E-Service Project

- **Abstract** **[1]**
- **Overview** **[1]**
- **Functional Requirement** **[2]**
- **Non-Functional Requirement** **[2]**
- **Structure** **[4]**

ABSTRACT

The e-Service Management System is an online system developed with the use of **JSP (Java Server Pages), Servlets and Core Java** as back end technologies. It has **MySQL** as the database. This project is intended to enhance the efficient provision and management of services across various areas enabling easy interaction between the customer and the provider of the services.

The project employs a **Model-View-Controller** (MVC) structure for its implementation in order to avoid intermingling of concerns making the project easy to extend and grow. The system allows users to create accounts, search the list of services offered, make orders for certain services, check the orders and their current state; it also allows admin users to control accounts of normal users, service operators and service types.

Overview

Engaged with this web based e service. Everything integrated into one place for the management and dispatch of service providers, clients and administrators. Customers are brought into the equation making service provision void of bursting into fragmentation providing a unified approach to make service provision modernized, improve performance, increase the availability of offered services, and enhance the quality of service delivery.

Goal of e-Service Management System:

The e-Service Management System provides an intuitive and uncomplicated interface that enables:

- Customers to search, order, and monitor services at their own convenience.
- Service providers to organize their products/sw services together with customer orders.
- Administrators in this case perform a supervisory role over the ecosystem where users, services, and transactions are. Control's features are offered

E-Service Project

Functional Requirements

1. User Registration and Authentication

User Registration:
A user should be able to register with their username, password, and email.
User information should be efficiently protected in the Store user details.

User Login:
Integrate a login feature. This feature should allow the user to authorize themselves by entering their username and password.

Undertake user session management to manage users who are logged in.

2. Service Booking

Service Request:
Users must be able to schedule the session with available services by entering the name of the service and the date when it should be reserved.
Add the sessions to the data base against the logged in user in the data base.

3. Session Management

Session Handling:
Use of Http sessions for users who have logged in the system.
Let the users log out so ending their session.

Non Functional Requirements

1. Security

Data Encryption:
Pass on the sensitive particulars to be encrypted (for example passwords) to the data base only after encryption.
Provide usage of WWW secure through HTTPS.

Authentication and Authorization:

Integrate a mechanism to stressed restricted access to some operations based on certain roles of users.

2. Performance

Scalability:
Conduct systematic planning allowing the system to serve a number of users and a number of bookings at the same time.
Make recommendations for improving database queries and optimization of indexing.

Response Time:
Make sure that actions taken by the user of the system such as registration, login, or making a booking are completed within seconds.

Technologies Utilized:

- **Frontend**: JSP, HTML, CSS, JavaScript (with optional frameworks for styling).
- **Backend**: Core Java, Servlets.
- **Database**: MySQL, integrated using JDBC.
- **Server**: Apache Tomcat for hosting the application.

E-Service Project

E-Service-Project Structure

```
|
├── src
|   ├── main
|   |   ├── java
|   |   |   └── com
|   |   |       └── eservice
|   |   |           ├── controller
|   |   |           |   ├── RegisterController.java
|   |   |           |   ├── LoginController.java
|   |   |           |   ├── BookServiceController.java
|   |   |           |   └── LogoutController.java
|   |   |           ├── model
|   |   |           |   ├── User.java
|   |   |           |   ├── Booking.java
|   |   |           |   ├── UserDAO.java
|   |   |           |   ├── BookingDAO.java
|   |   |           └── util
|   |   |               └── DatabaseUtil.java
|   |   ├── resources
|   |   |   └── database-config.xml
|   |   └── webapp
|   |       ├── WEB-INF
|   |       |   └── web.xml
|   |       ├── views
|   |       |   ├── register.jsp
```

```
│   │   │   ├── login.jsp
│   │   │   ├── services.jsp
│   │   │   ├── booking.jsp
│   │   │   └── logout.jsp
│   │   ├── css
│   │   │   └── style.css (optional, if using CSS)
│   │   └── index.jsp
```