**D-532 Applied Database Final Project**

**Team: Amit Banerjee, Pravallika Pentapati, Sahil Dhingra.**

**(Summer 2022)**

**Final Project Phase-3: WebApp Design**

**Table of Contents**

[Part 1. Web App Architecture 3](#_Toc108955981)

[Overview 3](#_Toc108955982)

[1.Data Storage: - 3](#_Toc108955983)

[2.Back End Languages 3](#_Toc108955984)

[3.Database Access/Connections/Security: - 3](#_Toc108955985)

[4.Front End Languages: - 4](#_Toc108955986)

[5. Deployment 4](#_Toc108955987)

[6. App Interactivity 5](#_Toc108955988)

[7. Architecture Diagram 5](#_Toc108955989)

[Part-2: Web App Layout 6](#_Toc108955990)

[1.Initial layout: 7](#_Toc108955991)

[2.Sign up page: 7](#_Toc108955992)

[3.Login: 8](#_Toc108955993)

[Warning message: 8](#_Toc108955994)

[Success Message: 9](#_Toc108955995)

[6.Search Listings: 9](#_Toc108955996)

[7.Add Listing page: 10](#_Toc108955997)

[8.Update/Delete Listing Page: 10](#_Toc108955998)

[Example: Change feature form 11](#_Toc108955999)

[9.Delete listing: 12](#_Toc108956000)

[10.Statistical plots: 13](#_Toc108956001)

[Part-3: Assessment Table: 14](#_Toc108956002)

Webapp Design

# Part 1. Web App Architecture

## Overview

Model: MySQL is being used as the model layer to store and model data.

View: For the front end (view) layer of the application we are using the Streamlit library in python.

Controller: We are using python as our backend controller. This is where all the application logic resides.

## 1. Data Storage: -

MySQL is used as the backend data store by our application. The real estate data is organized into normalized data structures to ensure CRUD performance for the application. The entire data model, applicable constraints and references have been defined in the second phase of the project.

Currently the database is locally hosted, but the recommendation for a production version of the app is to install the DB on AWS RDS or Heroku server or an equivalent Cloud service.

## 2. Back End Languages

For controlling the app flow the middle layer controller language being used is python. We are using python libraries mysql.connector and pandas extensively to control the application interactions.

## 3. Database Access/Connections/Security: -

To connect to the backend database, we use the MySQL Connector/Python which enables python programs to access MySQL databases using an API that is compliant with the Python Database Specification v2.0 (PEP 249).

For the local app version, we connect using a secrets file in the apps’ root directory as shown below. For a production deployment, the plan is to store the secrets in a more secure mechanism like Hashicorp Vault or AWS Secret Manager.

Text

Description automatically generated

Text

Description automatically generated

## 4. Front End Languages: -

For the front-end UI layer, we use the Streamlit framework and library in python. Streamlit is an open-source framework to rapidly build and share webapps. Streamlit abstracts a lot of the front-end programming complexities of html and jsp’s and makes web development very simple for data engineers and scientists.

Text

Description automatically generated

## 5. Deployment

Currently app is hosted locally but we are in process of hosting the app on Heroku in upcoming week.

## 6. App Interactivity

We have created an app for real estate listings. We have divided CRUD operationality on multiple screens. This is because we do not want user to delete or update a listing, that should be the task for the admin. Below are the details on operations:

View Listing: On the “Search Listings” page, User can view the listings. To help user navigate to their needs, there are several search criteria which can be deployed. For example user can select Price range, City, State, Number of Bedrooms/Bathrooms etc. There are dropdown for all reference data elements. Once user has selected all relevant filter conditions, user can click search to view the listings based upon that.

Insert Listing: On next page we have “Insert Listing” utility where admin can add a listing to the database through UI. Here, all relevant details can be selected and/or entered depending upon the field type. User will select data for Listing Type, Property Type, Bed/Bath details etc.. In addition, user will add address details and click save to process the record. Once save button is clicked, a procedure in the background will check for business rules. If all rules are satisfied, an entry will be made and User will be shown corresponding “Listing ID”. If there’s business rule failure, user will be shown respective error message. For example, if user adds duplicate data on address, system will show error message “Duplicate Address Entry” and will not add that to database.

