Course Project

Description

1. Consider the design of a database for a website for managing driveway-sealing for a contractor David Smith. A client will need to register with the website with the following information: first name, last name, address, credit card information, phone number, and email. A unique client ID will be generated for the client when they register themselves. The workflow starts from a client to submit a request for quote, in which the client can submit the request for sealing ONE driveway with the following information: the address of the property, the square feet of the driveway, a proposed price for the work, and five pictures of the driveway from different angles. A note will also be sent as part of the request to allow the client to add whatever information they see fit. After receiving the request, David Smith can either reject the request with a note so that the client knows why the request is rejected, or respond with a quote in which an initial counter proposal of price is sent for the work as well as a time window (from when to when) that the work will be performed. After receiving the quote, the client can accept the quote immediately, which creates an order of work that forms the contract between David Smith and the client. Otherwise, the client can resubmit the request with a new note commenting and negotiating the terms for the quote: lower price, another day for the work. After receiving the revised request of quote, David Smith can modify the price or work period and send the revised quote to the client. Such negotiation can be looped as many times as necessary until the client accepts the offer or one of the parties quit from negotiation. In the case of quit, no agreement will be reached, the quote will simplify fail as the final step and will be closed. A new quote request will need to be submitted thereafter if the client changes their mind and like to pursue a new round of negotiation. After the work specified in the order of work has been completed, David Smith will be able to generate a bill from the order of work and sends it to the client. After receiving the bill, the client can either pay the bill immediately using their credit card, or reject it with a note to explain the concerns and complaints. David Smith can resubmit the bill with his own note explaining or addressing the above concerns, including applying a discount if necessary. The negotiation on the bill can be sent back and forth as many times as possible. The bill will be concluded only when the client pays for the bill; otherwise, the bill will always be pending in a dispute status, possibly the two parties need to resort to external process such as a lawsuit, which is out of the scope of this project. In this project, you will need to model each response of the quote and each response of the bill as they are the evidence in a lawsuit. In terms of interfaces, we need to implement the following functionalities: 1) **Dashboard for David Smith:** a) check all incoming request of quote and examine their content, responses and status and b) check all orders of work, their content and status; c) check all bills of work, their content and status. David Smith should also be able to respond to the most recent response of quote and bill. A report of revenue can also

be generated easily for a particular period. 2) **Dashboard for clients**: a) check the information of quotes, orders and bills. A client should be able to respond to the most recent response from David Smith.

Projects from previous semesters:

1) **https://www.youtube.com/watch?v=nJa_pHEDbFE**
2) **https://www.youtube.com/watch?v=NBJ5TSXiqfU**
3) https://rumble.com/vqdauj-csc6710-group-5-project-part-3-demo.html
4) Previous example project sample video (Courtesy of Rajiur Rahman)

For all parts of this project, your system must be web-based. Some simple GUI interfaces are required for each functionality. **All functionality must be performed via the interface of your system, direct SQL statement execution via any tools (MySQL workbench) can only be used for debugging purposes.**

Workload for this project:

1. Draw an E-R diagram for the system, in particular, use arrows or thick lines to represent constraints appropriately. Write down your assumptions and justifications briefly and clearly. Translate the above E-R diagram into a relational model, i.e., write a set of CREATE TABLE statements. In particular, specify primary key, foreign key and other constraints whenever possible.
2. Implement all interfaces and functionality described above and then implement the following functionality for the Dashboard for David Smith.
3. [**Big clients**]: List the clients that David Smith completed the most number of orders. List one client if there is no tie, list all the top clients if there is a tie. Please verify your query result is correct after you have done the query to have full credit. In your submission, please show the result and explain why that result, change the database, and then you get another result, explain the new result as well, and why this change of result. The same requirement for all following queries.
4. [**Difficult clients**]: List the clients who sent three different requests to David Smith but then never followed up afterwards.
5. [**This month quotes**]. List all the AGREED quotes in this month (Say December 2024).
6. [**Prospective clients**]: List all the clients that have registered but never submitted any request for quotes.
7. **[Largest driveway]:** List the locations of the driveways that have the largest square feet that David Smith ever worked, list all locations if there is a tie or list just one location if there is no tie.
8. [**Overdue bills**]. List all the bills that have not been paid after one week the bill is generated.
9. [**Bad clients]**. List all the clients that have never paid any bill after it is due. Suppose the bill is due one week after it is generated. A client is not bad if there is no bill for them at all.
10. [**Good clients**]. List all the clients that have paid their bills right after the bills are generated, meaning they paid the bill within 24 hours when the bill is generated.

Part 4 (only for honor students)

1. Set up your project on a remote server (there are some free hosting services and commercial hosting services such as Amazon) that is accessible from the Internet so that other people can access your website remotely. Optionally, you can have an official domain name by registration at DNS authority such as godaddy.com.

2. Write a professional user's manual so that a user will know how to use your website. The user's manual is accessible from the Website too.
3. Write a developer's manual so that future developers of the system will know how to extend the functionality of the system. The manual can include architectural diagrams, API descriptions, and suggestions for future functionality.
4. The TA and the instructor will examine the above three items carefully to determine whether the honor students pass the honor credits requirements.

How to submit:

1. Submit a PDF that contains: 1) A title: "project part 3", 2) project partners: both parners' access IDs and names, list "no partner" if you work alone; 3) a URL to your presentation; 4) A list of SQL statements in a file sql.txt, with each one corresponds to the solution for each query.
2. The video needs to be presented by both partners. We only need you to record your screen and your voice for the project demo, not your face. Please do not show and explain the source code in the video, you only need to demonstrate the functionality, including failure handing.
3. In your video, for each query, you need show a result, then update the database and show a new result. Please explain both the old result and the new result for each query as well as how the change happens. Both results cannot be empty.
4. Please do not show and explain your source code (except your SQL statements).
5. To save the time for the video, you can pause the recording when you are doing housekeeping work (such as changing the database or debugging), and then resume the recording when you are ready again.
6. A zip file that contains all your source code. Sql.txt. and a readme file that lists the instructions to install, configure and run your project. In the readme file, please clearly list the contributions made by each partner to this project, including the number of hours working together.

Start your project early, and ask questions if you have doubts. Do not wait until the last minute.

Good luck!