ANGEL STORE

AN ANGULAR APP

BY – NELLORE LAHARI

To Create a compressive single page application – SPA using Angular .

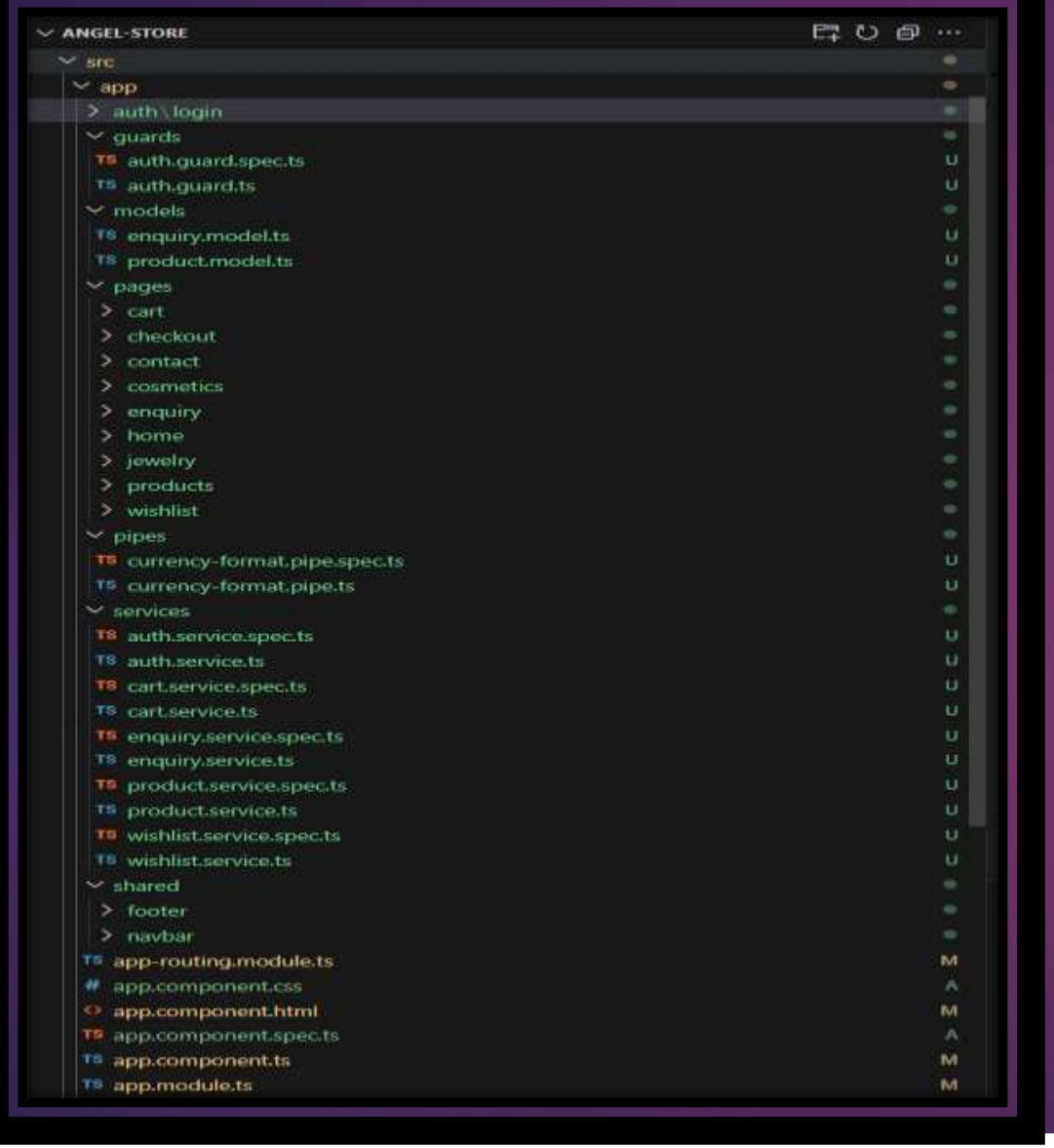
Angel Store (Online Cosmetics & Jewelry Shopping App)
Project Overview:

- **Angel Store** is a **Single Page Application (SPA)** developed using **Angular**. It is an online store for **jewelry & cosmetics**, featuring various pages like **Home, Products, Wishlist, Cart, Checkout, Login, and Contact Us**. The project incorporates key Angular concepts such as **components, routing, services, observables, API calls, form handling, pipes, local storage, and authentication with route guards**.
- **Step 1:**
  Created an angular project named **angel-store** and created the required **page components** like Home , Products , Cart , Wishlist , Checkout , Contact , Enquiry and Login

  Command for new project – **ng new angel-store –no-standalone**
  Command for new components – **ng g c page/componentname**
- **Step 2:**
  Created services for authentication , cart , enquiry , product , wishlist .

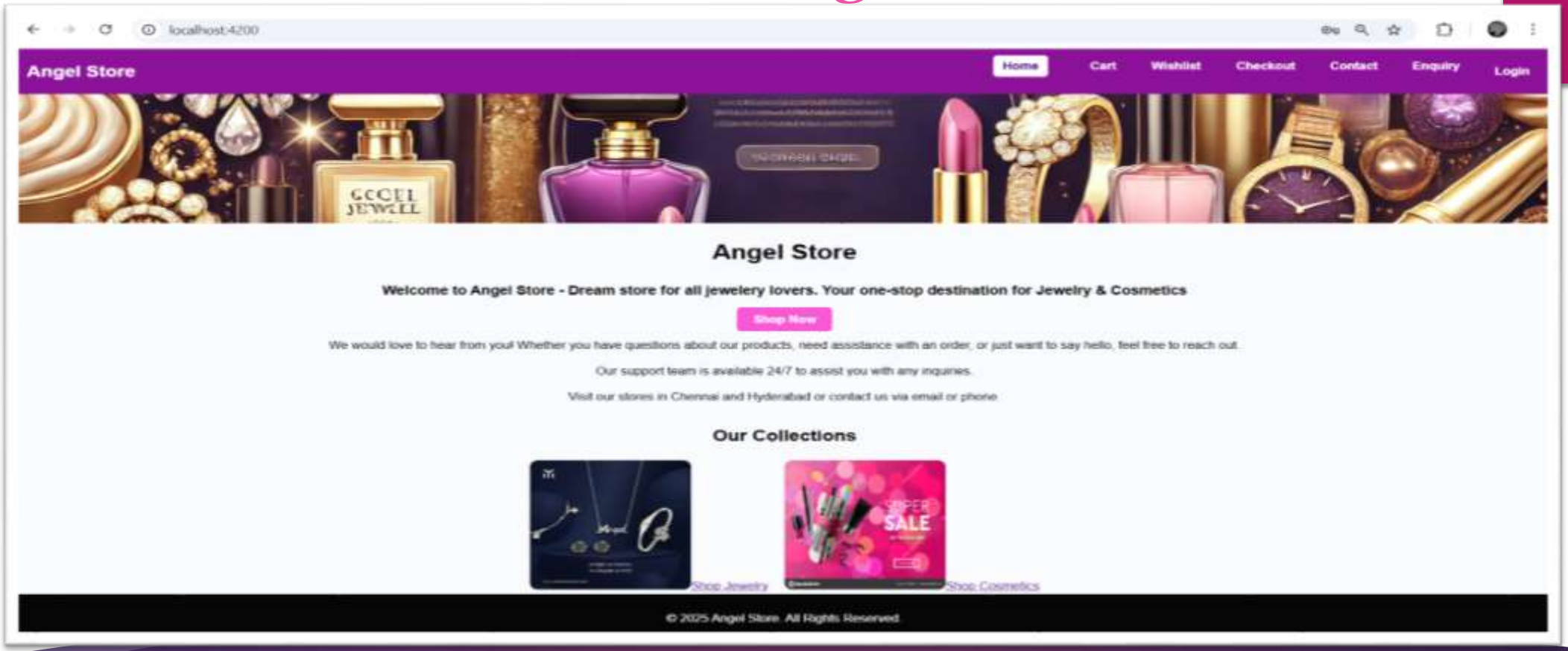
  Auth.service - Handles user **authentication** (login, signup, logout) and **route guarding.**
  cart.service - Manages the **cart functionality**, including adding/removing items and persisting data in local storage.
  enquiry.service- Processes **customer enquiries** submitted via the **contact form** and sends data to the backend.
  product.service- Fetches **product data** from an **API** and provides it to components.
  wishlist.service- Manages the **wishlist feature**, allowing users to **add/remove favorite products** and persist them.

