#######################################################################################

#Importing

from tkinter import \*

from tkinter import ttk , messagebox as msg

from threading import Thread

from time import sleep

from mysql import connector as ms

import tkinter as tk

import pygame

import random

import time

import os

from datetime import datetime

from PIL import Image , ImageTk

#######################################################################################

root = Tk()

root.resizable(width = False , height = False)

root.wm\_state('zoomed')

root.title("Appstore By AVIONICS and RRAJJ")

frame = Frame(root, bg='grey')

#######################################################################################

root.iconbitmap('Photos/Icon.ico')

back\_lg1 = PhotoImage(file = r'Photos/back small.png')

home\_photo = PhotoImage(file = r'Photos/home small.png')

appstore\_lg2 = PhotoImage(file = r'Photos/Appstore Small.png')

front\_photo\_variable = PhotoImage(file = r'Photos/applicationmanagement.png')

person1 = Image.open('Photos/person\_1.jpeg')

person2 = Image.open('Photos/person\_2.jpg')

person\_1 = person1.resize((300 , 300) , Image.ANTIALIAS)

person\_2 = person2.resize((300 , 300) , Image.ANTIALIAS)

person1\_photo = ImageTk.PhotoImage (person\_1)

person2\_photo = ImageTk.PhotoImage(person\_2)

pas = ''

#######################################################################################

##Classes

class Calculator(Toplevel):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.geometry('300x350')

self.title("Calculator")

self.iconbitmap('Photos/calculator\_icon.ico')

self.bgcolor = '#c5c5c5'

self.config(bg = self.bgcolor)

self.resizable(width = False , height = False)

self.scvalue = tk.StringVar()

self.scvalue.set("0")

self.screen = tk.Entry(self , text = self.scvalue , font = 'helvatica 19 bold' ,relief = 'sunken' , width = 300)

self.screen.pack(pady = 6 , padx = 3 , ipadx = 9 , ipady = 9)

self.protocol("WM\_DELETE\_WINDOW" , self.close)

#This save variable is to check wether the last thing done was root , equal , int or log

#If yes then we have to clear the screen of entry for the next thing to be entered

self.save = bool #Used in the click function

self.create\_buttons()

def close(self):

self.destroy()

global root

root.wm\_state('zoomed')

#Creating buttons

def create\_buttons(self):

self.lst = [['<--' , 'CE', 'C' , 'root' , 'log'] , ['7' , '8' , '9' , '/' , '%'] , ['4', '5' , '6' , '\*' , '1/x'], ['1' , '2' , '3' , '-' , '='] , ['0' , '.' , '+' , 'int']]

tk.Label(self , text = '' , bg = self.bgcolor).pack(pady = 4)

for i in self.lst :

self.frame = tk.Frame(self , bg = self.bgcolor)

for j in i :

if j == '0' :

self.button = tk.Button(self.frame , text = j , width = 10 , height = 2 , relief = 'raised')

self.button.pack(side = 'left' , anchor = 'nw' , pady = 5 , padx = 5 , ipadx = 10)

self.button.bind("<Button-1>" , self.click)

else :

self.button = tk.Button(self.frame , text = j , width = 5 , height = 2 , relief = 'raised')

self.button.pack(side = 'left' , anchor = 'nw' , pady = 5 , padx = 5)

self.button.bind("<Button-1>" , self.click)

self.frame.pack()

#If the given string to this fucntion is digit then it will return True else False

def digit(self , string):

if string.isdigit() :

return True

try :

float(string)

return True

except :

return False

#All the click events here

def click(self , event):

self.text = event.widget.cget("text")

if self.scvalue.get() == '0' or self.scvalue.get() == "Something went wrong" or self.save == True:

self.save = False

self.scvalue.set("")

self.screen.update()

self.result()

else :

self.result()

#Change in the entry widget with the result output is done here

def result(self) :

if self.text == 'root' :

self.save = True

from math import sqrt

try :

self.scvalue.set(sqrt(float(self.scvalue.get())))

except :

self.scvalue.set("Something went wrong")

self.screen.update()

elif self.text == 'int' :

self.save = True

self.scvalue.set(int(float(self.scvalue.get())))

self.screen.update()

elif self.text == 'log' :

self.save = True

from math import log

try :

self.scvalue.set(log(float(self.scvalue.get())))

except :

self.scvalue.set("Something went wrong")

elif self.text == '=' :

self.save = True

try :

self.scvalue.set(eval(self.scvalue.get()))

self.screen.update()

except :

self.scvalue.set("Something went wrong")

self.screen.update()

#To clear the screen

elif self.text == 'C' or self.text == 'CE':

self.scvalue.set("0")

self.screen.update()

#Using this as backspace

elif self.text == '<--' :

val = self.scvalue.get()

if len(val) == 1 or len(val) == 0:

self.scvalue.set('0')

self.screen.update()

else :

res = ''

for i in range(len(val) - 1):

res += val[i]

self.scvalue.set(res)

self.screen.update()

else :

self.scvalue.set(str(self.scvalue.get()) + self.text)

self.screen.update()

#######################################################################################

class Game(Toplevel):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.title("Tic-Tac-Toe -App Store")

self.resizable(width = False , height = False)

self.click = True

self.count = 0

self.reset\_bool = False

self.winner = False

self.protocol("WM\_DELETE\_WINDOW" , self.close)

self.menus()

self.create\_buttons()

def close(self):

self.destroy()

global root

root.wm\_state('zoomed')

def create\_buttons(self):

self.new\_frame = tk.Frame(self)

if self.reset\_bool == True:

self.click = True

self.reset\_bool = False

#Creating buttons

self.b1 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b1))

self.b2 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b2))

self.b3 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b3))

self.b4 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b4))

self.b5 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b5))

self.b6 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b6))

self.b7 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b7))

self.b8 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b8))

self.b9 = tk.Button(self.new\_frame , text = ' ' , font = 'helvatica 13' , height = 3 , width =6 , command = lambda : self.clicked(self.b9))

#Gridding

self.new\_frame.grid()

self.b1.grid(row = 0 , column = 0)

self.b2.grid(row = 0 , column = 1)

self.b3.grid(row = 0 , column = 2)

self.b4.grid(row = 1 , column = 0)

self.b5.grid(row = 1 , column = 1)

self.b6.grid(row = 1 , column = 2)

self.b7.grid(row = 2 , column = 0)

self.b8.grid(row = 2 , column = 1)

self.b9.grid(row = 2 , column = 2)

def reset(self):

self.reset\_bool = True

self.count = 0

self.winner = False

self.new\_frame.destroy()

self.create\_buttons()

def menus(self):

self.Mainmenu = tk.Menu(self)

self.menu = tk.Menu(self.Mainmenu , tearoff = False)

self.menu.add\_command(label = 'Reset game' , command = self.reset)

self.Mainmenu.add\_cascade(label = 'Options' , menu = self.menu)

self.config(menu = self.Mainmenu)

#If the button is clicked then this will run

def clicked(self , b):

if b['text'] == ' ' and self.click == True:

b.config(text = 'X')

self.click = False

self.count += 1

if self.count >= 5:

self.check\_won("X")

elif b['text'] == ' ' and self.click == False:

b.config(text = 'O')

self.click = True

self.count += 1

if self.count >= 5:

self.check\_won("O")

else :

self.disable()

if msg.showerror(title = "Tic-Tac-Toe -App Store" , message= f'That box is already taken by {b["text"]}\nPlease click a box that is not clicked') == 'ok' :

self.enable()

def check\_won(self , who\_clicked):

global root

if self.b1['text'] == who\_clicked and self.b2['text'] == who\_clicked and self.b3['text'] == who\_clicked :

self.b2.config(bg = 'red')

self.b1.config(bg = 'red')

self.b3.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

elif self.b4['text'] == who\_clicked and self.b5['text'] == who\_clicked and self.b6['text'] == who\_clicked :

self.b4.config(bg = 'red')

self.b5.config(bg = 'red')

self.b6.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

elif self.b7['text'] == who\_clicked and self.b8['text'] == who\_clicked and self.b9['text'] == who\_clicked :

self.b7.config(bg = 'red')

self.b8.config(bg = 'red')

self.b9.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

elif self.b1['text'] == who\_clicked and self.b4['text'] == who\_clicked and self.b7['text'] == who\_clicked :

self.b4.config(bg = 'red')

self.b1.config(bg = 'red')

self.b7.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

elif self.b5['text'] == who\_clicked and self.b2['text'] == who\_clicked and self.b8['text'] == who\_clicked :

self.b2.config(bg = 'red')

self.b5.config(bg = 'red')

self.b8.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

elif self.b3['text'] == who\_clicked and self.b6['text'] == who\_clicked and self.b9['text'] == who\_clicked :

self.b3.config(bg = 'red')

self.b6.config(bg = 'red')

self.b9.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

elif self.b1['text'] == who\_clicked and self.b5['text'] == who\_clicked and self.b9['text'] == who\_clicked :

self.b1.config(bg = 'red')

self.b5.config(bg = 'red')

self.b9.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

elif self.b5['text'] == who\_clicked and self.b3['text'] == who\_clicked and self.b7['text'] == who\_clicked :

self.b3.config(bg = 'red')

self.b5.config(bg = 'red')

self.b7.config(bg = 'red')

self.disable()

msg.showinfo(title= "Tic-Tac-Toe -App Store" , message= f'CONGRATULATIONS!! \n{who\_clicked} Won!!!')

self.winner = True

if self.winner == False and self.count == 9 :

yesno = msg.askyesno(title = "Tic-Tac-Toe -App Store" , message= "The game ends with a tie\n Do you wan't to restart the game??")

self.disable()

if yesno :

self.enable()

self.reset()

else :

root.wm\_state('zoomed')

self.destroy()

if self.winner == True :

if msg.askyesno(title = "Tic-Tac-Toe -App Store" , message= "Do you want to try it again") == True :

self.reset()

self.enable()

else :

root.wm\_state('zoomed')

self.destroy()

def enable(self):

for but in self.new\_frame.grid\_slaves():

but['state'] = tk.NORMAL

def disable(self):

for but in self.new\_frame.grid\_slaves():

if type(but) == Button:

but.config(state = DISABLED)

#######################################################################################

class Snake():

def \_\_init\_\_(self):

pygame.mixer.init()

pygame.mixer.music.load('Game/Faded.mp3')

pygame.mixer.music.play(-1)

pygame.init()

