INSUREGO GENERAL INSURANCE

Team: *Keerthana and Hymavathi*   *- Backend(Web API)*
       Kanishka and Keerthi   *- Frontend(MVC and Views)*
       Mehraj Fathima   *- Database and Testing*

INSUREGO GENERAL INSURANCE

A User-Friendly Online Platform for Motor and Travel Insurance, addresses these issues by providing a secure, user-friendly online platform for direct customer interaction, enabling easy buying, renewing, estimating, and claiming motor and travel insurance policies. This responsive web application promotes cost savings, real-time status tracking, and digital verification to enhance customer experience in the finance domain.

PROJECT OVERVIEW

Topic: General Insurance

Domain: Finance

Project Name: InsureGO Insurance

Description: An online web application for buying, renewing, estimating, and claiming motor and travel insurance policies.

Target Users: Customers seeking easy, secure, and responsive insurance services without intermediaries.

TECHNOLOGIES USED

- **Fronted(View layer)**
  Razor View Engine(.cshtml)
  css(for styling)

- **Backend**
  ASP.Net MVC - Handles request routing controllers, and business logic
  ASP.NET Web API - used for data communication and api services

- **Databases**
  Microsoft SQL Server- Store application data

- **Data Access**
  Entity Framework - DB FirstApproach

- **Tools**
  Visual Studio -IDE
  GitHub - Source code management

FLOW CHART
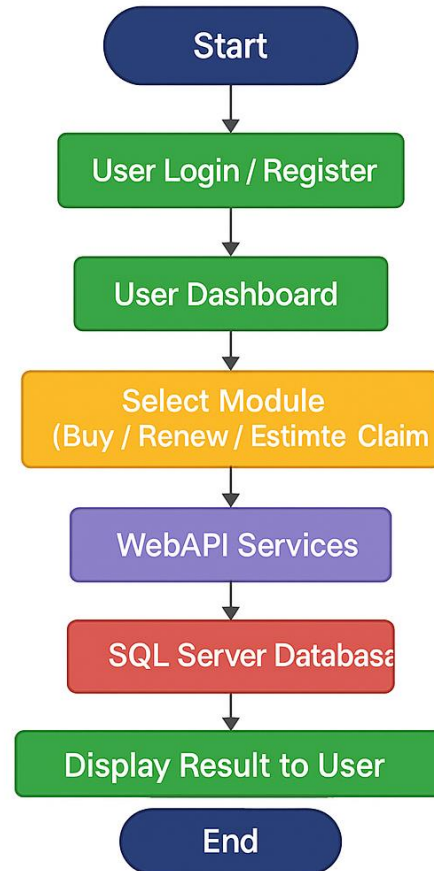
USER AUTHENTICATION AND ADMIN FEATURES

Description: Secure login/register for users; admin dashboard for policy/claim management with role-based access.

Key Features:

- **Login:** Email/password validation → personalized dashboard.

- **Register:** User details input → secure account creation.

- **Admin Page:** Manage policies, users, claims, reports.
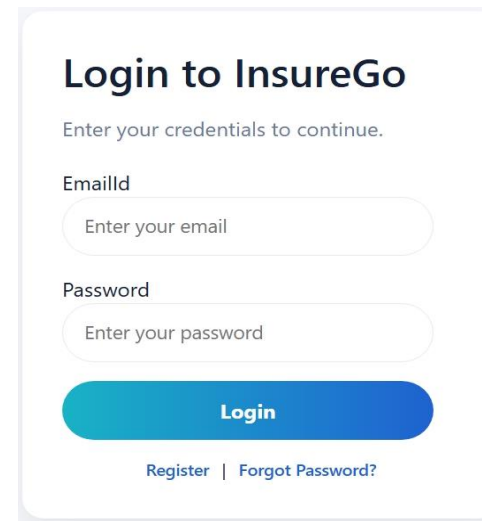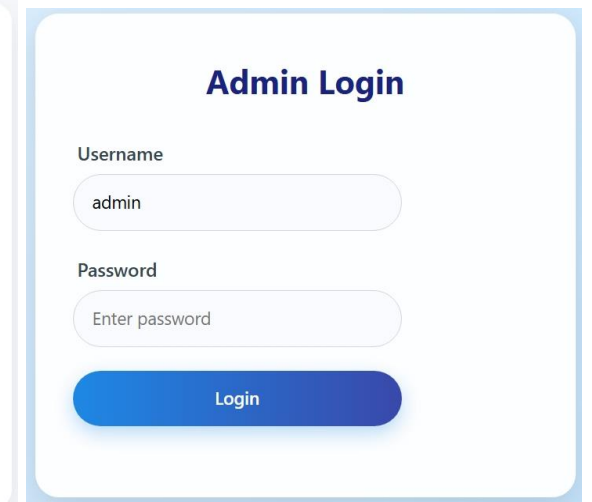
MANDATORY MODULES - OVERVIEW

Mandatory Modules

Four key modules:
Buy Insurance
Renew Insurance
Calculate Estimate
Claim Insurance

MODULE 1 - BUY INSURANCE

- User selects insurance type, Fills required information (e.g., vehicle details).

- Makes secure payment to Insurance.

- Receives policy number and instant coverage.

- Main Procedures/Methods: Controller method (e.g., BuyInsuranceController) validates user input and calls WebAPI for policy creation.

- Model classes store data (e.g., PolicyModel for details).

- View renders forms; API integrates with payment gateway for transactions.
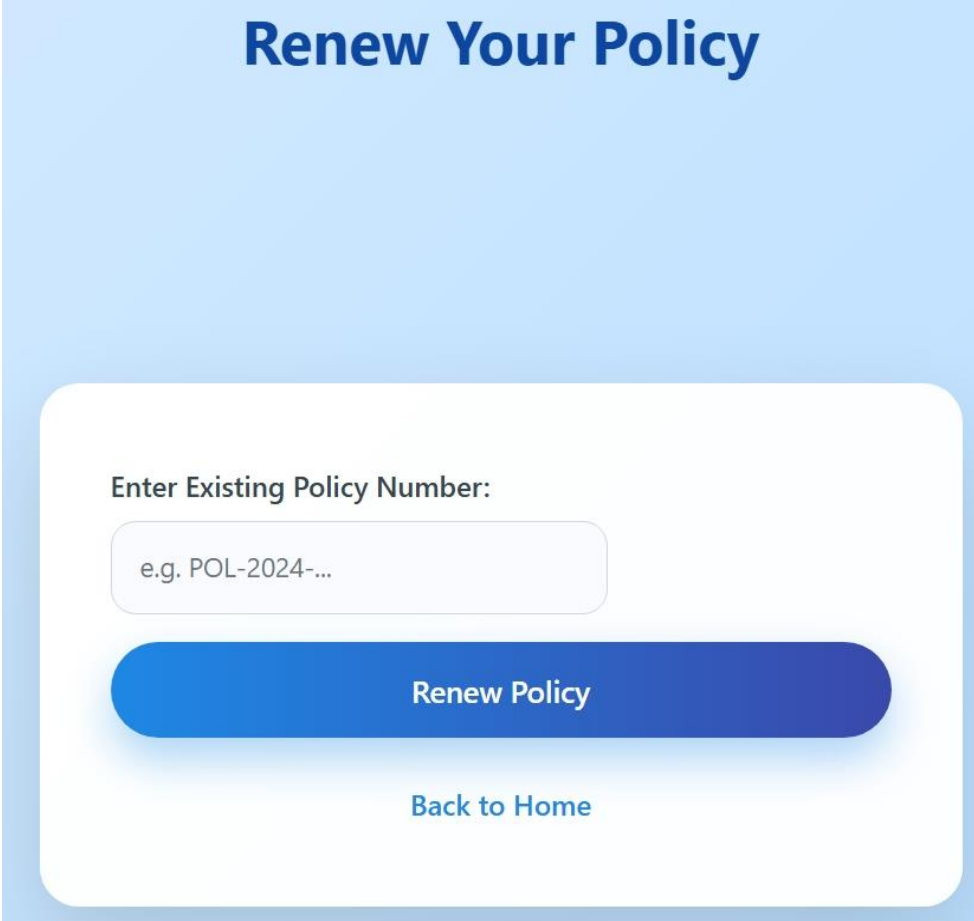
MODULE 2 - RENEW INSURANCE

- User clicks "Renew Insurance" link.

- Enters previous policy number.

- Makes payment based on policy value.

- Policy is renewed automatically.

- Main Procedures/Methods: Controller method (e.g., RenewInsuranceController) fetches existing policy via WebAPI and calculates renewal cost.

- Model updates policy expiry date.

- View displays renewal form; API handles payment and database updates for motor policies only.

Renew Your Policy

Enter Existing Policy Number:

e.g. POL-2024-...

Renew Policy

Back to Home

MODULE 3 - CALCULATE INSURANCE ESTIMATE

- User selects insurance type (motor or travel).

- Enters details (e.g., vehicle specs or trip info).

- Generates instant price estimate.

- Main Procedures/Methods: Controller method (e.g., EstimateController) processes inputs and calls WebAPI for pricing logic (e.g., based on risk factors).

- Model computes estimate without saving data.

- View shows calculator interface; API queries database for rate tables.

MODULE 4 - CLAIM INSURANCE

- User initiates claim process (details not fully specified in abstract).

- Assumed steps: Submit claim details, upload documents digitally, track status.

- Main Procedures/Methods: Controller method (e.g., ClaimInsuranceController) validates claim and uploads documents via WebAPI.

- Model stores claim data and status.

- View provides claim form and tracking dashboard; API integrates with verification services and updates database.

Claim History - POL20260128220432

Claim No	Date	Status	Amount
31	28-01-2026	Approved	₹ 7,000.00

← Back to Home

TESTING

- **Testing Approach:** Ensures reliability, security, and functionality of the application.
- **Types of Testing Used:**
  1. <u>Unit Testing:</u> Tests individual components (e.g., controllers and models) using tools like NUnit.

THANK YOU