

## **Project Reflection Report**

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

No changes were made to the initial idea. However, some parts of the initial proposal were reduced in scope due to a shortage of time.

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

The application has achieved its purpose of creating a platform that solves the problem of food wastage by distributing food among users who need it. It also enables restaurants and hotels to monitor their waste and make informed decisions about the supply-demand ratios. Unfortunately, due to the shortage of time, we could not implement the notification feature where the customer is notified about the order status.

**Discuss if you changed the schema or source of the data for your application**

1. In the Offers Table, we added a new attribute called “CurrQuantity” that keeps a record of the current quantity of the meals offered. We maintained the amount of food in each type offered by each provider in the Offers Table.
2. Each provider is supposed to update the Quantity of food offered every day.
3. However, to calculate wastage analytics, since we had to know the initial amount of meals offered, we had to add a column; if we had updated the Quantity column, we would have lost information about the number of meals provided at the start of the day.

**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 did not change the ER diagram.

**Discuss what functionalities you added or removed. Why? We added the functionality to search by food item name.**

1. Earlier, we decided to search only for provided limited umbrella options (like Veg, Non-veg, etc.)
2. However, we realized this is inadequate as a given user might be allergic to certain food items in a particular category and needs more refined results.

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

The main contribution of our application is to rate hotels based not only on user feedback but also on the amount of waste they generate. This enables us to find more sustainable and environmentally conscious restaurants. This also allows hotels to track their waste and make decisions to maximize efficiency based on historical trends.

**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.**

1. **Rugved:** We used react js for frontend and flask for backend i.e 2 different domains. In such a case, we faced Cross-Origin Resource Sharing (CORS) issue where even though backend APIs were working fine independently when testing using Postman software but when we were trying to make the same request from frontend, it was throwing the “Access to prq api has been blocked by CORS policy”. By default, browsers will only allow communication between client and server as long as they are in the same domain. To resolve this, we need to put appropriate headers and this can be done by using the “cors” library in flask.
2. **Mayank :** When using flask and sending insert queries through the backend, our changes were not showing in the database. We had

experience with using MYSQL. Later we realized, our transactions are not committed by default. Thus, we had to commit after each insert/delete explicitly as the connection was set up with autocommit off.

3. **Hardik:** Since we had created our database on Windows OS and then we shifted to a Linux based OS on GCP so that all the group members have easy access to the SQL database, there were several errors in the SQL code. After debugging we learned that MySQL is case-insensitive on Windows OS and case-sensitive in most of the Linux systems. So for this, we had to change various parts of our code to run the previous SQL code on GCP.
4. **Saurabh:** It was difficult for us to integrate a Python Based Backend (in Flask) with a JavaScript based Frontend (React.js). Moreover, to test the flask APIs, before we created a GCP instance of our database, each of the team members had to have a local instance of the database. Making the database accessible was difficult until we created a GCP instance of it. In addition, populating the data in the tables was a challenge too. We had to write stored procedures to populate a ton of data records in different tables. Connecting to the database via the backend was challenging too.

### **Are there other things that changed comparing the final application with the original proposal?**

1. Due to limited time, we had to reduce the scope of the project.
2. Initially we had proposed adding NGOs and displaying the providers on a map. As of now, this has been left for future work due to lack of data and time.

### **Describe future work that you think, other than the interface, that the application can improve on**

1. Implementing further checks on the application logic, preventing users from entering invalid data.
2. Authentication and Privacy
3. Adding map-based queries where users can query databases based on map inputs.

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

1. We had 2 main components - backend and frontend.
2. All 4 members contributed to both components.
3. Frontend was majorly done by Rugved and Backend by Saurabh, Hardik, and Mayank. We further broke down each part to modules and each module was assigned to 2 team members. This technique distributed the load among the entire team equally and this was managed by the team really well.
4. For advance database program:
  1. Stored Procedure was developed by Saurabh and Mayank
  2. Trigger was developed by Hardik
  3. The frontend and creative visualisation for stored procedure was developed by Rugved.