**Parking Portal**

1. **PURPOSE:**Setting up a new standard way to collect parking (VEHICLE) documents efficiently and enforcing them effectively through digitalization. Reduces operational time i.e. (It saves the time for managers in the process of collecting documents from every unit, verify them, scan them and again email them to the authorized person).
2. **SCOPE:**
3. To streamline the entire manual processes of documentation into digitalization.
4. To make data-driven decisions with real-time data.
5. To save the time of manager.
6. To create direct contact with tenants for the documentation process.
7. **DEFINATION:** To understand the need and importance of the documents, before allocation of the parking space and stickers to the tenants we need to verify whether the documents are expired or relevant, if documents are relevant, we will provide the stickers and parking space for the same. Tenant will receive the email notification 30 days prior of expiry date as per the documents received and they must provide the new relevant documents for the same.
8. **TECHNOLOGIES TO BE USED:**

**Frontend:** HTML, CSS, Jsp.

**Backend:** Java, Spring.

**Database:** MYSQL (Ver 8.0.20).

**Server:** Tomcat (Ver 8.5.84).

**IDE:** Spring Boot (Ver 4.17), JDK (Ver 8), MYSQL Workbench.

1. **IDENTIFICATION OF NEEDS:**

This project essentially needs functions:

1. It will be helpful for management to keep track of parking details.
2. Direct contact with tenants for the documentation process.
3. **USER CHARACTARISTICS:**

**i) ADMIN:** Can login/logout.

Can manage the parking system.

Can add/view/edit/update users database.

Can Accept/Reject Documents

Can allot parking space to tenants.

**ii) REGISTERED USER:** Can login/logout.

Can view/upload/update documents.

1. **SOFTWARE DEVELOPMENT PROCESS:** 
   * 1. **Prototype model:** Used for UI (User Interface) part of the project.

Simple data processing application.

Easily detect missing functionality.

* + 1. **Evolutionary model:** Used for object-oriented development.

Reduce the error/risk of software.

1. **USER CASE DIAGRAM**

**<Includes>**

**<Includes>**

**USER**

**<Includes>**

**<Includes>**

**<Includes>**

**<Includes>**

**ADMIN**

Checked Parking slot availability,

Allot parking space

1. **ACTIVITY DIAGRAM**

**No**

**yes**

Property Information

**Try again/forgot password**

Valid Username & password

Login

**USER**

**Admin**

Admin Login

**No**

Valid Username & password

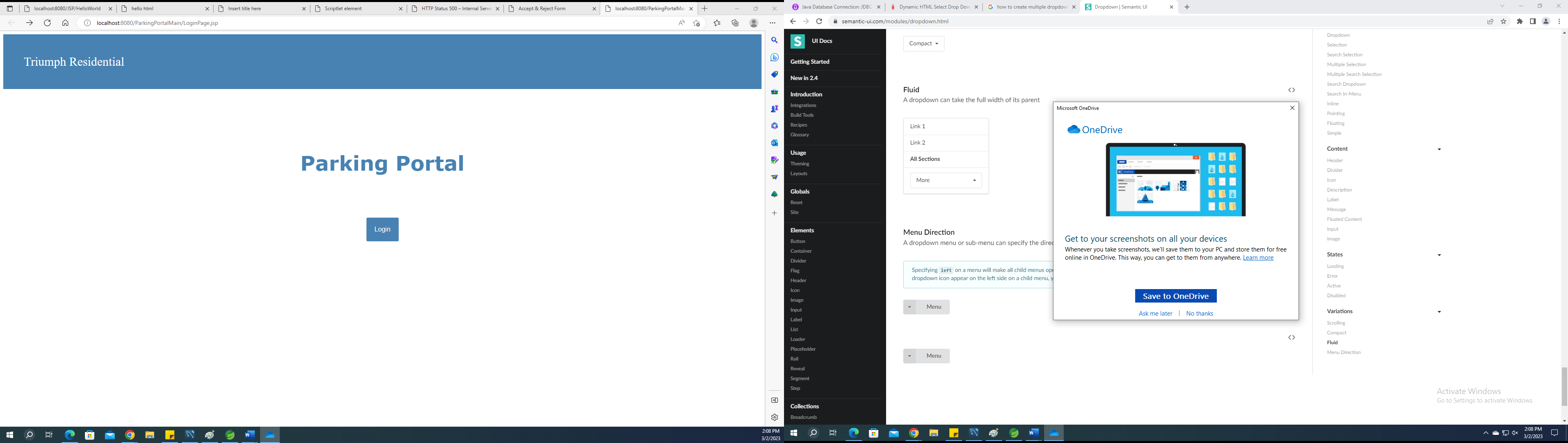
View/upload/update documents

**yes**

View, Update, Delete, Edit, Accept/Reject Doc database

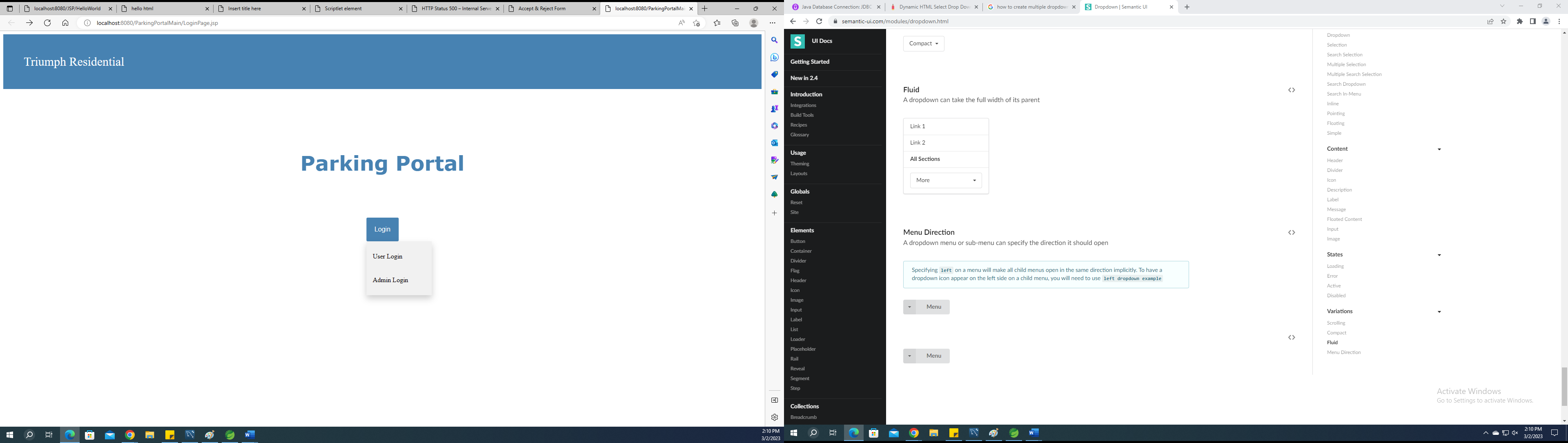
**Parking Portal**

Hover the curser on Login Button

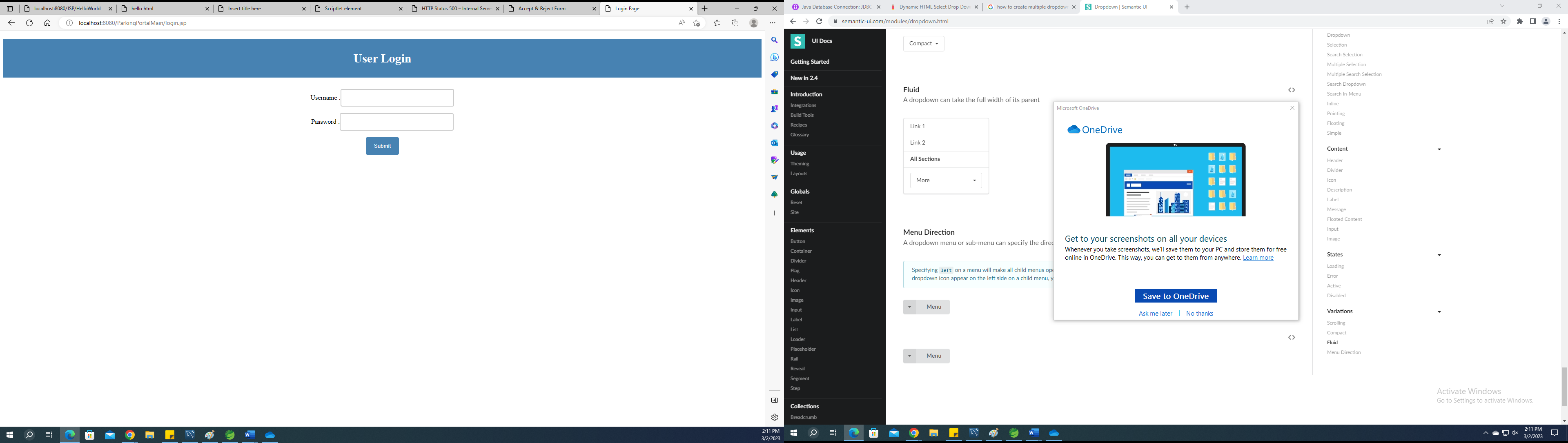


1. **Admin & User Login**

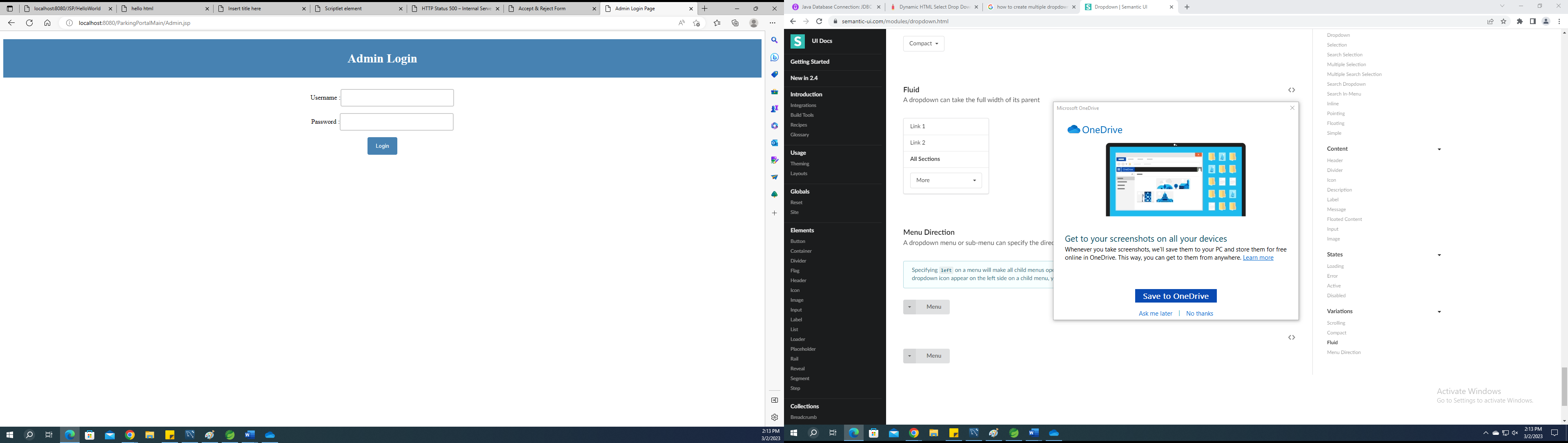
Click on UserLogin/AdminLogin



* 1. User Login



* 1. Admin Login

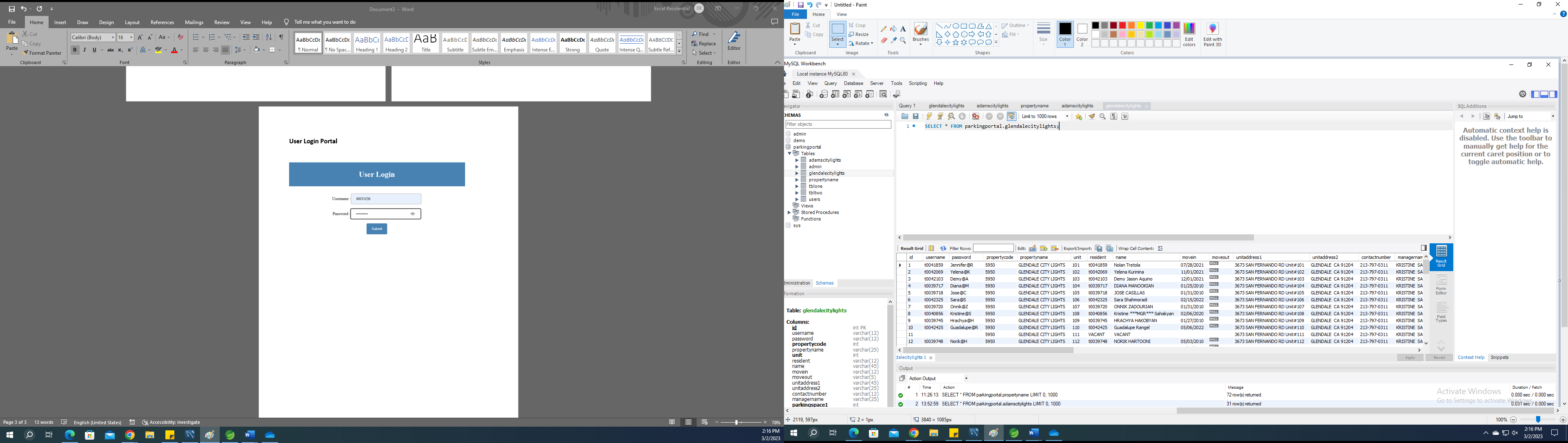


1. **User Login Portal**

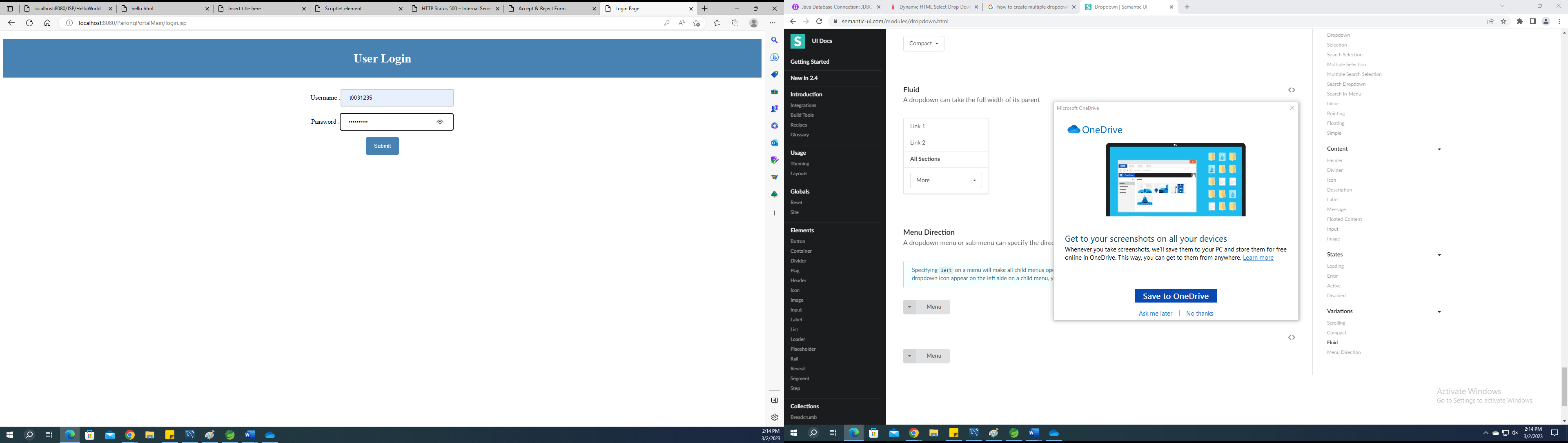
Username: t0031235

Password: Leighton@C

Database



Put username and password in login Field and then click on submit button



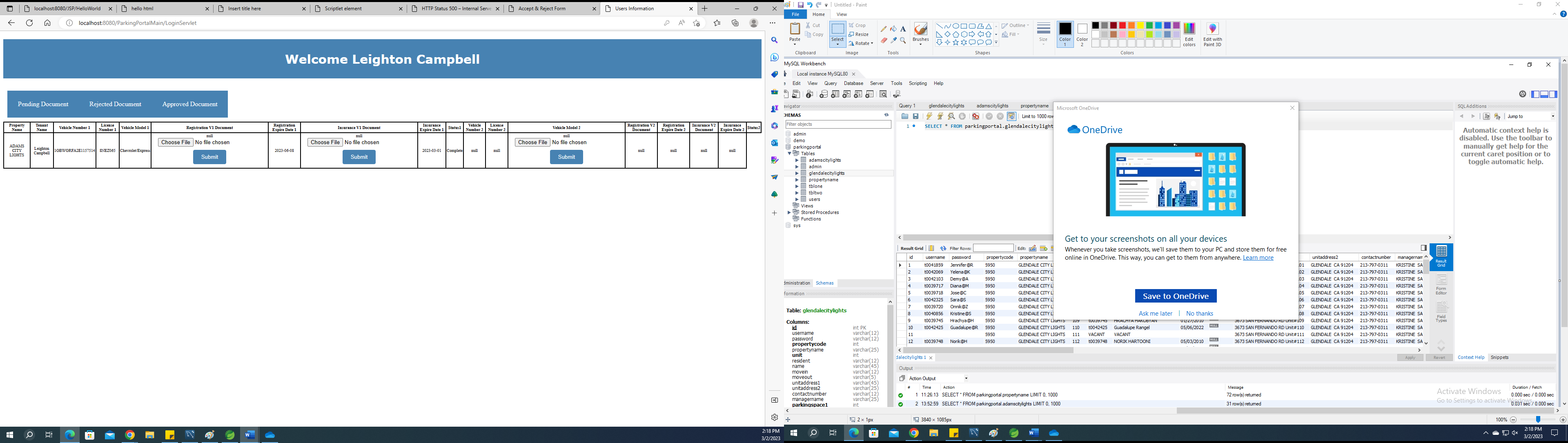
