Laboratory 6

Development of a web application with Streamlit and MySQL

Goal

The goal of this practice is to develop a simple web application based on Python (Streamlit), capable of interacting with a MySQL database to perform queries based on user interactions.

Preliminary steps

This practice makes use of the Streamlit WEB server and the MySQL database offered by XAMPP. In order to carry out this practice, both services must be started.

Start MySQL server on localhost and start Apache

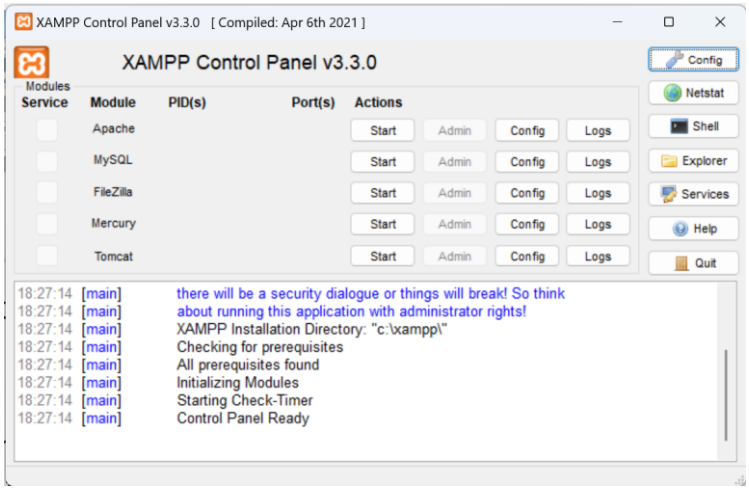
The execution of scripts with SQL commands for the creation and population of the database will be performed through the ***phpMyAdmin*** web interface of MySQL.

Before opening the Web interface of MySQL it is necessary to:

- Start the local Apache server;
- Start the local MySQL server.

Specifically, execute the following steps:

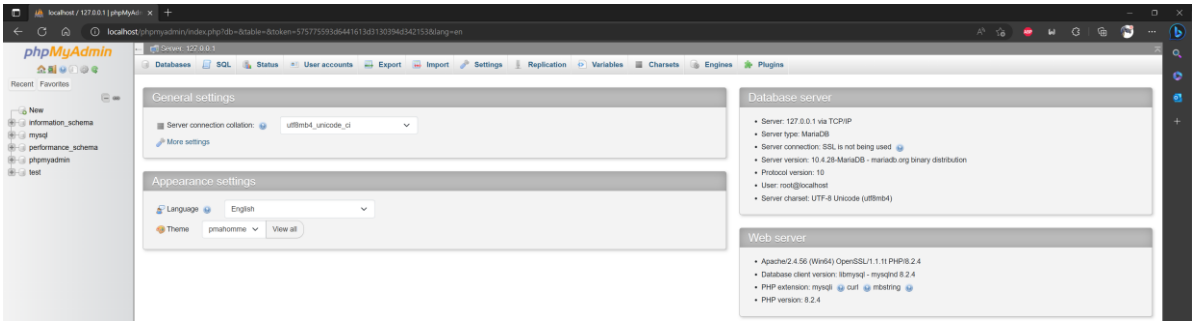
1. Start "XAMPP Control Panel".



2. Start Apache clicking the Start button in the row of "Apache" module.

3. Start MySQL clicking the Start button in the row of "MySQL" module.

4. Open the MySQLWeb interface clicking the Admin button in the row of "MySQL" module (the browser will automatically open the URL associated to the page of administration and MySQL querying. **Hint:** Don't use Microsoft Internet Explorer).



5. To execute a SQL script from the Web interface of MySQL:

- Select the "Import" panel.
- Select the file with the script you want to execute and click on "Go" button.

6. To execute the creation/population script more than once, you need to cancel any existing instance of the database, either directly from the "Database" panel or by including at the beginning of the script the commands for deleting the existing tables.

Running the Streamlit project

To create a new project you can use the *streamlitTutorial* repository with the *base* branch*:*

git clone https://github.com/Cryst4lDr4g0n/streamlitTutorial-eng.git

When creating the project and the environment remember to use a Python version > 3.7. To install the requirements:

pip install -r requirements.txt

To disable telemetry, verify or add the *$CWD/.streamlit/config.toml* file where *$CWD* is the folder from which Streamlit was started with the following option:

[browser]
gatherUsageStats = false

To run Streamlit:

streamlit run Home.py
python -m streamlit run Home.py

To disable the automatic launch of the browser use - *-server.headless true*.

Remember to update your credentials when connecting to the db. In the lab log in as *root* user, blank password (“”) and database *‘hotel’*.

Creation and population of the database

The database used during this practice is the same as the one you used for Laboratory 4 with some slight modifications.

You are requested to design the database for managing reservations for a hotel chain. The database must contain a list of rooms available for reservation. Each room is identified by a unique number, and are characterized by the floor, the surface area in square meters, and the list of amenities available (minibar, jacuzzi, balcony, etc.). The rooms can be single, double, triple, or suite. Among the different rooms available, for the suites is known the list of available spaces (bedroom, dining room, living room, etc.).

The database must contain a list of travel agencies, identified by an alphanumeric code and characterized by their address (street, number, ZIP code, city and state), telephone number and website (if available). You want to keep track of all the bookings stipulated by the various travel agencies for each room.

Different reservations may have been made for the same room, with the same travel agency or with different travel agencies. A reservation is characterized by a start date and an end date, a daily price. A room can only have one reservation in the same period of time. Each reservation is associated with a single room and a single agency.

In the CITY table there are the geographical coordinates of the cities.

Exercises

Create a multi-page application to display the main information contained in the database. Especially:

1. ***Home*** page: Header customization, general information, and student name
2. ***Agencies*** page: Visualize general information regarding available agencies and their geographical location
3. ***Room*** page: Visualize and filter available rooms
4. ***Reservations*** page: Show reservations' trends and prices

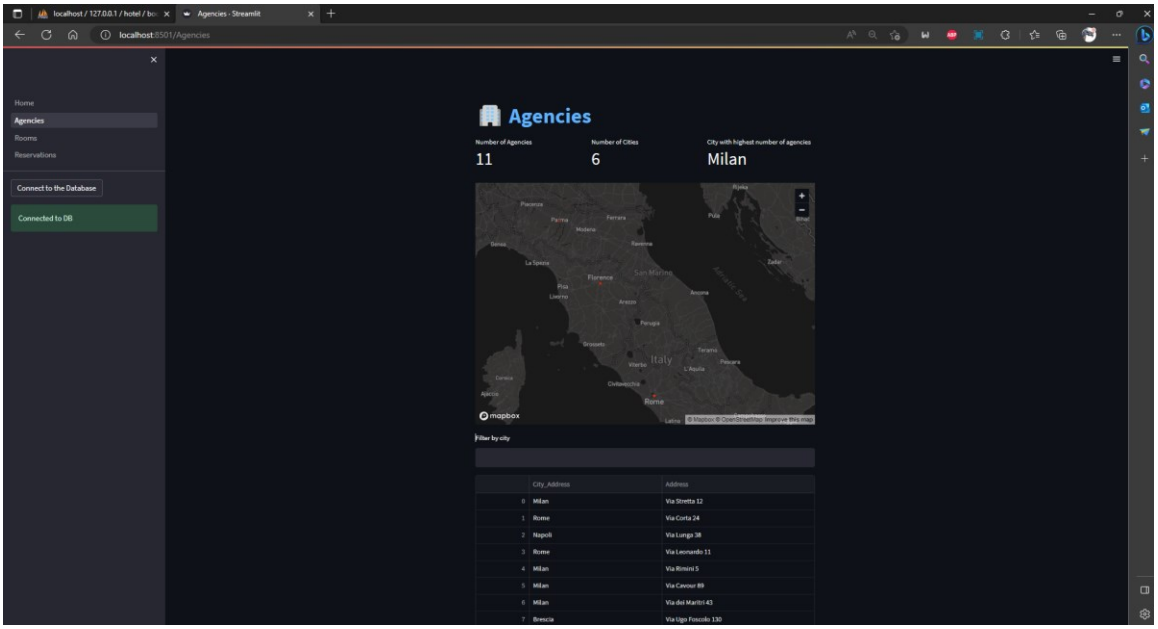
All pages should be customized with text elements (using markdown or Streamlit's pre-set widgets) so that headings, subheadings, and paragraphs highlight what is being represented. In addition, to make the vision and interface more intuitive and organized, the main layout elements must be used: expanders, columns, tabs.

Agency Page

1. Use 3 *metric* widgets to represent: number of distinct agencies saved to database, number of distinct cities, name of the city the highest number of registered agencies.
2. Use a *map* to represent the geographical location of available agencies.

3. Create a *table* to display agency information, merging the address information into a single column and the city (**Hint**: use *CONCAT(attribute1,attribute2)*).

4. Set a *text input* to allow the user to type as input a city for which he wants to find the available agencies in the table.



Rooms Page

1. Set up a "Filters" *expander* from which the user can select the preferred options for filtering the available rooms: the type (single, double, triple, suite, all) with a *radio button*, the options with a multiselect, if there is a kitchen among the available spaces with a *checkbox*.

2. Display in table format the rooms that meet the selected filters with the main information (CodR, Floor, SurfaceArea and Type).

OPTIONAL

3. Display a maximum of 5 rooms with the main information (CodR, Floor, SurfaceArea and Type), each represented separately by iterating through the list of results obtained (**Hint**: use *for index*, *row in df.iterrows()*).

4. Structure the information into two *columns*. In the first one instert the textual information, in the second associate for each room an image based on the type (one for single room, one for double room, one for triple room)

Reservations Page

1. Display for each month the code, floor, surface area and type of the room with the highest monthly average daily price. Represent with a line chart the trend of the average daily price compared to the months.