Structure :



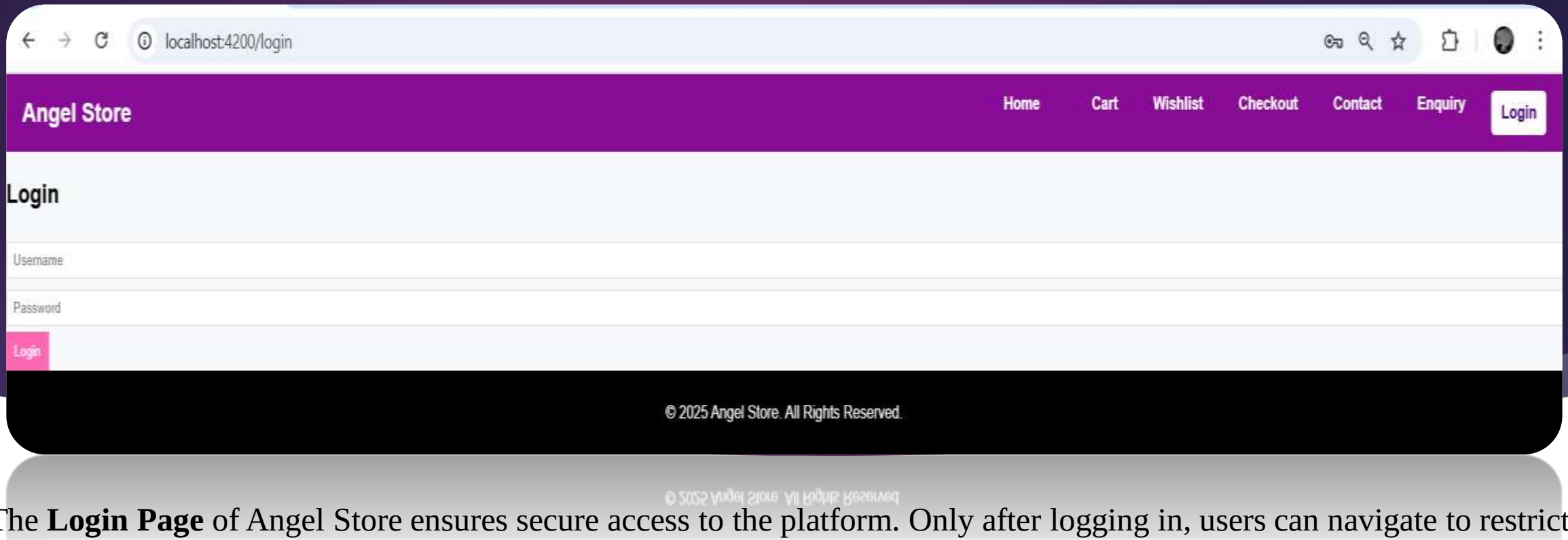
- **Step 4 :**
  Created guards for certain pages using **auth.guard.ts**
  •**Prevents unauthorized access** by checking if the user is logged in.
  •**Redirects users to the login page** if they try to access a protected route without authentication.
- **Step 5 :**
  Created **currency-format pipes**
  •**Product Listing & Details Pages** – To display product prices in a formatted currency style.
  •**Cart & Checkout Pages** – To show the total price of items in a user-friendly currency format.
- **Step 6 : API calls**
  used **api.json** to call the products list and user for login.
  product.service.ts fetches products from api.json
- **Step 7 : Form Handling**
  Created **contact form** using **Template based forms**
  Created **checkout form** using **Model based forms**

