MINI PROJECT REPORT

On

Secure Internet

Submitted by

Akshay Dubey (171500029)
Himanshu Pathak (171500136)
Ganesh Pandey(171500106)
Saurabh Singhal(171500304)

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University
Mathura,U.P.-281406, INDIA
2019

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

Declaration

I hereby declare that the work which is being presented in the Mini Project "**Secure Internet",** in fulfillment of the requirements for Mini Project viva voce, is an authentic record of our own work carried under the supervision of Technical Trainer "**Dr. Manoj Varshney**".

Team Members: Akshay Dubey (171500029)

Himanshu Pathak (171500136)

Ganesh Pandey (171500106)

Saurabh Singhal (171500304)

Course: B.tech. (Computer Science and Engineering)

Year: 3rd

Semester: 5th

Acknowledgement

We have taken efforts in this project "**Secure Internet**". However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Dr. Manoj Varshney** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our special gratitude and thanks to industry persons for giving me such attention and time.

Our thanks and appreciations also go to my colleague in developing the project and people who have willingly helped us out with their abilities.

Abstract

Web Content Filtering range from wanting to improve internal productivity, keep users off of dangerous and illegal sites or restrict sites based on the morals of those in charge of the organization. In general, the need to block content for whatever reason, can be met with Web Content Filtering.

Secure Internet can be used as a parental control solution that helps parents protect their children and monitor their Internet use. This can also be used in corporate and educational environment to reduce the Internet Slacking. An Internet content filtering tool gives organizations the ability to prevent web users visiting online destinations that may harbor hidden security risks or unacceptable content. As a precaution against the inadvertent downloading of malware, adware, and ransomware, tools for filtering Internet content are becoming necessary elements of an organization´s cybersecurity defenses.

After making this tool parents and teachers are able to restrict their children and students respectively from accessing unauthorized or illegal websites that makes the bad impact on their mindset.

We can also use this tool in an organization that will restrict their employees from using the particular websites like social networking sites during work hours, that will increase their productivity and reduce Internet Slacking.

An Internet filter tool can also increase workplace productivity and avoid potential HR issues. These goals are achieved by blocking access to personal social media accounts, online shopping portals, sites containing pornographic material and video streaming websites.

Contents

Chapter 1

Introduction

Introduction to Secure Internet

Secure Internet is a website filtering tool. A Web filter is a program that can screen an incoming Web page to determine whether some or all of it should not be displayed to the user. The filter checks the origin or content of a Web page against a set of rules provided by company or person who has installed the Web filter. A Web filter allows an enterprise or individual user to block out pages from Web sites that are likely to include objectionable advertising, pornographic content, spyware, viruses, and other objectionable content. Vendors of Web filters claim that their products will reduce recreational Internet surfing among employees and secure networks from Web-based threats.

Some Web filter products also provide reporting so that the installer can see what kind of traffic is being filtered and who has requested it. Some products provide soft blocking (in which a warning page is sent to the user instead of the requested page while still allowing access to the page) and an override capability that allows an administrator to unlock a page.

While a Web filter can screen out a certain amount of malware, security experts advise other forms of protection as well, such as the installation of desktop and network antivirus softwares. A Web filter is often installed as part of a proxy server and firewall.

Secure Internet can be used as a parental control solution that helps parents protect their children and monitor their Internet use. This can also be used in corporate and educational environment to reduce the Internet Slacking. An Internet content filtering tool gives organizations the ability to prevent web users visiting online destinations that may harbor hidden security risks or unacceptable content. As a precaution against the inadvertent downloading of malware, adware, and ransomware, tools for filtering Internet content are becoming necessary elements of an organization´s cybersecurity defenses.

Types of Filtering used :

We have used two types of web filtering that are as follows :

Particular Website Filtering:- In this filtering, user enters the website name that user want to block. After entering the website, the website is stored into the database. After the website stored into database, it is accessed and written to the hosts file, so that it will be redirected to the localhost. Hence, website will be blocked and website will not open in the browser. In this only the websites user wanted to block are only blocked. User can enter multiple websites and it can block multiple websites.

Categories Wise Filtering:- In this filtering, user selects the category that he/she wants to block. After that we check the keywords of the website and if the keywords matches the category, the website name will be written to the hosts file, so that it will be redirected to the localhost. Hence, website will be blocked and website will not open in the browser.In this user can select multiple categories that he/she wants to block.

Pre-requisites

Hands-on knowledge of tkinter(For User Interface) and pymysql(Dataset) and it also uses SQL server(For Database) is essential before working on Secure Internet. Make sure that you have the following packages(bs4,pywinauto,pymysql) installed and running before implementing it.

Chapter 2

Software Requirement

Introduction to Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).It is used for web development, software development,system scripting.

Advantages of Python :

1. Presence of third-party modules
2. Extensive support libraries
3. Open source

Applications of Python :

1. GUI based desktop applications(Games, Scientific Applications)
2. Web frameworks and applications
3. Enterprise and Business applications
4. Language Development

Introduction to SQL

Structured Query language (SQL) pronounced as "S-Q-L" or sometimes as "See-Quel". SQL is a standard language for accessing and manipulating databases. SQL programming can be effectively used to insert, search, update, delete database records.

Introduction to PyMySQL

PyMySQL is a pure-Python MySQL client library, based on PEP 249. Most public APIs are compatible with mysqlclient and MySQLdb. PyMySQL works with MySQL 5.5+ and MariaDB 5.5+.

MySQL is a leading open source database management system. It is a multiuser, multithreaded database management system. MySQL is especially popular on the web.

Introduction to Beautiful Soup

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping. It commonly saves programmers hours or days of work.

Introduction to ImageTk

The ImageTk module contains support to create and modify Tkinter BitmapImage and PhotoImage objects from PIL images.

Introduction to Time

Python has a module named time to handle time-related tasks. To use functions defined in the module, we need to import the module first.

The sleep() function suspends (delays) execution of the current thread for the given number of seconds.

Introduction to Tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter:

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

There are two main methods used you the user need to remember while creating the Python application with GUI:-

1. Tk(): To create a main window, tkinter offers a method ‘Tk()’. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

   m=tkinter.Tk() where m is the name of the main window object

2. mainloop(): There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

   m.mainloop()

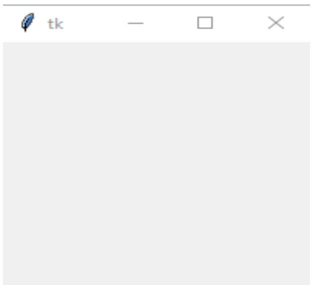


Fig 2.1 Window appears on executing mainloop()

There are mainly three geometry manager classes class:-

1. **pack() method**:It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method**:It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method**:It organizes the widgets by placing them on specific positions directed by the programmer.

There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

1. **Button**:To add a button in your application, this widget is used.
   The general syntax is:
   w=Button(master, option=value)
   master is the parameter used to represent the parent window.

Parameters:

1. activebackground: to set the background color when button is under the cursor.
2. activeforeground: to set the foreground color when button is under the cursor.
3. bg: to set he normal background color.
4. command: to call a function.
5. font: to set the font on the button label.
6. image: to set the image on the button.
7. width: to set the width of the button.
8. height: to set the height of the button.

2. **Entry**:It is used to input the single line text entry from the user.. For multi-line text input, Text widget is used.The general syntax is:
   w=Entry(master, option=value)

Parameters:

1. bd: to set the border width in pixels.
2. bg: to set the normal background color.
3. cursor: to set the cursor used.
4. width: to set the width of the button.
5. height: to set the height of the button.

3. **Label**: It refers to the display box where you can put any text or image which can be updated any time as per the code.The general syntax is: w=Label(master, option=value)

Parameters:

1. bg: to set he normal background color.
2. bg to set he normal background color.
3. font: to set the font on the button label.
4. image: to set the image on the button.
5. width: to set the width of the button.
6. height: to set the height of the button.

4. **Text**: To edit a multi-line text and format the way it has to be displayed.The general syntax is: w =Text(master, option=value)

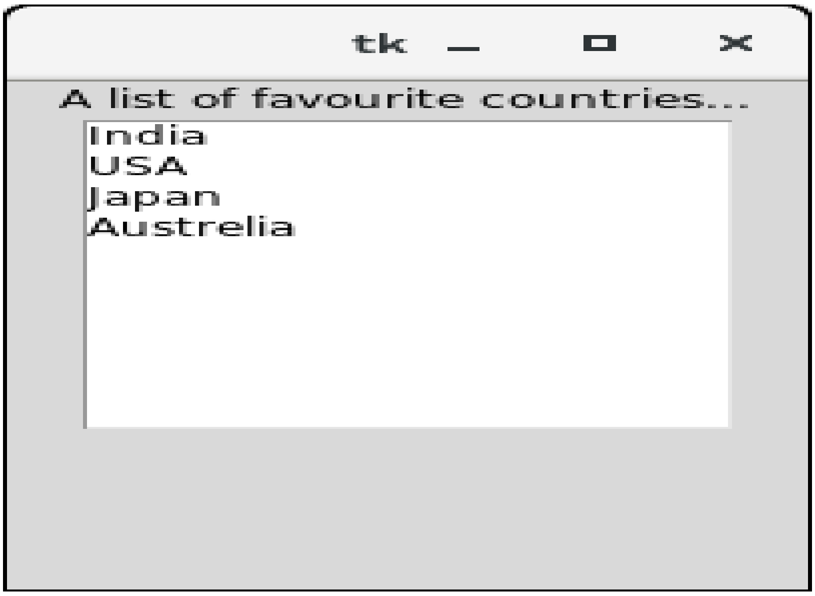
Parameters:

1. highlightcolor: To set the color of the focus highlight when widget has to be focused.
2. insertbackground: To set the background of the widget.
3. bg: to set he normal background color.
4. font: to set the font on the button label.
5. image: to set the image on the widget.
6. width: to set the width of the widget.
7. height: to set the height of the widget.

5. **Listbox**: It offers a list to the user from which the user can accept any number of options.The general syntax is: w = Listbox(master, option=value)

Parameters:

1. highlightcolor: To set the color of the focus highlight when widget has to be focused.
2. bg: to set he normal background color.
3. bd: to set the border width in pixels.
4. font: to set the font on the button label.
5. image: to set the image on the widget.
6. width: to set the width of the widget.
7. height: to set the height of the widget.



A list of favourite countries...

India
USA
Japan
Austrelia

Fig 2.2:-Listbox showing the list items

6. **Frame:** The Frame widget is very important for the process of grouping and organizing other widgets in a somehow friendly way. It works like a container, which is responsible for arranging the position of other widgets.
   It uses rectangular areas in the screen to organize the layout and to provide padding of these widgets. A frame can also be used as a foundation class to implement complex widgets.The general syntax is:Frame(master,options…..)

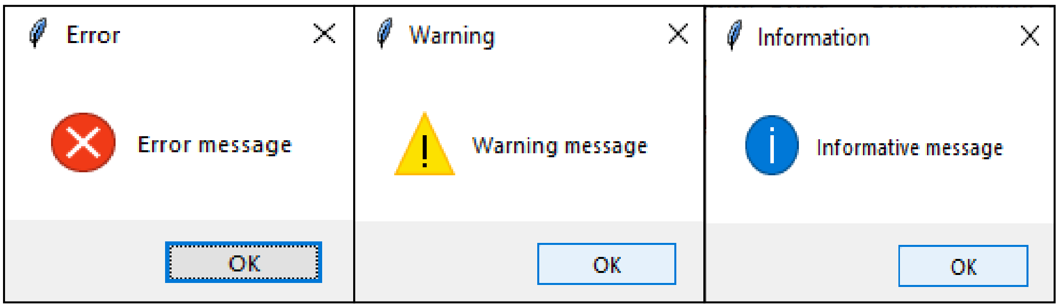
Parameters:

1. master – This represents the parent window.
2. options – Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.Options are bg, bd, cursor, height etc.

Messagebox module:- To show a minimalistic Tkinter message box, use the function showinfo() and showerror() where the parameters are the window title and text.

Ex:-

```
import tkinter
from tkinter import messagebox
root = tkinter.Tk()
root.withdraw()
messagebox.showerror("Error", "Error message")
messagebox.showwarning("Warning","Warning message")
messagebox.showinfo("Information","Informative message")
```



Error

Error message

Warning

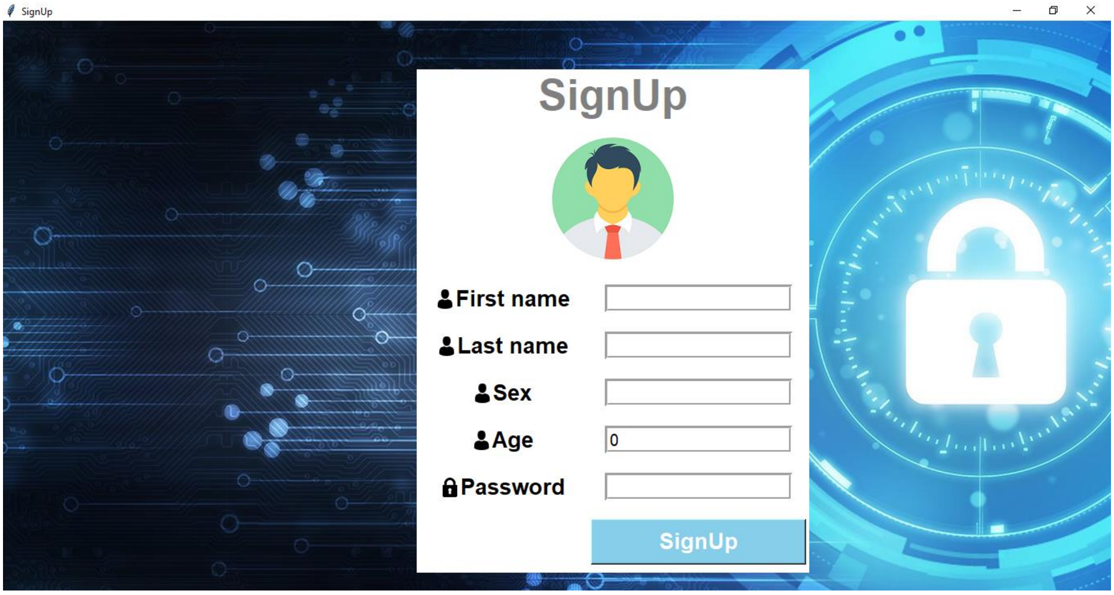
Warning message

Information

Informative message

Fig 2.3 All 3 types of message boxes

Chapter 3
Implementation and User Interface



First name

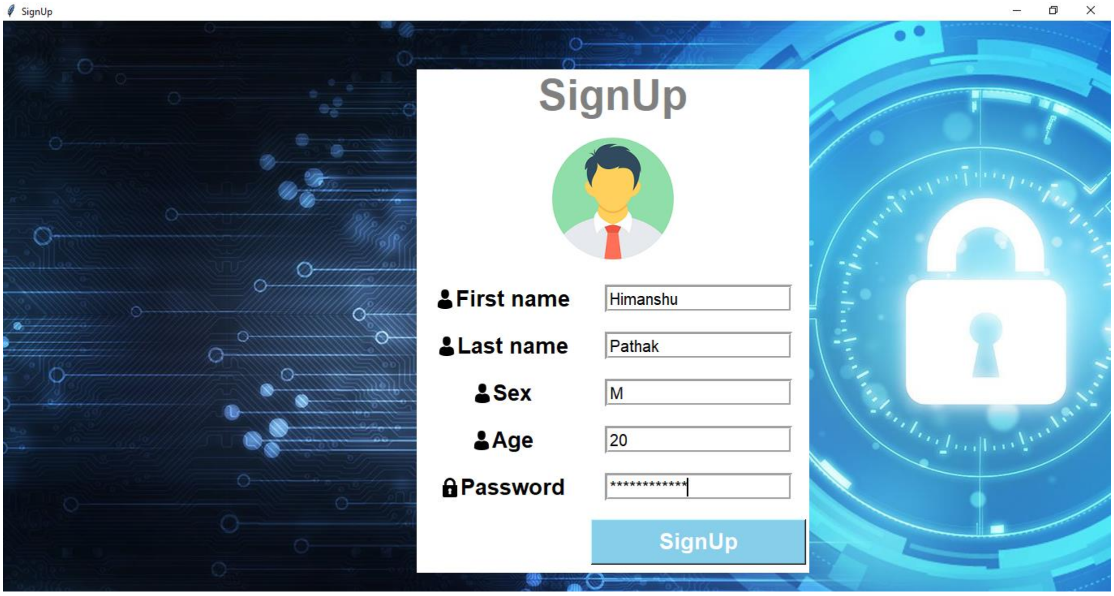
Last name

Sex

Age 0

Password

Fig. 3.1 Signup Window



First name Himanshu

Last name Pathak

Sex M

Age 20

Password ************

Fig. 3.2 Signup Window with filled entries

Here you can see that user filled all the entries and clicks the Signup button he/she will be signed up successfully and redirected to login window.

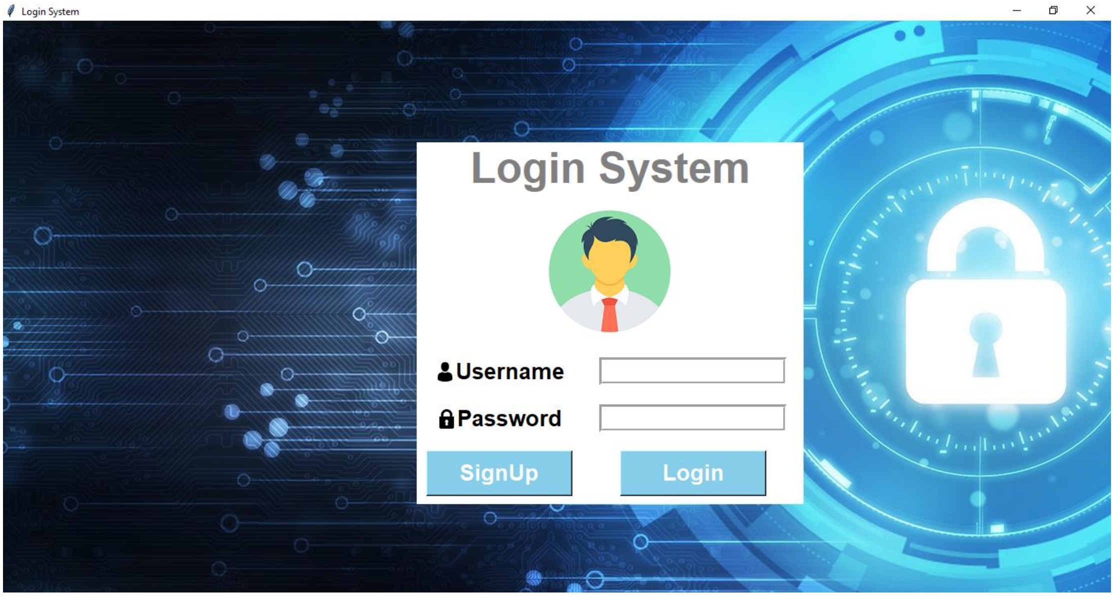
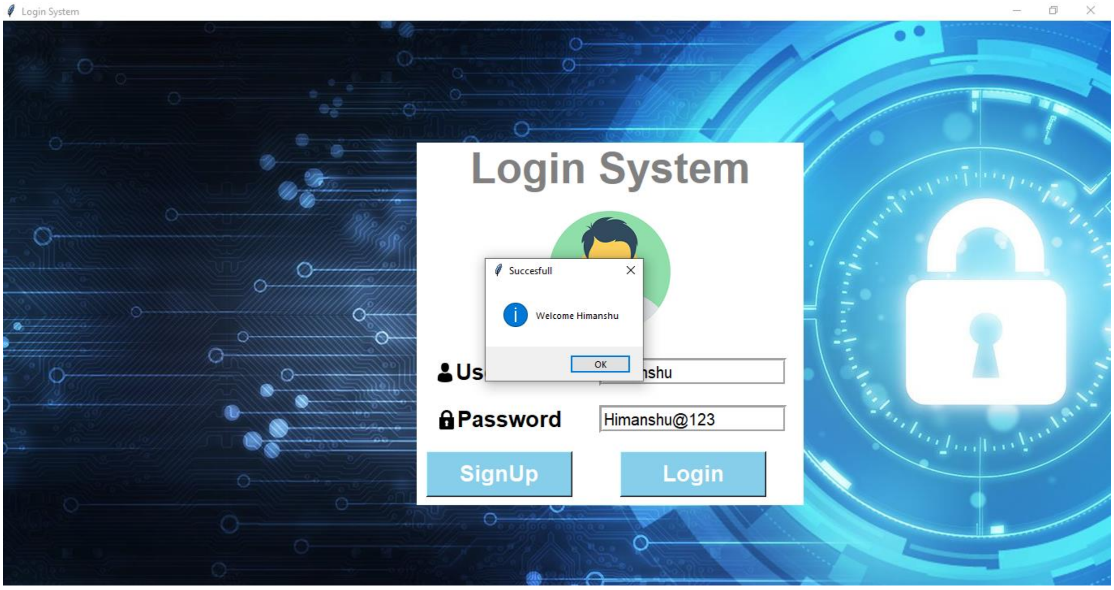


Fig. 3.3 Login Window

If the user is not signed up he click the Signup button to go to signup window.



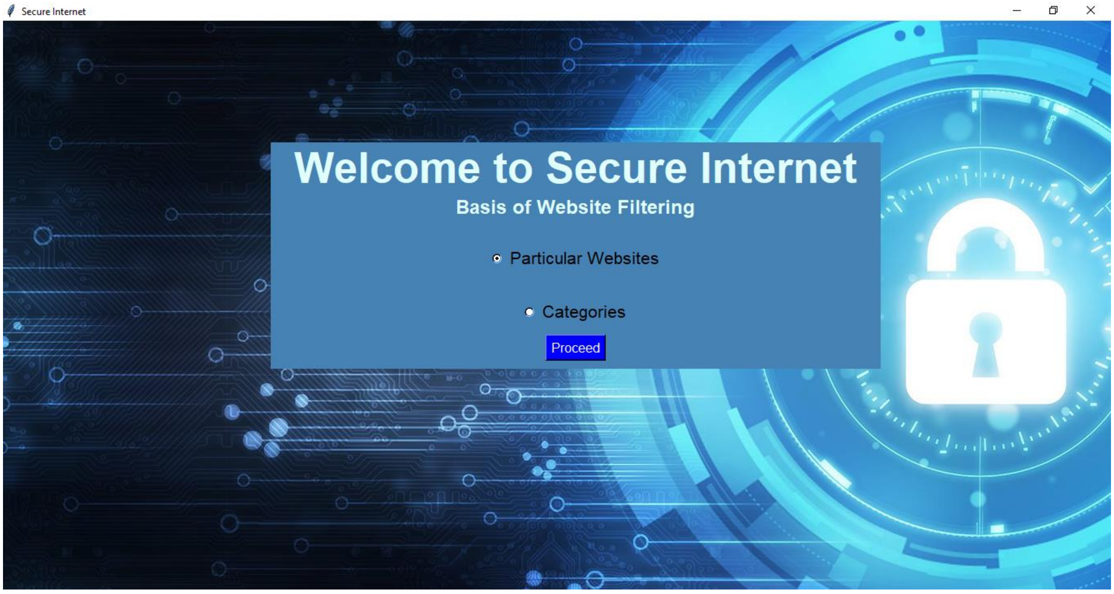
Welcome Himanshu

nshu

Himanshu@123

Fig. 3.4 User Logged-In

Here you can see that the user successfully logged in after filling the all details.



Welcome to Secure Internet

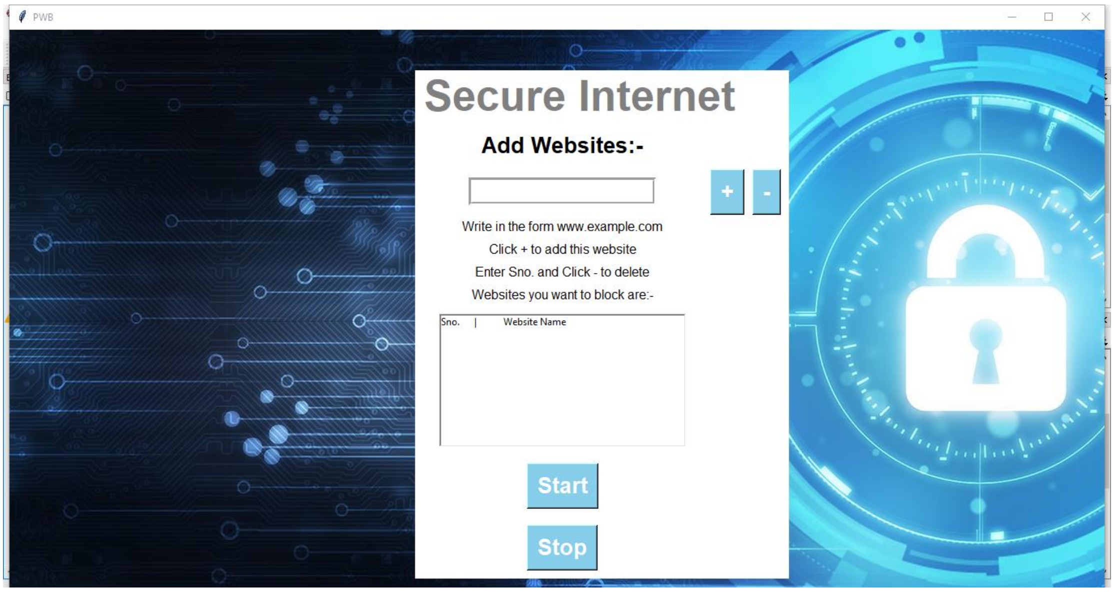
Basis of Website Filtering

● Particular Websites

○ Categories

Fig. 3.5 Selection Window of Website Filtering

In this window user can select the basis of website filtering that are specified above. After selecting the basis he/she can click Proceed button to move forward.



Write in the form www.example.com

Click + to add this website

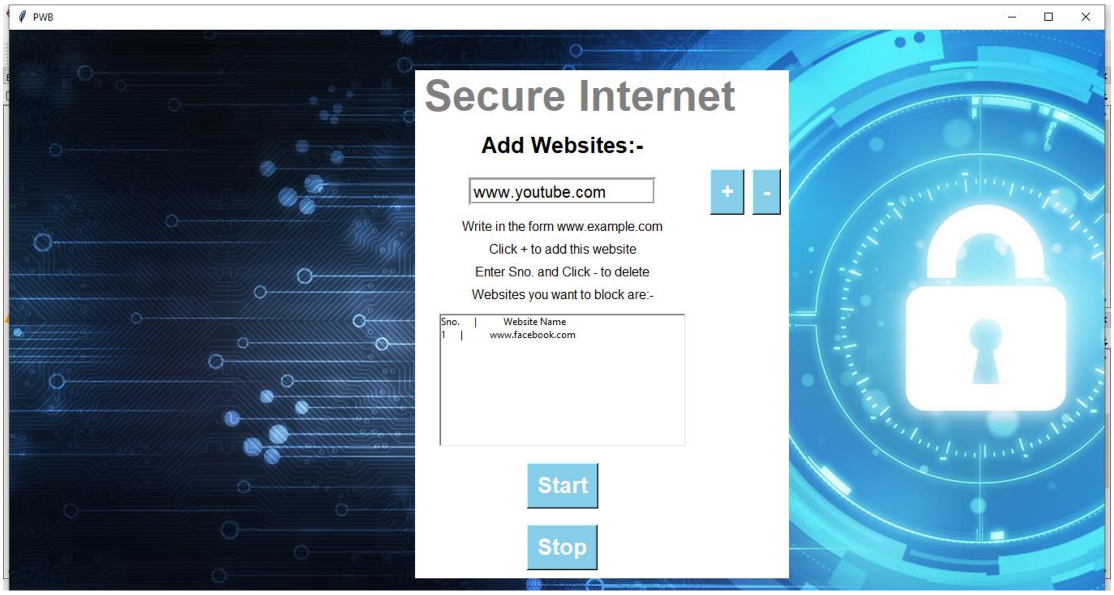
Enter Sno. and Click - to delete

Websites you want to block are:-

Sno.	Website Name

Fig. 3.6 Particular website adding and blocking window

User has selected the particular website option in last step so this window is popped up.



www.youtube.com

Write in the form www.example.com

Click + to add this website

Enter Sno. and Click - to delete

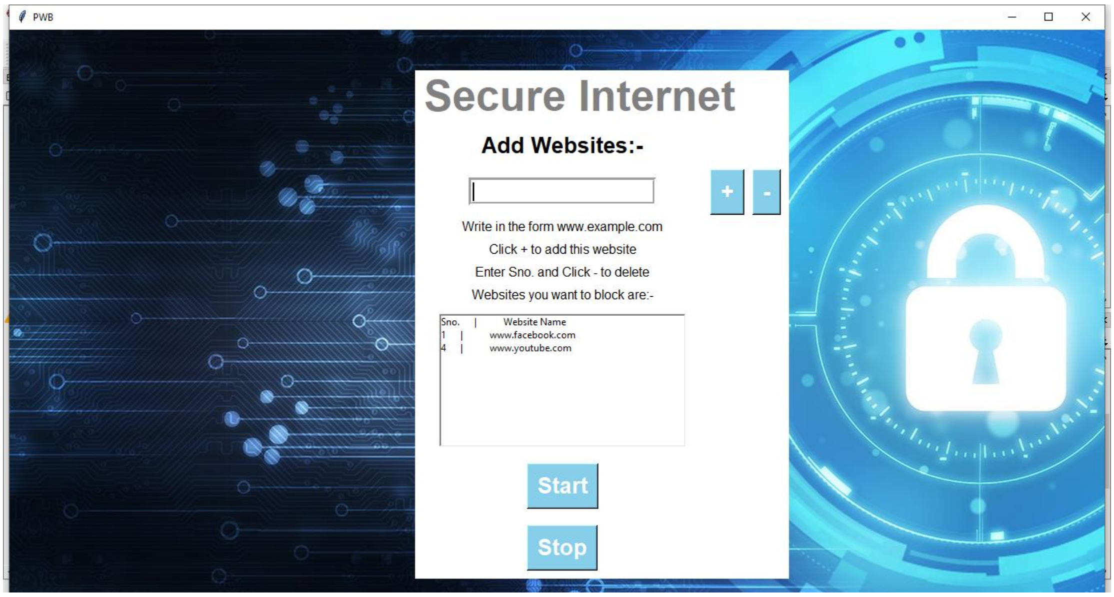
Websites you want to block are:-

```
Sno.    |     Website Name
1    |     www.facebook.com
```

Fig. 3.7 User enters the website

Here user is entering the website name which s/he wants to block.

When he clicks on + button , entered website will be added to the database.



Write in the form www.example.com

Click + to add this website

Enter Sno. and Click - to delete

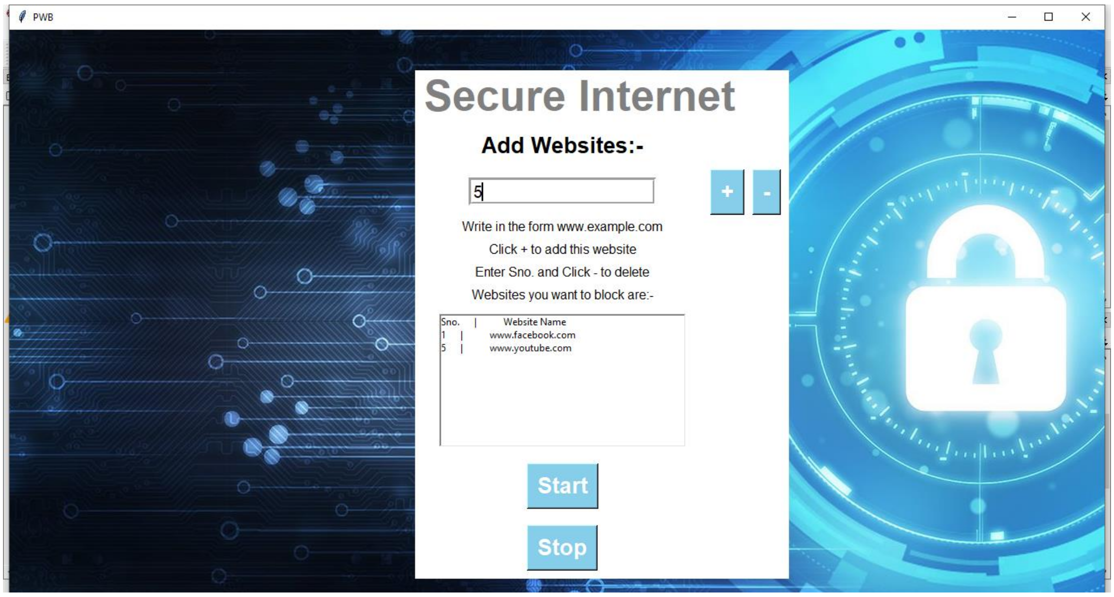
Websites you want to block are:-

```
Sno.    |     Website Name
1    |     www.facebook.com
4    |     www.youtube.com
```

Fig. 3.8 Website added successfully

All the websites entered by the user will be shown in the list box.

When he clicks the Start button all the websites shown in the list box get blocked.



Secure Internet

Add Websites:-

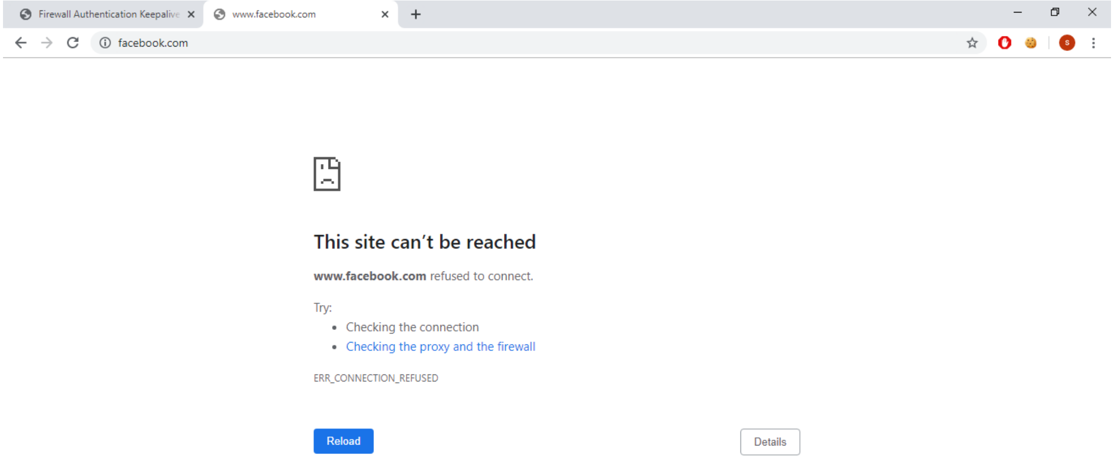
5

Write in the form www.example.com
Click + to add this website
Enter Sno. and Click - to delete
Websites you want to block are:-

```
Sno.   |      Website Name
1      |      www.facebook.com
5      |      www.youtube.com
```

Fig. 3.9 Deleting the website

Here user is entering the serial number of the website which he wants to delete from database. After giving the serial number when s/he clicks the – button the website will be deleted.



This site can’t be reached

www.facebook.com refused to connect.

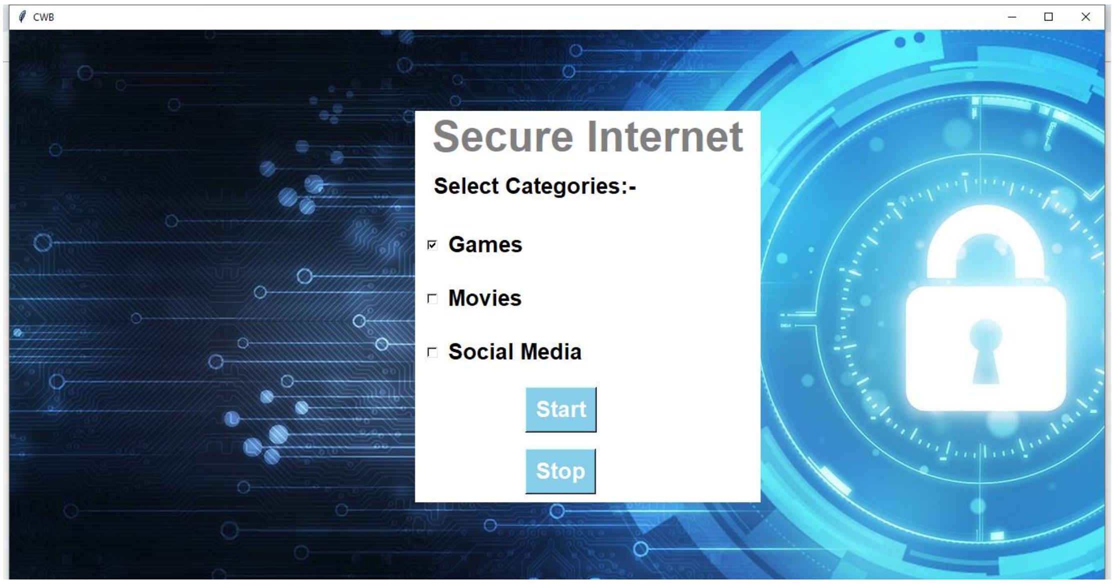
Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Fig. 3.10 Website successfully blocked

Here you can see that entered website is successfully blocked.



Secure Internet

Select Categories:-

☑ Games

☐ Movies

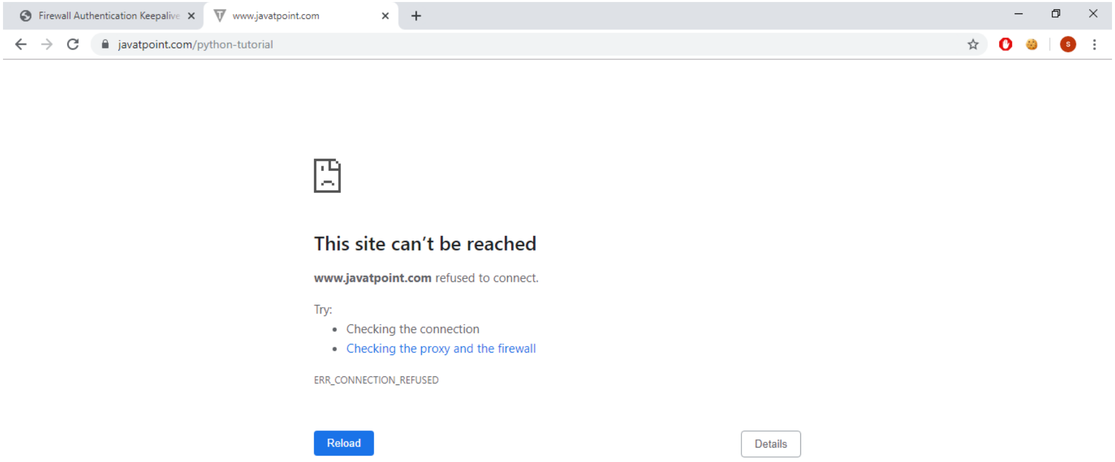
☐ Social Media

Fig. 3.11 Categories wise blocking window

This is the second option of the basis of the website filtering.

Here user can select the category/ categories which he/she wants to block.

After selecting the category/ categories click on start button to start blocking.


Firewall Authentication Keepalive
www.javatpoint.com
javatpoint.com/python-tutorial

This site can't be reached

www.javatpoint.com refused to connect.

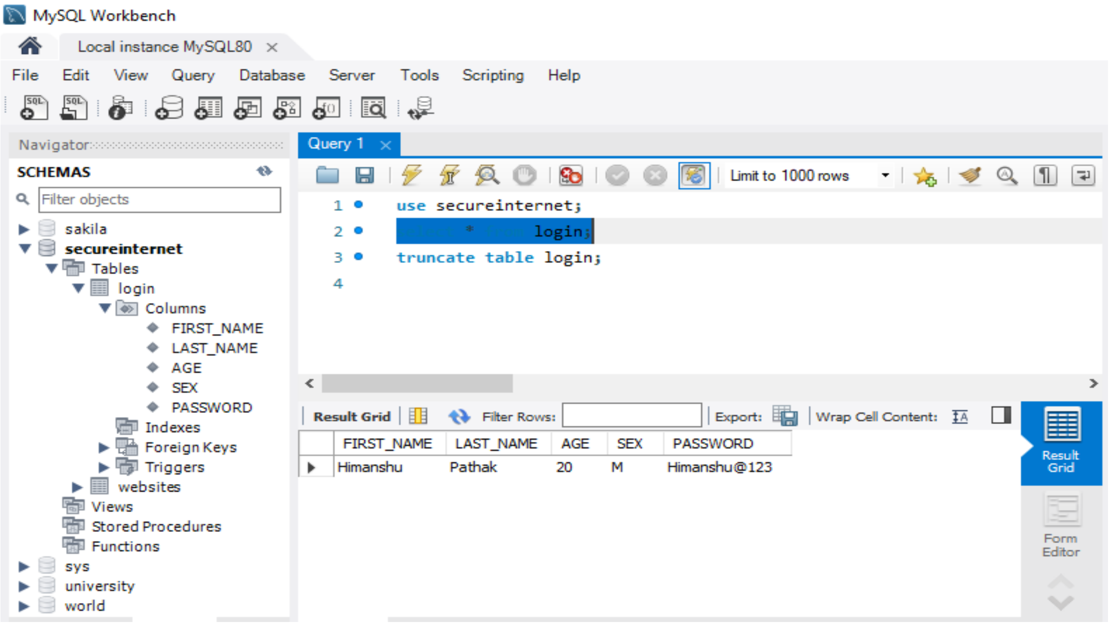
Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Fig. 3.12 Website blocked successfully

Here you can see that entered category of website is successfully blocked.

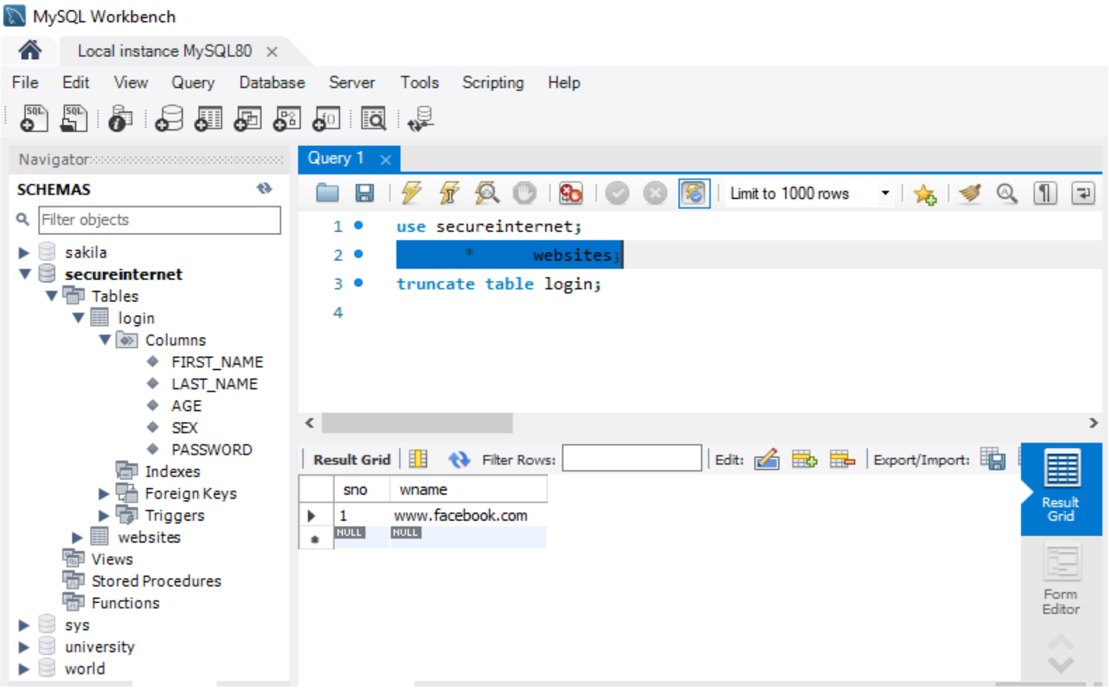


```
use secureinternet;
        *        login;
truncate table login;
```

FIRST_NAME	LAST_NAME	AGE	SEX	PASSWORD
Himanshu	Pathak	20	M	Himanshu@123

Fig. 3.13 Login database

This is the Database of the all users that signed up using signup window.



```
use secureinternet;
        *        websites;
truncate table login;
```

sno	wname
1	www.facebook.com
NULL	NULL

Fig. 3.14 Website database

This is the database of the website entered by user during particular website blocking.

Chapter 4

Conclusion

As title suggests the aim of project is to make browsing internet secure. This can be achieved by blocking the content or websites which are illegal, unethical, unwanted, malicious and unauthorized. Blocking websites is not a easy task. This can be done by checking the head content in the source code of website.

We will check the source code of website using beautifulsoup library in python and analyze that whether the content of website is legal or not. If the website does not contain any illegal content then we will allow user to freely surf it,

Otherwise we will redirect to our designed alertpage that shows the website you are accessing contains illegal, unauthorized and unethical content and you are not allowed to access this website or webpage.

We will also store the date and time with website name in database when the user opens blocked website which can be viewed only by superuser who have installed this tool on that PC.

This tool can be used by the organizations to block a particular websites that they don't allow their employees to access on organization's PC. This can be done by adding the particular website to this tool. This will reduce Internet Slacking.

After making this tool parents and teachers are able to restrict their children and students respectively from accessing unauthorized or illegal websites that makes the bad impact on their mindset.

We can also use this tool in an organization that will restrict their employees from using the particular websites like social networking sites during work hours, that will increase their productivity and reduce Internet Slacking.

An Internet filter tool can also increase workplace productivity and avoid potential HR issues. These goals are achieved by blocking access to personal social media accounts, online shopping portals, sites containing pornographic material and video streaming websites.

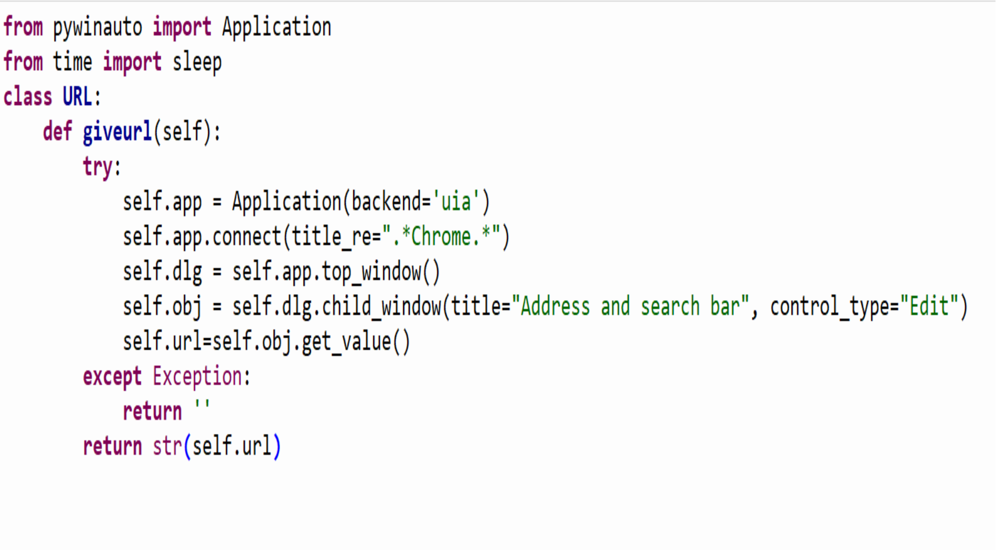
Chapter 5
Bibliography

- **geeksforgeeks.com**
- **w3schools.com**
- **towardsdatascience.com**
- **pythonspot.com**
- **javatpoint.com**

Chapter 6

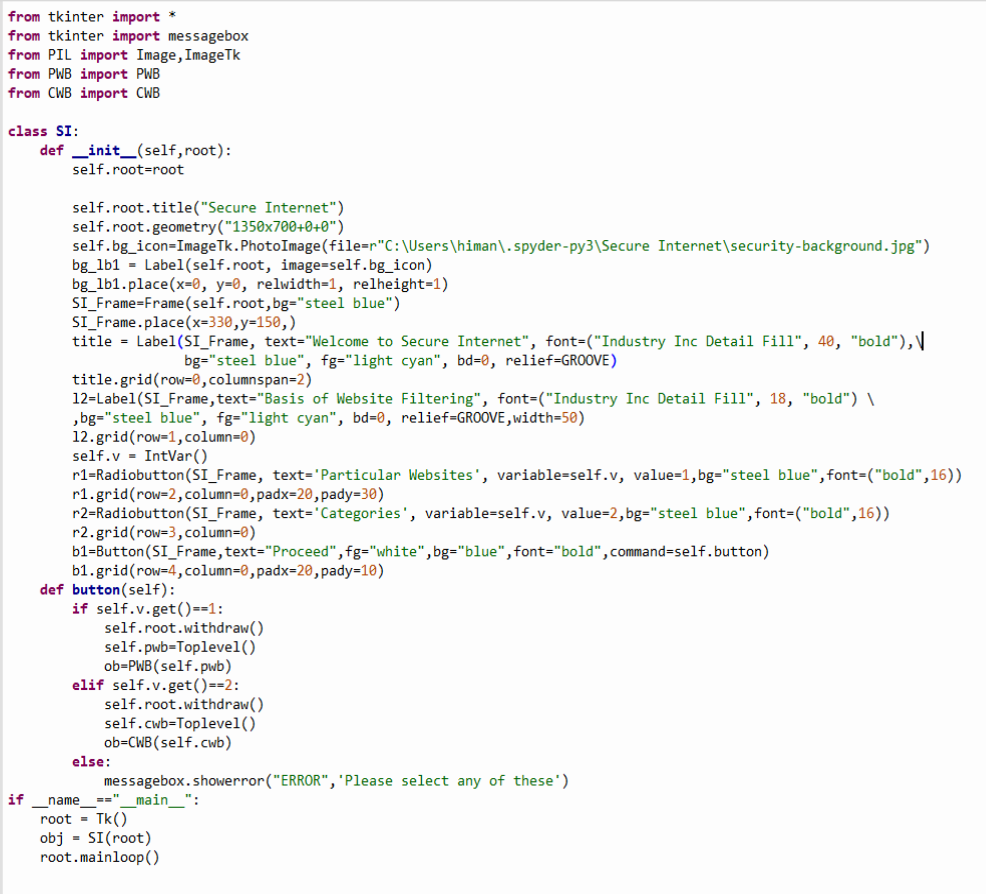
Appendices

URL :



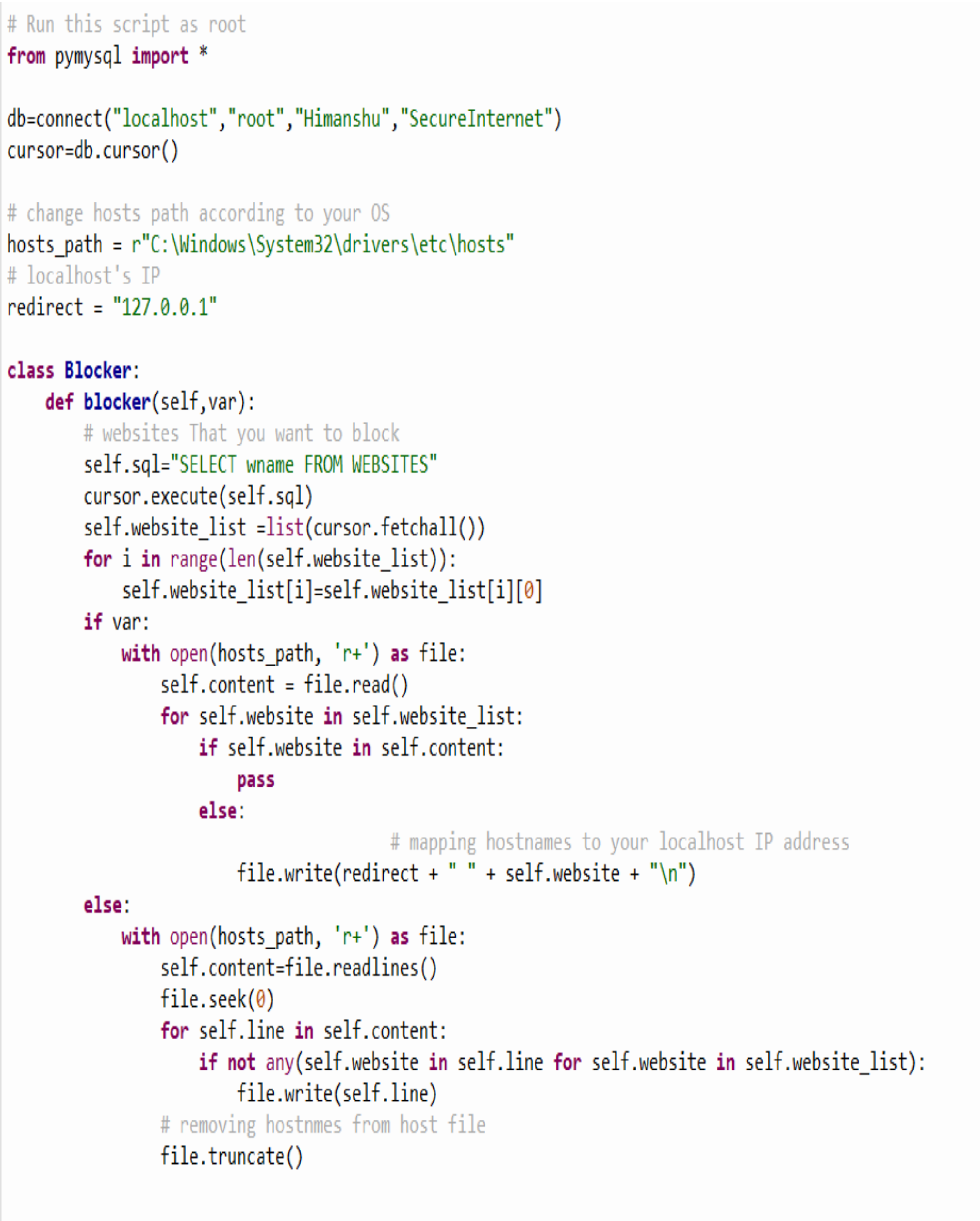
```
from pywinauto import Application
from time import sleep
class URL:
    def giveurl(self):
        try:
            self.app = Application(backend='uia')
            self.app.connect(title_re=".*Chrome.*")
            self.dlg = self.app.top_window()
            self.obj = self.dlg.child_window(title="Address and search bar", control_type="Edit")
            self.url=self.obj.get_value()
        except Exception:
            return ''
        return str(self.url)
```

Secure Internet :



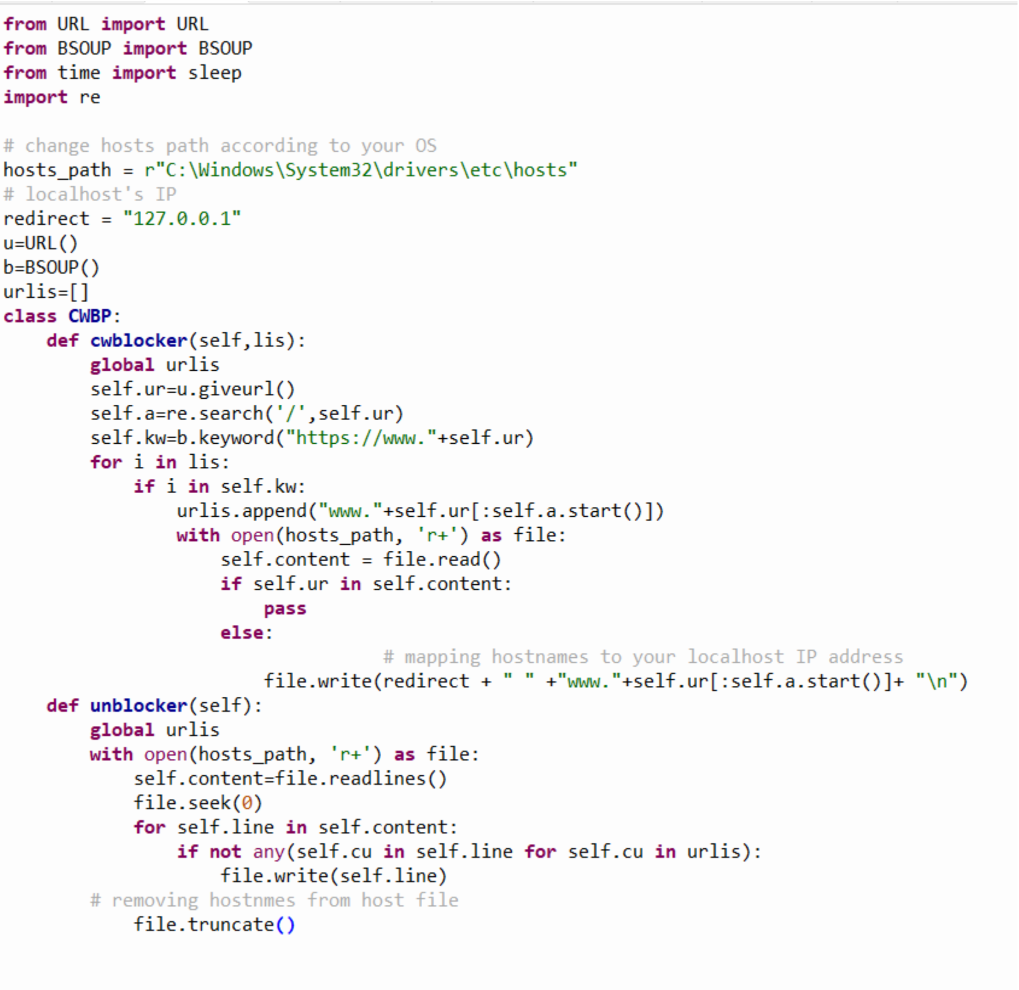
```
from tkinter import *
from tkinter import messagebox
from PIL import Image,ImageTk
from PWB import PWB
from CWB import CWB

class SI:
    def __init__(self,root):
        self.root=root

        self.root.title("Secure Internet")
        self.root.geometry("1350x700+0+0")
        self.bg_icon=ImageTk.PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\security-background.jpg")
        bg_lb1 = Label(self.root, image=self.bg_icon)
        bg_lb1.place(x=0, y=0, relwidth=1, relheight=1)
        SI_Frame=Frame(self.root,bg="steel blue")
        SI_Frame.place(x=330,y=150,)
        title = Label(SI_Frame, text="Welcome to Secure Internet", font=("Industry Inc Detail Fill", 40, "bold"),\
                      bg="steel blue", fg="light cyan", bd=0, relief=GROOVE)
        title.grid(row=0,columnspan=2)
        l2=Label(SI_Frame,text="Basis of Website Filtering", font=("Industry Inc Detail Fill", 18, "bold") \
        ,bg="steel blue", fg="light cyan", bd=0, relief=GROOVE,width=50)
        l2.grid(row=1,column=0)
        self.v = IntVar()
        r1=Radiobutton(SI_Frame, text='Particular Websites', variable=self.v, value=1,bg="steel blue",font=("bold",16))
        r1.grid(row=2,column=0,padx=20,pady=30)
        r2=Radiobutton(SI_Frame, text='Categories', variable=self.v, value=2,bg="steel blue",font=("bold",16))
        r2.grid(row=3,column=0)
        b1=Button(SI_Frame,text="Proceed",fg="white",bg="blue",font="bold",command=self.button)
        b1.grid(row=4,column=0,padx=20,pady=10)
    def button(self):
        if self.v.get()==1:
            self.root.withdraw()
            self.pwb=Toplevel()
            ob=PWB(self.pwb)
        elif self.v.get()==2:
            self.root.withdraw()
            self.cwb=Toplevel()
            ob=CWB(self.cwb)
        else:
            messagebox.showerror("ERROR",'Please select any of these')
if __name__=="__main__":
    root = Tk()
    obj = SI(root)
    root.mainloop()
```

PWBP



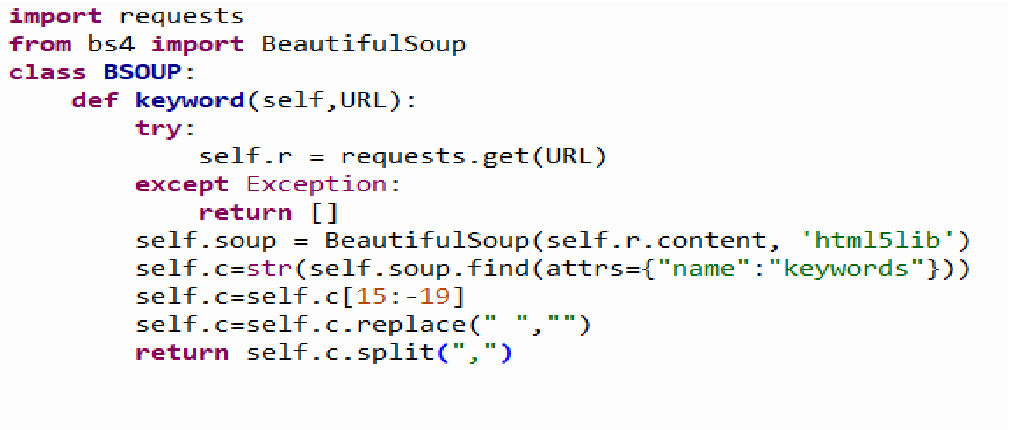
```
# Run this script as root
from pymysql import *

db=connect("localhost","root","Himanshu","SecureInternet")
cursor=db.cursor()

# change hosts path according to your OS
hosts_path = r"C:\Windows\System32\drivers\etc\hosts"
# localhost's IP
redirect = "127.0.0.1"

class Blocker:
    def blocker(self,var):
        # websites That you want to block
        self.sql="SELECT wname FROM WEBSITES"
        cursor.execute(self.sql)
        self.website_list =list(cursor.fetchall())
        for i in range(len(self.website_list)):
            self.website_list[i]=self.website_list[i][0]
        if var:
            with open(hosts_path, 'r+') as file:
                self.content = file.read()
                for self.website in self.website_list:
                    if self.website in self.content:
                        pass
                    else:
                                        # mapping hostnames to your localhost IP address
                        file.write(redirect + " " + self.website + "\n")
        else:
            with open(hosts_path, 'r+') as file:
                self.content=file.readlines()
                file.seek(0)
                for self.line in self.content:
                    if not any(self.website in self.line for self.website in self.website_list):
                        file.write(self.line)
                # removing hostnmes from host file
                file.truncate()
```

CWBP :



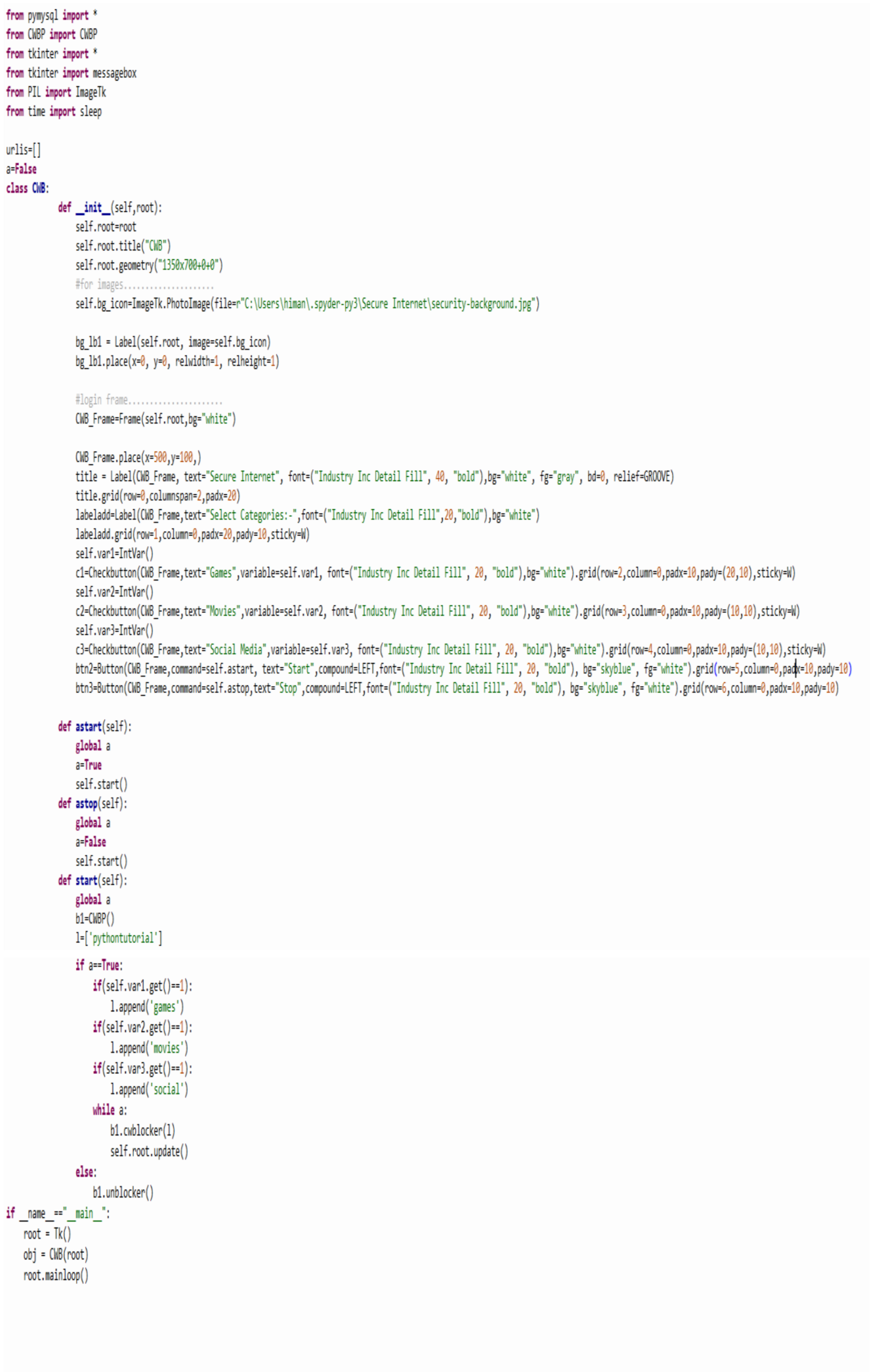
```python
from URL import URL
from BSOUP import BSOUP
from time import sleep
import re

# change hosts path according to your OS
hosts_path = r"C:\Windows\System32\drivers\etc\hosts"
# localhost's IP
redirect = "127.0.0.1"
u=URL()
b=BSOUP()
urlis=[]
class CWBP:
    def cwblocker(self,lis):
        global urlis
        self.ur=u.giveurl()
        self.a=re.search('/',self.ur)
        self.kw=b.keyword("https://www."+self.ur)
        for i in lis:
            if i in self.kw:
                urlis.append("www."+self.ur[:self.a.start()])
                with open(hosts_path, 'r+') as file:
                    self.content = file.read()
                    if self.ur in self.content:
                        pass
                    else:
                                   # mapping hostnames to your localhost IP address
                        file.write(redirect + " " +"www."+self.ur[:self.a.start()]+ "\n")
    def unblocker(self):
        global urlis
        with open(hosts_path, 'r+') as file:
            self.content=file.readlines()
            file.seek(0)
            for self.line in self.content:
                if not any(self.cu in self.line for self.cu in urlis):
                    file.write(self.line)
        # removing hostnmes from host file
            file.truncate()
```

BSOUP :



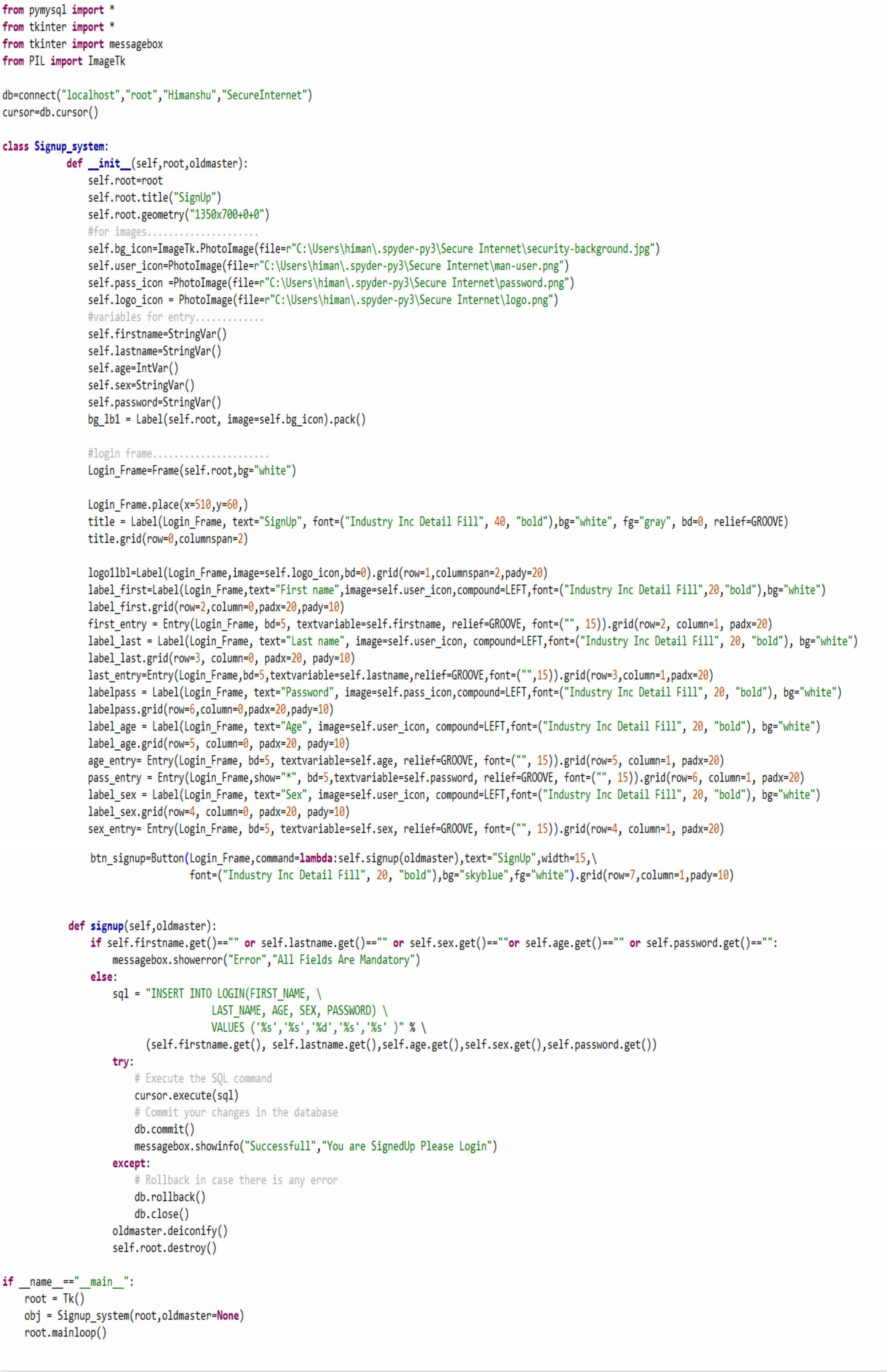
```python
import requests
from bs4 import BeautifulSoup
class BSOUP:
    def keyword(self,URL):
        try:
            self.r = requests.get(URL)
        except Exception:
            return []
        self.soup = BeautifulSoup(self.r.content, 'html5lib')
        self.c=str(self.soup.find(attrs={"name":"keywords"}))
        self.c=self.c[15:-19]
        self.c=self.c.replace(" ","")
        return self.c.split(",")
```

CWB :



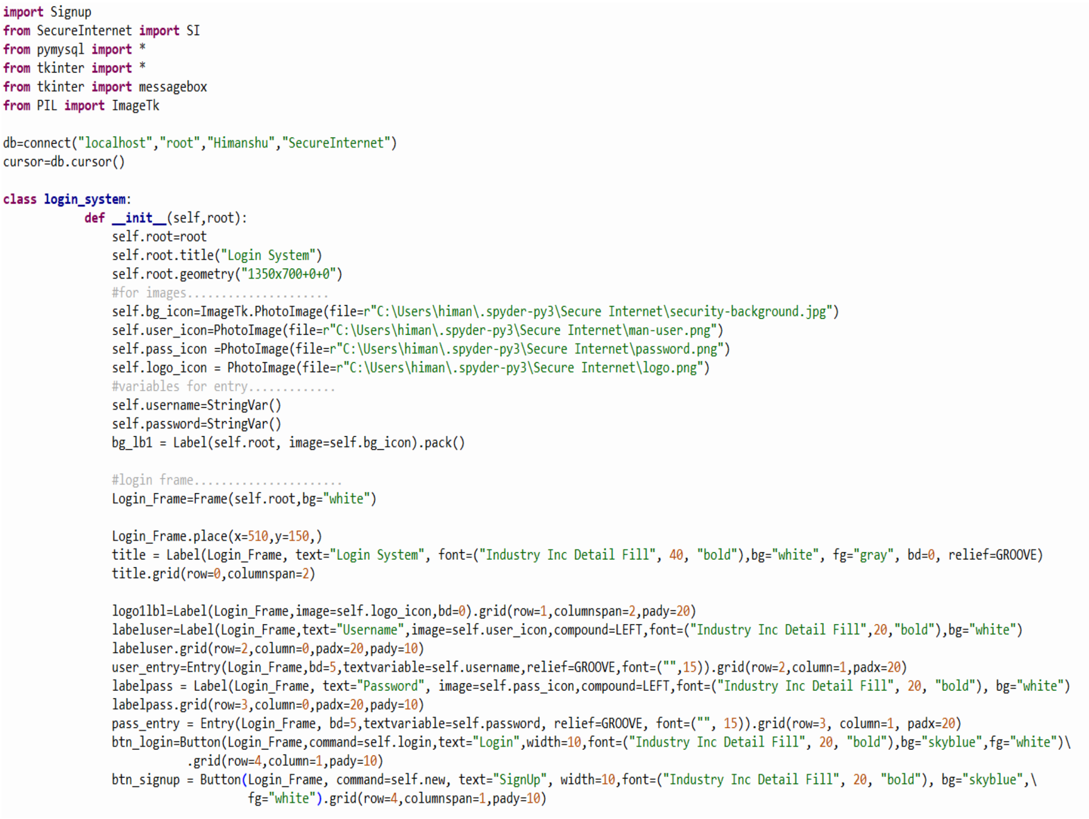
```
from pymysql import *
from CWBP import CWBP
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk
from time import sleep

urlis=[]
a=False
class CWB:
            def __init__(self,root):
                self.root=root
                self.root.title("CWB")
                self.root.geometry("1350x700+0+0")
                #for images....................
                self.bg_icon=ImageTk.PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\security-background.jpg")

                bg_lb1 = Label(self.root, image=self.bg_icon)
                bg_lb1.place(x=0, y=0, relwidth=1, relheight=1)

                #login frame.....................
                CWB_Frame=Frame(self.root,bg="white")

                CWB_Frame.place(x=500,y=100,)
                title = Label(CWB_Frame, text="Secure Internet", font=("Industry Inc Detail Fill", 40, "bold"),bg="white", fg="gray", bd=0, relief=GROOVE)
                title.grid(row=0,columnspan=2,padx=20)
                labeladd=Label(CWB_Frame,text="Select Categories:-",font=("Industry Inc Detail Fill",20,"bold"),bg="white")
                labeladd.grid(row=1,column=0,padx=20,pady=10,sticky=W)
                self.var1=IntVar()
                c1=Checkbutton(CWB_Frame,text="Games",variable=self.var1, font=("Industry Inc Detail Fill", 20, "bold"),bg="white").grid(row=2,column=0,padx=10,pady=(20,10),sticky=W)
                self.var2=IntVar()
                c2=Checkbutton(CWB_Frame,text="Movies",variable=self.var2, font=("Industry Inc Detail Fill", 20, "bold"),bg="white").grid(row=3,column=0,padx=10,pady=(10,10),sticky=W)
                self.var3=IntVar()
                c3=Checkbutton(CWB_Frame,text="Social Media",variable=self.var3, font=("Industry Inc Detail Fill", 20, "bold"),bg="white").grid(row=4,column=0,padx=10,pady=(10,10),sticky=W)
                btn2=Button(CWB_Frame,command=self.astart, text="Start",compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="skyblue", fg="white").grid(row=5,column=0,padx=10,pady=10)
                btn3=Button(CWB_Frame,command=self.astop,text="Stop",compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="skyblue", fg="white").grid(row=6,column=0,padx=10,pady=10)

            def astart(self):
                global a
                a=True
                self.start()
            def astop(self):
                global a
                a=False
                self.start()
            def start(self):
                global a
                b1=CWBP()
                l=['pythontutorial']

                if a==True:
                    if(self.var1.get()==1):
                        l.append('games')
                    if(self.var2.get()==1):
                        l.append('movies')
                    if(self.var3.get()==1):
                        l.append('social')
                    while a:
                        b1.cwblocker(l)
                        self.root.update()
                else:
                    b1.unblocker()
if __name__=="__main__":
    root = Tk()
    obj = CWB(root)
    root.mainloop()
```

Signup :

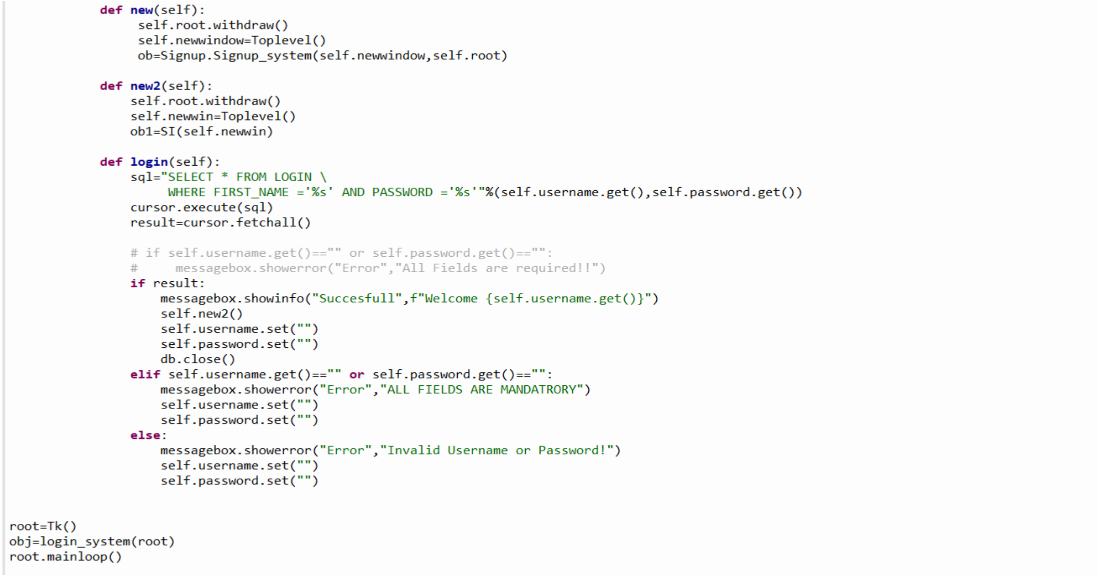


```
from pymysql import *
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk

db=connect("localhost","root","Himanshu","SecureInternet")
cursor=db.cursor()

class Signup_system:
            def __init__(self,root,oldmaster):
                self.root=root
                self.root.title("SignUp")
                self.root.geometry("1350x700+0+0")
                #for images....................
                self.bg_icon=ImageTk.PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\security-background.jpg")
                self.user_icon=PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\man-user.png")
                self.pass_icon =PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\password.png")
                self.logo_icon = PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\logo.png")
                #variables for entry............
                self.firstname=StringVar()
                self.lastname=StringVar()
                self.age=IntVar()
                self.sex=StringVar()
                self.password=StringVar()
                bg_lb1 = Label(self.root, image=self.bg_icon).pack()

                #login frame.....................
                Login_Frame=Frame(self.root,bg="white")

                Login_Frame.place(x=510,y=60,)
                title = Label(Login_Frame, text="SignUp", font=("Industry Inc Detail Fill", 40, "bold"),bg="white", fg="gray", bd=0, relief=GROOVE)
                title.grid(row=0,columnspan=2)

                logo1lbl=Label(Login_Frame,image=self.logo_icon,bd=0).grid(row=1,columnspan=2,pady=20)
                label_first=Label(Login_Frame,text="First name",image=self.user_icon,compound=LEFT,font=("Industry Inc Detail Fill",20,"bold"),bg="white")
                label_first.grid(row=2,column=0,padx=20,pady=10)
                first_entry = Entry(Login_Frame, bd=5, textvariable=self.firstname, relief=GROOVE, font=("", 15)).grid(row=2, column=1, padx=20)
                label_last = Label(Login_Frame, text="Last name", image=self.user_icon, compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="white")
                label_last.grid(row=3, column=0, padx=20, pady=10)
                last_entry=Entry(Login_Frame,bd=5,textvariable=self.lastname,relief=GROOVE,font=("",15)).grid(row=3,column=1,padx=20)
                labelpass = Label(Login_Frame, text="Password", image=self.pass_icon,compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="white")
                labelpass.grid(row=6,column=0,padx=20,pady=10)
                label_age = Label(Login_Frame, text="Age", image=self.user_icon, compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="white")
                label_age.grid(row=5, column=0, padx=20, pady=10)
                age_entry= Entry(Login_Frame, bd=5, textvariable=self.age, relief=GROOVE, font=("", 15)).grid(row=5, column=1, padx=20)
                pass_entry = Entry(Login_Frame,show="*", bd=5,textvariable=self.password, relief=GROOVE, font=("", 15)).grid(row=6, column=1, padx=20)
                label_sex = Label(Login_Frame, text="Sex", image=self.user_icon, compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="white")
                label_sex.grid(row=4, column=0, padx=20, pady=10)
                sex_entry= Entry(Login_Frame, bd=5, textvariable=self.sex, relief=GROOVE, font=("", 15)).grid(row=4, column=1, padx=20)

                btn_signup=Button(Login_Frame,command=lambda:self.signup(oldmaster),text="SignUp",width=15,\
                                   font=("Industry Inc Detail Fill", 20, "bold"),bg="skyblue",fg="white").grid(row=7,column=1,pady=10)


            def signup(self,oldmaster):
                if self.firstname.get()=="" or self.lastname.get()=="" or self.sex.get()==""or self.age.get()=="" or self.password.get()=="":
                    messagebox.showerror("Error","All Fields Are Mandatory")
                else:
                    sql = "INSERT INTO LOGIN(FIRST_NAME, \
                                       LAST_NAME, AGE, SEX, PASSWORD) \
                                       VALUES ('%s','%s','%d','%s','%s' )" % \
                           (self.firstname.get(), self.lastname.get(),self.age.get(),self.sex.get(),self.password.get())
                    try:
                        # Execute the SQL command
                        cursor.execute(sql)
                        # Commit your changes in the database
                        db.commit()
                        messagebox.showinfo("Successfull","You are SignedUp Please Login")
                    except:
                        # Rollback in case there is any error
                        db.rollback()
                        db.close()
                    oldmaster.deiconify()
                    self.root.destroy()

if __name__=="__main__":
    root = Tk()
    obj = Signup_system(root,oldmaster=None)
    root.mainloop()
```

login :

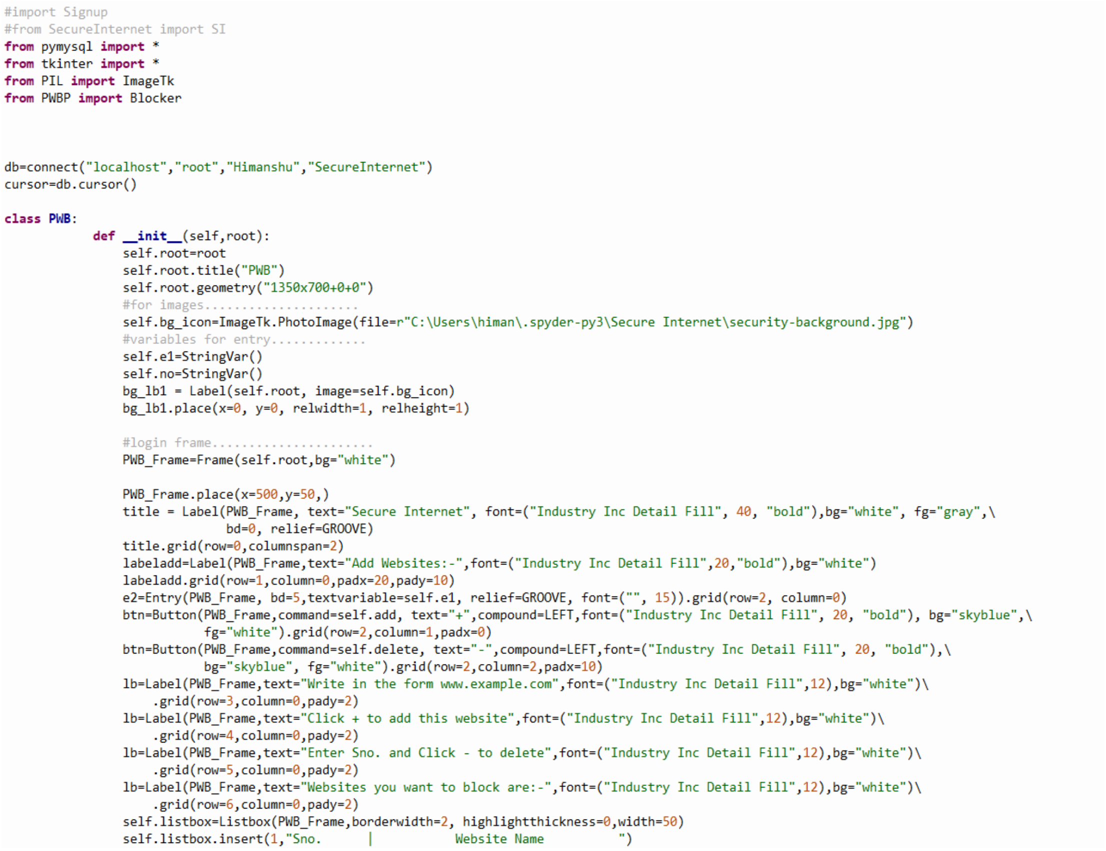


```
import Signup
from SecureInternet import SI
from pymysql import *
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk

db=connect("localhost","root","Himanshu","SecureInternet")
cursor=db.cursor()

class login_system:
            def __init__(self,root):
                self.root=root
                self.root.title("Login System")
                self.root.geometry("1350x700+0+0")
                #for images.....................
                self.bg_icon=ImageTk.PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\security-background.jpg")
                self.user_icon=PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\man-user.png")
                self.pass_icon =PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\password.png")
                self.logo_icon = PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\logo.png")
                #variables for entry.............
                self.username=StringVar()
                self.password=StringVar()
                bg_lb1 = Label(self.root, image=self.bg_icon).pack()

                #login frame.....................
                Login_Frame=Frame(self.root,bg="white")

                Login_Frame.place(x=510,y=150,)
                title = Label(Login_Frame, text="Login System", font=("Industry Inc Detail Fill", 40, "bold"),bg="white", fg="gray", bd=0, relief=GROOVE)
                title.grid(row=0,columnspan=2)

                logo1lbl=Label(Login_Frame,image=self.logo_icon,bd=0).grid(row=1,columnspan=2,pady=20)
                labeluser=Label(Login_Frame,text="Username",image=self.user_icon,compound=LEFT,font=("Industry Inc Detail Fill",20,"bold"),bg="white")
                labeluser.grid(row=2,column=0,padx=20,pady=10)
                user_entry=Entry(Login_Frame,bd=5,textvariable=self.username,relief=GROOVE,font=("",15)).grid(row=2,column=1,padx=20)
                labelpass = Label(Login_Frame, text="Password", image=self.pass_icon,compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="white")
                labelpass.grid(row=3,column=0,padx=20,pady=10)
                pass_entry = Entry(Login_Frame, bd=5,textvariable=self.password, relief=GROOVE, font=("", 15)).grid(row=3, column=1, padx=20)
                btn_login=Button(Login_Frame,command=self.login,text="Login",width=10,font=("Industry Inc Detail Fill", 20, "bold"),bg="skyblue",fg="white")\
                           .grid(row=4,column=1,pady=10)
                btn_signup = Button(Login_Frame, command=self.new, text="SignUp", width=10,font=("Industry Inc Detail Fill", 20, "bold"), bg="skyblue",\
                                    fg="white").grid(row=4,columnspan=1,pady=10)
```

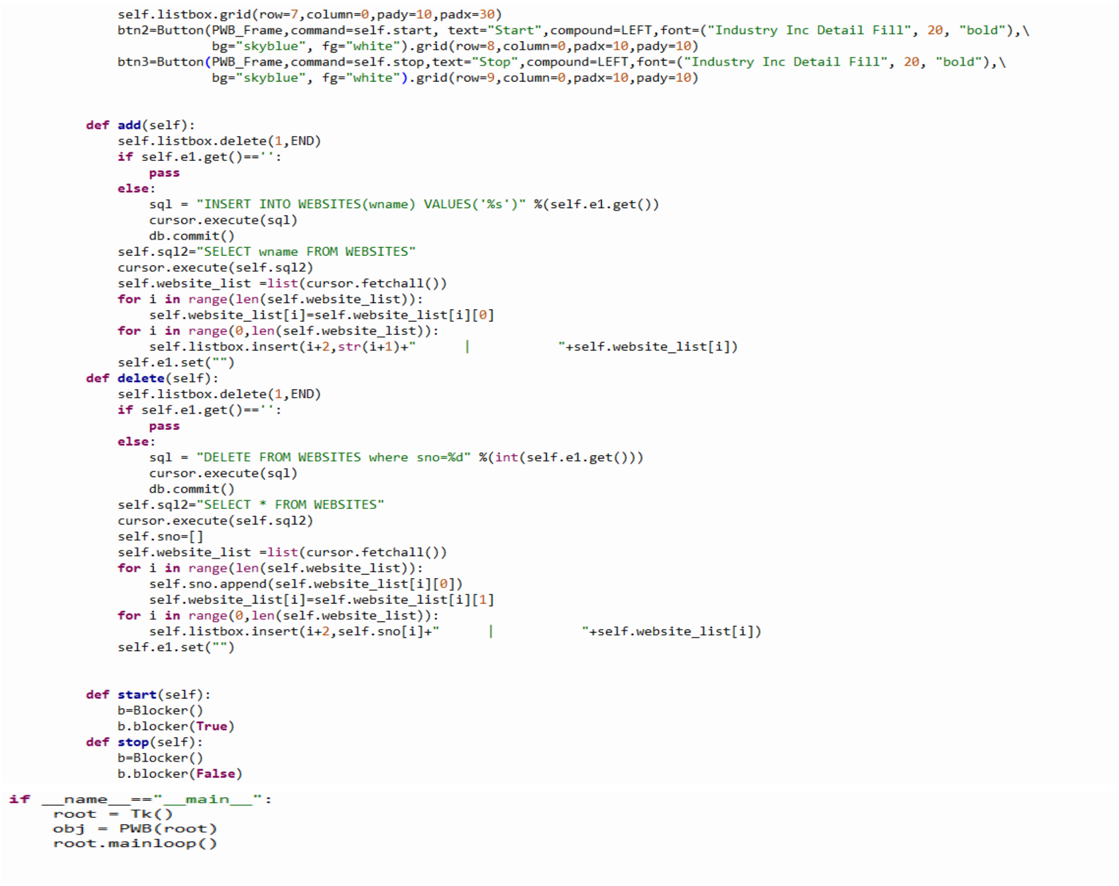


```
            def new(self):
                 self.root.withdraw()
                 self.newwindow=Toplevel()
                 ob=Signup.Signup_system(self.newwindow,self.root)

            def new2(self):
                self.root.withdraw()
                self.newwin=Toplevel()
                ob1=SI(self.newwin)

            def login(self):
                sql="SELECT * FROM LOGIN \
                     WHERE FIRST_NAME ='%s' AND PASSWORD ='%s'"%(self.username.get(),self.password.get())
                cursor.execute(sql)
                result=cursor.fetchall()

                # if self.username.get()=="" or self.password.get()=="":
                #     messagebox.showerror("Error","All Fields are required!!")
                if result:
                    messagebox.showinfo("Succesfull",f"Welcome {self.username.get()}")
                    self.new2()
                    self.username.set("")
                    self.password.set("")
                    db.close()
                elif self.username.get()=="" or self.password.get()=="":
                    messagebox.showerror("Error","ALL FIELDS ARE MANDATRORY")
                    self.username.set("")
                    self.password.set("")
                else:
                    messagebox.showerror("Error","Invalid Username or Password!")
                    self.username.set("")
                    self.password.set("")


root=Tk()
obj=login_system(root)
root.mainloop()
```

PWB:



```
#import Signup
#from SecureInternet import SI
from pymysql import *
from tkinter import *
from PIL import ImageTk
from PWBP import Blocker


db=connect("localhost","root","Himanshu","SecureInternet")
cursor=db.cursor()

class PWB:
            def __init__(self,root):
                self.root=root
                self.root.title("PWB")
                self.root.geometry("1350x700+0+0")
                #for images....................
                self.bg_icon=ImageTk.PhotoImage(file=r"C:\Users\himan\.spyder-py3\Secure Internet\security-background.jpg")
                #variables for entry............
                self.e1=StringVar()
                self.no=StringVar()
                bg_lb1 = Label(self.root, image=self.bg_icon)
                bg_lb1.place(x=0, y=0, relwidth=1, relheight=1)

                #login frame.....................
                PWB_Frame=Frame(self.root,bg="white")

                PWB_Frame.place(x=500,y=50,)
                title = Label(PWB_Frame, text="Secure Internet", font=("Industry Inc Detail Fill", 40, "bold"),bg="white", fg="gray",\
                              bd=0, relief=GROOVE)
                title.grid(row=0,columnspan=2)
                labeladd=Label(PWB_Frame,text="Add Websites:-",font=("Industry Inc Detail Fill",20,"bold"),bg="white")
                labeladd.grid(row=1,column=0,padx=20,pady=10)
                e2=Entry(PWB_Frame, bd=5,textvariable=self.e1, relief=GROOVE, font=("", 15)).grid(row=2, column=0)
                btn=Button(PWB_Frame,command=self.add, text="+",compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"), bg="skyblue",\
                           fg="white").grid(row=2,column=1,padx=0)
                btn=Button(PWB_Frame,command=self.delete, text="-",compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"),\
                           bg="skyblue", fg="white").grid(row=2,column=2,padx=10)
                lb=Label(PWB_Frame,text="Write in the form www.example.com",font=("Industry Inc Detail Fill",12),bg="white")\
                    .grid(row=3,column=0,pady=2)
                lb=Label(PWB_Frame,text="Click + to add this website",font=("Industry Inc Detail Fill",12),bg="white")\
                    .grid(row=4,column=0,pady=2)
                lb=Label(PWB_Frame,text="Enter Sno. and Click - to delete",font=("Industry Inc Detail Fill",12),bg="white")\
                    .grid(row=5,column=0,pady=2)
                lb=Label(PWB_Frame,text="Websites you want to block are:-",font=("Industry Inc Detail Fill",12),bg="white")\
                    .grid(row=6,column=0,pady=2)
                self.listbox=Listbox(PWB_Frame,borderwidth=2, highlightthickness=0,width=50)
                self.listbox.insert(1,"Sno.      |           Website Name          ")
```



```
                self.listbox.grid(row=7,column=0,pady=10,padx=30)
                btn2=Button(PWB_Frame,command=self.start, text="Start",compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"),\
                            bg="skyblue", fg="white").grid(row=8,column=0,padx=10,pady=10)
                btn3=Button(PWB_Frame,command=self.stop,text="Stop",compound=LEFT,font=("Industry Inc Detail Fill", 20, "bold"),\
                            bg="skyblue", fg="white").grid(row=9,column=0,padx=10,pady=10)


            def add(self):
                self.listbox.delete(1,END)
                if self.e1.get()=='':
                    pass
                else:
                    sql = "INSERT INTO WEBSITES(wname) VALUES('%s')" %(self.e1.get())
                    cursor.execute(sql)
                    db.commit()
                self.sql2="SELECT wname FROM WEBSITES"
                cursor.execute(self.sql2)
                self.website_list =list(cursor.fetchall())
                for i in range(len(self.website_list)):
                    self.website_list[i]=self.website_list[i][0]
                for i in range(0,len(self.website_list)):
                    self.listbox.insert(i+2,str(i+1)+"      |           "+self.website_list[i])
                self.e1.set("")
            def delete(self):
                self.listbox.delete(1,END)
                if self.e1.get()=='':
                    pass
                else:
                    sql = "DELETE FROM WEBSITES where sno=%d" %(int(self.e1.get()))
                    cursor.execute(sql)
                    db.commit()
                self.sql2="SELECT * FROM WEBSITES"
                cursor.execute(self.sql2)
                self.sno=[]
                self.website_list =list(cursor.fetchall())
                for i in range(len(self.website_list)):
                    self.sno.append(self.website_list[i][0])
                    self.website_list[i]=self.website_list[i][1]
                for i in range(0,len(self.website_list)):
                    self.listbox.insert(i+2,self.sno[i]+"      |           "+self.website_list[i])
                self.e1.set("")


            def start(self):
                b=Blocker()
                b.blocker(True)
            def stop(self):
                b=Blocker()
                b.blocker(False)

if __name__=="__main__":
    root = Tk()
    obj = PWB(root)
    root.mainloop()
```