Spring 2024 CS 230 Final Project Description and Policies

Part 1. Design and queries

The purpose of the **design** phase is to start thinking about what you might do before you jump into coding. Identify at least three different **queries** or questions you can ask about your data set. Try to phrase your questions so that they can have a parameter which can come from user input.

For example: (and these **queries** don't match the data sets you are given but are here to inspire you!)

- What's the cost of the most expensive <house_type> in <city>?
- Find all of the apartments in <city> that rent for under <amount>.

Then think about the interactive widgets from Streamlit that you can bring to your application to obtain user input. For example: you can use a numeric slider to have the user enter a monthly rental amount.

Next, think about how you will visually present the data or query results using charts, graphs, tables, or maps. Be sure your web pages and visualizations are "user friendly" and as "polished" as possible. Be sure to label controls requiring user interaction, make sure your charts have titles, legends or explanations that would be helpful to the user. Think about how the user will navigate from one part of your site to another.

Part 2. Coding

Create your Python application with a Streamlit UI and several visualizations.

Create charts and graphs of different types with custom legends, axis labels, tick marks, colors, other features), and at least one map showing locations and data based on latitude and longitude. Your application should tell a story, so be sure elements are labelled appropriately, and add any narrative that will help the reader understand your visualizations and to cue the reader about which values to specify, and the purpose of each chart or graph. You may wish to add a few sentences explaining each chart as a paragraph of text on the screen.

You might also use pandas to create summary report based on the data itself (max/min values, relationships between columns, etc). See the documentation for how to use different Streamlit features. You might make use of sidebars to place your widgets, multi-page applications, or caching to improve performance.

Read the documentation for PyDeck Maps. Our examples of maps in class were PyDeck's Scatterplot Layer or IconLayer, but PyDeck support several other styles such as Text and Heatmaps. Have a look. To explore another chart library, consider Seaborn charts which have additional chart types and customization options. You might also look at Folium maps (here's a simple tutorial) if you'd like to play with a different mapping library.

If your project contains more than one Python code file (i.e., one or more Python code files and images), create a zip file containing all of your project files and submit it. You do not have to submit the data file that you used.

Part 3. In-Class Presentation

The presentation will occur during the final exam period. You will be presenting to your instructor and answering questions.

You will have **five minutes** to present your project. Create a 2 minute video demonstrating the project's functionality, in which you give an overview of your project's capabilities. In this video, demonstrate what you feel

is the most interesting part of your project, and then **be prepared** to show how you implemented it in your program and answer questions on the implementation. You will be graded on the quality (not quantity of features) of your product, your understanding of how your code works, and your ability to modify it on request.

The video and in-class presentation are mandatory. Failure to show up and present your project will result in **a zero** for your final project grade. More details may be shared in class and by email.

Requirements

See the grading schema that defines **requirements** and refers to **features** that are defined below.

Grading

The Final Project counts for 20% of your final grade and is worth 50 points, as follows:

Requirement	Points
Project description document	2
Features	
Python Coding Features : at least 3 @ 2 points.	6
Streamlit Features:3 controls/widgets and at least one page design feature.	6
Visualizations: 2 charts and one map.	7
Data Analytics Capabilities (at least 4 – sort, filter, etc).	10
Quality and style of code: Well documented, efficient, modular code.	4
Presentation in class:	
Pre-recorded **video.**	5
Answers to questions.	**10**
Total:	**50**
Extra Credit: Publish to Streamlit Cloud	**2**

Project description document:

In 2 pages maximum, include a summary of all features/pages: how to find them and what they do. This document will serve as a guide to features during grading.

Features

Python Coding Features:

- A function with two or more parameters, one of which has a default value
- A function that returns more than one value
- A list comprehension
- A loop that iterates through items in a list, dictionary, or data frame
- At least two different methods of lists, dictionaries, or tuples.

Streamlit Features:

- At least three Streamlit widgets (sliders, drop downs, multi-selects, text box, etc.)
- Page design features (sidebar, fonts, colors, images, navigation)

Visualizations:

- At least two different charts with titles, colors, labels, legends, as appropriate
- At least one detailed map (st. map will only get you partial credit) – for full credit, include dots, icons, text that appears when hovering over a marker, or other map features

Data Analytics Capabilities:

- Sorting data in ascending or descending order, by one or more columns,

- Filtering data by one condition
- Filtering data by two or more conditions with AND or OR
- Computing summary statistics (total sum, max, min, etc.)
- Add/drop/select/create new/group columns, frequency count, other features
- Text analysis, etc.

Usual rules about writing "good" code apply:

- Make your code as modular and easy to follow as possible.
- Include a docstring, comments, and meaningful variable names.
- If you did something "cool" in your code that you are incredibly proud of, please write a comment call attention to what you did.
- If you referred to any online articles or other information beyond class examples, please be sure to list them as references / comments in your code.
- Make sure the program runs and the output is correct.

Documentation String

Use this documentation string at the top of your Python code file:

```
"""Name:      Your Name
CS230:     Section XXX
Data:      Which data set you used

Description:
This program ... (a few sentences about your program and the queries and charts)
"""
```

Submission

Submit your Python file (or zip file containing multiple files) and a video to Brightspace the **day before the final** exam (presentation) date. Do NOT submit the data file.

Getting Help and Academic Integrity:

- Do not discuss the overall design of your program with anyone other than your instructor.
- **You are prohibited from seeking help from any individual other than your instructor or CIS Sandbox tutors. Tutors can help you debug your code.**
- **You may use generative AI code co-Pilots to the extent that it helps you implement the individual features. The design and thinking behind the project must be your own. If you use the generated code, you are responsible for understanding how it works and should be able to modify any features you include in your code on request during the presentation.**
- **Any violation of these policies will result in a zero for the project at minimum or even a grade of 'F' for the course.**