PROJECT REFLECTION REPORT

1. **Please list out changes in your project directions if the final project is different from your original proposal (based on your stage 1 proposal submission).**
   We believe that we have retained the basic functionality that was promised in the original project proposal submitted for stage 1. We had promised that the scope of the project will include developing an interface for a prospective or current EV owner to access data on EV Vehicles and their maintenance, and our current interface makes that possible with the search option. We had set out with an intent to focus more on the search patterns of users. However, we didn't take those into consideration during the development of our web app.

2. **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**
   Our web app works as an effective tool to access the data on electric vehicles across locations. It also has data on the charger compatibility and maintenance of these vehicles, that can be put to effective use. However, we believe that our web app failed to provide the location-centric functionality that was implied in the proposal. We do believe that a few minor tweaks to the code base can make the web application suitable for such location-centric functionality also. We also failed to provide a mechanism for users to save their lookup information.

3. **Discuss if you changed the schema or source of the data for your application**
   We had set out with the assumption that enough data exists on the internet to supply our database. However, after thorough research, we concluded that there is an acute shortage of data when it comes to electric vehicles, and hence we had to resort to data simulations and synthetic data for the purposes of our project. In terms of Schema, we acted on all the feedback given to us by our project mentor, Hongtai, and we simplified the database, by reducing the number of tables from 9 to 6, getting rid of extraneous relationships and unnecessary weak entities, while still maintaining our promised functionality.

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?**

   We were using a highly complex ER diagram before we began the implementation on GCP. Our mentor, Hongtai suggested that we simplify the database and use fewer entities. Some key differences include the existence of the maintenance table versus including it as an attribute in the vehicle entity. We also had a separate login entity (weak entity) to account for user login, but now the user entity takes care of that. As visible, we also got rid of many extraneous and unnecessary entity relationships for the sake of simplicity. We believe that the current implementation is a more suitable design, as it is simple and still serves the same purpose as the original, complex ER design.

5. **Discuss what functionalities you added or removed. Why?**
   As required by the rubric of the project, we added a few customizable searches, and views. These included using advanced SQL queries and stored procedures. We also added the functionality of providing a database-wide search engine that lets users search for different vehicles and chargers. We, however, removed the cost calculator, as was promised in the project proposal. We also removed the testimonials and news section that was mentioned as the creative component in the project proposal.

6. **Explain how you think your advanced database programs complement your application.**
   Our advanced database programs (modified after the final demo) provide additional functionality for our web application. Our stored procedure called getBrandRanking allows a user to see how different brands are ranked according to a specified rubric depending on their price, charger type, and maintenance cost. Our trigger, on the other hand, accounts for a (hypothetical) government policy to provide incentives for car companies. We have tried to simulate some real-life demands and situations that might arise and can be handled by advanced database programs.

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.**

- Technical Challenge 1: Lack of Data: We found out that there isn't enough structured data in the EV dominion. The data that exists is scattered. We had to spend a lot of time cleaning the data according to our needs. We would advise any new member to go over our data schema and align new data with the set structure.
- Technical Challenge 2: Complexity of Schema: We started out with a very difficult schema that would have been impossible to implement using the relational model. Hence, any new modifications to the schema must focus on maintaining its simplicity, and proper relationships must be chalked out to still preserve the functionality of the web application.
- Technical Challenge 3: Using GET and POST requests: Over the course of the development of the project, we realized that get and post requests need special attention for customization to achieve the purpose that they are intended for. We ran into issues with server requests for activities like Login and Search. One must ensure that any additional functionality uses the right category of requests that is coherent with how the other components of the app have been implemented.

8. **Are there other things that changed comparing the final application with the original proposal?**
   As mentioned earlier, we have implemented our web application in a similar dominion as was proposed in the proposal. However, we altered some fundamental functionality as we advanced with the development of the project. We got rid of the calculator and navigation functionality, and instead put our focus on search and rating functionality. Similarly, we created a completely different interface for employees, which was not proposed in the project proposal draft.

9. **Describe future work that you think, other than the interface, that the application can improve on**
   We continue to find more structured and accurate data to supply our databases. We will also focus on refining our database schema to simplify it and more accurately represent the relationships between entities. Right now, we rely on rudimentary server-client requests to implement our functionality. Going into the future, we will also use more efficient and complex server-client requests. Our website has no implementation for security, as of now. However, we can enhance the security of our website too. Finally, we plan to include visual components such as map-APIs, graphical representations, etc to make our website more informative and applicable.

10. **Describe the final division of labor and how well you managed teamwork.**
    The withdrawal of our team captain, Francisco created a few hurdles when it came to managing the workload. However, the team members stepped in to take responsibility for Fransisco's part of the work. Since the database was hosted on Purvansh's GCP account, he was responsible for all the backend and connectivity tasks. Jinzhi helped in conceptualizing the ER and schema for the database, data collection, front-end development, and designing the advanced database programs. Akash also focused on front-end development and suggested helpful changes over the course of the project.

DETAILED CHANGES TO MAKE UP LOST POINTS.

- **STAGE 1:** Received 7 out of 8 points. Lost 1 point due to the wrong name of the ReadMe file. The name was changed, as asked by Hongtai, and can be found in the Github repo. As is reflected from stage 2 feedback, the necessary changes were made.

010	Rubric	Evaluation	Score
1	Updates in TeamInfo.md for teamName and the project summary.	Stage.0 feedback is not reflected. The Readme.md is not as required in stage.0.	0
2	Creation of the project proposal placed in the doc folder.	Good.	1
3	The project proposal includes the title and project summary	Good.	1
4	The project proposal includes a detailed description of your application	To see information on EV and plan a road trip.	1
5	The project proposal includes a detailed usefulness description of your application.	Please point out examples of those similar applications.	1
6	The project proposal includes a detailed realness description of your application	Links to potential dataset are provided.	1
7	The project proposal includes a detailed functionality description of your application	What are the complex functions and what are the create components?	1
8	The project proposal includes a detailed low-fidelity UI mockup of your application	Good.	1
9	The project proposal includes a detailed project work distribution across the team	Good.	1
10	Create a release with the correct tag for your submission	Good.	1
11	Total score		9

- **STAGE 2:** Received 12 out of 15 points. Lost 3 points due to the wrong implementation of the ER diagram. The feedback is outlined here. We conducted meetings and discussed all necessary changes with Hongati, and he finally approved our ER diagram, which can be seen in the Stage 3 feedback where he said there were no issues with the stage 2 changes. We updated our database schema, and the differences can be tracked between our stage 2 and stage 3 releases.

010	Rubric	Evaluation
1	Does not have a submission located within the doc folder (-0.5%)	Good.
2	Having a complete and correct ER/UML diagram (+6%)	Maintenance is not Vehicle and should not have Vehicle_id attribute. History should not contain User or Vehicle or Location or Charger attributes. User should not contain Location attribute. Charger should not contain Location attributes.
3	The ER/UML diagram has 5 or more entities (+2%)	User, Vehicle, History, Location, Charger 5 entities. Many-many and many-one relationship types. Maintenance is just attribute of vehicles.
4	Having a complete and correctly translated relational schema (+7%, -1% for each incorrect translation)	User misses attributes. History FK to Vehicle wrong, misses FK. Login primary key wrong.
5	Assumptions of the ER/UML diagram (-0.5% for each missing description for each entity)	Weak entity Login not explained. It is the same as User. Are there any relationship on Charger_type among History, Vehicle and Charger entities?
6	A description of each relationship and its cardinality (-1% for each missing description)	Relationship 2 mismatches What is Within_scope relationship?
7	Create a release with the correct tag for your submission (-2% for incorrect release)	Submitted after due date due to sick extensions.
8	Fix all suggestions for the previous stages (-2% for missing requirements from previous stages)	Fixed stage.0. No update on similar example websites or complex function description.
9	Total Score (15)	Curved Score 12

- **<u>STAGE 3:</u>** Received 20.5 out of 22 points. Lost 1.5 points for not including count queries to show a minimum of 1000 entries in at least 3 tables. Adding a screenshot here for reference and proof that our tables contained 1000 queries each.

010	Rubric	Evaluation	Score
1	Does not have a submission located within the doc folder (-0.5%)	Good.	
2	Database implementation is worth 7% and is graded (as a group) as follows:		
	1. +3% for implementing the database tables locally or on GCP, you should provide a screenshot of the connection (i.e. showing your terminal information)	Unclear locally of on GCP. Implemented 8D out of 8 tables.	
MIN 4	2. +2.5% for providing the DDL commands for your tables. (-0.5% for each mistake)	DDL for 8 tables.	
MIN 3	3. +1.5% for inserting at least 1000 rows in the tables. (You should do a count query to show this, 1% for each table)	No submission for this.	-1.5
3	Advanced Queries are worth 7% and are graded (as a group) as follows:		
	1. +5% for developing two advanced queries (see Implementation point 4 for this stage, 2.5% each)	Query 1 UNION and JOIN Query 2 GROUP BY and subquery	
	2. +2% for providing screenshots with the top 15 rows of the query results (1% each)	Show 2 query results.	
4	Indexing Analysis is worth 8% and is graded (as a group) as follows:		
	1. +3% on trying at least three different indexing designs (excluding the default index) for each advanced query.	Query 1: No indexing and indexing on Vehicle(Brand), Compatible_with(Chrager_Type), and Both. Query 2: No indexing, indexing on Brand, Price, Model.	
	2. +4% on the indexing analysis reports.	Have reports.	
	3. +1% on the accuracy and thoroughness of the analyses.	-1 Analysis does not consider time cost fluctuation. A small amount of time increase does not mean the plan created on the index is slower.	
5	Total Score (22)		20.5

```
mysql> show tables;
+------------------------------+
| Tables_in_electric_vehicles |
+------------------------------+
| About                        |
| Charger_Type                 |
| Charging_Point               |
| Compatible_With              |
| FinalTable                   |
| History                      |
| Location                     |
| User                         |
| Vehicle                      |
+------------------------------+
9 rows in set (0.20 sec)

mysql> SELECT COUNT(*) FROM Vehicle;
+----------+
| COUNT(*) |
+----------+
|     1202 |
+----------+
1 row in set (0.20 sec)

mysql> SELECT COUNT(*) FROM Location;
+----------+
| COUNT(*) |
+----------+
|     5000 |
+----------+
1 row in set (0.22 sec)

mysql> SELECT COUNT(*) FROM Compatible_With;
+----------+
| COUNT(*) |
+----------+
|     1200 |
+----------+
1 row in set (0.20 sec)

mysql>
```

- **STAGE 5:** Received 20 out of 23 points. Lost 2 points for not using advanced queries in Stored Procedure and one point for Trigger design. The new modified SQL script for the stored procedure has been pushed to the Github repo.

  We have also reimplemented our trigger to apply an extra 20% maintenance cost on any company's new vehicle if the company's average cost increases beyond 80000. This again is used to model a hypothetical scheme. We have pushed the modified trigger to the Github repo.