**HABIT TRACKER**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**BHARGAV JALAGAM - 1602-20-737-068**

**AKASH AAKI - 1602-20-737-105**

**NANDANI V - 1602-20-737-118**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2022**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

****

**DECLARATION BY THE CANDIDATE**

We, **NANDANI V, JALAGAM BHARGAV** and **AKASH A,** bearing hall ticket numbers, **1602-20-737-118, 1602-20-737-068** and **1602-20-737-105**, hereby declare that the project report entitled **“HABIT TRACKER“** is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**JALAGAM BHARGAV**

**1602-20-737-068**

**AKASH AAKI**

**1602-20-737-105**

**NANDANI V**

**1602-20-737-118**

(Faculty In-Charge) (Head, Dept of IT)

**ACKNOWLEDGEMENT**

The enduring pages of the work are the cumulative sequence of extensive guidance and arduous work. We wish to acknowledge and express our personal gratitude to all those without whom this work could not have been reality.

We feel very delighted to get this rare opportunity to show our profound senses of reverences and indebtedness to our esteemed teacher **DRL PRASANNA** ma’am and **RADHA GADIGE** ma’am, for their keen and sustained interest, valuable advice, throughout the course of which led our mini project, to a successful completion. For this kind act of consideration, we are beholder to them in special manner and no one can fully convey our feelings of respect and regard for them.

**ABSTRACT**

This is a Habit Tracker with Graphical User Interface. This can be used by anyone who would like to improve their life. The user must register himself and verify himself with his email account. The user should login with the registered account. Multiple users can be registered. The user will be provided with many habits, he should tick the habits once performed. The user’s streak on the habit will be incremented by 1 if the habit is done, which will motivate the user and is a reward for completing the habit. The steaks can be viewed by clicking a button present on the screen, which will reveal the steaks for all the habits.

This project is developed to provide fun while doing habits. We came across this project looking at various advertisements during the Indian premier league. For doing this project, we gained some basic knowledge about GUI programming in python. The package tkinter is used in this project, it will provide the required GUI.

**TABLE OF CONTENTS**

| S.No | Topic | Page No. |
| --- | --- | --- |
| 1. | Introduction | 6 |
| 2. | Technology | 7 |
| 3. | Design | 8 |
| 4. | Implementation | 13 |
| 5. | Testing | 20 |
| 6. | Additional Knowledge Acquired | 21 |
| 7. | Conclusion and Future work | 22 |
| 8. | References | 23 |

**INTRODUCTION**

If you want to stick with a habit for good, one simple and effective thing you can do is keep a habit tracker. Habit Tracker is as it says, a Habit Tracker. A habit tracker reminds you to act, it motivates you to continue, and provides immediate satisfaction. Habit tracker can help kickstart a habit or keep you on track with behaviors that you tend to forget or let slide when things get busy. You can keep track of your habits. It is a very simple code and can be used to keep track of your habits.

First of all, the user must register/ sign up first. This data is stored in the mysql database. Then, he can login with the id and password set by him earlier, if wrong id or password is entered, error will be shown. After log-in, he has got several habits visible to him, he can tick each one off whenever the habit is performed. He can check the streaks for all the habits with a click of the button available on the screen.

Even if the user logs-out of the Habit Tracker, the streaks will be stored safely in the mysql database. Once the user logs back in, the habits will be visible again and can be ticked off and increase the streak.

The user’s login id and password will be stored inside the mysql database, which can be accessed by the admin and can remove users.

**TECHNOLOGY**

**Hardware Requirements**

* 512 MB RAM
* 2GB HDD
* CORE i5

**Software Requirements**

* Windows XP/ Windows 2000
* PYTHON INTERPRETER

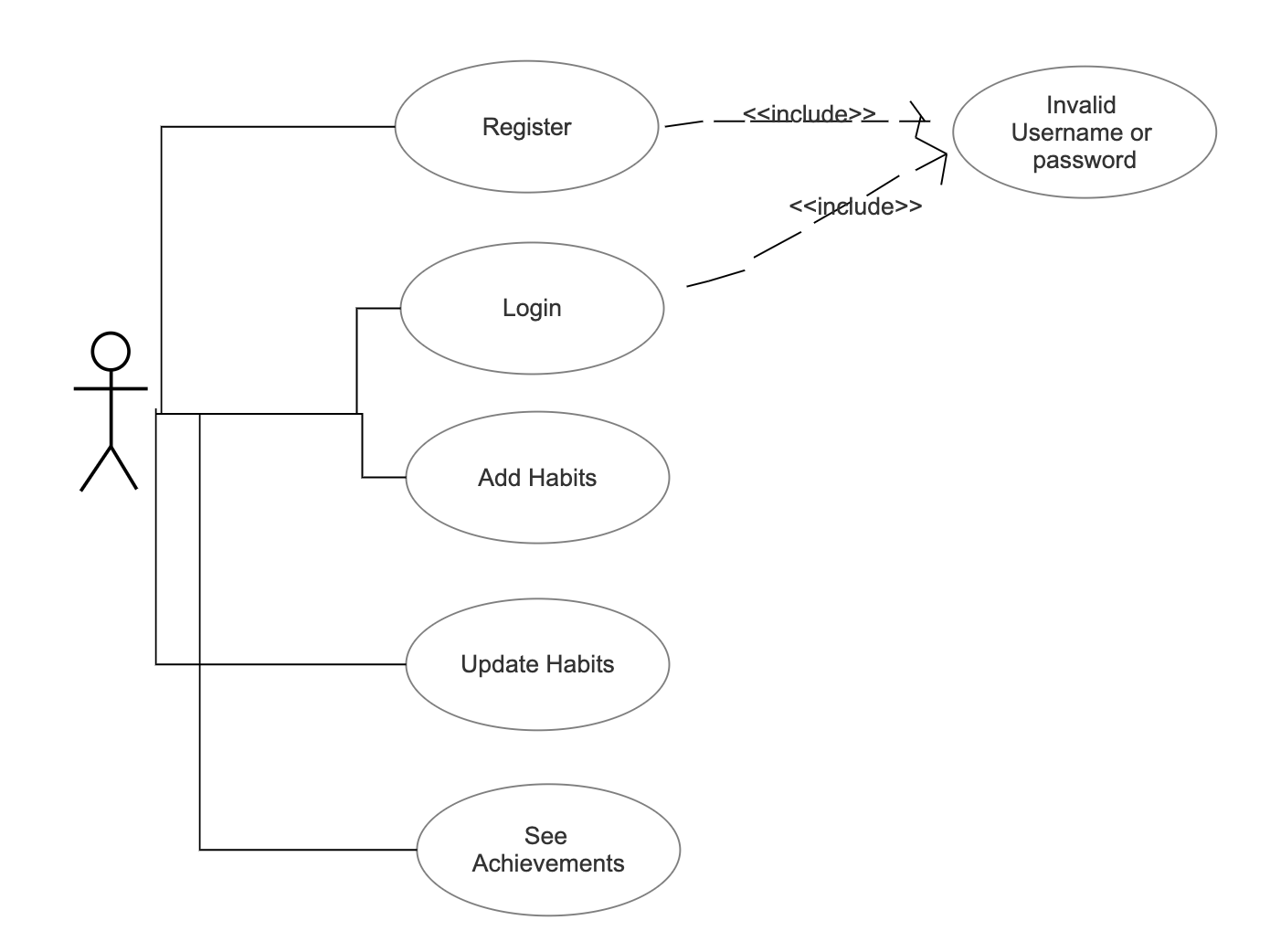
**Packages used**

* tkinter
* PIL
* msilib.schema

**PROPOSED WORK**

**DESIGN –**

USE CASE DIAGRAM –



**Use-Case Descriptions:**

**Use Case ID: UC01**

Name: Register

Actors: User

Description: Allows new user to register for an account

Pre-conditions: None

Post-conditions: An account is created for the user

Main flow:

| **user** | **system** |
| --- | --- |
| 1.Chooses the register option |  |
|  | 2.Prompts the data required for registration |
| 3.Enters the data prompted by the system |  |
|  | 4.Validates user information and creates account if its correct |

**Use Case ID: UC02**

Name: Login

Actors: User

Description: Allows registered users to Login

Pre-conditions: user should be registered with the system

Post-conditions: User logs in and all the options are displayed on the screen

Main Flow:

| **User** | **System** |
| --- | --- |
| Enters username and pw and chooses login option |  |
|  | Validates credentials:  Prompts error if incorrect  Displays the home page if correct |

**Use Case ID: UC03**

Name: Add habits

Actors: User

Description: Allows registered users to select the habits they like.

Pre-conditions: User should be successfully signed-in to the system.

Post-conditions: Habit is created

Main Flow:

| **User** | **System** |
| --- | --- |
| 1.Chooses add Habit |  |
|  | 2.Gives user the habit they choose in list  of habits |

**Use Case ID: UC04**

Name: Update habits

Actors: User

Description: Allows registered users to delete a habits.

Pre-conditions: User should be successfully signed-in to the system, and have the habit present to delete.

Post-conditions: Habit is deleted

Main Flow:

| **user** | **system** |
| --- | --- |
| 1.Chooses update Habit |  |
|  | 2.Displays habits |
| 3.Selects which habit to delete |  |
|  | 4. Deletes the habit |

**Use Case ID: UC05**

Name: See achievements

Actors: User

Description: Shows the user which level they are in.

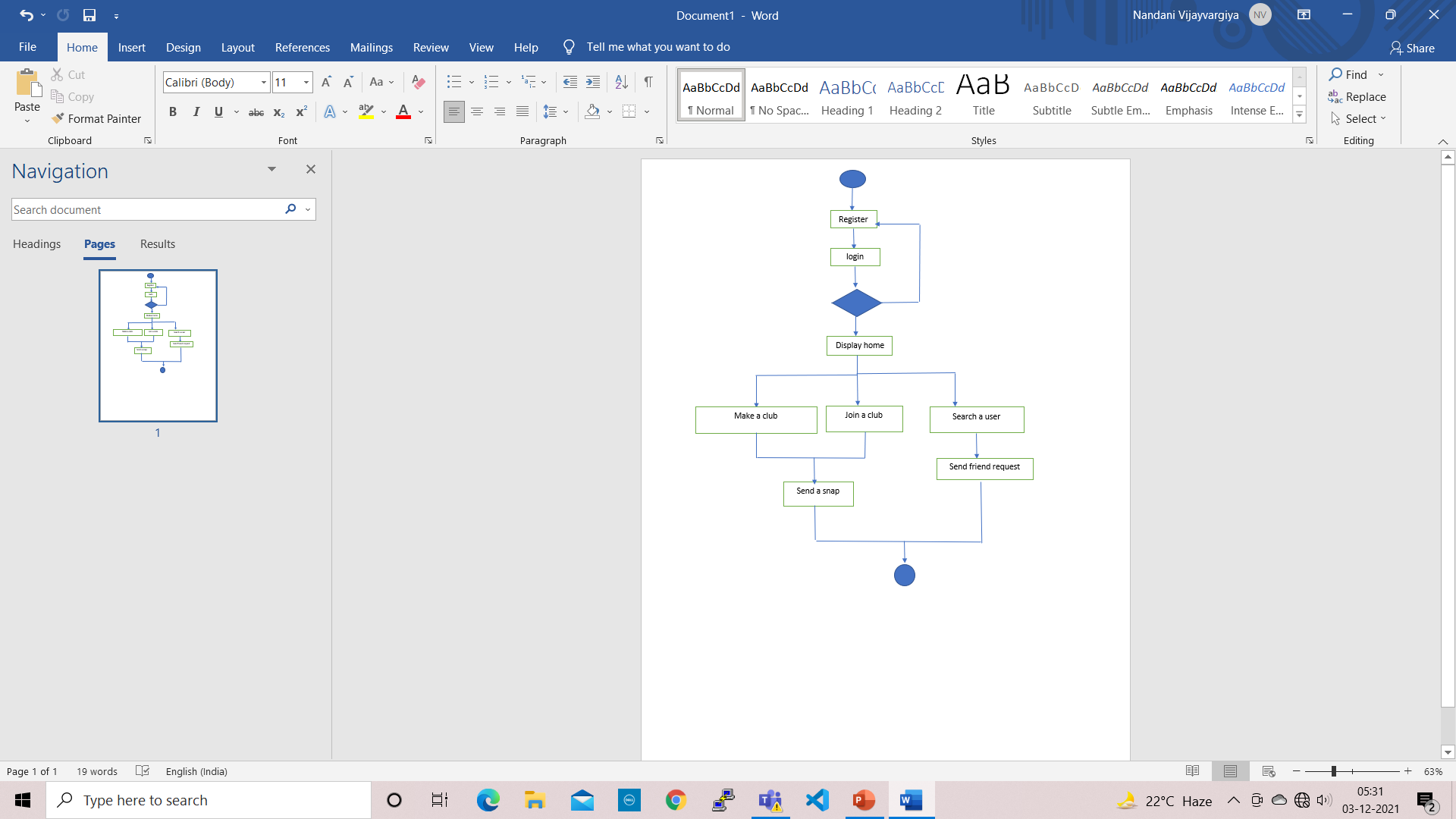
Pre-conditions: User should be successfully signed-in to the system.

Post-conditions: As habits are performed and registered in the app, the user will receive points and will be leveled-up according to the points they have.

