(Autonomous)

A Micro project Report

on

Analysing your level of English

By

PENUMOLU HAMPI
18BQ1A05F2

PENDYALA MOUNIKA
18BQ1A05F1

Under the Guidance of

Dr. T.SUDHIR

Department of Computer Science & Engineering

Vasireddy Venkatadri Institute of Technology, Nambur

1. Problem Statement:

The purpose of this program is to analyze your level of English.

2. Introduction to the Project:

In this if the user is new then the user should signup his details.If the user is existing user then the user should login as he have created an account already.

After this the user gives input or type about a topic in the given time.This input compares with three levels of words which are in the form of excel. Later, the user checks in which level his words are more. Finally we gives score to user.

3. Constraints/ Business Rules:

1. The user should enter his details in signup form if he is new.
2. The user should enter his details in login form if he is existing user.
3. The user should not quit the exam after the entrance of exam.

4. Technology/ Software Used:

We all know that to create windows in python with out importing any packages is impossible.So we have imported a

package named tkinter. We also know that to get an excel sheet as input and to get some of the features of python we have imported some of the packages like pandas,xlrd,sys etc.

Tkinter :

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

Some modules that provide Tk support include:

- tkinter.scrolledtext
- tkinter.colorchooser
- tkinter.commondialog
- tkinter.filedialog
- tkinter.font
- tkinter.messagebox
- tkinter.simpledialog
- tkinter.dnd
- turtle

Some inbuilt functions like:

- tk.Frame(self)
- tk.Label(self, text, font)

- tk.Button(self, text,height,width,fg,font,command=lambda: controller.fun_frame())
- tk.Text(self, height, width)
- tk.Listbox(self,height, width)
- tk.__init__(self)

Pandas :

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

There are several ways to create a DataFrame. One way way is to use a dictionary.Another way to create a DataFrame is by importing a new file using Pandas. The **read_excel()** method can read Excel 2003 (.xls) files using the xlrd Python module. Excel 2007+ (.xlsx) files can be read using either xlrd or openpyxl. Binary Excel (.xlsb) files can be read using pyxlsb. The **to_excel()** instance method is used for saving a DataFrame to Excel. Generally the semantics are similar to working with csv data.

Reading Excel files

In the most basic use-case, read_excel takes a path to an Excel file, and the sheet_name indicating which sheet to parse.

ExcelFile can also be called with a xlrd.book.Book object as a parameter. This allows the user to control how the excel file is read. For example, sheets can be loaded on demand by calling xlrd.open_workbook() with on_demand=True.

- The arguments sheet_name allows specifying the sheet or sheets to read.
- The default value for sheet_name is 0, indicating to read the first sheet
- Pass a string to refer to the name of a particular sheet in the workbook.
- Pass an integer to refer to the index of a sheet. Indices follow Python convention, beginning at 0.
- Pass a list of either strings or integers, to return a dictionary of specified sheets.
- Pass a None to return a dictionary of all available sheets.

Xlrd:

xlrd is a library for reading data and formatting information from Excel files, whether they are .xls or .xlsx files.

For example, reading, writing or modifying the data can be done in Python. Also, user might have to go through various sheets and retrieve data based on some criteria or modify some rows and columns and do a lot of work.xlrd module is used to extract data from a spreadsheet.

Sys module:

The sys module provides information about constants, functions and methods of the Python interpreter. dir(system) gives a summary of the available constants, functions and methods. Another possibility is the help() function. Using help(sys) provides valuable detail information.

It's also possible to redirect the output into a file:

$ python streams.py<number.txt>output.txt

Database Connection:

Postgresql is used for the database connection.

5.Code:

Analyze your level of English in detail:

The entire code is developed by creating different classes and each class works as a frame.

###This is to create different frames:

```
class AnalysingLevelOfEnglish(tk.Tk):

    def __init__(self, *args, **kwargs):

        tk.Tk.__init__(self, *args, **kwargs)
        container = tk.Frame(self)

        self.title("AnalysingLevelOfEnglish")
        self.geometry("1500x1500")
        container.pack(side="top", fill="both", expand = True)

        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)

        self.frames = {}
        self.connection = None
        self.cursor = None
        self.connect_to_db()

        for F in (Welcome, Signin, Signup , Login , Startexam , Exam ,
Finished , Improve , Thankyou):

            frame = F(container, self)

            self.frames[F] = frame

            frame.grid(row=0, column=0, sticky="nsew")

        self.show_frame(Welcome)

    def show_frame(self, cont):

        frame = self.frames[cont]
        frame.tkraise()
```

```
    def connect_to_db(self):
        try:
            # Connect to your PostgreSQL database
            print("Database connection successful")
        except Exception as e:
            print("Error connecting to database:", e)

    def fetch_users(self):
        try:
            query = "SELECT * FROM Users"
            self.cursor.execute(query)
            rows = self.cursor.fetchall()
            return rows  # Return the fetched rows
        except Exception as e:
            print("Error fetching users:", e)
            return []

    def close_db(self):
        ###to close db
            print("Database connection closed")

    def __del__(self):
        self.close_db()  # Ensure the database connection is closed when the
app is closed
```

```
class Welcome;

### To welcome the user.
```

class Signin:

###To ask the user whether he is new user or existing user.