Home Page

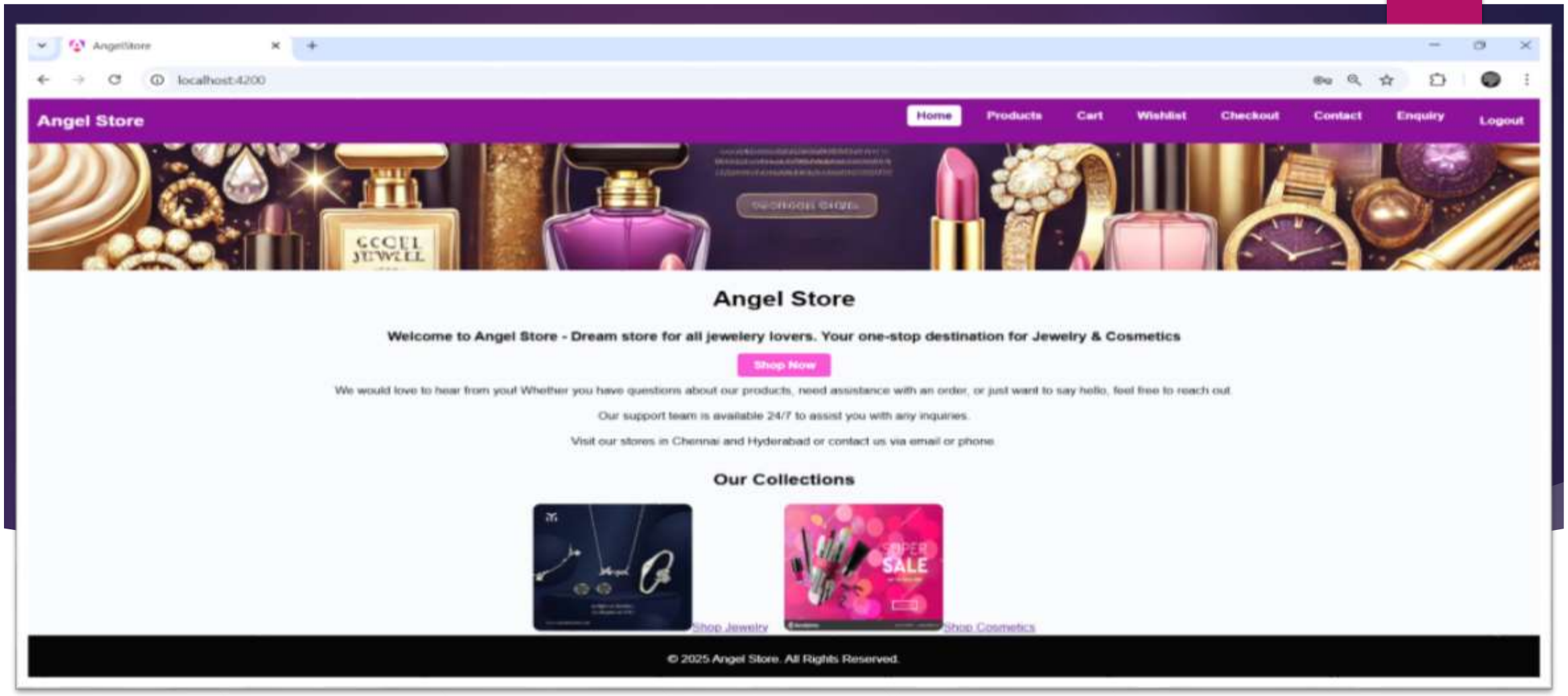


The **Angel Store** home page features a stylish and luxurious UI with a vibrant purple theme, perfect for a jewelry and cosmetics e-commerce store. A high-quality **banner** showcases premium products, creating an appealing first impression. The **navigation bar** provides easy access to key sections like Cart, Wishlist, and Checkout. A **welcome section** highlights the store's purpose, inviting users to explore. The page includes **Shop Jewelry & Shop Cosmetics** links for seamless browsing, along with a **Shop Now** button to boost

Login Page

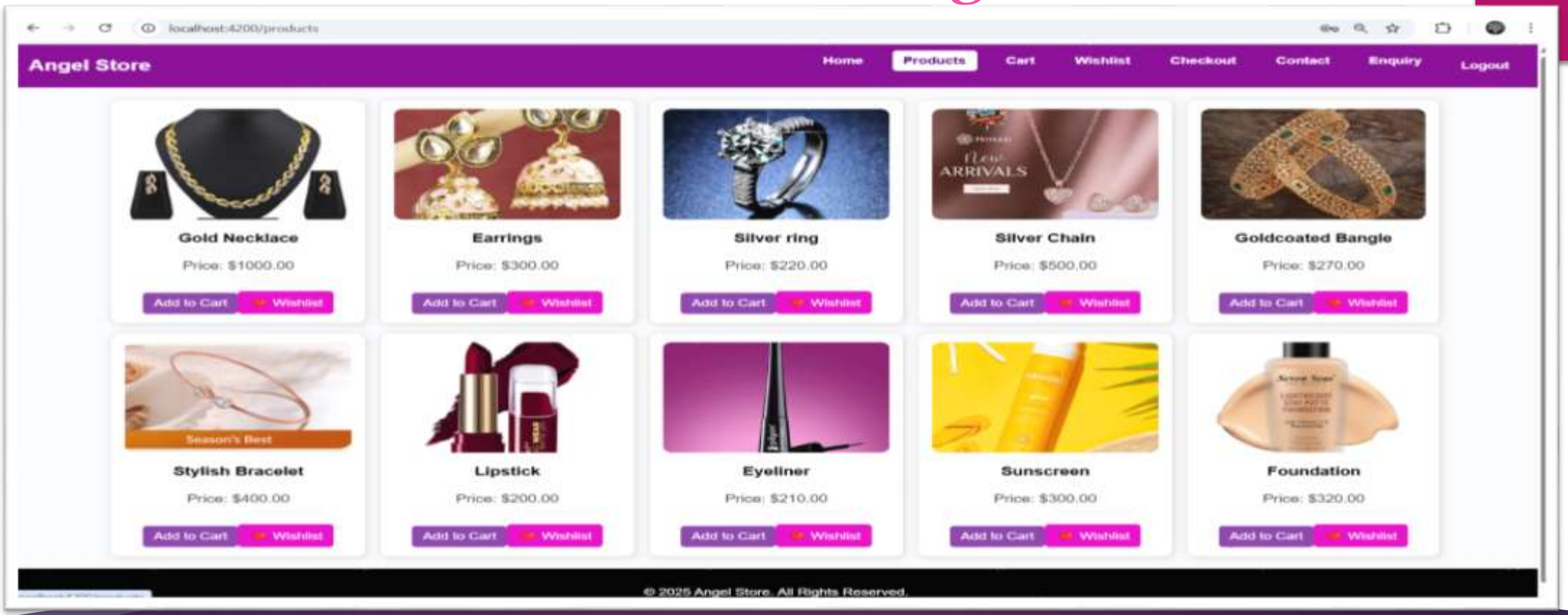


The **Login Page** of Angel Store ensures secure access to the platform. Only after logging in, users can navigate to restricted pages like **Cart, Wishlist, Checkout, and Products**, while **Home and Contact pages** remain accessible to all. If a user tries to access restricted pages without logging in, they are **redirected to the login page**. This is implemented using **Route Guards** in Angular to control page access. The login system enhances security and ensures a personalized shopping experience.

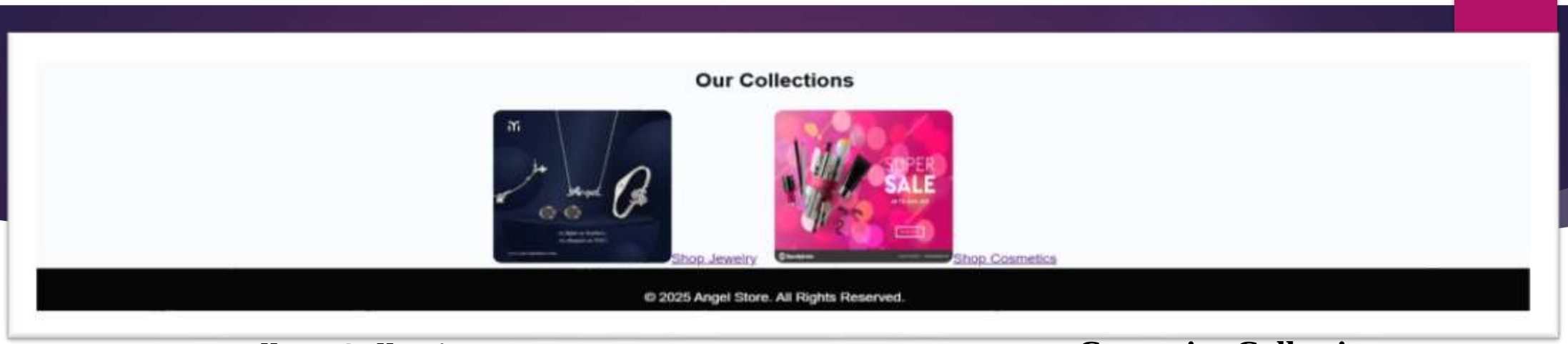


After logging in successfully , now the products page will be shown in the navigation bar and also as soon as login it will redirect to the products page

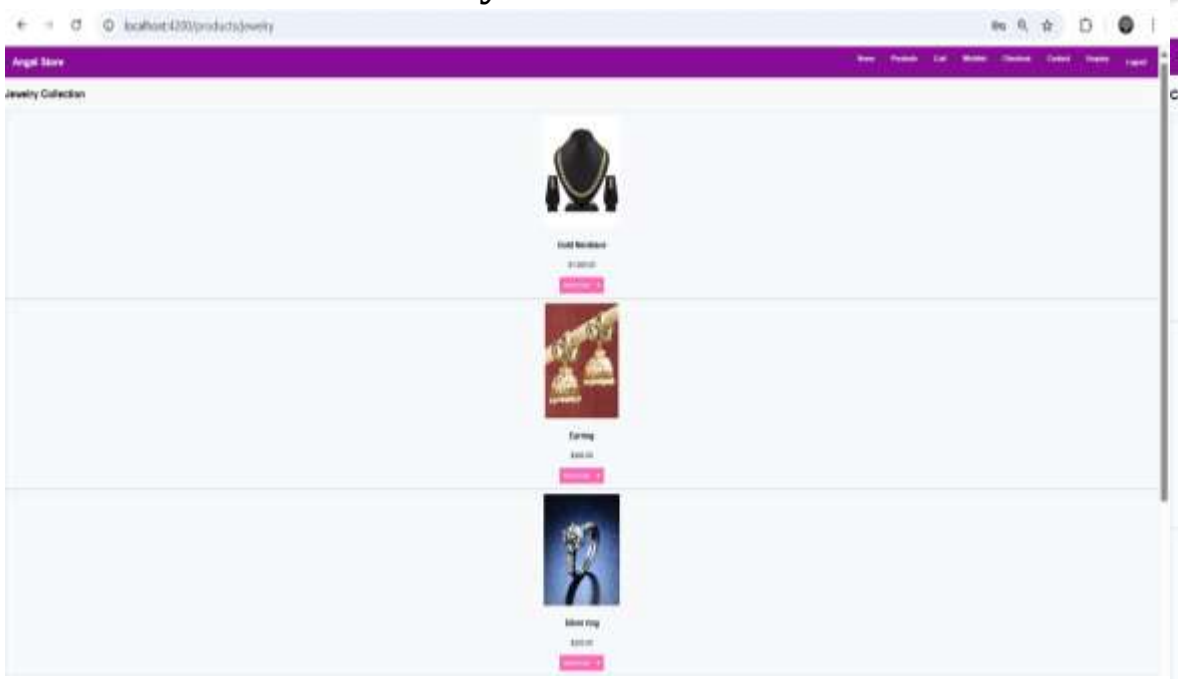
Products Page



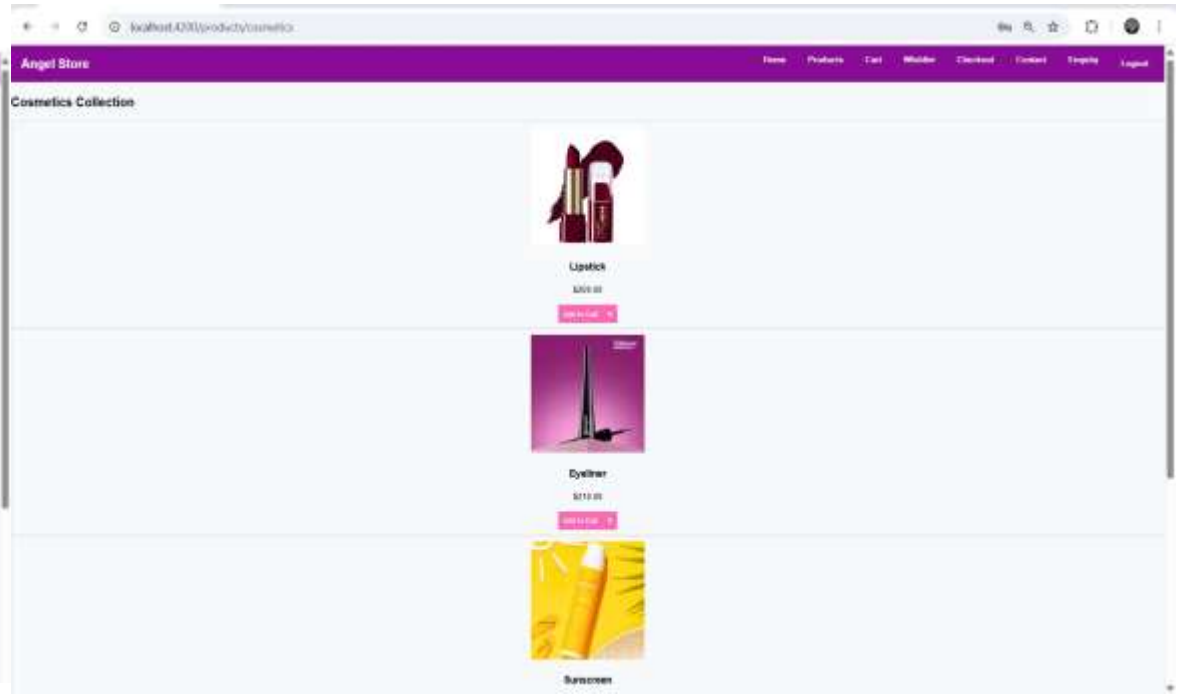
The **Product Page** in Angel Store displays a variety of jewelry and cosmetic products available for purchase. Users can view detailed product information, including images, descriptions, and prices. This page is **restricted to logged-in users**, ensuring only authenticated users can browse and add items to their cart or wishlist. Products are dynamically fetched and displayed using Angular components and services. The page enhances the shopping experience with a clean, user-friendly interface.