Main Flow:

| **user** | **system** |
| --- | --- |
| 1.Completes a habit |  |
|  | 2.Awards points to user |
| 3. Chooses See Achievements |  |
|  | 4. Displays the points the user has |

**ACTIVITY DIAGRAM**

****

**IMPLEMENTATION**

**Source code –**

from msilib.schema import CheckBox

from tkinter import \*

from tkinter import ttk

from PIL import ImageTk

from tkinter import messagebox

import mysql.connector

#from checkbox import isChecked

#from checkbox import isChecked

class Login:

def \_init\_(self,root) :

self.root=root

self.root.title("Login and registration system for apps")

self.root.geometry("1366x700+0+0")

self.root.resizable(False,False)

self.loginform()

def loginform(self):

Frame\_login= Frame(self.root,bg="white")

Frame\_login.place(x=0,y=0,height=700,width=1366)

#self.img= ImageTk.PhotoImage(file="background-2.jpg")

# img=Label(Frame\_login,image=self.img).place(x=0,y=0,width=1366,height=700)

frame\_input=Frame(self.root,bg="white")

frame\_input.place(x=320,y=130,height=450,width=350)

label1=Label(frame\_input,text="Login here",font=('impact',32,'bold'),fg="black",bg="white")

label1.place(x=75,y=20)

label2=Label(frame\_input,text="Username",font=('Goudy old style',32,'bold'),fg="orangered",bg="white")

label2.place(x=30,y=95)

self.email\_txt=Entry(frame\_input,font=('times new roman',15,'bold'),bg="lightgray")

self.email\_txt.place(x=30,y=145,height=35,width=270)

label3=Label(frame\_input,text="Password",font=('Goudy old style',20,'bold'),fg="orangered",bg="white")

label3.place(x=30,y=195)

self.password=Entry(frame\_input,font=('times new roman',15,'bold'),bg="lightgray")

self.password.place(x=30,y=245,height=35,width=270)

btn1= Button(frame\_input,text="forgot password?",cursor='hand2',font=('calibri',10),bg="white",fg="black",bd=0)

btn1.place(x=125,y=305)

btn2= Button(frame\_input,text="Login",command=self.login,cursor='hand2',font=('times new roman',15),bg="white",fg="orangered",bd=0,width=15,height=1)

btn2.place(x=90,y=340)

btn3= Button(frame\_input,text="Not Registered?Register",command=self.Register,cursor='hand2',font=('calibri',10),bg="white",fg="black",bd=0)

btn3.place(x=110,y=390)

def login(self):

if self.email\_txt.get()=="" or self.password.get()=="":

messagebox.showerror("Error","All feilds are required",parent=self.root)

else:

try:

con= mysql.connector.connect(host="localhost",user="Nandani",passwd="MiniProjecthehe1.",database="pythongui")

cur=con.cursor(buffered=True)

cur.execute('select \* from register where emailid=%s and password=%s',(self.email\_txt.get(),self.password.get()))

row=cur.fetchone()

if row==None:

messagebox.showerror('Error','Invalid username and password',parent=self.root)

self.loginclear()

self.email\_txt.focus()

else:

self.appscreen()

con.close()

except Exception as es:

messagebox.showerror('Error',f'Error Due to: {str(es)}',parent=self.root)

def Register(self):

Frame\_login1=Frame(self.root,bg="white")

Frame\_login1.place(x=0,y=0,height=700,width=1366)

frame\_input2=Frame(self.root,bg="white")

frame\_input2.place(x=320,y=130,height=450,width=630)

label1=Label(frame\_input2,text="Register here",font=('impact',32,'bold'),fg="black",bg="white")

label1.place(x=45,y=20)

label2=Label(frame\_input2,text="Username",font=('Goudy old style',20,'bold'),fg="orangered",bg="white")

label2.place(x=30,y=95)

self.entry=Entry(frame\_input2,font=('times new roman',15,'bold'),bg="lightgray")

self.entry.place(x=30,y=145,height=35,width=270)

label3=Label(frame\_input2,text="Password",font=('Goudy old style',20,'bold'),fg="orangered",bg="white")

label3.place(x=30,y=195)

self.entry2=Entry(frame\_input2,font=('times new roman',15,'bold'),bg="lightgray")

self.entry2.place(x=30,y=245,height=35,width=270)

label4=Label(frame\_input2,text="Email id",font=('Goudy old style',20,'bold'),fg="orangered",bg="white")

label4.place(x=330,y=95)

self.entry3=Entry(frame\_input2,font=('times new roman',15,'bold'),bg="lightgray")

self.entry3.place(x=330,y=145,height=35,width=270)

label5=Label(frame\_input2,text="confirm Password",font=('Gody old style',20,'bold'),fg="orangered",bg="white")

label5.place(x=330,y=195)

self.entry4=Entry(frame\_input2,font=('times new roman',15,'bold'),bg="lightgray")

self.entry4.place(x=330,y=245,height=35,width=270)

btn2= Button(frame\_input2,text="Register",command=self.register,cursor='hand2',font=('times new roman',15),bg="white",fg="orangered",bd=0,width=15,height=1)

btn2.place(x=90,y=340)

btn3= Button(frame\_input2,text="Already Registered?login",command=self.loginform,cursor='hand2',font=('calibri',10),bg="white",fg="black",bd=0)

btn3.place(x=110,y=390)

def register(self):

if self.entry.get()==""or self.entry2.get()==""or self.entry3.get()==""or self.entry4.get()=="":

messagebox.showerror("Error","All feilds are required",parent=self.root)

elif self.entry2.get()!=self.entry4.get():

messagebox.showerror("Error","Password and confirm password should be same",parent=self.root)

else:

#try:

con= mysql.connector.connect(host="localhost",user="Nandani",passwd="MiniProjecthehe1.",database="pythongui")

cur=con.cursor(buffered=True)

#cur.execute('select \* from register where emailid=%s ',(self.entry3.get()))

#row=cur.fetchall()

#if row!=None:

#messagebox.showerror("Error","user already exists,please try with another email id",parent=self.root)

#self.regclear()

#self.entry.focus()

#else:

#string= self.entry.get()

cur.execute('INSERT INTO register VALUES("'+str(self.entry.get())+'","'+str(self.entry3.get())+'","'+str(self.entry2.get())+'","'+str(self.entry4.get())+'","'+str(0)+'","'+str(0)+'");')

con.commit()

con.close()

#messagebox.showerror("register successful",parent=self.root)

self.regclear()

#except Exception as es:

#messagebox.showerror("error",f"error due to:{str(es)}",parent=self.root)

def isCheckedYoga(self):

frame\_inputt=Frame(self.root,bg="black")

frame\_inputt.place(x=500,y=500,height=300,width=400)

if self.CheckVar1.get() == 1:

label4=Label(frame\_inputt,text="You achieved a streak in yoga",font=('times new roman',12,'bold'),fg="orangered",bg="white")

label4.place(x=40,y=30)

try:

con= mysql.connector.connect(host="localhost",user="Nandani",passwd="MiniProjecthehe1.",database="pythongui")

cur=con.cursor(buffered=True)

#self.name=self.email\_txt.get()

#cur.execute('update register set YogaStreaks=YogaStreaks+1 where emailid=self.name ;')

cur.execute('update register set YogaStreaks=YogaStreaks+1 where emailid=%s and password=%s',(self.email\_txt.get(),self.password.get()))

con.commit()

#cur.execute("select \* from register")

#for i in cur:

# print(i)

except Exception as es:

messagebox.showerror("error",f"error due to:{str(es)}",parent=self.root)

def isCheckedHomework(self):

frame\_inputt=Frame(self.root,bg="black")

frame\_inputt.place(x=500,y=500,height=300,width=400)

if self.CheckVar2.get() == 1:

label4=Label(frame\_inputt,text="You achieved a streak in homework",font=('times new roman',12,'bold'),fg="orangered",bg="white")

label4.place(x=40,y=30)

try:

con= mysql.connector.connect(host="localhost",user="Nandani",passwd="MiniProjecthehe1.",database="pythongui")

cur=con.cursor(buffered=True)

#cur.execute('update register set HomeworkStreaks=HomeworkStreaks+1 where username="Nandani" ;')

cur.execute('update register set HomeworkStreaks=HomeworkStreaks+1 where emailid=%s and password=%s',(self.email\_txt.get(),self.password.get()))

con.commit()

#cur.execute("select \* from register")

#for i in cur:

# print(i)

except Exception as es:

messagebox.showerror("error",f"error due to:{str(es)}",parent=self.root)

def displayStreaks(self):

con= mysql.connector.connect(host="localhost",user="Nandani",passwd="MiniProjecthehe1.",database="pythongui")

cur=con.cursor(buffered=True)

sql='SELECT YogaStreaks,HomeworkStreaks FROM register where username="Krishna"'

#sql=('SELECT YogaStreaks,HomeworkStreaks FROM register where emailid=%s',(self.email\_txt.get()))

#cur.execute('SELECT YogaStreaks,HomeworkStreaks FROM register where emailid=%s',(self.email\_txt.get()))

cur.execute(sql)

rows=cur.fetchall()

Frame\_loginn=Frame(self.root,bg="white")

Frame\_loginn.place(x=500,y=500,height=450,width=630)

tv=ttk.Treeview(Frame\_loginn,columns=(1,2),show="headings",height=7)

tv.pack()

tv.heading(1,text="YogaStreaks")

tv.heading(2,text="HomeworkStreaks")

for i in rows:

tv.insert('','end',values=i)

def appscreen(self):

Frame\_login=Frame(self.root,bg="white")

Frame\_login.place(x=0,y=0,height=700,width=1366)

label1=Label(Frame\_login,text="Welcome to Habit Tracker",font=('times new roman',32,'bold'),fg="black",bg="white")

label1.place(x=375,y=100)

label2=Label(Frame\_login,text="Consistency is the key",font=('times new roman',12,'bold'),fg="black",bg="white")

label2.place(x=700,y=160)

label3=Label(Frame\_login,text="Select the tasks performed",font=('times new roman',25,'bold'),fg="black",bg="white")

label3.place(x=340,y=220)

btn2= Button(Frame\_login,text="log out",command=self.loginform,cursor='hand2',font=('times new roman',15),bg="white",fg="orangered",bd=0,width=15,height=1)

btn2.place(x=1000,y=10)

self.CheckVar1 = BooleanVar()

self.CheckVar2 = BooleanVar()

Checkbutton(Frame\_login, text="Yoga", variable=self.CheckVar1, onvalue=1, offvalue=0, command=self.isCheckedYoga).place(x=400,y=500)

Checkbutton(Frame\_login, text="Homework", variable=self.CheckVar2, onvalue=1, offvalue=0, command=self.isCheckedHomework).place(x=400,y=550)

btn3= Button(Frame\_login,text="Check Streaks",command=self.displayStreaks,cursor='hand2',font=('times new roman',15),bg="white",fg="orangered",bd=0,width=15,height=1)

btn3.place(x=900,y=350)

def regclear(self):

self.entry.delete(0,END)

self.entry2.delete(0,END)

self.entry3.delete(0,END)

self.entry4.delete(0,END)

def loginclear(self):

self.email\_txt.delete(0,END)

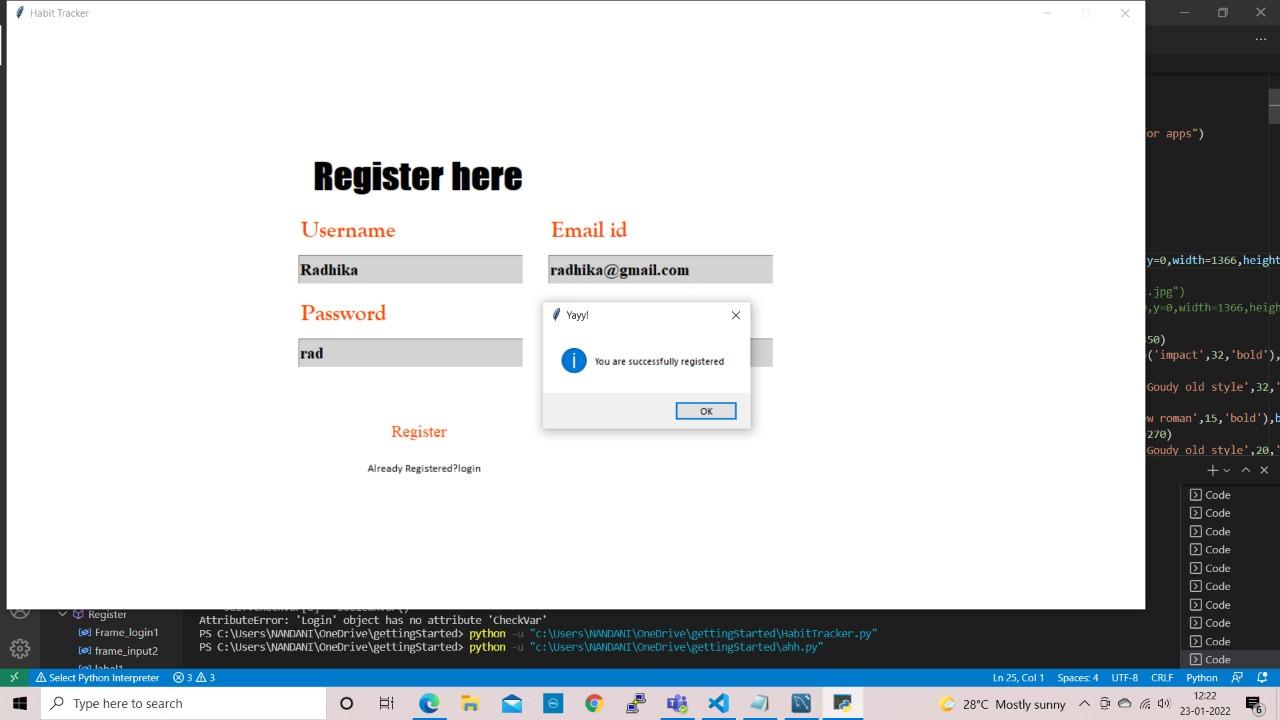
self.password.delete(0,END)

root=Tk()

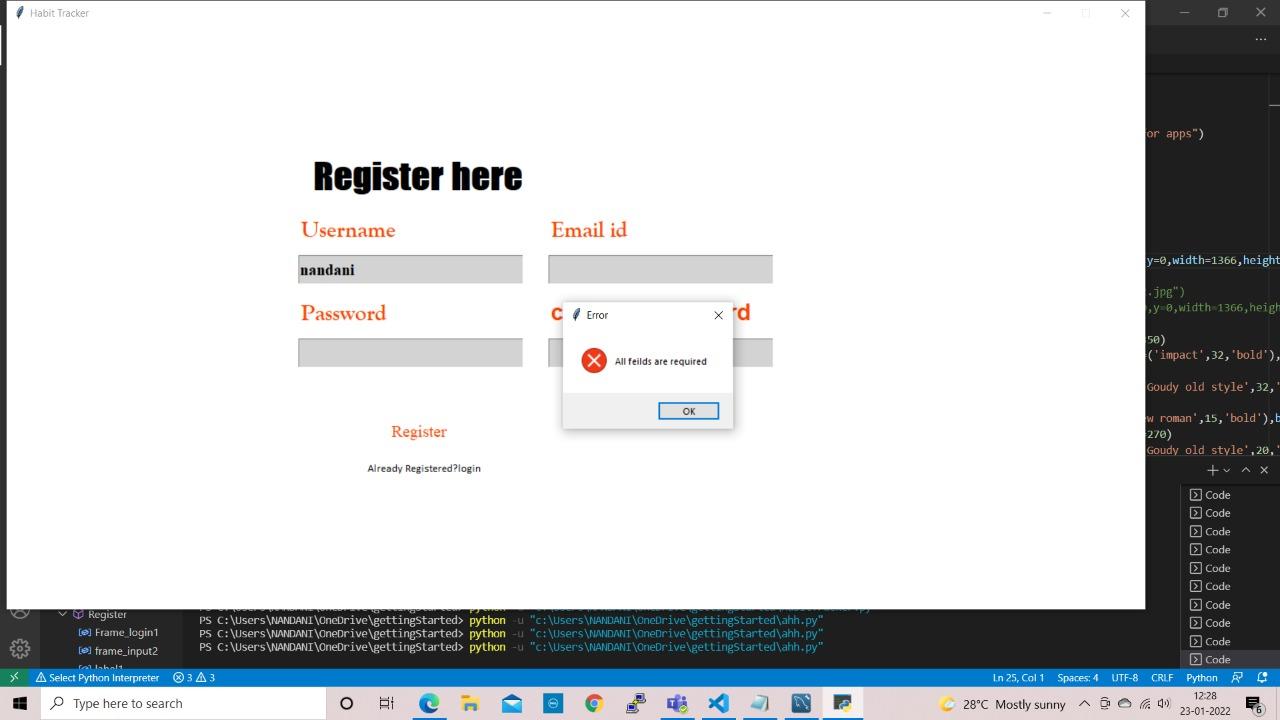
ob=Login(root)

root.mainloop()

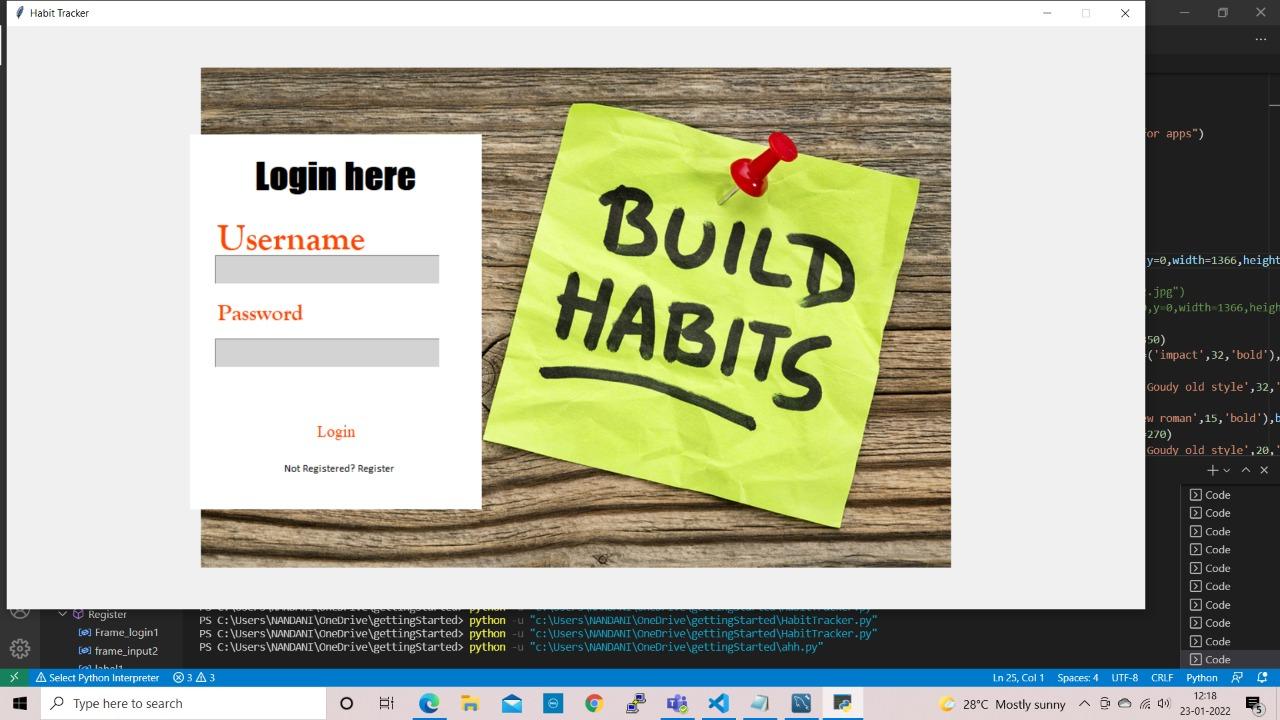
**SCREENSHOTS OF APPLICATION TESTCASES –**

****

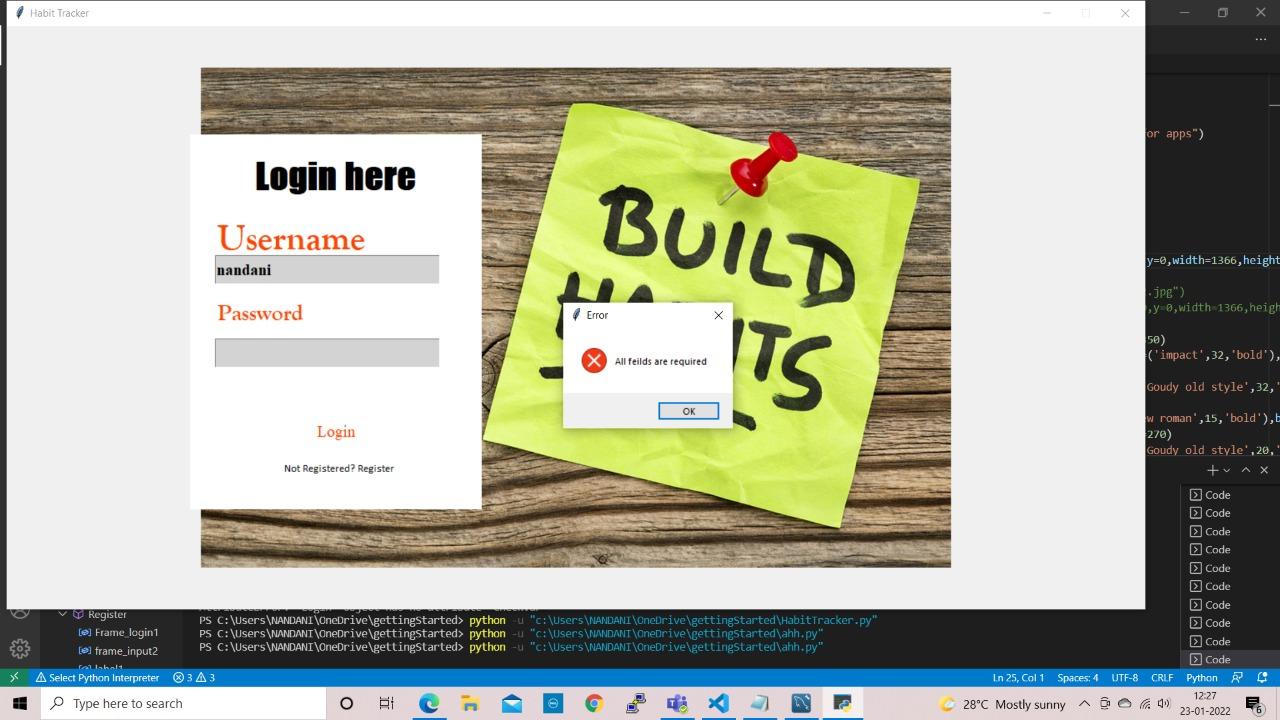
**This is the register page, the user should enter username, email id, password and confirm password to register.**

****

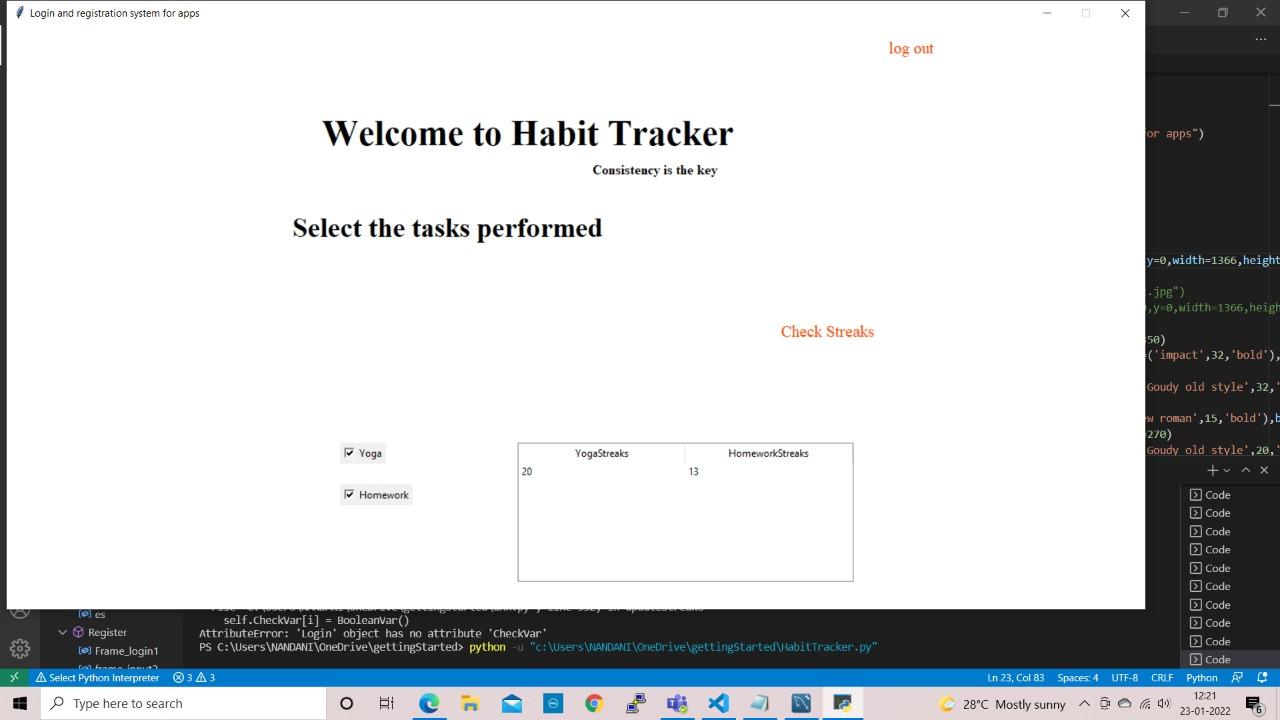
**In case the user did not enter all the fields, the system will show an error says “All fields are required”.**

****

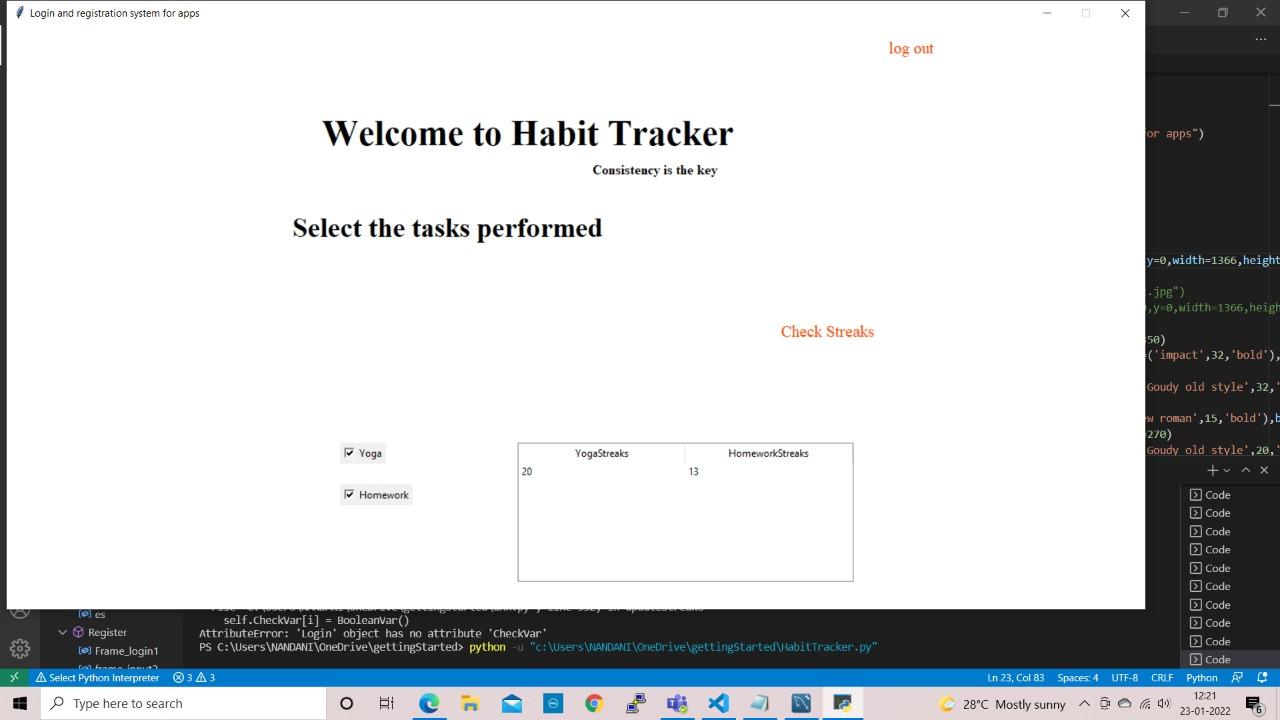
**This is the login page.**

****

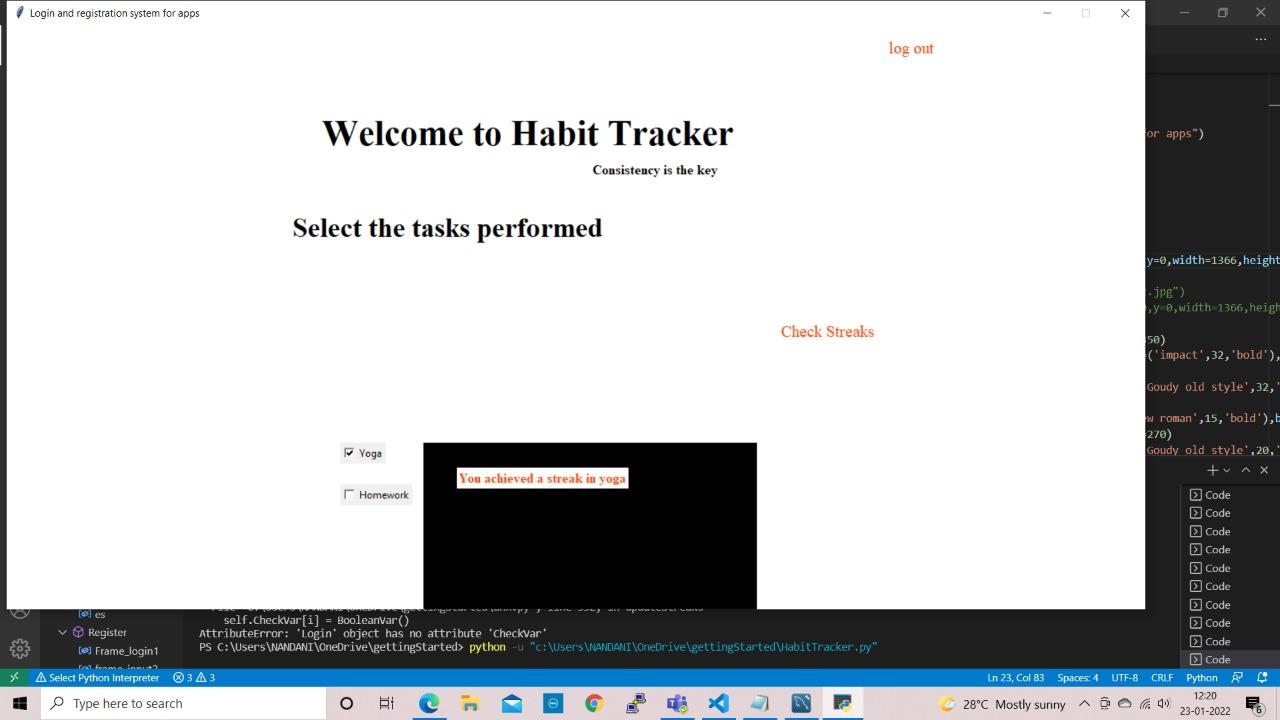
**In case the user does not enter all the fields, the system will show an error saying “All fields required”.**

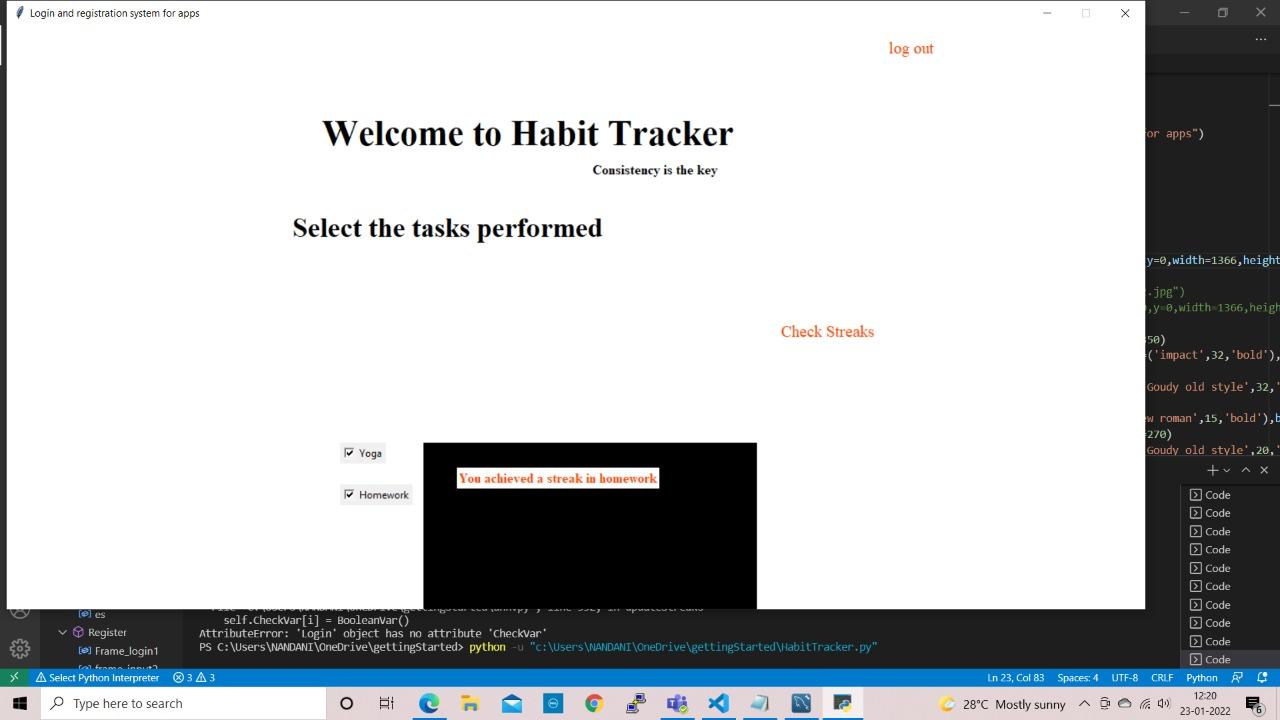
****

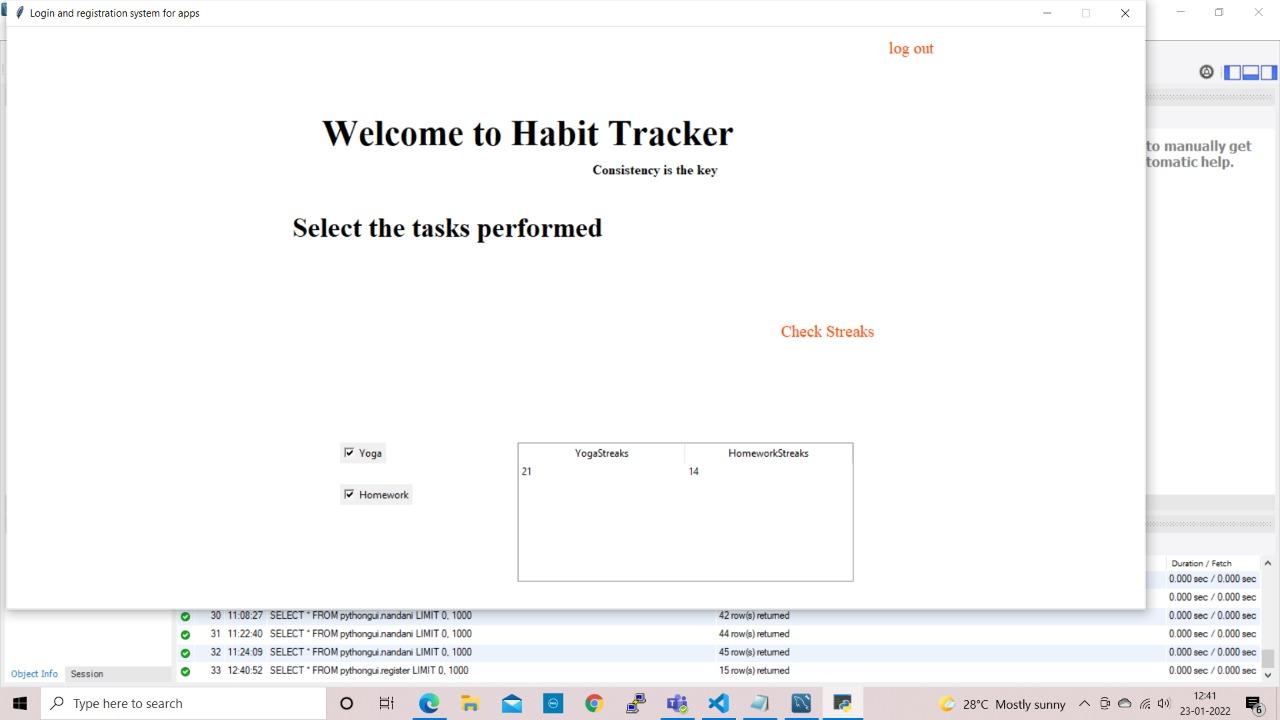
**Once the user is successfully logged-in, the home page is displayed to the user.The user is displayed with the habits he has, the user can update the habits if he has performed the habits.**

****

**Once the habits is updated, the used can click “Check Streaks” and check the streaks of all the habits the user has.**

****

**The message “You achieved a streak in (habit)” will be displayed once the habit is checked.**

****

**After performing the habits, by clicking the “Check Streaks” again, the updated streaks will be shown.**

**Github link**

**https://github.com/nandaniv**

**ADDITIONAL KNOWLEDGE ACQUIRED**

* Python GUI programming using the module tkinter.
* mysql to store and retrieve data.
* Use case diagrams.
* Activity diagram.

**CONCLUSION –**

Habit Tracker is a simple but useful application that is used to keep track of habits. It is a very basic code that provides a loads of value and is helpful.

**FUTURE WORK –**

* The user can connect through his Facebook account and make habits with his friends.
* Make it a web application rather than a console one.
* Improve the GUI and include the feature to send photos/snaps.
* To include the data on to a database and make it a kind of a mobile app.
* To include a calendar and show when the habits a done and when they are skipped.

**REFERENCES –**

Basic Python

1. Course covered during 2nd semester by Dr. Ramesh Vassapanavara Sir.
2. Ppts and handouts provided by the sir.
3. Python Programming - Using Problem Solving Approach First Edition by Reema Thareja.

Tkinter tutorials

1.<https://www.youtube.com/watch?v=YXPyB4XeYLA>

2. <https://www.youtube.com/watch?v=VMP1oQOxfM0&t=271s>

3. <https://www.youtube.com/watch?v=NkAwxoQkdOA>

Use cases and activity diagram

1. Hand-outs provided by DRL PRASANNA ma’am.
2. <https://www.youtube.com/watch?v=zid-MVo7M-E>
3. <https://www.youtube.com/watch?v=knM8BGY9yVI&t=161s>