**D-532 Applied Database Final Project**

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

**(Summer 2022)**

**Final Project Phase-4: Technical Report**

**Table of Contents**

[Project Title:. 3](#_Toc109863244)

[Team Name: 3](#_Toc109863245)

[Application URL: 3](#_Toc109863246)

[Full GitHub URL: 3](#_Toc109863247)

[Project Summary: 3](#_Toc109863248)

[Tools Used: 3](#_Toc109863249)

[Data Storage Tool: - 3](#_Toc109863250)

[Back End Languages: - 4](#_Toc109863251)

[Database Access/Connections/Security: - 4](#_Toc109863252)

[Front End Languages: - 4](#_Toc109863253)

[Deployment: - 4](#_Toc109863254)

[Data Used: 4](#_Toc109863255)

[List of Functionalities for Users: 5](#_Toc109863256)

[User Functionalities 5](#_Toc109863257)

[Teamwork: 6](#_Toc109863258)

# Project Title: D-532 Applied Database US Real Estate Listing Site.

# Team Name: Data Wizards.

# Application URL:

We have hosted an application on Streamlit that allows to build, deploy, and operate applications in the cloud without exposing sensitive data, having a free presentable app hosted on GitHub for the sensitive data

URL: <https://prpent-database-final-project-real-e-real-estate-listing-dhaxum.streamlitapp.com/>

# Full GitHub URL:

We have uploaded the application source code with github.iu.edu at following URL.  
URL: <https://github.com/prpent/Database-Final-project-Real-estate-listings-webapp>

# Project Summary:

As per the statistics from [review42](https://review42.com/resources/real-estate-statistics/), 50% of buyers find their new home online. In addition, with the record number of tech-savvy millennials jumping into real estate, this number will only grow. Looking at this interesting fact and using real estate data from Kaggle, our application will showcase the house listings in the US.

# Tools Used:

We have implemented entire application with MVC (Model View Controller) architecture

pattern and Python language for server-side programming.

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.

### Data Storage Tool: -

MySQL is used as the backend data store by our application. The application DB is hosted as Clear DB MySQL on Heroku.

### 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.

### 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).

A secrets.toml file is added to the application settings which contains the connection details required to connect application with MYSQL database hosted on Heroku.

That’s the only place where connection details exist, which makes the database secure and free from any maligned uses. Only application owner can access the secrets file.

### 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.

### Deployment: -

Our app is hosted on Streamlit Cloud using our GitHub code repository. Using this we can now connect our database (remotely hosted on Heroku) to Streamlit cloud and run our web app without exposing sensitive data or getting trouble with a very large amount of data.

# Data Used:

This dataset contains a list of house sale prices in the USA region. Data was scraped from a real estate listings website operated by the News Corp subsidiary Move, Inc. based in Santa Clara, California. It is the second most visited real estate listings website in the United States as of 2021, with over 100 million monthly active users, then hosted data on Kaggle in CSV format for performing the statistical analysis and predicting the house prices. We used this dataset for building the web app in the final project. This dataset in Kaggle will be updated frequently. There are 203,216 entries and 12 columns. We cleaned the data by adding more columns and removing duplicates. The dataset in Kaggle could be found at <https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset>.

# List of Functionalities for Users:

Users will be able to find their preferred home based on multiple search criteria. In addition, we will provide potential home buyers with relevant statistical visualizations to make an informed decision. There would be functionality for listing a new property and making amends to already existing ones. There will be a total of 5 main pages

The architecture of the web page is as follows:

1. All users can read about the application on the home page.
2. Registered users will be able to login into the application from the home page and manage the listings.
3. Registered Users can add, update, delete and view a record after successful login.
4. Registered Users can add a new listing from ‘Add Listings’ page.
5. Registered users can view statistics through visualizations on the statistical plots page.

### User Functionalities

View Listing:

On the “Search Listings” page, User can view the listings. To help users 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. in any combination. There are dropdowns for all reference data elements which directly fetch data from backend. Once user has selected all relevant filter conditions, user can click search to view the corresponding listings.

Insert Listing:

On the 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 the 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 a success message will be shown with corresponding “Listing ID”. If there’s any business rule failure, corresponding error message will be shown on the screen. For example, if address data in entered again for a new entry, 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 chooses a listing ID and clicks the “Delete” button, that entry will be hard deleted from the system.

For updates, the user has multiple options. Here the user has an option to select any of the operations 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 into selected forms. Once done, the user can click the “Update Task” button, and queries will be fired in the backend to make the necessary changes as selected by the user.

Visualization:

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

# Teamwork:

|  |  |  |
| --- | --- | --- |
| Team member | Questions | Comments |
| Pravallika Pentapati | Are you satisfied with the task completion (scale 1-10) | Yes, I am satisfied with the task completion. (10) |
| Team Work | The three members of our team are collaboratively working together and share the work portions equally. I think we achieved the collaborative effort as a group to achieve a build the app. |
| Time Commitment | We worked a good number of hours towards the project deliverables for each phase. |
| What could be done better in future: | we achieved the primary objective of the app for implementing the CRUD functionalities. I think in the future we can enhance the app by adding some more additional functionalities like logout, a screen to show the annual percentage rate, and monthly payment. |

# 