

Local Crafts Store Project Documentation

Organization: Artikeys Company

Developed By: Nour Othman

Date: 26/1/2025

Local Crafts Store Project

Project Overview

The Local Crafts Store project is a full-stack web application designed to bring together artisans selling handcrafted goods and buyers looking for unique products. The platform offers personalized features for sellers to manage their online presence, buyers to shop seamlessly across multiple sellers, and administrators to oversee the marketplace efficiently.

The system is highly modular, using the MVC architecture to ensure clean separation of responsibilities and maintainability. Key roles in the system include **Sellers**, **Buyers**, **Admin**, and an automated **Robot** that handles messaging.

Key Features

Sellers

- **Account Creation and Login:** Sellers can create accounts and log in to manage their activities.
- **Personalized Store Pages:** Each seller has a unique page where they can display their products, complete with product details such as images, descriptions, and prices.
- **Order Management Dashboard:** Sellers can view orders placed by buyers. Each order item can be approved or rejected, with updates reflected in real-time.
- **Automated Notifications:** Sellers receive automated messages based on the status of their orders (e.g., when an order is approved, rejected, or pending).
- **Product Management:** Sellers can add, edit, or remove products directly from their dashboards.

Buyers

- **Multi-Seller Orders:** Buyers can add products from multiple sellers into a single order. Each product is routed to its respective seller for processing.
- **Order Status Updates:** Buyers are notified about the progress of their orders through automated messages (e.g., pending approval, approved, or rejected).
- **Browsing and Searching:** Buyers can explore products through categories or use the search functionality to find specific items.
- **Seamless Checkout:** A user-friendly checkout system that handles multiple seller transactions in a single order.

Admin Role

- **Category Management:** Admins can create, edit, or delete categories to keep the platform organized.
- **Seller Oversight:** Admins monitor seller accounts and have the authority to delete accounts or remove inappropriate products.
- **Order Monitoring:** Admins can view and oversee the status of all orders, ensuring smooth operations.
- **Product Moderation:** Ensures that products comply with marketplace policies.

Robot Role

- **Automated Messaging:** The system includes a robot that automates communication with users. The robot:
 - Notifies sellers about order statuses.
 - Updates buyers when their orders are approved, rejected, or still pending.
 - This automation ensures timely updates and minimizes manual intervention.
-

Technical Implementation

MVC Architecture

The project adopts the **Model-View-Controller (MVC)** architecture:

- **Models:** Define the data and business logic. Examples include `Product.cs`, `Seller.cs`, and `Order.cs`.
- **Views:** Handle user interfaces, using `.cshtml` files to display content dynamically.
- **Controllers:** Act as intermediaries between models and views, handling user inputs and returning appropriate responses.

ViewModels

ViewModels play a crucial role in transferring data between the back-end and the front-end. Examples include:

- **CheckoutViewModel:** Simplifies data required during the checkout process.
- **SellerProductsViewModel:** Provides data for managing a seller's product list.
- **SearchResultsViewModel:** Returns results for buyer queries.

Components

Reusable components improve development efficiency and maintain a consistent UI:

- **CategoryViewComponent:** Dynamically loads and displays product categories.
- **SellerViewComponent:** Renders seller-specific data, such as their profile or featured products.

Areas

To ensure modularity and a clean structure, the application uses **Areas**:

- **Admin Area:**
 - Controllers for managing categories, seller accounts, and orders.
 - Views for handling administrative tasks like reviewing pending items or approving/rejecting seller requests.
- **Main Area:**
 - User-facing features such as browsing products, purchasing items, and managing accounts.

Database

- The application uses **Entity Framework Core (EF Core)** for ORM functionality.
- Migrations ensure smooth updates to the database schema without data loss.
- The database includes tables for products, sellers, orders, categories, and user accounts.

Security

- **Role-Based Access Control (RBAC)** ensures secure access:
 - Buyers can only view and purchase products.
 - Sellers manage their own products and orders.
 - Admins have full control over the platform, including moderation and account management.
- **Authentication:** Handles user logins securely, ensuring only authenticated users can access their respective dashboards.

Automated Features

The built-in robot automates key processes:

- Sends messages to buyers and sellers regarding order statuses.
 - Enhances user experience by providing quick updates, minimizing manual follow-up.
-

Code Highlights

Sellers

- Controllers such as `SellersController.cs` handle seller-specific operations like managing products and processing orders.
- Views include `Index.cshtml`, `Details.cshtml`, and others, providing a clear interface for sellers to interact with.

Buyers

- The `CartController.cs` and `CheckoutController.cs` manage buyer operations, including adding items to a cart, viewing the cart, and completing purchases.
- Views like `Cart/Index.cshtml` and `Cart/Checkout.cshtml` provide buyers with an easy-to-use UI.

Admin

- The Admin area contains controllers such as `CategoriesController.cs` and `OrdersController.cs` to handle platform-wide operations.
- Views include templates for managing categories, viewing pending items, and handling seller accounts.

Robot Integration

- The robot logic resides in the back-end and interacts with the controllers to send notifications dynamically.
- It uses pre-defined templates to deliver concise and relevant updates.

Conclusion

The Local Crafts Store project offers a comprehensive solution for connecting artisans with buyers while maintaining a high level of organization and automation. The use of MVC, ViewModels, and reusable components ensures a scalable and maintainable platform. Additionally, the automated robot enhances user experience by providing timely updates, making the platform user-friendly for all stakeholders.