Jewellery Collections

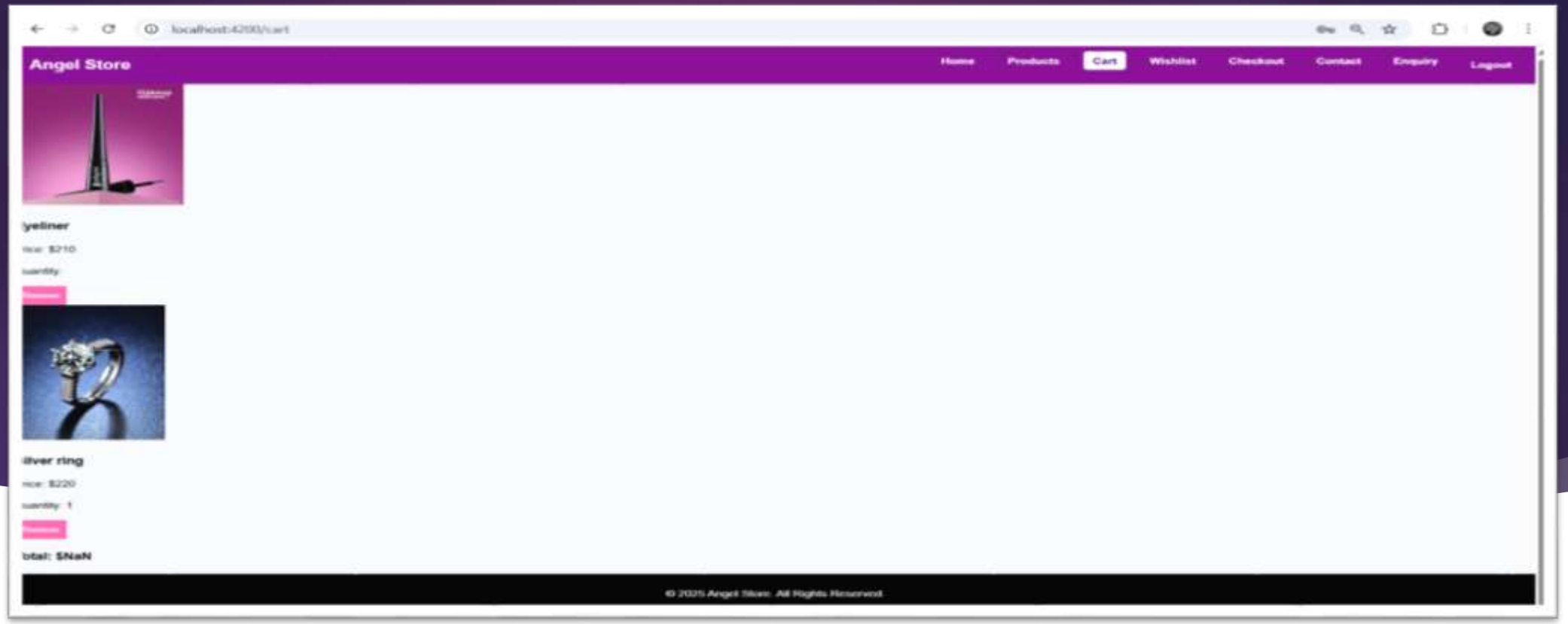


Cosmetics Collections



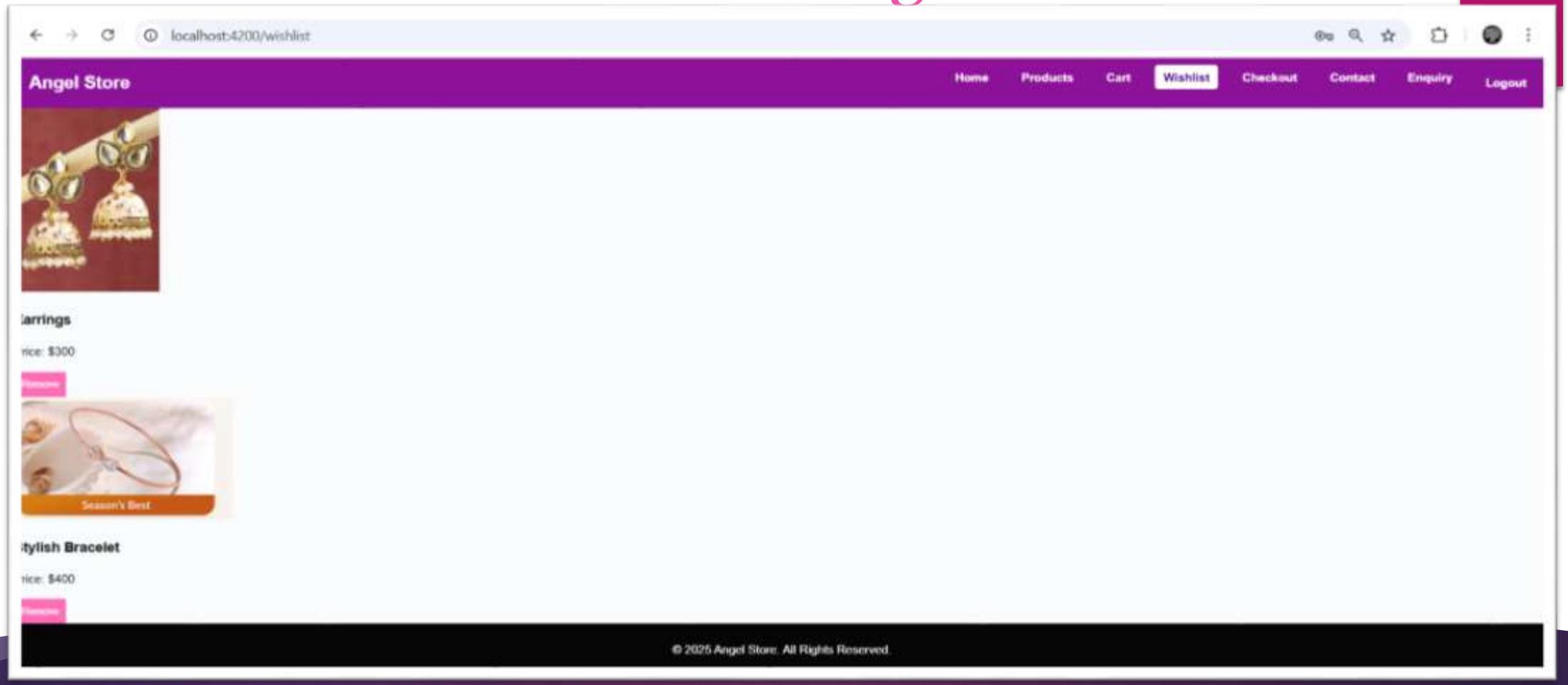
In Home Page , the 2 categories mentioned in the end of the page are navigated to that specific category of shopping that user specificly want to shop and as per their wish for easy access .

