

USER MANAGEMENT SYSTEM

Gayathri Ramadurgum

1NH21CS271

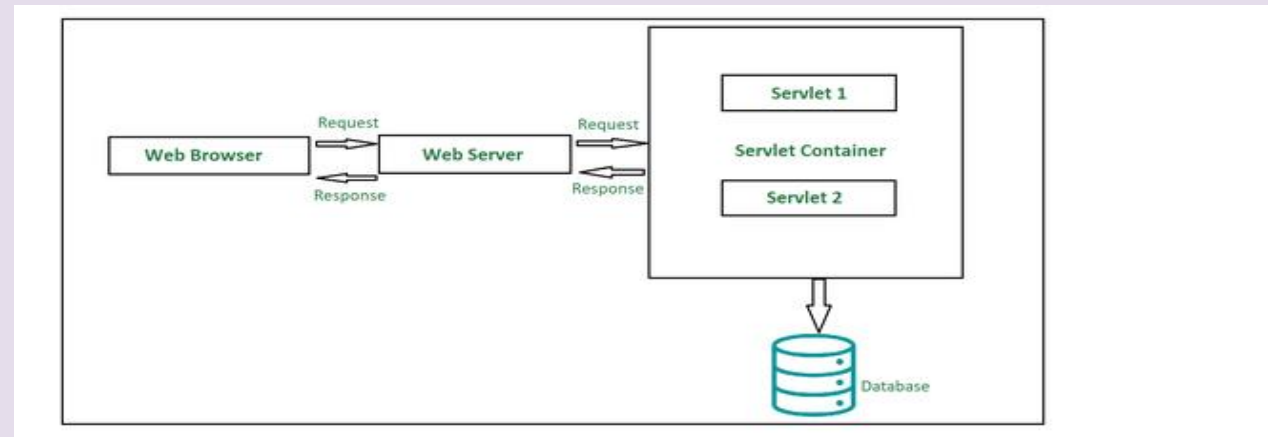
CSE 5E

Project Guide : Ms. Kavitha.U, Assistant Professor, CSE Department



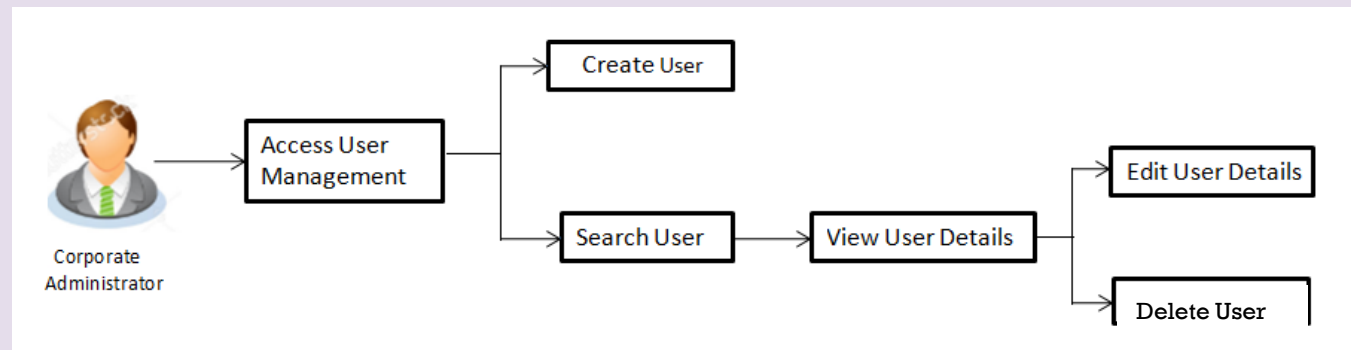
AIM OF THE PROJECT

- The User Management System is a web-based application developed using technologies like Servlets, Java Server Pages (JSP), and JDBC
- The User Management System offers a user-friendly interface accessible via any web browser, allowing administrators to perform various user management tasks such as user registration, profile management, and communication, while enabling users to access their information securely and receive important updates from administrators.



IMPORTANCE OF THE PROJECT

- The User Management System project serves as a crucial tool for organizations, providing a centralized platform for administrators to efficiently manage employee profiles and circulate important information.
- The system facilitates effective communication within the organization.
- Administrators can create and share important notes through the platform, providing a unified space for crucial announcements, policy updates, or any information pertinent to the employees.



OBJECTIVES

1. **Authentication and Authorization:** Establish role-based authorization, distinguishing between administrators and regular employees.
2. **Database Integration:** Utilize JDBC to seamlessly integrate the system with a MySQL database for efficient storage and retrieval of user data.
3. **User Profile Management:** Enable administrators to perform CRUD (Create, Read, Update, and Delete) operations on user profiles, ensuring accurate and up-to-date employee information.
4. **User-Friendly Interface:** Design an intuitive and user-friendly interface for both administrators and regular users to enhance overall user experience.



SOFTWARES USED

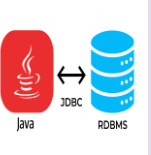
1. Servlets

- They are server-side Java components that handle HTTP requests and generate dynamic responses. Servlets are responsible for processing user input from the registration form. They capture user data, validate it, and perform necessary actions such as storing data in the database or retrieving user information.



2. JDBC

- A Java API that allows Java applications to interact with relational databases like MySQL. JDBC is used to establish a connection between your Java application (Servlets) and the MySQL database. It provides the necessary libraries and methods to connect, query, and manipulate the database.



3. Apache Tomcat

- A widely used open-source application server that implements the Java Servlet in this project. Apache Tomcat is responsible for hosting and running your Java web application. It handles the execution of Servlets when HTTP requests are made to your web application.



4. JSP - Java Server Pages

- It is a server-side technology which is used for creating web applications. It is used to create dynamic web content. JSP consists of both HTML tags and JSP tags. In this, JSP tags are used to insert JAVA code into HTML pages.



IMPLEMENTATION

Initial Page Load - index.html and login.jsp:

- When a user accesses the system, the initial request is sent to retrieve the login.jsp page.
- login.jsp presents a login form where users input their credentials, such as username and password.
- Upon submitting the login form, the request is sent to validate the user's credentials.

User Authentication - LoginServlet:

- The submitted credentials are processed by the LoginServlet.
- LoginServlet verifies the user's credentials by querying the MySQL database using JDBC.
- If the credentials are valid, the user is authenticated, and access to the system is granted.

User Registration - RegisterServlet:

- After successful authentication of the admin, they can access the registration form in index.html.
- Admins input employee's personal information, including name, email, mobile number, date of birth, city, and gender.
- Clicking the "Register" button submits the registration details to the RegisterServlet.



Displaying Registered Users - ShowDataServlet:

- After successful registration or at any time, admins can click the "Show Users" button in index.html to view registered users.
- The "Show Users" button triggers an HTTP request to the ShowDataServlet.
- Inside ShowDataServlet, a query using JDBC retrieves a list of all previously registered users from the MySQL database.
- The retrieved user data is formatted into an HTML table dynamically and sent as a response to the user's browser for display.

User Editing - EditServlet:

- On the "Show Users" page, admins can click on the "Edit" link associated with a specific user.
- Clicking "Edit" invokes the EditServlet.
- Inside EditServlet, the user's current information is retrieved from the database based on the user's ID.
- Admins can make changes to their data, such as updating their email, password, contact information, or other profile attributes.
- EditServlet securely updates the user's information in the MySQL database using JDBC.



User Deletion - DeleteServlet:

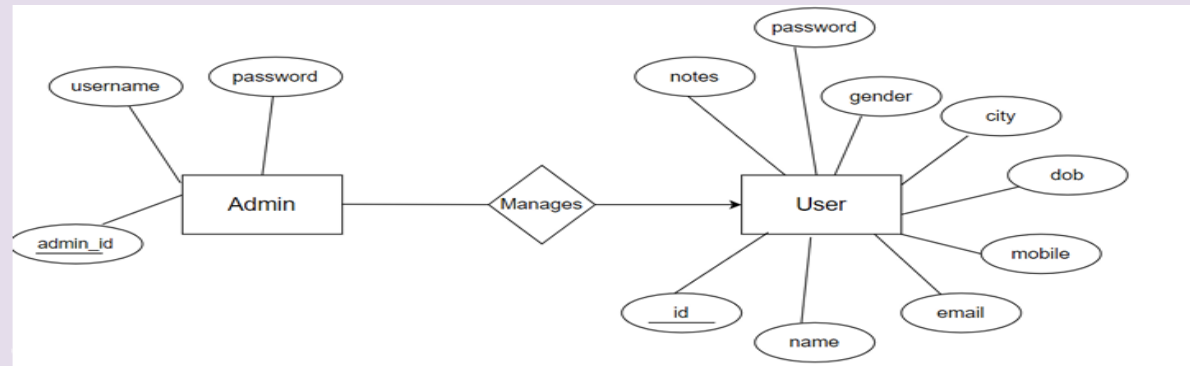
- On the "Show Users" page, users can click on the "Delete" link associated with a specific user.
- Clicking "Delete" invokes the DeleteServlet.
- Inside DeleteServlet, the user's account is deleted securely from the MySQL database using JDBC.

User Access – loginServlet:

- If the user radio button is selected, then it means a regular user is trying to login
- Upon successful authentication as a regular user, users are redirected to their dashboard. The loginServlet is invoked when credentials of the user are entered
- The user dashboard provides personalized information and features, including viewing notes or communications from the admin.



DATABASE-MYSQL



- In the User Management System's database, there are two essential tables: the Admin table and the User table.
- The Admin table is responsible for storing administrator credentials, featuring fields for username and password. This table ensures secure access to the system, allowing only authorized administrators to manage user data.
- On the other hand, the User table serves as the repository for all user-related information. This includes details such as name, email, mobile number, date of birth, city, and gender.
- Each user's entry in this table is uniquely identified by a primary key, typically the user ID, which ensures data integrity and facilitates efficient retrieval and management of user records.



CONCLUSION

- In conclusion, the User Management System developed using Java Servlets, JDBC, JSP, HTML, CSS, and MySQL represents a comprehensive solution for efficiently managing user data and facilitating communication between administrators and employees.
- Through the seamless integration of these technologies, the project successfully addresses the core objectives of user authentication, registration, profile management, and note communication.
- Leveraging MySQL as the backend database management system ensures secure and reliable storage of user information, while Java Servlets serve as the backbone for implementing business logic and facilitating data interaction. The use of JSP and HTML provides dynamic and user-friendly interfaces, enhancing the overall user experience.



Thank You!

