Activity #4 - Introduction to GUI Development using Pycharm

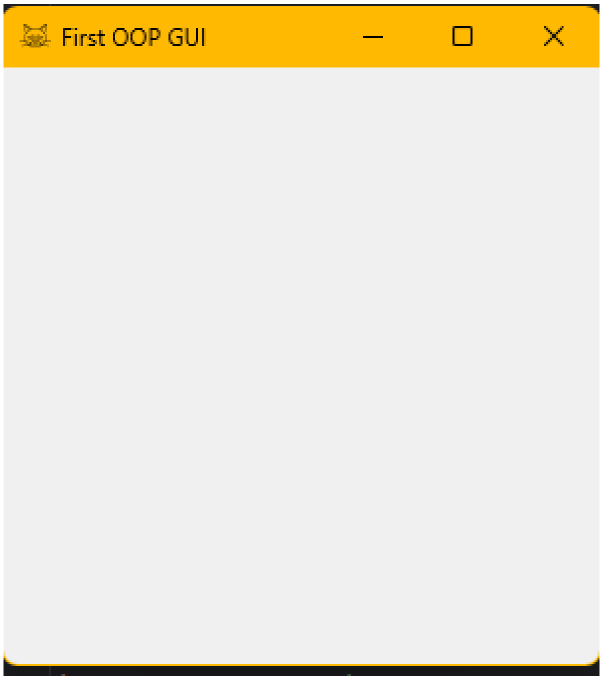
Valleser, Kenn Jie L..	14/10/2024
CPE009B/CPE21S4	Engr. Ma. Rizette Sayo

5. Procedure

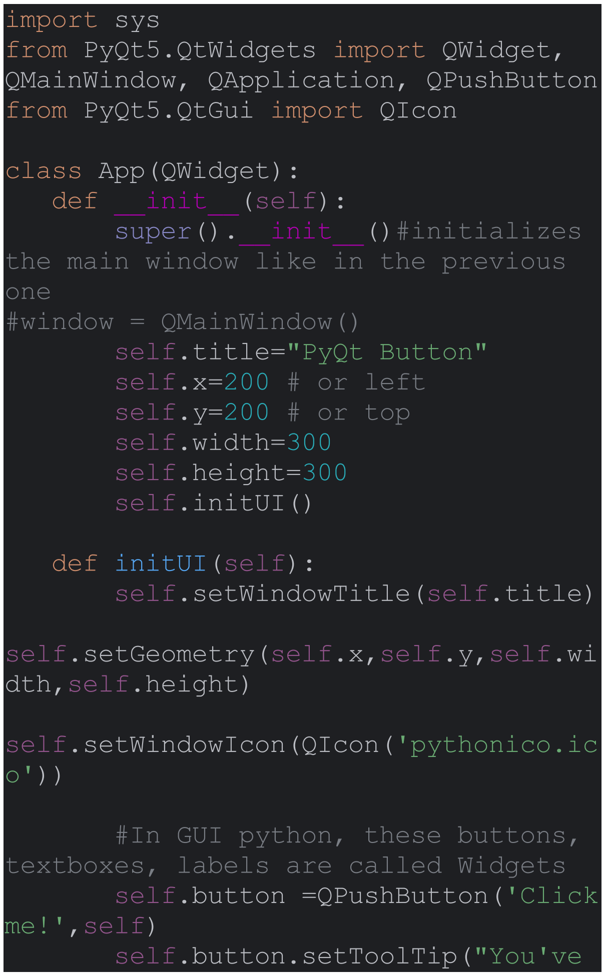
Source Code

```
import sys
from PyQt5.QtWidgets import
QMainWindow, QApplication
from PyQt5.QtGui import QIcon

class App(QMainWindow):
   def __init__(self):
       super().__init__()#initializes
the main window like in the previous
one
#window = QMainWindow()
       self.title="First OOP GUI"
       self.initUI()

   def initUI(self):
       self.setWindowTitle(self.title)

self.setGeometry(200,200,300,300)

self.setWindowIcon(QIcon('pythonico.ic
o'))#sets an icon
       self.show()

if __name__=='__main__':
   app=QApplication(sys.argv)
   Main=App()
   sys.exit(app.exec_())
```

