**MLS Buddy**

**“A super simple crm directly linked to the rmls.”**

**General Info**

The RMLS (Regional Multiple Listing Service) is a local, public platform that Realtors use to sell houses. It has a 'private remarks' section that only Realtors can access. In order to list a house on the RMLS, you either have to be a Realtor or pay a flat fee for a Realtor to list your house on the RMLS. Redfin, Zillow, Trulia and the like have a direct feed of new listings from the RMLS. Unfortunately, the RMLS has maintained a 'scattered' display of listing information. When Realtors look at listings, or send listings from the RMLS to their clients, the viewer is better off looking up the address on Redfin because the information will be easier to read.

Although Redfin has already done a great job of re-arranging the information on an RMLS listing, this display is only available for Realtors who hang their license off of Redfin. This extension will be available for every Realtor, regardless of the Agency where they hang their license.

**Objective**

It is a tool that improves the RMLS experience in ways that make it easier for everyone to use. Initially created as a chrome extension for realtors possibly flourishing into an app for buyers/sellers as well.

Realtors of all ages will be able to:

- Find information on a listing with more ease with highlighted words

- Easily explore the neighborhood with Google Maps in an extension

- Save client info

etc..

So I'll be update the file on my end first (in a folder named 'The Better RMLS') then dragging it into the corresponding github branch when done updating.

**Approach**

I have a repository called The-Better-RMLS-git on github where I will be updating all the files associated with this project regularly.

Since I don't need a branch created for every person on this project because it's just me; I will be creating a branch for every part of the project;

- Project-Management: containing this doc and other docs that outline my process. This doc will be updated with all of the changes my process goes through during its execution.

- Phase 1: containing the code associated with phase 1- an extension that simply highlights certain words on particular websites.

-Phase 2: …

- etc…

**Phases**

**Phase 1: Highlighting Text**

Event: visiting Redfin, RMLS, Zillow, Realtor.com, Trulia

Text to be highlighted:

- Address -listing price – property taxes

-Sqft - squarefeet -total SQFT -school -neighborhood -beds -baths -year built

-school -neighborhood -DOM -Days on market -Parking -sewer -heat -cool

Next Open house?

**Tasks**:

1. Create manifest.json with content script info and other relevant information

2. trigger the highlight function when those urls (with wildcards) are visited.

**Phase 2: Landing popup**

Event: Clicking the icon (whether on a listing site or not)

A search box requesting an address (or listing #). Clicking search pulls info from the easiest source (probably Redfin).

Tasks:

- Create a CSS container with a width and height border (if necessary)

- Text: 'address'

- Text box

- 'User input' searches on Redfin (or whatever site) and opens results in content with highlighted results (which will be there anyway becase one of the urls was visited)

**Phase 3: Display highlights on Popup**

Event: Clicking on the icon while on the listing

Text to be highlighted:

[square 1]

- Address -listing price – property taxes

-Sqft - squarefeet -total SQFT -school -neighborhood -beds -baths -year built

[square 2]

-school -neighborhood -DOM -Days on market -Parking -sewer -heat -cool

[square 3}

Next Open house?

Tasks:

**Clicking when on listing**:

- I*n the* ***content*** *script* collect all the information between the space before the highlighted and two or three spaces after it (depending on the information that's being displayed) into a variable. Each square has its own variable. If the formatting for this doesn't look right, then create a variable for each individual piece of information.

- *In the* ***background*** *script*mention that when URLS are visited, i = 0 (or whatever arbitrary reference makes sense)

- In *the* ***popup*** *script* when i=0 (when the URLS are being visited), display the variables associated with the highlighted info. \*You may have to specify to only display the highlighted when there is one of each variable (if more than one of purchase price variable i=1), this way, when a bunch of different

- *In the* ***popup*** *script* display the variables that were collected in the content script

**When 'search button' on popup has been clicked**:

- Event: address searched on popup

- Tasks:

display highlighted info when i=0 is TRUE

**Phase 3: Listing Pics On Popup**

Event: opening a listing with the

**Phase 4: Add Google Maps**

Event: Opening a listing in the popup