2.1) If username and password valid -> Open the User detail page

* User can upload and update their documents in portal

Steps for uploading Documents:

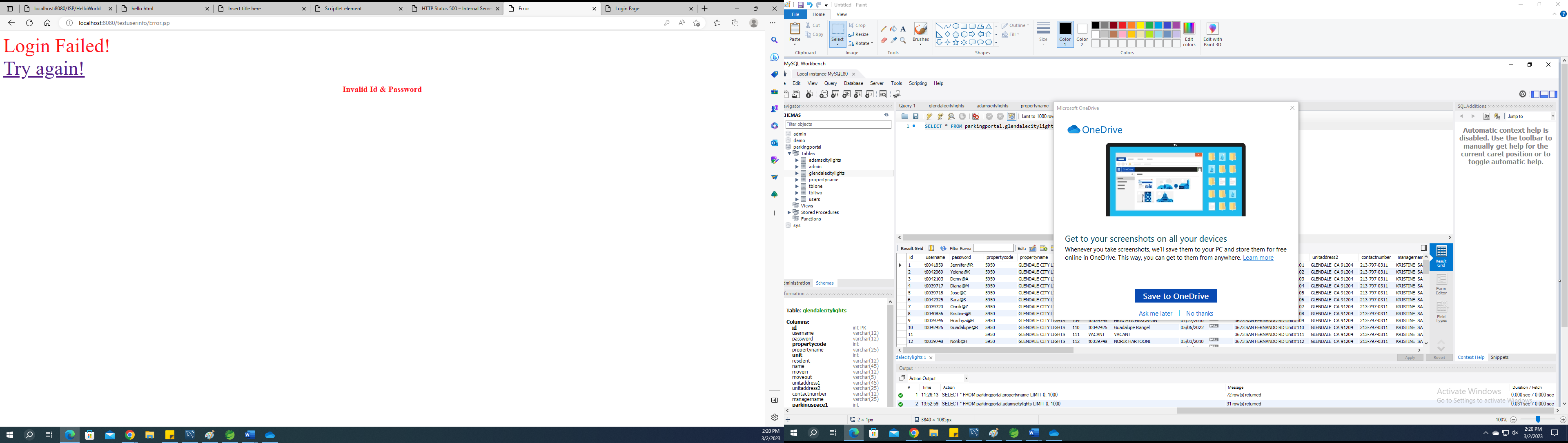
1. Click on Choose file
2. Select your file
3. Click on Submit button

* User will get notification of Expire documents before 30 days prior
* If admin Accept/Reject any doc users get notification in Reject/Approve Document page



2.2) If username and password Invalid ->

Click on Try again

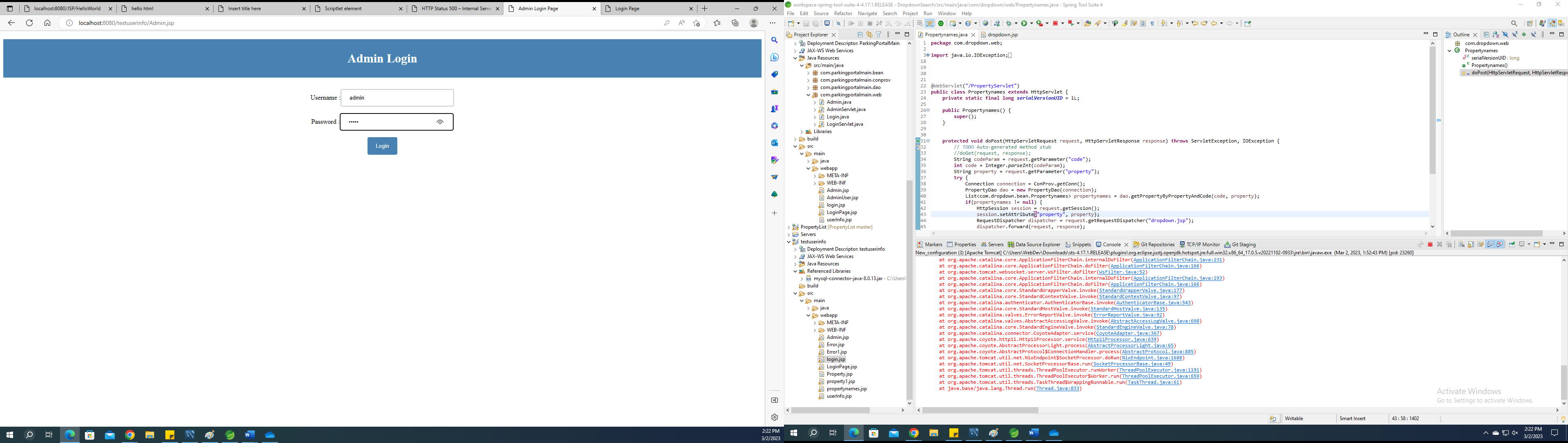


1. **Admin Login**

Put username and password in login Field and then click on submit button

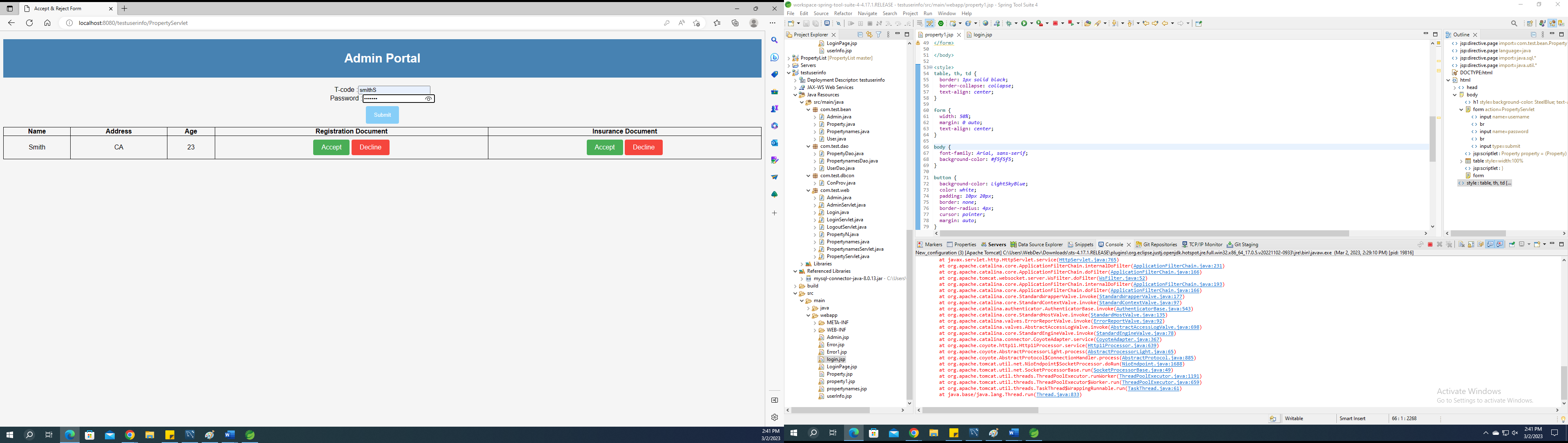
Username: admin

Password: admin



3.1) If username and password valid -> Open the Admin Login Portal

Put users T-code and password in input field and then click on submit button



Admin can Accept/Reject documents through accept and Decline button

**Implementation**

**User’s(Tenants) Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Tittle** | **FE Description** | **Status** | **BE Description** | **Status** |
| 1. | Login page | * Input field: username, password. * Button submit | Complete | Tenant login validation using database. | Complete |
| 2. | User information | * In header display welcome tenant name. * Navbar: pending/approved/reject Document. * pending/approved/reject Document page creation. * Display tenant detail. * Upload document button. * Document submission button. * Logout button |  | * After login api display the user information using login validation. * Display only not null columns. * Upload/update Document * Submit document (send notification in admin portal notification icon) * Expire date notification * Status(complete/incomplete) * If admin approved all document status show as complete, if admin approve only one document status show as incomplete) * Logout button |  |
| 3. | Pending Documents | Pending document notification |  | Pending document notification |  |
| 4. | Approved Documents | Approved document notification |  | Approved document notification |  |
| 5. | Reject Documents | Reject document notification |  | Reject document notification |  |

**Admin Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Tittle** | **FE Description** | **Status** | **BE Description** | **Status** |
| 1. | Login page | * Input field: username, password. * Button submit * Notification icon | Complete | * Tenant login validation * Get notification of tenant document upload(tenant name upload document name) | Complete |
| 2. | Property name | * Input field: t-code, property name * Dropdown search |  | * After selecting **dropdown** users t-code and property name api opens the accept & reject document page * Database connection with all 72 property table |  |
| 3. | Accept & Reject page | * Users information * Accept and reject button * View document * Logout button |  | * Display users information as per their t-code and property name * Display only not null columns. * Accept/reject document * Logout button |  |

**Database**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Tittle** | **Description** | **Status** |
| **1.** | **Create schema** | **Create schema parking portal** | **Complete** |
| **2.** | **Create tables** | **Property name, properties 72** |  |
| **3.** | **Import tables** | **Import property names list, 72 properties table** |  |
| **4.** | **Relation** | **Create relation between property name list and properties table** |  |