

# Project Notebook

**Team name:** Team Pantry

**App Name:** Our Kitchen

## Team members

Jinseok Heo, jheo61, [jheo61@gatech.edu](mailto:jheo61@gatech.edu)

Manasa Golla, mgolla3, [mgolla3@gatech.edu](mailto:mgolla3@gatech.edu)

Navya Gupta, ngupta361, [ngupta361@gatech.edu](mailto:ngupta361@gatech.edu)

Olufunmilola Obielodan, oobielodan3, [oobielodan3@gatech.edu](mailto:oobielodan3@gatech.edu)

## Team Agreement

Our team agrees to meet on Mondays and Wednesdays at 11:00 AM at the IC, with additional meetings scheduled as needed. We plan to communicate primarily through our group chat (text/phone), responding within 24 hours.

Each member is expected to participate actively by attending meetings, preparing in advance, contributing to discussions, completing assigned tasks on time, and notifying the group if conflicts arise.

Decisions will be made by consensus whenever possible, and if consensus cannot be reached, a majority vote will determine the outcome.

Roles will be assigned or rotated each sprint, including a facilitator to lead meetings, a recorder to document decisions, and a task manager to track deliverables, with all members sharing accountability for team success.

To stay organized, we will use Google Drive for file sharing with a shared folder holding all of our material, our group chat for communication, and a shared spreadsheet for task tracking. If disputes occur, the issue will first be raised respectfully, discussed openly, and resolved through consensus or majority vote.

We will also have interim deadlines so as to ensure completion and quality assurance of our final product.

Finally, this agreement will be treated as a living document that we review and update at the start of each sprint to reflect new processes, tools, roles, or timelines, ensuring it remains relevant to our evolving work.

## Problem Overview, Domain Research, and Competitive Analysis

Sprint 5:

### Updated Problem Overview

By Sprint 5, our understanding of the motivation gap expanded beyond internal intention and long-term habit formation to include the timing and structure of the user's day. Even when users were motivated to cook, many reported simply forgetting, losing momentum after work or classes, or abandoning plans when their day became unpredictable. This clarified that motivation is not just a behavioral deficit: it is also a temporal one. To address this, we introduced lightweight, personalized notifications and a dedicated Logging tab for meals, food experiences, and journals. User testing showed that supportive nudges tied to streaks or goals helped convert intention into action, and the separate logging space strengthened users' sense of progression without adding cognitive load. These additions reinforced a key insight from earlier sprints: users need micro-reflections and timely triggers that make cooking feel achievable in the moment, not just in theory.

Sprint 5 also deepened our understanding of how social structures support long-term engagement. Our expanded micro-communities and enhanced badge system showed that users respond best to small, goal-oriented groups where progress feels visible but not performative. These refinements made the platform feel more coherent and gave users clearer pathways for encouragement and accountability. A major shift this sprint was the addition of an admin dashboard, our second-platform approach, which enables community leaders or organizers to view high-level engagement patterns. This broadened the problem space by showing that motivation is not just an individual journey but something that can be strengthened through light, well-informed community oversight. Overall, Sprint 5 revealed that sustaining cooking habits requires the interplay of personal reflection, timely prompts, intimate social support, and structured community stewardship, all working together to transform cooking from an isolated task into a repeatable practice embedded within users' daily routines.

Sprint 4:

### Updated Problem Overview

While the modern digital ecosystem is filled with recipe databases, AI meal planners, and pantry trackers, the true obstacle users face is not access to information, it is sustained motivation. Our Sprint 3 findings revealed that users, particularly students and working professionals, consistently know what to cook but struggle to maintain consistency in

actually cooking. This disconnect stems from a lack of accountability structures and emotional reinforcement within current cooking tools. Existing apps typically serve as one-time utilities for discovering or organizing recipes but fail to create a feedback loop that reinforces behavior over time. Interviews and prototype testing underscored that users often begin with enthusiasm, only to disengage once the novelty fades, demonstrating that access to recipes or automation alone does not produce durable habits. The behavioral challenge, therefore, lies in helping users perceive cooking not as a burdensome task, but as a meaningful, repeatable act of self-care and creativity that can be sustained over weeks and months.

Our refined understanding of the problem centers on designing a digital framework that transforms motivation into routine through small, supportive accountability mechanisms. From our Sprint 3 prototype, we learned that visible streaks, personal milestones, and lightweight journaling provided a sense of achievement that users found rewarding without feeling pressured. However, participants also emphasized the importance of social reinforcement, gentle encouragement from peers within relatable, small-scale communities that feel personal rather than performative. This reinforces our conclusion that large, open networks or impersonal feeds are ineffective motivators for consistent behavior. Instead, what users need is a hybrid experience: one that enables private progress through solo tracking while introducing gradual social accountability through micro-communities once users feel comfortable. Our central problem, then, is not to create a better recipe or pantry app, but to build an environment that makes cooking a habit by weaving motivation, reflection, and community into a single, sustained behavioral loop.

### **Problem Overview:**

#### **Sprint 3:**

Micro-communities target the issue many users face, motivation and accountability, through turning cooking from a solo chore into a lightweight, social habit. Small, themed groups (our micro-communities) (e.g., “30-Minute Dinner Gang,” “Beginner Bakers,” “Parents with Picky Eaters”) create gentle, right-sized pressure and visible progress (“90% of us cooked this week”), with simple check-ins (photo + caption), prompts, and opt-in challenges that reward effort over perfection. Quick reactions and comments deliver immediate social reinforcement, while seasonal themes and group milestones build a shared narrative that keeps people coming back. Unlike big forums or rigid gamified tracks, these communities emphasize belonging and consistency without stress: accountability feels supportive, not punitive.

At the same time, the app must be genuinely valuable for solo users so it attracts and retains people even if they never post or do not want to interact with the public component of the app. A private dashboard lets them log meals, earn milestones, and see gentle, non-punitive streaks; configurable nudges and weekly recaps make progress feel tangible without pressure. Users can passively browse community content for inspiration, save ideas, and join later, meaning there's zero social tax on day one. This dual path differentiates us from recipe browsers (great content, no habit loop), self-contained streak apps (progress without people), and AI coaches (low trust for taste). Our overall goal for the app is to implement an approach that retains motivation and accountability through not only a community aspect proven to increase motivation, but also through the independent user experience of the app that aims to attract a wider range of users and promote engagement and habit-building. Unlike last sprint, where our problem framing focused too broadly on recipe complexity and pantry management, this sprint hones in on motivation and accountability as the core behavioral challenge. Our current focus is to test the individual user experience, specifically whether personal milestones, streaks, nudges, and journaling can motivate consistent cooking on their own, before layering in the community features to ensure the solo experience is viable and compelling.

## Sprint 2:

Many young adults want to cook more often to save money, improve their health, or explore creativity, but they struggle with motivation and accountability to maintain consistent cooking habits. In our previous sprint, this problem was framed too broadly. We emphasized how people struggle with cooking due to recipe complexity, planning, and pantry management without clearly identifying what prevents people from sustaining the practice. From user insights and researching other apps, we now understand that the central challenge is not a lack of ideas or convenience features, but the absence of structures that provide motivation and accountability. While existing cooking platforms excel at content discovery (offering endless recipes, meal plans, and inspiration) they fail to address the core behavioral challenge: cooking feels isolating and unrewarding compared to other lifestyle activities. Unlike fitness apps that provide social engagement, progress tracking, and community support, cooking tools leave users practicing alone without accountability systems or meaningful rewards for consistency. This creates a critical gap between intention and action, where people have access to cooking content but lack the motivational structures needed to transform cooking into a sustainable daily habit. We hope to create a platform that instills motivation and accountability to make cooking feel rewarding and sustainable. In this sprint, the most important questions we intended to answer revolved around what our problem actually is, clarified, and what

solution would be the best approach to solving that problem through sustained research, interviews, and user as well as classroom feedback. Specifically, the questions we want to address include: Is our real problem motivation/accountability? Do micro-communities motivate consistent cooking better than a Duolingo style app or an AI coach? How can we clarify past solution approaches to address our problem directly? Which solution approach has the most evidence to support it?

#### *Original Problem Statement (Sprint 1):*

*Many people want to cook more often (whether for health, creativity, or saving money) but struggle to make it a consistent habit. Cooking often feels isolating, time-consuming, and uninspiring, especially when compared to the social, gamified experiences people get from apps like fitness trackers or social media. Existing recipe platforms (e.g., Pinterest, AllRecipes) provide content but lack community-driven motivation, habit-building tools, or personalized guidance. This creates a gap: people want cooking to feel fun, social, and integrated into their lifestyle, but current tools don't help them turn it into a sustainable routine. Our targeted audience would primarily involve young adults, either in college or post-graduates.*

#### **Interviews**

##### **Sprint 2:**

- In this interview, I interviewed a working professional with not much time on their hands. They mentioned barriers to cooking were struggling in getting the groceries needed for recipes, the time to cook, and a lack of accountability. The main insights I obtained were that current cooking apps tend not to have an easy to use UI or the recipe function proves difficult. Gamifying cooking would be a great idea, and would motivate users. In a cooking app, they would want a place to source recipes, save them, take photos of the food you made to track your own recipes, and a place where you can write down your own recipe. They like the idea of a pantry-tracking function, especially if a simple photo of the contents could provide recipe ideas. They also mentioned that after a grocery run, if the AI could scan a photo of the receipt to track what is in the pantry, that would be a great addition. They would like a community aspect where there are certain forums that could be used like Reddit threads, and something that would be helpful for specific dietary goals (protein-heavy, low-calorie, etc.). They also mentioned that if the app's scope is a bit broad (i.e, having too many functions) this may be a bit overwhelming. They mentioned they would use an app with a homepage, a pantry page, and a community page. They would like notifications in a cooking app, but would like to customize these so that you could see when a friend or fitness influencer posts a recipe.

- In this interview, I spoke with a full-time master's student who also works part-time. Cooking is part of her daily routine, and while she enjoys making her own meals, she admitted that staying consistent and motivated can still be a challenge, especially on busier days. She typically plans meals around what she already has on hand, but she noted that keeping track of pantry items or logging groceries would only work if it didn't require much extra effort. She also reflected on what motivates her to cook. While she likes the idea of being part of a small cooking group, she worried that too many structured commitments might actually be discouraging. Similarly, streaks or rigid challenges feel stressful to her rather than rewarding. Instead, she finds encouragement from hitting personal milestones and being able to see visible progress, such as improvements in nutrition or the variety of meals she's cooked over time. When it comes to inspiration, she's open to ideas generated by both AI and communities, but she emphasized that accountability and motivation from a community of peers would keep her engaged more than AI alone. Tracking nutrition and progress, especially if connected to her health goals, was the feature she said would motivate her most. From this conversation, I learned that while many people may already cook frequently, consistency and motivation remain ongoing struggles. Strict systems like streaks or challenges can backfire, while flexible, milestone-based progress and supportive communities may be more effective in keeping people engaged. The discussion also highlighted how pantry tracking needs to be nearly effortless to be useful, and that trust in AI alone is not enough to sustain long-term engagement.
- I interviewed a 24 year old student who is interning in cosmetics marketing who enjoys cooking and usually shops once a week to stock up on ingredients. She primarily cooks alone and is motivated by the creativity of trying new foods, often discovering recipes on Instagram and YouTube. Her cooking goals include losing weight at a steady pace while keeping meals affordable, making both health and budget key priorities. However, she faces several challenges that affect her engagement. She often disengages when she does not have the right ingredient at home or lacks the proper tools, such as an oven. She has never tracked her pantry before, but she expressed frustration when ingredients remain unused for too long or when she struggles to figure out what to cook with leftovers. She also frequently forgets where she saved recipes, which makes it harder to follow through on her cooking plans. These insights suggest that she would benefit from a solution that simplifies pantry management, provides recipe suggestions tailored to available ingredients, and centralizes recipe storage. In addition, since she enjoys sharing her travel experiences on social media, she is also interested in uploading her cooking,

which points to an opportunity for community features that make cooking more social and engaging.

- In my first interview, I talked with a young adult who just moved in with her sisters. She stated that when she lived alone, she cooked about five days a week, but now she does not cook as much. She explained that she likes to cook for herself because she likes to have leftovers, but when she cooks for multiple people, there is much less of a possibility for this. Moreover, she said that time is a huge barrier to cooking for her, as cooking takes a while and she wants to eat ASAP when she is hungry. She mainly follows recipes on TikTok when cooking and likes to experiment with twists and substitutions, stating that the possibility of having her own unique recipe is what motivates her to do so. She likes this method of recipe use because you can see the end product and see real (human) reviews in the comments. However, she does not like how sometimes the ingredients and instructions are not in the caption, and so she has to watch the video over and over again in order to get everything. In terms of motivation to cook in general, she likes the satisfaction of the end product and plating and taking pictures of the food but is not a fan of the actual cooking process. She also mentioned that she has used AI to plan out her meals for a week before. She had the agent tailor the meals to her dietary restrictions. However, she explained that she did not follow all of it, mostly because she didn't like half of the recipes it suggested, and she ended up having to ask it for different recipes multiple times. Finally, she gave her thoughts on a community-based recipe/cooking app, stating that she'd want to post her own cooking pictures and would like if they praised her for them. This interview showed me that while AI can be a useful tool in many ways, it's not always reliable in providing something the user actually wants. On the other hand, a community-driven solution would be a greater motivator because of the possibility of positive feedback.
- In my second interview, I talked with a 4th-year college senior that cooks about 3-4 times a week, usually for others and not herself. She stated that she feels like she should cook more often, more for herself, and more for fun but is only motivated to cook if she knows other people will eat the food, as she loves to see the joy on their faces when they eat her food. Beyond cooking more, her cooking goals include making more cultural foods, especially Nigerian food. She explained that she wants to dive more into her cultural foods, as it sometimes feels embarrassing to not know how to make these meals. From this interview, I again learned that other people are a huge motivator for cooks. This interviewee mentioned that she

doesn't even really like most of the food she makes, but other people do, so she continues to make it.

Across our 5 interviews this sprint, we learned that motivation and accountability are indeed the core challenges, not recipe discovery or convenience features. We saw consistent themes emerge: users value community support and social connection over individual gamification, prefer human-created content over AI recommendations, and need flexible rather than rigid habit-building structures. These insights pointed us toward redefining our solution approaches from Sprint 1 to focus primarily on micro-community accountability rather than pantry management or AI assistance.

#### Sprint 1:

- The interview I conducted was with a post-graduate working a corporate job, with a bit more time than the average student. The interviewee cooks mainly out of financial need and health motivations, nothing that cooking makes them feel healthier. The apps they currently rely on include recipe websites and social media like Instagram or TikTok. Their likes are that they can explore other recipes, and dislikes involve frustration over unnecessary backstories or the lack of written instructions, causing them to rewatch a video multiple times. As for cooking everyday, they do not view this as much of a challenge, but trying new recipes they rated as a higher challenge to do. They mentioned cooking was more enjoyable when sharing with others, either cooking with someone else or showing off their dishes, and social media provides some motivation to cook. They expressed strong interest in an app that would build cooking streaks and share ratings and recipes with friends such as Beli, and would enjoy features like commenting on others' photos and saving other friends' recipes or saving recipes from social media. They do not trust AI to simplify recipes, but they mentioned it could be valuable for measurement adjustment, substitutions, or nutritional calculation. Overall, their responses reinforce the value of social features, streaks in cooking apps, and streamlined recipe presentation in a potential solution to the problem. They rated everyday cooking as a "2" in difficulty, suggesting it is somewhat challenging to maintain consistently. Cooking becomes more enjoyable when shared with others—either cooking with friends or showing off what they've made—and social media already provides some motivation.
- I interviewed two participants who highlighted different aspects of cooking habits and motivations. The first was a college student who loves cooking and engages in it multiple times a week, considering it one of her main hobbies. She enjoys trying new recipes and often discovers them through Pinterest, food blogs (especially for

baking), and online platforms like Food Network, TikTok, and Instagram Reels. While TikTok and Instagram's algorithms are helpful for saving recipes to try later, Food Network is mainly used when she has a specific recipe in mind. On Pinterest, she often searches for targeted recipes, such as "easy vegetarian recipes," but finds it time-consuming due to ads and navigating multiple links. She follows specific creators on TikTok, such as Plant You Susi and Naughty Fork, and will adapt existing recipes based on available ingredients or formulate her own ideas. As a vegetarian college student, she prioritizes easy, fast, and efficient recipes with simple ingredients that are both tasty and affordable, particularly when cooking for herself. She generally avoids unhealthy meals and strives to maintain balanced eating habits. She often cooks during the day, preparing larger portions to last several meals, and enjoys cooking with friends, which makes the experience more motivating and enjoyable. However, when cooking alone, her motivation decreases, especially if she is busy, stressed, or short on time. I also spoke with a post-graduate student currently in medical school, who also enjoys cooking and finds most of his recipes on YouTube. He appreciates that these platforms are concise, though he notes that exact measurements (e.g., in grams) are sometimes missing, which can be frustrating. Unlike the first participant, he does not have explicit goals regarding protein, efficiency, healthiness, or cost when following a recipe; however, for daily cooking, he keeps these practical considerations in mind. These interviews revealed a common insight: while people enjoy cooking, especially socially, motivation drops when cooking alone, and there are frustrations around recipe clarity, discovery, and personalization. This highlights an opportunity for solutions that combine community support, habit-building, and streamlined recipe guidance.

## Domain Research & Competitive Analysis

### Sprint 5:

#### 1. Habitica: <https://habitica.com/>

Habitica is a gamified habit-building platform that transforms daily tasks into RPG-style quests. Users earn experience points, maintain streaks, and unlock badges by completing real-life actions, creating a strong sense of progression and accountability. While Habitica succeeds at turning routine behaviors into fun, repeatable habits, its approach is highly generalized and not tailored to domain-specific contexts like cooking. The platform lacks emotional resonance around food experiences, community cooking goals, or reflective journaling about meals. It also leans heavily on extrinsic rewards (XP, avatars, gear), which can lose impact over time without meaningful, domain-specific ties to users' daily environments. These limitations highlight an opportunity for a cooking-focused

app that blends habit-loop design with reflective logging, recipe-centered action, and small social communities rooted in shared interests and personal growth rather than gaming mechanics alone.

## 2. MIT Research Article:

<https://news.mit.edu/2022/motivation-students-habits-learning-0407>

This MIT research article examines how consistent learning habits form when students receive timely prompts, supportive accountability structures, and opportunities for low-pressure self-reflection. The study found that even highly motivated individuals struggle to maintain routines without external triggers and small community-based reinforcement, patterns that closely mirror what we observed in our cooking-motivation research. The findings emphasize that habits stick when they are supported by moment-based cues, peer involvement, and micro-structures that reward incremental progress. This directly aligns with our rationale for introducing notifications, community streaks, and structured journaling within Pantry. By translating these motivational principles into a food-centered setting, our platform can support durable behavioral change rather than one-off bursts of enthusiasm.

Our expanded domain research and competitive analysis reinforce that the strongest models for sustained engagement come from behavior-driven platforms rather than traditional cooking apps. Tools like Strava, Noom, and Headspace demonstrate that long-term participation grows from lightweight accountability, streak preservation, well-timed prompts, and reflective practices that help users feel emotionally connected to their progress, validating our emphasis on visible progress loops, meal journals, and micro-community reinforcement as core elements of Pantry's motivational design. In contrast, much of the cooking-app landscape continues to prioritize convenience and content discovery while neglecting habit formation and social accountability. Apps such as Tasty, Yummly, and Nourishly provide inspiration or reflection but do not offer the structures needed to sustain daily or weekly cooking behaviors, and platforms like Cookpad or Pepper incorporate community features without the guided accountability or milestone-based progression required to maintain momentum over time. These gaps directly inform our Sprint 5 direction: strengthening individual motivation tools like streaks, personal milestones, notifications, and logs, while laying the foundation for micro-communities that create the intimate, supportive environments users need to turn cooking from an occasional activity into a repeatable and rewarding habit.

Sprint 4:

1. Tasty: <https://tasty.co/app>

Tasty, developed by BuzzFeed, is one of the most recognizable digital cooking platforms. Its focus is on short-form, visually appealing recipe videos that make cooking approachable. Users can filter recipes, save favorites, and follow step-by-step visual guides. However, Tasty's model centers on consumption rather than participation. There is little built-in accountability, no habit-tracking or community reinforcement, and engagement is limited to passive scrolling or liking videos. While it lowers the barrier to entry for cooking, it fails to sustain users' interest beyond one-off recipe discovery. This reinforces our finding that inspiration alone does not translate to lasting motivation or behavioral change.

2. Nourishly: <https://nourishly.com/>

Nourishly takes a mindful approach to food tracking, focusing on journaling and emotional reflection rather than calorie counting. Users can log meals, moods, and goals to foster healthier relationships with food. Its reflective design resonates with our journaling component, showing that self-awareness can increase intrinsic motivation. However, Nourishly lacks peer engagement, users are isolated in their reflections. For Pantry, integrating Nourishly's reflective strengths with micro-community accountability offers a more balanced motivational structure that supports both introspection and connection.

Our expanded domain research and competitive analysis continue to confirm clear gaps in the cooking and wellness app landscape. What works are platforms like Strava, Noom, and Headspace, which demonstrate that consistent engagement comes from behavioral reinforcement rather than pure functionality. Lightweight accountability, streaks, and small-group motivation drive sustained user involvement, while personalized milestones and emotional reflection help transform actions into lasting habits. The success of these models validates our approach to combine visible progress with gentle social reinforcement through micro-communities. On the other hand, what doesn't work are the majority of cooking apps, such as Tasty, Yummly, and Nourishly, which still emphasize discovery and convenience but neglect motivation, habit-building, and community belonging. Others, like Cookpad or Pepper, include social interaction but fail to offer structured accountability, leaving users without a clear feedback loop to sustain behavior over time. This sprint, our focus builds directly on these insights: refining the individual user experience to ensure that features such as streaks, milestones, and journals can independently support motivation, while strategically planning the rollout of micro-community features that reinforce long-term engagement.

Sprint 3:

1. Yummly platform: <https://www.yummly.com/>

Yummly is a popular recipe discovery and meal-planning platform that uses personalization algorithms to recommend recipes based on dietary preferences, ingredients on hand, and cooking skill level. Its strengths lie in its powerful filtering options, integration with smart kitchen devices, and user-friendly recipe organization. However, while it excels at personalization and convenience, Yummly focuses almost entirely on content curation rather than behavior change. It lacks structured accountability features such as streaks, challenges, or social interactions that could help users build consistent cooking habits. Our app differentiates itself by centering motivation and accountability through micro-communities and personal progress, rather than focusing solely on recipe recommendation.

2. SideChef Assistant: <https://www.sidechef.com/>

SideChef is a step-by-step cooking assistant that offers recipes, guided instructions, meal plans, and integrations with smart appliances and grocery delivery services. It positions itself as a comprehensive cooking solution, but its approach is top-down and instructional, emphasizing guided content rather than peer interaction or community-driven motivation. While the app can help beginners learn techniques, it does not provide ongoing habit reinforcement structures or social accountability mechanisms. Unlike SideChef, our app focuses on sustaining cooking behavior over time through milestone tracking, streaks, and micro-community challenges, turning cooking into a repeatable, social, and personally rewarding practice rather than just a one-time instructional experience.

Our updated domain research and competitive analysis continue to reveal clear gaps in the current cooking app landscape. What works are platforms like Strava and Duolingo which prove that lightweight social accountability, streaks, and milestone-based feedback drive sustained engagement, while research consistently shows that micro-communities are more effective than large forums for building motivation and habits. What doesn't work are most cooking apps that still emphasize recipe discovery and planning (e.g., Flavorish, Yummly, KitchenPal) without supporting long-term behavior change, or they rely on isolated individual tracking (e.g., Cookpanion) without meaningful social reinforcement. Even apps that include community elements (e.g., Cookpad, Pepper, Allcooks) lack the habit-forming structure and targeted micro-community design our research highlights as critical. This sprint, we're focusing on testing the individual user experience, validating whether features like personal dashboards, streaks, milestones, and prompts can motivate consistent cooking on their own, while laying the foundation for micro-community accountability to enhance motivation at scale.

## Sprint 2:

1. <https://www.nature.com/articles/s41599-024-03719-6>

The above article shows results from a randomized experiment where researchers assessed the impact of online learning communities on motivation and test scores. The results support that micro-communities encourage motivation and learning, substantiating our approach to motivating users to cook.

2. [https://books.google.com/books?hl=en&lr=&id=KPknDwAAQBAJ&oi=fnd&pg=PA\\_189&dq=micro+communities+increase+motivation&ots=xNEBRzyMZe&sig=L8iiEC7Aotiixe2od20REyyNWTs#v=onepage&q&f=false](https://books.google.com/books?hl=en&lr=&id=KPknDwAAQBAJ&oi=fnd&pg=PA_189&dq=micro+communities+increase+motivation&ots=xNEBRzyMZe&sig=L8iiEC7Aotiixe2od20REyyNWTs#v=onepage&q&f=false)

The study conducted in the above book further substantiates our solution, again emphasizing that communities do support and increase motivation in learning. This would allow us, again, to substantiate our problem and our solution approach. We see a direct correlation of motivation to micro-communities, directly addressing our approach to the problem set at hand.

3. Strava Article - Money Maze Podcast

<https://www.moneymazepodcast.com/blog/strava>

This article highlights how Strava became a leading social network for fitness by focusing on community, accountability, and progress tracking. Their motto, “Join for tracking, stay for the community,” really resonates with the type of experience I want to build. What stood out to me is that Strava isn’t just about elite athletes, it builds motivation by making progress visible and creating a social environment where people encourage each other. From this, I learned that a food app can adopt a similar model by centering its identity around the idea that “food brings people together.” Cooking progress and recipe sharing could be the equivalent of workout tracking, helping users stay accountable while also finding joy in the community aspect.

4. Strava Case Study - Social Plus

<https://www.social.plus/blog/how-strava-formed-the-worlds-biggest-team-a-community-case-study>

This case study dives deeper into Strava’s success, especially its use of gamified challenges and “friendly rivalry.” Users are motivated through in-app challenges, virtual groups, and the ability to comment on and compare activities. I learned that the competitive but supportive nature of these features can transfer to cooking by introducing recipe challenges (like “cook with 5 ingredients this week”),

collaborative cooking groups, or progress comparisons in a fun and low-pressure way. Another key takeaway is that Strava remains engaging even without heavy social interaction, which is important for making sure the app is enjoyable for solo users while still offering rich community features. It's important to build a rich experience for users to get a compelling narrative.

5. Duolingo

<https://ronntorossianupdate.com/how-duolingo-reached-new-heights-with-an-innovative-app-marketing-strategy#:~:text=Duolingo's%20marketing%20success%20is%20largely,become%20both%20fun%20and%20addictive>

This article explains how Duolingo's success comes from gamification, streaks, and personalized milestones that make the app feel both fun and addictive. The streak feature is especially powerful for building consistent engagement, and the personalized feedback loops help users feel rewarded and motivated to keep going. From this, I learned that cooking motivation can be framed in a similar way: streaks for daily or weekly cooking, personalized milestones (like "5 home-cooked meals this week"), and simple but rewarding feedback loops. These strategies show how I can make the app engaging by tapping into people's natural desire for progress and accomplishment.

6. KitchenPal

[https://play.google.com/store/apps/details?id=fr.icuisto.icuisto&hl=en\\_US](https://play.google.com/store/apps/details?id=fr.icuisto.icuisto&hl=en_US)

KitchenPal is one of the more popular pantry management apps, known for integrating multiple features into a single platform. Its strengths lie in functionalities such as barcode scanning for quick item entry, expiry notifications to prevent waste, and recipe suggestions based on available ingredients, all of which encourage healthier eating habits and better ingredient management. These features allow users to rapidly add items, track what they already have, and discover meals they can cook with existing supplies. However, despite these advantages, KitchenPal also faces challenges. Recipes are often provided only as links to external articles rather than fully integrated content, which disrupts the user experience. In addition, the app still places a heavy burden during the initial setup process, as users must manually input or scan a large number of items before the system becomes useful.

7. Cookpanion: Recipes, Meal Plan (Previously called Whisk)

<https://www.whiskapp.net/>

Cookpanion is a smart recipe and meal planning app that promotes easy recipe management, custom AI-made recipes, collaborative meal planning, dynamic grocery lists, saving recipes from the web, and daily recommendations. The app helps to bolster creativity in cooking, makes it easy to create meals tailored to dietary preferences and health goals, and helps users save time due to its AI-powered suggestions and recipe creations. Users report ease of use and enjoyment while using it. The application also has goals and streaks to motivate users to cook daily, but these are entirely self-contained with no community accountability. In fact, Cookpanion offers no community motivation beyond meal planning together. The application also allows for both manual and photo-based pantry addition, with many features being AI-powered. While this does allow for advanced features, it does not take into account the possible mistrust of AI and allow for non-use.

#### 8. Designli Article <https://designli.co/blog/how-communities-help-retain-app-users/>

This article talks about the “reward of being part of a tribe.” It claims that the human desire to connect with each other helps users form identity around an activity when it has a tight-knit community. This is what makes community so powerful. In turn, a community encourages you when your motivation wanes and drives you forward with positive reinforcement from the group. From this article, I learned that community is a great way to help people build habits, a direct connection to our approach of building micro-communities. Additionally, the article suggests that this is a good way to retain users.

#### 9. Allcooks <https://allcooksapp.com/>

Note: It looks like the app is still in the development process.

Allcooks was built to be a go-to for all things culinary, designed for users to get ideas and inspiration, help combat food insecurity, order ingredients, and connect with a cooking community. It also promotes progress through badges and rewards and has an AI sous chef assistant that gives personalized cooking assistance and recommendations. What differentiates our application are the micro-communities we intend to implement. While the application has a community aspect, it's more of a general community, we intend to get people even more connected by more specific interests. Additionally, it doesn't seem like there is much filtering when it comes to recipe inspiration.

