Lab 4 (in pair) DB intro

Hide Folder Information

Group Category

assignmentPair

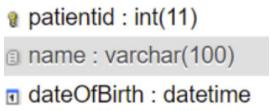
Group Name

g7

Instructions

Using two different origins like the previous lab, write a server side script to create a DB with one table in MySQL or MarinaDB (or other SQL based DB)to store patients' info and also to accept SQL queries on that table via http POST request to INSERT data or read via SELECT statements .

Your DB schema could follow the ERD below (patient)



At the time of creation of the table use `Engine=innoDB.`

Your nodejs script creates such table (above) every time they don't exist (every time the user presses the insert button described below, your script has to first check if the table exists, otherwise it has to recreate it .

Server1, index.html (origin 1)

A.

An Insert button to insert the following rows via POST request, every time pressed

```
'Sara Brown', '1901-01-01'),
'John Smith', '1941-01-01'),
'Jack Ma', '1961-01-30'),
'Elon Musk', '1999-01-01');
```

If the user presses the button multiple times, your app shall generate those rows multiple times and your table grows. The client displays a response from the server too

Note:

Your nodejs script creates such table (above) every time they don't exist (every time the user presses the insert button described below, your script has to first check if the table exists, otherwise it has to recreate it .

B.

A textarea to accept a SQL query to retrieve data and a submit button to send that query onto the server via GET (if we are reading). Then the client displays the response received from the server. The purpose of this text area is to let visitors of your website run SQl statements to ready from that table. However they may enter any SQL statements and it is **your responsibility** to ensure that harmful statements such as **DELETE, DROP, CREATE or UPDATE** cannot be executed. This protection must be enforced **on the server side** by configuring/programming the database user with **restricted privileges** — for example, a **read-only / SELECT-only role**. This is the only way that if a malicious or accidental SQL command is submitted, the database user should not have permission to modify or delete data.

On your server side (DB users) you need to take care of these privileges correctly so that doing update or drop is fully blocked; the only secure way is via setting limited privileges for the corresponding user

Server2, node js code (origin 2)

Responds to the POST or GET requests as appropriate (in this lab you are not allowed to use express or anything for routing, but for DB you can use any modules)

Deliverables

1. All of the server1 code in a folder named server1, all of the server2 code in a folder named server2. Copy both folders in a folder named YourTeam#Lab5 (e.g D3Lab5) and zip it as YourTeam#Lab5.zip and upload to learning Hub

At the learning hub comment section:

2. Post URLs of the hosted application at the comment section of learning hub

2.1 the url(s) of the client hosted at server1

2.2 the url of the API server hosted on the server2 with a sample GET SQL request as follows: YourDomain.xyz/...lab5/api/v1/sql/"select * from patient" (encode the whole url correctly so that clicking on in runs it correctly)

Entering this in browser address bar shall return all rows of your patient table

3. include Attribution to chatGPT if you used* (or any other resources, provided their license permits you)

Rubrics:

8/10 for working and bug free application

- 6 for the server side
- 2 for the client side

1/10 Working home page (with the two buttons and one text area)

Deduction rubrics

-3 if you do not properly prevent DROP or DELETE operations by applying the Principle of Least Privilege when configuring database user permissions.

-3 mark will be deducted if URLs are not posted correctly and hyperlinked at comment section of your learning hub submission) (you need to include https and press space bar when you post your url)

(-4) if the part client and server are from the same origin (e.g. if are hosted on the same server) (if you don't have a teammate, please find a free service to host the client side (server1) as it is all static files and you don't need a DB or backend scripting; remember it has to have https)

(-4) if you use JQuery or other libraries/modules not allowed in your program (express or any routing modules not allowed but yo ucan use any module for your DB)

10% mark deduction for each day late submission (0 after three days late submission)

-2 if you use var for variable declaration

-2 if you do not separately store the user strings at the top of your code or in a separate file

-2 for every question asked about your project code while marking your assignment in-person. Even if its your partner's code, you still need to know it

-4 if you are not accompanying your partner in-person while your lab being marked

Needless to mention, any other deduction will be decided while marking

***chatGPT**

as already mentioned in the course outline, using chatGPT is fine but you need to

1- mention that in BOTH your code comment and the learning hub comment . e.g. ChatGPT-3.5 (https://chat.openai.com/) was used to code solutions presented in this assignment. otherwise would be treated as Academic Dishonesty
2- you have to be familiar with every single line of the code)
3- you take full responsibility of the work you submitted

4- you cannot copy/paste from code generated by chatGPT, you will need to retype or rewrite .

All that is to benefit you

Due on Feb 15, 2026 11:59 PM

Submit Assignment

Files to submit

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File **Record Audio** **Record Video**

Comments