Mini Project Report on

GUI Based English Dictionary

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor Of Engineering

in

Electronics and Telecommunication Engineering

by

Atharva Deherkar (201902008)
Purva Hambire (201902017)
Sumeet Sharma (201902029)
Kritika Singh (201902033)

For the completion of termwork for Skill Lab

UNIVERSITY OF MUMBAI

Department of Electronics and Telecommunication Engineering
Xavier Institute of Engineering
Mahim(West), Mumbai-400016
(2020-2021)

Introduction

Fast-evolving technology, in recent years, has seen the emergence of abundant digital dictionaries. There is a large number of studies on English learners' E-dictionary use, and the results suggest that paper dictionaries are losing popularity and that E-dictionaries are gaining importance among English learners.

This ***GUI Based English Dictionary*** provides the user with the meaning and as well as antonyms and synonyms of the word.

Ojective

The objective of our project is to create a simplified system for learners which can eliminate the need of carrying huge paper dictionaries and make tasks easier and faster. This project can be used by students , foreign students and also normal people. The dictionary use can assist the learners to tackle the unknown words' meanings.

Problem Statement

The conventional, bulky, fragile, paper dictionary has limited number of vocabulary in it. Therefore, a user friendly E-dictionary is needed to make search of words easier and more accessible.

Implementation

Concepts Used

Here are the concepts used in our project.

- GUI based Application (using PyQt5 module)
- Object Oriented Programming
  - Classes
  - Objects
  - Functions

- Conditonal Statements
- Loops
- Exception Handling

Algorithm

Step 01: Start the program

Step 02: Enter word to find it's Meaning/Synonym/Antonym

Step 03: If internet is connected, goto **Step 05**, else goto **Step 04**

Step 04: Display internet connection error message

Step 05: If **Meaning button** is pressed, then goto **Step 08**

Step 06: If **Synonym button** is pressed, then goto **Step 08**

Step 07: If **Antonym button** is pressed, then goto **Step 08**

Step 08: If the word present in Dictionary, goto **Step 09**, else goto **Step 10**

Step 09: Display **Meaning/Synonym/Antonym** in Answer Window

Step 10: Display word not found error message

Step 11: If **Close button** is pressed, then the program will stop, else repeat from **Step 02**

