Complete File Documentation for apps/web/src

(81 Files Total)

# ROOT LEVEL FILES (4 files)

### 1. App.tsx

**Location**: /apps/web/src/App.tsx  
**Imports**: react-router-dom (BrowserRouter, Routes, Route)  
**Functionality**: Main application router that sets up three primary routes: "/" (ShopPage), "/admin" (Dashboard), "/admin/login" (Login)  
**Exports**: Default export of App component

### 2. main.tsx

**Location**: /apps/web/src/main.tsx  
**Imports**: React, ReactDOM, App component, CSS styles  
**Functionality**: Application entry point that renders React app to DOM with StrictMode  
**Exports**: None (side-effects only)

### 3. vit-env.d.ts

**Location**: /apps/web/src/vit-env.d.ts  
**Imports**: None (type definitions only)  
**Functionality**: Vite environment variable type definitions  
**Exports**: ImportMetaEnv interface with VITE\_API\_URL

### 4. reportWebVitals.ts

**Location**: /apps/web/src/reportWebVitals.ts  
**Imports**: web-vitals library types  
**Functionality**: Performance monitoring utility that dynamically imports web-vitals metrics (CLS, INP, FCP, LCP, TTFB)  
**Exports**: Default export of reportWebVitals function

# TYPES DIRECTORY (9 files)

## Main Index: /apps/web/src/types/index.ts

**Imports**: All type definitions from models/, api/, ui/ subdirectories  
**Functionality**: Barrel export aggregating all type definitions  
**Exports**: Re-exports from models/, api/, and ui/ subdirectories

## Models Directory (/apps/web/src/types/models/)

### 1. card.ts (280 lines)

**Location**: /apps/web/src/types/models/card.ts  
**Imports**: None (type-only module)  
**Functionality**: Comprehensive card type system with admin/storefront distinction, card variation management, and utility functions for card display and filtering  
**Exports**:

**Interfaces**: Card, CardVariation, StorefrontCard, AdminCard, CardSearchResult, BrowseVariation, BrowseBaseCard, PartialCard, DisplayCard

**Type Guards**: isStorefrontCard(), isAdminCard(), hasVariationMetadata()

**Utilities**: formatTreatment(), formatFinish(), isFoilCard(), hasSpecialTreatment(), getCardDisplayName(), getUniqueTreatments(), getUniqueFinishes()

### 2. inventory.ts

**Location**: /apps/web/src/types/models/inventory.ts  
**Imports**: None  
**Functionality**: Defines data structures for inventory management  
**Exports**: InventoryItem, InventoryUpdate interfaces

### 3. order.ts

**Location**: /apps/web/src/types/models/order.ts  
**Imports**: None  
**Functionality**: Defines order processing and customer data structures  
**Exports**: OrderItem, Order, OrderStatus type, CustomerData

## API Types Directory (/apps/web/src/types/api/)

### 1. requests.ts

**Location**: /apps/web/src/types/api/requests.ts  
**Imports**: None  
**Functionality**: Defines shapes for API request parameters  
**Exports**: SearchFilters, PaginationParams, CardSearchParams

### 2. responses.ts

**Location**: /apps/web/src/types/api/responses.ts  
**Imports**: None  
**Functionality**: Defines shapes for API responses  
**Exports**: PaginatedResponse, ApiResponse, FilterCounts

## UI Types Directory (/apps/web/src/types/ui/)

### 1. cart.ts

**Location**: /apps/web/src/types/ui/cart.ts  
**Imports**: None  
**Functionality**: Defines shopping cart UI data structures  
**Exports**: CartItem, Cart

### 2. common.ts

**Location**: /apps/web/src/types/ui/common.ts  
**Imports**: None  
**Functionality**: Common UI types used across the application  
**Exports**: Currency, ViewMode, ToastMessage, LoadingState

# LIB DIRECTORY (11 files)

## API Directory (/apps/web/src/lib/api/)

### 1. client.ts (177 lines)

**Location**: /apps/web/src/lib/api/client.ts  
**Imports**: zod (for validation), throttledFetch, logError  
**Functionality**: Type-safe HTTP client with request/response handling. ApiClient class provides methods for GET, POST, PUT, PATCH, DELETE with header management, URL building, Zod schema validation, and error handling  
**Exports**: api (singleton instance), API\_BASE constant

