אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev
جامعة بن-غوريون في النقب

בניית מערכות ממוחשבות מבוססות אינטרנט (WEB)
364-1-1381

Assignment 4

Part A

1. Create MySQL database and the table called '*users*' inside this database (the columns are according to your list of users from assignment 3).
2. Take the list of users you've created in assignment 3 and insert it into MySQL database inside a table called '*users'*.
3. Create a new **blueprint** called: '*assignment_4*'. Attach this blueprint to your website as it was shown in the lecture. Inside this **blueprint** define the route called '*assignment_4*'. This route will lead to a new template (inside the brand-new blueprint you just created) called *'assignment4.html'*. This template will contain four parts:
   a. **Insertion form**
   b. **Update form**
   c. **Deletion form**
   d. **List of users**
4. **Insertion form**: create a form that helps to insert a new user to the table 'users'.
5. **Update form**: create a form that helps to update specific user in the the table 'users' (name, age, etc.).
6. **Deletion form**: create a form that helps to delete any user according to its id or any other unique identifier from the table 'users'.
7. **List of users**: get list of users from the table 'users' and display it on the page below the Insertion, Update and Deletion forms.
8. Each submission of any of these forms (Insertion, Update, Deletion) must take us back to the route '*assignment4*', where we can see the full updated list of users.
9. Make it clear for the person that uses your website, that some changes were made in the database after pressing the 'submit' button. (For example: an alert, a small message at the top of the page, etc.)

Part B

1. Right down inside the report of this assignment the properties of a table 'users' in your DB (it's columns, their types, primary key, order).
2. Create a new route called: '*assignment4/users*' (no need for a new blueprint).
3. This route ('*assignment4/users*') will return a list of users from a table 'users' from your DB in a **json** format.
4. Create a new route called: '*assignment4/outer_source*' (no need for a new blueprint as well). There will be two simple forms to extract a single user from reqres.in website. To do this use the following link: https://reqres.in/api/users/{x} where

instead of {x} will be some number (1, 2, 3, …) that identifies the id of a user that was received from one of the forms.

5. First form will do it from frontend (fetch function - js). Second form will do it from backend (import requests - python).

Part C

1. Create a new route called: '*assignment4/restapi_users/< USER_ID>*' (no need for a new blueprint).
2. This route will return user's data in **json** format (use 'users' table from previous parts).
3. If there is no user with such USER_ID, the route will return some error message also in **json** format.
4. If no USER_ID was provided, this route will return some default user. The response comes in **json** format as well (the route: '*assignment4/restapi_users*').
5. Use *int* or *float* validation for type of USER_ID inside the route (as it was shown in the lecture).

Additional Remarks:

- You can add as many new templates and static files as you desire

Additional resources:

- Install MySQL. Example: https://medium.com/365datascience/installing-mysql-and-getting-acquainted-with-the-interface-cf0f98e599f2
- Install *mysql.connector*:
  https://www.w3schools.com/python/python_mysql_getstarted.asp
  https://pypi.org/project/flask-mysql-connector/
- MySQL cheatsheet: https://gist.github.com/hofmannsven/9164408
- Check syntax of MySQL queries: https://www.piliapp.com/mysql-syntax-check/
- MySQL Docs: https://dev.mysql.com/doc/connector-python/en/
- Fetch:
  https://www.youtube.com/watch?v=Oive66jrwBs&ab_channel=TraversyMedia,
  https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
- Requests: https://realpython.com/python-requests/
- Python and REST APIs: Interacting with Web Services: https://realpython.com/api-integration-in-python/

Submission of the assignment:

- You need to create a **short** report with descriptions of what you did in **bullet points**. It will help us to see the changes you did and where we can find them in your GitHub repo.
- Attach the link to your GitHub repo at the top of the report.
- Write down your name and ID at the title.
- The length of the report doesn't have to be more than one paper.
- The format of the report – **PDF** file.
- You need upload this report in a special submission box in the Moodle website.