

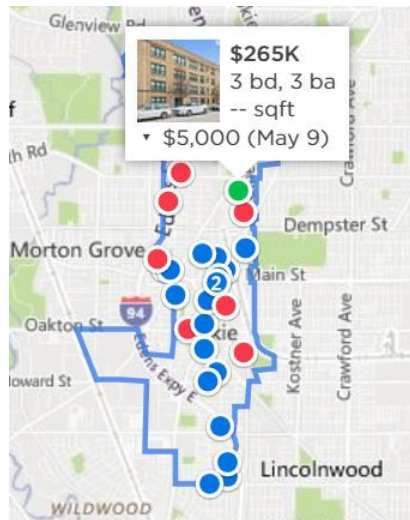


Job Description: Create Dynamic, Live, Customizable Map

Required Features and Attributes:

The map should be completely dynamic: one can easily zoom-in, zoom-out, and move around.

- 1) **Markers:** Show markers according to the latitude and longitude given. Markers should have a pop-up information window when clicked on it. Markers should be of different colors depending on the property type. One should be able to select which property types to view. View [picture below](#) for details.

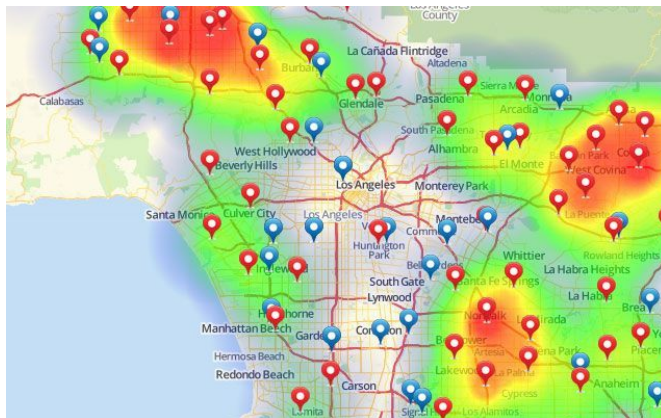


<input type="text" value="SKOKIE Illinois"/>					
Listing Type ^	Up To \$300K v	3+ Beds v	Home Type v		

- 2) **Overlay of Gradient:** This gradient should be based on the data provided. One should be able to zoom in as much as necessary (even at the zip-code level) without any gradient problem. See the image below.



- 3) **Toggle Between Gradient Overlay and Pointmap:** One should be able to toggle among heatmap, pointmap, or both. Simple toggle switches should be made.



Heatmap **Pointmap**




On



Off

4) Select the Data Shown: One should be able to toggle among the data being shown. One can select gradient overlay of dataset 1, dataset 2, dataset 3, etc. A simple dropdown menu should be created for that.



Organize Data By:

Price of the Property

Property Size

Some Other Property

5) Time Scroll: Each property will have a date when it was posted. I want a range selector. Then only property posted during the selected time range is shown on the map (both point map and gradient overlay). The range should be for 10 years, with intervals for each month and week. Additionally, one should be manually able to input (type) start and end date.



6) Draw Polygons: If some addresses are provided, then a polygon should be drawn.

Miscellaneous: It is essential that when a zip code or a county is selected, the boundary of the selected region shows up. With that, this should be well documented. In the end, source code must be provided to us, so our data science and software team can work with the real data. You will only be provided sample data. The source code should be in separate files: HTML, CSS, JS. Each mechanism should be a different function for documentation reason. Lastly, use NodeJS server and Dockerize, generate a docker image to run in a container.

The material in this document is protected by trademark and copyright. Do not reproduce or use in any scope without permission from Steignet.