### 2. endpoints.ts (157 lines)

**Location**: /apps/web/src/lib/api/endpoints.ts  
**Imports**: None  
**Functionality**: Provides API endpoint constants organized by category (AUTH, CARDS, STOREFRONT, GAMES, SETS, SEARCH, ORDERS, ADMIN, HEALTH) and utility functions for building query strings  
**Exports**: ENDPOINTS object with 50+ endpoint definitions, buildQueryString() and withQuery() helper functions

### 3. index.ts

**Location**: /apps/web/src/lib/api/index.ts  
**Imports**: api, API\_BASE, ENDPOINTS, buildQueryString  
**Functionality**: Barrel export for API utilities  
**Exports**: Re-exports all API utilities

## Constants Directory (/apps/web/src/lib/constants/)

### 1. index.ts (222 lines)

**Location**: /apps/web/src/lib/constants/index.ts  
**Imports**: None  
**Functionality**: Centralized application constants and configuration including API settings, UI configuration, feature flags, and image placeholders  
**Exports**: API\_CONFIG, FILTER\_CONFIG, VIRTUAL\_SCROLL\_CONFIG, BREAKPOINTS, GRID\_CONFIG, ACCESSIBILITY\_CONFIG, ANIMATION\_CONFIG, CART\_CONFIG, IMAGE\_CONFIG, CURRENCY\_CONFIG, VALIDATION\_CONFIG, ERROR\_CONFIG, FEATURES, UI\_TEXT, KEYBOARD\_SHORTCUTS, PERFORMANCE\_CONFIG, PLACEHOLDER\_CARD\_IMAGE, ADMIN\_CONFIG

### 2. validation.ts (73 lines)

**Location**: /apps/web/src/lib/constants/validation.ts  
**Imports**: VALIDATION\_CONFIG from index  
**Functionality**: Provides form validation rules and constraints for customer data, addresses, orders, and admin functions  
**Exports**: FORM\_VALIDATION object with customer, address, order, and admin validation rules

## Utils Directory (/apps/web/src/lib/utils/)

### 1. index.ts

**Location**: /apps/web/src/lib/utils/index.ts  
**Imports**: All utility functions from csv, format, groupCards, sanitization, virtualScroll modules  
**Functionality**: Barrel export for utility functions  
**Exports**: Aggregates exports from all utility submodules

### 2. format.ts (45 lines)

**Location**: /apps/web/src/lib/utils/format.ts  
**Imports**: Currency type from @/types  
**Functionality**: Currency and price formatting utilities with support for different currencies and display contexts  
**Exports**: formatCurrency(), formatPrice(), formatCurrencySimple(), formatOrderTotal()

### 3. groupCards.ts (99 lines)

**Location**: /apps/web/src/lib/utils/groupCards.ts  
**Imports**: Card types, CardVariation, formatting utilities, BrowseBaseCard, BrowseVariation  
**Functionality**: Groups catalog rows into browseable base cards, organizing by set\_id/card\_number with variation sorting for improved browsing experience  
**Exports**: groupCardsForBrowse()

### 4. sanitization.ts (114 lines)

**Location**: /apps/web/src/lib/utils/sanitization.ts  
**Imports**: buffer.constants (unused import)  
**Functionality**: XSS prevention through comprehensive input sanitization with functions for text, HTML, email, phone, address, and complete customer data  
**Exports**: sanitizeText(), sanitizeHTML(), sanitizeEmail(), sanitizePhone(), sanitizeAddress(), sanitizeCustomerData()

### 5. csv.ts (297 lines)

**Location**: /apps/web/src/lib/utils/csv.ts  
**Imports**: None  
**Functionality**: CSV import/export utilities for bulk inventory operations with parsing, validation, formatting, and download capabilities  
**Exports**: arrayToCSV(), csvToArray(), downloadCSV(), validateInventoryCSV(), formatInventoryForExport(), generateInventoryTemplate()

### 6. virtualScroll.ts (107 lines)

**Location**: /apps/web/src/lib/utils/virtualScroll.ts  
**Imports**: React hooks (useState, useCallback, useEffect, useMemo)  
**Functionality**: Virtual scrolling hook for performance optimization of large lists with responsive columns, overscan buffering, and scroll handling  
**Exports**: useVirtualScroll hook

# SERVICES DIRECTORY (3 files)

## Error Handling Directory (/apps/web/src/services/error/)

### 1. types.ts

**Location**: /apps/web/src/services/error/types.ts  
**Imports**: None  
**Functionality**: Defines error type classifications and interfaces  
**Exports**: ErrorType enum (NETWORK, AUTH, VALIDATION, API, UNKNOWN), FormattedError, ErrorTemplate interfaces

### 2. handler.ts (125 lines)

**Location**: /apps/web/src/services/error/handler.ts  
**Imports**: Error types from types.ts  
**Functionality**: Centralized error handling and logging system with categorization, formatting, logging, and React hook integration  
**Exports**: categorizeError(), formatError(), logError(), useErrorHandler()

## HTTP Directory (/apps/web/src/services/http/)

### 1. throttler.ts (89 lines)

**Location**: /apps/web/src/services/http/throttler.ts  
**Imports**: None  
**Functionality**: Request throttling and deduplication system with RequestThrottler class that prevents API spam and deduplicates simultaneous requests  
**Exports**: RequestThrottler class, defaultThrottler instance, throttledFetch function

# SHARED COMPONENTS DIRECTORY (25 files)

## UI Components (/apps/web/src/shared/ui/)

### 1. index.ts

**Location**: /apps/web/src/shared/ui/index.ts  
**Imports**: Toast, EmptyState, SectionHeader, VariationBadge, MobileFilterModal, FilterSidebar, CurrencySelector components  
**Functionality**: Barrel exports for shared UI components  
**Exports**: Toast, EmptyState, SectionHeader, VariationBadge, MobileFilterModal, FilterSidebar, CurrencySelector

### 2. Toast.tsx

**Location**: /apps/web/src/shared/ui/Toast.tsx  
**Imports**: React hooks, lucide-react icons  
**Functionality**: Toast notification system with context provider and individual toast items for user feedback  
**Exports**: ToastProvider, useToast hook, Toast component

### 3. Button.tsx, Modal.tsx, EmptyState.tsx

**Location**: /apps/web/src/shared/ui/[Button|Modal|EmptyState].tsx  
**Imports**: Minimal or none  
**Functionality**: Basic UI components (implementations appear minimal/placeholder)  
**Exports**: Respective component exports

### 4. VariationBadge.tsx, FilterSidebar.tsx, CurrencySelector.tsx, MobileFilterModal.tsx, SectionHeader.tsx

**Location**: /apps/web/src/shared/ui/[component].tsx  
**Imports**: React, relevant types, styling utilities  
**Functionality**: Specialized UI components for variation display, filtering, currency selection, mobile filtering, and section headers  
**Exports**: Respective component exports

## Card Components (/apps/web/src/shared/card/)

### 1. index.ts

**Location**: /apps/web/src/shared/card/index.ts  
**Imports**: CardItem, CardRow, CardSkeleton components  
**Functionality**: Barrel exports for card display components  
**Exports**: CardItem, CardRow, CardSkeleton

### 2. CardItem.tsx, CardRow.tsx, CardSkeleton.tsx

**Location**: /apps/web/src/shared/card/[component].tsx  
**Imports**: React, card types, styling utilities  
**Functionality**: Card display components for different layouts and loading states  
**Exports**: Respective card component exports

## Layout Components (/apps/web/src/shared/layout/)

### 1. index.ts

**Location**: /apps/web/src/shared/layout/index.ts  
**Imports**: ErrorBoundary, KeyboardShortcuts, CardGrid, CardList components  
**Functionality**: Barrel exports for layout components  
**Exports**: ErrorBoundary, KeyboardShortcuts, CardGrid, CardList

### 2. ErrorBoundary.tsx

**Location**: /apps/web/src/shared/layout/ErrorBoundary.tsx  
**Imports**: React, React.ErrorBoundary  
**Functionality**: React error boundary for crash handling and graceful error display  
**Exports**: ErrorBoundary component

### 3. KeyboardShortcuts.tsx

**Location**: /apps/web/src/shared/layout/KeyboardShortcuts.tsx  
**Imports**: React, keyboard constants  
**Functionality**: Keyboard shortcut UI display and help system  
**Exports**: KeyboardShortcuts component

### 4. CardGrid.tsx, CardList.tsx

**Location**: /apps/web/src/shared/layout/[component].tsx  
**Imports**: React, card components, layout utilities  
**Functionality**: Card display layouts for grid and list views  
**Exports**: Respective layout component exports

**Search Components (/apps/web/src/shared/search/)**

### 1. index.ts

**Location**: /apps/web/src/shared/search/index.ts  
**Imports**: FiltersPanel, CardSearchBar, ActiveFilters, MobileFilterButton, VariationFilter components  
**Functionality**: Barrel exports for search and filter components  
**Exports**: FiltersPanel, CardSearchBar, ActiveFilters, MobileFilterButton, VariationFilter

