

## LetsEat Technical Summary

Sarah Stevens

LetsEat is a restaurant recommendation system that uses user inputs and machine learning to give a personalized suggestion. The system will use many technologies, including REACT, Scikit Learn, Yelp API, Selenium WebScraper, FastAPI, and MySQL. The front end of the application will utilize REACT. Scikit Learn will be used for the machine learning library. Data about the restaurants will be retrieved from the Yelp API. Other restaurant data and machine learning training data will be acquired through Selenium web scraping. The server will be hosted with FastAPI. MySQL will store user data. LetsEat is novel because it gives one, personalized recommendation. This differs from applications like Yelp/OpenTable, which give lists of restaurants for the user to decide between. In addition, current applications recommend restaurants based on basic user input, while LetsEat will utilize more specific, personalized data to give a recommendation also based on atmosphere. LetsEat simplifies the decision-making process by giving a single, personalized recommendation for that experience.

LetsEat's objective is to have an integrated, functional, full-stack application. This includes a working machine learning algorithm that uses user and restaurant data to give accurate recommendations, based on relevant training data. For this, we must answer how to acquire enough training data for the model. We want our model to improve accuracy with time, so we must answer how we will increase accuracy as we get more usage. Within the application, there will be a completed front end communicating with back end. Successful back end includes user data storage and restaurant data collection. For this to be successful, we must answer how we will collect and communicate this data securely and quickly. Lastly, because we are relying on user information and inputs, we must answer how we will ensure our system is secure and safe.

There are existing tools that will assist us in being successful. Creating a full-stack application can be accomplished using online documentation. It is important we can acquire enough data to give accurate recommendations. Such data is also online, and, for collection, there exists web scraping technology. For the machine learning model, Scikit Learn has the necessary libraries. Overall, this project is do-able because we have access to relevant documentation for each aspect of the application and have resources to collect relevant data.

There are few development costs for LetsEat. Software costs include data storage and application hosting, with no hardware costs. The major project milestones include fully integrating the front and back end, which will be completed by the December Alpha-release. The machine learning algorithm will also be integrated and provide low-accuracy recommendations, including full user/restaurant data collection on my end. In the following two months, we will optimize the application in terms of ease-of-usage, speed, and improved recommendation accuracy. Following this, additional features will be added. Overall, this project will be around 1250 lines of code. Front end will be around 750 because all interface features are explicitly stated. Back end will be around 500 because frameworks exist to accomplish many of the tasks.