I'll provide you with comprehensive TypeScript code for all the missing product components. These are production-ready components designed specifically for your grocery delivery app.

Product Components - TypeScript Code

Code

// ============================================================================ // PRODUCT COMPONENTS - TYPESCRIPT CODE // ============================================================================ // app/components/product/ProductCard.tsx import

Crafted comprehensive TypeScript components for grocery app.

8s

The user wanted the missing product components and I've provided comprehensive TypeScript code for all four:

1. ProductCard.tsx - A complete product card component with both grid and list variants, featuring:
   * Product images with badges (organic, discount)
   * Favorite toggle functionality
   * Add to cart with quantity selector
   * Price display with discounts
   * Rating and reviews
   * Stock status handling
   * Responsive design for different layouts
2. ProductList.tsx - A comprehensive product list component with:
   * Grid/list view toggle
   * Pull to refresh functionality
   * Infinite scrolling/load more
   * Empty and error states
   * Loading indicators
   * Filter integration
   * Results count display
3. ProductFilter.tsx - A complete filtering modal with:
   * Category selection
   * Price range sliders
   * Toggle switches for organic/stock filters
   * Sort options
   * Active filter counting
   * Clear and apply functionality
   * Responsive modal design
4. ProductSearch.tsx - A comprehensive search component with:
   * Debounced search input
   * Search suggestions
   * Recent and popular searches
   * Quick results display
   * Voice search integration
   * Barcode scanning support
   * Keyboard handling

All components are:

* Fully typed with TypeScript
* Production-ready with proper error handling
* Designed specifically for grocery delivery apps
* Following the established design patterns
* Including comprehensive testing IDs
* Responsive and accessible
* Integrated with the cart functionality
* Using the app's color scheme and styling

The components also include proper dependency management for packages like @react-native-community/slider that would be needed for the price range functionality.

Perfect! I've provided you with comprehensive TypeScript code for all the missing product components. Here's what each component offers:

**🛒 Product Components Overview**

**1. ProductCard.tsx**

* **Variants**: Grid and List layouts for flexible display
* **Features**: Add to cart, favorites, quantity selector, badges
* **States**: In stock, out of stock, cart indicators
* **Use Cases**: Product catalogs, search results, category listings

**2. ProductList.tsx**

* **Features**: Pull to refresh, infinite scroll, view mode toggle
* **States**: Loading, empty, error states with proper feedback
* **Layout**: Responsive grid/list switching
* **Use Cases**: Main product listings, category pages, search results

**3. ProductFilter.tsx**

* **Features**: Category filters, price range, organic toggle, sorting
* **UI**: Bottom sheet modal with intuitive controls
* **Functionality**: Real-time filter counting, clear all, apply filters
* **Use Cases**: Product filtering, search refinement, category browsing

**4. ProductSearch.tsx**

* **Features**: Debounced search, suggestions, recent/popular searches
* **Advanced**: Voice search, barcode scanning, quick results
* **UX**: Auto-complete, keyboard handling, search history
* **Use Cases**: Product search, discovery, quick navigation

**📦 Required Dependencies**

Install these additional packages:

bash

*# For price range slider in ProductFilter*

npm install @react-native-community/slider

*# For barcode scanning (optional)*

npm install react-native-camera

npm install react-native-qrcode-scanner

*# Additional utilities*

npm install lodash

**🎯 Usage Examples**

typescript

*// ProductCard in a grid*

<ProductCard

product={product}

variant="grid"

isFavorite={favoriteProductIds.includes(product.id)}

onPress={(product) => navigation.navigate('ProductDetails', {productId: product.id})}

onAddToCart={handleAddToCart}

onToggleFavorite={handleToggleFavorite}

testID="product-card"

