LetsEat

LetsEat: Alpha Prototype

Group Members: Sarah Stevens, Zach Bloomstein, Max Englander, Don Kim

2022.12.07

01.
Summary

The Issue

"I don't care where we go, I just don't want Italian"

"We can't figure out where to eat, it is too hard to decide"

"It is so long and difficult for everyone to agree on a restaurant"

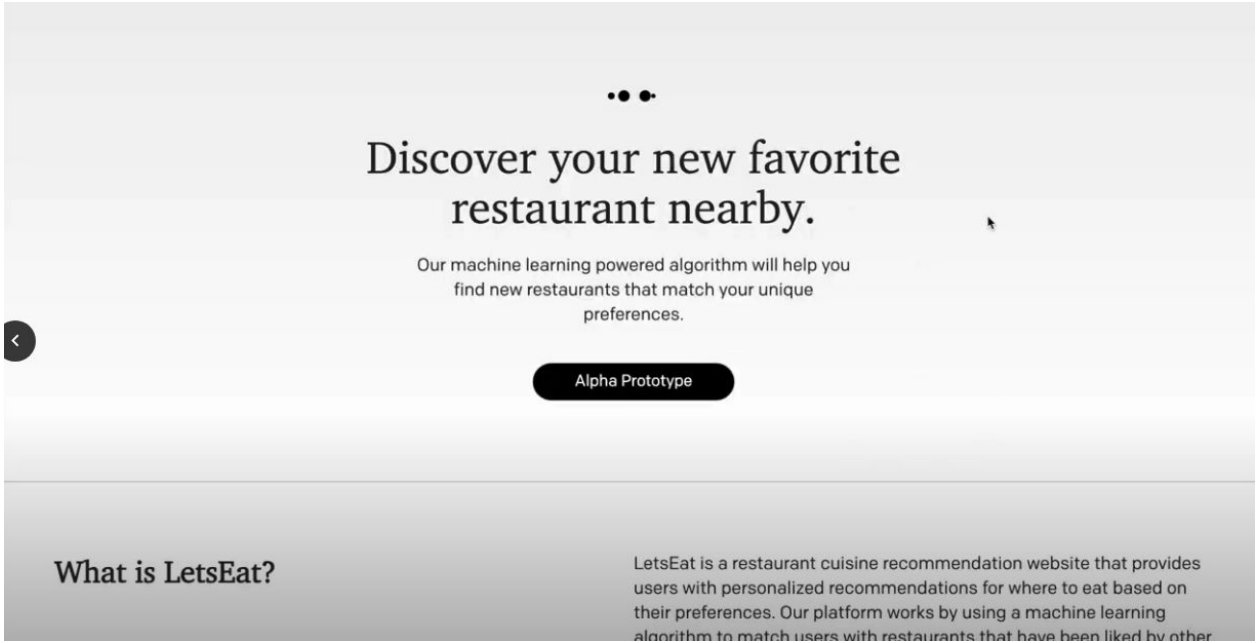
The Solution

We make the decision for you!

LetsEat is a restaurant recommendation application that utilizes machine learning to quickly generate a **single personalized recommendation** based on the user's input.