###This is the class to the take details of users if he is new.

```
class Signup(tk.Frame):


   def __init__(self, parent, controller):
      tk.Frame.__init__(self, parent)


      label1 = tk.Label(self, text="Enter your details!!!",fg="green", font=LARGE_FONT)
      label1.pack(pady=10,padx=10)


      self.name = tk.Label(self, text="Name:", fg="blue",font=LARGE_FONT)
      self.name.pack(pady=20,padx=20)


      self.textBox=tk.Text(self, height=3, width=30)
      self.textBox.pack()


      self.mobile = tk.Label(self, text="Mobile Number:",fg="blue", font=LARGE_FONT)
      self.mobile.pack(pady=25,padx=5)
```

```
        self.textBox=tk.Text(self, height=3, width=30)
        self.textBox.pack()


        self.email = tk.Label(self, text="Email ID:",fg="blue",
font=LARGE_FONT)
        self.email.pack(pady=35,padx=5)


        self.textBox=tk.Text(self, height=3, width=30)
        self.textBox.pack()
        button1 = tk.Button(self, text="Enter into exam
->",height=2,width=25,fg="blue",font=LARGE_FONT,command=lambd
a: controller.show_frame(Startexam))
        button1.pack()
        label = tk.Label(self, text="or", font=LARGE_FONT)
        label.pack(pady=10,padx=10)


        button2 = tk.Button(self,
text="Back",height=2,width=25,fg="blue",font=LARGE_FONT,comman
d=lambda: controller.show_frame(Signin))
        button2.pack()
```

class Login:

This is the class to the take details of users if he is existing user.

This is the class to perform the action according to the users choice i.e, if he wants to start exam or quit.

```
class Startexam(tk.Frame):


    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)


        button1 = tk.Button(self, text="Start Exam",height=2,width=25,fg="blue",font=LARGE_FONT,command=lambda: controller.show_frame(Exam))
        button1.pack()


        label = tk.Label(self, text="or", font=LARGE_FONT)
        label.pack(pady=10,padx=10)


        button2 = tk.Button(self, text="Quit",height=2,width=25,fg="blue",font=LARGE_FONT,command=lambda: controller.show_frame(Thankyou))
        button2.pack()
```

This is to read data from excel files:

```
        sheet1 = pd.read_excel(r'Word Lists.xlsx')
    wb = xlrd.open_workbook("Word Lists.xlsx")
    sheet = wb.sheet_by_index(0)
    sheet.cell_value(0, 0)
    a=b=c=[]
    wb1 = xlrd.open_workbook("Book1.xlsx")
    sheet1 = wb1.sheet_by_index(0)
    sheet1.cell_value(0, 0)

    wb2 = xlrd.open_workbook("Book2.xlsx")
    sheet2 = wb2.sheet_by_index(0)
    sheet2.cell_value(0, 0)



    for i in range(sheet.nrows):
        a.append(sheet.cell_value(i, 0))
    for i in range(sheet1.nrows):
        b.append(sheet1.cell_value(i, 0))
    for i in range(sheet2.nrows):
        c.append(sheet2.cell_value(i, 0))
```

