administrators to create, manage, and schedule exams, while providing students with a secure platform to take exams online. The system supports features such as question bank management, real-time exam monitoring, automated grading, and result generation. JSP is used for dynamic web pages, ensuring efficient data handling and seamless interaction between users and the server. The system enhances the convenience of online assessments, reducing manual effort and improving accessibility for both students and examiners.

**PROBLEM STEMENT:**

The problem statement for an Online Examination System using JSP revolves around the need for a digital platform that simplifies and automates the process of conducting exams. Traditional paper-based exams are time-consuming, prone to human errors, and inefficient in terms of data management. This system aims to provide an efficient solution by enabling exam creation, scheduling, and real-time monitoring of test-taking. It also ensures automated grading, secure access, and quick result processing. The system should support user authentication, question bank management, and a user-friendly interface, while ensuring scalability, security, and reliability for both students and administrators.

**TECHNOLOGIES USED:**

**Frontend Technologies:**

1. **HTML**: Used for structuring web pages and content.
2. **CSS:** Provides styling and layout to the web pages, ensuring a visually appealing design.
3. **JavaScript:** For client-side scripting, including form validation, interactive features, and dynamic content updates.

**Backend Technologies:**

1. **Java: The** core programming language used for backend logic, including user management, exam scheduling, and result processing.
2. **JSP (Java Server Pages):** Used for generating dynamic content and presenting it to users based on data processed by Java.
3. **Servlets:** Handle HTTP requests, manage sessions, and perform logic like user authentication and exam submission.
4. **JDBC (Java Database Connectivity):** Used to interact with the database for storing and retrieving data like user profiles, exam questions, and results.
5. **MySQL/PostgreSQL:** Relational databases for storing and managing data related to users, exams, and results.
6. **Apache Tomcat:** A servlet container used for deploying and running JSP and Servlet-based web applications.

**Architecture:**

The architecture of an online examination system using JSP (Java Server Pages) typically follows a multi-tiered approach to ensure modularity, scalability, and security. It consists of the following layers:

1. **Presentation Layer:** The presentation layer, implemented with JSP and HTML/CSS, provides the user interface for students, administrators, and examiners, enabling login, exam interaction, and dynamic content generation based on user actions.
2. **Business Logic Layer:** The middle layer uses Java Servlets and JavaBeans to handle authentication, question management, answer evaluation, and result processing, ensuring efficient execution of examination workflows**.**
3. **Data Access Layer:** The data access layer uses JDBC to securely manage user profiles, question banks, exam schedules, and results by interacting with the database.
4. **Database Layer:** The database stores all system data, including user credentials, questions, answers, and exam results. Popular databases like MySQL or PostgreSQL are commonly used.
5. **Security:** Authentication, session management, and encryption mechanisms protect user data and exam integrity.

This architecture ensures a seamless, secure, and scalable online examination experience.

**System Workflow:**

An online examination system using JSP follows a structured workflow to ensure seamless operation:

1. **User Authentication:**

* Login Page: Students and administrators access the system via a JSP-based login interface.
* Authentication: Credentials are verified against a database using JSP servlets.

1. **Dashboard Access:**

* Student Dashboard: Displays available exams, results, and profile settings.
* Admin Dashboard: Allows creation, scheduling, and management of exams, questions, and user accounts.

1. **Exam Creation and Management (Admin):**

* Create Exam: Admin uses JSP forms to input exam details, upload questions (multiple-choice, essay), and set time limits.
* Schedule Exam: Define availability dates and times, ensuring synchronization with the database.

1. **Exam Enrolment and Notification (Student):**

* Enrol in Exam: Students select and register for available exams.
* Notifications: JSP-generated alerts inform students of upcoming exams.

1. **Taking the Exam:**

* Exam Interface: JSP pages present questions sequentially with timers.
* Answer Submission: Responses are captured via JSP forms and sent to the server for processing.

1. **Automatic Grading and Manual Review:**

* Auto-Grading: Objective questions are graded automatically using JSP logic.
* Manual Review: Subjective answers are reviewed by admins if necessary.

1. **Result Generation and Display:**

* Result Compilation: Scores are calculated and stored in the database.
* Result Page: Students view their performance through JSP-rendered result pages.

1. **Feedback and Reporting:**

* Feedback Forms: Students provide feedback via JSP forms.
* Reports: Admins generate performance reports using JSP and display analytics.

This JSP-based workflow ensures a dynamic, interactive, and efficient online examination environment, leveraging Java’s server-side capabilities for robust performance.

**Security:**

Securing an online examination system using JSP involves several key steps:

* **User Authentication:** Implement secure login mechanisms using hashed passwords and two-factor authentication to verify candidate identities.
* **Role-based Access Control:** Restrict access based on user roles (admin, examiner, candidate) to prevent unauthorized access.
* **Secure Data Transmission:** Use HTTPS for encrypted communication to protect sensitive data during transit.
* **Input Validation:** Prevent SQL injection and XSS attacks by sanitizing user inputs.
* **Session Management:** Implement secure session handling using tokens and timeout features to prevent hijacking.
* **Encryption:** Encrypt sensitive data like exam results and user credentials in the database.
* **Monitoring and Logging:** Track system activities to identify and respond to potential breaches.
* **Backup and Recovery:** Maintain regular backups and a disaster recovery plan for data security.

**Testing an Online Examination System Using JSP:**

1. **Requirement Analysis:** Identify the system requirements, such as user roles (students, administrators), functionality (login, exam, results), and test objectives.
2. **Environment Setup:** Configure the testing environment, including a Java-enabled server (e.g., Apache Tomcat) and database connection.
3. **Unit Testing:** Test individual modules like login validation, question retrieval, and answer submission using mock data.
4. **Integration Testing:** Verify data flow between modules, such as question selection and score calculation.
5. **Functional Testing:** Check core functionalities, such as user registration, exam scheduling, and result display.
6. **Performance Testing:** Measure system response under different loads.
7. **Security Testing:** Validate login authentication, data encryption, and protection against SQL injection.
8. **Bug Fixes and Retesting:** Resolve identified issues and retest.
9. **User Acceptance Testing (UAT):** Involve end-users to ensure the system meets expectations.
10. **Deployment and Maintenance:** Deploy the tested system and monitor for issues post-launch.

**Procedure:**

1. **User Authentication:**

Create a login page with JSP to validate users (students/admins) using a database.

1. **Dashboard Setup:**

Redirect authenticated users to a dashboard with roles (e.g., exam admin or student).

1. **Exam Management:**

Admins use JSP pages to add/edit/delete exam details, questions, and timing.

1. **Question Display**:

Dynamically load questions from the database using JSP and JavaBeans.

1. **Answer Submission:**

Collect answers via forms and store them in the database.

1. **Result Evaluation:**

Automatically calculate results using backend logic.

1. **Result Display:**

Show results with JSP and provide detailed feedback.

**SOURCE CODE**

**index.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" %>

<! DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Online Exam System</title>

<! -- Link to External CSS File -->

<link rel="stylesheet" type="text/css" href="css/styles.css">

</head>

<body>

<div class="container">

<h1>Welcome to the Online Examination System</h1>

<a href="quiz.jsp">Take Quiz</a>

<a href="admin/manage.jsp">Admin Panel</a>

</div>

</body>

</html>

**manage.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" %>

<! DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin Panel</title>

<! -- Link to External CSS File -->

<link rel="stylesheet" type="text/css" href="../css/styles.css">

</head>

<body>

<div class="admin-container">

<h1>Welcome to Admin Panel</h1>

<a href="addQuestion.jsp">Add Questions</a>

</div>

</body>

</html>

**addQuestion.jsp**

<%@ page import="db.DBConnection" %>

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8" %>

<! DOCTYPE html>

<html>

<head>

<title>Add Question</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 0;

}

h1 {

text-align: center;

color: #333;

}

form {

width: 50%;

margin: 20px auto;

padding: 20px;

background: #fff;

border-radius: 8px;

box-shadow: 0 2px 4px rgba (0, 0, 0, 0.1);

}

p {

margin: 10px 0;

}

input [type="text"] {

width: calc (100% - 20px);

padding: 8px;

margin: 5px 0;

border: 1px solid #ddd;

border-radius: 4px;

}

button {

background-color: #4CAF50;

color: white;

border: none;

padding: 10px 20px;

cursor: pointer;

border-radius: 5px;

font-size: 16px;

}

button: hover {

background-color: #45a049;

}

a {

display: inline-block;

margin-top: 15px;

text-decoration: none;

color: #007BFF;

font-size: 14px;

}

a: hover {

text-decoration: underline;

}

.success-message {

text-align: center;

margin-top: 20px;

color: green;

}

</style>

</head>

<body>

<h1>Add a New Question</h1>

<form method="post">

<p>

Question: <input type="text" name="question">

</p>

<p>

Option 1: <input type="text" name="option1"><br>

Option 2: <input type="text" name="option2"><br>

Option 3: <input type="text" name="option3"><br>

Option 4: <input type="text" name="option4"><br>

</p>

<p>

Correct Option: <input type="text" name="correct\_option">

</p>

<button type="submit">Add Question</button>

</form>

<%

if (request.getMethod ().equalsIgnoreCase ("POST")) {

Connection conn = db.DBConnection.getConnection ();

PreparedStatement pstmt = conn.prepareStatement (

"INSERT INTO questions (question, option1, option2, option3, option4, correct\_option) VALUES (?,?,?,?,?,?)");

pstmt.setString (1, request.getParameter ("question"));

pstmt.setString (2, request.getParameter ("option1"));

pstmt.setString (3, request.getParameter ("option2"));

pstmt.setString (4, request.getParameter ("option3"));

pstmt.setString (5, request.getParameter ("option4"));

pstmt.setInt (6, Integer.parseInt (request.getParameter ("correct\_option")));

pstmt.executeUpdate ();

conn.close ();

%>

<div class="success-message">Question added successfully!</div>

<%

}

%>

<a href="manage.jsp">Back to Admin Panel</a>

</body>

</html>

**quiz.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>

<%@ page import="java.sql.\*" %>

<!DOCTYPE html>

<html>

<head>

<title>Quiz</title>

<style>

body {

font-family: 'Arial', sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

min-height: 100vh;

}

h1 {

color: #333;

font-weight: normal;

margin-bottom: 20px;

text-align: center;

}

form {

background-color: #fff;

padding: 20px 30px;

border: 1px solid #ddd;

border-radius: 5px;

width: 100%;

max-width: 500px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

p {

margin-bottom: 15px;

font-size: 1em;

color: #555;

}

input[type="radio"] {

margin-right: 8px;

}

button {

display: block;

width: 100%;

background-color: #007BFF;

color: #fff;

border: none;

padding: 10px;

border-radius: 5px;

font-size: 1em;

cursor: pointer;

margin-top: 20px;

transition: background-color 0.3s;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<h1>Take the Quiz</h1>

<form action="result.jsp" method="post">

<%

// Database connection

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

try {

// Load the JDBC driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Establish connection (update DB URL, user, and password)

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/onlineexam", "root", "aimer");

// Fetch questions

stmt = conn.createStatement();

rs = stmt.executeQuery("SELECT \* FROM questions");

// Dynamically render questions

int questionNumber = 1;

while (rs.next()) {

%>

<p>

<%= questionNumber++ %>. <%= rs.getString("question") %><br>

<input type="radio" name="q<%= rs.getInt("id") %>" value="1"> <%= rs.getString("option1") %><br>

<input type="radio" name="q<%= rs.getInt("id") %>" value="2"> <%= rs.getString("option2") %><br>

<input type="radio" name="q<%= rs.getInt("id") %>" value="3"> <%= rs.getString("option3") %><br>

<input type="radio" name="q<%= rs.getInt("id") %>" value="4"> <%= rs.getString("option4") %><br>

</p>

<%

}

} catch (Exception e) {

out.println("<p>Error loading questions: " + e.getMessage() + "</p>");

} finally {

// Close resources

if (rs != null) rs.close();

if (stmt != null) stmt.close();

if (conn != null) conn.close();

}

%>

<button type="submit">Submit</button>

</form>

</body>

</html>

**result.jsp**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=UTF-8" %>

<!DOCTYPE html>

<html>

<head>

<title>Result</title>

<style>

body {

font-family: 'Arial', sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

min-height: 100vh;

}

h1 {

color: #333;

font-weight: normal;

margin-bottom: 20px;

}

p {

font-size: 1.2em;

color: #555;

}

a {

display: inline-block;

margin-top: 20px;

text-decoration: none;

background-color: #007BFF;

color: #fff;

padding: 10px 20px;

border-radius: 5px;

transition: background-color 0.3s;

}

a:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<h1>Your Results</h1>

<%

// Initialize the score variable

int score = 0;

// Fetch the user's answers and compare with the correct answers stored in the database

Connection conn = db.DBConnection.getConnection();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM questions");

// Iterate through each question

while (rs.next()) {

int questionId = rs.getInt("id");

int correctAnswer = rs.getInt("correct\_option");

// Get the user's answer for the current question

String userAnswer = request.getParameter("q" + questionId);

// If the user answered the question, compare the answer with the correct one

if (userAnswer != null && Integer.parseInt(userAnswer) == correctAnswer) {

score++;

}

}

conn.close();

%>

<p>Your Score: <%= score %> out of 4</p>

<%

if (score == 4) {

%>

<p>Excellent! You scored 100%!</p>

<%

} else if (score >= 3) {

%>

<p>Great job! You scored <%= score %> out of 4!</p>

<%

} else {

%>

<p>Good effort! You scored <%= score %> out of 4. Keep practicing!</p>

<%

}

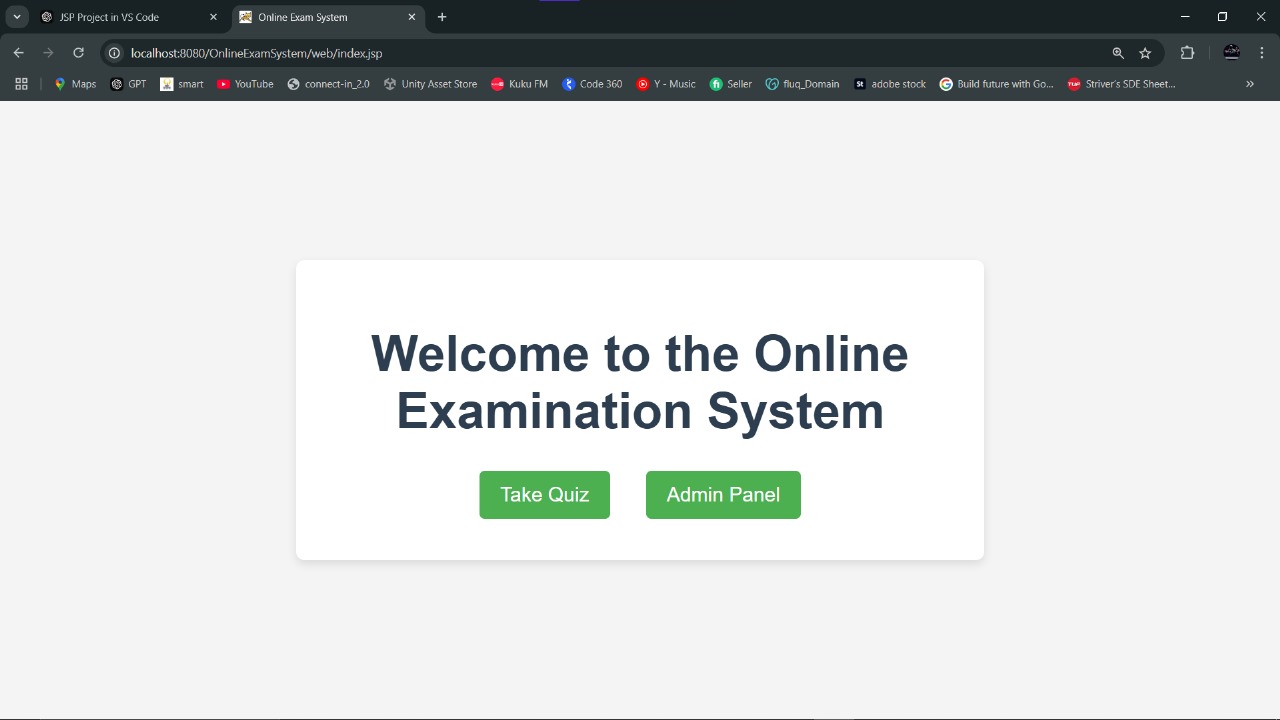
%>

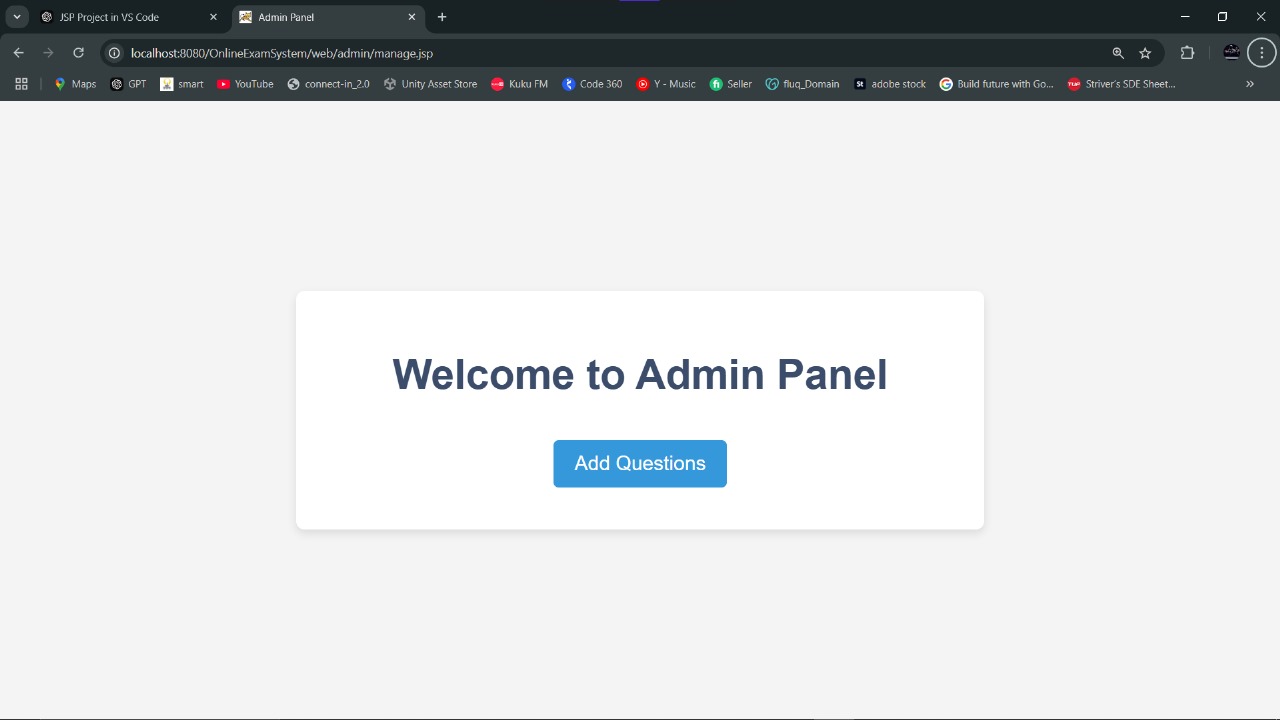
<a href="index.jsp">Go Home</a>

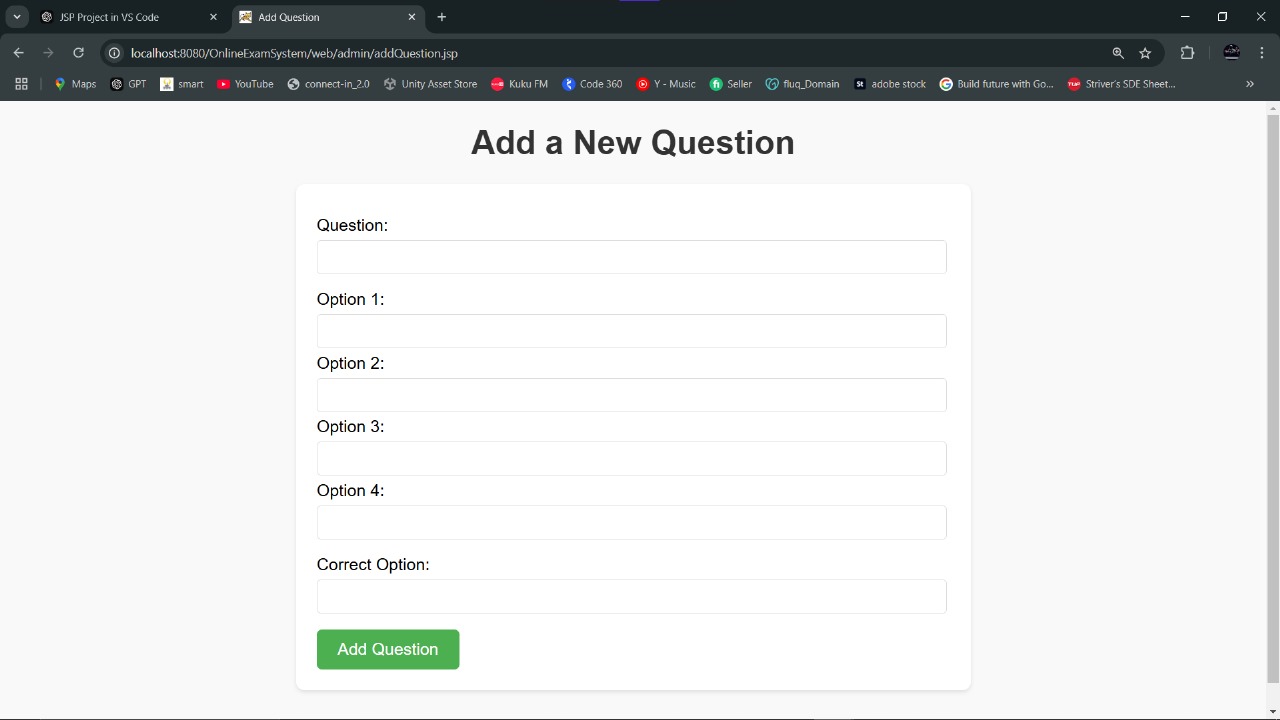
</body>

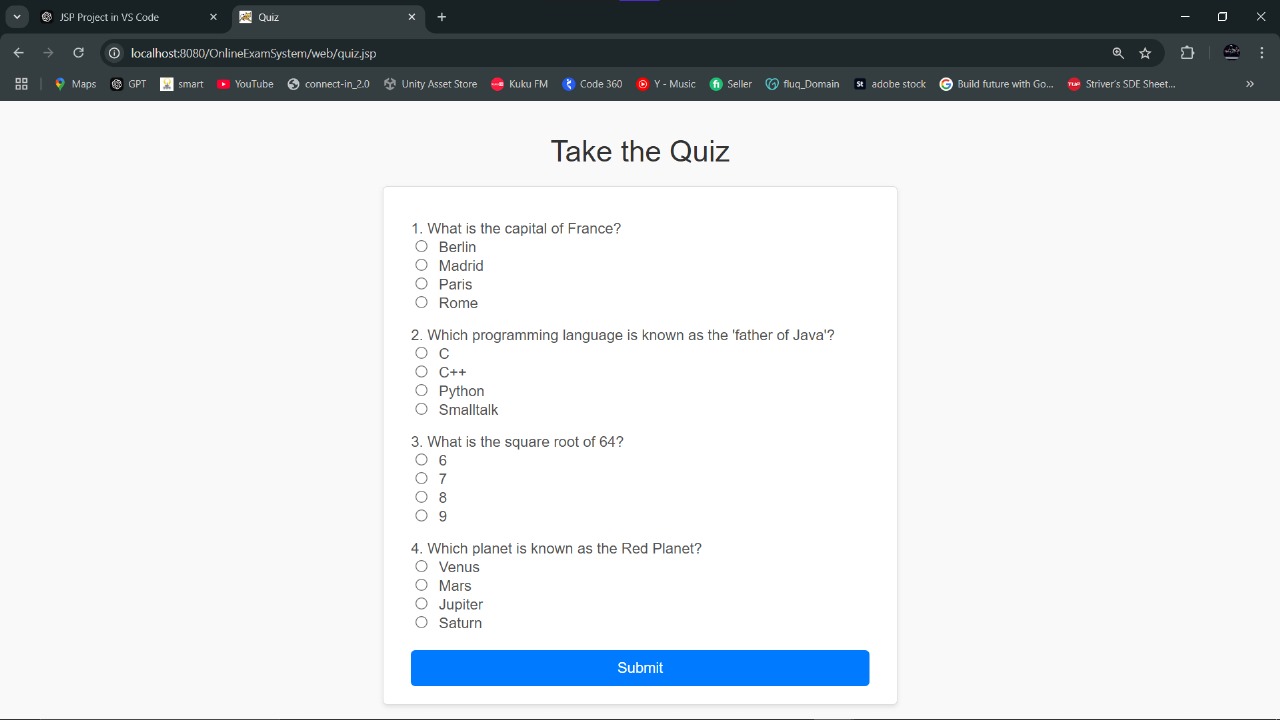
</html>

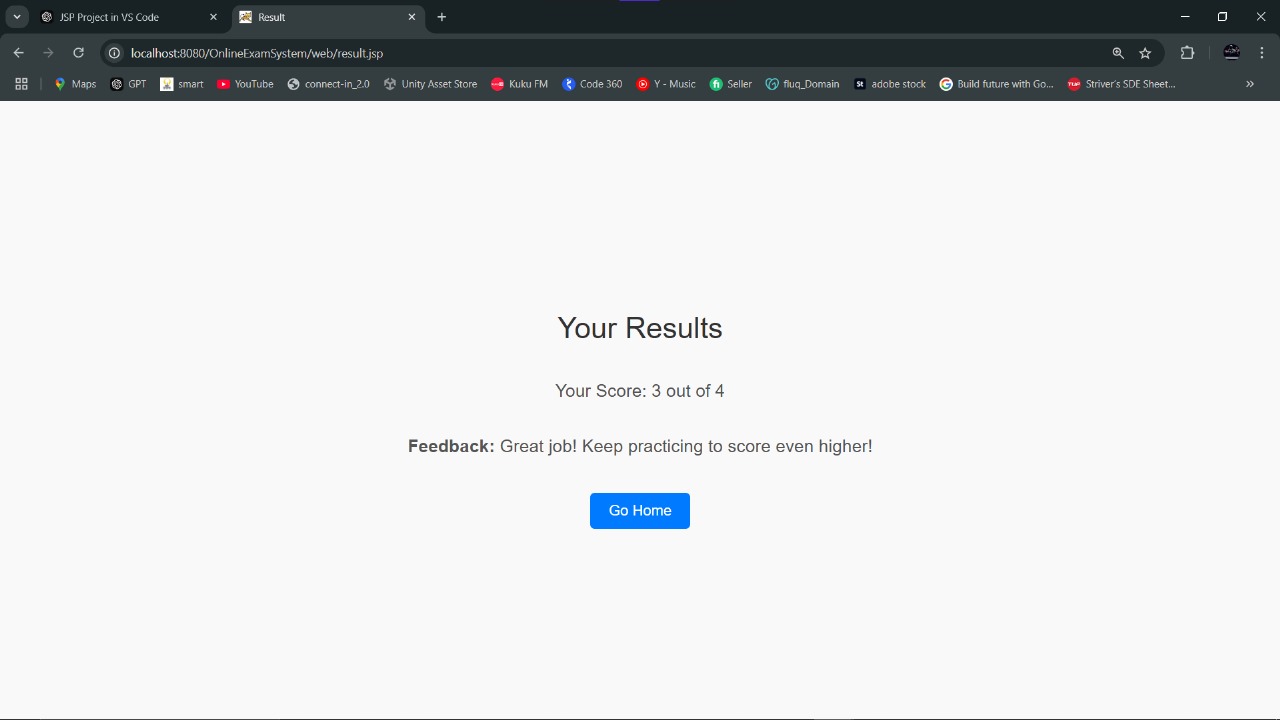
**RESULT:**

****









**GITHUB LINK:**

<https://github.com/Akhil05000/OnlineTicketBookingSystem>

**CONCLUSION:**

The implementation of an online examination system using Java Server Pages (JSP) demonstrates an efficient, scalable, and user-friendly solution for modern education and assessment needs. This system addresses key challenges such as accessibility, convenience, and resource optimization, offering significant advantages over traditional pen-and-paper methods.

By leveraging JSP and related technologies, the system provides a dynamic and interactive web-based platform that facilitates seamless interactions between administrators, educators, and students. It ensures the secure and automated management of question banks, user authentication, examination scheduling, and real-time grading, which reduces manual intervention and minimizes errors. Additionally, the use of JSP enables robust integration with databases, ensuring reliable data storage and retrieval while supporting large-scale operations.

The system promotes inclusivity by enabling remote access to exams, thus benefiting students in diverse locations. Features like time tracking, instant feedback, and detailed analytics further enhance the user experience and provide valuable insights for educators to improve learning outcomes.

In conclusion, the online examination system built with JSP is a cost-effective, adaptable, and sustainable solution. It not only modernizes the examination process but also aligns with the growing need for digital transformation in education, ensuring a seamless transition to the future of learning and assessment.