Our domain research and competitive analysis reveals significant gaps in the current market. What is successful: Apps like Strava demonstrate that community-driven accountability works for habit formation, and research confirms micro-communities increase motivation more than large forums. What is NOT successful: Existing cooking apps either focus on recipe discovery without accountability (Flavorish, KitchenPal) or provide only individual tracking without meaningful community support (Cookpanion). The few apps with community features (Cookpad, Pepper) lack the targeted micro-community structure and habit-building focus that our research shows users need.

#### Sprint 1:

1. Link 1 and research summary: <https://app.flavorish.ai/>

Flavorish is an AI-powered recipe and meal-planning app that lets users save and organize recipes from any source, generate personalized meals based on ingredients or diets, track nutrition, and create grocery lists. However, its approach is largely limited to organizing and generating recipes from a restricted dataset, which can make the experience feel static rather than interactive. It lacks meaningful community contribution, where users could share, adapt, and improve recipes together, and it does not address real-world challenges such as managing the kitchen environment or making use of leftover pantry items. As a result, while Flavorish is strong as a digital cookbook, it falls short of delivering a truly dynamic, collaborative, and problem-solving cooking experience for everyday users.

2. Link 2 and research summary: <https://cookpad.com/us>

Cookpad is a global social cooking platform, heavy on community recipe sharing. Its mission is to make everyday cooking fun by sharing their recipes and cooking experiences. The platform enables sharing cooksnaps (user-cooked photos), commenting, recipe discussions, and even keeping a cooking log to track progress. Updates like dedicated 'Search', 'Create', 'Activity', and 'You' tabs help users interact more fluidly across discovery, contribution, and personal activity. While a community exists (cooksnaps, logs), Cookpad has several shortcomings. It lacks habit-building features like streaks, challenges, or reminders to encourage consistent cooking, its search functionality can be confusing or restricted without premium access, and the quality of user-generated recipes is inconsistent. Additionally, Cookpad is unavailable in the U.S., leaving a significant market gap. These limitations highlight opportunities for an app that combines community, habit-forming gamification, and AI-assisted recipe guidance to create a more engaging and reliable cooking experience.

3. Link 3 and research summary:

<https://www.peppertheapp.com/blog/the-story-of-pepper>

Pepper is marketed as the Instagram of cooking. It is designed as a social cooking app and boasts many users. It has recipe sharing, digital cookbooks, challenges, and grocery integration to make cooking more community-driven. While it does offer vibrant feeds and playful features, it primarily focuses on discovering new recipes/foods rather than building consistent habits in cooking through something akin to streaks. Its gaps include limited personalization, lack of meal planning or reminders, and few real-time social experiences. Our app could differ significantly from this and address these gaps by emphasizing forming habits through streaks, reminders, and shared cooking challenges. We could also implement personalized recommendations that address user lifestyle goals, integrate meal planning with smart grocery support, and create live or buddy cook-alongs. This shifts the focus of the app from just recipe discovery and sharing to making cooking a sustainable, fun routine for young adults.

## Solution Approaches

Sprint 5:

### Final Solution Approach

During Sprint 5, our solution approach continued to evolve as we incorporated new rounds of user testing, classroom feedback, and expanded domain research. The central theme of this sprint was understanding how moment-to-moment triggers, clearer progress tracking, and community-supported structure can strengthen long-term cooking habits. User feedback from our Sprint 4 prototype showed that while streaks and journals helped with internal motivation, many users still struggled to remember or initiate cooking in real time. This directly influenced our decision to introduce lightweight, personalized notifications and a dedicated Logging tab for meals and journaling. Early user tests confirmed that timely nudges and a cleaner, more intentional space for reflection made the app feel more actionable and reduced the friction of tracking progress. Classroom feedback also emphasized the need to clarify how these features contribute to measurable behavioral outcomes, encouraging us to refine success indicators such as log frequency, notification engagement, and week-over-week streak continuation. These insights helped solidify the idea that motivation is shaped not only by emotional reinforcement but by well-timed, low-pressure prompts built into the user's daily routine.

User interviews this sprint also deepened our understanding of how social accountability should be introduced. While some users were excited by the idea of communities, many wanted clearer structure, consistent goals, and more meaningful badges before engaging socially. This led us to strengthen our badge system, refine the design of micro-communities, and clarify how group spaces should support, not overwhelm, users' building habits. Feedback underscored that users prefer community features that feel purposeful, supportive, and tied to their actual cooking behavior. Domain research from Habitica, Strava, and micro-community literature reaffirmed that personal progress loops and structured social groups are most effective when each contributes distinctly to motivation. These findings continued to validate our hybrid, staged rollout: ensure that individual tools (logs, streaks, milestones, notifications) form a stable motivational core, then use micro-communities to reinforce those patterns through gentle social accountability and shared goals.

A major evolution in Sprint 5 was the introduction of the admin dashboard: our second-platform approach which emerged from both domain research and classroom discussions about scalability and real-world use. User interviews with potential community leaders (e.g., student orgs, clubs, RAs) showed that having visibility into group engagement patterns would help them create challenges, monitor participation, and support their members more effectively. This insight shifted our perspective from purely individual motivation to a dual-system model where communities have light-touch stewardship supported by data, mirroring models used by platforms like Strava Clubs or academic learning communities. Technically, this required us to clarify how user activity data could be aggregated and displayed without compromising privacy or overcomplicating our Firebase architecture. This two-platform perspective brought a new dimension to our solution: the idea that habit formation can be strengthened when community leaders are empowered to understand and guide group activity.

Across Sprint 5, we translated these insights into concrete refinements. We introduced notification testing to evaluate how timing and tone affect engagement, expanded the Logging tab to encourage consistent journaling and meal tracking, strengthened badge logic to increase reward clarity, and refined communities to offer more structured, goal-oriented spaces. We also began integrating admin-level analytics to test whether leaders could meaningfully motivate small groups. At the same time, we continued tracking metrics such as streak continuation, log frequency, notification interactions, and early indicators of community participation to determine whether these features genuinely enhance long-term cooking habits. This ongoing behavioral measurement ensures that our approach remains grounded in user behavior rather than assumptions.

Overall, Sprint 5 solidified our staged, evidence-based solution strategy: build and validate strong individual habit loops, introduce well-structured micro-communities that support those loops, and extend the ecosystem through a second platform that empowers community leaders to sustain engagement over time. This approach directly reflects accumulated user feedback, technical constraints, and domain research, ensuring that our expanding feature set remains coherent, scalable, and truly aligned with motivating users to cook consistently.

## Use Cases

### Use Case 1: Aisha – The Health-Conscious Professional (Goal Tracking and Long-Term Motivation)

Aisha is a 30-year-old marketing manager focused on maintaining her health and work-life balance. She uses Pantry to set personal challenges like “Cook 4 healthy meals per week” and receives encouraging notifications when she’s close to achieving them. The app’s visual streak tracker and milestone rewards keep her motivated, while gentle reminders ensure she doesn’t lose momentum during busy weeks. Aisha occasionally browses the “Healthy Weeknight Meals” group for inspiration but mainly values her solo progress tracking. For her, Pantry functions as a personalized accountability tool, reinforcing intrinsic motivation and measurable, sustainable growth.

### Use Case 2: Jordan – The Social Foodie (Motivated by Shared Progress and Group Challenges)

Jordan, a 24-year-old grad student, thrives on social interaction and loves sharing food experiences. He joins several micro-communities like “Weekend Brunchers” and “Quick College Meals.” When his group launches a new challenge, “Cook one meal inspired by another member this week”, Jordan participates enthusiastically, commenting on others’ posts and reacting to progress updates. His motivation stems from connection and light-hearted competition. Over time, his streaks and group milestones reinforce a sense of belonging. For Jordan, Pantry is not just a cooking app, it’s a digital social circle where mutual encouragement transforms small habits into shared accomplishment.

### Use Case 3: Priya – The New Cook (Confidence-Building Through Supportive Feedback)

Priya is a 19-year-old freshman who recently moved out and feels intimidated by cooking. She downloads Pantry to document her early attempts and joins the “Beginner Cooks” micro-community. When she posts a picture of her first homemade meal, others respond with encouragement rather than critique. Her dashboard shows a simple streak counter and progress badges like “First 3 Meals Cooked.” Seeing her improvement visually helps

her overcome fear of failure. For Priya, Pantry provides a psychologically safe entry point into cooking, helping her build confidence through visible progress, positive reinforcement, and gentle social accountability.

#### Use Case 4: Ravi – The Platform Administrator (Monitoring Consumer Patterns to Guide Feature Decisions)

Ravi is a 32-year-old product administrator responsible for overseeing overall user engagement across the Pantry platform. Using the admin dashboard, he reviews aggregated metrics such as daily log submissions, streak continuation rates, notification interaction patterns, and activity levels across micro-communities. He notices, for example, that users engage heavily with logging during weekday evenings but drop off on weekends, prompting him to adjust the timing and tone of automated notifications. He also observes that certain micro-communities, like “Healthy Weeknight Meals”, drive significantly more weekly activity than others, helping him prioritize which group structures should be refined or expanded. When a new badge is released, Ravi tracks how quickly users earn it and whether it increases meal-logging consistency. The dashboard helps him identify friction points, highlight high-performing features, and share data-driven recommendations with the development team. For Ravi, Pantry becomes not just an app, but a behavioral insights platform that enables him to support users at scale by aligning product improvements with real patterns of consumer interaction.

#### Use Case 4: Mateo – The Busy Parent (Time-Sensitive Prompts and Low-Effort Logging)

Mateo is a 38-year-old parent of two who loves cooking but often finds himself overwhelmed by tight schedules and unpredictable evenings. He isn’t interested in browsing recipes or participating in social groups, his priority is simply staying consistent. Pantry’s lightweight notifications help him remember when he planned to cook, and the new Logging tab makes it quick to record meals without disrupting his routine. Mateo often logs simple dinners like tacos or pasta right after cooking, earning streaks and badges that make him feel like he’s still making progress despite limited free time. The app’s quiet, unobtrusive prompts and clean tracking interface fit seamlessly into his day, giving him a sense of control even during hectic weeks. For Mateo, Pantry succeeds because it meets him where he is, offering timely support without requiring extra energy, scrolling, or social interaction.

### Final Solution Approach (Sprint 4)

#### Final Solution Approach

For Sprint 4, our team is refining our solution approach based on accumulated evidence from user testing, classroom feedback, and technical considerations. Our final approach centers on a hybrid model that prioritizes individual habit-building mechanisms while strategically layering in micro-community features. This decision stems directly from Sprint 3 prototype testing, where users found immediate value in solo features like streaks, personal dashboards, and meal logging, with many expressing that these elements alone made them feel more motivated to cook consistently. Classroom feedback emphasized the need to define concrete behavioral metrics and clarify how our technical architecture would scale as we introduce community features. Reviewers called for key performance indicators such as streak continuation rates, weekly active users, and average posts per user to quantify whether our design fosters sustainable cooking habits rather than just novelty-driven use. This reinforced that Sprint 4 must move beyond sentiment-based testing toward longitudinal behavioral measurement, tracking whether users actually return and cook consistently over time.

The choice to validate the solo experience before fully implementing micro-communities is grounded in both user research and technical pragmatism. Sprint 3 interviews revealed that not all users want immediate social engagement, with some preferring private tracking and personal milestones without pressure to interact with others. Feedback encouraged us to test solo motivation independently, leading us to prioritize building a robust individual experience that provides value regardless of community participation. Technically, this allows us to establish a stable data model for core features before introducing the complexity of community collections and challenges. Concerns about Firebase sub-collections becoming unwieldy prompted us to carefully plan our database schema. By validating solo engagement first, we can establish baseline metrics for retention and habit formation, then measure the incremental impact of community features, giving us clear evidence of whether micro-communities truly enhance motivation.

Our domain research validates this staged approach. Platforms like Strava and Duolingo demonstrate that habit-building apps succeed by creating feedback loops that reward consistency through streaks and milestones, features that work independently before social layers enhance them. Existing cooking apps like Cookpad and Pepper offer social sharing but lack structured accountability mechanisms, resulting in engagement that fades once novelty wears off. Research on micro-communities showed that small, goal-oriented groups increase motivation but only when paired with meaningful individual progress tracking. Feedback also suggested features like push notifications, visibility toggles for privacy, and interaction elements like reactions to posts, which we are incorporating into Sprint 4 to bridge the gap between solo and social modes.

Moving forward in Sprint 4, we are implementing several key refinements. First, we are tracking concrete behavioral metrics including streak continuation rate, day seven retention, weekly active users, and which interactions drive repetitive use. Second, we are prototyping lightweight notifications and reminders to test whether gentle prompts increase cooking frequency without feeling intrusive. Third, we are clarifying our technical architecture by documenting how individual and group data will synchronize in Firebase. Fourth, we are exploring features that support smooth transitions from solo to community mode, such as nudges to join groups after achieving milestones and visibility settings. Finally, we are conducting longitudinal testing to measure whether engagement over multiple weeks reflects genuine habit formation, using metrics like week-over-week active days and second-week retention.

This refined solution approach directly addresses core critiques from Sprint 3 while staying true to our evidence-based process. By validating the solo experience with concrete behavioral data in Sprint 4, we ensure our app provides immediate value to all users while laying the foundation for micro-communities to amplify motivation at scale. This staged approach reduces technical risk, allows us to test assumptions incrementally, and ensures that every feature we add is validated by user behavior rather than assumptions.

## Use Cases

### Use Case 1: Aisha – The Health-Conscious Professional (Goal Tracking and Long-Term Motivation)

Aisha is a 30-year-old marketing manager focused on maintaining her health and work-life balance. She uses Pantry to set personal challenges like “Cook 4 healthy meals per week” and receives encouraging notifications when she’s close to achieving them. The app’s visual streak tracker and milestone rewards keep her motivated, while gentle reminders ensure she doesn’t lose momentum during busy weeks. Aisha occasionally browses the “Healthy Weeknight Meals” group for inspiration but mainly values her solo progress tracking. For her, Pantry functions as a personalized accountability tool, reinforcing intrinsic motivation and measurable, sustainable growth.

### Use Case 2: Jordan – The Social Foodie (Motivated by Shared Progress and Group Challenges)

Jordan, a 24-year-old grad student, thrives on social interaction and loves sharing food experiences. He joins several micro-communities like “Weekend Brunchers” and “Quick College Meals.” When his group launches a new challenge, “Cook one meal inspired by

another member this week”, Jordan participates enthusiastically, commenting on others’ posts and reacting to progress updates. His motivation stems from connection and light-hearted competition. Over time, his streaks and group milestones reinforce a sense of belonging. For Jordan, Pantry is not just a cooking app, it’s a digital social circle where mutual encouragement transforms small habits into shared accomplishment.

### Use Case 3: Priya – The New Cook (Confidence-Building Through Supportive Feedback)

Priya is a 19-year-old freshman who recently moved out and feels intimidated by cooking. She downloads Pantry to document her early attempts and joins the “Beginner Cooks” micro-community. When she posts a picture of her first homemade meal, others respond with encouragement rather than critique. Her dashboard shows a simple streak counter and progress badges like “First 3 Meals Cooked.” Seeing her improvement visually helps her overcome fear of failure. For Priya, Pantry provides a psychologically safe entry point into cooking, helping her build confidence through visible progress, positive reinforcement, and gentle social accountability.