###This is to insert three level words into listbox

```
def write(self,d):
    k=0
```

```
    for key in d:
        k+=1
        self.listbox.insert(END, '{}: {} - {}'.format(k,key, d[key]))
out=write(self,d1)
print(out)
def write1(self,d2):
    k=0
    for key in d2:
        k+=1
        self.listbox1.insert(END, '{}: {} - {}'.format(k,key, d2[key]))
 out1=write1(self,d2)
print(out1)
def write2(self,d3):
      k=0
     for key in d3:
         k+=1
         self.listbox2.insert(END, '{}: {} -{}'.format(k,key, d3[key]))
out2=write2(self,d3)
print(out2)


   def __init__(self, parent, controller):
       tk.Frame.__init__(self, parent)

       label1 = tk.Label(self, text="          ", font=LARGE_FONT)
       label1.pack(pady=50,padx=50)

       label = tk.Label(self, text="Edit your text!!!",fg="green",
font=('arial',16,'bold'))
       label.pack(pady=10,padx=10)

       self.text=tk.Text(self,height=30, width=45)
       self.text.pack(side=LEFT, expand=1)


       self.listbox3 = tk.Listbox(self,height=15, width=17)
```

```
        self.listbox3.pack(side=RIGHT)

        self.listbox2 = tk.Listbox(self,height=15, width=17)
        self.listbox2.pack(side=RIGHT)

        self.listbox1 = tk.Listbox(self,height=15, width=17)
        self.listbox1.pack(side=RIGHT)


        self.listbox = tk.Listbox(self,height=15, width=17)
        self.listbox.pack(side=RIGHT)


        label1 = tk.Label(self, text="          ", font=LARGE_FONT)
        label1.pack(pady=300,padx=300)

        self.buttonCal = tk.Button(self,
text="Check",height=2,width=20,fg="blue",font=('arial',16,'bold'),
command=self.Split)
        self.buttonCal.pack(side=LEFT)

        button2 = tk.Button(self,
text="Submit",height=2,width=20,fg="blue",font=('arial',16,'bold'),comm
and=lambda: controller.show_frame(Finished))
        button2.pack(side=RIGHT)



class Finished(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)

        label = tk.Label(self, text="Thank you for taking the
exam",fg="blue",font=('arial',16,'bold'))
        label.pack(pady=10,padx=10)


        button2 = tk.Button(self,
text="next",height=2,width=25,fg="blue",font=('arial',16,'bold'),comman
d=lambda: controller.show_frame(Improve))
        button2.pack()


class Improve(tk.Frame):

    def     init    (self, parent, controller):
```

```
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Do you want to improve your level then
enter into exam", fg="blue",font=LARGE_FONT)
        label.pack(pady=10,padx=10)

        button1 = tk.Button(self, text="Start
Exam",height=2,width=25,fg="blue",font=('arial',16,'bold'),command=la
mbda: controller.show_frame(Exam))
        button1.pack()

        label = tk.Label(self, text="or", font=LARGE_FONT)
        label.pack(pady=10,padx=10)

        button2 = tk.Button(self,
text="Quit",height=2,width=25,fg="blue",font=('arial',16,'bold'),comman
d=lambda: controller.show_frame(Thankyou))
        button2.pack()

class Thankyou(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)

        label1 = tk.Label(self, text="          ", font=LARGE_FONT)
        label1.pack(pady=50,padx=50)


        label = tk.Label(self, text="Thank you",fg="blue",
font=('arial',25,'bold'))
        label.pack(pady=10,padx=10)

app = AnalysingLevelOfEnglish()
app.configure()
app.mainloop()
```

Here we used the last classes like Finished(),class Improve(), Thankyou() for :

class Finished():

###To display the completion of exam to user.

class Improve():

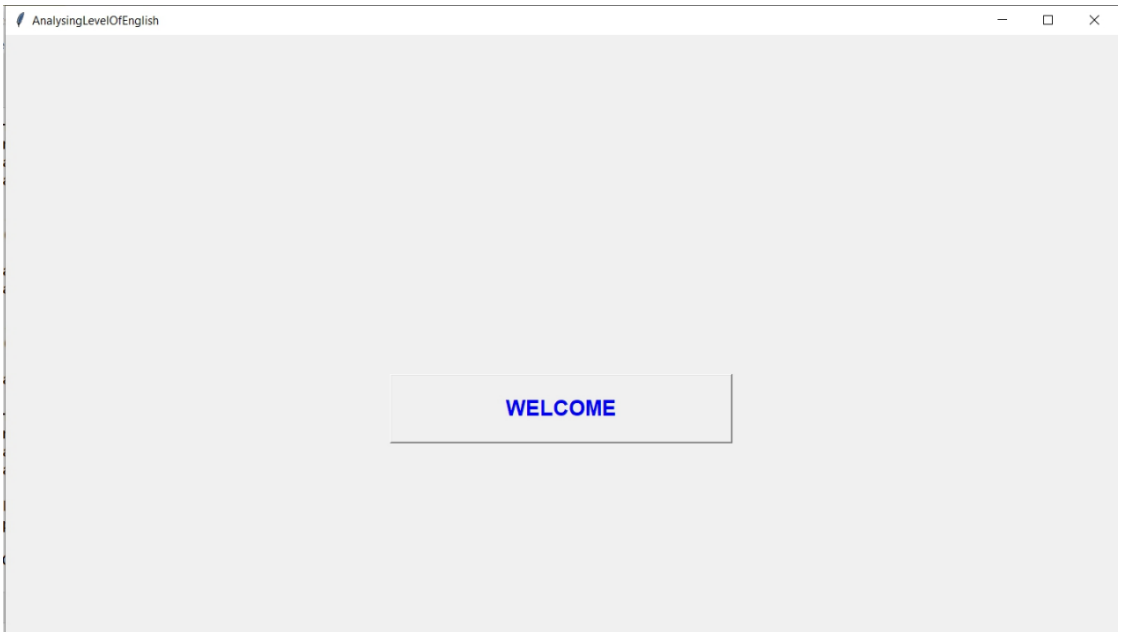
To go back and improve by writing again.

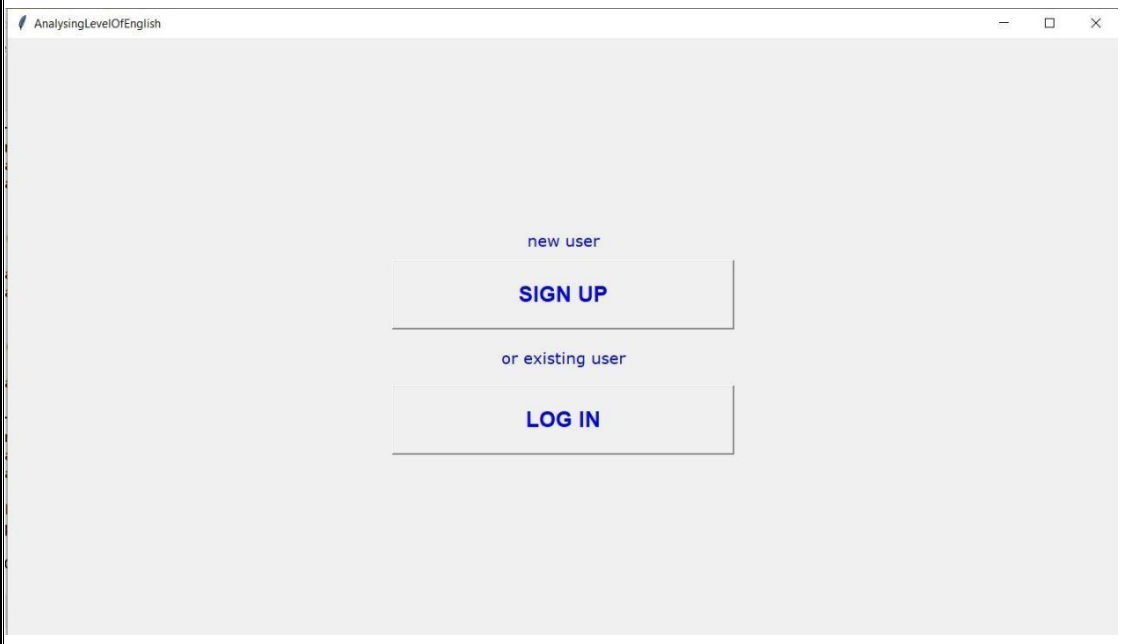
class Thankyou():

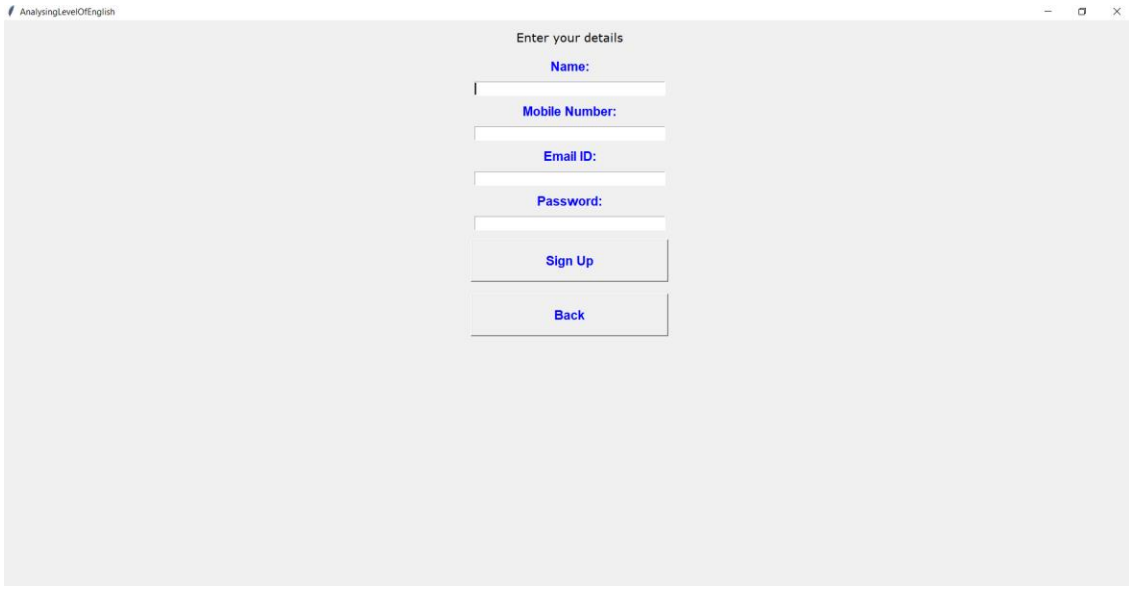
###To display thankyou when the user wants to quit.

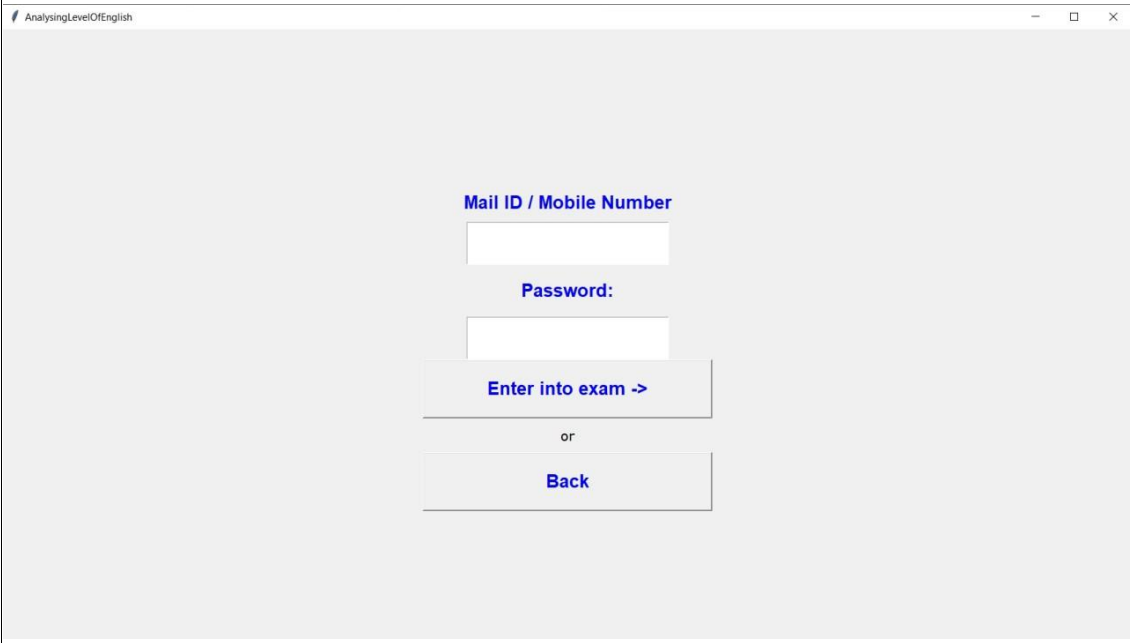
Let us enter to see our outputs:

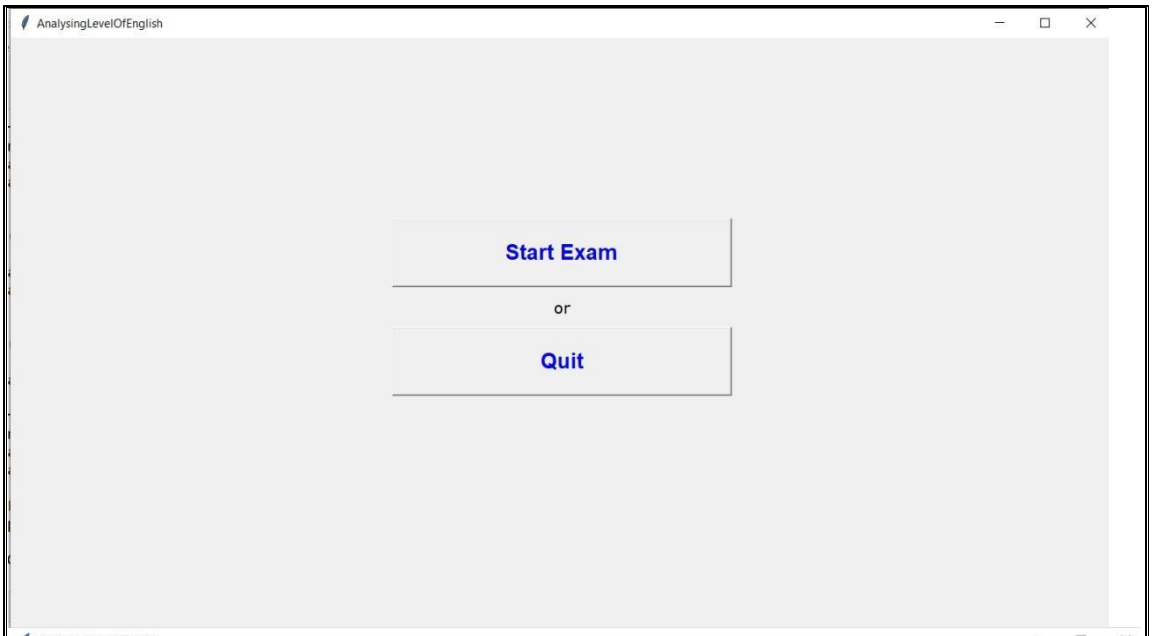
6.Output Screens/Screenshots:

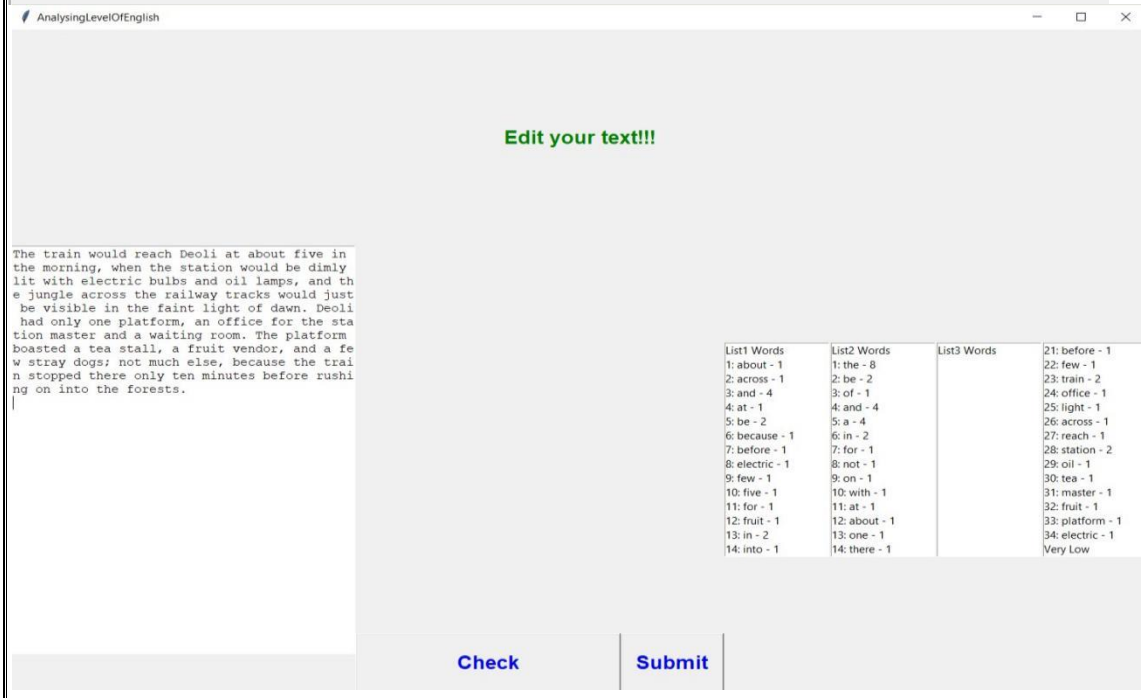


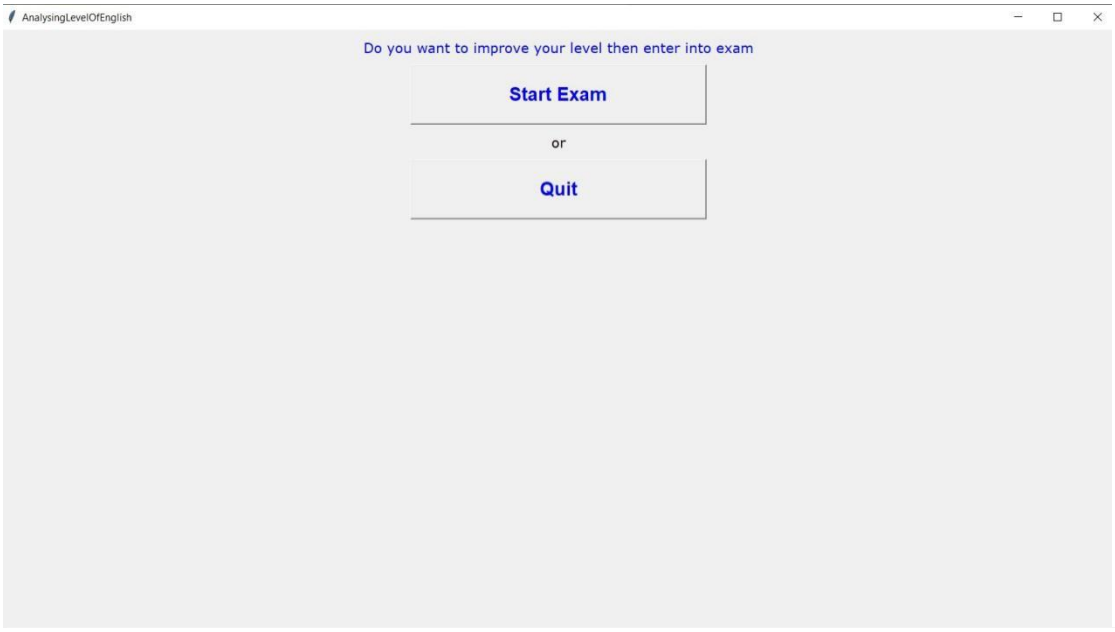


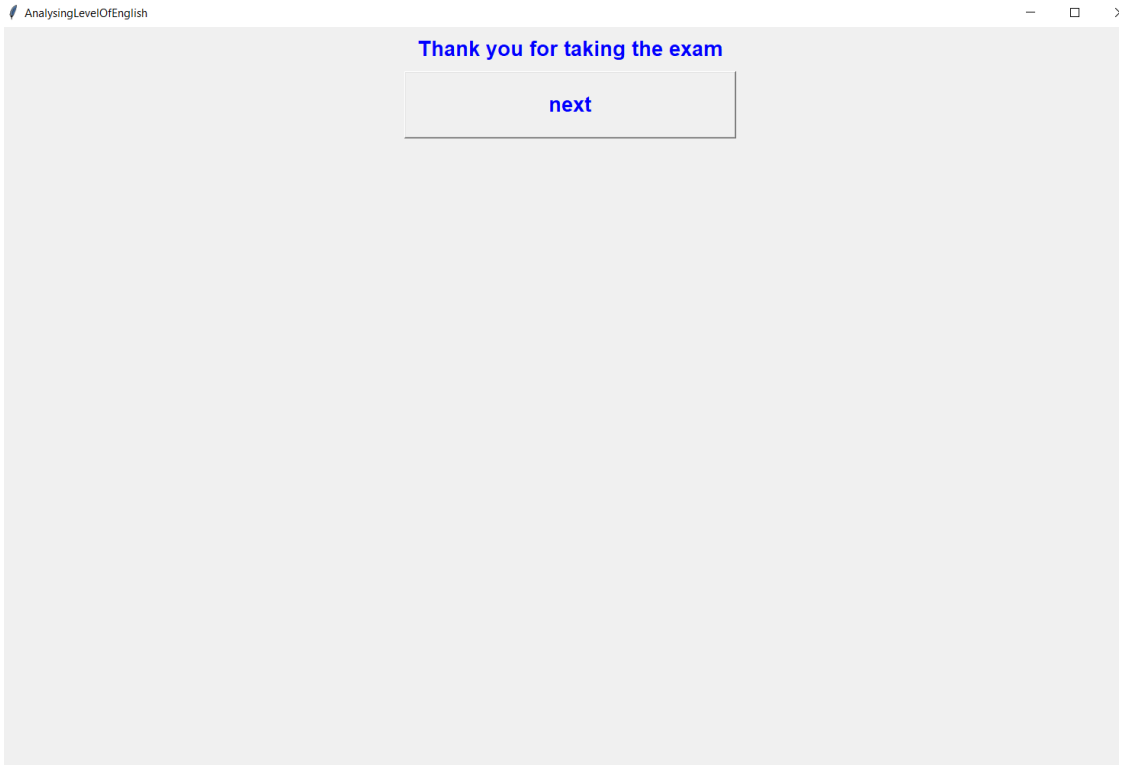

AnalysingLevelOfEnglish
Enter your details
Name:
Mobile Number:
Email ID:
Password:
Sign Up
Back

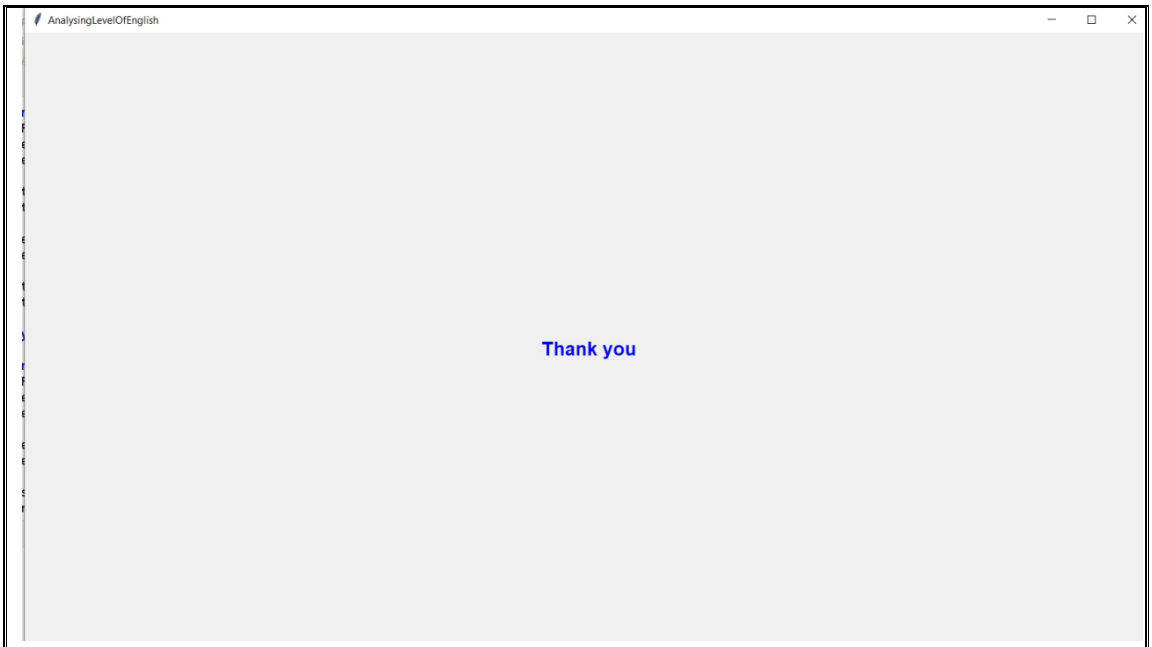

AnalysingLevelOfEnglish
Mail ID / Mobile Number
Password:
Enter into exam ->
or
Back


AnalysingLevelOfEnglish
Start Exam
or
Quit


AnalysingLevelOfEnglish
Edit your text!!!
The train would reach Deoli at about five in the morning, when the station would be dimly lit with electric bulbs and oil lamps, and the jungle across the railway tracks would just be visible in the faint light of dawn. Deoli had only one platform, an office for the station master and a waiting room. The platform boasted a tea stall, a fruit vendor, and a few stray dogs; not much else, because the train stopped there only ten minutes before rushing on into the forests.
List1 Words
1: about - 1
2: across - 1
3: and - 4
4: at - 1
5: be - 2
6: because - 1
7: before - 1
8: electric - 1
9: few - 1
10: five - 1
11: for - 1
12: fruit - 1
13: in - 2
14: into - 1
List2 Words
1: the - 8
2: be - 2
3: of - 1
4: and - 4
5: a - 4
6: in - 2
7: for - 1
8: not - 1
9: on - 1
10: with - 1
11: at - 1
12: about - 1
13: one - 1
14: there - 1
List3 Words
21: before - 1
22: few - 1
23: train - 2
24: office - 1
25: light - 1
26: across - 1
27: reach - 1
28: station - 2
29: oil - 1
30: tea - 1
31: master - 1
32: fruit - 1
33: platform - 1
34: electric - 1
Very Low
Check
Submit


AnalysingLevelOfEnglish
Do you want to improve your level then enter into exam
Start Exam
or
Quit


AnalysingLevelOfEnglish
Thank you for taking the exam
next


AnalysingLevelOfEnglish
Thank you

7. Internet References Used:

In this online sources like python tutorials,tkinter tutorials etc are very helpful to develop this program.

8. Conclusion :

Jonathon stated that,”The English language is a work in progress have fun with it!”

As per his his statement,this exam helps the user about his level in English.This warns the person if he is at lower level to increase his level.This encourages the person if he is at higher level.This also helps the person to improve his level in English.