

**AUV**

[GUI]

Subject Goes Here

Name:Akshay Parseja

4/2/2020

**Table of Contents**

[Abstract 3](#_Toc15063064)

[Introduction 4](#_Toc15063065)

[Subpoint 1 4](#_Toc15063066)

[Subpoint 2 4](#_Toc15063067)

[Subpoint of Subpoint 2 4](#_Toc15063068)

[Heading 1 5](#_Toc15063069)

[Result 6](#_Toc15063070)

[Conclusion 7](#_Toc15063071)

[References and Links 8](#_Toc15063072)

# 

# **Abstract**

This repository is created by member of Trident labs and will guide you through our Graphical User interface(a.k.a GUI) desktop application which is being used by us to perform various tasks and also accomplish our testing successfully and eaily!

# 

# **Introduction**

We are using PyQt a python interface to create our responsive GUI.

PyQt is a GUI widgets toolkit. It is a Python interface for **Qt**, one of the most powerful, and popular cross-platform GUI library. PyQt was developed by RiverBank Computing Ltd. The latest version of PyQt can be downloaded from its official website − [riverbankcomputing.com](https://riverbankcomputing.com/software/pyqt/download)

PyQt API is a set of modules containing a large number of classes and functions. While **QtCore** module contains non-GUI functionality for working with file and directory etc., **QtGui** module contains all the graphical controls. In addition, there are modules for working with XML **(QtXml)**, SVG **(QtSvg)**, and SQL **(QtSql)**, etc.

# 

# **How did we create :**

* Firstly we created a list of our requirements which were needed to be performed using GUI ( THE NEEDFUL)
* Moving forward we made our front end because the easier it is accessible for our team to access the better our GUI
* After finishing with These we started with making our classes for each task and inserted our logic!

# **HOW TO USE :**

1. open server.py check port , note down the port
2. open Trident.py, check ip:
   * + - 1. if local host , ip='127.0.0.1' (no rpi, server file running on same pc as on Trident.py)
         2. if using Rpi, ip= raspberry pi ip dress

('192.168.2.10)(black sd card)

('192.168.2.150)(red sd card)

1. check port , port in Trident.py should be the same as of server.py

* if ip is right , port is checked :

click on path (G:\Trident labs\Trident Labs 19-20\gui\graphical\_user\_interface-Trident2.2.5\Trident.2.2.3\Trident.2.2.5) and write cmd instead of it

run command --> python server.py (to run server.py file)

open another cmd

next run command --> python Trident.py(to open gui)

press connect

1. ERRORS:

* if local bind issue "port in use"
  + - * 1. 1)open server.py , change port
        2. 2)open Trident.py , make sure to change port in Trident.py and keep it same as server.py
        3. 3) run server,run Trident, connect

1. Steps to follow for proper exiting from gui which donot affect bot functionalities:(in Bot)
   * 1. CLose gui using Esc
     2. CLose server file in rpi
     3. HOFF

# 

# **Result:**

1. Testing time reduced really well.
2. Error detection in communication performed well.
3. State of bot underwater can be interpreted at much better level which in turn helps the execution of further tasks.
4. Performing tuning of various kinds have become easier.

# 

# **Conclusion**

# **References and Links**

Ref 1

Ref 2

Ref 3