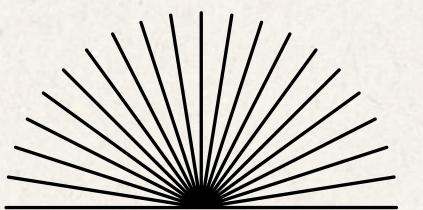


SMART HEALTH INSURANCE MANAGEMENT SYSTEM

PRESENTED BY:
Anuska Maity



Agenda

03	Overview
04	Problem Statement
05	Solution and Key Features
06	Technology Stack
07	System Architecture
08	User Management Screen
09	Analytics and Report
10	Results and Achievement

Overview

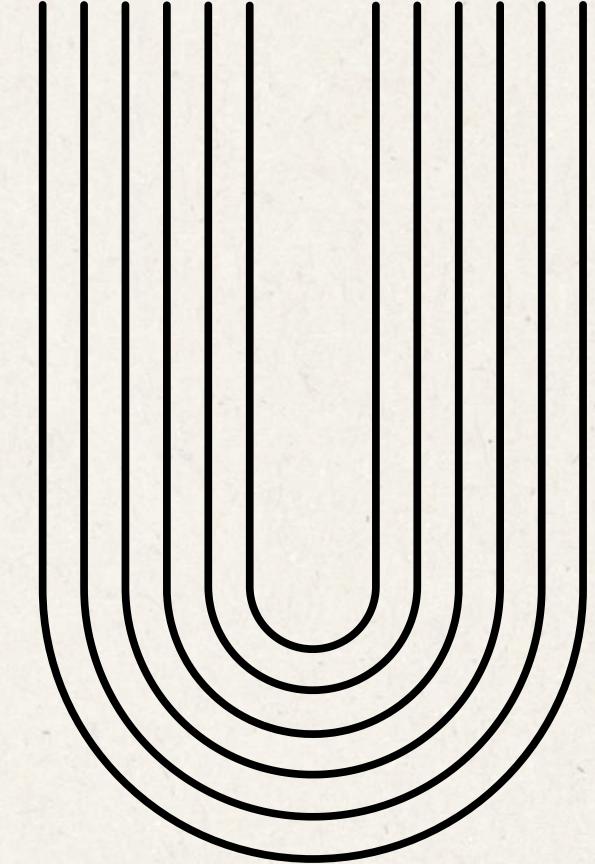
This is end-to-end process flow of the Smart Health Insurance Management System. It explains how different users interact with the system and how data flows from policy creation to claim settlement.

The system is designed as a full-stack web application using ASP.NET Core Web API (backend) and Angular (frontend), with secure role-based access.

01 Technology Stack:
.NET Core, Angular,
SQL Server

02 Duration:
9 days

Problem Statement



The existing health insurance management process is largely manual, fragmented, and inefficient, leading to delays in policy management, claim processing, and poor coordination between customers, hospitals, and insurance staff. There is a need for a secure, centralized, and automated system that can manage insurance policies, claims, payments, and reporting in a seamless manner.

1 Policy Management Problem

2 Claim Processing Problem

3 Hospital & Provider Coordination Problem

4 Security & Access Control Problem

Solution And Key Features

AUTOMATE INSURANCE PLAN MANAGEMENT, POLICY ISSUANCE, AND END-TO-END CLAIM PROCESSING. IT CENTRALIZES USER AND ROLE CONTROL THROUGH AN ADMIN-DRIVEN WORKFLOW WHILE SECURELY MANAGING THE COMPLETE CLAIM LIFECYCLE FROM SUBMISSION TO APPROVAL OR REJECTION.

1. User & Role Management	2. Security & Access Control
3. Insurance Plan Management	4. Policy Management
5. Claims Management	6. Hospital & Claims Officer Workflow

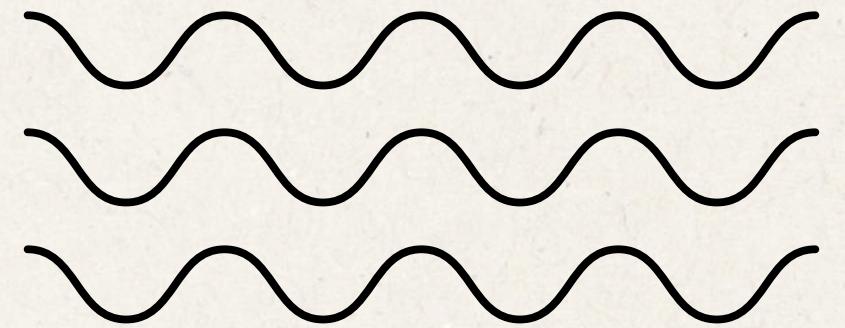
Backend #1

- ASP.NET Core Web API implementing business logic and workflows
- Handles authentication, role management, policies, and claims processing
- Secure REST APIs with JWT-based authorization

06/10

Frontend #2

- Angular application with standalone components and modular architecture
- Role-based dashboards with Angular Material UI
- Implements guards, interceptors, and reactive forms



Technology Stack

Roles #3

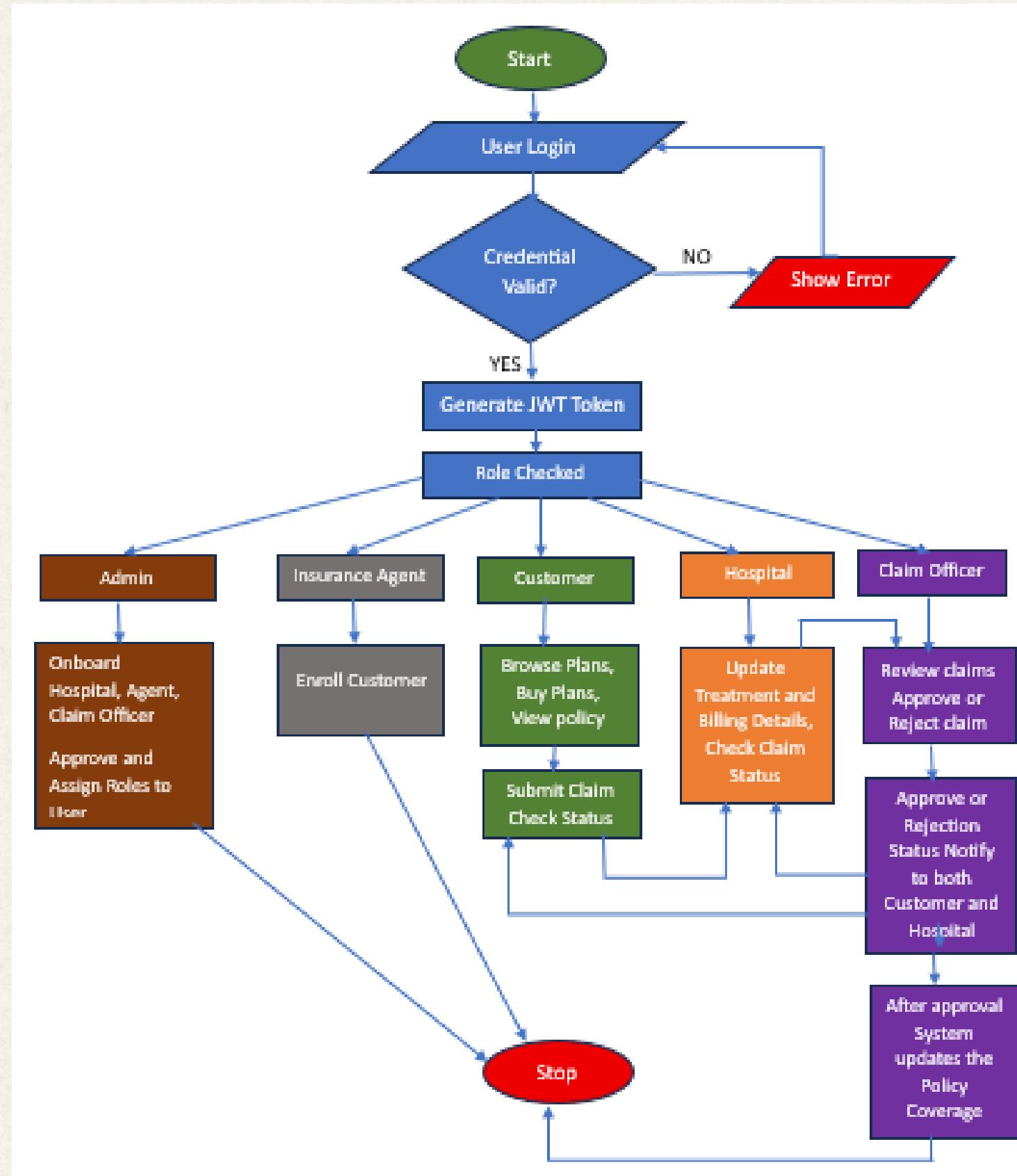
- Admin, Customer, Hospital Provider, Claims Officer, Insurance Agent
- Role-based access control ensures secure and restricted functionality
- Each role has a dedicated dashboard and permissions