### Final Solution Approach (Sprint 3)

For subsequent sprints, our team refined and evolved our approach based on extensive user interviews, classroom feedback, and domain research, ultimately centering our solution around micro-communities paired with a strong individual user experience. Early in the process, we learned that our original framing which focused on recipe complexity, planning, and pantry management, was too broad and didn’t address the core behavioral barrier. Through interviews with students, working professionals, and young adults, a clear pattern emerged: users weren’t struggling to find recipes, they were struggling to stay motivated and accountable to cook consistently. Classroom feedback reinforced this insight, emphasizing the need to clearly define our problem scope and ensure our solution directly targets motivation rather than piling on more convenience features. This led us to pivot from AI coaching or gamified quest-based approaches (which received mixed to negative feedback) toward micro-communities, which users consistently described as engaging, motivating, and approachable.

Our domain research and competitive analysis further strengthened this decision. Platforms like Strava demonstrate how lightweight social accountability can transform routine activities into sustained habits, while research on micro-communities shows they outperform large forums in driving motivation. At the same time, existing cooking apps like Flavorish, Yummly, and KitchenPal excel at content curation but lack behavioral reinforcement mechanisms, and apps with community features like Cookpad and Pepper do not provide the structured, small-scale accountability that users need. Interviewees

shared this gap: they valued peer motivation, positive feedback, and the ability to share progress in small, themed groups more than they valued AI suggestions or rigid gamified progression. This evidence collectively validated our choice to focus on micro-communities and personal progress tracking as the backbone of our solution.

In Sprint 3, we're applying these insights to test the viability of the individual experience before layering in richer community interactions. Interviewees made it clear that not all users want to engage socially right away, so we're emphasizing features like personal dashboards, streaks, and milestones to ensure the app provides immediate value to solo users. This allows us to validate whether our habit-building mechanisms work independently, while laying a strong foundation for micro-community accountability to enhance motivation at scale. This evolution, from a broad, feature-heavy framing to a focused, behaviorally grounded solution, has been driven by consistent user feedback, competitive research, and classroom critique, ensuring our approach remains evidence-based and user-centered.

### **Use Cases:**

#### **Use Case 1: Emily – The Busy Student (Lightweight Accountability through Communities)**

Emily is a 20-year-old college student living with roommates, juggling classes, extracurriculars, and part-time work. She wants to eat healthier and save money but struggles to maintain consistent cooking habits because she's often tired at the end of the day. In the app, Emily joins a micro-community called "30-Minute Dinners," which focuses on quick, low-effort meals. After cooking dinner one night, she posts a quick photo and caption to her community feed. Her friends cheer her post, and she sees the group's progress bar showing that most members cooked twice this week, motivating her not to fall behind. She earns her "First Week of Home Dinners" milestone and gets a light nudge a few days later when she hasn't logged a meal. For Emily, the combination of low-stakes community accountability, visible group progress, and gentle reminders helps her stay consistent without feeling overwhelmed.

#### **Use Case 2: Maria – The Independent Cook (Solo Experience as Core Value)**

Maria is a 35-year-old single mom who cooks primarily for her two kids. She has little time for social interaction in apps and wants to keep things simple, but she does want to track her family's home-cooked meals and occasionally find inspiration. Maria uses the app's personal dashboard to log meals, add notes about what her kids liked, and track her "10 Family Dinners Cooked" milestone. She occasionally browses community posts for quick recipe ideas but doesn't actively participate. The streaks, milestones, and private

journaling features give her a sense of progress and personal accomplishment, even without joining communities. For Maria, the app functions like a lightweight cooking log and motivator, helping her sustain healthy habits without demanding social engagement. This use case ensures our solution remains valuable for users who prefer to stay mostly solo.

#### Use Case 3: David – The Aspiring Home Chef (Skill-Building through Themed Micro-Communities)

David is a 28-year-old working professional who enjoys cooking but struggles with consistency and wants to expand his skills. He joins a themed community called “Beginner Bakers” to learn new techniques and keep himself accountable. Each month, the group launches a new challenge, such as “Try two new baking methods.” David participates by documenting his attempts, posting progress photos, and receiving feedback from peers. He also uses his personal dashboard to track which recipes he’s tried, noting improvements over time. Seeing other members’ posts inspires him to experiment, while the community’s shared goals and streaks push him to stay engaged week after week. For David, the app becomes both a learning tool and a motivational structure, blending community accountability with individual growth.

#### Refined Solution Approaches (Sprint 2):

Taking into consideration the feedback we received in Sprint 1 as well as user interviews and domain research, we redefined both our problem scope and solution direction.

Sprint 1 showed us that our framing was too broad. Additionally, reviewer feedback on our initial solution approaches emphasized that streaks or AI alone would not sustain engagement, while pantry setup carried too much friction. From additionally user interviews, we also learned that people don’t just struggle with recipes or planning; they struggle with accountability and consistency in their cooking habits. In Sprint 2 we are focusing on motivation and accountability through community as a core lens, while treating convenience features and AI support as secondary, future-ready enhancements.

From feedback, research, and additional interviews, we evolved our three Sprint 1 approaches into testable concepts that directly address motivation and accountability:

#### **Approach 1: Duolingo-Style Application for Cooking (Evolution of Health App Integration)**

Our original health app integration approach focused on connecting cooking activity to existing fitness platforms. We refined this into a standalone gamified experience that

could provide similar habit-building reinforcement without requiring external app dependencies. This concept tackles the core challenge of motivation and accountability by **transforming cooking into a gamified practice**. Users build streaks, go through levels of cooking new recipes like 'budget-friendly,' 'less time,' and 'new ideas,' and receive light gamified nudges, making consistency in cooking feel rewarding rather than a chore.

Gamification has proven successful in habit formation (Duolingo, fitness apps), and this approach could provide structured motivation without requiring social interaction. It addresses the accountability challenge through consistent progress tracking and rewards.

However, there are some downsides to consider for this solution. First, we see many other apps exist that target something very similar to this approach. Candidates include Zest cooking, which has levels, streaks, and badges, as well as Parsnip, which is an AI version that again tracks progress and has levels and streaks. We need to test whether users find cooking levels and quests personally engaging, if the progression system motivates consistent cooking, and whether the gamified structure feels natural for cooking activities.

Example use case: A busy college student opens the app after classes and sees a "Budget-Friendly" cooking quest waiting for them. They follow the simple pasta recipe, take a photo when complete, and unlock the next level while earning their first "Rookie Chef" badge. The progress bar and streak counter motivate them to return tomorrow for another cooking challenge.

### **Approach 2: AI Cooking Coach (Evolution of pantry management assistant)**

Our original pantry management assistant focused on smart inventory tracking and recipe suggestions based on available ingredients. We refined this into an AI coach that could provide the same personalized guidance and recommendations but expanded beyond pantry management to offer comprehensive cooking support and accountability. The goal of this AI agent is to **track progress, suggest recipes, and act as a coach for the user**. In this way, we target accountability through an AI coach and focus on motivation through progress tracking. An AI coach could provide personalized accountability without the social pressure some users might find intimidating. It addresses both motivation (through encouragement) and guidance (through recipe suggestions) while being available whenever users need support.

A downside to consider is that an AI assistant would be an expensive solution, and if our interviewees did not see the value in it, we do not want to invest significant cost in developing this approach. We need to test whether users trust AI recommendations for

recipes, if AI coaching feels personally engaging enough to drive behavior change, and whether the coaching interaction creates sufficient accountability.

Example Use case: After dinner, a graduate student opens the app to see their AI coach reviewing their cooking history. The coach notices they've been making repetitive meals and suggests a quick stir-fry with a shopping list and step-by-step guidance. After cooking, the user uploads a photo and receives personalized feedback and encouragement, with their progress dashboard showing improved variety and nutrition trends.

### **Approach 3: Micro-Communities (Chosen Solution)**

Our third solution is similar to solution 2 in Sprint 1, centered on **micro-communities for cooking accountability**.

Unlike existing cooking apps, which focus on passive browsing or recipe databases, our design is rooted in active habit formation. The micro-communities will be intentionally small, lightweight, and goal-oriented to create a sense of accountability and belonging. There are not large, anonymous forums; instead, they are designed to provide a personal, low-friction way to build habits with others.

We were inspired by Strava's community model, where friendly rivalry and lightweight sharing drive engagement among everyday users. Similarly, our app will promote approachable accountability. Lightweight participation will be built into the design with daily or weekly check-ins, optional challenges, progress prompts, and friendly comparisons with peers. This encourages consistency without overwhelming users and makes cooking approachable with emphasis on effort and consistency, not skill (e.g. cooking twice in a week "counts" regardless of outcome). This framing makes the app inclusive for beginners while motivating committed home cooks.

At the same time, our research made it clear an engaging solo experience is critical for the product's success. We want to ensure users can benefit regardless of joining communities or interacting with friends. Users will still be able to use personal progress tracking with milestones, journaling, and stats. This ensures a base layer of value. Another key learning from our research is that narrative and aesthetics matter. Cooking/food is heavily tied to memory, identity, and shared experience. Our solution will be designed to feel more like documenting a journey than logging chores. Seasonal challenges (e.g. a fall baking challenge) or group milestones (e.g. 90% of our community cooked this week) will provide users a sense of story and collective momentum.

To avoid the setup friction flagged in Sprint 1, pantry-based suggestions are deprioritized in the MVP. Instead of requiring users to log their pantry, we reserve AI-powered photo

recognition for later. This way, the app avoids early complexity while staying future-ready for when user trust in AI grows. In the learning prototype section, we see interviewees clearly appreciate the app interface and design, as well as how they themselves would interact with an app like this. In our research, we have found practical studies that show how micro-community engagement is linked to higher motivation, directly addressing our problem statement and contributing to its being solved.

Overall, our approach differs from existing apps by anchoring on habit formation through micro-community accountability rather than recipes or discovery. We balance individual progress and collective motivation, avoid early gimmicks, and build a product that works for a solo user but grows stickier with social participation. By rooting our solution in feedback, user interviews, and proven models, we've arrived at an approach that directly addresses Sprint 1 critiques: clarified problem scope, reduced friction, and a habit-focused solution.

We need to test whether users prefer small, goal-oriented groups over existing platforms, if micro-community accountability translates to sustained cooking habits, and whether the approach provides sufficient solo value for less socially-inclined users.

#### Sprint 1:

- **Solution Approach 1:** Cooking habit coach with wearable/health app integration. This would be more akin to habit-building and would link cooking activity, like logging meals or taking photos of a dish, to fitness platforms like Fitbit or Apple health. This would also integrate streaks, dashboards, and rewards into apps that users already check daily, making it much easier for users to stay motivated and consistent. This approach would directly connect cooking frequency to wellness and thus prioritize habit building over other features. Unlike typical recipe or social feed apps, this approach directly connects cooking frequency to wellness tracking, reinforcing the habit through the same tech people use to monitor their overall health.
- **Solution Approach 2:** Our second proposed solution is a social app designed around micro-communities where people can connect, share, and stay motivated toward specific lifestyle goals, starting with cooking. Instead of broad, noisy feeds like Instagram or TikTok, users can join or create small groups focused on particular themes (e.g., “cooking for fitness,” “budget-friendly meals for students,” or “exploring global cuisines”). Within these communities, users can share photos of what they’ve made, link recipes, ask questions, run polls, and exchange inspiration in a supportive environment. Posts will include tags for diet, allergens, and meal type (snack, breakfast, dinner, drink, etc.), making it easy to browse and discover

ideas that fit personal needs. To make the experience rewarding, users earn streaks by uploading at least once per week, building a visual “Cooking Passport” on their profile that showcases their taste profile through photos of everything they’ve cooked. This passport not only provides a sense of accomplishment but also creates a fun, aesthetic collection that friends and community members can explore. AI can optionally enhance the experience by generating high-level recipe summaries from links, simplifying nutrition insights, or even reimagining food photos into whimsical, shareable art styles for added appeal.

- **Solution Approach 3:** Students and everyday users often struggle with the demands of grocery shopping and meal planning, which can lead to repetitive meals, forgotten ingredients, and wasted food. The pantry management assistant solution is designed to ease this burden by giving users a smarter way to manage their kitchen inventory. Through smart pantry tracking, users can log or scan items as they shop, monitor what they already have, and receive reminders before items expire. With recipe suggestions based on inventory, the assistant recommends meals that fit the ingredients users already own, while also pointing out simple substitutions for missing items. To reduce planning effort, the tool provides shopping list automation, creating organized lists of what users need to restock or buy for upcoming meals. It also incorporates budget-friendly planning, helping students estimate total grocery costs, find affordable alternatives, and prioritize purchases that stretch their budget. Together, these features allow users to save time, cut down on food waste, and make healthier, more cost-effective meal choices with less effort.

## Learning Prototype (Methodology & Testing Process)

Sprint 5:

### What We Built

In this sprint, we focused on testing whether timely prompts and structured or self-directed goal-setting could help users follow through with cooking habits during moments when motivation typically drops. To explore this, we built a Thanksgiving Challenge prototype supported by an A/B test across two versions of the challenge. Version A provided a guided structure with a pre-defined checklist of 5 curated tasks designed to reduce cognitive load for students and newer cooks. Version B offered a self-directed goal-setting experience, prompting users to define their own Thanksgiving cooking goals and action items, supporting autonomy for working professionals and

experienced cooks. Both versions were supported by a multi-platform personalized notification system that delivered timely reminders, streak-preservation nudges, and community activity alerts to test whether temporal interventions increased follow-through. We also introduced a streamlined quick-entry logging experience that allowed users to log meals with photos, captions, and optional reflections, immediately updating streaks and badges. All components were implemented using React Native + Expo, with Firebase providing backend logic, Cloud Functions for A/B assignment and analytics tagging, and real-time Firestore synchronization. Together, these features created a controlled environment for observing how structure, autonomy, and timing influence users' real-world cooking behavior.

Our Sprint 5 prototype focuses on solving the temporal motivation problem: users who intend to cook still fall off when real-life interruptions break their momentum. We built three interconnected features to test whether timely, lightweight interventions can convert intention into action:

### 1. Thanksgiving Challenge with A/B Testing

- a. Version A (Guided Structure): Pre-defined challenge with 5 structured tasks focused on community engagement and personal cooking goals. Users see a checklist of specific, curated actions to complete during Thanksgiving week.
- b. Version B (Self-Directed): Users define their own Thanksgiving cooking goals and create custom action items. Provides flexibility and autonomy in goal-setting.
- c. Badge System: Special "Thanksgiving Challenge" badge awarded upon completion, integrated into our existing gamification infrastructure.

