**Pod 2: Afra Noman, Kailey Bergeron**, **Anh Nguyen**

# Project 1: Tech Workshop Development – Documentation

**Introduction:** Graphical User Interface (GUI) is a type of user interface that allows users to interact with software applications through graphical elements such as buttons, text fields, menus, and icons. GUIs make applications more user-friendly by enabling intuitive interactions, typically using mouse clicks or touchscreen gestures, rather than text-based commands.

**To create a GUI, you will need the following:**

1. Programming language
2. Terminal
3. IDE (Integrated Development Environment)
4. GUI Library & Design Tools

**1. Programming Language**

Foundation for writing your code and creating the logic behind the GUI.

Python Installation: download the latest version from the official Python website: <https://www.python.org/downloads/>

**2. Terminal**

Used to run commands, install packages, and execute Python scripts.

Terminals: Command Prompt (Windows) or Terminal (Mac/Linux): execute the command: python GUI.py (GUI.py is the Python script).

\*These are usually pre-installed on your system.

**3. IDE (Integrated Development Environment)**

Provided a user-friendly interface for writing, testing and debugging code. For GUI development in Python, one of the most popular and versatile IDEs is Visual Studio Code (VS Code)

Visual Studio Code Installation: download on this page <https://code.visualstudio.com/>

(Refer to pod 4 - IDE documentation for more information)

After that, install VS Code and Python Extensions.

**4. GUI Library & Design Tools**

*GUI Libraries* contains the tools and widgets needed to build the graphical interface.

* Tkinter is the built-in, standard GUI library for Python, included with most Python installations. It’s great for simple, lightweight applications.
* Install PyQt using the command: pip install pyqt5
  + Offers a wide range of widgets and tools for creating complex GUIs.

*Design Tools* help visually design the layout of GUI, making it easier to create and adjust the interface.

* Tkinter includes a basic design tool called tkinter.ttk which allows for some visual design capabilities. PySimpleGUI wraps around Tkinter to provide a more straightforward way to create GUIs.
  + Install PySimpleGUI using the command: pip install pysimplegui

Tkinter Installation: Tkinter usually comes pre-installed with Python, but if it’s not available, install it using the following command: pip install tk

**Basic Structure of a Tkinter GUI:**

* Import Tkinter: Import the Tkinter library.
* Create the Main Window: The root window is the main application window.
* Add Widgets: Widgets like buttons, labels, and entry fields are added to allow user interaction. Some common widgets: Button(), Entry(), Text(), Label(), Frame(), etc.
* Main Loop: The mainloop() method listens for events (button clicks, key presses) and keeps the application running.

**GitHub links to supplemental code that will help further your understanding of GUIs and how to build them with Tkinter:**

* Supplemental
  + CalculatorTk.py
  + WeatherTk.py
  + MenuBarTk.py
  + AllTkApps.py
  + MCQuizTk.py
  + MCQuizTkData.json

**Practical exercise to complete after the workshop:**

1. You will study, follow along, and recreate the MCQuizTk application.
2. After ensuring that the original works properly, change the MCQuizTkData file to questions and answers about your own workshop.
3. Study the AllTkApps application.
4. You will modify your MCQuizTk code by making an established frame for the results to display, like how the AllTkApps application uses frames.
5. Make some stylistic changes
6. Your ***deliverable*** is your modified code, modified data file, and a screen recording of your quiz working.
7. Turn this in your pod’s submissions folder.