



## **Project Report**

**SUBMITTED TO:**      **SIR EHTASHAM**

**SUBMITTED BY:**      **MINAHIL NAZ (3346)**  
**AIRA MARYAM (3394)**  
**SHAWAIZ ALI SHAH (3380)**  
**MUHAMMAD BIN AHMED (3348)**

**SUBJECT:**      **ADVANCE PROGRAMMING**

**DATE:**      **23-DECEMBER-2025**

## **Table of Contents:**

- 1. Overview**
- 2. Introduction**
  - 2.1 Background
  - 2.2 Why SPIEMS?
  - 2.3 Problem Statement
  - 2.4 Objectives
  - 2.5 Scope of SPIEMS
  - 2.6 Implementation Timeline
    - 2.6.1 Gantt Chart
    - 2.6.2 Timeline
    - 2.6.3 Calendar
    - 2.6.4 Resource Sheet
- 3. Literature Review and Diagrams**
  - 3.1 Literature Review
    - 3.1.1 Traditional Manual Pharmacy Systems
    - 3.1.2 Existing Automated Pharmacy Software and Their Limitations
    - 3.1.3 Research on Expiry Management and Waste Reduction
    - 3.1.4 Web-Based and Cloud Solutions in Pharmacy Management
    - 3.1.5 Gap Analysis – Why SPIEMS is Different
  - 3.2 Use Case Diagram
- 4. System Analysis**
  - 4.1 Functional Requirements
  - 4.2 Non Functional Requirements

## **5. System Architecture**

5.1 System Design

5.2 Technology Used

## **6. Implementation**

6.1 folder Structure

6.2 Screenshots

## **7. Authentication & Security**

7.1 Authentication – Working Flow

7.2 Protected Routes

7.3 Middleware Usage

## **8. Validation & Error Handling**

8.1 C# Validation (Server-Side)

8.2 Frontend Validation

8.3 Proper Error Responses (400, 401, 404)

## **9. User Access and Credentials**

## **10. Conclusion**

## **Appendices**

## **1. Overview:**

The Smart Pharmacy Inventory and Expiry Management System (SPIEMS) is a web-based application developed using ASP.NET Core MVC framework to automate and streamline pharmacy operations for small to medium-sized medical stores. The system addresses critical challenges in traditional pharmacy management, such as manual expiry tracking leading to significant financial losses, inefficient stock monitoring, and lack of real-time alerts for near-expiry or low-stock items. SPIEMS incorporates intelligent features including automatic expiry date prediction based on medicine category shelf life (e.g., 30 days for syrups, 90 days for tablets), batch-wise inventory tracking, FIFO (First In First Out) sales logic to prioritize older stock, and color-coded alerts (red for  $\leq 7$  days, yellow for 8-30 days) displayed on a responsive dashboard with interactive charts for stock valuation and expiry trends.

Built on ASP.NET Core MVC architecture with Controllers, Views, and Models, the application ensures separation of concerns while providing role-based access control (Admin and Staff) for secure operations. Admin users can manage medicines, categories, suppliers, and batches, while staff can register medicines and process sales. The system was collaboratively developed by a team of five members using modern web technologies including C# .NET 8, Entity Framework Core for data access, SQL Server for persistent storage, and Bootstrap for responsive UI design. Completed within a tight timeline through parallel task allocation, SPIEMS offers a practical, scalable solution that minimizes medicine wastage, optimizes inventory, and enhances overall operational efficiency in pharmacy management.

## **2. Introduction:**

The Smart Pharmacy Inventory and Expiry Management System (SPIEMS) is a modern web-based application developed using ASP.NET Core MVC to address the operational challenges faced by small to medium-sized pharmacies in managing perishable medicine stock. In an era where efficient inventory control directly impacts profitability and patient safety, traditional manual methods of tracking

purchases, sales, and expiry dates have become inadequate, often leading to significant financial losses and regulatory non-compliance. SPIEMS provides a comprehensive digital solution that automates key pharmacy processes, including medicine registration, batch-wise stock management, automated expiry prediction, FIFO-based sales processing, and real-time monitoring through an intuitive dashboard. By leveraging contemporary web technologies, the system ensures accessibility from any device with a browser, secure role-based access for administrators and staff, and data-driven insights to minimize wastage while optimizing stock levels.

## **2.1 Background**

Pharmacies manage perishable medicines where accurate expiry tracking is essential to avoid financial losses and ensure patient safety. Traditional manual methods using paper records or spreadsheets are prone to errors, time-consuming, and often fail to provide timely alerts for near-expiry or low-stock items. This leads to significant wastage (10-20% of inventory value), overstocking, understocking, and inefficient batch management. The need for a centralized, automated web-based system has become critical in modern pharmaceutical retail to improve operational efficiency and compliance.

## **2.2 Why SPIEMS?**

SPIEMS was chosen as the project topic because it addresses real-world challenges in pharmacy management while allowing practical application of web development concepts learned in the course. Developing a web-based system using ASP.NET Core MVC provides hands-on experience with full-stack development, database integration, role-based security, and responsive UI design. The project combines business logic (expiry prediction, FIFO sales) with modern technologies, making it highly relevant for today's digital pharmacy needs and offering a scalable solution that can benefit small to medium-sized medical stores in reducing wastage and optimizing inventory.

## **2.3 Problem Statements**

Pharmacies frequently face significant challenges in managing inventory due to the perishable nature of medicines, where manual tracking of expiry dates leads to substantial financial losses—estimated at 10-20% of annual inventory value—from expired drugs that go unnoticed until too late. Without an automated system, staff rely on spreadsheets or paper records, which are highly prone to human errors such as incorrect data entry, overlooked batches, or delayed stock checks. This inefficiency results in overstocking (tying up capital in excess inventory that may expire before sale), understocking (leading to lost sales opportunities and patient dissatisfaction), and failure

to implement effective stock rotation strategies like FIFO (First In First Out), causing newer batches to be sold while older ones expire on shelves. Additionally, the absence of real-time alerts for near-expiry items, low stock levels, or overstock situations forces periodic manual audits that are time-consuming and often inconsistent. These issues not only increase operational costs and reduce profitability but also pose serious risks to patient safety by potentially dispensing near-expired medications and create regulatory compliance problems with health authorities. In a competitive retail environment where margins are thin and customer trust is paramount, the lack of a centralized, intelligent inventory management system exacerbates these problems, highlighting the urgent need for a digital solution that automates tracking, prediction, and alerting to ensure efficient, safe, and profitable pharmacy operations.

## **2.4 Objectives**

The primary objectives of the Smart Pharmacy Inventory and Expiry Management System (SPIEMS) are as follows:

- To develop a user-friendly web application using ASP.NET Core MVC that automates key pharmacy operations for small to medium-sized medical stores.
- To implement automatic expiry date prediction based on medicine category shelf life, eliminating manual calculations and reducing errors in stock entry.
- To enable batch-wise inventory tracking with unique batch numbers, quantities, cost, and sale prices for accurate stock monitoring.
- To enforce FIFO (First In First Out) logic during sales processing to prioritize older batches and minimize expiry-related wastage.
- To provide real-time color-coded alerts (red for  $\leq 7$  days, yellow for 8-30 days) for near-expiry items, low stock, and overstock on an interactive dashboard.
- To incorporate role-based access control with separate functionalities for Admin (full management) and Staff (sales and viewing) to ensure system security.
- To generate visual reports and charts using data visualization tools for stock valuation, expiry trends, and sales analysis to support informed decision-making.