Cart Page



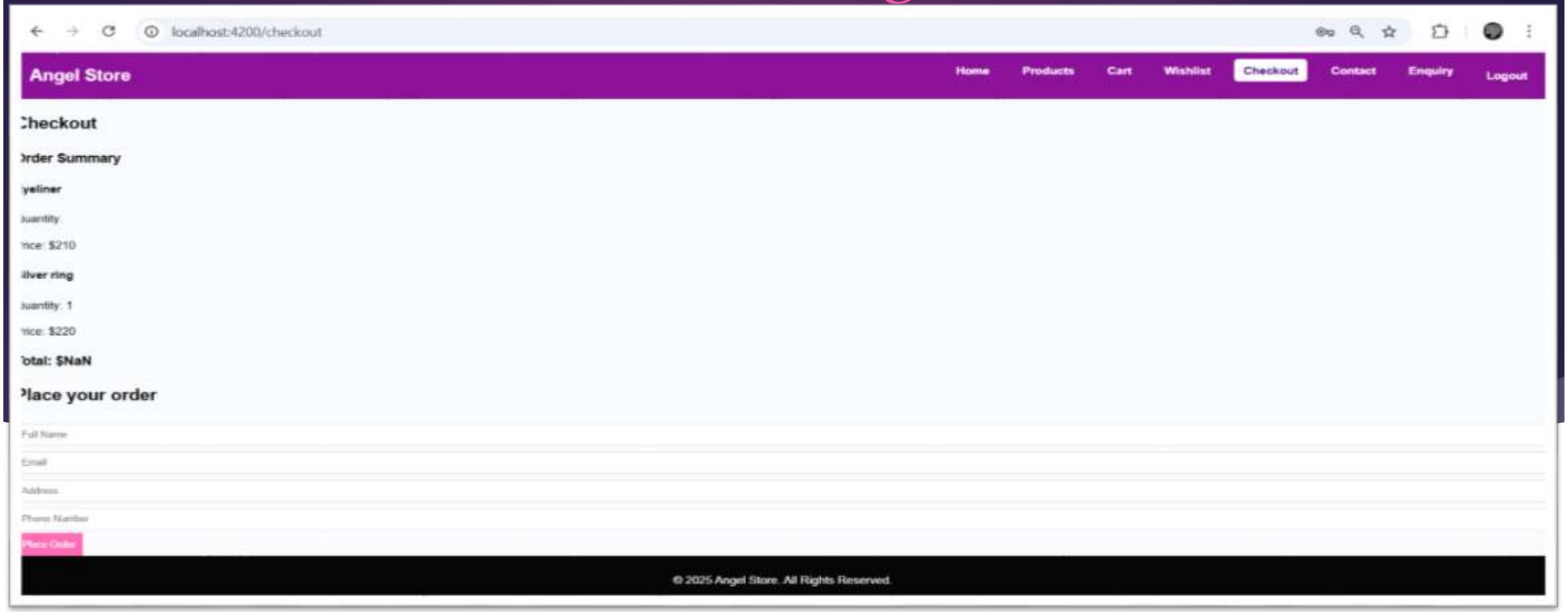
The **Cart Page** in Angel Store displays all the products added by the user for purchase. It allows users to update quantities, remove items, and view the total price dynamically. **Local Storage** is used to persist cart data, ensuring items remain in the cart even if the page is refreshed. This enhances the shopping experience by maintaining user selections until checkout.

Wishlist Page



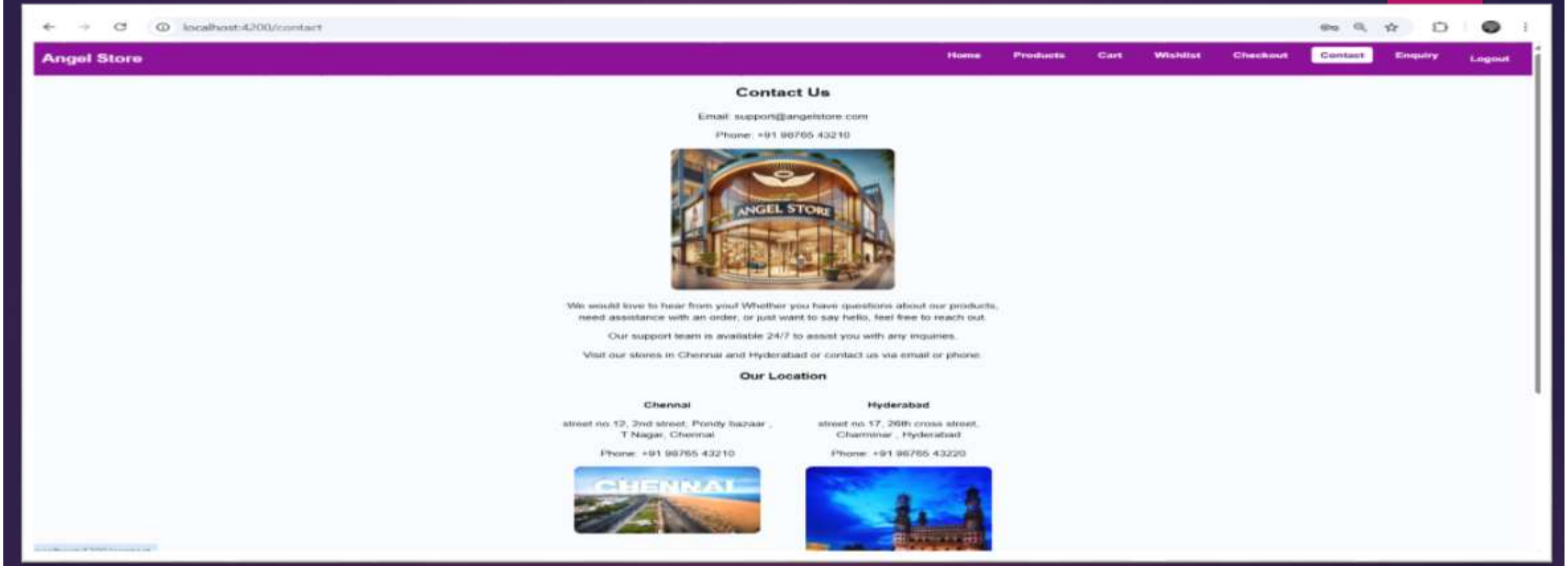
The **Wishlist Page** in Angel Store lets users save their favorite products for future purchases. It provides an option to move items to the cart or remove them from the list. **Local Storage** ensures that wishlist items are retained even after a page refresh. This feature enhances user convenience by allowing them to revisit their saved products anytime.

