

# JAVA BASED ONLINE QUIZ PLATFORM

PROBLEM UNDERSTANDING - The Quiz Platform allows:

Admins to approve/reject quizzes.

Quiz creators to create quizzes and add questions. –

Participants to take quizzes and view their results.

Key problems solved: -

Managing quizzes online.

Assigning quizzes to users.

Evaluating participant results.

## 1. Project Definition

This project is a **Web-Based Online Quiz Platform** developed using

**Java, JSP, Servlets, JDBC, MySQL, Bootstrap, and MVC architecture.**

It allows Admins, Quiz Creators, and Participants to manage, create, and attempt quizzes online.

---

## 2. Purpose of the System

The system automates quiz creation and evaluation.

It replaces manual test creation with a **fast, efficient, user-friendly** online platform.

---

## 3. Main Objectives

- Provide **role-based access control** (Admin, Creator, Participant).
  - Allow quiz creators to easily create quizzes and add questions.
  - Allow participants to take quizzes and view results.
  - Allow admins to manage users and approve quizzes.
  - Provide a modern, clean UI using Bootstrap.
- 

## 4. Overall Workflow

1. User logs in using email & password.
  2. Server checks their role from the database.
  3. Based on the role → Redirect to correct dashboard.
  4. Each user performs role-specific actions.
-

## 5. Technologies Used

- **Frontend:** JSP, HTML, CSS, Bootstrap
  - **Backend:** Java Servlets, JDBC
  - **Database:** MySQL
  - **Tools:** NetBeans IDE, Apache Tomcat Server
  - **Architecture:** MVC (Models, Views, Controllers)
- 

## 6. MVC Architecture Usage

- **Model:** Java classes like *User, Quiz, Question*
  - **View:** JSP pages like *login.jsp, dashboard.jsp, take-quiz.jsp*
  - **Controller:** Servlets such as *LoginServlet, CreateQuizServlet*
- 

## 7. Database Integration (JDBC)

The system uses JDBC to:

- Insert quizzes
- Store questions
- Validate login
- Fetch quizzes for participants
- Approve/reject quizzes

Your **DBConnection.java** class manages MySQL connectivity using a config.properties file.

---

## 8. User Roles in the System

The system has 3 major user roles:

### ✓ Admin

- Manages users
- Approves or rejects quizzes
- Sees reports

### ✓ Quiz Creator

- Creates quizzes
- Adds questions
- Reviews submissions

### ✓ Participant

- Views available quizzes
  - Takes quizzes
  - Sees results
- 

## 9. User Authentication

LoginServlet handles:

- Fetching user from DB
  - Checking password
  - Creating a session
  - Redirecting based on role
- 

## 10. Session Management

Session tracking ensures users cannot:

- Open admin pages without logging in
- Access pages of another role
- Access system after logout

Filters enforce access control.

---

## 11. Role-Based Filters

Three filters:

- **AdminFilter** → Protects admin pages
- **CreatorFilter** → Protects creator pages
- **ParticipantFilter** → Protects participant pages

These ensure **security and access control**.

---

## **12. Quiz Creation Process**

1. Creator enters quiz title + description.
  2. CreateQuizServlet inserts quiz into DB.
  3. Creator is redirected to add-question.jsp.
  4. Add-Question form allows adding unlimited questions.
- 

## **13. Adding Questions**

Each question includes:

- Question text
- Four options
- Correct answer

Stored inside questions table linked by quiz\_id.

---

## **14. Quiz Approval Process**

Admin views all quizzes in *approve-quiz.jsp*.

Admin can:

- Accept quiz (ApproveQuizServlet)
- Reject quiz (RejectQuizServlet)

Only approved quizzes show for participants.

---

## **15. Participant Quiz Flow**

1. Participant logs in
  2. Sees list of available approved quizzes
  3. Opens a quiz
  4. Questions display one after another
  5. Participant selects correct answers
  6. System calculates score and stores results
-

## **16. Result Calculation**

Performed in **SubmitQuizServlet**:

- Fetch correct answers
  - Compare with participant answers
  - Calculate total marks
  - Store result in results table
- 

## **17. Bootstrap UI Integration**

All JSP pages use:

- Navbar
- Cards
- Buttons
- Tables

UI is modern, clean, and responsive.

---

## **18. Important Servlet Files (Backbone of Project)**

Essential servlets include:

- LoginServlet
- LogoutServlet
- CreateQuizServlet
- AddQuestionServlet
- TakeQuizServlet
- SubmitQuizServlet
- ApproveQuizServlet
- RejectQuizServlet
- ViewQuizServlet
- ResultServlet

These control all major actions.

---

## **19. Folder Structure Explanation**

Your project maintains:

- **model** → Java classes
- **dao** → Database handling
- **controller** → Servlets
- **filters** → Security filters
- **Web Pages** → All JSP files
- **resources** → config.properties

This ensures a clean maintainable project.

---

## **20. Final Outcome**

The final system:

Is fully functional

Supports all 3 roles

Has secure authentication

Uses filters for role checks

Allows full quiz creation & evaluation

Has a modern UI

Works as a real professional web application

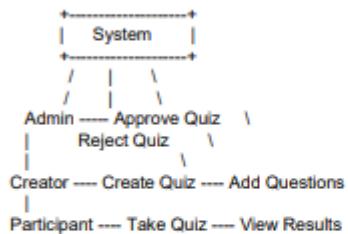
## 2. USE CASE DIAGRAM

---

Actors:

- Admin
- Quiz Creator
- Participant
- System

Use Case Diagram (ASCII Representation):



## 3. SEQUENCE DIAGRAM (CREATE QUIZ)

---

Creator → System: Open create-quiz.jsp  
Creator → System: Submit quiz details  
System → DB: INSERT quiz  
DB → System: quizId returned  
System → Creator: Redirect to add-question.jsp?quizId=...

## 4. ER DIAGRAM (ASCII)

---

```
+-----+ +-----+ +-----+  
| quizzes | 1 --- n | questions | n --- 1 | options |  
+-----+ +-----+ +-----+
```

```
+-----+ +-----+ +-----+  
| quiz_id PK| | question_id PK| | option_id PK|  
| title | | quiz_id FK | | question_id FK |  
| desc | | question_text | | option_text |  
+-----+ +-----+ +-----+
```