**Login page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>DRIVING SCHOOL</title>

<link rel="stylesheet" href="style.css">

<style >

body {

font-family: sans-serif;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

background-color: #f0f0f0;

}

.container {

background-color: #fff;

padding: 30px;

border-radius: 5px;

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

text-align: center;

}

h1 {

font-size: 24px;

margin-bottom: 10px;

}

h2 {

font-size: 20px;

margin-bottom: 20px;

}

.form-group {

margin-bottom: 15px;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"],

input[type="password"] {

width: 100%;

padding: 10px;

border: 1px solid #ccc;

border-radius: 3px;

font-size: 16px;

}

button {

background-color: #333;

color: #fff;

padding: 10px 20px;

border: none;

border-radius: 3px;

cursor: pointer;

}

button:hover {

background-color: #222;

}

</style>

</head>

<body>

<div class="container">

<h1>DRIVING SCHOOL</h1>

<h2>Admin Login</h2>

<form action="#">

<div class="form-group">

<label for="username">Admin ID:</label>

<input type="text" id="username" name="username" required>

</div>

<div class="form-group">

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

</div>

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

</form>

</div>

</body>

</html>

public class LoginPageController

{

private readonly IUserService \_userService;

public LoginPageController(IUserService userService)

{

\_userService = userService;

}

public IActionResult Login(string username, string password)

{

if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password))

{

return BadRequest("Username and password cannot be empty");

}

var user = \_userService.Authenticate(username, password);

if (user != null)

{

// Login successful, logic to handle successful login (e.g., generate token)

return Ok("Login successful");

}

else

{

// Login failed, logic to handle failed login (e.g., display error message)

return Unauthorized("Invalid username or password");

}

}

}

**HOMEPAGE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>DRIVING SCHOOL Admin HomePage</title>

<link rel="stylesheet" href="style.css">

<style>

body {

font-family: sans-serif;

margin: 0;

padding: 0;

}

.container {

display: flex;

flex-direction: column;

width: 100vw;

height: 100vh;

background-color: #f0f0f0;

}

.header {

display: flex;

justify-content: space-between;

align-items: center;

padding: 10px 20px;

background-color: #333;

color: #fff;

}

.header .title {

font-size: 20px;

}

.header .logout{

margin-left: 1005px;

border: 1px solid white;

width: 60px;

height: 28px;

text-align: center;

padding-top: 13px;

font-size: 16px;

cursor: pointer;

}

.header .exit {

font-size: 16px;

cursor: pointer;

border: 1px solid;

color: whitesmoke;

width: 52px;

height: 29px;

text-align: center;

padding-top: 12px;

}

.menu {

display: flex;

/\* flex-direction: column; \*/

padding: 32px 219px;

background-color: #612f2f;

}

.menu a {

text-decoration: none;

color: #333;

margin-bottom: 5px;

padding: 52px;

border-radius: 3px;

border: 1px solid;

color: whitesmoke;

margin-left: 57px;

background-color: blue;

font-size: 16px;

font-weight: 600;

}

.menu a.active {

background-color: #ccc;

}

.content {

padding: 20px;

}

.content h2 {

text-align: center;

font-size: 26px;

margin-bottom: 10px;

}

.content p {

margin-bottom: 5px;

}

.buttons button{

padding: 15px;

margin-left: 160px;

background-color: #8fcb54;

color: whitesmoke;

font-size: 20px;

border: 2px solid whitesmoke;

}

.buttons

{

margin-top: 30px;

/\* padding: 0px 265px; \*/

margin-left: 68px;

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<div class="title">DRIVING SCHOOL</div>

<div class="logout">Logout</div>

<div class="exit">Exit</div>

</div>

<div class="menu">

<a href="#">Manage Students</a>

<a href="#">Manage Courses</a>

<a href="#">Manage Instructors</a>

</div>

<div class="content">

<h2>Manage Students</h2>

</div>

<div class="buttons">

<button>Enrolled Students</button>

<button>Registered Courses</button>

<button>Active Instructors</button>

</div>

</div>

</body>

</html>

**CLASSES**

public class Student

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string Phone { get; set; }

// Additional properties can be added as needed (e.g., address, enrolled courses)

}

public class Course

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

// Additional properties can be added as needed (e.g., duration, instructor)

}

public class Instructor

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string Phone { get; set; }

// Additional properties can be added as needed (e.g., courses assigned)

}

**SERVICES**

public interface IStudentService

{

void AddStudent(Student student);

// Additional methods for functionalities like getting students, updating student information, etc.