Update and Delete: Next, we have utility for update and delete where admin will be able to enter or select the listing ID, and kind of operation that needs to be performed on that record. If user selects Delete listing form and choose an listing ID and click “Delete” button, that entry will be hard deleted from the system.

For update, user has multiple options. Here user has an option to select any of the operation like Change Address, Change Listing Type (ex. mark as SOLD), Change price, features (bed, baths, acre lot, size). Depending upon form selection, relevant updated data can be entered to selected forms. Once done, user can click “Update Task” button, queries will be fired in the backend to make the necessary changes as selected by the user.

Visualization: On visualization page, user has option to select the kind of plot the want to see. Multiple plots have multiple search criteria to show. For example, If user wants to see average housing price by region, then user can select if the region should be States, or Cities in a state, or Zip codes in a state. Depending upon the selected filters, plot will be generated.

## 7. Architecture Diagram

Graphical user interface, diagram

Description automatically generated

# Part-2: Web App Layout

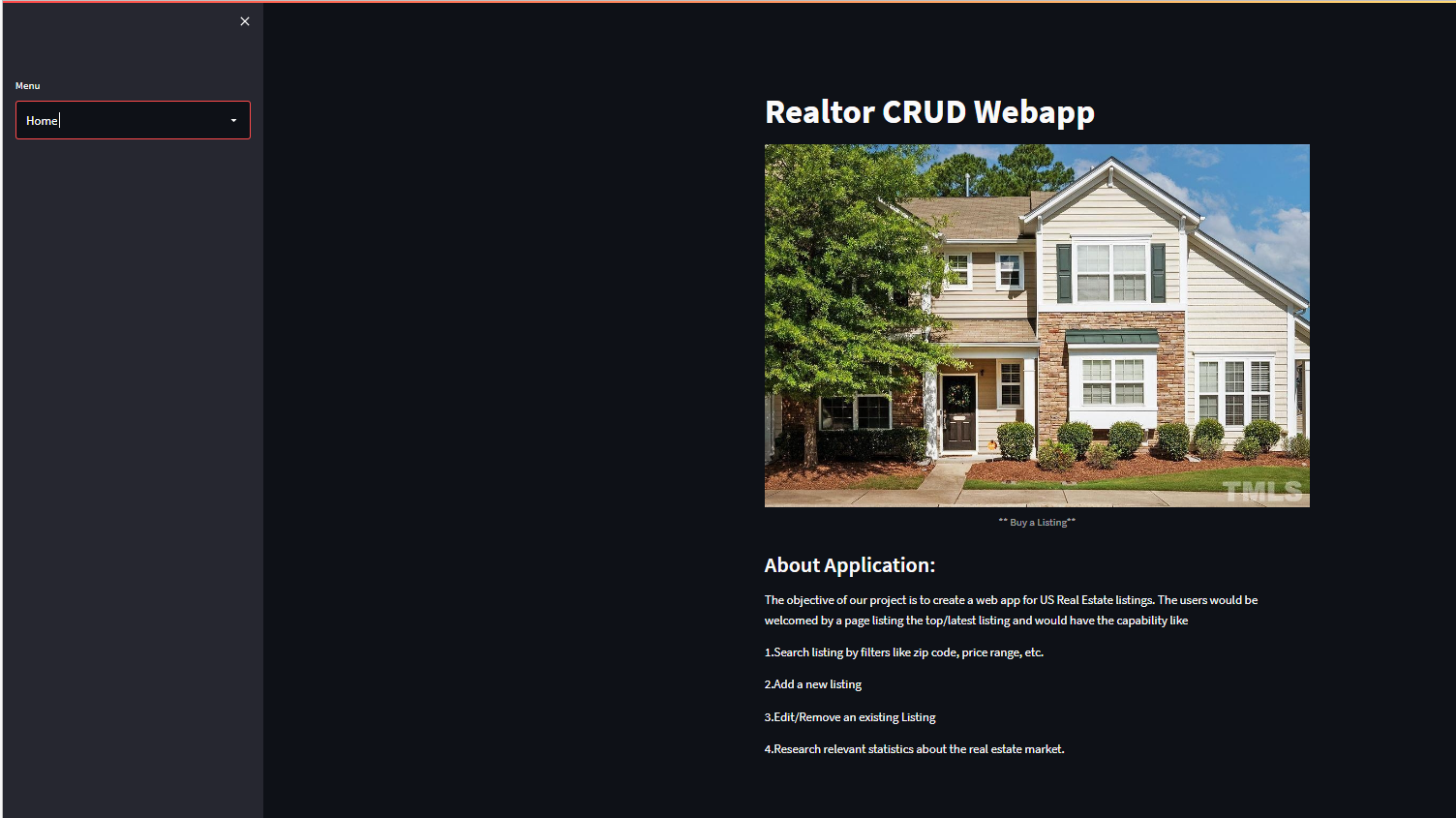
We used Streamlit with python to design the web page layouts for our application. There will be a total of 5 main pages as per the below screenshots

The architecture of the web page is as follows:

1. User/Admin can read about the application on the home page
2. Admin will be able to login into the application from the home page and can manage the listings.
3. Admin can add, update, delete and view a record after successful login.
4. Admin can add a new listing from ‘Add Listings’ page.
5. Admin can modify/delete the listing from the Update/Delete listing page
6. Admin can view statistics through visualizations on the statistical plots page.
7. Users can view all the listings as per the requirement using different filters on the search listings page.

Color schema: The color schema of the application is completely in simple white and Black color. The black color was chosen as the background and White color for the text labels. On mouse over, the text turns red.

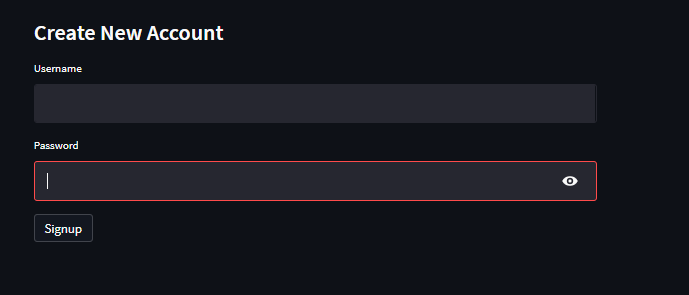
## 1.Initial layout:



In the application, we have a menu tab available on the left side which is in form of a drop-down. Users can select Home, Login, Signup, and Search listings as per their needs. Upon successful login, other options to manage the listings like Add listings, Update/Delete listings, and Statistical plots are visible.

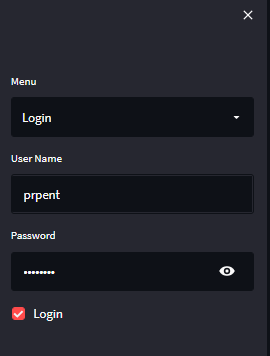
## 2.Sign up page:

On this page, the user will be able to create a new account to manage the listings. Two input boxes were provided to fill the required information for sign-up and a button to submit as shown in the below screenshot.



## 3.Login:

We also have a login option available on the sidebar of the menu. We have a login available as a check box. Username and password are available as input text boxes as shown in the below screenshot.



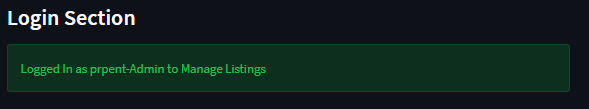
### Warning message:

If the user types an incorrect username and/or password, then a warning message will be displayed as shown in the below screenshot.



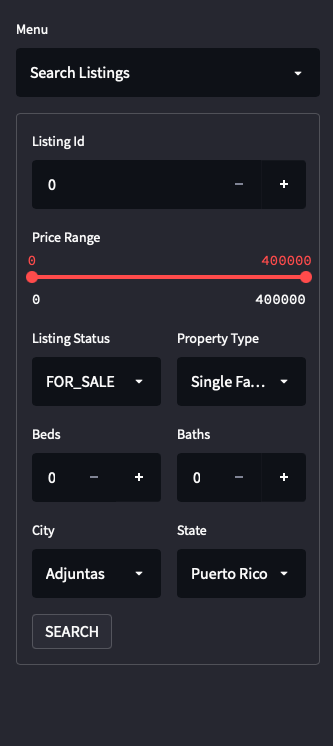
### Success Message:

If the user enters the correct username and password then a success message will be displayed as shown below.



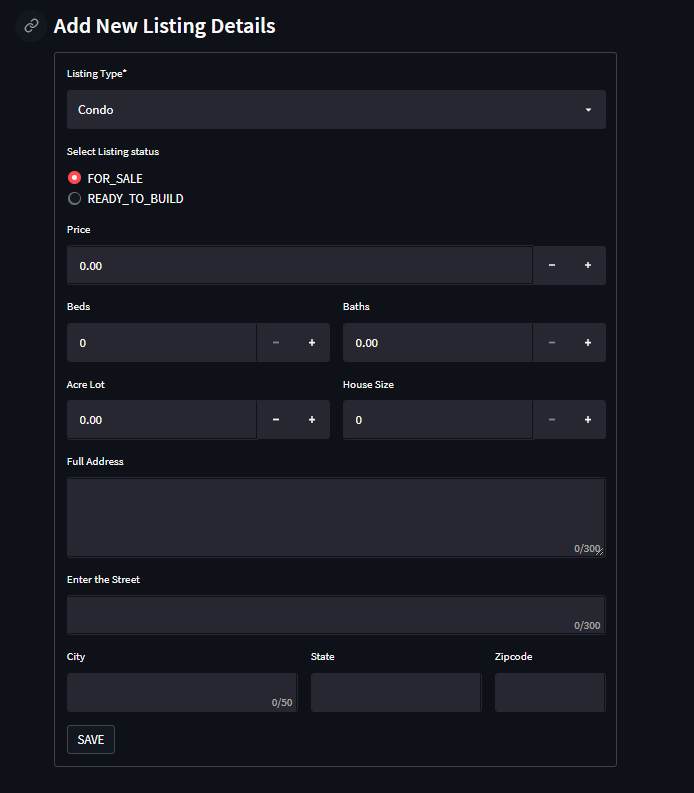
## 4.Search Listings:

On the search listings page, the directory listings based on user requirements are displayed. User always has the choice to add/remove additional filters leveraging the parameters like filter by city or filter by state or Property type, etc. Based on the filter, the listings will be displayed in the search listings page. No login is required to perform a search on listings. Everyone can search the listings.



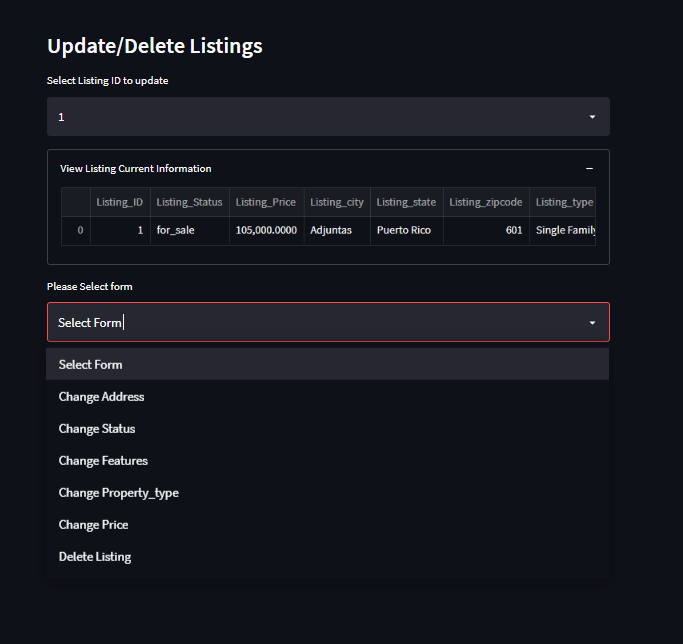
## 5.Add Listing page:

On this page, Admin will be able to add a new listing record. Once the admin fills the information in all required fields then a success message will be displayed along with the Listing id number on add listing page.