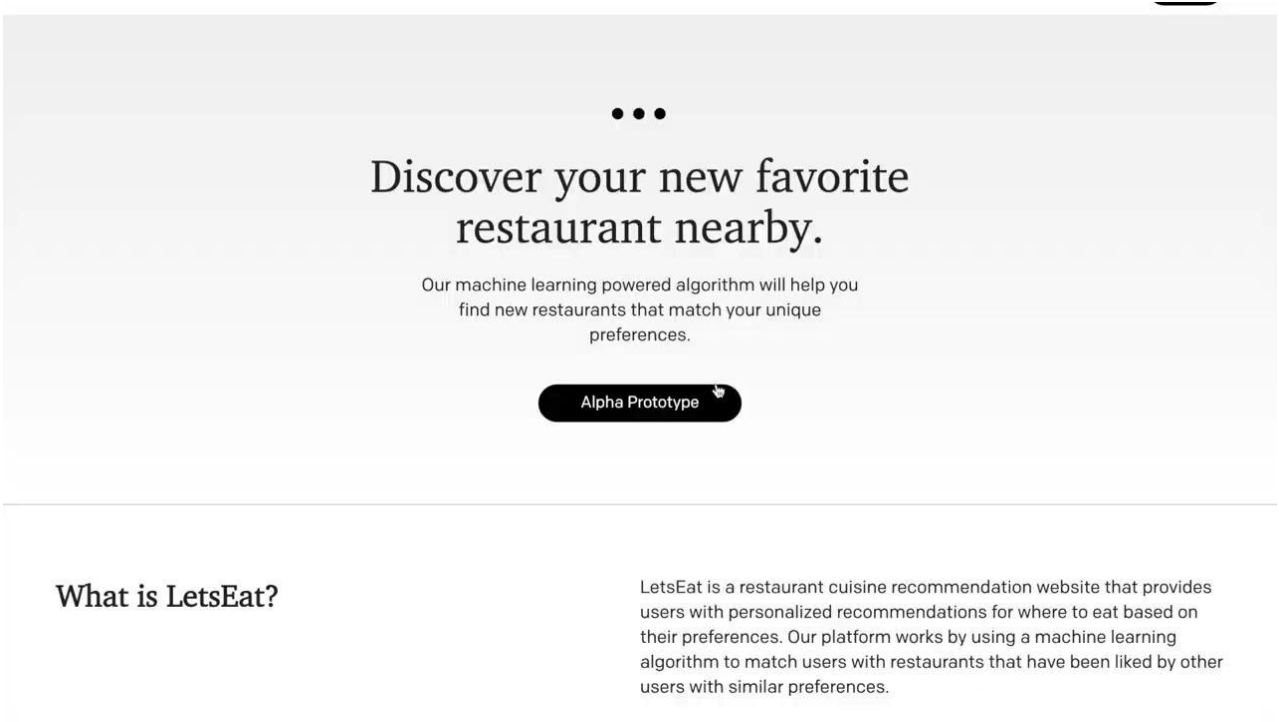
02.

