

# DineDash

## Project Summary

Our project is to create a website, DineDash, to find a restaurant that matches your specific needs such as specific dietary restrictions, food preferences, pricing, etc. Deciding on a restaurant that meets the desired combinations can be a very difficult task so we wanted to create DineDash to solve this problem.

## Description

DineDash is designed to help users find optimal dining recommendations based on their needs and wants. The user logs into the application and inputs data based on their preferences and dietary restrictions through an intuitive user interface. The app then collects this user data, processes it, and queries a database of restaurants to find the best possible matches. It filters and ranks the restaurants based on the users' criteria. The user receives a list of recommended restaurants with relevant information such as name, address, rating, pricing, and menu items on the user interface.

## Usefulness

Our proposed application is similar to yelp and uber however, we plan to add feature(s) not available in either application. Some users may have dietary restrictions so they have the ability to filter restaurants by ingredients to make sure the restaurant of their choice is safe. Similar to Uber Eats, restaurants can be chosen by category, price, distance and our search and filter functions will apply the same features with added specifications.

Additionally, we will add the ability for users to create a cost estimation for the items they wish to order from a selected restaurant.

Similar to going on yelp to find a place to eat based on preference

- Allergy filtration sets it apart
- Creative Component/ Challenge part:
  - When picking a restaurant they can add what they wish to order to a 'cart' and we estimate the cost based on what they want to order

## Realness

**Source:** <https://www.kaggle.com/datasets/ahmedshahriarsakib/uber-eats-usa-restaurants-menus>

This dataset contains 2 csv files:

1. Restaurants - id, name, score, ratings, category, price\_range, address
2. Restaurant Menu - restaurant\_id, category, menu\_item name, description, price

We will extract data from these tables and place them in separate tables like score and ratings, addresses and more.

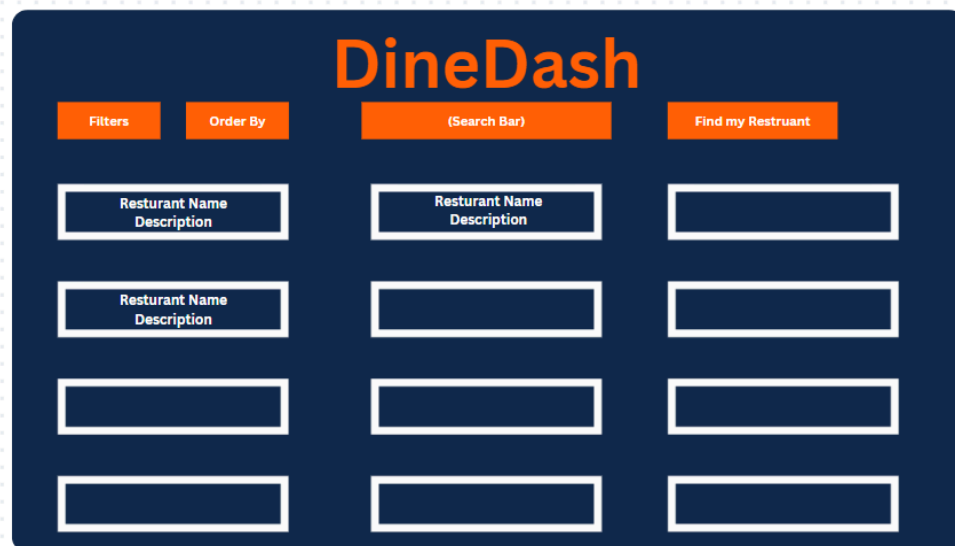
Dataset size from the original file is 1048576 rows.

## Functionality Description

Users on the application can search for certain restaurants and receive a list of the restaurants fitting their criteria. They will have the ability to filter out restaurants based on price, distance, allergens, cuisine type, ratings, etc. We can allow multiple filters to be added and deleted to the search and the button “Find my Restaurant” can allow the user to update restaurants if they change their filter preferences. We can utilize React Select feature to allow multiple variations to be selected by the user allowing preferences to be unique. Additionally, we can implement an “Order By” drop down that will order the restaurants chosen based on location, price rating or relevance/none. The search bar can be used to directly find a restaurant that matches the keyword they typed.

Users will have the ability to select a restaurant they choose and be shown the multiple items, price and description of items in the restaurant. Users will be able to add the item of their choice to a list which will output a total price of their order if they decide to purchase the item(s).

## UI Mockup



The mockup shows a dark blue header with the title "DineDash" in orange. Below the header is a navigation bar with four orange buttons: "Filters", "Order By", "(Search Bar)", and "Find my Restruant". The main content area is a 4x3 grid of white-outlined boxes. The first two columns are labeled "Resturant Name" and "Description" respectively, while the third column is empty.



The mockup shows a dark blue header with the title "(Restarant Name)" in orange. Below the header is a navigation bar with three orange buttons: "Filters", "Find my Restruant", and an empty space. The main content area is a 3x3 grid of orange boxes. Each box contains the text "Item Name", "Price", and "Description".

## **Work Distribution**

The work will be distributed primarily based on each member's experience and their availability to take on a larger workload. We plan on utilizing branches to work on our separate parts for testing and organization. Currently our breakdown is based on experience and will further be split more narrowly when we begin designing the project further.

### **Work Distribution**

Ashley: Fronted UI Design, Primarily focusing on creating the search bar functionality, Buttons, general layout of text, buttons, etc within the page itself. Will work on routing the website front end with the database using Flask.

Deepika: Integrating frontend and backend by working on the Flask for data handling. Working with Lindsey and Ashley to connect the two aspects they are focusing on primarily. Working on fetching data from the back-end to show dynamic content on the page when the user selects filters. Additionally working on applying filters applied by the user is handled as a query.

Manas: Working with Lindsey on query creation and backend. Will be working on database creation for SQL.

Lindsey: Designing the entities/relationships for the project and splitting up the data appropriately to meet requirements of the project. Will work on query creation for filtering capability.