Checkout Page

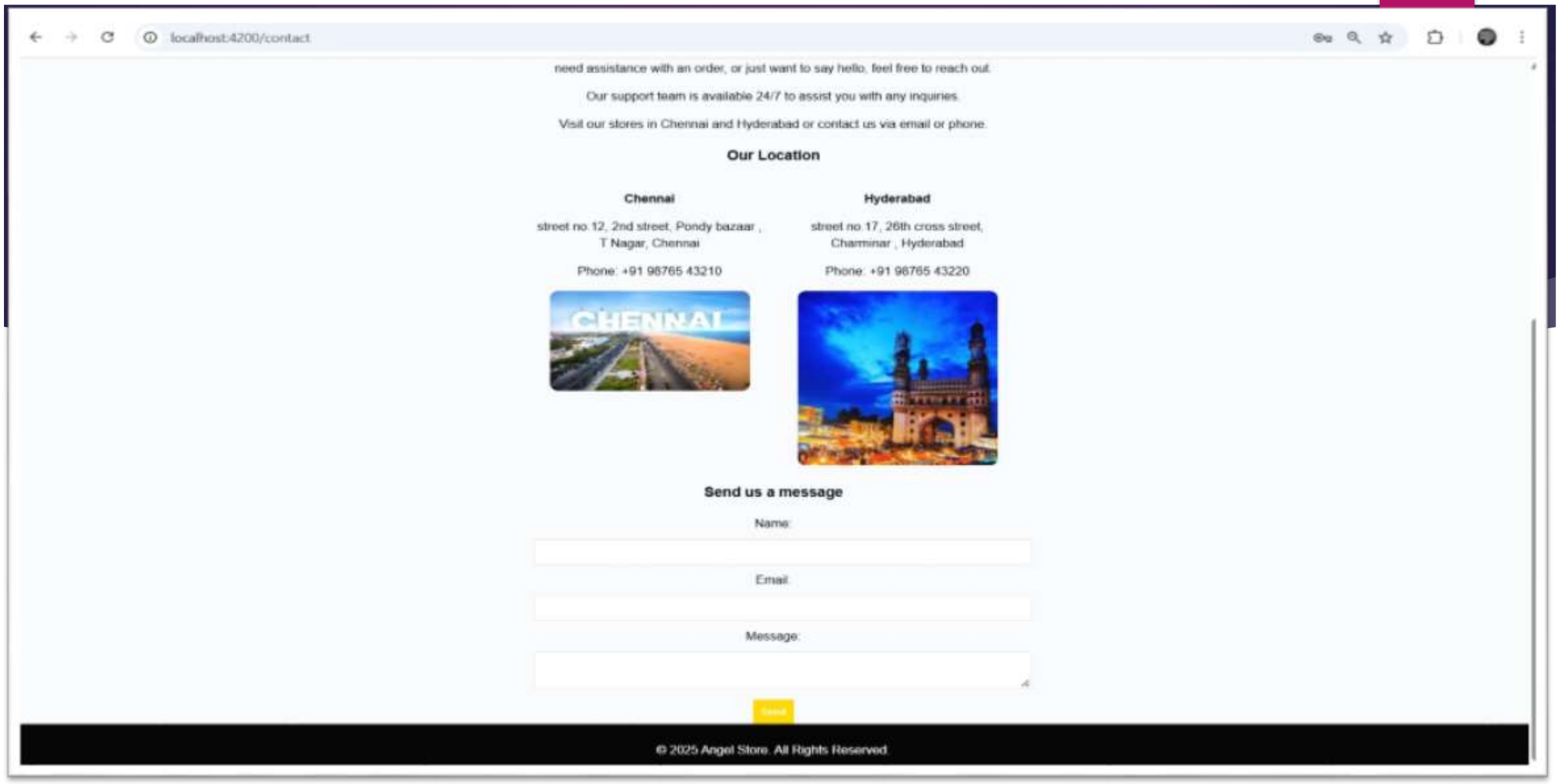


The **Checkout Page** in Angel Store allows users to review their selected items before making a purchase. It provides a seamless payment and order confirmation process. Secure transactions and a user-friendly interface ensure a smooth shopping experience.

Contact Page

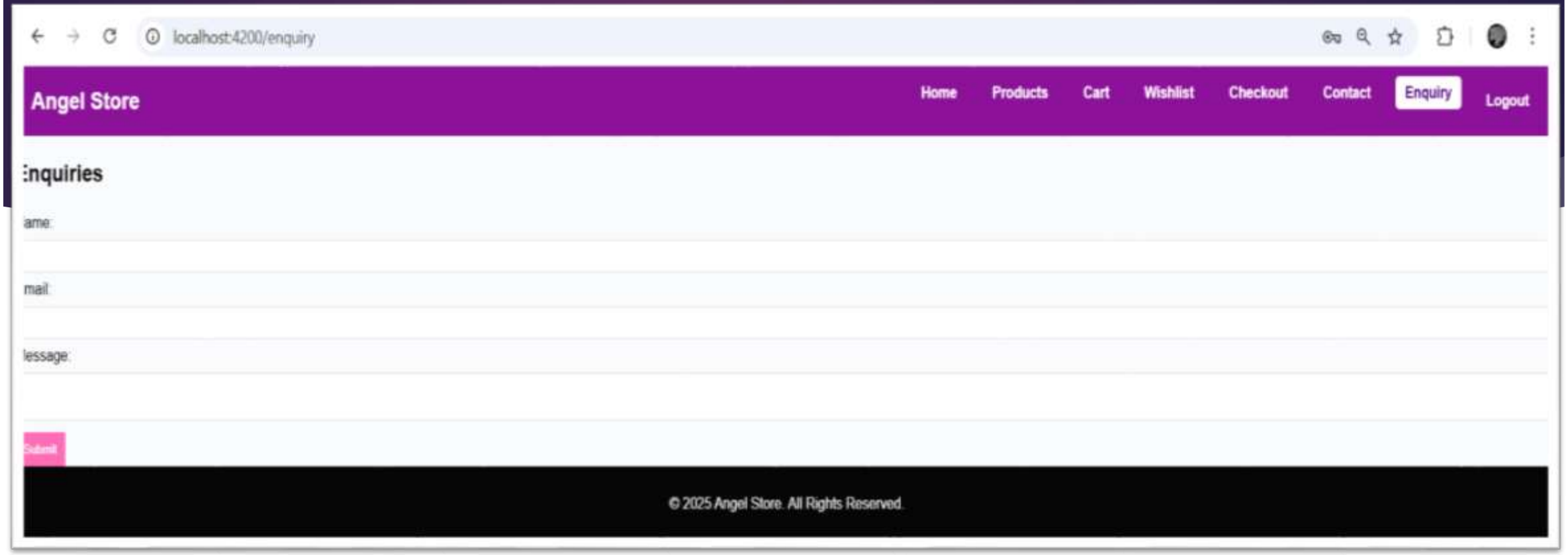


The **Contact Page** in Angel Store provides customers with essential details like store location, email, and phone number for inquiries. It includes an enquiry form where users can submit messages, which are stored in the backend. A visually appealing layout with a store image enhances the user experience. The page ensures easy communication between customers and the store.



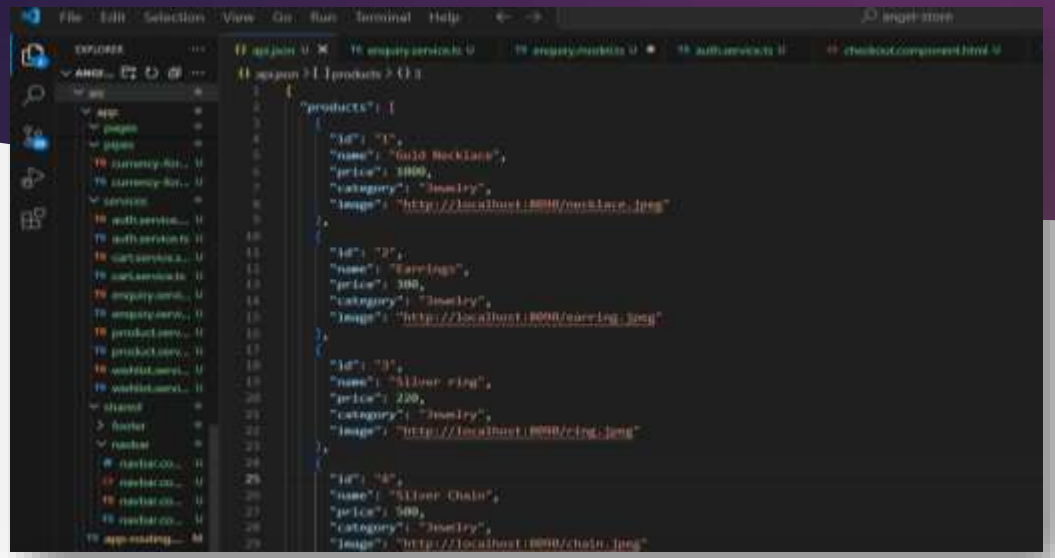
Contact page with template based forms .