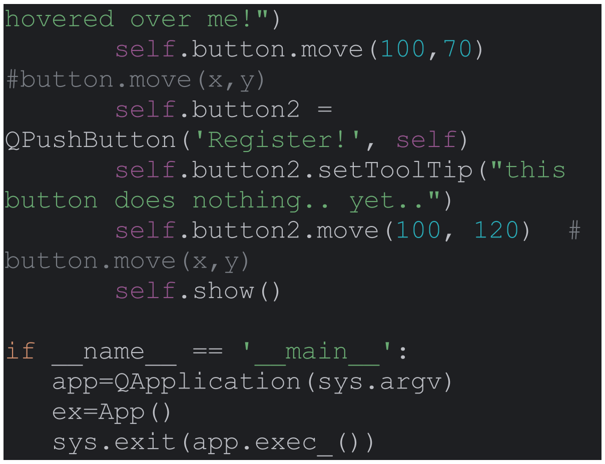
Output



Source Code

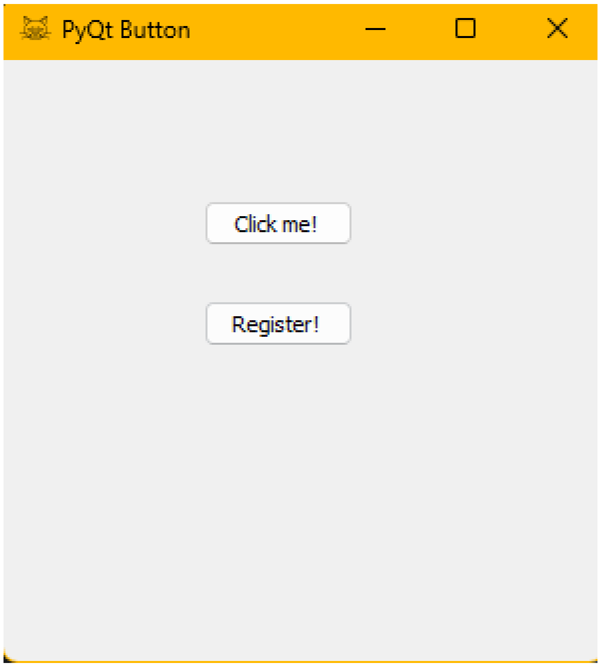


```
import sys
from PyQt5.QtWidgets import QWidget,
QMainWindow, QApplication, QPushButton
from PyQt5.QtGui import QIcon

class App(QWidget):
   def __init__(self):
       super().__init__()#initializes
the main window like in the previous
one
#window = QMainWindow()
       self.title="PyQt Button"
       self.x=200 # or left
       self.y=200 # or top
       self.width=300
       self.height=300
       self.initUI()

   def initUI(self):
       self.setWindowTitle(self.title)

self.setGeometry(self.x,self.y,self.wi
dth,self.height)

self.setWindowIcon(QIcon('pythonico.ic
o'))

       #In GUI python, these buttons,
textboxes, labels are called Widgets
       self.button =QPushButton('Click
me!',self)
       self.button.setToolTip("You've
```

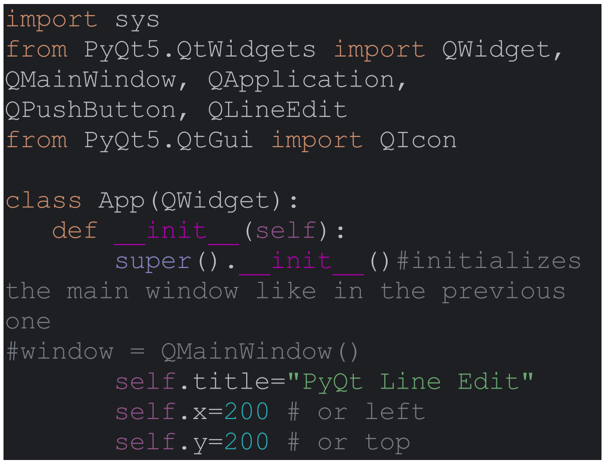


```
hovered over me!")
       self.button.move(100,70)
#button.move(x,y)
       self.button2 =
QPushButton('Register!', self)
       self.button2.setToolTip("this
button does nothing.. yet..")
       self.button2.move(100, 120)   #
button.move(x,y)
       self.show()

if __name__ == '__main__':
   app=QApplication(sys.argv)
   ex=App()
   sys.exit(app.exec_())
```

Output



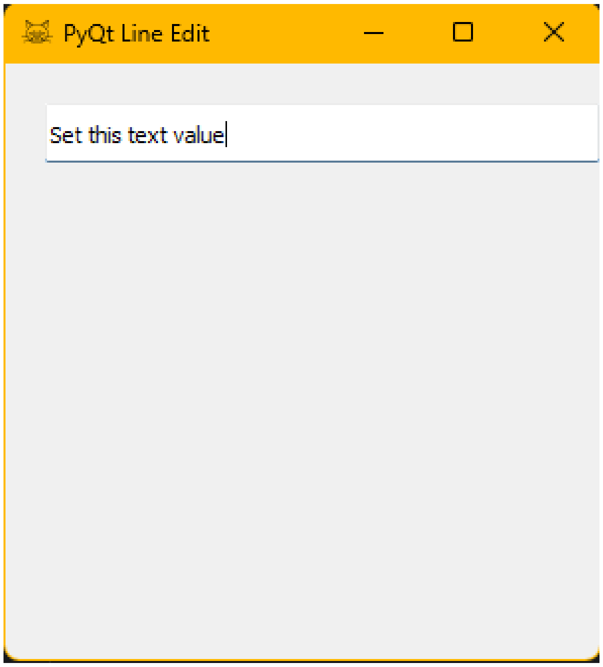
Source Code



```
import sys
from PyQt5.QtWidgets import QWidget,
QMainWindow, QApplication,
QPushButton, QLineEdit
from PyQt5.QtGui import QIcon

class App(QWidget):
   def __init__(self):
       super().__init__()#initializes
the main window like in the previous
one
#window = QMainWindow()
       self.title="PyQt Line Edit"
       self.x=200 # or left
       self.y=200 # or top
```

```
        self.width=300
        self.height=300
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x,
self.y, self.width, self.height)

self.setWindowIcon(QIcon('pythonico.ic
o'))

        #Create text
        self.textbox=QLineEdit(self)
        self.textbox.move(20,20)
        self.textbox.resize(280,30)
        self.textbox.setText("Set this
text value")
        self.show()



if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```

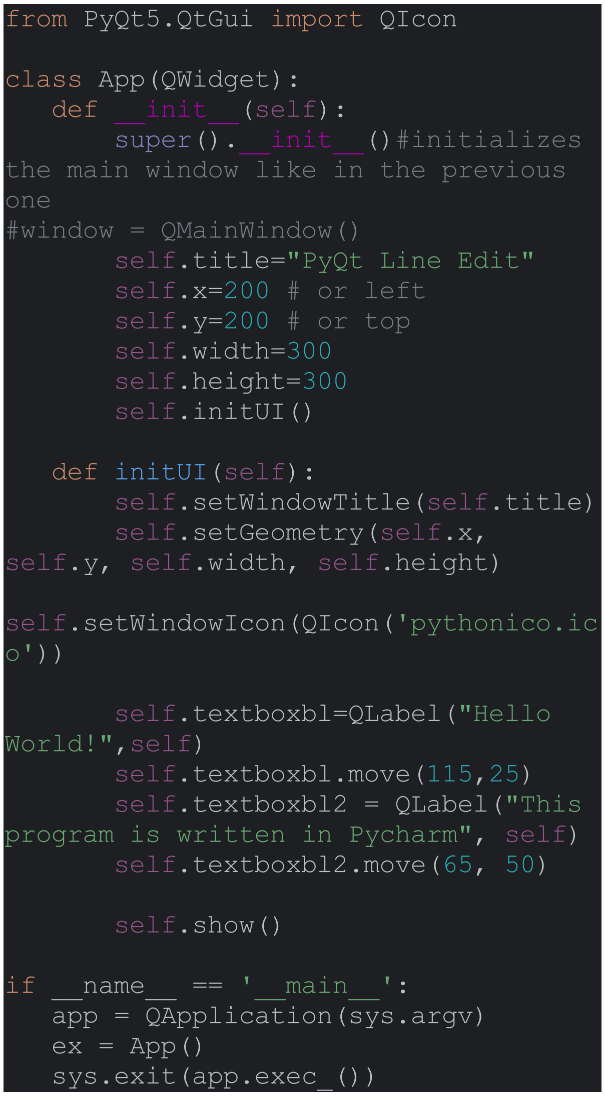
Output



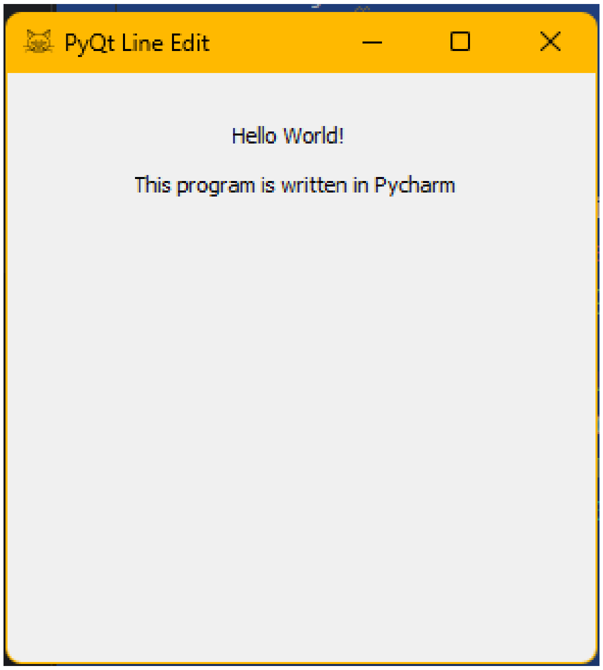
Set this text value

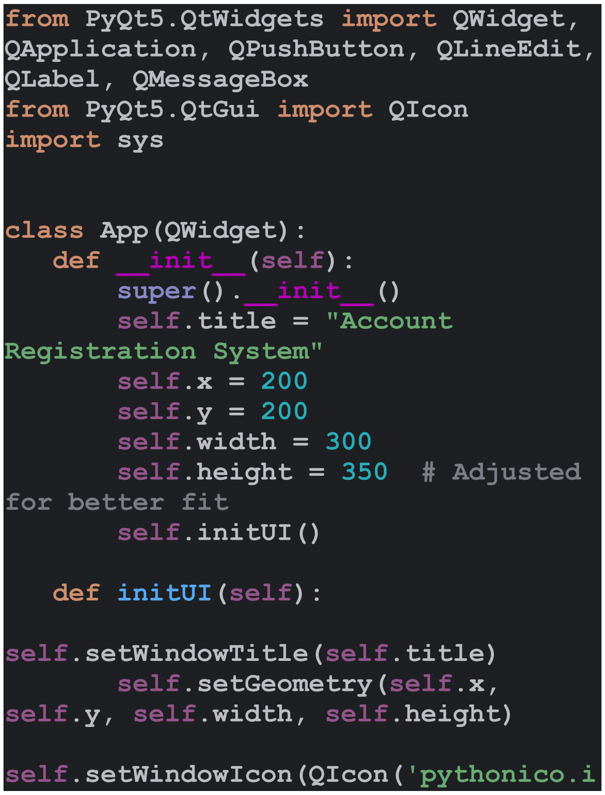
Source Code

```
import sys
from PyQt5.QtWidgets import QWidget,
QMainWindow, QApplication,
QPushButton, QLineEdit, QLabel
```

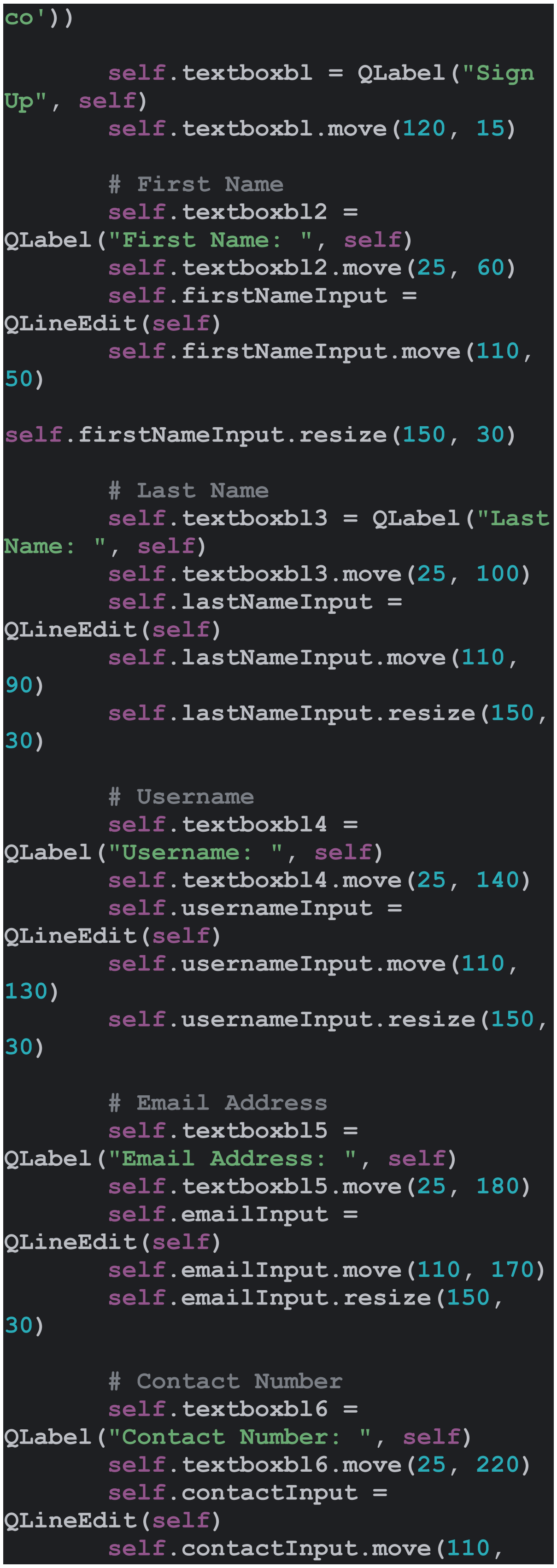


```
from PyQt5.QtGui import QIcon

class App(QWidget):
    def __init__(self):
        super().__init__()#initializes
the main window like in the previous
one
#window = QMainWindow()
        self.title="PyQt Line Edit"
        self.x=200 # or left
        self.y=200 # or top
        self.width=300
        self.height=300
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x,
self.y, self.width, self.height)

self.setWindowIcon(QIcon('pythonico.ic
o'))

        self.textboxbl=QLabel("Hello
World!",self)
        self.textboxbl.move(115,25)
        self.textboxbl2 = QLabel("This
program is written in Pycharm", self)
        self.textboxbl2.move(65, 50)

        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```

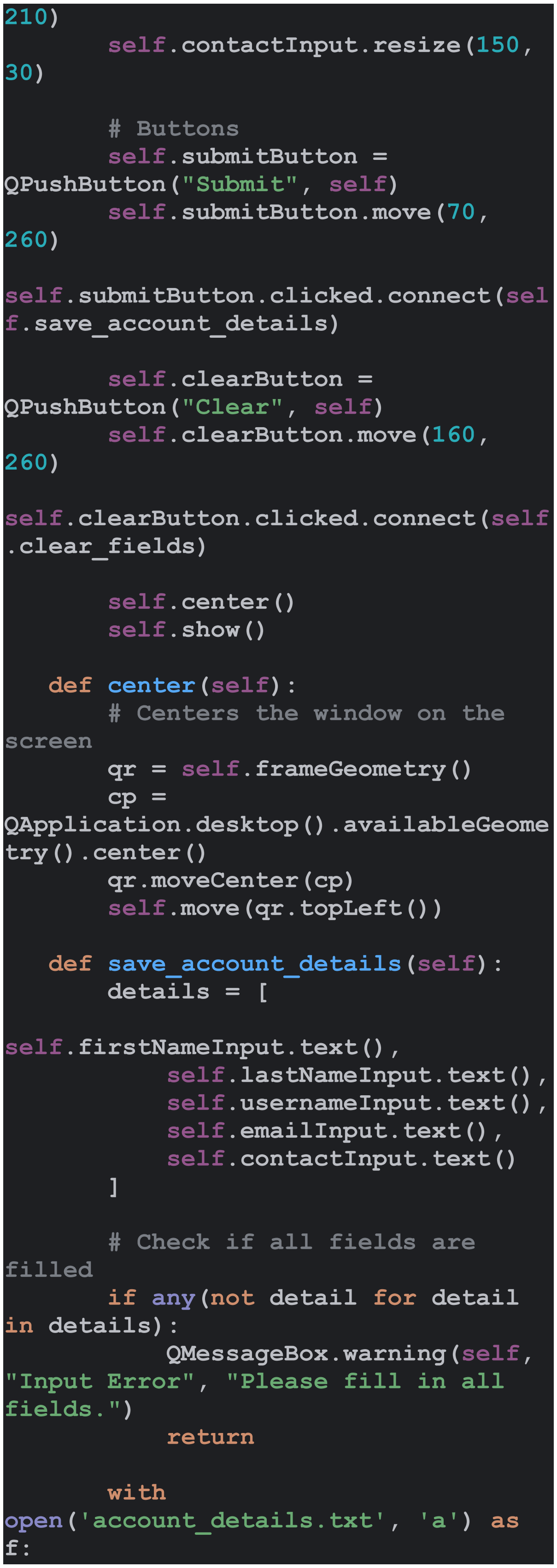
Output	 Hello World! This program is written in Pycharm

6. Supplementary Activity	
Source Code(Registration.py)	

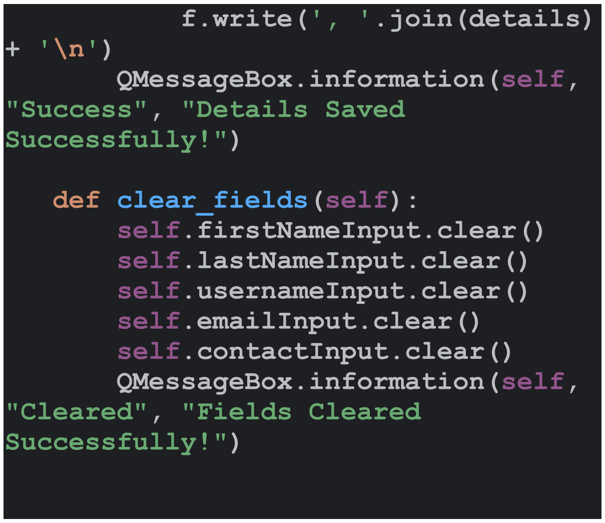
```
from PyQt5.QtWidgets import QWidget,
QApplication, QPushButton, QLineEdit,
QLabel, QMessageBox
from PyQt5.QtGui import QIcon
import sys


class App(QWidget):
   def __init__(self):
       super().__init__()
       self.title = "Account
Registration System"
       self.x = 200
       self.y = 200
       self.width = 300
       self.height = 350  # Adjusted
for better fit
       self.initUI()

   def initUI(self):

self.setWindowTitle(self.title)
       self.setGeometry(self.x,
self.y, self.width, self.height)

self.setWindowIcon(QIcon('pythonico.i
```



```
co'))

        self.textboxbl = QLabel("Sign
Up", self)
        self.textboxbl.move(120, 15)

        # First Name
        self.textboxbl2 =
QLabel("First Name: ", self)
        self.textboxbl2.move(25, 60)
        self.firstNameInput =
QLineEdit(self)
        self.firstNameInput.move(110,
50)

self.firstNameInput.resize(150, 30)

        # Last Name
        self.textboxbl3 = QLabel("Last
Name: ", self)
        self.textboxbl3.move(25, 100)
        self.lastNameInput =
QLineEdit(self)
        self.lastNameInput.move(110,
90)
        self.lastNameInput.resize(150,
30)

        # Username
        self.textboxbl4 =
QLabel("Username: ", self)
        self.textboxbl4.move(25, 140)
        self.usernameInput =
QLineEdit(self)
        self.usernameInput.move(110,
130)
        self.usernameInput.resize(150,
30)

        # Email Address
        self.textboxbl5 =
QLabel("Email Address: ", self)
        self.textboxbl5.move(25, 180)
        self.emailInput =
QLineEdit(self)
        self.emailInput.move(110, 170)
        self.emailInput.resize(150,
30)

        # Contact Number
        self.textboxbl6 =
QLabel("Contact Number: ", self)
        self.textboxbl6.move(25, 220)
        self.contactInput =
QLineEdit(self)
        self.contactInput.move(110,
```

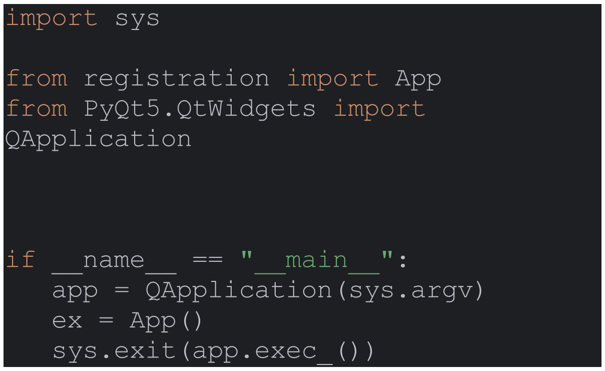


```
210)
       self.contactInput.resize(150,
30)

       # Buttons
       self.submitButton =
QPushButton("Submit", self)
       self.submitButton.move(70,
260)

self.submitButton.clicked.connect(sel
f.save_account_details)

       self.clearButton =
QPushButton("Clear", self)
       self.clearButton.move(160,
260)

self.clearButton.clicked.connect(self
.clear_fields)

       self.center()
       self.show()

   def center(self):
       # Centers the window on the
screen
       qr = self.frameGeometry()
       cp =
QApplication.desktop().availableGeome
try().center()
       qr.moveCenter(cp)
       self.move(qr.topLeft())

   def save_account_details(self):
       details = [

self.firstNameInput.text(),
           self.lastNameInput.text(),
           self.usernameInput.text(),
           self.emailInput.text(),
           self.contactInput.text()
       ]

       # Check if all fields are
filled
       if any(not detail for detail
in details):
           QMessageBox.warning(self,
"Input Error", "Please fill in all
fields.")
           return

       with
open('account_details.txt', 'a') as
f:
```



