Project Requirement

1. Introduction

- **Purpose**: The purpose of this document is to define the requirements for the **TechXpress E-commerce Platform**, a scalable and maintainable web application for selling electronics. The platform will allow users to browse products, add items to a shopping cart, and complete purchases using integrated payment gateways. It will also include an admin panel for managing products, categories, and orders.
- **Scope**: The platform will be developed using **ASP.NET Core MVC**, **Entity Framework Core**, and **Stripe** for payment integration. It will follow **NTier Architecture**, **Repository Pattern**, and **Unit of Work Pattern** for efficient development and separation of concerns.

2. Functional Requirements

The functional requirements describe the features and behaviors of the system. These are categorized by user roles and system functionalities.

2.1 User Roles

1. **Customer**:
    - Browse products by category (e.g., laptops, mobiles, cameras).
    - Search for products using keywords and filters (e.g., price range, brand).
    - Add products to the shopping cart.
    - View and manage the shopping cart (update quantities, remove items).
    - Proceed to checkout and complete the purchase using Stripe.
    - View order history and track order status.
    - Manage user profile (update personal details, change password).
2. **Admin**:
    - Manage product listings (add, update, delete products).
    - Manage product categories (add, update, delete categories).
    - View and manage orders (update order status, cancel orders).
    - View sales reports and analytics.
    - Manage user roles and permissions.

2.2 Product Management

1. The system shall allow admins to add new products with details such as:
    - Product name, description, price, stock quantity, category, and image.
2. The system shall allow admins to update product details.
3. The system shall allow admins to delete products.

4. The system shall display products to customers with filters and sorting options (e.g., price low to high, popularity).

2.3 Shopping Cart

1. The system shall allow customers to add products to the shopping cart.
2. The system shall display the shopping cart with the following details:
    - Product name, price, quantity, and total price.
3. The system shall allow customers to update the quantity of items in the cart.
4. The system shall allow customers to remove items from the cart.
5. The system shall calculate the total price of items in the cart, including taxes and discounts (if applicable).

2.4 Checkout and Payment

1. The system shall allow customers to proceed to checkout from the shopping cart.
2. The system shall collect shipping and payment details during checkout.
3. The system shall integrate with **Stripe** to process payments securely.
4. The system shall display an order confirmation page with order details and a unique order ID.
5. The system shall send an email confirmation to the customer after a successful purchase.

2.5 Order Management

1. The system shall allow customers to view their order history.
2. The system shall allow customers to track the status of their orders (e.g., pending, shipped, delivered).
3. The system shall allow admins to update the status of orders.
4. The system shall allow admins to cancel orders.

2.6 User Authentication and Authorization

1. The system shall allow users to register and log in using **ASP.NET Identity**.
2. The system shall enforce role-based access control (RBAC):
    - Customers can only access customer-related features.
    - Admins can access the admin panel and manage products, categories, and orders.

Non-Functional Requirements

These requirements describe the quality attributes of the system.

1. **Performance**:
    - Product search results shall load within **2 seconds** under normal load.
2. **Security**:
    - The system shall encrypt sensitive data (e.g., passwords, payment details) using **AES-256 encryption**.
    - The system shall prevent SQL injection and cross-site scripting (XSS) attacks.
    - The system shall enforce HTTPS for all communication.
3. **Usability**:
    - The system shall provide clear error messages and validation feedback to users.
4. **Scalability**:
    - The system shall be designed to scale horizontally to support future growth in users and products.

4. User Requirements

1. **Customer Experience**:
    - Customers shall be able to easily browse and search for products.
    - Customers shall have a seamless checkout experience with minimal steps.
    - Customers shall receive timely notifications about their order status.
2. **Admin Experience**:
    - Admins shall have an intuitive dashboard for managing products, categories, and orders.
    - Admins shall be able to generate sales reports and analytics.

---

6. External Interfaces

1. **Stripe API**:
    - The system shall integrate with Stripe for payment processing.
    - The system shall handle payment success and failure responses from Stripe.

      // not required paid service

2. **Email Service**:

- The system shall send email notifications (e.g., order confirmation, password reset) using an external email service (e.g., SendGrid).

  //not required

---

7. Assumptions and Constraints

1. **Assumptions**:
   - The system will be deployed on **Microsoft Azure**.
2. **Constraints**:
   - The project must be completed within **4 weeks**.

Database Requirements for TechXpress E-commerce Platform

1. Database Overview

The database will store all data related to products, categories, users, orders, and payments. It will be designed using Entity Framework Core and follow the Repository Pattern and Unit of Work Pattern for efficient data access and management. The database will be hosted on Microsoft Azure SQL Database for scalability and reliability.