}

**CONTROLLER**

public class StudentController : Controller

{

private readonly IStudentService \_studentService;

public StudentController(IStudentService studentService)

{

\_studentService = studentService;

}

public IActionResult AddStudent(string firstName, string lastName, string email, string phone)

{

if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(lastName) ||

string.IsNullOrEmpty(email) || string.IsNullOrEmpty(phone))

{

return BadRequest("All fields are required");

}

var student = new Student

{

FirstName = firstName,

LastName = lastName,

Email = email,

Phone = phone

};

\_studentService.AddStudent(student);

return Ok("Student added successfully");

}

}

**Manage students**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>DRIVING SCHOOL</title>

<link rel="stylesheet" href="style.css">

<style>

body {

font-family: sans-serif;

margin: 0;

padding: 0;

}

.container {

display: flex;

flex-direction: column;

width: 100vw;

height: 100vh;

}

.header {

display: flex;

justify-content: space-between;

align-items: center;

padding: 10px 20px;

background-color: #333;

color: #fff;

}

.header .title {

display: flex;

align-items: center;

}

.header .title img {

width: 50px;

margin-right: 10px;

}

.header .menu {

flex: 1;

display: flex;

justify-content: space-evenly;

}

.header .menu a {

text-decoration: none;

color: #fff;

}

.header .menu a.active {

border-bottom: 2px solid #fff;

}

.header .logout,

.header .exit {

font-size: 16px;

cursor: pointer;

padding: 13px;

border: 1px solid whitesmoke;

margin-right: 5px;

}

.content {

flex: 1;

padding: 20px;

}

.content h2 {

font-size: 20px;

margin-bottom: 10px;

}

.student-table {

width: 100%;

border-collapse: collapse;

}

.course-table {

width: 100%;

border-collapse: collapse;

}

.course-table th,

.course-table td {

padding: 10px;

border: 1px solid #ccc;

text-align: left;

}

.course-table thead th {

background-color: #ddd;

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<div class="title">

<h1>DRIVING SCHOOL</h1>

</div>

<div class="menu">

<a href="#" class="active">Dashboard</a>

<a href="#">Manage Students</a>

<a href="#">Manage Courses</a>

<a href="#">Manage Instructors</a>

</div>

<div class="logout">Logout</div>

<div class="exit">Exit</div>

</div>

<div class="content">

<div class="course-table">

<table>

<thead>

<tr>

<th>Student ID</th>

<th>Full Name</th>

<th>Email</th>

<th>Contact</th>

<th>DOB</th>

<th>Gender</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>John Doe</td>

<td>john.doe@example.com</td>

<td>1234567890</td>

<td>2000-01-01</td>

<td>Male</td>

<td><a href="#">Edit</a> | <a href="#">Delete</a></td>

</tr>

<tr>

<td>2</td>

<td>Jane Doe</td>

<td>jane.doe@example.com</td>

<td>0987654321</td>

<td>2001-02-02</td>

<td>Female</td>

<td><a href="#">Edit</a> | <a href="#">Delete</a></td>

</tr>

<tr>

<td>3</td>

<td>Mary Smith</td>

<td>mary.smith@example.com</td>

<td>9876543210</td>

<td>2002-03-03</td>

<td>Female</td>

<td><a href="#">Edit</a> | <a href="#">Delete</a></td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</body>

</html>

public class StudentController

{

private readonly IStudentService \_studentService;

public StudentController(IStudentService studentService)

{

\_studentService = studentService;

}

public IActionResult AddStudent(string firstName, string lastName, string email, string phone)

{

if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(lastName) ||

string.IsNullOrEmpty(email) || string.IsNullOrEmpty(phone))

{

return BadRequest("All fields are required");

}

var student = new Student

{

FirstName = firstName,

LastName = lastName,

Email = email,

Phone = phone

};

\_studentService.AddStudent(student);

return Ok("Student added successfully");

}

}

**MANAGE COURSES**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>DRIVING SCHOOL</title>

<link rel="stylesheet" href="style.css">

<style>

body {

font-family: sans-serif;

margin: 0;

padding: 0;

}

.container {

display: flex;

flex-direction: column;

width: 100vw;

height: 100vh;

}

.header {

display: flex;

align-items: center;

padding: 10px 20px;

background-color: #333;

color: #fff;

}

.header .title {

display: flex;

align-items: center;

}

.header .title img {

width: 50px;

margin-right: 10px;

}

.header .menu {

flex: 1;

display: flex;

justify-content: space-evenly;

}

.header .menu a {

text-decoration: none;

color: #fff;

}

.header .menu a.active {

border-bottom: 2px solid #fff;

}

.header .logout,

.header .exit {

font-size: 16px;

cursor: pointer;

padding: 13px;

border: 1px solid whitesmoke;

margin-right: 5px;

}

.content {

flex: 1;

padding: 20px;

}

.content h2 {

font-size: 20px;

margin-bottom: 10px;

}

.course-table {

width: 100%;

border-collapse: collapse;

}

.course-table th,

.course-table td {

padding: 10px;

border: 1px solid #ccc;

text-align: left;

}

.course-table thead th {

background-color: #ddd;

}

.course-table button {

background-color: transparent;

border: none;

color: #333;

cursor: pointer;

}

.course-table button:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<div class="title">

<h1>DRIVING SCHOOL</h1>

</div>

<div class="menu">

<a href="#" class="active">Dashboard</a>

<a href="#">Manage Students</a>

<a href="#">Manage Courses</a>

<a href="#">Manage Instructors</a>

</div>

<div class="logout">Logout</div>

<div class="exit">Exit</div>

</div>

<div class="content">

<h2>Manage Courses</h2>

<div class="course-table">

<table>

<thead>

<tr>

<th>Course ID</th>

<th>Type</th>

<th>Fee</th>

<th>Delete</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Introductory</td>

<td>₹50.00</td>

<td><button>Delete</button></td>

</tr>

<tr>

<td>2</td>

<td>Standard</td>

<td>₹70.00</td>

<td><button>Delete</button></td>

</tr>

<tr>

<td>3</td>

<td>Pass Plus</td>

<td>₹90.00</td>

<td><button>Delete</button></td>

</tr>

<tr>

<td>4</td>

<td>Driving Test</td>

<td>₹100.00</td>

<td><button>Delete</button></td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</body>

</html>

public class Student

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string Phone { get; set; }

// Additional properties can be added as needed (e.g., address, enrolled courses)

}

public class Course

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public decimal Fee { get; set; } // Assuming a decimal data type for fee

// Additional properties can be added as needed (e.g., duration, instructor)

}

public class Instructor

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string Phone { get; set; }

// Additional properties can be added as needed (e.g., courses assigned)

}

**SERVICES**

public interface IStudentService

{

void AddStudent(Student student);

// Additional methods for functionalities like getting students, updating student information, etc.

}

**CONTROLLER**

public class StudentController : Controller

{

private readonly IStudentService \_studentService;

public StudentController(IStudentService studentService)

{

\_studentService = studentService;

}

public IActionResult AddStudent(string firstName, string lastName, string email, string phone)

{

if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(lastName) ||

string.IsNullOrEmpty(email) || string.IsNullOrEmpty(phone))

{

return BadRequest("All fields are required");

}

var student = new Student

{

FirstName = firstName,

LastName = lastName,

Email = email,

Phone = phone

};

\_studentService.AddStudent(student);

return Ok("Student added successfully");

}

}