Project Milestone 2

Team Members - Group 3			
Eric Phann, *product manager*	**Yaxin** Zhao, *data/prompt engineer*	**Lakshmi** Jayanth Kumar, *app developer*	**Gaurav** Samdani, *LLM operations*

[Click Here for Group 3 Repository]

Summary (#8)

We have solidified our project use case as Charlotte Eatz, a foodie companion app for the local Charlotte market. The main addition to our codebase was retrieval augmented generation (RAG), which was tailored to our product using synthetic data generated from Anthropic AI. Next, we added additional Pytests specific to our LLM agent and tools. Finally, we also stepped up our GitHub workflow and best practices by incorporating issues and pull requests as well as a more meaningful repo title and description. These best practices will continue throughout the project.

Milestone 2 requirements were decomposed into issues and the RAG and Pytest components were covered in pull request #10 and #11 respectively **[1]**.

Challenges

One of the primary challenges we had was integrating all the components together using our current framework. For example, while the *implementation* of RAG was not too difficult, the *integration* of the component into our codebase was a challenge, and we will need to revisit it later on. This is a similar challenge for Pytests and metrics. We will continue iterating and stay open to researching and implementing alternatives to what is the most compatible with our code.

A secondary challenge was crunch time. The bulk of the work was completed closer towards the Milestone 2 deadline. We hope to implement smaller goals and more intentional deadlines to help move the project along at a more consistent and sustainable pace and ensure we are always on the same page, even if we all have different levels of familiarity and expertise.

Reflections

Overall, we believe this was an all-around strong improvement from Milestone 1; all aspects of the project, both technical and non-technical, have improved. Additionally, we were able to get the RAG working. While standalone and not fully integrated to our app, it was still tailored to our product using synthetic, local Charlotte data.

Next steps

We hope to integrate everything and refine our product into a MVP before assessing overall app performance and carrying out our user testing plan below. By the next milestone, we hope to have a more refined application with a thoughtful user experience from beginning to end, with all key features and user feedback implemented *and* integrated.

Another step is to complete a broader codebase review to ensure all team members understand each component and its significance for the project. At the same time, we will clean up code, the project repository, and add any docs as needed using tools such as Black, Bandit, or isort.

Through an enhanced GitHub workflow and targeted action items and deadlines, we're confident we can sustainably achieve this and create an amazing product by the final submission.

Project Use Case Definition (#6)

Product Overview

Charlotte Eatz is a large language model (LLM) foodie companion app powered by retrieval-augmented generation (RAG) and agentic automation. The app enables users to discover and book reservations at local restaurants in Charlotte by combining real-time data from the Yelp API with pre-processed restaurant reviews from the Kaggle Yelp dataset. It also provides personalized restaurant suggestions and automates the reservation process.

Focused on Charlotte neighborhoods and specific zip codes, the system ensures users receive up-to-date, relevant options and can seamlessly reserve a table.

Target User(s)

- **Visitors and Tourists:** People visiting Charlotte who seek an easy-to-use tool for finding local gems and making reservations without navigating multiple apps or websites.
- **Busy Professionals:** Individuals with tight schedules who want quick and reliable restaurant recommendations for business lunches, dinners, or social gatherings. They appreciate the automation of the reservation process, saving them time.
- **Locals**: Those from Charlotte that want to discover trendy restaurants and experiences.

Problem Statement

Charlotte lacks a foodie companion app for helping customers satisfy their culinary cravings through local restaurants and experiences. Charlotte Eatz hopes to bridge this gap and bring local foodies one step closer to their new favorite spot.

Key Features

- Search bar and filters for location and cuisine

- Personalized recommendations based on chat history
- Name entity recognition (NER) for better responses e.g., GLiNER
- Automated reservation system for convenience
- By and for the local Charlotte market

Scope Definition

All of the key features mentioned we believe are in-scope. We believe we can successfully implement them by the time we start testing our product and continue to improve upon them. Some out-of-scope features, or features that could come later on, are cross-city expansions, social media features, and being able to provide users with exclusive deals and discounts.

Unique Value Proposition

Our product stands out by offering hyper-local restaurant recommendations tailored specifically to Charlotte, paired with automated reservations for a seamless user experience. The combination of localized focus, automation, and tailored data handling makes our app unique in simplifying and innovating restaurant discovery and booking for the local Charlotte market.

Success Metrics

- **User Satisfaction**: Gathering feedback through user testing to assess user satisfaction with restaurant suggestions, booking automation, and the overall experience.
- **Local Partner Feedback**: Gathering insights from local businesses like the 1st Floor Coffee Shop to evaluate the app's effectiveness in supporting business needs.
- **Reservation Completion Rate**: Measuring the percentage of users who successfully complete automated reservations after receiving restaurant recommendations.

Potential Challenges

One challenge that we anticipate is real data scarcity due to API limitations. It may be a financial strain to rely on APIs from Yelp or Google for local Charlotte restaurant data over the long-term. Solutions may be to web scrape or, for the time being, rely on synthetically produced data.

Another technical challenge is successfully bringing all of the features together in the app (e.g., filters, NER, etc.). While they work standalone, it's important to seamlessly integrate them together in one app to maintain a streamlined user experience.

Ethical Considerations

Our app could disrupt existing restaurant booking systems, leading to potential negative impacts on smaller establishments that may struggle to compete with larger chains or platforms. We will actively engage with local restaurants to understand their needs and concerns, ensuring our platform adds value to their business. For example, offering features like promotions or visibility boosts for smaller restaurants can foster a positive relationship and support local economies.

User Testing Plan (#7)

- **Target user group**: Local Charlotte users who want an optimized and streamlined restaurant discovery experience. No technical experience necessary.
- **Recruitment strategy**: Reach out to our immediate social circle i.e., classmates, family, friends, and local food businesses (such as the 1st floor coffee shop).
- **Testing methodology**: The methodology will be simple—ask a user to explore the app, focus on identifying a desired restaurant and then booking it through the chatbot.
- **Feedback collection**: In-person interviews
- **Metrics**: Simple binary questions with additional comments requested
    - Was the application easy to use? Why or why not?
    - Was the application better than just researching and booking yourself?
    - Did you like interacting with the chatbot and its responses?
    - Would you use our app again?
- **Timeline**
    - Finish developing MVP, integrating all features, and metrics by November 10th
    - Take the following week to conduct in-person demos and tests (Nov. 11th-15th)
    - Tweak and refine based on feedback and requirements for Milestone 3 (Nov. 21st)

Appendix

[1] Screenshot of Milestone 2 on GitHub