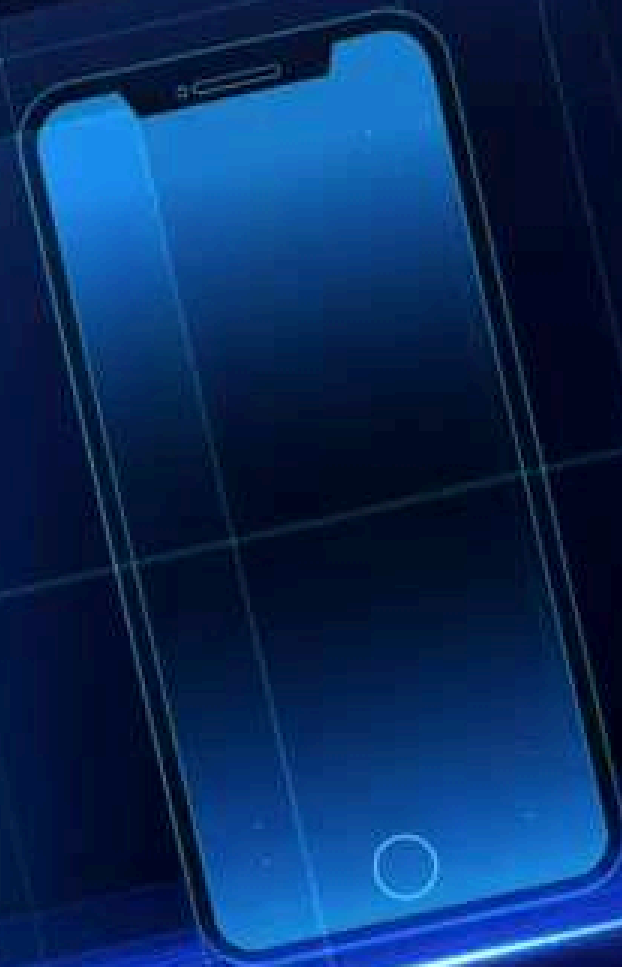ZIRA

AGENDA

01

Project Overview

We built a modern, user-friendly online clothing store designed for shoppers and admins to manage items, orders, and browsing seamlessly.

02

What Did We Do ?

Our team developed the store using ASP.NET Core MVC, .NET 8.0, SQL Server, and a clean front-end to support real-time item filtering, cart management, and dynamic image handling.

03

Project Preview

We'll walk through a live demo of the platform, showcasing features like category-based browsing, cart and favorites, item variant selection, and admin item editing.

04

Project Journey

key milestones, challenges faced, and progress made so far.

Project Overview

What is our project?

Our project is a full-stack web application . It's an Online Clothing Store that allows store owners to manage their products, variants, and inventory, while providing users with a smooth browsing and shopping experience.

Why did we choose this project?

We chose this project because a real client approached us with a request to build an Online Clothing Store. This presented a valuable opportunity to work on a real-world application that not only meets an actual business need, but also allows us to practice and apply our skills for our college project. It's the perfect blend of practical experience and academic relevance.

Target Audience

Store Owners: To help them manage their inventory, products, and orders efficiently.

Customers: To provide a modern, user-friendly platform to browse and purchase clothing items with ease.

What Did We Do?

Our team developed a complete Online Clothing Store using ASP.NET Core MVC (.NET 8.0), SQL Server, and a responsive front-end to support features such as real-time product filtering, shopping cart management, and dynamic image handling.

MVC Architecture

We chose the Model-View-Controller (MVC) pattern because the client specifically needed a web-only solution. MVC allowed us to clearly separate concerns, making the application easier to maintain, test, and scale in the future.

Repository Pattern

To keep our data access layer clean and decoupled, we used the Repository Pattern. This helped us abstract the database operations and made our code easier to test and more modular, while promoting single responsibility and separation of concerns.

Authentication & Authorization

For secure user access, we implemented Authentication and Authorization using ASP.NET Identity. This allowed us to manage roles, such as Store Owners, and Customers, ensuring each user has access to the appropriate parts of the application.

Dependency Injection

We took full advantage of .NET's built-in Dependency Injection to manage our services and repositories efficiently. This approach made our application more flexible and testable, while reducing tight coupling between components.

SQL Server

We used SQL Server as our relational database to store all data related to users, products, orders, and store settings. It provided strong support for data integrity, relationships, and scalability.

Project Preview

View More!

Stay bold. Stay warm.

🌐 **Project Demo**   🔗 **LinkedIn Post**

Project Journey

- **Excalidraw**

Notion

MEET OUR TEAM

ZIAD GHORABA

Team Leader/ Backend

SARA HESHAM

Backend/ Tester

MOHAND EL-MESH

Frontend

OMAR ATEF

Frontend

HASSAN ARAFAT

Frontend

Thank You!