PROJECT

Done : Navaneethan.R

1. Java-Based Student Management System : A MVC Approach

Description

- Develop a Student Information System using Java technologies (JDBC,Servlets, JSP) following the MVC design pattern. The system will provide a user-friendly interface for student login, data entry,storage in a JDBC database, and display of student details in a grid format.

Key Features

1. Login Page:
    - Create a simple login page for students to enter their credentials (e.g., username and password).
    - Implement authentication to ensure only authorized users can access the system.

2. Student Details Form (JSP):
    - Design a JSP page with a form to capture student details, including fields such as Name, Roll Number, Course, and Grade.
    - Ensure proper validation for the form inputs to maintain data integrity.

3. Servlets (Controller):
    - Implement a servlet acting as the controller to handle requests from the JSP pages.
    - Use the servlet to process login requests, validate user credentials.

4. DAO Layer (Data Access Object):
    - Develop a DAO layer to interact with the database.
    - Include methods to insert student details into the database and retrieve student information for display.

5. JDBC Database:
    - Set up a relational database (e.g., MySQL) to store student information.
    - Create a table to store student details with appropriate columns (e.g., ID, Name, Roll Number, Course, Grade).

6. Insertion of Student Details:
    - Upon form submission, use the servlet to capture the entered data.
    - Call the DAO layer to insert the student details into the database.

7. Display Students in Grid Format (JSP):
    - Create a JSP page to display student information in a grid format.

8. MVC Design Pattern:
    - Model: Represent the student data structure and database interactions in the DAO layer.
    - View: Use JSP pages for the user interface.
    - Controller: Implement servlets to manage the flow of data between the view and the model.

9. Exception Handling:
    - Implement proper exception handling throughout the application to manage errors gracefully.
    - Display meaningful error messages to users in case of any issues.