### 2. Personalized Notification System

- a. Multi-platform push notifications (Android + Web) that fire at user-defined optimal times
- b. Streak-preservation alerts when users haven't logged activity in several days
- c. Goal completion reminders tied to weekly progress
- d. Community activity notifications when friends/community members post

### 3. Quick-Entry Logging Tab

- a. Streamlined meal logging with photo upload
- b. Optional journaling/reflection prompts
- c. Private vs. public post options
- d. Community sharing checkboxes for cross-posting to relevant groups
- e. Immediate streak and badge updates upon submission

Technical Implementation:

- Built with React Native + Expo for cross-platform compatibility (Android + Web)
- Firebase backend with 8+ Cloud Functions for server-side logic
- Real-time Firestore synchronization for instant updates
- A/B test group assignment via Cloud Function with atomic transactions
- Firebase Analytics integration for behavioral tracking and segmentation

## Questions for Testing

The primary research question guiding this sprint was: Will timely, lightweight prompts combined with structured or self-directed goal-setting help users follow through in the moment when motivation typically drops? To explore this question, we tested three categories of hypotheses. First, we examined structure vs. autonomy, predicting that students and newer cooks (H1a) would benefit more from Version A's guided checklist due to reduced planning friction, while working professionals and experienced cooks (H1b) would prefer the flexibility and ownership offered by Version B. Second, we evaluated temporal intervention effectiveness, hypothesizing that users who received timely notifications would complete their first Thanksgiving action at higher rates (H2a), and that pairing notifications with quick logging would reduce activation energy and increase follow-through (H2b). Third, we investigated social vs. individual motivation, predicting that community-focused challenge tasks would generate more engagement than purely personal tasks (H3). Ultimately, the results of this test will inform a major product decision: Should future habit-building features be guided (curated challenges and structured plans) or self-driven (user-defined goals and flexible frameworks)? This decision directly influences the design of our onboarding experience, challenge formats, and long-term goal-setting systems.

## Methodology for Testing

Our methodology centered on a controlled A/B test conducted during Thanksgiving week (Nov 23–29), involving 10+ users recruited through university channels, social networks, and prior prototype testers. Users were randomly assigned via a server-side Cloud Function to either Version A (Guided Structure) or Version B (Self-Directed Goals), with an additional segmentation layer identifying students versus working professionals. Version A participants received a pre-defined checklist of tasks—such as sharing a Thanksgiving recipe, engaging with community posts, logging a Thanksgiving meal, writing a journal reflection, and maintaining their weekly streak—which they tracked through a dashboard, earning a badge upon completion. Version B participants instead created their own 5 Thanksgiving-related goals, optionally shared them with communities, and earned a badge upon marking all personal goals complete. Both groups used the same notification

system, logging interface, community feed, and badge/streak infrastructure, ensuring that the only variable manipulated was the structure of goal-setting.

Data collection relied primarily on Firebase Analytics, where every interaction was tagged with A/B test group and user segment. Activation metrics included completion of the first Thanksgiving action within 24 hours, time from goal-setting to first action, and notification open-to-action conversion. Engagement metrics measured challenge completion rate, number of posts, journal entries, and community interactions. Retention metrics captured daily return rates, streak continuation, and post-challenge usage. Behavioral patterns, such as time-of-day activity or photo upload frequency, helped identify friction points and user preferences. To complement quantitative data, we also conducted qualitative interviews with high performers, drop-offs, and behavioral outliers, using insights to validate assumptions and reveal contextual factors that analytics could not capture. A real-time analytics dashboard displayed completion rates, segment breakdowns, notification funnels, and daily activity trends, enabling us to monitor experiment progress and ensure data-driven iteration throughout the sprint.

### Outcomes + what we were looking for:

#### Success Indicators for Version A (Guided Structure):

- Higher completion rates among students and newer cooks
- Lower time-to-first-action (users start faster with clear instructions)
- More uniform completion patterns (everyone does similar tasks)
- Comments from users: "I liked knowing exactly what to do"

#### Success Indicators for Version B (Self-Directed):

- Higher completion rates among working professionals and experienced cooks
- Greater variety in types of goals set
- Higher satisfaction scores in post-test surveys
- More personalized, creative goal definitions
- Comments from users: "I appreciated the flexibility"

### Cross-Version Insights We're Seeking:

#### 1. Community vs. Personal Task Preference:

- a. Which type of task gets completed first?
- b. Do community-focused tasks drive more app visits due to social notifications?
- c. Does social accountability increase completion rates?

## 2. Notification Effectiveness:

- a. What % of completed actions happen within 1 hour of a notification?
- b. Do users who customize notification timing have higher completion rates
- c. At what point do notifications become annoying vs. helpful?

## 3. Logging Friction:

- a. How often do users upload photos vs. text-only logs?
- b. Do simpler prompts increase logging frequency?
- c. Does the ability to make posts private increase overall logging?

## 4. Temporal Patterns:

- a. When during Thanksgiving week do most users engage?
- b. Do we see the "motivation drop" we hypothesized (declining activity mid-week)?
- c. Can we identify optimal notification timing from user behavior?

## Analysis Approach

### 1. Quantitative Analysis (From Firebase Analytics):

#### Completion Rate by Group:

- Version A (Guided): X% completed all tasks
- Version B (Self-Directed): Y% completed all goals
- Statistical significance test: Chi-square test,  $p < 0.05$  threshold

#### Segmentation Analysis:

- Students in Version A: completion %
- Students in Version B: completion %
- Professionals in Version A: completion %
- Professionals in Version B: completion %
- Identify interaction effects (does user type moderate version effectiveness?)

#### Time-to-First-Action:

- Version A median time: X hours
- Version B median time: Y hours

#### Engagement Metrics:

- Posts per user by version
- Community interaction rate by version
- Streak continuation rate by version

## 2. Qualitative Validation (From Selective Interviews):

After identifying patterns in analytics (e.g., "Students in Version B dropped off more"), we interview 3-5 users from that segment:

- "Can you describe your experience with setting your own goals?"
- "What made you decide to stop participating?"
- "If you could change one thing about the challenge, what would it be?"

This confirms whether our analytical conclusions match user perception and reveals context we couldn't capture in metrics alone.

Sprint 4:

### Prototype Background:

We continued building our application using React Native for the front end and Firebase for the backend. We focused on implementing the essential functionality needed to interact with other people, set goals, and accomplish streaks through activity tracking. These functions of the app were deemed essential for testing the motivational aspect of the application.

The front end currently includes the Home tab. Here, users can submit a journal entry, set goals for the week, and see and interact with other users' posts. There is also the Log tab. Here, users can upload a log of what they cooked for that meal with a caption and the choice to make it a private or a public log. The private log still updates their meal count and contributes to their weekly goal but does not appear on the community log. A public post will have all the same benefits and also post to the community log. Finally, there's the Profile tab. Here, users can see their progress on their weekly goals, the logs they have created (both private and public), their streaks, their badges (calculation implementation in progress), and change their username and profile pictures (through the settings page). The community and explore tabs are yet to be implemented.

The interface is designed to be lightweight and intuitive, minimizing friction so users can easily log their activity without complex setup or entries. This build lays the foundation for testing whether simple, personal habit-tracking features or community features better support motivation.

This sprint focused on implementing the minimal viable prototype to allow for users to complete either personal or community-based tasks. Future iterations will expand on

these features, making them more complete and building on them to grow a more robust application. This will include things such as notifications, friend functionality, community challenges, personal goals and journal prompts that update, and overall more structured activity.

### **Main Hypothesis and Questions:**

Our project is centered around the idea that while access to information and recipes is important, the greater barrier to consistent home cooking is sustained motivation and engagement. In Sprint 4, our team continued to develop our learning prototype to evaluate our central hypothesis—combining information with social support through micro-communities addresses the gaps in this area and leads to long-term habit change. We have also realized through previous sprints that the individual components of the application are additionally important. Features such as streaks and journal reflections also encourage users to continue their cooking journey. Our main questions for Sprint 4 is to see if the micro-communities and or positive social reinforcement, are enough to drive user engagement? Or is micro-community involvement more effective when paired with personal features?

### **Methodology for Testing:**

To test our learning prototype, we conducted an experiment in which we sent out our MVP to 10 individuals to use for about 2.5 days. Half of these users were put into the “Group A” category upon sign-up, and half were put into “Group B.” Group A was given access to all of the currently developed features, including the setting of personal goals, streaks, and community feed. Group B, however, did not have access to any of their personal habit-building features of the app. The only way they could make use of the platform was to create and interact with other people’s posts. Additionally, we collected usage data through the Firebase analytics page, which we then analyzed to compare which group interacted with the app more. Finally, some users provided feedback notes on how they felt about using the app for those couple of days.

### **Sprint 3:**

#### **Prototype Background:**

We built a mobile application using React Native for the frontend and Firebase for the backend, focusing on implementing the essential functionality to support habit formation through cooking activity tracking. The frontend currently includes a Home tab, where users can see journal prompts and their most recent check-ins, a Post interface that allows users to upload images and captions of their cooking, and a Profile tab that displays

streaks and basic personal stats. The interface is designed to be lightweight and intuitive, minimizing friction so users can easily log their activity without complex setup or pantry entry. This initial build lays the foundation for testing whether simple, personal habit-tracking features can effectively support motivation before introducing community features.

While this sprint focused on implementing users, posts, check-ins, and streaks, future work will expand the architecture to include communities and memberships, challenges and milestones, and notifications for nudges and weekly recaps. These features will build on the current foundation to support more structured social accountability, goal-setting, and engagement at scale. We also plan to provide more detailed architecture diagrams, code repository links, and extended API documentation as the system evolves.

### **Main Hypotheses:**

In Sprint 3, our team developed a first learning prototype to evaluate our central hypothesis that, while access to information and recipes is important, the greater barrier to consistent home cooking is sustained motivation and engagement. Our testing suggested that knowledge alone does not translate into long-term habit change; users often know what to cook but struggle to maintain the consistency required to build healthier routines. The micro-community model directly addresses this gap by combining information with social support, enabling users to form small groups, share meals, set challenges, and celebrate achievements together. This structure transforms cooking into a shared experience, providing both the resources to learn and the accountability to act. Based on this approach, we plan to refine the app with features that reinforce engagement such as streak tracking, peer challenges, and collaborative goal-setting, ensuring that users stay motivated not only through access to content but also through encouragement from their community.

### **Methodology for Testing:**

To test our hypothesis that motivation is a stronger driver of sustained cooking behavior than access to information, we conducted a structured user experiment with 9 participants, primarily college students and young professionals aged 20–30. Each participant was asked to complete three prototype tasks: Following these tasks, participants completed a structured survey and interview focusing on usability and motivational drivers. The evaluation assessed how comfortable the app felt to use, what aspects encouraged participants to cook, what aspects motivated them to engage with the app itself, and how open they felt about sharing their posts. Each dimension was captured through a mix of 1–5 scale ratings and qualitative feedback, allowing us to link

quantitative outcomes with richer user reflections. We then analyzed the results by comparing task completion rates and satisfaction scores across participants. This experimental design provided a quantitative basis for evaluating whether the micro-community approach effectively addressed motivational barriers, thereby informing our decision to advance this design direction.

Sprint 2:

### **Methodology**

We created prototypes for each of the three refined approaches and conducted structured user testing sessions. Each participant interacted with the prototypes for 10-15 minutes per approach, followed by interviews focusing on engagement, trust, and likelihood of sustained usage. We interviewed 6 participants: college students and young professionals aged 18-35 who cook 2-5 times per week. Our testing order varied to avoid bias, and we asked participants to think aloud while navigating each prototype before discussing their reactions.

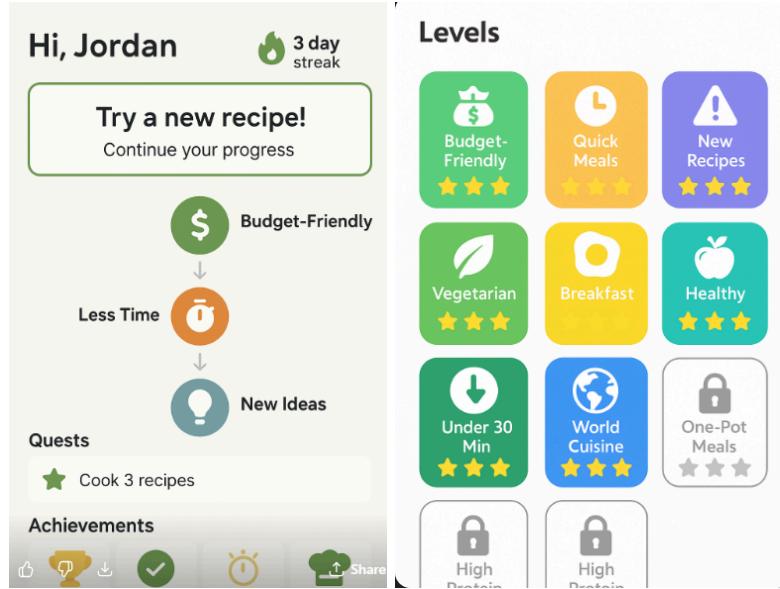
### **Mockups Created**

For each approach, we developed testable prototypes:

#### **Approach 1: Duolingo for Cooking**

We created a brief mockup and interviewed users on what they think of this idea to address their lack of motivation and accountability in cooking. Here, we gamify the process of cooking and attempt to create levels such that users feel motivated to continue cooking every day. This mockup features cooking levels, quest-based recipe assignments, progress bars, badges, and streak counters. Users could navigate through different "tracks" like "Budget-Friendly" or "Quick Meals" with locked progression.

Below is the mockup of this approach:

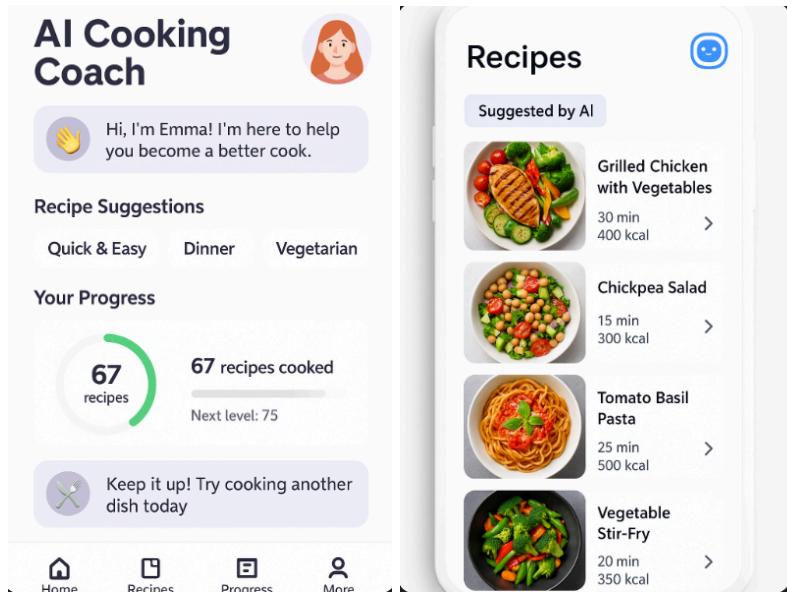


**Primary Usecase:** After a long day of classes, the user wants to save money and eat something homemade instead of ordering takeout, so they open the app for inspiration. The first quest waiting for them is a simple pasta dish from the Budget-Friendly track. They follow the steps, cook the meal, and feel confidence when the app unlocks the next level. A progress bar marks their advancement, and a cheerful notification awards them their first badge, the "Rookie Chef." With a streak now underway, the user feels excited to return tomorrow, motivated by small wins and slowly building both confidence and skill in the kitchen.

