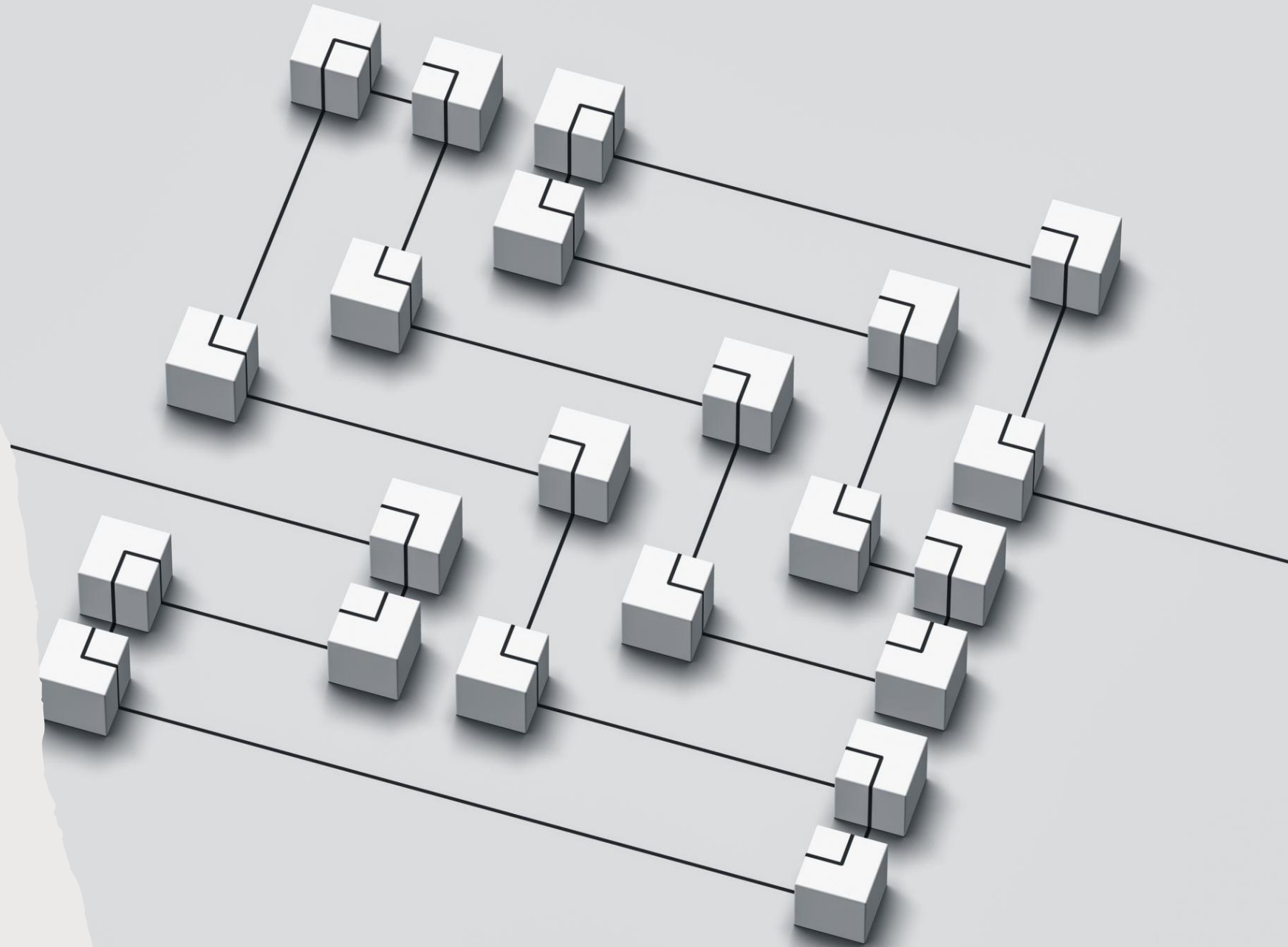


# Milestone 3 : Project Implementation

By Ryan Davis, Cory Vitanza, Ivan  
Farfan, and Johan Jaramillo



# Roles and Tasks

## Functional Requirements for Milestone 3:

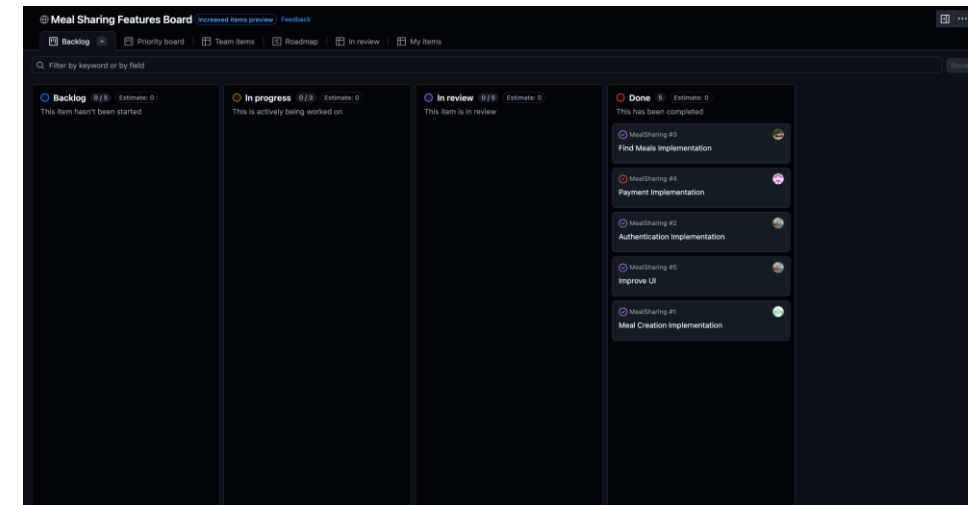
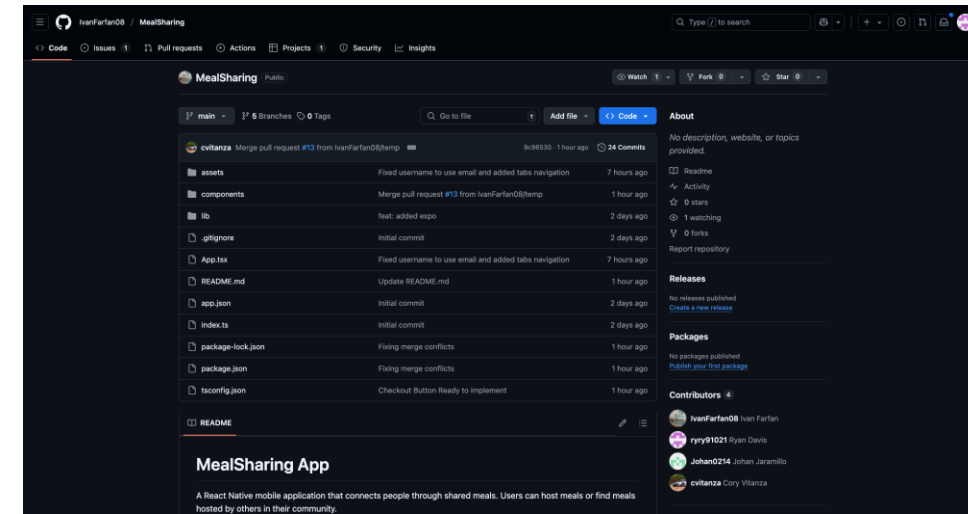
- A user can log in to the system
- A user can register an account for the first time.
- A host can create a meal, with the required content and information (photo, ingredients, etc.)
- A guest can browse previously posted meals and select a meal to view details.

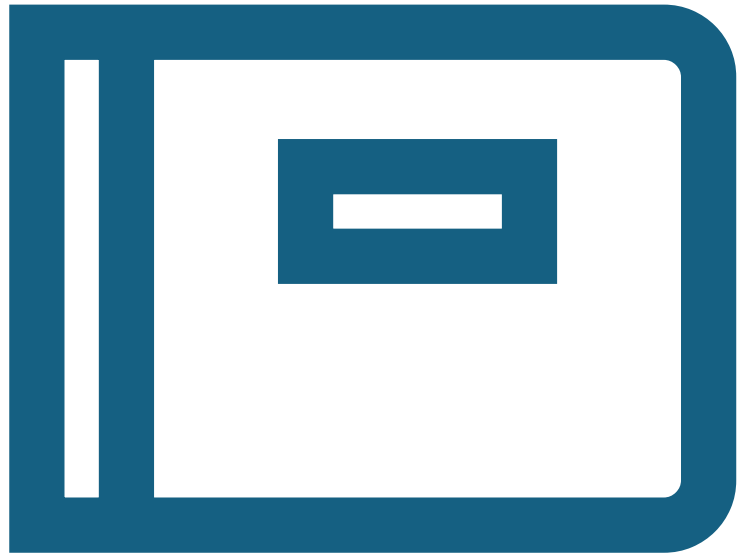
## Team Roles:

- Ryan Davis- Stripe API implementation (CheckoutButton.tsx, App.tsx), documentation.
- Ivan Farfan- Manages React project. Implemented user account logic (Account.tsx, Auth.tsx, App.tsx), Manages GitHub and Pull Requests.
- Johan Jaramillo- Implemented Meal creation for user type “host” to provide meal cards (HostMeals.tsx).
- Cory Vitanza- Updated the "Find Meals" page layout and functionality (FindMeals.tsx). Implemented bugfixes for misc. errors/bugs.

# Project Repository

- **GitHub:** Utilized for version control and collaboration through branches and pull requests.
- **Branching:** Allowed each developer to work on different components of the application independently.
- **Pull Requests:** Used for code review before merging completed tasks into the main branch for deployment.
- **Features Board:** An agile board was used to manage the assignment and completion of tasks..
- <https://github.com/IvanFarfan08/MealSharing/tree/main>





Our Demo

# Challenges and Strengths

## Challenges

- **Different Development Environments:** The team faced difficulties with setting configurations that work universally across different machines.
- **Merge Conflicts:** Multiple merge conflicts arose due to conflicting files and differences.
- **Configuration Changes:** Developing on separate branches led to changes in multiple files, which contributed to changes in certain areas of the code.

## Strengths

- **Effective Use of React:** The team successfully developed the app using React, with all developers completing their tasks.
- **Pull Requests for Code Review:** Tasks were reviewed through pull requests, ensuring code quality and usability.
- **Strong Communication:** Effective communication between developers facilitated the seamless integration of multiple components.

# Looking Forward

**Continued Implementation:**

The team will further develop the software to achieve full functionality.

**Iterative Process:**

The team will maintain an iterative development approach to create the software.



**Version Control:** GitHub will continue to be used for proper version control and project management.



**Task Division:** Functional requirements continue to be divided into tasks for better management and completion.



**Sprint Requirements:** The team will meet the necessary requirements of each sprint for continued progress.



# Thank you!

---

Any questions?