**GRUPPO 6: WEB APP**

**Capaldo Gennaro, Diletto Alessandro Pio, Senatore Carmela Pia**

Il nostro gruppo ha creato una Web App che mette in comunicazione il nostro database con un’interfaccia grafica più semplice ed intuitiva rispetto al programma Python. Per la creazione della Web App abbiamo utilizzato un mini-framework Web scritto in Python: Flask.

**Installazione di Flask e connessione al database**

* Aprire il prompt dei comandi ed eseguire il codice “pip install flask”



* Nello script Python importare Flask con il seguente codice:



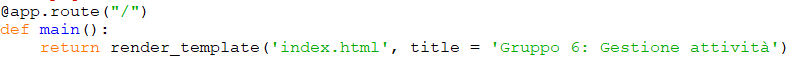
* Ora possiamo effettuare la connessione al nostro database e importare il resto dei pacchetti con il seguente codice:

Immagine che contiene testo

Descrizione generata automaticamente

In questa sezione per collegare il database al nostro script è necessario fornire a Python l’indirizzo del nostro db. In questo caso, visto che stiamo lavorando in un database in locale come host forniamo ‘localhost’, user = ‘root’, password = password per accedere a MySQL e infine il database a cui dobbiamo collegarci, in questo caso ‘hairdresser’.

**Come creare una pagina web in Flask**

* 

Con il seguente codice stiamo definendo il percorso di base del nostro sito.   
“/” = localhost:5000. Ritorniamo con la funzione main un template per visualizzare graficamente la nostra homepage.

* Immagine che contiene testo

  Descrizione generata automaticamente

Ora verifichiamo se il file eseguito è il programma principale e lanciamo l’app.

**Come utilizzare la web app**

* **Importare il database hairdresser.sql in MySQL**
* **Aprire il file app.py e configurare le credenziali di accesso per collegare il programma al database.**
* **Eseguire il file app.py**
* **Aprire un motore di ricerca e recarsi all’indirizzo “localhost:500”**

**Funzioni implementate sulla Web App:**

Immagine che contiene testo

Descrizione generata automaticamente

Con questa funzionalità possiamo inserire all’interno del nostro database una nuova prenotazione. In questo caso siccome dobbiamo aggiungere una riga ad una tabella e quindi apportare modifiche al database ci serviamo di una richiesta POST. Abbiamo creato con @app.route una pagina web all’indirizzo “localhost:5000/InsPrenotazione”. All’interno di questa route abbiamo creato la funzione Query1, che prende dei dati in input che vengono inseriti sul sito, gli assegna una variabile e tramite l’apertura di un cursore esegue la query che va effettivamente a modificare il database. Ovviamente per poter inserire dei dati sulla nostra pagina web ci siamo serviti di un codice html, in particolare abbiamo usato il <form>. All’interno del form è possibile creare degli spazi per poter inserire dati, i quali poi verranno utilizzati dalla funzione Query1 per effettuare la query.

Il codice html:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Analogamente alla funzionalità precedente, quest’ultima permette l’inserimento di un nuovo dipendente. Viene utilizzata anche qui una richiesta POST e tramite i dati inseriti sul web è possibile andare ad aggiungere una nuova riga alla tabella personale\_t.

Il codice html:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Con questa funzionalità invece aggiungiamo analogamente alle funzionalità precedenti un nuovo prodotto all’interno del database.

Il codice html:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

In questo caso siamo di fronte ad una richiesta GET, ovvero vogliamo interrogare il database e ottenere informazioni. In particolare, vogliamo sapere tramite l’inserimento dell’ID della prenotazione la data, l’ora e l’id del cliente che ha effettuato quella prenotazione. Abbiamo creato 4 funzioni.

* + Prima funzione:

Definiamo il metodo di richiesta GET e inseriamo in action il link a cui il form deve passare le informazioni appena inserite. In questo caso al link “localhost:5000/VisPrenotazioni1/search”

* + Seconda funzione:

Per assegnare a delle variabili le informazioni ottenute nel form è necessario passare a Python il comando “request.args.get(nome\_variabile).

* + Quarta funzione:

Apriamo un cursore, assegniamo ad una variabile la query da effettuare e la eseguiamo con il cursor.execute(nome\_query, [nome\_variabili]).

A questo punto non ci resta che mettere insieme i risultati della query con cursor.fetchall() e tramite la renderizzazione di un template e quindi di un codice html, tramite un ciclo for creiamo una tabella in cui inseriamo i dati.

Codice html:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Siamo anche qui di fronte ad una richiesta GET, in particolare si vuole visualizzare il flusso di cassa ottenuto dall’attività in un giorno stabilito dall’interlocutore. Allo stesso modo precedente prendiamo in input dal form la data e tramite queste funzioni passiamo la data scelta alla query.

Il codice html:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Con questa funzione stiamo facendo una richiesta GET, in particolare si vuole visualizzare tramite l’id del cliente tutte le prenotazioni che ha effettuato.

Il codice html:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

**Con questa funzionalità si vuole visualizzare il feedback che i clienti hanno lasciato ad un membro specifico del personale. Una volta aperta la pagina, visualizzerete anche un link per poter accedere ad un'altra pagina web in cui è possibile vedere i feedback relativi a tutto al personale.**

**Il codice html:**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Immagine che contiene testo

Descrizione generata automaticamente**

**Con questa funzionalità è possibile vedere i trattamenti effettuati da un membro del personale col vincolo di 2 date. Siamo sempre di fronte ad una richiesta di tipo GET.**

**Il codice html:**

**Immagine che contiene testo

Descrizione generata automaticamente**

Immagine che contiene testo

Descrizione generata automaticamente**qImmagine che contiene testo

Descrizione generata automaticamente**

**Con questa funzionalità si vuole visualizzare il fatturato che l’attività ha ottenuto dato il vincolo di 2 date. Siamo sempre di fronte ad una richiesta GET.**

**Il codice html:**

**Immagine che contiene testo

Descrizione generata automaticamente**



Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

**Con questa funzionalità si vuole visualizzare tramite l’inserimento di una anagrafica il totale che ha fatturato sulla base delle prenotazioni effettuate.**

**Il codice html:**

**Immagine che contiene testo

Descrizione generata automaticamente**