

Database Systems Course (2019-2020) – Final Project

The goal of the project: design and implement a useful and interesting web application related to music. The application should provide functionality based on your desired target audience – for example, music producers might be interested in analyzing the characterizations of hit records (e.g. popular collaborators, songs length, song temp...), while guitar players would like to know what guitar models are used by their favorite bands. The possibilities are endless – be creative!

Project objectives:

- Design and create a music-related web-application, with an underlying MySQL database schema, where the information originates from **at least two sources**:
 - One of the sources should be accessed via an API (there are many music-related APIs online – contact me if you’re having trouble finding one)
 - The other source could be a different API, or any other publicly available information (for example, a .csv file)
- The database should be populated with at least 10,000 records, and have at least 5 tables.
- Your application should use at least 7 different SELECT queries:
 - One full-text query (make sure the data is indexed correctly)
 - 6 complex queries (e.g. nested queries, group by, aggregations, EXISTS, etc.)

Project requirements:

- Your application should **not rely on continued usage** (for example, interactions with different users).
- You should provide an assistance mechanism to help the user enter the correct input (e.g. auto complete, dropdown)
- Your server should be deployed on “delta-tomcat-vm”
 - Usage Instructions: <https://www.cs.tau.ac.il/system/django>
- The MySQL server you should use is “mysqlsrv1.cs.tau.ac.il”. Users and password will be given after you sent me an email with your group details (full names and ids)

Working Teams:

- Work should be done in groups of 4 students.
- After I register your group, you will be given a user for mysqlsrv1.cs.tau.ac.il
 - All DB interactions should be done with this user

Coding Guidelines:

- The server side should be written in Python
- You can use external libraries, as long as:
 - It works properly on the university server
 - It does not automatically create a DB schema

- It does not perform any database optimizations
 - It does not generate SQL queries automatically
- For the client-side, you can use any JavaScript library you want
- Your code should be readable
- Errors should be handled

DB Design:

- The DB should be designed according to the principals taught in class, including:
 - Meaningful names for tables, columns, indices, keys etc.
 - Primary and foreign keys
 - Use of indices to optimize your queries
- Explain why you chose your DB design:
 - Why do you think it is the most efficient for your needs
 - What are the disadvantages of other designs you have considered

Documentation:

- User manual:
 - What does the application do? (overview)
 - The screens of the application: how to get there and what are their features
- Software documentation:
 - DB scheme structure (also explain your choices while designing the DB)
 - DB optimization performed (e.g. your use of indices)
 - Description of your 7 main queries:
 - What does it do
 - How did you optimize this query
 - How does your DB design supports this query
- Code structure
- Description of your API + how did you use it
- External libraries you used
- General flow of the application

Submission:

- As mentioned above, your server should run on “delta-tomcat-vm” and the MySQL server should be “mysqlsrv1.cs.tau.ac.il” with the user you will be assigned
- You should submit your source code, the SQL code for creating the DB, and the documentation in a single .zip file in the following structure:
 - /SRC
 - /API-DATA-RETRIEVE (the code which inserts the data to your DB)
 - /APPLICATION-SOURCE-CODE (server code)
 - /CREATE-DB-SCRIPT.sql (an .sql script which creates the DB)

- /DOCUMENTATION
 - /URL-TO-THE-APP.txt (port and path on “delta-tomcat-vm”)
 - /NAMES-AND-IDS.txt (group members details)
 - /USER-MANUAL.pdf (see “documentation” section)
 - /SOFTWARE-DOCS.pdf (see “documentation” section)
 - /MYSQL-USER-AND-PASSWORD.txt (the MySQL user you were assigned)

Bonus (up to 10 points) will be given:

- For original ideas and features
- For exceptionally convenient and aesthetic UI

Tips and advices:

- Make sure you invest efforts in the right places. This is a Database project, and the focus should be on the database design, optimization, and queries rather than the UI
- Use your team members efficiently (what work could be done independently?)
- All APIs have a daily/hourly usage limits. Make sure you retrieve data efficiently so you're not exceeding the limit. Make sure to start this process in advance as it might take several days.
- As your application must be deployed on the university servers, make sure **in advance** that it is running successfully (e.g. does not require any external library and is able to communicate with the sql server)
 - For any system issue please contact system@cs.tau.ac.il and address the course staff if your issues are not resolved

Good Luck!