

1. Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).

We completed many of our goals for the project, including adding data about a user and crimes related to the user. We presented a visual component to display the severity of crimes in a given area. In our project proposal, we stated that we would map the severity of crimes based on colors including red, yellow, and green. Although we created such a visual component, there were several changes in our direction of the project from stage 1 to stage 6. We redesigned our front end to display various information from the data, such as victim count, query history, and crime data, which was not our original proposal. However, we planned to implement geofencing in our project which would have allowed greater user interaction with the application to interact with map boundaries and crime areas in Los Angeles. We instead created a main page which allows users to enter a longitude, latitude, and radius to ultimately display the severity of a crime in an area, color coded with red, yellow, and green square.

2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.

Our application achieved several functionalities and provided further insights regarding the Los Angeles crime data. Our application successfully drew information from various data tables in order to allow users to search severity of crime data in a specified location, view areas in Los Angeles and access the number of crimes in the area, and view weapons associated with the crime and the number of crimes connected to a specific weapon. We also added a new feature and functionality that was not initially planned in the project proposal in which we allow users to further interact with the application and access query history. The query history page allows users to enter a user name and password which then allows the user to see and past queries that they performed such as adding data, updating data, and finally removing any user or data. Additionally, the application was shown to be of use for individuals to have access to the crime data in a given area. The color coded crime severity visualization provides use as the colors show how safe or dangerous a particular region of Los Angeles is. Although we implemented the color coded severity visualization, we did not accomplish a component of developing a crime map to color code real time severity and crime data. We initially planned to display in the frontend a map of Los Angeles with all regions colored red, yellow, or green based on the severity of crime in a given location. Although we did not complete this task, we instead

created a smaller color coded square to model the severity of data within a given longitude, latitude, and radius.

3. Discuss if you change the schema or source of the data for your application

The source of our data remained the same throughout the entire project, which was the crime data in Los Angeles. However, we ended up changing our schema later in the project so that it made more sense for our application. We originally had tables for each component of the crime data such as the premise, area, weapon used, etc. These all had different primary keys. However, later in stage 3 we realized that our original foreign keys that we had set up for our tables were not unique, so we redesigned our database so that each tables' primary key was the DR_NO, which is a unique identifier for a crime committed. We then later changed our database design again in stage 5 to include fewer tables because we didn't need all the data. We ended up with the tables: Areas, Crime_Types, Query, User, Victims, and Weapons. We ended up creating a new table "Query" which stores the outcomes from our stored procedure.

4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

Throughout our project, our team was committed to designing a robust and efficient database system. We took a thoughtful and collaborative approach to developing our ER diagram, and as a result, we did not encounter any significant differences between our original and final designs. We were able to maintain a high level of consistency and coherence in our approach, which helped us achieve our goals effectively. However, as we delved deeper into the implementation stage, we discovered some areas where we could streamline and optimize our table structure. After careful consideration, we made the decision to combine certain tables to create a more concise and efficient database. This allowed us to simplify our data management processes, which ultimately improved the overall functionality and performance of our system. Overall, we believe that our final design strikes the ideal balance between functionality, efficiency, and ease of use. By working collaboratively and remaining open to feedback and suggestions, we were able to develop a robust and effective database system that meets the needs of our project.

5. Discuss what functionalities you added or removed. Why?

We initially planned to create a map visualization of crime data in Los Angeles. Within this map, we planned to create a drop down menu which would essentially allow users to select a type of crime and the map would reflect which areas have the certain crime and count the number of crimes within each location in Los Angeles. We additionally planned in our project proposal to display how many crimes occurred within a specified time range. However, upon completing each stage of the project, we made changes to our overall project proposal in order to complete each stage in a timely manner while also satisfying project requirements. As a result, we modified our project design, however we visually represented the severity of crimes within a certain radius with a square that shows the crime severity based on green, yellow, and red colors as well as the number of crimes that occurred in the area. Additionally, we redesigned our application to display various crime data upon further looking at the data in the database. We decided to instead create a webpage with several tabs in which a user is able to enter latitude, longitude, and radius in order to display severity of crime in the location. We also created a page to show the number of crimes in a certain Los Angeles location, and a page to display the weapons used and the number of crimes that the weapon was used in. Additionally, we decided to implement a user query page as this functionality would allow a user to view identifiable information. A user is asked to enter their username and password, and then a page displays any changes made to the user's data. This was a new functionality that we added as this would allow users to more directly engage with the application and see changes and updates made to that user's information.

6. Explain how you think your advanced database programs complement your application.

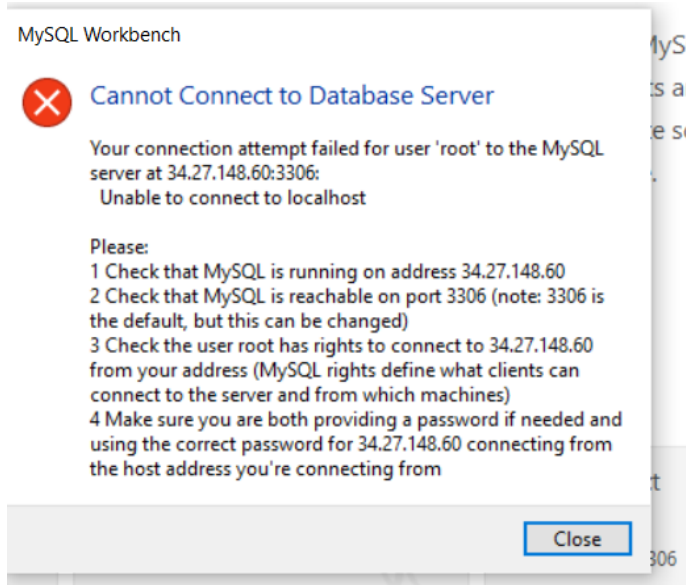
Throughout the project, we interacted with MySQL in order to create advanced queries, stored procedures, and triggers. Our advanced queries were used directly in the front end of our application in order to display information related to the data in the Los Angeles Crime database. Upon viewing the various tables in the provided crimes database, we implemented one advanced query to display the number of crimes in a given area of Los Angeles. Additionally, we implemented another advanced query to display the weapon type and the

number of reported crimes associated with the weapon. Both advanced queries complemented our application as we were able to display important and useful information regarding crimes in Los Angeles to users of the application. Additionally, our stored procedure prompts a user to input longitude, latitude, and radius in order to display the three most common weapon types, three most common crime types, crime count, and severity of crimes within the given location. This stored procedure was utilized in our application's main page in order to provide a visual representation of crime severity in a given location by displaying a green, yellow, or red box with the crime count in the specified location. Additionally, our created trigger was utilized in our query history page of the application in order to allow a user to interact more directly with the data and application. A process is triggered when a user's data is deleted and their record and user id is also deleted. In the query history, upon entering the user's username and password, the user has the option to see any changes made to their data, while also having the ability to delete their record. This trigger is of use to the application as removing the user id when the user record is deleted ensures that the database is storing the proper and updated information.

7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project

Throughout completing stages of the project, we faced various issues with the database, frontend and backend, however we worked collaboratively as a team to solve problems that we encountered.

Emily: As we were unfamiliar with GCP, we faced issues in properly setting up GCP and ensuring that our SQL workbench was properly connected. Upon creating the Google GCP instance, we struggled for a while to create a database instance in MySQL as we continued to face errors. We encountered the error "cannot connect to database server." I figured out that the GCP was not properly connected to MySQL due to invalid ip address connections. Since I was in charge of creating the GCP instance, I navigated to the networking tool where I was able to enter each team member's IP address in order to properly connect to GCP and MySQL workbench. We learned that each time MySQL workbench is opened, each user's new ip address must be modified in Google GCP for proper connection. Solving this problem allowed us to access, create, and interact with our database in MySQL workbench properly.



Tiffany: One technical issue we faced was cloning the github repository. We were faced with a security issue presumably because of security put in place on the github organization. We eventually figured it out by running “gh auth login” which allowed me to authenticate with github in VSCode. This then allowed me to clone the repository, however my other team members were still not able to clone the repository which they resolved by working on the github codespace.

Prem: One notable technical challenge that our team encountered during the development of our full-stack database application was related to data synchronization. As our application relied on real-time database updates, we faced issues with backend updates not reflecting on the frontend in a timely manner. This was particularly problematic when we attempted to query or update the database and encountered delays in rendering changes on the frontend. To address this issue, our team devoted significant time and resources to exploring potential solutions. After careful consideration and experimentation, we arrived at the decision to leverage the useEffect hook in order to fetch data from the API. Specifically, we developed a trigger that would call the useEffect function each time a specific event handler was triggered, allowing us to ensure that database updates were being rendered in real-time.

Scott (whhuang4): During the integration of the frontend and backend, one of the major technical challenges our team faced was related to data transfer. This issue was prevalent throughout the entire data pipeline, spanning from the database to the backend and finally to the frontend. The following points detail the challenges and solutions we encountered, which could be helpful for future teams working on similar projects or maintaining our project:

1. Handling pipeline breakdowns: If the database or backend were to go down, the entire data pipeline would be disrupted. As a result, when encountering issues in the program, the first step should be to check the availability and functionality of the database, backend, and frontend to ensure they are operating correctly.
2. Consistent naming conventions: Inconsistencies in data naming, such as "user_name" vs. "user_id", led to improper data transfer. To address this, it is crucial to maintain a consistent naming convention across all functionalities and APIs.
3. Understanding HTTP request formats: One particularly difficult issue involved the data format of HTTP requests. Initially, I used Headers to handle data transfer in GET requests. However, the data was not being properly sent from the frontend to the backend. After thoroughly examining the data pipeline and studying HTTP requests, I discovered that switching from Headers to Params resolved this issue. This experience highlights the importance of understanding HTTP requests when working on projects involving data transfer.

By addressing these challenges, future teams can ensure a more efficient and seamless data transfer process in their projects.

8. Are there other things that changed comparing the final application with the original proposal?

As with many projects, there were changes made between the original proposal

and the final application for this particular project. Two significant changes that were made to the project include: The implementation of the map view and geo-fencing was ultimately deemed too time-intensive, and therefore the team decided to provide the user with a list of information rather than a visual. This change allowed the team to focus on other important aspects of the application while still providing users with valuable information. However, it's worth noting that this is a feature that could be implemented in the future. The creative component of the application was originally planned to be a time range to detect crime. However, the team ultimately changed this to a color that maps to a crime intensity level, with red indicating high intensity, green indicating low intensity, and yellow indicating moderate intensity. This change allowed the team to provide users with a more intuitive and accessible representation of crime levels

in the area. Overall, while some changes were made to the project between the original proposal and the final application, these changes ultimately allowed the team to deliver a product that was both valuable and feasible within the given timeline and resources.

9. Describe future work that you think, other than the interface, that the application can improve on

In the future, our application could be improved by including data from all across the country rather than just Los Angeles. This would make the application more useful to users that are outside of Los Angeles, but still want to know about crime data in their city. Another part of the application that could be improved would be being able to update the database with new current data, rather than just data that users put in. In reality, there is a lot of crime being committed, but our database is just based on the data we were given and it isn't being constantly updated in our database.

10. Describe the final division of labor and how well you managed teamwork

Although we had our assigned tasks in our project proposal, as we reached each stage, we worked collaboratively on the given tasks at hand. Overall the division of labor included Emily completing the GCP setup and MySQL connection. Tiffany and Emily worked on creating the advanced queries. Scott, Prem, Emily, and Tiffany worked together on importing the numerous data and tables from the Los Angeles crime data into MySQL workbench as there were several tables that needed to be imported at a time. Tiffany and Emily worked on the frontend of the application in stage 4 and Prem and Scott worked on the backend while also connecting the frontend and backend of our application to work properly. All team members worked on finalizing the frontend and backend with Scott also working on creating the triggers and stored procedure to integrate into our final application. Throughout our project, we worked collaboratively as a team to aid each member in completing any challenges that were faced. Most of our team meetings occurred in person which allowed us to work more efficiently and allowed us to help each other when needed.