

Relazione progetto programmazione di reti

Traccia 2 – Web Server a.a 2020 / 2021

Sansone Lorenzo

0000887714

Analisi

Si vuole sviluppare un WebServer in Python per la gestione di un sito che mostra informazioni e servizi del Policlinico S. Orsola – Malpighi di Bologna

Requisiti

- Il web server dovrà permettere l'accesso a più utenti contemporaneamente
- Dovrà essere implementata l'interruzione da tastiera (o da console) dell'esecuzione del web server liberando la risorsa socket
- Nella pagina principale dovrà anche essere presente un link per il download di un file pdf
- La pagina iniziale deve consentire la visualizzazione della lista dei servizi erogati dall'azienda ospedaliera e per ogni servizio avere un link di riferimento ad una pagina dedicata
- Autenticare l'utente nella fase iniziale della connessione

Descrizione

Dopo aver avviato il server, essersi autenticato ed aver raggiunto il webserver dal browser verrà mostrata la pagina principale dove vengono indicati i servizi principali:

- Reparti
- Privacy cittadini
- Cerca un medico
- Numeri utili
- Libera professione

Policlinico S. Orsola - Malpighini

Home	Reparti	Cerca un medico	Privacy cittadini	Numeri utili	Libera professione	Download info pdf	Aggiorna contenuti
------	---------	-----------------	-------------------	--------------	--------------------	-------------------	--------------------

L'Azienda Ospedaliero-Universitaria di Bologna Policlinico S. Orsola - Malpighi e' un ospedale molto antico (il primo nucleo risale al 1592) ed e' sede della Facolta' di Medicina e Chirurgia dell'Universita' Alma Mater Studiorum di Bologna. Si colloca nel cuore della citta' di Bologna con un'estensione di circa 1,8 Km e un'organizzazione logistica che si struttura in 27 Padiglioni che ospitano le Unità Operative del Policlinico. Ogni giorno circa 20.000 persone accedono al Policlinico (personale dipendente, studenti e docenti universitari, pazienti, visitatori e fornitori). Il Policlinico e' centro di riferimento nazionale ed internazionale per diverse patologie; ogni anno sono organizzati, nelle sue sedi interne, eventi didattico-formativi ai quali partecipano professionisti di fama nazionale e internazionale. L'organizzazione interna e' strutturata in 9 Dipartimenti ad attività integrata (ospedaliera e universitaria), una tipologia di organizzazione che consente di assicurare l'esercizio delle attività assistenziali, didattiche e di ricerca, cui afferiscono le 87 Unità Operative. E' dotato di 1.515 posti letto con un organico di 6807 dipendenti, compresi i ricercatori e i medici universitari; vi si effettuano circa 49.000 ricoveri all'anno e oltre 3.300.000 prestazioni specialistiche per esterni.

I servizi si possono scegliere dalla barra di navigazione o dalla seguente lista

[Reparti](#)
[Cerca un medico](#)
[Privacy cittadini](#)
[Numeri utili](#)
[Libera professione](#)

Per ogni servizio è stata creata una pagina dedicata. Inoltre è presente una sezione download da dove si può scaricare un pdf che recensisce i servizi. È presente anche un tasto per aggiornare i contenuti.

Reparti

Verrà mostrata una lista dei reparti del Policlinico con relativi link per accedere al sito ufficiale e permettere di approfondire.

Cerca un medico

Verrà presentata una lista di medici a cui fare riferimento per alcuni reparti dell'ospedale.

Privacy cittadini

Saranno specificate le politiche di gestione della privacy per i pazienti che usufruiranno dei servizi dell'ospedale.

Numeri utili

Verranno riportati i numeri per poter contattare il call center e gli uffici principali della struttura.

Cerca un medico

In questa pagina sarà spiegato come funziona il servizio di libera professionista e come potervi accedere.

Dettagli

Autenticazione

Per fare l'autenticazione si procede da linea di comando una volta che il server viene attivato.

L'user da inserire è "user123" e la password "standardpass"

```
#variabile per l'accesso
logged = True
while logged == False:
    user = input('Inserire username:')
    password = input('Inserire password:')
    if user in users_data and password == users_data[user]:
        logged = True
        print('Logged with account:')
        print('Username:',user)
        print('Password:',password)
    else:
        print('Dati non corretti')
```

L'username e password vengono richiesti fintanto che non si inseriscono le credenziali corrette

```
Inserire username:user123
Inserire password:standardpass

Logged with account:
Username: user123
Password: standardpass
Porta collegata: 8080
updating all contents
finished update
```

Interruzione da tastiera

La seguente funzione permette di uscire dal processo tramite il comando Ctrl-C

```
# definiamo una funzione per permetterci di uscire dal processo tramite Ctrl-C
def signal_handler(signal, frame):
    print( 'Exiting http server (Ctrl+C pressed)')
    try:
        if(server):
            server.server_close()
    finally:
        # fermo il thread del refresh senza busy waiting
        waiting_refresh.set()
        sys.exit(0)
```

Librerie usate

Le librerie utilizzate nella scrittura del codice sono:

- signal** è stato utilizzato per gestire l'interruzione da tastiera.
- sys** per lanciare comandi che permettono di terminare il programma o prendere input da console.
- http.server** definisce classi per la costruzione di http server.
- threading** permette di utilizzare un'interfaccia ad alto livello sul modulo di basso livello dei thread. Utilizzato per aggiornare i contenuti periodicamente.
- socketserver** semplifica il compito di costruire server di rete.

Avvio del WebServer

Per poter avviare il server ed accedere al sito sarà sufficiente farlo partire o dall'IDE o digitando il comando:

```
py WebServer.py port
```

port è la porta che verrà usata per accedere al sito, se non viene specificata verrà impostata di default la porta 8080

Una volta avviato il server si potranno inserire le credenziali:

user: user123

password: standardpass

Infine si potrà accedere il sito dal browser specificando <http://localhost:port/index.html> oppure 127.0.0.1:port/index.html

Esempio con porta 8080

<http://localhost:8080/index.html>

127.0.0.1:8080/index.html

