**Documentation for Day 4 - Dynamic Frontend Components**  
**Marketplace Name: [BazaarNest]**

**1. Technical Report**

**Modern Implementation Approach**

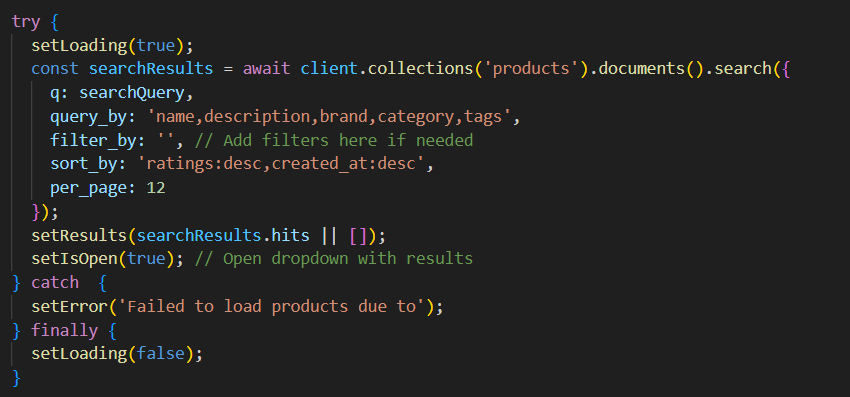
* **Next.js 15**: Leveraged the latest features like Server Components, API Routes, and streaming for dynamic data handling.
* **Client-Side Fetching**: Used fetch with revalidate for dynamic updates and caching.
* **Typesense Integration**: Implemented serverless search via Next.js API routes for **lightning-fast, typo-tolerant search**.

**Steps Taken to Build and Integrations**

1. **Project Setup**
   * Initialized Next.js 15 with the App Router.
   * Connected Sanity CMS using @sanity/client for real-time data.
   * Configured **Typesense** for search via Next.js API routes.
2. **Component Development**

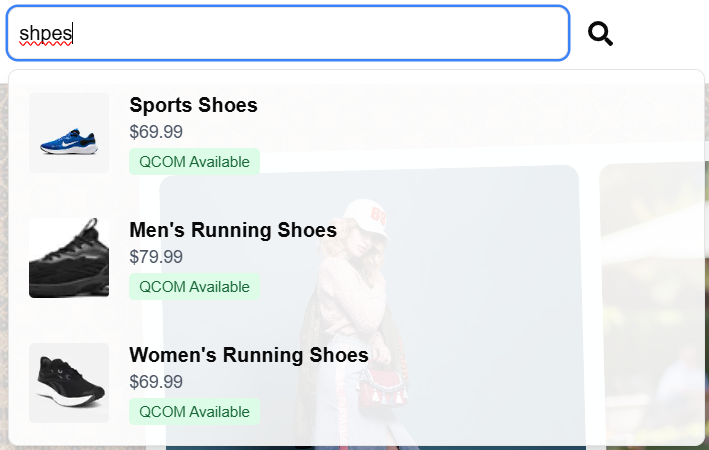
**Product Listing**:

