Auto Insurance System – Complete Line-by-Line Walkthrough

Each file is included with beginner-friendly inline comments per line, followed by an end-to-end flow summary.

# Auto-Insurance/AutoInsuranceSystemAPI/AutoInsuranceSystemAPI.csproj

0001: <Project Sdk="Microsoft.NET.Sdk.Web"> // markup element

0002:

0003: <PropertyGroup> // markup element

0004: <TargetFramework>net9.0</TargetFramework> // markup element

0005: <Nullable>enable</Nullable> // markup element

0006: <ImplicitUsings>enable</ImplicitUsings> // markup element

0007: </PropertyGroup> // markup element

0008:

0009: <ItemGroup> // markup element

0010: <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="9.0.5" /> // markup element

0011: <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.0" /> // markup element

0012: <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" /> // markup element

0013: </ItemGroup> // markup element

0014:

0015: </Project> // markup element

# Auto-Insurance/AutoInsuranceSystemAPI/Controllers/PoliciesController.cs

0001: using AutoInsuranceSystemAPI.Models; // import namespace

0002: using AutoInsuranceSystemAPI.Services; // import namespace

0003: using Microsoft.AspNetCore.Mvc; // import namespace

0004:

0005: namespace AutoInsuranceSystemAPI.Controllers // declare namespace

0006: { // statement

0007: [ApiController] // attribute/annotation

0008: [Route("api/admin/policies")] // attribute/annotation

0009: public class PoliciesController : ControllerBase // class declaration

0010: { // statement

0011: private readonly IPolicyService policyService; // member or method declaration

0012:

0013: public PoliciesController(IPolicyService policyService) // member or method declaration

0014: { // statement

0015: this.policyService = policyService; // statement

0016: } // statement

0017:

0018: [HttpGet] // attribute/annotation

0019: public async Task<ActionResult<IEnumerable<Policy>>> GetAll() // member or method declaration

0020: { // statement

0021: var items = await policyService.GetAllAsync(); // local variable declaration

0022: return Ok(items); // return value

0023: } // statement

0024:

0025: [HttpGet("{id:int}")] // attribute/annotation

0026: public async Task<ActionResult<Policy>> GetById(int id) // member or method declaration

0027: { // statement

0028: var policy = await policyService.GetByIdAsync(id); // local variable declaration

0029: if (policy == null) return NotFound(); // conditional check

0030: return Ok(policy); // return value

0031: } // statement

0032:

0033: [HttpPost] // attribute/annotation

0034: public async Task<ActionResult<Policy>> Create([FromBody] Policy policy) // member or method declaration

0035: { // statement

0036: var created = await policyService.CreateAsync(policy); // local variable declaration

0037: return CreatedAtAction(nameof(GetById), new { id = created.PolicyId }, created); // return value

0038: } // statement

0039:

0040: [HttpPut("{id:int}")] // attribute/annotation

0041: public async Task<IActionResult> Update(int id, [FromBody] Policy policy) // member or method declaration

0042: { // statement

0043: var ok = await policyService.UpdateAsync(id, policy); // local variable declaration

0044: if (!ok) return NotFound(); // conditional check

0045: return NoContent(); // return value

0046: } // statement

0047:

0048: [HttpDelete("{id:int}")] // attribute/annotation

0049: public async Task<IActionResult> Delete(int id) // member or method declaration

0050: { // statement

0051: var ok = await policyService.DeleteAsync(id); // local variable declaration

0052: if (!ok) return NotFound(); // conditional check

0053: return NoContent(); // return value

0054: } // statement

0055: } // statement

0056: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/Controllers/WeatherForecastController.cs

0001: using Microsoft.AspNetCore.Mvc; // import namespace

0002:

0003: namespace AutoInsuranceSystemAPI.Controllers // declare namespace

0004: { // statement

0005: [ApiController] // attribute/annotation

0006: [Route("[controller]")] // attribute/annotation

0007: public class WeatherForecastController : ControllerBase // class declaration

0008: { // statement

0009: private static readonly string[] Summaries = new[] // member or method declaration

0010: { // statement

0011: "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot", "Sweltering", "Scorching" // statement

0012: }; // statement

0013:

0014: private readonly ILogger<WeatherForecastController> \_logger; // member or method declaration

0015:

0016: public WeatherForecastController(ILogger<WeatherForecastController> logger) // member or method declaration

0017: { // statement

0018: \_logger = logger; // statement

0019: } // statement

0020:

0021: [HttpGet(Name = "GetWeatherForecast")] // attribute/annotation

0022: public IEnumerable<WeatherForecast> Get() // member or method declaration

0023: { // statement

0024: return Enumerable.Range(1, 5).Select(index => new WeatherForecast // return value

0025: { // statement

0026: Date = DateOnly.FromDateTime(DateTime.Now.AddDays(index)), // statement

0027: TemperatureC = Random.Shared.Next(-20, 55), // statement

0028: Summary = Summaries[Random.Shared.Next(Summaries.Length)] // statement

0029: }) // statement

0030: .ToArray(); // statement

0031: } // statement

0032: } // statement

0033: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/Data/AutoInsuranceApiDbContext.cs

0001: using AutoInsuranceSystemAPI.Models; // import namespace

0002: using Microsoft.EntityFrameworkCore; // import namespace

0003:

0004: namespace AutoInsuranceSystemAPI.Data // declare namespace

0005: { // statement

0006: public class AutoInsuranceApiDbContext : DbContext // class declaration

0007: { // statement

0008: public AutoInsuranceApiDbContext(DbContextOptions<AutoInsuranceApiDbContext> options) : base(options) { } // member or method declaration

0009:

0010: public virtual DbSet<Policy> Policy { get; set; } // member or method declaration

0011: } // statement

0012: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/Models/Policy.cs

0001: using System.ComponentModel.DataAnnotations; // import namespace

0002: using System.ComponentModel.DataAnnotations.Schema; // import namespace

0003:

0004: namespace AutoInsuranceSystemAPI.Models // declare namespace

0005: { // statement

0006: [Table("Policy")] // attribute/annotation

0007: public class Policy // class declaration

0008: { // statement

0009: [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)] // attribute/annotation

0010: public int PolicyId { get; set; } // member or method declaration

0011:

0012: [Required, MaxLength(50)] // attribute/annotation

0013: public string PolicyNumber { get; set; } // member or method declaration

0014:

0015: [Required, MaxLength(200)] // attribute/annotation

0016: public string VehicleDetails { get; set; } // member or method declaration

0017:

0018: [Required] // attribute/annotation

0019: [Column(TypeName = "decimal(10,2)")] // attribute/annotation

0020: public decimal CoverageAmount { get; set; } // member or method declaration

0021:

0022: [Required, MaxLength(50)] // attribute/annotation

0023: public string CoverageType { get; set; } // member or method declaration

0024:

0025: [Required] // attribute/annotation

0026: [Column(TypeName = "decimal(10,2)")] // attribute/annotation

0027: public decimal PremiumAmount { get; set; } // member or method declaration

0028:

0029: [Required] // attribute/annotation

0030: public DateTime StartDate { get; set; } // member or method declaration

0031:

0032: [Required] // attribute/annotation

0033: public DateTime EndDate { get; set; } // member or method declaration

0034:

0035: [Required] // attribute/annotation

0036: public PolicyStatus PolicyStatus { get; set; } // member or method declaration

0037: } // statement

0038:

0039: public enum PolicyStatus // member or method declaration

0040: { // statement

0041: ACTIVE, // statement

0042: INACTIVE, // statement

0043: RENEWED // statement

0044: } // statement

0045: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/Program.cs

0001: using Microsoft.EntityFrameworkCore; // import namespace

0002:

0003: var builder = WebApplication.CreateBuilder(args); // local variable declaration

0004:

0005: // Add services to the container. // comment

0006:

0007: builder.Services.AddControllers(); // statement

0008:

0009: builder.Services.AddEndpointsApiExplorer(); // statement

0010: builder.Services.AddSwaggerGen(options => // statement

0011: { // statement

0012: options.SwaggerDoc("v1", new Microsoft.OpenApi.Models.OpenApiInfo // statement

0013: { // statement

0014: Title = "Auto Insurance Admin API", // statement

0015: Version = "v1" // statement

0016: }); // statement

0017: }); // statement

0018:

0019: builder.Services.AddDbContext<AutoInsuranceSystemAPI.Data.AutoInsuranceApiDbContext>(options => // statement

0020: { // statement

0021: options.UseSqlServer(builder.Configuration.GetConnectionString("SqlCon")); // statement

0022: }); // statement

0023:

0024: builder.Services.AddScoped<AutoInsuranceSystemAPI.Services.IPolicyService, AutoInsuranceSystemAPI.Services.PolicyService>(); // statement

0025:

0026: var app = builder.Build(); // local variable declaration

0027:

0028: // Configure the HTTP request pipeline. // comment

0029: app.UseSwagger(); // statement

0030: app.UseSwaggerUI(c => // statement

0031: { // statement

0032: c.SwaggerEndpoint("/swagger/v1/swagger.json", "Auto Insurance Admin API v1"); // statement

0033: }); // statement

0034:

0035: app.UseHttpsRedirection(); // statement

0036:

0037: app.UseAuthorization(); // statement

0038:

0039: app.MapControllers(); // statement

0040:

0041: app.Run(); // statement

# Auto-Insurance/AutoInsuranceSystemAPI/Properties/launchSettings.json

0001: ﻿{ // statement

0002: "$schema": "https://json.schemastore.org/launchsettings.json", // statement

0003: "profiles": { // statement

0004: "http": { // statement

0005: "commandName": "Project", // statement

0006: "dotnetRunMessages": true, // statement

0007: "launchBrowser": false, // statement

0008: "applicationUrl": "http://localhost:5241", // statement

0009: "environmentVariables": { // statement

0010: "ASPNETCORE\_ENVIRONMENT": "Development" // statement

0011: } // statement

0012: }, // statement

0013: "https": { // statement

0014: "commandName": "Project", // statement

0015: "dotnetRunMessages": true, // statement

0016: "launchBrowser": false, // statement

0017: "applicationUrl": "https://localhost:7227;http://localhost:5241", // statement

0018: "environmentVariables": { // statement

0019: "ASPNETCORE\_ENVIRONMENT": "Development" // statement

0020: } // statement

0021: } // statement

0022: } // statement

0023: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/Services/IPolicyService.cs

0001: using AutoInsuranceSystemAPI.Models; // import namespace

0002:

0003: namespace AutoInsuranceSystemAPI.Services // declare namespace

0004: { // statement

0005: public interface IPolicyService // interface declaration

0006: { // statement

0007: Task<List<Policy>> GetAllAsync(); // statement

0008: Task<Policy?> GetByIdAsync(int id); // statement

0009: Task<Policy> CreateAsync(Policy policy); // statement

0010: Task<bool> UpdateAsync(int id, Policy policy); // statement

0011: Task<bool> DeleteAsync(int id); // statement

0012: } // statement

0013: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/Services/PolicyService.cs

0001: using AutoInsuranceSystemAPI.Data; // import namespace

0002: using AutoInsuranceSystemAPI.Models; // import namespace

0003: using Microsoft.EntityFrameworkCore; // import namespace

0004:

0005: namespace AutoInsuranceSystemAPI.Services // declare namespace

0006: { // statement

0007: public class PolicyService : IPolicyService // class declaration

0008: { // statement

0009: private readonly AutoInsuranceApiDbContext dbContext; // member or method declaration

0010: public PolicyService(AutoInsuranceApiDbContext dbContext) // member or method declaration

0011: { // statement

0012: this.dbContext = dbContext; // statement

0013: } // statement

0014:

0015: public async Task<List<Policy>> GetAllAsync() // member or method declaration

0016: { // statement

0017: return await dbContext.Policy.AsNoTracking().ToListAsync(); // return value

0018: } // statement

0019:

0020: public async Task<Policy?> GetByIdAsync(int id) // member or method declaration

0021: { // statement

0022: return await dbContext.Policy.AsNoTracking().FirstOrDefaultAsync(p => p.PolicyId == id); // return value

0023: } // statement

0024:

0025: public async Task<Policy> CreateAsync(Policy policy) // member or method declaration

0026: { // statement

0027: policy.PolicyId = 0; // statement

0028: dbContext.Policy.Add(policy); // statement

0029: await dbContext.SaveChangesAsync(); // statement

0030: return policy; // return value

0031: } // statement

0032:

0033: public async Task<bool> UpdateAsync(int id, Policy policy) // member or method declaration

0034: { // statement

0035: var existing = await dbContext.Policy.FirstOrDefaultAsync(p => p.PolicyId == id); // local variable declaration

0036: if (existing == null) return false; // conditional check

0037:

0038: existing.PolicyNumber = policy.PolicyNumber; // statement

0039: existing.VehicleDetails = policy.VehicleDetails; // statement

0040: existing.CoverageAmount = policy.CoverageAmount; // statement

0041: existing.CoverageType = policy.CoverageType; // statement

0042: existing.PremiumAmount = policy.PremiumAmount; // statement

0043: existing.StartDate = policy.StartDate; // statement

0044: existing.EndDate = policy.EndDate; // statement

0045: existing.PolicyStatus = policy.PolicyStatus; // statement

0046:

0047: await dbContext.SaveChangesAsync(); // statement

0048: return true; // return value

0049: } // statement

0050:

0051: public async Task<bool> DeleteAsync(int id) // member or method declaration

0052: { // statement

0053: var existing = await dbContext.Policy.FirstOrDefaultAsync(p => p.PolicyId == id); // local variable declaration

0054: if (existing == null) return false; // conditional check

0055: dbContext.Policy.Remove(existing); // statement

0056: await dbContext.SaveChangesAsync(); // statement

0057: return true; // return value

0058: } // statement

0059: } // statement

0060: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/WeatherForecast.cs

0001: namespace AutoInsuranceSystemAPI // declare namespace

0002: { // statement

0003: public class WeatherForecast // class declaration

0004: { // statement

0005: public DateOnly Date { get; set; } // member or method declaration

0006:

0007: public int TemperatureC { get; set; } // member or method declaration

0008:

0009: public int TemperatureF => 32 + (int)(TemperatureC / 0.5556); // member or method declaration

0010:

0011: public string? Summary { get; set; } // member or method declaration

0012: } // statement

0013: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/appsettings.Development.json

0001: { // statement

0002: "Logging": { // statement

0003: "LogLevel": { // statement

0004: "Default": "Information", // statement

0005: "Microsoft.AspNetCore": "Warning" // statement

0006: } // statement

0007: } // statement

0008: } // statement

# Auto-Insurance/AutoInsuranceSystemAPI/appsettings.json

0001: { // statement

0002: "Logging": { // statement

0003: "LogLevel": { // statement

0004: "Default": "Information", // statement

0005: "Microsoft.AspNetCore": "Warning" // statement

0006: } // statement

0007: }, // statement

0008: "AllowedHosts": "\*", // statement

0009: "ConnectionStrings": { // statement

0010: "SqlCon": "Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=AutoInsuranceSystemManagementDB;Integrated Security=True;TrustServerCertificate=True" // statement

0011: }, // statement

0012: "ApiSettings": { // statement

0013: "AdminApiKey": "changeme-admin-key" // statement

0014: } // statement

0015: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Auto\_Insurance\_System.csproj

0001: <Project Sdk="Microsoft.NET.Sdk.Web"> // markup element

0002:

0003: <PropertyGroup> // markup element

0004: <TargetFramework>net9.0</TargetFramework> // markup element

0005: <Nullable>enable</Nullable> // markup element

0006: <ImplicitUsings>enable</ImplicitUsings> // markup element

0007: </PropertyGroup> // markup element

0008:

0009: <ItemGroup> // markup element

0010: <PackageReference Include="Microsoft.EntityFrameworkCore" Version="9.0.7" /> // markup element

0011: <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.7" /> // markup element

0012: <PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="9.0.7"> // markup element

0013: <PrivateAssets>all</PrivateAssets> // markup element

0014: <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets> // markup element

0015: </PackageReference> // markup element

0016: <PackageReference Include="Microsoft.VisualStudio.Web.CodeGeneration.Design" Version="9.0.0" /> // markup element

0017: <PackageReference Include="System.Configuration.ConfigurationManager" Version="9.0.7" /> // markup element

0018: </ItemGroup> // markup element

0019:

0020: </Project> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Controllers/AdminController.cs

0001: using Auto\_Insurance\_System.Data; // import namespace

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Models; // import namespace

0004: using Microsoft.AspNetCore.Mvc; // import namespace

0005: using Microsoft.EntityFrameworkCore; // import namespace

0006: using Auto\_Insurance\_System.Filters; // import namespace

0007:

0008: namespace Auto\_Insurance\_System.Controllers // declare namespace

0009: { // statement

0010: [RoleAuthorizationFilter("ADMIN")] // attribute/annotation

0011: public class AdminController : Controller // class declaration

0012: { // statement

0013: private readonly IUsersService usersService; // member or method declaration

0014: private readonly IPolicyService policyService; // member or method declaration

0015: private readonly IClaimService claimService; // member or method declaration

0016: private readonly ISupportTicketService supportTicketService; // member or method declaration

0017: private readonly AutoInsuranceDbContext dbContext; // member or method declaration

0018:

0019: public AdminController( // member or method declaration

0020: IUsersService usersService, // statement

0021: IPolicyService policyService, // statement

0022: IClaimService claimService, // statement

0023: ISupportTicketService supportTicketService, // statement

0024: AutoInsuranceDbContext dbContext) // statement

0025: { // statement

0026: this.usersService = usersService; // statement

0027: this.policyService = policyService; // statement

0028: this.claimService = claimService; // statement

0029: this.supportTicketService = supportTicketService; // statement

0030: this.dbContext = dbContext; // statement

0031: } // statement

0032:

0033: private bool IsAdmin() // member or method declaration

0034: { // statement

0035: return HttpContext.Session.GetString("UserRole") == "ADMIN"; // return value

0036: } // statement

0037:

0038: private void PopulateDashboardStats() // member or method declaration

0039: { // statement

0040: var policies = policyService.GetAllPolicies(); // local variable declaration

0041: var claims = claimService.GetAllClaims(); // local variable declaration

0042: var tickets = supportTicketService.GetAllTickects(); // local variable declaration

0043: var users = usersService.GetAllUsers(); // local variable declaration

0044: var payments = dbContext.Payments.AsNoTracking().ToList(); // local variable declaration

0045:

0046: ViewBag.TotalPolicies = policies.Count; // statement

0047: ViewBag.ActivePolicies = policies.Count(p => p.PolicyStatus == PolicyStatus.ACTIVE); // statement

0048: ViewBag.PendingClaims = claims.Count(c => c.ClaimStatus == ClaimStatus.OPEN); // statement

0049: ViewBag.ApprovedClaims = claims.Count(c => c.ClaimStatus == ClaimStatus.APPROVED); // statement

0050: ViewBag.OpenTickets = tickets.Count(t => t.TicketStatus == TicketStatus.OPEN); // statement

0051: ViewBag.TotalUsers = users.Count; // statement

0052:

0053: var now = DateTime.Now; // local variable declaration

0054: ViewBag.RevenueThisMonth = payments // statement

0055: .Where(p => p.PaymentStatus == PaymentStatus.SUCCESS && p.PaymentDate.Month == now.Month && p.PaymentDate.Year == now.Year) // statement

0056: .Sum(p => (decimal?)p.PaymentAmount) ?? 0m; // statement

0057: } // statement

0058:

0059: public IActionResult Index() // member or method declaration

0060: { // statement

0061: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0062: PopulateDashboardStats(); // statement

0063: return View(); // return value

0064: } // statement

0065:

0066: public IActionResult Console() // member or method declaration

0067: { // statement

0068: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0069: PopulateDashboardStats(); // statement

0070: return View(); // return value

0071: } // statement

0072:

0073: public IActionResult UsersHub() // member or method declaration

0074: { // statement

0075: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0076: PopulateDashboardStats(); // statement

0077: var users = usersService.GetAllUsers(); // local variable declaration

0078: return View(users); // return value

0079: } // statement

0080:

0081: public IActionResult Users() // member or method declaration

0082: { // statement

0083: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0084: PopulateDashboardStats(); // statement

0085: var users = usersService.GetAllUsers(); // local variable declaration

0086: return View(users); // return value

0087: } // statement

0088:

0089: [HttpGet] // attribute/annotation

0090: public IActionResult UserCreate() // member or method declaration

0091: { // statement

0092: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0093: PopulateDashboardStats(); // statement

0094: return View(new Users()); // return value

0095: } // statement

0096:

0097: [HttpPost] // attribute/annotation

0098: public IActionResult UserCreate(Users model) // member or method declaration

0099: { // statement

0100: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0101: PopulateDashboardStats(); // statement

0102: if (!ModelState.IsValid) // conditional check

0103: { // statement

0104: return View(model); // return value

0105: } // statement

0106: var created = usersService.RegisterUser(model); // local variable declaration

0107: if (!created) // conditional check

0108: { // statement

0109: ViewBag.Error = "Failed to create user."; // statement

0110: return View(model); // return value

0111: } // statement

0112: TempData["Success"] = "User created successfully."; // statement

0113: return RedirectToAction("UserCreate"); // return value

0114: } // statement

0115:

0116: [HttpGet] // attribute/annotation

0117: public IActionResult UserUpdate() // member or method declaration

0118: { // statement

0119: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0120: PopulateDashboardStats(); // statement

0121: return View(model: null); // return value

0122: } // statement

0123:

0124: [HttpPost] // attribute/annotation

0125: public IActionResult UserUpdate(Users model) // member or method declaration

0126: { // statement

0127: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0128: PopulateDashboardStats(); // statement

0129: if (model.UserId <= 0) // conditional check

0130: { // statement

0131: ViewBag.Error = "UserId is required for update."; // statement

0132: return View(model); // return value

0133: } // statement

0134: var existing = dbContext.Users.Find(model.UserId); // local variable declaration

0135: if (existing == null) // conditional check

0136: { // statement

0137: ViewBag.Error = $"No user found with id {model.UserId}"; // statement

0138: return View(model); // return value

0139: } // statement

0140: existing.Email = model.Email; // statement

0141: existing.Username = model.Username; // statement

0142: existing.Role = model.Role; // statement

0143: if (!string.IsNullOrWhiteSpace(model.Password)) existing.Password = model.Password; // conditional check

0144: dbContext.SaveChanges(); // statement

0145: TempData["Success"] = "User updated successfully."; // statement

0146: return RedirectToAction("UserUpdate"); // return value

0147: } // statement

0148:

0149: [HttpGet] // attribute/annotation

0150: public IActionResult UserDelete() // member or method declaration

0151: { // statement

0152: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0153: PopulateDashboardStats(); // statement

0154: return View(); // return value

0155: } // statement

0156:

0157: [HttpPost] // attribute/annotation

0158: public IActionResult UserDelete(int userId) // member or method declaration

0159: { // statement

0160: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0161: PopulateDashboardStats(); // statement

0162: var existing = dbContext.Users.Find(userId); // local variable declaration

0163: if (existing == null) // conditional check

0164: { // statement

0165: ViewBag.Error = $"No user found with id {userId}"; // statement

0166: return View(); // return value

0167: } // statement

0168: dbContext.Users.Remove(existing); // statement

0169: dbContext.SaveChanges(); // statement

0170: TempData["Success"] = "User deleted successfully."; // statement

0171: return RedirectToAction("UserDelete"); // return value

0172: } // statement

0173:

0174: // Policies hub + pages // comment

0175: public IActionResult Policies() // member or method declaration

0176: { // statement

0177: return RedirectToAction("PoliciesHub"); // return value

0178: } // statement

0179:

0180: public IActionResult PoliciesHub() // member or method declaration

0181: { // statement

0182: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0183: PopulateDashboardStats(); // statement

0184: return View(); // return value

0185: } // statement

0186:

0187: public IActionResult PoliciesAll() // member or method declaration

0188: { // statement

0189: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0190: PopulateDashboardStats(); // statement

0191: var list = policyService.GetAllPolicies().OrderBy(p => p.PolicyId).ToList(); // local variable declaration

0192: return View(list); // return value

0193: } // statement

0194:

0195: [HttpGet] // attribute/annotation

0196: public IActionResult PolicyCreate() // member or method declaration

0197: { // statement

0198: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0199: return NotFound(); // return value

0200: } // statement

0201:

0202: [HttpPost] // attribute/annotation

0203: public IActionResult PolicyCreate(Policy model) // member or method declaration

0204: { // statement

0205: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0206: return NotFound(); // return value

0207: } // statement

0208:

0209: [HttpGet] // attribute/annotation

0210: public IActionResult PolicyDetails() // member or method declaration

0211: { // statement

0212: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0213: PopulateDashboardStats(); // statement

0214: return View(model: null); // return value

0215: } // statement

0216:

0217: [HttpPost] // attribute/annotation

0218: public IActionResult PolicyDetails(int policyId) // member or method declaration

0219: { // statement

0220: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0221: PopulateDashboardStats(); // statement

0222: var policy = policyService.GetPolicyById(policyId.ToString()); // local variable declaration

0223: if (policy == null) ViewBag.Error = $"No policy found with id {policyId}"; // conditional check

0224: return View(policy); // return value

0225: } // statement

0226:

0227: [HttpGet] // attribute/annotation

0228: public IActionResult PolicyUpdate() // member or method declaration

0229: { // statement

0230: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0231: PopulateDashboardStats(); // statement

0232: return View(model: null); // return value

0233: } // statement

0234:

0235: [HttpPost] // attribute/annotation

0236: public IActionResult PolicyUpdate(Policy model) // member or method declaration

0237: { // statement

0238: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0239: PopulateDashboardStats(); // statement

0240: if (model.PolicyId <= 0) // conditional check

0241: { // statement

0242: ViewBag.Error = "PolicyId is required for update."; // statement

0243: return View(model); // return value

0244: } // statement

0245: var ok = policyService.UpdatePolicy(model); // local variable declaration

0246: if (ok) // conditional check

0247: { // statement

0248: TempData["Success"] = "Policy updated successfully."; // statement

0249: return RedirectToAction("PolicyUpdate"); // return value

0250: } // statement

0251: ViewBag.Error = "Failed to update policy."; // statement

0252: return View(model); // return value

0253: } // statement

0254:

0255: [HttpGet] // attribute/annotation

0256: public IActionResult PolicyDelete() // member or method declaration

0257: { // statement

0258: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0259: PopulateDashboardStats(); // statement

0260: return View(); // return value

0261: } // statement

0262:

0263: [HttpPost] // attribute/annotation

0264: public IActionResult PolicyDelete(int policyId) // member or method declaration

0265: { // statement

0266: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0267: PopulateDashboardStats(); // statement

0268: var ok = policyService.DeletePolicy(policyId.ToString()); // local variable declaration

0269: if (ok) // conditional check

0270: { // statement

0271: TempData["Success"] = "Policy deleted successfully."; // statement

0272: return RedirectToAction("PolicyDelete"); // return value

0273: } // statement

0274: ViewBag.Error = "Failed to delete policy."; // statement

0275: return View(); // return value

0276: } // statement

0277:

0278: // Claims hub and pages // comment

0279: public IActionResult Claims() // member or method declaration

0280: { // statement

0281: return RedirectToAction("ClaimsHub"); // return value

0282: } // statement

0283:

0284: public IActionResult ClaimsHub() // member or method declaration

0285: { // statement

0286: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0287: PopulateDashboardStats(); // statement

0288: return View(); // return value

0289: } // statement

0290:

0291: public IActionResult ClaimsAll() // member or method declaration

0292: { // statement

0293: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0294: PopulateDashboardStats(); // statement

0295: var claims = claimService.GetAllClaims().OrderBy(c => c.ClaimDate).ToList(); // local variable declaration

0296: return View(claims); // return value

0297: } // statement

0298:

0299: [HttpGet] // attribute/annotation

0300: public IActionResult ClaimDetails() // member or method declaration

0301: { // statement

0302: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0303: PopulateDashboardStats(); // statement

0304: return View(model: null); // return value

0305: } // statement

0306:

0307: [HttpPost] // attribute/annotation

0308: public IActionResult ClaimDetails(int claimId) // member or method declaration

0309: { // statement

0310: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0311: PopulateDashboardStats(); // statement

0312: var claim = claimService.GetClaimDetails(claimId.ToString()); // local variable declaration

0313: if (claim == null) // conditional check

0314: { // statement

0315: ViewBag.Error = $"No claim found with id {claimId}"; // statement

0316: } // statement

0317: return View(claim); // return value

0318: } // statement

0319:

0320: [HttpGet] // attribute/annotation

0321: public IActionResult ClaimSubmit() // member or method declaration

0322: { // statement

0323: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0324: PopulateDashboardStats(); // statement

0325: return View(); // return value

0326: } // statement

0327:

0328: [HttpPost] // attribute/annotation

0329: public IActionResult ClaimSubmit(Claim model) // member or method declaration

0330: { // statement

0331: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0332: PopulateDashboardStats(); // statement

0333: if (!ModelState.IsValid) // conditional check

0334: { // statement

0335: ViewBag.Error = "Please fill all required fields."; // statement

0336: return View(model); // return value

0337: } // statement

0338: var ok = claimService.SubmitClaim(model); // local variable declaration

0339: if (ok) // conditional check

0340: { // statement

0341: TempData["Success"] = "Claim submitted successfully."; // statement

0342: return RedirectToAction("ClaimSubmit"); // return value

0343: } // statement

0344: ViewBag.Error = "Failed to submit claim."; // statement

0345: return View(model); // return value

0346: } // statement

0347:

0348: [HttpGet] // attribute/annotation

0349: public IActionResult ClaimUpdateStatus() // member or method declaration

0350: { // statement

0351: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0352: PopulateDashboardStats(); // statement

0353: return View(); // return value

0354: } // statement

0355:

0356: [HttpPost] // attribute/annotation

0357: public IActionResult ClaimUpdateStatus(int claimId, ClaimStatus status, int? adjusterId) // member or method declaration

0358: { // statement

0359: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0360: PopulateDashboardStats(); // statement

0361: var claim = dbContext.Claims.Find(claimId); // local variable declaration

0362: if (claim == null) // conditional check

0363: { // statement

0364: ViewBag.Error = $"No claim found with id {claimId}"; // statement

0365: return View(); // return value

0366: } // statement

0367: claim.ClaimStatus = status; // statement

0368: if (adjusterId.HasValue) claim.AdjusterId = adjusterId.Value; // conditional check

0369: dbContext.SaveChanges(); // statement

0370: ViewBag.Success = "Claim status updated."; // statement

0371: return View(); // return value

0372: } // statement

0373:

0374: // Payments hub and pages // comment

0375: public IActionResult Payments() // member or method declaration

0376: { // statement

0377: return RedirectToAction("PaymentsHub"); // return value

0378: } // statement

0379:

0380: public IActionResult PaymentsHub() // member or method declaration

0381: { // statement

0382: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0383: PopulateDashboardStats(); // statement

0384: return View(); // return value

0385: } // statement

0386:

0387: public IActionResult PaymentsAll() // member or method declaration

0388: { // statement

0389: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0390: PopulateDashboardStats(); // statement

0391: var payments = dbContext.Payments // local variable declaration

0392: .AsNoTracking() // statement

0393: .Include(p => p.Policy) // statement

0394: .OrderBy(p => p.PaymentDate) // statement

0395: .ToList(); // statement

0396: return View("Payments", payments); // return value

0397: } // statement

0398:

0399: [HttpGet] // attribute/annotation

0400: public IActionResult PaymentMake() // member or method declaration

0401: { // statement

0402: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0403: return NotFound(); // return value

0404: } // statement

0405:

0406: [HttpPost] // attribute/annotation

0407: public IActionResult PaymentMake(Payment model) // member or method declaration

0408: { // statement

0409: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0410: return NotFound(); // return value

0411: } // statement

0412:

0413: [HttpGet] // attribute/annotation

0414: public IActionResult PaymentUpdateStatus() // member or method declaration

0415: { // statement

0416: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0417: PopulateDashboardStats(); // statement

0418: return View(); // return value

0419: } // statement

0420:

0421: [HttpPost] // attribute/annotation

0422: public IActionResult PaymentUpdateStatus(int paymentId, PaymentStatus status) // member or method declaration

0423: { // statement

0424: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0425: PopulateDashboardStats(); // statement

0426: var pay = dbContext.Payments.Find(paymentId); // local variable declaration

0427: if (pay == null) // conditional check

0428: { // statement

0429: ViewBag.Error = $"No payment found with id {paymentId}"; // statement

0430: return View(); // return value

0431: } // statement

0432: pay.PaymentStatus = status; // statement

0433: dbContext.SaveChanges(); // statement

0434: ViewBag.Success = "Payment status updated."; // statement

0435: return View(); // return value

0436: } // statement

0437:

0438: [HttpGet] // attribute/annotation

0439: public IActionResult PaymentsByPolicy() // member or method declaration

0440: { // statement

0441: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0442: PopulateDashboardStats(); // statement

0443: return View(model: null); // return value

0444: } // statement

0445:

0446: [HttpPost] // attribute/annotation

0447: public IActionResult PaymentsByPolicy(int policyId) // member or method declaration

0448: { // statement

0449: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0450: PopulateDashboardStats(); // statement

0451: var list = dbContext.Payments.Include(p => p.Policy).Where(p => p.PolicyId == policyId).OrderBy(p => p.PaymentDate).ToList(); // local variable declaration

0452: if (!list.Any()) ViewBag.Error = $"No payments found for policy {policyId}"; // conditional check

0453: return View(list); // return value

0454: } // statement

0455:

0456: public IActionResult Tickets() // member or method declaration

0457: { // statement

0458: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0459: PopulateDashboardStats(); // statement

0460: var tickets = supportTicketService.GetAllTickects(); // local variable declaration

0461: return View(tickets); // return value

0462: } // statement

0463:

0464: // Tickets hub and pages // comment

0465: public IActionResult TicketsHub() // member or method declaration

0466: { // statement

0467: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0468: PopulateDashboardStats(); // statement

0469: return View(); // return value

0470: } // statement

0471:

0472: public IActionResult TicketsAll() // member or method declaration

0473: { // statement

0474: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0475: PopulateDashboardStats(); // statement

0476: var tickets = supportTicketService.GetAllTickects(); // local variable declaration

0477: return View(tickets); // return value

0478: } // statement

0479:

0480: [HttpGet] // attribute/annotation

0481: public IActionResult TicketAssign() // member or method declaration

0482: { // statement

0483: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0484: PopulateDashboardStats(); // statement

0485: return View(model: null); // return value

0486: } // statement

0487:

0488: [HttpPost] // attribute/annotation

0489: public IActionResult TicketAssign(int ticketId, int agentId) // member or method declaration

0490: { // statement

0491: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0492: PopulateDashboardStats(); // statement

0493: var ok = supportTicketService.AssignTicket(ticketId, agentId); // local variable declaration

0494: if (!ok) // conditional check

0495: { // statement

0496: ViewBag.Error = $"Unable to assign ticket {ticketId}."; // statement

0497: return View(); // return value

0498: } // statement

0499: ViewBag.Success = "Ticket assigned."; // statement

0500: return View(); // return value

0501: } // statement

0502:

0503: [HttpGet] // attribute/annotation

0504: public IActionResult TicketResolve() // member or method declaration

0505: { // statement

0506: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0507: PopulateDashboardStats(); // statement

0508: return View(model: null); // return value

0509: } // statement

0510:

0511: [HttpPost] // attribute/annotation

0512: public IActionResult TicketResolve(int ticketId) // member or method declaration

0513: { // statement

0514: if (!IsAdmin()) return RedirectToAction("Login", "Auth"); // conditional check

0515: PopulateDashboardStats(); // statement

0516: var ok = supportTicketService.ResolvedTickect(ticketId); // local variable declaration

0517: if (!ok) // conditional check

0518: { // statement

0519: ViewBag.Error = $"No ticket found with id {ticketId}"; // statement

0520: return View(); // return value

0521: } // statement

0522: ViewBag.Success = "Ticket resolved."; // statement

0523: return View(); // return value

0524: } // statement

0525:

0526: // JSON endpoints for admin operations ------------------------- // comment

0527: [HttpPost] // attribute/annotation

0528: public IActionResult AdminCreateUser([FromForm] Users user) // member or method declaration

0529: { // statement

0530: if (!IsAdmin()) return Unauthorized(); // conditional check

0531: if (string.IsNullOrWhiteSpace(user.Username) || string.IsNullOrWhiteSpace(user.Password) || string.IsNullOrWhiteSpace(user.Email)) // conditional check

0532: return BadRequest("Invalid user input"); // return value

0533: var ok = usersService.RegisterUser(user); // local variable declaration

0534: return ok ? Ok(new { message = "User created" }) : BadRequest("Failed to create user"); // return value

0535: } // statement

0536:

0537: [HttpPost] // attribute/annotation

0538: public IActionResult AdminUpdateUser([FromForm] Users user) // member or method declaration

0539: { // statement

0540: if (!IsAdmin()) return Unauthorized(); // conditional check

0541: var existing = dbContext.Users.Find(user.UserId); // local variable declaration

0542: if (existing == null) return NotFound(); // conditional check

0543: existing.Email = user.Email; // statement

0544: existing.Username = user.Username; // statement

0545: existing.Role = user.Role; // statement

0546: if (!string.IsNullOrWhiteSpace(user.Password)) existing.Password = user.Password; // conditional check

0547: dbContext.SaveChanges(); // statement

0548: return Ok(new { message = "User updated" }); // return value

0549: } // statement

0550:

0551: [HttpPost] // attribute/annotation

0552: public IActionResult AdminDeleteUser([FromForm] int userId) // member or method declaration

0553: { // statement

0554: if (!IsAdmin()) return Unauthorized(); // conditional check

0555: var existing = dbContext.Users.Find(userId); // local variable declaration

0556: if (existing == null) return NotFound(); // conditional check

0557: dbContext.Users.Remove(existing); // statement

0558: dbContext.SaveChanges(); // statement

0559: return Ok(new { message = "User deleted" }); // return value

0560: } // statement

0561:

0562: [HttpPost] // attribute/annotation

0563: public IActionResult AdminCreatePolicy([FromForm] Policy model) // member or method declaration

0564: { // statement

0565: if (!IsAdmin()) return Unauthorized(); // conditional check

0566: var ok = policyService.CreatePolicy(model); // local variable declaration

0567: return ok ? Ok(new { message = "Policy created" }) : BadRequest("Failed to create policy"); // return value

0568: } // statement

0569:

0570: [HttpPost] // attribute/annotation

0571: public IActionResult AdminUpdatePolicy([FromForm] Policy model) // member or method declaration

0572: { // statement

0573: if (!IsAdmin()) return Unauthorized(); // conditional check

0574: var ok = policyService.UpdatePolicy(model); // local variable declaration

0575: return ok ? Ok(new { message = "Policy updated" }) : BadRequest("Failed to update policy"); // return value

0576: } // statement

0577:

0578: [HttpPost] // attribute/annotation

0579: public IActionResult AdminDeletePolicy([FromForm] int policyId) // member or method declaration

0580: { // statement

0581: if (!IsAdmin()) return Unauthorized(); // conditional check

0582: var ok = policyService.DeletePolicy(policyId.ToString()); // local variable declaration

0583: return ok ? Ok(new { message = "Policy deleted" }) : BadRequest("Failed to delete policy"); // return value

0584: } // statement

0585:

0586: [HttpGet] // attribute/annotation

0587: public IActionResult AdminGetPolicy(int id) // member or method declaration

0588: { // statement

0589: if (!IsAdmin()) return Unauthorized(); // conditional check

0590: var policy = policyService.GetPolicyById(id.ToString()); // local variable declaration

0591: if (policy == null) return NotFound(); // conditional check

0592: return Json(policy); // return value

0593: } // statement

0594:

0595: [HttpPost] // attribute/annotation

0596: public IActionResult AdminSubmitClaim([FromForm] Claim model) // member or method declaration

0597: { // statement

0598: if (!IsAdmin()) return Unauthorized(); // conditional check

0599: var ok = claimService.SubmitClaim(model); // local variable declaration

0600: return ok ? Ok(new { message = "Claim submitted" }) : BadRequest("Failed to submit claim"); // return value

0601: } // statement

0602:

0603: [HttpPost] // attribute/annotation

0604: public IActionResult AdminUpdateClaimStatus([FromForm] int claimId, [FromForm] ClaimStatus status, [FromForm] int? adjusterId) // member or method declaration

0605: { // statement

0606: if (!IsAdmin()) return Unauthorized(); // conditional check

0607: var claim = dbContext.Claims.Find(claimId); // local variable declaration

0608: if (claim == null) return NotFound(); // conditional check

0609: claim.ClaimStatus = status; // statement

0610: if (adjusterId.HasValue) claim.AdjusterId = adjusterId.Value; // conditional check

0611: dbContext.SaveChanges(); // statement

0612: return Ok(new { message = "Claim status updated" }); // return value

0613: } // statement

0614:

0615: [HttpPost] // attribute/annotation

0616: public IActionResult AdminResolveTicket([FromForm] int ticketId) // member or method declaration

0617: { // statement

0618: if (!IsAdmin()) return Unauthorized(); // conditional check

0619: var ticket = dbContext.SupportTickects.Find(ticketId); // local variable declaration

0620: if (ticket == null) return NotFound(); // conditional check

0621: ticket.TicketStatus = TicketStatus.RESOLVED; // statement

0622: ticket.ResolvedDate = DateTime.Now; // statement

0623: dbContext.SaveChanges(); // statement

0624: return Ok(new { message = "Ticket resolved" }); // return value

0625: } // statement

0626: } // statement

0627: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Controllers/AgentController.cs

0001: using Auto\_Insurance\_System.Data; // import namespace

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Models; // import namespace

0004: using Microsoft.AspNetCore.Mvc; // import namespace

0005: using Microsoft.EntityFrameworkCore; // import namespace

0006: using Auto\_Insurance\_System.Filters; // import namespace

0007:

0008: namespace Auto\_Insurance\_System.Controllers // declare namespace

0009: { // statement

0010: [RoleAuthorizationFilter("AGENT")] // attribute/annotation

0011: public class AgentController : Controller // class declaration

0012: { // statement

0013: private readonly IUsersService usersService; // member or method declaration

0014: private readonly IPolicyService policyService; // member or method declaration

0015: private readonly IClaimService claimService; // member or method declaration

0016: private readonly ISupportTicketService supportTicketService; // member or method declaration

0017: private readonly AutoInsuranceDbContext dbContext; // member or method declaration

0018:

0019: public AgentController( // member or method declaration

0020: IUsersService usersService, // statement

0021: IPolicyService policyService, // statement

0022: IClaimService claimService, // statement

0023: ISupportTicketService supportTicketService, // statement

0024: AutoInsuranceDbContext dbContext) // statement

0025: { // statement

0026: this.usersService = usersService; // statement

0027: this.policyService = policyService; // statement

0028: this.claimService = claimService; // statement

0029: this.supportTicketService = supportTicketService; // statement

0030: this.dbContext = dbContext; // statement

0031: } // statement

0032:

0033: private bool IsAgent() // member or method declaration

0034: { // statement

0035: return HttpContext.Session.GetString("UserRole") == "AGENT"; // return value

0036: } // statement

0037:

0038: // Users // comment

0039: public IActionResult UsersHub() // member or method declaration

0040: { // statement

0041: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0042: return View(); // return value

0043: } // statement

0044:

0045: public IActionResult Users() // member or method declaration

0046: { // statement

0047: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0048: var users = usersService.GetAllUsers(); // local variable declaration

0049: return View(users); // return value

0050: } // statement

0051:

0052: [HttpGet] // attribute/annotation

0053: public IActionResult UserCreate() // member or method declaration

0054: { // statement

0055: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0056: return View(new Users { Role = UserRole.CUSTOMER }); // return value

0057: } // statement

0058:

0059: [HttpPost] // attribute/annotation

0060: public IActionResult UserCreate(Users model) // member or method declaration

0061: { // statement

0062: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0063: if (!ModelState.IsValid) // conditional check

0064: { // statement

0065: ViewBag.Error = "Please correct the errors and try again."; // statement

0066: return View(model); // return value

0067: } // statement

0068: if (model.Role == UserRole.ADMIN) // conditional check

0069: { // statement

0070: ViewBag.Error = "Agents cannot create ADMIN users."; // statement

0071: return View(model); // return value

0072: } // statement

0073: var ok = usersService.RegisterUser(model); // local variable declaration

0074: if (ok) // conditional check

0075: { // statement

0076: TempData["Success"] = "User created successfully."; // statement

0077: return RedirectToAction("UserCreate"); // return value

0078: } // statement

0079: ViewBag.Error = "Failed to create user."; // statement

0080: return View(model); // return value

0081: } // statement

0082:

0083: // Policies // comment

0084: public IActionResult PoliciesHub() // member or method declaration

0085: { // statement

0086: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0087: return View(); // return value

0088: } // statement

0089:

0090: public IActionResult PoliciesAll() // member or method declaration

0091: { // statement

0092: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0093: var list = policyService.GetAllPolicies().OrderBy(p => p.PolicyId).ToList(); // local variable declaration

0094: return View(list); // return value

0095: } // statement

0096:

0097: [HttpGet] // attribute/annotation

0098: public IActionResult PolicyDetails() // member or method declaration

0099: { // statement

0100: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0101: return View(model: null); // return value

0102: } // statement

0103:

0104: [HttpPost] // attribute/annotation

0105: public IActionResult PolicyDetails(int policyId) // member or method declaration

0106: { // statement

0107: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0108: var policy = policyService.GetPolicyById(policyId.ToString()); // local variable declaration

0109: if (policy == null) ViewBag.Error = $"No policy found with id {policyId}"; // conditional check

0110: return View(policy); // return value

0111: } // statement

0112:

0113: [HttpGet] // attribute/annotation

0114: public IActionResult PolicyCreate() // member or method declaration

0115: { // statement

0116: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0117: return View(new Policy { // return value

0118: PolicyNumber = (new Random()).Next(10000, 99999).ToString(), // statement

0119: StartDate = DateTime.Today, EndDate = DateTime.Today.AddYears(1), PolicyStatus = PolicyStatus.ACTIVE }); // statement

0120: } // statement

0121:

0122: [HttpPost] // attribute/annotation

0123: public IActionResult PolicyCreate(Policy model) // member or method declaration

0124: { // statement

0125: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0126: if (!ModelState.IsValid) // conditional check

0127: { // statement

0128: ViewBag.Error = "Please correct the errors and try again."; // statement

0129: return View(model); // return value

0130: } // statement

0131: var ok = policyService.CreatePolicy(model); // local variable declaration

0132: if (ok) // conditional check

0133: { // statement

0134: TempData["Success"] = "Policy created successfully."; // statement

0135: return RedirectToAction("PolicyCreate"); // return value

0136: } // statement

0137: ViewBag.Error = "Failed to create policy."; // statement

0138: return View(model); // return value

0139: } // statement

0140:

0141: // Claims // comment

0142: public IActionResult ClaimsHub() // member or method declaration

0143: { // statement

0144: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0145: return View(); // return value

0146: } // statement

0147:

0148: public IActionResult ClaimsAll() // member or method declaration

0149: { // statement

0150: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0151: var claims = claimService.GetAllClaims().OrderBy(c => c.ClaimDate).ToList(); // local variable declaration

0152: return View(claims); // return value

0153: } // statement

0154:

0155: [HttpGet] // attribute/annotation

0156: public IActionResult ClaimDetails() // member or method declaration

0157: { // statement

0158: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0159: return View(model: null); // return value

0160: } // statement

0161:

0162: [HttpPost] // attribute/annotation

0163: public IActionResult ClaimDetails(int claimId) // member or method declaration

0164: { // statement

0165: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0166: var claim = claimService.GetClaimDetails(claimId.ToString()); // local variable declaration

0167: if (claim == null) // conditional check

0168: { // statement

0169: ViewBag.Error = $"No claim found with id {claimId}"; // statement

0170: } // statement

0171: return View(claim); // return value

0172: } // statement

0173:

0174: [HttpGet] // attribute/annotation

0175: public IActionResult ClaimSubmit() // member or method declaration

0176: { // statement

0177: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0178: return View(); // return value

0179: } // statement

0180:

0181: [HttpPost] // attribute/annotation

0182: public IActionResult ClaimSubmit(Claim model) // member or method declaration

0183: { // statement

0184: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0185: if (!ModelState.IsValid) // conditional check

0186: { // statement

0187: ViewBag.Error = "Please fill all required fields."; // statement

0188: return View(model); // return value

0189: } // statement

0190: var ok = claimService.SubmitClaim(model); // local variable declaration

0191: if (ok) // conditional check

0192: { // statement

0193: TempData["Success"] = "Claim submitted successfully."; // statement

0194: return RedirectToAction("ClaimSubmit"); // return value

0195: } // statement

0196: ViewBag.Error = "Failed to submit claim."; // statement

0197: return View(model); // return value

0198: } // statement

0199:

0200: // Payments // comment

0201: public IActionResult PaymentsHub() // member or method declaration

0202: { // statement

0203: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0204: return View(); // return value

0205: } // statement

0206:

0207: public IActionResult PaymentsAll() // member or method declaration

0208: { // statement

0209: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0210: var payments = dbContext.Payments // local variable declaration

0211: .AsNoTracking() // statement

0212: .Include(p => p.Policy) // statement

0213: .OrderBy(p => p.PaymentDate) // statement

0214: .ToList(); // statement

0215: return View("Payments", payments); // return value

0216: } // statement

0217:

0218: [HttpGet] // attribute/annotation

0219: public IActionResult PaymentsByPolicy() // member or method declaration

0220: { // statement

0221: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0222: return View(model: null); // return value

0223: } // statement

0224:

0225: [HttpPost] // attribute/annotation

0226: public IActionResult PaymentsByPolicy(int policyId) // member or method declaration

0227: { // statement

0228: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0229: var list = dbContext.Payments.Include(p => p.Policy).Where(p => p.PolicyId == policyId).OrderBy(p => p.PaymentDate).ToList(); // local variable declaration

0230: if (!list.Any()) ViewBag.Error = $"No payments found for policy {policyId}"; // conditional check

0231: return View(list); // return value

0232: } // statement

0233:

0234: [HttpGet] // attribute/annotation

0235: public IActionResult PaymentMake() // member or method declaration

0236: { // statement

0237: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0238: return View(new Payment { PaymentDate = DateTime.Today, PaymentStatus = PaymentStatus.SUCCESS }); // return value

0239: } // statement

0240:

0241: [HttpPost] // attribute/annotation

0242: public IActionResult PaymentMake(Payment model) // member or method declaration

0243: { // statement

0244: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0245: if (!ModelState.IsValid) // conditional check

0246: { // statement

0247: ViewBag.Error = "Please fill all required fields."; // statement

0248: return View(model); // return value

0249: } // statement

0250: // Always record agent payments as SUCCESS // comment

0251: model.PaymentStatus = PaymentStatus.SUCCESS; // statement

0252: var policy = dbContext.Policy.Find(model.PolicyId); // local variable declaration

0253: if (policy == null) // conditional check

0254: { // statement

0255: ViewBag.Error = $"No policy found with id {model.PolicyId}"; // statement

0256: return View(model); // return value

0257: } // statement

0258: if (model.PaymentAmount != policy.PremiumAmount) // conditional check

0259: { // statement

0260: ViewBag.Error = $"Amount must equal the policy premium ({policy.PremiumAmount:0.00})."; // statement

0261: return View(model); // return value

0262: } // statement

0263: var paymentService = new Services.PaymentService(dbContext); // local variable declaration

0264: var ok = paymentService.MakePayment(model); // local variable declaration

0265: if (ok) // conditional check

0266: { // statement

0267: TempData["Success"] = "Payment recorded successfully."; // statement

0268: return RedirectToAction("PaymentMake"); // return value

0269: } // statement

0270: ViewBag.Error = "Failed to record payment."; // statement

0271: return View(model); // return value

0272: } // statement

0273:

0274: // Tickets // comment

0275: public IActionResult TicketsHub() // member or method declaration

0276: { // statement

0277: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0278: return View(); // return value

0279: } // statement

0280:

0281: public IActionResult Tickets() // member or method declaration

0282: { // statement

0283: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0284: var tickets = supportTicketService.GetAllTickects(); // local variable declaration

0285: return View(tickets); // return value

0286: } // statement

0287:

0288: [HttpGet] // attribute/annotation

0289: public IActionResult TicketCreate() // member or method declaration

0290: { // statement

0291: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0292: return View(new SupportTicket{ CreatedDate = DateTime.Now, TicketStatus = TicketStatus.OPEN }); // return value

0293: } // statement

0294:

0295: [HttpPost] // attribute/annotation

0296: public IActionResult TicketCreate(SupportTicket model) // member or method declaration

0297: { // statement

0298: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0299: // Normalize server-side fields and avoid validating navigation property // comment

0300: ModelState.Remove(nameof(SupportTicket.CreatedDate)); // statement

0301: ModelState.Remove(nameof(SupportTicket.User)); // statement

0302: model.CreatedDate = DateTime.Now; // statement

0303: if (model.TicketStatus == 0) model.TicketStatus = TicketStatus.OPEN; // conditional check

0304:

0305: if (!ModelState.IsValid) // conditional check

0306: { // statement

0307: ViewBag.Error = "Please fill all required fields."; // statement

0308: return View(model); // return value

0309: } // statement

0310:

0311: // Validate FK: User must exist // comment

0312: if (!dbContext.Users.Any(u => u.UserId == model.UserId)) // conditional check

0313: { // statement

0314: ViewBag.Error = $"User with id {model.UserId} not found."; // statement

0315: return View(model); // return value

0316: } // statement

0317:

0318: var ok = supportTicketService.CreateTicket(model); // local variable declaration

0319: if (ok) // conditional check

0320: { // statement

0321: TempData["Success"] = "Ticket created successfully."; // statement

0322: return RedirectToAction("TicketCreate"); // return value

0323: } // statement

0324: ViewBag.Error = "Failed to create ticket."; // statement

0325: return View(model); // return value

0326: } // statement

0327:

0328: [HttpGet] // attribute/annotation

0329: public IActionResult TicketResolve() // member or method declaration

0330: { // statement

0331: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0332: return View(model: null); // return value

0333: } // statement

0334:

0335: [HttpPost] // attribute/annotation

0336: public IActionResult TicketResolve(int ticketId) // member or method declaration

0337: { // statement

0338: if (!IsAgent()) return RedirectToAction("Login", "Auth"); // conditional check

0339: var ticket = dbContext.SupportTickects.Find(ticketId); // local variable declaration

0340: if (ticket == null) // conditional check

0341: { // statement

0342: ViewBag.Error = $"No ticket found with id {ticketId}"; // statement

0343: return View(); // return value

0344: } // statement

0345: var currentUser = HttpContext.Session.GetString("UserName"); // local variable declaration

0346: var agent = usersService.GetAllUsers().FirstOrDefault(u => u.Username == currentUser); // local variable declaration

0347: if (ticket.AssignedAgentId != agent?.UserId) // conditional check

0348: { // statement

0349: ViewBag.Error = "Ticket is not assigned to you. You cannot resolve it."; // statement

0350: return View(); // return value

0351: } // statement

0352: var ok = supportTicketService.ResolvedTickect(ticketId); // local variable declaration

0353: if (!ok) // conditional check

0354: { // statement

0355: ViewBag.Error = "Failed to resolve ticket."; // statement

0356: return View(); // return value

0357: } // statement

0358: ViewBag.Success = "Ticket resolved."; // statement

0359: return View(); // return value

0360: } // statement

0361: } // statement

0362: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Controllers/AuthController.cs

0001: ﻿using Auto\_Insurance\_System.Interfaces; // statement

0002: using Auto\_Insurance\_System.Models; // import namespace

0003: using Microsoft.AspNetCore.Mvc; // import namespace

0004: using Auto\_Insurance\_System.Data; // import namespace

0005:

0006: namespace Auto\_Insurance\_System.Controllers // declare namespace

0007: { // statement

0008: public class AuthController : Controller // class declaration

0009: { // statement

0010: private readonly IUsersService userService; // member or method declaration

0011: private readonly IPolicyService? policyService; // member or method declaration

0012: private readonly IClaimService? claimService; // member or method declaration

0013: private readonly ISupportTicketService? supportTicketService; // member or method declaration

0014: private readonly AutoInsuranceDbContext? dbContext; // member or method declaration

0015:

0016: public AuthController(IUsersService userService, IPolicyService? policyService = null, IClaimService? claimService = null, ISupportTicketService? supportTicketService = null, AutoInsuranceDbContext? dbContext = null) // member or method declaration

0017: { // statement

0018: this.userService = userService; // statement

0019: this.policyService = policyService; // statement

0020: this.claimService = claimService; // statement

0021: this.supportTicketService = supportTicketService; // statement

0022: this.dbContext = dbContext; // statement

0023: } // statement

0024:

0025: private void PopulateAdminStats() // member or method declaration

0026: { // statement

0027: if (policyService == null || claimService == null || supportTicketService == null || dbContext == null) // conditional check

0028: return; // statement

0029:

0030: var policies = policyService.GetAllPolicies(); // local variable declaration

0031: var claims = claimService.GetAllClaims(); // local variable declaration

0032: var tickets = supportTicketService.GetAllTickects(); // local variable declaration

0033: var payments = dbContext.Payments.ToList(); // local variable declaration

0034:

0035: ViewBag.ActivePolicies = policies.Count(p => p.PolicyStatus == PolicyStatus.ACTIVE); // statement

0036: ViewBag.PendingClaims = claims.Count(c => c.ClaimStatus == ClaimStatus.OPEN); // statement

0037: var now = DateTime.Now; // local variable declaration

0038: ViewBag.RevenueThisMonth = payments // statement

0039: .Where(p => p.PaymentStatus == PaymentStatus.SUCCESS && p.PaymentDate.Month == now.Month && p.PaymentDate.Year == now.Year) // statement

0040: .Sum(p => (decimal?)p.PaymentAmount) ?? 0m; // statement

0041: ViewBag.OpenTickets = tickets.Count(t => t.TicketStatus == TicketStatus.OPEN); // statement

0042: } // statement

0043:

0044: // GET: /User/Register // comment

0045: public IActionResult Register() // member or method declaration

0046: { // statement

0047: return View(); // return value

0048: } // statement

0049:

0050: [HttpPost] // attribute/annotation

0051: public IActionResult Register(Users user) // member or method declaration

0052: { // statement

0053: try // begin try block

0054: { // statement

0055: user.Role = Enum.Parse<UserRole>(user.Role.ToString(), true); // statement

0056:

0057: if (userService.RegisterUser(user)) // conditional check

0058: { // statement

0059: ViewBag.SuccessMessage = $"Hello {user.Username}, you are successfully registered as {user.Role}."; // statement

0060: ViewBag.TriggerRedirect = true; // statement

0061: return View(); // return value

0062: } // statement

0063:

0064: ModelState.AddModelError("", "Registration failed. Please try again."); // statement

0065: return View(user); // return value

0066: } // statement

0067: catch (Exception ex) // handle exception

0068: { // statement

0069: ModelState.AddModelError("", $"Error during registration: {ex.Message}"); // statement

0070: return View(user); // return value

0071: } // statement

0072: } // statement

0073:

0074: // GET: /Auth/ForgotPassword // comment

0075: [HttpGet] // attribute/annotation

0076: public IActionResult ForgotPassword() // member or method declaration

0077: { // statement

0078: return View(); // return value

0079: } // statement

0080:

0081: // POST: /Auth/ForgotPassword // comment

0082: [HttpPost] // attribute/annotation

0083: [ValidateAntiForgeryToken] // attribute/annotation

0084: public IActionResult ForgotPassword(string? Username, string? Email, string NewPassword, string ConfirmPassword) // member or method declaration

0085: { // statement

0086: if (dbContext == null) // conditional check

0087: { // statement

0088: ModelState.AddModelError("", "Unexpected error: database not available."); // statement

0089: return View(); // return value

0090: } // statement

0091:

0092: if (string.IsNullOrWhiteSpace(Username) && string.IsNullOrWhiteSpace(Email)) // conditional check

0093: { // statement

0094: ModelState.AddModelError("", "Please provide Username or Email."); // statement

0095: return View(); // return value

0096: } // statement

0097: if (string.IsNullOrWhiteSpace(NewPassword) || string.IsNullOrWhiteSpace(ConfirmPassword)) // conditional check

0098: { // statement

0099: ModelState.AddModelError("", "Password and Confirm Password are required."); // statement

0100: return View(); // return value

0101: } // statement

0102: if (!string.Equals(NewPassword, ConfirmPassword)) // conditional check

0103: { // statement

0104: ModelState.AddModelError("", "Passwords do not match."); // statement

0105: return View(); // return value

0106: } // statement

0107:

0108: var user = !string.IsNullOrWhiteSpace(Username) // local variable declaration

0109: ? dbContext.Users.FirstOrDefault(u => u.Username == Username) // statement

0110: : dbContext.Users.FirstOrDefault(u => u.Email == Email); // statement

0111:

0112: if (user == null) // conditional check

0113: { // statement

0114: ModelState.AddModelError("", "No matching user found."); // statement

0115: return View(); // return value

0116: } // statement

0117:

0118: user.Password = NewPassword; // statement

0119: dbContext.SaveChanges(); // statement

0120: TempData["Success"] = "Password updated successfully. Please login."; // statement

0121: return RedirectToAction("Login"); // return value

0122: } // statement

0123:

0124: // GET: /User/Login // comment

0125: public IActionResult Login() // member or method declaration

0126: { // statement

0127: return View(); // return value

0128: } // statement

0129:

0130: // POST: /User/Login // comment

0131: [HttpPost] // attribute/annotation

0132: public IActionResult Login(string Username, string Password, string Role) // member or method declaration

0133: { // statement

0134: var userLogin = new Users // local variable declaration

0135: { // statement

0136: Username = Username, // statement

0137: Password = Password, // statement

0138: Role = Enum.Parse<UserRole>(Role) // statement

0139: }; // statement

0140:

0141: if (userService.VerifyUserLogin(userLogin)) // conditional check

0142: { // statement

0143: HttpContext.Session.SetString("UserName", Username); // statement

0144: HttpContext.Session.SetString("UserRole", Role); // statement

0145:

0146: return Role switch // return value

0147: { // statement

0148: "ADMIN" => RedirectToAction("AdminDashboard"), // statement

0149: "AGENT" => RedirectToAction("AgentDashboard"), // statement

0150: "CUSTOMER" => RedirectToAction("CustomerDashboard"), // statement

0151: \_ => RedirectToAction("Landing", "Home") // statement

0152: }; // statement

0153: } // statement

0154:

0155: TempData["LoginError"] = "Invalid username, password, or role."; // statement

0156: return RedirectToAction("Landing", "Home"); // return value

0157: } // statement

0158:

0159:

0160: public IActionResult AdminDashboard() // member or method declaration

0161: { // statement

0162: if (HttpContext.Session.GetString("UserRole") != "ADMIN") // conditional check

0163: return RedirectToAction("Login"); // return value

0164:

0165: PopulateAdminStats(); // statement

0166: return View(); // return value

0167: } // statement

0168:

0169: public IActionResult AgentDashboard() // member or method declaration

0170: { // statement

0171: if (HttpContext.Session.GetString("UserRole") != "AGENT") // conditional check

0172: return RedirectToAction("Login"); // return value

0173: PopulateAdminStats(); // statement

0174: return View(); // return value

0175: } // statement

0176:

0177: public IActionResult CustomerDashboard() // member or method declaration

0178: { // statement

0179: if (HttpContext.Session.GetString("UserRole") != "CUSTOMER") // conditional check

0180: return RedirectToAction("Login"); // return value

0181: return View(); // return value

0182: } // statement

0183:

0184: public IActionResult Profile() // member or method declaration

0185: { // statement

0186: string username = HttpContext.Session.GetString("UserName"); // statement

0187: var user = userService.GetAllUsers().FirstOrDefault(u => u.Username == username); // local variable declaration

0188: return View(user); // return value

0189: } // statement

0190:

0191: public IActionResult Logout() // member or method declaration

0192: { // statement

0193: HttpContext.Session.Clear(); // statement

0194: return RedirectToAction("Landing", "Home"); // return value

0195: } // statement

0196:

0197:

0198: } // statement

0199: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Controllers/CustomerController.cs

0001: using Auto\_Insurance\_System.Data; // import namespace

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Models; // import namespace

0004: using Microsoft.AspNetCore.Mvc; // import namespace

0005: using Microsoft.EntityFrameworkCore; // import namespace

0006: using Auto\_Insurance\_System.Filters; // import namespace

0007:

0008: namespace Auto\_Insurance\_System.Controllers // declare namespace

0009: { // statement

0010: [RoleAuthorizationFilter("CUSTOMER")] // attribute/annotation

0011: public class CustomerController : Controller // class declaration

0012: { // statement

0013: private readonly IPolicyService policyService; // member or method declaration

0014: private readonly IClaimService claimService; // member or method declaration

0015: private readonly IPaymentService paymentService; // member or method declaration

0016: private readonly ISupportTicketService ticketService; // member or method declaration

0017: private readonly AutoInsuranceDbContext db; // member or method declaration

0018:

0019: public CustomerController(IPolicyService policyService, IClaimService claimService, IPaymentService paymentService, ISupportTicketService ticketService, AutoInsuranceDbContext db) // member or method declaration

0020: { // statement

0021: this.policyService = policyService; // statement

0022: this.claimService = claimService; // statement

0023: this.paymentService = paymentService; // statement

0024: this.ticketService = ticketService; // statement

0025: this.db = db; // statement

0026: } // statement

0027:

0028: private bool IsCustomer() // member or method declaration

0029: { // statement

0030: return HttpContext.Session.GetString("UserRole") == "CUSTOMER"; // return value

0031: } // statement

0032:

0033: private Users? CurrentUser() // member or method declaration

0034: { // statement

0035: var name = HttpContext.Session.GetString("UserName"); // local variable declaration

0036: return db.Users.FirstOrDefault(u => u.Username == name); // return value

0037: } // statement

0038:

0039: // Dashboard hubs // comment

0040: public IActionResult PoliciesHub() // member or method declaration

0041: { // statement

0042: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0043: return View(); // return value

0044: } // statement

0045:

0046: public IActionResult ClaimsHub() // member or method declaration

0047: { // statement

0048: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0049: return View(); // return value

0050: } // statement

0051:

0052: public IActionResult PaymentsHub() // member or method declaration

0053: { // statement

0054: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0055: return View(); // return value

0056: } // statement

0057:

0058: public IActionResult TicketsHub() // member or method declaration

0059: { // statement

0060: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0061: return View(); // return value

0062: } // statement

0063:

0064: // Policy: create // comment

0065: [HttpGet] // attribute/annotation

0066: public IActionResult PolicyCreate() // member or method declaration

0067: { // statement

0068: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0069: return View(new Policy { // return value

0070: PolicyNumber = (new Random()).Next(10000, 99999).ToString(), // statement

0071: StartDate = DateTime.Today, EndDate = DateTime.Today.AddYears(1), PolicyStatus = PolicyStatus.ACTIVE }); // statement

0072: } // statement

0073:

0074: [HttpPost] // attribute/annotation

0075: public IActionResult PolicyCreate(Policy model) // member or method declaration

0076: { // statement

0077: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0078: if (!ModelState.IsValid) // conditional check

0079: { // statement

0080: ViewBag.Error = "Please correct the errors and try again."; // statement

0081: return View(model); // return value

0082: } // statement

0083: //var ok = policyService.CreatePolicy(model); // comment

0084: //https://localhost:7227/api/admin/policies // comment

0085: HttpClient client = new HttpClient(); // statement

0086: client.BaseAddress = new Uri("https://localhost:7227/api/"); // statement

0087: var result = client.PostAsJsonAsync("admin/policies", model); // local variable declaration

0088: result.Wait(); // statement

0089: bool ok = false; // statement

0090: if (result.IsCompleted) // conditional check

0091: { // statement

0092: ok = true; // statement

0093: } // statement

0094:

0095: if (ok) // conditional check

0096: { // statement

0097: TempData["Success"] = "Policy created successfully."; // statement

0098: return RedirectToAction("PolicyCreate"); // return value

0099: } // statement

0100: ViewBag.Error = "Failed to create policy."; // statement

0101: return View(model); // return value

0102: } // statement

0103:

0104: // Claims: submit and list my claims // comment

0105: [HttpGet] // attribute/annotation

0106: public IActionResult ClaimSubmit() // member or method declaration

0107: { // statement

0108: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0109: return View(new Claim { ClaimDate = DateTime.Today, ClaimStatus = ClaimStatus.OPEN }); // return value

0110: } // statement

0111:

0112: [HttpPost] // attribute/annotation

0113: public IActionResult ClaimSubmit(Claim model) // member or method declaration

0114: { // statement

0115: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0116: if (!ModelState.IsValid) // conditional check

0117: { // statement

0118: ViewBag.Error = "Please fill all required fields."; // statement

0119: return View(model); // return value

0120: } // statement

0121: // Force OPEN status // comment

0122: model.ClaimStatus = ClaimStatus.OPEN; // statement

0123: var ok = claimService.SubmitClaim(model); // local variable declaration

0124: if (ok) // conditional check

0125: { // statement

0126: TempData["Success"] = "Claim submitted successfully."; // statement

0127: return RedirectToAction("ClaimSubmit"); // return value

0128: } // statement

0129: ViewBag.Error = "Failed to submit claim."; // statement

0130: return View(model); // return value

0131: } // statement

0132:

0133: public IActionResult Claims() // member or method declaration

0134: { // statement

0135: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0136: var me = CurrentUser(); // local variable declaration

0137: // Assuming AdjusterId is not the customer, use Policy ownership once available; for now show none unless policy filter is used // comment

0138: var myPolicyIds = db.Policy.Select(p => p.PolicyId).ToList(); // local variable declaration

0139: var claims = db.Claims.AsNoTracking() // local variable declaration

0140: .Where(c => myPolicyIds.Contains(c.PolicyId)) // statement

0141: .OrderBy(c => c.ClaimDate) // statement

0142: .ToList(); // statement

0143: return View(claims); // return value

0144: } // statement

0145:

0146: // Payments: make and list my payments // comment

0147: [HttpGet] // attribute/annotation

0148: public IActionResult PaymentMake() // member or method declaration

0149: { // statement

0150: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0151: return View(new Payment { PaymentDate = DateTime.Today, PaymentStatus = PaymentStatus.SUCCESS }); // return value

0152: } // statement

0153:

0154: [HttpPost] // attribute/annotation

0155: public IActionResult PaymentMake(Payment model) // member or method declaration

0156: { // statement

0157: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0158: if (!ModelState.IsValid) // conditional check

0159: { // statement

0160: ViewBag.Error = "Please fill all required fields."; // statement

0161: return View(model); // return value

0162: } // statement

0163: // Force status to SUCCESS regardless of posted value // comment

0164: model.PaymentStatus = PaymentStatus.SUCCESS; // statement

0165: var policy = db.Policy.Find(model.PolicyId); // local variable declaration

0166: if (policy == null) // conditional check

0167: { // statement

0168: ViewBag.Error = $"No policy found with id {model.PolicyId}"; // statement

0169: return View(model); // return value

0170: } // statement

0171: if (model.PaymentAmount != policy.PremiumAmount) // conditional check

0172: { // statement

0173: ViewBag.Error = $"Amount must equal the policy premium ({policy.PremiumAmount:0.00})."; // statement

0174: return View(model); // return value

0175: } // statement

0176: var ok = paymentService.MakePayment(model); // local variable declaration

0177: if (ok) // conditional check

0178: { // statement

0179: TempData["Success"] = "Payment recorded successfully."; // statement

0180: return RedirectToAction("PaymentMake"); // return value

0181: } // statement

0182: ViewBag.Error = "Failed to record payment."; // statement

0183: return View(model); // return value

0184: } // statement

0185:

0186: public IActionResult Payments() // member or method declaration

0187: { // statement

0188: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0189: var me = CurrentUser(); // local variable declaration

0190: var myPolicyIds = db.Policy.Select(p => p.PolicyId).ToList(); // local variable declaration

0191: var payments = db.Payments.AsNoTracking() // local variable declaration

0192: .Where(p => myPolicyIds.Contains(p.PolicyId)) // statement

0193: .OrderBy(p => p.PaymentDate) // statement

0194: .ToList(); // statement

0195: return View(payments); // return value

0196: } // statement

0197:

0198: // Tickets: create and list my tickets // comment

0199: [HttpGet] // attribute/annotation

0200: public IActionResult TicketCreate() // member or method declaration

0201: { // statement

0202: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0203: return View(new SupportTicket { CreatedDate = DateTime.Now, TicketStatus = TicketStatus.OPEN }); // return value

0204: } // statement

0205:

0206: [HttpPost] // attribute/annotation

0207: public IActionResult TicketCreate(SupportTicket model) // member or method declaration

0208: { // statement

0209: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0210: ModelState.Remove(nameof(SupportTicket.CreatedDate)); // statement

0211: ModelState.Remove(nameof(SupportTicket.User)); // statement

0212: model.CreatedDate = DateTime.Now; // statement

0213: model.TicketStatus = TicketStatus.OPEN; // statement

0214: if (!ModelState.IsValid) // conditional check

0215: { // statement

0216: ViewBag.Error = "Please fill all required fields."; // statement

0217: return View(model); // return value

0218: } // statement

0219: var ok = ticketService.CreateTicket(model); // local variable declaration

0220: if (ok) // conditional check

0221: { // statement

0222: TempData["Success"] = "Ticket created successfully."; // statement

0223: return RedirectToAction("TicketCreate"); // return value

0224: } // statement

0225: ViewBag.Error = "Failed to create ticket."; // statement

0226: return View(model); // return value

0227: } // statement

0228:

0229: public IActionResult Tickets() // member or method declaration

0230: { // statement

0231: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0232: var me = CurrentUser(); // local variable declaration

0233: var tickets = db.SupportTickects.Include(t => t.User).Where(t => t.UserId == me!.UserId).OrderBy(t => t.CreatedDate).ToList(); // local variable declaration

0234: return View(tickets); // return value

0235: } // statement

0236:

0237: public IActionResult Policies() // member or method declaration

0238: { // statement

0239: if (!IsCustomer()) return RedirectToAction("Login", "Auth"); // conditional check

0240: var me = CurrentUser(); // local variable declaration

0241: var list = db.Policy.AsNoTracking().OrderBy(p => p.PolicyId).ToList(); // local variable declaration

0242: return View(list); // return value

0243: } // statement

0244: } // statement

0245: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Controllers/HomeController.cs

0001: using System.Diagnostics; // import namespace

0002: using Auto\_Insurance\_System.Models; // import namespace

0003: using Microsoft.AspNetCore.Mvc; // import namespace

0004:

0005: namespace Auto\_Insurance\_System.Controllers // declare namespace

0006: { // statement

0007: public class HomeController : Controller // class declaration

0008: { // statement

0009: private readonly ILogger<HomeController> \_logger; // member or method declaration

0010:

0011: public HomeController(ILogger<HomeController> logger) // member or method declaration

0012: { // statement

0013: \_logger = logger; // statement

0014: } // statement

0015:

0016: public IActionResult Index() // member or method declaration

0017: { // statement

0018: return RedirectToAction("Landing"); // return value

0019: } // statement

0020:

0021: public IActionResult Privacy() // member or method declaration

0022: { // statement

0023: return View(); // return value

0024: } // statement

0025:

0026: [Route("landing")] // attribute/annotation

0027: public IActionResult Landing() // member or method declaration

0028: { // statement

0029: return View(); // return value

0030: } // statement

0031:

0032: public IActionResult About() // member or method declaration

0033: { // statement

0034: return View(); // return value

0035: } // statement

0036:

0037:

0038: [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)] // attribute/annotation

0039: public IActionResult Error() // member or method declaration

0040: { // statement

0041: return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier }); // return value

0042: } // statement

0043:

0044: public IActionResult AccessDenied() // member or method declaration

0045: { // statement

0046: return View(); // return value

0047: } // statement

0048: } // statement

0049: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Data/AutoInsuranceDbContext.cs

0001: ﻿using Auto\_Insurance\_System.Models; // statement

0002: using Microsoft.EntityFrameworkCore; // import namespace

0003:

0004: namespace Auto\_Insurance\_System.Data // declare namespace

0005: { // statement

0006: public class AutoInsuranceDbContext : DbContext // class declaration

0007: { // statement

0008: public AutoInsuranceDbContext(DbContextOptions<AutoInsuranceDbContext> options) : base(options) { } // member or method declaration

0009: public virtual DbSet<Policy> Policy { get; set; } // member or method declaration

0010: public virtual DbSet<Claim> Claims { get; set; } // member or method declaration

0011: public virtual DbSet<Payment> Payments { get; set; } // member or method declaration

0012: public virtual DbSet<SupportTicket> SupportTickects { get; set; } // member or method declaration

0013: public virtual DbSet<Users> Users { get; set; } // member or method declaration

0014: } // statement

0015: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Filters/AdminAuthorizationFilterAttribute.cs

0001: ﻿using Microsoft.AspNetCore.Mvc; // statement

0002: using Microsoft.AspNetCore.Mvc.Filters; // import namespace

0003:

0004: namespace Auto\_Insurance\_System.Filters // declare namespace

0005: { // statement

0006: public class AdminAuthorizationFilterAttribute : Attribute, IAuthorizationFilter // class declaration

0007: { // statement

0008: public void OnAuthorization(AuthorizationFilterContext context) // member or method declaration

0009: { // statement

0010: string userId = context.HttpContext.Session.GetString("UserName"); // statement

0011: if (userId != "Admin") // conditional check

0012: { // statement

0013: context.HttpContext.Session.SetString("Error", "You are not authorized to access the given activity or resource"); // statement

0014: context.Result = new ViewResult // statement

0015: { // statement

0016: ViewName = "Error" // statement

0017: }; // statement

0018: } // statement

0019: } // statement

0020: } // statement

0021:

0022: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Filters/AutoInsuranceExceptionFilter.cs

0001: ﻿using Microsoft.AspNetCore.Mvc; // statement

0002: using Microsoft.AspNetCore.Mvc.Filters; // import namespace

0003:

0004: namespace Auto\_Insurance\_System.Filters // declare namespace

0005: { // statement

0006: public class AutoInsuranceExceptionFilter : IExceptionFilter // class declaration

0007: { // statement

0008: public void OnException(ExceptionContext context) // member or method declaration

0009: { // statement

0010: Exception ex = context.Exception; // statement

0011: context.ExceptionHandled = true; // statement

0012: string errorMessage = "Server Error " + ex.Message; // statement

0013: context.HttpContext.Session.SetString("Error", errorMessage); // statement

0014: context.Result = new ViewResult // statement

0015: { // statement

0016: ViewName = "Error" // statement

0017: }; // statement

0018: } // statement

0019: } // statement

0020: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Filters/RoleAuthorizationFilterAttribute.cs

0001: using Microsoft.AspNetCore.Mvc; // import namespace

0002: using Microsoft.AspNetCore.Mvc.Filters; // import namespace

0003:

0004: namespace Auto\_Insurance\_System.Filters // declare namespace

0005: { // statement

0006: public class RoleAuthorizationFilterAttribute : Attribute, IAuthorizationFilter // class declaration

0007: { // statement

0008: private readonly string requiredRole; // member or method declaration

0009:

0010: public RoleAuthorizationFilterAttribute(string requiredRole) // member or method declaration

0011: { // statement

0012: this.requiredRole = requiredRole; // statement

0013: } // statement

0014:

0015: public void OnAuthorization(AuthorizationFilterContext context) // member or method declaration

0016: { // statement

0017: var sessionRole = context.HttpContext.Session.GetString("UserRole"); // local variable declaration

0018: if (string.IsNullOrEmpty(sessionRole) || !string.Equals(sessionRole, requiredRole, StringComparison.OrdinalIgnoreCase)) // conditional check

0019: { // statement

0020: context.Result = new RedirectToActionResult("AccessDenied", "Home", null); // statement

0021: } // statement

0022: } // statement

0023: } // statement

0024: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Interfaces/IClaimService.cs

0001: ﻿using Auto\_Insurance\_System.Models; // statement

0002: namespace Auto\_Insurance\_System.Interfaces // declare namespace

0003: { // statement

0004: public interface IClaimService // interface declaration

0005: { // statement

0006: bool SubmitClaim(Claim obj); // statement

0007: Claim GetClaimDetails(string id); // statement

0008: bool UpdateClaimStatus(Claim obj); // statement

0009: List<Claim> GetAllClaims(); // statement

0010: } // statement

0011: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Interfaces/IPaymentService.cs

0001: ﻿using Auto\_Insurance\_System.Models; // statement

0002:

0003: namespace Auto\_Insurance\_System.Interfaces // declare namespace

0004: { // statement

0005: public interface IPaymentService // interface declaration

0006: { // statement

0007: bool MakePayment(Payment obj); // statement

0008: Payment GetPaymentDetails(string id); // statement

0009: List<Payment> GetPaymentsByPolicy(int policyId); // statement

0010: } // statement

0011: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Interfaces/IPolicyService.cs

0001: ﻿using Auto\_Insurance\_System.Models; // statement

0002:

0003: namespace Auto\_Insurance\_System.Interfaces // declare namespace

0004: { // statement

0005: public interface IPolicyService // interface declaration

0006: { // statement

0007: bool CreatePolicy(Policy obj); // statement

0008: List<Policy> GetAllPolicies(); // statement

0009: Policy GetPolicyById(string id); // statement

0010: bool UpdatePolicy(Policy obj); // statement

0011: bool DeletePolicy(string id); // statement

0012: } // statement

0013: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Interfaces/ISupportTicketService.cs

0001: ﻿using Auto\_Insurance\_System.Models; // statement

0002:

