CAR

VIEWS

Car

ADD car :

@model MVCCoreApp.Models.Car

@{

ViewData["Title"] = "Add Car";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h2>Add Car Details</h2>

@if (ViewBag.Message != null)

{

<div class="alert alert-success">@ViewBag.Message</div>

}

@if (ViewBag.ErrorMessage != null)

{

<div class="alert alert-danger">@ViewBag.ErrorMessage</div>

}

<**form** **asp-action**="AddCar" method="post">

<table class="table table-hover">

<tr>

<th>

<**label** **asp-for**="CarId"></**label**>

</th>

<td>

<**input** **asp-for**="CarId" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="CarModel"></**label**>

</th>

<td>

<**input** **asp-for**="CarModel" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="Color"></**label**>

</th>

<td>

<**input** **asp-for**="Color" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="TotalSeats"></**label**>

</th>

<td>

<**input** **asp-for**="TotalSeats" **type**="number" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="HasAC"></**label**>

</th>

<td>

<**input** **asp-for**="HasAC" **type**="checkbox" />

</td>

</tr>

<tr>

<th></th>

<td>

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

</td>

</tr>

</table>

</**form**>

Edit car:

@model MVCCoreApp.Models.Car

@{

ViewData["Title"] = "Edit Car";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h2>Edit Car Details</h2>

@if (ViewBag.ErrorMessage != null)

{

<div class="alert alert-danger">@ViewBag.ErrorMessage</div>

}

<**form** **asp-action**="EditCar" method="post">

<table class="table table-hover">

<tr>

<th>

<**label** **asp-for**="CarId"></**label**>

</th>

<td>

<**input** **asp-for**="CarId" class="form-control" readonly /> <!-- CarId is readonly -->

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="CarModel"></**label**>

</th>

<td>

<**input** **asp-for**="CarModel" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="Color"></**label**>

</th>

<td>

<**input** **asp-for**="Color" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="TotalSeats"></**label**>

</th>

<td>

<**input** **asp-for**="TotalSeats" **type**="number" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="HasAC"></**label**>

</th>

<td>

<**input** **asp-for**="HasAC" **type**="checkbox" />

</td>

</tr>

<tr>

<th></th>

<td>

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

<**a** **asp-action**="Index" class="btn btn-secondary">Cancel</**a**>

</td>

</tr>

</table>

</**form**>

Index car:

@model IEnumerable<MVCCoreApp.Models.Car>

@{

ViewData["Title"] = "Car List";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h2>Car List</h2>

@if (TempData["Message"] != null)

{

<div class="alert alert-success">@TempData["Message"]</div>

}

@if (TempData["ErrorMessage"] != null)

{

<div class="alert alert-danger">@TempData["ErrorMessage"]</div>

}

<**a** **asp-action**="AddCar" class="btn btn-primary mb-3">Add New Car</**a**>

<table class="table table-bordered table-hover">

<thead>

<tr>

<th>Car ID</th>

<th>Car Model</th>

<th>Color</th>

<th>Total Seats</th>

<th>Has AC</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

@foreach (var car in Model)

{

<tr>

<td>@car.CarId</td>

<td>@car.CarModel</td>

<td>@car.Color</td>

<td>@car.TotalSeats</td>

<td>@(car.HasAC ? "Yes" : "No")</td>

<td>

<**a** **asp-action**="EditCar" **asp-route-carId**="@car.CarId" class="btn btn-warning btn-sm">Edit</**a**>

<**form** **asp-action**="DeleteCar" method="post" style="display:inline">

<input type="hidden" name="carId" value="@car.CarId" />

<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this car?')">Delete</button>

</**form**>

</td>

</tr>

}

</tbody>

</table>

DRIVER

Add driver:

@model MVCCoreApp.Models.Driver

@{

ViewData["Title"] = "Add Driver";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Add Driver Details</h1>

<**form** **asp-action**="AddDriver" method="post">

<table class="table table-hover">

<tr>

<th>

<**label** **asp-for**="DriverName"></**label**>

</th>

<td>

<**input** **asp-for**="DriverName" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="MobileNo"></**label**>

</th>

<td>

<**input** **asp-for**="MobileNo" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="Age"></**label**>

</th>

<td>

<**input** **asp-for**="Age" class="form-control" />

</td>

</tr>

<tr>

<td colspan="2">

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

</td>

</tr>

</table>

</**form**>

Edit driver:

