Sprint 3: Adding a Database

Directions:

As discussed in class we want you the student to be able to use a database in your project.

You will read the resources directly below and then you must implement the requirements further down on the page using the python library pymysql in conjunction with the work you did on the server in the past week, ideally you should not have to change anything on your front-end:

As always the lectures are available on the course website.

pymysql:

- https://github.com/PyMySQL/PyMySQL/blob/master/example.py
- https://pymysql.readthedocs.io/en/latest/

Databases:

- https://www.liquidweb.com/kb/create-a-mysql-database-on-linux-via-command-line/ https://www.tutorialspoint.com/mysql/mysql-create-tables.htm
- https://www.w3schools.com/sql/sql_insert.asp
- https://www.w3schools.com/sql/sql_select.asp
- https://www.w3schools.com/sql/sql_and_or.asp
- https://www.w3schools.com/sql/sql_distinct.asp

You should also remember that these are not the only available resources on the topic. Before coming to office hours or emailing us for help it is your responsibility as self interested learners to search. Google, and stack-overflow are your friends and the questions you will have are easily answerable one google away.

Database Setup: You set-up a Mysql database to connect to your project. Mysql is the leading relational database in the world.

- This involves creating a table for TODOS
- Ensuring that table has an "Auto Incrementing Primary Key"
- Ensuring whatever other information you need for your server and front-end is available.
- Point Value: 20

CREATE: When you add a new TODO on your front-end a request must be sent to the server, the server must create a new record in the database storing the todo.

- Example SQL: `INSERT INTO table_name(column_name,...) VALUES(val,...);`
- Point Value: 20

Update: On your front end, you should be able to change an existing TODO to be renamed to something else. When this happens a request must be sent to your server and the server must update the record that exists from when you created the initial todo.

- Example SQL: UPDATE table_name SET column_name = value;
- Point Value: 20

READ: When you refresh your page, close your browser and reopen to the front-end your front-end must send a request to the server which fetches all of the TODOS that have been inserted into the database from previous sessions.

- SELECT column_name,... FROM table_name;
- Point Value: 20

DELETE: On your front-end when you remove a TODO a request must be sent to the server that will delete the associated record from the database.

- DELETE FROM table_name WHERE column_name = value;
- Point Value: 20