Alpha Prototype

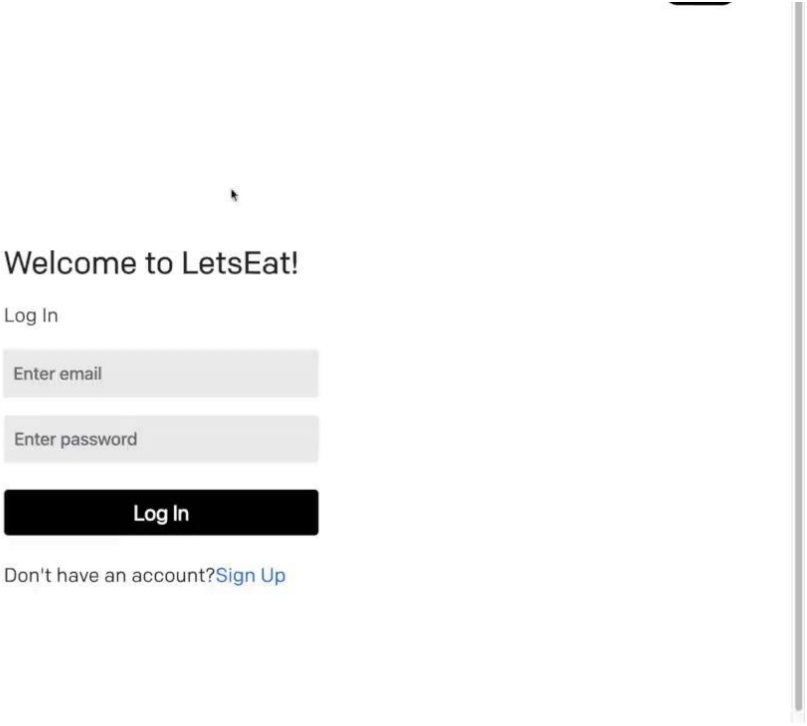


Integrated, Working Application

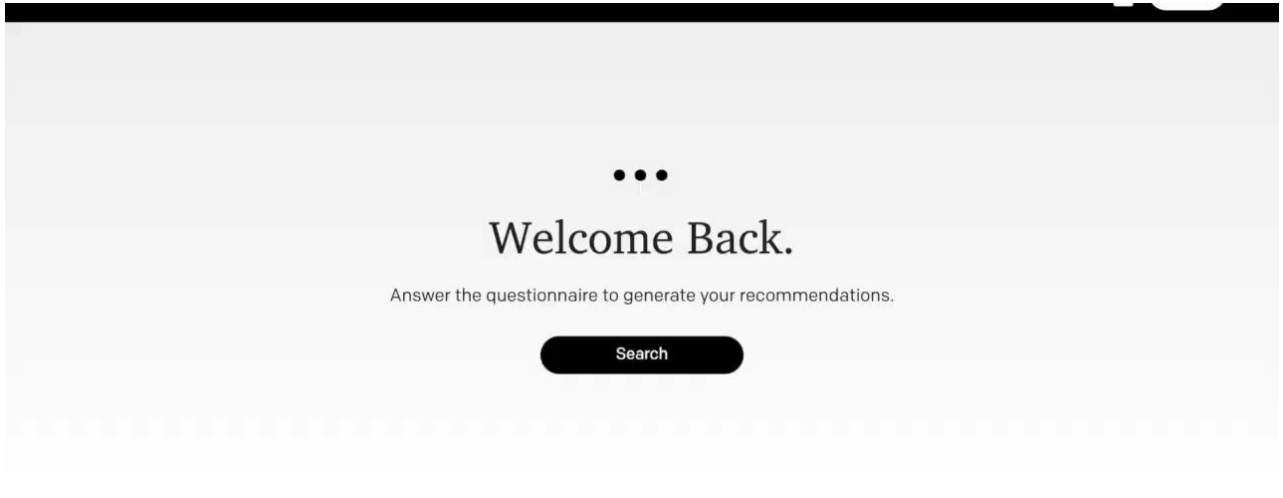
- Able to create a profile
- Fill out the questionnaires
- Receive restaurant recommendation
