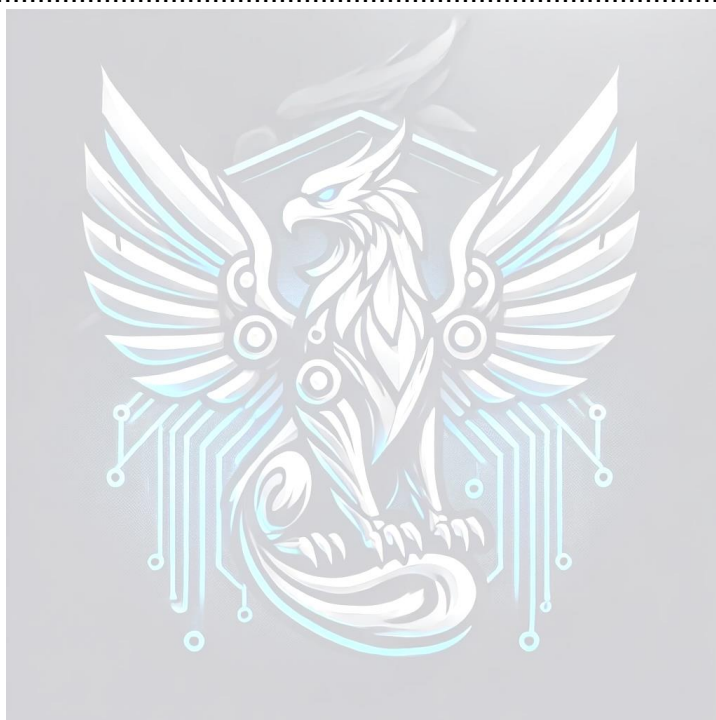


# REPORT SOFTWARE



SOFTWARE .....	3
HOMEPAGE .....	4
HomePage CLI.....	5
PORTSCANNER .....	7
Chi Siamo.....	10

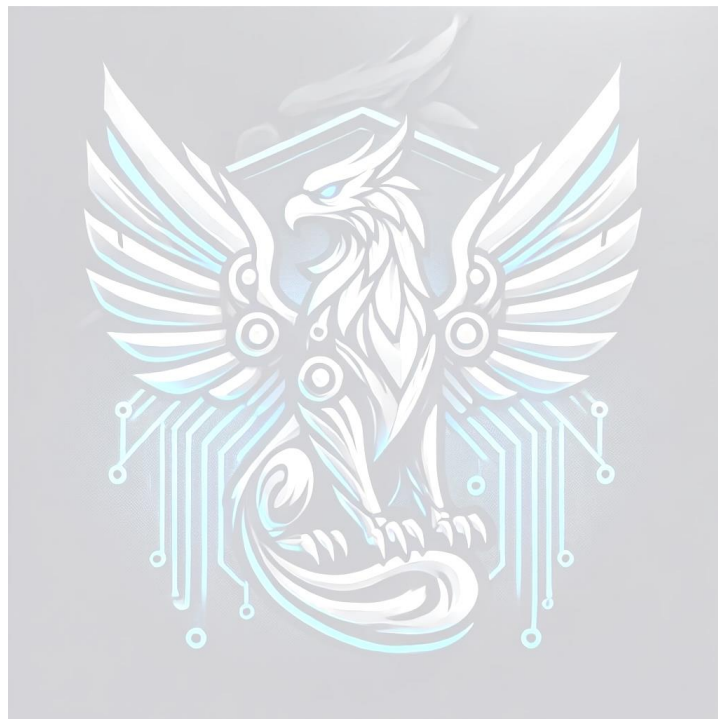


# SOFTWARE

Oggi presenteremo il software dei Security Griffins mostrando le principali funzionalità...

Prima di andare a descrivere le funzionalità del software bisogna specificare dei particolari; l'obiettivo è stato quello di creare un software semplice nell'utilizzo, leggero per gli occhi (utilizzo di colori spenti) e soprattutto un software che permettesse al cliente di lavorare in contemporanea con altre applicazioni

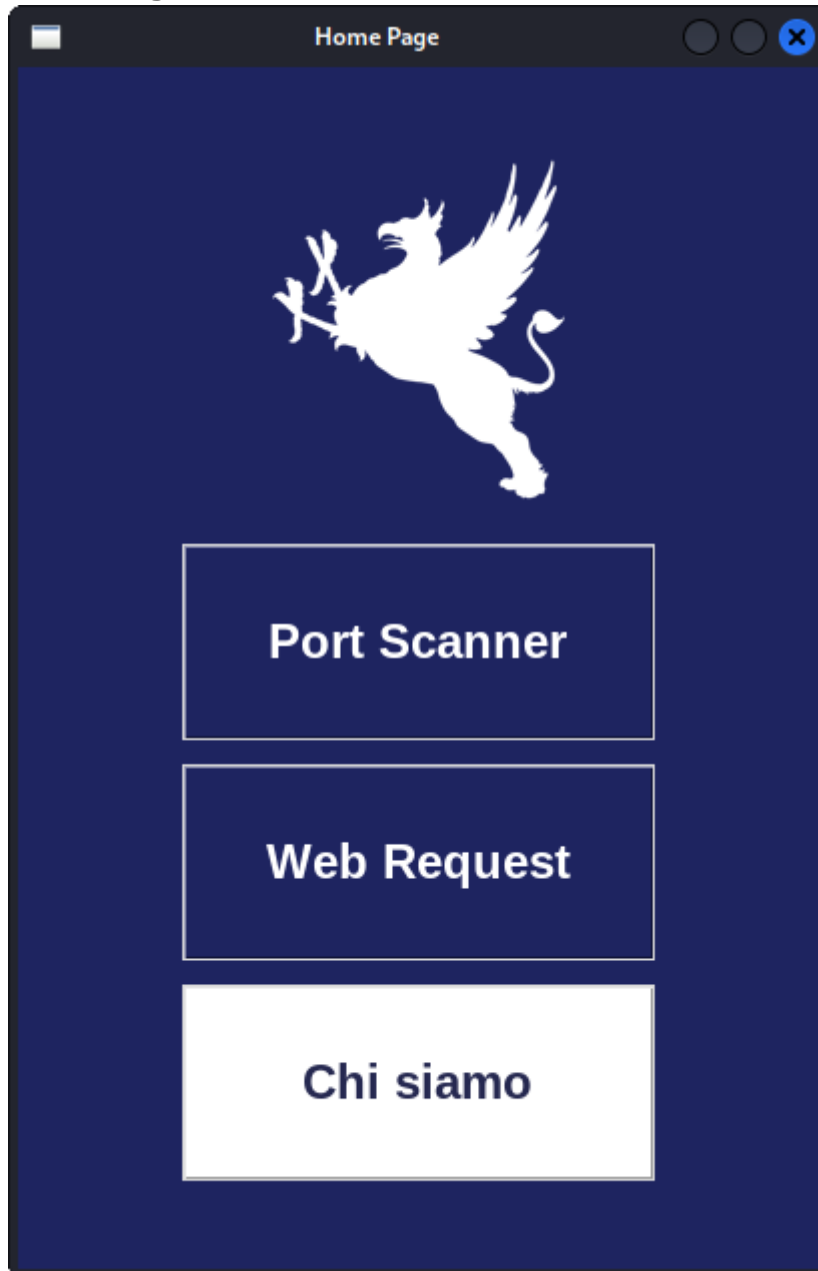
*“Le librerie utilizzate, sono tutte elencate a **pag 2** del file portscanner”*



# HOME PAGE

La homepage del software è rappresentata dal file “eseguibile.py”. Il nome scelto è “eseguibile” proprio per far capire al cliente quale file deve eseguire per poter aprire il software

## HomePage GUI



## HomePage CLI

```
import tkinter
from PIL import Image, ImageTk #Perchè di base tkinter non supporta .jpeg e .png
import subprocess #serve a collegare pagine esterne
import os
```

### 1. Importazione librerie

```
#Finestra Default
finestra = tkinter.Tk()
finestra.title("Home Page")
finestra.geometry("400x620")
finestra.configure(bg="#e7cf8a")
frame = tkinter.Frame(bg="#e7cf8a") #Container
```

### 2. Creazione della pagina root

```
#Gestione Immagine
image_path = os.path.join(directory_attuale_GUI, "Grifone.png")
Img = Image.open(image_path)
Img_dimensione = Img.resize((200, 200))
Img_tk = ImageTk.PhotoImage(Img_dimensione)
```

### 3. Gestione immagine .png tramite libreria PIL



```

#Creazione Widget
BottonePort = tkinter.Button(
    frame, text="Port Scanner", width=16, height=3, font=("Arial", 18),
    bg="black", fg="#e7cf8a", command=ApriPortScanner
)

BottoneRequest = tkinter.Button(
    frame, text="Web Request", width=16, height=3, font=("Arial", 18),
    bg="black", fg="#e7cf8a"
)

BottoneChiSiamo = tkinter.Button(
    frame, text="Chi siamo", width=16, height=3, font=("Arial", 18),
    bg="black", fg="#e7cf8a", command=ChiSiamoButton
)

ImgGrifone = tkinter.Label(frame, image=Img_tk, bg="#e7cf8a")

#Posizionamento Widget
ImgGrifone.grid(row=0)
BottonePort.grid(row=1, column=0, columnspan=2)
BottoneRequest.grid(row=2, column=0, columnspan=2, pady=8)
BottoneChiSiamo.grid(row=3, column=0, columnspan=2)

frame.pack(side="top", pady=40)

#MainLoop
finestra.mainloop()

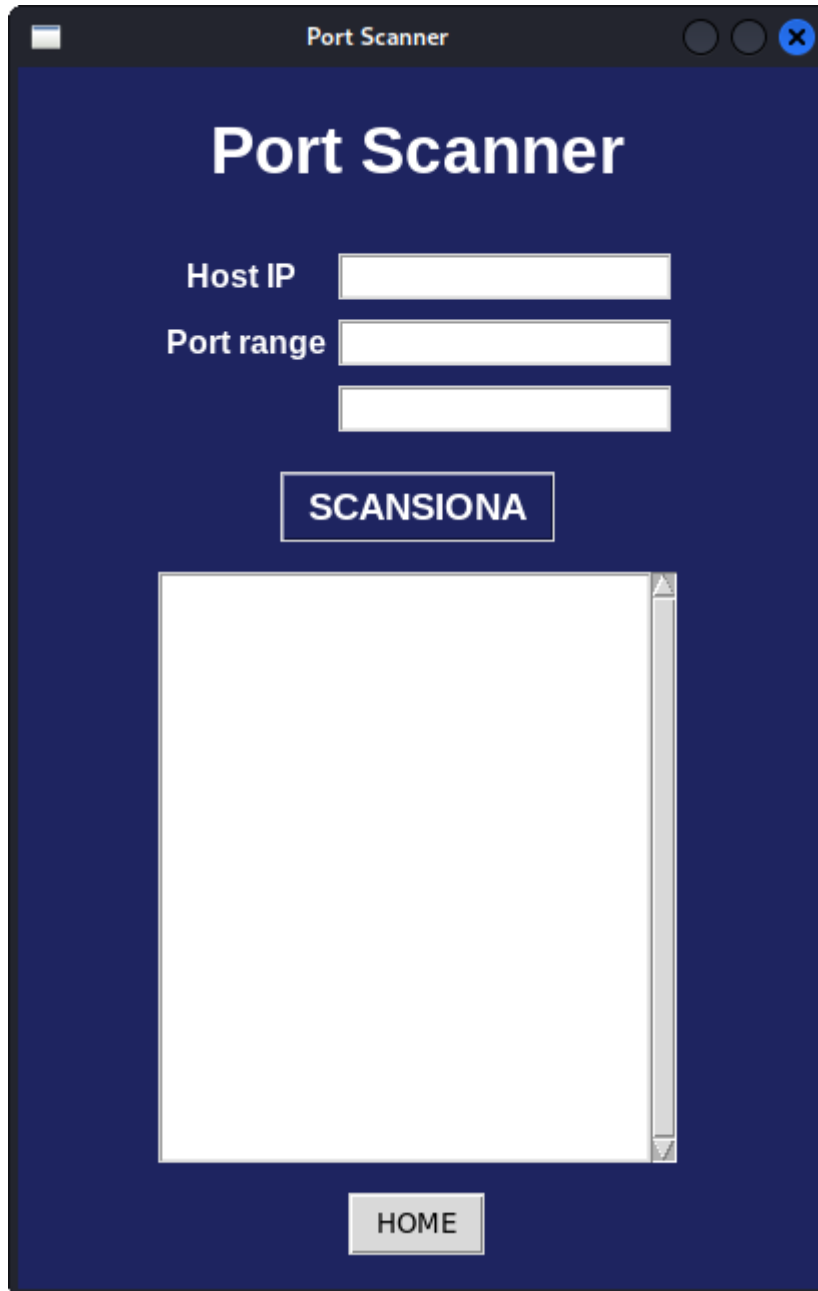
```

#### 4. Parte grafica - creazione widget e buttons

# PORTSCANNER

La GUI del portscanner ci permette di poter scannerizzare tutte le porte in range di un Host a nostra scelta. L'interfaccia grafica è stata creata appositamente per aiutare il cliente finale ad avere comodità nell'utilizzo del software

## Portscanner GUI





## Portscanner CLI

```
import tkinter
from tkinter import scrolledtext
from portscanner import scan_ports
import threading
import queue
import os
import subprocess
```

### 1. Importazione delle librerie

```
finestra = tkinter.Tk()
finestra.title("Port Scanner")
finestra.geometry("400x610")
finestra.configure(bg="#e7cf8a")

frame = tkinter.Frame(bg="#e7cf8a") #Container

#PathPrincipale
directory_attuale_GUI = os.path.dirname(os.path.abspath(__file__))
```

### 2. Creazione finestra root

```
#Creazione degli widget
spazio_titolo = tkinter.Label(frame, text="Port Scanner",
    bg="#e7cf8a", font=("Arial", 25, "bold")
)

spazio_Host = tkinter.Label(frame, text="Host IP ",
    bg="#e7cf8a", font=("Arial", 12, "bold")
)
inserisci_Host = tkinter.Entry(frame)

spazio_portainizio = tkinter.Label(frame, text="Port range",
    bg="#e7cf8a", font=("Arial", 12, "bold")
)
inserisci_portainizio = tkinter.Entry(frame)

inserisci_portafine = tkinter.Entry(frame)

bottone_scansiona = tkinter.Button(frame, text="SCANSIONA", bg="black",
    fg="#e7cf8a", font=("Arial", 14, "bold"), command=scansiona
)

result_text = scrolledtext.ScrolledText(frame, width=30, height=17)

bottone_home = tkinter.Button(frame, text="HOME", command=Home)
```

### 3. Creazione widget e personalizzazione



```

#Posizionamento degli Widget
spazio_titolo.grid(row=0, column=0, columnspan=2, sticky="news", pady=20)
spazio_Host.grid(row=1, column=0)
inserisci_Host.grid(row=1, column=1, pady=10)
spazio_portainizio.grid(row=2, column=0)
inserisci_portainizio.grid(row=2, column=1, pady=10)
inserisci_portafine.grid(row=3, column=1, pady=10)
bottone_scansiona.grid(row=4, column=0, columnspan=2, pady=10)
result_text.grid(row=5, column=0, columnspan=2, pady=5)
bottone_home.grid(row=6, column=0, columnspan=2, pady=10)

frame.pack(side="top", pady=0)

# Impostiamo il peso (weight) per la riga e le colonne in modo che possano espandersi correttamente
frame.grid_rowconfigure(6, weight=1) # Per la riga 6 (dove c'è il bottone "HOME")
frame.grid_columnconfigure(0, weight=1)
frame.grid_columnconfigure(1, weight=1)

finestra.mainloop() #fino quando c'è questo loop l'app funzionerà

```

#### 4. Posizionamento Widget

```

def scansiona(): #Bottone
    target_host = inserisci_Host.get()

    port1 = inserisci_portainizio.get()
    inizio_port = int(port1)

    port2 = inserisci_portafine.get()
    fine_port = int(port2)

    result_queue = queue.Queue()

    scan_ports(target_host, inizio_port, fine_port, result_queue)

    result_text.delete(1.0, tkinter.END)
    result_text.insert(tkinter.END, f"Porte aperte su {target_host}\n")

    while not result_queue.empty():
        port = result_queue.get()
        result_text.insert(tkinter.END, f"Porta {port} aperta\n")

def Home():
    finestra.destroy()
    Path_Home = os.path.join(directory_attuale_GUI, "eseguibile2.py")
    subprocess.run(["python", Path_Home])

scan_thread = threading.Thread(target=scansiona)
scan_thread.start()

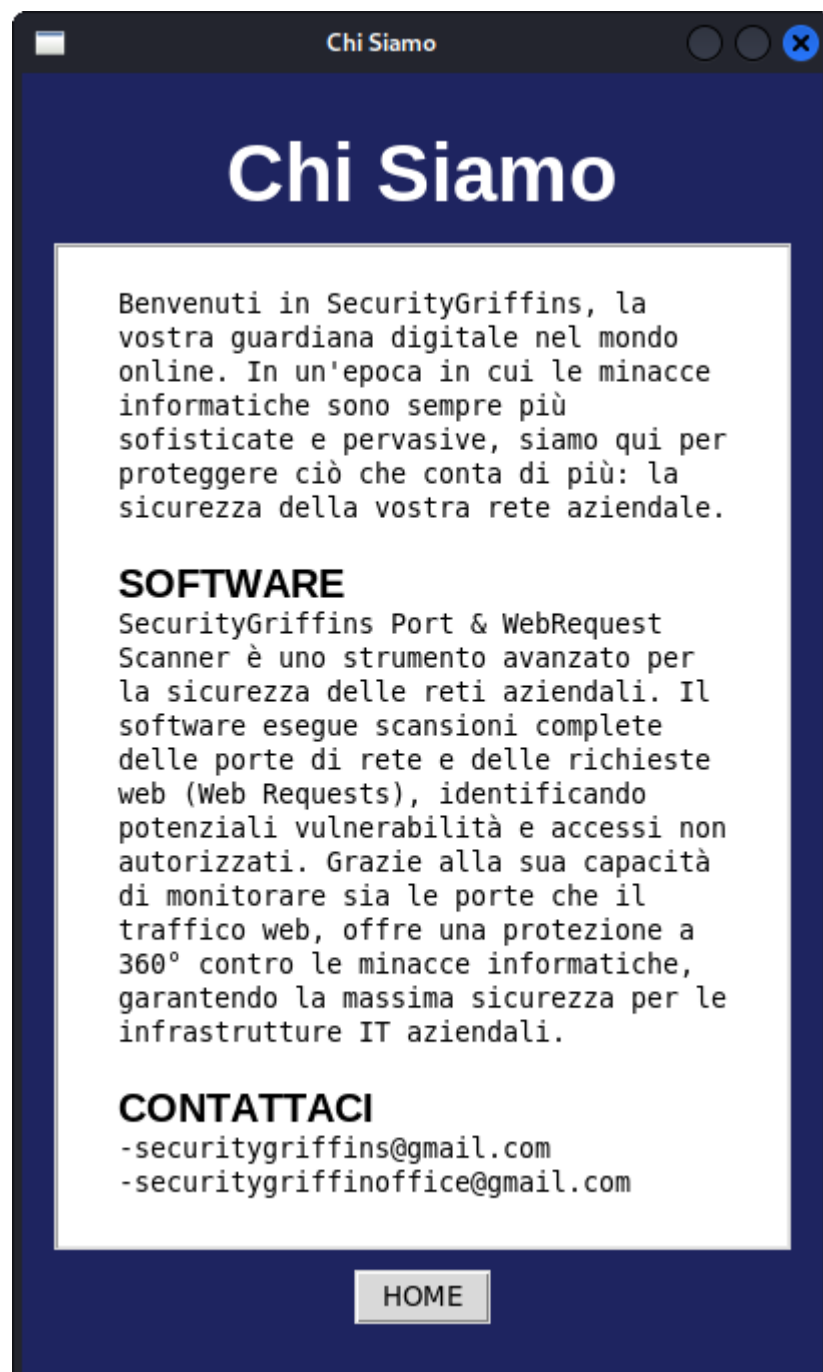
```

#### 5. Creazione funzioni per il button + funzione tasto Home

# Chi Siamo

L'interfaccia 'Chi Siamo' è stata progettata per fornire al cliente una chiara comprensione delle nostre attività e delle principali funzionalità del software

## Chi Siamo GUI



## Chi Siamo CLI

```
import tkinter
import tkinter.scrolledtext
import subprocess
import os
```

### 1. Importazione librerie

```
#Widget
chisiamo = tkinter.Label(frame, text="Chi Siamo",
    background="#e7cf8a", font=("Arial", 30, "bold")
)

didascaliala = tkinter.Text(frame, width=38, height=27,
    wrap=tkinter.WORD, padx=30, pady=20
)

bottone_home = tkinter.Button(frame, text="HOME", command=Home)

#Posizionamento
chisiamo.grid(row=0, column=1, columnspan=2)
didascaliala.grid(row=1, column=1, columnspan=2, pady=10)
bottone_home.grid(row=6, column=0, columnspan=2, pady=0)
```

### 2. Creazione Widget e personalizzazione widget + posizionamento

```
#Inserisci il testo
testochisiamo = """Benvenuti in SecurityGriffins, la vostra guardiana digitale nel mondo online. In
"""
testosoftware = """SecurityGriffins Port & WebRequest Scanner è uno strumento avanzato per la sicurezza
"""
email1 = """-securitygriffins@gmail.com
"""
email2 = """-securitygriffinoffice@gmail.com
"""

didascaliala.tag_configure("titolo", font=("Arial", 15, "bold"))

#Insert
didascaliala.insert(tkinter.END, testochisiamo)
didascaliala.insert(tkinter.END, "\n")
didascaliala.insert(tkinter.END, "SOFTWARE\n", "titolo")
didascaliala.insert(tkinter.END, testosoftware)
didascaliala.insert(tkinter.END, "\n")
didascaliala.insert(tkinter.END, "CONTATTACI\n", "titolo")
didascaliala.insert(tkinter.END, email1)
didascaliala.insert(tkinter.END, email2)

# Impostiamo il peso (weight) per la riga e le colonne in modo che possano espandersi correttamente
frame.grid_rowconfigure(6, weight=1) # Per la riga 6 (dove c'è il bottone "HOME")
frame.grid_columnconfigure(0, weight=1)
frame.grid_columnconfigure(1, weight=1)

#FRAME
frame.pack(side="top", pady=25)

finestra.mainloop()
```

### 3. Inserimento testo nei widget

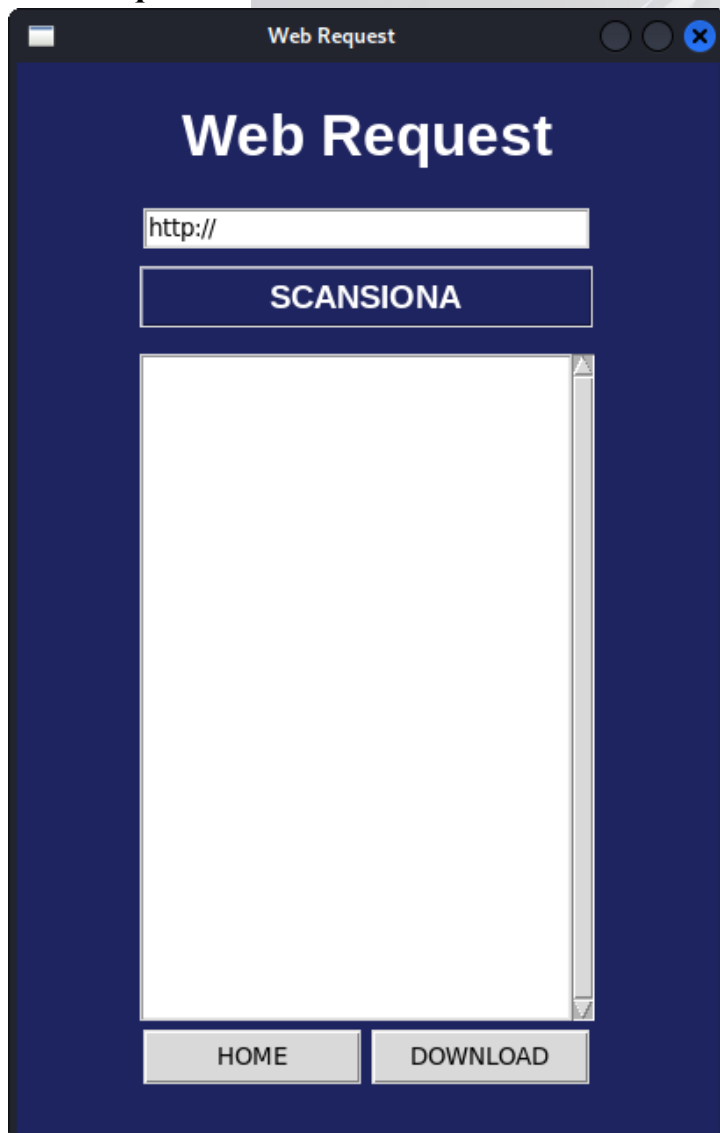
```
def Home():  
    finestra.destroy()  
    Path_Home = os.path.join(directory_attuale_GUI, "eseguibile2.py")  
    subprocess.run(["python", Path_Home])
```

### 4. Funzione button Home

## WEB REQUEST

La Web Request è un'operazione in cui un client (come un browser o un'applicazione) invia una richiesta a un server web per ottenere informazioni o eseguire un'azione

### Web Request GUI



## Web Request CLI

```
import tkinter
from tkinter import scrolledtext, filedialog, messagebox
import os
from httprequest2 import run_request
import subprocess
```

### 1. Importazione Librerie

```
def esegui_request():
    url = inserisci_url.get()

    if not url:
        messagebox.showerror("Inserisci un URL valido.")
        return

    if not url.startswith("http//"):
        url = "http://" + url

    try:
        log_file_path = run_request(url)
        result_text.delete(1.0, tkinter.END)
        result_text.insert(tkinter.END, f"Le richieste sono state completate con successo. \nSalvato in:\n{log_file_path}\n")
    except Exception as e:
        messagebox.showerror("Errore")

def Home():
    finestra.destroy()
    Path_Home = os.path.join(directory_attuale_GUI, "eseguibile2.py")
    subprocess.run(["python", Path_Home])

def download_buttonexc():
    log_dir = "http_requests_logs"
    if not os.path.exists(log_dir):
        messagebox.ERROR("Non ci sono log da scaricare")
        return

    filepath = filedialog.asksaveasfilename(defaultextension=".txt", filetypes=[("Text Files", "*.txt")])
    if filepath:
        latest_log_file = max([os.path.join(log_dir, f) for f in os.listdir(log_dir)], key=os.path.getctime)
        with open(latest_log_file, "r") as file:
            content = file.read()

        with open(filepath, "w") as file:
            file.write(content)
```

### 2. Definizione funzioni tasti Download, Scansiona ed Home

```

finestra = tkinter.Tk()
finestra.title("Web Request")
finestra.geometry("400x610")
finestra.configure(bg="#1e2460")

frame = tkinter.Frame(bg="#1e2460") #Container

#PathPrincipale
directory_attuale_GUI = os.path.dirname(os.path.abspath(__file__))

```

### 3. Creazione finestra root + integrazione del frame

```

#Widget
spazio_titolo = tkinter.Label(frame, text="Web Request",
    bg="#1e2460", fg="white", font=("Arial", 25, "bold")
)

spazio_IP = tkinter.Label(frame, text="URL ",
    bg="#1e2460", font=("Arial", 12, "bold")
)

inserisci_url = tkinter.Entry(frame, width=31)
inserisci_url.insert(tkinter.END, "http://")

bottone_scansiona = tkinter.Button(frame, text="SCANSIONA", bg="#1e2460",
    fg="white", font=("Arial", 14, "bold"), command=esegui_request, width=23
)

result_text = scrolledtext.ScrolledText(frame, width=30, height=22)

download_button = tkinter.Button(frame, text="DOWNLOAD", command=download_buttonexc, width=12)

bottone_home = tkinter.Button(frame, text="HOME", command=Home, width=12)

#Posizionamento Widget
spazio_titolo.grid(row=0, column=0, columnspan=2, sticky="news", pady=20)
inserisci_url.grid(row=1, column=0, columnspan=2)
bottone_scansiona.grid(row=2, column=0, columnspan=2, pady=10)
result_text.grid(row=3, column=0, columnspan=2, pady=5)
download_button.grid(row=4, column=1, columnspan=1)
bottone_home.grid(row=4, column=0, columnspan=1)

frame.pack(side="top", pady=0)

finestra.mainloop()

```

### 4. Creazione e posizionamento degli widget