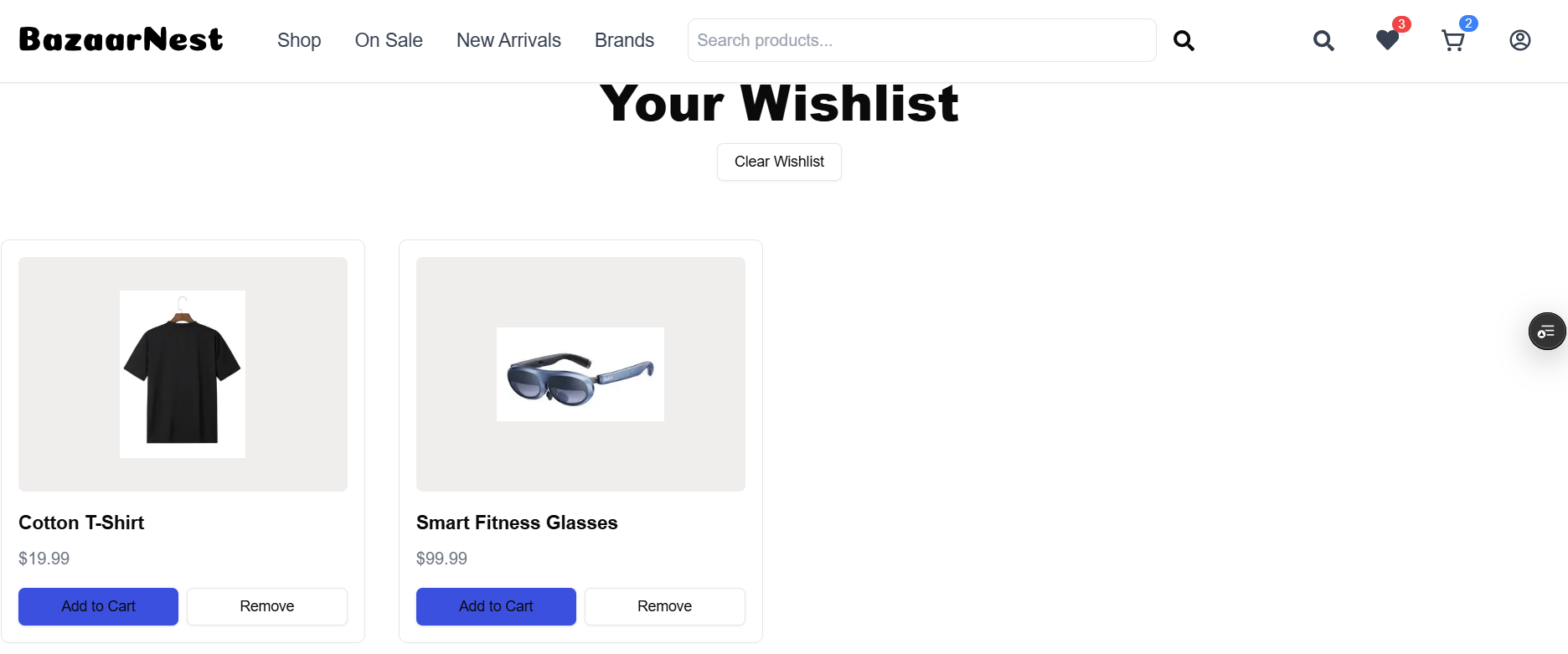
* Fetched data client-side

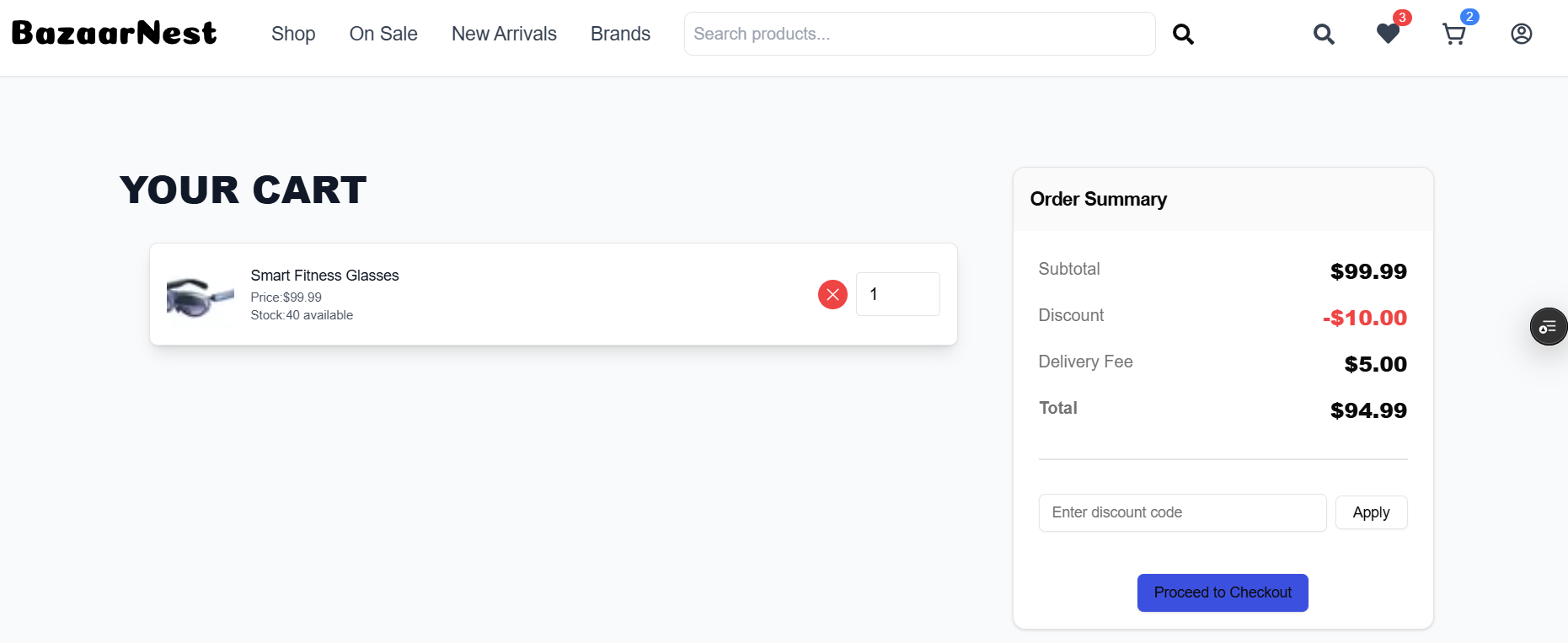


* Rendered products in a responsive grid.

**Typesense-Powered Search Bar**:

* Integrated lightning fast search engine with typo tolerance
* 
* **Cart & Wishlist**:
  + Used zustand or useContext + localStorage for state management.





1. **Styling**
   * Used **Tailwind CSS** with @layer for responsive design.
   * Added CSS transitions for interactive elements (e.g., hover effects on product cards).
   * Used modern animation library Framer Motion

**Challenges & Solutions**

1. **Real-Time Search Performance**
   * **Issue**: High latency when querying Typesense directly from the client.
   * **Solution**: Proxied requests via Next.js API routes for edge caching and reduced latency.
2. **Debounce Optimization**
   * **Issue**: Excessive API calls during rapid typing.
   * **Solution**: Implemented lodash.debounce with a 300ms delay.
3. **State Hydration**
   * **Issue**: Cart/wishlist data lost after page refresh.
   * **Solution**: Synced client state with localStorage using useEffect.

**Best Practices**

* **Modern Data Fetching**:
  + Used Next.js 15’s fetch with revalidate for incremental updates.
  + Separated client/server logic with API routes.
* **Search Optimization**:
  + Configured Typesense with num\_typos: 2 and query\_by: "name,description" for typo tolerance.
  + Edge-deployed Typesense for sub-50ms response times.
* **Code Quality**:
  + Adopted atomic component design (e.g., <SearchBar />, <ProductCard />).
  + Used TypeScript for type safety.
* **Self-Validation Checklist**
* **Modern Next.js 15 Features**: API routes, dynamic routing, and client components.
* **Typesense Search**: Typo tolerance, sub-50ms response times.
* **State Management**: Cart/wishlist persistence with localStorage.
* **Responsive Design**: Mobile-first Tailwind CSS implementation.
* **Submitted By**: Ibad Uddin  
    
  **Format**: Word/Pdf file
* **Why This Approach?**
* **Next.js 15**: Streamlined server/client integration and improved performance.
* **Typesense**: Outperforms traditional search solutions with its speed and typo tolerance.
* **Modularity**: Reusable components and API routes ensure scalability for future features.
* This setup mirrors industry standards for high-performance marketplaces like Etsy or Shopify. 🚀