@model MVCCoreApp.Models.Driver

@{

ViewData["Title"] = "Edit Driver";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Edit Driver Details</h1>

<**form** **asp-action**="EditDriver" method="post">

<**input** **type**="hidden" **asp-for**="DriverId" />

<table class="table table-hover">

<tr>

<th>

<**label** **asp-for**="DriverName"></**label**>

</th>

<td>

<**input** **asp-for**="DriverName" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="MobileNo"></**label**>

</th>

<td>

<**input** **asp-for**="MobileNo" class="form-control" />

</td>

</tr>

<tr>

<th>

<**label** **asp-for**="Age"></**label**>

</th>

<td>

<**input** **asp-for**="Age" class="form-control" />

</td>

</tr>

<tr>

<td colspan="2">

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

</td>

</tr>

</table>

</**form**>

Index :

@model IE

numerable<MVCCoreApp.Models.Driver>

@{

ViewData["Title"] = "Driver List";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Driver List</h1>

<**a** **asp-action**="AddDriver" class="btn btn-primary">Add New Driver</**a**>

@if (!Model.Any())

{

<div class="alert alert-warning">No drivers available.</div>

}

else

{

<table class="table table-hover">

<thead>

<tr>

<th>Driver ID</th>

<th>Driver Name</th>

<th>Mobile No</th>

<th>Age</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

@foreach (var driver in Model)

{

<tr>

<td>@driver.DriverId</td>

<td>@driver.DriverName</td>

<td>@driver.MobileNo</td>

<td>@driver.Age</td>

<td>

<**a** **asp-action**="EditDriver" **asp-route-driverId**="@driver.DriverId" class="btn btn-warning btn-sm">Edit</**a**>

<**form** **asp-action**="DeleteDriver" method="post" style="display:inline">

<input type="hidden" name="driverId" value="@driver.DriverId" />

<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Are you sure you want to delete this driver?')">Delete</button>

</**form**>

</td>

</tr>

}

</tbody>

</table>

}

HOME

Index:

@{

ViewData["Title"] = "Home Page";

}

<div class="text-center">

<h1 class="display-4">Welcome</h1>

<p>Learn about <a href="https://learn.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>

</div>

Privacy:

@{

ViewData["Title"] = "Privacy Policy";

}

<h1>@ViewData["Title"]</h1>

<p>Use this page to detail your site's privacy policy.</p>

JOURNEY

@{

ViewData["Title"] = "Welcome";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Welcome</h1>

SCHEDULE

Add schedule:

@model MVCCoreApp.Models.Schedules

@{

ViewData["Title"] = "Add Schedule";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Add Schedule</h1>

<**form** **asp-action**="AddSchedule" method="post">

<table class="table table-hover">

<tr>

<th><**label** **asp-for**="JourneyDate">Journey Date</**label**></th>

<td><**input** **asp-for**="JourneyDate" class="form-control" **type**="date" /></td>

</tr>

<tr>

<th><**label** **asp-for**="CarId">Select Car</**label**></th>

<td>

<**select** **asp-for**="CarId" class="form-control" **asp-items**="ViewBag.Cars">

<**option** **value**="">-- Select a Car --</**option**>

</**select**>

</td>

</tr>

<tr>

<th><**label** **asp-for**="DriverId">Select Driver</**label**></th>

<td>

<**select** **asp-for**="DriverId" class="form-control" **asp-items**="ViewBag.Drivers">

<**option** **value**="">-- Select a Driver --</**option**>

</**select**>

</td>

</tr>

<tr>

<th><**label** **asp-for**="StartLocation">Start Location</**label**></th>

<td><**input** **asp-for**="StartLocation" class="form-control" /></td>

</tr>

<tr>

<th><**label** **asp-for**="Destination">Destination</**label**></th>

<td><**input** **asp-for**="Destination" class="form-control" /></td>

</tr>

<tr>

<th><**label** **asp-for**="TotalKmDriven">Total KM Driven</**label**></th>

<td><**input** **asp-for**="TotalKmDriven" class="form-control" /></td>

</tr>

<tr>

<th><**label** **asp-for**="RatePerKm">Rate per KM</**label**></th>

<td><**input** **asp-for**="RatePerKm" class="form-control" /></td>

</tr>

<tr>

<td colspan="2">

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

</td>

</tr>

</table>

</**form**>

@section Scripts {

<**partial** **name**="\_ValidationScriptsPartial" />

}

Delete schedule:

@model MVCCoreApp.Models.Schedules

@{

ViewData["Title"] = "Delete Schedule";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Delete Schedule</h1>

<p>Are you sure you want to delete the following schedule?</p>

<table class="table">

<tr>

<th>Journey Date</th>

<td>@Model.JourneyDate.ToString("dd-MM-yyyy")</td>

</tr>

<tr>

<th>Car</th>

<td>@Model.CarId</td>

</tr>

<tr>

<th>Driver</th>

<td>@Model.DriverId</td>

</tr>

<tr>

<th>Start Location</th>

<td>@Model.StartLocation</td>

</tr>

<tr>

<th>Destination</th>

<td>@Model.Destination</td>

</tr>

<tr>

<th>Total KM Driven</th>

<td>@Model.TotalKmDriven</td>

</tr>

<tr>

<th>Rate per KM</th>

<td>@Model.RatePerKm</td>

</tr>

</table>

<**form** **asp-action**="DeleteSchedule" method="post">

<**input** **type**="hidden" **asp-for**="ScheduleId" />

<button type="submit" class="btn btn-danger">Confirm Delete</button>

<**a** **asp-action**="Index" class="btn btn-secondary">Cancel</**a**>

</**form**>

Editschedule:

@model MVCCoreApp.Models.Schedules

@{

ViewData["Title"] = "Edit Schedule";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Edit Schedule</h1>

@if (TempData["ErrorMessage"] != null)

{

<div class="alert alert-danger">@TempData["ErrorMessage"]</div>

}

<**form** **asp-action**="EditSchedule" method="post">

<**input** **type**="hidden" **asp-for**="ScheduleId" />

<table class="table table-hover">

<tr>

<th><**label** **asp-for**="JourneyDate">Journey Date</**label**></th>

<td><**input** **asp-for**="JourneyDate" class="form-control" **type**="date" /></td>

</tr>

<tr>

<th><**label** **asp-for**="CarId">Select Car</**label**></th>

<td>

<**select** **asp-for**="CarId" class="form-control" **asp-items**="ViewBag.Cars">

<**option** **value**="">-- Select a Car --</**option**>

</**select**>

</td>

</tr>

<tr>

<th><**label** **asp-for**="DriverId">Select Driver</**label**></th>

<td>

<**select** **asp-for**="DriverId" class="form-control" **asp-items**="ViewBag.Drivers">

<**option** **value**="">-- Select a Driver --</**option**>

</**select**>

</td>

</tr>

<tr>

<th><**label** **asp-for**="StartLocation">Start Location</**label**></th>

<td><**input** **asp-for**="StartLocation" class="form-control" /></td>

</tr>

<tr>

<th><**label** **asp-for**="Destination">Destination</**label**></th>

<td><**input** **asp-for**="Destination" class="form-control" /></td>

</tr>

<tr>

<th><**label** **asp-for**="TotalKmDriven">Total KM Driven</**label**></th>

<td><**input** **asp-for**="TotalKmDriven" class="form-control" /></td>

</tr>

<tr>

<th><**label** **asp-for**="RatePerKm">Rate per KM</**label**></th>

<td><**input** **asp-for**="RatePerKm" class="form-control" /></td>

</tr>

<tr>

<td colspan="2">

<button type="submit" class="btn btn-primary">Update Schedule</button>

</td>

</tr>

</table>

</**form**>

Index:

@model IEnumerable<MVCCoreApp.Models.Schedules>

@{

ViewData["Title"] = "Schedule List";

Layout = "~/Views/Shared/JourneyLayout.cshtml";

}

<h1>Schedule List</h1>

<**a** **asp-action**="AddSchedule" class="btn btn-success">Add New Schedule</**a**>

<table class="table table-bordered table-striped">

<thead>

<tr>

<th>Journey Date</th>

<th>Car ID</th>

<th>Driver ID</th>

<th>Start Location</th>

<th>Destination</th>

<th>Total KM Driven</th>

<th>Rate Per KM</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

@foreach (var schedule in Model)

{

<tr>

<td>@schedule.JourneyDate.ToString("dd-MM-yyyy")</td>

<td>@schedule.CarId</td>

<td>@schedule.DriverId</td>

<td>@schedule.StartLocation</td>

<td>@schedule.Destination</td>

<td>@schedule.TotalKmDriven</td>

<td>@schedule.RatePerKm</td>

<td>

<**a** **asp-action**="EditSchedule" **asp-route-id**="@schedule.ScheduleId" class="btn btn-primary">Edit</**a**>

<**a** **asp-action**="DeleteSchedule" **asp-route-id**="@schedule.ScheduleId" class="btn btn-danger">Delete</**a**>

</td>

</tr>

}

</tbody>

</table>

Header:

<header class="header">

<h1 class="h1">

<**img** style="border-radius:50px;" src="~/images/car.jpeg" height="100" width="100" class="img-thumbnail" />

Online Journey Portal

</h1>

<hr style="border:1px solid green" />

<nav class="nav">

| <**a** **asp-action**="Welcome" **asp-controller**="Journey" class="bg-primary">Home</**a**> | <**a** **asp-action**="Index" **asp-controller**="Car" class="bg-primary">Cars</**a**> | <**a** **asp-action**="Index" **asp-controller**="Driver" class="bg-primary">Drivers</**a**> | <**a** **asp-action**="Index" **asp-controller**="Schedule" class="bg-primary">Schedules</**a**> |

</nav>

<hr style="border:1px solid green" />

</header>

Footer:

<footer class="footer" style="position:absolute;bottom:0px;width:100%">

<hr style="border:1px solid green" />

<p style="text-align:center;font-weight:bold">

All Rights Reserved. &copy; Copyright Cognizant 2024-2025.

</p>

</footer>

MODELS

CAR:

using System.ComponentModel.DataAnnotations;

namespace MVCCoreApp.Models

{

public class Car

{

[Key] // Primary key attribute

public string CarId { get; set; }

public string CarModel { get; set; }

public string Color { get; set; }

public int TotalSeats { get; set; }

public bool HasAC { get; set; }

}

}

DRIVER:

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace MVCCoreApp.Models

{

[Table("DriverDetails")] // Maps this model to the DriverDetails table in the database

public class Driver

{

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

public int DriverId { get; set; } // Primary Key (matches with the DriverId column in the database)

[Required(ErrorMessage = "Driver name is required")]

public string DriverName { get; set; }

[Required(ErrorMessage = "Mobile number is required")]

[MaxLength(10), RegularExpression(@"[7-9]{1}[0-9]{9}", ErrorMessage = "Invalid mobile number")]

public string MobileNo { get; set; }

public int Age { get; set; }

}

}

ERRORVIEW:

namespace MVCCoreApp.Models

{

public class ErrorViewModel

{

public string? RequestId { get; set; }

public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

}

}

JOURNEYDBCONTEXT:

using Microsoft.EntityFrameworkCore;

namespace MVCCoreApp.Models

{

public class JourneyDbContext : DbContext

{

public JourneyDbContext(DbContextOptions<JourneyDbContext> options) : base(options) { }

public DbSet<Car> Cars { get; set; }

public DbSet<Driver> Drivers { get; set; } // The `Driver` class now maps to DriverDetails

public DbSet<Login> Logins { get; internal set; }

public DbSet<Schedules> Schedules { get; set; }

}

}

LOGIN:

using System.ComponentModel.DataAnnotations;

namespace MVCCoreApp.Models

{

public class Login

{

[Key]

public string UserId { get; set; }

[Required]

public string Password { get; set; }

[Required]

public string UserType { get; set; } // Example: Admin, User, etc.

}

}

SCHEDULES:

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

using Microsoft.AspNetCore.Mvc.ModelBinding.Validation;

using MVCCoreApp.Models;

namespace MVCCoreApp.Models

{

public class Schedules

{

[Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]

public int ScheduleId { get; set; }

[Required(ErrorMessage = "Journey Date must not be left empty")]

[DataType(DataType.Date), DisplayFormat(DataFormatString = "{0:dd-MM-yyyy}")]

public DateTime JourneyDate { get; set; }

[Required(ErrorMessage = "Please select a Car")]

public string CarId { get; set; }

[Required(ErrorMessage = "Please select a Driver")]

public int DriverId { get; set; }

public string StartLocation { get; set; }

public string Destination { get; set; }

[Required(ErrorMessage = "Total KM Driven must be specified")]

public decimal TotalKmDriven { get; set; }

[Required(ErrorMessage = "Rate per KM must be specified")]

public int RatePerKm { get; set; }

[ValidateNever]

public virtual Car Car { get; set; }

[ValidateNever]

public virtual Driver Driver { get; set; }

}

}

CONTROLLERS

CARCONTROLLER:

using Microsoft.AspNetCore.Mvc;

using MVCCoreApp.Models;

namespace MVCCoreApp.Controllers

{

public class CarController : Controller

{

private readonly JourneyDbContext \_context;

public CarController(JourneyDbContext context)

