

نحوه پیاده سازی:

در ابتدا کاربر وارد می شود در این بخش ما یک متود به عنوان متود login_a داریم ک با استفاده از کتابخانه TKINTER در پایتون پیاده سازی شده که ما ابتدا یوزرنیم و پسورد را می گیریم و در این متود با مقدار یوزر و پسورد برنامه که admin به ازای جفت آنها می باشد تطابق می دهیم و اگر درست بود وارد منوی اصلی برنامه می شویم .

```
def login_a():  
    global username,password,entry1,entry2  
    username="admin"  
    password="admin"  
    def login():  
        value1 = entry1.get()  
        value2 = entry2.get()  
        if value1 == username and value2 == password:  
            messagebox.showinfo("Success", "You log in !")  
            gui1.quit()  
        else:  
            messagebox.showinfo("Error", "Wrong password or username")  
  
    gui1 = Tk()  
    gui1.resizable(width=False, height=False)  
    gui1.title('proxy server')  
    lable = Label(gui1, text="PROXY SERVER")  
    lable.grid(row=0, column=1)  
    lable.config(font='Times 22', fg='red')  
    entry1 = Entry(gui1, width=30)  
    entry2 = Entry(gui1, width=30)  
    entry2.config(show='*')  
    lblentry1 = ttk.Label(text="Your Username : ")  
    lblentry2 = ttk.Label(text="Your Password : ")  
    lblentry1.grid(row=1, column=0, sticky='W', pady=10)  
    lblentry2.grid(row=2, column=0, sticky='W')  
    entry1.grid(row=1, column=1, sticky=W)  
    entry2.grid(row=2, column=1, sticky=W)  
    button1 = Button(gui1, text='Login', command=login)  
    button1.grid(row=3, column=1, sticky=W, padx=50, pady=6)  
    gui1.mainloop()
```

در منوی اصلی ک مانند صفحه لاگین از همان کتاب خانه استفاده شده ار چهار دکمه استفاده کردیم که یکی آنها برای شروع برنامه می باشد ک نام آن استارت است .

بعد از شروع برنامه باید در تنظیمات پراکسی مشتری ادرس ایپی که در برنامه می باشد و همین طور شماره پورت را وارد کرده و

Connection Settings

Configure Proxy Access to the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8078

☒ Also use this proxy for FTP and HTTPS

HTTPS Proxy Port 0

FTP Proxy Port 0

SOCKS Host Port 0

☐ SOCKS v4 ☒ SOCKS v5

☐ Automatic proxy configuration URL

Reload

No proxy for

OK Cancel Help

شروع به استفاده از اینترنت کنیم. برنامه ما ابتدا با استفاده از متود های سوکت به ترتیب:

1. متود socket ک منابع را از ویندوز رزرو می کند

2. bind برنامه به عنوان سرور آماده دریافت و اتصال روی آیی و پورت مورد نظر می شود

3. listen که با این متود تعداد اتصالات به سرور را مشخص می کند

4. accept با این متود مشخصات کلاینت متصل شده را ذخیره می کند

```
def main_a():
    global data, a, browser, webserver, temp, client, spec, buffer, counter, flag_filter, username, password, entry1, entry2
    flag_filter = 0
    counter = 0
    buffer = 1000000
    #create_table_filter()
    #create_table()
    a = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    a.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    a.bind(("127.0.0.1", 8078))
    a.listen(1)
    client, spec = a.accept()

    while True:
        try:
            data = client.recv(buffer)
            threading.Thread(target=position()).start()
        except Exception as e:
            print(e)
```

*از چند نخی برای صدا کردن متود position استفاده می کنیم

در ادامه شروع به دریافت اطلاعات از ماشین کلاینت می شود ک هر بار که اطلاعات را دریافت میکند وارد متود position می شود و ادرس ایپی و url سایت را از اطلاعات دریافت می کند و به متود se می فرستد .

```

def position():
    try:

        first_line=data.decode("ANSI").split("\n")[0]

        url = first_line.split(" ")[1]

        http_pos = url.find("://")
        if http_pos == -1:
            temp = url
        else:
            temp = url[(http_pos + 3):]

        temp = url[(http_pos + 3):]#url site

        port_pos = temp.find(":")

        webserver_pos = temp.find("/")
        if webserver_pos == -1:
            webserver_pos = len(temp)
        webserver = ""
        port = -1
        if port_pos == -1 or webserver_pos < port_pos:
            port = 80
            webserver = temp[:webserver_pos]
        else:
            port = int(temp[(port_pos + 1):][:webserver_pos - port_pos - 1])
            webserver = temp[:port_pos]

        se(webserver, port, client, data, spec,temp)
    except Exception as e:

```

در خط اول اطلاعات از هم جدا می شوند در خط های بعدی به دنبال آییی ماشین متصل و آدرس سایتی که ماشین به آن متصل می شود را پیدا میکنیم

در ادامه اطلاعات به متود se فرستاده می شود که وظیفه این متود ایجاد یک کلاینت مجزا و ارتباط با سرور آن سایت می باشد و هر بار که جوابی دریافت می کند اگر جواب ناقص باشد یا سایت به طور کامل بارگذاری نشود دوباره سعی در گرفتن اطلاعات از سرور سایت میکند

```

def se(webserver,port,clinet,data,spec,temp):
    try:
        check_filter(temp)
        if flag_filter == 0:
            b = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            b.connect((webserver, port))
            b.send(data)
            insert(spec[0], webserver)#insert to database
            while 1:
                reply = b.recv(10000)
                if len(reply) > 0:
                    client.sendall(reply)
                else:
                    break
            b.close()
    except Exception as e:
        print(e)

```

دکمه ی دیتا بیس در صفحه اصلی و پیاده سازی پایگاه داده:

با زدن دکمه دیتا بیس وارد یک صفحه جدید می شویم که اطلاعات ایپی ها و سایت های متصل شده را در آنجا نگه داری میکنیم

برای دیتا بیس ما ابتدا کتاب خانه sqlite را وارد میکنیم و با چند متود با آن کار میکنیم :

```

def create_table():
    x = db.connect('database.db')
    cur = x.cursor()
    cur.execute("create table a1 (id integer primary key autoincrement,ip text , adress text ,)")
    x.close()

```

این متود ابتدا یک دیتا بیس میسازد و نام آنرا database می گذاریم سپس یک جدول با نام a1 می سازیم که سه attribute دارد به نام های id ,ip,text که ایدی یک کلید اصلی است و ایپی همان ایپی می باشد و تکست آدرس سایت می باشد

```

def create_table_filter():
    x=db.connect('database.db')
    cur=x.cursor()
    cur.execute("create table filter (url text PRIMARY KEY)")
    x.close()

def insert(ip1,adress1):
    x = db.connect('database.db')
    cur = x.cursor()
    adress=adress1
    ip=ip1
    ips = x.cursor()
    if check(ip,adress) == 0:
        cur.execute("insert into a1(ip, adress) values ('{}', '{}')".format
            ip, adress)
        x.commit()
    x.close()

def insert_filter():
    x=db.connect('database.db')
    cur = x.cursor()
    url=input("import url filter:")
    cur.execute("insert into filter(url) values ('{}')".format(url))
    x.commit()
    x.close()

```

متود create table filter برای ساخت جدول ادرس های فیلتر شده است
 متود insert برای اضافه شدن ادرس ایپی و سایت بر روی جدول می باشد
 متود inser filter برای اضافه کردن سایت برای فیلتر شدن می باشد