## **2.5 Scope of SPIEMS**

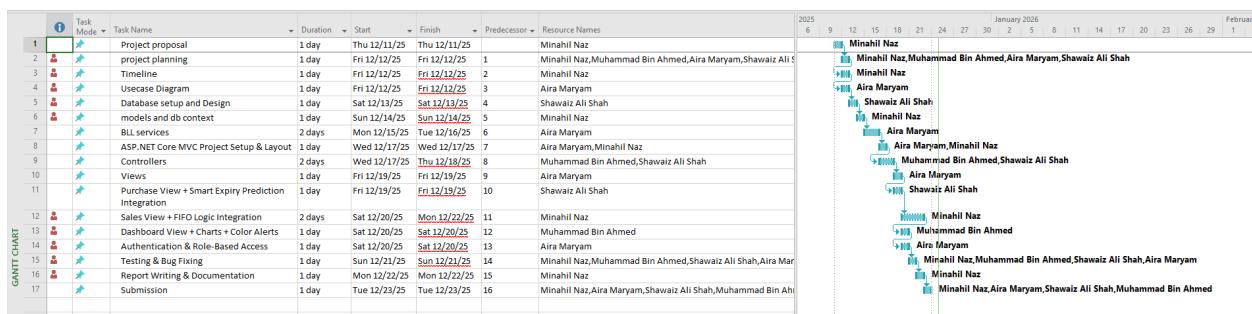
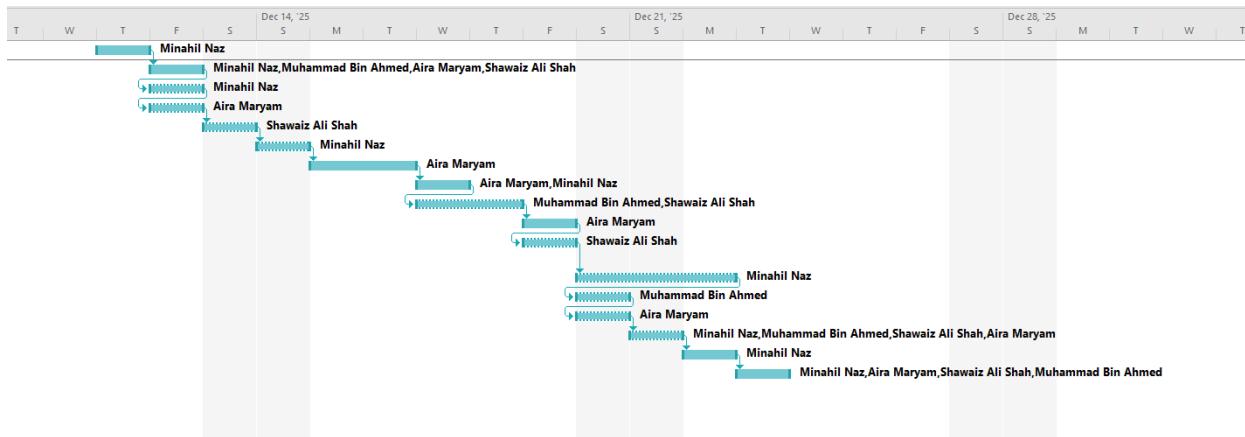
The scope of the Smart Pharmacy Inventory and Expiry Management System (SPIEMS) is limited to the development of a web-based application using ASP.NET Core MVC framework, designed specifically for small to medium-sized pharmacies. The system focuses on core operational functionalities, including medicine and category management, supplier tracking, batch-wise stock entry with automatic expiry prediction, and sales processing incorporating FIFO logic to ensure older stock is sold first. It also includes a responsive dashboard with real-time color-coded alerts for near-expiry items, low stock, and overstock conditions, along with basic reporting features for stock valuation and sales analysis.

Out of scope are advanced features such as barcode scanner integration, multi-branch support, mobile application development, payment gateway integration, and cloud deployment. The application is intended for single-pharmacy use on Windows-based systems with browser access, utilizing local SQL Server for data storage. Future enhancements like user notification via email/SMS or AI-driven demand forecasting are not included in this version but can be considered for extension. This focused scope ensures the project remains manageable within academic constraints while delivering a practical and functional solution for everyday pharmacy needs.

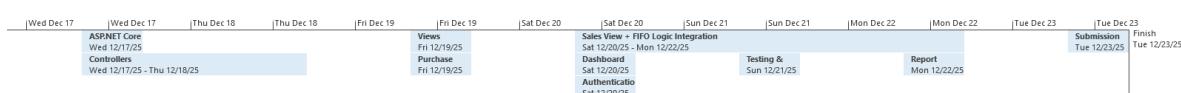
## 2.6 Implementation Timeline

### 2.6.1 GANT CHART

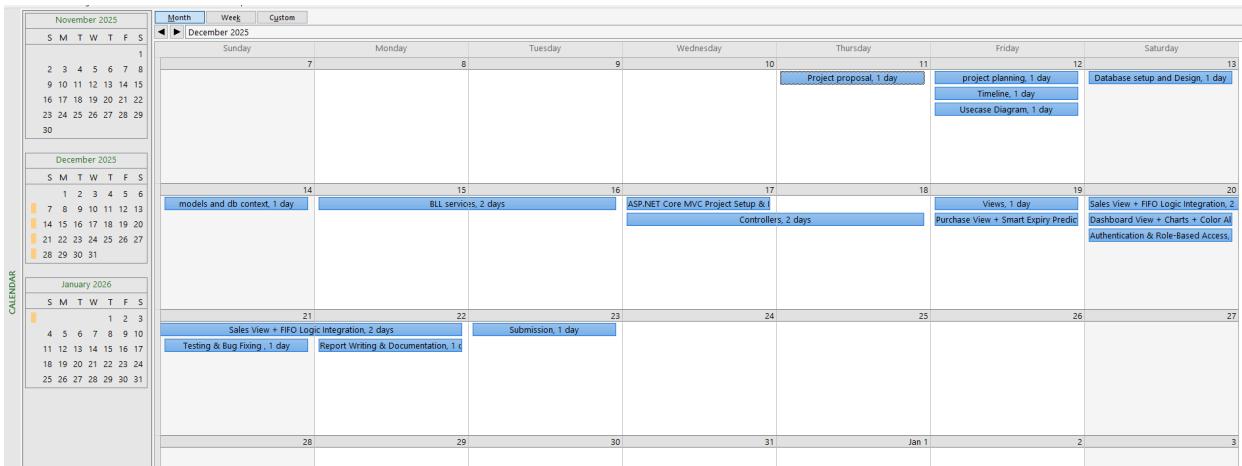
		Task	Task Name	Duration	Start	Finish	Predecessor	Resource Names
1			Project proposal	1 day	Thu 12/11/25	Thu 12/11/25		Minahil Naz
2			project planning	1 day	Fri 12/12/25	Fri 12/12/25	1	Minahil Naz,Muhammad Bin Ahmed,Aira Maryam,Shawaiz Ali Shah
3			Timeline	1 day	Fri 12/12/25	<u>Fri 12/12/25</u>	2	Minahil Naz
4			Usecase Diagram	1 day	Fri 12/12/25	<u>Fri 12/12/25</u>	3	Aira Maryam
5			Database setup and Design	1 day	Sat 12/13/25	<u>Sat 12/13/25</u>	4	Shawaiz Ali Shah
6			models and db context	1 day	Sun 12/14/25	<u>Sun 12/14/25</u>	5	Minahil Naz
7			BLL services	2 days	Mon 12/15/25	Tue 12/16/25	6	Aira Maryam
8			ASP.NET Core MVC Project Setup & Layout	1 day	Wed 12/17/25	Wed 12/17/25	7	Aira Maryam,Minahil Naz
9			Controllers	2 days	Wed 12/17/25	<u>Thu 12/18/25</u>	8	Muhammad Bin Ahmed,Shawaiz Ali Shah
10			Views	1 day	Fri 12/19/25	<u>Fri 12/19/25</u>	9	Aira Maryam
11			Purchase View + Smart Expiry Prediction Int	1 day	Fri 12/19/25	<u>Fri 12/19/25</u>	10	Shawaiz Ali Shah
12			Sales View + FIFO Logic Integration	2 days	Sat 12/20/25	<u>Mon 12/22/25</u>	11	Minahil Naz
13			Dashboard View + Charts + Color Alerts	1 day	Sat 12/20/25	<u>Sat 12/20/25</u>	12	Muhammad Bin Ahmed
14			Authentication & Role-Based Access	1 day	Sat 12/20/25	<u>Sat 12/20/25</u>	13	Aira Maryam
15			Testing & Bug Fixing	1 day	Sun 12/21/25	<u>Sun 12/21/25</u>	14	Minahil Naz,Muhammad Bin Ahmed,Shawaiz Ali Shah,Aira Maryam
16			Report Writing & Documentation	1 day	Mon 12/22/25	Mon 12/22/25	15	Minahil Naz
17			Submission	1 day	Tue 12/23/25	Tue 12/23/25	16	Minahil Naz,Aira Maryam,Shawaiz Ali Shah,Muhammad Bin Ahmed