0003: namespace Auto\_Insurance\_System.Interfaces // declare namespace

0004: { // statement

0005: public interface ISupportTicketService // interface declaration

0006: { // statement

0007: bool CreateTicket(SupportTicket ticket); // statement

0008: SupportTicket GetTickectDetails(int id); // statement

0009: bool ResolvedTickect(int ticketId); // statement

0010: bool AssignTicket(int ticketId, int agentUserId); // statement

0011: List<SupportTicket> GetAllTickects(); // statement

0012: } // statement

0013: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Interfaces/IUsersService.cs

0001: ﻿using Auto\_Insurance\_System.Models; // statement

0002:

0003: namespace Auto\_Insurance\_System.Interfaces // declare namespace

0004: { // statement

0005: public interface IUsersService // interface declaration

0006: { // statement

0007: bool RegisterUser(Users user); // statement

0008: Users Login(string username, string password); // statement

0009: Users GetUserProfile(int userId); // statement

0010: bool Logout(int userId); // statement

0011: List<Users> GetAllUsers(); // statement

0012: bool ChangePassword(Users userLogin); // statement

0013: bool VerifyUserLogin(Users userLogin); // statement

0014:

0015: } // statement

0016: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Migrations/20250825124659\_AutoInsuranceContext.Designer.cs

0001: ﻿// <auto-generated /> // statement

0002: using System; // import namespace

0003: using Auto\_Insurance\_System.Data; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005: using Microsoft.EntityFrameworkCore.Infrastructure; // import namespace

0006: using Microsoft.EntityFrameworkCore.Metadata; // import namespace

0007: using Microsoft.EntityFrameworkCore.Migrations; // import namespace

0008: using Microsoft.EntityFrameworkCore.Storage.ValueConversion; // import namespace

0009:

0010: #nullable disable // statement

0011:

0012: namespace Auto\_Insurance\_System.Migrations // declare namespace

0013: { // statement

0014: [DbContext(typeof(AutoInsuranceDbContext))] // attribute/annotation

0015: [Migration("20250825124659\_AutoInsuranceContext")] // attribute/annotation

0016: partial class AutoInsuranceContext // class declaration

0017: { // statement

0018: /// <inheritdoc /> // comment

0019: protected override void BuildTargetModel(ModelBuilder modelBuilder) // member or method declaration

0020: { // statement

0021: #pragma warning disable 612, 618 // statement

0022: modelBuilder // statement

0023: .HasAnnotation("ProductVersion", "9.0.7") // statement

0024: .HasAnnotation("Relational:MaxIdentifierLength", 128); // statement

0025:

0026: SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder); // statement

0027:

0028: modelBuilder.Entity("Auto\_Insurance\_System.Models.Claim", b => // statement

0029: { // statement

0030: b.Property<int>("ClaimId") // statement

0031: .ValueGeneratedOnAdd() // statement

0032: .HasColumnType("int"); // statement

0033:

0034: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ClaimId")); // statement

0035:

0036: b.Property<int>("AdjusterId") // statement

0037: .HasColumnType("int"); // statement

0038:

0039: b.Property<decimal>("ClaimAmount") // statement

0040: .HasColumnType("decimal(18,2)"); // statement

0041:

0042: b.Property<DateTime>("ClaimDate") // statement

0043: .HasColumnType("datetime2"); // statement

0044:

0045: b.Property<int>("ClaimStatus") // statement

0046: .HasColumnType("int"); // statement

0047:

0048: b.Property<int>("PolicyId") // statement

0049: .HasColumnType("int"); // statement

0050:

0051: b.HasKey("ClaimId"); // statement

0052:

0053: b.HasIndex("AdjusterId"); // statement

0054:

0055: b.HasIndex("PolicyId"); // statement

0056:

0057: b.ToTable("Claims"); // statement

0058: }); // statement

0059:

0060: modelBuilder.Entity("Auto\_Insurance\_System.Models.Payment", b => // statement

0061: { // statement

0062: b.Property<int>("PaymentId") // statement

0063: .ValueGeneratedOnAdd() // statement

0064: .HasColumnType("int"); // statement

0065:

0066: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("PaymentId")); // statement

0067:

0068: b.Property<decimal>("PaymentAmount") // statement

0069: .HasColumnType("decimal(10,2)"); // statement

0070:

0071: b.Property<DateTime>("PaymentDate") // statement

0072: .HasColumnType("datetime2"); // statement

0073:

0074: b.Property<int>("PaymentStatus") // statement

0075: .HasColumnType("int"); // statement

0076:

0077: b.Property<int>("PolicyId") // statement

0078: .HasColumnType("int"); // statement

0079:

0080: b.HasKey("PaymentId"); // statement

0081:

0082: b.HasIndex("PolicyId"); // statement

0083:

0084: b.ToTable("Payments"); // statement

0085: }); // statement

0086:

0087: modelBuilder.Entity("Auto\_Insurance\_System.Models.Policy", b => // statement

0088: { // statement

0089: b.Property<int>("PolicyId") // statement

0090: .ValueGeneratedOnAdd() // statement

0091: .HasColumnType("int"); // statement

0092:

0093: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("PolicyId")); // statement

0094:

0095: b.Property<decimal>("CoverageAmount") // statement

0096: .HasColumnType("decimal(10,2)"); // statement

0097:

0098: b.Property<string>("CoverageType") // statement

0099: .IsRequired() // statement

0100: .HasMaxLength(50) // statement

0101: .HasColumnType("nvarchar(50)"); // statement

0102:

0103: b.Property<DateTime>("EndDate") // statement

0104: .HasColumnType("datetime2"); // statement

0105:

0106: b.Property<string>("PolicyNumber") // statement

0107: .IsRequired() // statement

0108: .HasMaxLength(50) // statement

0109: .HasColumnType("nvarchar(50)"); // statement

0110:

0111: b.Property<int>("PolicyStatus") // statement

0112: .HasColumnType("int"); // statement

0113:

0114: b.Property<decimal>("PremiumAmount") // statement

0115: .HasColumnType("decimal(10,2)"); // statement

0116:

0117: b.Property<DateTime>("StartDate") // statement

0118: .HasColumnType("datetime2"); // statement

0119:

0120: b.Property<string>("VehicleDetails") // statement

0121: .IsRequired() // statement

0122: .HasMaxLength(200) // statement

0123: .HasColumnType("nvarchar(200)"); // statement

0124:

0125: b.HasKey("PolicyId"); // statement

0126:

0127: b.ToTable("Policy"); // statement

0128: }); // statement

0129:

0130: modelBuilder.Entity("Auto\_Insurance\_System.Models.SupportTicket", b => // statement

0131: { // statement

0132: b.Property<int>("TicketId") // statement

0133: .ValueGeneratedOnAdd() // statement

0134: .HasColumnType("int"); // statement

0135:

0136: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("TicketId")); // statement

0137:

0138: b.Property<int?>("AssignedAgentId") // statement

0139: .HasColumnType("int"); // statement

0140:

0141: b.Property<DateTime>("CreatedDate") // statement

0142: .HasColumnType("datetime2"); // statement

0143:

0144: b.Property<string>("IssueDescription") // statement

0145: .IsRequired() // statement

0146: .HasColumnType("nvarchar(max)"); // statement

0147:

0148: b.Property<DateTime?>("ResolvedDate") // statement

0149: .HasColumnType("datetime2"); // statement

0150:

0151: b.Property<int>("TicketStatus") // statement

0152: .HasColumnType("int"); // statement

0153:

0154: b.Property<int>("UserId") // statement

0155: .HasColumnType("int"); // statement

0156:

0157: b.HasKey("TicketId"); // statement

0158:

0159: b.HasIndex("UserId"); // statement

0160:

0161: b.ToTable("SupportTickets"); // statement

0162: }); // statement

0163:

0164: modelBuilder.Entity("Auto\_Insurance\_System.Models.Users", b => // statement

0165: { // statement

0166: b.Property<int>("UserId") // statement

0167: .ValueGeneratedOnAdd() // statement

0168: .HasColumnType("int"); // statement

0169:

0170: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("UserId")); // statement

0171:

0172: b.Property<string>("Email") // statement

0173: .IsRequired() // statement

0174: .HasMaxLength(100) // statement

0175: .HasColumnType("nvarchar(100)"); // statement

0176:

0177: b.Property<string>("Password") // statement

0178: .IsRequired() // statement

0179: .HasMaxLength(255) // statement

0180: .HasColumnType("nvarchar(255)"); // statement

0181:

0182: b.Property<int>("Role") // statement

0183: .HasColumnType("int"); // statement

0184:

0185: b.Property<string>("Username") // statement

0186: .IsRequired() // statement

0187: .HasMaxLength(50) // statement

0188: .HasColumnType("nvarchar(50)"); // statement

0189:

0190: b.HasKey("UserId"); // statement

0191:

0192: b.ToTable("Users"); // statement

0193: }); // statement

0194:

0195: modelBuilder.Entity("Auto\_Insurance\_System.Models.Claim", b => // statement

0196: { // statement

0197: b.HasOne("Auto\_Insurance\_System.Models.Users", "Adjuster") // statement

0198: .WithMany() // statement

0199: .HasForeignKey("AdjusterId") // statement

0200: .OnDelete(DeleteBehavior.Cascade) // statement

0201: .IsRequired(); // statement

0202:

0203: b.HasOne("Auto\_Insurance\_System.Models.Policy", "Policy") // statement

0204: .WithMany() // statement

0205: .HasForeignKey("PolicyId") // statement

0206: .OnDelete(DeleteBehavior.Cascade) // statement

0207: .IsRequired(); // statement

0208:

0209: b.Navigation("Adjuster"); // statement

0210:

0211: b.Navigation("Policy"); // statement

0212: }); // statement

0213:

0214: modelBuilder.Entity("Auto\_Insurance\_System.Models.Payment", b => // statement

0215: { // statement

0216: b.HasOne("Auto\_Insurance\_System.Models.Policy", "Policy") // statement

0217: .WithMany() // statement

0218: .HasForeignKey("PolicyId") // statement

0219: .OnDelete(DeleteBehavior.Cascade) // statement

0220: .IsRequired(); // statement

0221:

0222: b.Navigation("Policy"); // statement

0223: }); // statement

0224:

0225: modelBuilder.Entity("Auto\_Insurance\_System.Models.SupportTicket", b => // statement

0226: { // statement

0227: b.HasOne("Auto\_Insurance\_System.Models.Users", "User") // statement

0228: .WithMany() // statement

0229: .HasForeignKey("UserId") // statement

0230: .OnDelete(DeleteBehavior.Cascade) // statement

0231: .IsRequired(); // statement

0232:

0233: b.Navigation("User"); // statement

0234: }); // statement

0235: #pragma warning restore 612, 618 // statement

0236: } // statement

0237: } // statement

0238: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Migrations/20250825124659\_AutoInsuranceContext.cs

0001: ﻿using System; // statement

0002: using Microsoft.EntityFrameworkCore.Migrations; // import namespace

0003:

0004: #nullable disable // statement

0005:

0006: namespace Auto\_Insurance\_System.Migrations // declare namespace

0007: { // statement

0008: /// <inheritdoc /> // comment

0009: public partial class AutoInsuranceContext : Migration // class declaration

0010: { // statement

0011: /// <inheritdoc /> // comment

0012: protected override void Up(MigrationBuilder migrationBuilder) // member or method declaration

0013: { // statement

0014: migrationBuilder.CreateTable( // statement

0015: name: "Policy", // statement

0016: columns: table => new // statement

0017: { // statement

0018: PolicyId = table.Column<int>(type: "int", nullable: false) // statement

0019: .Annotation("SqlServer:Identity", "1, 1"), // statement

0020: PolicyNumber = table.Column<string>(type: "nvarchar(50)", maxLength: 50, nullable: false), // statement

0021: VehicleDetails = table.Column<string>(type: "nvarchar(200)", maxLength: 200, nullable: false), // statement

0022: CoverageAmount = table.Column<decimal>(type: "decimal(10,2)", nullable: false), // statement

0023: CoverageType = table.Column<string>(type: "nvarchar(50)", maxLength: 50, nullable: false), // statement

0024: PremiumAmount = table.Column<decimal>(type: "decimal(10,2)", nullable: false), // statement

0025: StartDate = table.Column<DateTime>(type: "datetime2", nullable: false), // statement

0026: EndDate = table.Column<DateTime>(type: "datetime2", nullable: false), // statement

0027: PolicyStatus = table.Column<int>(type: "int", nullable: false) // statement

0028: }, // statement

0029: constraints: table => // statement

0030: { // statement

0031: table.PrimaryKey("PK\_Policy", x => x.PolicyId); // statement

0032: }); // statement

0033:

0034: migrationBuilder.CreateTable( // statement

0035: name: "Users", // statement

0036: columns: table => new // statement

0037: { // statement

0038: UserId = table.Column<int>(type: "int", nullable: false) // statement

0039: .Annotation("SqlServer:Identity", "1, 1"), // statement

0040: Username = table.Column<string>(type: "nvarchar(50)", maxLength: 50, nullable: false), // statement

0041: Password = table.Column<string>(type: "nvarchar(255)", maxLength: 255, nullable: false), // statement

0042: Email = table.Column<string>(type: "nvarchar(100)", maxLength: 100, nullable: false), // statement

0043: Role = table.Column<int>(type: "int", nullable: false) // statement

0044: }, // statement

0045: constraints: table => // statement

0046: { // statement

0047: table.PrimaryKey("PK\_Users", x => x.UserId); // statement

0048: }); // statement

0049:

0050: migrationBuilder.CreateTable( // statement

0051: name: "Payments", // statement

0052: columns: table => new // statement

0053: { // statement

0054: PaymentId = table.Column<int>(type: "int", nullable: false) // statement

0055: .Annotation("SqlServer:Identity", "1, 1"), // statement

0056: PolicyId = table.Column<int>(type: "int", nullable: false), // statement

0057: PaymentAmount = table.Column<decimal>(type: "decimal(10,2)", nullable: false), // statement

0058: PaymentDate = table.Column<DateTime>(type: "datetime2", nullable: false), // statement

0059: PaymentStatus = table.Column<int>(type: "int", nullable: false) // statement

0060: }, // statement

0061: constraints: table => // statement

0062: { // statement

0063: table.PrimaryKey("PK\_Payments", x => x.PaymentId); // statement

0064: table.ForeignKey( // statement

0065: name: "FK\_Payments\_Policy\_PolicyId", // statement

0066: column: x => x.PolicyId, // statement

0067: principalTable: "Policy", // statement

0068: principalColumn: "PolicyId", // statement

0069: onDelete: ReferentialAction.Cascade); // statement

0070: }); // statement

0071:

0072: migrationBuilder.CreateTable( // statement

0073: name: "Claims", // statement

0074: columns: table => new // statement

0075: { // statement

0076: ClaimId = table.Column<int>(type: "int", nullable: false) // statement

0077: .Annotation("SqlServer:Identity", "1, 1"), // statement

0078: PolicyId = table.Column<int>(type: "int", nullable: false), // statement

0079: ClaimAmount = table.Column<decimal>(type: "decimal(18,2)", nullable: false), // statement

0080: ClaimDate = table.Column<DateTime>(type: "datetime2", nullable: false), // statement

0081: ClaimStatus = table.Column<int>(type: "int", nullable: false), // statement

0082: AdjusterId = table.Column<int>(type: "int", nullable: false) // statement

0083: }, // statement

0084: constraints: table => // statement

0085: { // statement

0086: table.PrimaryKey("PK\_Claims", x => x.ClaimId); // statement

0087: table.ForeignKey( // statement

0088: name: "FK\_Claims\_Policy\_PolicyId", // statement

0089: column: x => x.PolicyId, // statement

0090: principalTable: "Policy", // statement

0091: principalColumn: "PolicyId", // statement

0092: onDelete: ReferentialAction.Cascade); // statement

0093: table.ForeignKey( // statement

0094: name: "FK\_Claims\_Users\_AdjusterId", // statement

0095: column: x => x.AdjusterId, // statement

0096: principalTable: "Users", // statement

0097: principalColumn: "UserId", // statement

0098: onDelete: ReferentialAction.Cascade); // statement

0099: }); // statement

0100:

0101: migrationBuilder.CreateTable( // statement

0102: name: "SupportTickets", // statement

0103: columns: table => new // statement

0104: { // statement

0105: TicketId = table.Column<int>(type: "int", nullable: false) // statement

0106: .Annotation("SqlServer:Identity", "1, 1"), // statement

0107: UserId = table.Column<int>(type: "int", nullable: false), // statement

0108: AssignedAgentId = table.Column<int>(type: "int", nullable: true), // statement

0109: IssueDescription = table.Column<string>(type: "nvarchar(max)", nullable: false), // statement

0110: TicketStatus = table.Column<int>(type: "int", nullable: false), // statement

0111: CreatedDate = table.Column<DateTime>(type: "datetime2", nullable: false), // statement

0112: ResolvedDate = table.Column<DateTime>(type: "datetime2", nullable: true) // statement

0113: }, // statement

0114: constraints: table => // statement

0115: { // statement

0116: table.PrimaryKey("PK\_SupportTickets", x => x.TicketId); // statement

0117: table.ForeignKey( // statement

0118: name: "FK\_SupportTickets\_Users\_UserId", // statement

0119: column: x => x.UserId, // statement

0120: principalTable: "Users", // statement

0121: principalColumn: "UserId", // statement

0122: onDelete: ReferentialAction.Cascade); // statement

0123: }); // statement

0124:

0125: migrationBuilder.CreateIndex( // statement

0126: name: "IX\_Claims\_AdjusterId", // statement

0127: table: "Claims", // statement

0128: column: "AdjusterId"); // statement

0129:

0130: migrationBuilder.CreateIndex( // statement

0131: name: "IX\_Claims\_PolicyId", // statement

0132: table: "Claims", // statement

0133: column: "PolicyId"); // statement

0134:

0135: migrationBuilder.CreateIndex( // statement

0136: name: "IX\_Payments\_PolicyId", // statement

0137: table: "Payments", // statement

0138: column: "PolicyId"); // statement

0139:

0140: migrationBuilder.CreateIndex( // statement

0141: name: "IX\_SupportTickets\_UserId", // statement

0142: table: "SupportTickets", // statement

0143: column: "UserId"); // statement

0144: } // statement

0145:

0146: /// <inheritdoc /> // comment

0147: protected override void Down(MigrationBuilder migrationBuilder) // member or method declaration

0148: { // statement

0149: migrationBuilder.DropTable( // statement

0150: name: "Claims"); // statement

0151:

0152: migrationBuilder.DropTable( // statement

0153: name: "Payments"); // statement

0154:

0155: migrationBuilder.DropTable( // statement

0156: name: "SupportTickets"); // statement

0157:

0158: migrationBuilder.DropTable( // statement

0159: name: "Policy"); // statement

0160:

0161: migrationBuilder.DropTable( // statement

0162: name: "Users"); // statement

0163: } // statement

0164: } // statement

0165: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Migrations/AutoInsuranceDbContextModelSnapshot.cs

0001: ﻿// <auto-generated /> // statement

0002: using System; // import namespace

0003: using Auto\_Insurance\_System.Data; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005: using Microsoft.EntityFrameworkCore.Infrastructure; // import namespace

0006: using Microsoft.EntityFrameworkCore.Metadata; // import namespace

0007: using Microsoft.EntityFrameworkCore.Storage.ValueConversion; // import namespace

0008:

0009: #nullable disable // statement

0010:

0011: namespace Auto\_Insurance\_System.Migrations // declare namespace

0012: { // statement

0013: [DbContext(typeof(AutoInsuranceDbContext))] // attribute/annotation

0014: partial class AutoInsuranceDbContextModelSnapshot : ModelSnapshot // class declaration

0015: { // statement

0016: protected override void BuildModel(ModelBuilder modelBuilder) // member or method declaration

0017: { // statement

0018: #pragma warning disable 612, 618 // statement

0019: modelBuilder // statement

0020: .HasAnnotation("ProductVersion", "9.0.7") // statement

0021: .HasAnnotation("Relational:MaxIdentifierLength", 128); // statement

0022:

0023: SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder); // statement

0024:

0025: modelBuilder.Entity("Auto\_Insurance\_System.Models.Claim", b => // statement

0026: { // statement

0027: b.Property<int>("ClaimId") // statement

0028: .ValueGeneratedOnAdd() // statement

0029: .HasColumnType("int"); // statement

0030:

0031: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("ClaimId")); // statement

0032:

0033: b.Property<int>("AdjusterId") // statement

0034: .HasColumnType("int"); // statement

0035:

0036: b.Property<decimal>("ClaimAmount") // statement

0037: .HasColumnType("decimal(18,2)"); // statement

0038:

0039: b.Property<DateTime>("ClaimDate") // statement

0040: .HasColumnType("datetime2"); // statement

0041:

0042: b.Property<int>("ClaimStatus") // statement

0043: .HasColumnType("int"); // statement

0044:

0045: b.Property<int>("PolicyId") // statement

0046: .HasColumnType("int"); // statement

0047:

0048: b.HasKey("ClaimId"); // statement

0049:

0050: b.HasIndex("AdjusterId"); // statement

0051:

0052: b.HasIndex("PolicyId"); // statement

0053:

0054: b.ToTable("Claims"); // statement

0055: }); // statement

0056:

0057: modelBuilder.Entity("Auto\_Insurance\_System.Models.Payment", b => // statement

0058: { // statement

0059: b.Property<int>("PaymentId") // statement

0060: .ValueGeneratedOnAdd() // statement

0061: .HasColumnType("int"); // statement

0062:

0063: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("PaymentId")); // statement

0064:

0065: b.Property<decimal>("PaymentAmount") // statement

0066: .HasColumnType("decimal(10,2)"); // statement

0067:

0068: b.Property<DateTime>("PaymentDate") // statement

0069: .HasColumnType("datetime2"); // statement

0070:

0071: b.Property<int>("PaymentStatus") // statement

0072: .HasColumnType("int"); // statement

0073:

0074: b.Property<int>("PolicyId") // statement

0075: .HasColumnType("int"); // statement

0076:

0077: b.HasKey("PaymentId"); // statement

0078:

0079: b.HasIndex("PolicyId"); // statement

0080:

0081: b.ToTable("Payments"); // statement

0082: }); // statement

0083:

0084: modelBuilder.Entity("Auto\_Insurance\_System.Models.Policy", b => // statement

0085: { // statement

0086: b.Property<int>("PolicyId") // statement

0087: .ValueGeneratedOnAdd() // statement

0088: .HasColumnType("int"); // statement

0089:

0090: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("PolicyId")); // statement

0091:

0092: b.Property<decimal>("CoverageAmount") // statement

0093: .HasColumnType("decimal(10,2)"); // statement

0094:

0095: b.Property<string>("CoverageType") // statement

0096: .IsRequired() // statement

0097: .HasMaxLength(50) // statement

0098: .HasColumnType("nvarchar(50)"); // statement

0099:

0100: b.Property<DateTime>("EndDate") // statement

0101: .HasColumnType("datetime2"); // statement

0102:

0103: b.Property<string>("PolicyNumber") // statement

0104: .IsRequired() // statement

0105: .HasMaxLength(50) // statement

0106: .HasColumnType("nvarchar(50)"); // statement

0107:

0108: b.Property<int>("PolicyStatus") // statement

0109: .HasColumnType("int"); // statement

0110:

0111: b.Property<decimal>("PremiumAmount") // statement

0112: .HasColumnType("decimal(10,2)"); // statement

0113:

0114: b.Property<DateTime>("StartDate") // statement

0115: .HasColumnType("datetime2"); // statement

0116:

0117: b.Property<string>("VehicleDetails") // statement

0118: .IsRequired() // statement

0119: .HasMaxLength(200) // statement

0120: .HasColumnType("nvarchar(200)"); // statement

0121:

0122: b.HasKey("PolicyId"); // statement

0123:

0124: b.ToTable("Policy"); // statement

0125: }); // statement

0126:

0127: modelBuilder.Entity("Auto\_Insurance\_System.Models.SupportTicket", b => // statement

0128: { // statement

0129: b.Property<int>("TicketId") // statement

0130: .ValueGeneratedOnAdd() // statement

0131: .HasColumnType("int"); // statement

0132:

0133: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("TicketId")); // statement

0134:

0135: b.Property<int?>("AssignedAgentId") // statement

0136: .HasColumnType("int"); // statement

0137:

0138: b.Property<DateTime>("CreatedDate") // statement

0139: .HasColumnType("datetime2"); // statement

0140:

0141: b.Property<string>("IssueDescription") // statement

0142: .IsRequired() // statement

0143: .HasColumnType("nvarchar(max)"); // statement

0144:

0145: b.Property<DateTime?>("ResolvedDate") // statement

0146: .HasColumnType("datetime2"); // statement

0147:

0148: b.Property<int>("TicketStatus") // statement

0149: .HasColumnType("int"); // statement

0150:

0151: b.Property<int>("UserId") // statement

0152: .HasColumnType("int"); // statement

0153:

0154: b.HasKey("TicketId"); // statement

0155:

0156: b.HasIndex("UserId"); // statement

0157:

0158: b.ToTable("SupportTickets"); // statement

0159: }); // statement

0160:

0161: modelBuilder.Entity("Auto\_Insurance\_System.Models.Users", b => // statement

0162: { // statement

0163: b.Property<int>("UserId") // statement

0164: .ValueGeneratedOnAdd() // statement

0165: .HasColumnType("int"); // statement

0166:

0167: SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("UserId")); // statement

0168:

0169: b.Property<string>("Email") // statement

0170: .IsRequired() // statement

0171: .HasMaxLength(100) // statement

0172: .HasColumnType("nvarchar(100)"); // statement

0173:

0174: b.Property<string>("Password") // statement

0175: .IsRequired() // statement

0176: .HasMaxLength(255) // statement

0177: .HasColumnType("nvarchar(255)"); // statement

0178:

0179: b.Property<int>("Role") // statement

0180: .HasColumnType("int"); // statement

0181:

0182: b.Property<string>("Username") // statement

0183: .IsRequired() // statement

0184: .HasMaxLength(50) // statement

0185: .HasColumnType("nvarchar(50)"); // statement

0186:

0187: b.HasKey("UserId"); // statement

0188:

0189: b.ToTable("Users"); // statement

0190: }); // statement

0191:

0192: modelBuilder.Entity("Auto\_Insurance\_System.Models.Claim", b => // statement

0193: { // statement

0194: b.HasOne("Auto\_Insurance\_System.Models.Users", "Adjuster") // statement

0195: .WithMany() // statement

0196: .HasForeignKey("AdjusterId") // statement

0197: .OnDelete(DeleteBehavior.Cascade) // statement

0198: .IsRequired(); // statement

0199:

0200: b.HasOne("Auto\_Insurance\_System.Models.Policy", "Policy") // statement

0201: .WithMany() // statement

0202: .HasForeignKey("PolicyId") // statement

0203: .OnDelete(DeleteBehavior.Cascade) // statement

0204: .IsRequired(); // statement

0205:

0206: b.Navigation("Adjuster"); // statement

0207:

0208: b.Navigation("Policy"); // statement

0209: }); // statement

0210:

0211: modelBuilder.Entity("Auto\_Insurance\_System.Models.Payment", b => // statement

0212: { // statement

0213: b.HasOne("Auto\_Insurance\_System.Models.Policy", "Policy") // statement

0214: .WithMany() // statement

0215: .HasForeignKey("PolicyId") // statement

0216: .OnDelete(DeleteBehavior.Cascade) // statement

0217: .IsRequired(); // statement

0218:

0219: b.Navigation("Policy"); // statement

0220: }); // statement

0221:

0222: modelBuilder.Entity("Auto\_Insurance\_System.Models.SupportTicket", b => // statement

0223: { // statement

0224: b.HasOne("Auto\_Insurance\_System.Models.Users", "User") // statement

0225: .WithMany() // statement

0226: .HasForeignKey("UserId") // statement

0227: .OnDelete(DeleteBehavior.Cascade) // statement

0228: .IsRequired(); // statement

0229:

0230: b.Navigation("User"); // statement

0231: }); // statement

0232: #pragma warning restore 612, 618 // statement

0233: } // statement

0234: } // statement

0235: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Models/Claim.cs

0001: ﻿using System.ComponentModel.DataAnnotations; // statement

0002: using System.ComponentModel.DataAnnotations.Schema; // import namespace

0003: using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; // import namespace

0004:

0005: namespace Auto\_Insurance\_System.Models // declare namespace

0006: { // statement

0007: [Table("Claims")] // attribute/annotation

0008: public class Claim // class declaration

0009: { // statement

0010: [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)] // attribute/annotation

0011: public int ClaimId { get; set; } // member or method declaration

0012:

0013: //[ForeignKey("Policy")] // comment

0014: [Required(ErrorMessage = "PolicyId is required")] // attribute/annotation

0015: public int PolicyId { get; set; } // member or method declaration

0016:

0017: [ValidateNever] // attribute/annotation

0018: public Policy? Policy { get; set; } // Navigation property (optional) // member or method declaration

0019:

0020: [Required(ErrorMessage = "Claim amount is required")] // attribute/annotation

0021: //[Column(TypeName = "decimal(10,2)")] // comment

0022: public decimal ClaimAmount { get; set; } // member or method declaration

0023:

0024: [Required(ErrorMessage = "Claim date is mandatory")] // attribute/annotation

0025: public DateTime ClaimDate { get; set; } // member or method declaration

0026:

0027: [Required(ErrorMessage = "Claim status is required")] // attribute/annotation

0028: public ClaimStatus ClaimStatus { get; set; } // member or method declaration

0029:

0030: [ForeignKey("Adjuster")] // attribute/annotation

0031: [Required(ErrorMessage = "AdjusterId is required")] // attribute/annotation

0032: public int AdjusterId { get; set; } // member or method declaration

0033:

0034: [ValidateNever] // attribute/annotation

0035: public Users? Adjuster { get; set; } // member or method declaration

0036: } // statement

0037:

0038: public enum ClaimStatus // member or method declaration

0039: { // statement

0040: OPEN, // statement

0041: APPROVED, // statement

0042: REJECTED // statement

0043: } // statement

0044: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Models/ErrorViewModel.cs

0001: namespace Auto\_Insurance\_System.Models // declare namespace

0002: { // statement

0003: public class ErrorViewModel // class declaration

0004: { // statement

0005: public string? RequestId { get; set; } // member or method declaration

0006:

0007: public bool ShowRequestId => !string.IsNullOrEmpty(RequestId); // member or method declaration

0008: } // statement

0009: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Models/Payment.cs

0001: ﻿using System; // statement

0002: using System.ComponentModel.DataAnnotations; // import namespace

0003: using System.ComponentModel.DataAnnotations.Schema; // import namespace

0004: using Microsoft.AspNetCore.Mvc.ModelBinding.Validation; // import namespace

0005:

0006: namespace Auto\_Insurance\_System.Models // declare namespace

0007: { // statement

0008: [Table("Payments")] // attribute/annotation

0009: public class Payment // class declaration

0010: { // statement

0011: [Key] // attribute/annotation

0012: [DatabaseGenerated(DatabaseGeneratedOption.Identity)] // attribute/annotation

0013: public int PaymentId { get; set; } // member or method declaration

0014:

0015: [Required(ErrorMessage = "PolicyId is required")] // attribute/annotation

0016: public int PolicyId { get; set; } // member or method declaration

0017:

0018: [ForeignKey(nameof(PolicyId))] // attribute/annotation

0019: [ValidateNever] // attribute/annotation

0020: public Policy? Policy { get; set; } // member or method declaration

0021:

0022: [Required(ErrorMessage = "Payment amount is required")] // attribute/annotation

0023: [Column(TypeName = "decimal(10,2)")] // attribute/annotation

0024: public decimal PaymentAmount { get; set; } // member or method declaration

0025:

0026: [Required(ErrorMessage = "Payment date is required")] // attribute/annotation

0027: public DateTime PaymentDate { get; set; } // member or method declaration

0028:

0029: [Required(ErrorMessage = "Payment status is required")] // attribute/annotation

0030: public PaymentStatus PaymentStatus { get; set; } // member or method declaration

0031: } // statement

0032:

0033: public enum PaymentStatus // member or method declaration

0034: { // statement

0035: SUCCESS, // statement

0036: FAILED, // statement

0037: PENDING // statement

0038: } // statement

0039: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Models/Policy.cs

0001: ﻿using System.ComponentModel.DataAnnotations; // statement

0002: using System.ComponentModel.DataAnnotations.Schema; // import namespace

0003:

0004: namespace Auto\_Insurance\_System.Models // declare namespace

0005: { // statement

0006: [Table("Policy")] // attribute/annotation

0007: public class Policy // class declaration

0008: { // statement

0009: [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)] // attribute/annotation

0010: public int PolicyId { get; set; } // member or method declaration

0011:

0012: [Required(ErrorMessage = "Policy Number is mandatory"), MaxLength(50)] // attribute/annotation

0013: public string PolicyNumber { get; set; } // member or method declaration

0014:

0015: [Required(ErrorMessage = "Vehicle Details are mandatory"), MaxLength(200)] // attribute/annotation

0016: public string VehicleDetails { get; set; } // member or method declaration

0017:

0018: [Required(ErrorMessage = "Coverage Amount is required")] // attribute/annotation

0019: [Column(TypeName = "decimal(10,2)")] // attribute/annotation

0020: public decimal CoverageAmount { get; set; } // member or method declaration

0021:

0022: [Required(ErrorMessage = "Coverage Type is required"), MaxLength(50)] // attribute/annotation

0023: public string CoverageType { get; set; } // member or method declaration

0024:

0025: [Required(ErrorMessage = "Premium Amount is required")] // attribute/annotation

0026: [Column(TypeName = "decimal(10,2)")] // attribute/annotation

0027: public decimal PremiumAmount { get; set; } // member or method declaration

0028:

0029: [Required(ErrorMessage = "Policy Start Date is mandatory")] // attribute/annotation

0030: public DateTime StartDate { get; set; } // member or method declaration

0031:

0032: [Required(ErrorMessage = "Policy End Date is mandatory")] // attribute/annotation

0033: public DateTime EndDate { get; set; } // member or method declaration

0034:

0035: [Required(ErrorMessage = "Policy Status is required")] // attribute/annotation

0036: public PolicyStatus PolicyStatus { get; set; } // member or method declaration

0037: } // statement

0038:

0039: public enum PolicyStatus // member or method declaration

0040: { // statement

0041: ACTIVE, // statement

0042: INACTIVE, // statement

0043: RENEWED // statement

0044: } // statement

0045: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Models/SupportTicket.cs

0001: ﻿using System; // statement

0002: using System.ComponentModel.DataAnnotations; // import namespace

0003: using System.ComponentModel.DataAnnotations.Schema; // import namespace

0004:

0005: namespace Auto\_Insurance\_System.Models // declare namespace

0006: { // statement

0007: [Table("SupportTickets")] // attribute/annotation

0008: public class SupportTicket // class declaration

0009: { // statement

0010: [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)] // attribute/annotation

0011: public int TicketId { get; set; } // member or method declaration

0012:

0013: [Required(ErrorMessage = "User Id is required")] // attribute/annotation

0014: public int UserId { get; set; } // member or method declaration

0015:

0016: [ForeignKey(nameof(UserId))] // attribute/annotation

0017: public Users User { get; set; } // member or method declaration

0018:

0019: public int? AssignedAgentId { get; set; } // member or method declaration

0020:

0021: [Required(ErrorMessage = "Issue description is required")] // attribute/annotation

0022: public string IssueDescription { get; set; } // member or method declaration

0023:

0024: [Required(ErrorMessage = "Ticket status is required")] // attribute/annotation

0025: public TicketStatus TicketStatus { get; set; } // member or method declaration

0026:

0027: [Required(ErrorMessage = "Created date is required")] // attribute/annotation

0028: public DateTime CreatedDate { get; set; } // member or method declaration

0029:

0030: public DateTime? ResolvedDate { get; set; } // member or method declaration

0031: } // statement

0032:

0033: public enum TicketStatus // member or method declaration

0034: { // statement

0035: OPEN, // statement

0036: RESOLVED // statement

0037: } // statement

0038: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Models/Users.cs

0001: ﻿using System; // statement

0002: using System.ComponentModel.DataAnnotations; // import namespace

0003: using System.ComponentModel.DataAnnotations.Schema; // import namespace

0004:

0005: namespace Auto\_Insurance\_System.Models // declare namespace

0006: { // statement

0007: [Table("Users")] // attribute/annotation

0008: public class Users // class declaration

0009: { // statement

0010: [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)] // attribute/annotation

0011: public int UserId { get; set; } // member or method declaration

0012:

0013: [Required(ErrorMessage = "Username is required")] // attribute/annotation

0014: [MaxLength(50, ErrorMessage = "Username cannot exceed 50 characters")] // attribute/annotation

0015: public string Username { get; set; } // member or method declaration

0016:

0017: [Required(ErrorMessage = "Password is required")] // attribute/annotation

0018: [MaxLength(255, ErrorMessage = "Password cannot exceed 255 characters")] // attribute/annotation

0019: public string Password { get; set; } // member or method declaration

0020:

0021: [Required(ErrorMessage = "Email is required")] // attribute/annotation

0022: [MaxLength(100, ErrorMessage = "Email cannot exceed 100 characters")] // attribute/annotation

0023: [EmailAddress(ErrorMessage = "Invalid email format")] // attribute/annotation

0024: public string Email { get; set; } // member or method declaration

0025:

0026: [Required(ErrorMessage = "Role is required")] // attribute/annotation

0027: public UserRole Role { get; set; } // member or method declaration

0028: } // statement

0029:

0030: public enum UserRole // member or method declaration

0031: { // statement

0032: ADMIN, // statement

0033: AGENT, // statement

0034: CUSTOMER // statement

0035: } // statement

0036: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Program.cs

0001: using Auto\_Insurance\_System.Data; // import namespace

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Services; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005: using Auto\_Insurance\_System.Filters; // import namespace

0006:

0007: var builder = WebApplication.CreateBuilder(args); // local variable declaration

0008: builder.Services.AddDbContextPool<AutoInsuranceDbContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("SqlCon"))); // statement

0009: builder.Services.AddTransient<IPolicyService, PolicyService>(); // statement

0010: builder.Services.AddTransient<IClaimService, ClaimService>(); // statement

0011: builder.Services.AddTransient<IPaymentService, PaymentService>(); // statement

0012: builder.Services.AddTransient<ISupportTicketService, SupportTicketService>(); // statement

0013: builder.Services.AddTransient<IUsersService, UsersService>(); // statement

0014: builder.Services.AddSession(); // statement

0015: // Add services to the container. // comment

0016: builder.Services.AddControllersWithViews(options => // statement

0017: { // statement

0018: options.Filters.Add<AutoInsuranceExceptionFilter>(); // statement

0019: }); // statement

0020:

0021: var app = builder.Build(); // local variable declaration

0022:

0023: // Configure the HTTP request pipeline. // comment

0024: if (!app.Environment.IsDevelopment()) // conditional check

0025: { // statement

0026: app.UseExceptionHandler("/Home/Error"); // statement

0027: // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts. // comment

0028: app.UseHsts(); // statement

0029: } // statement

0030:

0031: app.UseHttpsRedirection(); // statement

0032: app.UseRouting(); // statement

0033:

0034: app.UseSession(); // statement

0035: app.UseAuthorization(); // statement

0036:

0037: app.MapStaticAssets(); // statement

0038:

0039: app.MapControllerRoute( // statement

0040: name: "default", // statement

0041: pattern: "{controller=Home}/{action=Landing}/{id?}") // statement

0042: .WithStaticAssets(); // statement

0043:

0044:

0045: app.Run(); // statement

# Auto-Insurance/Auto\_Insurance\_System/Properties/launchSettings.json

0001: ﻿{ // statement

0002: "$schema": "https://json.schemastore.org/launchsettings.json", // statement

0003: "profiles": { // statement

0004: "http": { // statement

0005: "commandName": "Project", // statement

0006: "dotnetRunMessages": true, // statement

0007: "launchBrowser": true, // statement

0008: "launchUrl": "landing", // statement

0009: "applicationUrl": "http://localhost:5190", // statement

0010: "environmentVariables": { // statement

0011: "ASPNETCORE\_ENVIRONMENT": "Development" // statement

0012: } // statement

0013: }, // statement

0014: "https": { // statement

0015: "commandName": "Project", // statement

0016: "dotnetRunMessages": true, // statement

0017: "launchBrowser": true, // statement

0018: "launchUrl": "landing", // statement

0019: "applicationUrl": "https://localhost:7194;http://localhost:5190", // statement

0020: "environmentVariables": { // statement

0021: "ASPNETCORE\_ENVIRONMENT": "Development" // statement

0022: } // statement

0023: } // statement

0024: } // statement

0025: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Services/ClaimService.cs

0001: ﻿using Auto\_Insurance\_System.Data; // statement

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Models; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005:

0006: namespace Auto\_Insurance\_System.Services // declare namespace

0007: { // statement

0008: public class ClaimService : IClaimService // class declaration

0009: { // statement

0010: private readonly AutoInsuranceDbContext context; // member or method declaration

0011: public ClaimService(AutoInsuranceDbContext db) // member or method declaration

0012: { // statement

0013: context = db; // statement

0014: } // statement

0015:

0016: public List<Claim> GetAllClaims() // member or method declaration

0017: { // statement

0018: DbSet<Claim> claims = context.Claims; // statement

0019: return claims.ToList(); // return value

0020: } // statement

0021:

0022: public Claim GetClaimDetails(string id) // member or method declaration

0023: { // statement

0024: int claimId = Convert.ToInt32(id); // statement

0025: DbSet<Claim> claims = context.Claims; // statement

0026: Claim claim = claims.Find(claimId); // statement

0027: return claim; // return value

0028: } // statement

0029:

0030: public bool SubmitClaim(Claim obj) // member or method declaration

0031: { // statement

0032: try // begin try block

0033: { // statement

0034: DbSet<Claim> claims = context.Claims; // statement

0035: claims.Add(obj); // statement

0036: context.SaveChanges(); // statement

0037: return true; // return value

0038: } // statement

0039: catch (Exception ex) // handle exception

0040: { // statement

0041: Console.WriteLine(ex.ToString()); // statement

0042: return false; // return value

0043: } // statement

0044: } // statement

0045:

0046: public bool UpdateClaimStatus(Claim obj) // member or method declaration

0047: { // statement

0048: try // begin try block

0049: { // statement

0050: DbSet<Claim> claims = context.Claims; // statement

0051: Claim c = claims.Find(obj.ClaimId); // statement

0052: if (c != null) // conditional check

0053: { // statement

0054: c.PolicyId = obj.PolicyId; // statement

0055: c.ClaimStatus = obj.ClaimStatus; // statement

0056: c.ClaimAmount = obj.ClaimAmount; // statement

0057: c.ClaimDate = obj.ClaimDate; // statement

0058: c.AdjusterId = obj.AdjusterId; // statement

0059: context.SaveChanges(); // statement

0060: return true; // return value

0061: } // statement

0062: else // statement

0063: { // statement

0064: return false; // return value

0065: } // statement

0066: } // statement

0067: catch (Exception ex) // handle exception

0068: { // statement

0069: return false; // return value

0070: } // statement

0071: } // statement

0072: } // statement

0073: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Services/PaymentService.cs

0001: ﻿using Auto\_Insurance\_System.Data; // statement

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Models; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005: using System; // import namespace

0006: using System.Collections.Generic; // import namespace

0007: using System.Diagnostics; // import namespace

0008: using System.Linq; // import namespace

0009:

0010: namespace Auto\_Insurance\_System.Services // declare namespace

0011: { // statement

0012: public class PaymentService : IPaymentService // class declaration

0013: { // statement

0014: private readonly AutoInsuranceDbContext context; // member or method declaration

0015:

0016: public PaymentService(AutoInsuranceDbContext db) // member or method declaration

0017: { // statement

0018: context = db; // statement

0019: } // statement

0020:

0021: public bool MakePayment(Payment obj) // member or method declaration

0022: { // statement

0023: try // begin try block

0024: { // statement

0025: var policy = context.Policy.Find(obj.PolicyId); // local variable declaration

0026: if (policy == null) return false; // conditional check

0027: if (obj.PaymentAmount != policy.PremiumAmount) return false; // conditional check

0028:

0029: DbSet<Payment> payments = context.Payments; // statement

0030: payments.Add(obj); // statement

0031: context.SaveChanges(); // statement

0032: return true; // return value

0033: } // statement

0034: catch (Exception ex) // handle exception

0035: { // statement

0036: Console.WriteLine(ex.ToString()); // statement

0037: return false; // return value

0038: } // statement

0039: } // statement

0040:

0041: public Payment GetPaymentDetails(string id) // member or method declaration

0042: { // statement

0043: try // begin try block

0044: { // statement

0045: int payId = Convert.ToInt32(id); // statement

0046: DbSet<Payment> payments = context.Payments; // statement

0047: Payment p = payments.Find(payId); // statement

0048: return p; // return value

0049: } // statement

0050: catch (Exception ex) // handle exception

0051: { // statement

0052: Debug.WriteLine("Error fetching payment details: " + ex.Message); // statement

0053: return null; // return value

0054: } // statement

0055: } // statement

0056:

0057: public List<Payment> GetPaymentsByPolicy(int policyId) // member or method declaration

0058: { // statement

0059: try // begin try block

0060: { // statement

0061: DbSet<Payment> payments = context.Payments; // statement

0062: return payments // return value

0063: .Where(p => p.PolicyId == policyId) // statement

0064: .Include(p => p.Policy) // statement

0065: .ToList(); // statement

0066: } // statement

0067: catch (Exception ex) // handle exception

0068: { // statement

0069: Debug.WriteLine("Error fetching payments by policy: " + ex.Message); // statement

0070: return new List<Payment>(); // return value

0071: } // statement

0072: } // statement

0073: } // statement

0074: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Services/PolicyService.cs

0001: ﻿using Auto\_Insurance\_System.Data; // statement

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Models; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005: using System.Diagnostics; // import namespace

0006:

0007: namespace Auto\_Insurance\_System.Services // declare namespace

0008: { // statement

0009: public class PolicyService : IPolicyService // class declaration

0010: { // statement

0011: private readonly AutoInsuranceDbContext context; // member or method declaration

0012: public PolicyService(AutoInsuranceDbContext db) // member or method declaration

0013: { // statement

0014: context = db; // statement

0015: } // statement

0016:

0017: public bool CreatePolicy(Policy obj) // member or method declaration

0018: { // statement

0019: try // begin try block

0020: { // statement

0021: DbSet<Policy> policies = context.Policy; // statement

0022: policies.Add(obj); // statement

0023: context.SaveChanges(); // statement

0024: return true; // return value

0025: } // statement

0026: catch(Exception ex) // handle exception

0027: { // statement

0028: Debug.WriteLine(ex); // statement

0029: return false; // return value

0030: } // statement

0031:

0032: } // statement

0033:

0034: public List<Policy> GetAllPolicies() // member or method declaration

0035: { // statement

0036: DbSet<Policy> policies = context.Policy; // statement

0037: return policies.ToList(); // return value

0038: } // statement

0039:

0040: public Policy GetPolicyById(string id) // member or method declaration

0041: { // statement

0042: int policyId = Convert.ToInt32(id); // statement

0043: DbSet<Policy> policies = context.Policy; // statement

0044: Policy policy = policies.Find(policyId); // statement

0045: return policy; // return value

0046: } // statement

0047:

0048: public bool UpdatePolicy(Policy obj) // member or method declaration

0049: { // statement

0050: try // begin try block

0051: { // statement

0052: DbSet<Policy> policies = context.Policy; // statement

0053: Policy policy = policies.Find(obj.PolicyId); // statement

0054: if (policy != null) // conditional check

0055: { // statement

0056: policy.PolicyNumber = obj.PolicyNumber; // statement

0057: policy.PolicyStatus = obj.PolicyStatus; // statement

0058: policy.VehicleDetails = obj.VehicleDetails; // statement

0059: policy.CoverageAmount = obj.CoverageAmount; // statement

0060: policy.CoverageType = obj.CoverageType; // statement

0061: policy.PremiumAmount = obj.PremiumAmount; // statement

0062: policy.StartDate = obj.StartDate; // statement

0063: policy.EndDate = obj.EndDate; // statement

0064:

0065: context.SaveChanges(); // statement

0066: return true; // return value

0067: } // statement

0068:

0069: else // statement

0070: { // statement

0071: return false; // return value

0072: } // statement

0073: } // statement

0074: catch (Exception ex) // handle exception

0075: { // statement

0076: Debug.WriteLine("-----------------------------------------------"); // statement

0077: Debug.WriteLine(ex.Message); // statement

0078: return false; // return value

0079: } // statement

0080: } // statement

0081:

0082: public bool DeletePolicy(string id) // member or method declaration

0083: { // statement

0084: try // begin try block

0085: { // statement

0086: DbSet<Policy> policies = context.Policy; // statement

0087: int policyId = Convert.ToInt32(id); // statement

0088: Policy policy = policies.Find(policyId); // statement

0089: if (policy != null) // conditional check

0090: { // statement

0091: policies.Remove(policy); // statement

0092: context.SaveChanges(); // statement

0093: return true; // return value

0094: } // statement

0095:

0096: else // statement

0097: { // statement

0098: return false; // return value

0099: } // statement

0100: } // statement

0101: catch (Exception ex) // handle exception

0102: { // statement

0103: Debug.WriteLine("-----------------------------------------------"); // statement

0104: Debug.WriteLine(ex.Message); // statement

0105: return false; // return value

0106: } // statement

0107: } // statement

0108:

0109: } // statement

0110: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Services/SupportTicketService.cs

0001: ﻿using Auto\_Insurance\_System.Data; // statement

0002: using Auto\_Insurance\_System.Models; // import namespace

0003: using Auto\_Insurance\_System.Interfaces; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005:

0006: namespace Auto\_Insurance\_System.Services // declare namespace

0007: { // statement

0008: public class SupportTicketService : ISupportTicketService // class declaration

0009: { // statement

0010: private readonly AutoInsuranceDbContext context; // member or method declaration

0011: public SupportTicketService(AutoInsuranceDbContext db) // member or method declaration

0012: { // statement

0013: context = db; // statement

0014: } // statement

0015: public bool CreateTicket(SupportTicket ticket) // member or method declaration

0016: { // statement

0017: try // begin try block

0018: { // statement

0019: DbSet<SupportTicket> tickets = context.SupportTickects; // statement

0020: tickets.Add(ticket); // statement

0021: context.SaveChanges(); // statement

0022: return true; // return value

0023: } // statement

0024: catch (Exception ex) // handle exception

0025: { // statement

0026: return false; // return value

0027: } // statement

0028: } // statement

0029: public SupportTicket GetTickectDetails(int id) // member or method declaration

0030: { // statement

0031: return context.SupportTickects.Include(t=>t.User).FirstOrDefault(t => t.TicketId == id); // return value

0032: } // statement

0033:

0034: public bool ResolvedTickect(int ticketId) // member or method declaration

0035: { // statement

0036: var t = context.SupportTickects.Find(ticketId); // local variable declaration

0037: if (t == null) return false; // conditional check

0038: t.TicketStatus = TicketStatus.RESOLVED; // statement

0039: t.ResolvedDate = DateTime.Now; // statement

0040: context.SaveChanges(); // statement

0041: return true; // return value

0042: } // statement

0043:

0044: public bool AssignTicket(int ticketId, int agentUserId) // member or method declaration

0045: { // statement

0046: var t = context.SupportTickects.Find(ticketId); // local variable declaration

0047: if (t == null) return false; // conditional check

0048: t.AssignedAgentId = agentUserId; // statement

0049: context.SaveChanges(); // statement

0050: return true; // return value

0051: } // statement

0052:

0053: public List<SupportTicket> GetAllTickects() // member or method declaration

0054: { // statement

0055: return context.SupportTickects.Include(t => t.User).ToList(); // return value

0056: } // statement

0057:

0058: } // statement

0059:

0060:

0061: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Services/UsersService.cs

0001: ﻿using Auto\_Insurance\_System.Data; // statement

0002: using Auto\_Insurance\_System.Interfaces; // import namespace

0003: using Auto\_Insurance\_System.Models; // import namespace

0004: using Microsoft.EntityFrameworkCore; // import namespace

0005: using System.Linq; // import namespace

0006:

0007: namespace Auto\_Insurance\_System.Services // declare namespace

0008: { // statement

0009: public class UsersService : IUsersService // class declaration

0010: { // statement

0011: private readonly AutoInsuranceDbContext context; // member or method declaration

0012:

0013: public UsersService(AutoInsuranceDbContext context) // member or method declaration

0014: { // statement

0015: this.context = context; // statement

0016: } // statement

0017:

0018: public bool RegisterUser(Users user) // member or method declaration

0019: { // statement

0020: try // begin try block

0021: { // statement

0022: DbSet<Users> users = context.Users; // statement

0023: users.Add(user); // statement

0024: context.SaveChanges(); // statement

0025: return true; // return value

0026: } // statement

0027: catch (Exception ex) // handle exception

0028: { // statement

0029: return false; // return value

0030: } // statement

0031: } // statement

0032:

0033: public Users Login(string username, string password) // member or method declaration

0034: { // statement

0035: return context.Users.FirstOrDefault(u => u.Username == username && u.Password == password); // return value

0036: } // statement

0037:

0038: public Users GetUserProfile(int userId) // member or method declaration

0039: { // statement

0040: return context.Users.Find(userId); // return value

0041: } // statement

0042:

0043: public List<Users> GetAllUsers() // member or method declaration

0044: { // statement

0045: return context.Users.ToList(); // return value

0046: } // statement

0047: public bool Logout(int userId) // member or method declaration

0048: { // statement

0049: Users user =GetAllUsers().FirstOrDefault(u => u.UserId == userId); // statement

0050: if (user == null) // conditional check

0051: { // statement

0052: return false; // return value

0053: } // statement

0054: context.Users.Remove(user); // statement

0055: context.SaveChanges(); // statement

0056: return true; // return value

0057: } // statement

0058:

0059:

0060:

0061: public bool ChangePassword(Users userLogin) // member or method declaration

0062: { // statement

0063: Users user = GetAllUsers().FirstOrDefault(u => u.UserId == userLogin.UserId); // statement

0064: if (user == null) // conditional check

0065: { // statement

0066: return false; // return value

0067: } // statement

0068: user.Password = userLogin.Password; // statement

0069: context.SaveChanges(); // statement

0070: return true; // return value

0071: } // statement

0072:

0073:

0074: public bool VerifyUserLogin(Users userLogin) // member or method declaration

0075: { // statement

0076: Users user = context.Users.FirstOrDefault(u => // statement

0077: u.Username == userLogin.Username && // statement

0078: u.Password == userLogin.Password && // statement

0079: u.Role == userLogin.Role); // statement

0080:

0081: return user != null; // return value

0082: } // statement

0083:

0084: } // statement

0085: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/ClaimDetails.cshtml

0001: @model Auto\_Insurance\_System.Models.Claim // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Claim Details"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Get Claim Details</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm mb-3"> // markup element

0013: <div class="card-body"> // markup element

0014: <form method="post" class="row g-2"> // markup element

0015: <div class="col-md-4"> // markup element

0016: <input class="form-control" type="number" name="claimId" placeholder="Enter Claim Id" required /> // markup element

0017: </div> // markup element

0018: <div class="col-md-2"> // markup element

0019: <button class="btn btn-primary w-100" type="submit">Search</button> // markup element

0020: </div> // markup element

0021: </form> // markup element

0022: @if (ViewBag.Error != null) // Razor conditional rendering

0023: { // statement

0024: <div class="alert alert-warning mt-3">@ViewBag.Error</div> // markup element

0025: } // statement

0026: </div> // markup element

0027: </div> // markup element

0028:

0029: @if (Model != null) // Razor conditional rendering

0030: { // statement

0031: <div class="card shadow-sm"> // markup element

0032: <div class="card-header bg-light">Claim #@Model.ClaimId</div> // markup element

0033: <div class="card-body"> // markup element

0034: <div class="row g-3"> // markup element

0035: <div class="col-md-3"><strong>Policy:</strong> @Model.PolicyId</div> // markup element

0036: <div class="col-md-3"><strong>Amount:</strong> @String.Format("{0:N2}", Model.ClaimAmount)</div> // markup element

0037: <div class="col-md-3"><strong>Date:</strong> @Model.ClaimDate.ToString("dd-MMM-yyyy")</div> // markup element

0038: <div class="col-md-3"><strong>Status:</strong> @Model.ClaimStatus</div> // markup element

0039: <div class="col-md-3"><strong>Adjuster:</strong> @Model.AdjusterId</div> // markup element

0040: </div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: } // statement

0044: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/ClaimSubmit.cshtml

0001: @model Auto\_Insurance\_System.Models.Claim // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Submit Claim"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Submit Claim</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div asp-validation-summary="All" class="text-danger"></div> // markup element

0025:

0026: <div class="col-12"> // markup element

0027: <label class="form-label">Policy Id</label> // markup element

0028: <input class="form-control" type="number" asp-for="PolicyId" required /> // markup element

0029: <div class="form-text">Enter an existing policy id. Create one first if needed.</div> // markup element

0030: <span class="text-danger" asp-validation-for="PolicyId"></span> // markup element

0031: </div> // markup element

0032:

0033: <div class="col-12"> // markup element

0034: <label class="form-label">Amount</label> // markup element

0035: <input class="form-control" type="number" step="0.01" asp-for="ClaimAmount" required /> // markup element

0036: <span class="text-danger" asp-validation-for="ClaimAmount"></span> // markup element

0037: </div> // markup element

0038:

0039: <div class="col-12"> // markup element

0040: <label class="form-label">Date</label> // markup element

0041: <input class="form-control" type="date" asp-for="ClaimDate" required /> // markup element

0042: <span class="text-danger" asp-validation-for="ClaimDate"></span> // markup element

0043: </div> // markup element

0044:

0045: <div class="col-12"> // markup element

0046: <label class="form-label">Status</label> // markup element

0047: <select class="form-select" asp-for="ClaimStatus"> // markup element

0048: <option>OPEN</option> // markup element

0049: <option>APPROVED</option> // markup element

0050: <option>REJECTED</option> // markup element

0051: </select> // markup element

0052: <span class="text-danger" asp-validation-for="ClaimStatus"></span> // markup element

0053: </div> // markup element

0054:

0055: <div class="col-12"> // markup element

0056: <label class="form-label">Adjuster Id</label> // markup element

0057: <input class="form-control" type="number" asp-for="AdjusterId" required /> // markup element

0058: <div class="form-text">Enter an existing user id (e.g., an AGENT).</div> // markup element

0059: <span class="text-danger" asp-validation-for="AdjusterId"></span> // markup element

0060: </div> // markup element

0061:

0062: <div class="col-12 d-flex gap-2"> // markup element

0063: <button class="btn btn-primary">Submit</button> // markup element

0064: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Cancel</a> // markup element

0065: </div> // markup element

0066: </form> // markup element

0067: </div> // markup element

0068: </div> // markup element

0069: </div> // markup element

0070:

0071: @section Scripts{ // statement

0072: <partial name="\_ValidationScriptsPartial" /> // markup element

0073: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/ClaimUpdateStatus.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Update Claim Status"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0007: <h2 class="mb-0">Update Claim Status</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0009: </div> // markup element

0010:

0011: @if (ViewBag.Error != null) // Razor conditional rendering

0012: { // statement

0013: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0014: } // statement

0015: @if (ViewBag.Success != null) // Razor conditional rendering

0016: { // statement

0017: <div class="alert alert-success">@ViewBag.Success</div> // markup element

0018: } // statement

0019:

0020: <div class="card shadow-sm"> // markup element

0021: <div class="card-body"> // markup element

0022: <form method="post" class="row g-3"> // markup element

0023: <div class="col-md-4"> // markup element

0024: <label class="form-label">Claim Id</label> // markup element

0025: <input class="form-control" type="number" name="claimId" required /> // markup element

0026: </div> // markup element

0027: <div class="col-md-4"> // markup element

0028: <label class="form-label">Status</label> // markup element

0029: <select class="form-select" name="status"> // markup element

0030: <option value="OPEN">OPEN</option> // markup element

0031: <option value="APPROVED">APPROVED</option> // markup element

0032: <option value="REJECTED">REJECTED</option> // markup element

0033: </select> // markup element

0034: </div> // markup element

0035: <div class="col-md-4"> // markup element

0036: <label class="form-label">Adjuster Id (optional)</label> // markup element

0037: <input class="form-control" type="number" name="adjusterId" /> // markup element

0038: </div> // markup element

0039: <div class="col-12"> // markup element

0040: <button class="btn btn-primary">Update</button> // markup element

0041: </div> // markup element

0042: </form> // markup element

0043: </div> // markup element

0044: </div> // markup element

0045: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/Claims.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Claim> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Claims"; // statement

0004: } // statement

0005:

0006: <div class="container-fluid py-3"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Claims</h2> // markup element

0009: <div class="d-flex gap-2"> // markup element

0010: <button class="btn btn-success" data-bs-toggle="modal" data-bs-target="#claimModal">Submit Claim</button> // markup element

0011: <a class="btn btn-outline-secondary" asp-action="Console">Admin Console</a> // markup element

0012: </div> // markup element

0013: </div> // markup element

0014:

0015: <div class="card shadow-sm"> // markup element

0016: <div class="card-body"> // markup element

0017: <div class="row g-2 mb-3"> // markup element

0018: <div class="col-md-3"> // markup element

0019: <select id="claimStatusFilter" class="form-select"> // markup element

0020: <option value="">All Statuses</option> // markup element

0021: <option value="OPEN">Open</option> // markup element

0022: <option value="APPROVED">Approved</option> // markup element

0023: <option value="REJECTED">Rejected</option> // markup element

0024: </select> // markup element

0025: </div> // markup element

0026: <div class="col-md-4"> // markup element

0027: <input id="claimSearch" class="form-control" placeholder="Search by claim id, policy id or adjuster" /> // markup element

0028: </div> // markup element

0029: </div> // markup element

0030:

0031: <div class="table-responsive"> // markup element

0032: <table class="table table-hover align-middle" id="claimsTable"> // markup element

0033: <thead class="table-light"> // markup element

0034: <tr> // markup element

0035: <th>Claim Id</th> // markup element

0036: <th>Policy Id</th> // markup element

0037: <th>Amount</th> // markup element

0038: <th>Date</th> // markup element

0039: <th>Status</th> // markup element

0040: <th>Adjuster</th> // markup element

0041: <th>Actions</th> // markup element

0042: </tr> // markup element

0043: </thead> // markup element

0044: <tbody> // markup element

0045: @foreach (var c in Model.OrderByDescending(x => x.ClaimDate)) // statement

0046: { // statement

0047: <tr data-status="@c.ClaimStatus"> // markup element

0048: <td>@c.ClaimId</td> // markup element

0049: <td>@c.PolicyId</td> // markup element

0050: <td>@String.Format("{0:N2}", c.ClaimAmount)</td> // markup element

0051: <td>@c.ClaimDate.ToString("dd-MMM-yyyy")</td> // markup element

0052: <td><span class="badge @GetClaimBadge(c.ClaimStatus)">@c.ClaimStatus</span></td> // markup element

0053: <td>@c.AdjusterId</td> // markup element

0054: <td> // markup element

0055: <button class="btn btn-sm btn-outline-success me-1" onclick="openStatus(@c.ClaimId, 'APPROVED')">Approve</button> // markup element

0056: <button class="btn btn-sm btn-outline-danger me-1" onclick="openStatus(@c.ClaimId, 'REJECTED')">Reject</button> // markup element

0057: <button class="btn btn-sm btn-outline-secondary" onclick="prefillClaim(@c.ClaimId,@c.PolicyId,@c.ClaimAmount,'@c.ClaimDate.ToString("yyyy-MM-dd")',@((int)c.ClaimStatus),@c.AdjusterId)" data-bs-toggle="modal" data-bs-target="#claimModal">Edit</button> // markup element

0058: </td> // markup element

0059: </tr> // markup element

0060: } // statement

0061: </tbody> // markup element

0062: </table> // markup element

0063: </div> // markup element

0064: </div> // markup element

0065: </div> // markup element

0066: </div> // markup element

0067:

0068: <!-- Claim Modal (reuse) --> // markup element

0069: <div class="modal fade" id="claimModal" tabindex="-1" aria-hidden="true"> // markup element

0070: <div class="modal-dialog"> // markup element

0071: <div class="modal-content"> // markup element

0072: <div class="modal-header"><h5 class="modal-title">Claim</h5><button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button></div> // markup element

0073: <div class="modal-body"> // markup element

0074: <form id="claimForm" class="row g-2"> // markup element

0075: <input type="hidden" name="ClaimId" id="ClaimId" /> // markup element

0076: <div class="col-12"><input class="form-control" type="number" name="PolicyId" id="PolicyId" placeholder="Policy Id" required></div> // markup element

0077: <div class="col-12"><input class="form-control" type="number" step="0.01" name="ClaimAmount" id="ClaimAmount" placeholder="Amount" required></div> // markup element

0078: <div class="col-12"><input class="form-control" type="date" name="ClaimDate" id="ClaimDate" required></div> // markup element

0079: <div class="col-12"> // markup element

0080: <select class="form-select" name="ClaimStatus" id="ClaimStatus" required> // markup element

0081: <option value="OPEN">OPEN</option> // markup element

0082: <option value="APPROVED">APPROVED</option> // markup element

0083: <option value="REJECTED">REJECTED</option> // markup element

0084: </select> // markup element

0085: </div> // markup element

0086: <div class="col-12"><input class="form-control" type="number" name="AdjusterId" id="AdjusterId" placeholder="Adjuster Id" required></div> // markup element

0087: </form> // markup element

0088: </div> // markup element

0089: <div class="modal-footer"> // markup element

0090: <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button> // markup element

0091: <button type="button" class="btn btn-primary" id="submitClaimBtn">Save</button> // markup element

0092: </div> // markup element

0093: </div> // markup element

0094: </div> // markup element

0095: </div> // markup element

0096:

0097: @section Scripts{ // statement

0098: <script> // markup element

0099: function prefillClaim(id, policyId, amount, date, status, adjuster){ // statement

0100: document.getElementById('ClaimId').value=id||''; // statement

0101: document.getElementById('PolicyId').value=policyId||''; // statement

0102: document.getElementById('ClaimAmount').value=amount||''; // statement

0103: document.getElementById('ClaimDate').value=date||''; // statement

0104: const sel=document.getElementById('ClaimStatus'); // statement

0105: sel.value = status===0?'OPEN':status===1?'APPROVED':'REJECTED'; // statement

0106: document.getElementById('AdjusterId').value=adjuster||''; // statement

0107: } // statement

0108:

0109: function openStatus(id, status){ // statement

0110: const form=new FormData(); form.append('claimId', id); form.append('status', status); // statement

0111: fetch('/Admin/AdminUpdateClaimStatus', {method:'POST', body:form}).then(r=>{ // statement

0112: if(r.ok){ location.reload(); } else { alert('Failed to update'); } // conditional check

0113: }); // statement

0114: } // statement

0115:

0116: // Filters // comment

0117: const statusFilter = document.getElementById('claimStatusFilter'); // statement

0118: const search = document.getElementById('claimSearch'); // statement

0119: const table = document.getElementById('claimsTable'); // statement

0120: function applyFilters() { // statement

0121: const status = statusFilter.value; // statement

0122: const q = search.value.toLowerCase(); // statement

0123: Array.from(table.querySelectorAll('tbody tr')).forEach(r => { // statement

0124: const matchStatus = !status || r.getAttribute('data-status') === status; // statement

0125: const text = r.innerText.toLowerCase(); // statement

0126: const matchText = text.includes(q); // statement

0127: r.style.display = matchStatus && matchText ? '' : 'none'; // statement

0128: }); // statement

0129: } // statement

0130: statusFilter.addEventListener('change', applyFilters); // statement

0131: search.addEventListener('input', applyFilters); // statement

0132:

0133: // Submit/save claim // comment

0134: const submitClaimBtn=document.getElementById('submitClaimBtn'); // statement

0135: submitClaimBtn.addEventListener('click', async ()=>{ // statement

0136: const form=new FormData(document.getElementById('claimForm')); // statement

0137: const res=await fetch('/Admin/AdminSubmitClaim',{method:'POST', body:form}); // statement

0138: if(res.ok){ location.reload(); } else { alert('Failed'); } // conditional check

0139: }); // statement

0140: </script> // markup element

0141: } // statement

0142:

0143: @functions{ // statement

0144: string GetClaimBadge(ClaimStatus status) // statement

0145: { // statement

0146: return status switch // return value

0147: { // statement

0148: ClaimStatus.OPEN => "bg-warning text-dark", // statement

0149: ClaimStatus.APPROVED => "bg-success", // statement

0150: ClaimStatus.REJECTED => "bg-danger", // statement

0151: \_ => "bg-light text-dark" // statement

0152: }; // statement

0153: } // statement

0154: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/ClaimsAll.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Claim> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "All Claims"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">All Claims</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="row g-2 mb-3"> // markup element

0015: <div class="col-md-3"> // markup element

0016: <select id="statusFilter" class="form-select"> // markup element

0017: <option value="">All Statuses</option> // markup element

0018: <option>OPEN</option> // markup element

0019: <option>APPROVED</option> // markup element

0020: <option>REJECTED</option> // markup element

0021: </select> // markup element

0022: </div> // markup element

0023: <div class="col-md-4"> // markup element

0024: <input id="q" class="form-control" placeholder="Search claim id, policy id, adjuster" /> // markup element

0025: </div> // markup element

0026: </div> // markup element

0027:

0028: <div class="table-responsive"> // markup element

0029: <table class="table table-hover align-middle" id="tbl"> // markup element

0030: <thead class="table-light"> // markup element

0031: <tr> // markup element

0032: <th>Claim Id</th> // markup element

0033: <th>Policy</th> // markup element

0034: <th>Amount</th> // markup element

0035: <th>Date</th> // markup element

0036: <th>Status</th> // markup element

0037: <th>Adjuster</th> // markup element

0038: </tr> // markup element

0039: </thead> // markup element

0040: <tbody> // markup element

0041: @foreach (var c in Model) // statement

0042: { // statement

0043: <tr data-status="@c.ClaimStatus"> // markup element

0044: <td>@c.ClaimId</td> // markup element

0045: <td>@c.PolicyId</td> // markup element

0046: <td>@String.Format("{0:N2}", c.ClaimAmount)</td> // markup element

0047: <td>@c.ClaimDate.ToString("dd-MMM-yyyy")</td> // markup element

0048: <td>@c.ClaimStatus</td> // markup element

0049: <td>@c.AdjusterId</td> // markup element

0050: </tr> // markup element

0051: } // statement

0052: </tbody> // markup element

0053: </table> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057: </div> // markup element

0058:

0059: @section Scripts{ // statement

0060: <script> // markup element

0061: const status = document.getElementById('statusFilter'); // statement

0062: const q = document.getElementById('q'); // statement

0063: const tbl = document.getElementById('tbl'); // statement

0064: function apply(){ // statement

0065: const s=status.value; const query=q.value.toLowerCase(); // statement

0066: Array.from(tbl.querySelectorAll('tbody tr')).forEach(r=>{ // statement

0067: const okS = !s || r.getAttribute('data-status')===s; // statement

0068: const okQ = r.innerText.toLowerCase().includes(query); // statement

0069: r.style.display = okS && okQ ? '' : 'none'; // statement

0070: }); // statement

0071: } // statement

0072: status.addEventListener('change', apply); // statement

0073: q.addEventListener('input', apply); // statement

0074: </script> // markup element

0075: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/ClaimsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Claims Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Claims</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AdminDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="ClaimsAll"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-list-task"></i></div> // markup element

0017: <div class="fs-5 fw-bold">Get All Claims</div> // markup element

0018: <div class="text-muted small">Browse all</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="ClaimDetails"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-warning text-dark mb-3"><i class="bi bi-search"></i></div> // markup element

0028: <div class="fs-5 fw-bold">Claim Details</div> // markup element

0029: <div class="text-muted small">Find a claim by id</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: @\* <div class="col-md-3"> // comment

0035: <a class="text-decoration-none" asp-action="ClaimSubmit"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-plus-circle"></i></div> // markup element

0039: <div class="fs-5 fw-bold">Submit Claim</div> // markup element

0040: <div class="text-muted small">Add a new claim</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> \*@ // statement

0045: <div class="col-md-3"> // markup element

0046: <a class="text-decoration-none" asp-action="ClaimUpdateStatus"> // markup element

0047: <div class="tile card shadow-sm border-0 h-100"> // markup element

0048: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0049: <div class="icon-circle bg-info text-white mb-3"><i class="bi bi-arrow-repeat"></i></div> // markup element

0050: <div class="fs-5 fw-bold">Update Status</div> // markup element

0051: <div class="text-muted small">Approve/Reject/Open</div> // markup element

0052: </div> // markup element

0053: </div> // markup element

0054: </a> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057: </div> // markup element

0058:

0059: <style> // markup element

0060: .claimslike-bg { // statement

0061: background: linear-gradient(180deg,#f8fbff,#eef5ff); // statement

0062: } // statement

0063:

0064: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0065: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0066: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0067: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/Console.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Admin Console"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <h2 class="mb-3">Admin Console</h2> // markup element

0007: @await Html.PartialAsync("~/Views/Shared/\_AdminStats.cshtml") // statement

0008:

0009: <div class="row g-3"> // markup element

0010: <div class="col-md-3"> // markup element

0011: <div class="card h-100 shadow-sm border-0" role="button" data-bs-toggle="modal" data-bs-target="#userModal"> // markup element

0012: <div class="card-body"> // markup element

0013: <div class="fs-5 fw-bold mb-1">User Management</div> // markup element

0014: <div class="text-muted small">Create, update, assign roles, delete</div> // markup element

0015: </div> // markup element

0016: </div> // markup element

0017: </div> // markup element

0018: <div class="col-md-3"> // markup element

0019: <div class="card h-100 shadow-sm border-0" role="button" data-bs-toggle="modal" data-bs-target="#policyModal"> // markup element

0020: <div class="card-body"> // markup element

0021: <div class="fs-5 fw-bold mb-1">Policy Oversight</div> // markup element

0022: <div class="text-muted small">Create, update, delete, fetch by id</div> // markup element

0023: </div> // markup element

0024: </div> // markup element

0025: </div> // markup element

0026: <div class="col-md-3"> // markup element

0027: <div class="card h-100 shadow-sm border-0" role="button" data-bs-toggle="modal" data-bs-target="#claimModal"> // markup element

0028: <div class="card-body"> // markup element

0029: <div class="fs-5 fw-bold mb-1">Claim Supervision</div> // markup element

0030: <div class="text-muted small">Submit, approve/reject, details</div> // markup element

0031: </div> // markup element

0032: </div> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <div class="card h-100 shadow-sm border-0" role="button" data-bs-toggle="modal" data-bs-target="#ticketModal"> // markup element

0036: <div class="card-body"> // markup element

0037: <div class="fs-5 fw-bold mb-1">Ticket Management</div> // markup element

0038: <div class="text-muted small">Resolve support tickets</div> // markup element

0039: </div> // markup element

0040: </div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043:

0044: <div class="mt-4"> // markup element

0045: <a class="btn btn-outline-secondary me-2" asp-action="Users">Open Users</a> // markup element

0046: <a class="btn btn-outline-success me-2" asp-action="Policies">Open Policies</a> // markup element

0047: <a class="btn btn-outline-warning me-2" asp-action="Claims">Open Claims</a> // markup element

0048: <a class="btn btn-outline-info me-2" asp-action="Payments">Open Payments</a> // markup element

0049: <a class="btn btn-outline-danger" asp-action="Tickets">Open Tickets</a> // markup element

0050: </div> // markup element

0051: </div> // markup element

0052:

0053: <!-- User Modal --> // markup element

0054: <div class="modal fade" id="userModal" tabindex="-1" aria-hidden="true"> // markup element

0055: <div class="modal-dialog modal-lg"> // markup element

0056: <div class="modal-content"> // markup element

0057: <div class="modal-header"> // markup element

0058: <h5 class="modal-title">User Management</h5> // markup element

0059: <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> // markup element

0060: </div> // markup element

0061: <div class="modal-body"> // markup element

0062: <form id="userForm" class="row g-2"> // markup element

0063: <input type="hidden" name="UserId" id="UserId" /> // markup element

0064: <div class="col-md-4"><input class="form-control" name="Username" placeholder="Username" required></div> // markup element

0065: <div class="col-md-4"><input class="form-control" type="email" name="Email" placeholder="Email" required></div> // markup element

0066: <div class="col-md-4"><input class="form-control" type="password" name="Password" placeholder="Password"></div> // markup element

0067: <div class="col-md-4"> // markup element

0068: <select class="form-select" name="Role" required> // markup element

0069: <option value="ADMIN">ADMIN</option> // markup element

0070: <option value="AGENT">AGENT</option> // markup element

0071: <option value="CUSTOMER">CUSTOMER</option> // markup element

0072: </select> // markup element

0073: </div> // markup element

0074: </form> // markup element

0075: <div class="small text-muted mt-2">Leave password empty to keep existing when updating.</div> // markup element

0076: </div> // markup element

0077: <div class="modal-footer"> // markup element

0078: <button type="button" class="btn btn-outline-danger" id="deleteUserBtn">Delete</button> // markup element

0079: <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button> // markup element

0080: <button type="button" class="btn btn-primary" id="saveUserBtn">Save</button> // markup element

0081: </div> // markup element

0082: </div> // markup element

0083: </div> // markup element

0084: </div> // markup element

0085:

0086: <!-- Policy Modal --> // markup element

0087: <div class="modal fade" id="policyModal" tabindex="-1" aria-hidden="true"> // markup element

0088: <div class="modal-dialog modal-lg"> // markup element

0089: <div class="modal-content"> // markup element

0090: <div class="modal-header"><h5 class="modal-title">Policy Oversight</h5><button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button></div> // markup element

0091: <div class="modal-body"> // markup element

0092: <form id="policyForm" class="row g-2"> // markup element

0093: <input type="hidden" name="PolicyId" id="PolicyId" /> // markup element

0094: <div class="col-md-4"><input class="form-control" name="PolicyNumber" placeholder="Policy Number" required></div> // markup element

0095: <div class="col-md-8"><input class="form-control" name="VehicleDetails" placeholder="Vehicle Details" required></div> // markup element

0096: <div class="col-md-4"><input class="form-control" type="number" step="0.01" name="CoverageAmount" placeholder="Coverage Amount" required></div> // markup element

0097: <div class="col-md-4"><input class="form-control" name="CoverageType" placeholder="Coverage Type" required></div> // markup element

0098: <div class="col-md-4"><input class="form-control" type="number" step="0.01" name="PremiumAmount" placeholder="Premium Amount" required></div> // markup element

0099: <div class="col-md-6"><input class="form-control" type="date" name="StartDate" required></div> // markup element

0100: <div class="col-md-6"><input class="form-control" type="date" name="EndDate" required></div> // markup element

0101: <div class="col-md-4"> // markup element

0102: <select class="form-select" name="PolicyStatus" required> // markup element

0103: <option value="ACTIVE">ACTIVE</option> // markup element

0104: <option value="INACTIVE">INACTIVE</option> // markup element

0105: <option value="RENEWED">RENEWED</option> // markup element

0106: </select> // markup element

0107: </div> // markup element

0108: </form> // markup element

0109: <div class="input-group mt-3"> // markup element

0110: <input id="policyLookupId" type="number" class="form-control" placeholder="Get Policy By Id"> // markup element

0111: <button id="policyLookupBtn" class="btn btn-outline-secondary">Fetch</button> // markup element

0112: </div> // markup element

0113: </div> // markup element

0114: <div class="modal-footer"> // markup element

0115: <button type="button" class="btn btn-outline-danger" id="deletePolicyBtn">Delete</button> // markup element

0116: <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button> // markup element

0117: <button type="button" class="btn btn-primary" id="savePolicyBtn">Save</button> // markup element

0118: </div> // markup element

0119: </div> // markup element

0120: </div> // markup element

0121: </div> // markup element

0122:

0123: <!-- Claim Modal --> // markup element

0124: <div class="modal fade" id="claimModal" tabindex="-1" aria-hidden="true"> // markup element

0125: <div class="modal-dialog"> // markup element

0126: <div class="modal-content"> // markup element

0127: <div class="modal-header"><h5 class="modal-title">Claim Supervision</h5><button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button></div> // markup element

0128: <div class="modal-body"> // markup element

0129: <form id="claimForm" class="row g-2"> // markup element

0130: <input type="hidden" name="ClaimId" id="ClaimId" /> // markup element

0131: <div class="col-12"><input class="form-control" type="number" name="PolicyId" placeholder="Policy Id" required></div> // markup element

0132: <div class="col-12"><input class="form-control" type="number" step="0.01" name="ClaimAmount" placeholder="Amount" required></div> // markup element

0133: <div class="col-12"><input class="form-control" type="date" name="ClaimDate" required></div> // markup element

0134: <div class="col-12"> // markup element

0135: <select class="form-select" name="ClaimStatus" required> // markup element

0136: <option value="OPEN">OPEN</option> // markup element

0137: <option value="APPROVED">APPROVED</option> // markup element

0138: <option value="REJECTED">REJECTED</option> // markup element

0139: </select> // markup element

0140: </div> // markup element

0141: <div class="col-12"><input class="form-control" type="number" name="AdjusterId" placeholder="Adjuster Id" required></div> // markup element

0142: </form> // markup element

0143: <div class="d-flex gap-2 mt-3"> // markup element

0144: <button id="approveClaimBtn" class="btn btn-success flex-fill">Approve</button> // markup element

0145: <button id="rejectClaimBtn" class="btn btn-danger flex-fill">Reject</button> // markup element

0146: </div> // markup element

0147: </div> // markup element

0148: <div class="modal-footer"> // markup element

0149: <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button> // markup element

0150: <button type="button" class="btn btn-primary" id="submitClaimBtn">Submit</button> // markup element

0151: </div> // markup element

0152: </div> // markup element

0153: </div> // markup element

0154: </div> // markup element

0155:

0156: <!-- Ticket Modal --> // markup element

0157: <div class="modal fade" id="ticketModal" tabindex="-1" aria-hidden="true"> // markup element