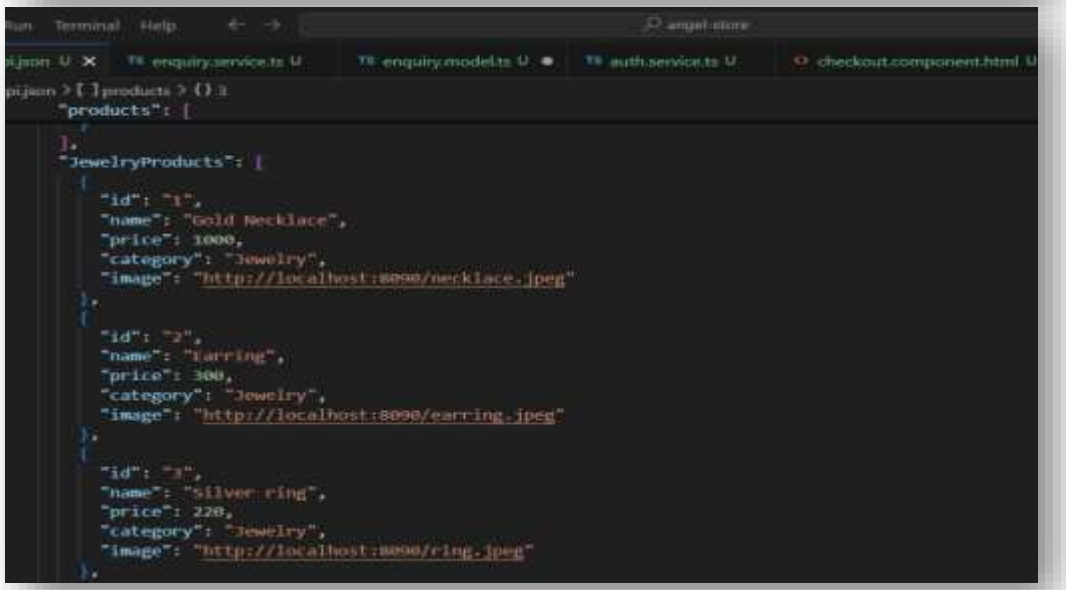
Enquiry Page

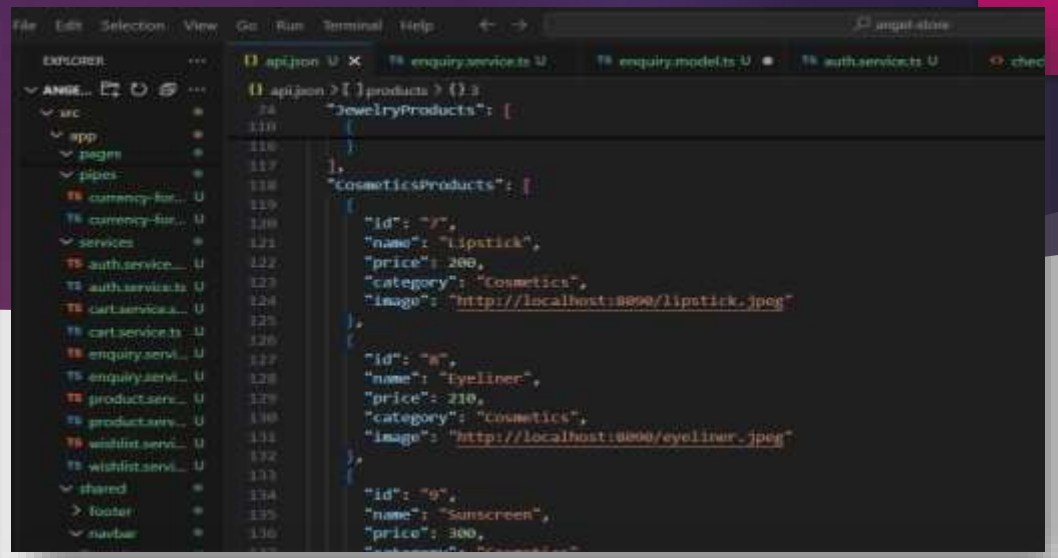


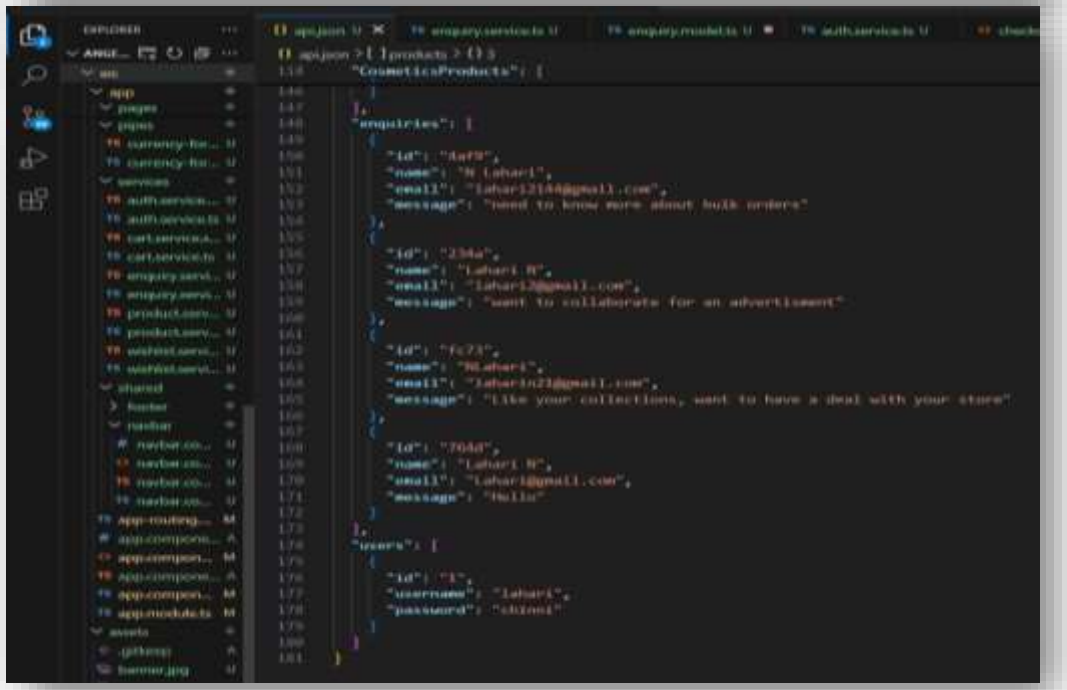
The **Enquiry Page** in Angel Store allows users to submit their queries through a form. The form uses **template-driven form handling** with ngModel for binding input fields like name, email, and message. On submission, the data is stored and processed for customer support. A confirmation message appears to assure users that their enquiry has been received.

Api.json









thank you