**Final Project Description and Requirements**

The Final Project consists of designing, developing, and presenting a Python-based application of interest to solve a problem of your choice. You can think of your product as a commercial, for-profit product, or as a non-profit, public service product.

Your application will need to scrape or otherwise obtain data from three (or more) different online sources, either by web scraping or using API access. The code that you write will then need to clean, organise and analyse the obtained data to produce interesting results for the user to look at (statistical results, tables, plots, ...) and perhaps interact with.

Since this is a Python course, all of this will need to be in Python code: you may NOT download data using a browser, clean and organize the data in Excel, and then just do the analysis and display output in Python. Your entire project must be Python from beginning to end.

# Group Project Ideas

What decisions do you make yourself that involve looking at multiple data sources, probably on the web?

For example, if you would like to take a vacation, you might look up:

1. Available flights and costs
2. The predicted weather where and when you plan to go
3. Available hotels and maybe rental car information
4. Restaurants and entertainment events

One past project combined these sources of information to give a user a single application for vacation planning.

If you want to eat a nice meal, you might:

1. Select the type(s) of food you are interested in: Mexican, French, BBQ, ...
2. Look up menus, prices, and ratings of nearby restaurants offering that kind of food
3. Look up highly rated recipes that you could make at home, perhaps with cost estimates

Think about what information YOU use from multiple sources to decide about where to go, what to buy, what to invest in, which player to draft in your fantasy sports league, which doctor or dentist to go to, where you want to live, etc.

***One restriction: no movie recommendation applications: these are too easy, and have been done too many times.***

# Deliverables

There are 4 main deliverables for the final project – (1) a draft “pitch deck” explaining the proposed outline of the project, (2) a final slide deck once the project is complete, (3) a group video explaining and demonstrating the end product and finally (4) the codebase used to create the final output.

The following sections discuss these deliverables in more detail.

## Project Draft – Pitch Deck 10 points

The first phase of the Group Project will consist of:

1. **Form a Group.** The Group should have 4 members. Students will be assigned randomly to groups initially, but you may choose to switch between groups. Ultimately, groups should be between 3 to 5 in size, no smaller and no larger.
2. **Decide on the product or solution to develop.**
3. **Develop and submit a "Pitch Deck"**, that is, a draft PowerPoint presentation describing your group, your product and ideas. During the development of your product, you will revise this Draft Pitch Deck into a Final Presentation Deck, from which your group will make its presentation during the last week of classes.

The **draft\_pitchdeck\_template.pptx** will be available in our Canvas page.

For this project you will need to obtain data from THREE or more web sites, at least one of which must be scraped: more sources and a larger variety of data is better than less. In lectures, we will look at using Beautiful Soup to scrape tabular data from a web site. **If you wish to scrape data from an active JavaScript web site, you may have to learn about Selenium or Scrapy, as well.** You can also pull data from an API at a web site. Using different scraping/downloading methods in your Group Project code will be more impressive than using just one method.

You can think about creating a commercial product like an app, or instead think about a solution that provides some benefit to society in the form of a Python program, and aim your pitch at policy decision makers. In the end, either way, **you are making a pitch about your identified problem area and potential solution**.

You may create your own pitch deck from scratch or modify the template with images, change the colors, use photos, etc as long as it contains similar information in a similar order as shown in the template.

The goal of the presentation is to lay out your thoughts, ideas and potential solutions. Do not get too worried about the “flash” or quality of the content, the most important thing is to collect your thoughts and make a strong case for developing your intended solution.

In Phase 1, you produce a DRAFT pitch deck. It does not need to be perfect, and *you can change it* as your Group Project evolves and develops, but the draft pitch deck should be complete and coherent. At the end of the Project, you will have to deliver a FINAL presentation deck, which should be more professional and polished. This FINAL presentation deck will also be the basis for your Group Presentation.

## Final Slide Deck – Project Prototype 10 points

The goal of the Group Project is to use THREE sources of data, with AT LEAST ONE obtained by web scraping, others perhaps by direct download, by using an API, or whatever makes the most sense. In class we will investigate Beautiful Soup, but you may also wish to investigate Selenium, Scrapy, or other web scrapers.

Hint: Commercial web sites often detect and try to prevent web scraping. It is often possible to work around this, but it can be a challenge. Government web sites, which are excellent sources of economic, financial, health, crime, and other kinds of data, usually do NOT try to prevent scraping.

You may NOT use data from a "data set aggregator" site like Kaggle, where the data sets have already been cleaned for you. One purpose of the project is to gain experience dealing with dirty data. But if the original source of data provides clean data, that's fine.

By the time of this deliverable’s due date, you will have learned about basic Python: int, float, str, bool, None; arithmetic operators; lists, loops and range(); equality, relational, and logical operators; if/elif/else decision; built-in data structures (list, tuple, set, dict, with named and symbolic operators); sequence slices; user and text file input/output; and Beautiful Soup for scraping HTML web sites.

But note that we will be learning more, including NumPy for N-dimensional arrays, random numbers, and statistical functions; Pandas for Series (like time series) and DataFrames (like spreadsheets); Regular Expressions for general pattern matching; matplotlib for graphics/visualization; and other data cleaning/wrangling techniques.

Hence, your Group Project code, in parallel with doing something that is at least potentially useful and valuable to some customer, should demonstrate a solid knowledge of most of the topics we will cover. This does not mean that you must use tuples, or that you must use floor division, or that you must use a binary data file. But it does mean that you will use many features of the Python tools that we will cover.

In the final slidedeck you should include more detail including:

* a clear description of what you are trying to do and what data sources you will be using
* details about each DATA SOURCE that you are using
* some screenshots, or images of a *representative sample* of data from each source.

These elements should also be included:

* The NUMBER of data sources you are using,
* The VARIETY of scraping/downloading/API technologies you are using to obtain data,
* The COMPLEXITY of the data you are using,
* The VARIETY OF KIND of data sources: don't just create an aggregated list of trucks available for purchase by downloading and combining data from several vehicle sales websites.

You can use the draft pitch deck as the foundation for this final presentation, or develop some other materials to summarize your work. Again, the deliverable is not so much “how fancy and glitzy is your presentation? But rather how strong is the content you are sharing.

The FINAL Pitch Deck however should be professional and polished. This FINAL Pitch Deck will be the basis for your Group Presentation, which will be delivered by you in week 7. The upload should be a PPT or a PDF file, in a zip archive.

The goal of the Group Project is to use THREE sources of data, with AT LEAST ONE obtained by web scraping, others perhaps by direct download, by using an API, or whatever makes the most sense. In class we will investigate Beautiful Soup, but you may also wish to investigate Selenium, Scrapy, or other web scrapers.

Hint: Commercial web sites often detect and try to prevent web scraping. It is often possible to work around this, but it can be a challenge. Government web sites, which are excellent sources of economic, financial, health, crime, and other kinds of data, usually do NOT try to prevent scraping.

You may NOT use data from a "data set aggregator" site like Kaggle, where the data sets have already been cleaned for you. One purpose of the project is to gain experience dealing with dirty data. But if the original source of data provides clean data, that's fine.

By the time of this deliverable’s due date, you will have learned about basic Python: int, float, str, bool, None; arithmetic operators; lists, loops and range(); equality, relational, and logical operators; if/elif/else decision; built-in data structures (list, tuple, set, dict, with named and symbolic operators); sequence slices; user and text file input/output; and Beautiful Soup for scraping HTML web sites.

But note that we will be learning more, including NumPy for N-dimensional arrays, random numbers, and statistical functions; Pandas for Series (like time series) and DataFrames (like spreadsheets); Regular Expressions for general pattern matching; matplotlib for graphics/visualization; and other data cleaning/wrangling techniques.

Hence, your Group Project code, in parallel with doing something that is at least potentially useful and valuable to some customer, should demonstrate a solid knowledge of most of the topics we will cover. This does not mean that you must use tuples, or that you must use floor division, or that you must use a binary data file. But it does mean that you will use many features of the Python tools that we will cover.

## Project Final – Video Presentation 30 points

Create a short video (10 minutes or less) demonstrating your project being run. If certain things take a long time to run (like web scraping), show the beginning and ending of that in the video, and clip out the whirling dots/thumb twiddling/waiting-waiting-waiting part from your video. We should be able to post videos directly on Canvas – I will make a resource available either as an assignment, or maybe a Discussion so we can post and share all videos directly there. Make sure you record your video at high enough volume and use the proper equipment like a microphone as necessary to be sure the content can be heard clearly. AVOID BACKGROUND MUSIC or keep the volume very low so the **voiceover is clear and easy to understand**. The video should NOT contain information about how to INSTALL your project: that should be in the "README" file. (Likewise, the "README" file must contain written instructions about how to RUN your project: it MUST NOT just say, "watch the video."). You should describe the goal of the project, how it was constructed in a very basic way and demonstrate using the tool(s).

*Each person in the Project Group must give an approximately equal part of the presentation* (within a minute or so – it will not be timed, but there should not be presentations where one person talks for 7 minutes, and the other three talk for 1 minute each). Under the time constraint, you do not need to spend a lot of time introducing yourselves but you can say your name when explaining your part of the project. Keep the presentation focused on "here is how this will help our customers / solve a problem or pain point" level Provide an overview demo of your application but you do not need dive deeply into details of the code.

The FOCUS of your live presentation should be to get users excited about using your app. You need to show enough detail about what your application does to support this goal (like example output), but not so much detail (like source code details) that the presentation becomes boring.

## Project Final – Source Code 50 points

This deliverable requires you to submit all your source code to Canvas. We must be able to download your project code from Canvas, install it, and run it smoothly and without any modifications to a base Python environment (except for installing any required packages). Adding a few packages is OK, but the final project should probably not require a large number of different Python tools, add-ins, etc.

You must write exactly ONE “README” file that provides instructions on how to install and run your project if there are necessary modification, including installing any additional Python modules or packages; setting environment variables *(please avoid if possible)*; obtaining an API key; or the like. (If your project requires packages/modules, be sure to explain clearly what needs to be done MANUALLY to install those: ***DO NOT auto-install anything within your project***: it is very bad practice to do this and will result in a loss of points. Also, please include the names and AndrewIDs of all Group members in the “README” file. Provide screen shots to look at if that would be helpful. DO NOT write a “README” file that refers us to other “README” files elsewhere. To repeat: ONE “README” file only.

You may also consider adding a “requirements.txt” file that lists all the necessary additions to run the final project. We will talk about this in class and information about “requirements” files will be posted on Canvas.

Additional instructions:

1. Be sure that the name of your folder holding the project clearly identifies your team, and you also put your team name in the README and any other materials associated with the project. Even in the Python scripts and modules, be sure to add some comments somewhere near the top with the Team name for clarity.
2. There must be ONE main program file (**group\_4\_nifty\_project.py** or whatever) that we can load into IDLE, Spyder or VSCode and run to get your project to work. This does NOT mean that you need to write all your source code in a single code file: if the main program file requires other modules/program files in order to work, then it can **import** those modules, etc., to make everything work.
3. Be sure to use NO hard-coded file or directory pathnames. Put everything in a SINGLE directory, so that everything is in one location and the code works properly without branching into various other levels of the host computer.
4. If web scraping takes a long time (perhaps so long that it might time out on a slower web connection), make it possible to use a previously scraped copy of some data, rather than forcing the user to wait for a long and perhaps unsuccessful download. In your application, give the user the option to download fresh data, with a warning about how long that might take, OR to use previously downloaded data.
5. You should have at least SOME comments in each code file, including the name of the file, the AndrewIDs of the group members, the team name and a description of the purpose of the code in the file, which other module(s) import it, and which other module(s) it imports. If you have functions or classes or algorithms that are not obvious, add a short comment at the top describing what is happening.
6. Do not install (and expect us to install) software that duplicates Python language or Python module functionality, even if it would be more convenient and/or more efficient and/or cooler. For example, do not install a database system: use Python lists, sets, dicts, NumPy ndarrays, and/or Pandas DataFrames and/or Series. This is a Python course, after all! We want to see how much Python you have learned, not how well you know other software like SQL or JavaScript or whatever.

Points review

|  |  |  |
| --- | --- | --- |
| Deliverable | Timing | Points |
| Project Draft | 3rd week | 10 |
| First pitch deck | 6th week | 10 |
| Video and demonstration | Final class date | 30 |
| Final full codebase | Last day of classes | 50 |
| TOTAL | - | 100 |

# User Interfaces – A Trivial Example

Your Group Project program does not need to have a fancy user interface. In Canvas, under **Files | Group Project Files**, download the Python application file **trivial\_project.py**. Open and execute this application in IDLE, Spyder, or some other Python development environment.

This is a trivial menu-oriented program that:

1. Displays a data table of temperatures in Pittsburgh
2. Displays a plot of high and low temperatures
3. Displays simple statistics about the temperatures

Of course, your Group Project program needs to do much more than this trivial program does, but it does not need to have a much fancier user interface than this.

If you decide to put extra effort into a nice GUI for your program (which is also likely to improve your grade), check out **tkinter** at <https://docs.python.org/3/library/tkinter.html>. You may also consider packages like [Streamlit](https://streamlit.io/), [Dash](https://dash.plotly.com/), [Flask](https://flask.palletsprojects.com/en/3.0.x/), [Voila](https://voila.readthedocs.io/en/stable/using.html), etc. These are strictly not necessary, but if someone in the team happens to have some experience and can pull off bundling all the code necessary to output your project to an interactive webpage, that is fine. These routes have a high level of risk / reward because the application must run perfectly. If it fails to run with minimal adjustments to the Python environment, then the project would fail for not being fit for use. So use extreme caution if choosing to go this way.