ONLINE QUIZ PORTAL WRITEUP

USER REGISTRATION: - Implement the `User` entity class with attributes like name, email, and password. - Create the `UserDAO` class to handle user-related database operations (e.g., register,

login, retrieve user details). - Develop the registration form (`userregistration.jsp`) that collects user information such as

name, email, password, and date of birth. - Implement the `RegisterServlet` to handle user registration, which calls the appropriate

methods from `UserDAO` to store the user details in the database.

USER LOGIN: - Create the login form (`userlogin.jsp`) that collects user credentials (email and password). - Implement the `UserLoginServlet` to handle user login, which calls the `validateUser`

method from `UserDAO` to check if the provided email and password are valid. - If the credentials are valid, create a session for the user and redirect them to the user

dashboard (`userdashboard.jsp`). - If the credentials are invalid, display an error message and redirect the user back to the

login page.

USER DASHBOARD: - Implement the `UserDashboardServlet` to handle requests to the user dashboard. - Retrieve the user details from the session and pass them to the user dashboard page

(`userdashboard.jsp`). - Display a welcome message to the user along with any relevant information or actions

available in the user dashboard.

VIEW QUIZZES: - Retrieve the list of available quizzes from the `QuizDAO` in the user dashboard servlet.

- Pass the list of quizzes to the user dashboard page (`userdashboard.jsp`) for display. - Display the quiz details (e.g., title, description) along with a link or button for each quiz

that redirects the user to the quiz details page.

QUIZ DETAILS AND SUBMISSION: - Implement the `QuizDetailsServlet` to handle requests for viewing quiz details. - Retrieve the quiz details, including questions and options, from the `QuizDAO` and pass

them to the quiz details page (`quizdetails.jsp`). - Display the questions and options to the user, along with a form for submitting their

answers. - Implement the `SubmitQuizServlet` to handle quiz submission. - Retrieve the submitted answers from the request parameters and compare them with the

correct answers stored in the database. - Calculate the user's score and store the quiz result in the database. - Redirect the user to the result page (`result.jsp`) to display their score and provide

feedback on their answers.

ADMIN LOGIN: - Create the admin login form (`adminlogin.jsp`) that collects admin credentials (email and

password). - Implement the `AdminLoginServlet` to handle admin login, which calls the

`validateAdmin` method from `AdminDAO` to check if the provided email and password are

valid. - If the credentials are valid, create a session for the admin and redirect them to the admin

dashboard (`admindashboardpage.jsp`). - If the credentials are invalid, display an error message and redirect the admin back to the

login page.

ADMIN DASHBOARD: - Implement the `AdminDashboardServlet` to handle requests to the admin dashboard. - Retrieve the admin details from the session and pass them to the admin dashboard page

(`admindashboardpage.jsp`).

- Display a welcome message to the admin along with any relevant information or actions available in the admin dashboard.

MANAGE QUIZZES: - Implement the `AddSQLQuestionServlet

` to handle requests for adding SQL questions to the quiz. - Create the form (`addquestionsql.jsp`) to collect the SQL question details (e.g., question

text, options, correct answer) and submit them to the servlet. - Implement the `ViewSQLQuestionServlet` to retrieve and display a list of SQL questions

in the admin dashboard. - Create the page (`viewsqlquestions.jsp`) to display the list of SQL questions and provide

options to edit or delete each question. - Implement the `EditSQLQuestionServlet` to handle requests for editing an existing SQL

question. - Create the form (`editsqlquestion.jsp`) to display the details of an SQL question and allow

the admin to modify them. - Implement the `DeleteSQLQuestionServlet` to handle requests for deleting an SQL

question from the database.

MANAGE USERS: - Implement the `EditUserServlet` to handle requests for editing user details. - Create the form (`edituser.jsp`) to display the details of a user and allow the admin to

modify them. - Implement the `DeleteUserServlet` to handle requests for deleting a user from the

database.

LOGOUT: - Implement the `LogoutServlet` to handle user and admin logout requests. - Invalidate the session and redirect the user or admin to the logout page (`loggedout.jsp`).