Source Code

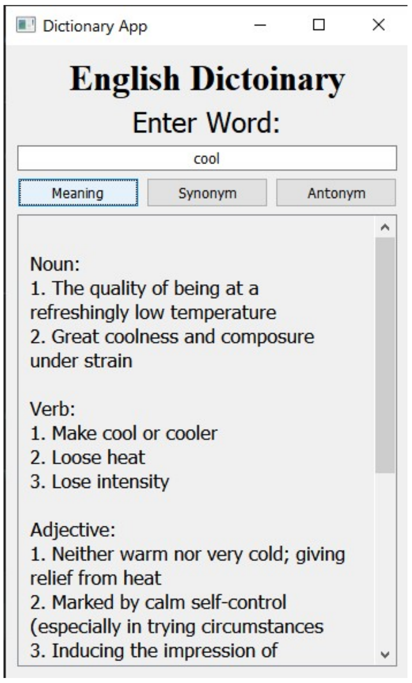
```
from PyQt5 import QtCore, QtGui, QtWidgets
from PyDictionary import PyDictionary as Eng_dict
from urllib.request import urlopen


class Ui_MainWindow(object):
        def setupUi(self, MainWindow):
                MainWindow.setObjectName("MainWindow")
                MainWindow.resize(350, 550)

                self.centralwidget = QtWidgets.QWidget(
                    MainWindow)
```

```
self.centralwidget.setObjectName(”centralwidget”
   )

self.gridLayout = QtWidgets.QGridLayout(self.
   centralwidget)
self.gridLayout.setObjectName(”gridLayout”)

self.Title = QtWidgets.QLabel(self.centralwidget
   )
font = QtGui.QFont()
font.setFamily(”Times␣New␣Roman”)
font.setPointSize(20)
font.setBold(True)
font.setWeight(75)
self.Title.setFont(font)
self.Title.setAlignment(QtCore.Qt.AlignCenter)
self.Title.setObjectName(”Title”)
self.gridLayout.addWidget(self.Title, 0, 0, 1,
   3)

self.Search_Box = QtWidgets.QLineEdit(self.
   centralwidget)
self.Search_Box.setText(””)
self.Search_Box.setAlignment(QtCore.Qt.
   AlignCenter)
self.Search_Box.setObjectName(”Search_Box”)
self.gridLayout.addWidget(self.Search_Box, 2, 0,
    1, 3)

self.Meaning_button = QtWidgets.QPushButton(self
   .centralwidget)
self.Meaning_button.setObjectName(”
   Meaning_button”)
self.gridLayout.addWidget(self.Meaning_button,
   3, 0, 1, 1)
self.Meaning_button.clicked.connect(self.
   clicked_search_meaning)

self.Synonym_button = QtWidgets.QPushButton(self
    .centralwidget)
self.Synonym_button.setObjectName("
    Synonym_button")
self.gridLayout.addWidget(self.Synonym_button,
    3, 1, 1, 1)
self.Synonym_button.clicked.connect(self.
    clicked_search_synonym)

self.Antonym_button = QtWidgets.QPushButton(self
    .centralwidget)
self.Antonym_button.setObjectName("
    Antonym_button")
self.gridLayout.addWidget(self.Antonym_button,
    3, 2, 1, 1)
self.Antonym_button.clicked.connect(self.
    clicked_search_antonym)

self.Enter_Statement = QtWidgets.QLabel(self.
    centralwidget)
font = QtGui.QFont()
font.setPointSize(16)
self.Enter_Statement.setFont(font)
self.Enter_Statement.setAlignment(QtCore.Qt.
    AlignCenter)
self.Enter_Statement.setObjectName("
    Enter_Statement")
self.gridLayout.addWidget(self.Enter_Statement,
    1, 0, 1, 3)

self.scrollArea = QtWidgets.QScrollArea(self.
    centralwidget)
self.scrollArea.setWidgetResizable(True)
self.scrollArea.setObjectName("scrollArea")
self.scrollAreaWidgetContents = QtWidgets.
    QWidget()
self.scrollAreaWidgetContents.setGeometry(QtCore
    .QRect(0, 0, 320, 335))
self.scrollAreaWidgetContents.setObjectName("
    scrollAreaWidgetContents")
```

```
        self.gridLayout_2 = QtWidgets.QGridLayout(self.
            scrollAreaWidgetContents)
        self.gridLayout_2.setObjectName("gridLayout_2")

        self.Answer_window = QtWidgets.QLabel(self.
            scrollAreaWidgetContents)
        font = QtGui.QFont()
        font.setPointSize(11)
        self.Answer_window.setFont(font)
        self.Answer_window.setText("")
        self.Answer_window.setWordWrap(True)
        self.Answer_window.setObjectName("Answer_window"
            )
        self.gridLayout_2.addWidget(self.Answer_window,
            0, 0, 1, 1)
        self.scrollArea.setWidget(self.
            scrollAreaWidgetContents)
        self.gridLayout.addWidget(self.scrollArea, 4, 0,
             1, 3)

        MainWindow.setCentralWidget(self.centralwidget)
        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow
            )

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow
            ", "Dictionary␣App"))
        self.Synonym_button.setText(_translate("
            MainWindow", "Synonym"))
        self.Title.setText(_translate("MainWindow", "
            English␣Dictoinary"))
        self.Meaning_button.setText(_translate("
            MainWindow", "Meaning"))
        self.Enter_Statement.setText(_translate("
            MainWindow", "Enter␣Word:"))
        self.Antonym_button.setText(_translate("
            MainWindow" , "Antonym"))

    def is_internet_available(self):
        try:
            urlopen('http://216.58.192.142', timeout
              =1)
            return True
        except:
            return False
    print(is_internet_available)

    def clicked_search_meaning(self):
        try:
            if self.is_internet_available():
                self.Eng_meaning = Eng_dict.
                  meaning(self.Search_Box.text
                  ().casefold())
                self.mean = ""
                for k, v in self.Eng_meaning.
                  items():
                    c = 0
                    if k == 'Noun' or k == '
                      Verb' or k == '
                      Adjective' or k == '
                      Adverb':
                        self.mean = self
                          .mean + "\n"
                          + k + ":\n"
                    for m in v:
                        c += 1
                        self.mean = self
                          .mean + str(c
                          ) + ".␣" + m.
                          capitalize()
                          + "\n"
                self.Answer_window.setText(self.
                  mean)
                print(self.mean)
                print(self.Eng_meaning)
            else:
                self.Answer_window.setText("You␣
                  are␣not␣connected␣to␣internet
                  !")
        except:
            self.Answer_window.setText("Your␣word␣is
              ␣not␣present␣in␣the␣Dictionary!")

    def clicked_search_synonym(self):
        try:
            if self.is_internet_available():
                self.Eng_synonym = Eng_dict.
                  synonym(self.Search_Box.text
                  ().casefold())
                self.syn = "Synonym:\n"
                c = 0
                for s in self.Eng_synonym:
                    c += 1
                    self.syn = self.syn +
                      str(c) + ".␣" + s.
                      capitalize() + "\n"
                self.Answer_window.setText(self.
                  syn)
                print(self.syn)
                print(self.Eng_synonym)
            else:
                self.Answer_window.setText("You␣
                  are␣not␣connected␣to␣internet
                  !")
        except:
                self.Answer_window.setText("Your
                  ␣word␣is␣not␣present␣in␣the␣
                  Dictionary!")

    def clicked_search_antonym(self):
        try:
            if self.is_internet_available():
                self.Eng_antonym = Eng_dict.
                  antonym(self.Search_Box.text
                                ().casefold())
                            self.ant = "Antonym:\n"
                            c = 0
                            for a in self.Eng_antonym:
                                    c += 1
                                    self.ant = self.ant +
                                        str(c) + ". " + a.
                                        capitalize() + "\n"
                            self.Answer_window.setText(self.
                                ant)
                            print(self.ant)
                            print(self.Eng_antonym)
                    else:
                            self.Answer_window.setText("You 
                                are not connected to internet
                                !")
            except:
                    self.Answer_window.setText("Your word is
                         not present in the Dictionary!")


if __name__ == "__main__":
        import sys
        app = QtWidgets.QApplication(sys.argv)
        MainWindow = QtWidgets.QMainWindow()
        ui = Ui_MainWindow()
        ui.setupUi(MainWindow)
        MainWindow.show()
        sys.exit(app.exec_())
```

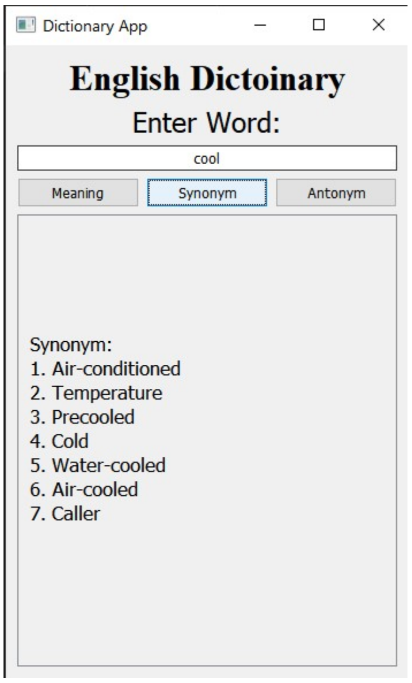
Result

These are the output images of our project:



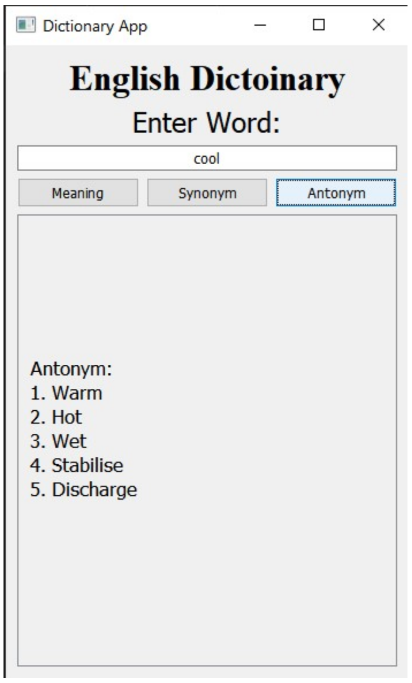
```
Noun:
1. The quality of being at a
refreshingly low temperature
2. Great coolness and composure
under strain

Verb:
1. Make cool or cooler
2. Loose heat
3. Lose intensity

Adjective:
1. Neither warm nor very cold; giving
relief from heat
2. Marked by calm self-control
(especially in trying circumstances
3. Inducing the impression of
```

Figure 1: GUI Application showing **Meaning** of the entered word



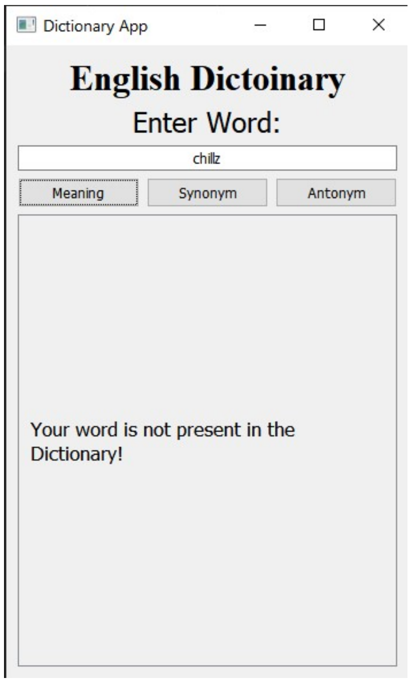
```
Synonym:
1. Air-conditioned
2. Temperature
3. Precooled
4. Cold
5. Water-cooled
6. Air-cooled
7. Caller
```

Figure 2: GUI Application showing **Synonym** of the entered word



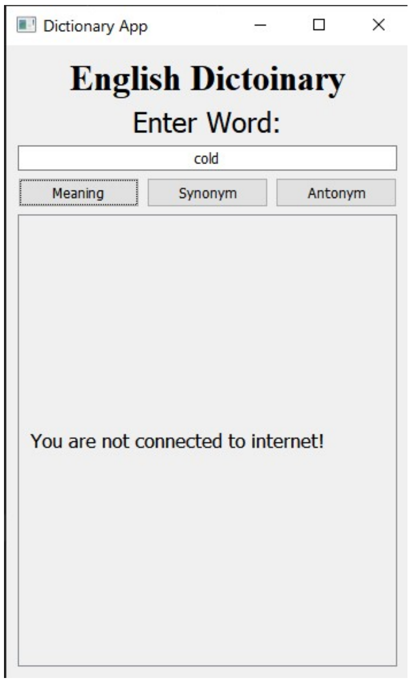
Antonym:
1. Warm
2. Hot
3. Wet
4. Stabilise
5. Discharge

Figure 3: GUI Application showing **Antonym** of the entered word



Your word is not present in the Dictionary!

(a) **No word found** error



You are not connected to internet!

(b) **Internet connection** error

Figure 4: GUI Application showing different types of **Errors**

Conclusion

The digitalization of paper dictionary can help us to find meaning of words along with their synonyms and antonyms within seconds. Further this project can also be used for other languages in future.

In this report, we have given an overview of our mini-project ***GUI Based English Dictionary*** using **Python**. Using the concepts mentioned above, we constructed the program and executed it. We have used **PyCharm IDE** as it is user-friendly, and is time-saving since it gives the expected results instantly by pointing out any error (if present) while editing the code.

References

[1] Etsuko Toyoda, ”Usage and efficacy of electronic dictionaries for a language without word boundaries”, *The EuroCALL Review*, Volume 24, Number 2, September 2016

[2] The Learners Way, https://thelearnersway.net/ideas/2014/12/7/dictionaries-vs-internet

[3] Language Learning, https://languagelearning.stackexchange.com/questions/502/what-are-the-advantages-of-a-paper-dictionary-over-an-online-dictionary/519

[4] PyDictionary 2.0.1, https://pypi.org/project/PyDictionary/#description

[5] Getting Started with PyQt, https://wiki.python.org/moin/PyQt/Tutorials