Database #4

- SQL Server with Entity Framework Core
- Structured schema for users, roles, policies, plans, and claims
- Ensures data integrity, relationships, and auditability

System Architecture

07/10



User Management Screen

08/10

- Admin-only interface for managing all system users.
- Displays registered users with username, email, registration date, status, and role.
- Supports approval of newly registered users.
- Enables role assignment after approval.
- Prevents role modification once assigned (Role Locked).
- Allows filtering to view pending users only.
- Ensures controlled and secure access to system dashboards.

The screenshot shows the 'User Approval' section of an admin interface. On the left is a dark sidebar with navigation links: Dashboard, User Approval (which is highlighted), Insurance Plans, Hospitals, Agents, and Claims Officer. The main area is titled 'User Approval' and contains a sub-instruction 'Approve users first, then assign their access role'. A 'Show Pending Only' button is located in the top right corner of this title bar. The central part of the screen is a table with the following columns: Username, Email, Registered On, Status, Role, and Actions. There are 12 rows of data, each representing a user account. The 'Status' column uses color-coded buttons: green for APPROVED and red for PENDING. The 'Actions' column includes a dropdown menu labeled 'Select role' and an 'Assign' button for approved users, and an 'Approve' button for pending users. Some rows show 'Role Locked' status next to the role name. The table has horizontal and vertical scroll bars on the right and bottom respectively.

Username	Email	Registered On	Status	Role	Actions
priya	anuska09.maity@gmail.com	Jan 4, 2026, 9:50:00 PM	APPROVED	-	Select role Assign
officer3	22053227@kiit.ac.in	Jan 5, 2026, 9:20:51 PM	PENDING	-	Approve
officer2	anuska09.maity@gmail.com	Jan 4, 2026, 10:30:35 PM	APPROVED	ClaimsOfficer	Role Locked
rani	rani@gmail.com	Jan 4, 2026, 9:55:19 PM	APPROVED	Customer	Role Locked
puja	puja@gmail.com	Jan 4, 2026, 9:53:10 PM	APPROVED	-	Select role Assign
peerless	test@gmail.com	Jan 4, 2026, 10:22:43 PM	APPROVED	HospitalProvider	Role Locked
agent100	22053227@kiit.ac.in	Jan 5, 2026, 9:49:37 PM	PENDING	-	Approve
officer4	anuska09.maity@gmail.com	Jan 5, 2026, 9:27:56 PM	PENDING	-	Approve
ritika	anuska09.maity@gmail.com	Jan 4, 2026, 9:51:54 PM	PENDING	-	Approve
apollo	test@gmail.com	Jan 6, 2026, 4:30:53 AM	APPROVED	HospitalProvider	Role Locked
urmi	rani@gmail.com	Jan 6, 2026, 4:27:14 AM	APPROVED	Customer	Role Locked
diya	rani@gmail.com	Jan 5, 2026, 9:03:00 PM	APPROVED	Customer	Role Locked

Analytics and Reports

09/10

Chart 1: Plan-wise Claim Count

Compares number of active policies vs claims across insurance plans.

Helps identify plans with higher claim ratios.

Supports decision-making and fraud/risk evaluation.

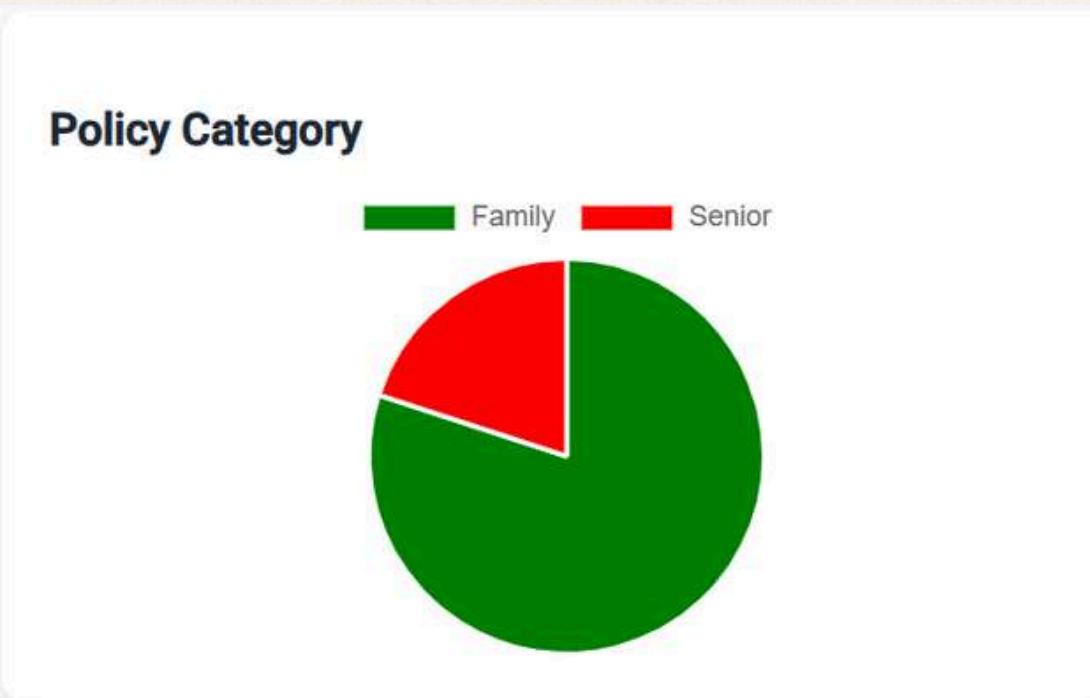


Chart 2: Policy Category Distribution

Visual representation of policy categories (Family, Senior).

Shows customer enrollment trends.

Assists in understanding demographic risk exposure.



Results & Achievements

- Successfully developed a fully functional role-based health insurance management system.
- Automated the complete insurance and claim lifecycle, reducing manual intervention.
- Implemented secure authentication and authorization for five distinct user roles.
- Enabled end-to-end claim tracking from submission to approval or rejection.
- Improved operational efficiency through dashboard-based workflows.
- Integrated analytics and reports to support data-driven claim decisions.
- Ensured transparency and accountability across all stakeholders.
- Built a scalable and modular architecture suitable for real-world deployment.

LIVE DEMONSTRATION

SEE THE APPLICATION IN ACTION