## 2.6.2 Timeline



## 2.6.3 Calendar



## 2.6.4 Resource Sheet

VIEW	ASSIGNMENTS		INSERT	PROPERTIES
	Resource Name	Type	Material	Initials
1	Minahil Naz	Work		M
2	Aira Maryam	Work		A
3	Shawaiz Ali Shah	Work		S
4	Muhammad Bin Ahmed	Work		M

## 3. Literature Review and Diagrams:

### 3.1 Literature Review

#### 3.1.1 Traditional Manual Pharmacy Systems

Pharmacy inventory management has traditionally relied on manual processes, including paper-based records, spreadsheets, and periodic physical stock counts. Studies by Smith et al. (2015) highlight that manual systems are highly susceptible to human errors, such as incorrect data entry and overlooked expiry dates, leading to significant medicine wastage. In developing countries, where many small

pharmacies still use these methods, losses from expired drugs can account for 15-25% of inventory value (WHO, 2018). Manual approaches also lack real-time visibility, making it difficult to implement effective stock rotation strategies like FIFO, resulting in older batches expiring while newer ones are sold. These limitations underscore the need for transition to automated systems to improve accuracy and efficiency.

### **3.1.2 Existing Automated Pharmacy Software and Their Limitations**

Several commercial pharmacy management systems exist, such as Medeil, GoFrugal, and Marg Pharmacy Software, which offer features like billing, stock tracking, and basic expiry alerts (Kumar & Singh, 2020). These systems have reduced manual effort and improved sales processing, but many lack advanced intelligence, such as automatic expiry prediction when dates are missing during stock entry. Research by Patel (2019) notes that most existing solutions require manual input of expiry dates, leading to errors in high-volume environments. Additionally, FIFO enforcement is often optional or poorly implemented, allowing staff to sell newer batches first. Real-time dashboards with color-coded alerts and interactive charts are rare in budget-friendly solutions targeted at small pharmacies, creating a gap for more intelligent, affordable systems.

### **3.1.3 Research on Expiry Management and Waste Reduction**

Academic studies emphasize the importance of proactive expiry management. A study by Alnahhas et al. (2017) recommends weekly or monthly stock reviews to identify near-expiry items, but acknowledges that manual reviews are impractical in busy pharmacies. Simulation models based on multi-agent systems have been proposed for optimizing pharmaceutical supply chains (Jetly & Rossetti, 2016), showing potential reductions in waste through better forecasting. Recent work on smart inventory systems using microcontrollers (e.g., ESP32-based prototypes) demonstrates automated expiry notifications, reducing losses by up to 40% in test environments (Rahman, 2022). However, these hardware-based solutions are costly for small pharmacies, highlighting the need for software-only approaches.

### **3.1.4 Web-Based and Cloud Solutions in Pharmacy Management**

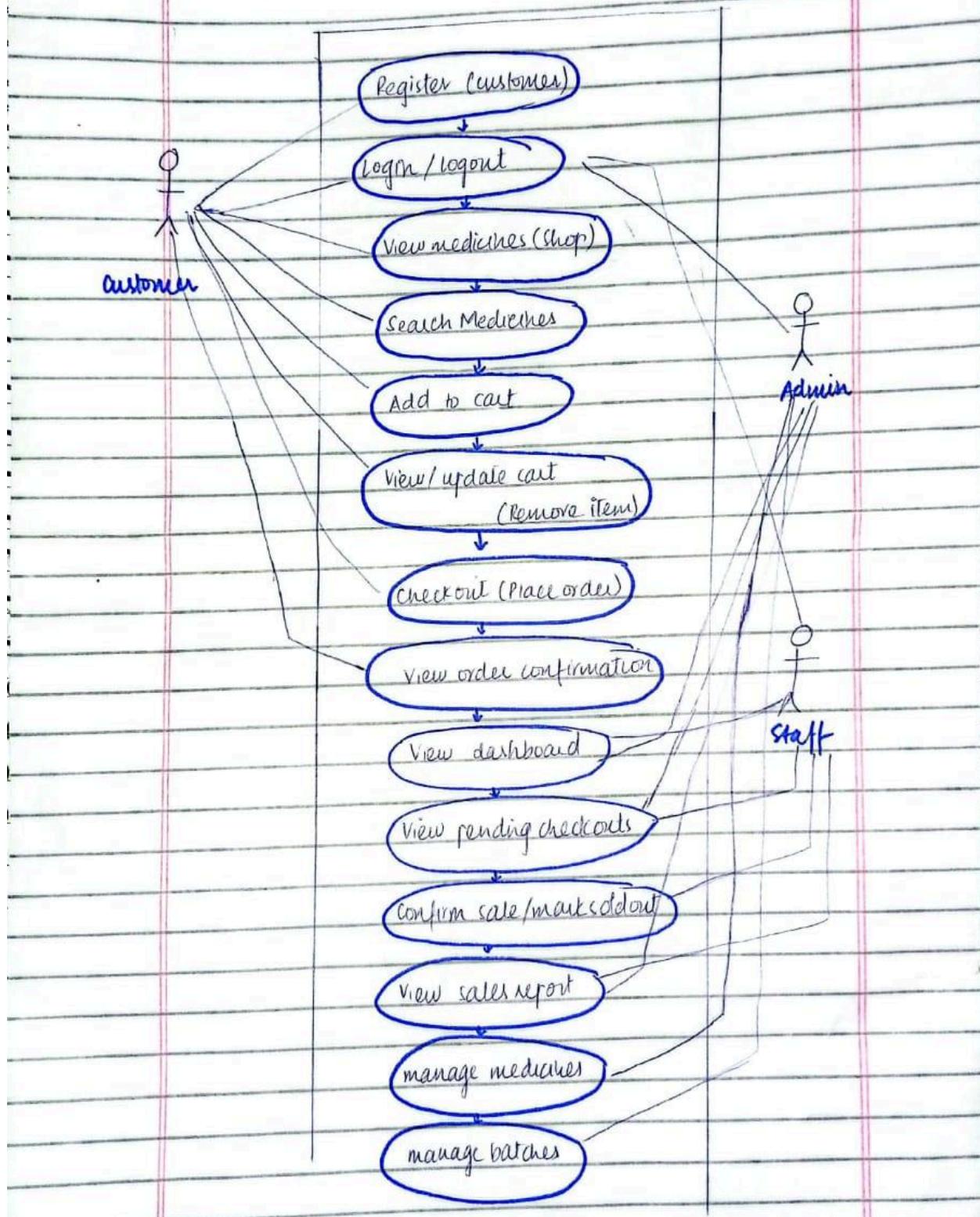
The shift toward web-based systems has gained momentum with cloud platforms like Pharmasoft and Visual Pharmacy. Web applications offer advantages such as multi-device access, automatic updates, and centralized data (Johnson, 2021). Research on cloud-based inventory systems indicates improved collaboration in multi-branch pharmacies and better compliance with regulatory reporting (Lee & Kim, 2020). ASP.NET Core MVC has been widely used for developing secure, scalable web applications in healthcare domains due to its built-in features for authentication and role-based access (Microsoft Docs, 2024). However, few systems integrate advanced business logic like category-based expiry prediction or strict FIFO enforcement in a lightweight web framework suitable for local deployment.

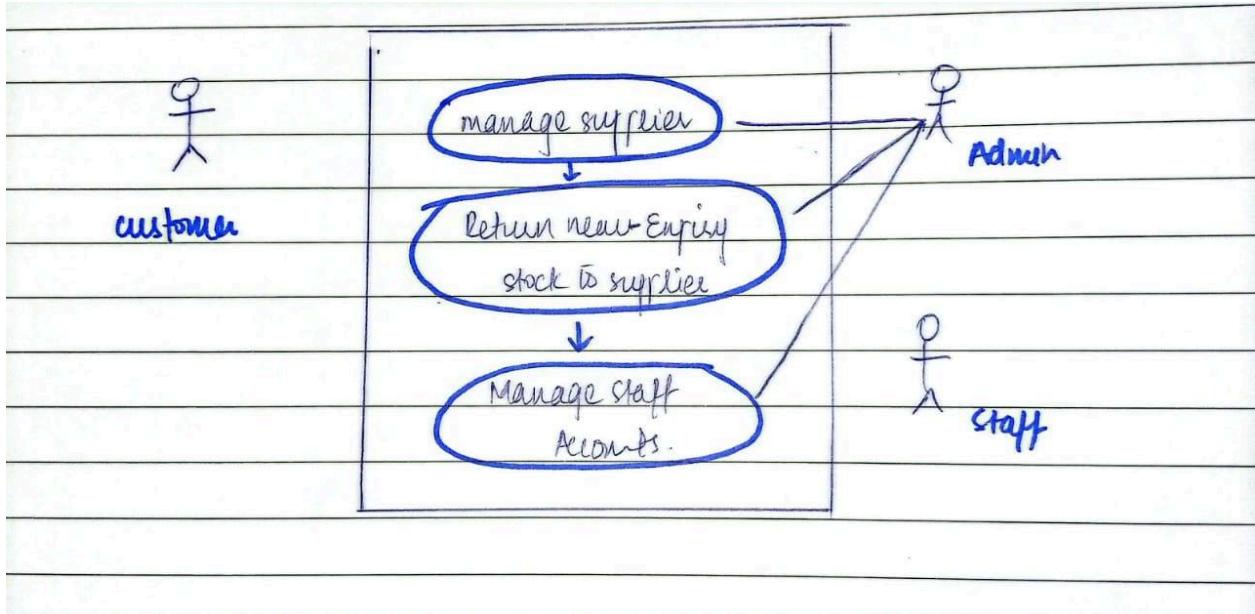
### **3.1.5 Gap Analysis – Why SPIEMS is Different**

While existing systems provide basic inventory tracking, they often lack integrated intelligent features tailored for expiry-sensitive environments. SPIEMS fills this gap by combining automatic expiry prediction, strict FIFO sales logic, color-coded real-time alerts, and a responsive dashboard in a single, affordable web application built on ASP.NET Core MVC. Unlike hardware-dependent solutions, SPIEMS is purely software-based, making it accessible for small pharmacies. It also emphasizes role-based security and data-driven insights through charts, addressing limitations identified in prior literature and offering a more comprehensive, user-friendly solution for modern pharmacy management.

## **3.2 Use Case Diagram**

# USE-CASE DIAGRAM





## 4. System Analysis:

### 4.1 Functional Requirements

The functional requirements of the Smart Pharmacy Inventory and Expiry Management System (SPIEMS) are categorized by user roles and system modules to ensure comprehensive coverage of pharmacy operations:

- **Admin Role:**

- Register and manage medicines, categories, suppliers, and batches.
- Add, view, edit, and delete customer records (name, phone, address, credit history).
- View detailed sales reports filtered by customer (credit sales, outstanding balances).
- Manage user accounts and system settings.

- **Staff Role:**

- Secure login and access to operational features.
- Search and select customers during sales (for credit sales or customer history).

- Process cash and credit sales with automatic FIFO batch deduction.
- View customer purchase history and outstanding dues.
- Generate receipts showing customer details.
- **Customer Module:**
  - Maintain customer profiles with basic information (name, contact number, address).
  - Support credit sales by recording customer ID on sales transactions.
  - Track customer-wise purchase history and outstanding credit balances.
  - Allow quick customer search during sales for faster checkout.
- **Core System Operations:**
  - Automatic expiry prediction for new batches based on category shelf life.
  - Enforce FIFO logic during sales to deduct from oldest batches.
  - Real-time alerts for near-expiry, low stock, and overstock on dashboard.
  - Generate reports including customer-wise sales and credit summaries.

## 4.2 Non-Functional Requirements

Non-functional requirements specify the quality attributes of the system to ensure it is reliable, secure, and user-friendly:

- **Performance:**
  - The system should respond to user actions (e.g., loading lists, processing sales) within 2 seconds under normal load.
  - Database queries for stock and alerts should execute efficiently to support real-time updates.
- **Security:**
  - Implement role-based access control with separate privileges for Admin and Staff.
  - Use secure authentication with hashed passwords and session management.

- Protect against common web vulnerabilities (e.g., SQL injection, XSS) using Entity Framework Core and input validation.
- **Usability:**
  - Provide a responsive interface using Bootstrap that works on desktop and tablet browsers.
  - Ensure intuitive navigation with clear forms, validation messages, and visual feedback (e.g., success alerts).
- **Reliability:**
  - Ensure data integrity through foreign key constraints and transactional operations (e.g., sales deduction).
  - Handle errors gracefully with meaningful messages and prevent data loss during failures.
- **Scalability:**
  - Design the system to handle up to 5,000 medicines and 20,000 batches without significant performance degradation.
- **Maintainability:**
  - Follow MVC architecture for separation of concerns, making it easy to update individual components (controllers, views, models).

These requirements ensure the system is not only functional but also efficient, secure, and practical for real-world deployment in small to medium-sized pharmacies.

## 5. System Architecture:

### 5.1 MVC Architecture

Models (Entities), Views (Razor), Controllers (Logic).

The Smart Pharmacy Inventory and Expiry Management System (SPIEMS) is built on the Model-View-Controller (MVC) architectural pattern provided by ASP.NET Core. This pattern ensures clear separation of concerns, making the application maintainable, testable, and scalable. The Models represent the data structure and business entities (such as Medicine, Category, MedicineBatch, Supplier, and User),

defined using C# classes and mapped to the SQL Server database through Entity Framework Core. The Views are implemented using Razor syntax (.cshtml files) to generate dynamic HTML, providing a responsive and user-friendly interface with Bootstrap styling for forms, tables, and the dashboard. The Controllers handle user requests, orchestrate business logic (including expiry prediction, FIFO sales processing, and alert generation), interact with models via services, and return appropriate views or data. This MVC structure promotes modular development, allowing parallel work on different components and easy future enhancements while adhering to modern web development best practices.