0158: <div class="modal-dialog"> // markup element

0159: <div class="modal-content"> // markup element

0160: <div class="modal-header"><h5 class="modal-title">Support Ticket Management</h5><button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button></div> // markup element

0161: <div class="modal-body"> // markup element

0162: <div class="input-group mb-2"> // markup element

0163: <input id="ticketId" class="form-control" type="number" placeholder="Ticket Id" /> // markup element

0164: </div> // markup element

0165: <button id="resolveTicketBtn" class="btn btn-success w-100">Resolve</button> // markup element

0166: </div> // markup element

0167: <div class="modal-footer"> // markup element

0168: <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button> // markup element

0169: </div> // markup element

0170: </div> // markup element

0171: </div> // markup element

0172: </div> // markup element

0173:

0174: @section Scripts{ // statement

0175: <script> // markup element

0176: // User ops // comment

0177: const saveUserBtn=document.getElementById('saveUserBtn'); // statement

0178: saveUserBtn.addEventListener('click', async ()=>{ // statement

0179: const form=new FormData(document.getElementById('userForm')); // statement

0180: const userId=form.get('UserId'); // statement

0181: const url=userId?'/Admin/AdminUpdateUser':'/Admin/AdminCreateUser'; // statement

0182: const res=await fetch(url,{method:'POST',body:form}); // statement

0183: alert(res.ok?'Saved':'Failed'); // statement

0184: }); // statement

0185:

0186: document.getElementById('deleteUserBtn').addEventListener('click', async ()=>{ // statement

0187: const id=document.getElementById('UserId').value; // statement

0188: if(!id) return alert('Enter UserId'); // conditional check

0189: const form=new FormData(); form.append('userId', id); // statement

0190: const res=await fetch('/Admin/AdminDeleteUser',{method:'POST', body:form}); // statement

0191: alert(res.ok?'Deleted':'Failed'); // statement

0192: }); // statement

0193:

0194: // Policy ops // comment

0195: const savePolicyBtn=document.getElementById('savePolicyBtn'); // statement

0196: savePolicyBtn.addEventListener('click', async ()=>{ // statement

0197: const form=new FormData(document.getElementById('policyForm')); // statement

0198: const id=form.get('PolicyId'); // statement

0199: const url=id?'/Admin/AdminUpdatePolicy':'/Admin/AdminCreatePolicy'; // statement

0200: const res=await fetch(url,{method:'POST',body:form}); // statement

0201: alert(res.ok?'Saved':'Failed'); // statement

0202: }); // statement

0203:

0204: document.getElementById('deletePolicyBtn').addEventListener('click', async ()=>{ // statement

0205: const id=document.getElementById('PolicyId').value; if(!id) return alert('Enter PolicyId'); // statement

0206: const form=new FormData(); form.append('policyId', id); // statement

0207: const res=await fetch('/Admin/AdminDeletePolicy',{method:'POST', body:form}); // statement

0208: alert(res.ok?'Deleted':'Failed'); // statement

0209: }); // statement

0210:

0211: document.getElementById('policyLookupBtn').addEventListener('click', async ()=>{ // statement

0212: const id=document.getElementById('policyLookupId').value; if(!id) return; // statement

0213: const res=await fetch('/Admin/AdminGetPolicy?id='+encodeURIComponent(id)); // statement

0214: if(!res.ok){ alert('Not found'); return; } // conditional check

0215: const p=await res.json(); // statement

0216: const f=document.getElementById('policyForm'); // statement

0217: f.PolicyId.value=p.policyId||p.PolicyId; f.PolicyNumber.value=p.policyNumber||p.PolicyNumber; f.VehicleDetails.value=p.vehicleDetails||p.VehicleDetails; // statement

0218: f.CoverageAmount.value=p.coverageAmount||p.CoverageAmount; f.CoverageType.value=p.coverageType||p.CoverageType; f.PremiumAmount.value=p.premiumAmount||p.PremiumAmount; // statement

0219: f.StartDate.value=(p.startDate||p.StartDate).substring(0,10); f.EndDate.value=(p.endDate||p.EndDate).substring(0,10); f.PolicyStatus.value=p.policyStatus||p.PolicyStatus; // statement

0220: }); // statement

0221:

0222: // Claim ops // comment

0223: const submitClaimBtn=document.getElementById('submitClaimBtn'); // statement

0224: submitClaimBtn.addEventListener('click', async ()=>{ // statement

0225: const form=new FormData(document.getElementById('claimForm')); // statement

0226: const res=await fetch('/Admin/AdminSubmitClaim',{method:'POST',body:form}); // statement

0227: alert(res.ok?'Submitted':'Failed'); // statement

0228: }); // statement

0229:

0230: document.getElementById('approveClaimBtn').addEventListener('click', ()=>updateClaimStatus('APPROVED')); // statement

0231: document.getElementById('rejectClaimBtn').addEventListener('click', ()=>updateClaimStatus('REJECTED')); // statement

0232: async function updateClaimStatus(status){ // statement

0233: const id=document.getElementById('ClaimId').value; const adjuster=document.querySelector('#claimForm [name="AdjusterId"]').value; // statement

0234: if(!id) return alert('Enter ClaimId (if updating existing).'); // conditional check

0235: const form=new FormData(); form.append('claimId', id); form.append('status', status); form.append('adjusterId', adjuster); // statement

0236: const res=await fetch('/Admin/AdminUpdateClaimStatus',{method:'POST', body:form}); // statement

0237: alert(res.ok?'Updated':'Failed'); // statement

0238: } // statement

0239:

0240: // Ticket ops // comment

0241:

0242: document.getElementById('resolveTicketBtn').addEventListener('click', async ()=>{ // statement

0243: const ticketId=document.getElementById('ticketId').value; if(!ticketId) return alert('Enter ticket id'); // statement

0244: const form=new FormData(); form.append('ticketId', ticketId); // statement

0245: const res=await fetch('/Admin/AdminResolveTicket',{method:'POST', body:form}); // statement

0246: alert(res.ok?'Resolved':'Failed'); // statement

0247: }); // statement

0248: </script> // markup element

0249: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PaymentMake.cshtml

0001: @model Auto\_Insurance\_System.Models.Payment // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Make Payment"; // statement

0004: } // statement

0005:

0006: <div class="container py-4 pay-bg"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Make Payment</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div asp-validation-summary="All" class="text-danger"></div> // markup element

0025: <div class="col-12"> // markup element

0026: <label class="form-label">Policy Id</label> // markup element

0027: <input class="form-control" asp-for="PolicyId" type="number" required /> // markup element

0028: <span class="text-danger" asp-validation-for="PolicyId"></span> // markup element

0029: </div> // markup element

0030: <div class="col-12"> // markup element

0031: <label class="form-label">Amount</label> // markup element

0032: <input class="form-control" asp-for="PaymentAmount" type="number" step="0.01" required /> // markup element

0033: <span class="text-danger" asp-validation-for="PaymentAmount"></span> // markup element

0034: </div> // markup element

0035: <div class="col-12"> // markup element

0036: <label class="form-label">Date</label> // markup element

0037: <input class="form-control" asp-for="PaymentDate" type="date" required /> // markup element

0038: <span class="text-danger" asp-validation-for="PaymentDate"></span> // markup element

0039: </div> // markup element

0040: <div class="col-12"> // markup element

0041: <label class="form-label">Status</label> // markup element

0042: <select class="form-select" asp-for="PaymentStatus"> // markup element

0043: <option>SUCCESS</option> // markup element

0044: <option>PENDING</option> // markup element

0045: <option>FAILED</option> // markup element

0046: </select> // markup element

0047: <span class="text-danger" asp-validation-for="PaymentStatus"></span> // markup element

0048: </div> // markup element

0049: <div class="col-12 d-flex gap-2"> // markup element

0050: <button class="btn btn-primary">Save</button> // markup element

0051: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Cancel</a> // markup element

0052: </div> // markup element

0053: </form> // markup element

0054: <partial name="\_ValidationScriptsPartial" /> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057: </div> // markup element

0058:

0059: <style> // markup element

0060: .pay-bg{background:#ffffff;} // statement

0061: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PaymentUpdateStatus.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Update Payment Status"; // statement

0003: } // statement

0004:

0005: <div class="container py-4 pay-bg"> // markup element

0006: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0007: <h2 class="mb-0">Update Payment Status</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0009: </div> // markup element

0010:

0011: @if (ViewBag.Success != null) // Razor conditional rendering

0012: { // statement

0013: <div class="alert alert-success">@ViewBag.Success</div> // markup element

0014: } // statement

0015: @if (ViewBag.Error != null) // Razor conditional rendering

0016: { // statement

0017: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0018: } // statement

0019:

0020: <div class="card shadow-sm"> // markup element

0021: <div class="card-body"> // markup element

0022: <form method="post" class="row g-3"> // markup element

0023: <div class="col-md-4"> // markup element

0024: <label class="form-label">Payment Id</label> // markup element

0025: <input class="form-control" type="number" name="paymentId" required /> // markup element

0026: </div> // markup element

0027: <div class="col-md-4"> // markup element

0028: <label class="form-label">Status</label> // markup element

0029: <select class="form-select" name="status"> // markup element

0030: <option value="SUCCESS">SUCCESS</option> // markup element

0031: <option value="PENDING">PENDING</option> // markup element

0032: <option value="FAILED">FAILED</option> // markup element

0033: </select> // markup element

0034: </div> // markup element

0035: <div class="col-12 d-flex gap-2"> // markup element

0036: <button class="btn btn-primary">Update</button> // markup element

0037: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Cancel</a> // markup element

0038: </div> // markup element

0039: </form> // markup element

0040: </div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043:

0044: <style> // markup element

0045: .pay-bg{background:#ffffff;} // statement

0046: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/Payments.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Payment> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Manage Payments"; // statement

0004: } // statement

0005:

0006: <div class="container-fluid"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Payments</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="row g-2 mb-3"> // markup element

0015: <div class="col-md-3"> // markup element

0016: <select id="paymentStatusFilter" class="form-select"> // markup element

0017: <option value="">All Statuses</option> // markup element

0018: <option value="SUCCESS">Success</option> // markup element

0019: <option value="PENDING">Pending</option> // markup element

0020: <option value="FAILED">Failed</option> // markup element

0021: </select> // markup element

0022: </div> // markup element

0023: <div class="col-md-4"> // markup element

0024: <input id="paymentSearch" class="form-control" placeholder="Search by payment id, policy number or amount" /> // markup element

0025: </div> // markup element

0026: </div> // markup element

0027:

0028: <div class="table-responsive"> // markup element

0029: <table class="table table-striped align-middle" id="paymentsTable"> // markup element

0030: <thead class="table-light"> // markup element

0031: <tr> // markup element

0032: <th>Payment Id</th> // markup element

0033: <th>Policy Number</th> // markup element

0034: <th>Amount</th> // markup element

0035: <th>Date</th> // markup element

0036: <th>Status</th> // markup element

0037: </tr> // markup element

0038: </thead> // markup element

0039: <tbody> // markup element

0040: @foreach (var p in Model) // statement

0041: { // statement

0042: <tr data-status="@p.PaymentStatus"> // markup element

0043: <td>@p.PaymentId</td> // markup element

0044: <td> // markup element

0045: @if (p.Policy != null) // Razor conditional rendering

0046: { // statement

0047: <div>@p.Policy.PolicyNumber</div> // markup element

0048: } // statement

0049: else // statement

0050: { // statement

0051: <div class="text-muted">(Policy Id: @p.PolicyId)</div> // markup element

0052: } // statement

0053: </td> // markup element

0054: <td>@String.Format("{0:N2}", p.PaymentAmount)</td> // markup element

0055: <td>@p.PaymentDate.ToString("dd-MMM-yyyy")</td> // markup element

0056: <td><span class="badge @GetPaymentBadge(p.PaymentStatus)">@p.PaymentStatus</span></td> // markup element

0057: </tr> // markup element

0058: } // statement

0059: </tbody> // markup element

0060: </table> // markup element

0061: </div> // markup element

0062: </div> // markup element

0063: </div> // markup element

0064: </div> // markup element

0065:

0066: @functions{ // statement

0067: string GetPaymentBadge(PaymentStatus status) // statement

0068: { // statement

0069: return status switch // return value

0070: { // statement

0071: PaymentStatus.SUCCESS => "bg-success", // statement

0072: PaymentStatus.PENDING => "bg-warning text-dark", // statement

0073: PaymentStatus.FAILED => "bg-danger", // statement

0074: \_ => "bg-light text-dark" // statement

0075: }; // statement

0076: } // statement

0077: } // statement

0078:

0079: @section Scripts{ // statement

0080: <script> // markup element

0081: const statusFilter = document.getElementById('paymentStatusFilter'); // statement

0082: const search = document.getElementById('paymentSearch'); // statement

0083: const table = document.getElementById('paymentsTable'); // statement

0084:

0085: function applyFilters() { // statement

0086: const status = statusFilter.value; // statement

0087: const q = search.value.toLowerCase(); // statement

0088: Array.from(table.querySelectorAll('tbody tr')).forEach(r => { // statement

0089: const matchStatus = !status || r.getAttribute('data-status') === status; // statement

0090: const text = r.innerText.toLowerCase(); // statement

0091: const matchText = text.includes(q); // statement

0092: r.style.display = matchStatus && matchText ? '' : 'none'; // statement

0093: }); // statement

0094: } // statement

0095: statusFilter.addEventListener('change', applyFilters); // statement

0096: search.addEventListener('input', applyFilters); // statement

0097: </script> // markup element

0098: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PaymentsByPolicy.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Payment> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Payments By Policy"; // statement

0004: } // statement

0005:

0006: <div class="container py-4 pay-bg"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Get Payments By Policy</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm mb-3"> // markup element

0013: <div class="card-body"> // markup element

0014: <form method="post" class="row g-2"> // markup element

0015: <div class="col-md-4"><input class="form-control" type="number" name="policyId" placeholder="Enter Policy Id" required /></div> // markup element

0016: <div class="col-md-2"><button class="btn btn-primary w-100" type="submit">Search</button></div> // markup element

0017: </form> // markup element

0018: @if (ViewBag.Error != null) // Razor conditional rendering

0019: { // statement

0020: <div class="alert alert-warning mt-3">@ViewBag.Error</div> // markup element

0021: } // statement

0022: </div> // markup element

0023: </div> // markup element

0024:

0025: @if (Model != null) // Razor conditional rendering

0026: { // statement

0027: <div class="card shadow-sm"> // markup element

0028: <div class="card-body"> // markup element

0029: <div class="table-responsive"> // markup element

0030: <table class="table table-hover align-middle"> // markup element

0031: <thead class="table-light"> // markup element

0032: <tr> // markup element

0033: <th>Payment Id</th> // markup element

0034: <th>Policy Number</th> // markup element

0035: <th>Amount</th> // markup element

0036: <th>Date</th> // markup element

0037: <th>Status</th> // markup element

0038: </tr> // markup element

0039: </thead> // markup element

0040: <tbody> // markup element

0041: @foreach (var p in Model) // statement

0042: { // statement

0043: <tr> // markup element

0044: <td>@p.PaymentId</td> // markup element

0045: <td>@(p?.Policy?.PolicyNumber ?? $"(Policy Id: {p.PolicyId})")</td> // markup element

0046: <td>@String.Format("{0:N2}", p.PaymentAmount)</td> // markup element

0047: <td>@p.PaymentDate.ToString("dd-MMM-yyyy")</td> // markup element

0048: <td>@p.PaymentStatus</td> // markup element

0049: </tr> // markup element

0050: } // statement

0051: </tbody> // markup element

0052: </table> // markup element

0053: </div> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056: } // statement

0057: </div> // markup element

0058:

0059: <style> // markup element

0060: .pay-bg{background:#ffffff;} // statement

0061: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PaymentsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Payments Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Payments</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AdminDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="PaymentsAll"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3">📋</div> // markup element

0017: <div class="fs-5 fw-bold">Get All Payments</div> // markup element

0018: <div class="text-muted small">Full transaction list</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="PaymentsByPolicy"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-warning text-dark mb-3">🧾</div> // markup element

0028: <div class="fs-5 fw-bold">Get Payments By Policy</div> // markup element

0029: <div class="text-muted small">Filter by policy</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <a class="text-decoration-none" asp-action="PaymentUpdateStatus"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-info text-white mb-3">🔄</div> // markup element

0039: <div class="fs-5 fw-bold">Update Payment Status</div> // markup element

0040: <div class="text-muted small">Success/Failed/Pending</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </div> // markup element

0047:

0048: <style> // markup element

0049: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0050: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0051: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0052: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/Policies.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Policy> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Manage Policies"; // statement

0004: } // statement

0005:

0006: <div class="container-fluid"> // markup element

0007: <h2 class="mb-3">Policies</h2> // markup element

0008:

0009: <div class="card shadow-sm"> // markup element

0010: <div class="card-body"> // markup element

0011: <div class="row g-2 mb-3"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <select id="statusFilter" class="form-select"> // markup element

0014: <option value="">All Statuses</option> // markup element

0015: <option value="ACTIVE">Active</option> // markup element

0016: <option value="INACTIVE">Inactive</option> // markup element

0017: <option value="RENEWED">Renewed</option> // markup element

0018: </select> // markup element

0019: </div> // markup element

0020: <div class="col-md-4"> // markup element

0021: <input id="policySearch" class="form-control" placeholder="Search by number, vehicle or coverage" /> // markup element

0022: </div> // markup element

0023: </div> // markup element

0024:

0025: <div class="table-responsive"> // markup element

0026: <table class="table table-striped align-middle" id="policiesTable"> // markup element

0027: <thead class="table-light"> // markup element

0028: <tr> // markup element

0029: <th>Policy Id</th> // markup element

0030: <th>Number</th> // markup element

0031: <th>Vehicle</th> // markup element

0032: <th>Coverage</th> // markup element

0033: <th>Premium</th> // markup element

0034: <th>Start</th> // markup element

0035: <th>End</th> // markup element

0036: <th>Status</th> // markup element

0037: </tr> // markup element

0038: </thead> // markup element

0039: <tbody> // markup element

0040: @foreach (var p in Model) // statement

0041: { // statement

0042: <tr data-status="@p.PolicyStatus"> // markup element

0043: <td>@p.PolicyId</td> // markup element

0044: <td>@p.PolicyNumber</td> // markup element

0045: <td>@p.VehicleDetails</td> // markup element

0046: <td>@String.Format("{0:N2}", p.CoverageAmount) (@p.CoverageType)</td> // markup element

0047: <td>@String.Format("{0:N2}", p.PremiumAmount)</td> // markup element

0048: <td>@p.StartDate.ToString("dd-MMM-yyyy")</td> // markup element

0049: <td>@p.EndDate.ToString("dd-MMM-yyyy")</td> // markup element

0050: <td> // markup element

0051: <span class="badge @GetStatusBadge(p.PolicyStatus)">@p.PolicyStatus</span> // markup element

0052: </td> // markup element

0053: </tr> // markup element

0054: } // statement

0055: </tbody> // markup element

0056: </table> // markup element

0057: </div> // markup element

0058: </div> // markup element

0059: </div> // markup element

0060: </div> // markup element

0061:

0062: @functions{ // statement

0063: string GetStatusBadge(PolicyStatus status) // statement

0064: { // statement

0065: return status switch // return value

0066: { // statement

0067: PolicyStatus.ACTIVE => "bg-success", // statement

0068: PolicyStatus.RENEWED => "bg-info", // statement

0069: PolicyStatus.INACTIVE => "bg-secondary", // statement

0070: \_ => "bg-light text-dark" // statement

0071: }; // statement

0072: } // statement

0073: } // statement

0074:

0075: @section Scripts{ // statement

0076: <script> // markup element

0077: const statusFilter = document.getElementById('statusFilter'); // statement

0078: const search = document.getElementById('policySearch'); // statement

0079: const table = document.getElementById('policiesTable'); // statement

0080:

0081: function applyFilters() { // statement

0082: const status = statusFilter.value; // statement

0083: const q = search.value.toLowerCase(); // statement

0084: Array.from(table.querySelectorAll('tbody tr')).forEach(r => { // statement

0085: const matchStatus = !status || r.getAttribute('data-status') === status; // statement

0086: const text = r.innerText.toLowerCase(); // statement

0087: const matchText = text.includes(q); // statement

0088: r.style.display = matchStatus && matchText ? '' : 'none'; // statement

0089: }); // statement

0090: } // statement

0091: statusFilter.addEventListener('change', applyFilters); // statement

0092: search.addEventListener('input', applyFilters); // statement

0093: </script> // markup element

0094: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PoliciesAll.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Policy> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "All Policies"; // statement

0004: } // statement

0005:

0006: <div class="container py-4 policies-bg"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">All Policies</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="row g-2 mb-3"> // markup element

0015: <div class="col-md-3"> // markup element

0016: <select id="status" class="form-select"> // markup element

0017: <option value="">All Statuses</option> // markup element

0018: <option value="ACTIVE">ACTIVE</option> // markup element

0019: <option value="INACTIVE">INACTIVE</option> // markup element

0020: <option value="RENEWED">RENEWED</option> // markup element

0021: </select> // markup element

0022: </div> // markup element

0023: <div class="col-md-4"><input id="q" class="form-control" placeholder="Search policy number, vehicle or coverage" /></div> // markup element

0024: </div> // markup element

0025:

0026: <div class="table-responsive"> // markup element

0027: <table id="tbl" class="table table-hover align-middle"> // markup element

0028: <thead class="table-light"> // markup element

0029: <tr> // markup element

0030: <th>Policy Id</th> // markup element

0031: <th>Number</th> // markup element

0032: <th>Vehicle</th> // markup element

0033: <th>Coverage</th> // markup element

0034: <th>Premium</th> // markup element

0035: <th>Start</th> // markup element

0036: <th>End</th> // markup element

0037: <th>Status</th> // markup element

0038: </tr> // markup element

0039: </thead> // markup element

0040: <tbody> // markup element

0041: @foreach (var p in Model) // statement

0042: { // statement

0043: <tr data-status="@p.PolicyStatus"> // markup element

0044: <td>@p.PolicyId</td> // markup element

0045: <td>@p.PolicyNumber</td> // markup element

0046: <td>@p.VehicleDetails</td> // markup element

0047: <td>@p.CoverageType (@String.Format("{0:N2}", p.CoverageAmount))</td> // markup element

0048: <td>@String.Format("{0:N2}", p.PremiumAmount)</td> // markup element

0049: <td>@p.StartDate.ToString("dd-MMM-yyyy")</td> // markup element

0050: <td>@p.EndDate.ToString("dd-MMM-yyyy")</td> // markup element

0051: <td>@p.PolicyStatus</td> // markup element

0052: </tr> // markup element

0053: } // statement

0054: </tbody> // markup element

0055: </table> // markup element

0056: </div> // markup element

0057: </div> // markup element

0058: </div> // markup element

0059: </div> // markup element

0060:

0061: @section Scripts{ // statement

0062: <script> // markup element

0063: const status=document.getElementById('status'); // statement

0064: const q=document.getElementById('q'); // statement

0065: const tbl=document.getElementById('tbl'); // statement

0066: function apply(){ // statement

0067: const s=status.value; const query=q.value.toLowerCase(); // statement

0068: Array.from(tbl.querySelectorAll('tbody tr')).forEach(r=>{ // statement

0069: const okS=!s || r.getAttribute('data-status')===s; // statement

0070: const okQ=r.innerText.toLowerCase().includes(query); // statement

0071: r.style.display = okS && okQ ? '' : 'none'; // statement

0072: }); // statement

0073: } // statement

0074: status.addEventListener('change', apply); // statement

0075: q.addEventListener('input', apply); // statement

0076: </script> // markup element

0077: } // statement

0078:

0079: <style> // markup element

0080: .policies-bg{background:#ffffff;} // statement

0081: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PoliciesHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Policies Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4 claimslike-bg"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Policies</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AdminDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="PoliciesAll"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-list-check"></i></div> // markup element

0017: <div class="fs-5 fw-bold">Get All Policies</div> // markup element

0018: <div class="text-muted small">Browse the full list</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="PolicyDetails"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-warning text-dark mb-3"><i class="bi bi-search"></i></div> // markup element

0028: <div class="fs-5 fw-bold">Get Policy By ID</div> // markup element

0029: <div class="text-muted small">Find a specific policy</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <a class="text-decoration-none" asp-action="PolicyUpdate"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-info text-white mb-3"><i class="bi bi-pencil-square"></i></div> // markup element

0039: <div class="fs-5 fw-bold">Update Policy</div> // markup element

0040: <div class="text-muted small">Modify an existing policy</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> // markup element

0045: <div class="col-md-3"> // markup element

0046: <a class="text-decoration-none" asp-action="PolicyDelete"> // markup element

0047: <div class="tile card shadow-sm border-0 h-100"> // markup element

0048: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0049: <div class="icon-circle bg-danger text-white mb-3"><i class="bi bi-trash"></i></div> // markup element

0050: <div class="fs-5 fw-bold">Delete Policy</div> // markup element

0051: <div class="text-muted small">Remove by id</div> // markup element

0052: </div> // markup element

0053: </div> // markup element

0054: </a> // markup element

0055: </div> // markup element

0056: @\* <div class="col-md-3"> // comment

0057: <a class="text-decoration-none" asp-action="PolicyCreate"> // markup element

0058: <div class="tile card shadow-sm border-0 h-100"> // markup element

0059: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0060: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-plus-circle"></i></div> // markup element

0061: <div class="fs-5 fw-bold">Create Policy</div> // markup element

0062: <div class="text-muted small">Add a new policy</div> // markup element

0063: </div> // markup element

0064: </div> // markup element

0065: </a> // markup element

0066: </div> \*@ // statement

0067: </div> // markup element

0068: </div> // markup element

0069:

0070: <style> // markup element

0071: .claimslike-bg{background:#ffffff;} // statement

0072: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0073: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0074: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0075: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PolicyCreate.cshtml

0001: @model Auto\_Insurance\_System.Models.Policy // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Create Policy"; // statement

0004: } // statement

0005:

0006: <div class="container py-4 policies-bg"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Create Policy</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-12"> // markup element

0025: <label class="form-label">Policy Number</label> // markup element

0026: <input class="form-control" asp-for="PolicyNumber" required /> // markup element

0027: <span class="text-danger" asp-validation-for="PolicyNumber"></span> // markup element

0028: </div> // markup element

0029: <div class="col-12"> // markup element

0030: <label class="form-label">Vehicle Details</label> // markup element

0031: <input class="form-control" asp-for="VehicleDetails" required /> // markup element

0032: <span class="text-danger" asp-validation-for="VehicleDetails"></span> // markup element

0033: </div> // markup element

0034: <div class="col-12"> // markup element

0035: <label class="form-label">Coverage Amount</label> // markup element

0036: <input class="form-control" type="number" step="0.01" asp-for="CoverageAmount" required /> // markup element

0037: <span class="text-danger" asp-validation-for="CoverageAmount"></span> // markup element

0038: </div> // markup element

0039: <div class="col-12"> // markup element

0040: <label class="form-label">Coverage Type</label> // markup element

0041: <input class="form-control" asp-for="CoverageType" required /> // markup element

0042: <span class="text-danger" asp-validation-for="CoverageType"></span> // markup element

0043: </div> // markup element

0044: <div class="col-12"> // markup element

0045: <label class="form-label">Premium Amount</label> // markup element

0046: <input class="form-control" type="number" step="0.01" asp-for="PremiumAmount" required /> // markup element

0047: <span class="text-danger" asp-validation-for="PremiumAmount"></span> // markup element

0048: </div> // markup element

0049: <div class="col-md-6"> // markup element

0050: <label class="form-label">Start Date</label> // markup element

0051: <input class="form-control" type="date" asp-for="StartDate" required /> // markup element

0052: <span class="text-danger" asp-validation-for="StartDate"></span> // markup element

0053: </div> // markup element

0054: <div class="col-md-6"> // markup element

0055: <label class="form-label">End Date</label> // markup element

0056: <input class="form-control" type="date" asp-for="EndDate" required /> // markup element

0057: <span class="text-danger" asp-validation-for="EndDate"></span> // markup element

0058: </div> // markup element

0059: <div class="col-12"> // markup element

0060: <label class="form-label">Status</label> // markup element

0061: <select class="form-select" asp-for="PolicyStatus"> // markup element

0062: <option>ACTIVE</option> // markup element

0063: <option>INACTIVE</option> // markup element

0064: <option>RENEWED</option> // markup element

0065: </select> // markup element

0066: <span class="text-danger" asp-validation-for="PolicyStatus"></span> // markup element

0067: </div> // markup element

0068: <div class="col-12 d-flex gap-2"> // markup element

0069: <button class="btn btn-primary">Create</button> // markup element

0070: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Cancel</a> // markup element

0071: </div> // markup element

0072: </form> // markup element

0073: </div> // markup element

0074: </div> // markup element

0075: </div> // markup element

0076:

0077: @section Scripts{ // statement

0078: <partial name="\_ValidationScriptsPartial" /> // markup element

0079: } // statement

0080:

0081: <style> // markup element

0082: .policies-bg{background:#ffffff;} // statement

0083: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PolicyDelete.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Delete Policy"; // statement

0003: } // statement

0004:

0005: <div class="container py-4 policies-bg"> // markup element

0006: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0007: <h2 class="mb-0 text-danger">Delete Policy</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0009: </div> // markup element

0010:

0011: @if (TempData["Success"] != null) // Razor conditional rendering

0012: { // statement

0013: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0014: } // statement

0015: @if (ViewBag.Error != null) // Razor conditional rendering

0016: { // statement

0017: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0018: } // statement

0019:

0020: <div class="card shadow-sm"> // markup element

0021: <div class="card-body"> // markup element

0022: <form method="post" class="row g-3"> // markup element

0023: <div class="col-md-4"> // markup element

0024: <label class="form-label">Policy Id</label> // markup element

0025: <input class="form-control" type="number" name="policyId" placeholder="Enter policy id to delete" required /> // markup element

0026: </div> // markup element

0027: <div class="col-12"> // markup element

0028: <button class="btn btn-danger">Delete</button> // markup element

0029: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Cancel</a> // markup element

0030: </div> // markup element

0031: </form> // markup element

0032: </div> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035:

0036: <style> // markup element

0037: .policies-bg{background:#ffffff;} // statement

0038: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PolicyDetails.cshtml

0001: @model Auto\_Insurance\_System.Models.Policy // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Policy Details"; // statement

0004: } // statement

0005:

0006: <div class="container py-4 policies-bg"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Get Policy By ID</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm mb-3"> // markup element

0013: <div class="card-body"> // markup element

0014: <form method="post" class="row g-2"> // markup element

0015: <div class="col-md-4"><input class="form-control" type="number" name="policyId" placeholder="Enter Policy Id" required /></div> // markup element

0016: <div class="col-md-2"><button class="btn btn-primary w-100" type="submit">Search</button></div> // markup element

0017: </form> // markup element

0018: @if (ViewBag.Error != null) // Razor conditional rendering

0019: { // statement

0020: <div class="alert alert-warning mt-3">@ViewBag.Error</div> // markup element

0021: } // statement

0022: </div> // markup element

0023: </div> // markup element

0024:

0025: @if (Model != null) // Razor conditional rendering

0026: { // statement

0027: <div class="card shadow-sm"> // markup element

0028: <div class="card-header bg-light">Policy #@Model.PolicyId</div> // markup element

0029: <div class="card-body"> // markup element

0030: <div class="row g-3"> // markup element

0031: <div class="col-md-4"><strong>Number:</strong> @Model.PolicyNumber</div> // markup element

0032: <div class="col-md-8"><strong>Vehicle:</strong> @Model.VehicleDetails</div> // markup element

0033: <div class="col-md-4"><strong>Coverage:</strong> @Model.CoverageType (@String.Format("{0:N2}", @Model.CoverageAmount))</div> // markup element

0034: <div class="col-md-4"><strong>Premium:</strong> @String.Format("{0:N2}", @Model.PremiumAmount)</div> // markup element

0035: <div class="col-md-2"><strong>Start:</strong> @Model.StartDate.ToString("dd-MMM-yyyy")</div> // markup element

0036: <div class="col-md-2"><strong>End:</strong> @Model.EndDate.ToString("dd-MMM-yyyy")</div> // markup element

0037: <div class="col-md-3"><strong>Status:</strong> @Model.PolicyStatus</div> // markup element

0038: </div> // markup element

0039: </div> // markup element

0040: </div> // markup element

0041: } // statement

0042: </div> // markup element

0043:

0044: <style> // markup element

0045: .policies-bg{background:#ffffff;} // statement

0046: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/PolicyUpdate.cshtml

0001: @model Auto\_Insurance\_System.Models.Policy // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Update Policy"; // statement

0004: } // statement

0005:

0006: <div class="container py-4 policies-bg"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Update Policy</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-12"> // markup element

0025: <label class="form-label">Policy Id</label> // markup element

0026: <input class="form-control" asp-for="PolicyId" type="number" placeholder="Existing policy id" required /> // markup element

0027: </div> // markup element

0028: <div class="col-12"> // markup element

0029: <label class="form-label">Policy Number</label> // markup element

0030: <input class="form-control" asp-for="PolicyNumber" required /> // markup element

0031: </div> // markup element

0032: <div class="col-12"> // markup element

0033: <label class="form-label">Vehicle Details</label> // markup element

0034: <input class="form-control" asp-for="VehicleDetails" required /> // markup element

0035: </div> // markup element

0036: <div class="col-12"> // markup element

0037: <label class="form-label">Coverage Amount</label> // markup element

0038: <input class="form-control" type="number" step="0.01" asp-for="CoverageAmount" required /> // markup element

0039: </div> // markup element

0040: <div class="col-12"> // markup element

0041: <label class="form-label">Coverage Type</label> // markup element

0042: <input class="form-control" asp-for="CoverageType" required /> // markup element

0043: </div> // markup element

0044: <div class="col-12"> // markup element

0045: <label class="form-label">Premium Amount</label> // markup element

0046: <input class="form-control" type="number" step="0.01" asp-for="PremiumAmount" required /> // markup element

0047: </div> // markup element

0048: <div class="col-md-6"> // markup element

0049: <label class="form-label">Start Date</label> // markup element

0050: <input class="form-control" type="date" asp-for="StartDate" required /> // markup element

0051: </div> // markup element

0052: <div class="col-md-6"> // markup element

0053: <label class="form-label">End Date</label> // markup element

0054: <input class="form-control" type="date" asp-for="EndDate" required /> // markup element

0055: </div> // markup element

0056: <div class="col-12"> // markup element

0057: <label class="form-label">Status</label> // markup element

0058: <select class="form-select" asp-for="PolicyStatus"> // markup element

0059: <option>ACTIVE</option> // markup element

0060: <option>INACTIVE</option> // markup element

0061: <option>RENEWED</option> // markup element

0062: </select> // markup element

0063: </div> // markup element

0064: <div class="col-12 d-flex gap-2"> // markup element

0065: <button class="btn btn-primary">Update</button> // markup element

0066: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Cancel</a> // markup element

0067: </div> // markup element

0068: </form> // markup element

0069: </div> // markup element

0070: </div> // markup element

0071: </div> // markup element

0072:

0073: <style> // markup element

0074: .policies-bg{background:#ffffff;} // statement

0075: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/TicketAssign.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Assign Ticket"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0007: <h2 class="mb-0">Assign Ticket</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="card shadow-sm"> // markup element

0012: <div class="card-body"> // markup element

0013: <form method="post" class="row g-3"> // markup element

0014: <div class="col-md-4"> // markup element

0015: <label class="form-label">Ticket Id</label> // markup element

0016: <input class="form-control" type="number" name="ticketId" required /> // markup element

0017: </div> // markup element

0018: <div class="col-md-4"> // markup element

0019: <label class="form-label">Agent User Id</label> // markup element

0020: <input class="form-control" type="number" name="agentId" required /> // markup element

0021: </div> // markup element

0022: <div class="col-12 d-flex gap-2"> // markup element

0023: <button class="btn btn-primary">Assign</button> // markup element

0024: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Cancel</a> // markup element

0025: </div> // markup element

0026: </form> // markup element

0027: @if (ViewBag.Success != null) // Razor conditional rendering

0028: { // statement

0029: <div class="alert alert-success mt-3">@ViewBag.Success</div> // markup element

0030: } // statement

0031: @if (ViewBag.Error != null) // Razor conditional rendering

0032: { // statement

0033: <div class="alert alert-warning mt-3">@ViewBag.Error</div> // markup element

0034: } // statement

0035: </div> // markup element

0036: </div> // markup element

0037: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/TicketResolve.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Resolve Ticket"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0007: <h2 class="mb-0">Resolve Ticket</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="card shadow-sm"> // markup element

0012: <div class="card-body"> // markup element

0013: <form method="post" class="row g-3"> // markup element

0014: <div class="col-md-4"> // markup element

0015: <label class="form-label">Ticket Id</label> // markup element

0016: <input class="form-control" type="number" name="ticketId" required /> // markup element

0017: </div> // markup element

0018: <div class="col-12 d-flex gap-2"> // markup element

0019: <button class="btn btn-primary">Resolve</button> // markup element

0020: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Cancel</a> // markup element

0021: </div> // markup element

0022: </form> // markup element

0023: @if (ViewBag.Success != null) // Razor conditional rendering

0024: { // statement

0025: <div class="alert alert-success mt-3">@ViewBag.Success</div> // markup element

0026: } // statement

0027: @if (ViewBag.Error != null) // Razor conditional rendering

0028: { // statement

0029: <div class="alert alert-warning mt-3">@ViewBag.Error</div> // markup element

0030: } // statement

0031: </div> // markup element

0032: </div> // markup element

0033: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/Tickets.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.SupportTicket> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Manage Tickets"; // statement

0004: } // statement

0005:

0006: <div class="container-fluid"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Support Tickets</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0010: </div> // markup element

0011: @await Html.PartialAsync("~/Views/Shared/\_AdminStats.cshtml") // statement

0012:

0013: <div class="card shadow-sm"> // markup element

0014: <div class="card-body"> // markup element

0015: <div class="row g-2 mb-3"> // markup element

0016: <div class="col-md-3"> // markup element

0017: <select id="ticketStatusFilter" class="form-select"> // markup element

0018: <option value="">All Statuses</option> // markup element

0019: <option value="OPEN">Open</option> // markup element

0020: <option value="RESOLVED">Resolved</option> // markup element

0021: </select> // markup element

0022: </div> // markup element

0023: <div class="col-md-4"> // markup element

0024: <input id="ticketSearch" class="form-control" placeholder="Search by ticket id, user id or description" /> // markup element

0025: </div> // markup element

0026: </div> // markup element

0027:

0028: <div class="table-responsive"> // markup element

0029: <table class="table table-striped align-middle" id="ticketsTable"> // markup element

0030: <thead class="table-light"> // markup element

0031: <tr> // markup element

0032: <th>#</th> // markup element

0033: <th>User Id</th> // markup element

0034: <th>Description</th> // markup element

0035: <th>Status</th> // markup element

0036: <th>Created</th> // markup element

0037: <th>Resolved</th> // markup element

0038: </tr> // markup element

0039: </thead> // markup element

0040: <tbody> // markup element

0041: @foreach (var t in Model) // statement

0042: { // statement

0043: <tr data-status="@t.TicketStatus"> // markup element

0044: <td>@t.TicketId</td> // markup element

0045: <td>@t.UserId</td> // markup element

0046: <td>@t.IssueDescription</td> // markup element

0047: <td><span class="badge @GetTicketBadge(t.TicketStatus)">@t.TicketStatus</span></td> // markup element

0048: <td>@t.CreatedDate.ToString("dd-MMM-yyyy")</td> // markup element

0049: <td>@(t.ResolvedDate.HasValue ? t.ResolvedDate.Value.ToString("dd-MMM-yyyy") : "-")</td> // markup element

0050: </tr> // markup element

0051: } // statement

0052: </tbody> // markup element

0053: </table> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057: </div> // markup element

0058:

0059: @functions{ // statement

0060: string GetTicketBadge(TicketStatus status) // statement

0061: { // statement

0062: return status switch // return value

0063: { // statement

0064: TicketStatus.OPEN => "bg-danger", // statement

0065: TicketStatus.RESOLVED => "bg-success", // statement

0066: \_ => "bg-light text-dark" // statement

0067: }; // statement

0068: } // statement

0069: } // statement

0070:

0071: @section Scripts{ // statement

0072: <script> // markup element

0073: const statusFilter = document.getElementById('ticketStatusFilter'); // statement

0074: const search = document.getElementById('ticketSearch'); // statement

0075: const table = document.getElementById('ticketsTable'); // statement

0076:

0077: function applyFilters() { // statement

0078: const status = statusFilter.value; // statement

0079: const q = search.value.toLowerCase(); // statement

0080: Array.from(table.querySelectorAll('tbody tr')).forEach(r => { // statement

0081: const matchStatus = !status || r.getAttribute('data-status') === status; // statement

0082: const text = r.innerText.toLowerCase(); // statement

0083: const matchText = text.includes(q); // statement

0084: r.style.display = matchStatus && matchText ? '' : 'none'; // statement

0085: }); // statement

0086: } // statement

0087: statusFilter.addEventListener('change', applyFilters); // statement

0088: search.addEventListener('input', applyFilters); // statement

0089: </script> // markup element

0090: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/TicketsAll.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.SupportTicket> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "All Tickets"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Tickets</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-hover align-middle"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Ticket Id</th> // markup element

0019: <th>User</th> // markup element

0020: <th>Description</th> // markup element

0021: <th>Assigned Agent</th> // markup element

0022: <th>Status</th> // markup element

0023: <th>Created</th> // markup element

0024: <th>Resolved</th> // markup element

0025: </tr> // markup element

0026: </thead> // markup element

0027: <tbody> // markup element

0028: @foreach (var t in Model) // statement

0029: { // statement

0030: <tr> // markup element

0031: <td>@t.TicketId</td> // markup element

0032: <td>@t.User?.Username</td> // markup element

0033: <td>@t.IssueDescription</td> // markup element

0034: <td>@(t.AssignedAgentId?.ToString() ?? "-")</td> // markup element

0035: <td><span class="badge @(t.TicketStatus==TicketStatus.OPEN?"bg-danger":"bg-success")">@t.TicketStatus</span></td> // markup element

0036: <td>@t.CreatedDate.ToString("dd-MMM-yyyy")</td> // markup element

0037: <td>@(t.ResolvedDate.HasValue ? t.ResolvedDate.Value.ToString("dd-MMM-yyyy") : "-")</td> // markup element

0038: </tr> // markup element

0039: } // statement

0040: </tbody> // markup element

0041: </table> // markup element

0042: </div> // markup element

0043: </div> // markup element

0044: </div> // markup element

0045: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/TicketsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Tickets Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Tickets</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AdminDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="TicketsAll"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-list-check"></i></div> // markup element

0017: <div class="fs-5 fw-bold">All Tickets</div> // markup element

0018: <div class="text-muted small">Browse all</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="TicketAssign"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-warning text-dark mb-3"><i class="bi bi-person-check"></i></div> // markup element

0028: <div class="fs-5 fw-bold">Assign Ticket</div> // markup element

0029: <div class="text-muted small">Set agent</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <a class="text-decoration-none" asp-action="TicketResolve"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-check2-circle"></i></div> // markup element

0039: <div class="fs-5 fw-bold">Resolve Ticket</div> // markup element

0040: <div class="text-muted small">Mark as resolved</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </div> // markup element

0047:

0048: <style> // markup element

0049: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0050: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0051: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0052: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/UserCreate.cshtml

0001: @model Auto\_Insurance\_System.Models.Users // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Create User"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Create User</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="UsersHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-12"> // markup element

0025: <label class="form-label">Username</label> // markup element

0026: <input class="form-control" asp-for="Username" required /> // markup element

0027: <span class="text-danger" asp-validation-for="Username"></span> // markup element

0028: </div> // markup element

0029: <div class="col-12"> // markup element

0030: <label class="form-label">Email</label> // markup element

0031: <input class="form-control" type="email" asp-for="Email" required /> // markup element

0032: <span class="text-danger" asp-validation-for="Email"></span> // markup element

0033: </div> // markup element

0034: <div class="col-12"> // markup element

0035: <label class="form-label">Password</label> // markup element

0036: <input class="form-control" type="password" asp-for="Password" required /> // markup element

0037: <span class="text-danger" asp-validation-for="Password"></span> // markup element

0038: </div> // markup element

0039: <div class="col-12"> // markup element

0040: <label class="form-label">Role</label> // markup element

0041: <select class="form-select" asp-for="Role"> // markup element

0042: <option>ADMIN</option> // markup element

0043: <option>AGENT</option> // markup element

0044: <option>CUSTOMER</option> // markup element

0045: </select> // markup element

0046: <span class="text-danger" asp-validation-for="Role"></span> // markup element

0047: </div> // markup element

0048: <div class="col-12 d-flex gap-2"> // markup element

0049: <button class="btn btn-primary">Create</button> // markup element

0050: <a class="btn btn-outline-secondary" asp-action="UsersHub">Cancel</a> // markup element

0051: </div> // markup element

0052: </form> // markup element

0053: </div> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056:

0057: @section Scripts{ // statement

0058: <partial name="\_ValidationScriptsPartial" /> // markup element

0059: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/UserDelete.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Delete User"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0007: <h2 class="mb-0">Delete User</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-action="UsersHub">Back</a> // markup element

0009: </div> // markup element

0010:

0011: @if (TempData["Success"] != null) // Razor conditional rendering

0012: { // statement

0013: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0014: } // statement

0015: @if (ViewBag.Error != null) // Razor conditional rendering

0016: { // statement

0017: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0018: } // statement

0019:

0020: <div class="card shadow-sm"> // markup element

0021: <div class="card-body"> // markup element

0022: <form method="post" class="row g-3"> // markup element

0023: <div class="col-md-4"> // markup element

0024: <label class="form-label">User Id</label> // markup element

0025: <input class="form-control" name="userId" type="number" required /> // markup element

0026: </div> // markup element

0027: <div class="col-12 d-flex gap-2"> // markup element

0028: <button class="btn btn-danger">Delete</button> // markup element

0029: <a class="btn btn-outline-secondary" asp-action="UsersHub">Cancel</a> // markup element

0030: </div> // markup element

0031: </form> // markup element

0032: </div> // markup element

0033: </div> // markup element

0034: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/UserUpdate.cshtml

0001: @model Auto\_Insurance\_System.Models.Users // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Update User"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Update User</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="UsersHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-md-4"> // markup element

0025: <label class="form-label">User Id</label> // markup element

0026: <input class="form-control" asp-for="UserId" type="number" required /> // markup element

0027: </div> // markup element

0028: <div class="col-12"> // markup element

0029: <label class="form-label">Username</label> // markup element

0030: <input class="form-control" asp-for="Username" required /> // markup element

0031: <span class="text-danger" asp-validation-for="Username"></span> // markup element

0032: </div> // markup element

0033: <div class="col-12"> // markup element

0034: <label class="form-label">Email</label> // markup element

0035: <input class="form-control" type="email" asp-for="Email" required /> // markup element

0036: <span class="text-danger" asp-validation-for="Email"></span> // markup element

0037: </div> // markup element

0038: <div class="col-12"> // markup element

0039: <label class="form-label">New Password (optional)</label> // markup element

0040: <input class="form-control" type="password" asp-for="Password" /> // markup element

0041: <span class="text-danger" asp-validation-for="Password"></span> // markup element

0042: </div> // markup element

0043: <div class="col-12"> // markup element

0044: <label class="form-label">Role</label> // markup element

0045: <select class="form-select" asp-for="Role"> // markup element

0046: <option>ADMIN</option> // markup element

0047: <option>AGENT</option> // markup element

0048: <option>CUSTOMER</option> // markup element

0049: </select> // markup element

0050: <span class="text-danger" asp-validation-for="Role"></span> // markup element

0051: </div> // markup element

0052: <div class="col-12 d-flex gap-2"> // markup element

0053: <button class="btn btn-primary">Update</button> // markup element

0054: <a class="btn btn-outline-secondary" asp-action="UsersHub">Cancel</a> // markup element

0055: </div> // markup element

0056: </form> // markup element

0057: </div> // markup element

0058: </div> // markup element

0059: </div> // markup element

0060:

0061: @section Scripts{ // statement

0062: <partial name="\_ValidationScriptsPartial" /> // markup element

0063: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/Users.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Users> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Manage Users"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Users</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="UsersHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body no-hscroll"> // markup element

0014: <div class="row g-2 mb-2 align-items-center"> // markup element

0015: <div class="col-md-6 col-lg-4"> // markup element

0016: <input id="userSearch" class="form-control" placeholder="Search by username or email" /> // markup element

0017: </div> // markup element

0018: <div class="col-auto"> // markup element

0019: <button id="searchBtn" class="btn btn-primary">Search</button> // markup element

0020: </div> // markup element

0021: <div class="col-12"> // markup element

0022: <div id="searchFeedback" class="text-danger small d-none">Not found. Provide right details.</div> // markup element

0023: </div> // markup element

0024: </div> // markup element

0025: <div> // markup element

0026: <table class="table table-hover align-middle mb-0 w-100" id="usersTable"> // markup element

0027: <thead class="table-light"> // markup element

0028: <tr> // markup element

0029: <th>Id</th> // markup element

0030: <th>Username</th> // markup element

0031: <th>Email</th> // markup element

0032: <th>Role</th> // markup element

0033: </tr> // markup element

0034: </thead> // markup element

0035: <tbody> // markup element

0036: @foreach (var u in Model) // statement

0037: { // statement

0038: <tr> // markup element

0039: <td>@u.UserId</td> // markup element

0040: <td>@u.Username</td> // markup element

0041: <td class="text-break">@u.Email</td> // markup element

0042: <td><span class="badge bg-primary">@u.Role</span></td> // markup element

0043: </tr> // markup element

0044: } // statement

0045: </tbody> // markup element

0046: </table> // markup element

0047: </div> // markup element

0048: </div> // markup element

0049: </div> // markup element

0050: </div> // markup element

0051:

0052: @section Scripts{ // statement

0053: <script> // markup element

0054: const searchInput = document.getElementById('userSearch'); // statement

0055: const searchBtn = document.getElementById('searchBtn'); // statement

0056: const table = document.getElementById('usersTable'); // statement

0057: const feedback = document.getElementById('searchFeedback'); // statement

0058:

0059: function applySearch(){ // statement

0060: const q = searchInput.value.trim().toLowerCase(); // statement

0061: let matches = 0; // statement

0062: Array.from(table.querySelectorAll('tbody tr')).forEach(row => { // statement

0063: const text = row.innerText.toLowerCase(); // statement

0064: const show = q === '' || text.includes(q); // statement

0065: row.style.display = show ? '' : 'none'; // statement

0066: if(show) matches++; // conditional check

0067: }); // statement

0068: feedback.classList.toggle('d-none', matches > 0); // statement

0069: } // statement

0070:

0071: searchBtn.addEventListener('click', applySearch); // statement

0072: searchInput.addEventListener('keydown', (e)=>{ if(e.key === 'Enter'){ e.preventDefault(); applySearch(); }}); // statement

0073: </script> // markup element

0074: } // statement

0075:

0076: <style> // markup element

0077: .no-hscroll{ overflow-x: hidden; } // statement

0078: #usersTable td{ word-break: break-word; } // statement

0079: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/UsersHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Users Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Users</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AdminDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012:

0013: <div class="col-md-3"> // markup element

0014: <a class="text-decoration-none" asp-action="UserCreate"> // markup element

0015: <div class="tile card shadow-sm border-0 h-100"> // markup element

0016: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0017: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-person-plus"></i></div> // markup element

0018: <div class="fs-5 fw-bold">Create User</div> // markup element

0019: <div class="text-muted small">Add a new user</div> // markup element

0020: </div> // markup element

0021: </div> // markup element

0022: </a> // markup element

0023: </div> // markup element

0024:

0025: <div class="col-md-3"> // markup element

0026: <a class="text-decoration-none" asp-action="Users"> // markup element

0027: <div class="tile card shadow-sm border-0 h-100"> // markup element

0028: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0029: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-people"></i></div> // markup element

0030: <div class="fs-5 fw-bold">View All Users</div> // markup element

0031: <div class="text-muted small">Browse and search</div> // markup element

0032: </div> // markup element

0033: </div> // markup element

0034: </a> // markup element

0035: </div> // markup element

0036: <div class="col-md-3"> // markup element

0037: <a class="text-decoration-none" asp-action="UserUpdate"> // markup element

0038: <div class="tile card shadow-sm border-0 h-100"> // markup element

0039: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0040: <div class="icon-circle bg-warning text-dark mb-3"><i class="bi bi-pencil-square"></i></div> // markup element

0041: <div class="fs-5 fw-bold">Update User</div> // markup element

0042: <div class="text-muted small">Modify user details</div> // markup element

0043: </div> // markup element

0044: </div> // markup element

0045: </a> // markup element

0046: </div> // markup element

0047: <div class="col-md-3"> // markup element

0048: <a class="text-decoration-none" asp-action="UserDelete"> // markup element

0049: <div class="tile card shadow-sm border-0 h-100"> // markup element

0050: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0051: <div class="icon-circle bg-danger text-white mb-3"><i class="bi bi-person-dash"></i></div> // markup element

0052: <div class="fs-5 fw-bold">Delete User</div> // markup element

0053: <div class="text-muted small">Remove by id</div> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056: </a> // markup element

0057: </div> // markup element

0058: </div> // markup element

0059: </div> // markup element

0060:

0061: <style> // markup element

0062: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0063: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0064: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0065: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Admin/\_UsersModals.cshtml

0001: <!-- Create User Modal --> // markup element

0002: <div class="modal fade" id="userCreateModal" tabindex="-1" aria-hidden="true"> // markup element

0003: <div class="modal-dialog"> // markup element

0004: <div class="modal-content"> // markup element

0005: <div class="modal-header"><h5 class="modal-title">Create User</h5><button type="button" class="btn-close" data-bs-dismiss="modal"></button></div> // markup element

0006: <div class="modal-body"> // markup element

0007: <form id="createForm" class="row g-2"> // markup element

0008: <div class="col-12"><input class="form-control" name="Username" placeholder="Username" required></div> // markup element

0009: <div class="col-12"><input class="form-control" type="email" name="Email" placeholder="Email" required></div> // markup element

0010: <div class="col-12"><input class="form-control" type="password" name="Password" placeholder="Password" required></div> // markup element

0011: <div class="col-12"> // markup element

0012: <select class="form-select" name="Role" required> // markup element

0013: <option>ADMIN</option> // markup element

0014: <option>AGENT</option> // markup element

0015: <option>CUSTOMER</option> // markup element

0016: </select> // markup element

0017: </div> // markup element

0018: </form> // markup element

0019: </div> // markup element

0020: <div class="modal-footer"><button class="btn btn-secondary" data-bs-dismiss="modal">Close</button><button id="createBtn" class="btn btn-primary">Create</button></div> // markup element

0021: </div> // markup element

0022: </div> // markup element

0023: </div> // markup element

0024:

0025: <!-- Update User Modal --> // markup element

0026: <div class="modal fade" id="userUpdateModal" tabindex="-1" aria-hidden="true"> // markup element

0027: <div class="modal-dialog"> // markup element

0028: <div class="modal-content"> // markup element

0029: <div class="modal-header"><h5 class="modal-title">Update User</h5><button type="button" class="btn-close" data-bs-dismiss="modal"></button></div> // markup element

0030: <div class="modal-body"> // markup element

0031: <form id="updateForm" class="row g-2"> // markup element

0032: <div class="col-12"><input class="form-control" type="number" name="UserId" placeholder="User Id" required></div> // markup element

0033: <div class="col-12"><input class="form-control" name="Username" placeholder="Username" required></div> // markup element

0034: <div class="col-12"><input class="form-control" type="email" name="Email" placeholder="Email" required></div> // markup element

0035: <div class="col-12"><input class="form-control" type="password" name="Password" placeholder="New Password (optional)"></div> // markup element

0036: <div class="col-12"> // markup element

0037: <select class="form-select" name="Role" required> // markup element

0038: <option>ADMIN</option> // markup element

0039: <option>AGENT</option> // markup element

0040: <option>CUSTOMER</option> // markup element

0041: </select> // markup element

0042: </div> // markup element

0043: </form> // markup element

0044: </div> // markup element

0045: <div class="modal-footer"><button class="btn btn-secondary" data-bs-dismiss="modal">Close</button><button id="updateBtn" class="btn btn-primary">Update</button></div> // markup element

0046: </div> // markup element

0047: </div> // markup element

0048: </div> // markup element

0049:

0050: <!-- Delete User Modal --> // markup element

0051: <div class="modal fade" id="userDeleteModal" tabindex="-1" aria-hidden="true"> // markup element

0052: <div class="modal-dialog"> // markup element

0053: <div class="modal-content"> // markup element

0054: <div class="modal-header"><h5 class="modal-title">Delete User</h5><button type="button" class="btn-close" data-bs-dismiss="modal"></button></div> // markup element

0055: <div class="modal-body"> // markup element

0056: <form id="deleteForm" class="row g-2"> // markup element

0057: <div class="col-12"><input class="form-control" type="number" name="userId" placeholder="User Id" required></div> // markup element

0058: </form> // markup element

0059: </div> // markup element

0060: <div class="modal-footer"><button class="btn btn-secondary" data-bs-dismiss="modal">Close</button><button id="deleteBtn" class="btn btn-danger">Delete</button></div> // markup element

0061: </div> // markup element

0062: </div> // markup element

0063: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/ClaimDetails.cshtml

0001: @model Auto\_Insurance\_System.Models.Claim // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Claim Details"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Claim Details</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (ViewBag.Error != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0015: } // statement

0016:

0017: <div class="card shadow-sm"> // markup element

0018: <div class="card-body"> // markup element

0019: <form method="post" class="row g-3"> // markup element

0020: <div class="col-md-4"> // markup element

0021: <label class="form-label">Claim Id</label> // markup element

0022: <input class="form-control" name="claimId" type="number" required /> // markup element

0023: </div> // markup element

0024: <div class="col-12 d-flex gap-2"> // markup element

0025: <button class="btn btn-primary">Get Details</button> // markup element

0026: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Cancel</a> // markup element

0027: </div> // markup element

0028: </form> // markup element

0029:

0030: @if (Model != null) // Razor conditional rendering

0031: { // statement

0032: <hr /> // markup element

0033: <dl class="row"> // markup element

0034: <dt class="col-sm-3">Policy Id</dt><dd class="col-sm-9">@Model.PolicyId</dd> // markup element

0035: <dt class="col-sm-3">Amount</dt><dd class="col-sm-9">@Model.ClaimAmount</dd> // markup element

0036: <dt class="col-sm-3">Date</dt><dd class="col-sm-9">@Model.ClaimDate.ToShortDateString()</dd> // markup element

0037: <dt class="col-sm-3">Status</dt><dd class="col-sm-9">@Model.ClaimStatus</dd> // markup element

0038: <dt class="col-sm-3">Adjuster</dt><dd class="col-sm-9">@Model.AdjusterId</dd> // markup element

0039: </dl> // markup element

0040: } // statement

0041: </div> // markup element

0042: </div> // markup element

0043: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/ClaimSubmit.cshtml

0001: @model Auto\_Insurance\_System.Models.Claim // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Submit Claim"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Submit Claim</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-md-4"> // markup element

0025: <label class="form-label">Policy Id</label> // markup element

0026: <input class="form-control" asp-for="PolicyId" type="number" required /> // markup element

0027: </div> // markup element

0028: <div class="col-md-4"> // markup element

0029: <label class="form-label">Amount</label> // markup element

0030: <input class="form-control" asp-for="ClaimAmount" type="number" step="0.01" required /> // markup element

0031: </div> // markup element

0032: <div class="col-md-4"> // markup element

0033: <label class="form-label">Date</label> // markup element

0034: <input class="form-control" asp-for="ClaimDate" type="date" required /> // markup element

0035: </div> // markup element

0036: <input type="hidden" asp-for="ClaimStatus" value="OPEN" /> // markup element

0037: <div class="col-md-4"> // markup element

0038: <label class="form-label">Adjuster Id</label> // markup element

0039: <input class="form-control" asp-for="AdjusterId" type="number" required /> // markup element

0040: </div> // markup element

0041: <div class="col-12 d-flex gap-2"> // markup element

0042: <button class="btn btn-primary">Submit</button> // markup element

0043: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Cancel</a> // markup element

0044: </div> // markup element

0045: </form> // markup element

0046: </div> // markup element

0047: </div> // markup element

0048: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/ClaimsAll.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Claim> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Claims"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Claims</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-hover align-middle mb-0"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Claim Id</th> // markup element

0019: <th>Policy</th> // markup element

0020: <th>Amount</th> // markup element

0021: <th>Date</th> // markup element

0022: <th>Status</th> // markup element

0023: <th>Adjuster</th> // markup element

0024: </tr> // markup element

0025: </thead> // markup element

0026: <tbody> // markup element

0027: @foreach (var c in Model) // statement

0028: { // statement

0029: <tr> // markup element

0030: <td>@c.ClaimId</td> // markup element

0031: <td>@c.PolicyId</td> // markup element

0032: <td>@c.ClaimAmount</td> // markup element

0033: <td>@c.ClaimDate.ToShortDateString()</td> // markup element

0034: <td><span class="badge bg-secondary">@c.ClaimStatus</span></td> // markup element

0035: <td>@c.AdjusterId</td> // markup element

0036: </tr> // markup element

0037: } // statement

0038: </tbody> // markup element

0039: </table> // markup element

0040: </div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/ClaimsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Agent • Claims Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Claims</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AgentDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="ClaimsAll"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-list"></i></div> // markup element

0017: <div class="fs-5 fw-bold">All Claims</div> // markup element

0018: <div class="text-muted small">Browse and search</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="ClaimDetails"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-info text-white mb-3"><i class="bi bi-search"></i></div> // markup element

0028: <div class="fs-5 fw-bold">Get Claim Details</div> // markup element

0029: <div class="text-muted small">Find by id</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <a class="text-decoration-none" asp-action="ClaimSubmit"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-upload"></i></div> // markup element

0039: <div class="fs-5 fw-bold">Submit Claim</div> // markup element

0040: <div class="text-muted small">Create new claim</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </div> // markup element

0047:

0048: <style> // markup element

0049: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0050: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0051: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0052: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/PaymentMake.cshtml

0001: @model Auto\_Insurance\_System.Models.Payment // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Make Payment"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Make Payment</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-md-4"> // markup element

0025: <label class="form-label">Policy Id</label> // markup element

0026: <input class="form-control" asp-for="PolicyId" type="number" required /> // markup element

0027: </div> // markup element

0028: <div class="col-md-4"> // markup element

0029: <label class="form-label">Amount</label> // markup element

0030: <input class="form-control" asp-for="PaymentAmount" type="number" step="0.01" required /> // markup element

0031: <small class="text-muted">Amount must equal the policy's premium.</small> // markup element

0032: </div> // markup element

0033: <input type="hidden" asp-for="PaymentStatus" value="SUCCESS" /> // markup element

0034: <div class="col-md-4"> // markup element

0035: <label class="form-label">Date</label> // markup element

0036: <input class="form-control" asp-for="PaymentDate" type="date" required /> // markup element

0037: </div> // markup element

0038: <div class="col-12 d-flex gap-2"> // markup element

0039: <button class="btn btn-primary">Record</button> // markup element

0040: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Cancel</a> // markup element

0041: </div> // markup element

0042: </form> // markup element

0043: </div> // markup element

0044: </div> // markup element

0045: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/Payments.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Payment> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Payments"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Payments</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-striped align-middle"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Payment Id</th> // markup element

0019: <th>Policy</th> // markup element

0020: <th>Amount</th> // markup element

0021: <th>Status</th> // markup element

0022: <th>Date</th> // markup element

0023: </tr> // markup element

0024: </thead> // markup element

0025: <tbody> // markup element

0026: @foreach (var p in Model) // statement

0027: { // statement

0028: <tr> // markup element

0029: <td>@p.PaymentId</td> // markup element

0030: <td>@p.PolicyId</td> // markup element

0031: <td>@p.PaymentAmount</td> // markup element

0032: <td><span class="badge bg-secondary">@p.PaymentStatus</span></td> // markup element

0033: <td>@p.PaymentDate.ToShortDateString()</td> // markup element

0034: </tr> // markup element

0035: } // statement

0036: </tbody> // markup element

0037: </table> // markup element

0038: </div> // markup element

0039: </div> // markup element

0040: </div> // markup element

0041: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/PaymentsByPolicy.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Payment> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Payments by Policy"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Payments by Policy</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (ViewBag.Error != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0015: } // statement

0016:

0017: <div class="card shadow-sm"> // markup element

0018: <div class="card-body"> // markup element

0019: <form method="post" class="row g-3"> // markup element

0020: <div class="col-md-4"> // markup element

0021: <label class="form-label">Policy Id</label> // markup element

0022: <input class="form-control" name="policyId" type="number" required /> // markup element

0023: </div> // markup element

0024: <div class="col-12 d-flex gap-2"> // markup element

0025: <button class="btn btn-primary">Get Payments</button> // markup element

0026: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Cancel</a> // markup element

0027: </div> // markup element

0028: </form> // markup element

0029:

0030: @if (Model != null) // Razor conditional rendering

0031: { // statement

0032: <hr /> // markup element

0033: <div class="table-responsive"> // markup element

0034: <table class="table table-hover align-middle mb-0"> // markup element

0035: <thead class="table-light"> // markup element

0036: <tr> // markup element

0037: <th>Payment Id</th> // markup element

0038: <th>Amount</th> // markup element

0039: <th>Status</th> // markup element

0040: <th>Date</th> // markup element

0041: </tr> // markup element

0042: </thead> // markup element

0043: <tbody> // markup element

0044: @foreach (var p in Model) // statement

0045: { // statement

0046: <tr> // markup element

0047: <td>@p.PaymentId</td> // markup element

0048: <td>@p.PaymentAmount</td> // markup element

0049: <td><span class="badge bg-secondary">@p.PaymentStatus</span></td> // markup element

0050: <td>@p.PaymentDate.ToShortDateString()</td> // markup element

0051: </tr> // markup element

0052: } // statement

0053: </tbody> // markup element

0054: </table> // markup element

0055: </div> // markup element

0056: } // statement

0057: </div> // markup element

0058: </div> // markup element

0059: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/PaymentsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Agent • Payments Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Payments</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AgentDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="PaymentsAll"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-cash-stack"></i></div> // markup element

0017: <div class="fs-5 fw-bold">All Payments</div> // markup element

0018: <div class="text-muted small">Browse and search</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="PaymentsByPolicy"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-info text-white mb-3"><i class="bi bi-search"></i></div> // markup element

0028: <div class="fs-5 fw-bold">By Policy Id</div> // markup element

0029: <div class="text-muted small">Filter by policy</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <a class="text-decoration-none" asp-action="PaymentMake"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-plus-circle"></i></div> // markup element

0039: <div class="fs-5 fw-bold">Make Payment</div> // markup element

0040: <div class="text-muted small">Record a payment</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </div> // markup element

0047:

0048: <style> // markup element

0049: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0050: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0051: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0052: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/PoliciesAll.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Policy> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Policies"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Policies</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-hover align-middle mb-0"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Policy Id</th> // markup element

0019: <th>Number</th> // markup element

0020: <th>Vehicle</th> // markup element

0021: <th>Coverage</th> // markup element

0022: <th>Premium</th> // markup element

0023: <th>Status</th> // markup element

0024: </tr> // markup element

0025: </thead> // markup element

0026: <tbody> // markup element

0027: @foreach (var p in Model) // statement

0028: { // statement

0029: <tr> // markup element

0030: <td>@p.PolicyId</td> // markup element

0031: <td>@p.PolicyNumber</td> // markup element

0032: <td>@p.VehicleDetails</td> // markup element

0033: <td>@p.CoverageType</td> // markup element

0034: <td>@p.PremiumAmount.ToString("C")</td> // markup element

0035: <td><span class="badge bg-secondary">@p.PolicyStatus</span></td> // markup element

0036: </tr> // markup element

0037: } // statement

0038: </tbody> // markup element

0039: </table> // markup element

0040: </div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/PoliciesHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Agent • Policies Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Policies</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AgentDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="PoliciesAll"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-journal-richtext"></i></div> // markup element

0017: <div class="fs-5 fw-bold">All Policies</div> // markup element

0018: <div class="text-muted small">Browse and search</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="PolicyDetails"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-info text-white mb-3"><i class="bi bi-search"></i></div> // markup element

0028: <div class="fs-5 fw-bold">Get Policy By ID</div> // markup element

0029: <div class="text-muted small">Find a policy</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <a class="text-decoration-none" asp-action="PolicyCreate"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-plus-square"></i></div> // markup element

0039: <div class="fs-5 fw-bold">Create Policy</div> // markup element

0040: <div class="text-muted small">Add new policy</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </div> // markup element

0047:

0048: <style> // markup element

0049: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0050: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0051: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0052: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/PolicyCreate.cshtml

0001: @model Auto\_Insurance\_System.Models.Policy // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Create Policy"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Create Policy</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-12"> // markup element

0025: <label class="form-label">Policy Number</label> // markup element

0026: <input type="hidden" class="form-control" asp-for="PolicyNumber" required /> @Model.PolicyNumber // statement

0027: @\* <span class="text-danger" asp-validation-for="PolicyNumber"></span> \*@ // comment

0028: </div> // markup element

0029: <div class="col-12"> // markup element

0030: <label class="form-label">Vehicle Details</label> // markup element

0031: <input class="form-control" asp-for="VehicleDetails" required /> // markup element

0032: <span class="text-danger" asp-validation-for="VehicleDetails"></span> // markup element

0033: </div> // markup element

0034: <div class="col-12"> // markup element

0035: <label class="form-label">Coverage Amount</label> // markup element

0036: <input class="form-control" type="number" step="0.01" asp-for="CoverageAmount" required /> // markup element

0037: <span class="text-danger" asp-validation-for="CoverageAmount"></span> // markup element

0038: </div> // markup element

0039: <div class="col-12"> // markup element

0040: <label class="form-label">Coverage Type</label> // markup element

0041: <input class="form-control" asp-for="CoverageType" required /> // markup element

0042: <span class="text-danger" asp-validation-for="CoverageType"></span> // markup element

0043: </div> // markup element

0044: <div class="col-12"> // markup element

0045: <label class="form-label">Premium Amount</label> // markup element

0046: <input class="form-control" type="number" step="0.01" asp-for="PremiumAmount" required /> // markup element

0047: <span class="text-danger" asp-validation-for="PremiumAmount"></span> // markup element

0048: </div> // markup element

0049: <div class="col-md-6"> // markup element

0050: <label class="form-label">Start Date</label> // markup element

0051: <input class="form-control" type="date" asp-for="StartDate" required /> // markup element

0052: <span class="text-danger" asp-validation-for="StartDate"></span> // markup element

0053: </div> // markup element

0054: <div class="col-md-6"> // markup element

0055: <label class="form-label">End Date</label> // markup element

0056: <input class="form-control" type="date" asp-for="EndDate" required /> // markup element

0057: <span class="text-danger" asp-validation-for="EndDate"></span> // markup element

0058: </div> // markup element

0059: <input type="hidden" asp-for="PolicyStatus" value="ACTIVE" /> // markup element

0060: <div class="col-12 d-flex gap-2"> // markup element

0061: <button class="btn btn-primary">Create</button> // markup element

0062: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Cancel</a> // markup element

0063: </div> // markup element

0064: </form> // markup element

0065: </div> // markup element

0066: </div> // markup element

0067: </div> // markup element

0068:

0069: @section Scripts{ // statement

0070: <partial name="\_ValidationScriptsPartial" /> // markup element

0071: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/PolicyDetails.cshtml

0001: @model Auto\_Insurance\_System.Models.Policy // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Policy Details"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Policy Details</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (ViewBag.Error != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0015: } // statement

0016:

0017: <div class="card shadow-sm"> // markup element

0018: <div class="card-body"> // markup element

0019: <form method="post" class="row g-3"> // markup element

0020: <div class="col-md-4"> // markup element

0021: <label class="form-label">Policy Id</label> // markup element

0022: <input class="form-control" name="policyId" type="number" required /> // markup element

0023: </div> // markup element

0024: <div class="col-12 d-flex gap-2"> // markup element

0025: <button class="btn btn-primary">Get Details</button> // markup element

0026: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Cancel</a> // markup element

0027: </div> // markup element

0028: </form> // markup element

0029:

0030: @if (Model != null) // Razor conditional rendering

0031: { // statement

0032: <hr /> // markup element

0033: <dl class="row"> // markup element

0034: <dt class="col-sm-3">Policy Number</dt><dd class="col-sm-9">@Model.PolicyNumber</dd> // markup element

0035: <dt class="col-sm-3">Vehicle</dt><dd class="col-sm-9">@Model.VehicleDetails</dd> // markup element

0036: <dt class="col-sm-3">Coverage</dt><dd class="col-sm-9">@Model.CoverageType</dd> // markup element

0037: <dt class="col-sm-3">Premium</dt><dd class="col-sm-9">@Model.PremiumAmount</dd> // markup element

0038: <dt class="col-sm-3">Dates</dt><dd class="col-sm-9">@Model.StartDate.ToShortDateString() - @Model.EndDate.ToShortDateString()</dd> // markup element

0039: <dt class="col-sm-3">Status</dt><dd class="col-sm-9">@Model.PolicyStatus</dd> // markup element

0040: </dl> // markup element

0041: } // statement

0042: </div> // markup element

0043: </div> // markup element

0044: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/TicketCreate.cshtml

0001: @model Auto\_Insurance\_System.Models.SupportTicket // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Create Ticket"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Create Ticket</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-md-4"> // markup element

0025: <label class="form-label">User Id</label> // markup element

0026: <input class="form-control" asp-for="UserId" type="number" required /> // markup element

0027: </div> // markup element

0028: <div class="col-md-8"> // markup element

0029: <label class="form-label">Issue Description</label> // markup element

0030: <input class="form-control" asp-for="IssueDescription" required /> // markup element

0031: </div> // markup element

0032: <input type="hidden" asp-for="TicketStatus" value="OPEN" /> // markup element

0033: <div class="col-12 d-flex gap-2"> // markup element

0034: <button class="btn btn-primary">Create</button> // markup element

0035: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Cancel</a> // markup element

0036: </div> // markup element

0037: </form> // markup element

0038: </div> // markup element

0039: </div> // markup element

0040: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/TicketResolve.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Agent • Resolve Ticket"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0007: <h2 class="mb-0">Resolve Ticket</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0009: </div> // markup element

0010:

0011: @if (ViewBag.Success != null) // Razor conditional rendering

0012: { // statement

0013: <div class="alert alert-success">@ViewBag.Success</div> // markup element

0014: } // statement

0015: @if (ViewBag.Error != null) // Razor conditional rendering

0016: { // statement

0017: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0018: } // statement

0019:

0020: <div class="card shadow-sm"> // markup element

0021: <div class="card-body"> // markup element

0022: <form method="post" class="row g-3"> // markup element

0023: <div class="col-md-4"> // markup element

0024: <label class="form-label">Ticket Id</label> // markup element

0025: <input class="form-control" name="ticketId" type="number" required /> // markup element

0026: </div> // markup element

0027: <div class="col-12 d-flex gap-2"> // markup element

0028: <button class="btn btn-primary">Resolve</button> // markup element

0029: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Cancel</a> // markup element

0030: </div> // markup element

0031: </form> // markup element

0032: </div> // markup element

0033: </div> // markup element

0034: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/Tickets.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.SupportTicket> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Tickets"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Support Tickets</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="row g-2 mb-3"> // markup element

0015: <div class="col-md-3"> // markup element

0016: <select id="ticketStatusFilter" class="form-select"> // markup element

0017: <option value="">All Statuses</option> // markup element

0018: <option value="OPEN">Open</option> // markup element

0019: <option value="RESOLVED">Resolved</option> // markup element

0020: </select> // markup element

0021: </div> // markup element

0022: <div class="col-md-4"> // markup element

0023: <input id="ticketSearch" class="form-control" placeholder="Search by ticket id, user id or description" /> // markup element

0024: </div> // markup element

0025: </div> // markup element

0026:

0027: <div class="table-responsive"> // markup element

0028: <table class="table table-striped align-middle" id="ticketsTable"> // markup element

0029: <thead class="table-light"> // markup element

0030: <tr> // markup element

0031: <th>Ticket Id</th> // markup element

0032: <th>User Id</th> // markup element

0033: <th>Description</th> // markup element

0034: <th>Status</th> // markup element

0035: <th>Created</th> // markup element

0036: <th>Resolved</th> // markup element

0037: </tr> // markup element

0038: </thead> // markup element

0039: <tbody> // markup element

0040: @foreach (var t in Model) // statement

0041: { // statement

0042: <tr data-status="@t.TicketStatus"> // markup element

0043: <td>@t.TicketId</td> // markup element

0044: <td>@t.UserId</td> // markup element

0045: <td>@t.IssueDescription</td> // markup element

0046: <td><span class="badge @(t.TicketStatus == TicketStatus.OPEN ? "bg-danger" : "bg-success")">@t.TicketStatus</span></td> // markup element

0047: <td>@t.CreatedDate.ToString("dd-MMM-yyyy")</td> // markup element

0048: <td>@(t.ResolvedDate.HasValue ? t.ResolvedDate.Value.ToString("dd-MMM-yyyy") : "-")</td> // markup element

0049: </tr> // markup element

0050: } // statement

0051: </tbody> // markup element

0052: </table> // markup element

0053: </div> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057:

0058: @section Scripts{ // statement

0059: <script> // markup element

0060: const statusFilter = document.getElementById('ticketStatusFilter'); // statement

0061: const search = document.getElementById('ticketSearch'); // statement

0062: const table = document.getElementById('ticketsTable'); // statement

0063:

0064: function applyFilters() { // statement

0065: const status = statusFilter.value; // statement

0066: const q = search.value.toLowerCase(); // statement

0067: Array.from(table.querySelectorAll('tbody tr')).forEach(r => { // statement

0068: const matchStatus = !status || r.getAttribute('data-status') === status; // statement

0069: const text = r.innerText.toLowerCase(); // statement

0070: const matchText = text.includes(q); // statement

0071: r.style.display = matchStatus && matchText ? '' : 'none'; // statement

0072: }); // statement

0073: } // statement

0074: statusFilter.addEventListener('change', applyFilters); // statement

0075: search.addEventListener('input', applyFilters); // statement

0076: </script> // markup element

0077: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/TicketsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Agent • Tickets Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Support Tickets</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AgentDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="Tickets"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-list"></i></div> // markup element

0017: <div class="fs-5 fw-bold">All Tickets</div> // markup element

0018: <div class="text-muted small">Browse and search</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="TicketCreate"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-plus-square"></i></div> // markup element

0028: <div class="fs-5 fw-bold">Create Ticket</div> // markup element

0029: <div class="text-muted small">Log an issue</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: <div class="col-md-3"> // markup element

0035: <a class="text-decoration-none" asp-action="TicketResolve"> // markup element

0036: <div class="tile card shadow-sm border-0 h-100"> // markup element

0037: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0038: <div class="icon-circle bg-warning text-dark mb-3"><i class="bi bi-check2-square"></i></div> // markup element

0039: <div class="fs-5 fw-bold">Resolve Ticket</div> // markup element

0040: <div class="text-muted small">Resolve assigned</div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </a> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </div> // markup element

0047:

0048: <style> // markup element

0049: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0050: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0051: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0052: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/UserCreate.cshtml

0001: @model Auto\_Insurance\_System.Models.Users // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Create User"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Create User</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="UsersHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-12"> // markup element

0025: <label class="form-label">Username</label> // markup element

0026: <input class="form-control" asp-for="Username" required /> // markup element

0027: <span class="text-danger" asp-validation-for="Username"></span> // markup element

0028: </div> // markup element

0029: <div class="col-12"> // markup element

0030: <label class="form-label">Email</label> // markup element

0031: <input class="form-control" type="email" asp-for="Email" required /> // markup element

0032: <span class="text-danger" asp-validation-for="Email"></span> // markup element

0033: </div> // markup element

0034: <div class="col-12"> // markup element

0035: <label class="form-label">Password</label> // markup element

0036: <input class="form-control" type="password" asp-for="Password" required /> // markup element

0037: <span class="text-danger" asp-validation-for="Password"></span> // markup element

0038: </div> // markup element

0039: <div class="col-12"> // markup element

0040: <label class="form-label">Role</label> // markup element

0041: <select class="form-select" asp-for="Role"> // markup element

0042: <option>AGENT</option> // markup element

0043: <option>CUSTOMER</option> // markup element

0044: </select> // markup element

0045: <span class="text-danger" asp-validation-for="Role"></span> // markup element

0046: </div> // markup element

0047: <div class="col-12 d-flex gap-2"> // markup element

0048: <button class="btn btn-primary">Create</button> // markup element

0049: <a class="btn btn-outline-secondary" asp-action="UsersHub">Cancel</a> // markup element

0050: </div> // markup element

0051: </form> // markup element

0052: </div> // markup element

0053: </div> // markup element

0054: </div> // markup element

0055:

0056: @section Scripts{ // statement

0057: <partial name="\_ValidationScriptsPartial" /> // markup element

0058: } // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/Users.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Users> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Agent • Users"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Users</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="UsersHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body no-hscroll"> // markup element

0014: <div class="row g-2 mb-2 align-items-center"> // markup element

0015: <div class="col-md-6 col-lg-4"> // markup element

0016: <input id="userSearch" class="form-control" placeholder="Search by username or email" /> // markup element

0017: </div> // markup element

0018: <div class="col-auto"> // markup element

0019: <button id="searchBtn" class="btn btn-primary">Search</button> // markup element

0020: </div> // markup element

0021: <div class="col-12"> // markup element

0022: <div id="searchFeedback" class="text-danger small d-none">Not found. Provide right details.</div> // markup element

0023: </div> // markup element

0024: </div> // markup element

0025: <div> // markup element

0026: <table class="table table-hover align-middle mb-0 w-100" id="usersTable"> // markup element

0027: <thead class="table-light"> // markup element

0028: <tr> // markup element

0029: <th>Id</th> // markup element

0030: <th>Username</th> // markup element

0031: <th>Email</th> // markup element

0032: <th>Role</th> // markup element

0033: </tr> // markup element

0034: </thead> // markup element

0035: <tbody> // markup element

0036: @foreach (var u in Model) // statement

0037: { // statement

0038: <tr> // markup element

0039: <td>@u.UserId</td> // markup element

0040: <td>@u.Username</td> // markup element

0041: <td class="text-break">@u.Email</td> // markup element

0042: <td><span class="badge bg-primary">@u.Role</span></td> // markup element

0043: </tr> // markup element

0044: } // statement

0045: </tbody> // markup element

0046: </table> // markup element

0047: </div> // markup element

0048: </div> // markup element

0049: </div> // markup element

0050: </div> // markup element

0051:

0052: @section Scripts{ // statement

0053: <script> // markup element

0054: const searchInput = document.getElementById('userSearch'); // statement

0055: const searchBtn = document.getElementById('searchBtn'); // statement

0056: const table = document.getElementById('usersTable'); // statement

0057: const feedback = document.getElementById('searchFeedback'); // statement

0058:

0059: function applySearch(){ // statement

0060: const q = searchInput.value.trim().toLowerCase(); // statement

0061: let matches = 0; // statement

0062: Array.from(table.querySelectorAll('tbody tr')).forEach(row => { // statement

0063: const text = row.innerText.toLowerCase(); // statement

0064: const show = q === '' || text.includes(q); // statement

0065: row.style.display = show ? '' : 'none'; // statement

0066: if(show) matches++; // conditional check

0067: }); // statement

0068: feedback.classList.toggle('d-none', matches > 0); // statement

0069: } // statement

0070:

0071: searchBtn.addEventListener('click', applySearch); // statement

0072: searchInput.addEventListener('keydown', (e)=>{ if(e.key === 'Enter'){ e.preventDefault(); applySearch(); }}); // statement

0073: </script> // markup element

0074: } // statement

0075:

0076: <style> // markup element

0077: .no-hscroll{ overflow-x: hidden; } // statement

0078: #usersTable td{ word-break: break-word; } // statement

0079: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Agent/UsersHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Agent • Users Hub"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Users</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="AgentDashboard">Dashboard</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="Users"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-primary text-white mb-3"><i class="bi bi-people"></i></div> // markup element

0017: <div class="fs-5 fw-bold">View All Users</div> // markup element

0018: <div class="text-muted small">Browse and search</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="UserCreate"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-success text-white mb-3"><i class="bi bi-person-plus"></i></div> // markup element

0028: <div class="fs-5 fw-bold">Create User</div> // markup element

0029: <div class="text-muted small">Add a new user</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: </div> // markup element

0036:

0037: <style> // markup element

0038: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0039: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0040: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0041: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Auth/AdminDashboard.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Admin Dashboard"; // statement

0003: var username = Context.Session.GetString("UserName"); // local variable declaration

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <h2 class="mb-1">Welcome, @username!</h2> // markup element

0008: <p class="text-muted">Choose a module to manage.</p> // markup element

0009:

0010: @await Html.PartialAsync("~/Views/Shared/\_AdminStats.cshtml") // statement

0011:

0012: <div class="row g-4 mt-1"> // markup element

0013: <div class="col-md-4 col-lg-3"> // markup element

0014: <a class="text-decoration-none" asp-controller="Admin" asp-action="UsersHub"> // markup element

0015: <div class="tile card shadow-sm border-0 h-100"> // markup element

0016: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0017: <div class="icon-circle bg-primary text-white mb-3">👥</div> // markup element

0018: <div class="fs-5 fw-bold">Users</div> // markup element

0019: <div class="text-muted small">Create ,view,update, delete</div> // markup element

0020: </div> // markup element

0021: </div> // markup element

0022: </a> // markup element

0023: </div> // markup element

0024: <div class="col-md-4 col-lg-3"> // markup element

0025: <a class="text-decoration-none" asp-controller="Admin" asp-action="PoliciesHub"> // markup element

0026: <div class="tile card shadow-sm border-0 h-100"> // markup element

0027: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0028: <div class="icon-circle bg-success text-white mb-3">📑</div> // markup element

0029: <div class="fs-5 fw-bold">Policies</div> // markup element

0030: <div class="text-muted small">Update, list, delete</div> // markup element

0031: </div> // markup element

0032: </div> // markup element

0033: </a> // markup element

0034: </div> // markup element

0035: <div class="col-md-4 col-lg-3"> // markup element

0036: <a class="text-decoration-none" asp-controller="Admin" asp-action="ClaimsHub"> // markup element

0037: <div class="tile card shadow-sm border-0 h-100"> // markup element

0038: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0039: <div class="icon-circle bg-warning text-dark mb-3">📝</div> // markup element

0040: <div class="fs-5 fw-bold">Claims</div> // markup element

0041: <div class="text-muted small">Update, list</div> // markup element

0042: </div> // markup element

0043: </div> // markup element

0044: </a> // markup element

0045: </div> // markup element

0046: <div class="col-md-4 col-lg-3"> // markup element

0047: <a class="text-decoration-none" asp-controller="Admin" asp-action="PaymentsHub"> // markup element

0048: <div class="tile card shadow-sm border-0 h-100"> // markup element

0049: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0050: <div class="icon-circle bg-info text-white mb-3">💳</div> // markup element

0051: <div class="fs-5 fw-bold">Payments</div> // markup element

0052: <div class="text-muted small">List, by policy, update</div> // markup element

0053: </div> // markup element

0054: </div> // markup element

0055: </a> // markup element

0056: </div> // markup element

0057: <div class="col-md-4 col-lg-3"> // markup element

0058: <a class="text-decoration-none" asp-controller="Admin" asp-action="TicketsHub"> // markup element

0059: <div class="tile card shadow-sm border-0 h-100"> // markup element

0060: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0061: <div class="icon-circle bg-danger text-white mb-3">🎫</div> // markup element

0062: <div class="fs-5 fw-bold">Tickets</div> // markup element

0063: <div class="text-muted small">Assign, resolve, list</div> // markup element

0064: </div> // markup element

0065: </div> // markup element

0066: </a> // markup element

0067: </div> // markup element

0068: </div> // markup element

0069: </div> // markup element

0070:

0071: <style> // markup element

0072: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0073: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0074: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0075: </style> // markup element

0076:

0077:

# Auto-Insurance/Auto\_Insurance\_System/Views/Auth/AgentDashboard.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Agent Dashboard"; // statement

0003: var username = Context.Session.GetString("UserName"); // local variable declaration

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="mb-4"> // markup element

0008: <h1 class="display-5 fw-semibold text-dark">Welcome, @username!</h1> // markup element

0009: <p class="text-muted">This is your Agent Dashboard.</p> // markup element

0010: </div> // markup element

0011:

0012: @await Html.PartialAsync("~/Views/Shared/\_AdminStats.cshtml") // statement

0013:

0014: <div class="row g-4"> // markup element

0015: <div class="col-md-4 col-lg-3"> // markup element

0016: <a class="text-decoration-none" asp-controller="Agent" asp-action="UsersHub"> // markup element

0017: <div class="tile card shadow-sm border-0 h-100"> // markup element

0018: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0019: <div class="icon-circle bg-primary text-white mb-3">👥</div> // markup element

0020: <div class="fs-5 fw-bold">Users</div> // markup element

0021: <div class="text-muted small">View / Create</div> // markup element

0022: </div> // markup element

0023: </div> // markup element

0024: </a> // markup element

0025: </div> // markup element

0026: <div class="col-md-4 col-lg-3"> // markup element

0027: <a class="text-decoration-none" asp-controller="Agent" asp-action="PoliciesHub"> // markup element

0028: <div class="tile card shadow-sm border-0 h-100"> // markup element

0029: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0030: <div class="icon-circle bg-success text-white mb-3">📄</div> // markup element

0031: <div class="fs-5 fw-bold">Policies</div> // markup element

0032: <div class="text-muted small">All / By Id / Create</div> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: </a> // markup element

0036: </div> // markup element

0037: <div class="col-md-4 col-lg-3"> // markup element

0038: <a class="text-decoration-none" asp-controller="Agent" asp-action="ClaimsHub"> // markup element

0039: <div class="tile card shadow-sm border-0 h-100"> // markup element

0040: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0041: <div class="icon-circle bg-warning text-dark mb-3">🧾</div> // markup element

0042: <div class="fs-5 fw-bold">Claims</div> // markup element

0043: <div class="text-muted small">All / Details / Submit</div> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </a> // markup element

0047: </div> // markup element

0048: <div class="col-md-4 col-lg-3"> // markup element

0049: <a class="text-decoration-none" asp-controller="Agent" asp-action="PaymentsHub"> // markup element

0050: <div class="tile card shadow-sm border-0 h-100"> // markup element

0051: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0052: <div class="icon-circle bg-info text-white mb-3">💳</div> // markup element

0053: <div class="fs-5 fw-bold">Payments</div> // markup element

0054: <div class="text-muted small">All / By Policy / Make</div> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057: </a> // markup element

0058: </div> // markup element

0059: <div class="col-md-4 col-lg-3"> // markup element

0060: <a class="text-decoration-none" asp-controller="Agent" asp-action="TicketsHub"> // markup element

0061: <div class="tile card shadow-sm border-0 h-100"> // markup element

0062: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0063: <div class="icon-circle bg-danger text-white mb-3">🎫</div> // markup element

0064: <div class="fs-5 fw-bold">Tickets</div> // markup element

0065: <div class="text-muted small">All / Create / Resolve</div> // markup element

0066: </div> // markup element

0067: </div> // markup element

0068: </a> // markup element

0069: </div> // markup element

0070: </div> // markup element

0071: </div> // markup element

0072:

0073: <style> // markup element

0074: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0075: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0076: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0077: </style> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Auth/CustomerDashboard.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Customer Dashboard"; // statement

0003: var username = Context.Session.GetString("UserName"); // local variable declaration

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="mb-4"> // markup element

0008: <h1 class="display-5 fw-semibold text-dark">Welcome, @username!</h1> // markup element

0009: <p class="text-muted">This is your Customer Dashboard.</p> // markup element

0010: </div> // markup element

0011:

0012: <div class="row g-4"> // markup element

0013: <div class="col-md-4 col-lg-3"> // markup element

0014: <a class="text-decoration-none" asp-controller="Customer" asp-action="PoliciesHub"> // markup element

0015: <div class="tile card shadow-sm border-0 h-100"> // markup element

0016: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0017: <div class="icon-circle bg-success text-white mb-3">📄</div> // markup element

0018: <div class="fs-5 fw-bold">Policies</div> // markup element

0019: <div class="text-muted small">Create policy</div> // markup element

0020: </div> // markup element

0021: </div> // markup element

0022: </a> // markup element

0023: </div> // markup element

0024: <div class="col-md-4 col-lg-3"> // markup element

0025: <a class="text-decoration-none" asp-controller="Customer" asp-action="ClaimsHub"> // markup element

0026: <div class="tile card shadow-sm border-0 h-100"> // markup element

0027: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0028: <div class="icon-circle bg-warning text-dark mb-3">🧾</div> // markup element

0029: <div class="fs-5 fw-bold">Claims</div> // markup element

0030: <div class="text-muted small">Submit / view</div> // markup element

0031: </div> // markup element

0032: </div> // markup element

0033: </a> // markup element

0034: </div> // markup element

0035: <div class="col-md-4 col-lg-3"> // markup element

0036: <a class="text-decoration-none" asp-controller="Customer" asp-action="PaymentsHub"> // markup element

0037: <div class="tile card shadow-sm border-0 h-100"> // markup element

0038: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0039: <div class="icon-circle bg-info text-white mb-3">💳</div> // markup element

0040: <div class="fs-5 fw-bold">Payments</div> // markup element

0041: <div class="text-muted small">Make / view</div> // markup element

0042: </div> // markup element

0043: </div> // markup element

0044: </a> // markup element

0045: </div> // markup element

0046: <div class="col-md-4 col-lg-3"> // markup element

0047: <a class="text-decoration-none" asp-controller="Customer" asp-action="TicketsHub"> // markup element

0048: <div class="tile card shadow-sm border-0 h-100"> // markup element

0049: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0050: <div class="icon-circle bg-primary text-white mb-3">🎫</div> // markup element

0051: <div class="fs-5 fw-bold">Tickets</div> // markup element

0052: <div class="text-muted small">Create / view status</div> // markup element

0053: </div> // markup element

0054: </div> // markup element

0055: </a> // markup element

0056: </div> // markup element

0057: </div> // markup element

0058: </div> // markup element

0059:

0060: <style> // markup element

0061: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0062: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0063: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0064: </style> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Auth/ForgotPassword.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Forgot Password · Auto Insurance System"; // statement

0003: } // statement

0004:

0005: <div class="container py-5"> // markup element

0006: <div class="row justify-content-center"> // markup element

0007: <div class="col-md-6"> // markup element

0008: <div class="card border-0 shadow-lg"> // markup element

0009: <div class="card-body p-4"> // markup element

0010: <h2 class="mb-3">Reset Password</h2> // markup element

0011:

0012: @if (ViewData.ModelState.ErrorCount > 0) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-danger"> // markup element

0015: @foreach (var error in ViewData.ModelState.Values.SelectMany(v => v.Errors)) // statement

0016: { // statement

0017: <div>@error.ErrorMessage</div> // markup element

0018: } // statement

0019: </div> // markup element

0020: } // statement

0021:

0022: <form asp-action="ForgotPassword" method="post"> // markup element

0023: @Html.AntiForgeryToken() // statement

0024: <div class="form-text mb-2">Provide either your Username or Email.</div> // markup element

0025: <div class="form-floating mb-3"> // markup element

0026: <input type="text" class="form-control" id="Username" name="Username" placeholder="Username" /> // markup element

0027: <label for="Username">Username</label> // markup element

0028: </div> // markup element

0029: <div class="form-floating mb-3"> // markup element

0030: <input type="email" class="form-control" id="Email" name="Email" placeholder="Email" /> // markup element

0031: <label for="Email">Email</label> // markup element

0032: </div> // markup element

0033:

0034: <div class="form-floating mb-3"> // markup element

0035: <input type="password" class="form-control" id="NewPassword" name="NewPassword" placeholder="New Password" required /> // markup element

0036: <label for="NewPassword">New Password</label> // markup element

0037: </div> // markup element

0038: <div class="form-floating mb-3"> // markup element

0039: <input type="password" class="form-control" id="ConfirmPassword" name="ConfirmPassword" placeholder="Confirm Password" required /> // markup element

0040: <label for="ConfirmPassword">Confirm Password</label> // markup element

0041: </div> // markup element

0042:

0043: <button type="submit" class="btn btn-app-primary w-100">Update Password</button> // markup element

0044: </form> // markup element

0045:

0046: <div class="text-center mt-3"> // markup element

0047: <a asp-controller="Auth" asp-action="Login">Back to Login</a> // markup element

0048: </div> // markup element

0049: </div> // markup element

0050: </div> // markup element

0051: </div> // markup element

0052: </div> // markup element

0053: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Auth/Login.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Login · Auto Insurance System"; // statement

0003: } // statement

0004:

0005: <div class="container py-5"> // markup element

0006: <div class="row justify-content-center"> // markup element

0007: <div class="col-md-6"> // markup element

0008: <div class="card border-0 shadow-lg"> // markup element

0009: <div class="card-body p-4"> // markup element

0010: <div class="d-flex align-items-center mb-3"> // markup element

0011: <div class="icon-circle bg-app-primary text-white">🔐</div> // markup element

0012: <h2 class="ms-2 mb-0">Login</h2> // markup element

0013: </div> // markup element

0014:

0015: @if (ViewData.ModelState.ErrorCount > 0) // Razor conditional rendering

0016: { // statement

0017: <div class="alert alert-danger text-center"> // markup element

0018: @foreach (var error in ViewData.ModelState.Values.SelectMany(v => v.Errors)) // statement

0019: { // statement

0020: <div>@error.ErrorMessage</div> // markup element

0021: } // statement

0022: </div> // markup element

0023: } // statement

0024:

0025: <form asp-action="Login" method="post"> // markup element

0026: <div class="form-floating mb-3"> // markup element

0027: <input type="text" name="Username" id="Username" class="form-control" placeholder="Username" required /> // markup element

0028: <label for="Username">Username</label> // markup element

0029: </div> // markup element

0030:

0031: <div class="form-floating mb-3"> // markup element

0032: <input type="password" name="Password" id="Password" class="form-control" placeholder="Password" required /> // markup element

0033: <label for="Password">Password</label> // markup element

0034: </div> // markup element

0035:

0036: <div class="form-floating mb-3"> // markup element

0037: <select name="Role" id="Role" class="form-select" required> // markup element

0038: <option value="">Select Role</option> // markup element

0039: <option value="ADMIN">Admin</option> // markup element

0040: <option value="AGENT">Agent</option> // markup element

0041: <option value="CUSTOMER">Customer</option> // markup element

0042: </select> // markup element

0043: <label for="Role">User Type</label> // markup element

0044: </div> // markup element

0045:

0046: <button type="submit" class="btn btn-app-primary w-100">Login</button> // markup element

0047: </form> // markup element

0048:

0049: <div class="text-center mt-3 auth-links"> // markup element

0050: <a href="/Auth/ForgotPassword">Forgot Password?</a> | // statement

0051: <a href="/Auth/Register">Register</a> // markup element

0052: </div> // markup element

0053: </div> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057: </div> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Auth/Register.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Register"; // statement

0003: } // statement

0004:

0005: <!-- Font Awesome for icons --> // markup element

0006: <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" /> // markup element

0007:

0008: <div class="container mt-5"> // markup element

0009: @\* Success Message Block \*@ // comment

0010: @{ // Razor code block start

0011: var successMessage = ViewBag.SuccessMessage as string; // local variable declaration

0012: var triggerRedirect = ViewBag.TriggerRedirect as bool?; // local variable declaration

0013: } // statement

0014:

0015: @if (!string.IsNullOrEmpty(successMessage)) // Razor conditional rendering

0016: { // statement

0017: <div class="alert alert-success text-center mt-4"> // markup element

0018: <i class="fas fa-check-circle text-success me-2"></i> // markup element

0019: @successMessage // statement

0020: </div> // markup element

0021:

0022: <script> // markup element

0023: setTimeout(function () { // statement

0024: window.location.href = '@Url.Action("Login", "Auth")'; // statement

0025: }, 3000); // statement

0026: </script> // markup element

0027: } // statement

0028: <h2 class="text-center mb-4">Choose Your Role to Register</h2> // markup element

0029:

0030: <div class="row text-center mb-5"> // markup element

0031: <!-- Admin Card --> // markup element

0032: <div class="col-md-4"> // markup element

0033: <div class="role-card admin-card" onclick="showForm('ADMIN')"> // markup element

0034: <i class="fas fa-user-shield fa-2x mb-2 text-danger"></i> // markup element

0035: <h4>Admin</h4> // markup element

0036: <p>Manage users, policies, and system settings.</p> // markup element

0037:

0038: <div class="form-container" id="form-ADMIN" style="display:none;"> // markup element

0039: <form asp-action="Register" method="post"> // markup element

0040: <input type="hidden" name="Role" value="ADMIN" /> // markup element

0041: <input type="text" name="Username" class="form-control mb-2" placeholder="Username" required /> // markup element

0042: <input type="password" name="Password" class="form-control mb-2" placeholder="Password" required /> // markup element

0043: <input type="email" name="Email" class="form-control mb-3" placeholder="Email" required /> // markup element

0044: <button type="submit" class="btn btn-danger w-100">Register as Admin</button> // markup element

0045: </form> // markup element

0046: </div> // markup element

0047: </div> // markup element

0048: </div> // markup element

0049:

0050: <!-- Agent Card --> // markup element

0051: <div class="col-md-4"> // markup element

0052: <div class="role-card agent-card" onclick="showForm('AGENT')"> // markup element

0053: <i class="fas fa-user-tie fa-2x mb-2 text-warning"></i> // markup element

0054: <h4>Agent</h4> // markup element

0055: <p>Handle claims, assist customers, and track policies.</p> // markup element

0056:

0057: <div class="form-container" id="form-AGENT" style="display:none;"> // markup element

0058: <form asp-action="Register" method="post"> // markup element

0059: <input type="hidden" name="Role" value="AGENT" /> // markup element

0060: <input type="text" name="Username" class="form-control mb-2" placeholder="Username" required /> // markup element

0061: <input type="password" name="Password" class="form-control mb-2" placeholder="Password" required /> // markup element

0062: <input type="email" name="Email" class="form-control mb-3" placeholder="Email" required /> // markup element

0063: <button type="submit" class="btn btn-warning w-100 text-dark">Register as Agent</button> // markup element

0064: </form> // markup element

0065: </div> // markup element

0066: </div> // markup element

0067: </div> // markup element

0068:

0069: <!-- Customer Card --> // markup element

0070: <div class="col-md-4"> // markup element

0071: <div class="role-card customer-card" onclick="showForm('CUSTOMER')"> // markup element

0072: <i class="fas fa-user fa-2x mb-2 text-success"></i> // markup element

0073: <h4>Customer</h4> // markup element

0074: <p>View policies, file claims, and manage your account.</p> // markup element

0075:

0076: <div class="form-container" id="form-CUSTOMER" style="display:none;"> // markup element

0077: <form asp-action="Register" method="post"> // markup element

0078: <input type="hidden" name="Role" value="CUSTOMER" /> // markup element

0079: <input type="text" name="Username" class="form-control mb-2" placeholder="Username" required /> // markup element

0080: <input type="password" name="Password" class="form-control mb-2" placeholder="Password" required /> // markup element

0081: <input type="email" name="Email" class="form-control mb-3" placeholder="Email" required /> // markup element

0082: <button type="submit" class="btn btn-success w-100">Register as Customer</button> // markup element

0083: </form> // markup element

0084: </div> // markup element

0085: </div> // markup element

0086: </div> // markup element

0087: </div> // markup element

0088: </div> // markup element

0089:

0090: <!-- Styling --> // markup element

0091: <style> // markup element

0092: .role-card { // statement

0093: border: 2px solid #ccc; // statement

0094: border-radius: 15px; // statement

0095: padding: 25px; // statement

0096: min-height: 320px; // statement

0097: transition: all 0.3s ease; // statement

0098: position: relative; // statement

0099: background: linear-gradient(to bottom right, #f8f9fa, #e9ecef); // statement

0100: box-shadow: 0 4px 8px rgba(0,0,0,0.1); // statement

0101: cursor: pointer; // statement

0102: } // statement

0103:

0104: .role-card:hover { // statement

0105: transform: scale(1.03); // statement

0106: box-shadow: 0 6px 12px rgba(0,0,0,0.15); // statement

0107: } // statement

0108:

0109: .admin-card { // statement

0110: border-left: 6px solid #dc3545; // statement

0111: } // statement

0112:

0113: .agent-card { // statement

0114: border-left: 6px solid #ffc107; // statement

0115: } // statement

0116:

0117: .customer-card { // statement

0118: border-left: 6px solid #28a745; // statement

0119: } // statement

0120:

0121: .form-container { // statement

0122: margin-top: 20px; // statement

0123: animation: fadeIn 0.4s ease-in-out; // statement

0124: } // statement

0125:

0126: body { // statement

0127: background: linear-gradient(to bottom right, #f0f4f8, #d9e2ec); // statement

0128: } // statement

0129:

0130:

0131: .form-control { // statement

0132: border-radius: 8px; // statement

0133: } // statement

0134:

0135: .btn { // statement

0136: border-radius: 8px; // statement

0137: font-weight: bold; // statement

0138: } // statement

0139: </style> // markup element

0140:

0141: <style> // markup element

0142: @@keyframes fadeIn { // statement

0143: from { // statement

0144: opacity: 0; // statement

0145: transform: translateY(10px); // statement

0146: } // statement

0147:

0148: to { // statement

0149: opacity: 1; // statement

0150: transform: translateY(0); // statement

0151: } // statement

0152: } // statement

0153:

0154: .form-container { // statement

0155: animation: fadeIn 0.4s ease-in-out; // statement

0156: } // statement

0157: </style> // markup element

0158:

0159:

0160: <!-- Script --> // markup element

0161: <script> // markup element

0162: function showForm(role) { // statement

0163: // Hide all forms // comment

0164: document.querySelectorAll('.form-container').forEach(form => form.style.display = 'none'); // statement

0165:

0166: // Show selected form // comment

0167: document.getElementById(`form-${role}`).style.display = 'block'; // statement

0168:

0169: // Scroll to selected card // comment

0170: const card = document.querySelector(`.${role.toLowerCase()}-card`); // statement

0171: window.scrollTo({ top: card.offsetTop, behavior: 'smooth' }); // statement

0172: } // statement

0173: </script> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/ClaimSubmit.cshtml

0001: @model Auto\_Insurance\_System.Models.Claim // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • Submit Claim"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Submit Claim</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-md-4"> // markup element

0025: <label class="form-label">Policy Id</label> // markup element

0026: <input class="form-control" asp-for="PolicyId" type="number" required /> // markup element

0027: </div> // markup element

0028: <div class="col-md-4"> // markup element

0029: <label class="form-label">Amount</label> // markup element

0030: <input class="form-control" asp-for="ClaimAmount" type="number" step="0.01" required /> // markup element

0031: </div> // markup element

0032: <div class="col-md-4"> // markup element

0033: <label class="form-label">Date</label> // markup element

0034: <input class="form-control" asp-for="ClaimDate" type="date" required /> // markup element

0035: </div> // markup element

0036: <input type="hidden" asp-for="ClaimStatus" value="OPEN" /> // markup element

0037: <div class="col-md-4"> // markup element

0038: <label class="form-label">Adjuster Id</label> // markup element

0039: <input class="form-control" asp-for="AdjusterId" type="number" required /> // markup element

0040: </div> // markup element

0041: <div class="col-12 d-flex gap-2"> // markup element

0042: <button class="btn btn-primary">Submit</button> // markup element

0043: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Cancel</a> // markup element

0044: </div> // markup element

0045: </form> // markup element

0046: </div> // markup element

0047: </div> // markup element

0048: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/Claims.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Claim> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • My Claims"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">My Claims</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="ClaimsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-hover align-middle mb-0"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Claim Id</th> // markup element

0019: <th>Policy</th> // markup element

0020: <th>Amount</th> // markup element

0021: <th>Date</th> // markup element

0022: <th>Status</th> // markup element

0023: </tr> // markup element

0024: </thead> // markup element

0025: <tbody> // markup element

0026: @foreach (var c in Model) // statement

0027: { // statement

0028: <tr> // markup element

0029: <td>@c.ClaimId</td> // markup element

0030: <td>@c.PolicyId</td> // markup element

0031: <td>@c.ClaimAmount</td> // markup element

0032: <td>@c.ClaimDate.ToShortDateString()</td> // markup element

0033: <td><span class="badge bg-secondary">@c.ClaimStatus</span></td> // markup element

0034: </tr> // markup element

0035: } // statement

0036: </tbody> // markup element

0037: </table> // markup element

0038: </div> // markup element

0039: </div> // markup element

0040: </div> // markup element

0041: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/ClaimsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Customer • Claims"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Claims</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="CustomerDashboard">Back</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="ClaimSubmit"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-success text-white mb-3">🧾</div> // markup element

0017: <div class="fs-5 fw-bold">Submit Claim</div> // markup element

0018: <div class="text-muted small">Create new claim</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="Claims"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-primary text-white mb-3">📃</div> // markup element

0028: <div class="fs-5 fw-bold">My Claims</div> // markup element

0029: <div class="text-muted small">View submitted</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: </div> // markup element

0036:

0037: <style> // markup element

0038: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0039: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0040: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0041: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/PaymentMake.cshtml

0001: @model Auto\_Insurance\_System.Models.Payment // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • Make Payment"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Make Payment</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-md-4"> // markup element

0025: <label class="form-label">Policy Id</label> // markup element

0026: <input class="form-control" asp-for="PolicyId" type="number" required /> // markup element

0027: </div> // markup element

0028: <div class="col-md-4"> // markup element

0029: <label class="form-label">Amount</label> // markup element

0030: <input class="form-control" asp-for="PaymentAmount" type="number" step="0.01" required /> // markup element

0031: <small class="text-muted">Amount must equal the policy's premium.</small> // markup element

0032: </div> // markup element

0033: <input type="hidden" asp-for="PaymentStatus" value="SUCCESS" /> // markup element

0034: <div class="col-md-4"> // markup element

0035: <label class="form-label">Date</label> // markup element

0036: <input class="form-control" asp-for="PaymentDate" type="date" required /> // markup element

0037: </div> // markup element

0038: <div class="col-12 d-flex gap-2"> // markup element

0039: <button class="btn btn-primary">Record</button> // markup element

0040: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Cancel</a> // markup element

0041: </div> // markup element

0042: </form> // markup element

0043: </div> // markup element

0044: </div> // markup element

0045: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/Payments.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Payment> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • My Payments"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">My Payments</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PaymentsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-striped align-middle"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Payment Id</th> // markup element

0019: <th>Policy</th> // markup element

0020: <th>Amount</th> // markup element

0021: <th>Status</th> // markup element

0022: <th>Date</th> // markup element

0023: </tr> // markup element

0024: </thead> // markup element

0025: <tbody> // markup element

0026: @foreach (var p in Model) // statement

0027: { // statement

0028: <tr> // markup element

0029: <td>@p.PaymentId</td> // markup element

0030: <td>@p.PolicyId</td> // markup element

0031: <td>@p.PaymentAmount</td> // markup element

0032: <td><span class="badge bg-secondary">@p.PaymentStatus</span></td> // markup element

0033: <td>@p.PaymentDate.ToShortDateString()</td> // markup element

0034: </tr> // markup element

0035: } // statement

0036: </tbody> // markup element

0037: </table> // markup element

0038: </div> // markup element

0039: </div> // markup element

0040: </div> // markup element

0041: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/PaymentsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Customer • Payments"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Payments</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="CustomerDashboard">Back</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="PaymentMake"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-success text-white mb-3">💳</div> // markup element

0017: <div class="fs-5 fw-bold">Make Payment</div> // markup element

0018: <div class="text-muted small">Record a payment</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="Payments"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-primary text-white mb-3">📃</div> // markup element

0028: <div class="fs-5 fw-bold">My Payments</div> // markup element

0029: <div class="text-muted small">View history</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: </div> // markup element

0036:

0037: <style> // markup element

0038: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0039: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0040: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0041: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/Policies.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.Policy> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • My Policies"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">My Policies</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-hover align-middle mb-0"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Policy Id</th> // markup element

0019: <th>Number</th> // markup element

0020: <th>Vehicle</th> // markup element

0021: <th>Coverage</th> // markup element

0022: <th>Premium</th> // markup element

0023: <th>Status</th> // markup element

0024: </tr> // markup element

0025: </thead> // markup element

0026: <tbody> // markup element

0027: @foreach (var p in Model) // statement

0028: { // statement

0029: <tr> // markup element

0030: <td>@p.PolicyId</td> // markup element

0031: <td>@p.PolicyNumber</td> // markup element

0032: <td>@p.VehicleDetails</td> // markup element

0033: <td>@p.CoverageType</td> // markup element

0034: <td>@p.PremiumAmount.ToString("C")</td> // markup element

0035: <td><span class="badge bg-secondary">@p.PolicyStatus</span></td> // markup element

0036: </tr> // markup element

0037: } // statement

0038: </tbody> // markup element

0039: </table> // markup element

0040: </div> // markup element

0041: </div> // markup element

0042: </div> // markup element

0043: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/PoliciesHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Customer • Policies"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Policies</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="CustomerDashboard">Back</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="PolicyCreate"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-success text-white mb-3">📄</div> // markup element

0017: <div class="fs-5 fw-bold">Create Policy</div> // markup element

0018: <div class="text-muted small">Add a new policy</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="Policies"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-primary text-white mb-3">📃</div> // markup element

0028: <div class="fs-5 fw-bold">My Policies</div> // markup element

0029: <div class="text-muted small">View list</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: </div> // markup element

0036:

0037: <style> // markup element

0038: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0039: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0040: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0041: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/PolicyCreate.cshtml

0001: @model Auto\_Insurance\_System.Models.Policy // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • Create Policy"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Create Policy</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-12"> // markup element

0025: <label class="form-label">Policy Number</label> // markup element

0026: <input type="hidden" class="form-control" asp-for="PolicyNumber" required /> @Model.PolicyNumber // statement

0027: </div> // markup element

0028: <div class="col-12"> // markup element

0029: <label class="form-label">Vehicle Details</label> // markup element

0030: <input class="form-control" asp-for="VehicleDetails" required /> // markup element

0031: </div> // markup element

0032: <div class="col-12"> // markup element

0033: <label class="form-label">Coverage Amount</label> // markup element

0034: <input class="form-control" type="number" step="0.01" asp-for="CoverageAmount" required /> // markup element

0035: </div> // markup element

0036: <div class="col-12"> // markup element

0037: <label class="form-label">Coverage Type</label> // markup element

0038: <input class="form-control" asp-for="CoverageType" required /> // markup element

0039: </div> // markup element

0040: <div class="col-12"> // markup element

0041: <label class="form-label">Premium Amount</label> // markup element

0042: <input class="form-control" type="number" step="0.01" asp-for="PremiumAmount" required /> // markup element

0043: </div> // markup element

0044: <div class="col-md-6"> // markup element

0045: <label class="form-label">Start Date</label> // markup element

0046: <input class="form-control" type="date" asp-for="StartDate" required /> // markup element

0047: </div> // markup element

0048: <div class="col-md-6"> // markup element

0049: <label class="form-label">End Date</label> // markup element

0050: <input class="form-control" type="date" asp-for="EndDate" required /> // markup element

0051: </div> // markup element

0052: <input type="hidden" asp-for="PolicyStatus" value="ACTIVE" /> // markup element

0053: <div class="col-12 d-flex gap-2"> // markup element

0054: <button class="btn btn-primary">Create</button> // markup element

0055: <a class="btn btn-outline-secondary" asp-action="PoliciesHub">Cancel</a> // markup element

0056: </div> // markup element

0057: </form> // markup element

0058: </div> // markup element

0059: </div> // markup element

0060: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/TicketCreate.cshtml

0001: @model Auto\_Insurance\_System.Models.SupportTicket // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • Create Ticket"; // statement

0004: } // statement

0005:

0006: <div class="container py-4"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">Create Ticket</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: @if (TempData["Success"] != null) // Razor conditional rendering

0013: { // statement

0014: <div class="alert alert-success">@TempData["Success"]</div> // markup element

0015: } // statement

0016: @if (ViewBag.Error != null) // Razor conditional rendering

0017: { // statement

0018: <div class="alert alert-warning">@ViewBag.Error</div> // markup element

0019: } // statement

0020:

0021: <div class="card shadow-sm"> // markup element

0022: <div class="card-body"> // markup element

0023: <form method="post" class="row g-3"> // markup element

0024: <div class="col-md-4"> // markup element

0025: <label class="form-label">User Id</label> // markup element

0026: <input class="form-control" asp-for="UserId" type="number" required /> // markup element

0027: </div> // markup element

0028: <div class="col-md-8"> // markup element

0029: <label class="form-label">Issue Description</label> // markup element

0030: <input class="form-control" asp-for="IssueDescription" required /> // markup element

0031: </div> // markup element

0032: <input type="hidden" asp-for="TicketStatus" value="OPEN" /> // markup element

0033: <div class="col-12 d-flex gap-2"> // markup element

0034: <button class="btn btn-primary">Create</button> // markup element

0035: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Cancel</a> // markup element

0036: </div> // markup element

0037: </form> // markup element

0038: </div> // markup element

0039: </div> // markup element

0040: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/Tickets.cshtml

0001: @model IEnumerable<Auto\_Insurance\_System.Models.SupportTicket> // strongly-typed view model declaration

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Customer • My Tickets"; // statement

0004: } // statement

0005:

0006: <div class="container"> // markup element

0007: <div class="d-flex align-items-center justify-content-between mb-3"> // markup element

0008: <h2 class="mb-0">My Tickets</h2> // markup element

0009: <a class="btn btn-outline-secondary" asp-action="TicketsHub">Back</a> // markup element

0010: </div> // markup element

0011:

0012: <div class="card shadow-sm"> // markup element

0013: <div class="card-body"> // markup element

0014: <div class="table-responsive"> // markup element

0015: <table class="table table-striped align-middle"> // markup element

0016: <thead class="table-light"> // markup element

0017: <tr> // markup element

0018: <th>Ticket Id</th> // markup element

0019: <th>Description</th> // markup element

0020: <th>Status</th> // markup element

0021: <th>Created</th> // markup element

0022: <th>Resolved</th> // markup element

0023: </tr> // markup element

0024: </thead> // markup element

0025: <tbody> // markup element

0026: @foreach (var t in Model) // statement

0027: { // statement

0028: <tr> // markup element

0029: <td>@t.TicketId</td> // markup element

0030: <td>@t.IssueDescription</td> // markup element

0031: <td><span class="badge @(t.TicketStatus == TicketStatus.OPEN ? "bg-danger" : "bg-success")">@t.TicketStatus</span></td> // markup element

0032: <td>@t.CreatedDate.ToString("dd-MMM-yyyy")</td> // markup element

0033: <td>@(t.ResolvedDate.HasValue ? t.ResolvedDate.Value.ToString("dd-MMM-yyyy") : "-")</td> // markup element

0034: </tr> // markup element

0035: } // statement

0036: </tbody> // markup element

0037: </table> // markup element

0038: </div> // markup element

0039: </div> // markup element

0040: </div> // markup element

0041: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Customer/TicketsHub.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Customer • Tickets"; // statement

0003: } // statement

0004:

0005: <div class="container py-4"> // markup element

0006: <div class="d-flex justify-content-between align-items-center mb-3"> // markup element

0007: <h2 class="mb-0">Support Tickets</h2> // markup element

0008: <a class="btn btn-outline-secondary" asp-controller="Auth" asp-action="CustomerDashboard">Back</a> // markup element

0009: </div> // markup element

0010:

0011: <div class="row g-4 mt-1"> // markup element

0012: <div class="col-md-3"> // markup element

0013: <a class="text-decoration-none" asp-action="TicketCreate"> // markup element

0014: <div class="tile card shadow-sm border-0 h-100"> // markup element

0015: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0016: <div class="icon-circle bg-success text-white mb-3">🎫</div> // markup element

0017: <div class="fs-5 fw-bold">Create Ticket</div> // markup element

0018: <div class="text-muted small">Log an issue</div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </a> // markup element

0022: </div> // markup element

0023: <div class="col-md-3"> // markup element

0024: <a class="text-decoration-none" asp-action="Tickets"> // markup element

0025: <div class="tile card shadow-sm border-0 h-100"> // markup element

0026: <div class="card-body d-flex flex-column justify-content-center align-items-start"> // markup element

0027: <div class="icon-circle bg-primary text-white mb-3">📃</div> // markup element

0028: <div class="fs-5 fw-bold">My Tickets</div> // markup element

0029: <div class="text-muted small">View status</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </a> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: </div> // markup element

0036:

0037: <style> // markup element

0038: .icon-circle{ width:46px;height:46px;border-radius:50%;display:inline-flex;align-items:center;justify-content:center; } // statement

0039: .tile{ transition:transform .15s ease, box-shadow .15s ease; } // statement

0040: .tile:hover{ transform:translateY(-2px); box-shadow:0 .75rem 1rem rgba(0,0,0,.08)!important; } // statement

0041: </style> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Home/About.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "About · Auto Insurance System"; // statement

0003: ViewBag.FullWidth = true; // statement

0004: } // statement

0005:

0006: <style> // markup element

0007: /\* Scoped styles for About page \*/ // comment

0008: .about-hero { // statement

0009: background: linear-gradient(135deg, #0d6efd 0%, #0b5ed7 40%, #084298 100%); // statement

0010: color: #fff; // statement

0011: position: relative; // statement

0012: overflow: hidden; // statement

0013: } // statement

0014: .about-hero::after { // statement

0015: content: ""; // statement

0016: position: absolute; // statement

0017: right: -10rem; // statement

0018: top: -10rem; // statement

0019: width: 32rem; // statement

0020: height: 32rem; // statement

0021: background: radial-gradient(closest-side, rgba(255,255,255,.15), rgba(255,255,255,0)); // statement

0022: filter: blur(2px); // statement

0023: border-radius: 50%; // statement

0024: } // statement

0025: .about-badge { // statement

0026: display: inline-flex; // statement

0027: align-items: center; // statement

0028: gap: .5rem; // statement

0029: background: rgba(255,255,255,.12); // statement

0030: border: 1px solid rgba(255,255,255,.2); // statement

0031: color: #fff; // statement

0032: padding: .375rem .75rem; // statement

0033: border-radius: 999px; // statement

0034: font-size: .875rem; // statement

0035: letter-spacing: .02em; // statement

0036: backdrop-filter: blur(2px); // statement

0037: } // statement

0038: .about-card { // statement

0039: border: 1px solid #eef1f5; // statement

0040: box-shadow: 0 6px 18px rgba(16,24,40,.06); // statement

0041: border-radius: 1rem; // statement

0042: transition: transform .2s ease, box-shadow .2s ease; // statement

0043: background: #fff; // statement

0044: } // statement

0045: .about-card:hover { transform: translateY(-2px); box-shadow: 0 10px 24px rgba(16,24,40,.12); } // statement

0046:

0047: .feature-icon { // statement

0048: width: 44px; height: 44px; // statement

0049: border-radius: .75rem; // statement

0050: display: inline-flex; align-items: center; justify-content: center; // statement

0051: background: #eef4ff; color: #0d6efd; // statement

0052: box-shadow: inset 0 0 0 1px rgba(13,110,253,.15); // statement

0053: } // statement

0054: .timeline { // statement

0055: position: relative; // statement

0056: padding-left: 1.75rem; // statement

0057: } // statement

0058: .timeline::before { // statement

0059: content: ""; position: absolute; top: .25rem; bottom: .25rem; left: .5rem; width: 2px; background: #e5e7eb; // statement

0060: } // statement

0061: .timeline .dot { width: .75rem; height: .75rem; background: #0d6efd; border-radius: 999px; position: absolute; left: .125rem; margin-top: .5rem; } // statement

0062:

0063: .faq-item { border-bottom: 1px solid #eef1f5; } // statement

0064: .faq-item:last-child { border-bottom: 0; } // statement

0065:

0066: .cta-banner { background: linear-gradient(135deg, #f8fafc 0%, #f1f5f9 100%); border: 1px solid #edf2f7; } // statement

0067:

0068: /\* Page background wrapper \*/ // comment

0069: .about-page-bg { background: linear-gradient(180deg, #f6f9ff 0%, #f8fafc 100%); } // statement

0070: </style> // markup element

0071: <div class="about-page-bg"> // markup element

0072:

0073: <section class="about-hero py-5 py-lg-6 full-bleed"> // markup element

0074: <div class="container-xl"> // markup element

0075: <div class="row align-items-center g-4"> // markup element

0076: <div class="col-12 col-lg-8"> // markup element

0077:

0078: <h1 class="display-5 fw-bold mb-3">About Auto Insurance System</h1> // markup element

0079: <p class="lead mb-4 opacity-75"> // markup element

0080: A modern, secure platform for managing policies, claims, payments, and support—crafted for customers, // statement

0081: agents, and administrators. // statement

0082: </p> // markup element

0083: <div class="d-flex flex-wrap gap-2"> // markup element

0084: <a class="btn btn-light text-primary fw-semibold" asp-controller="Home" asp-action="Landing">Explore Portals</a> // markup element

0085: <a class="btn btn-outline-light" asp-controller="Home" asp-action="Privacy">Privacy Policy</a> // markup element

0086: </div> // markup element

0087: </div> // markup element

0088: </div> // markup element

0089: </div> // markup element

0090: </section> // markup element

0091:

0092: <section class="py-5"> // markup element

0093: <div class="container"> // markup element

0094: <div class="row g-4"> // markup element

0095: <div class="col-md-4"> // markup element

0096: <div class="about-card h-100 p-4"> // markup element

0097: <div class="feature-icon mb-3">🔒</div> // markup element

0098: <h5 class="fw-semibold mb-2">Security First</h5> // markup element

0099: <p class="text-secondary mb-0">Role-based access ensures Admins, Agents, and Customers see exactly what they need—nothing more.</p> // markup element

0100: </div> // markup element

0101: </div> // markup element

0102: <div class="col-md-4"> // markup element

0103: <div class="about-card h-100 p-4"> // markup element

0104: <div class="feature-icon mb-3">⚡</div> // markup element

0105: <h5 class="fw-semibold mb-2">Fast & Responsive</h5> // markup element

0106: <p class="text-secondary mb-0">Optimized pages and clean UI components deliver a smooth experience on any device.</p> // markup element

0107: </div> // markup element

0108: </div> // markup element

0109: <div class="col-md-4"> // markup element

0110: <div class="about-card h-100 p-4"> // markup element

0111: <div class="feature-icon mb-3">📈</div> // markup element

0112: <h5 class="fw-semibold mb-2">Operational Insight</h5> // markup element

0113: <p class="text-secondary mb-0">Admins get at‑a‑glance metrics across policies, claims, payments, and tickets.</p> // markup element

0114: </div> // markup element

0115: </div> // markup element

0116: </div> // markup element

0117: </div> // markup element

0118: </section> // markup element

0119:

0120: <section class="py-5 bg-light"> // markup element

0121: <div class="container"> // markup element

0122: <div class="row g-4"> // markup element

0123: <div class="col-lg-6"> // markup element

0124: <div class="about-card p-4 p-lg-5 h-100"> // markup element

0125: <h4 class="fw-semibold mb-3">Who’s it for?</h4> // markup element

0126: <div class="row row-cols-1 row-cols-sm-2 g-3"> // markup element

0127: <div class="col"> // markup element

0128: <div class="p-3 border rounded-3 h-100"> // markup element

0129: <div class="small text-uppercase text-secondary mb-1">Customers</div> // markup element

0130: <div class="mb-2">View active policies, file claims, and pay securely.</div> // markup element

0131: <a class="link-primary small" asp-controller="Auth" asp-action="Login" asp-route-Role="CUSTOMER">Login as Customer</a> // markup element

0132: </div> // markup element

0133: </div> // markup element

0134: <div class="col"> // markup element

0135: <div class="p-3 border rounded-3 h-100"> // markup element

0136: <div class="small text-uppercase text-secondary mb-1">Agents</div> // markup element

0137: <div class="mb-2">Manage claims, users, and tickets with clear workflows.</div> // markup element

0138: <a class="link-warning small" asp-controller="Auth" asp-action="Login" asp-route-Role="AGENT">Login as Agent</a> // markup element

0139: </div> // markup element

0140: </div> // markup element

0141: <div class="col"> // markup element

0142: <div class="p-3 border rounded-3 h-100"> // markup element

0143: <div class="small text-uppercase text-secondary mb-1">Admins</div> // markup element

0144: <div class="mb-2">Oversee users, policies, payments, and system health.</div> // markup element

0145: <a class="link-success small" asp-controller="Auth" asp-action="Login" asp-route-Role="ADMIN">Login as Admin</a> // markup element

0146: </div> // markup element

0147: </div> // markup element

0148: <div class="col"> // markup element

0149: <div class="p-3 border rounded-3 h-100"> // markup element

0150: <div class="small text-uppercase text-secondary mb-1">Auditors</div> // markup element

0151: <div class="mb-0">Export reports and review historical data for compliance.</div> // markup element

0152: </div> // markup element

0153: </div> // markup element

0154: </div> // markup element

0155: </div> // markup element

0156: </div> // markup element

0157: <div class="col-lg-6"> // markup element

0158: <div class="about-card p-4 p-lg-5 h-100"> // markup element

0159: <h4 class="fw-semibold mb-3">How it works</h4> // markup element

0160: <div class="timeline"> // markup element

0161: <div class="position-relative ps-4 mb-3"> // markup element

0162: <div class="dot"></div> // markup element

0163: <div class="fw-semibold">Authenticate</div> // markup element

0164: <small class="text-secondary">Sign in with your role to access tailored tools.</small> // markup element

0165: </div> // markup element

0166: <div class="position-relative ps-4 mb-3"> // markup element

0167: <div class="dot"></div> // markup element

0168: <div class="fw-semibold">Take Action</div> // markup element

0169: <small class="text-secondary">Create policies, submit or review claims, and manage payments.</small> // markup element

0170: </div> // markup element

0171: <div class="position-relative ps-4"> // markup element

0172: <div class="dot"></div> // markup element

0173: <div class="fw-semibold">Track & Report</div> // markup element

0174: <small class="text-secondary">Monitor statuses and export insights for decision‑making.</small> // markup element

0175: </div> // markup element

0176: </div> // markup element

0177: </div> // markup element

0178: </div> // markup element

0179: </div> // markup element

0180: </div> // markup element

0181: </section> // markup element

0182:

0183: <section class="py-5"> // markup element

0184: <div class="container"> // markup element

0185: <div class="row g-4"> // markup element

0186: <div class="col-lg-7"> // markup element

0187: <div class="about-card p-4 p-lg-5 h-100"> // markup element

0188: <h4 class="fw-semibold mb-3">Frequently Asked Questions</h4> // markup element

0189: <div class="faq-item py-3"> // markup element

0190: <button class="btn btn-link text-decoration-none p-0 w-100 text-start d-flex justify-content-between align-items-center" data-faq> // markup element

0191: <span class="fw-semibold">Can I switch roles after logging in?</span> // markup element

0192: <span class="text-secondary">▼</span> // markup element

0193: </button> // markup element

0194: <div class="collapse show mt-2" data-faq-panel> // markup element

0195: <small class="text-secondary">Yes. Log out and sign back in with the intended role. Each role has unique permissions.</small> // markup element

0196: </div> // markup element

0197: </div> // markup element

0198: <div class="faq-item py-3"> // markup element

0199: <button class="btn btn-link text-decoration-none p-0 w-100 text-start d-flex justify-content-between align-items-center" data-faq> // markup element

0200: <span class="fw-semibold">How are payments handled?</span> // markup element

0201: <span class="text-secondary">▼</span> // markup element

0202: </button> // markup element

0203: <div class="collapse mt-2" data-faq-panel> // markup element

0204: <small class="text-secondary">Payments are tracked with clear statuses. Admins can review transactions and monthly revenue.</small> // markup element

0205: </div> // markup element

0206: </div> // markup element

0207: <div class="faq-item py-3"> // markup element

0208: <button class="btn btn-link text-decoration-none p-0 w-100 text-start d-flex justify-content-between align-items-center" data-faq> // markup element

0209: <span class="fw-semibold">What happens if I’m not authorized?</span> // markup element

0210: <span class="text-secondary">▼</span> // markup element

0211: </button> // markup element

0212: <div class="collapse mt-2" data-faq-panel> // markup element

0213: <small class="text-secondary">You’ll be redirected to an Access Denied page. Contact an administrator if you require elevated permissions.</small> // markup element

0214: </div> // markup element

0215: </div> // markup element

0216: </div> // markup element

0217: </div> // markup element

0218: <div class="col-lg-5"> // markup element

0219: <div class="about-card p-4 p-lg-5 cta-banner h-100"> // markup element

0220: <div class="d-flex align-items-center mb-3"> // markup element

0221: <div class="feature-icon me-3">🚀</div> // markup element

0222: <div class="fw-semibold">Get started now</div> // markup element

0223: </div> // markup element

0224: <p class="text-secondary">Choose your portal and experience a streamlined insurance workflow.</p> // markup element

0225: <div class="d-grid gap-2"> // markup element

0226: <a class="btn btn-primary" asp-controller="Auth" asp-action="Login" asp-route-Role="CUSTOMER">Login as Customer</a> // markup element

0227: <a class="btn btn-outline-primary" asp-controller="Auth" asp-action="Login" asp-route-Role="AGENT">Login as Agent</a> // markup element

0228: <a class="btn btn-light" asp-controller="Auth" asp-action="Login" asp-route-Role="ADMIN">Login as Admin</a> // markup element

0229: </div> // markup element

0230: </div> // markup element

0231: </div> // markup element

0232: </div> // markup element

0233: </div> // markup element

0234: </section> // markup element

0235:

0236: </div> // markup element

0237:

0238: <script> // markup element

0239: // Minimal JS to enhance FAQ toggles // comment

0240: document.querySelectorAll('[data-faq]').forEach(function(btn){ // statement

0241: btn.addEventListener('click', function(){ // statement

0242: const panel = this.parentElement.querySelector('[data-faq-panel]'); // statement

0243: document.querySelectorAll('[data-faq-panel]').forEach(p => { if (p !== panel) p.classList.remove('show'); }); // statement

0244: panel.classList.toggle('show'); // statement

0245: }); // statement

0246: }); // statement

0247: </script> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Home/Index.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Home"; // statement

0003: } // statement

0004: <!-- Redirect handled by HomeController.Index() to Landing --> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Home/Landing.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Auto Insurance System"; // statement

0003: ViewBag.FullWidth = true; // statement

0004: ViewBag.NoTopPadding = true; // statement

0005: ViewBag.BodyClass = "is-landing"; // statement

0006: } // statement

0007:

0008: @if (TempData["LoginError"] != null) // Razor conditional rendering

0009: { // statement

0010: <div class="alert alert-danger text-center"> // markup element

0011: @TempData["LoginError"] // statement

0012: </div> // markup element

0013: } // statement

0014:

0015: <section class="hero-banner text-white"> // markup element

0016: <div class="px-3 px-md-4 py-4 py-lg-5"> // markup element

0017: <div class="row align-items-center g-4"> // markup element

0018: <div class="col-12 col-lg-10"> // markup element

0019: <div class="d-flex align-items-center mb-3"> // markup element

0020: <div class="logo-wrap me-3"> // markup element

0021: <svg width="56" height="56" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg"> // markup element

0022: <rect width="24" height="24" rx="6" fill="#ffffff"/> // markup element

0023: <path d="M5 13c1.5-3.5 4.5-5.5 7-5.5s5.5 2 7 5.5" stroke="#0d6efd" stroke-width="2" stroke-linecap="round"/> // markup element

0024: <path d="M5 15h14" stroke="#0d6efd" stroke-width="2" stroke-linecap="round"/> // markup element

0025: <circle cx="12" cy="10" r="2" fill="#0d6efd"/> // markup element

0026: </svg> // markup element

0027: </div> // markup element

0028: <div> // markup element

0029: <h1 class="display-5 fw-bold m-0">Auto Insurance System</h1> // markup element

0030: <p class="lead m-0">Comprehensive claims and policy management.</p> // markup element

0031: </div> // markup element

0032: </div> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: </div> // markup element

0036: </section> // markup element

0037:

0038: <section class="py-5"> // markup element

0039: <div class="container"> // markup element

0040: <div class="row g-4"> // markup element

0041: <div class="col-md-4"> // markup element

0042: <div class="card card-portal h-100"> // markup element

0043: <div class="card-body"> // markup element

0044: <div class="portal-icon bg-primary-subtle text-app-primary"> // markup element

0045: <svg width="28" height="28" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg"> // markup element

0046: <circle cx="12" cy="8" r="3.5" stroke="#0d6efd" stroke-width="2" /> // markup element

0047: <path d="M5 20c0-3.5 3.5-6 7-6s7 2.5 7 6" stroke="#0d6efd" stroke-width="2" stroke-linecap="round"/> // markup element

0048: </svg> // markup element

0049: </div> // markup element

0050: <h5 class="card-title mt-3">Customer Portal</h5> // markup element

0051: <p class="card-text">Access your policies, file claims, and track claim status.</p> // markup element

0052: <a class="stretched-link" asp-controller="Auth" asp-action="Login" asp-route-Role="CUSTOMER" aria-label="Login as Customer"></a> // markup element

0053: <a class="btn btn-outline-primary w-100" asp-controller="Auth" asp-action="Login" asp-route-Role="CUSTOMER">Login as Customer</a> // markup element

0054: </div> // markup element

0055: </div> // markup element

0056: </div> // markup element

0057: <div class="col-md-4"> // markup element

0058: <div class="card card-portal h-100"> // markup element

0059: <div class="card-body"> // markup element

0060: <div class="portal-icon bg-warning-subtle text-warning"> // markup element

0061: <svg width="28" height="28" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg"> // markup element

0062: <path d="M5 5h14v10H5z" stroke="#f0ad4e" stroke-width="2"/> // markup element

0063: <path d="M9 19h6" stroke="#f0ad4e" stroke-width="2" stroke-linecap="round"/> // markup element

0064: <circle cx="12" cy="10" r="2" fill="#f0ad4e"/> // markup element

0065: </svg> // markup element

0066: </div> // markup element

0067: <h5 class="card-title mt-3">Agent Workspace</h5> // markup element

0068: <p class="card-text">Manage assigned claims, update statuses, and assist customers.</p> // markup element

0069: <a class="stretched-link" asp-controller="Auth" asp-action="Login" asp-route-Role="AGENT" aria-label="Login as Agent"></a> // markup element

0070: <a class="btn btn-outline-warning w-100" asp-controller="Auth" asp-action="Login" asp-route-Role="AGENT">Login as Agent</a> // markup element

0071: </div> // markup element

0072: </div> // markup element

0073: </div> // markup element

0074: <div class="col-md-4"> // markup element

0075: <div class="card card-portal h-100"> // markup element

0076: <div class="card-body"> // markup element

0077: <div class="portal-icon bg-success-subtle text-success"> // markup element

0078: <svg width="28" height="28" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg"> // markup element

0079: <path d="M5 7h14v10H5z" stroke="#28a745" stroke-width="2"/> // markup element

0080: <path d="M8 10h5M8 13h8" stroke="#28a745" stroke-width="2" stroke-linecap="round"/> // markup element

0081: </svg> // markup element

0082: </div> // markup element

0083: <h5 class="card-title mt-3">Admin Panel</h5> // markup element

0084: <p class="card-text">Oversee users, policies, claims, payments, and support tickets.</p> // markup element

0085: <a class="stretched-link" asp-controller="Auth" asp-action="Login" asp-route-Role="ADMIN" aria-label="Login as Admin"></a> // markup element

0086: <a class="btn btn-success w-100" asp-controller="Auth" asp-action="Login" asp-route-Role="ADMIN">Login as Admin</a> // markup element

0087: </div> // markup element

0088: </div> // markup element

0089: </div> // markup element

0090: </div> // markup element

0091: </div> // markup element

0092: </section> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Home/Privacy.cshtml

0001: ﻿@{ // statement

0002: ViewData["Title"] = "Privacy Policy"; // statement

0003: } // statement

0004:

0005: <section class="pb-2"> // markup element

0006: <h1 class="h4 fw-semibold mb-1">@ViewData["Title"]</h1> // markup element

0007: <p class="text-secondary small mb-0">Your privacy matters. This Policy explains what we collect, why we collect it, and how you can manage your information when using Auto Insurance System.</p> // markup element

0008: </section> // markup element

0009:

0010: <hr/> // markup element

0011:

0012: <section class="pt-3"> // markup element

0013: <div class="row g-4"> // markup element

0014: <div class="col-12"> // markup element

0015: <h2 id="info-we-collect" class="h5">Information We Collect</h2> // markup element

0016: <p class="small">We collect information to deliver insurance services, process claims, and secure your account.</p> // markup element

0017: <ul class="small"> // markup element

0018: <li><strong>Account details</strong>: name, email, phone, address, and authentication data.</li> // markup element

0019: <li><strong>Policy & claim data</strong>: policy numbers, coverage details, claim submissions, evidence you upload.</li> // markup element

0020: <li><strong>Payments</strong>: transaction identifiers and status. Card data is handled by our payment processor and is not stored on our servers.</li> // markup element

0021: <li><strong>Device & usage</strong>: IP address, browser, pages visited, and interactions to improve reliability and security.</li> // markup element

0022: <li><strong>Cookies</strong>: small files to keep you signed in and to remember preferences. See Cookies below.</li> // markup element

0023: </ul> // markup element

0024:

0025: <h2 id="how-we-use" class="h5 mt-4">How We Use Information</h2> // markup element

0026: <ul class="small"> // markup element

0027: <li>Provide, operate, and maintain policy and claims features.</li> // markup element

0028: <li>Authenticate users, prevent fraud, and secure the platform.</li> // markup element

0029: <li>Process payments and send transactional communications.</li> // markup element

0030: <li>Respond to support requests and improve our services.</li> // markup element

0031: <li>Comply with legal obligations.</li> // markup element

0032: </ul> // markup element

0033:

0034:

0035: <h2 id="sharing" class="h5 mt-4">Sharing & Disclosure</h2> // markup element

0036: <ul class="small"> // markup element

0037: <li><strong>Service providers</strong>: infrastructure, payments, communications, and analytics providers under contract.</li> // markup element

0038: <li><strong>Legal & compliance</strong>: to comply with law, enforce terms, or protect rights.</li> // markup element

0039: <li><strong>Business transfers</strong>: in connection with a merger, sale, or acquisition.</li> // markup element

0040: </ul> // markup element

0041:

0042:

0043: <h2 id="security" class="h5 mt-4">Security</h2> // markup element

0044: <p class="small">We use administrative, technical, and physical safeguards to protect information, including encryption in transit, access controls, and monitoring. No method of transmission or storage is completely secure; we work continuously to enhance our protections.</p> // markup element

0045:

0046: <h2 id="your-rights" class="h5 mt-4">Your Rights</h2> // markup element

0047: <p class="small">Depending on your location, you may have rights to:</p> // markup element

0048: <ul class="small"> // markup element

0049: <li>Access, correct, or delete your personal information.</li> // markup element

0050: <li>Object to or restrict certain processing.</li> // markup element

0051: <li>Port a copy of your information.</li> // markup element

0052: <li>Withdraw consent where processing is based on consent.</li> // markup element

0053: </ul> // markup element

0054: <p class="small">To exercise these rights, see <a href="#contact">Contact Us</a>. You may also have the right to lodge a complaint with your local data protection authority.</p> // markup element

0055:

0056: <h2 id="cookies" class="h5 mt-4">Cookies</h2> // markup element

0057: <p class="small">We use essential cookies for authentication and session management, and optional cookies to remember preferences and measure usage. You can control cookies in your browser settings; disabling essential cookies may limit functionality.</p> // markup element

0058:

0059:

0060: <h2 id="children" class="h5 mt-4">Children’s Privacy</h2> // markup element

0061: <p class="small">Our services are not directed to children under 13 (or the age required by your jurisdiction). We do not knowingly collect information from children. If you believe a child has provided personal information, please contact us and we will take appropriate steps.</p> // markup element

0062:

0063: <h2 id="changes" class="h5 mt-4">Changes to This Policy</h2> // markup element

0064: <p class="small">We may update this Policy from time to time. We will post the updated version and revise the date in the file history. Material changes will be communicated as required by law.</p> // markup element

0065:

0066: <h2 id="contact" class="h5 mt-4">Contact Us</h2> // markup element

0067: <p class="small mb-0"> // markup element

0068: Auto Insurance System<br/> // statement

0069: Email: <a href="mailto:privacy@autoinsurance.local">privacy@autoinsurance.local</a><br/> // statement

0070: Address: 123 Example Street, Suite 100, Your City, Your Country // statement

0071: </p> // markup element

0072: </div> // markup element

0073: </div> // markup element

0074: </section> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Shared/AccessDenied.cshtml

0001: @{ // Razor code block start

0002: ViewData["Title"] = "Access Denied"; // statement

0003: } // statement

0004:

0005: <div class="container py-5"> // markup element

0006: <h1 class="text-danger">Access Denied</h1> // markup element

0007: <p>You do not have permission to access this resource.</p> // markup element

0008: <p> // markup element

0009: <a class="btn btn-primary" href="@Url.Action("Landing", "Home")">Go to Home</a> // markup element

0010: </p> // markup element

0011: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Shared/Error.cshtml

0001: ﻿@model ErrorViewModel // statement

0002: @{ // Razor code block start

0003: ViewData["Title"] = "Error"; // statement

0004: } // statement

0005:

0006: <h1 class="text-danger">Error.</h1> // markup element

0007: <h2 class="text-danger">An error occurred while processing your request.</h2> // markup element

0008:

0009: @if (Model.ShowRequestId) // Razor conditional rendering

0010: { // statement

0011: <p> // markup element

0012: <strong>Request ID:</strong> <code>@Model.RequestId</code> // markup element

0013: </p> // markup element

0014: } // statement

0015:

0016: <h3>Development Mode</h3> // markup element

0017: <p> // markup element

0018: Swapping to <strong>Development</strong> environment will display more detailed information about the error that occurred. // statement

0019: </p> // markup element

0020: <p> // markup element

0021: <strong>The Development environment shouldn't be enabled for deployed applications.</strong> // markup element

0022: It can result in displaying sensitive information from exceptions to end users. // statement

0023: For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE\_ENVIRONMENT</strong> environment variable to <strong>Development</strong> // statement

0024: and restarting the app. // statement

0025: </p> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Shared/\_AdminStats.cshtml

0001: @{ // Razor code block start

0002: var activePolicies = ViewBag.ActivePolicies as int? ?? 0; // local variable declaration

0003: var pendingClaims = ViewBag.PendingClaims as int? ?? 0; // local variable declaration

0004: var revenueThisMonth = ViewBag.RevenueThisMonth as decimal? ?? 0m; // local variable declaration

0005: var openTickets = ViewBag.OpenTickets as int? ?? 0; // local variable declaration

0006: } // statement

0007:

0008: <div class="row g-3 mb-4"> // markup element

0009: <div class="col-md-3"> // markup element

0010: <div class="card shadow-sm border-0 stat-card h-100"> // markup element

0011: <div class="card-body"> // markup element

0012: <div class="d-flex align-items-center"> // markup element

0013: <div class="portal-icon bg-success-subtle text-success me-2">📄</div> // markup element

0014: <div> // markup element

0015: <div class="text-muted small">Active Policies</div> // markup element

0016: <div class="fs-2 fw-semibold">@activePolicies</div> // markup element

0017: </div> // markup element

0018: </div> // markup element

0019: </div> // markup element

0020: </div> // markup element

0021: </div> // markup element

0022: <div class="col-md-3"> // markup element

0023: <div class="card shadow-sm border-0 stat-card h-100"> // markup element

0024: <div class="card-body"> // markup element

0025: <div class="d-flex align-items-center"> // markup element

0026: <div class="portal-icon bg-warning-subtle text-warning me-2">⏳</div> // markup element

0027: <div> // markup element

0028: <div class="text-muted small">Pending Claims</div> // markup element

0029: <div class="fs-2 fw-semibold">@pendingClaims</div> // markup element

0030: </div> // markup element

0031: </div> // markup element

0032: </div> // markup element

0033: </div> // markup element

0034: </div> // markup element

0035: <div class="col-md-3"> // markup element

0036: <div class="card shadow-sm border-0 stat-card h-100"> // markup element

0037: <div class="card-body"> // markup element

0038: <div class="d-flex align-items-center"> // markup element

0039: <div class="portal-icon bg-info-subtle text-info me-2">💰</div> // markup element

0040: <div> // markup element

0041: <div class="text-muted small">Revenue This Month</div> // markup element

0042: <div class="fs-2 fw-semibold">@String.Format(System.Globalization.CultureInfo.GetCultureInfo("en-IN"), "₹{0:N2}", revenueThisMonth)</div> // markup element

0043: </div> // markup element

0044: </div> // markup element

0045: </div> // markup element

0046: </div> // markup element

0047: </div> // markup element

0048: <div class="col-md-3"> // markup element

0049: <div class="card shadow-sm border-0 stat-card h-100"> // markup element

0050: <div class="card-body"> // markup element

0051: <div class="d-flex align-items-center"> // markup element

0052: <div class="portal-icon bg-danger-subtle text-danger me-2">🎫</div> // markup element

0053: <div> // markup element

0054: <div class="text-muted small">Open Tickets</div> // markup element

0055: <div class="fs-2 fw-semibold">@openTickets</div> // markup element

0056: </div> // markup element

0057: </div> // markup element

0058: </div> // markup element

0059: </div> // markup element

0060: </div> // markup element

0061: </div> // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/Shared/\_Layout.cshtml

0001: ﻿ <!DOCTYPE html> // statement

0002: <html lang="en"> // markup element

0003: <head> // markup element

0004: <meta charset="utf-8" /> // markup element

0005: <meta name="viewport" content="width=device-width, initial-scale=1.0" /> // markup element

0006: <title>@(ViewData["Title"] ?? "Auto Insurance System") - Auto\_Insurance\_System</title> // markup element

0007: <script type="importmap"></script> // markup element

0008: <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" /> // markup element

0009: <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" integrity="sha384-9Gz6TgM8LzJY0w7L9a4m0mQh1m7mUQ4pE2Fz9qj6cQq7k6OQF3yXl5C4jD9v9V8e" crossorigin="anonymous"> // markup element

0010: <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" /> // markup element

0011: <link rel="stylesheet" href="~/Auto\_Insurance\_System.styles.css" asp-append-version="true" /> // markup element

0012: </head> // markup element

0013: <body class="@((ViewBag.BodyClass as string) ?? "")"> // markup element

0014: <header class="shadow-sm"> // markup element

0015: <nav class="navbar navbar-expand-lg navbar-dark bg-app-primary"> // markup element

0016: <div class="container-fluid"> // markup element

0017: @{ // Razor code block start

0018: var role = Context.Session.GetString("UserRole"); // local variable declaration

0019: var userName = Context.Session.GetString("UserName"); // local variable declaration

0020: var brandTitle = "Auto Insurance System"; // local variable declaration

0021: if (!string.IsNullOrEmpty(role)) // conditional check

0022: { // statement

0023: brandTitle = role switch // statement

0024: { // statement

0025: "ADMIN" => "Admin Dashboard · Auto Insurance System", // statement

0026: "AGENT" => "Agent Dashboard · Auto Insurance System", // statement

0027: "CUSTOMER" => "Customer Dashboard · Auto Insurance System", // statement

0028: \_ => "Auto Insurance System" // statement

0029: }; // statement

0030: } // statement

0031: var currentController = ViewContext.RouteData.Values["controller"]?.ToString(); // local variable declaration

0032: var currentAction = ViewContext.RouteData.Values["action"]?.ToString(); // local variable declaration

0033: var isFullWidth = (ViewBag.FullWidth as bool?) ?? false; // local variable declaration

0034: var noTopPadding = (ViewBag.NoTopPadding as bool?) ?? false; // local variable declaration

0035: } // statement

0036:

0037: <a class="navbar-brand fw-semibold d-flex align-items-center" asp-area="" asp-controller="Home" asp-action="Landing"> // markup element

0038: <span class="badge rounded-circle bg-light text-app-primary me-2">AI</span> // markup element

0039: <span>@brandTitle</span> // markup element

0040: </a> // markup element

0041:

0042: <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#mainNavbar" aria-controls="mainNavbar" aria-expanded="false" aria-label="Toggle navigation"> // markup element

0043: <span class="navbar-toggler-icon"></span> // markup element

0044: </button> // markup element

0045:

0046: <div class="collapse navbar-collapse" id="mainNavbar"> // markup element

0047: <ul class="navbar-nav me-auto mb-2 mb-lg-0"> // markup element

0048: <li class="nav-item"><a class="nav-link" asp-controller="Home" asp-action="Landing">Home</a></li> // markup element

0049: <li class="nav-item"><a class="nav-link" asp-controller="Home" asp-action="About">About</a></li> // markup element

0050: <li class="nav-item"><a class="nav-link" asp-controller="Home" asp-action="Privacy">Privacy</a></li> // markup element

0051: </ul> // markup element

0052:

0053: <ul class="navbar-nav ms-auto align-items-lg-center"> // markup element

0054: @if (!string.IsNullOrEmpty(role)) // Razor conditional rendering

0055: { // statement

0056: <li class="nav-item dropdown"> // markup element

0057: <a class="nav-link dropdown-toggle d-flex align-items-center" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false"> // markup element

0058: <span class="icon-circle bg-light text-dark me-2">👤</span> // markup element

0059: <span class="text-capitalize">@role.ToLower()</span> // markup element

0060: </a> // markup element

0061: <ul class="dropdown-menu dropdown-menu-end"> // markup element

0062: <li class="dropdown-item-text"><small class="text-muted">Signed in as</small><div class="fw-semibold">@userName</div></li> // markup element

0063: <li><hr class="dropdown-divider" /></li> // markup element

0064: <li><a class="dropdown-item" asp-controller="Auth" asp-action="Logout">Logout</a></li> // markup element

0065: </ul> // markup element

0066: </li> // markup element

0067: } // statement

0068: else // statement

0069: { // statement

0070: <li class="nav-item"> // markup element

0071: <a class="btn btn-outline-light btn-sm" asp-controller="Auth" asp-action="Login">Login</a> // markup element

0072: </li> // markup element

0073: } // statement

0074: </ul> // markup element

0075: </div> // markup element

0076: </div> // markup element

0077: </nav> // markup element

0078:

0079: @\* @if (Context.Session.GetString("UserRole") == "ADMIN") // comment

0080: { // statement

0081: <div class="bg-light-subtle border-bottom"> // markup element

0082: <div class="container py-2"> // markup element

0083: <ul class="nav gap-3 align-items-center"> // markup element

0084: <li class="nav-item"> // markup element

0085: <a class="nav-link d-flex align-items-center" asp-controller="Admin" asp-action="UsersHub"> // markup element

0086: <i class="bi bi-people me-1"></i> Users // statement

0087: </a> // markup element

0088: </li> // markup element

0089: <li class="nav-item"> // markup element

0090: <a class="nav-link d-flex align-items-center" asp-controller="Admin" asp-action="ClaimsHub"> // markup element

0091: <i class="bi bi-clipboard-check me-1"></i> Claims // statement

0092: </a> // markup element

0093: </li> // markup element

0094: <li class="nav-item"> // markup element

0095: <a class="nav-link d-flex align-items-center" asp-controller="Admin" asp-action="PoliciesHub"> // markup element

0096: <i class="bi bi-file-earmark-text me-1"></i> Policies // statement

0097: </a> // markup element

0098: </li> // markup element

0099: <li class="nav-item"> // markup element

0100: <a class="nav-link d-flex align-items-center" asp-controller="Admin" asp-action="PaymentsHub"> // markup element

0101: <i class="bi bi-credit-card me-1"></i> Payments // statement

0102: </a> // markup element

0103: </li> // markup element

0104: <li class="nav-item"> // markup element

0105: <a class="nav-link d-flex align-items-center" asp-controller="Admin" asp-action="TicketsHub"> // markup element

0106: <i class="bi bi-life-preserver me-1"></i> Tickets // statement

0107: </a> // markup element

0108: </li> // markup element

0109: </ul> // markup element

0110: </div> // markup element

0111: </div> // markup element

0112: } \*@ // statement

0113: </header> // markup element

0114:

0115: <div class="@(isFullWidth ? "container-fluid px-0" : "container") @(noTopPadding ? "py-0" : "py-4")"> // markup element

0116: <main id="mainContent" role="main"> // markup element

0117: @RenderBody() // statement

0118: </main> // markup element

0119: </div> // markup element

0120:

0121: <footer class="border-top footer text-muted bg-white-50 footer-compact"> // markup element

0122: <div class="container-sm d-flex justify-content-between align-items-center py-2"> // markup element

0123: <span>&copy; @DateTime.UtcNow.Year - Auto Insurance System</span> // markup element

0124: <a asp-area="" asp-controller="Home" asp-action="Privacy" class="text-decoration-none">Privacy</a> // markup element

0125: </div> // markup element

0126: </footer> // markup element

0127: <script src="~/lib/jquery/dist/jquery.min.js"></script> // markup element

0128: <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script> // markup element

0129: <script src="~/js/site.js" asp-append-version="true"></script> // markup element

0130: @await RenderSectionAsync("Scripts", required: false) // statement

0131: </body> // markup element

0132: </html> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/Shared/\_ValidationScriptsPartial.cshtml

0001: ﻿<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script> // statement

0002: <script src="~/lib/jquery-validation-unobtrusive/dist/jquery.validate.unobtrusive.min.js"></script> // markup element

# Auto-Insurance/Auto\_Insurance\_System/Views/\_ViewImports.cshtml

0001: ﻿@using Auto\_Insurance\_System // statement

0002: @using Auto\_Insurance\_System.Models // statement

0003: @addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers // statement

# Auto-Insurance/Auto\_Insurance\_System/Views/\_ViewStart.cshtml

0001: ﻿@{ // statement

0002: Layout = "\_Layout"; // statement

0003: } // statement

# Auto-Insurance/Auto\_Insurance\_System/appsettings.Development.json

0001: { // statement

0002: "Logging": { // statement

0003: "LogLevel": { // statement

0004: "Default": "Information", // statement

0005: "Microsoft.AspNetCore": "Warning" // statement

0006: } // statement

0007: } // statement

0008: } // statement

# Auto-Insurance/Auto\_Insurance\_System/appsettings.json

0001: { // statement

0002: "Logging": { // statement

0003: "LogLevel": { // statement

0004: "Default": "Information", // statement

0005: "Microsoft.AspNetCore": "Warning" // statement

0006: } // statement

0007: }, // statement

0008: "ConnectionStrings": { // statement

0009: "SqlCon": "Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=AutoInsuranceSystemManagementDB;Integrated Security=True;TrustServerCertificate=True" // statement

0010: }, // statement

0011: "AllowedHosts": "\*" // statement

0012: } // statement

# Auto-Insurance/Auto\_Insurance\_System\_Soln.sln

0001: ﻿ // statement

0002: Microsoft Visual Studio Solution File, Format Version 12.00 // statement

0003: # Visual Studio Version 17 // statement

0004: VisualStudioVersion = 17.14.36202.13 // statement

0005: MinimumVisualStudioVersion = 10.0.40219.1 // statement

0006: Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Auto\_Insurance\_System", "Auto\_Insurance\_System\Auto\_Insurance\_System.csproj", "{5B6A4D34-72D0-4AAF-9B1A-63E3D5DBB14A}" // statement

0007: EndProject // statement

0008: Project("{2150E333-8FDC-42A3-9474-1A3956D46DE8}") = "Solution Items", "Solution Items", "{8EC462FD-D22E-90A8-E5CE-7E832BA40C5D}" // statement

0009: ProjectSection(SolutionItems) = preProject // statement

0010: SqlQueryPro.sql = SqlQueryPro.sql // statement

0011: SqlQuery\_2.sql = SqlQuery\_2.sql // statement

0012: EndProjectSection // statement

0013: EndProject // statement

0014: Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "AutoInsuranceSystemAPI", "AutoInsuranceSystemAPI\AutoInsuranceSystemAPI.csproj", "{5B566C93-5B54-48EC-8300-F94DDC380F40}" // statement

0015: EndProject // statement

0016: Global // statement

0017: GlobalSection(SolutionConfigurationPlatforms) = preSolution // statement

0018: Debug|Any CPU = Debug|Any CPU // statement

0019: Release|Any CPU = Release|Any CPU // statement

0020: EndGlobalSection // statement

0021: GlobalSection(ProjectConfigurationPlatforms) = postSolution // statement

0022: {5B6A4D34-72D0-4AAF-9B1A-63E3D5DBB14A}.Debug|Any CPU.ActiveCfg = Debug|Any CPU // statement

0023: {5B6A4D34-72D0-4AAF-9B1A-63E3D5DBB14A}.Debug|Any CPU.Build.0 = Debug|Any CPU // statement

0024: {5B6A4D34-72D0-4AAF-9B1A-63E3D5DBB14A}.Release|Any CPU.ActiveCfg = Release|Any CPU // statement

0025: {5B6A4D34-72D0-4AAF-9B1A-63E3D5DBB14A}.Release|Any CPU.Build.0 = Release|Any CPU // statement

0026: {5B566C93-5B54-48EC-8300-F94DDC380F40}.Debug|Any CPU.ActiveCfg = Debug|Any CPU // statement

0027: {5B566C93-5B54-48EC-8300-F94DDC380F40}.Debug|Any CPU.Build.0 = Debug|Any CPU // statement

0028: {5B566C93-5B54-48EC-8300-F94DDC380F40}.Release|Any CPU.ActiveCfg = Release|Any CPU // statement

0029: {5B566C93-5B54-48EC-8300-F94DDC380F40}.Release|Any CPU.Build.0 = Release|Any CPU // statement

0030: EndGlobalSection // statement

0031: GlobalSection(SolutionProperties) = preSolution // statement

0032: HideSolutionNode = FALSE // statement

0033: EndGlobalSection // statement

0034: GlobalSection(ExtensibilityGlobals) = postSolution // statement

0035: SolutionGuid = {04DF0C42-2F71-4313-A90B-405E2E42A8D8} // statement

0036: EndGlobalSection // statement

0037: EndGlobal // statement

# Auto-Insurance/SqlQueryPro.sql

0001: ﻿create database AutoInsuranceSystemManagementDB; // statement

0002: use AutoInsuranceSystemManagementDB // statement

# Auto-Insurance/SqlQuery\_2.sql

0001: ﻿ // statement

# Application Working Flow Summary

1) A user hits a URL; routing (Program.cs) maps to {controller}/{action}/{id?}.

2) The Controller action runs; model binding fills parameters/models from the request.

3) The Controller validates input (ModelState), then calls a DI-injected Service.

4) The Service uses EF Core via AutoInsuranceDbContext (or API project DbContext) to query/update DbSet<T>.

5) For Customer policy creation, MVC calls the Admin API (HttpClient) which persists via AutoInsuranceApiDbContext.

6) The Service returns domain models/booleans; the Controller decides to render a View, redirect, or return JSON.

7) Razor Views render HTML with Bootstrap styling; \_Layout.cshtml provides shared nav and layout.

8) Session-based RoleAuthorizationFilter checks Session["UserRole"] to guard role-specific controllers/actions.

9) The response (HTML/JSON/redirect) is sent back to the browser.