**Part I: Basic Application**

**Frontend (HTML and CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Rentify - Registration</title>

<style>

/\* Add your CSS styles here \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f0f0f0;

}

.container {

max-width: 800px;

margin: 20px auto;

background-color: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

}

input[type="text"],

input[type="email"],

input[type="tel"],

input[type="password"],

select {

width: calc(100% - 20px);

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

button {

width: 100%;

padding: 10px;

background-color: #007bff;

color: #fff;

border: none;

border-radius: 5px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<div class="container">

<h1>Rentify - Registration</h1>

<form action="#" method="POST" id="registrationForm">

<!-- Login Section -->

<h2>Login</h2>

<input type="text" name="username" placeholder="Username" required>

<input type="password" name="password" placeholder="Password" required>

<!-- Personal Information Section -->

<h2>Personal Information</h2>

<input type="text" name="firstName" placeholder="First Name" required>

<input type="text" name="lastName" placeholder="Last Name" required>

<input type="email" name="email" placeholder="Email" required>

<input type="tel" name="phoneNumber" placeholder="Phone Number" required>

<!-- Role Selection -->

<h2>Select Role</h2>

<label><input type="radio" name="role" value="seller" required> Seller</label>

<label><input type="radio" name="role" value="buyer" required> Buyer</label>

<!-- Property Details Section (for Seller) -->

<div id="sellerDetails" style="display: none;">

<h2>Property Details</h2>

<input type="text" name="propertyName" placeholder="Property Name">

<input type="text" name="area" placeholder="Area">

<input type="number" name="bedrooms" placeholder="Number of Bedrooms">

<input type="number" name="bathrooms" placeholder="Number of Bathrooms">

<!-- Add more property details fields here -->

</div>

<!-- Submit Button -->

<button type="submit">Register</button>

</form>

</div>

<script>

// Add JavaScript code here to show/hide property details section based on role selection

document.querySelector('input[name="role"]').addEventListener('change', function() {

var sellerDetails = document.getElementById('sellerDetails');

if (this.value === 'seller') {

sellerDetails.style.display = 'block';

} else {

sellerDetails.style.display = 'none';

}

});

</script>

</body>

</html>

#### Backend (Java with Spring Boot):

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

@SpringBootApplication

@Controller

public class RentifyApplication {

public static void main(String[] args) {

SpringApplication.run(RentifyApplication.class, args);

}

@RequestMapping("/")

public String showRegistrationForm(Model model) {

model.addAttribute("user", new User());

return "registration";

}

@PostMapping("/register")

public String register(User user) {

// Save user data to the database or perform necessary actions

return "redirect:/";

}

@GetMapping("/properties")

public String showProperties(Model model) {

// Retrieve property data from the database and pass it to the view

return "properties";

}

}

class User {

private String username;

private String password;

private String firstName;

private String lastName;

private String email;

private String phoneNumber;

private String role;

private String propertyName;

private String area;

private int bedrooms;

private int bathrooms;

// Getters and setters

}

### Part II: Add-On Features

#### Pagination and Form Validation:

Now we can implement pagination and form validation using JavaScript libraries like jQuery or frontend frameworks like React, Angular, or Vue.js. For this example, let's add some simple form validation using JavaScript:

document.getElementById("registrationForm").addEventListener("submit", function(event) {

var form = event.target;

var username = form.username.value;

var password = form.password.value;

var firstName = form.firstName.value;

var lastName = form.lastName.value;

var email = form.email.value;

var phoneNumber = form.phoneNumber.value;

var role = form.role.value;

if (!username || !password || !firstName || !lastName || !email || !phoneNumber || !role) {

alert("All fields are required");

event.preventDefault();

}

});