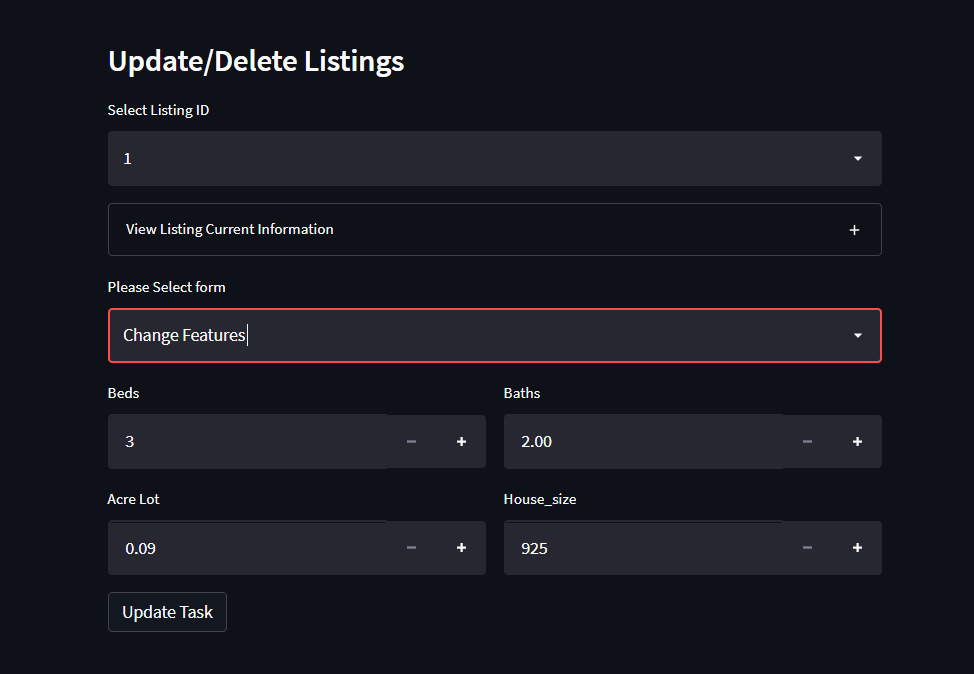
## 6.Update/Delete Listing Page:

On the update/delete listings page, Admin can search the listing id using the listing id search box. The admin can also view the current information of a particular listing id on the same page. Admin has the option to update the forms as per requirement.



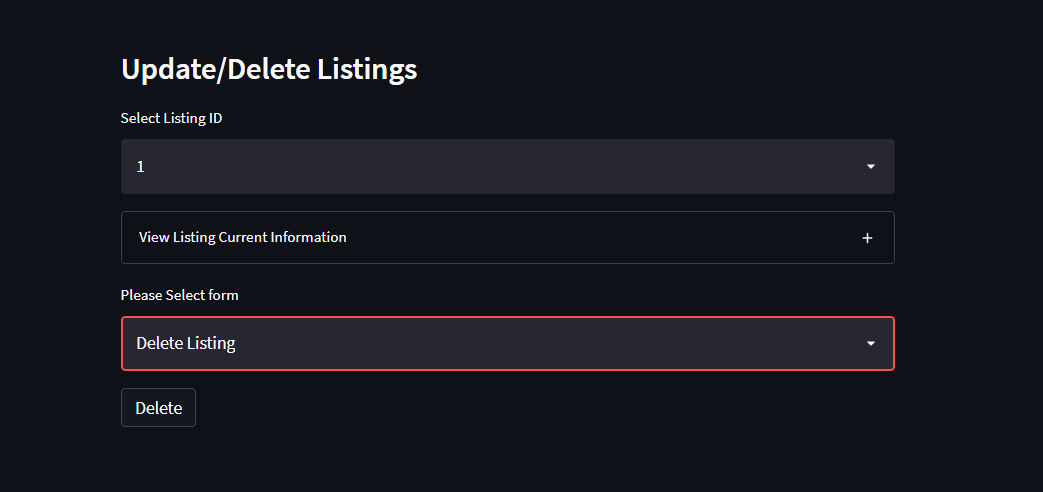
### Example: Change feature form

If we select the change feature form then existing information will auto-populate for all the fields and the admin can update any fields in the form as per the requirement after changes are done then we need to select the update button to save the new changes and we will show the success message as “Record is updated”.



