Updated: January, 24, 2022

Data Analysis Course 1 Project Requirements

Project Deadline: Last Friday of the class, by 11:59am (noon)

Please read this document in full. Make sure to ask any questions before the project is due. If there is any confusion please ask a mentor or a member of the Code Louisville staff.

The purpose of the capstone project is to reinforce what you've learned and show off your skills. Your projects will be your go-to tool to show off to potential employers and demonstrate your knowledge. Projects should show that you have a good understanding of Python, it's methods and implementation.

This is your project, make something fun and challenging. You are hoping to impress an employer with it!

What to expect:

Weeks 1-3: Immerse yourself into Python and it's methods. Think about the types of programs you have been exposed to. Go over this document with mentors.

Weeks 4-5: As you continue to learn and work through your videos, think of ideas that sound fun and interesting. Think about and practice implementation.

Weeks 6-10: Project should be worked on in this time frame. You should discuss ideas and implementation with your mentors as you move forward.

Week 11: The project **"Turn In Form"** should have been sent out. Please submit your project ahead of the due date. You are able to keep working till the due date/time. **Last Friday of the class; by 11:59am (noon).** Project should be tested, and looked over by a mentor.

Week 12: Testing should be done, tidy up the codebase and finalize your project, make sure you have commits to Github. Project should be submitted. **Late projects will not be accepted, failure to meet the project requirements will result in not completing the class.**

Project Requirements Overview:

Create an application that demonstrates your knowledge of the Python development skills you've been learning. This application can be anything of your choosing **as long as it demonstrates the below requirements.**

REQUIREMENTS:

The below items are the requirements for your personal project for the Python class. Projects are graded on a pass/fail system, thus **every requirement** must be met in order for your project to pass. You should discuss your project/plan with mentors regularly to ensure you're on the right path to passing.

Program staff and mentors will review projects after they are turned in and will determine if your project meets these requirements:

- The project is uploaded to your GitHub repository and shows at minimum 5 separate commits
    - **Using GitHub's file uploader does not count as a check-in.** You must upload via Git
    - Gitignore should be used to keep any secrets/passwords used to access APIs / data sources out of the Github repository
- The project includes a README file that explains the following:
    - A one-paragraph or longer description of what your project is about.
    - **Relevant packages that need to be installed to run the project.**
    - Which 3+ features you have included from the below lists to meet the requirements
    - Any special instructions are required for the reviewer to run your project. (For example: "**run python main.py" from the command line**)

    - Guide to using markdown for README.md files (https://guides.github.com/features/mastering-markdown/)
    - Describe the data used in the analysis in the README file.
- The project should implement a simple data analysis by reading data, performing calculations on the data, and displaying the results.
- Choose **at least** 1 item from each category on the **Features List** below and implement them in your project
    - *We recommend you pick a 4th item (or more!) to add, just in case something goes wrong with one of your other items - 3 is only the minimum requirement*

FEATURE LIST:

- Category 1: Python Programming Basics:
    - Implement a “master loop” console application where the user can repeatedly enter commands/perform actions, including choosing to exit the program.
    - Create a class, then create at least one object of that class and populate it with data. The value of at least one object must be used somewhere in your code.
    - Create a dictionary or list, populate it with several values, retrieve at least one value, and use it in your program.
    - Create and call at least 3 functions or methods, at least one of which must return a value that is used somewhere else in your code. To clarify, at least one function should be called in your code, that function should calculate, retrieve, or otherwise set the value of a variable or data structure, return a value to where it was called, and use that value somewhere else in your code. *For example, you could create a function that reads how many items there are in a text file, returns that value, and later uses that value to execute a loop a certain number of times.*
    - Implement a regular expression (regex) to ensure a field either a phone number or an email address is always stored and displayed in the same format.
    - Build a conversion tool that converts user input to another type and displays it (ex: converts cups to grams).
    - Calculate and display data based on an external factor (ex: get the current date, and display how many days remain until some event).
    - Analyze text and display information about it (ex: how many words in a paragraph).
- Category 2: Utilize External Data:
    - Read data from an external file, such as text, JSON, CSV, etc, and use that data in your application.
    - Connect to an external/3rd party API and read data into your app
    - Connect to a database and read data using SQL.
- Category 3: Data Display
    - Visualize data in a graph, chart, or other visual representation of data.
    - Display data in tabular form
- *Category 4: Best Practices*

    - Implement a log that records errors, invalid inputs, or other important events and writes them to a text file.
    - Create 3 or more unit tests for your application.
    - The program should utilize a virtual environment and document library dependencies in a requirements.txt file.
- *Other features can be added to this list with mentor or staff permission, but we want to see you stretch your skills, so you'll want to pick something challenging.*

"STRETCH" FEATURE LIST:

These count too! But they will require going outside of the base curriculum to learn about and may be more challenging.

- Implement a "scraper" that can be fed a type of file or URL and pull information off of it. For example, a web scraper that lets you provide any website URL and it will find certain keywords on the page.
- Implement optical character recognition (OCR) that you can upload PDFs to and it will generate the text.
- Use pandas, matplotlib, and/or numpy to perform a data analysis project. Ingest 2 or more pieces of data, analyze that data in some manner, and display a new result to a graph, chart, or other display.
- Use a Jupyter notebook to document your data analysis.

Clarifications and Commonly Asked Questions

- **How projects are reviewed**
    - The project reviewer will use git to clone your project to their local machine and follow the directions listed in your readme file.
    - They will look for the features you documented in your readme, verify they work properly, and display an understanding of the code.
    - If all requirements are met, your project passes.
    - We are looking to ensure you understand what you wrote, so a poorly implemented project may still not pass if we believe you do not understand the code.

- **Naming your project**
    - Choose a name for your GitHub repo that is relevant to the subject of your project. Do not name your repo "CodeLouisvilleProject" or similar. Choose a name based on what your project is about. For example "Brian's House of Pancakes", "Recipe conversion calculator", etc.

- **GitHub**

    - Yes, GitHub is a requirement. It’s not just how we find and view your project, it’s a critical skill you need to understand. Not having your project checked in on GitHub will result in not completing the class.
    - 5 commits is a minimum to show you’ve made multiple updates. Hopefully, you have dozens of commits!

TESTING YOUR PROJECT

You should test your project on another computer by having someone else obtain and run your project. Several projects in the past have failed to meet requirements because it was written in a way that only worked on that person's computer and not the reviewer's. Your project reviewer will not be responsible for tracking down why your project does not work and this may result in a failed project.

MENTORS

You should talk about your project early and often with your mentors. Explain your idea and the features you are thinking of implementing. They will help you understand if it will meet the requirements or if you've possibly decided to tackle too large of a project for the 12-week time period. Towards the end of the session, you should again show your project to your mentors and get confirmation that it meets the requirements of the project. You are perfectly fine to ask them to confirm so there should be no surprises about whether your project will meet the requirements when it comes time to submit.