**Assignment Cover Sheet**

To be completed **electronically** by the student and submitted with each piece of work. Please upload this completed cover sheet via Turnitin

**Assignment Title:** Code Lab 2 (Data-Driven Application)

**Tutor:** Arshiya Subhani

**Student Name: Adhenz Miranda**

**Student Number: 521536**

A black and white image of a letter s

Description automatically generated**Date of Submission: 01/15/2023**

Signed: Date: 01/11/2023

Your data driven application must be accompanied by a Development Document of 1000 words. This development document should consist of the following elements:

* *Abstract:* A short description of your application including which API you have selected. You should also provide a link to your Github repository, Google drive link to the executable folder, evidence of design and evidence of testing.
* *Project Plan:* Project plan that breaks the project down into key milestones (e.g planning, design, development & testing) and tasks. This plan should allocate the estimated time to complete each of the listed tasks.
* *Evidence of design:* This may include but is not limited to: specification list of your program requirements, Flowchart, Pseudo-Code, Wireframes, UML Data Structure Diagram. A minimum expectation for this section is the inclusion of a wireframe for the design of the GUI.
* *Technical Description & Walkthrough:* In this section you should provide a link to a video which includes a walkthrough of your program running as well as a technical breakdown of your code. This technical breakdown should explain how the key features of your program have been implemented via code. The video technical description & walkthrough contributes to the overall word count. The anticipated length of the video for walkthrough is 3-5 minutes.The anticipated length of the video for technical description video is max 8 minutes.
* *Testing:* Test plan and evidence of testing (e.g. screenshots and/or videos). The minimum expectation for this section is a testing table complete with test cases covering the core functionality of your app.
* *Critical Reflection:* An open and detailed evaluation of your application that notes what is compelling about the work, what could be improved, and what you need to learn to make these improvements.

*Appendix:* A copy of your code should be included in an appendix at the end of your documentation. To provide this simply copy and paste the code from each of your project files (do not provide the code in screenshots).

**Abstract**

This is a Python Tkinter application that’s based on the OpenTrivia Database API that is themed based around the Technology and Science subjects. There are 5 questions for each subject and 3 levels of difficulties built within the application, it can be replayed repeatedly until the player feels satisfied.

**Project Plan**

The plan for this project application is to make a trivia-based app based around technology and science, based off the OpenTrivia Database API and try make it with my own interpretation. My goal for this Application so that it is interactive for the user, the user can be at the least be educated, and the user should be capable of navigating properly through the app itself. I estimate in at least around 5-6 days this app should be completed and be functional by then, along with the writing of the development document.

The design of this app is going to be simple, and I will try make the design in my Figma file similar to the Tkinter, but I fear as I develop the app, it may not be the exact same thing or envision the same, but it is one of the plans, if not, I will try improvise the code and design so it is also easy for the user to navigate and interact within the app. The design should take at least no more than a day or 1-2 hours.

Developing this app is a challenge, as I need to debug and understand the basic functionality of how my app should be, and see which classes and variables I can apply to the app, As such also how I am going to deal with the APIs within the app and how I am going to deal with the contents of it, and looking at the API itself, it will take a while to develop the implementation, along with how I am going to structure the code and the main functionality of each class, I will create the app in one Python file, and it will have multiple classes made for each frame, and from what I can understand the OpenTrivia API is mostly just dictionaries with key-value pairs, different with each link, and it doesn’t require an API key for it to be initialize. I estimate the development to most likely around 4-5 days of development.

Testing the app itself will be a way for m to debug the app itself, see if there are any errors that can be debugged or abused in my app, and if there might be any visual overflows and as such, this should also take up the time alongside the development and estimated time of 4-5 days.

**Evidence of Design**

|  |
| --- |
| **Hi-fi Figma Prototype** |
|  |
| **Flowchart** |
|  |

**Technical Description & Walkthrough**

**Testing**

**Critical Reflection**