**Database and Analytics Programming Project:**

Project Journal drafted by Polina Prinii – x21137757

Fri, Oct 29th:

Team meeting held with a duration of 30 minutes to pitch ideas for project topic. Proposed to cover Covid-19 on a country basis with the aim of analysing impact/spread/vaccination rate and so on.

Team agreed on the idea and countries were assigned to each.

Discussed project objectives such as database and programming language requirements. As well as kicked around the idea of potentially placing our combined data analysis on a dashboard hosted on a webservice, dependant on time available to complete project.

Thu, Nov 4th:

Research conducted around possible dataset to use for project. Duration of about roughly 2 hours.

Identified an appropriate dataset by Our World Data found here: <https://github.com/owid/covid-19-data>

Fri, Nov 5th:

Team meeting held. 30 minutes duration

Discussed the following:

* Journals and the format to submit in.
* Questions around which we required additional clarification from the lecturer.
* Understanding the question our dataset is looking to answer.
* Potential use of AWS
* The challenges behind working locally with our VM's, specifically towards the end of the project where we are looking to connect all 4 DB's for comparative analysis.

Fri, Nov 12th:

Team meeting held. 30 minute duration

Reviewed feedback from lecturer. Conducted a pivot from original theme for the project as clarification around the nature of datasets was received. Understanding is that each dataset must be unique therefore, the use of one dataset split on country basis was a no go.

Alun suggested to stick on the topic of Covid-19. Requested I cover the area around vaccinations.

Reviewed deadlines via Gantt Chart and set a soft deadline of the 5th of Dec for the Code part of the project.

Mon, Nov 15th:

Git repository created and invites extended to the whole team. This took about 30 minutes in total to set up.

Tue, Nov 16th:

Held a Quick Git Demo to get the team familiarised with GitHub. Covered areas such as creating folders and files, committing and pulling to Git through Desktop and the importance of the ReadMe file. Call lasted about 30 minutes.

Thu, Nov 18th:

Refined my section of the Git repository.

Deleted any test files and or folders. Created a README file which outlines the project objectives along with the aim of analysis for the chosen dataset.

Created a Word file to compile the Journal and updated the journal. All changes are reflected in the History section on Git Desktop or commits on git web.

Overall activity took about 2 hours.

Fri, Nov 19th:

Commenced project by setting up appropriate environments for the project via PyCharm the chosen IDE to write code and Oracle VM VirtualBox Manager.

Challenge encountered around the setup of the Python main file which is to be responsible for all project code.

As the chosen IDE is PyCharm, the project creation in PyCharm required for the environment to be specified. For PyCharm to connect to my Virtual Machine it is first necessary to have Python installed.

Having identified the correct steps to for installation Python 3.9 was installed on the VM.

Next challenge presented around installing pip package on the virtual machine as it is a necessary pre-requisite to connecting PyCharm to VM.

Work commenced @ 5:40pm and concluded @ 7:30pm. Work resumed @ 9:15pm.

Continued to attempt connection to VM from PyCharm upon which stumbled across my next challenge. Initially the overall understanding around env setup and connection to the VM was incorrect.

It was understood that both the latest version of Python and pip was to be installed on the VM which would be the environment (to be specified in PyCharm) in order to connect to the IDE located locally on the machine. This was incorrect.

The correct understanding or in other words process for env setup for the main.py file (to hold all python code) was to create the correct environment in PyCharm (in my case “virtualenv” was selected) which was created in the git repository stored locally, ultimately setting up the interpreter. From there within the IDE the main.py file was created. The main.py file will connect with MongoDB and PostgreSQL which are hosted on the VM.

Work concluded @ 10:50pm.

Sat, Nov 20th:

Work commenced at 2:20pm.

Today I am working with the main.py file. The aim is to ingest the files which are stored on the Git repository managed by Our World in Data.

For learning purposes, I have chosen to work with .csv files in order to convert it to JSON and then using the JSON files import the datasets into MongoDB.

Additionally, for contingency I have cloned the git repository from which my data is derived in the case the repository is deleted.

Issue encountered around editing the main.py fail, kept receiving the following error: “This file does not belong to the project”, additionally kept getting an error when trying to import pandas package.

Believe it might be an issue with initial set-up with the interpret and the project env.

Decided to uninstall PyCharm and start from scratch.

Identified the root cause was due to the project environment being setup in a folder which was not empty. Solution was to create a “Main Project” folder which would hold the main project drafted within PyCharm.

PyCharm works with empty directories aka folders when creating projects however, since I created my Journal and a README file in my project folder before setting up the PyCharm env, the project creation was experiencing issues.

Project space in PyCharm created and successfully synced with Git.

Work to be continued later, concluded at 4pm.

Sat, Nov 20th:

Work commenced at 2:40pm.

Continuing work which was initially intended for yesterday. Though to a few issues with PyCharm it was postponed. Today’s aim is to ingest the chosen datasets through Python. I’ll be using the pandas package to read the csv files directly from GitHub.

Found out the hard way that PyCharm does not automatically install the extensive packages available to Python like Jupyter Notebook.

Additionally, found that PyCharm for some odd reason limits the print output of a data-frame to two columns and 10 rows. To get around this I must use pd options around display to show more than 10 rows and 2 columns at a time. Having applied the pd options to expand the display of columns and rows, it would expand the view of the columns and not the rows. Unsure if there is some issue around the rows but the pd option won’t apply to rows.

For this reason I will just perform a number of print functions to validate if the data came through okay, such as print columns values, to show columns names, a count function to see how many rows and so on.

Included exception handling to cover the following cases:

* No Internet
* Any other possible errors.

Additionally included an if statement allowing the user to pull the data file from local machine in the case the code is being ran outside an internet connection.

Lastly included the additional two dataset around manufacturer of vaccines and age groups as well as cleaned up the code a little.

Tomorrow to commence converting to JSON. Work concluded @ 9:45pm.

Tue, Nov 23rd:

Work commenced at 7:40pm.

Today I’m working on taking the declared csv files using Pandas as converting them to a JSON object (file) which will be stored in my Git repo. I’ll be doing this convert in a form of a function additionally, I’ll be including a clause which will be replacing the JSON files on each run of the code as we are working with Git repository at the source dataset and said datasets undergo occasional updates which I want to reflect in the project.

Successfully manged to write a function which conducts the necessary csv to json convert and stores the .json files in a separate folder in Git. Additionally, manage to make the JSON file pretty from the beginning by using the indent clause in the .write() function of pandas.

Next, I move to the clause around the files to be replaced each time the code is ran. This not only helps in testing purposes but also allows me to work with the most up-to-date data in the datasets as I’ve specified the pd.read\_csv function to call form an URL (git repo).

Successfully implemented the replace functionality using the OS package in Python.

Next, I’ll be moving on to importing the three datasets into MongoDB for transforming the data.