TD Project HTML/CSS + DB

Displaying Data from MySQL Database on the Web

Priyanka RAWAT

Project Overview:

This is an inter-disciplinary project that combines the key aspects of the two modules: "Bases de donnes (Database)" from the bloc DATA and "Technologies Web (HTML/CSS)" from the bloc DEV.

In this project you will create a website for viewing and publishing data from a MariaDB database. This involves demonstrating how to connect to the MariaDB database, retrieve data, and display it on a webpage. Eventually, you will be creating a data update page.

For this project, use the **VM "Debian9-SQL-2022" from the module "Bases de données**". You can access mariaDB database via VM on your laptop. You can restore an existing sql dump file.

You can do this project in a group of 2 or 3 students.

VM Debian9-SQL-2022:

You can use the VM installed with Apache web server, PHP, and MariaDB. PhpMyAdmin (on Apache + PHP).

Other similar tools are: XAMPP, WAMPServer etc.

General Guidelines for Compte Rendu:

- Upload all your project files + technical document on the platform Mootse.
- One upload for 1 group.
- **Technical document (pdf):** covers the details and explanation on your choices concerning the method/approach, tools and technologies, and answers to all the  "questions" in this subject. You can use any method or approach like PHP or Python or any other method for the backend. You should explain your technical choices in the technical document. You can add relevant screenshots etc., in your document.
- You must use HTML/CSS for this project (for example, HTML5 tags, semantic tags, CSS flexbox model etc.,).
- Follow the basic rule of separating the content and the style: the HTML and CSS.

Notes:

This project will result in 3 notes:

- 1 for the program part and its elements: 30%
- 1 for the website (features, quality, user-friendly etc.): 30%
    - User-friendly: in terms of design, structure, easy-to-navigate, RWD etc.
- 1 for the technical document explaining the functioning/operation and your technical choices: 40%

Consult the section "General guidelines for Compte Renud" for the guidelines concerning the compte rendu.

1. Context:

Figure 1 shows the integration of different elements such as a server-side scripting language (e.g., PHP) and MySQL database engine (a relational database engine), within a system. At one end of the system, you have a visitor to your website using a web browser to request a webpage. That web browser expects to receive a standard HTML document in return. At the other end of the system, you have the content of your website, which resides in one or more tables in a MySQL database that only understands how to respond to SQL queries (SQL commands).

This project focuses on how to retrieve data stored in a MySQL database and display it on a web page.

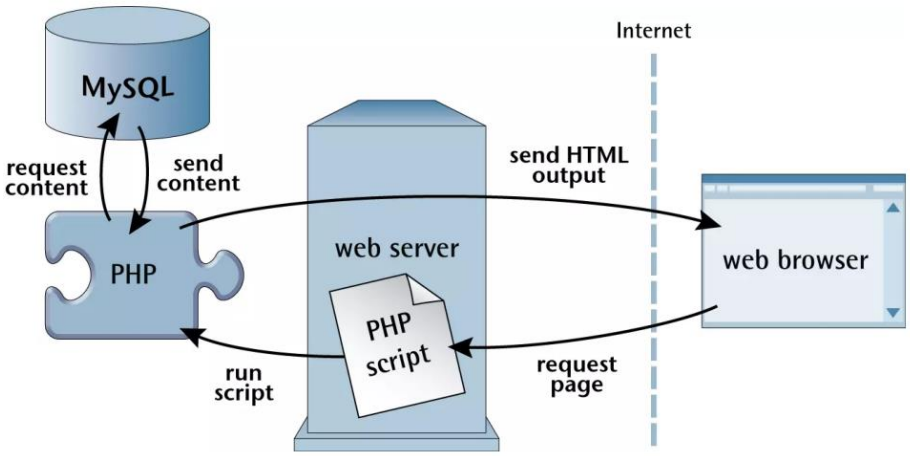

Figure 1: What happens when there is a visitor to a page on your website.

1.Explain Figure1. List and explain the different steps involved in the process when there is a visitor to a page on your website.

2. Identify the different components in Figure1. Explain the role of each component. Provide 2 examples of a Web Server.

3. What is the role of the PHP script in Figure 1? Provide an example of an alternative solution to PHP.

2. Objective:

Create a HTML/CSS website to view and publish/display data stored in a MariaDB database (a fork of MySQL). The data or content can be in the form of tables, forms, or any other type of data. Eventually, create a data update page.

Task: For example, you will be using the tools like MySQL and a server-side scripting language to create a website where users can view data from the database.

3. Webpage Features:

Some of the features that you should implement in your webpage/website.

- Restore the existing database and Display existing content from the database in a user-friendly manner.
- Populate (update) the database and display the new data from the database on your webpage.
- Create a data update page.

4. Methods, Tools and Technologies:

a. Front end: HTML/CSS for the front end

b. DB: mariaDB for the database.
You can access a mariadb DB via a VM (use the VM from the module "Bases de données": Debian9-SQL-2022). As a first example, you can restore the DB using a sql dump file. Then, you can populate a new database, or update the existing database to create a data update webpage.

c. Back end: for the back end, you can use any approach or method, for example:
   a. Method 1: PHP for the back end or
   b. Method 2: Python with Flask for the back end or
   c. Any other method

5. Example guidelines:

In this project you are free to use any method of your choice to create an HTML/CSS webpage to view and display content from a MariaDB database. For example, some approaches or methods you can use are PHP, Python etc.

5.1 Example Method 1: Using PHP for Back End

Environment set up on your machine/laptop:

- Ensure you have PHP (a server-side scripting language) and a web server (e.g., Apache / Nginx) installed.
- For example, to check if PHP is installed on your system, you can use the command:
  $ php -v
  $ dpkg -l | grep php

- For client-side language use HTML/CSS.
- You can access MySQL database server via VM Debian9-SQL-2022.
- **VM Debian9-SQL-2022:** Normally, you should have a VM installed with Apache web server, PHP, and MariaDB.

Database Creation: In general, one can use a tool like phpMyAdmin or the MySQL command line to create a database. In this project, you can access MySQL database via VM Debian9-SQL-2022.

4.Why and when do you need to use a database for your website? Explain.

Database Connection: Create a PHP script to connect to the mariaDB database and retrieve data. In your server-side script (e.g., PHP), establish a connection to the mariaDB database using appropriate credentials.

5. How do you establish a connection to a MySQL database in a PHP script? Give an example of a function that is used for database connection.

6. What SQL query you would use to retrieve specific data from a MySQL database table.

Retrieve Data: Write SQL queries (SQL commands) to retrieve the desired data from mariaDB database tables. Use PHP (or any other server-side language of your choice) to execute the SQL queries and fetch the data.

Format Data for Display: If required, process the retrieved data as needed (e.g., format dates, covert data types etc.) before sending it to the web page.
For example, you can create a php script to connect to the database, fetch the data and return it in json format.

Create the Website/Web page (HTML/CSS): Create an HTML/CSS front end for displaying data and a form or table to publish new data.

Integration for Displaying Data on the Web page: Integrate the server-side script (e.g., PHP script) with your web page (HTML front end) to fetch and display the data. Eventually, create a data update page.

5.2 Example Method 2: Using Python with Flask for Back End

You will need to first create a Python environment and install Flask and the MySQL connector library, for example, using `pip`.

Flask: is a micro web framework for Python
pymysql: is a Python MySQL library that can be used with MariaDB

$ pip install Flask pymysql

Database Connection: Create a Flask application and connect it to the MariaDB database using the MySQL connector. Retrieve data and provide API endpoints to access data.

Create the web page: Create an HTML/CSS front end to display data and a form to publish new data.

Integration: Integrate the Flask application with your webpage to display data.

6. Annex: Useful commands and Examples.

➔ You can try the following commands on VM "Debian9-SQL-2022":

```
$ mysql --version
10.5.13-MariaDB

MariaDB (a fork of MySQL).
$ mariadb --version
10.5.13-MariaDB
```

- Log in to the "MySQL server" as "root" user.
- Start the "MySQL client" to access the MySQL database server.

```
$ mysql -u root -p

MariaDB [(none)]> show databases;
```

- Select a database (change to a different database)

```
MariaDB [(none)]> USE humanressources;
```

- List all the tables in the selected database

```
MariaDB [humanressources]> show tables;
```

- Retrieve data from a table: retrieve all rows and columns from the specified table

```
MariaDB [humanressources]> SELECT * FROM countries;
```

- Describe a table's structure:
  `DESC table_name;`

- Insert data into a table (add new records to a table):
  `INSERT INTO table_name (col1, col2, …) VALUES (val1, val2, …);`

- Update data in a table:
  `UPDATE table_name SET col1 = val1, col2 = val2 WHERE condition;`

- Delete data from a table:
  `DELETE FROM table_name WHERE condition;`

- Create a new table:
  `CREATE TABLE table_name (col1 data_type, col2 data_type, …);`

- Drop a table: (Use it with caution)
  `DROP TABLE table_name;`

- Restore a database from a sql dump file:
  For example, for sql dump file **Lab12_TD1_SQL_humanressources.sql**

```
$mysql -u root -p humanressources < Lab12_TD1_SQL_humanressources.sql
```

➔ PHP + MySQL: We can use PHP to connect to and manipulate MySQL database.
➔ An example PHP script to connect to a database “connect_to_db.php”.

Improvise the code according to your requirements.

- Database connection: connect_to_db.php

```
<?php
//Replace with your database credentials
$hostName = "localhost";
$userName = "root";
$password = "your db password";
$databaseName = "name of your db";

// Create a connection
$conn = new mysqli($hostName, $userName, $password, $databaseName);
// Check the connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully”;
?>
```

- Fetch data from MySQL table: view_data.php
  - Include a db connection file (e.g., connect_to_db.php)
  - assign $conn to a new variable $db and table name to another variable $table
  - Define columns name in an indexed array and assign them to the $columns
  - Also, assign fetch_data() function to the $fetchData
- **fetch_data() :** this function accepts 3 parameters like $db, $table & $column and it contains MySQLi SELECT query that will return records in an array format by fetching from the database

```
// Include the database connection script (php script)
include("connect_to_db.php");

// Retrieve and display data

   $sql = "SELECT * FROM your_table_name";
   $result = $conn->query($sql);

   if ($result->num_rows > 0) {
       echo "<ul>";
       while ($row = $result->fetch_assoc()) {
           echo "<li>" .$row["content"] . "</li>";
       }
       echo "</ul>";
   } else {
       echo "No data found.";
   }

// close the database connection
$conn->close();
```

Example code: **Improvise the code according to your requirements.**

```
<?php
// Include the database connection script (php script)
include("connect_to_db.php");

$db= $conn;
$tableName="your table name";
$columns= [xxxxxx];
$fetchData = fetch_data($db, $tableName, $columns);

function fetch_data($db, $tableName, $columns){
 if(empty($db)){
  $msg= "Database connection error";
 }elseif (empty($columns) || !is_array($columns)) {
  $msg="columns Name must be defined in an indexed array";
 }elseif(empty($tableName)){
   $msg= "Table Name is empty";
}else{

$columnName = implode(", ", $columns);
$query = "SELECT ".$columnName." FROM $tableName"." ORDER BY id
DESC";
$result = $db->query($query);

if($result== true){
 if ($result->num_rows > 0) {
    $row= mysqli_fetch_all($result, MYSQLI_ASSOC);
    $msg= $row;
 } else {
    $msg= "No Data Found";
 }
}else{
  $msg= mysqli_error($db);
}
}
return $msg;
}
?>
```

7. Useful links (Références et apprentissage)

- PHP + MySQL: https://www.w3schools.com/php/php_mysql_intro.asp
- PHP: https://www.w3schools.com/php/default.asp
- https://www.mysql.com/
- HTML Standard (whatwg.org)
- http://www.w3.org/Style/CSS/
- http://www.w3schools.com/css/
- https://devdocs.io/css/
- http://caniuse.com/
- https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML