NYU TANDON SCHOOL OF ENGINEERING

CS-GY 6083 – B, Spring 2022
Principles of Database Systems

Project - Part II, Requirement Document [100 points, 15% weight]

General description

In the second part of the project, you have to create a web-based user interface for the Database schema designed in the first part of project. In particular, users should be able to register, login, and can perform business activities (CURD – Create (Insert), Update, Read, Delete records). Please note that the users could be either customers or employees of WOW, and they have different authorization on data.

Note that you have more freedom in this second part of project to design your own system. You still have to follow the basic guidelines, but you can choose the actual look and feel of the site, and offer other features that you find useful. In general, design an overall neat and functional system. If you are doing the project as a group of three, note that every students of each team have to attend the demo and know ALL details of the design. So, work together with your partner, not separately. You can revise your design from the first part of project as needed.

Users should be able to perform all operations via a standard web browser. This should be implemented by writing a program that is called by a web server, connects to your database, then calls appropriate stored procedures that you have defined in the database (or maybe send queries), and finally returns the results as a web page. You can implement the interface in different ways, you may use frameworks such as PHP, Java, Django, Python or Ruby on Rails to connect to your backend database. Contact the TAs for technical questions that you may have. The backend should be a relational database (MySQL recommended) based on the schema you designed in the first part, with suitable improvements as needed. The APIs of your website is recommended to be designed with RESTful style. For sensitive data such as password, it should be stored in database after encrypted.

Your interface must take appropriate measures to guard against SQL injection and cross site scripting attacks. To prevent SQL injection, you could use stored procedures and prepare statements (if your programming language supports them). If your language doesn't support prepared statements, your code should check and sanitize inputs from users before concatenating them into query strings. Some languages provide functions, such as PHP's htmlspecialchars, to help with this. You should also define appropriate transactions to make sure that multiple users can use the site at the same time. Also, one thing to consider is how to keep state for a user session. Make sure to address these requirements (protection against SQL injections etc., and concurrency) in your development.

Project Demo: 80 Points

Every group is expected to demo their project to one of the TAs at the end of the semester. Project demo will be conducted remotely in Zoom session. TAs will send out notification to register for project part II demo and accordingly group will register for the time slots on a first come first service request basis. All students of each group need to be present at the demo. If you use your own installation, make sure you can access it during the demo. One popular choice is to use a local web server, database and browser on your laptop, and share your screen with TAs during the demo. (In this case, your project can just run locally on your laptop, but you can also make your project Cloud-based). Grading will be done on the entire project based on what features are supported, how attractive and convenient the system is for users, your project description and documentation(important!), and the appropriateness of your design in terms of overall architecture and use of the RDBMS. Make sure to input some interesting data so you can give a good demo. You should also be able to show and explain your source code during the demo.

Project Report: 20 points

All students of each project group is required to submit a project document (PDF only) before the deadline of Project Part II on NYU Classes. The project document should have following as minimum,

- Name and section of the course, Name and Student ID of team members, and Submission date
- Table of content
- Execute summary (about a page), that describe the business case, approach that you have used towards business solution, and how it will help business improving business performance. Logical and Relational Model design along with any assumptions made in support to designs.
- Brief details of software, programming language, and database used
- DDL file content, generated from Data Modeler Relational Model
- List of tables, and total number of records of each table
- Screenshots of some sessions, pages, menus of your Web Application
- Details of security features that you have implemented on Web Application development

- Lesson learned, detailing your reflections about project work, what have you learned, what went well and what did not. Constraints you faced, if any (e.g. time management, coordinating project with team member remotely etc.)

- Business analysis with 6 SQLs using your project data.
  Write SQL queries using each of following,
  Q1) Table joins with at least 3 tables in join
  Q2) Multi-row subquery
  Q3) Correlated subquery
  Q4) SET operator query
  Q5) Query with in line view or WITH clause
  Q6) TOP-N query
  For each of above queries use proper column alias, built-in functions, appropriate sorting and submit following three items for each of above queries.
  A1) Select query
  A2) Result of the query
  A3) Brief explanation about the business information you intend to retrieve

Extra Credit

There will be opportunity to get extra credit by implementing cool extra features. Extra credit is up to 6. EXTRA CREDIT WILL BE APPLIED TOWARDS SCORED POINTS OF PROJECT PART II ONLY. IF YOUR SCORE GOES OVER 100 POINTS AFTER APPLYING EXTRA CREDIT, THEN OVERFLOW SCORE WILL BE APPLIED TOWARDS ANOTHER PRIOR ASSIGMENT/PROJECT/EXAM WITH 15% WEIGHTAGE.

Extra Features
Please note that the extra points are for those who want to add more interesting features than basic requirements. Typical extra features could be:

- Better design to use Cache/Containers/Serverless etc. to make your website more robust with high availability and scalability.

- Building correct index on database to deal with the query with high frequency. In this case, you need to show us how and why you build an index, what and why this index can help to your system, and the process of your experiment/analyze to improve the performance of system by indexing.

- Interesting data visualization and methods for user to interact with data

- Security check on password reset, stored procedures, user functions, history tables etc.