Angular Project: E-commerce Website – SheStyle

1. Components

Components are the building blocks of the Angular application. Each component represents a specific UI section, such as the product list, product details, shopping cart, and checkout. In our e-commerce website, we use multiple components to structure different sections of the website, ensuring modularity and ease of maintenance.

2. Parent-Child Components

The application follows a structured parent-child relationship between components. Ex: The product list component (parent) contains multiple product items (child). The child components receive data from the parent using @Input and send events back to the parent using @Output. This allows dynamic rendering and interaction between components, improving reusability.

3. Structural and Attribute Directives

Structural Directives like *ngIf, *ngFor, and *ngSwitch control the DOM structure dynamically. For example, *ngFor is used to loop through the list of products and display them in the UI.

Attribute Directives such as [ngClass] and [ngStyle] help in applying styles dynamically based on conditions, like highlighting a product when it is on sale.

4. @Input

@Input is used to pass data from a parent component to a child component. In the e-commerce website, this is used to send product details from the product list component to the individual product card component, enabling dynamic rendering of different products based on API responses.

5. Basic Routing

Angular's routing module is implemented to navigate between different sections of the website, such as the home page, product details, cart, and checkout. The router ensures seamless navigation by managing URL parameters and linking components dynamically. Lazy loading is also used for performance optimization.

6. Services

Services are used to share data and business logic across components. A ProductService is created to fetch product data from an API and provide it to multiple components. Similarly, a CartService is used to manage the shopping cart, ensuring that cart-related data persists across different pages.

7. Observables

Observables are extensively used to handle asynchronous data streams. The HttpClient module returns data as Observables when making API calls, ensuring smooth data fetching. Observables also enable real-time updates, such as reflecting cart changes immediately when a user adds or removes items.

8. API Calls

The application retrieves product data through an API (api.json). The HttpClientModule is used to send HTTP requests to fetch product listings, categories, and user details. API calls are managed inside services to keep the components clean and focused on UI rendering.

9. Template and Reactive Forms

Template-driven forms are used for simpler forms like the contact form or user login, where two-way binding with ngModel is implemented.

Reactive forms are used for complex forms like the checkout form, where form validation, custom validation rules, and dynamic form fields are required. This approach ensures better scalability and validation handling.

10. Pipes

Pipes are used to transform and format data before displaying it in the UI. Common pipes used in the application include:

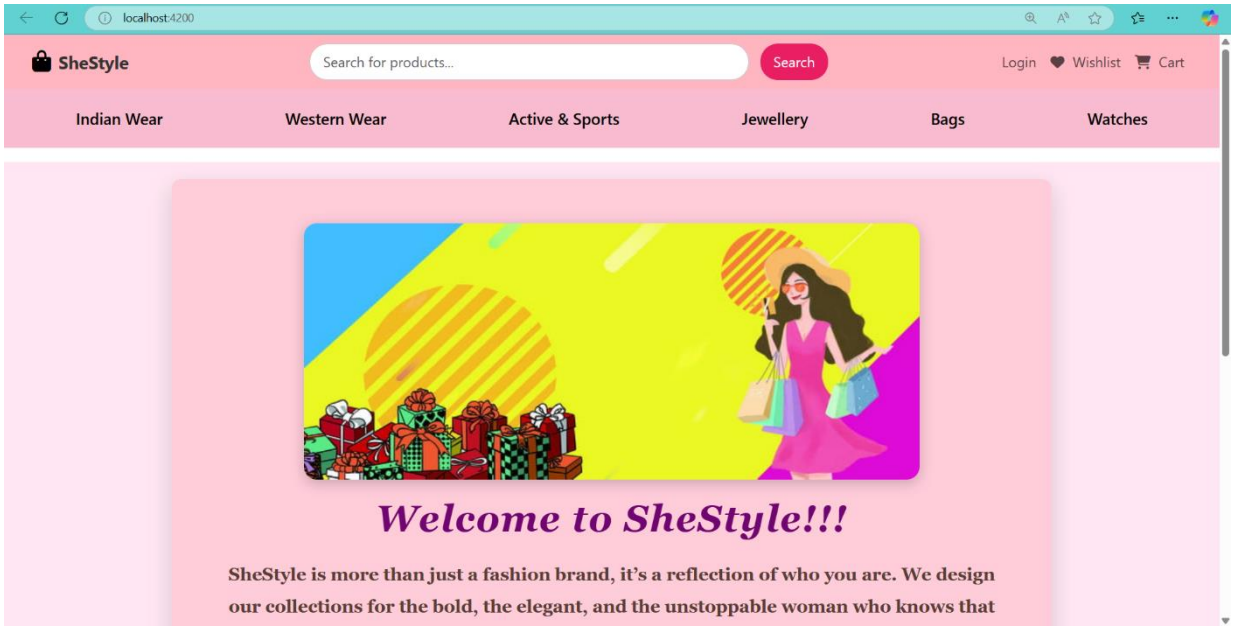
CurrencyPipe to format prices correctly

DatePipe to format order dates

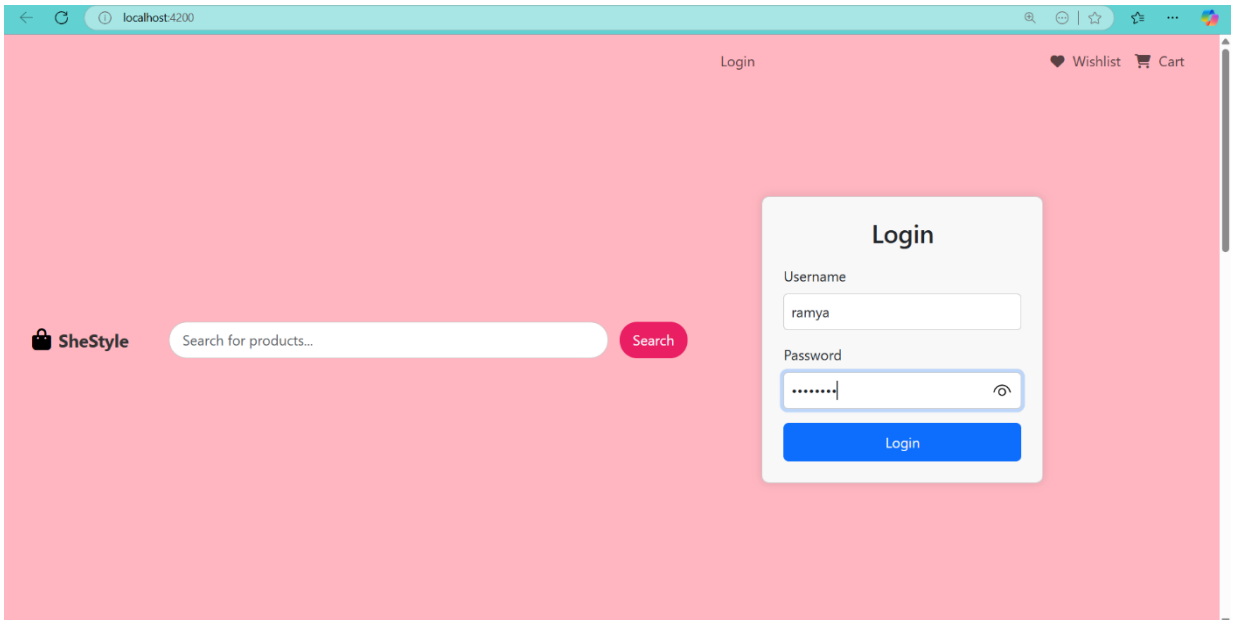
Custom Pipes such as a filter pipe to search and filter products dynamically based on user input

App Features:

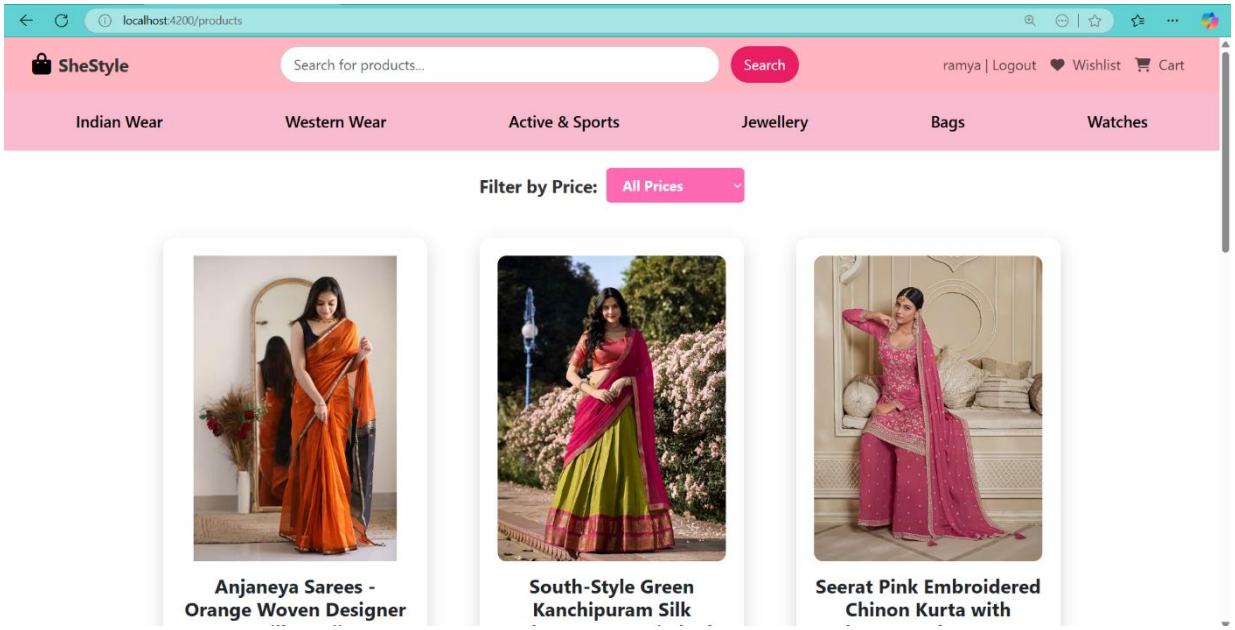
1.Home Screen:



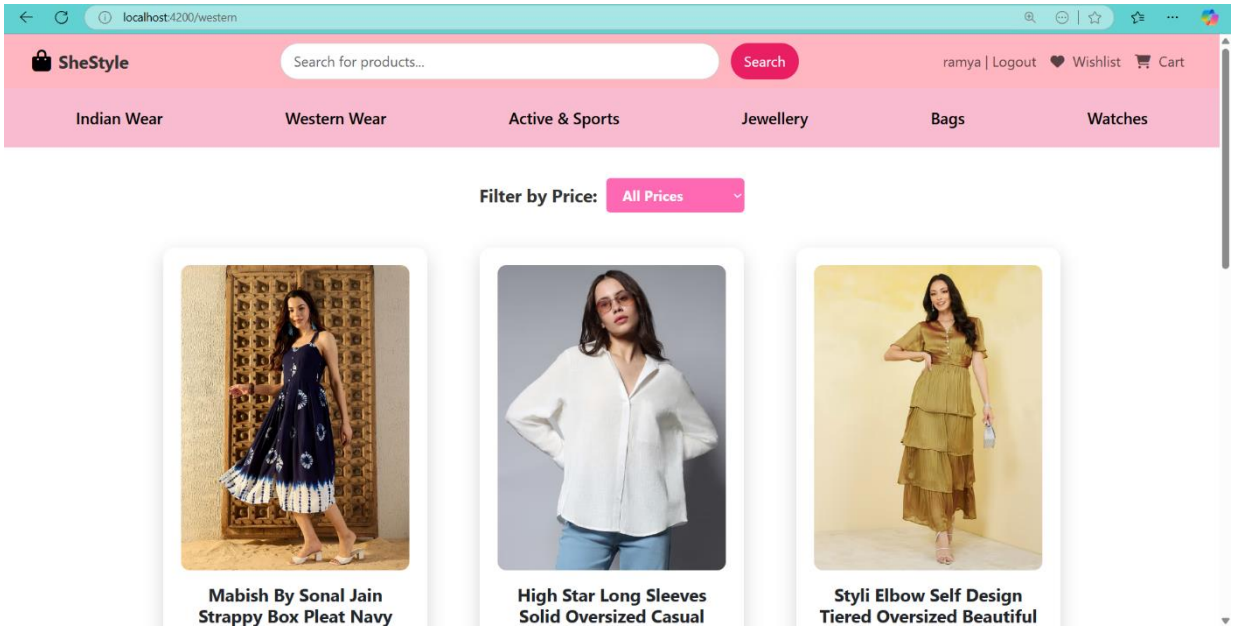
2.Login Page:



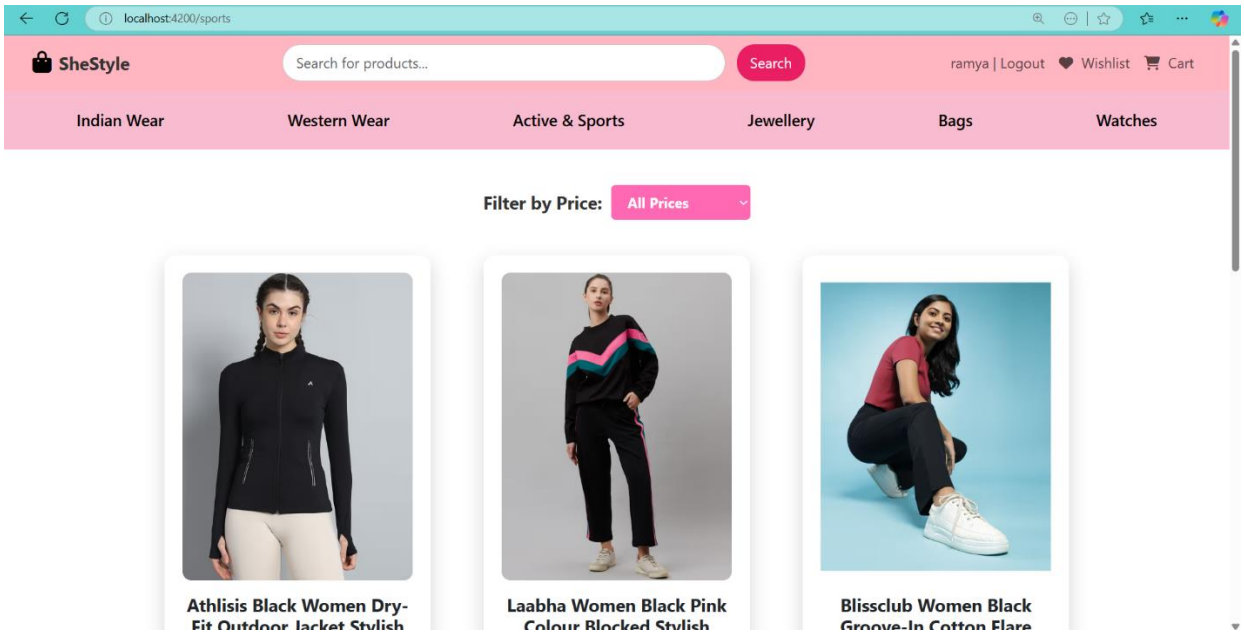
3.Indian Wear:



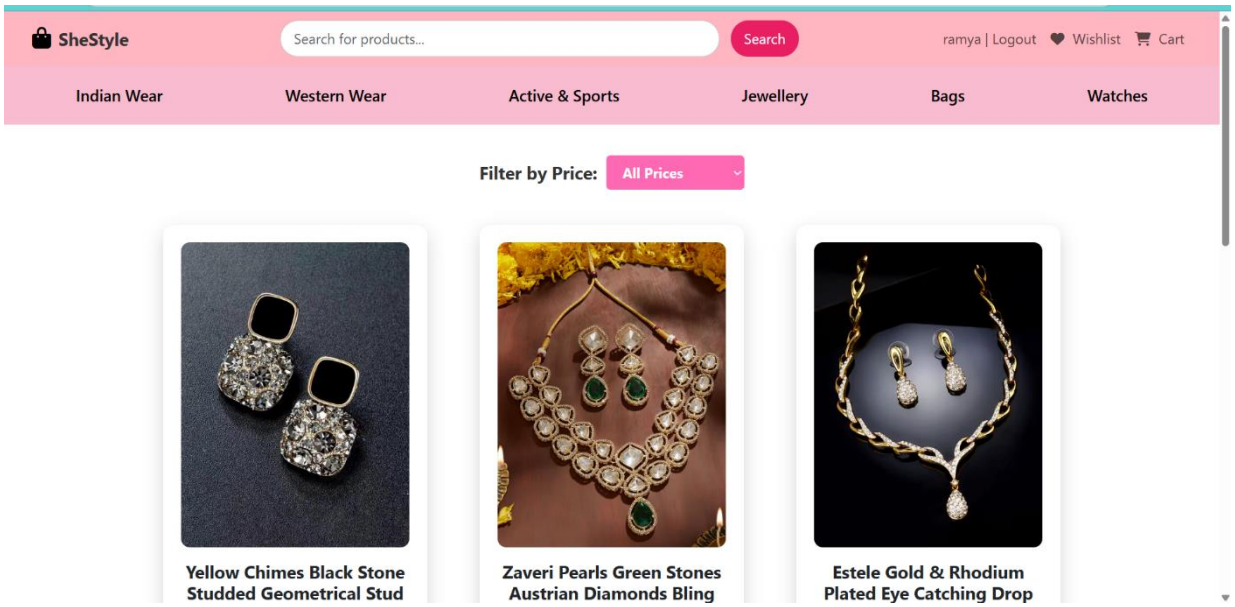
4.Western Wear:



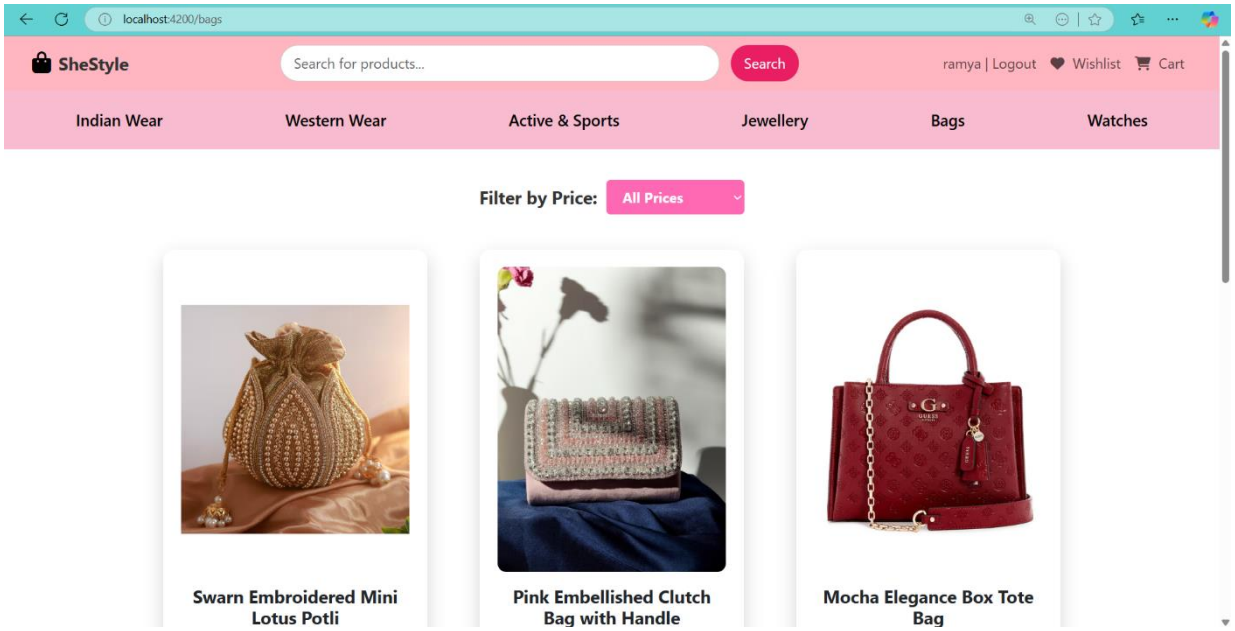
5. Active & Sports:



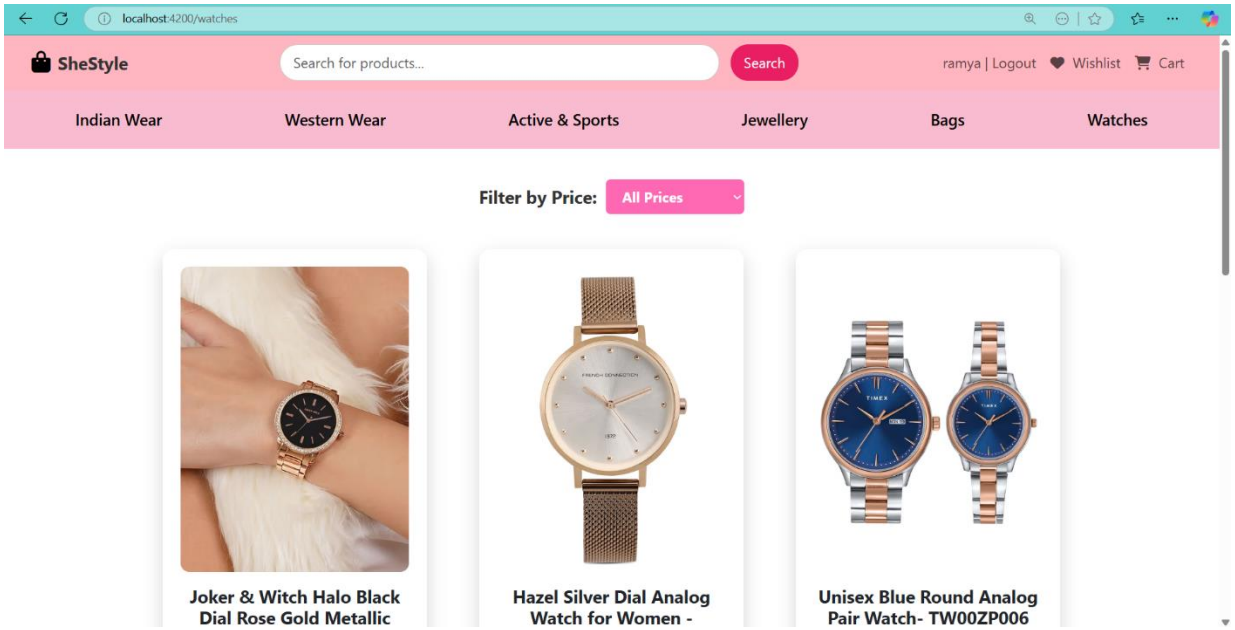
6.Jewllery:



7.Bags:



8.Watches:



9.Filters: