Here is a **fully organized, clean, copy-and-paste–ready Google Doc** draft you can paste directly into Google Docs.
It includes:

✅ What the Teensy code does
✅ What the GUI does
✅ Exactly what should happen during operation
✅ Installation steps for Python in Visual Studio Code
✅ Instructions for running both sides
✅ Troubleshooting

**Project 2 — Teensy 4.1 + IMU + GPS + Python GUI

User Guide & Setup Instructions**

---

1. Overview

This project uses a **Teensy 4.1 microcontroller** to read sensor data from:

- **LSM6DSOX IMU** (Acceleration + Gyroscope via I²C)
- **LIS3MDL Magnetometer** (Magnetic field via I²C)
- **STEMMA Mini GPS** (Latitude, Longitude, Altitude, Satellites via UART)

The Teensy sends all sensor values through the USB serial port to a custom **Python GUI**, built using **Tkinter**, which displays the data in real-time.

This document explains:

- How the system works
- What the GUI does

- How to install everything required
- How to run both the Teensy code and the GUI
- What the user should see at each step
- Troubleshooting

---

2. Hardware Components

The final project uses:

- **Teensy 4.1 microcontroller**
- **LSM6DSOX accelerometer/gyroscope**
- **LIS3MDL magnetometer**
- **STEMMA Mini GPS module**
- **HC-05 Bluetooth module** (tested separately)
- **LiPo battery + PowerBoost 1000C**
- **Built-in Teensy microSD slot**
- **USB cable (for power + serial communication)**

Connections:

- **IMU & Magnetometer** → **I²C** (SDA pin 18, SCL pin 19)
- **GPS Module** → **UART1** (RX1 pin 0, TX1 pin 1)
- **USB** → **PC** (for serial data and GUI connection)

---

3. What the Teensy Program Does

The Teensy code continuously:

1. Reads IMU data

- Angular Velocity X, Y, Z (rad/s)
- Acceleration X, Y, Z (m/s²)

2. Reads Magnetometer data

- Magnetic Field X, Y, Z (µT)

3. Reads GPS data

- Latitude (deg)
- Longitude (deg)
- Elevation (MSL & Ellipsoid)
- Number of locked satellites

4. Combines all data into a single CSV line

Example:

```
1234,0.12,0.03,9.80,0.001,0.40,0.005,22.5,18.6,-41.2,36.125000,-97.061000,262.0,7
```

5. Sends the data through USB Serial

→ This is what the Python GUI reads.

---

4. What the Python GUI Does

When the user runs the GUI:

✔️ Main Menu

- **Start Display**
- **Exit**

✔️ Start Display Screen

The GUI:

1. Automatically searches for the Teensy's serial port
2. If *no port is found*, it shows an error message and returns to the main menu
3. If successful, it displays live sensor data in **two sections**:
    - **GPS Section**
    - **IMU Section**
4. Fields update in **real time** (no scrolling):
    - **GPS:** Latitude, Longitude, Elevation (MSL), Elevation (Ellipsoid), Satellites
    - **IMU:** Angular Velocity X, Y, Z; Acceleration X, Y, Z; Magnetic Field X, Y, Z

✔️ End Display Button

Returns the user to the main menu.

✔️ Exit Button

Closes the application safely at any time.

---

5. Exactly What Should Happen (End-to-End)

Step 1 — Power + Upload

- You plug Teensy into PC

- You upload the final `.ino` from Arduino IDE
- Teensy starts running and streaming sensor data to USB

Step 2 — Test the Teensy Serial Output

- Open Arduino Serial Monitor
- Set **115200 baud**
- You should see constant sensor data every ~100ms

If the Teensy is working, continue.

Step 3 — Run the GUI

- Close Arduino Serial Monitor (important—only one program can own the port)
- Open VS Code Terminal

Run:

```
python localization_gui.py
```

-

Step 4 — Start Display

- GUI window opens
- Click **Start Display**
- The GUI automatically connects to the Teensy's USB port
- IMU + GPS numbers immediately begin updating
    - GPS may show zeros indoors
    - IMU should update instantly

Step 5 — End Display

- User can click **End Display** to return to main menu

Step 6 — Exit

- User can click **Exit** to close program safely

---

6. Installing Python and Requirements (If User Has Nothing Installed)

A. Install Python

1. Go to python.org
2. Download **Python 3.11 or 3.12**
3. Check the box: **“Add Python to PATH”**
4. Install

B. Install Visual Studio Code

1. Go to https://code.visualstudio.com
2. Install VS Code
3. Open VS Code
4. Go to **Extensions**
5. Install the extension named: **Python**

C. Required Python Packages

In the VS Code terminal run:

```
pip install pyserial
```

Tkinter usually comes built-in with Python.

D. Verify

Run this:

```
python --version
```

```
pip --version
```

If both return valid versions, Python is correctly installed.

---

7. How to Run the GUI (Step-By-Step)

1. Save the file

Save your Python GUI code as:

```
localization_gui.py
```

2. Open VS Code

- Open the folder containing your `.py` file

3. Open Terminal

```
View → Terminal
```

4. Run the program:

```
python localization_gui.py
```

5. Main Menu

Click **Start Display**

6. Live Data Should Appear

- IMU values changing as you move the board
- Magnetometer values changing
- GPS values populated if you have a fix

7. End Display or Exit

Buttons always return you safely.

---

8. Troubleshooting

1. GUI says “No USB device found”

- Close Arduino Serial Monitor
- Replug the Teensy
- Try a different USB cable

2. GPS only shows zeros

- GPS does NOT work inside most buildings
- Move near a window or outside
- Allow 30–120 seconds for first fix

3. GUI crashes at startup

Run:

```
pip install pyserial
```

4. Serial Monitor is blank

- Confirm baud is **115200**
- Ensure correct port is selected

---

9. Summary

This system ties together:

- **Teensy 4.1 hardware**
- **I²C IMU + Magnetometer**
- **UART GPS**
- **USB Serial streaming**
- **Python GUI visualizer**

The Teensy gathers all sensor data → sends it over USB → Python GUI displays it in real time.

When running correctly:

- Teensy continuously streams CSV sensor data
- GUI auto-detects Teensy
- GPS + IMU fields update instantly
- User can end or exit at any time