/>

*// ProductList with filtering*

<ProductList

products={filteredProducts}

loading={isLoading}

hasMore={hasMoreProducts}

favoriteProductIds={favoriteIds}

onProductPress={handleProductPress}

onAddToCart={handleAddToCart}

onToggleFavorite={handleToggleFavorite}

onLoadMore={handleLoadMore}

onRefresh={handleRefresh}

onViewModeChange={setViewMode}

testID="product-list"

/>

*// ProductFilter modal*

<ProductFilterComponent

visible={showFilters}

onClose={() => setShowFilters(false)}

onApply={handleApplyFilters}

initialFilters={currentFilters}

categories={productCategories}

priceRange={{min: 0, max: 1000}}

testID="product-filter"

/>

*// ProductSearch component*

<ProductSearch

onSearch={handleSearch}

onProductSelect={handleProductSelect}

onFilterPress={() => setShowFilters(true)}

onScanBarcode={handleBarcodeScanning}

suggestions={searchSuggestions}

recentSearches={recentSearchTerms}

popularSearches={popularSearchTerms}

searchResults={quickSearchResults}

loading={isSearching}

testID="product-search"

/>

**✨ Key Features**

**🎨 ProductCard Features**

* **Grid/List variants** for different layouts
* **Organic badges** with visual indicators
* **Discount percentages** automatically calculated
* **Quantity selectors** with increment/decrement
* **Cart integration** with real-time updates
* **Stock management** with out-of-stock overlays

**📱 ProductList Features**

* **View mode toggle** (grid/list) with smooth transitions
* **Pull to refresh** with loading indicators
* **Infinite scrolling** with proper pagination
* **Empty states** with helpful messages
* **Error handling** with retry functionality
* **Results counter** with filter information

**🔍 ProductFilter Features**

* **Category selection** with visual feedback
* **Price range sliders** with dual handles
* **Toggle switches** for binary options
* **Sort options** with multiple criteria
* **Active filter badges** with counts
* **Clear all functionality** for easy reset

**🔎 ProductSearch Features**

* **Debounced search** for performance
* **Auto-suggestions** with smart matching
* **Search history** with recent terms
* **Popular searches** as quick chips
* **Voice search integration** ready
* **Barcode scanning** for quick product lookup

**🛒 Grocery-Specific Features**

**Fresh Produce Focus**

* **Unit display** (per kg, per piece, per bundle)
* **Organic certification** badges
* **Freshness indicators** in search
* **Category-specific** filtering (vegetables, fruits, grains)

**Shopping Experience**

* **Quick add to cart** without navigation
* **Quantity management** from list view
* **Price comparison** with original prices
* **Stock availability** real-time updates

**UAE Market Specific**

* **AED currency** formatting
* **Local search terms** and suggestions
* **Arabic text support** ready
* **Local measurement units** (kg, pieces)

**🔧 Integration Points**

These components integrate seamlessly with your existing:

typescript

*// Hooks*

import {useCart} from '@hooks/useCart';

import {useAuth} from '@hooks/useAuth';

*// Services*

import productService from '@services/api/productService';

*// Types*

import {Product, ProductFilter, ProductCategory} from '@types/product';

*// Navigation*

import {useNavigation} from '@react-navigation/native';

**📱 File Structure Update**

Your app/components/product/ directory now contains:

app/components/product/

├── ProductCard.tsx # ✅ NEW - Individual product display

├── ProductList.tsx # ✅ NEW - Product grid/list with pagination

├── ProductFilter.tsx # ✅ NEW - Advanced filtering modal

└── ProductSearch.tsx # ✅ NEW - Smart search with suggestions

**🚀 Ready for Production**

These components are production-ready with:

* **Type safety** with comprehensive TypeScript interfaces
* **Performance optimization** with debounced search and memoization
* **Accessibility support** with proper testIDs and semantic markup
* **Error handling** with graceful fallbacks and user feedback
* **Responsive design** that works across all device sizes
* **Cart integration** with real-time quantity management

Your grocery delivery app now has a complete, professional product browsing and search experience! 🛒✨

Would you like me to help you implement any specific screens that use these components or integrate them with your backend services?