**LAB 2: CSS AND BOOTSTRAP**

Disha Jain - 220905554 - 61

1)Design the student bio-data form using button, label, textbox, radio button, table and checkbox.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Student Bio-Data Form</title>

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

<style>

body {

font-family: 'Arial', sans-serif;

background-color: #f4ece0;

color: #5d4037;

}

h2 {

color: #3e2723;

font-family: 'Georgia', serif;

}

.form-group label {

color: #3e2723;

font-weight: bold;

}

.form-check-label {

font-size: 16px;

color: #3e2723;

}

.btn-primary {

background-color: #6d4c41;

border-color: #5d4037;

font-size: 16px;

}

.btn-primary:hover {

background-color: #5d4037;

border-color: #3e2723;

}

.container {

background-color: #fff;

border-radius: 8px;

padding: 30px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

margin-top: 50px;

}

.subjects-section {

margin-top: 15px;

}

.table th, .table td {

vertical-align: middle;

}

.form-group-submit {

margin-top: 30px;

}

</style>

</head>

<body>

<div class="container">

<h2 class="text-center mb-4">Student Bio-Data Form</h2>

<form>

<div class="form-group">

<label for="fullName">Full Name</label>

<input type="text" class="form-control" id="fullName" placeholder="Enter Full Name">

</div>

<div class="form-group">

<label for="dob">Date of Birth</label>

<input type="date" class="form-control" id="dob">

</div>

<div class="form-group">

<label for="age">Age</label>

<input type="number" class="form-control" id="age" placeholder="Enter Age" required>

</div>

<div class="form-group">

<label>Gender</label>

<div class="form-check form-check-inline">

<input class="form-check-input" type="radio" name="gender" id="genderMale" value="Male">

<label class="form-check-label" for="genderMale">Male</label>

</div>

<div class="form-check form-check-inline">

<input class="form-check-input" type="radio" name="gender" id="genderFemale" value="Female">

<label class="form-check-label" for="genderFemale">Female</label>

</div>

</div>

<div class="form-group">

<label for="registrationNumber">Registration Number</label>

<input type="text" class="form-control" id="registrationNumber" placeholder="Enter 9-digit Registration Number" maxlength="9" pattern="\d{9}" required>

<small class="form-text text-muted">Please enter exactly 9 digits.</small>

</div>

<div class="form-group">

<label for="email">Email ID</label>

<input type="email" class="form-control" id="email" placeholder="Enter Email ID" required>

</div>

<div class="form-group">

<label for="contactNumber">Contact Number</label>

<input type="tel" class="form-control" id="contactNumber" placeholder="Enter Contact Number" maxlength="10" pattern="\d{10}" required>

<small class="form-text text-muted">Please enter a 10-digit contact number.</small>

</div>

<div class="form-group">

<label>Branch</label>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchCSE" value="CSE">

<label class="form-check-label" for="branchCSE">

CSE

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchCSEAI" value="CSE AI & ML">

<label class="form-check-label" for="branchCSEAI">

CSE AI & ML

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchDSE" value="DSE">

<label class="form-check-label" for="branchDSE">

DSE

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchCCE" value="CCE">

<label class="form-check-label" for="branchCCE">

CCE

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchIT" value="IT">

<label class="form-check-label" for="branchIT">

IT

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchBIOMED" value="BIOMED">

<label class="form-check-label" for="branchBIOMED">

BIOMED

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchBIOTECH" value="BIOTECH">

<label class="form-check-label" for="branchBIOTECH">

BIOTECH

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchCIVIL" value="CIVIL">

<label class="form-check-label" for="branchCIVIL">

CIVIL

</label>

</div>

<div class="form-check">

<input class="form-check-input" type="radio" name="branch" id="branchMECHANICAL" value="MECHANICAL">

<label class="form-check-label" for="branchMECHANICAL">

MECHANICAL

</label>

</div>

</div>

<div class="form-group">

<label for="previousEducation">Previous Education</label>

<table class="table table-bordered">

<thead>

<tr>

<th>Education Level</th>

<th>Institution Name</th>

<th>Percentage</th>

<th>Year of Passing</th>

</tr>

</thead>

<tbody>

<tr>

<td>School</td>

<td><input type="text" class="form-control" placeholder="Enter School Name"></td>

<td><input type="number" class="form-control" placeholder="Enter Percentage" step="0.1" min="0" max="100"></td>

<td><input type="number" class="form-control" placeholder="Enter Year" min="1900" max="2099"></td>

</tr>

<tr>

<td>Intermediate</td>

<td><input type="text" class="form-control" placeholder="Enter Intermediate Institution Name"></td>

<td><input type="number" class="form-control" placeholder="Enter Percentage" step="0.1" min="0" max="100"></td>

<td><input type="number" class="form-control" placeholder="Enter Year" min="1900" max="2099"></td>

</tr>

</tbody>

</table>

</div>

<div class="form-group-submit">

<button type="submit" class="btn btn-primary btn-block">Submit</button>

</div>

</form>

</div>

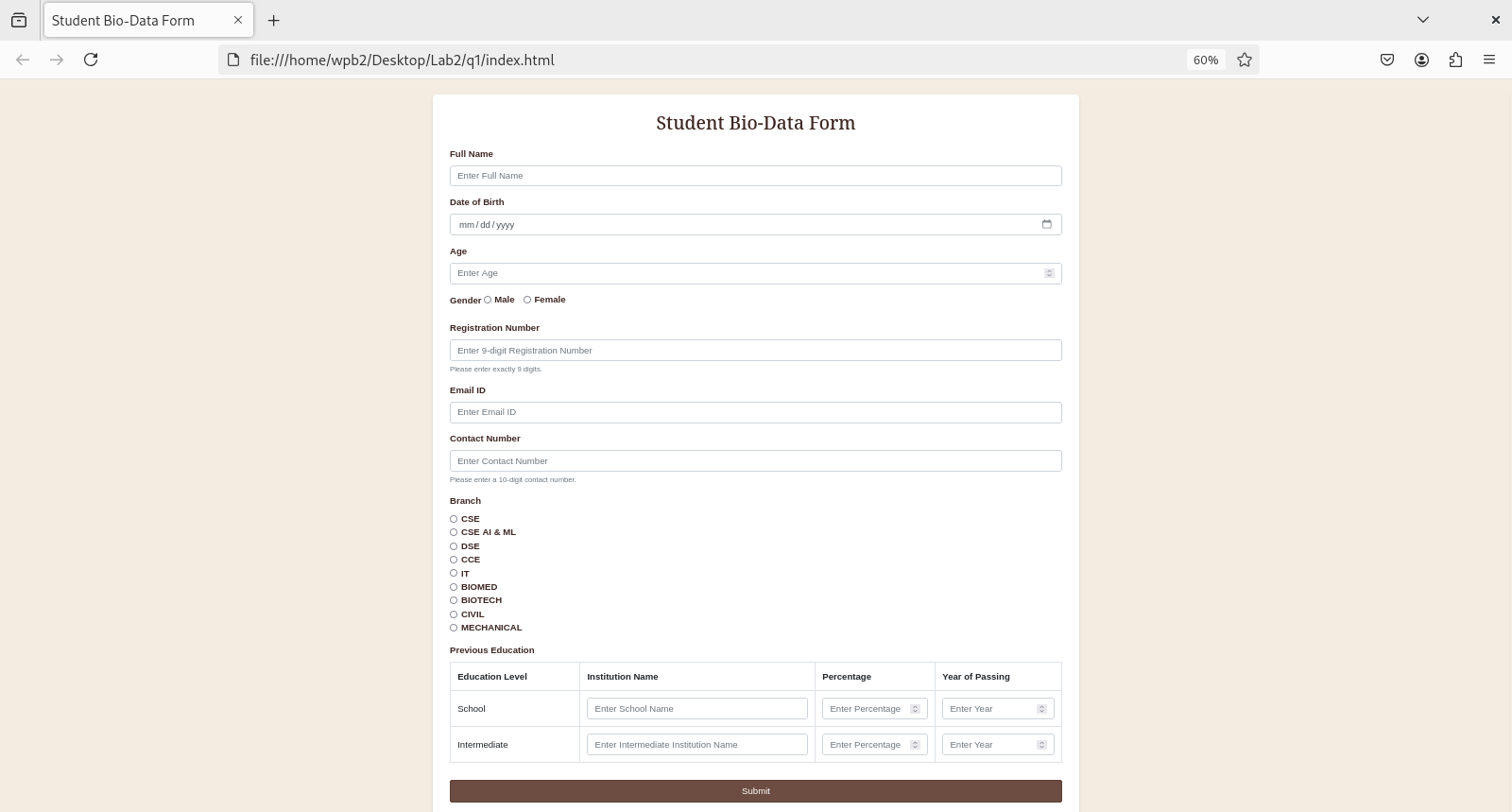
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>



2) Design a web page which shows the database oriented CRUD operation. Consider Employee data.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Employee CRUD Operations</title>

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

<style>

body {

background-color: #f4e1d2;

color: #5b4e3d;

}

.container {

margin-top: 30px;

background-color: #fff4e1;

padding: 20px;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.form-group label {

font-weight: bold;

color: #5b4e3d;

}

.table th, .table td {

text-align: center;

padding: 12px;

}

.table-bordered {

border-color: #8f7b5a;

}

.table th {

background-color: #d8c1a0;

color: #5b4e3d;

}

.table-striped tbody tr:nth-child(odd) {

background-color: #f4e1d2;

}

.employee-table-wrapper {

max-height: 120px;

overflow-y: auto;

border: 1px solid #8f7b5a;

margin-top: 20px;

}

.details-table {

margin-top: 20px;

}

#employeeDetailsSection {

display: none;

}

button {

background-color: #8f7b5a;

color: white;

border: none;

padding: 8px 16px;

border-radius: 4px;

cursor: pointer;

font-size: 14px;

}

button:hover {

background-color: #6b5b3c;

}

button:focus {

outline: none;

box-shadow: 0 0 0 2px rgba(139, 123, 90, 0.5);

}

h2, h3 {

color: #5b4e3d;

}

.form-control {

background-color: #f4e1d2;

border-color: #8f7b5a;

color: #5b4e3d;

}

</style>

</head>

<body>

<div class="container">

<h2 class="text-center">Employee CRUD Operations</h2>

<form id="employeeForm">

<div class="form-group">

<label for="empName">Employee Name</label>

<input type="text" class="form-control" id="empName" placeholder="Enter Employee Name" required>

</div>

<div class="form-group">

<label for="empPosition">Position</label>

<input type="text" class="form-control" id="empPosition" placeholder="Enter Employee Position" required>

</div>

<div class="form-group">

<label for="empSalary">Salary</label>

<input type="number" class="form-control" id="empSalary" placeholder="Enter Salary" required>

</div>

<div class="form-group">

<label for="empDepartment">Department</label>

<input type="text" class="form-control" id="empDepartment" placeholder="Enter Department" required>

</div>

<input type="hidden" id="empId">

<button type="submit" id="submitBtn">Add Employee</button>

</form>

<h3 class="mt-5">Employee List</h3>

<div class="employee-table-wrapper">

<table class="table table-bordered" id="employeeTable">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<!-- Data will be dynamically populated here -->

</tbody>

</table>

</div>

<div id="employeeDetailsSection">

<h3 class="mt-5">Employee Details</h3>

<table class="table table-bordered details-table" id="employeeDetailsTable">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Position</th>

<th>Salary</th>

<th>Department</th>

</tr>

</thead>

<tbody>

<!-- Data will be dynamically populated here when Show Details is clicked -->

</tbody>

</table>

</div>

</div>

<!-- Bootstrap JS and dependencies -->

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<script>

// Simulating a database with an array

let employees = [

{ id: 1, name: 'Disha Jain', position: 'Software Engineer', salary: 50000, department: 'IT' },

{ id: 2, name: 'Snehansh Jerath', position: 'Manager', salary: 70000, department: 'HR' },

{ id: 3, name: 'Adel Sarah Dsouza', position: 'Designer', salary: 45000, department: 'Design' },

{ id: 4, name: 'Komal Mathur', position: 'Developer', salary: 55000, department: 'IT' },

{ id: 5, name: 'Anshu Kantipudi', position: 'Manager', salary: 65000, department: 'HR' }

];

// Function to fetch and display employees in the table

function fetchEmployees() {

let rows = '';

employees.forEach(emp => {

rows += `

<tr>

<td>${emp.id}</td>

<td>${emp.name}</td>

<td>

<button class="btn btn-warning btn-sm" onclick="editEmployee(${emp.id})">Edit</button>

<button class="btn btn-danger btn-sm" onclick="deleteEmployee(${emp.id})">Delete</button>

<button class="btn btn-info btn-sm" onclick="showEmployeeDetails(${emp.id})">Show Details</button>

</td>

</tr>

`;

});

$('#employeeTable tbody').html(rows);

}

// Add or Update Employee

$('#employeeForm').on('submit', function(e) {

e.preventDefault();

const employee = {

name: $('#empName').val(),

position: $('#empPosition').val(),

salary: $('#empSalary').val(),

department: $('#empDepartment').val()

};

const empId = $('#empId').val();

if (empId) {

// Update employee

const index = employees.findIndex(emp => emp.id == empId);

if (index !== -1) {

employees[index] = { ...employees[index], ...employee }; // Update the employee data

alert('Employee updated successfully!');

}

$('#submitBtn').text('Add Employee');

} else {

// Create new employee

const newEmployee = {

id: employees.length + 1,

...employee

};

employees.push(newEmployee);

alert('Employee added successfully!');

}

// Reset form and refresh employee list

$('#employeeForm')[0].reset();

$('#empId').val('');

fetchEmployees();

});

// Edit Employee

function editEmployee(id) {

const emp = employees.find(emp => emp.id === id);

if (emp) {

$('#empName').val(emp.name);

$('#empPosition').val(emp.position);

$('#empSalary').val(emp.salary);

$('#empDepartment').val(emp.department);

$('#empId').val(emp.id);

$('#submitBtn').text('Update Employee');

}

}

// Delete Employee

function deleteEmployee(id) {

const index = employees.findIndex(emp => emp.id === id);

if (index !== -1) {

if (confirm('Are you sure you want to delete this employee?')) {

employees.splice(index, 1); // Remove the employee from the array

alert('Employee deleted successfully!');

fetchEmployees();

}

}

}

// Show Employee Details in a new row table

function showEmployeeDetails(id) {

const emp = employees.find(emp => emp.id === id);

if (emp) {

// Populate the Employee Details Table with the selected employee data

const detailRow = `

<tr>

<td>${emp.id}</td>

<td>${emp.name}</td>

<td>${emp.position}</td>

<td>${emp.salary}</td>

<td>${emp.department}</td>

</tr>

`;

$('#employeeDetailsTable tbody').html(detailRow);

$('#employeeDetailsSection').show();

}

}

// Initialize the page

$(document).ready(function() {

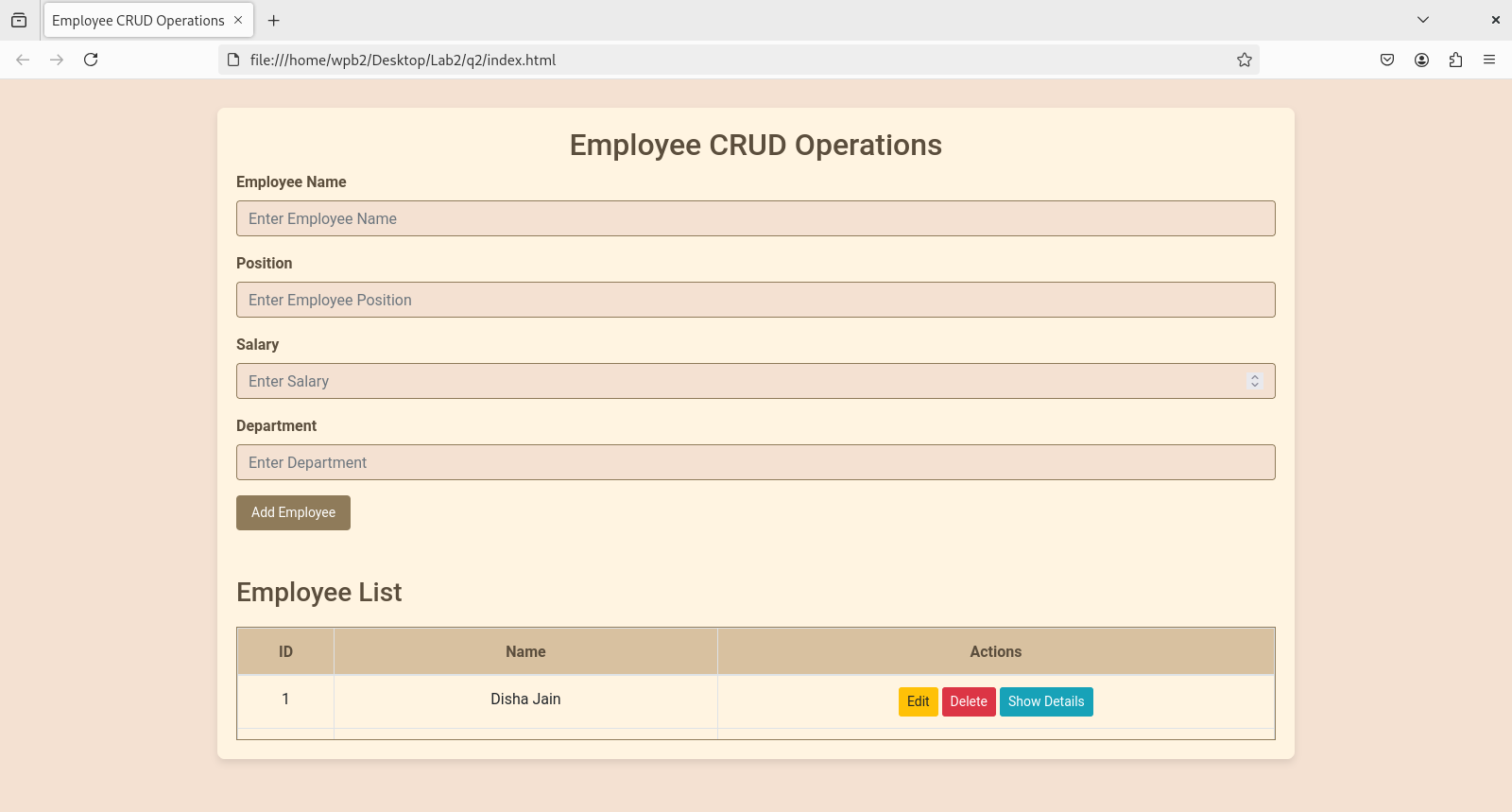
fetchEmployees();

});

</script>

</body>

</html>



3) Create a web page using bootstrap as mentioned. Divide the page in to 2 parts top and bottom, then divide the bottom into 3 parts and design each top and bottom part using different input groups, input, badges, buttons and button groups. Make the design more attractive.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Recipe Collection</title>

<!-- Bootstrap CSS -->

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

<style>

body {

background-color: #f4f1e1; /\* Light beige background \*/

color: #4e3629; /\* Dark brown text color \*/

font-family: 'Georgia', serif;

}

.container {

margin-top: 20px;

}

h1 {

font-family: Georgia, 'Times New Roman', Times, serif;

color: #6f4f37; /\* Dark brown header \*/

}

.input-group-text {

font-weight: bold;

background-color: #6f4f37; /\* Dark brown input group \*/

color: white;

}

.recipe-card {

margin-top: 20px;

background-color: #e9d8a6; /\* Beige background for recipe cards \*/

border: 1px solid #6f4f37; /\* Dark brown border \*/

}

.badge-custom {

background-color: #6f4f37;

color: white;

}

.recipe-card-body {

text-align: left;

}

.bottom-section {

margin-top: 40px;

}

.category-filter {

display: flex;

flex-direction: column;

gap: 10px;

}

.category-filter button {

background-color: #6f4f37; /\* Dark brown button \*/

color: white;

}

.recipe-collection {

max-width: 800px; /\* Makes the recipe collection wider \*/

}

.btn-primary {

background-color: #6f4f37;

border-color: #6f4f37;

}

.btn-warning {

background-color: #d49f1c; /\* Golden-yellow button for recipe of the day \*/

border-color: #d49f1c;

}

.card-header {

background-color: #6f4f37; /\* Dark brown header for Recipe of the Day \*/

color: white;

}

.card-body {

background-color: #e9d8a6; /\* Beige background for card body \*/

}

.card-text {

color: #4e3629; /\* Dark brown text color \*/

}

</style>

</head>

<body>

<div class="container">

<!-- Top Section - Add Recipe Form -->

<h1 class="text-center">MY RECIPE BOOK</h1>

<form id="recipeForm">

<div class="form-group">

<label for="recipeName">Recipe Name</label>

<input type="text" class="form-control" id="recipeName" placeholder="Enter recipe name" required>

</div>

<div class="form-group">

<label for="ingredients">Ingredients</label>

<textarea class="form-control" id="ingredients" rows="3" placeholder="Enter ingredients" required></textarea>

</div>

<div class="form-group">

<label for="instructions">Instructions</label>

<textarea class="form-control" id="instructions" rows="3" placeholder="Enter cooking instructions" required></textarea>

</div>

<div class="form-group">

<label for="cookingTime">Cooking Time (in minutes)</label>

<input type="number" class="form-control" id="cookingTime" placeholder="Enter cooking time" required>

</div>

<div class="form-group">

<label for="recipeCategory">Category</label>

<select class="form-control" id="recipeCategory" required>

<option value="Vegetarian">Vegetarian</option>

<option value="Non-Vegetarian">Non-Vegetarian</option>

<option value="Dessert">Dessert</option>

</select>

</div>

<button type="submit" class="btn btn-primary btn-block">Add Recipe</button>

</form>

<!-- Bottom Section - Divided into 3 Parts -->

<div class="bottom-section">

<div class="row">

<!-- First Part: Category Filter (Vertical) -->

<div class="col-md-3">

<h3>Filter by Category</h3>

<div class="category-filter">

<button type="button" class="btn btn-secondary" onclick="filterRecipes('All')">All</button>

<button type="button" class="btn btn-secondary" onclick="filterRecipes('Vegetarian')">Vegetarian</button>

<button type="button" class="btn btn-secondary" onclick="filterRecipes('Non-Vegetarian')">Non-Vegetarian</button>

<button type="button" class="btn btn-secondary" onclick="filterRecipes('Dessert')">Dessert</button>

</div>

</div>

<!-- Second Part: Recipe List (Wider) -->

<div class="col-md-7 recipe-collection">

<h3>Recipe Collection</h3>

<div id="recipeList">

<!-- Recipe cards will be dynamically populated here -->

</div>

</div>

<!-- Third Part: Recipe of the Day -->

<div class="col-md-2">

<h3>Recipe of the Day</h3>

<div class="card" id="recipeOfTheDayCard">

<div class="card-header">

<h5 id="recipeOfTheDayName">Recipe Suggestion</h5>

</div>

<div class="card-body">

<p class="card-text" id="recipeOfTheDayDescription">Here is a recipe suggestion for you based on the most popular or recent recipe.</p>

<button class="btn btn-warning" id="recipeOfTheDayLink" onclick="getRandomRecipe()">Different Recipe</button>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Bootstrap JS and dependencies -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<script>

// Dummy Recipe Data

let recipes = [

{

name: 'Tomato Soup',

ingredients: 'Tomatoes, Salt, Water, Olive Oil',

instructions: '1. Boil tomatoes in water. 2. Add salt and olive oil. 3. Blend and serve.',

cookingTime: 15,

category: 'Vegetarian'

},

{

name: 'Grilled Chicken',

ingredients: 'Chicken, Salt, Pepper, Olive Oil',

instructions: '1. Season the chicken. 2. Grill on both sides until done.',

cookingTime: 25,

category: 'Non-Vegetarian'

},

{

name: 'Chocolate Cake',

ingredients: 'Flour, Cocoa Powder, Sugar, Eggs',

instructions: '1. Mix ingredients and bake at 350°F for 30 minutes.',

cookingTime: 30,

category: 'Dessert'

},

{

name: 'Fruit Salad',

ingredients: 'Banana, Apple, Grapes, Lemon',

instructions: '1. Chop fruits and toss with lemon juice.',

cookingTime: 10,

category: 'Vegetarian'

},

{

name: 'Scrambled Eggs',

ingredients: 'Eggs, Butter, Salt, Pepper',

instructions: '1. Whisk eggs. 2. Cook with butter in a pan.',

cookingTime: 5,

category: 'Non-Vegetarian'

}

];

// Function to display recipes

function displayRecipes(recipesToShow) {

let recipeCards = '';

recipesToShow.forEach(recipe => {

recipeCards += `

<div class="card recipe-card">

<div class="card-header">

<h5 class="card-title">${recipe.name}</h5>

</div>

<div class="card-body recipe-card-body">

<p><strong>Ingredients:</strong> ${recipe.ingredients}</p>

<p><strong>Instructions:</strong> ${recipe.instructions}</p>

<p><strong>Cooking Time:</strong> ${recipe.cookingTime} minutes</p>

<p><strong>Category:</strong> <span class="badge badge-custom">${recipe.category}</span></p>

</div>

</div>

`;

});

document.getElementById('recipeList').innerHTML = recipeCards;

// Update "Recipe of the Day" when recipes change

updateRecipeOfTheDay(recipesToShow);

}

// Function to filter recipes based on category

function filterRecipes(category) {

let filteredRecipes = [];

if (category === 'All') {

filteredRecipes = recipes;

} else {

filteredRecipes = recipes.filter(recipe => recipe.category === category);

}

displayRecipes(filteredRecipes);

}

// Function to update the "Recipe of the Day"

function updateRecipeOfTheDay(recipesToShow) {

const randomRecipe = recipesToShow[Math.floor(Math.random() \* recipesToShow.length)];

const recipeOfTheDayCard = document.getElementById('recipeOfTheDayCard');

document.getElementById('recipeOfTheDayName').innerText = randomRecipe.name;

document.getElementById('recipeOfTheDayDescription').innerText = `Ingredients: ${randomRecipe.ingredients}. Instructions: ${randomRecipe.instructions}`;

document.getElementById('recipeOfTheDayLink').href = "#";

}

// Function to get a random recipe when the "Different Recipe" button is clicked

function getRandomRecipe() {

// Get the current filtered recipes

const recipesToShow = recipes;

// Select a random recipe from the list

const randomRecipe = recipesToShow[Math.floor(Math.random() \* recipesToShow.length)];

document.getElementById('recipeOfTheDayName').innerText = randomRecipe.name;

document.getElementById('recipeOfTheDayDescription').innerText = `Ingredients: ${randomRecipe.ingredients}. Instructions: ${randomRecipe.instructions}`;

}

// Handle form submission

$('#recipeForm').on('submit', function(e) {

e.preventDefault();

const recipe = {

name: $('#recipeName').val(),

ingredients: $('#ingredients').val(),

instructions: $('#instructions').val(),

cookingTime: $('#cookingTime').val(),

category: $('#recipeCategory').val()

};

recipes.push(recipe);

// Reset the form and display the updated list of recipes

$('#recipeForm')[0].reset();

displayRecipes(recipes);

});

// Initialize the page

$(document).ready(function() {

displayRecipes(recipes); // Display all recipes initially

});

</script>

</body>

</html>

