Tastely Progress Report 2

Team Members: Yug Patel, Kush Solanki, Arjun Kirubakaran, Abraham Yirga

This week, we continued development based on the foundation established in the previous phase. We successfully completed backend user authentication with Flask, implementing secure login, registration, and session management. The user profile API was finalized, enabling users to view and update profile settings such as dietary preferences. On the frontend, we developed and styled login and signup forms using JavaScript and Tailwind CSS, and successfully integrated these with the Flask backend through API calls. We created the initial user dashboard layout, displaying saved recipes and profile information. In addition, we finalized the initial UML class diagram to better document system architecture and began setting up integration with the Spoonacular API to preload the database with a variety of recipes for improved user experience.

Several aspects of the project progressed well this week. Backend user authentication was completed and tested with minimal issues. Integration between frontend and backend components worked smoothly, with clear communication between JavaScript fetch requests and Flask API responses. Our database schema proved to be robust, requiring no structural changes as new features were implemented. The team also collaborated effectively by distributing tasks across frontend, backend, and documentation workstreams, allowing parallel progress on multiple parts of the system.

There are still areas where improvements are needed. Frontend form validation and user error feedback (such as displaying invalid login or signup errors) require further refinement to ensure a more intuitive user experience. Integration with the Spoonacular API encountered unexpected challenges related to authentication and data formatting, requiring additional time to resolve. Backend unit testing needs to be expanded to cover more service and edge cases to ensure reliability as we add complexity. We also observed that allocating more time earlier in the sprint to frontend UI/UX improvements could reduce friction during final integration stages.

Overall, we are steadily advancing toward building a fully functional MVP, and we have clear action items for the next sprint to address these areas and continue feature development.