

Complete Homepage Architecture Analysis

Based on my comprehensive analysis, here's the complete picture of the **primary homepage (/)** and all its components:

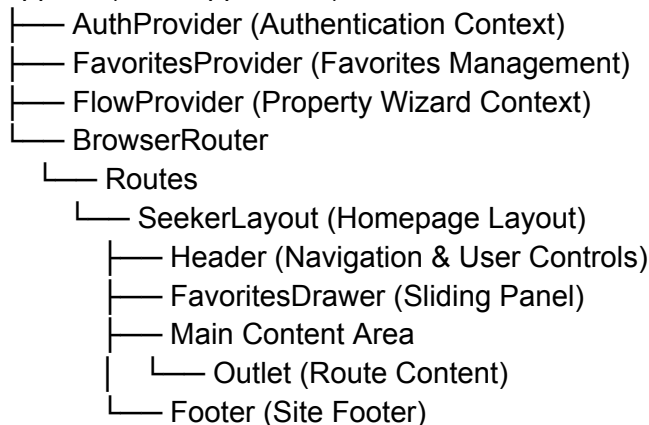
Homepage Route Structure

Root Path (/) - Primary Homepage

- **Component:** `PropertyMapHome` (via `SeekerLayout`)
 - **File:** `src/modules/seeker/index.tsx`
 - **Purpose:** Interactive property search with map integration
 - **User Experience:** Full property browsing interface
-

Complete Component Hierarchy

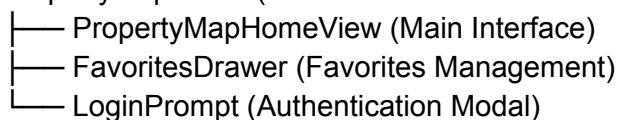
App.tsx (Root Application)



PropertyMapHome Architecture

Main Component Structure

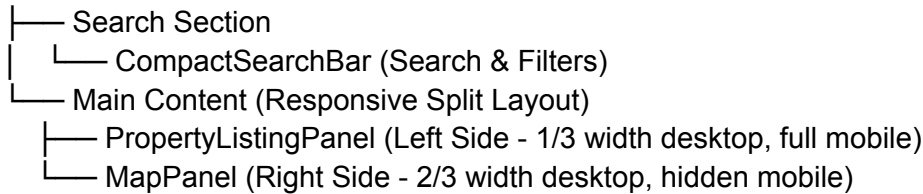
PropertyMapHome (src/modules/seeker/index.tsx)



PropertyMapHomeView - Core Interface

File: `src/modules/seeker/components/PropertyMapHomeView.tsx`

PropertyMapHomeView

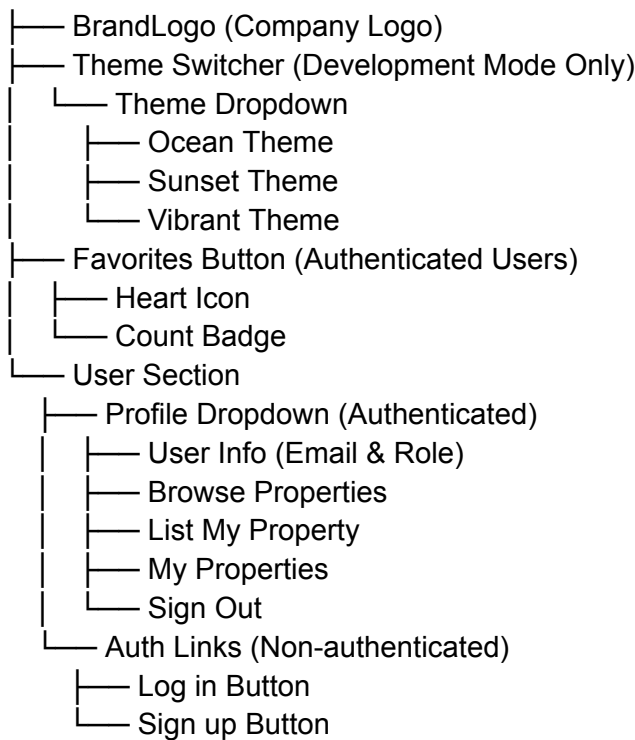


Detailed Component Breakdown

1. Header Component

File: `src/components/Header.tsx` **Purpose:** Universal navigation and user controls

Header



Key Features:

- **Responsive Design:** Adapts to mobile/desktop
- **Role Detection:** Shows appropriate options based on user role

- **Theme Management:** Live theme switching (dev mode)
- **Favorites Integration:** Real-time count updates

2. CompactSearchBar Component

File: `src/modules/seeker/components/CompactSearchBar.tsx` **Purpose:** Property search and filtering

CompactSearchBar



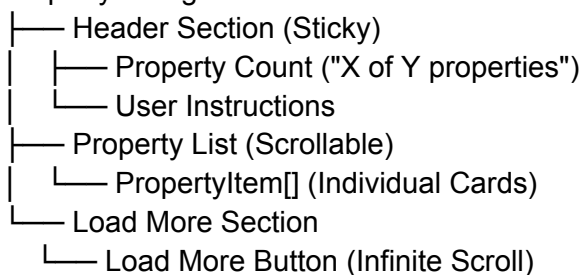
Key Features:

- **Location Suggestions:** Auto-complete for locations
- **Property Type Filtering:** Residential, Commercial, Land
- **Price Range:** Slider-based range selection
- **Filter Persistence:** Maintains filter state
- **Recent Searches:** Quick access to previous searches

3. PropertyListingPanel Component

File: `src/modules/seeker/components/PropertyListingPanel.tsx` **Purpose:** Property list display and management

PropertyListingPanel



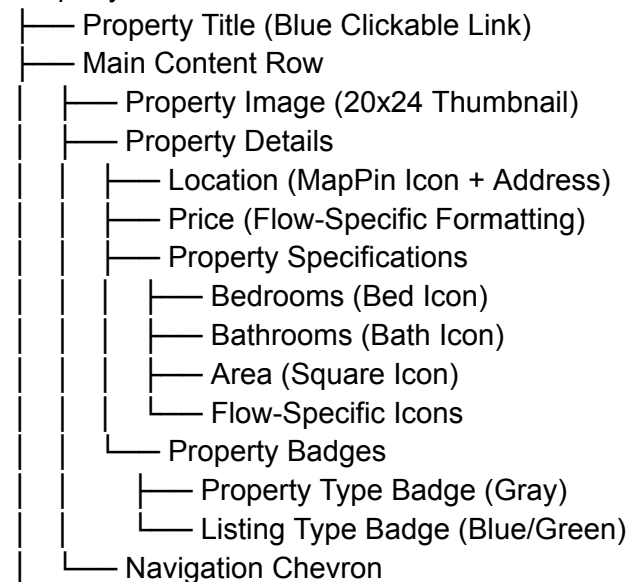
Key Features:

- **Infinite Scrolling:** Loads properties on demand
- **Image Optimization:** Cached image loading from Supabase
- **Loading States:** Skeleton placeholders
- **Empty States:** "No properties found" messaging
- **Favorites Integration:** Heart button on each card

4. PropertyItem Component

File: `src/modules/seeker/components/PropertyItem.tsx` **Purpose:** Individual property card display

PropertyItem



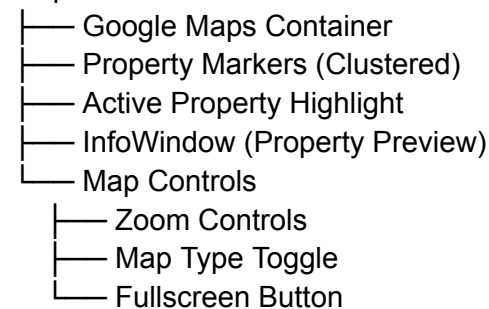
Flow-Specific Adaptations:

- **Residential Rent:** BHK, Bath, Area, Furnishing
- **Commercial Sale:** Area, Age, Floor Info
- **Coworking:** Seats, Area, Amenities
- **Land Sale:** Plot Area, Approvals, Land Type
- **PG/Hostel:** Room Type, Food, Gender Preference

5. MapPanel Component

File: `src/modules/seeker/components/MapPanel1.tsx` **Purpose:** Interactive Google Maps integration

MapPanel



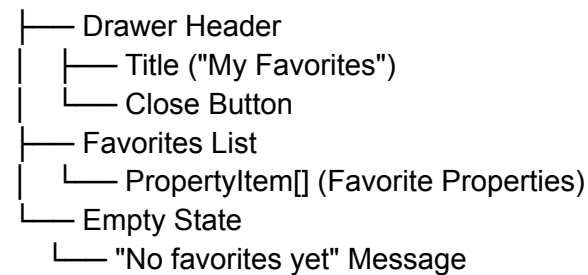
Key Features:

- **Property Markers:** Clustered markers for better performance
- **Hover Synchronization:** Highlights markers when list items hovered
- **InfoWindow:** Quick property preview on marker click
- **Map Interactions:** Pan, zoom, marker selection

6. FavoritesDrawer Component

File: `src/modules/seeker/components/FavoritesDrawer.tsx` **Purpose:** Sliding panel for favorites management

