**EFFICIENT USER GOVERNANCE SYSTEM**

**Project Overview:**

This project is a basic web application designed for managing user information using JSP (JavaServer Pages) and a MySQL database. The system allows users to register, log in, update their information, and delete records. Here's a detailed explanation of each component:

**1. register.html**

* **Purpose**: Provides a registration form for new users to sign up.
* **Form Fields**: Collects first name, last name, email, and password.
* **Action**: Submits the form data to reg-process.jsp for processing.

**2. index.html**

* **Purpose**: Provides a login form for existing users.
* **Form Fields**: Accepts username (referred to as id here) and password.
* **Action**: Submits the form data to login.jsp for authentication.
* **Link**: Includes a link to the registration page (register.html).

**3. Login.jsp**

* **Purpose**: Handles user login attempts.
* **Functionality**:
  + Retrieves the username (first name) and password from the form.
  + Connects to the MySQL database and queries the users table to verify the credentials.
  + Displays a welcome message upon successful login or an error message if credentials are invalid.
* **Post-login Options**: Provides buttons to navigate to pages for updating or deleting user records.

**4. reg-process.jsp**

* **Purpose**: Processes user registration data.
* **Functionality**:
  + Retrieves user input from the registration form.
  + Connects to the MySQL database and inserts the new user record into the users table.
  + Displays a confirmation message and a link to the login page upon successful registration.

**5. Retrieve-update.jsp**

* **Purpose**: Displays a list of all users in the database.
* **Functionality**:
  + Connects to the database and retrieves all user records.
  + Displays user information in a table with an option to update each user's details via a link.

**6. Delete1.jsp**

* **Purpose**: Displays a list of all users with options to delete.
* **Functionality**:
  + Connects to the database and retrieves all user records.
  + Displays user information in a table with a "Delete" button for each user.
  + The button links to delete.jsp, which handles the actual deletion.

**7. Delete.jsp**

* **Purpose**: Handles the deletion of a user record.
* **Functionality**:
  + Retrieves the user ID from the request.
  + Connects to the database and executes a SQL DELETE statement to remove the user record with the specified ID.
  + Displays a success message after deletion.

**8. Update.jsp**

* **Purpose**: Displays a form to update a user's information.
* **Functionality**:
  + Retrieves the user ID from the request and fetches the corresponding user record from the database.
  + Displays a form pre-filled with the user's current data.
  + Submits the updated data to update3.jsp for processing.

**9. Update3.jsp**

* **Purpose**: Processes the user information update.
* **Functionality**:
  + Retrieves updated user data from the form.
  + Connects to the database and executes a SQL UPDATE statement to modify the user record with the new details.
  + Displays a success or error message based on the outcome of the update operation.

**Key Features**

* **User Registration**: Allows new users to create accounts.
* **User Login**: Authenticates users and provides access to additional functionalities.
* **Data Retrieval**: Lists all users and allows for individual record updates.
* **Record Update**: Enables modification of user details.
* **Record Deletion**: Permits removal of user records from the database.

**Recommendations for Improvement**

1. **Security Enhancements**:
   * Implement password hashing to store passwords securely.
   * Use PreparedStatement to prevent SQL injection attacks.
   * Improve session management and access controls.
2. **Code Refactoring**:
   * Separate database logic from presentation logic for better maintainability.
   * Utilize Java Servlets for handling business logic and JSP for presentation.
3. **User Experience**:
   * Improve the user interface with better styling and validation.
   * Add client-side and server-side validation for form inputs.
4. **Error Handling**:
   * Provide user-friendly error messages and handle exceptions more gracefully.
5. **Database Management**:
   * Implement connection pooling for better performance and resource management.

By addressing these points, you can enhance the application's functionality, security, and user experience.

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated