Project Team Members

Team Leader

- **Ahmed Nabil Mohammed Elzayda**

Team Members

1. **Zeyad Elsayed Abo El3neen**
2. **Ahmed Emad Fawzy Abd Elnaby**
3. **Ahmed Osama Mohammed Khairy**
4. **Islam Mohammed Mahmoud Mohammed**
5. **Kaled Mohammed Mahmoud Ebraheem**

Case study: E-Commerce Website for Clothing

Objective:
To design, develop, test, and document an E-Commerce website specifically for selling clothing. The website will allow customers to browse, add items to their cart, and purchase products, while administrators manage inventory, orders, and customer accounts.

Technologies Used:

1. **SQL Database**: For data storage and retrieval.
2. **C#**: Primary programming language for backend logic.
3. **ASP.NET MVC**: Framework for implementing the Model-View-Controller architecture.
4. **Visual Studio**: Integrated Development Environment (IDE) for building and debugging the project.

Stakeholders

1. **Customers**: Individuals who use the website to browse and purchase clothing.
2. **Administrators**: Managers responsible for handling product listings, orders, and user accounts.
3. **Developers**: Team responsible for the system's development.
4. **Business Client**: The entity providing requirements and feedback.

Functional Requirements

1. **Customer Registration and Authentication**
    - Allow customers to register using an email and password.
    - Securely store passwords (e.g., using bcrypt hashing).
    - Provide secure login and logout options, with a password reset feature.
    - Separate login interface for administrators.
2. **Customer Profile Management**
    - Enable customers to view and edit profile information.
    - Allow customers to update passwords and manage contact details.
3. **Product Catalog and Search**
    - Display a categorized catalog of clothing items, organized by type (men, women, children), size, and price.
    - Allow customers to search for items using product name, category, price, or keywords.
    - Provide detailed information for each product (description, size, color, price, availability).
4. **Shopping Cart**
    - Allow customers to add, update, or remove items in the shopping cart.
    - Calculate the total price including taxes and shipping.
5. **Order Processing and Payment**
    - Offer multiple payment options (credit cards, PayPal, etc.).
    - Securely process payment details and confirm successful payment.
6. **Inventory Management**
    - Allow administrators to add, edit, and remove products from the catalog.
    - Track and update product availability in real-time.

7. **Order Management**
    - Display order details to administrators, including status tracking (processing, shipped, delivered).
    - Update order status and notify customers of any changes.
8. **Notifications**
    - Send email confirmations to customers for registration, purchase, and order status updates.
    - Notify customers about special offers or new arrivals.

Non-Functional Requirements

1. **Performance**
    - Support up to 200 concurrent users without performance degradation.
    - Load pages within 2 seconds under normal conditions.
    - Estimate capacity to ensure efficiency with high data and user loads.
2. **Security**
    - Enforce HTTPS for secure data transmission.
    - Protect payment and sensitive information and safeguard against common attacks.
3. **Usability**
    - Ensure an intuitive and user-friendly interface for seamless browsing and purchasing.
    - Provide a responsive design compatible with mobile devices, tablets, and desktops.
4. **Scalability**
    - Design to support an increase in users, products, and categories over time.
5. **Reliability and Availability**
    - Ensure 99.9% availability during business hours.
    - Handle errors gracefully and log important system events.
6. **Maintainability**
    - Write clean, modular code following best practices and standards.
    - Use version control to manage changes effectively.

Technical Requirements

1. **Programming Language and Framework**
    - Use C# with ASP.NET MVC for implementing backend logic and the MVC architecture.
2. **Database**
    - Utilize SQL for relational data storage.
    - Design a well-structured database schema with efficient indexing.
3. **Development Environment**
    - Develop using Visual Studio for its powerful tools and debugging features.
4. **Testing Frameworks**
    - Use NUnit for unit testing to ensure code reliability and performance.
5. **Deployment**
    - Optionally, deploy the website using cloud services (e.g., Heroku, AWS Free Tier) for accessibility and scalability.

End Goal

To deliver a fully-functional, user-friendly, and secure e-commerce platform for selling clothing, providing a smooth purchasing experience for customers and efficient management tools for administrators.