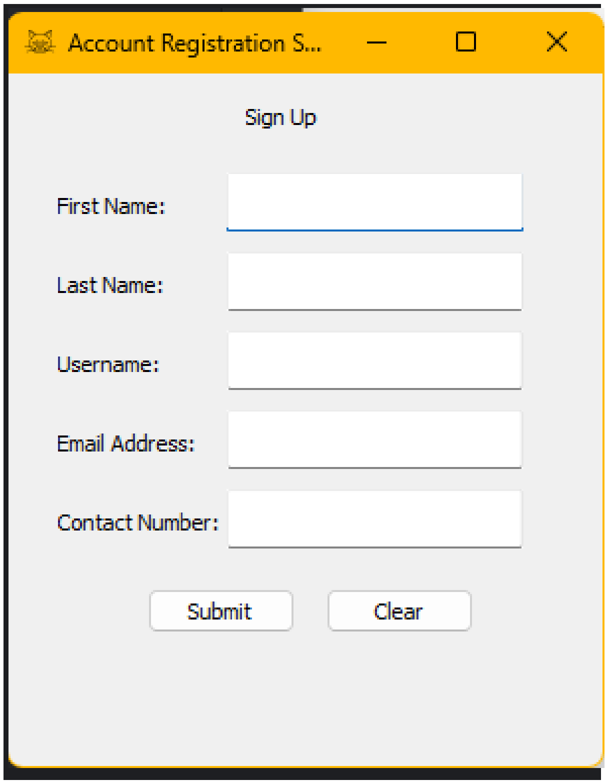
```
            f.write(', '.join(details)
+ '\n')
        QMessageBox.information(self,
"Success", "Details Saved
Successfully!")

    def clear_fields(self):
        self.firstNameInput.clear()
        self.lastNameInput.clear()
        self.usernameInput.clear()
        self.emailInput.clear()
        self.contactInput.clear()
        QMessageBox.information(self,
"Cleared", "Fields Cleared
Successfully!")
```

Source Code (Main.py)



```
import sys

from registration import App
from PyQt5.QtWidgets import
QApplication



if __name__ == "__main__":
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```

Output



Questions:

1. What are the common GUI Applications that general end-users such as home users, students, and office employees use? (give at least 3 and describe each)
Web Browsers (like Google Chrome). Web browsers let you go online and look at websites. They have buttons and menus that make it easy to search for information, bookmark favorite sites, and watch videos.
Office Suites (like Microsoft Office). Office suites include programs like Word for writing documents and Excel for spreadsheets. They help you create and edit text or numbers easily, with tools that make things like formatting and calculations straightforward.
Media Players (like VLC Media Player). Media players are used to play music and videos. They have simple controls like play, pause, and volume, so you can easily enjoy your favorite songs or movies.

2. Based from your answer in question 1, why do you think home users, students, and office employees use those GUI programs?
Easy to Use the graphical interface helps people navigate without needing to know complicated commands.
Programs like word processors make it easier to write essays or reports for school or work.
User-Friendly of Many of these apps have help features, so even beginners can figure them out.

3. How does Pycharm help developers in making GUI applications, what would be the difference if developers made GUI programs without GUI Frameworks such as Pycharm or Tkinter?

Smart Suggestions of the PyCharm can suggest code, which saves time and reduces mistakes.
Debugging of the PyCharm helps you find and fix errors in your code more easily.
Framework Support of the PyCharm works well with GUI frameworks (like Tkinter), making it easier to create windows and buttons without too much hassle.

4. What are the different platforms a GUI program may be created and deployed on? (Three is required then state why might a program be created on that specific platform)

- Windows

Many people use Windows, so developers often make applications for it. It's also compatible with a lot of different software and hardware.

- macOS

Developers create apps for macOS because it's popular among Apple users. The design is nice, and it works well with other Apple products.

- Linux

Linux is often used for programming and servers. It's free and customizable, which attracts developers who like to tinker with their software.

5. What is the purpose of app = QApplication(sys.argv), ex = App(), and sys.exit(app.exec_())?

app = QApplication(sys.argv): This starts the application and prepares it to run. It also lets you use command-line arguments if needed.

ex = App(): This creates your main window where all the GUI stuff happens, like buttons and text fields.

sys.exit(app.exec_()): This starts the program and keeps it running until you close it. When you close it, it makes sure everything stops nicely.

7. Conclusion

In conclusion, GUI applications like web browsers, office suites, and media players simplify tasks for home users, students, and office workers by providing user-friendly interfaces. Programming tools like PyCharm make it easier for developers to create these applications by offering helpful features and framework support. Different platforms such as Windows, macOS, and Linux allow developers to reach various audiences and utilize unique functionalities. Understanding the basics of GUI application structure is essential for anyone that are interested in building their own software.