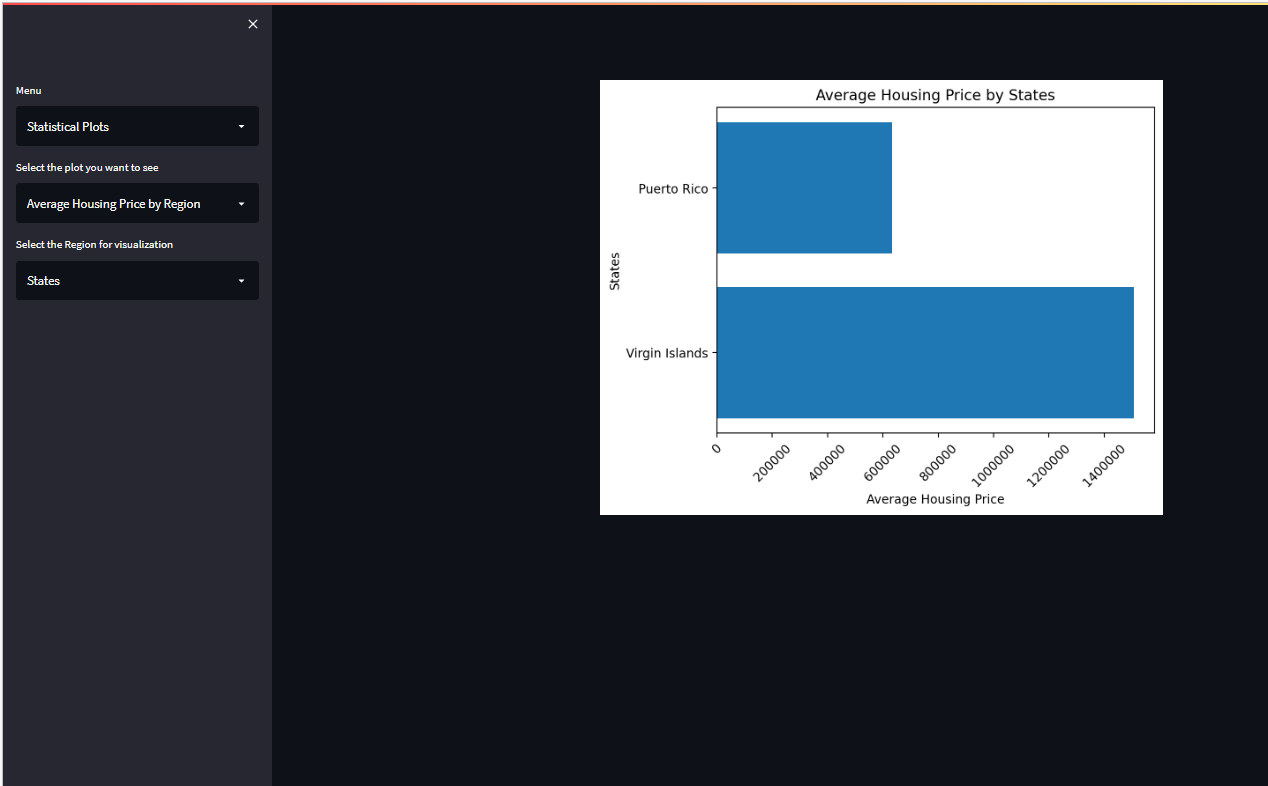
## 7.Delete listing:

if the admin selects form as delete listing, then the delete button will display, once the admin clicks it then a particular listing id will be deleted.



## 8.Statistical plots:

In the statistical plots page, Admin can view statistics about house prices for a particular city or state or zip through visualization we have two drop-down buttons for select plots and select region.



# Part-3: Assessment Table: