Gisma University of Applied Sciences

Assessment Submission Form

Student Number (If this is group work, please include the student numbers of all group participants)	GH1024695 Mahdieh Rajabi
Assessment Title	Build a GUI
Module Code	M604R
Module Title	Advance programming
Module Tutor	William Morrison
Date Submitted	11.01.2024

Declaration of Authorship

I declare that all material in this assessment is my own work except where there is clear acknowledgement and appropriate reference to the work of others.

I fully understand that the unacknowledged inclusion of another person's writings or ideas or works in this work may be considered plagiarism and that, should a formal investigation process confirms the allegation, I would be subject to the penalties associated with plagiarism, as per GISMA Business School, University of Applied Sciences' regulations for academic misconduct.

Signed……Mahdieh Rajabi…….. Date ……11.01.2024……

Link: https://github.com/Mahdiehrajabi/Advance-Programming2

Table of contents

Introduction

As a novice software engineer, I was asked to create a graphical user interface using a necessary dataset. The term GUI (graphical user interface) describes a paradigm that enables people to interact with computers and software programs through buttons, menus, logos, and icons.

I discovered a dataset about society and services related to the busiest airports, per the assessment brief. I published it to the GitHub repository after downloading it from the Kaggle website.

I started by installing a few necessary programs, such as tkinter and pandas.

Using the PyCharm terminal, type pip install pandas.

The same step applies to Tkinter as well.

I then used a function to develop the GUI that I intended to build.

```
#import libraries
import sys
import tkinter as tk
import csv
import matplotlib.pyplot as plt
import requests
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
from matplotlib.figure import Figure

```

I used 'import' to load various libraries or packages at the start of the code.

As an illustration:

Bring in os

It is possible to interact with the operating system and perform various actions on files, directories, environment variables, and other things using the OS module.

Bring in pandas as pd.

It's a well-known Python library. Pandas objects, classes, and functions were used throughout the code. This is a standard procedure to ensure that the code is both more legible and concise.

As tk, import tkinter.

Tkinter is the standard Python package used to create graphical user interfaces (GUIs). It is easier to refer to the library's elements in code when you import it as "tk," which eases the names that need to be written when using tkinter elements and widgets.

used the ‘tk’ in code to access and interact with the classes and functions of the tkinter library.

Import csv

imports a CSV module that is used to parse data structures that resemble tabular data, like Excel files. The module with the.csv extension provides many classes that can be used to read and write this file.

Import matplotlib

The most important part of this library is the Figure class, which displays the entire figure or the top-level container containing all of the rendered plotting elements, like labels and axes.

and "requests," a different package, was required in order to retrieve the data from the API. Application Programming Interface, or API, is a collection of protocols that enables interoperability between various programs. (They are essential)

Create GUI

```
# Create a window
root = tk.Tk()
root.title("Airport Analyzer")

data_selection_frame = tk.Frame(root)
data_selection_frame.pack(pady=10)

airport_loc_label = tk.Label(data_selection_frame, text="Select airport location:")
airport_loc_label.pack(side=tk.LEFT)

airport_loc_options = ["All", "Name", "TotalSeats", "Country Name"]
airport_loc_combobox = tk.ttk.Combobox(data_selection_frame, values=airport_loc_options)
airport_loc_combobox.pack(side=tk.LEFT)

# Button for visualization
generate_visualization_button = tk.Button(data_selection_frame, text="Generate Visualization", command=generate_visualization)
generate_visualization_button.pack()


visualization_frame = tk.Frame(root)
visualization_frame.pack(pady=20)

# Canvas for visualization
visualization_canvas = tk.Canvas(visualization_frame, width=500, height=400)
visualization_canvas.pack()

    visualization_canvas.delete('all')
    visualization_canvas.create_image(200, 150, image=visualization)

def airport_data():

    data = pd.read_csv('airport_volume_airport_locations.csv')
    return data

def create_visualization(data):

    pie_chart = plt.pie(data['locations'].value_counts())
    plt.title('Airport location')
    plt.show()

root.mainloop()
```

And then fetch data from our dataset:

```
91
92      def fetch(self, dataset):
93
94
95          api_url = f"https://www.kaggle.com/datasets/zvr842/all-global-airports"
96          response = requests.get(api_url)
97
98
99
100     def plot_data(self, data):
101
102
103         x_values = list(range(len(data)))
104         y_values = [item['value'] for item in data]
105
106         figure, ax = plt.subplots()
107         ax.plot(x, y, label="Values")
108         ax.set_xlabel("X")
109         ax.set_ylabel("Y")
110         ax.legend()
111
112     def clear_data(self):
113         self.airport_data = None
114         self.airport_list_widget.clear()
115         self.details_table.setRowCount(0)
116         self.ax.clear()
117         self.canvas.draw()
118         self.update_status_bar('Data cleared successfully.')
119
120
121         canvas = FigureCanvasTkAgg(fig, master=self.root)
122         canvas_widget = canvas.get_tk_widget()
123         canvas_widget.grid(row=2, column=0, columnspan=2, pady=10)
124
```

Data can be centrally stored on a server or database. To bring information from these centralized sources into the local application environment, it needs to be fetched. Before the data is viewed or evaluated, preprocessing may be required. This could include data transformation and cleaning. Gathering data should come first before carrying out any necessary preparatory steps.

The next step involves data visualization, which makes use of matplotlib to produce resizable visualizations of the dataset (such graphs and line charts).

References:

Codemy.com. (2019, January 10). *Create graphical user interfaces with Python and TKINTEr* [Video]. YouTube. https://www.youtube.com/watch?v=yQSEXcf6s2I

Developedbyed. (2023, January 13). *Build a modern Python GUI Project | Step by step tutorial* [Video]. YouTube. https://www.youtube.com/watch?v=NI9LXzo0UY0

Python Simplified. (2022, June 15). *Create GUI App with Tkinter and SQLite - Step by Step Python Tutorial for Beginners* [Video]. YouTube. https://www.youtube.com/watch?v=5qOnzF7RsNA