{

\_context = context;

}

// \*\*GET: Car/Index (Display all cars)\*\*

public IActionResult Index()

{

var cars = \_context.Cars.ToList();

return View(cars);

}

// \*\*GET: Car/AddCar (Render the add form)\*\*

[HttpGet]

public IActionResult AddCar()

{

return View(new Car());

}

// \*\*POST: Car/AddCar (Submit form data)\*\*

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult AddCar(Car model)

{

if (ModelState.IsValid)

{

\_context.Cars.Add(model);

\_context.SaveChanges();

TempData["Message"] = "Car added successfully!";

return RedirectToAction("Index");

}

TempData["ErrorMessage"] = "Failed to add car.";

return View(model);

}

// \*\*GET: Car/EditCar (Render edit form)\*\*

[HttpGet]

public IActionResult EditCar(string carId)

{

var car = \_context.Cars.Find(carId); // Find car by ID

if (car == null)

{

TempData["ErrorMessage"] = "Car not found.";

return RedirectToAction("Index");

}

return View(car); // Pass the car to the edit form

}

// \*\*POST: Car/EditCar (Submit updated data)\*\*

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult EditCar(Car model)

{

if (ModelState.IsValid)

{

var car = \_context.Cars.Find(model.CarId);

if (car != null)

{

car.CarModel = model.CarModel;

car.Color = model.Color;

car.TotalSeats = model.TotalSeats;

car.HasAC = model.HasAC;

\_context.SaveChanges(); // Save the updated car details

TempData["Message"] = "Car updated successfully!";

return RedirectToAction("Index");

}

TempData["ErrorMessage"] = "Car not found.";

}

return View(model);

}

}

}

DRIVER CONTROLLER:

using Microsoft.AspNetCore.Mvc;

using MVCCoreApp.Models;

using System.Linq;

namespace MVCCoreApp.Controllers

{

public class DriverController : Controller

{

private readonly JourneyDbContext \_context;

public DriverController(JourneyDbContext context)

{

\_context = context;

}

// \*\*GET: Driver/Index (Display all drivers)\*\*

public IActionResult Index()

{

var drivers = \_context.Drivers.ToList(); // Fetch all drivers from DriverDetails table

if (drivers == null || !drivers.Any())

{

drivers = new List<Driver>();

}

return View(drivers);

}

// \*\*GET: Driver/AddDriver\*\*

[HttpGet]

public IActionResult AddDriver()

{

return View(new Driver());

}

// \*\*POST: Driver/AddDriver\*\*

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult AddDriver(Driver model)

{

if (ModelState.IsValid)

{

\_context.Drivers.Add(model);

\_context.SaveChanges();

TempData["Message"] = "Driver added successfully!";

return RedirectToAction("Index");

}

TempData["ErrorMessage"] = "Failed to add driver.";

return View(model);

}

// \*\*GET: Driver/EditDriver\*\*

[HttpGet]

public IActionResult EditDriver(int driverId)

{

var driver = \_context.Drivers.Find(driverId);

if (driver == null)

{

TempData["ErrorMessage"] = "Driver not found.";

return RedirectToAction("Index");

}

return View(driver);

}

// \*\*POST: Driver/EditDriver\*\*

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult EditDriver(Driver model)

{

if (ModelState.IsValid)

{

var driver = \_context.Drivers.Find(model.DriverId);

if (driver != null)

{

driver.DriverName = model.DriverName;

driver.MobileNo = model.MobileNo;

driver.Age = model.Age;

\_context.SaveChanges();

TempData["Message"] = "Driver updated successfully!";

return RedirectToAction("Index");

}

TempData["ErrorMessage"] = "Driver not found.";

}

return View(model);

}

// \*\*POST: Driver/DeleteDriver\*\*

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult DeleteDriver(int driverId)

{

var driver = \_context.Drivers.Find(driverId);

if (driver != null)

{

\_context.Drivers.Remove(driver);

\_context.SaveChanges();

TempData["Message"] = "Driver deleted successfully!";

}

else

{

TempData["ErrorMessage"] = "Driver not found.";

}

return RedirectToAction("Index");

