**Property Wizard: Project Documentation**

**Project Overview**

Property Wizard is a real estate management system designed to simplify the process of managing properties, users, and transactions. The project uses a combination of Java technologies such as JSP, Servlets, and JDBC, and it is built using the Maven framework. The system allows users to view, edit, and manage property listings as well as perform user administration tasks.

**Features**

**General Features**

* **User Authentication**: Secure login functionality for administrators and regular users.
* **Property Management**: Create, update, view, and delete property listings.
* **User Administration**: Manage user roles and permissions.
* **Dynamic UI**: Built using JSP to display dynamic content.

**Pages Included**

1. **index.jsp**: Entry point of the application. Displays navigation options.
2. **login.jsp**: Provides a login form for user authentication.
3. **welcome.jsp**: Displays the list of available properties.
4. **edit.jsp**: Allows users to edit the details of a property.
5. **userform.jsp**: Enables administrators to add new users.
6. **userlist.jsp**: Displays a list of users in a tabular format.
7. **view.jsp**: Provides detailed information about a specific property.

**Technical Details**

**Technologies Used**

* **Backend**: Java, Servlets
* **Frontend**: JSP (Java Server Pages), HTML, CSS
* **Database**: MySQL (or any relational database)
* **Build Tool**: Maven
* **Server**: Apache Tomcat

**Key Components**

**Database Schema**

1. **Users Table**:
   * id (Primary Key)
   * name
   * email
   * role
   * password
2. **Properties Table**:
   * id (Primary Key)
   * name
   * location
   * price

**Key Classes and Interfaces**

1. **PropertyServlet**:
   * Handles CRUD operations for properties.
2. **UserServlet**:
   * Manages user-related operations.
3. **DatabaseUtil**:
   * Provides a utility for establishing database connections using JDBC.

**JSP Pages Workflow**

* **login.jsp**: Takes user credentials and forwards them to LoginServlet for verification.
* **welcome.jsp**: Displays the properties fetched from the database.
* **edit.jsp**: Prefills property data for editing.
* **user-list.jsp**: Dynamically fetches and displays user data from the database.

**Installation and Setup**

1. **Prerequisites**:
   * JDK 8 or higher
   * Apache Maven
   * Apache Tomcat
   * MySQL Server
2. **Steps to Setup**:
   * Clone the repository:
   * git clone <repository-url>
   * Navigate to the project directory:
   * cd property-wizard
   * Configure the database:
     + Create a database named property\_wizard.
     + Import the provided SQL script to set up tables.
   * Update src/main/resources/db.properties with database credentials.
   * Build the project:
   * mvn clean install
   * Deploy the WAR file to Tomcat:
     + Copy the generated property-wizard.war to Tomcat's webapps directory.
   * Start the Tomcat server and access the application at:
   * http://localhost:8080/property-wizard

**Usage Guide**

1. **Login**:
   * Navigate to login.jsp.
   * Enter your credentials and click "Login."
2. **View Properties**:
   * Upon successful login, the welcome.jsp page displays available properties.
   * Click "View" to see detailed property information.
3. **Add/Edit Properties**:
   * Use edit.jsp to modify an existing property.
   * Use the "Add Property" option (if implemented) to create new listings.
4. **Manage Users**:
   * Go to user-list.jsp to view all users.
   * Click "Edit" to update user details or "Delete" to remove a user.

**Future Enhancements**

* Add **search and filter** capabilities for property listings.
* Implement **role-based access control** for admin and regular users.
* Enable **image uploads** for property listings.
* Develop a **REST API** for third-party integrations.
* Introduce **pagination** for large data sets in user-list.jsp and welcome.jsp.

**Troubleshooting**

* **Login Issues**:
  + Ensure the database has valid user credentials.
  + Check LoginServlet for potential bugs.
* **Database Connectivity**:
  + Verify db.properties contains the correct database URL, username, and password.
  + Check that the database server is running.
* **Deployment Errors**:
  + Ensure the WAR file is correctly deployed to the webapps directory.
  + Check Tomcat logs for deployment errors.

**Contact Information**

For further assistance or inquiries, please contact:

* **Developer**: Dhruv and Team
* **Email**: [[itsdhruvbhardwaj@](mailto:your-email@example.com)gmail.com]