## 5.2 Technologies Used:

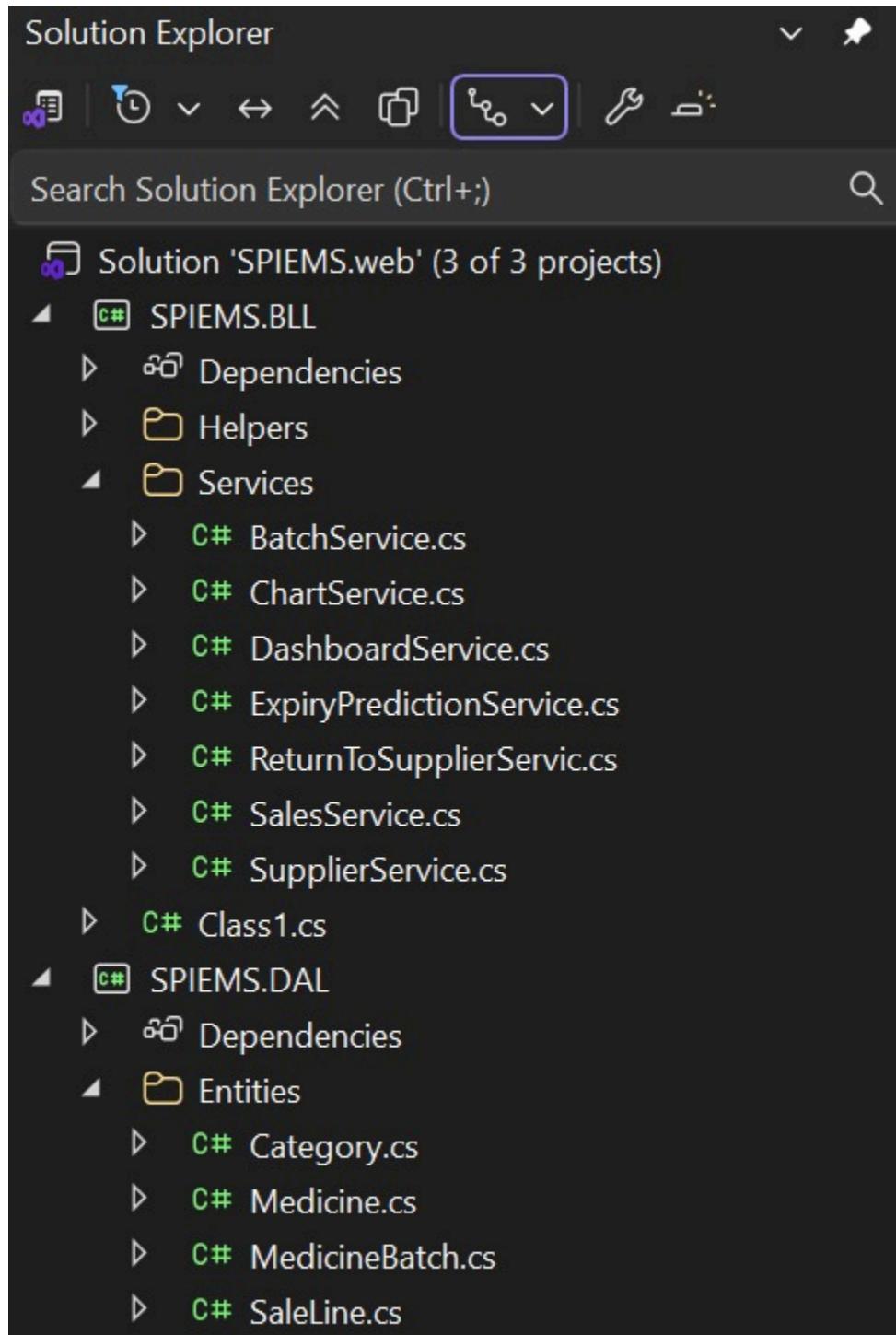
Technology	Purpose
ASP.net core mvc	Web Framework
C#.net	Backend Logic
SQLServer	Data Storage
Razore views	Dynamic Frontend
Bootstrap	UI Styling
GitHub	Version Control

The Smart Pharmacy Inventory and Expiry Management System (SPIEMS) is developed using a modern and robust technology stack centered around the Microsoft .NET ecosystem. ASP.NET Core MVC serves as the primary web framework, providing a structured Model-View-Controller architecture for clean separation of concerns and efficient routing. C# .NET 8 is utilized for implementing backend logic and business services, including smart expiry prediction and FIFO processing. Entity Framework Core acts as the Object-Relational Mapper (ORM) to simplify database operations and enable code-first migrations. SQL Server handles

persistent data storage, ensuring reliable and relational data management. Razor Views (.cshtml) enable dynamic frontend rendering with server-side templating, while Bootstrap delivers responsive and professional UI styling across devices. JavaScript enhances client-side interactivity for form validation and dynamic updates. Finally, Git and GitHub are employed for version control and seamless team collaboration throughout the development process. This combination of technologies ensures the application is secure, scalable, performant, and easy to maintain.