### 2. FiltersPanel.tsx, SearchBar.tsx, ActiveFilters.tsx, MobileFilterButton.tsx, VariationFilter.tsx

**Location**: /apps/web/src/shared/search/[component].tsx  
**Imports**: React, search types, filter utilities, UI components  
**Functionality**: Search and filter UI components for various filtering needs  
**Exports**: Respective search component exports

## Media Components (/apps/web/src/shared/media/)

### 1. OptimizedImage.tsx (156 lines)

**Location**: /apps/web/src/shared/media/OptimizedImage.tsx  
**Imports**: React hooks (useState, useEffect, useRef, useCallback)  
**Functionality**: Optimized image component with lazy loading using Intersection Observer, blur placeholder, error fallback, and loading states for performance  
**Exports**: OptimizedImage component

### 2. index.ts

**Location**: /apps/web/src/shared/media/index.ts  
**Imports**: OptimizedImage component  
**Functionality**: Barrel export for media components  
**Exports**: OptimizedImage

## Accessibility (/apps/web/src/a11y/)

### 1. Announcer.tsx (10 lines)

**Location**: /apps/web/src/a11y/Announcer.tsx  
**Imports**: React hooks  
**Functionality**: Screen reader announcements for dynamic content using aria-live region with polite updates  
**Exports**: Announcer component

# FEATURES DIRECTORY (33+ files)

## Hooks Directory (/apps/web/src/features/hooks/)

### 1. index.ts

**Location**: /apps/web/src/features/hooks/index.ts  
**Imports**: All custom hooks from the hooks directory  
**Functionality**: Barrel export for all custom React hooks  
**Exports**: useCardFetching, useCart, useRecentlyViewed, useShopFilters, useShopKeyboardShortcuts, useShopViewMode, useVariationSelection, useCardDisplayArea

### 2. useCart.ts (138 lines)

**Location**: /apps/web/src/features/hooks/useCart.ts  
**Imports**: React hooks, CartItem/Cart types  
**Functionality**: Shopping cart management with localStorage persistence, providing cart state and manipulation functions  
**Exports**: useCart hook with cart object and functions (addItem, updateQuantity, removeItem, clearCart, etc.)

### 3. useShopFilters.ts (263 lines)

**Location**: /apps/web/src/features/hooks/useShopFilters.ts  
**Imports**: React hooks, react-router-dom, api, ENDPOINTS, SearchFilters  
**Functionality**: Filter management with URL synchronization and API loading, handles filter parsing from URL parameters  
**Exports**: useShopFilters hook with filters, options, loading state, and filter manipulation functions

### 4. useCardFetching.ts (117 lines)

**Location**: /apps/web/src/features/hooks/useCardFetching.ts  
**Imports**: React hooks, api, buildQueryString, useErrorHandler, StorefrontCard types  
**Functionality**: Card fetching with game/set resolution and search term handling, includes client-side treatment filtering  
**Exports**: useCardFetching hook returning cards, loading, error, refetch

### 5. useFilterCounts.ts (180 lines)

**Location**: /apps/web/src/features/hooks/useFilterCounts.ts  
**Imports**: React hooks, constants, api  
**Functionality**: Gets total card count with caching, debouncing, and retry logic with exponential backoff  
**Exports**: useFilterCounts hook with total count and legacy compatibility methods

### 6. useShopViewMode.ts (40 lines)

**Location**: /apps/web/src/features/hooks/useShopViewMode.ts  
**Imports**: React hooks, Currency type  
**Functionality**: Manages view mode (grid/list) and currency selection state  
**Exports**: useShopViewMode hook with viewMode, currency, isOffline state

### 7. useVariationSelection.ts (40 lines)

**Location**: /apps/web/src/features/hooks/useVariationSelection.ts  
**Imports**: React hooks, StorefrontCard type  
**Functionality**: Tracks selected variations per card with functions to select, get, and clear stale selections  
**Exports**: useVariationSelection hook with selection state and manipulation functions

### 8. useShopKeyboardShortcuts.ts (58 lines)

**Location**: /apps/web/src/features/hooks/useShopKeyboardShortcuts.ts  
**Imports**: React useEffect  
**Functionality**: Handles keyboard shortcuts for shop (Ctrl+K search, ? help, Escape close) with event handlers  
**Exports**: useShopKeyboardShortcuts hook

### 9. useRecentlyViewed.tsx (216 lines)

**Location**: /apps/web/src/features/hooks/useRecentlyViewed.tsx  
**Imports**: React hooks, lucide-react icons, OptimizedImage, Card type  
**Functionality**: Tracks and displays recently viewed cards with localStorage persistence (max 20 items) and provides a display component  
**Exports**: useRecentlyViewed hook and RecentlyViewedCards component

### 10. useCardDisplayArea.tsx (225 lines)

**Location**: /apps/web/src/features/hooks/useCardDisplayArea.tsx  
**Imports**: React hooks, CardGrid/CardList/ErrorBoundary, CardItem, types  
**Functionality**: Display area rendering with grouping and sorting logic, groups by name/set/rarity/price and provides view switching  
**Exports**: useCardDisplayArea component with grid/list view switching

## Shop Feature (/apps/web/src/features/shop/)

### 1. ShopPage.tsx (100+ lines)

**Location**: /apps/web/src/features/shop/ShopPage.tsx  
**Imports**: React, react-router-dom, all shop hooks, shop components, shared components, formatting utilities  
**Functionality**: Main shop/storefront page with URL parameter management, card display, filtering, cart functionality, view mode toggle, search, and keyboard shortcuts  
**Exports**: ShopPage component

### 2. components/index.ts

**Location**: /apps/web/src/features/shop/components/index.ts  
**Imports**: ShopHeader, ResultsHeader, RecentlyViewedCards, cart components  
**Functionality**: Barrel export for shop components  
**Exports**: ShopHeader, ResultsHeader, RecentlyViewedCards, cart-related exports (AddToCartModal, CartItem, CartModal, MiniCart, Checkout)

**3-7. Cart Components: AddToCartModal.tsx, CartItem.tsx, CartModal.tsx, MiniCart.tsx, Checkout.tsx**

**Location**: /apps/web/src/features/shop/components/cart/[component].tsx  
**Imports**: React, cart types, UI components, validation utilities  
**Functionality**: Shopping cart UI components with modal interfaces, item management, and checkout process including form validation and order submission with sanitization  
**Exports**: Respective cart component exports

## Admin Feature (/apps/web/src/features/admin/)

### 1. Login.tsx (150 lines)

**Location**: /apps/web/src/features/admin/Login.tsx  
**Imports**: React, react-router-dom, lucide-react icons, api, ENDPOINTS  
**Functionality**: Admin authentication form with credential validation, error handling, and responsive design  
**Exports**: Login component

### 2. Dashboard.tsx (80+ lines)

**Location**: /apps/web/src/features/admin/Dashboard.tsx  
**Imports**: React, lucide-react, api, ENDPOINTS, CurrencySelector, admin tab components  
**Functionality**: Main admin dashboard with tabbed interface (inventory, all-cards, analytics, orders), auth checking, logout, and currency selector  
**Exports**: Dashboard component

### 3-6. Admin Tab Components:

* **AnalyticsTab.tsx** - Analytics dashboard
* **CardsTab.tsx** - Card inventory management
* **AddToInventoryModal.tsx** - Add card modal
* **BulkManager.tsx** - Bulk operations
* **OrdersTab.tsx** - Order management

**Location**: /apps/web/src/features/admin/components/[category]/[component].tsx  
**Imports**: React, admin types, API utilities, UI components  
**Functionality**: Specialized admin interface components for different management tasks  
**Exports**: Respective admin component exports

# ARCHITECTURE SUMMARY

**Key Patterns Identified:**

1. **Type Safety**: Comprehensive TypeScript with Zod-validated API requests
2. **Component Organization**: Barrel exports in index.ts files for clean imports
3. **State Management**: React hooks with localStorage persistence
4. **API Integration**: Centralized client with throttling and error handling
5. **Error Handling**: Categorized errors with user-friendly messages
6. **Accessibility**: ARIA labels, keyboard shortcuts, screen reader support
7. **Performance**: Virtual scrolling, lazy loading, image optimization, request throttling
8. **Data Sanitization**: XSS prevention on customer inputs
9. **CSV Operations**: Bulk import/export for inventory management

**Total File Count: 81 Files**

* **Root**: 4 files (app entry, routing, config)
* **Types**: 9 files (comprehensive type system)
* **Lib**: 11 files (utilities, API, constants)
* **Services**: 3 files (error handling, HTTP throttling)
* **Shared**: 25 files (reusable UI components)
* **Features**: 33+ files (shop, admin, hooks)

This represents a complete, well-architected e-commerce platform for trading card singles with comprehensive admin tools, strong type safety, and excellent performance optimizations.