



IE-322. Final Report

Real Estate Prices in Google Maps

Spring 2025



Team Members & IDs:

Hamad Adil Alharthi - 2136345

Mohammed Asaad Zahid - 2250957

Abdulrahman Sindi - 2338464

Salman Ali Al Homoudi - 1945850

Khaled Ibraheem Alsaleh - 2236020



DR. Atif Shahzad

Section: HA

Project Overview

The **RealEstateMapApp** is a web-based interactive application designed to help users explore real estate listings on a map interface. The application aims to provide a visual and intuitive way for users to find properties based on location, pricing, and other relevant details. It integrates mapping services and user interface components to display listings with detailed information and images.



Core Features

- **Interactive Map Interface:** Users can view real estate listings pinned on a map, allowing for spatial awareness of property locations.
- **Property Popups:** Clicking on a map marker displays a popup with key property details (e.g., price, type, address).
- **Filtering and Search Options:** Users can search and filter properties based on criteria like price, location, or property type.
- **Responsive Design:** The layout adjusts seamlessly across devices, ensuring usability on desktops, tablets, and smartphones.
- **Image Previews:** Listings include images of the properties to aid in decision-making.

Technologies Used

- **HTML5 / CSS3:** For page structure and styling.
- **JavaScript (ES6):** Core scripting language for dynamic functionality.
- **Leaflet.js:** An open-source JavaScript library for interactive maps.
- **GOOGLE MAPS:** Mapping service providing geolocation and tile layers.
- **Bootstrap (if applicable):** For layout and responsive UI components

How It Works

1. The application loads the map using Leaflet.js and fetches real estate data.
2. Each listing is represented as a marker on the map.
3. When a user clicks a marker, a popup appears showing the property's price, type, and address.
4. The user can interact with map controls to zoom, scroll, and explore listings.
5. Additional UI elements (e.g., filters or search bars) allow the user to refine the displayed data.

Challenges & Solutions

- Data Representation: Converting raw property data into geo-coordinates and rendering it correctly on the map.

Solution: Used GeoJSON format and Leaflet's marker functionalities to visualize data.

- Performance: Handling multiple markers efficiently without slowing down the map rendering.

Solution: Clustered markers or simplified data rendering for better performance.

- Responsiveness: Ensuring UI adapts to different screen sizes.

Solution: Employed media queries and flex/grid layout strategies.

Challenges & Solutions

- Data Representation: Converting raw property data into geo-coordinates and rendering it correctly on the map.

Solution: Used GeoJSON format and Leaflet's marker functionalities to visualize data.

- Performance: Handling multiple markers efficiently without slowing down the map rendering.

Solution: Clustered markers or simplified data rendering for better performance.

- Responsiveness: Ensuring UI adapts to different screen sizes.

Solution: Employed media queries and flex/grid layout strategies.

- A picture from Visual Studio Representing Our Work