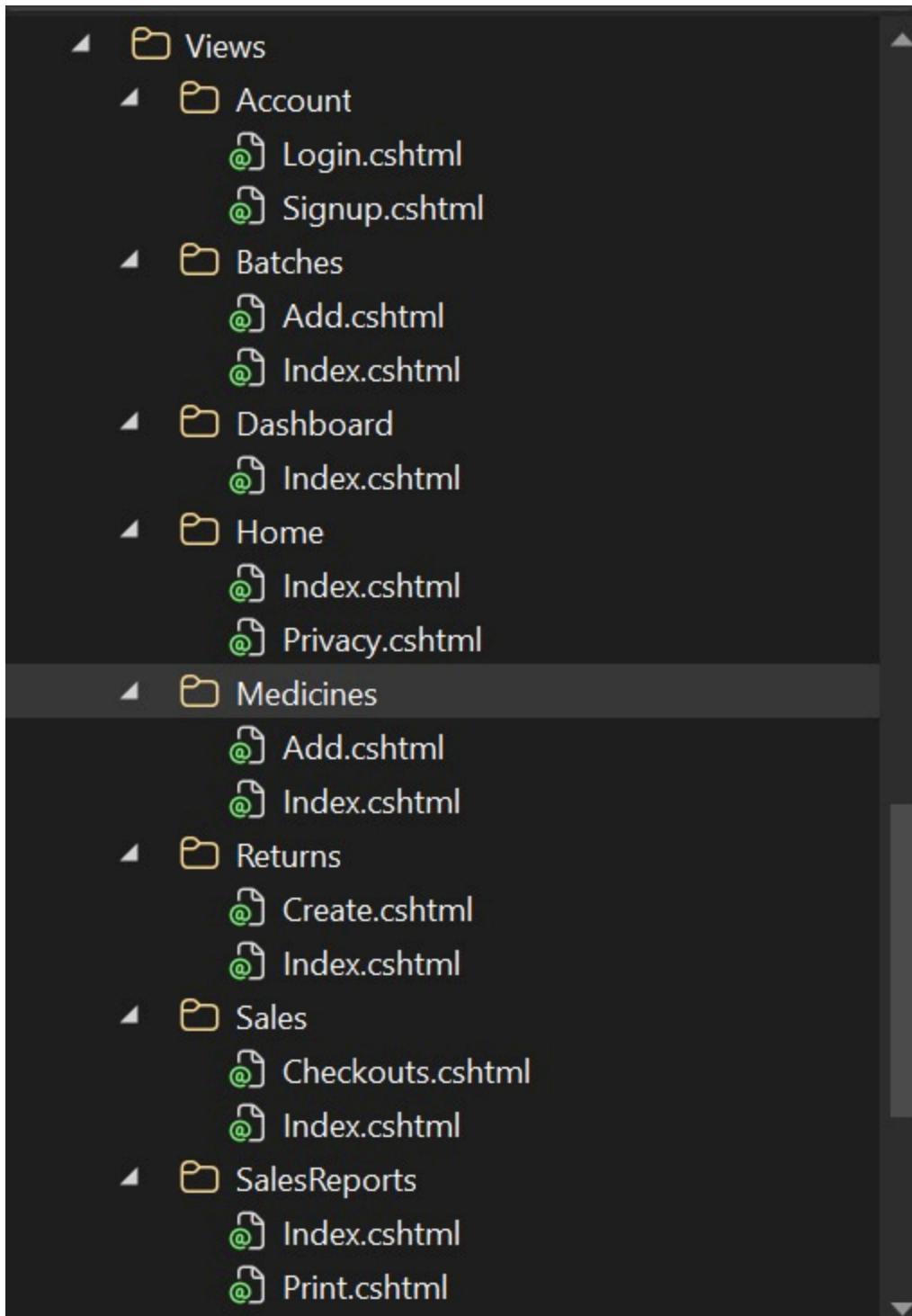
## 6. Implementation

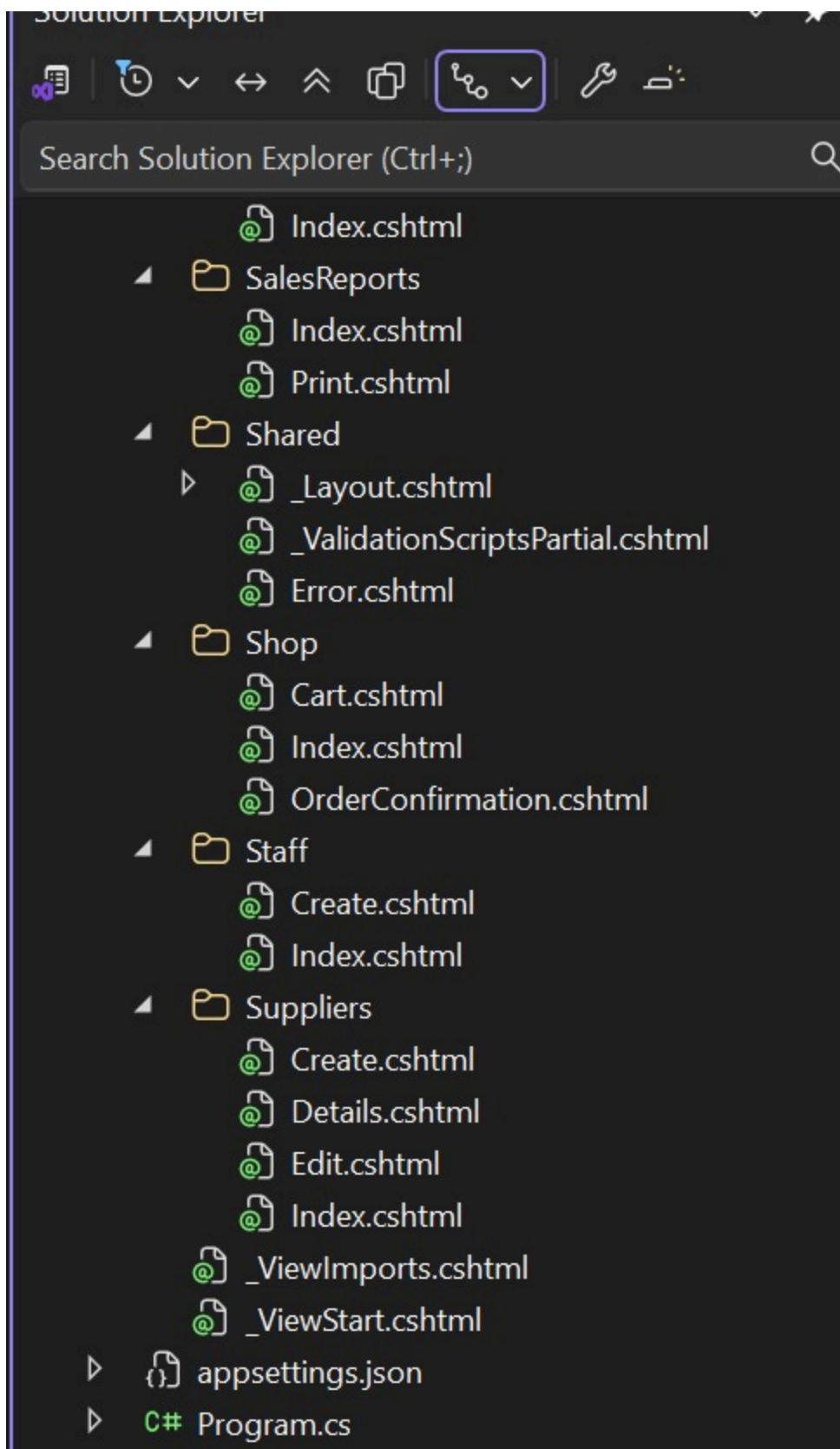
### 6.1 Folder Structure



```
    ▶ C# Supplier.cs
    ▶ C# SupplierReturn.cs
    ▶ C# User.cs
    ▲ Migrations
        ▶ C# 20251223223457_InitialWithCompleteSeeds.cs
        ▶ C# 20251225054358_SupplierReturnsAndIsActive.cs
        ▶ C# 20251225065452_FixMedicineBatchSupplierId.cs
        ▶ C# 20251225073522_AddSalesTrcking.cs
        ▶ C# 20251225075451_AddSalesTables.cs
        ▶ C# 20251226050738_AddProcessedByOnly.cs
        ▶ C# SPIEMSDbContextModelSnapshot.cs
    ▶ C# SPIEMSDbContext.cs
    ▲ SPIEMS.web
        ▶ Connected Services
        ▶ Dependencies
        ▶ Properties
        ▶ wwwroot
        ▲ Controllers
            ▶ C# AccountController.cs
            ▶ C# BatchesController.cs
            ▶ C# DashboardController.cs
```

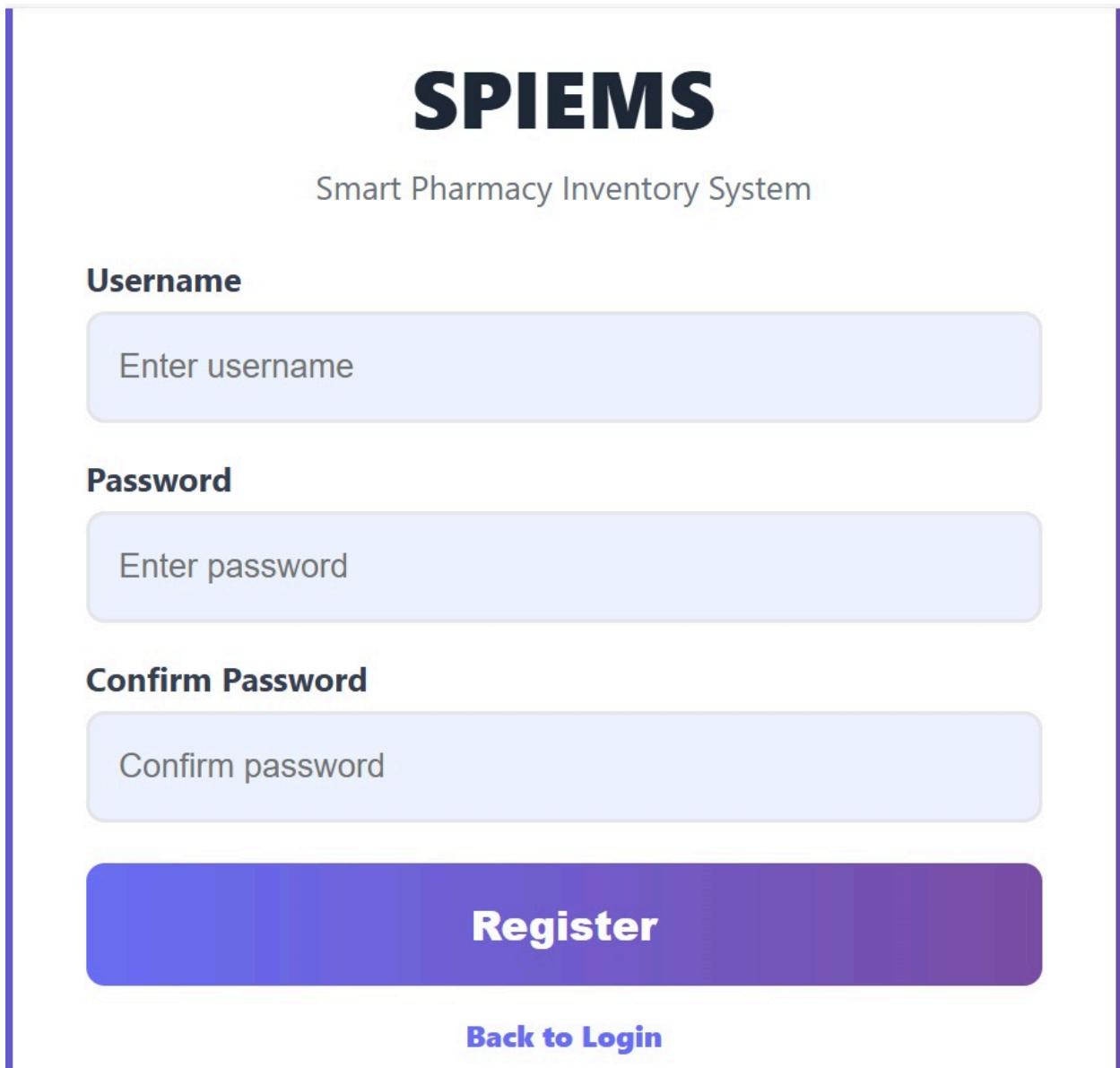
```
    ▲ Controllers
        ▷ C# AccountController.cs
        ▷ C# BatchesController.cs
        ▷ C# DashboardController.cs
        ▷ C# HomeController.cs
        ▷ C# MedicinesController.cs
        ▷ C# ReturnController.cs
        ▷ C# SalesController.cs
        ▷ C# SalesReportsController.cs
        ▷ C# ShopController.cs
        ▷ C# StaffController.cs
        ▷ C# SuppliersController.cs
    ▲ Models
        ▷ C# CartItemVm.cs
        ▷ C# DashboardVm.cs
        ▷ C# ErrorViewModel.cs
        ▷ C# LoginVm.cs
        ▷ C# SignupVm.cs
        ▷ C# StaffCreateVm.cs
    ▲ Views
        ▲ Account
```





## 6.2 Screenshots

### Login Page



### Sign up Page



# SPIEMS Login

Smart Pharmacy Inventory System

**Username**

Enter your username

**Password**

Enter your password

**Login**

Don't have an account? [Sign up here](#)

## Dashboard

The dashboard features a top navigation bar with user info ('Logged in as: AiraMaryam (Admin)'), navigation links ('View Charts', 'Sales Report', 'Logout'), and a sidebar with 'Quick Actions' like 'Add Medicine', 'Add Batch', etc.

