

# Muhammad Ahmad

## .Net Developer

✉ ahmed.asif4849@gmail.com

📍 Faisalabad

☎ +923147809083

🌐 <https://www.linkedin.com/in/m-ahmed-4849as>

### Profile

Junior Software Developer with hands-on experience in **C#, ASP.NET Core Web API, Code-First and Entity Framework Core**. Familiar with **HTML, CSS, and Bootstrap** for responsive UI development. Seeking a junior-level role to grow skills and contribute to backend-focused projects.

### Languages

English	<div></div>
Urdu	<div></div>

### Skills

C#	<div></div>
HTML	<div></div>
SQL	<div></div>
BOOTSTRAP	<div></div>
Agentic AI	<div></div>
Generative Ai	<div></div>
EF Code-First	<div></div>
Angular	<div></div>

### Projects

#### Employee Management System

**Technologies:** ASP.NET Core, Entity Framework Core, SQL Server, HTML, CSS, Bootstrap, JWT, Swagger.

**Description:** Developed a full backend system for managing employees, projects, and timesheets, with secure authentication and role-based access control. Built responsive UI pages for demonstrations.

#### Key Achievements:

- Implemented **JWT-based authentication** with register, login, and password reset functionality.
- Modeled **Many-to-Many relationships** between Employees and Projects using a custom join entity.

### Education

#### Bachelor In Computer Science

Gc University Faisalabad  
2022 – 2025 | Faisalabad

#### Intermediate

Kips College Jail Road Faisalabad  
07/2019 – 2021 | Faisalabad

### Professional Experience

#### Junior Software Developer (Intern)

NTRC Faisalabad

**Junior / Full Stack Developer Encodal | 1 Year (3 Months Internship + 9 Months Full-Time)**

#### Role & Responsibilities:

- Learned and applied **Angular 19, HTML5, CSS3, and Bootstrap** for building responsive frontend pages.
- Learned and applied **C# programming** for backend development using ASP.NET Core.
- Developed **RESTful APIs** with ASP.NET Core Web API, Code First approach in Entity Framework (EF and Entity Framework Core).
- Implemented **JWT-based authentication** and **role-based authorization**.
- Designed and managed **SQL Server databases** with proper relationships and constraints.
- Applied **Clean Architecture principles** to structure scalable applications.
- Used **Git** and **GitHub** for version control and collaboration.
- Documented APIs using Swagger.

#### Key Achievements:

- Built **Employee Management System backend, E-Commerce backend API, and XPrice – Multi-Store Product Price Comparison (Angular 19 & TypeScript)** as hands-on projects, focusing on clean architecture, API development, and responsive web application design.
- Successfully implemented secure login, registration, and password reset functionality.
- Strengthened problem-solving skills through real-world development tasks and debugging.

- Built **Timesheet submission and approval workflow** (Draft → Submitted → Approved/Rejected).
- Applied **Clean Architecture principles** for scalable and maintainable code.

### Country-API

Role: Backend Developer

**Technologies:** ASP.NET Core, Entity Framework Core, SQL Server, AutoMapper, Swagger

**Description:** Developed a backend API to manage country data and handle file uploads, storing uploaded documents securely in the **wwwroot folder**.

#### Key Achievements:

- Implemented **CRUD operations** for country management.
- Built **file upload functionality**, saving documents in the **wwwroot directory** for secure access.
- Created **DTOs and AutoMapper profiles** for clean data transfer between API and database.
- Designed and managed **SQL Server database** with proper relationships and constraints.
- Documented APIs using **Swagger** for testing and collaboration.

### E-commerce Web Application

Role: Backend Developer

**Technologies:** ASP.NET [↗](#) Core, Entity Framework Core, SQL Server, JWT, AutoMapper, Swagger.

**Description:** Developed a fully functional **E-commerce platform backend** with secure authentication and role-based access, allowing users to browse products, manage their carts, and place orders. Built responsive UI components for product display and order management.

#### Key Achievements:

- Implemented **JWT-based authentication** with user registration, login, and role-based authorization.
- Designed **relational database** for products, users, orders, and carts using **EF Core**.
- Integrated **business logic** for product management, order placement, and order tracking.
- Built **responsive frontend pages** using HTML, CSS, and Bootstrap for product catalog and cart functionality.
- Documented APIs using **Swagger** and maintained project version control using **Git/GitHub**.
- Applied **Clean Architecture principles** to ensure scalable and maintainable code.

### Ordering System Backend APIs | ASP.NET Core & EF Core

Role: Backend Developer

- Gained full-stack exposure by integrating **Angular frontend pages** with backend APIs, enhancing user interactivity and experience.

## Certificates

### Introduction to Generative AI [↗](#)

Completed a certified course on **Generative AI fundamentals** from Simplilearn, **covering LLM basics**, use cases, and responsible AI concepts.

### Startup School: Prompt to Prototype

Completed the *Startup School: Prompt to Prototype* program, learning how to turn ideas into working prototypes using practical AI tools and prompt engineering. Gained skills in **idea validation, rapid prototyping, and problem-solving**, and learned how to leverage AI to **accelerate product development** from concept to functional prototype.

### Introduction to AI – 365 DataScience (Certified) [↗](#)

Completed the *Introduction to AI* course from **365 DataScience**, gaining a solid understanding of **artificial intelligence concepts**, including machine learning, data-driven decision making, and the fundamentals of AI applications. Learned the differences between **narrow AI and general AI**, and explored practical examples of how AI is used in modern software and business solutions.

### Introduction to AI Agents and Agentic AI – 365 DataScience (Certified) [↗](#)

Completed the *Introduction to AI Agents and Agentic AI* course from 365 DataScience, learning about **intelligent agents, agentic behavior, and decision-making systems**. Gained knowledge of how AI agents perceive environments, make autonomous decisions, and execute actions to achieve goals. Explored practical examples of **agent-based AI applications** in software and automation using n8n.

### Introduction to Generative AI Studio – Simplilearn & Google (Certified) [↗](#)

Certified in *Generative AI Studio* (Simplilearn & Google), focused on **generative AI tools, prompt engineering, and AI-driven content creation**.

**Technologies:** ASP.NET Core, Entity Framework Core, SQL Server, AutoMapper, Swagger

**Description:** Built backend APIs for an ordering system handling **customers, products, orders, and order details**. Focused on clean architecture, database design, and efficient business logic.

• **Key Highlights:**

- Implemented backend logic for **customer registration, product management, and order processing**.
- Designed **relational database models** for customers, orders, order details, and products.
- Used **DTOs and AutoMapper** for clean data transfer between API and database.
- Documented APIs with **Swagger** and maintained version control with **Git/GitHub**.

**BookStore API**

**Technologies:** ASP.NET Core, Entity Framework Core, SQL Server, Custom Filter, Swagger

**Description:** Developed a RESTful API for managing books with structured controllers, filters, and models, focusing on clean architecture and maintainable code.

**Key Achievements:**

- Implemented **BooksController** for CRUD operations on book data.
- Created **custom filters** (AdminOnlyFilter, ApiKeyAuthFilter, HandleException Filter, LogActionFilter, WrapResultFilter) for authorization, exception handling, logging, and response wrapping.
- Configured **Swagger/OpenAPI** with API key authentication to simplify testing and documentation.
- Designed **Book model** and integrated with API endpoints for structured data management.
- Set up **Program.cs** for dependency injection, controllers, and middleware pipeline.

**XPrice – Multi-Store Product Price Comparison (Angular 19, TypeScript)**

**Description:** Developed a web application to **compare product prices across multiple stores**, providing users with an **interactive interface** to quickly identify the **lowest price** for each product. Focused on **responsive design, reusable components, and dynamic data handling**.

**Key Achievements:**

- Implemented XPrice Component to display **product price comparisons** in a structured table view.
- Created ProductService to fetch and manage **product and store price data**, enabling scalability.
- Designed Product and StorePrice models for **structured and maintainable data management**.

- Utilized **Angular features** like component-based architecture, dependency injection, and observables for **reactive data handling**.
- Added **dynamic lowest-price highlighting** and **currency formatting** to enhance **user experience**.