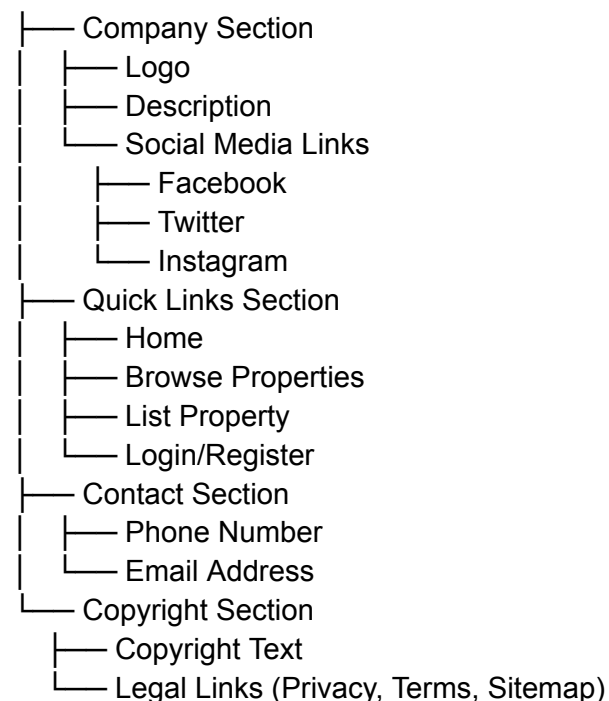
FavoritesDrawer



7. Footer Component

File: `src/components/Footer.tsx` **Purpose:** Site-wide footer information

Footer



Context Providers & State Management

AuthContext

File: `src/contexts/AuthContext.tsx` **Purpose:** User authentication state management

Provides:

- User session state
- Authentication methods (OTP, Password, Admin)
- Sign in/out functionality
- Registration handling

FavoritesContext

File: `src/contexts/FavoritesContext.tsx` **Purpose:** Favorites management across the app

Provides:

- Favorite properties list
- Add/remove favorites
- Favorite count for header badge
- Persistence with Supabase

ThemeContext

File: `src/contexts/ThemeContext.tsx` **Purpose:** Theme management system

Provides:

- Current theme state (Ocean, Sunset, Vibrant)
- Theme switching functionality
- CSS variable updates
- Local storage persistence

FlowContext

File: `src/contexts/FlowContext.tsx` **Purpose:** Property listing wizard flow management

Provides:

- Current flow state
 - Step navigation
 - Form data persistence
 - Flow type detection
-

Shared Utilities & Services

Property Data Services

- **File:** `src/modules/seeker/services/seekerService.ts`
- **Purpose:** Property data fetching and management
- **Functions:** Search, filter, pagination, favorites

Map Integration

- **File:** `src/modules/seeker/hooks/useGoogleMaps.ts`
- **Purpose:** Google Maps loading and management
- **Features:** Map initialization, error handling, loading states

Property Title Utilities

- **File:** `src/modules/seeker/utils/propertyTitleUtils.ts`
 - **Purpose:** Property display formatting
 - **Functions:** Flow type detection, title formatting, location formatting
-

Key Interactions & Data Flow

Property Search Flow

1. **User enters search** → CompactSearchBar
2. **Search triggers** → usePropertyMapData hook
3. **API call** → seekerService
4. **Results displayed** → PropertyListingPanel
5. **Map updated** → MapPanel with new markers

Favorites Management Flow

1. **User clicks heart** → PropertyItem
2. **Toggle favorite** → PropertyListingPanel
3. **API call** → FavoritesContext
4. **Update count** → Header badge
5. **Sync drawer** → FavoritesDrawer

Map Synchronization Flow

1. **Hover property card** → PropertyItem
2. **Trigger hover** → PropertyListingPanel
3. **Update map** → MapPanel highlights marker
4. **Click marker** → MapPanel

5. **Update active** → PropertyListingPanel highlights card
-

Mobile Responsiveness

Desktop Layout (≥768px)

- **Split View:** Property list (1/3) + Map (2/3)
- **Full Features:** All components visible
- **Hover Effects:** Card-to-map synchronization

Mobile Layout (<768px)

- **Single View:** Property list only (map hidden)
 - **Touch Optimized:** Larger touch targets
 - **Simplified UI:** Reduced information density
-

Performance Optimizations

Image Loading

- **Lazy Loading:** Images loaded as needed
- **Caching Strategy:** Local cache prevents re-downloads
- **Fallback Handling:** Default images for errors
- **Signed URLs:** Secure Supabase storage access

Component Optimization

- **Memo Usage:** Prevents unnecessary re-renders
 - **Debounced Search:** Reduces API calls
 - **Virtual Scrolling:** Handles large property lists
 - **State Batching:** Groups related state updates
-

This comprehensive architecture provides a modern, responsive, and performant real estate platform with rich interactive features, proper state management, and excellent user experience across all device types.