

PROGRAMMAZIONE DI RETI – LABORATORIO

PROGETTO DI FINE CORSO

Traccia 3

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.

NOME E COGNOME : Elisa Barberini

NUMERO DI MATRICOLA : 0000934354

L'obiettivo che ha guidato lo sviluppo del progetto è stata la creazione di un Web Server Python per un'agenzia viaggi e i differenti servizi da essa offerti, ovvero vacanze, alloggi, voli, crociere e attività.

Segue l'homepage che rappresenta l'applicazione.



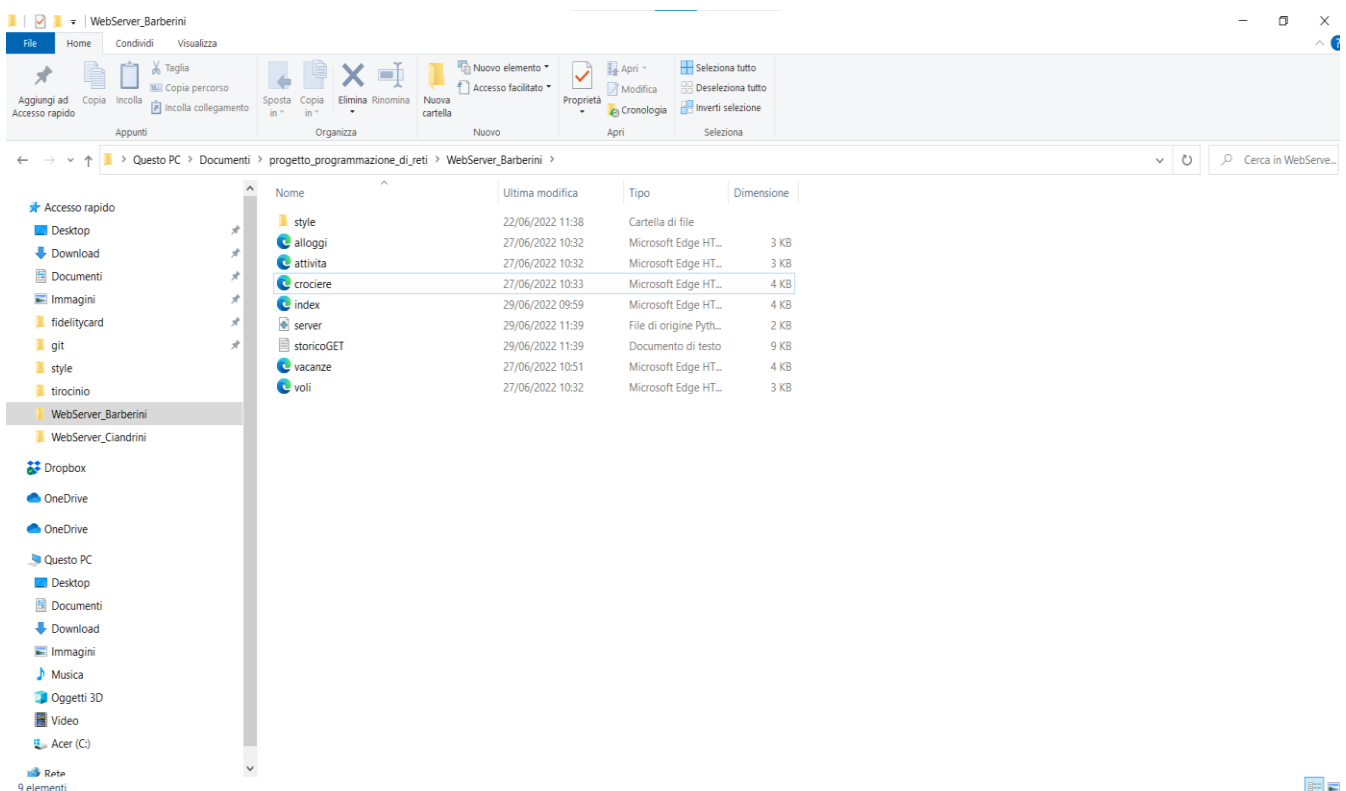
LA STRUTTURA DEL SITO

La web page si compone di 6 pagine HTML ed una cartella style contenente le varie immagini utilizzate ed un foglio di stile rinominato style.css.

Ciascuna delle pagine HTML fa riferimento ad uno dei servizi offerti dall'agenzia di viaggi, mentre la pagina index.html contiene una serie di articoli, ognuno relativo ad un differente servizio, per dare informazioni generali.

Le 6 pagine HTML sono:

- index.html
- vacanze.html
- alloggi.html
- crociere.html
- voli.html
- attività.html.



Il web Server supporta più connessioni in contemporanea, in particolare, è stata creata una classe **ServerHandler** che mantiene le funzioni di **SimpleHTTPRequestHandler** ma implementa il metodo **do_GET**.

do_GET, è un metodo utilizzato per salvare nell'URL i parametri richiesti, ad esempio per una query di ricerca.

Nel nostro caso il metodo precedentemente citato viene utilizzato per scrivere sul file storicoGET.txt le richieste fatte dall'utente.

Il metodo **do_GET** è implementato come segue:

def do_GET(self):

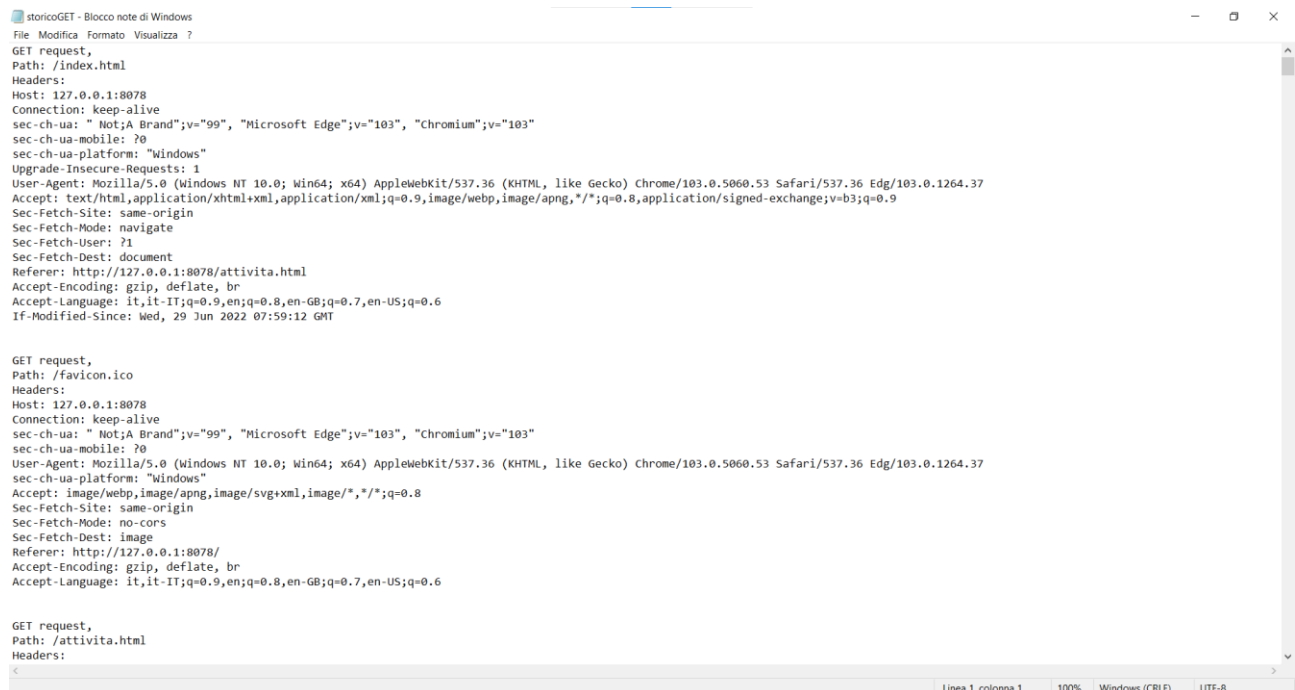
with open("storicoGET.txt", "a") as out:

info = "GET request,\nPath: " + str(self.path) + "\nHeaders:\n" +
str(self.headers) + "\n"

out.write(str(info))

http.server.SimpleHTTPRequestHandler.do_GET(self)

Segue un esempio del file storicoGET.txt:



```
storicoGET - Blocco note di Windows
File Modifica Formato Visualizza ?
GET request,
Path: /index.html
Headers:
Host: 127.0.0.1:8078
Connection: keep-alive
sec-ch-ua: " Not;A Brand";v="99", "Microsoft Edge";v="103", "Chromium";v="103"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36 Edg/103.0.1264.37
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1:8078/attivita.html
Accept-Encoding: gzip, deflate, br
Accept-Language: it,it-IT;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
If-Modified-Since: Wed, 29 Jun 2022 07:59:12 GMT

GET request,
Path: /favicon.ico
Headers:
Host: 127.0.0.1:8078
Connection: keep-alive
sec-ch-ua: " Not;A Brand";v="99", "Microsoft Edge";v="103", "Chromium";v="103"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36 Edg/103.0.1264.37
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://127.0.0.1:8078/
Accept-Encoding: gzip, deflate, br
Accept-Language: it,it-IT;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6

GET request,
Path: /attivita.html
Headers:
```