Key statistics displayed:

- Near Expiry (30 days): 2
- Low Stock: 2
- Total Medicines: 6
- Total Batches: 4

A critical alert box shows: 1 EXPIRED Batch - Remove immediately! with a 'View' button.

## Add Medicine

### + Add Medicine

← Back to Medicines

Generic Name

Brand Name

Company

Category  
 Capsules

Reorder Level  
 10

Save Cancel

## Add Batch

### + Add New Batch

[← Back to Batches](#)

Medicine

-- Select Medicine --

Supplier (optional)

-- Select Supplier --

Linking supplier enables Supplier → Batches view.

Batch No

Purchase Date      Expiry Date (optional)

12/27/2025      mm/dd/yyyy

Quantity      Cost Price      Sale Price

0      0      0

Save Batch    Cancel

## Add Staff

### Add Staff

Username

Password

Create Staff    Back

## Suppliers

Suppliers				
Name	Phone	Address	Status	Actions
airzzz	0334-1284537	RWP	Active	Batches  Edit  Deactivate
MedSupply Co	021-1234567	Karachi	Inactive	Batches  Edit  Activate
PharmaDirect	051-9876543	Islamabad	Inactive	Batches  Edit  Activate

## Add Suppliers

# + Add Supplier

[← Back](#)

Name

Phone

Address

Save

## Staff Accounts

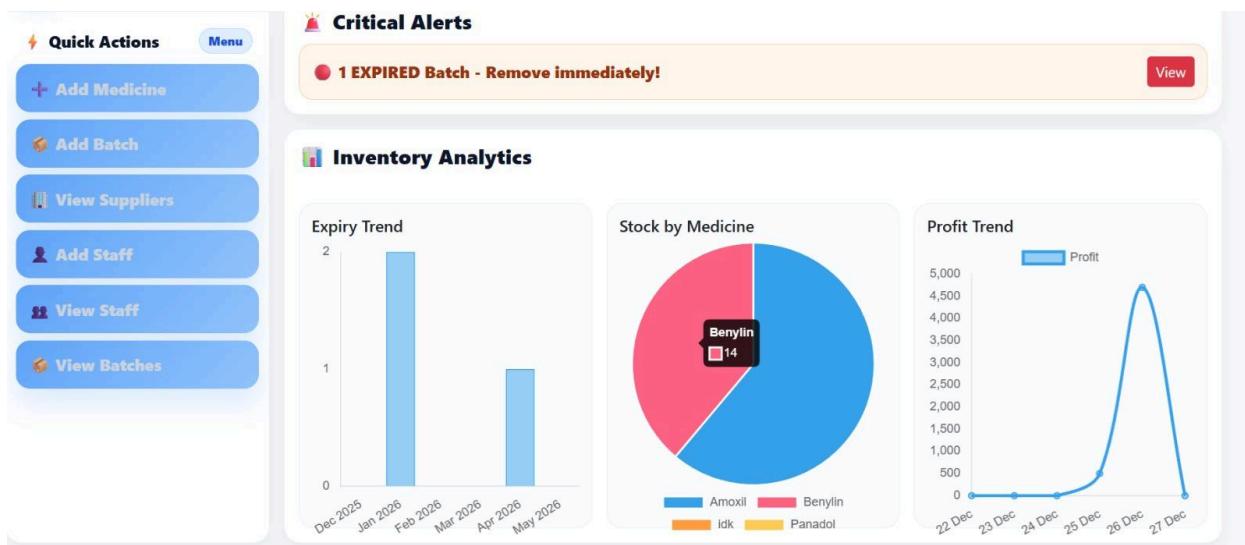
Staff Accounts		Role	Action
ID	Username		
8	airammm	Staff	<button>Delete</button>
4	aira	Staff	<button>Delete</button>
2	staff	Staff	<button>Delete</button>

## Medicine Batches

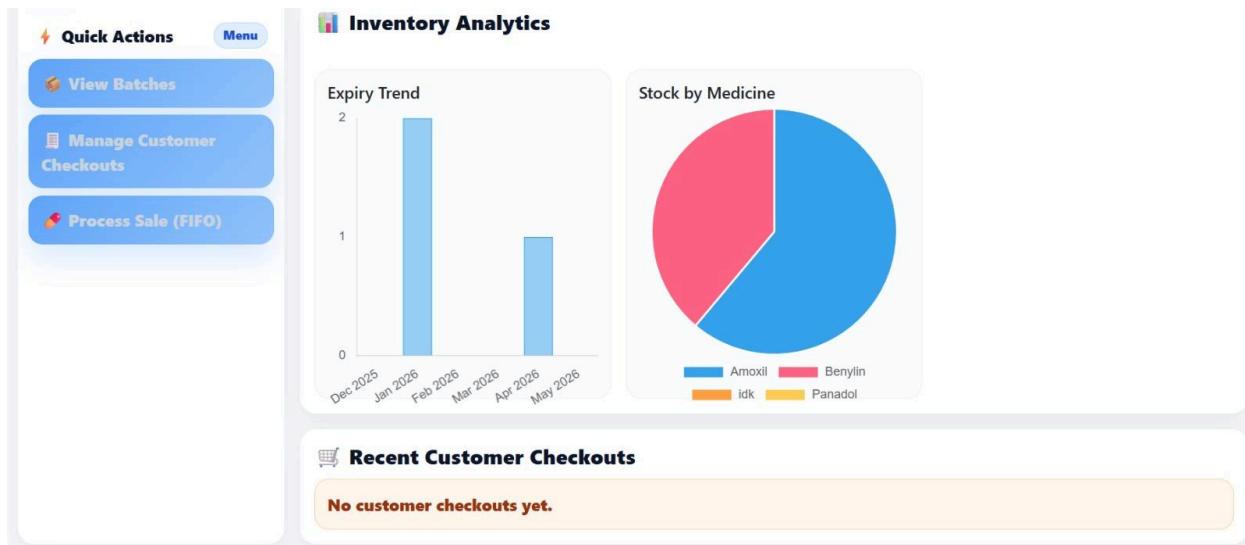
Medicine Batches									
Batch No	Medicine	Supplier	Purchase Date	Expiry Date	Days Left	Status	Quantity	Cost	Sale Price
2	Cough Syrup	airzzz	12/25/2025	1/24/2026	28	<span>🟡 Warning</span>	14	300.00	500.00
2	Amoxicillin		12/24/2025	4/23/2026	117	<span>🟢 Safe</span>	22	200.00	300.00
3	brufen		12/24/2025	1/23/2026	27	<span>🟡 Warning</span>	0	200.00	300.00
4	Paracetamol		12/25/2024	3/25/2025	-277	<span>🔴 Critical</span>	0	200.00	300.00

## Sales Report

Sales Report (Admin)							
Sale #	Date	User (Staff)	Role	Status	Total	Medicines	Items
11	12/26/2025 10:28:34 AM	aira	Staff	SoldOut	300.00	Amoxicillin	1



## Staff



The dashboard displays the following information:

- SPIEMS Dashboard**
- Quick Actions:**
  - View Batches
  - Manage Customer Checkouts
  - Process Sale (FIFO)
- Near Expiry (30 days):** 2 items
- Low Stock:** 2 items
- Total Medicines:** 6 items
- Total Batches:** 4 items
- Recent Customer Checkouts:** No customer checkouts yet.

## CUSTOMER MODULE

### Shop Medicines

Search Clear View Cart

<b>Amoxicillin</b> Amoxil <b>Company:</b> Pfizer <b>Price:</b> \$300.00 <b>Stock:</b> 22 <input type="button" value="1"/> <input type="button" value="Add"/>	<b>brufen</b> idk <b>Company:</b> idk <b>Price:</b> N/A <b>Stock:</b> Out of Stock <input type="button" value="1"/> <input type="button" value="Add"/>	<b>Cough Syrup</b> Benylin <b>Company:</b> Johnson & Johnson <b>Price:</b> \$500.00 <b>Stock:</b> 4 <input type="button" value="1"/> <input type="button" value="Add"/>	<b>Ibuprofen</b> Brufen <b>Company:</b> Abbott <b>Price:</b> N/A <b>Stock:</b> Out of Stock <input type="button" value="1"/> <input type="button" value="Add"/>
<b>Paracetamol</b> Panadol <b>Company:</b> GSK <b>Price:</b> N/A <b>Stock:</b> Out of Stock <input type="button" value="1"/> <input type="button" value="Add"/>	<b>paracetamol</b> iiddk <b>Company:</b> medical huh <b>Price:</b> N/A <b>Stock:</b> Out of Stock <input type="button" value="1"/> <input type="button" value="Add"/>		

## Shopping Cart

Shopping Cart

Medicine	Quantity	Est. Price	Total	Action
Amoxicillin (Amoxil)	1	\$300.00	\$300.00	<a href="#">Remove</a>
<b>Total (Estimated): \$300.00</b>				

[Continue Shopping](#) [Checkout](#)

## Order Placement



## Order Placed!

Thank you for your purchase. Your order has been processed successfully.

[Continue Shopping](#)

## 7. Authentication and Security

The Smart Pharmacy Inventory and Expiry Management System (SPIEMS) incorporates robust authentication and security mechanisms to protect sensitive pharmacy data and ensure that only authorized users can access specific functionalities. Built on ASP.NET Core's built-in Identity system, the application enforces secure user authentication and role-based authorization, preventing unauthorized access to administrative features while allowing controlled operations for staff members.

## **7.1 Authentication – Working Flow**

Authentication in SPIEMS follows a standard ASP.NET Core Identity workflow. Users (Admin or Staff) are prompted to log in with their username and password on a dedicated login page. Upon submission, the credentials are validated against hashed passwords stored in the database using BCrypt hashing for enhanced security. If valid, the system generates an authentication cookie and establishes a user session, redirecting the user to the dashboard. Invalid credentials trigger an error message without revealing whether the username or password was incorrect, following security best practices to prevent enumeration attacks. Session timeout and logout functionality are implemented to ensure users are automatically signed out after inactivity or upon explicit logout.

## **7.2 Protected Routes**

Protected routes are secured using ASP.NET Core's authorization attributes. The [Authorize] attribute is applied to controllers or actions that require authentication, automatically redirecting unauthenticated users to the login page. Role-specific access is enforced using [Authorize(Roles = "Admin")] for administrative actions (e.g., adding medicines, managing categories, viewing reports) and [Authorize(Roles = "Staff")] for staff operations (e.g., processing sales). This ensures that staff cannot access management features, while admins have full privileges. The system also includes fallback policies to deny access by default, enhancing the principle of least privilege.

## **7.3 Middleware Usage**

ASP.NET Core middleware pipeline is leveraged for centralized security handling. Authentication middleware configures cookie-based authentication with secure settings (HTTPS only in production, expiration policies). Authorization middleware evaluates user claims and roles for every request. Custom middleware can be added for logging authentication events or blocking suspicious activity. Additionally, anti-forgery tokens are used in forms to prevent CSRF attacks, and Entity Framework Core's parameterized queries protect against SQL injection.

These middleware components work together to provide layered security throughout the application lifecycle.

## 8. Validation & Error Handling

SPIEMS implements comprehensive validation and error handling at both server and client sides to ensure data integrity, improve user experience, and maintain system reliability. Proper validation prevents invalid data from entering the database, while meaningful error responses guide users and protect against potential security vulnerabilities.

### 8.1 C# Validation (Server-Side)

Server-side validation is performed using Data Annotations on model classes and Fluent Validation rules where needed. Required fields (e.g., medicine name, batch quantity), data types (e.g., positive numbers for quantity and price), and custom rules (e.g., purchase date cannot be in the future) are validated before database operations. ModelState.IsValid checks in controllers ensure that only valid data proceeds to the business logic layer. This prevents corrupted data and provides immediate feedback for correction.

### 8.2 Frontend Validation

Frontend validation is achieved through HTML5 attributes (required, type, pattern) and Bootstrap styling for visual feedback. JavaScript enhances the experience with real-time validation (e.g., checking quantity availability during sales). Unobtrusive validation using jQuery Validation and ASP.NET Core's client-side validation scripts display error messages instantly without page reloads, improving usability and reducing server load.

### 8.3 Proper Error Responses (400, 401, 404)

The system returns appropriate HTTP status codes for different scenarios:

- **400 Bad Request:** Returned for invalid or malformed input data (e.g., negative quantity, missing required fields).
- **401 Unauthorized:** Triggered when authentication fails or session expires, redirecting to login page.
- **404 Not Found:** Displayed when requested resources (e.g., medicine ID) do not exist. Custom error pages with friendly messages and logging of exceptions ensure users receive clear guidance while administrators can monitor issues through logs. This structured approach enhances both user experience and system maintainability.

## 9. User Access and Credentials

The Smart Pharmacy Inventory and Expiry Management System (SPIEMS) employs a role-based access control system to ensure secure and efficient operations, defining three main user roles: Admin, Staff, and Customer. The Admin role has comprehensive privileges, including adding, editing, and deleting medicines, categories, suppliers, batches, and customer records, as well as generating reports and managing user accounts. The Staff role is focused on operational tasks such as viewing stock, searching customers, processing cash or credit sales, and accessing the dashboard with alerts, without the ability to modify master data. Customers are incorporated as a non-login role for tracking purposes, allowing admins or staff to manage customer profiles (name, contact, address) for credit sales, purchase history, and outstanding balances, but customers do not have direct system access or credentials. Authentication is handled securely through ASP.NET Core's built-in system for Admin and Staff, requiring valid credentials, while customer data is maintained passively for enhanced sales and relationship management. This multi-role structure promotes data security, operational efficiency, and customer-centric features in a real-world pharmacy setting.

## **10. Conclusion:**

The Smart Pharmacy Inventory and Expiry Management System (SPIEMS) successfully fulfills its primary objectives by providing a comprehensive, web-based solution for automating pharmacy operations in small to medium-sized medical stores. Through the effective implementation of ASP.NET Core MVC architecture, the system delivers key features such as automatic expiry date prediction, batch-wise inventory tracking, FIFO-based sales processing, real-time color-coded alerts, role-based access control for Admin, Staff, and customer management, and an interactive dashboard with charts for stock and sales insights. The addition of the customer module further enhances practicality by supporting credit sales, purchase history tracking, and outstanding balance management. Developed collaboratively by the team with parallel task allocation, the project demonstrates strong application of web development principles, database design, security practices, and responsive UI, resulting in a reliable, user-friendly application that significantly reduces medicine wastage, optimizes stock levels, and improves overall operational efficiency.

## **Appendices**

[1] Report Repository-

<https://github.com/minahilnaz2005-lgtm/Smart-Pharmacy-Inventory-System>

[2] Source Code Repository-

<https://github.com/minahilnaz2005-lgtm/Smart-Pharmacy-Inventory-System>

[3] Project Video Demo(aira)-

<https://youtu.be/npcOKAi86v0?si=Z6dzMIZUuSY8AMfH>