#Colors

self.white = (225 ,225 ,225)

self.dark\_red = (200 , 0 , 0)

self.red = (225 , 0 ,0)

self.black = (0, 0 , 0)

self.green = (0 , 128 ,0)

self.light\_green = (0 , 225 ,0)

self.grey = (128 , 128 ,128)

self.blue = (0 ,0 , 225)

self.pink = (225, 0 , 225)

self.yellow = (200, 200 ,0)

#Global variables

self.game\_width = 600

self.game\_height = 600

self.game\_window = pygame.display.set\_mode((self.game\_width , self.game\_height))

pygame.display.set\_caption('Snake With Roshan')

self.clock = pygame.time.Clock()

self.count = 0

self.score\_lst = []

self.fps = 30

self.esc\_exit = False

self.Ones = ['','First','Second','Third','Fourth','Fifth','Sixth','Seventh','Eighth','Ninth']

self.Tens = ['Tenth','Eleventh','Twelveth','Thirteenth','Fourteenth','Fifteenth','Sixteenth','Seventeenth','Eigtheenth','Nineteenth']

self.Multiple\_of\_ten = ['','','Twenty','Thirty','Fourty','Fifty','Sixty','Seventy','Eigthy','Ninty']

self.Power\_of\_ten = ['','Hundred','Thousand','Lakh','Crore']

#self.WelcomeScreen()

#Funtions

def ExitScreen(self):

pygame.mixer.music.load('Game/Ahrix.mp3')

pygame.mixer.music.play(-1)

exit\_game = False

space = True

while not exit\_game:

self.game\_window.fill(self.white)

bgimg = pygame.image.load('Game/ExitScreen.jpg')

bgimg = pygame.transform.scale(bgimg , (600 , 600)).convert\_alpha()

self.game\_window.blit(bgimg , (0 , 0))

self.text\_screen(self.game\_window , "Made by Roshan Raj" , self.pink , 400, 580 , 25)

if self.esc\_exit :

if self.count-1 == 0 :

self.text\_screen(self.game\_window , "You exited the game in your running first game" , self.red , 100 , 260 , 30)

elif self.count-1 == 1 :

self.text\_screen(self.game\_window , "You played one game" , self.black , 200 , 200 , 30 )

self.text\_screen(self.game\_window , "Score :" + str(self.score\_lst[0]) , self.black , 200 , 230 , 30 )

self.text\_screen(self.game\_window , "You exited the game in your running second game" , self.red , 100 , 260 , 30)

else :

self.text\_screen(self.game\_window , "You played " + str(self.count-1) + " games" , self.yellow , 5, 5 , 30)

self.text\_screen(self.game\_window , "Your scores are the following :" , self.yellow , 5 , 30 , 30)

i = 0

while i < len(self.score\_lst) :

self.text\_screen(self.game\_window , self.numbers\_to\_words(i+1) +": " + str(self.score\_lst[i]) , self.yellow , 5 , 60 + (i\* 30), 30)

i += 1

self.text\_screen(self.game\_window , "You exited the game in your running\n" + self.numbers\_to\_words(i+1) + " game" , self.yellow , 5 , 60 + (i\* 30) , 30)

else :

if self.count == 0 :

self.text\_screen(self.game\_window , "You haven't tried the game" , self.black , 190, 200 , 30)

self.text\_screen(self.game\_window , "You should try it" , self.black , 200, 230 , 30)

elif self.count == 1 :

self.text\_screen(self.game\_window , "You played one game" , self.black , 200 , 200 , 30 )

self.text\_screen(self.game\_window , "Score :" + str(self.score\_lst[0]) , self.black , 200 , 230 , 30 )

else :

self.text\_screen(self.game\_window , "You played " + str(self.count) + " games" , self.yellow , 5, 5 , 30)

self.text\_screen(self.game\_window , "Your scores are the following :" , self.yellow , 5 , 30 , 30)

for i in range(len(self.score\_lst)) :

self.text\_screen(self.game\_window , self.numbers\_to\_words(i+1) +": " + str(self.score\_lst[i]) , self.yellow , 5 , 60 + (i\* 30), 30)

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.QUIT :

exit\_game = True

break

elif event.type == pygame.KEYDOWN :

if event.key == pygame.K\_SPACE and space:

pygame.mixer.music.pause()

space = False

elif event.key == pygame.K\_SPACE and not space :

pygame.mixer.music.unpause()

space = True

if event.key == pygame.K\_ESCAPE :

exit\_game = True

break

elif event.key == pygame.K\_r:

self.esc\_exit = False

self.GameLoop()

elif event.key == pygame.K\_s :

self.shortcuts()

pygame.quit()

exit()

def GameLoop(self) :

pygame.mixer.music.load('Game/back.mp3')

pygame.mixer.music.play(-1)

#Variables

self.count +=1

game\_over = False

exit\_game = False

snake\_x = 295

snake\_y = 295

snake\_size = 10

velocity = 5

velocity\_x = 0

velocity\_y = 0

food\_size = snake\_size

food\_x = random.randint(0 + snake\_size , self.game\_width - snake\_size)

food\_y = random.randint(0 + snake\_size , self.game\_height - snake\_size)

big\_food\_size = 15

big\_food\_x = random.randint(40 + snake\_size ,self.game\_width - snake\_size - 30)

big\_food\_y = random.randint(60 , self.game\_width - snake\_size - 30)

score = 0

snake\_length = 1

snake\_list = []

last\_key = 0

big\_count = False

space = True

start\_time = 0

end\_time = 0

if not os.path.exists('Game/HighScore.txt') :

with open('Game/HighScore.txt' , 'w') as f :

f.write("0")

with open("Game/HighScore.txt" , 'r') as f :

HighScore = f.read()

#######################################################################################

#GameLoop

while not exit\_game :

if game\_over :

with open('Game/HighScore.txt' , 'w') as f :

f.write(str(HighScore))

self.game\_window.fill(self.white)

game\_bgimg = pygame.image.load('Game/GameOver.jpg')

game\_bgimg = pygame.transform.scale(game\_bgimg , (600 , 600)).convert\_alpha()

self.game\_window.blit(game\_bgimg , (0 , 0))

self.text\_screen(self.game\_window , "Game Over! press enter to restart" , self.red ,135 , self.game\_height/2-60 , 35)

self.text\_screen(self.game\_window , "Esc to exit the game" , self.red ,180 , self.game\_height/2-25 , 35)

self.text\_screen(self.game\_window , "Made by Roshan Raj" , self.pink , 400, 580 , 25)

for event in pygame.event.get():

if event.type == pygame.QUIT :

self.ExitScreen()

elif event.type == pygame.KEYDOWN :

if event.key == pygame.K\_SPACE and space:

pygame.mixer.music.pause()

space = False

elif event.key == pygame.K\_SPACE and not space :

pygame.mixer.music.unpause()

space = True

if event.key == pygame.K\_RETURN or event.type == pygame.K\_r :

self.GameLoop()

elif event.key == pygame.K\_ESCAPE:

self.ExitScreen()

elif event.key == pygame.K\_s :

self.shortcuts()

else :

for event in pygame.event.get():

if event.type == pygame.QUIT :

exit\_game = True

self.esc\_exit = True

self.ExitScreen()

elif event.type == pygame.MOUSEBUTTONDOWN :

if event.pos[0] <= 50 and event.pos[1] <= 50:

self.shortcuts()

elif (event.pos[0] >= 60 and event.pos[0] <= 110) and event.pos[1] <= 50:

unpause = False

while not unpause :

for i in pygame.event.get():

if i.type == pygame.QUIT :

unpause = True

exit\_game = True

break

elif (i.type == pygame.KEYDOWN and i.key == pygame.K\_p):

unpause = True

break

elif i.type == pygame.MOUSEBUTTONDOWN and ((i.pos[0] >= 120 and i.pos[0]) and i.pos[1]<= 50):

unpause = True

break

elif event.type == pygame.KEYDOWN :

if event.key == pygame.K\_SPACE and space:

pygame.mixer.music.pause()

space = False

elif event.key == pygame.K\_SPACE and not space :

pygame.mixer.music.unpause()

space = True

if event.key == pygame.K\_ESCAPE :

exit\_game = True

self.esc\_exit = True

self.ExitScreen()

elif event.key == pygame.K\_s:

self.shortcuts()

elif event.key == pygame.K\_r :

self.GameLoop()

elif event.key == pygame.K\_p :

unpause = False

while not unpause :

for i in pygame.event.get():

if i.type == pygame.QUIT :

unpause = True

exit\_game = True

break

elif (i.type == pygame.KEYDOWN and i.key == pygame.K\_p):

unpause = True

break

elif i.type == pygame.MOUSEBUTTONDOWN and ((i.pos[0] >= 120 and i.pos[0]) and i.pos[1]<= 50):

unpause = True

break

elif event.key == pygame.K\_b :

pygame.quit()

quit()

elif event.key == pygame.K\_RIGHT :

if last\_key == pygame.K\_LEFT :

continue

velocity\_x = velocity

velocity\_y = 0

last\_key = pygame.K\_RIGHT

elif event.key == pygame.K\_DOWN :

if last\_key == pygame.K\_UP :

continue

velocity\_y = velocity

velocity\_x = 0

last\_key = pygame.K\_DOWN

elif event.key == pygame.K\_UP :

if last\_key == pygame.K\_DOWN :

continue

velocity\_y = -velocity

velocity\_x = 0

last\_key = pygame.K\_UP

elif event.key == pygame.K\_LEFT :

if last\_key == pygame.K\_RIGHT :

continue

velocity\_x = -velocity

velocity\_y = 0

last\_key = pygame.K\_LEFT

snake\_x += velocity\_x

snake\_y += velocity\_y

if abs(food\_x - snake\_x) < 7 and abs(food\_y - snake\_y) < 7 :

beepSound = pygame.mixer.Sound('Game/beep.wav')

beepSound.play()

score += 1

velocity += 0.2

big\_count = True

food\_x = random.randint(0 + 5 \*snake\_size ,self.game\_width - 5 \*snake\_size)

food\_y = random.randint(60 , self.game\_height - 5 \*snake\_size)

snake\_length += 1

if score > int(HighScore) :

HighScore = score

if score %5 == 0:

start\_time = time.time()

i = 1

#Setting logo

self.game\_window.fill(self.grey)

setting = pygame.image.load('Game/setting.png')

setting = pygame.transform.scale(setting , (50 , 50)).convert\_alpha()

self.game\_window.blit(setting , (0 , 0))

#Pause logo

pause = pygame.image.load('Game/pause.png')

pause = pygame.transform.scale(pause , (50 , 50)).convert\_alpha()

self.game\_window.blit(pause , (60 , 0))

#Resume logo

resume = pygame.image.load('Game/resume.png')

resume = pygame.transform.scale(resume , (50 , 50)).convert\_alpha()

self.game\_window.blit(resume , (120 , 0))

#game\_bgimg = pygame.image.load('snake.png')

#game\_bgimg = pygame.transform.scale(game\_bgimg , (600 , 600)).convert\_alpha()

#self.game\_window.blit(game\_bgimg , (0 , 0))

if score%5 == 0 and big\_count:

pygame.draw.circle(self.game\_window , self.red , [big\_food\_x ,big\_food\_y] , big\_food\_size)

if abs(big\_food\_x - snake\_x) < 12 and abs(big\_food\_y - snake\_y) < 12:

big\_count = False

beepSound = pygame.mixer.Sound('Game/beep\_big.wav')

beepSound.play()

score += 5

big\_food\_x = random.randint(0 + 5 \*snake\_size , self.game\_width - 5 \*snake\_size)

big\_food\_y = random.randint(60 , self.game\_height - 5 \*snake\_size)

snake\_length += 1

if score > int(HighScore) :

HighScore = score

end\_time = time.time()

if i < 2 :

self.text\_screen(self.game\_window , "Timer : " + str(4 -i) , self.green , 5 , 560 , 35)

elif i <3:

self.text\_screen(self.game\_window , "Timer : " + str(4 -i) , self.yellow , 5 , 560 , 35)

elif i < 4:

self.text\_screen(self.game\_window , "Timer : " + str(4 -i) , self.red , 5 , 560 , 35)

if (end\_time - start\_time) >= i :

i += 1

if (end\_time - start\_time) >= 4:

big\_count = False

head = []

head.append(snake\_x)

head.append(snake\_y)

snake\_list.append(head)

if len(snake\_list) > snake\_length :

del snake\_list[0]

if snake\_x>(self.game\_width - snake\_size) or snake\_y>(self.game\_height - snake\_size) or snake\_x<(snake\_size/2) or snake\_y<(snake\_size/2):

game\_over = True

self.score\_lst.append(score)

pygame.mixer.music.load('Game/Astronomia.mp3')

pygame.mixer.music.play(-1)

elif head in snake\_list[:-1] :

game\_over = True

self.score\_lst.append(score)

pygame.mixer.music.load('Game/adhi.mp3')

pygame.mixer.music.play(-1)

self.text\_screen(self.game\_window , "Score : " +str(score) , self.blue , 260 ,5 , 35)

self.text\_screen(self.game\_window , "High Score : " + str(HighScore) , self.blue , 400,5 , 35)

self.text\_screen(self.game\_window , "Made by Roshan Raj" , self.pink , 400, 580 , 25)

pygame.draw.rect(self.game\_window , self.red , [food\_x , food\_y , food\_size , food\_size])

self.plot\_snake(self.game\_window , self.black , snake\_list , snake\_size)

pygame.display.update()

self.clock.tick(self.fps)

pygame.quit()

def numbers\_to\_words(self ,n):

s=0

w=''

while n>0:

if s==1:

r=n%10

if r!=0:

w = self.Ones[r] + ' Hundred ' + w

n=n//10

else:

r=n%100

x=r%10

m=r//10

if m==1:

w = self.Tens[x] + ' ' + self.Power\_of\_ten[s] + ' ' + w

elif m==0:

w= self.Ones[x] + ' ' + self.Power\_of\_ten[s] + ' ' + w

else:

w =self.Multiple\_of\_ten[m] + ' ' + self.Ones[x] + ' ' + self.Power\_of\_ten[s] + ' ' + w

n=n//100

s+=1

return w

def plot\_snake(self , game\_window , color , snake\_list , snake\_size):

for x , y in snake\_list :

pygame.draw.rect(game\_window , color , [x , y , snake\_size, snake\_size])

x , y = snake\_list[-1][0] , snake\_list[-1][1]

pygame.draw.rect(game\_window , self.light\_green , [x , y , snake\_size, snake\_size])

def shortcuts(self):

exit\_game = False

while not exit\_game :

self.game\_window.fill(self.white)

self.text\_screen(self.game\_window , "" , self.black , 200 , 200 , 30)

for event in pygame.event.get():

if event.type == pygame.QUIT :

exit\_game = True

elif event.type == pygame.KEYDOWN :

if event.key == pygame.K\_ESCAPE or event.key == pygame.K\_s or event.key == pygame.K\_RETURN :

exit\_game = True

self.text\_screen(self.game\_window , "Game Shortcuts" ,self.black , 5 ,5 , 40)

self.text\_screen(self.game\_window , "P : Pause/Unpause Game" , self.grey ,5 ,50 , 30)

self.text\_screen(self.game\_window , "Spacebar : Pause/Unpause Music" , self.grey ,5 ,80 , 30)

self.text\_screen(self.game\_window , "R : Restart Game" , self.grey ,5 ,110 , 30)

self.text\_screen(self.game\_window , "S : Shortcuts" , self.grey ,5 ,140 , 30)

self.text\_screen(self.game\_window , "Esc : Exit Game" , self.grey ,5 ,170 , 30)

pygame.display.update()

def text\_screen(self , game\_window , text , color , x , y , size) :

font = pygame.font.SysFont(None , size)

screen\_text = font.render(text , True , color)

self.game\_window.blit(screen\_text , [x , y])

def WelcomeScreen(self):

exit\_game = False

while not exit\_game:

self.game\_window.fill((233,210,229))

bgimg = pygame.image.load('Game/Welcome.jpg')

bgimg = pygame.transform.scale(bgimg , (600 , 600)).convert\_alpha()

self.game\_window.blit(bgimg , (0 , 0))

self.text\_screen(self.game\_window , "Welcome to snakes" , self.black , 200 , 250 , 30)

self.text\_screen(self.game\_window , "Press SpaceBar to Play" , self.black , 180 , 275 , 30)

self.text\_screen(self.game\_window , "Made by Roshan Raj" , self.pink , 10, 580 , 25)

for event in pygame.event.get():

if event.type == pygame.QUIT :

exit\_game = True

if event.type == pygame.MOUSEBUTTONDOWN:

self.GameLoop()

if event.type == pygame.KEYDOWN :

if event.key == pygame.K\_SPACE :

self.GameLoop()

elif event.key == pygame.K\_ESCAPE :

exit\_game = True

elif event.key == pygame.K\_s :

self.shortcuts()

pygame.display.update()

self.clock.tick(self.fps)

self.ExitScreen()

#######################################################################################

#Functions

def clock():

global labl

cur = datetime.now()

hour = str(cur.hour)

minute = str(cur.minute)

second = str(cur.second)

labl.config(text = hour + ':' + minute + ':' + second)

labl.after(1000 , clock)

def close():

yesno = msg.askyesno(title= "Exit",message= 'Are you sure?')

if yesno:

root.destroy()

def gridding(): #Anything to be added in body\_right will be done here

global body\_right , labl

homepage\_frame = Frame(body\_right, height=100, width=100, bg='#606060', highlightbackground="black", highlightthickness=1)

homepage\_title = Label(homepage\_frame, text='ATOMIC ENERGY CENTRAL SCHOOL - 1, JADUGODA', bg='#00b050', font='Bebas 40', height= 2, fg='White')

homepage\_title.grid(row=0, column=0, columnspan=3, sticky='NEW', padx= 10, pady=10)

homepage\_subtitle1 = Label(homepage\_frame, text='COMPUTER PROJECT', bg='#92D050', font='Bebas 25', height= 2)

homepage\_subtitle1.grid(row=1, column=0, columnspan=3, sticky='NEW', padx= 10)

BOX1 = Frame(homepage\_frame, bg='#606060')

BOX1.grid(row=2, column=0, sticky='NEWS', padx=10, pady=10)

homepage\_frame.rowconfigure(2, weight=1)

BOX1.columnconfigure(0, weight=1)

project\_name\_label = Label(BOX1, text='Project Name',bg='#262626', fg='white', height=3, font='Roboto') ;project\_name\_label.grid(row=0, sticky='WE', pady=5)

name\_of\_student\_label = Label(BOX1, text='Name of the Students',bg='#262626', fg='white', height=3, font='Roboto'); name\_of\_student\_label.grid(row=1, sticky='WE', pady=5)

name\_of\_teacher\_label = Label(BOX1, text='Name of the Teacher',bg='#262626', fg='white', height=3, font='Roboto'); name\_of\_teacher\_label.grid(row=2, sticky='WE', pady=5)

BOX2 = Frame(homepage\_frame, bg='#606060')

BOX2.grid(row=2, column=1, sticky='NEWS', padx=10, pady=10, columnspan=2)

BOX2.columnconfigure(1, weight=1)

BOX2.columnconfigure(2, weight=1)

project\_name= Label(BOX2, text='Application Management', fg='white', height=3, bg='#262626', font='Roboto') ; project\_name.grid(row=0, columnspan=3, sticky='NEWS', pady=5)

name\_of\_student= Label(BOX2, text='Avitesh Murmu \n Roshan Raj', fg='white', height=3, bg='#262626', font='Roboto') ; name\_of\_student.grid(row=1, columnspan=3, sticky='NEWS', pady=5)

name\_of\_teacher= Label(BOX2, text='S. K. Mukherjee', fg='white', height=3, bg='#262626', font='Roboto') ; name\_of\_teacher.grid(row=2, columnspan=3, sticky='NEWS', pady=5)

for i in range(3):

homepage\_frame.columnconfigure(i, weight=1)

homepage\_frame.grid(row = 2 , column = 0, sticky='EW', padx=35, pady=10)

body\_right.rowconfigure(2, weight=1)

def validate\_login(): #Login works here

global body\_right

mydb = ms.connect(host = 'localhost' , user = 'root' , passwd = pas , database = 'project')

mycursor = mydb.cursor()

def already\_exists():

global result\_label

usd = username.get()

pw = password.get()

mycursor = mydb.cursor()

mycursor.execute("Select \* from login")

res = mycursor.fetchall()

for i in range(len(res)) :

if res[i][0].lower() == usd.lower():

if res[i][1] == pw :

result\_label.config(text = 'Already an account available\nSignined to that id')

result\_label.grid(row =1000 , column = 0)

else :

if res[i][1].lower() == pw.lower():

result\_label.config(text = 'This Username already exists\nPassword is wrong\nYou can sign in')

result\_label.grid(row = 1000 , column = 0)

return False

return True

def threads():

try :

username.set("")

password.set("")

home\_command()

except :

pass

def logout():

global sign\_up , header , logined

logined = False

sign\_in.config(text = 'Sign In' , command = signin\_window, font='Roboto 9')

sign\_up =Button(header, text='Sign Up', command=signup\_window ,height= 3,width=7, bg='#385723', relief=GROOVE , fg='white', font='Roboto 9')

sign\_up.grid(column=4,row=0, sticky=(E), padx=10, pady=5)

header.columnconfigure(2, weight=1)

home\_command()

def login():

global mycursor , result\_label , logined

usd = username.get()

passWord = password.get()

mycursor = mydb.cursor()

mycursor.execute("Select \* from login")

res = mycursor.fetchall()

try :

result\_label.config()

except :

result\_label = Label(body\_right , font = 'Robot 20 bold' , bg = '#262626')

if valididty(usd , passWord) == False:

count = False

for i in range(len(res)) :

if res[i][0].lower() == usd.lower():

count = True

if res[i][1] == passWord :

result\_label.config(text = "Sign in done" , fg = '#28e84a')

result\_label.grid(row = 1000 , column = 0)

logined = True

sign\_up.destroy()

sign\_in.config(text = 'Logout' , command = logout)

else :

if res[i][1].lower() == passWord.lower():

result\_label.config(text = "Please check the lower case/upper case in the password" , fg = "#f1ec14")

result\_label.grid(row = 1000 , column = 0)

else :

result\_label.config(text = "Sorry wrong password" , fg = 'red',)

result\_label.grid(row = 1000 , column = 0)

break

if not count:

result\_label.config(text = f"You should first sign up\nNo id named : {usd}" , fg = '#70d1cc')

result\_label.grid(row = 1000 , column = 0)

def valididty(user , passwrd):

global result\_label

try :

result\_label.config()

except :

result\_label = Label(body\_right , font = 'Robot 20 bold' , bg = '#262626')

if (user == '' or user.isspace() or user.isalpha() == False) and (passwrd == '' or passwrd.isspace() or passwrd.isalpha() == False):

result\_label.config(text = "Sorry not a valid username and password" , fg = 'red') ; result\_label.grid(row = 1000 , column = 0)

elif user == '' or user.isspace() or user.isalpha() == False:

result\_label.config(text = "Sorry not a valid username" , fg = 'red') ; result\_label.grid(row = 1000 , column = 0)

elif passwrd == '' or passwrd.isspace() or passwrd.isalpha() == False:

result\_label.config(text = "Sorry not a valid password" , fg = 'red') ; result\_label.grid(row = 1000 , column = 0)

elif len(passwrd) < 8 and len(user) < 8 :

result\_label.config(text = 'Sorry not a valid Username and Password\nUsername Password too small\nMust have 8 letters' , fg = 'red')

result\_label.grid(row = 1000 , column = 0)

elif len(passwrd) < 8 :

result\_label.config(text = 'Sorry not a valid Password\nPassword too small \nMust have 8 letters' , fg = 'red')

result\_label.grid(row = 1000 , column = 0)

elif len(user) < 8 :

result\_label.config(text = 'Sorry not a valid Username\nUsername too small \nMust have 8 letters' , fg = 'red')

result\_label.grid(row = 1000 , column = 0)

else :

return False

def makeId():

global result\_label , logined

usd = username.get()

pw = password.get()

try :

result\_label.config()

except :

result\_label = Label(body\_right , font = 'Robot 20 bold' , bg = '#262626')

if valididty(usd , pw) == False :

mycursor = mydb.cursor()

if already\_exists() == True :

s = 'insert into login values(%s , %s);'

tup = (usd , pw )

mycursor.execute(s , tup)

mycursor.execute("commit")

result\_label.config(text = "Sign up done" , fg = '#28e84a')

result\_label.grid(row = 1000 , column = 0)

logined = True

sign\_in.config(text = 'Logout' , command = logout)

sign\_up.destroy()

if sup :

makeId()

elif sin :

login()

if logined :

Thread(target= threads).start()

def clear\_entry():

username.set("")

password.set("")

def destroy\_everything():

for i in body\_right.grid\_slaves():

i.destroy()

for i in range(200):

body\_right.rowconfigure(i, weight=0)

def signup\_window():

global body\_right , password , username , login\_frame , sin , history\_list , sup

if history\_list[-1] != 'signup' :

history\_list.append('signup')

sup = True

sin = False

clear\_entry()

destroy\_everything()

login\_frame = Frame( body\_right , bg='#606060', height=200 , highlightbackground="black", highlightthickness=1)

login\_label\_heading = Label( login\_frame , text = 'Sign Up', bg='#262626', font='Bebas 25', height=2 , fg='White')

login\_label = Label( login\_frame , text = 'Username', bg='#222222', fg='white', font='Roboto')

login\_field = Entry( login\_frame , textvariable = username)

password\_label = Label( login\_frame , text='Password', bg='#222222', fg='white', font='Roboto')

password\_field = Entry( login\_frame , textvariable = password , show ="\u2022")

submit\_button = Button( login\_frame , text='Submit', width=20 , command = validate\_login, font='Roboto')

#sign\_in\_reference = Button(login\_frame, text='Already Have')

login\_frame.grid(row = 33,column=0,sticky='NEW', padx=35, pady=10)

login\_label\_heading.grid(row = 0, column=0,sticky='WE', padx=5, pady=5, columnspan=2)

login\_label.grid(row =1,column=0,sticky='WE', padx=5, pady=5, ipadx=10, ipady=10)

login\_field.grid(row =1,column=1,sticky='WE', padx=5, pady=5, ipadx=10, ipady=10)

password\_label.grid(row =2,column=0 ,sticky='WE', padx=5,pady=5, ipadx=10, ipady=10)

password\_field.grid(row =2,column=1,sticky='WE', padx=5, pady=5, ipadx=10, ipady=10)

submit\_button.grid(row =3,column=0,columnspan=2, pady=5)

body\_right.rowconfigure(33, weight=1) #Expanding right body's (login frame) vertically

login\_frame.columnconfigure(0, weight=1) #Expanding login frame's (login label) horizontally; column one

login\_frame.columnconfigure(1, weight=1) #Expanding login frame's (login label) horizontally; colmn two

def signin\_window():

global body\_right , password , username , login\_frame , sin , history\_list , sup

if history\_list[-1] != 'signin' :

history\_list.append('signin')

sin = True

sup = False

clear\_entry()

destroy\_everything()

login\_frame = Frame( body\_right , bg='#606060', height=200 , highlightbackground="black", highlightthickness=1)

login\_label\_heading = Label( login\_frame , text = 'Sign In', bg='#262626', font='Bebas 25', height=2 , fg='White')

login\_label = Label( login\_frame , text = 'Username', bg='#222222', fg='white', font='Roboto')

login\_field = Entry( login\_frame , textvariable = username)

password\_label = Label( login\_frame , text='Password', bg='#222222', fg='white', font='Roboto')

password\_field = Entry( login\_frame , textvariable = password , show ="\u2022")

submit\_button = Button( login\_frame , text='Submit', width=20 , command = validate\_login, font='Roboto')

#sign\_in\_reference = Button(login\_frame, text='Already Have')

login\_frame.grid(row = 34,column=0,sticky='NEW', padx=35, pady=10)

login\_label\_heading.grid(row = 0, column=0,sticky='WE', padx=5, pady=5, columnspan=2)

login\_label.grid(row =1,column=0,sticky='WE', padx=5, pady=5, ipadx=10, ipady=10)

login\_field.grid(row =1,column=1,sticky='WE', padx=5, pady=5, ipadx=10, ipady=10)

password\_label.grid(row =2,column=0 ,sticky='WE', padx=5,pady=5, ipadx=10, ipady=10)

password\_field.grid(row =2,column=1,sticky='WE', padx=5, pady=5, ipadx=10, ipady=10)

submit\_button.grid(row =3,column=0,columnspan=2, pady=5)

body\_right.rowconfigure(34, weight=1) #Expanding right body's (login frame) vertically

login\_frame.columnconfigure(0, weight=1) #Expanding login frame's (login label) horizontally; column one

login\_frame.columnconfigure(1, weight=1) #Expanding login frame's (login label) horizontally; colmn two

def check\_connection\_with\_mysql(check = False):

#Checking that project database is their or not if not then will create that

my = ms.connect(host = 'localhost' , user = 'root' , passwd = pas)

mycur = my.cursor()

mycur.execute("Show databases;" )

databases = mycur.fetchall()

if ('project',) not in databases :

mycur.execute("Create database project;")

#Checking that login table is in the database if not then creating that

global mydb ; mydb = ms.connect(host = 'localhost' , user = 'root' , passwd = pas , database = 'project')

mycur = mydb.cursor()

mycur.execute("Show tables ;")

res = mycur.fetchall()

if ('login',) not in res :

mycur.execute("Create table login(Id varchar(30) , PassWord varchar(20)) ;")

if check == True :

try :

for line in open("APPS.sql"):

mycur.execute(line)

mycur.execute('commit')

except :

pass

def back\_command():

global login\_frame , history\_list ,app\_list\_frame , contact\_us\_frame , delete\_frame , bonus\_zone

try :

app\_list\_frame.destroy()

result\_label.destroy()

except :

pass

try :

login\_frame.destroy()

result\_label.destroy()

except :

pass

try :

edit\_apps\_frame.destroy()

result\_label.destroy()

except :

pass

try :

contact\_us\_frame.destroy()

except :

pass

try :

delete\_update\_frame.destroy()

except :

pass

try :

bonus\_zone.destroy()

except :

pass

if len(history\_list) > 1:

del history\_list[-1]

if history\_list[-1] == 'main':

gridding()

elif history\_list[-1] == 'signin' :

del history\_list[-1]

signin\_window()

elif history\_list[-1] == 'signup' :

del history\_list[-1]

signup\_window()

elif history\_list[-1] == 'apps' :

del history\_list[-1]

all\_apps\_command()

elif history\_list[-1] == 'entertainment' :

del history\_list[-1]

entertainment\_command()

elif history\_list[-1] == 'games' :

del history\_list[-1]

games\_command()

elif history\_list[-1] == 'addapps' :

del history\_list[-1]

add\_apps\_command()

elif history\_list[-1] == 'contact' :

del history\_list[-1]

contact\_us\_command()

elif history\_list[-1] == 'delete' :

del history\_list[-1]

app\_delete\_command()

elif history\_list[-1] == 'bonus' :

del history\_list[-1]

bonus\_zone\_command()

def home\_command():

global login\_frame , app\_list\_frame , history\_list , edit\_apps\_frame

history\_list = ['main']

try :

app\_list\_frame.destroy() ;result\_label.destroy() ; gridding()

except :

destroy\_everything() ; gridding()

try :

login\_frame.destroy() ; destroy\_everything() ; gridding()

except :

destroy\_everything() ; gridding()

try :

edit\_apps\_frame.destroy() ; destroy\_everything() ; gridding()

except :

destroy\_everything() ; gridding()

try :

delete\_update\_frame.destroy() ; gridding()

except :

destroy\_everything() ; gridding()

try :

bonus\_zone.destroy() ; gridding()

except :

destroy\_everything() ; gridding()

def show\_apps(reslt = None):

global my\_tree

if reslt == None:

mydb = ms.connect(host = 'localhost' , user = 'root' , passwd = pas , database = 'project')

mycur = mydb.cursor()

if entertainment\_bool :

mycur.execute("Select \* from apps where category = 'entertainment' order by appid;")

elif games\_bool:

mycur.execute("Select \* from apps where category = 'games' order by appid;")

else :

mycur.execute('SELECT \* FROM apps order by appid;')

records = mycur.fetchall()

return records

else :

clear\_apps()

my\_tree.tag\_configure("oddrow", background='#859bbc',foreground='black')

my\_tree.tag\_configure("evenrow",background='#414141',foreground='white')

for i in range(len(reslt)):

if i%2 == 0:

my\_tree.insert(parent= '' , index= 'end' , iid=i , text = 'clear', values = reslt[i] , tags=('evenrow',))

elif i%2 != 0:

my\_tree.insert(parent= '' , index= 'end' , iid=i , text = 'clear', values = reslt[i] , tags=('oddrow',))

def clear\_apps():

global my\_tree

for i in my\_tree.get\_children():

my\_tree.delete(i)

def searching\_all\_apps():

global result\_label

entered\_name = search\_entry.get()

if entered\_name == '' or entered\_name.isspace():

show\_apps()

else :

mydb = ms.connect(host = 'localhost' , user = 'root' , passwd = pas , database = 'project')

mycur = mydb.cursor()

if entertainment\_bool :

mycur.execute(f"Select \* from apps where name LIKE '{entered\_name}%' and category = 'entertainment' order by appid")

elif games\_bool :

mycur.execute(f"Select \* from apps where name LIKE '{entered\_name}%' and category = 'Games' order by appid")

else :

mycur.execute(f"Select \* from apps where name LIKE '{entered\_name}%' order by appid")

res = mycur.fetchall()

if len(res) > 0:

show\_apps(reslt= res)

else :

if entertainment\_bool:

try :

result\_label.config(text = f"Sorry, No App Available With '{entered\_name}' Name\n In Entertainment category" , fg = '#ed4242')

result\_label.grid(row = 1000 , column = 0)

except :

result\_label = Label(body\_right , text = f"Sorry, No App Available With '{entered\_name}' Name\n In Entertainment category" , fg = '#ed4242' ,font = 'Robot 20 bold' , bg = '#262626')

result\_label.grid(row = 1000 , column = 0)

elif games\_bool :

try :

result\_label.config(text = f"Sorry, No App Available With '{entered\_name}' name\n In Games Gategory" , fg = '#ed4242')

result\_label.grid(row = 1000 , column = 0)

except :

result\_label = Label(body\_right , text = f"Sorry, No App Available With '{entered\_name}' name\n In Games Category" , fg = '#ed4242' ,font = 'Robot 20 bold' , bg = '#262626')

result\_label.grid(row = 1000 , column = 0)

else :

try :

result\_label.config(text = f"Sorry, No App Available With '{entered\_name}' name" , fg = '#ed4242')

result\_label.grid(row = 1000 , column = 0)

except :

result\_label = Label(body\_right , text = f"Sorry, No App Available With '{entered\_name}' Name" , fg = '#ed4242' ,font = 'Robot 20 bold' , bg = '#262626')

result\_label.grid(row = 1000 , column = 0)

def apps\_command():

global body\_right , history\_list , app\_list\_frame , searchforallapps , search\_entry , entertainment\_bool , games\_bool , search\_entry\_var , my\_tree

destroy\_everything()

if entertainment\_bool :

if history\_list[-1] != 'entertainment' :

history\_list.append('entertainment')

games\_bool = False

elif games\_bool :

if history\_list[-1] != 'games' :

history\_list.append('games')

entertainment\_bool = False

else :

if history\_list[-1] != 'apps' :

history\_list.append('apps')

result = show\_apps(reslt= None)

app\_list\_frame = Frame(body\_right , highlightbackground="black", highlightthickness=1, bg='#606060')

# my\_scrollbar = Scrollbar(app\_list\_frame)

# my\_scrollbar.grid(row = 1 , column = 1 , sticky = 'NES')

my\_tree = ttk.Treeview(app\_list\_frame) #, yscrollcommand = my\_scrollbar.set

search\_bar = Frame(app\_list\_frame)

search\_bar.grid( row = 0 , column = 0 , sticky='EW' , padx = 10 , pady = (10,0), ipadx = 10)

search\_bar.columnconfigure(0, weight=8)

search\_bar.columnconfigure(1, weight=1)

search\_entry\_var = StringVar()

search\_entry = Entry(search\_bar , bg = '#dedede' , textvariable = search\_entry\_var , highlightbackground = 'WHITE' , font = ("Roboto 19") , fg = 'black' , relief = SUNKEN ,borderwidth = 1)

search\_entry.grid( row = 0 , column = 0 , sticky = 'NEWS' , padx = (0,0))

searchforallapps = Button(search\_bar, bg = '#606060' , fg = 'white' , height = 2 , width = 12,relief = RAISED, text='Search',activebackground='#4F7942', activeforeground='white' , font='Roboto', command = searching\_all\_apps)

searchforallapps.grid( row = 0 , column = 1, sticky='NEWS')

# Creating columns

my\_tree['column'] = ('App IDs','Category','Name','Developer','Size','Views','Description')

my\_tree.column('#0' , minwidth = 0 , width = 0)

my\_tree.column('App IDs' , minwidth = 74 , width = 74 , anchor = W)

my\_tree.column('Category' , minwidth = 208 , width = 208 , anchor = W)

my\_tree.column('Name' , minwidth = 186 , width = 186 , anchor = W)

my\_tree.column('Developer' , minwidth = 186 , width = 186 , anchor = W)

my\_tree.column('Size' , minwidth = 70 , width = 70 , anchor = W)

my\_tree.column('Views' , minwidth = 186 , width = 186 , anchor = W)

my\_tree.column('Description', minwidth = 280 , width = 280 , anchor = W)

# Giving the columns heading

my\_tree.heading('#0' , text= 'Label' , anchor = W)

my\_tree.heading('App IDs' , text = 'App IDs' , anchor = W)

my\_tree.heading('Category' , text = 'Category' , anchor = W)

my\_tree.heading('Name' , text = 'Name' , anchor = W)

my\_tree.heading('Developer' , text = 'Developer' , anchor = W)

my\_tree.heading('Size' , text = 'Size' , anchor = W)

my\_tree.heading('Views' , text ='Views' , anchor = W)

my\_tree.heading('Description' , text = 'Description' , anchor = W)

# Styling

style = ttk.Style()

style.theme\_use('clam')

style.configure("Treeview", rowheight = 45 , fieldbackground = '#8fb198')

style.map('Treeview' , background = [('selected' , '#848179')])

# Giving rows colors

my\_tree.tag\_configure("oddrow", background='#859bbc',foreground='black')

my\_tree.tag\_configure("evenrow",background='#414141',foreground='white')

for i in range(len(result)):

if i%2 == 0:

my\_tree.insert(parent= '' , index= 'end' , iid=i , text = 'clear', values = result[i] , tags=('evenrow',))

elif i%2 != 0:

my\_tree.insert(parent= '' , index= 'end' , iid=i , text = 'clear', values = result[i], tags=('oddrow',))

# my\_scrollbar.config(command = my\_tree.yview)

app\_list\_frame.grid(padx=40, pady=10, ipady=10, ipadx=10)

my\_tree.grid(padx = 3)

app\_list\_frame.columnconfigure(0 , weight = 1)

app\_list\_frame.rowconfigure(1 , weight = 1)

def all\_apps\_command():

global entertainment\_bool , games\_bool

entertainment\_bool , games\_bool = False , False

apps\_command()

def entertainment\_command():

global entertainment\_bool , games\_bool

entertainment\_bool = True

games\_bool = False

apps\_command()

def games\_command():

global games\_bool , entertainment\_bool , search\_entry\_var

games\_bool = True

entertainment\_bool = False

apps\_command()

def submit():

global result\_label , category\_for\_storing

try :

result\_label.config(text = '')

except :

result\_label = Label(body\_right , font = 'Robot 20 bold' , bg = '#262626')

def valid():

global appid\_for\_storing , category\_for\_storing , name\_for\_storing , developer\_for\_storing , size\_for\_storing , views\_for\_storing , description\_for\_storing

global result\_label

if len(str(appid\_for\_storing.get())) <= 1:

result\_label.config(text = 'Appid too short' , fg = 'red') ; result\_label.grid(row = 1000 )

return False

elif name\_for\_storing.get().isspace() :

result\_label.config(text = 'Name too short' , fg = 'red') ; result\_label.grid(row = 1000 )

return False

elif developer\_for\_storing.get().isspace() :

result\_label.config(text = 'Developer name too short' , fg = 'red') ; result\_label.grid(row = 1000 )

return False

elif len(size\_for\_storing.get()) <= 0 :

result\_label.config(text = "Size of the game can't be this" , fg = 'red') ; result\_label.grid(row = 1000 )

return False

if valid() != False :

