

Product Catalog Web Application Xceed Task- Technical Documentation

Table of Contents

- [Architecture Overview](#)
 - [Backend Implementation](#)
 - [Frontend Implementation](#)
 - [Identity and Authorization](#)
 - [Advantages of the Approach](#)
 - [Best Practices and Defenses](#)
 - [System Screens](#)
-

Architecture Overview

The project follows an N-tier architecture for scalability and maintainability. Here's a breakdown of the solution structure:

1. Solution Structure

The solution contains 4 projects:

- **Web Layer:** Manages Areas (Admin, Customer) with their controllers, views, and identity pages.
- **Data Access Layer (DAL):** Includes DbContext, database migrations, and repositories.
- **Models Layer:** Contains models and view models for the application.
- **Utilities Layer:** Provides extensions and helper methods for standardizing roles and sending emails.

2. Web Layer Details

Areas:

- **Admin Area:** Contains controllers and views for content management (CRUD for categories/products).
- **Customer Area:** Contains controllers and views for displaying products based on availability.
- **Identity:** Includes registration and login with client-side and server-side validations.
- **Shared Views:** Contains master layout and reusable partial views.

3. Data Access Layer Details

- **DbContext:** Configured with EF Core for migrations and seeding the database.
- **Repositories:** Interfaces and concrete implementations for generic and specific operations:
 - IGenericRepository
 - ICategoryRepository
 - IProductRepository
 - IUnitOfWork

4. Models Layer Details

- **Entities:**
 - **Product:** Includes fields like name, price, creation date, etc.
 - **Category:** Categorizes products.
 - **ApplicationUser:** Extends IdentityUser for user management.
- **View Models:** Used for transferring data between layers instead of exposing raw entities.

5. Utilities Layer Details

- **Extensions:**
 - Role standardization methods.

- Email sender for handling registration.
-

Backend Implementation

1. Database Design and Models

- Created ERD diagram and modeled entities (Product, Category, ApplicationUser).
- Used EF Core for migrations and seeding initial data.
- Database tables:
 - Categories (seeded on migration).
 - Products (CRUD-enabled via repository).
 - Identity tables for users and roles.

2. Repository Pattern

Implemented **Repository Pattern** for loose coupling and database abstraction:

- Generic repository for common operations.
- Specific repositories for Category and Product.
- Unit of Work for combining multiple operations in a transaction.

3. Category Management

CRUD Operations for **Categories**:

- Server-side and client-side validations (Data Annotations, JavaScript).
- Notifications using **Toastr**.
- Partial views for reusable UI elements.

4. Product Management

CRUD Operations for **Products**:

- File upload for images (stored dynamically, not as static files).
- Rich Text Editor for product descriptions.

- **Sweet Alerts** for confirmation prompts.

5. Authorization

Role-based authorization using **Identity**:

- **Admins**: Full access to content management.
 - **Customers**: Restricted to viewing available products.
-

Frontend Implementation

1. Theme and Styling

- Integrated **Bootswatch** for a polished UI.
- **Bootstrap Icons** for intuitive navigation.

2. Dynamic Features

- **Client-Side Validation**: Enhanced UX with instant feedback.
- **Pagination**: Implemented for product listing pages using JavaScript.
- **Wildcard Search**: Users can search for products by keywords.

3. Notifications and Alerts

- **Toastr** for real-time success/error notifications.
- **Sweet Alerts** for delete confirmations and critical actions.

4. User Journey

- **Customer Flow**:
 - Registers with email, password, and address details.
 - Views available products based on time.
 - Adds products to cart (feature mitigation).
- **Admin Flow**:
 - Accesses Content Management from the navbar.

- Manages categories and products with CRUD operations.
-

Identity and Authorization

Extended Identity:

- Added additional fields to **ApplicationUser** (e.g., Name, Address).
- Scaffolded Identity pages for **Registration** and **Login**.

Role Assignment:

- **Admins** manage all content.
 - **Customers** access only current-time products.
-

Advantages of the Approach

- **Scalability:** N-tier architecture ensures modularity and allows future scaling.
 - **Security:** Role-based access and validations enhance application security.
 - **Maintainability:** Repository pattern and Unit of Work simplify database interactions.
 - **Reusability:** Partial views and helper methods reduce code duplication.
-

Best Practices and Defenses

- **Use of View Models:** Avoids exposing database entities directly to the frontend.
- **Repository Pattern:** Decouples data access logic from business logic.
- **Validation:** Combines server-side (Data Annotations) and client-side (JavaScript) for robust data integrity.
- **Notifications:** Real-time feedback improves user experience.
- **Image Uploads:** Dynamic file handling for flexibility in managing product images.

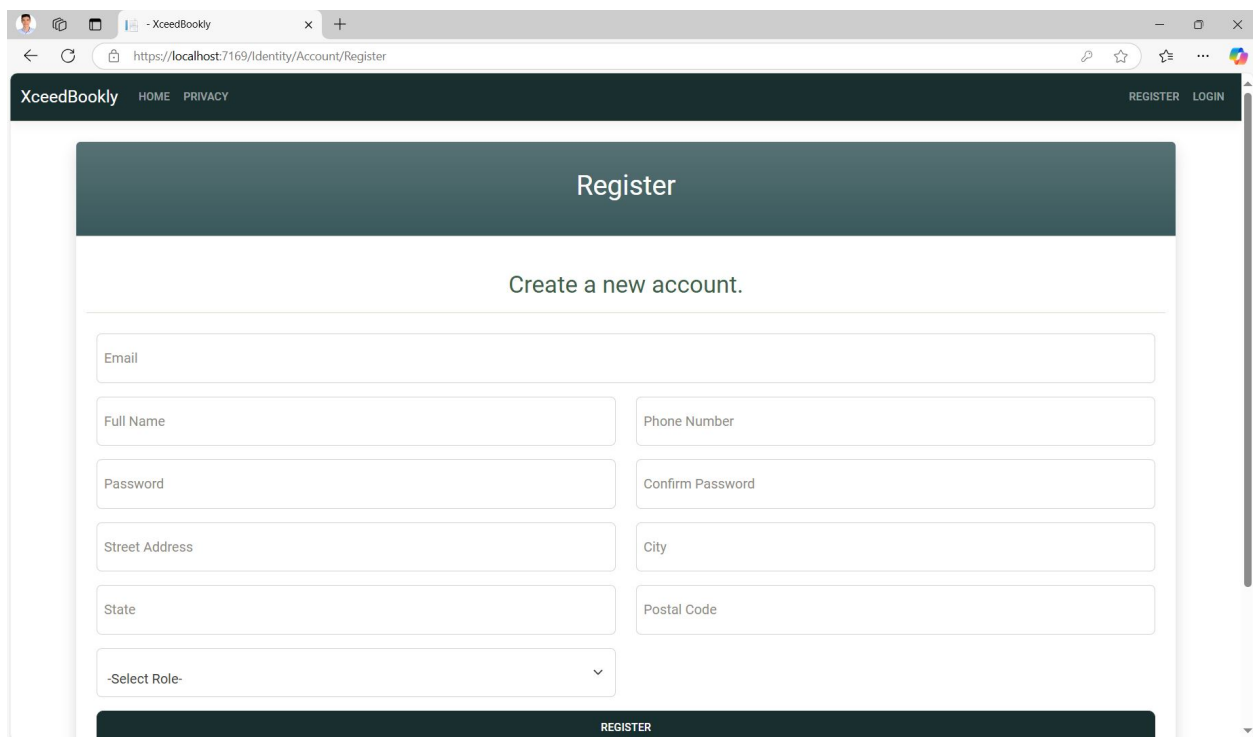
Dynamic Features

- **Custom Validations:** Prevents invalid data submission.
- **Sweet Alerts and Toastr:** Provides visual cues for user actions.
- **Projections in EF Core:** Optimizes data fetching for performance.

System Screens

1. Register Screen

The **Register Screen** allows users to create a new account with basic details such as email, password, and address.



The screenshot shows a web browser window with the URL `https://localhost:7169/Identity/Account/Register`. The page title is "XceedBookly" and the navigation bar includes "HOME" and "PRIVACY" links. The main heading is "Register" with the subtext "Create a new account." The form contains the following fields:

- Email
- Full Name
- Phone Number
- Password
- Confirm Password
- Street Address
- City
- State
- Postal Code
- Select Role-

A "REGISTER" button is located at the bottom right of the form.

2. Login Screen

The **Login Screen** provides users with fields for email and password, along with the option to reset their password.

XceedBookly HOME PRIVACY REGISTER LOGIN

Log in

Use a local account to log in.

Email
AdminXceed@gmail.com

Password

☐ Remember me?

LOG IN

[Forgot your password?](#) [Register as a new user](#) [Resend email confirmation](#)

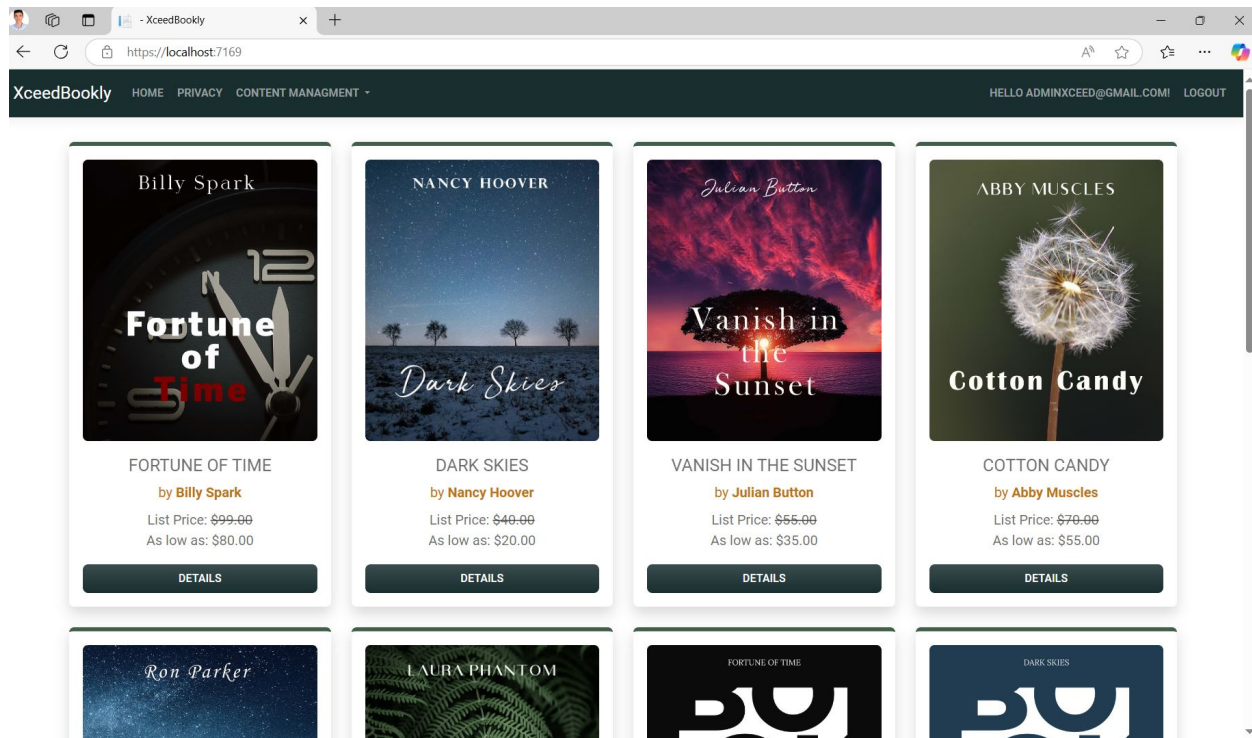
or

Use another service to log in.

There are no external authentication services configured. See this [article about setting up this ASP.NET application to support logging in via external services.](#)

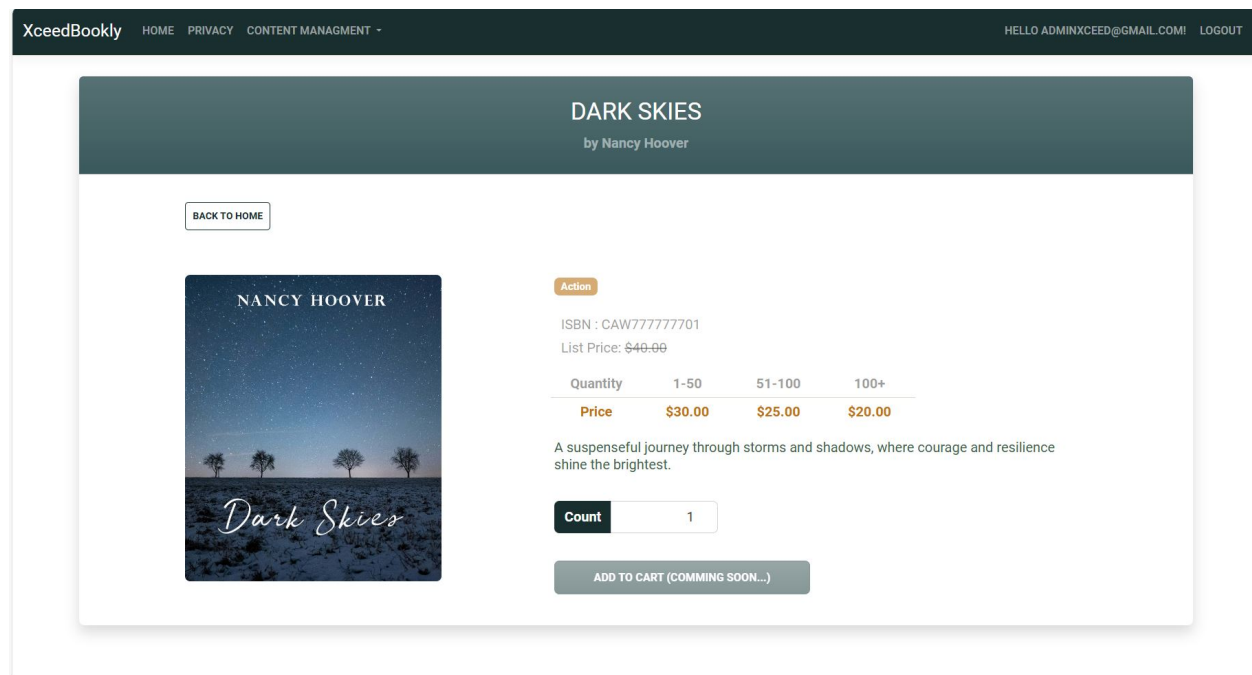
3. Home Screen

The **Home Screen** displays featured products to customers, with navigation options for browsing, logging in, or registering.



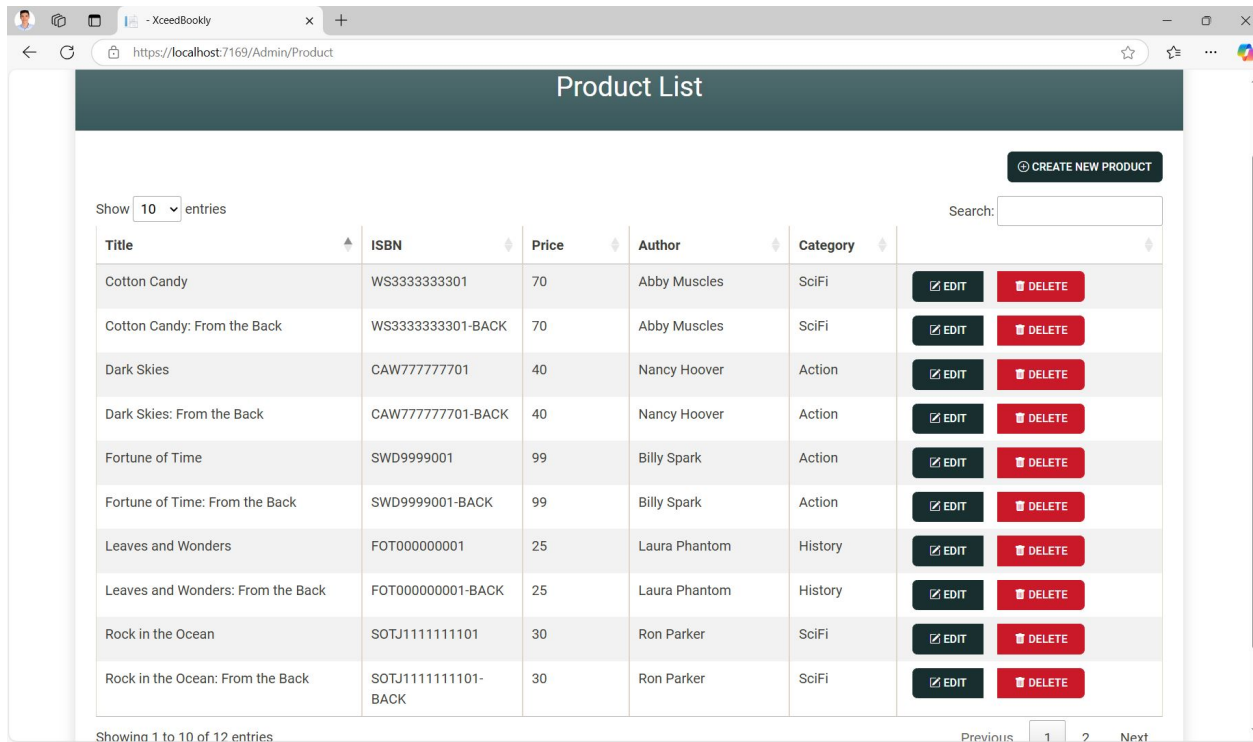
4. Product Details Screen

The **Product Details Screen** shows detailed information about a specific product, including images, description, and price.



5. Product CRUD Screen

The **Product CRUD Screen** enables administrators to create, update, or delete products with a clean interface and validation.



Title	ISBN	Price	Author	Category		
Cotton Candy	WS3333333301	70	Abby Muscles	SciFi	EDIT	DELETE
Cotton Candy: From the Back	WS3333333301-BACK	70	Abby Muscles	SciFi	EDIT	DELETE
Dark Skies	CAW777777701	40	Nancy Hoover	Action	EDIT	DELETE
Dark Skies: From the Back	CAW777777701-BACK	40	Nancy Hoover	Action	EDIT	DELETE
Fortune of Time	SWD99999001	99	Billy Spark	Action	EDIT	DELETE
Fortune of Time: From the Back	SWD99999001-BACK	99	Billy Spark	Action	EDIT	DELETE
Leaves and Wonders	FOT000000001	25	Laura Phantom	History	EDIT	DELETE
Leaves and Wonders: From the Back	FOT000000001-BACK	25	Laura Phantom	History	EDIT	DELETE
Rock in the Ocean	SOTJ1111111101	30	Ron Parker	SciFi	EDIT	DELETE
Rock in the Ocean: From the Back	SOTJ1111111101-BACK	30	Ron Parker	SciFi	EDIT	DELETE

Showing 1 to 10 of 12 entries

Previous 1 2 Next

6. Edit Screen

The **Edit Screen** allows admins to modify the product details such as name, price, and description.

Update Product

Title
Cotton Candy

Description
A delightful story about sweet memories and the vibrant world of small-town festivals.

ISBN
WS333333301

Author
Abby Muscles

List Price
70

Price for 1-50
65

ABBY MUSCLES
Cotton Candy

7. Delete Screen Sweet Alert

The **Delete Screen** provides admins with a confirmation dialog before deleting a product.

CREATE NEW PRODUCT

Show 10 entries

Search:

Title	ISBN	Price	Author	Category	EDIT	DELETE
Cotton Candy	WS333333301	70	Abby Muscles	Fiction	EDIT	DELETE
Cotton Candy: From the Back	WS333333301-BACK	70	Abby Muscles	Fiction	EDIT	DELETE
Dark Skies	CAW77777777	25	Laura Phantom	History	EDIT	DELETE
Dark Skies: From the Back	CAW77777777-BACK	25	Laura Phantom	History	EDIT	DELETE
Fortune of Time	SWD99999999	30	Ron Parker	SciFi	EDIT	DELETE
Fortune of Time: From the Back	SWD99999999-BACK	30	Ron Parker	SciFi	EDIT	DELETE
Leaves and Wonders	FOT000000001	25	Laura Phantom	History	EDIT	DELETE
Leaves and Wonders: From the Back	FOT000000001-BACK	25	Laura Phantom	History	EDIT	DELETE
Rock in the Ocean	SOTJ111111101	30	Ron Parker	SciFi	EDIT	DELETE
Rock in the Ocean: From the Back	SOTJ111111101-BACK	30	Ron Parker	SciFi	EDIT	DELETE

Are you sure?
You won't be able to revert this!

Yes, delete it! Cancel

8. pagination and Search

Show entries

Search:

Title	ISBN	Price	Author	Category	
Cotton Candy	WS3333333301	70	Abby Muscles	SciFi	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Cotton Candy: From the Back	WS3333333301-BACK	70	Abby Muscles	SciFi	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Dark Skies	CAW777777701	40	Nancy Hoover	Action	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Dark Skies: From the Back	CAW777777701-BACK	40	Nancy Hoover	Action	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Fortune of Time	SWD9999001	99	Billy Spark	Action	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Fortune of Time: From the Back	SWD9999001-BACK	99	Billy Spark	Action	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Leaves and Wonders	FOT000000001	25	Laura Phantom	History	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Leaves and Wonders: From the Back	FOT000000001-BACK	25	Laura Phantom	History	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Rock in the Ocean	SOTJ111111101	30	Ron Parker	SciFi	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE
Rock in the Ocean: From the Back	SOTJ111111101-BACK	30	Ron Parker	SciFi	<input checked="" type="checkbox"/> EDIT <input type="checkbox"/> DELETE

Showing 1 to 10 of 12 entries

Previous 2 Next

How To Run The System

Setup Instructions

Follow these steps to get the application up and running:

```

```bash
Navigate to the Project Folder
cd ProductCatalogWebApp

Update the Database Connection String
Open appsettings.json and set the connection string for your SQL Server.

Run Migrations to apply schema changes
dotnet ef migrations add InitialMigration
dotnet ef database update

```

```
Start the application
dotnet run
```