Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the implementation details for **PropView - Real Estate Listing Portal UI**. PropView is a frontend application for browsing, filtering, and viewing real estate listings, featuring image galleries and interactive maps. The portal is built using React, JavaScript, HTML, CSS, and Tailwind for responsive and dynamic user experiences.

---

1. System Components

Component	Description	Key Responsibilities
App	Root component, routing, context providers	App initialization, route management
ListingList	Displays list/grid of property cards	Fetch, filter, and render listings
ListingCard	Shows summary info and thumbnail for a property	Display property preview, handle selection
ListingDetail	Detailed view with gallery and map	Show full property info, images, map
FilterBar	UI for filtering/searching listings	Manage filter state, trigger search
ImageGallery	Carousel/gallery for property images	Display images, handle navigation
MapView	Interactive map with property markers	Show locations, handle marker interaction
APIService	Data fetching abstraction	Fetch listings, handle API errors

---

2. Class/Interface Overview

Component/Class	Key Props/State/Methods	Relationships
`ListingList`	`listings`, `filters`, `onSelectListing`	Uses `ListingCard`, receives from `App`
`ListingCard`	`listing`, `onClick`	Used by `ListingList`
`ListingDetail`	`listingId`, `onBack`	Uses `ImageGallery`, `MapView`
`FilterBar`	`filters`, `onFilterChange`	Used by `App`
`ImageGallery`	`images`	Used by `ListingDetail`
`MapView`	`coordinates`, `markers`, `onMarkerClick`	Used by `ListingDetail`, `ListingList`
`APIService`	`getListings(filters)`, `getListingById(id)`	Used by `App`, `ListingList`

Example: ListingCard (TypeScript-like)

```tsx
interface ListingCardProps {
  listing: Listing;
  onClick: (id: string) => void;
}
```

---

3. Data Structure Overview

Model	Fields
Listing	id, title, price, address, images[], coordinates, description, features
Filter	location, priceRange, bedrooms, propertyType, searchText
API Response	listings: Listing[], total: number, error: string?

Example: Listing

```tsx
type Listing = {
  id: string;
  title: string;
  price: number;
  address: string;
  images: string[];
  coordinates: { lat: number; lng: number };
  description: string;
  features: string[];
}
```

---

4. Algorithms/Logic

Listing Fetch & Filter Flow (Pseudocode)

```
onFilterChange(newFilters):
  setFilters(newFilters)
  listings = APIService.getListings(newFilters)
  setListings(listings)
```

Image Gallery Navigation

```
onNextImage():
  if currentIndex < images.length - 1:
    setCurrentIndex(currentIndex + 1)
```

---

5. Error Handling

Scenario	Handling Approach
API fetch failure	Show error message, retry option
No listings found	Display “No results” UI
Image load error	Show placeholder image
Map load failure	Show fallback UI, log error
Invalid filter input	Validate and show inline error

End of Document