

**Project Report**

**On**

**JUMBLE JUGGLE**

**Session 2018-19**

**Name of discipline**

**Submitted By**

**RAJAT SHRIVASTAVA**

**1803210117**

**Under the guidance of**

**Gopal Gupta**

**ABES ENGINEERING COLLEGE, GHAZIABAD**

**AFFILIATED TO**

**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW (Formerly UPTU)**

**STUDENT’S DECLARATION**

I hereby declare that the work being presented in this report entitled “JUMBLE JUGGLE” is an authentic record of my work carried out under the supervision of Mr. “GOPAL GUPTA”.

The matter embodied in this report has not been submitted by me for the award of any other degree.

**Dated: Signature of students**

**(Rajat Shrivastava)**

**Department:CSE**

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

¯

**Signature of Supervisor**

**TABLE OF CONTENTS** Page

DECLARATION ......................................................................................................ii

CHAPTER 1

1.1. Problem Introduction

1.1.1. Motivation …………………………………………………………........1

1.1.2. Project Objective ……………………………………………………….1

1.1.3 Scope of the Project ……………………………………………………1

1.2 Introduction to Technologies

1.2.1. Language ………………………………………………………………..2

1.2.2. Library / In-built function / package / modules ………………………3

1.2.3. Hardware Requirement ……………………………………………….3

1.2.4. Software Requirement …………………………………………………3

1.2.5. IDE ……………………………………………………………………….4

CHAPTER 2

2.1 Algorithm / Approach ………………………………………………………………..5

2.2 Work Flow Diagram …………………………………………………………………6

…

2.3 Explain your Code …………………………………………………………………..9

CHAPTER 3

3.1 Snapshots / Output of Interfaces ……………………………………………...…23

CHAPTER 4 (CONCLUSIONS) ...........................................................................27

REFERENCES…………………………………………………….……………….……....28

**CHAPTER 1**

**INTRODUCTION**

‘**Jumble Juggle**’ is a game which aims to improve the vocabulary by solving Jumble words. Jumble words contain some scrambled words which are to be arranged in a sequence such that it forms a meaningful sentence. The game displays various words on different boxes and user can click on the box to form a meaningful sentence. The more interactive and exiciting the learning environment is the more better a student learn. As practical exercise with visual touch are proven to be best for proper growth of children’s mind, ‘Jumble Juggle’ makes the vocabulary of a child better with the modern visual learning methods.

**Problem Introduction**

**Motivation**

The twelve days training session in python encouraged me to apply all those knowledge to build something productive. This game is a result of all those practical research which I did and the knowledge which I gained during training.

**Project Objective**

The objective of the game is to improve the vocabulary by solving Jumble words. The game contains many boxes on which different words are written. The players need to click on different boxes in such a way that when the words are combined they form a meaningful sentence. By this way the user will learn how the sentence are framed in English language and will also learn the basic and important grammar concept which will develop his vocabulary.

**Scope of the Project**

* First user need to create an account in order to play the game.
* After registering they can login to play the game.
* The game contain five levels with the length and difficulty increasing in each level.
* Each level contains five question each.
* 5 points is awarded if the solution is meaningful and 0 if not.
* In the first level only four word sentences are given and one word is increased in each level.
* The game can be played any number of times till all the question are attempted.
* The user can check the correct answer by clicking on ‘See detailed information’ under profile section.
* Only last 5 responses will be shown to the user.
* The user can click on “Restart game” any time to play game from beginning.
* The various other tools like reset password, view profile, edit profile, change password are also given.

**Introduction to Technologies**

**Language**

Python is a simple, general purpose, high level, interpreted, and object-oriented programming language. It is discovered by Guido Van Rossum in 1991. Its Features are:

* It is a Cross-Platform as well as Portable Language.
* It has got name “Python” because its founder was very fond of character Monty Python.
* Python is also called as “Batteries Included” because it’s all libraries are very fast in execution.
* It is known as a multiprogramming language because it can be used with web, enterprise, 3D CAD, etc.
* As it is a interpreted language, so it makes debugging very fast.

**Library / In-Built Programs / module**

In this project, I used the module “**tkinter**”. The module is use to create a GUI for python program. It has various widget namely: Button, Label, Entry, Frame. I have also used “**Messagebox”** module to display a pop-up error or information box in our GUI window. The various widget of ‘tkinter’ are explained below briefly:

1. **Button:** The button widget is used to create a button which may be clicked to execute a particular function or code.
2. **Label:** This widget is used to display single line text in our GUI window. The various properties of text like color, font, size can be set with the help of this widget.
3. **Entry:** This widget is used to create a text box where user can enter the required data.
4. **Frame:** This widget is used to create a frame inside a window which contains its own elements. It can be thought of a small dialogue box required for showing some details.
5. **Menubutton:** This widget is used to create a drop down menu with various options.

**Hardware Requirement**

During project design phase, the hardware used are:

* Processor
* Keyboard
* RAM
* Hard Drive
* Touchpad

**Software Requirement**

During project design phase, the software used are:

* Microsoft Word from Microsoft Office
* Windows 10, an OS as well as a Software, as OS is a superset of software and application programs.

**IDE**

During the design phase, I extensively use Anaconda for writing code. Anaconda is a free and open source distribution of Python and R programming languages for scientific computing that aims to simplify package management and deployment. Package versions are maintained by the package management system conda. Anaconda is written in Python.

**CHAPTER 2**

**SYSTEM DESIGN**

Here in this chapter, I am going to introduce the approach used for this project and explain the whole code thoroughly along with the outputs I get in each case.

**Algorithm**

**Step 1:** Start

**Step 2:** From Login page click on signup button to create an account.

**Step 3:** Enter Phone Number, Password, Name, Email and click on register button.

**Step 4:** Check if file named Phone Number exist.

If yes

Goto step 3

**Step 5:** Create a file named Phone Number and write all data into it.

**Step 6:** Goto Login Page.

**Step 7:** Input Phone Number and password and click on Login button.

**Step 8:** Check if File named Phone number Exist.

If no

Enter Signup

**Step 9:** Check for password.

If matched

Enter Game

**Step 10:** In game click on the required boxes to generate a sentence.

**Step 11:** Write the user’s response in file and check if sentence is meaningful.

If yes

Increment points by 5 and display next question.

Else

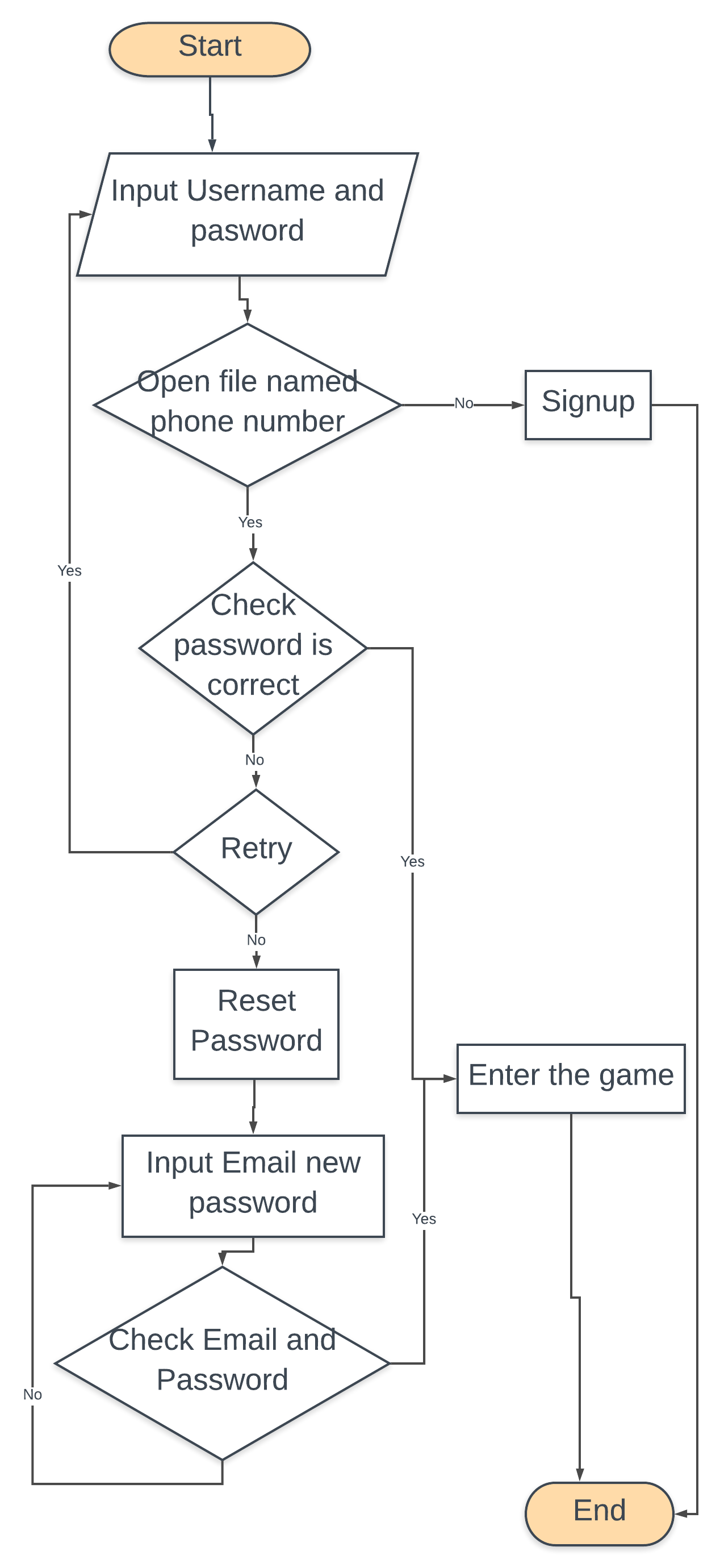
Display next question

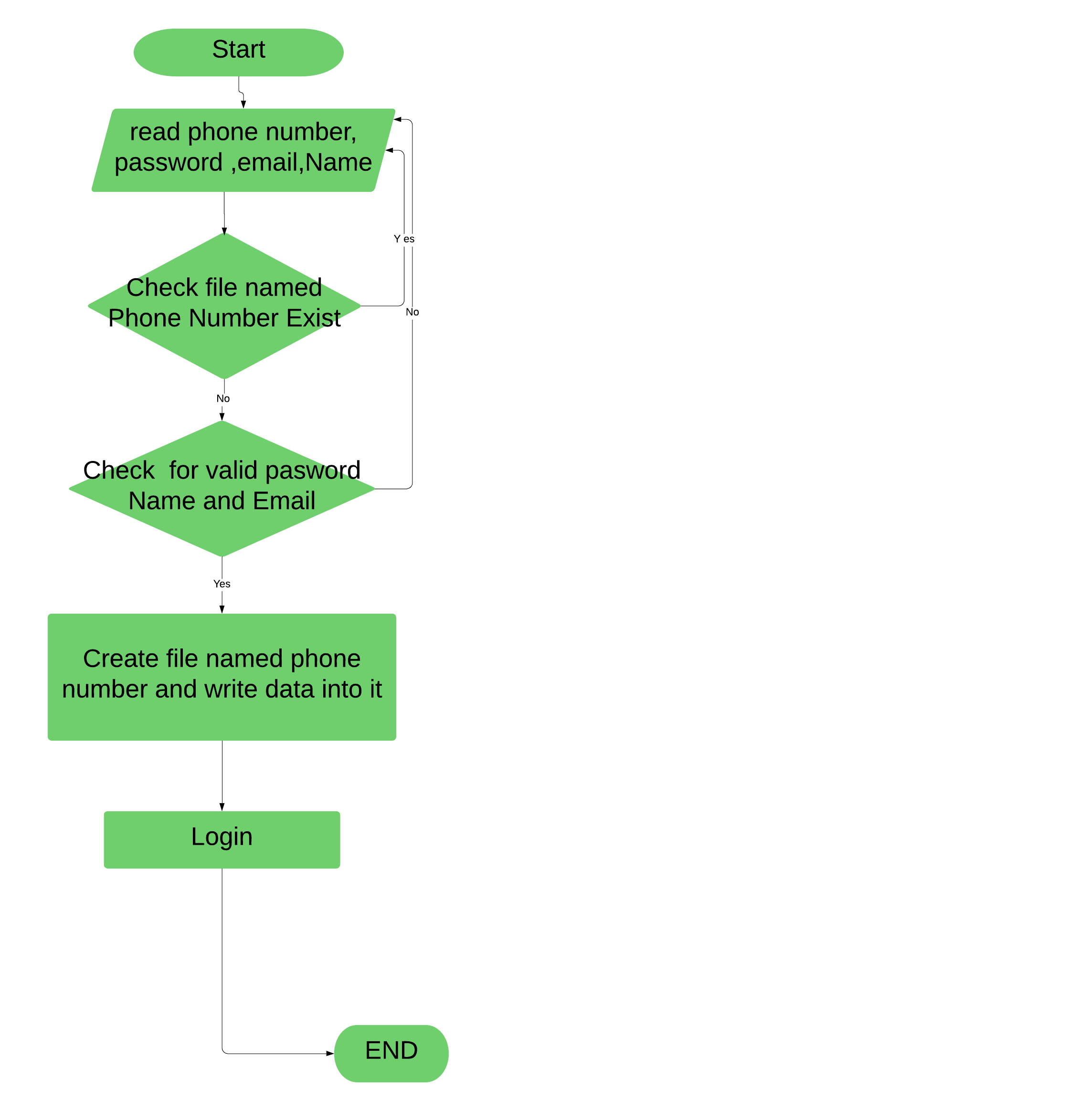
**Step 12:** Repeat step 10 and 11 till list of question reaches end.

**Step 13:** Show Result.

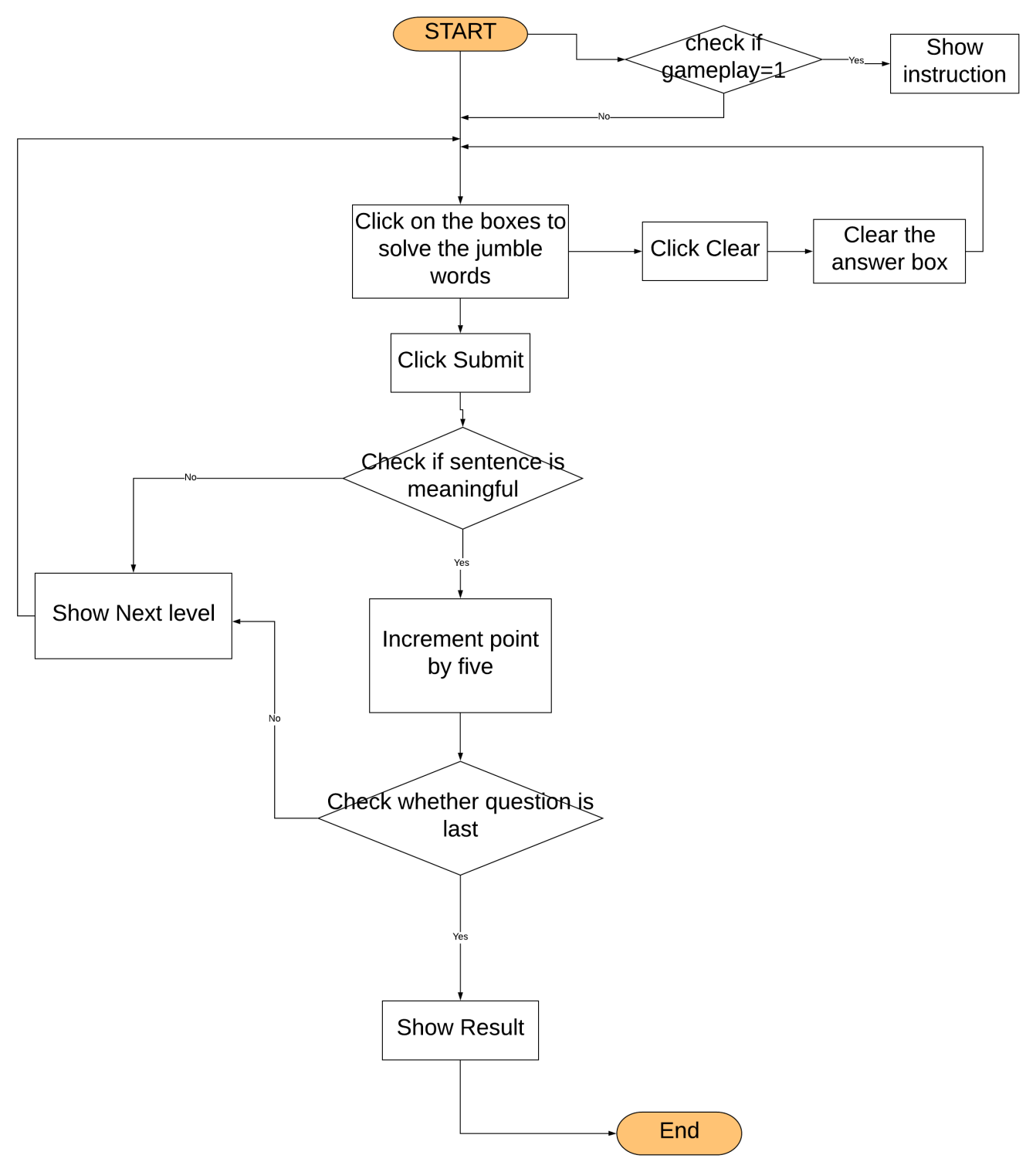
**Step 14:** End.

**Work Flow Diagram**

1. **Flow Chart for Login Module**
2. **Flow Chart for signup Module**

****

1. **Flow Chart for Game Module**

****

**Explain The Code**

import tkinter as tk **#The module for GUI**

from tkinter import \*

from tkinter import messagebox **#The module for Pop-up Box**

import os

**# Defining all question with options and answers**

question1={"option":{1:"twice",2:"a day",3:"brush your",4:"teeth"},"answer":"brush your teeth twice a day "}

question2={"option":{1:"a story",2:"narrating",3:"tomorrow",4:"I will be"},"answer":"I will be narrating a story tomorrow "}

question3={"option":{1:"pray to",2:"we should",3:"daily",4:"god"},"answer":"we should pray to god daily "}

question4={"option":{1:"father",2:"my",3:"me",4:"trust"},"answer":"my father trust me "}

question5={"option":{1:"Delhi",2:"fort",3:"is in",4:"Red"},"answer":"Red fort is in Delhi "}

question6={"option":{1:"the",2:"ball",3:"with",4:"Rohan",5:"is playing"},"answer":"Rohan is playing with the ball "}

question7={"option":{1:"love",2:"of others",3:"good\nmanners",4:"win the",5:"and respect"},"answer":"good manners win the love and respect of others "}

question8={"option":{1:"the",2:"the",3:"kept",4:"rail safe",5:"sleeper"},"answer":"the sleeper kept the rail safe "}

question9={"option":{1:"the",2:"jumped",3:"the dog",4:"pond",5:"into"},"answer":"the dog jumped into the pond "}

question10={"option":{1:"is",2:"to school",3:"late",4:"Rishu",5:"always"},"answer":"Rishu is always late to school "}

question11={"option":{1:"who is",2:"strangers",3:"respectful",4:"a person",5:"even",6:"like"},"answer":"even strangers like a person who is respectful "}

question12={"option":{1:"saves us",2:"turns away",3:"a\nsoft\nanswer",4:"anger and",5:"a pitfall",6:"from many"},"answer":"a soft answer turns away anger and saves us from many a pitfall "}

question13={"option":{1:"when",2:"best",3:"good\nmanners\ncan",4:"one is",5:"be learnt",6:"young"},"answer":"good manners can be learnt best when one is young "}

question14={"option":{1:"deadly\nweapons",2:"of",3:"science\nhas",4:"warfare",5:"given",6:"man"},"answer":"science has given man deadly weapons of warfare "}

question15={"option":{1:"armed",2:"which are",3:"miracles",4:"science\nhas",5:"man\nwith\ninventions",6:"not less than"},"answer":"science has armed man with inventions which are not less than miracles "}

**#creating a list and insrting all question in it**

level1=[]

level1.insert(0,question1)

level1.insert(1,question2)

level1.insert(2,question3)

level1.insert(3,question4)

level1.insert(4,question5)

level1.insert(5,question6)

level1.insert(6,question7)

level1.insert(7,question8)

level1.insert(8,question9)

level1.insert(9,question10)

level1.insert(10,question11)

level1.insert(11,question12)

level1.insert(12,question13)

level1.insert(13,question14)

level1.insert(14,question15)

exp=""  **#The variable to hold user answer**

i=0  **#The variable to hold question number**

point=0 **#Variable of points**

gameplay=0 **# The variable telling gameplay**

**#The login Function**

def main():

window=tk.Tk()

window.title("Jumple Juggle")

window.geometry('900x900')

window.configure(background="#82E0AA")

window.iconbitmap("Game.ico")

def login():

def forgotpass():

def check():

flag=1

file=open(user,"rt")

data=file.readlines()

file.close()

if(testem.get()!=data[2][:-1]):

messagebox.showerror("Forgot Password","The Email you entered did not matched with our records")

flag=0

else:

if(len(getpass.get())<8):

messagebox.showerror("Forgot Password","Password must be eight letters long")

flag=0

if(getpass.get()!=getpass1.get()):

messagebox.showerror("Forgot Password","Confirm Password don't match")

flag=0

if(flag==1):

data[0]=getpass.get()+"\n"

messagebox.showinfo("Forgot Password","Password reset successful")

file=open(user,"wt")

file.writelines(data)

file.close()

forgot.destroy()

window.destroy()

main()

def hide():

forgot.destroy()

forgot=Frame(window,bg="#E74C3C",height="50",width="60")

forgot.grid(row=3,column=2,rowspan=50,columnspan="90",sticky="N")

Label(forgot,text="Reset Password",font=("Britannic Bold","18"),bg="#E74C3C",width="50").grid(row=1,column=0,columnspan="4")

cut=Button(forgot,text="X",cursor="hand2",command=hide,bg="#E74C3C",font=("Arial Black","12"),bd=0,highlightcolor="Red")

cut.grid(row=1,column=4,sticky="E")

Label(forgot,text="Enter your Email",font=("Elephant","16"),bg="#E74C3C").grid(row=3,column=1,pady=4,ipady=2,sticky="E")

testem=Entry(forgot,bd=0,font=("Arial Black","16"),foreground="#17202A")

testem.grid(row=3,column=2,pady=4,ipady=2,sticky="W")

Label(forgot,text="Enter new password",font=("Elephant","16"),bg="#E74C3C").grid(row=5,column=1,pady=4,ipady=2,sticky="E")

getpass=Entry(forgot,show="\*",font=("Arial Black","16"),bd=0,foreground="#17202A")

getpass.grid(row=5,column=2,pady=4,ipady=2,sticky="W")

Label(forgot,text="Confirm password",font=("Elephant","16"),bg="#E74C3C").grid(row=7,column=1,pady=4,ipady=2,sticky="E")

getpass1=Entry(forgot,show="\*",font=("Arial Black","16"),bd=0,foreground="#17202A")

getpass1.grid(row=7,pady=4,ipady=2,column=2,sticky="W")

chk=Button(forgot,text="Reset Password",cursor="hand2",font=("Cooper Black","12"),command=check,bd=0,height="2",bg="#F7DC6F",fg="Black")

chk.grid(row=9,column=2,sticky="W",padx=2,ipadx=3,ipady=2,pady=2)

flag=1

user=phone.get()

pas=password.get()

if(str.isdigit(user)==False or len(user)!=10):

messagebox.showerror("Login", "Input a valid number")

loginph.set("")

flag=0

if(pas==""):

messagebox.showerror("Login", "Enter password first")

flag=0

if(flag==1):

if(os.path.exists(user)):

file=open(user,"rt")

text=file.readline()

text=text[0:-1]

if(text==pas):

window.destroy()

game(user)

else:

msg=messagebox.askretrycancel("Login","Wrong Password")

if(msg==False):

forgotpass()

lpass.set("")

else:

msg=messagebox.askquestion("Login","This number is not registered.Do you want to signup?",icon='warning')

if(msg=="yes"):

window.destroy()

signup()

Frame(window,bg="#82E0AA",height="900",width="160").grid(row=0,column=0,rowspan=50)

Label(window,text="LOGIN",fg="#E74C3C",bg="#82E0AA",font=("LOGIN", 48),justify="center").grid(row=1,column=1,columnspan=4)

Label(window,text="Phone Number",fg="Black",bg="#82E0AA",font=("Elephant", 20),justify="left").grid(sticky="E",row=3,column=2)

loginph= StringVar()

phone=Entry(window,fg="Black",bd=0,font=("Arial","19"),cursor="ibeam",textvariable=loginph)

phone.focus

phone.grid(row=3,column=3,ipadx=2,ipady=2,padx=2,pady=3)

lpass= StringVar()

Label(window,text="Password ",fg="Black",bg="#82E0AA",font=("Elephant", 15),justify="left").grid(sticky="E",row=4,column=2)

phone=Entry(window,fg="Black",bd=0,font=("Arial Black","16"),cursor="ibeam",bg="white",textvariable=loginph)

phone.focus

phone.grid(row=3,column=3,ipadx=2,ipady=2,padx=2,pady=3)

lpass= StringVar()

Label(window,text="Password",fg="Black",bg="#82E0AA",font=("Elephant", 20),justify="left").grid(sticky="E",row=4,column=2)

password=Entry(window,fg="Black",bd=0,cursor="ibeam",font=("Arial Black","16"),bg="White",show="\*",textvariable=lpass)

password.grid(row=4,column=3,ipadx=2,ipady=2,padx=2,pady=3)

Loginbt=Button(window,text="Login",bg="#E74C3C",fg="black",width="20",cursor="hand2",font=("Yelowtail","15"),relief="flat",activeforeground="white",activebackground="#E74C3C",command=login).grid(row=5,column=3)

Label(window,text="or",fg="Black",bg="#82E0AA",font=("Elephant", 20)).grid(row=6,column=3)

def signup1():

window.destroy()

signup()

Button(window,text="Signup",bg="#2ECC71",fg="White",bd=0,cursor="hand2",font=("Britanica","16"),width="20",command=signup1).grid(row=7,column=3)

window.mainloop()

**#The signup function**

def signup():

global point

global i

def register():

flag=1

phn=phone.get()

if(str.isdigit(phone.get())==False or len(phone.get())!=10):

messagebox.showerror("Signup", "Input a valid number")

num.set("")

flag=0

nam=name.get()

nam=list(nam.split())

email=Email.get()

pas=password.get()

if(len(pas)<8):

chkpas.set("")

messagebox.showerror("Signup", "Password must be 8 character long")

flag=0

for j in nam:

if(str.isalpha(j)==False):

messagebox.showerror("Signup","Name must be in letters")

namvar.set("")

flag=0

else:

name1=" ".join(nam)

if("@" not in email or ".com" not in email):

messagebox.showerror("Signup", "Input a valid Email")

em.set("")

flag=0

import os

if(flag==1):

if(os.path.exists(phone.get())):

messagebox.showerror("Signup", "This number is already registered")

num.set("")

else:

file=open(phn,"wt")

file.write(pas)

file.write("\n")

file.write(name1)

file.write("\n")

file.write(email)

file.write("\n")

file.write(str(point))

file.write("\n")

file.write(str(i))

file.close()

messagebox.showinfo("Signup", "You are registered succesfully")

sign.destroy()

main()

return

sign=tk.Tk()

sign.title("SIGN UP")

sign.iconbitmap("Game.ico")

sign.geometry("900x900")

sign.configure(background="#AAB7B8")

Frame(sign,bg="#AAB7B8",height="900",width="160").grid(row=1,column=1,rowspan=50)

Label(sign,text="REGISTER",fg="#9B59B6",bg="#AAB7B8",font=("REGISTER", 48)).grid(row=1,column=2,columnspan=4)

num= StringVar()

Label(sign,text="Phone Number",fg="Black",bg="#AAB7B8",font=("Phone Number", 20),justify="left").grid(sticky="E",row=2,column=2)

phone=Entry(sign,bd=0,fg="Black",font=("Arial Black","14"),cursor="ibeam",textvariable=num)

phone.focus()

phone.grid(row=2,column=3,padx=2,pady=2,ipadx=2,ipady=2)

Label(sign,text="Password",fg="Black",bg="#AAB7B8",font=("Password", 20),justify="left",).grid(sticky="E",column=2,row=3)

chkpas= StringVar()

password=Entry(sign,bd=0,fg="Black",font=("Arial Black","14"),cursor="ibeam",show="\*",textvariable=chkpas)

password.grid(row=3,column=3,padx=2,pady=2,ipadx=2,ipady=2)

Label(sign,text="Name",fg="Black",bg="#AAB7B8",font=("Name", 20),justify="left").grid(sticky="E",row=4,column=2)

namvar= StringVar()

name=Entry(sign,bd=0,fg="Black",font=("Arial Black","14"),cursor="ibeam",textvariable=namvar)

name.grid(row=4,column=3,padx=2,pady=2,ipadx=2,ipady=2)

Label(sign,text="Email",fg="Black",bg="#AAB7B8",font=("Email", 20),justify="left",).grid(sticky="E",row=5,column=2)

em= StringVar()

Email=Entry(sign,bd=0,fg="Black",font=("Arial Black","14"),cursor="ibeam",textvariable=em)

Email.grid(row=5,column=3,padx=2,pady=2,ipadx=2,ipady=2)

signupbt=Button(sign,text="Register",bg="#2ECC71",fg="White",width="25",cursor="hand2",bd=0,font=("Britanica","14"),command=register).grid(row=6,column=2,sticky="E",columnspan=2,ipadx=2,ipady=2,padx=2,pady=2)

Label(sign,text="or",fg="Black",bg="#AAB7B8",font=("Elephant", 20)).grid(row=7,column=3)

def login1():

sign.destroy()

main()

Button(sign,text="Login",bg="#E74C3C",fg="Black",bd=0,cursor="hand2",font=("Britanica","16"),width="24",command=login1).grid(row=8,column=2,columnspan=2,sticky="E")

sign.mainloop()

**#the game function**

def game(phno):

forward=phno

global point

global exp

global gameplay

global i

rajat=tk.Tk()

rajat.configure(bg="#F1C40F")

rajat.iconbitmap("Game.ico")

rajat.geometry("1500x1500")

**#function to give introduction**

def intro():

def destroy():

cross.configure(font="16")

frame.destroy()

frame=Frame(rajat,bg="#2ECC71")

frame.grid(row=2,column=1,rowspan="50",columnspan="90",sticky="N")

intro=Label(frame,text="Instruction",font=("Britannic Bold","18"),width="50",bg="#2ECC71")

intro.grid(row=1,column=0,rowspan=2)

cross=Button(frame,text="X",command=destroy,bg="#2ECC71",font="14",cursor="hand2",bd=0,activeforeground="White",activebackground="#2ECC71")

t1=Label(frame,text="> This game contains many boxes which contain different words written on them.",bg="#2ECC71",height="2",font=("Arial Black","14"))

t1=Label(frame,text="1. This game contain many boxes which contain different words written on them.",bg="#2ECC71",height="2",font=("Arial Black","14"))

t1.grid(row=3,sticky="W")

t2=Label(frame,text="2. You have to click on boxes in such a way that words when joined\ntogether form a meaningful sentece.",bg="#2ECC71",height="2",font=("Arial Black","14"))

t2.grid(row=5,sticky="W")

t3=Label(frame,text="3. Click on the submit button when you are sure sentence is meaningful.",bg="#2ECC71",height="2",font=("Arial Black","14"))

t3.grid(row=7,sticky="W")

t4=Label(frame,text="4. If you think you have clicked on wrong box, you can click on\nclear button to reset your answer.",bg="#2ECC71",height="2",font=("Arial Black","14"))

t4.grid(row=9,sticky="W")

t5=Label(frame,text="5. The game contain three level and 5 question in each level. ",bg="#2ECC71",height="2",font=("Arial Black","14"))

t5.grid(row=11,sticky="W")

t6=Label(frame,text="6. You will get 5 points on correct answer and 0 if answer is wrong. ",bg="#2ECC71",height="2",font=("Arial Black","14"))

t6.grid(row=13,sticky="W")

cross.grid(row=1,column=4,ipadx=2,ipady=2)

if(gameplay==0):

file=open(forward,"rt")

data=file.readlines()

i=int(data[4])

point=int(data[3][:-1])

**#function to change password**

def changepass():

def check():

flag=1

file=open(forward,"rt")

data=file.readlines()

file.close()

if(getcurrpass.get()!=data[0][:-1]):

messagebox.showerror("Change Password","Current password don't match")

flag=0

if(getpass.get()==data[0][:-1]):

messagebox.showerror("Change Password","New password chould be different from old password")

flag=0

if(len(getpass.get())<8):

messagebox.showerror("Change Password","Password must be eight letters long")

flag=0

if(getpass.get()!=getpass1.get()):

messagebox.showerror("Change Password","Confirm Password don't match")

flag=0

if(flag==1):

data[0]=getpass.get()+"\n"

messagebox.showinfo("Forgot Password","Password reset successful")

file=open(forward,"wt")

file.writelines(data)

file.close()

change.destroy()

def hide():

change.destroy()

change=Frame(rajat,bg="#E74C3C",height="50",width="60")

change.grid(row=3,column=2,rowspan=50,columnspan="90",sticky="N")

Label(change,text="Change Password",font=("Britannic Bold","18"),bg="#E74C3C",width="50").grid(row=1,column=0,columnspan="4")

cut=Button(change,text="X",cursor="hand2",command=hide,bg="#E74C3C",font=("Arial Black","12"),bd=0,highlightcolor="Red")

cut.grid(row=1,column=4,sticky="E")

Label(change,text="Enter current password",font=("Elephant","14"),bg="#E74C3C").grid(row=3,column=1,pady=4,ipady=2,sticky="E")

getcurrpass=Entry(change,show="\*",font=("Arial Black","14"),bd=0,foreground="#17202A")

getcurrpass.grid(row=3,column=2,pady=4,ipady=2,sticky="W")

Label(change,text="Enter new password",font=("Elephant","14"),bg="#E74C3C").grid(row=5,column=1,pady=4,ipady=2,sticky="E")

getpass=Entry(change,show="\*",font=("Arial Black","14"),bd=0,foreground="#17202A")

getpass.grid(row=5,column=2,pady=4,ipady=2,sticky="W")

Label(change,text="Confirm password",font=("Elephant","14"),bg="#E74C3C").grid(row=7,column=1,pady=4,ipady=2,sticky="E")

getpass1=Entry(change,show="\*",font=("Arial Black","14"),bd=0,foreground="#17202A")

getpass1.grid(row=7,pady=4,ipady=2,column=2,sticky="W")

chk=Button(change,text="Reset Password",cursor="hand2",font=("Arial Black","12"),command=check,bd=0,height="2",bg="#F7DC6F",fg="Black")

chk.grid(row=9,column=2,sticky="W",padx=2,ipadx=3,ipady=2,pady=2)

**#Function to view responce**

def responce():

responceframe=Frame(rajat,bg="#2ECC71",height="50",width="50")

responceframe.grid(row=3,column=2,rowspan=50,columnspan="90",sticky="N")

global i

global point

rowlabel=4

rowentry=5

filename=forward+"responce"

file=open(filename,"rt")

userans=file.readlines()

file.close()

counter=0

def hideframe():

responceframe.destroy()

Label(responceframe,text="NOTE:Only last 8 responce will be shown",font=("Arial Balck","12"),bg="#2ECC71").grid(row=3,column=2,sticky="E")

Label(responceframe,text="Detailed View",font=("Britannic Bold","18"),bg="#2ECC71",width="50").grid(row=1,column=0,columnspan="4")

Label(responceframe,text="Points",font=("Arial Balck","14"),bg="#2ECC71").grid(row=2,column=1,sticky="E")

crtpoint=StringVar()

Entry(responceframe,font=("Arial Black","18"),bd=0,state="disabled",disabledbackground="#2ECC71",disabledforeground="Black",textvariable=crtpoint).grid(row=2,column=2,sticky="W")

crtpoint.set(point)

Label(responceframe,text="Correct Answer Given",font=("Arial Balck","14"),bg="#2ECC71").grid(row=2,column=3)

crtans=StringVar()

Entry(responceframe,font=("Arial Black","18"),bd=0,disabledforeground="Black",state="disabled",disabledbackground="#2ECC71",textvariable=crtans).grid(row=2,column=4)

crtans.set(point//5)

cut=Button(responceframe,text="X",cursor="hand2",command=hideframe,bg="#2ECC71",font=("Arial Black","12"),bd=0,highlightcolor="Red")

cut.grid(row=1,column=4,sticky="E")

while counter<i:

if(userans[counter][:-1]==level1[counter]["answer"]):

color="Black"

else:

color="#E74C3C"

Label(responceframe,text="Your responce",font=("Elephant","14"),bg="#2ECC71").grid(row=rowlabel,column=1,pady=4,ipady=2,sticky="E")

setresponce=StringVar()

userresponce=Entry(responceframe,font=("Arial Black","14"),state="disabled",bd=0,disabledforeground=color,width="60",textvariable=setresponce)

setresponce.set(userans[counter])

userresponce.grid(row=rowlabel,column=2,pady=4,ipady=2,sticky="W",columnspan="4")

Label(responceframe,text="Correct answer",font=("Elephant","14"),bg="#2ECC71").grid(row=rowentry,column=1,pady=4,ipady=2,sticky="E")

setanswer=StringVar()

correctanswer=Entry(responceframe,font=("Arial Black","14"),state="disabled",bd=0,disabledforeground="#27AE60",width="60",textvariable=setanswer)

setanswer.set(level1[counter]["answer"])

correctanswer.grid(row=rowentry,column=2,pady=4,ipady=2,sticky="W",columnspan="4")

rowlabel+=2

rowentry+=2

if(rowlabel==14):

rowlabel=4

rowentry=5

counter+=1

**#Function to view profile**

def viewprofile():

global i

def hide():

profileframe.destroy()

def update(string,i):

if(i==1):

flag=1

if(string=="Enter new name"):

messagebox.showerror("Jumble Juggle","Please provide input")

flag=0

else:

string=list(string.split())

for j in string:

if(str.isalpha(j)==False):

messagebox.showerror("Signup","Name must be in letters")

getans.set("")

flag=0

else:

name1=" ".join(string)

if(flag==1):

file=open(forward,"rt")

data=file.readlines()

file.close()

data[1]=name1+"\n"

file=open(forward,"w")

file.writelines(data)

file.close()

messagebox.showinfo("Jumble Juggle", "Update Successful")

rajat.destroy()

game(forward)

else:

if(string=="Enter new Email"):

messagebox.showerror("Jumble Juggle","Please provide input")

elif("@" not in string or ".com" not in string):

messagebox.showerror("Jumble Juggle", "Input a valid Email")

else:

file=open(forward,"rt")

data=file.readlines()

file.close()

data[2]=string+"\n"

file=open(forward,"w")

file.writelines(data)

file.close()

messagebox.showinfo("Jumble Juggle", "Update Successful")

rajat.destroy()

game(forward)

def edit(i):

getans=Entry(profileframe,width="30",font=("Berlin Sans FB Demi","14"),bd=0)

getans.grid(row=9,column=1,columnspan=2,padx=2,pady=2,ipady=2)

if(i==1):

getans.insert(0,"Enter new name")

finalbt=Button(profileframe,bd=0,cursor="hand2",bg="#2D69AB",fg="White",font=("Cooper Black","12"),text="Update",command=lambda:update(getans.get(),1))

finalbt.grid(row=10,column=1,columnspan=2,padx=2,ipadx=2,ipady=2)

else:

getans.insert(0,"Enter new Email")

finalbt=Button(profileframe,bd=0,cursor="hand2",bg="#2D69AB",font=("Cooper Black","12"),fg="White",text="Update",command=lambda:update(getans.get(),2))

finalbt.grid(row=10,column=1,columnspan=2,padx=2,ipadx=2,ipady=2)

profileframe=Frame(rajat,bg="#E74C3C",height="50",width="60")

profileframe.grid(row=6,column=1,rowspan=50,columnspan="90",sticky="N")

Label(profileframe,text="Profile",font=("Britannic Bold","18"),bg="#E74C3C",width="50").grid(row=1,column=0,columnspan="4")

cut=Button(profileframe,text="X",command=hide,cursor="hand2",font="14",bg="#E74C3C",bd=0,activebackground="#E74C3C",activeforeground="White")

cut.grid(row=1,column=4,sticky="E")

file=open(forward,"rt")

file.readline()

Label(profileframe,text="Phone Number:",font=("Berlin Sans FB","14"),bg="#E74C3C").grid(row=3,column=1,sticky="E")

no=IntVar()

Entry(profileframe,state="disabled",textvariable=no,font=("Berlin Sans FB","14"),bd=0,disabledbackground="#E74C3C",disabledforeground="#17202A").grid(row=3,column=2,sticky="W")

no.set(forward)

Label(profileframe,text="Name:",font=("Berlin Sans FB","14"),bg="#E74C3C").grid(row=4,column=1,sticky="E")

Name=StringVar()

Entry(profileframe,state="disabled",font=("Berlin Sans FB","14"),textvariable=Name,bd=0,disabledbackground="#E74C3C",disabledforeground="#17202A").grid(row=4,column=2,sticky="W")

tempvar1=file.readline()

Name.set(tempvar1)

ed1=Button(profileframe,text="edit",cursor="hand2",bg="#E74C3C",fg="White",bd=0,command=lambda:edit(1))

ed1.grid(row=4,column=3,sticky="W")

Label(profileframe,text="Email Id:",font=("Berlin Sans FB","14"),bg="#E74C3C").grid(row=5,column=1,sticky="E")

Emailid=StringVar()

Entry(profileframe,state="disabled",font=("Berlin Sans FB","14"),textvariable=Emailid,bd=0,disabledbackground="#E74C3C",disabledforeground="#17202A").grid(row=5,column=2,sticky="W")

tempvar2=file.readline()

Emailid.set(tempvar2)

Label(profileframe,text="Points:",font=("Berlin Sans FB","14"),bg="#E74C3C").grid(row=6,column=1,sticky="E")

setpoint=StringVar()

Entry(profileframe,state="disabled",font=("Berlin Sans FB","14"),textvariable=setpoint,bd=0,disabledbackground="#E74C3C",disabledforeground="#17202A").grid(row=6,column=2,sticky="W")

tempvar3=file.readline()

setpoint.set(tempvar3)

Label(profileframe,text="Cuurently on level:",font=("Berlin Sans FB","14"),bg="#E74C3C").grid(row=7,column=1,sticky="E")

showlevel=StringVar()

Entry(profileframe,state="disabled",font=("Berlin Sans FB","14"),textvariable=showlevel,bd=0,disabledbackground="#E74C3C",disabledforeground="#17202A").grid(row=7,column=2,sticky="W")

tempvar4=file.readline()

showlevel.set(tempvar4)

Label(profileframe,text="Correct Answer given:",font=("Berlin Sans FB","14"),bg="#E74C3C").grid(row=8,column=1,sticky="E")

showques=StringVar()

Entry(profileframe,state="disabled",font=("Berlin Sans FB","14"),textvariable=showques,bd=0,disabledbackground="#E74C3C",disabledforeground="#17202A").grid(row=8,column=2,sticky="W")

showques.set(point//5)

ed3=Button(profileframe,text="View Detailed Information",bg="#E74C3C",cursor="hand2",fg="White",bd=0,command=lambda:responce())

ed3.grid(row=8,column=3,sticky="W")

ed2=Button(profileframe,text="edit",bg="#E74C3C",cursor="hand2",fg="White",bd=0,command=lambda:edit(2))

ed2.grid(row=5,column=3,sticky="W")

if(i==15):

Label(profileframe,text="Congratualation! You have succesfully completed the game.",fg="#F1C40F",font=("Elephant","18"),bg="#E74C3C").grid(row=11,column=1,columnspan=2,sticky="W")

file.close()

**#Function to Restart game**

def restart():

msg=messagebox.askquestion("Jumble Juggle","All your current progress will be lost. Do you want to continue?",icon='warning')

if(msg=="no"):

pass

else:

global point

global i

point=0

i=0

file=open(forward,"rt")

data=file.readlines()

file.close()

data[3]=str(point)+"\n"

data[4]=str(i)

file=open(forward,"wt")

file.writelines(data)

file.close()

rajat.destroy()

game(forward)

def logout():

global exp

global point

global i

global gameplay

file=open(forward,"rt")

data=file.readlines()

file.close()

data[3]=str(point)+"\n"

data[4]=str(i)

file=open(forward,"wt")

file.writelines(data)

file.close()

gameplay=0

exp=""

i=0

point=0

rajat.destroy()

main()

**#defining game interface**

if(i==15):

viewprofile()

setans= StringVar()

ansbox=Entry(rajat,cursor="arrow",text="",width="60",state="disabled",disabledforeground="#17202A",font=("Berlin Sans FB Demi","14"),bd=0,textvariable=setans)

ansbox.grid(row=5,column=1,columnspan=9,sticky="W",ipady="3",pady="4")

rajat.title("Jumble Juggle")

file=open(phno,"rt")

file.readline()

usernam=file.readline()

file.close()

usernam="Hello "+str.title(usernam)

user=Label(rajat,text=usernam,font=("Arial","18"),bg="#F1C40F",fg="#3498DB")

user.grid(row=1,column=4,columnspan=1,sticky="E")

mb= Menubutton (rajat,text="▼",bd=0,bg="#F1C40F",cursor="hand2",font=("Arial","14"),activebackground="#F1C40F",activeforeground="#CB4335")

mb.grid(row=1,column=5,columnspan=2,sticky="NW")

mb.menu = Menu ( mb, tearoff = 0 )

mb["menu"] = mb.menu

mb.menu.add\_cascade( label="View Profile",font=("Impact","14"),command=viewprofile)

mb.menu.add\_cascade( label="View Instruction",font=("Impact","14"),command=intro)

mb.menu.add\_cascade( label="Change Password",font=("Impact","14"),command=changepass)

mb.menu.add\_cascade( label="Restart Game",font=("Impact","14"),command=restart)

mb.menu.add\_cascade( label="View Detailed Information",font=("Impact","14"),command=responce)

mb.menu.add\_cascade( label="Logout",font=("Impact","14"),command=logout)

pointbox=Label(rajat,text="Points:",font=("Britannic Bold","18"),bg="#F1C40F")

pointbox.grid(row=1,column=0,sticky="E")

poin= IntVar()

poin.set(point)

setpoint=Entry(rajat,state="disabled",cursor="arrow",bd=0,textvariable=poin,font=("Britannic Bold","18"),disabledforeground="#17202A",disabledbackground="#F1C40F")

setpoint.grid(row=1,column=1,sticky="W",columnspan=2,ipadx=2,ipady=2,pady=2)

poin.set(point)

**#Function to add jumble words**

def add(got):

global exp

got=got.replace("\n"," ")

exp=exp+got

exp=exp+" "

print(exp)

setans.set(exp)

forward=phno

**#function to check answer**

def check():

global gameplay

global exp

global point

global i

filename=forward+"responce"

file=open(filename,"at")

file.write(exp)

file.write("\n")

file.close()

if(exp==level1[i]["answer"]):

exp=""

setans.set(exp)

point+=5

i+=1

gameplay+=1

rajat.destroy()

else:

exp=""

setans.set(exp)

i+=1

gameplay+=1

rajat.destroy()

file=open(forward,"rt")

data=file.readlines()

file.close()

data[3]=str(point)+"\n"

data[4]=str(i)

file=open(forward,"wt")

file.writelines(data)

file.close()

game(forward)

**#Function to clear answer box**

def clear():

global exp

exp=""

setans.set(exp)

bt1=Button(rajat,text=level1[i]["option"][1],height="4",bg="#5DADE2",cursor="hand2",fg="#CB4335",font=("Berlin Sans FB","14"),width="10",bd=0,command=lambda: add(level1[i]["option"][1]))

bt2=Button(rajat,text=level1[i]["option"][2],height="4",bg="#5DADE2",fg="#CB4335",cursor="hand2",font=("Berlin Sans FB","14"),width="10",bd=0,command=lambda: add(level1[i]["option"][2]))

bt3=Button(rajat,text=level1[i]["option"][3],height="4",bg="#5DADE2",fg="#CB4335",cursor="hand2",font=("Berlin Sans FB","14"),width="10",bd=0,command=lambda: add(level1[i]["option"][3]))

bt4=Button(rajat,text=level1[i]["option"][4],height="4",bg="#5DADE2",fg="#CB4335",cursor="hand2",font=("Berlin Sans FB","14"),width="10",bd=0,command=lambda: add(level1[i]["option"][4]))

bt1.grid(row=3,column=1,padx=1,sticky="W")

bt2.grid(row=3,column=2,padx=1,sticky="W")

bt3.grid(row=3,column=3,padx=1,sticky="W")

bt4.grid(row=3,column=4,padx=1,sticky="W")

if(i>=5):

bt5=Button(rajat,text=level1[i]["option"][5],height="4",bg="#5DADE2",cursor="hand2",fg="#CB4335",font=("Berlin Sans FB","14"),width="10",bd=0,command=lambda: add(level1[i]["option"][5]))

bt5.grid(row=3,column=5,padx=1,sticky="W")

if(i>=10):

bt6=Button(rajat,text=level1[i]["option"][6],height="4",bg="#5DADE2",cursor="hand2",fg="#CB4335",font=("Berlin Sans FB","14"),width="10",bd=0,command=lambda: add(level1[i]["option"][6]))

bt6.grid(row=3,column=6,padx=1,sticky="W")

chk=Button(rajat,text="Submit",font=("Cooper Black","12"),cursor="hand2",command=check,bd=0,height="2",bg="#E74C3C",fg="White")

chk.grid(row=10,column=4,sticky="W",padx=2,ipadx=3,ipady=2)

ans=Label(rajat,text="Answer:",font=("Britannic Bold","18"),bg="#F1C40F")

ans.grid(row=5,column=0,sticky="E")

clr=Button(rajat,text="Clear",font=("Cooper Black","12"),cursor="hand2",command=clear,bd=0,height="2",bg="#E74C3C",fg="White")

clr.grid(row=10,column=3,sticky="W",padx=2,ipadx=3,ipady=2)

if(gameplay==0):

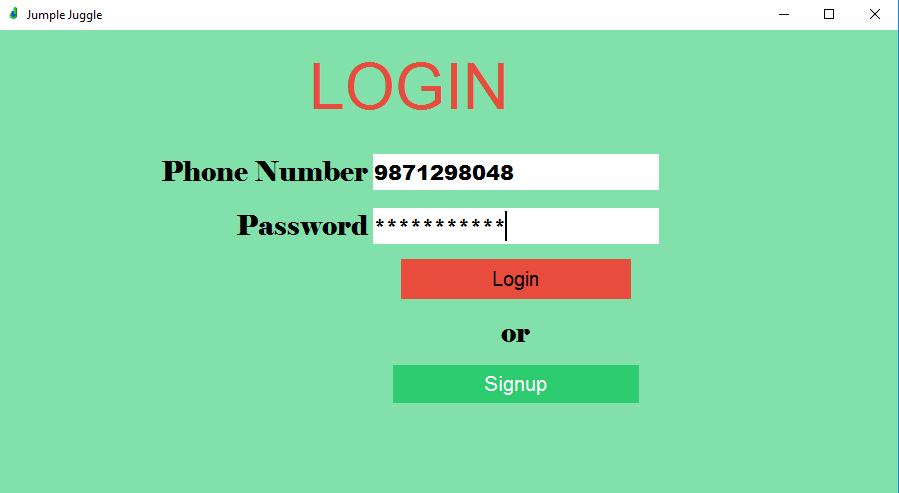
intro()

rajat.mainloop()

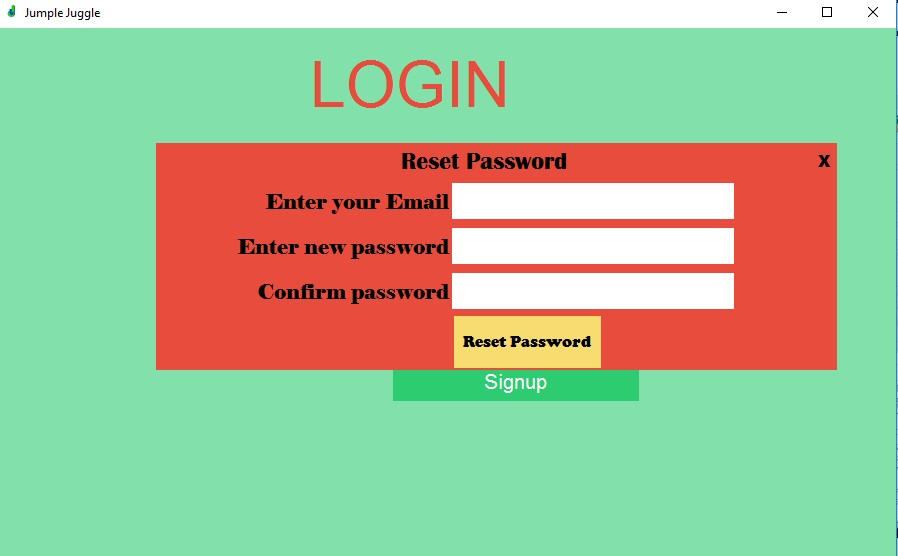
main() **#The final function call to start program**

**CHAPTER 3**

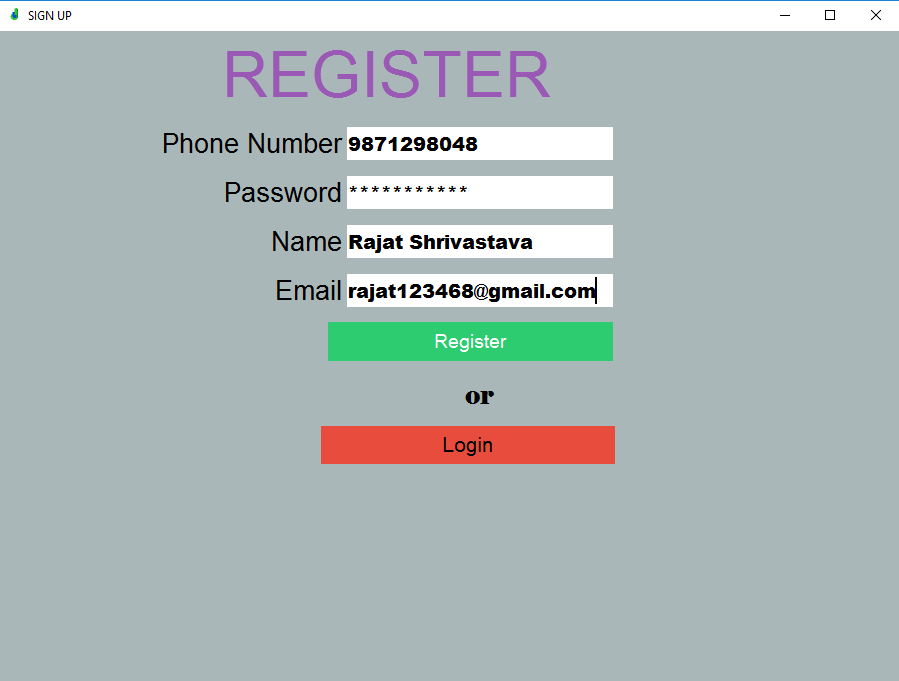
**Snapshots / Output of Interfaces**



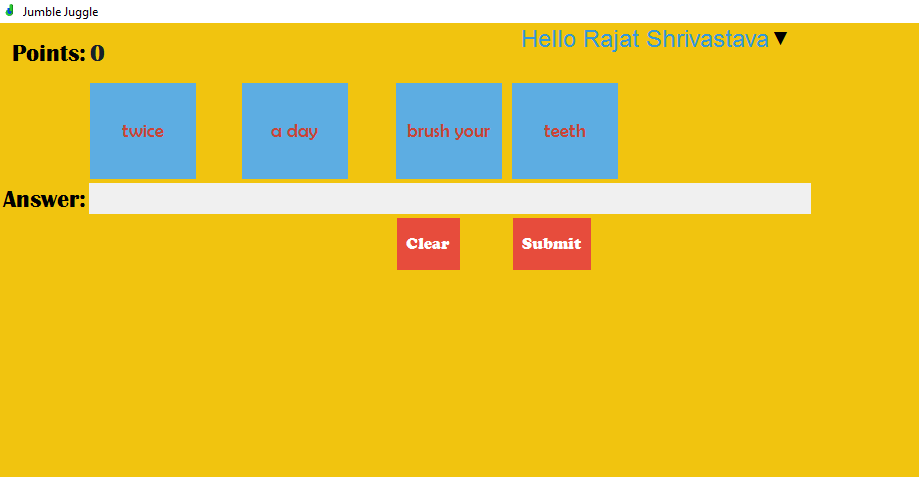
**Fig.1 The Login Page**



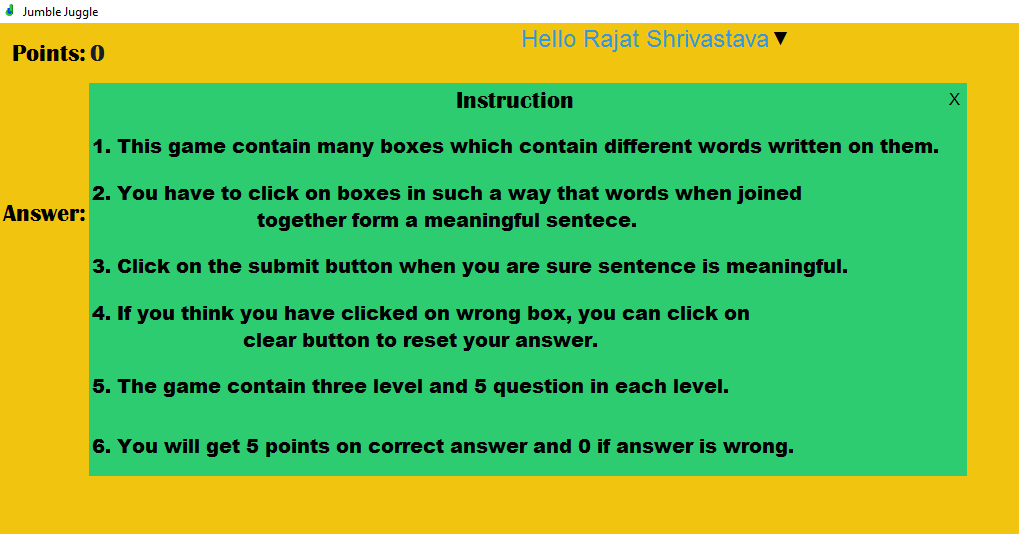
**Fig.2 Reset Password**



**Fig.3 The SignUp Page**

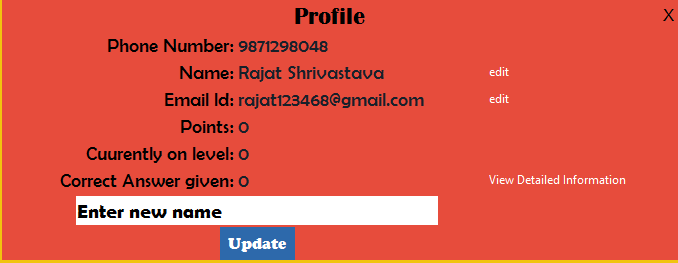


**Fig.4 The Game Interface**

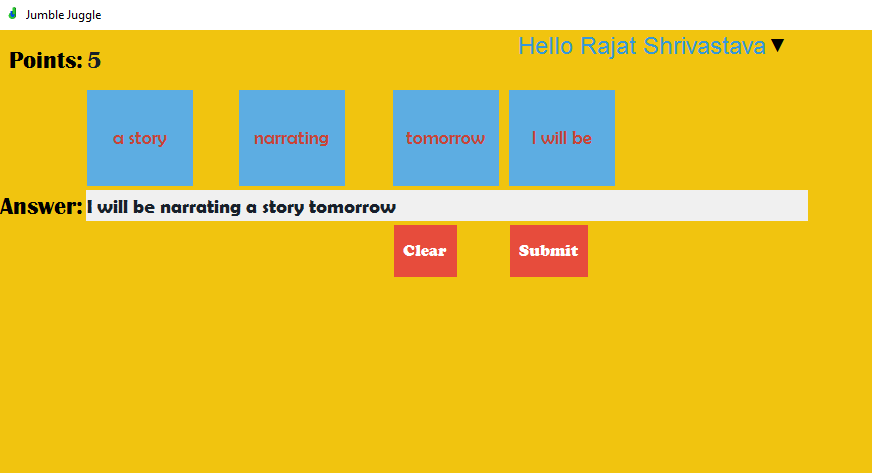
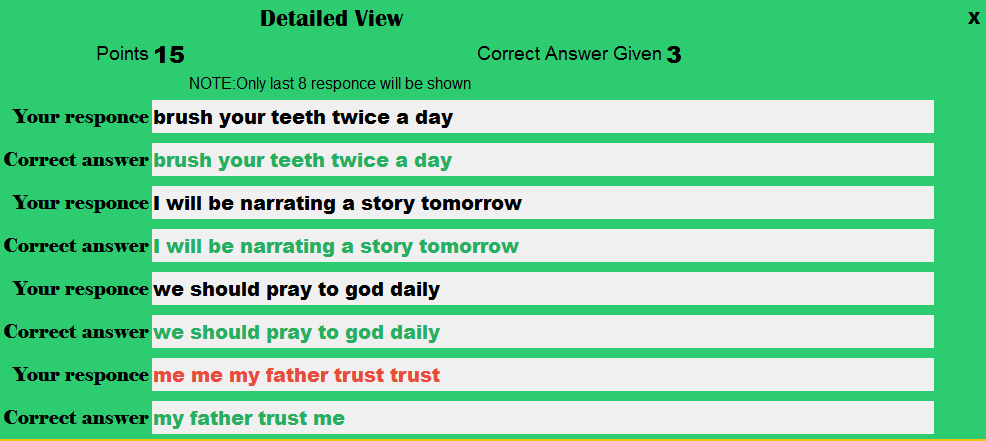
**Fig.5 The Instruction Panel**

****

**Fig.6 The Profile Panel**

****

**Fig.7 Editing Name in Profile Panel**

**Fig.8 Gameplay in progress**

**Fig.9 The Answer view Panel**

**Fig.10 The change password Panel**

**CHAPTER 4**

**CONCLUSION**

In this section, I summarize the experience gained by me during the development of this project. Writing a code for a game project is completely a new experience to me. It helps me to grow my Creative thinking and imagination capability.

The future plan to extend this project is to add more number of levels, introducing more features to it, and to make it more intresting by adding some instant replies when submit button is clicked.

At last, I want to say that I learned a lot through this project and it sharpen the concepts of functions as I extensively used functions in it.

**References**

1. <https://www.tutorialspoint.com/python/python_gui_programming.htm>
2. <https://www.geeksforgeeks.org/python-gui-tkinter/>
3. <https://www.studyadda.com/notes/5th-class/english/jumbled-words-and-sentences/jumbled-sentences-and-words/14438>
4. <http://perfectyourenglish.com/blog/rearrange-words-to-make-meaningful-sentences-class-4-grammar-worksheets/>
5. <https://www.lucidchart.com/pages/>