2. Database Schema

The database schema will consist of the following tables:

2.1 Users Table

- **Purpose: Stores user information for authentication and role-based access control.**
- **Fields:**
  - **UserId (Primary Key, GUID)**
  - **FirstName (String, Not Null)**
  - **LastName (String, Not Null)**
  - **Email (String, Unique, Not Null)**
  - **PasswordHash (String, Not Null)**
  - **Role (String, Not Null) – Values: Customer, Admin**
  - **CreatedAt (DateTime, Not Null)**
  - **LastLogin (DateTime, Nullable)**
  - **IsActive (Boolean, Default: True)**

2.2 Products Table

- **Purpose: Stores product details for display and management.**
- **Fields:**
    - **ProductId (Primary Key, GUID)**
    - **Name (String, Not Null)**
    - **Description (String, Not Null)**
    - **Price (Decimal, Not Null)**
    - **StockQuantity (Integer, Not Null)**
    - **CategoryId (Foreign Key, GUID, Not Null)**
    - **ImageUrl (String, Nullable)**
    - **CreatedAt (DateTime, Not Null)**
    - **UpdatedAt (DateTime, Nullable)**
    - **IsActive (Boolean, Default: True)**

2.3 Categories Table

- **Purpose: Organizes products into categories for better navigation.**
- **Fields:**
    - **CategoryId (Primary Key, GUID)**
    - **Name (String, Not Null)**
    - **Description (String, Nullable)**
    - **CreatedAt (DateTime, Not Null)**
    - **UpdatedAt (DateTime, Nullable)**
    - **IsActive (Boolean, Default: True)**

2.4 Orders Table

- **Purpose: Tracks customer orders and their status.**
- **Fields:**
    - **OrderId (Primary Key, GUID)**
    - **UserId (Foreign Key, GUID, Not Null)**
    - **OrderDate (DateTime, Not Null)**
    - **TotalAmount (Decimal, Not Null)**
    - **OrderStatus (String, Not Null) – Values: Pending, Processing, Shipped, Delivered, Cancelled**
    - **PaymentStatus (String, Not Null) – Values: Paid, Unpaid**
    - **ShippingAddress (String, Not Null)**
    - **TransactionId (String, Nullable) – Stores Stripe transaction ID.**

2.5 OrderDetails Table

- **Purpose: Stores individual items within an order.**
- **Fields:**
    - **OrderDetailId (Primary Key, GUID)**
    - **OrderId (Foreign Key, GUID, Not Null)**
    - **ProductId (Foreign Key, GUID, Not Null)**
    - **Quantity (Integer, Not Null)**
    - **UnitPrice (Decimal, Not Null)**
    - **TotalPrice (Decimal, Not Null)**

2.6 ShoppingCart Table

- **Purpose: Temporarily stores items added to the cart by users.**
- **Fields:**
    - **CartId (Primary Key, GUID)**
    - **UserId (Foreign Key, GUID, Not Null)**
    - **ProductId (Foreign Key, GUID, Not Null)**
    - **Quantity (Integer, Not Null)**
    - **AddedAt (DateTime, Not Null)**

---

3. Relationships

1. **Users ↔ Orders: One-to-Many (One user can have many orders).**
2. **Orders ↔ OrderDetails: One-to-Many (One order can have multiple items).**
3. **Products ↔ Categories: Many-to-One (Many products belong to one category).**
4. **Products ↔ ShoppingCart: Many-to-Many (Many products can be in many carts).**
5. **Products ↔ OrderDetails: Many-to-Many (Many products can be in many orders).**

4. Data Validation Rules

1. **Product Price:**
    - **Must be a positive decimal value.**
    - **Cannot be null.**
2. **Stock Quantity:**
    - **Must be a non-negative integer.**
    - **Cannot be null.**
3. **Email:**
    - **Must be unique.**
    - **Must follow a valid email format (e.g., user@example.com).**

4. **Order Status:**

- **Must be one of the predefined values: Pending, Processing, Shipped, Delivered, Cancelled.**

5. **Payment Status:**
   - **Must be one of the predefined values: Paid, Unpaid.**

---

5. Indexes

1. **Users Table:**
   - **Index on Email for fast lookup during login.**
2. **Products Table:**
   - **Index on CategoryId for efficient filtering by category.**
   - **Index on Name for fast search functionality.**
3. **Orders Table:**
   - **Index on UserId for quick retrieval of user orders.**
   - **Index on OrderDate for sorting and reporting.**

---

6. Security Requirements

1. **Data Encryption:**
   - **Sensitive data (e.g., PasswordHash, TransactionId) will be encrypted using AES-256 encryption.**
2. **Access Control:**
   - **Only authorized users (e.g., admins) can access and modify product and order data.**
   - **Customers can only access their own orders and shopping cart.**
3. **Audit Logs:**
   - **All changes to critical tables (e.g., Products, Orders) will be logged for auditing purposes.**

---

8. Performance Requirements

1. **Response Time:**
   - **Product search queries must return results within 2 seconds under normal load.**
   - **Order placement transactions must complete within 5 seconds.**