- Model is integrated with application
- Low confidence suggestion is given



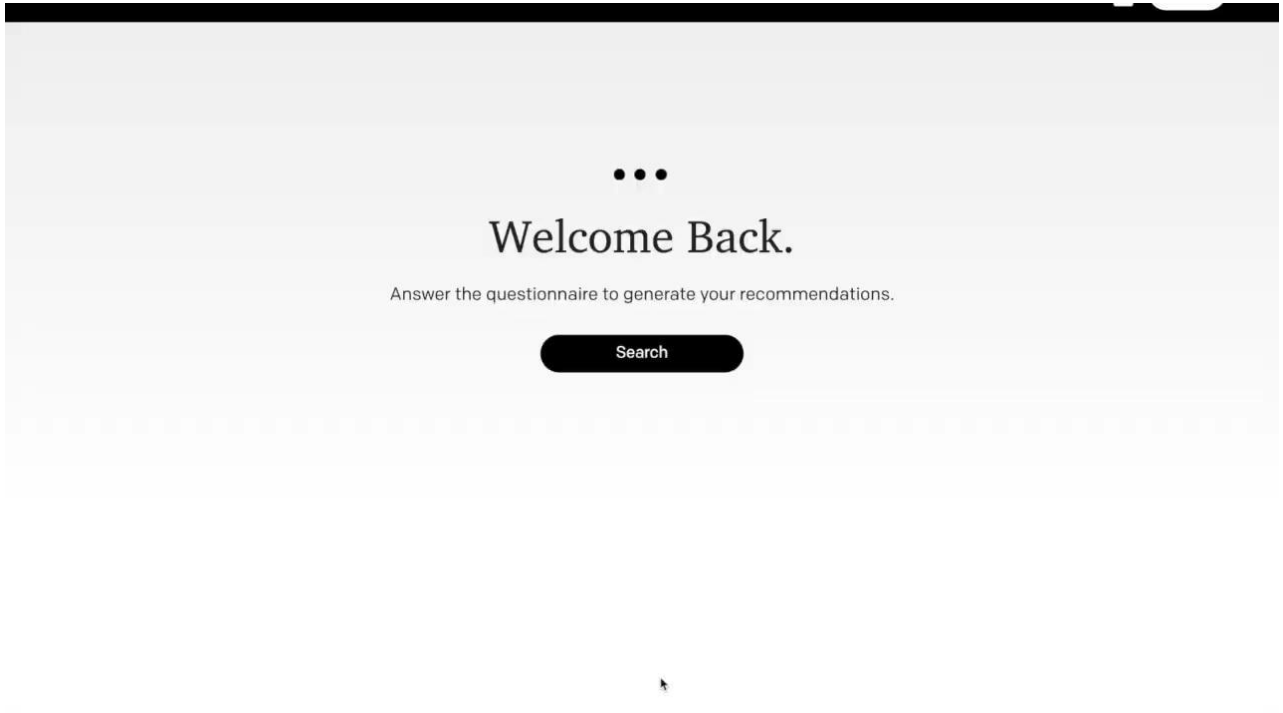
Creating Account


Welcome to LetsEat!
Log In
Enter email
Enter password
Log In
Don't have an account?Sign Up

