COLLEGE OF ENGINEERING, TRIVANDRUM

THIRUVANANTHAPURAM – 695 016

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Project Report
on

SIMPLE ADDRESS BOOK

Using PyQt

Submitted by

ABHISHEK T	TVE17CS004
ABDUL KHADER P V	LTVE17CS063
HISHAM ALI P	LTVE17CS066

Under the guidance

of

Thania Kumar

CONTENTS

Chapter 1

INTRODUCTION

Digitizing your data storage lets you store large volumes of data in less space than paper would take up. Simple Address Book is exactly that, an easy-to-use address book for your computer. You can use it to store your contacts and their addresses with a few quick clicks by use of a GUI application which comes with a window and some buttons that makes it more user friendly to enter and retrieve the data.

Adding contacts is as easy as filling out a form with a new contact's name followed by address. Saved data can be retrieved and analyzed. Information are stored using name as the keyword so that we can access the data by names. We filled out the form, pressed the "Add" button and our new contact was added after pressing “Submit” button; simple as that. If you want to change or update any of the information for your contacts, you can do so by clicking "Edit" button after you “Find” the contact..

Chapter 2

Py-Qt

PyQt is a Python Binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in. PyQt is a free software developed by the British firm Riverbank Computing. PyQt implements around 440 classes and over 6000 functions and methods including a substantial set of GUI widgets,classes for accessing SQL databases, Qscintilla, a scintilla-based rich text editor widget, an XML parser, SVG support, etc

The applications of PyQt include:

1. Openshot, a video editing program
2. Anki, a spaced repetition flashcard program
3. Dropbox, a file hosting service
4. Eric Python IDE
5. Kodos, Python Regular Expression Debugger
6. Orange, a data mining and visualization framework
7. QGIS, a free software desktop Geographic Information Systems(GIS) application
8. Veusz, a scientific plotting application
9. Spyder, a lightweight Python IDE
10.Leo, an outliner and literate programming editor

And many More.....

Chapter 3

DEVELOPMENT OF THE APPLICATION

PyQT is a Python wrapper around the QT framework for creating graphical user interfaces, or GUIs. First of all we need to install python and PyQT on the system by typing following command

```
$ apt-cache search pyqt
```

After that we need to create an interface for the application. We'll need some basic imports. Here, we're importing QtGui, which deals with all things GUI with PyQT.

```
import sys
from PyQt4 import QtGui, QtCore
```

First, we tell Python to load PyQt via the import statement: from PyQt4. QtWidgets import QApplication, Qlabel

Next, we create a Qapplication with the command:
app = Qapplication([])

This is a requirement of Qt: Every GUI app must have exactly one instance of QApplication. Many parts of Qt don't work until we executed the above line. So we need it in virtually every (Py) Qt app you write.

Everything we see in a (Py)Qt app is a widget: Buttons, labels, windows, dialogs, progress bars etc. Like HTML elements, widgets are often nested. For example, a window can contain a button, which in turn contains a label.

As before, we instantiate a QApplication. Then, we create a window. We use the most basic type Qwidget for it because it merely acts as a container and we don't want it to have any special behavior. Next, we create the layout and add QPushButtons to it. Finally, we tell the window to use this layout (and thus its contents). As in our first application, we end with calls to .show() and app.exec_().

The widgets can be created by the following functions. The below code shows creation of buttons in Graphical user interface.

```
nameLabel = QtGui.Qlabel("Name:")
self.nameLine = QtGui.QLineEdit()
self.nameLine.setReadOnly(True)
addressLabel = QtGui.QLabel("Address:")
self.addressText = QtGui.QTextEdit()
self.addressText.setReadOnly(True)
self.addButton = QtGui.QPushButton("&Add")
self.addButton.show()
```

The above lines of code is creating 3 buttons and making buttons to show.

Finally, once we're content with the GUI that we have built, we invoke:

```
window.show();
```

.show() brings the window to the screen for the user. ".show()" is a QT method.

Next step is we need to complile and run out program code. As because we installed python3 in the system, we need to compile the application code to make it run. For that we are compiling applications.

```
python fiename.py
```

Now we have a UI and following many functions to proper working of the application.

Now we can fill the required fields and, pressed the "Add" button and our new contact was added after pressing “Submit” button. If you want to change or update any of the information for your contacts, you can do so by clicking "Edit" button. Also there is an option to search contacts by name by clicking “Find” the contact button.

We can save the addresses we typed for the future accessing for addresses and export it to a folder. The “Load” button let’s to import the addresses loading into the application. If we need to remove the address of some particular person, we have also an option to “Remove” the contact.

Chapter 4

CONCLUSION

Storing addresses of your contacts in a digital format lets you store large volumes of addresses in less space than paper would take up in a better manner. This is a very basic Address Book program that was created with ease of use in mind. In this application we can store the data such as phone number,address and email id of a person and retrieve it whenever we wanted to check it.

The application worked as well. It is now performing all the features we aimed for. A project means a lot of experience. We sincerely thank our teachers to give us an opportunity to form groups and complete the project and help us improve our knowledge of Graphical User Interfaces(GUI) Programming

Chapter 5

REFERENCES

[1] https://www.tutorialspoint.com/pyqt/
[2] https://stackoverflow.com/
[3] https://www.geeksforgeeks.org/python-designing-gui-applications-using-pyqt/
[4] https://www.saltycrane.com/blog/2008/01/how-to-install-pyqt4-on-ubuntu-linux/
[5] https://www.riverbankcomputing.com/software/pyqt/