

LetsEat Technical Summary

Technical Innovation

The main technologies that will be utilized in this project are FastAPI, REACT, MySQL, and Scikit Learn. FastAPI will be used to host the web server, REACT for the front end web application, MySQL for the storage of user information, and Scikit Learn for the machine learning model. The novelty and uniqueness of this restaurant recommender is that it provides a single, personalized recommendation instead of a lengthy list based on basic filters. Yelp/OpenTable don't suggest restaurants based on atmospheric characteristics and don't provide single recommendations based on machine learning, the decision is still left up to the user. LetsEat will take out the confusing decision-making process and suggest the best possible restaurant given the user's preferences and the occasion due to its machine learning component.

Key objectives

The key objectives for this project would be to deliver a fully functioning full-stack application with a working frontend web application along with a backend with all the key components including a database for storage of user information, a machine learning model trained and able to give predictions on restaurant data, and security set up to prevent any potential attacks exposing the website and/or user information.

Questions that must be answered include:

- How will we make sure the website is properly secured so user data isn't exposed?
- Will the machine learning model be able to give accurate and quick suggestions?
- How do we plan on improving the model as we gather more data?
- How can we send and receive information quickly and accurately between the frontend and backend components?
- How will the user-base be built and how will retention of users be accomplished?

Technical Feasibility

Using FastAPI to build the web server is very do-able as there is lots of support and tutorials surrounding FastAPI on the internet, including how to integrate it with a REACT application. For the machine learning model, we can use XGBoost and Scikit Learn to train and use the model, which also has plenty of support and documentation online.

Costs, Risk and Risk Mitigation

The development cost in terms of hardware is none, as our project and the backend component specifically are entirely software-based. For the software side, the expectation is that a working machine learning model will be ready to deliver suggestions with relatively-low confidence scores by the end of December. After that, it may take an additional 1-2 months to fully refine the machine learning and web server code so that it is optimized and running as quickly as possible. In terms of lines of code, the estimation for the entire project would likely be in the 700-800 range due to the amount of code needed for the frontend, while the web server and machine learning side would be closer to 300-400 because that code is more concise.