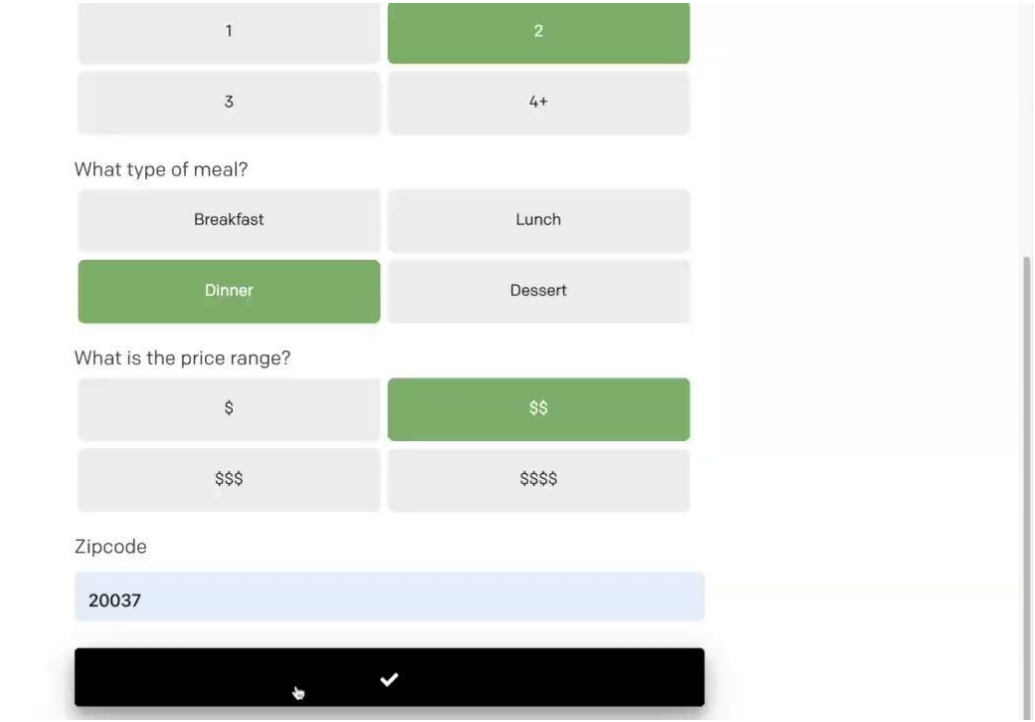
Log In



Profile Questions



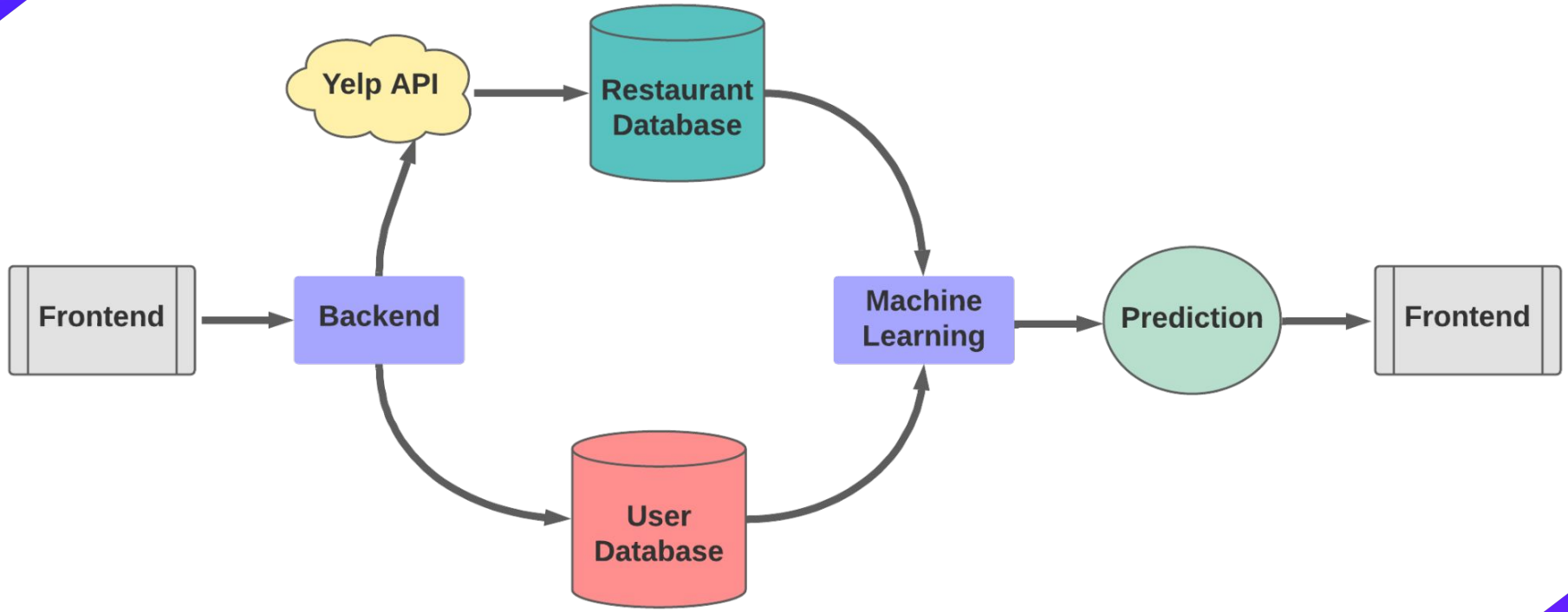
Search Questions



Restaurant Recommendation

03.

System Architecture


Yelp API
Restaurant Database
Frontend
Backend
Machine Learning
Prediction
Frontend
User Database

3.1
Frontend: Web App

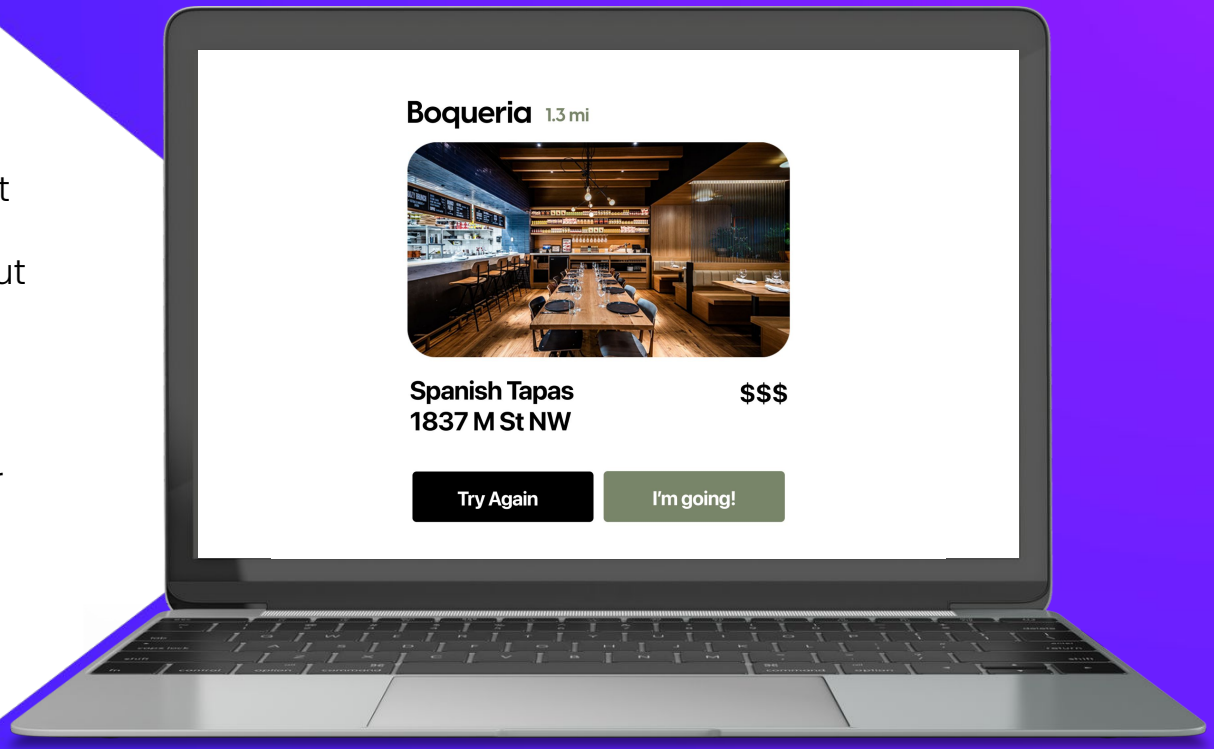
Technologies

REACT, Bootstrap

FRONT END

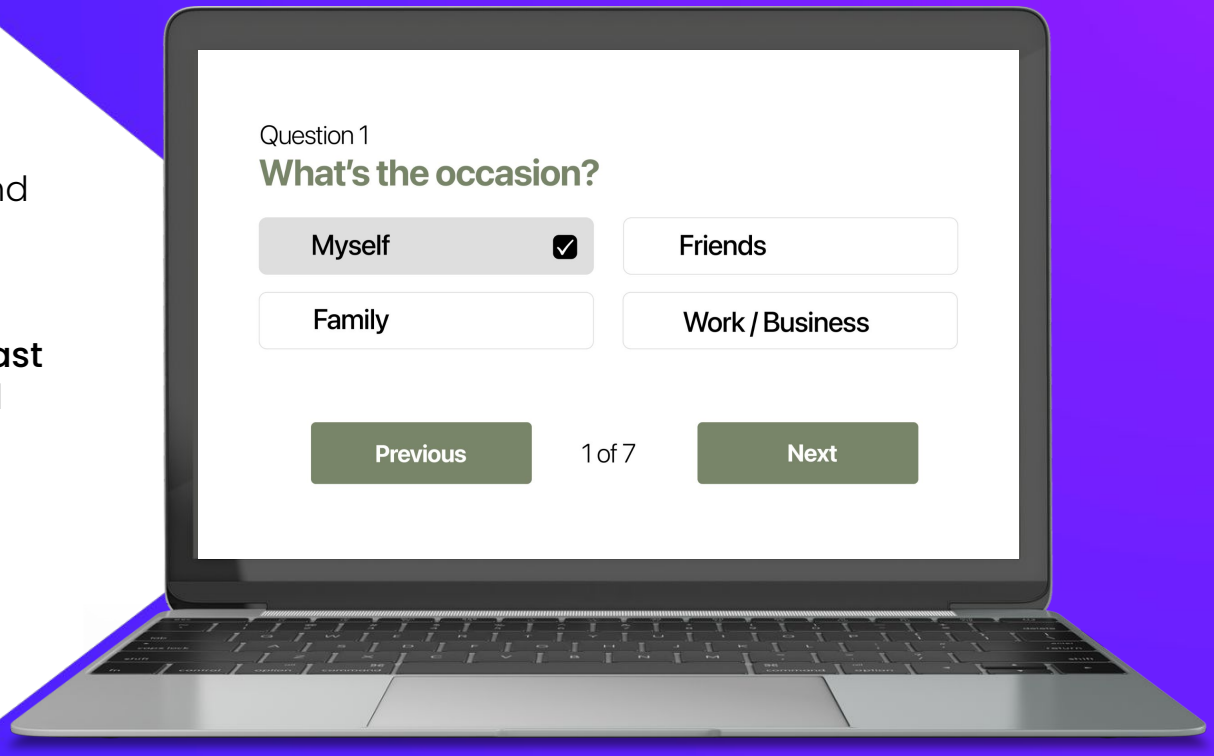
The **LetsEat** platform is built on the **REACT** library with **BootStrap** for styling. We put importance on having a modern and minimal user interface.

The user experience on our website is designed to be **seamless** so that the user can **quickly and easily** receive a restaurant recommendation.



FRONT END

The REACT frontend will send and receive data with the backend, such as the questionnaire and user answer choices, through **Fast API** which will be discussed further in the backend portion.

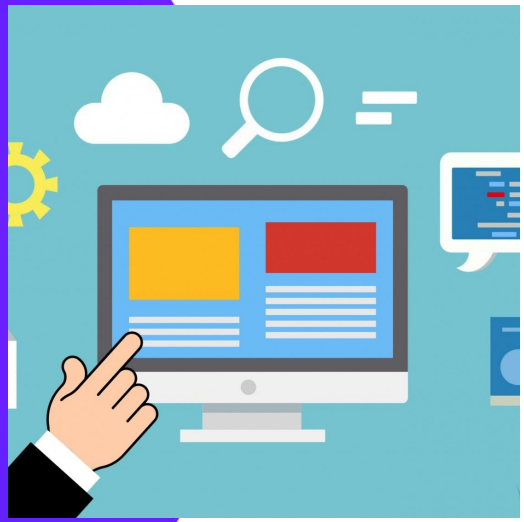


3.2

Backend: Web Server

Technologies

FastAPI, MySQL, OAuth



What is the Web Server?

Will host all the Backend Logic

Connects the front end application to all the functional aspects of the website, such as logging in, updating profile preferences, and picking a restaurant.

Storage of User Information

Users will login through the server and have their profile information saved.

How is the Web Server built?

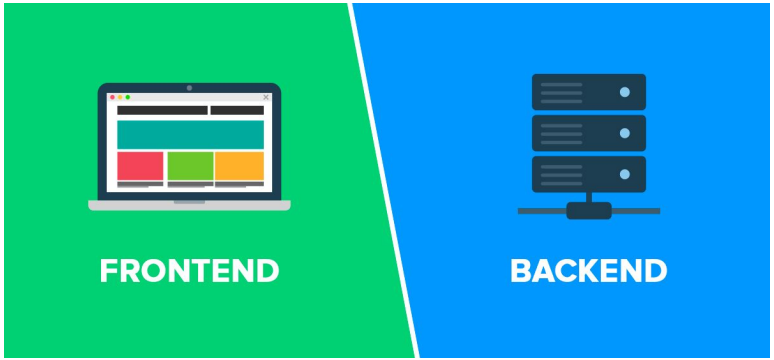
FastAPI

- Hosts API endpoints
- Connected to REACT app
- Receives and sends info

MySQL

- Stores user information
- Logins, cuisine preferences, dietary restrictions

Future releases?



Handling Edge Cases

Making sure that form validation is working and that data being sent to the server is properly formatted and won't cause unexpected exceptions when sent to the DB/ML model.

OAuth

- Allows users to create accounts easily
- Can connect directly to Google, Facebook

3.3

Backend: Data Collection

Technologies

Yelp API, Selenium WebDriver, Vader Sentiment Analysis, MySQL

Collecting Data

Yelp API

Get list of restaurants that meet general specifications of user (like location, allergies)

Get details of a specific restaurant

Selenium WebDriver

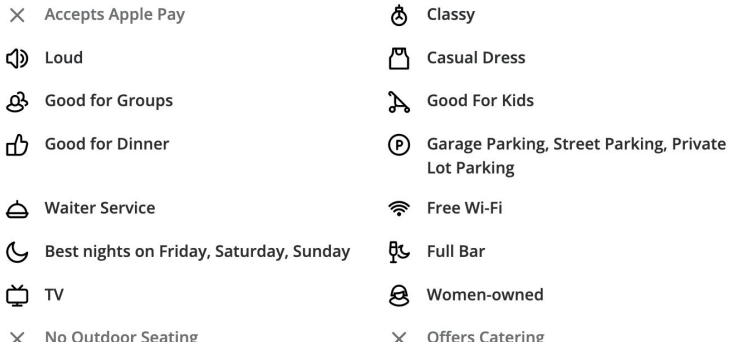
Web scrape Yelp Restaurant Website to get atmospheric attributes

Web scrape User Reviews to determine experience from content/keywords

Vader Sentiment Analysis & MySql

Vader: Use Sentiment Analysis to determine value based on review

MySql: Store all scraped data



Scraped Restaurant Data

Good for Dinner, Good for Groups, Good for Kids, Casual Dress

Scraped User Data

Henry had a positive experience at the restaurant for date night

3.4

Backend: Machine Learning

Technologies

Python, Scikit Learn, PyTorch, Numpy, XGBoost

Machine Learning

- Model takes in **answers of users** and **features of restaurants**
- Matches the users and restaurants
- Delivers **recommendation** based on **model training**

How it Works

- Find restaurants for a user based on what similar users liked
- **Similarity-based**

Collaborative Filtering

Future Releases

- Having the model retrain itself after time frame/uses
- Making sure model weights other properties correctly
- Tuning content filtering

Thanks !

Questions?