global mycur

result\_label.config(text = "Added" , fg = '#28e84a') ; result\_label.grid(row = 1000 )

mydb = ms.connect(host='localhost', user='root',passwd=pas, database='project')

mycur = mydb.cursor()

if description\_for\_storing.get().isspace() or description\_for\_storing.get() == '':

tup=(appid\_for\_storing.get() , category\_for\_storing.get() , name\_for\_storing.get() ,developer\_for\_storing.get() , size\_for\_storing.get() , views\_for\_storing)

s='insert into apps(appid , category , name , developer , size , views) values( %s , %s , %s , %s , %s , %s);'

else:

tup=(appid\_for\_storing.get() , category\_for\_storing.get() , name\_for\_storing.get() ,developer\_for\_storing.get() , size\_for\_storing.get() , views\_for\_storing , description\_for\_storing.get())

s='insert into apps values( %s , %s , %s , %s , %s , %s , %s);'

try :

mycur.execute(s, tup)

mycur.execute("commit")

except :

result\_label.config(text = "Something went wrong" , fg = 'red')

result\_label.grid(row = 1000 )

def add\_apps\_command():

destroy\_everything()

global appid\_for\_storing , category\_for\_storing , name\_for\_storing , developer\_for\_storing , size\_for\_storing , views\_for\_storing , description\_for\_storing

global history\_list , edit\_apps\_frame

if history\_list[-1] != 'addapps' :

history\_list.append('addapps')

edit\_apps\_frame = Frame(body\_right, bg='#606060', highlightbackground="black", highlightthickness=1)

name\_for\_storing = StringVar()

category\_for\_storing = StringVar() # Adding combobox drop down list

developer\_for\_storing = StringVar()

size\_for\_storing = StringVar()

description\_for\_storing = StringVar()

appid\_for\_storing = IntVar()

views\_for\_storing = 0

edit\_apps\_frame\_label= Label(edit\_apps\_frame, bg = '#222222', fg = 'white', height=2, font='Bebas 25', text = 'Add Apps')

idd = Label(edit\_apps\_frame, bg = '#222222', fg = 'white', height=2, font='Roboto' , text = 'Appid')

categoryy = Label(edit\_apps\_frame, bg = '#222222', fg = 'white', height=2, font='Roboto' , text = 'Category')

namee = Label(edit\_apps\_frame, bg = '#222222', fg = 'white', height=2, font='Roboto' , text = 'Name')

developerr = Label(edit\_apps\_frame, bg = '#222222', fg = 'white', height=2, font='Roboto' , text = 'Developer')

sizee = Label(edit\_apps\_frame, bg = '#222222', fg = 'white', height=2, font='Roboto' , text = 'Size')

descriptionn = Label(edit\_apps\_frame, bg = '#222222', fg = 'white', height=2, font='Roboto' , text = 'Description')

iddd = Entry( edit\_apps\_frame, font = 'Roboto 20', textvariable = appid\_for\_storing)

categoryyy = ttk.Combobox(edit\_apps\_frame, font = 'Roboto 20', textvariable = category\_for\_storing)

nameee = Entry( edit\_apps\_frame, font = 'Roboto 20', textvariable = name\_for\_storing)

developerrr = Entry( edit\_apps\_frame, font = 'Roboto 20', textvariable = developer\_for\_storing)

sizeee = Entry( edit\_apps\_frame, font = 'Roboto 20', textvariable = size\_for\_storing)

descriptionnn = Entry( edit\_apps\_frame, font = 'Roboto 20', textvariable = description\_for\_storing)

sub = Button( edit\_apps\_frame, text='Submit',command=submit,relief='groove', width=10, height=2)

body\_right.rowconfigure( 5, weight=1)

edit\_apps\_frame.grid( row=5, column=0, sticky='ENSW', padx = 35 , pady = 10)

edit\_apps\_frame\_label.grid(row=0, column=0, sticky='EW', padx = 10 , pady = (10,0) , columnspan=2)

idd.grid( row = 1, column=0, sticky='WEN', padx=10, pady=(10,0), ipadx=5, ipady=5)

categoryy.grid( row = 2, column=0, sticky='WEN', padx=10, pady=(10,0), ipadx=5, ipady=5)

namee.grid( row = 3, column=0, sticky='WEN', padx=10, pady=(10,0), ipadx=5, ipady=5)

developerr.grid( row = 4, column=0, sticky='WNE', padx=10, pady=(10,0), ipadx=5, ipady=5)

sizee.grid( row = 5, column=0, sticky='WNE', padx=10, pady=(10,0), ipadx=5, ipady=5)

descriptionn.grid( row = 6, column=0, sticky='WNE', padx=10, pady=(10,0), ipadx=5, ipady=5)

iddd.grid( row = 1, column=1, sticky='WNES', padx=10, pady=10, ipadx=5, ipady=5)

categoryyy.grid( row = 2, column=1 ,sticky='EWNS', padx=10, pady=10, ipadx=5, ipady=5)

nameee.grid( row = 3, column=1, sticky='WNES', padx=10, pady=10, ipadx=5, ipady=5)

developerrr.grid( row = 4, column=1, sticky='WNES', padx=10, pady=10, ipadx=5, ipady=5)

sizeee.grid( row = 5, column=1, sticky='WNES', padx=10, pady=10, ipadx=5, ipady=5)

descriptionnn.grid( row = 6, column=1, sticky='WNES', padx=10, pady=10, ipadx=5, ipady=5)

sub.grid( row=7, column=0, sticky='N', columnspan=2, padx=10, pady=(10,10), ipadx=5, ipady=5)

edit\_apps\_frame.columnconfigure(0, weight=1)

edit\_apps\_frame.columnconfigure(1, weight=2)

for i in range(7):

edit\_apps\_frame.rowconfigure(i, weight=1)

# Adding combobox drop down list

categoryyy['values'] = (' Entertainment', ' Games' , 'Business' ,'Lifestyle','Music & Audio','Photography','Social','Video Players & Editors')

categoryyy.current()

def contact\_us\_command(): ##Contact Us

global history\_list , contact\_us\_frame

if history\_list[-1] != 'contact' :

history\_list.append('contact')

destroy\_everything()

contact\_us\_frame= Frame(body\_right, bg='#606060')

contact\_us\_label= Label(contact\_us\_frame, text='Contact Us', bg='#262626', font='Bebas 25', height=1 , fg='White')

contact\_us\_frame.grid(row=10,column=0,sticky='NEWS', padx=40, pady=10) #Contact us frame

body\_right.rowconfigure(10, weight=1) #contact us frame

contact\_us\_frame.columnconfigure(0,weight=1) #expanding 2 columns in frame,

contact\_us\_frame.columnconfigure(1,weight=1)

person1= Frame(contact\_us\_frame, highlightbackground="black", highlightthickness=1, height=100,width=100) #for person 1 frame

person1.grid(row=1,column=0, sticky='EW', padx=50, pady=20)

person1.columnconfigure(0, weight=1)

person1\_show\_photo=Label(person1,image=person1\_photo) #person 1 photo

person1\_show\_photo.grid(row=0, padx=10, pady=10)

person1\_name=Label(person1, text='Avitesh Murmu', bg='#385723', fg='white',height=1, font='Bebas 25' )

person1\_name.grid(row=1,sticky='EW')

person1\_description=Label(person1, text='''

Class : XII

Section : A

Roll No : 36

''', justify=LEFT, bg='#92D050', fg='black', height=5, font='Roboto')

person1\_description.grid(row=2, sticky='EW')

person2= Frame(contact\_us\_frame, highlightbackground="black", highlightthickness=1, height=100,width=100) #for person 2 frame

person2.grid(row=1,column=1, sticky='EW', padx=50)

person2.columnconfigure(0, weight=1)

person2\_show\_photo=Label(person2,image=person2\_photo) #person 2 photo

person2\_show\_photo.grid(row=0, padx=10, pady=10)

person2\_name=Label(person2, text='Roshan Raj', bg='#385723', fg='white',height=1, font='Bebas 25')

person2\_name.grid(row=1,sticky='EW')

person2\_description=Label(person2, text='''

Class : XII

Section : A

Roll No : 13

''', justify=LEFT, bg='#92D050', fg='black', height=5, font='Roboto')

person2\_description.grid(row=2, sticky='EW')

contact\_us\_button.grid( row=100, column=0, sticky='NEW', ipadx=10, ipady=10, pady=5) #Contact Us button in the left

contact\_us\_label.grid(row=0, column=0,padx=10,ipadx=10, ipady=15, pady=10, sticky='WE', columnspan=2) #Contact us label/heading in the top

#######################################################################################

def snake\_run():

root.wm\_state('iconic')

try :

obj = Snake()

obj.WelcomeScreen()

except :

pass

root.wm\_state('zoomed')

def calculator\_run():

root.wm\_state('iconic')

Calculator()

def game\_run():

root.wm\_state('iconic')

Game()

def app\_delete\_command(appkaappid = None , appname = None):

global history\_list , delete\_update\_frame , delete\_frame , appkaappid\_no , appkanam

if history\_list[-1] != 'delete' :

history\_list.append('delete')

destroy\_everything()

def remove\_space(s):

r = ''

for i in range(len(s)):

if s[i] != ' ':

r += s[i]

return r

def delete\_app():

if appkaappid != None and appname != None:

mydb = ms.connect(host = 'localhost' , user = 'root' , passwd = pas , database = 'project')

mycur = mydb.cursor()

mycur.execute("Select \* from apps ;")

res = mycur.fetchall()

if appkaappid == '' and appname == '':

anymessage\_label.config(text = 'Appid and Appname are empty')

anymessage\_label .grid(row=4, column=0, padx=10, pady=5)

elif appkaappid == '':

anymessage\_label.config(text = 'Appid is empty')

anymessage\_label .grid(row=4, column=0, padx=10, pady=5)

elif appname == '':

anymessage\_label.config(text = 'Appname is empty')

anymessage\_label .grid(row=4, column=0, padx=10, pady=5)

else :

count = False

for row in res :

if row[0] == int(appkaappid) and remove\_space(row[2].lower()) == remove\_space(appname.lower()) :

mycur.execute(f'DELETE FROM apps WHERE appid= {int(appkaappid)}')

mycur.execute("commit")

count = True

anymessage\_label.config(text= "Deleting App Done", fg= 'Green')

anymessage\_label .grid(row=4, column=0, padx=10, pady=5)

break

elif row[0] == int(appkaappid) and remove\_space(row[2].lower()) != remove\_space(appname.lower()) :

anymessage\_label.config(text = 'Something wrong in App Name', fg = 'red')

anymessage\_label .grid(row=4, column=0, padx=10, pady=5)

count = True

break

elif row[0] != int(appkaappid) and remove\_space(row[2].lower()) == remove\_space(appname.lower()) :

anymessage\_label.config(text = 'Something wrong in App id', fg = 'red')

anymessage\_label .grid(row=4, column=0, padx=10, pady=5)

count = True

break

if count == False :

anymessage\_label.config(text = 'Nothing matches.... You should try again' , fg = 'red')

anymessage\_label .grid(row=4, column=0, padx=10, pady=5)

delete\_update\_frame = Frame( body\_right, bg='#606060')

delete\_frame = Frame( delete\_update\_frame, bg='#606060')

delete\_label = Label( delete\_frame, text='Delete App', bg='#262626', font='Bebas 25', height=2 , fg='White')

anymessage\_label = Label( delete\_frame, text = "Try deleting any app" ,bg='#262626', fg= 'Green', font='Roboto 18')

enter\_app\_id\_entry = Entry( delete\_frame, textvariable = appkaappid\_no , font = (17))

enter\_name\_entry = Entry( delete\_frame, textvariable = appkanam , font = (17))

submit2 = Button(delete\_frame, text='Submit', height=2 , width=20 , command = lambda : app\_delete\_command(appkaappid = appkaappid\_no.get() , appname= appkanam.get()))

try :

delete\_app()

except :

pass

delete\_label.grid(row=0, sticky='EW', padx=10, pady=10, columnspan=2)

enter\_app\_id\_label = Label(delete\_frame, text='Enter App Id', bg = '#222222', fg = 'white', height=2, font='Roboto')

enter\_app\_id\_label.grid(row=1, column=0, sticky='EW', padx=10, pady = (0, 5))

enter\_app\_name\_label = Label(delete\_frame, text='Enter App Name', bg = '#222222', fg = 'white', height=2, font='Roboto')

enter\_app\_name\_label.grid(row=2, column=0, sticky='EW', padx=10,pady = (5, 0))

enter\_app\_id\_entry.grid(row=1, column=1, sticky="EWNS", padx=10 , pady = (0, 5))

enter\_name\_entry .grid(row=2, column=1, sticky="EWNS", padx=10 , pady = (5, 0))

submit2 .grid(row=3, column=0, columnspan=2 , pady=10)

anymessage\_label .grid(row=4, column=0, pady=5 , padx=10, columnspan=2)

#######################################################################################

'''

# Update Apps

update\_frame = Frame(delete\_update\_frame, bg='#606060')

update\_frame.grid(row=1, sticky='NEWS') #update Frame

update\_frame.columnconfigure(0, weight=1)

update\_label = Label(update\_frame, text='Update App', bg='#262626', font='Bebas 25', height=2 , fg='White')

update\_label.grid(row=1, sticky='EW', padx=10, pady=20, columnspan=2)

'''

# Delete and Update Frame

delete\_update\_frame.grid(row= 0, sticky='NEWS', padx=35, pady=10)

body\_right.rowconfigure(0, weight=1)

delete\_update\_frame.columnconfigure(0, weight=1)

delete\_frame.grid(row=0, sticky='NEWS') #Delete Frame

delete\_frame.columnconfigure(0, weight=1)

delete\_frame.columnconfigure(1, weight=3)

def bonus\_zone\_command():

global history\_list , bonus\_zone , bonus\_snake\_logo , bonus\_tic\_tac\_toe\_logo , bonus\_calculator\_logo

if history\_list[-1] != 'bonus':

history\_list.append('bonus')

destroy\_everything()

t=1

bonus\_zone=Frame(body\_right, bg='#b8a753') # bonus Frame in Right

bonus\_zone.grid(row=45, sticky='NEWS', padx=40, pady=10)

body\_right.rowconfigure(45, weight=1)

bonus\_heading = Label(bonus\_zone, text='\u272F '+'Bonus Zone'+' \u272F', bg='#262626', font='Bebas 25', height=2 , fg='#c4a502') # Bonus Heading

bonus\_heading.grid(row=0, column=0,padx=10,ipadx=10, ipady=15, pady=10, sticky='WE', columnspan=3)

# Snake Game

bonus\_frame\_snake = Frame(bonus\_zone, bg='#8db705') # Snake Frame

bonus\_snake\_logo = PhotoImage(file= r'Photos/snake\_game.png')

bonus\_frame\_snake\_photo = Button(bonus\_frame\_snake, image=bonus\_snake\_logo, relief=GROOVE , command = snake\_run)

bonus\_snake\_label = Button(bonus\_frame\_snake, text='Play Snake', bg='#8db705', relief=GROOVE, activeforeground='#8db705', fg='White', font='Roboto' , command = snake\_run)

bonus\_frame\_snake.grid( row=t, column=0, padx=(25,0), pady=10, sticky='EW', ipady=(10))

bonus\_frame\_snake\_photo.grid( row=0, column=0, pady=(20,10))

bonus\_snake\_label.grid( row=1, sticky='news', padx=10)

bonus\_frame\_snake.columnconfigure(0, weight=1)

# Tic Tac Toe

bonus\_frame\_tic\_tac\_toe = Frame(bonus\_zone, height=300, bg='#eb5855') # Snake Tic Tac Toe

bonus\_tic\_tac\_toe\_logo = PhotoImage(file = r'Photos/tictactoe.png')

bonus\_tic\_tac\_toe\_photo = Button(bonus\_frame\_tic\_tac\_toe, image=bonus\_tic\_tac\_toe\_logo, relief=GROOVE , command = game\_run)

bonus\_tic\_tac\_toe\_label = Button(bonus\_frame\_tic\_tac\_toe, text='Play Tic Tac Toe', bg='#eb5855', relief=GROOVE, activeforeground='#eb5855', fg='White', font='Roboto' , command = game\_run)

bonus\_frame\_tic\_tac\_toe.grid( row=t, column=1, padx=(25,0), pady=10, sticky='EW', ipady=10)

bonus\_tic\_tac\_toe\_photo.grid( row=0, column=0, pady=(20,10))

bonus\_tic\_tac\_toe\_label.grid( row=1, sticky='news', padx=10)

bonus\_frame\_tic\_tac\_toe.columnconfigure(0, weight=1)

# Calculator

bonus\_frame\_calculator = Frame(bonus\_zone, height=300, bg='#b86d33') # Calculator Frame

bonus\_calculator\_logo = PhotoImage(file = r'Photos/calculator.png')

bonus\_calculator\_photo = Button(bonus\_frame\_calculator, image=bonus\_calculator\_logo, relief=GROOVE , command = calculator\_run)

bonus\_calculator\_label = Button(bonus\_frame\_calculator, text='Use Calculator', bg='#b86d33', relief=GROOVE, activeforeground='#b86d33', fg='White', font='Roboto' , command = calculator\_run)

bonus\_frame\_calculator.grid( row=t, column=2, padx=(25,25), pady=10, sticky='EW', ipady=10)

bonus\_calculator\_photo.grid( row=0, column=0, pady=(20,10))

bonus\_calculator\_label.grid( row=1, sticky='news', padx=10)

bonus\_frame\_calculator.columnconfigure(0, weight=1)

for i in range(3):

bonus\_zone.columnconfigure(i, weight=1)

def check\_connection\_and\_open\_main\_py():

if name\_variable.get() == '' and password\_variable.get() == '' and class\_variable.get() == '' and rollno\_variable.get() == '' :

anymessage.config(text = 'Please enter the above entries...\nAll entries empty' , bg = 'red' , fg = 'white')

elif name\_variable.get() != '' and password\_variable.get() != '':

global pas

try :

ms.connect(host = 'localhost' , user = 'root' , passwd = password\_variable.get() , database = 'project')

pas = password\_variable.get()

delete\_frame2()

hello\_user\_command()

retrieve\_frame1()

except :

anymessage.config(text = '!! Wrong !!\nPlease make sure that you entered your password CORRECTLY' , bg = 'red' , fg = 'white')

if name\_variable.get() == '' and password\_variable.get() == '':

label\_1.config(text = 'Your MySQL Password \*Required' , fg = 'red')

label\_2.config(text = 'Your Name \*Required' , fg = 'red')

elif name\_variable.get() != '' and password\_variable.get() == '' :

label\_1.config(text = 'Your MySQL Password \*Required' , fg = 'red')

label\_2.config(text = 'Your Name' , fg = 'white')

anymessage.config(text = '', bg='#262626')

elif name\_variable.get() == '' and password\_variable.get() != '':

label\_1.config(text = 'Your MySQL Password' , fg = 'white')

label\_2.config(text = 'Your Name \*Required' , fg = 'red')

anymessage.config(text = '', bg='#262626')

def start\_to\_main():

frame2.grid\_forget()

root.rowconfigure( 1 , weight=0)

frame.grid( sticky=(N, E, W, S), row=0)

root.rowconfigure( 0 , weight=1)

def delete\_frame2():

frame2.grid\_forget()

root.rowconfigure( 1 , weight=0)

def retrieve\_frame1():

frame.grid( sticky=(N, E, W, S), row=0)

root.rowconfigure( 0 , weight=1);check\_connection\_with\_mysql(check= True)

def delete\_signup\_and\_signin():

sign\_up.grid\_forget()

sign\_in.grid\_forget()

def hello\_user\_command():

if name\_variable.get() == '':

hello\_user = 'Welcome, Guest'

else:

hello\_user = 'Welcome \n'+ name\_variable.get()

hello\_frame = Frame(time\_greetings\_signup\_signin, bg='#385723')

hello\_frame.grid(row= 0, column=1, sticky='NEWS', ipadx=5, pady=10, padx=(0,10), ipady=3)

hello\_frame.rowconfigure(0, weight=1)

user\_display = Label(hello\_frame, image=user\_photo, bg = '#385723')

user\_display.grid(row=0, padx=10)

hello\_show = Label(hello\_frame , text = hello\_user , height= 3, bg='#385723', fg='white', font='Roboto 9')

hello\_show.grid(row= 0, column=1, sticky='NEWS')

#######################################################################################

#Global variables

username = StringVar()

password = StringVar()

sup = False ; sin = True

history\_list = ['main']

logined = False

entertainment\_bool = False

games\_bool = False

appkaappid\_no = StringVar()

appkanam = StringVar()

user\_photo = PhotoImage(file= r'Photos/user.png')

#######################################################################################

#Header

header = Frame(frame, height =5, bg='#00b050') #'#00A86B'

time\_greetings\_signup\_signin = Frame(header, bg='#00b050')

time\_greetings\_signup\_signin.grid(column=2, sticky='E')

#####################################

#for\_sign\_up\_and\_sign\_in

sign\_up =Button(time\_greetings\_signup\_signin, text = 'Sign Up',height= 3,width=7, bg='#385723', relief=GROOVE , fg='white', command=signup\_window, font='Roboto 9')

sign\_in =Button(time\_greetings\_signup\_signin, text = 'Sign In',height= 3,width=7, bg='#385723', relief=GROOVE , fg='white', command=signin\_window, font='Roboto 9')

labl = tk.Label(time\_greetings\_signup\_signin ,text = '', bg = '#aaff00' , fg = 'black' , font = 'Roboto 17')

#####################################

#Add Back ,Home, Logo

back\_and\_home = Frame(header , bg='#00b051')

back = Button(back\_and\_home, image = back\_lg1, bg='#339933', relief=GROOVE, command = back\_command) #"\u00AB BACK"

appstore = Label(frame, image = appstore\_lg2, bg='white')

home = Button(back\_and\_home, image = home\_photo, bg = '#339933', relief=GROOVE, command = home\_command, height=51, width=51)

light\_green =Frame(header, height=5, bg='#aaff00')

light\_green.grid(row=1, columnspan=5,sticky='WE')

######################################

#Body

body = Frame(frame)

##############################

#Left Body

body\_left = Frame(body, bg='#404040')

#Categories

category = Frame(body\_left, bg = '#404040')

apps = Button(category, bg = '#606060' , fg = 'white' , text='All Apps', activebackground='#4F7942', activeforeground='white' , font='Roboto', command = all\_apps\_command)

entertainment = Button(category, bg = '#606060' , fg = 'white' , text='Entertainment', activebackground='#4F7942', activeforeground='white' , font='Roboto', command = entertainment\_command)

games = Button(category, bg = '#606060' , fg = 'white' , text='Games', activebackground='#4F7942', activeforeground='white' , font='Roboto', command = games\_command)

edit\_apps = Button(category, bg = '#606060' , fg = 'white' , text='Add Apps', activebackground='#4F7942', activeforeground='white' , font='Roboto', command = add\_apps\_command)

contact\_us\_button= Button(category, bg = '#c4c4c4' , fg = 'black' , text='Contact Us', activebackground='#4F7942', activeforeground='white' , font='Roboto', command = contact\_us\_command)

delete\_button = Button(category, bg = '#606060' , fg = 'white' , text='Delete/Update \n Apps',activebackground='#4F7942', activeforeground='white' , font='Roboto', command = app\_delete\_command)

bonus\_zone\_button= Button(category, bg = '#FFD700' , fg = 'black' , text='\u272F'+'Bonus Zone'+'\u272F', activebackground='#cfb00c', activeforeground='white' , font='Roboto' , command = bonus\_zone\_command)

##############################

#Right body

body\_right = Frame(body, bg='#262626')

result\_label = Label(body\_right , font = 'Robot 20 bold' , bg = '#262626')

gridding()

##################################################################### delete and update Apps

#Footer

footer = Frame(frame, height=40, bg='#00b050')

credits = Button(footer,text='Copyright © 2020 AVIONICS & RRajj Inc. All rights reserved.' , bg = '#92D050', fg = '#222222', width=80, )

#######################################################################################

#First window starts here

frame2 = Frame(root, bg='#262626')

frame2.columnconfigure(0, weight=1)

frame2.rowconfigure(101, weight=1)

q=0

# Heading

heading\_label = Label(frame2, text='Please Enter', bg='#00b050', font='Bebas 30', fg='White')

heading\_label.grid(row=q, column=0, ipadx=30, ipady=15, sticky='EW', columnspan=2)

underline\_design = Frame(frame2, height=3, bg='White')

underline\_design.grid(row=q+1, sticky='EW', columnspan=2)

# Logo

front\_photo=Label(frame2,image=front\_photo\_variable)

front\_photo.grid(row=2, column=1, rowspan=102, padx=(0,0), pady=0)

# MySQL Password

label\_1 = Label(frame2, text='Your MySQL Password', bg='#262626', font='Bebas 17', fg='White')

label\_1 .grid(row=q+2, column=0, padx=30, pady=(10,0), sticky='W')

password\_variable=StringVar()

passw = Entry(frame2, font = 'Roboto 15', textvariable = password\_variable , show = '\*')

passw.grid(row=q+3, column=0, padx=30, pady=(0,0), sticky='NEWS')

# Name

label\_2 = Label(frame2, text='Your Name',height= 1, bg='#262626', font='Bebas 17', fg='White')

label\_2 .grid(row=q+11, column=0, padx=30, pady=(30,0), sticky='W')

name\_variable=StringVar()

name\_entry = Entry(frame2, font = 'Roboto 15', textvariable = name\_variable)

name\_entry.grid(row=q+12, column=0, padx=30, pady=(0,0), sticky='NEWS')

# Class

label\_3 = Label(frame2, text='Your Class',height= 1, bg='#262626', font='Bebas 17', fg='White')

label\_3.grid(row=q+21, column=0, padx=30, pady=(30,0), sticky='W')

class\_variable=StringVar()

class\_chosen = ttk.Combobox(frame2, font = 'Roboto 15', textvariable = class\_variable)

# Adding combobox drop down list

class\_chosen['values'] = ('X', 'XI', 'XII', 'None')

class\_chosen.current()

class\_chosen.grid(row=q+23, sticky='WE', padx=30, pady=(0,30), column=0)

# RollNO

label\_4 = Label(frame2, text='Your Roll No',height= 1, bg='#262626', font='Bebas 17', fg='White')

label\_4.grid(row=q+31, column=0, padx=30, pady=(0,0), sticky='W')

rollno\_variable =StringVar()

rollno\_chosen = ttk.Combobox(frame2, font = 'Roboto 15', textvariable = rollno\_variable)

# Adding combobox drop down list

value = []

for i in range(1,41):

value.append(i)

value.append('None')

rollno\_chosen['values'] = value

rollno\_chosen.current()

rollno\_chosen.grid(row=q+33, sticky='NEWS', padx=30, pady=(0,30))

# Section

subm = Button( frame2, text='Submit', command = lambda : check\_connection\_and\_open\_main\_py(), height=2, width=20, relief= GROOVE)

subm.grid(row=100, pady=(0,30))

anymessage=Label(frame2, font='Roboto 17', text = '', bg='#262626' , fg = 'white')

anymessage.grid(row=102, padx=30, pady=(0,15), sticky='EWNS')

#######################################################################################

#Griding

root.grid()

frame2.grid( sticky=(N, E, W, S), row=1)

#Header

header.grid( row=0, sticky=(N, E, W)) #Griding Header And its components

back.grid( row=0, column=0, sticky=(W), padx=10, pady=5)

back\_and\_home.grid( row=0, column=0, sticky='W')

appstore.grid( row=0, column=0, padx=10, pady=5)

home.grid( row=0, column=1, sticky='W')

sign\_up.grid( row=0, column=3, sticky=(E), padx=10, pady=5) #Done

sign\_in.grid( row=0, column=5, sticky=(E), padx=10, pady=5) #Done

labl.grid( row=0, column=0, sticky=(E), padx=(0 , 15), ipadx=15, ipady=9, pady=5)

clock()

#######################################################################################

#Griding Main Body, Row is equal to 1 because it should be in the frames's second row

body.grid( row=1, sticky=('NEWS'))

body\_left.grid( row=0, column=0, sticky='NEWS') #Left Body

category.grid( row=0, column=0, sticky='WNE', padx=10, pady=10) #Category

apps.grid( row=0, column=0, sticky='WE', ipadx=10, ipady=10, pady=5)

entertainment.grid( row=1, column=0, sticky='WE', ipadx=10, ipady=10, pady=5)

games.grid( row=2, column=0, sticky='WE', ipadx=10, ipady=10, pady=5)

edit\_apps.grid( row=3, column=0, sticky='EW', ipadx=10, ipady=10, pady=5)

contact\_us\_button.grid( row=100, column=0, sticky='NEW', ipadx=10, ipady=10, pady=5)

delete\_button.grid( row=14, column=0, sticky='NEWS')

bonus\_zone\_button.grid( row=4, column=0, sticky='EW', ipadx=10, ipady=10, pady=5)

#Body Right

body\_right.grid( row=0, column=1, sticky='NEWS') #Right Body

#footer

footer.grid( row=2, sticky='WE')

credits.grid( padx=10, pady=5)

#Configure

#Root Main Window

root.columnconfigure( 0 , weight=1)

root.rowconfigure( 0 , weight=0)

root.rowconfigure( 1 , weight=1)

# Header

frame.columnconfigure( 0 , weight=1) #Header - Column ,and must not be repeated

frame.rowconfigure( 0 , weight=0) #Header - Row

# Back

header.columnconfigure( 0 , weight=1) #Back

header.columnconfigure( 1 , weight=1) #AppStore (Heading)

header.columnconfigure( 2 , weight=1) #SignUp

# Body

# Body Left

frame.columnconfigure( 1 , weight=0) #Body - Column , for example here it should not be 1

frame.rowconfigure( 1 , weight=1) #Body - Row

body.rowconfigure( 0 , weight=1) #Body\_Left - Row

body.columnconfigure( 0 , weight=1) #Body\_Left - Column

body\_left.columnconfigure( 0 , weight=1) #Category

body\_left.rowconfigure( 0 , weight=1) #Category

category.columnconfigure( 0 , weight=1) #Entertainment

# Body Right

body.columnconfigure( 1 , weight=100) #Body\_Right - 1 for right Column

body\_right.columnconfigure( 0 , weight=1) #Expanding right body's (login frame) horizontally

body\_right.rowconfigure( 0 , weight=1) #Expanding right body's (login frame) vertically

body\_right.rowconfigure( 5 , weight=1)

# Footer

footer.columnconfigure( 0 , weight=1) #Credits

#######################################################################################

# Output

root.protocol("WM\_DELETE\_WINDOW" , close)

root.mainloop()