}

}

}

HOME CONTOLLER:

using Microsoft.AspNetCore.Mvc;

using MVCCoreApp.Models;

using System.Diagnostics;

namespace MVCCoreApp.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

JOURNEY CONTROLLER:

using Microsoft.AspNetCore.Mvc;

namespace MVCCoreApp.Controllers

{

public class JourneyController : Controller

{

public IActionResult Welcome()

{

return View();

}

}

}

SCHEDULE CONTROLLER:

using Microsoft.AspNetCore.Mvc;

using MVCCoreApp.Models;

using System.Linq;

using Microsoft.EntityFrameworkCore;

using Microsoft.AspNetCore.Mvc.Rendering;

using System.Security.Policy;

namespace MVCCoreApp.Controllers

{

public class ScheduleController : Controller

{

private readonly JourneyDbContext \_context;

public ScheduleController(JourneyDbContext context)

{

\_context = context;

}

// \*\*GET: Schedule/Index (Display all schedules)\*\*

public IActionResult Index()

{

var schedules = \_context.Schedules

.Include(s => s.Car)

.Include(s => s.Driver)

.ToList();

return View(schedules);

}

[HttpGet]

public IActionResult AddSchedule()

{

ViewBag.Cars = new SelectList(\_context.Cars, "CarId", "CarModel");

ViewBag.Drivers = new SelectList(\_context.Drivers, "DriverId", "DriverName");

return View();

}

[HttpPost]

public IActionResult AddSchedule(Schedules model)

{

if (ModelState.IsValid)

{

try

{

\_context.Schedules.Add(model); // Only add IDs and primitive fields

\_context.SaveChanges(); // Save to the database

TempData["SuccessMessage"] = "Schedule added successfully!";

return RedirectToAction("Index");

}

catch (Exception ex)

{

TempData["ErrorMessage"] = $"Error: {ex.Message}";

return View(model);

}

}

// If invalid, re-populate the ViewBag for dropdowns

ViewBag.Cars = new SelectList(\_context.Cars, "CarId", "CarModel");

ViewBag.Drivers = new SelectList(\_context.Drivers, "DriverId", "DriverName");

TempData["ErrorMessage"] = "Please fill all required fields.";

return View(model);

}

// GET: Edit Schedule

[HttpGet]

public IActionResult EditSchedule(int id)

{

var schedule = \_context.Schedules.Find(id);

if (schedule == null)

{

return NotFound();

}

// Populate dropdowns

ViewBag.Cars = new SelectList(\_context.Cars.ToList(), "CarId", "CarModel");

ViewBag.Drivers = new SelectList(\_context.Drivers.ToList(), "DriverId", "DriverName");

return View(schedule);

}

// POST: Edit Schedule

[HttpPost]

public IActionResult EditSchedule(Schedules model)

{

if (!ModelState.IsValid)

{

// Populate dropdowns in case of validation failure

ViewBag.Cars = new SelectList(\_context.Cars.ToList(), "CarId", "CarModel");

ViewBag.Drivers = new SelectList(\_context.Drivers.ToList(), "DriverId", "DriverName");

return View(model);

}

var scheduleToUpdate = \_context.Schedules.Find(model.ScheduleId);

if (scheduleToUpdate != null)

{

scheduleToUpdate.JourneyDate = model.JourneyDate;

scheduleToUpdate.CarId = model.CarId;

scheduleToUpdate.DriverId = model.DriverId;

scheduleToUpdate.StartLocation = model.StartLocation;

scheduleToUpdate.Destination = model.Destination;

scheduleToUpdate.TotalKmDriven = model.TotalKmDriven;

scheduleToUpdate.RatePerKm = model.RatePerKm;

\_context.Schedules.Update(scheduleToUpdate);

\_context.SaveChanges();

TempData["SuccessMessage"] = "Schedule updated successfully!";

return RedirectToAction("Index");

}

return NotFound();

}

// GET: Delete Schedule

[HttpGet]

public IActionResult DeleteSchedule(int id)

{

var schedule = \_context.Schedules.Find(id);

if (schedule == null)

{

return NotFound();

}

return View(schedule);

}

// POST: Delete Schedule

[HttpPost, ActionName("DeleteSchedule")]

public IActionResult ConfirmDeleteSchedule(int id)

{

var scheduleToDelete = \_context.Schedules.Find(id);

if (scheduleToDelete != null)

{

\_context.Schedules.Remove(scheduleToDelete);

\_context.SaveChanges();

TempData["SuccessMessage"] = "Schedule deleted successfully!";

return RedirectToAction("Index");

}

return NotFound();

}

}

}

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated