

## **Relazione dell'elaborato programmazione di reti traccia 3: python web server**

**Nome cognome:** Olivia Rannick Nguemo

**Email:** [oliviarannick.nguemo@studio.unibo.it](mailto:oliviarannick.nguemo@studio.unibo.it)

**Matricola:** 0000935099

### **Traccia 3: Python Web Server**

Si immagini di dover realizzare un Web Server in Python per una agenzia di viaggi. I requisiti del Web Server sono i seguenti:

- Il web server deve consentire l'accesso a più utenti in contemporanea
- La pagina iniziale deve consentire di visualizzare la lista dei servizi erogati dall'agenzia di viaggi e per ogni servizio avere un link di riferimento ad una pagina dedicata presente nella stessa working directory della pagina principale
- Nella pagina principale dovrà anche essere presente un link per il download di un file pdf da parte del browser
- Come requisito facoltativo si chiede di autenticare gli utenti nella fase iniziale della connessione.
- L'interruzione da tastiera (o da console) dell'esecuzione del web server deve essere opportunamente gestita in modo da liberare la risorsa socket.

## Scopo del progetto

Il progetto ha lo scopo di creare un web server in python dando dei collegamenti a dei servizi di viaggi, permettendo all'utente di pianificare un viaggio con serenità nella regio dell'emilia romagna.

## Descrizione della Home Page

La home page contiene dei pulsanti per accedere agli servizi tale che

- Home: che fa riferimento alla pagina iniziale
- Voli: che offre la possibilità di pagare dei biglietti di viaggio in aereo
- Alberghi: che presenta gli alberghi della regione
- Luoghi turistici: propone all'utente i luoghi da visitare
- Meteo: previsione meteorologiche di ogni città della regione
- Orario treni: orari di trenitalia per gli spostamenti nel paese
- Case vacanze: alcune proposte di case per vacanze
- Download :per scaricare un file pdf

## Home



## Benvenuto nell'Agenzia di Viaggio by Olivia



L'Agenzia di Viaggio by Olivia ha per obiettivo di aiutarvi a pianificare i vostri viaggi e un soggiorno tranquillo con le migliori offerte

## Dettagli dell'implementazione del codice

## 1. Costituzione delle pagine

```
308
309 #metodo per la creazione di una generica pagina
310 def create_page_servizio(title,page_html,page_body):
311     f = open(page_html,'w', encoding="utf-8")
312     try:
313         message = header_html + title + nav_bar + page_body + footer_html
314     except:
315         pass
316     f.write(message)
317     f.close()
318
319
```

Ogni pagina ha lo stesso header(costituito della parte css), nav\_bar(la barra dei pulsanti che offrono i servizi) ,e footer .

Ho usato la funzione create\_page\_servizio che costruisce ogni pagina, poi aprirla con la funzione open()

## 2. Autenticazione

```
335     waiting_refresh.set()
336     sys.exit(0)
337
338 # metodo utilizzato all'avvio del server
339 def main():
340     #controllo sulle credenziali di accesso
341     username = input("Inserire username: ")
342     pw = input("Inserire password: ")
343     #se usr o pw sono diverse, chiudo il server e esco
344     if (username != 'olivia' or pw != 'olivia') :
345         print("Errore durante l'autenticazione dell'utente, riprovare")
346         server.server_close()
347         sys.exit(0)
348     print("Autenticazione avvenuta con successo.\n\n")
349
```

Per stabilire la connessione al server, l'utente si deve autenticare con le credenziali seguenti

Username: "olivia"

Password: "olivia"

Che si deve inserire direttamente a riga di comando

### 3. Librerie

```
8
9  ''' Traccia 3- Elaborato Programmazione di Reti - Università di Bologna'''
10
11 import sys, signal
12 import http.server
13 import socketserver
14 import threading
15
```

**http.server:** definisce le classi per implementare i server HTTP

**sys:** fornisce l'accesso ad alcune variabili usate o mantenute dall'interprete, e a funzioni che interagiscono fortemente con l'interprete stesso.

**signal:** per gestire i segnali in Python.

**socketserver:** Il modulo Socket Server è una infrastruttura per la creazione di server di rete.

**threading:** usato per la gestione dei thread