### Approach 2: AI Coach

We created a brief prototype featuring an AI dashboard with personalized recipe suggestions, progress tracking, coaching feedback, and conversation interface for guidance and questions. We then interviewed users on what they think of this idea to address their lack of motivation and accountability in cooking. Here, we target accountability through an AI coach and focus on motivation through progress tracking.

Below is the mockup of this approach:

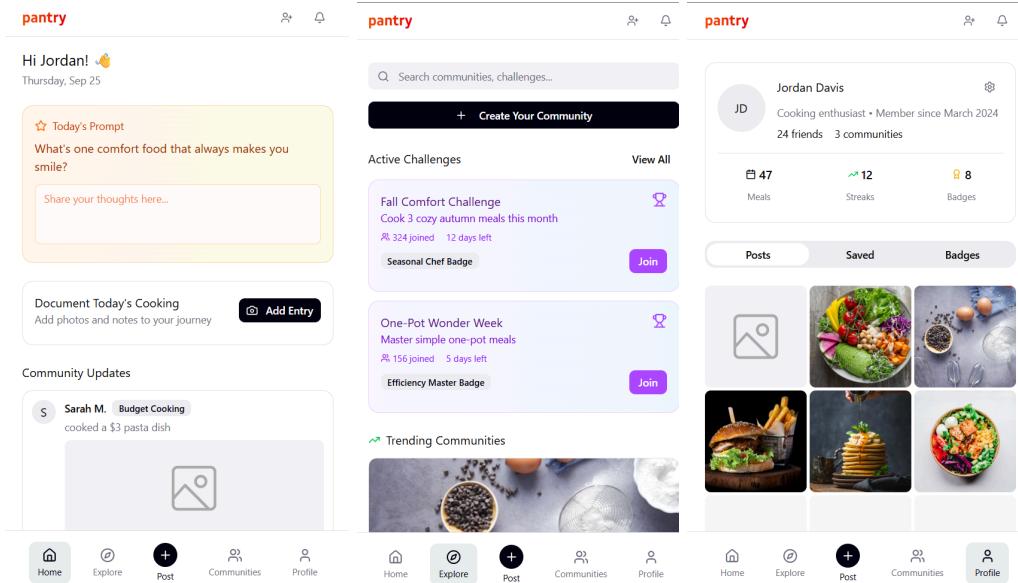


**Primary Usecase:** After finishing dinner one night, the user opens the app to see what their AI coach recommends next. The coach reviews their cooking history and notices they've been making a lot of pasta, so it suggests a quick stir-fry to add variety and balance, complete with a shopping list and step-by-step instructions. The next evening, the user follows the recipe, uploads a photo, and marks the dish as complete. The app logs the meal, updates their progress chart, and highlights their improvement: "67 recipes cooked – keep it up! Try another dish today." Over time, the dashboard begins to show trends such as healthier choices, faster prep times, and greater variety. The AI coach celebrates small milestones with motivational messages and tailored suggestions, helping the user feel supported and steadily improving.

### Approach 3: Micro-Communities

We created a brief mockup and interviewed users on what they think of this idea to address their lack of motivation and accountability in cooking. This is a clickable prototype featuring small themed communities, friend activity tracking, group challenges, personal progress dashboards, and community feeds with photos and interactions. Here, we focus on habit formation through micro-community accountability. Below are a few screenshots of this prototype, but more depth can be seen through this link:

<https://nicks-vocal-49864785.figma.site/>



**Primary Usecase:** The user realizes their eating habits have been mostly takeout, so they open the app looking for a way to cook more balanced meals. They join a micro-community called “Learn to Cook with Me” and begin posting their cooking attempts. That month, the group launched a light challenge – “Cook three cozy autumn meals this month.” Motivated not to fall behind, the user cooks, shares their progress, and earns recognition from the group for completing the challenge. Encouraged by the feedback, they continue documenting what worked and what didn’t while also picking up ideas from others. Over time, they feel less dependent on takeout, gain confidence in cooking, and gradually replace their unbalanced diet with home-cooked meals.

### Sprint 1:

- **Solution approach 1:** Would people really use the interconnection with the health app? How would we connect to a health app and be able to get the two apps to communicate? To go about developing learning prototypes for this and test hypotheses, we could take further interviews with other young adults and see their opinions or if they would use an app like this, or if they would expect more or less features for something like this. We could also do further research into the Health app functionalities and see if it is even a possibility to sync something like Apple Health with a third-party app. Our approach to testing will be to conduct about 5-10 interviews regarding this approach specifically to garner feedback on whether this is a viable, wanted approach.
- **Solution approach 2:**

- Adoption of Microcommunities: Unknown whether users are motivated to join small, goal-oriented groups rather than relying on existing large platforms (Reddit, Discord, Facebook groups). Hypothesis: People prefer smaller, curated communities where engagement feels more personal and progress-driven.
  - Streaks & Gamification Impact: Unclear whether streaks, passports, and gamification will drive consistent posting or just create short-term novelty. Hypothesis: Light gamification (like a “cooking passport”) will encourage users to continue posting without feeling burdensome.
  - AI Recipe Summaries & Aesthetics: Unknown whether users find value in automatically generated summaries (vs just linking the recipe). Unclear if stylized visuals add lasting appeal or feel like a gimmick. Hypothesis: AI summaries save time and add clarity, while AI-aesthetic features boost shareability and make profiles unique.
  - Differentiation vs. Existing Apps: Unknown whether users see enough value compared to Reddit communities, Discord servers, or Strava-style apps. Hypothesis: The “passport” concept plus goal-based microcommunities provides unique value not offered elsewhere.
- **Solution Approach 3:** The main challenge of Solution 3, pantry management, is that students may find detailed pantry tracking too tedious and time-consuming, particularly if it requires logging every item in precise quantities such as grams. To address this, we aim to identify methods that reduce the burden of scanning and tracking. For testing, we will develop a lightweight prototype that offers different high-level tracking approaches, such as barcode scanning, recipe-based questionnaires, or quick surveys. By asking users to log a small set of items using each method, we can measure the average time required and gather feedback on user preferences. This will help us determine whether students prioritize speed and convenience over accuracy, and ultimately identify the most practical tracking method for minimizing the effort of pantry management.

## Learning Prototype (Results)

**Sprint 5**

**A/B Test Result**

## # Demographic

Total Users: 16

Total Postings: 16

Working Professionals: 5

Students: 11

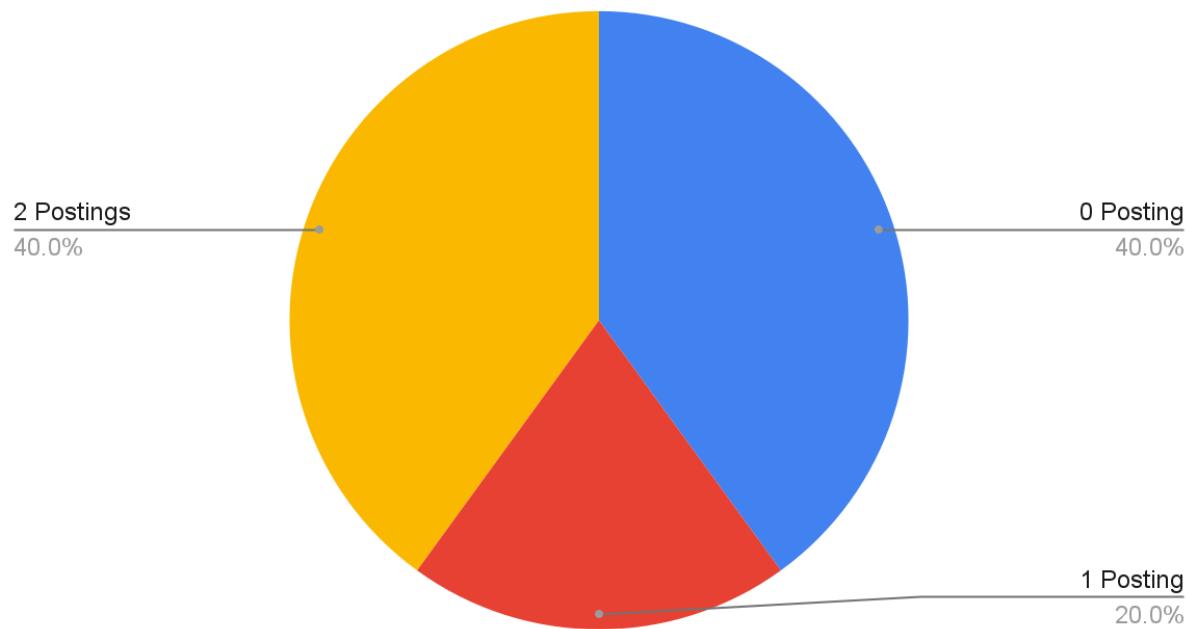
## # Posting Rate (Prof/Stud)

Posting 0: 6 (2/4)

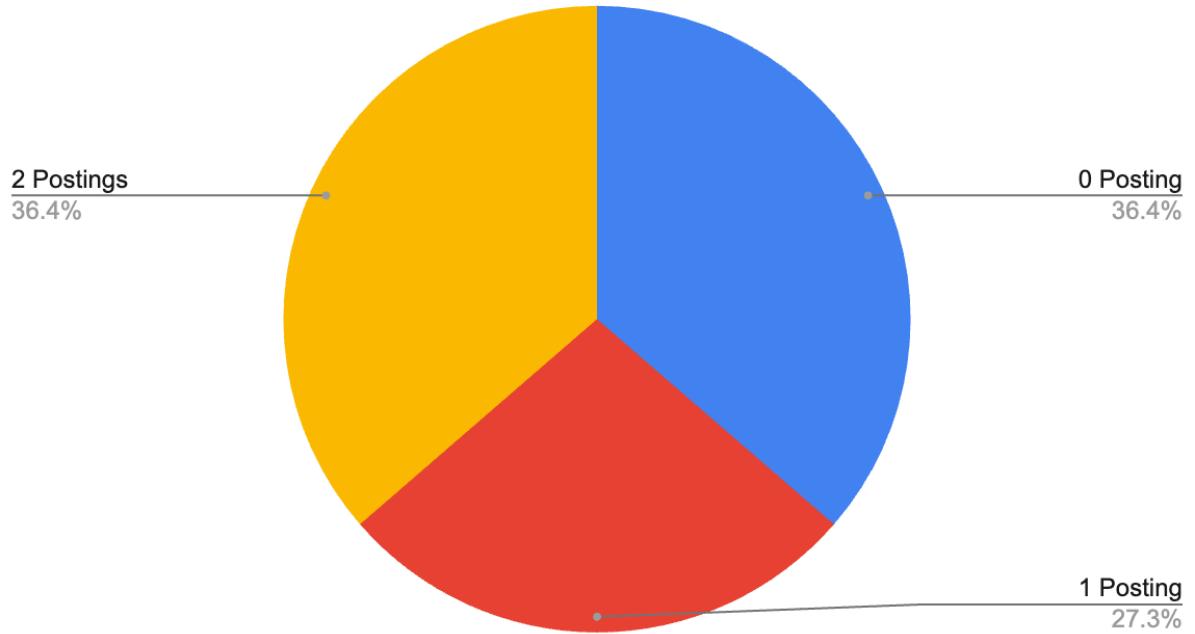
Posting 1: 4 (1/3)

Posting 2: 6 (2/4)

Working Professionals Posting Counts



## Students

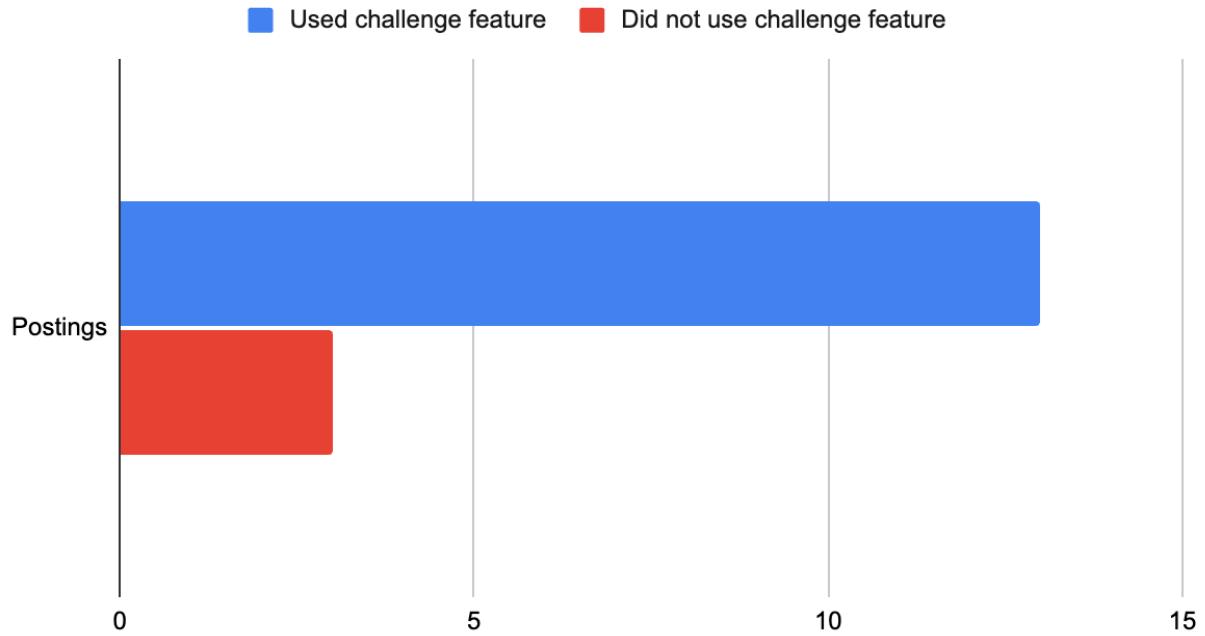


## # Use of the Challenge Feature

Posting from who used challenge feature: 13

Posting from who did not use challenge feature: 3

## Posting Rate based on use of challenge feature



### # AB testing

#### Group A

- 9 users
- 9 postings
- 4 challenge participants

#### Group B: 7 users

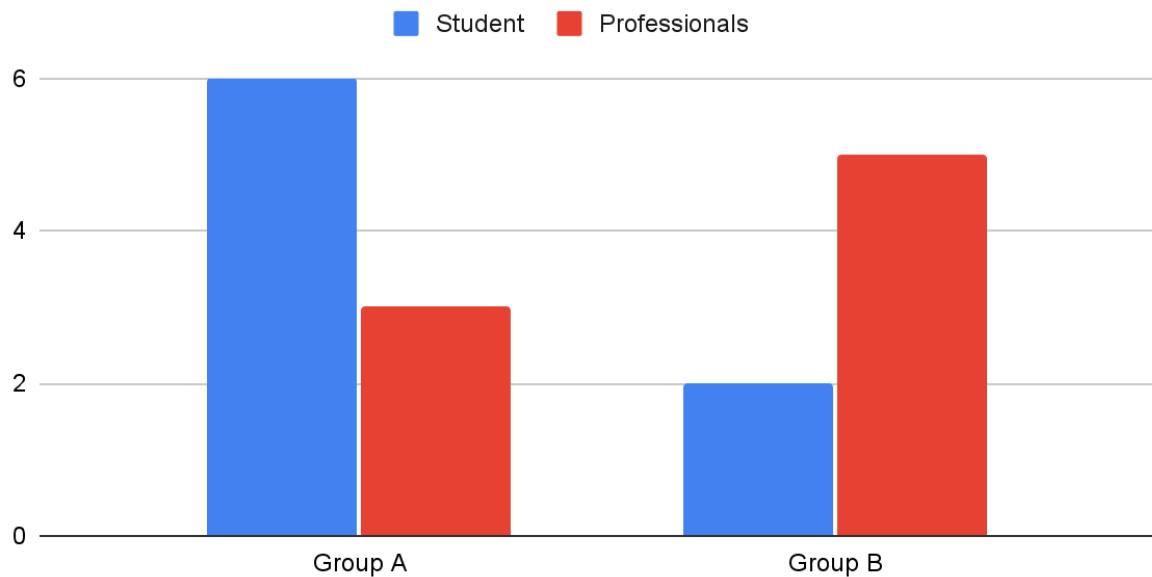
- 7 users
- 7 postings
- 5 challenge participants

### # Posting Rate (A/B vs Prof/Stud)

	Student (Postings)	Working Professionals (Postings)
--	--------------------	----------------------------------

Group A	6	3
Group B	2	5

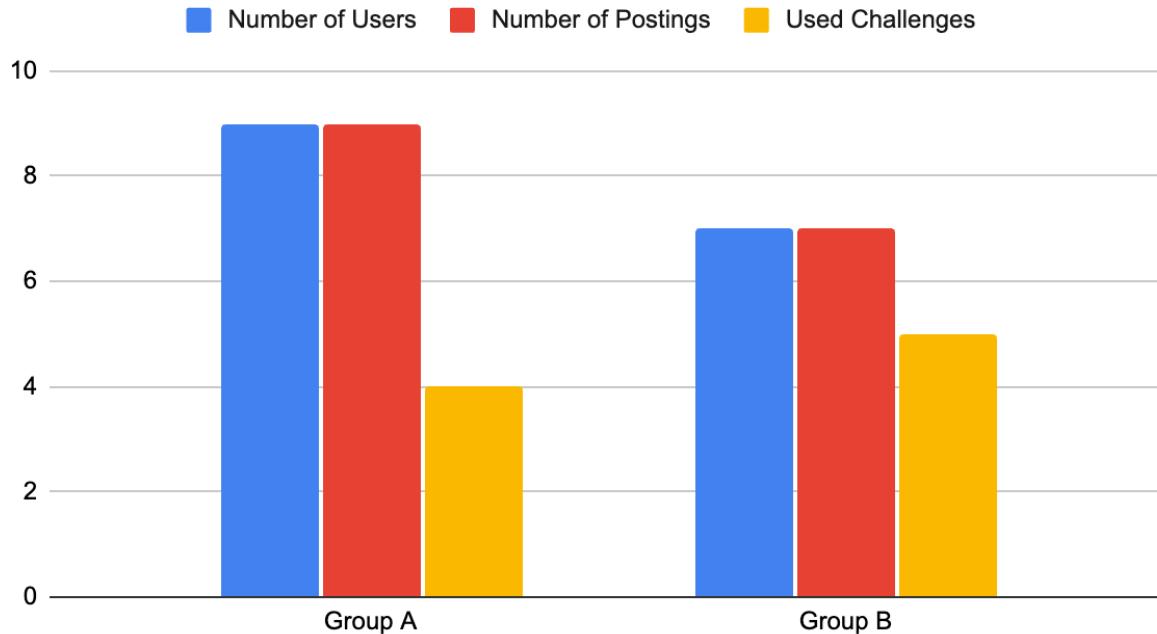
## Posting Engagement Comparison: A/B vs Student/ Professionals



### # Challenges (A/B)

# of completed challenges	Group A (people)	Group B (people)
Completed 0 challenges	5	2
Completed 1 challenges	1	0
Completed 2 challenges	0	1
Completed 3 challenges	1	0
Completed 4 challenges	0	1
Completed 5 challenges	2	2

## Group A vs Group B



### User Feedback Notes

#### User 1 (Student/Group A)

This user shared that posting during Thanksgiving felt especially motivating because she knew others would be active on the app. She could see herself using the app in the future and appreciated being able to set her own goals, noting that she completed all of them and liked how doing so helped her understand her own capabilities and standards. When reflecting on the challenge, she mentioned that it might be more engaging if users could also see other people's goals to make the experience feel more communal.

#### User 2 (Student/Group A)

This user liked that she was able to set her own goals but found it challenging to come up with them in the moment. She also only completed one of the goals she created. She enjoyed the community groups and could imagine using the app in the future, especially once she cooks more regularly since she likes taking photos of food. She suggested that having a list of preset goals to choose from—while still allowing users to add custom goals—would improve the experience. If she could change one thing about the challenge, it would be adding this option to help reduce the difficulty of generating personal goals on the spot.

## Learning Prototype Analysis and Discussion

The personal challenge feature positively influenced the overall increase in user engagement within the app, as users who participated in challenges tended to post more frequently. This suggests that personal goals positively influence user engagement. In the A/B test, Group A used a self-directed goal-setting format, while Group B used a guided, checklist-based challenge format. Differences in challenge format did not significantly affect posting frequency between the two groups. However, user interest in challenges varied, with 44.4% of Group A and 71.4% of Group B participating in at least one challenge, suggesting that users may be more motivated to participate when using the guided challenge format. This preference is likely driven by the structured nature of guided challenges, which reduce cognitive load and clarify expectations, whereas self-directed goals require greater personal planning and commitment.

### Sprint 4:

Firebase Overview: [!\[\]\(d24c3affefeb42bd070edd596d3c9a41\_img.jpg\) Firebase overview](#)

#### A/B Test Result

	Group A (Full Features)	Group B (Posts Only)
Number of Participants	5	5
Total Posts Created	10	6

A total of ten users participated in the study, with five assigned to Group A and five assigned to Group B. Over the 2.5-day trial, Group A, who had access to personal habit-building features such as streaks, goal-setting, and a progress dashboard, submitted 10 posts. Group B, who had access only to the community posting and browsing features, submitted 6 posts.

The difference in posting frequency indicates that the inclusion of personalized progress features encouraged more consistent interaction with the app. Group A users tended to cook and log meals more regularly, likely because the app provided a clear sense of progress and direction. In contrast, Group B users interacted with the app more passively, often browsing content without feeling compelled to contribute regularly.

## Learning Prototype Analysis and Discussion

The findings from this sprint suggest that habit-building features play a meaningful role in sustaining motivation and engagement. Being able to track progress through streaks and

weekly goals gave users a sense of direction and accomplishment, which encouraged them to return and log meals more consistently. These personal accountability features helped make cooking feel like gradual progress rather than isolated events. Our findings from the previous sprint's survey reinforced this pattern as well, with participants noting that seeing their own progress made cooking feel more purposeful and motivated them to continue. Together, the behavioral data and survey feedback indicate that simple self-tracking features are effective in supporting ongoing cooking habits.

### **User Feedback Notes**

#### **User 1 (Group A)**

I conducted a user interview with a master's student who tested the prototype, and overall she found the app intuitive and enjoyable to use. She highlighted that posting was extremely easy and appreciated the clean, minimal interface, noting that it made the app feel like a personal "cooking journal" where she could store photos and track meals without needing separate apps or folders. This personal collection aspect resonated with her, especially as someone trying to get into cooking more consistently. She also liked being able to immediately see updates from other users, which made the experience feel more social and motivating. Her main point of confusion was around the journal prompt — after submitting her reflection, her response disappeared right away, leaving her unsure whether it had been saved or how that feature worked. Despite this, she expressed that the app's simplicity, visual design, and social elements made her more excited to cook and share meals with friends.

#### **User 2 (Group B)**

This user found the app inviting and approachable, appreciating its personable copy and simple, non-intimidating design. She mentioned that seeing other users share what they cooked was motivating and reminded her of how seeing her friend's activity on Strava made her feel like she should run. However, she didn't feel inspired to cook by journaling prompts, stating "I don't like cooking enough to inspire me to journal." She said that she would've preferred the goal-setting feature since she wants to improve at cooking. She also suggested that the community feature could be more transparent when it's implemented and allow users to learn more about each community before joining one.

#### **User 3 (Group B)**

This user described the app as "cool" and liked being able to see what others were eating. She found it inspiring, noting that seeing others' meals gave her ideas for what she could

make in the future. Her comments suggest that she liked the social and visual aspects of the app, which helped spark ideas for her own cooking.

#### User 4 (Group A)

This user felt that the app would encourage her to cook because she enjoyed seeing what others had made as well. The posts from other users were motivating and made her want to try new recipes herself and post as well.

#### User 5 (Group B)

This user saw the app primarily as a social sharing platform and thought it was good for that purpose. However, she said it would encourage her more if it included recipes or ways to connect posts to cooking instructions. She suggested maybe adding some type of reverse image search to find the recipes of the food users had posted. Though recipe finding is out of our domain, this is an interesting consideration for the future, especially on posts where users don't share recipes.

#### User 6 (Group A)

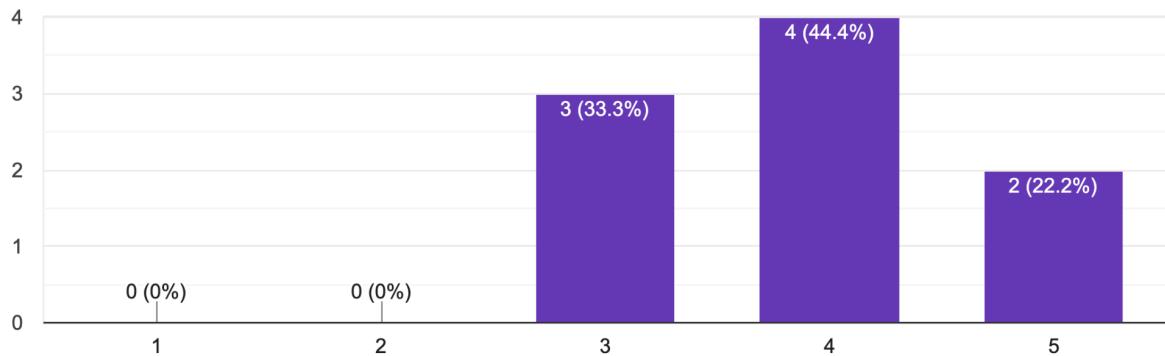
This user suggested adding reminders to help users remember to use the app. This indicates that while she may have liked the concept, she thinks people might forget to engage without some form of gentle prompt or notification. Her feedback highlights the need for reminders in sustaining ongoing participation and habit formation around cooking.

### Sprint 3:

#### Interview Results:

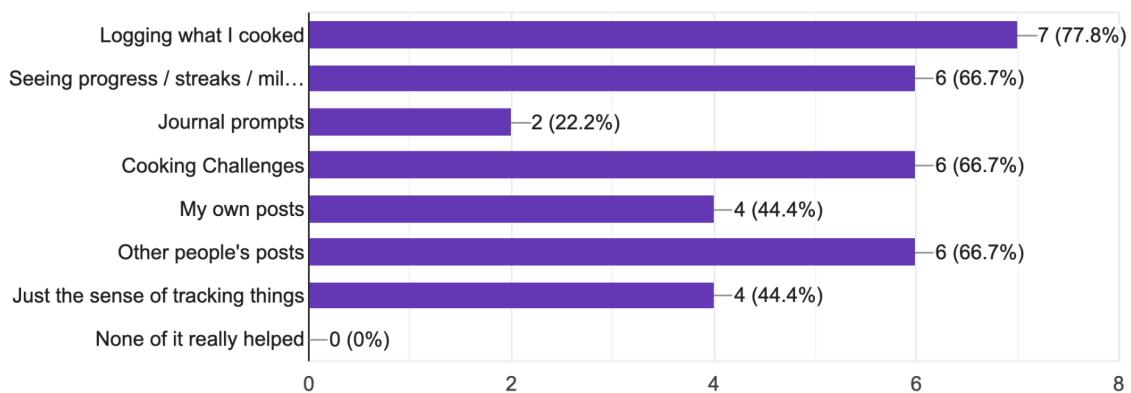
After using Pantry once, how motivated do you feel to cook again in the next few days?

9 responses

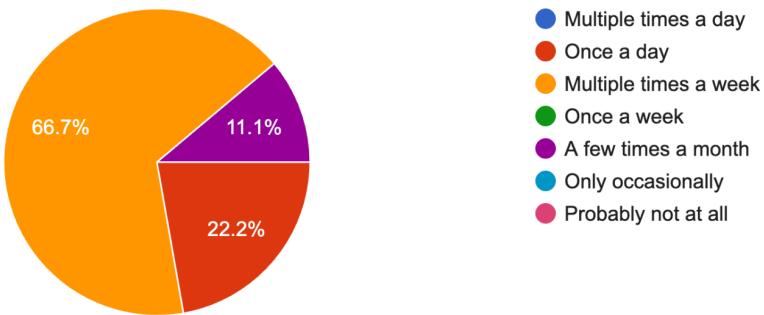


Which parts of the app (if any) helped you feel more motivated? (Select all that apply)

9 responses



Realistically, how often do you see yourself using Pantry if it stayed the way it is now?  
9 responses



Links to the interview data:

[https://docs.google.com/spreadsheets/d/1RpEqOjj7H\\_sP\\_tv14Mk5wFtVd-svx\\_2vS3uNeXurKnY/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1RpEqOjj7H_sP_tv14Mk5wFtVd-svx_2vS3uNeXurKnY/edit?usp=sharing)

## Learning Prototype Analysis and Discussion

Across nine survey responses, participants reported motivation scores ranging from 3 to 5, with an average of about 4, indicating generally strong motivation to cook again after using the app. The most consistently motivating features were logging meals, tracking progress through streaks, and participating in cooking challenges, alongside inspiration from other people's posts. A few participants also highlighted journal prompts and the overall "sense of tracking things" as valuable. When asked what was missing, users pointed to more recipes, notifications or reminders, daily challenges, and more visually engaging ways to view streaks or records; one also raised concern that store-bought food posts might feel discouraging.

In terms of usage frequency, most participants envisioned using the app multiple times a week, with some aiming for daily use and one only a few times a month. For long-term consistency, users emphasized the importance of low-pressure accountability, fun challenges, accessible recipes, subtle or gamified milestones, and small casual groups. While most found group statistics motivating, a minority felt pressured or neutral. Sharing behavior leaned toward occasional or casual posting, though some expressed interest in becoming more active if the community remained casual and supportive.

Sprint 2:

Through structured testing of our three approaches, we found that the micro-community-centered approach was strongly preferred by users, while the Duolingo and AI coach approaches were eliminated due to engagement and trust issues.

The results we found from our learning prototypes are written below: first with the key findings followed by a detailed overview of user interviews and insights regarding the different solution mockups.

### **Key Findings by Approach**

#### **1. Duolingo for Cooking**

Through interviews and testing a brief mockup, we found that users did not see much value in another app modeled so closely after Duolingo. Feedback revealed several flaws: participants reported low engagement, stress from not being able to choose what to cook, and general disinterest in using the application. While some appreciated the gamified elements, the level-based structure felt restrictive, leaving them frustrated about limited choice. Users also struggled to understand how cooking difficulty mapped to app levels and questioned how quest completion would be verified.

Key quotes:

- "Just looking at it, I'm not motivated... they just shove the recipe in your face."
- "The application would stress me out, as I wouldn't be able to choose what I wanted to cook."

Decision: Eliminated due to lack of personal engagement and procedural confusion that hindered user adoption.

#### **2. AI Coach**

Interviews on the AI coach mockup revealed that users were generally unwilling to trust AI for recipe suggestions or creation. Participants preferred recipes from real people and valued user input over automated guidance. While some thought an AI coach could be a cool feature, they felt it would not provide the same motivation as a community. Additionally, the high development cost of an AI assistant made this approach less feasible given the limited perceived value. Although AI integration remains a potential future option, we decided this approach is not the best path forward. Participants did appreciate aspects like a personalized interface and

progress tracking, but concerns about AI's inability to understand taste and cooking nuance outweighed these benefits.

Key Quotes:

- "AI has no idea what the food tastes like."
- "I don't really care for what AI has to say... if I had a human coach, I would care more."

Decision: Eliminated due to widespread AI distrust for cooking guidance and insufficient personal connection for accountability, combined with high development costs.

### 3. Micro-Communities

User Feedback Summary: Participants responded very positively to this approach, particularly appreciating the "friends' activity" feature and ability to create custom communities. Users expressed strong accountability motivation and liked the balance between social features and individual tracking.

Key Quotes:

- "I'd hate to be the one who hasn't cooked" when viewing group progress.
- "This would be a fun way to document an aspect of my life that I don't usually document."

Decision: Selected for continued development based on strong user engagement, clear accountability mechanisms, and positive feedback that aligned with our research on community-driven motivation.

### Detailed Interview Insights

#### Mockup Interviews Summary

1. I interviewed a senior in college who lives with roommates but is often out of her apartment and finds it difficult to cook often. She does like to cook but has a hard time doing so when she comes back late or after a long day. In this interview, I learned that for the Duolingo for Cooking solution approach, she did not feel a level of personal engagement with the app. She agreed the gamification was fun, and finding new recipes seemed easy enough, but the actual app did not entice her to

cook more, as she said she was not likely to use it as much. For the AI coach approach, she likes the personalization but again said the app seemed not to have a level of engagement, making it difficult for her to be held accountable. She liked the AI coach but did not trust its suggestions, especially for recipes. She preferred trying new recipes created by real people rather than AI suggestions. For the micro-community approach, she really liked the UI and the engagement aspect. She also liked that the app had a personal function through its badges, challenges, and streaks. She thought the simplified focus made much more sense and was consequently much more engaging to use. She also specifically pointed out the aspect of creating your own community, as she liked the idea of creating one for friend circles. Critiques she had in this prototype were to include a mobile app, as this would be used more, and filtering out access to your posts (friends-only, private, or public), as she likes the idea of posting in the community but may not want everyone in the community to view these posts. Of all three prototypes, she preferred the micro-community approach and really loved the simplistic, engaging UI. The relevance here is that we see users like her in college prefer an engaging, personalized app for cooking to increase motivation and accountability. We see that micro-communities engage individuals as well as gain feedback on details of features that students may want to see. We substantiate our approach here and clarify our solution further.

2. In this interview, I walked the participant through the solutions in the following order: 3 (micro-communities), 1 (Duolingo for cooking), and then 2 (AI coach). For the micro-communities mockup, the interviewee stated that she liked the “friends’ activity” feature, citing the ability to know what her friends are doing, her enjoyment of tracking activity, and the possibility of getting inspiration from her friends as the reasons for this. She also liked the challenges feature (though she did have some questions about prizes and accountability) and the weekly check-in discussion. Additionally, she said that she’d “hate to be the one who hasn’t cooked” when looking at the group progress bar, suggesting that contributing to a group’s success would be a good motivator for her completing a meal. When we moved to the Duolingo for cooking solutions, the interviewee immediately said that she liked the first one better, saying, “Just looking at it, I’m not motivated,” and “[In the other one] you could see what other people were doing; this one, I don’t know, they just shove the recipe in your face.” She stated that the application would stress her out, as she wouldn’t be able to choose what she wanted to cook, stating that maybe it would be better if there were a list of options rather than just one thing you had to cook before you could move on to the next level. Finally, the interview examined the AI coach solution, and she actually really liked this approach, saying that the

options and understanding of a recipe before having to prepare it make it much better than the Duolingo for cooking approach. She also mentioned that she likes how it tracks your progress and that there are different levels. She was concerned about the process of coaching, however, saying that if it were to use a camera to track her movements, she would immediately delete the app; if it were to require her to type in her questions, she would also not use it; but that she would use it if she could talk back and forth with it through voice commands. When asked to pick between the AI coach and micro-communities options, she chose the AI coach because she felt she would likely get faster results. However, she did say that she usually likes when people have made and tried a recipe, as AI has no idea what the food tastes like, suggesting that she does not trust the AI to have a good sense of flavor. While this interviewee clearly values speed over accuracy (even saying she could “deal with the gross cooking if it comes to that”), her responses still align with our domain research and other interviews—other people are a much greater motivator than doing things on your own, and people do not trust AI to make recipes for them.

3. In this interview, I walked the participant through the solutions in the following order: 2 (AI coach), 1 (Duolingo for cooking), and then 3 (micro-communities). This interviewee was not a huge fan of the AI coach approach, stating that she would prefer an actual person coaching her because it would allow for genuine connection and that if the application is not free, it would have to be significantly cheaper than an actual nutritionist because of that fact. She also did not understand what differentiated it from other AI coaching, explaining that one of her issues with it was that the concept felt pretty basic. Most of her questions centered around how she would be motivated and why he would want to use it. The interviewee expresses that she doesn't really care for what AI has to say; she can see that it is suggesting a recipe, but she doesn't care. On the other hand, if she had a human coach, she would care more, saying that she would “want to do good for my teacher” and that she just doesn't feel the same way about AI. The interviewee felt the same way about the Duolingo for cooking approach, saying that she actually doesn't like Duolingo because the streaks don't do anything for her. There is nothing in the approach to keep her attention, as she faces no real consequences if she decides to not cook. Finally, even before I introduced the micro-communities approach, the interviewee suggested that a social aspect to the solution would motivate her. When I showed her our approach, she said that she would “enjoy this,” citing it as a “less cliquey version of Belli.” She emphasized that she likes the more social aspects (of anything), as it makes her happy to keep up with other people. She stated that the solution would be a fun way to document an aspect of her life that

she doesn't usually document and that she likes the challenges because she likes being dared to do things. Again, this interview suggests that a social, community-based solution would be the best motivator for cooking, and less personal options just do not have the same incentive pull as connection.

### Mockup Interview Insights

1. For the Duolingo-like approach to address cooking motivations, interviews revealed that users feel a lack of personal engagement with this approach. Though they like the gamified aspect, they do not feel as engaged as they did with the community approach and have mentioned how this approach would not give them as much accountability as they would want in an app addressing this issue. The interviewees also had some procedural issues with it, suggesting in their responses that when they see an application such as this, they associate the levels with difficulty in cooking rather than the different types/styles of cooking we were trying to convey. They could not understand how one would progress through the levels or how this type of application would work with food. Additionally, there were questions about how the application would know that you completed the level and what types of recipes were in each level. Do you just have to do what it says? What kinds of foods does it cater to? Because of the strong negative feedback we got from this approach, both at the solution level and the feature level, we decided that we would not continue with this approach.
2. Through our interviews for the AI coach approach, we found that users lack trust in AI, a massive factor in determining whether this approach would work to solve the problem at hand. Users do not trust AI to suggest or create recipes for them and thus would not trust it to coach them through the process of cooking. Though users like the interface, they state a lack of engagement due to issues interacting with a 'robot.' They mention the need for genuine connection in caring about coaching feedback and seem to prefer real people and recipes created by individuals rather than recipes from AI tools, even ones configured to suggest online recipes. There was one outlier who preferred this approach to the others, though this was due solely to the speed of response rather than personal engagement. In fact, this participant still agreed that they preferred human-made recipes, as AI could have no insights on how the meal would taste. Due to these results and the expense of creating an AI coach, we have moved away from this approach as a solution to the problem. With the distrust and lack of personal connection to the AI interfering with motivation and accountability, we realized it would not be a valuable investment to develop this further.

3. The results of our interviews for the micro-communities approach revealed a lot of positive feedback and valuable insight as to how to move forward. Interviewees liked the layout, the individual personalization, and the option to join different communities based on different goals. Specifically, people loved the option to create custom communities amongst friends and the option to have friends on the app. They mentioned enjoying keeping up with friends and stated that the features in the application are great for that. Additionally, the fact that they would be responsible for contributing to community goals added additional motivation to complete challenges present in the application. We did get some feedback for improvements, such as having notifications for when friends post, notifications when new recipes are released, having a store for saved recipes, and having the option to make posts/communities private or public. These, among others, will be discussed further in the following section. Overall, reactions to this solution were quite positive and reinforced the idea that community engagement would lead to increased motivation and accountability in cooking for users. We see that this approach was most favored among our interviewees and resulted in feedback that clarified rather than negated the approach. This also further substantiates the result of our competitive analysis and domain research, all of which provided a strong basis for a community-centered platform to increase motivation and accountability.

### **Clickable Prototype Feedback for Micro-communities**

We have discussed the feedback we got about our three refined solutions. This feedback and our domain research leads us to continue with the micro-communities approach. In interviews where we walked through the prototype with the interviewees, participants stated that they enjoy friends' activity tracking, challenges, communities, weekly check-ins, and easy access to shared recipes. They also love the social inspiration aspect, stating that seeing what others cook motivates them to try it as well. They also explain that they think the app could be great for content creators, niche and/or cultural communities, and planning/group participation.

On the downside, users expressed the want for clarification of certain features. For instance, one user misunderstood the "Browse by Category" feature, thinking it referred to recipes rather than communities and implying that we should maybe let users search for or define their own categories. Some wanted to understand how challenges work (how do people prove they actually completed the challenge? are there prizes associated with this?) and suggested a built-in rewards system with proof and annual giveaways. One user wondered if answers to prompts were public, stating they would like if people could interact and respond to what they said. Another suggested that we have the option to

make all posts/communities private or public. Users also didn't understand whether or not they could save other people's recipes, suggesting that we add a storage feature for saved recipes. Additionally, there was some confusion with the stats, with users wanting badge descriptions so that they could know what they earned it for and not immediately understanding the stats on the profile page ("I cooked 47 meals this month??").

In concordance with keeping up with friends, users wanted notifications for when friends post or for when new recipes are released. They also talked about the option to go live with cooking and perhaps having "creator communities," where creators from other platforms can put more accessible versions of their recipes and monetize their content. However, this suggestion deviates slightly from the aim of this solution.

Overall, users thought this was a great solution but suggested some clarifications and added details to improve user experiences.

## Overall Conclusions

The testing validated our pivot toward micro-communities as the primary solution. Unlike the eliminated approaches, micro-communities received feedback focused on refinement rather than fundamental concerns, indicating strong product-market fit for our target users. The community-driven accountability model addresses the core behavioral challenge we identified while maintaining individual value for less social users.

## Solution Deep Dive

### Use Cases

Sprint 2:

#### 1. Persona 1: Emily, the Busy Student

- Background: 20 years old, lives with roommates, has limited time and budget for cooking.
- Goal: Wants to eat healthier and save money, but struggles with accountability.
- Interaction with App:
  - Joins a "30-Minute Dinner Gang" micro-community.
  - Posts quick dinner photos (low-effort participation).
  - Gets motivated by friendly comparisons – e.g., seeing her roommate also cooked twice this week.
  - Tracks her own streak of 5 home-cooked meals, celebrating milestones like "First Week of Home Dinners."

- Value: Emily gets light accountability and encouragement to keep cooking, without feeling overwhelmed by recipes or meal-prep logistics.

## 2. Persona 2: Maria, the Independent Cook

- Background: 35 years old, single mom, limited time for social apps, mostly cooks for her kids.
- Goal: Wants convenience, occasional inspiration, and a sense of progress, but not heavy social engagement.
- Interaction with App:
  - Uses a personal dashboard to log meals with photos and notes.
  - Hits milestones like “10 family dinners cooked.”
  - Receives daily lightweight prompts (“Here’s a 20-min comfort recipe for busy nights”).
  - Occasionally browses community recipes, but doesn’t actively post.
- Value: Maria gets solo value from the app. She feels rewarded for her cooking efforts even without joining communities.

## 3. Persona 3: David, the Aspiring Home Chef

- Background: 28 years old, works 9–5, enjoys cooking but struggles with consistency.
- Goal: Wants to improve his skills, experiment with recipes, and build a routine.
- Interaction with App:
  - Joins a themed community like “Beginner Bakers” or “Indian Cooking Explorers.”
  - Participates in monthly community challenges (e.g., “Try 2 new veggies this month”).
  - Uses the personal journaling feature to track his growth and document recipes he’s tried.
  - Feels part of a shared narrative when his community streak is maintained (“90% of us cooked this week!”).
- Value: David stays consistent because he feels supported by a group with shared goals, while also building a personal portfolio of meals.

## 4. Persona 4: Stephen, the Father of Young Children

- Background: 32 years old, has 2 young kids who are 6 and 2, has limited time to cook and his children are very picky eaters
- Goal: Wants to ensure that his kids are eating a healthy balanced diet and is looking for inspiration within their eating preferences.

- Interaction with App:
  - Joins a “Parents with picky eaters” micro-community.
  - Starts a new discussion thread detailing what his kids will and will not eat and asks for advice on what he could cook for them.
  - Motivated by a monthly challenge to introduce one new food into his kids’ diet.
  - Learns tips and tricks about different ways to balance their diets (e.g. snacks)
- Value: Stephen gets new ideas for foods that will both meet his kids’ dietary needs, and they enjoy. He also gets inspiration for adding new foods into their diet.

## 5. Persona 5: Izzy, the Beginner

- Background: 18 years old, just moved out and into her first apartment, is an aspiring chef
- Goal: Wants to learn how to cook but is unsure where to start. Ultimately, wants to become a world-renowned chef but understands she is at the beginning of her journey.
- Interaction with App:
  - Creates a “Learn to Cook with Me” micro-community.
  - Posts consistently about the foods she created and the exact steps in her process.
  - Gets feedback on her process and ideas about what she can do next time.
  - Also gives advice to others based on the things she’s learned in making new dishes.
- Value: Izzy now has a community of aspiring chefs helping and motivating each other to keep going in their journeys. Is also able to solicit feedback in a more casual (and less expensive) setting than culinary school.

## 6. Persona 6: Carlos, the Corporate Adult

- Background: 28 years old, single adult working a demanding job, but who is very picky and has very little time to cook.
- Goal: Inspiration to eat healthier meals with much less overwhelming effort and accountability to cook rather than order food.
- Interaction with App:
  - Joins a “Adults with Little Time” micro-community.
  - Shares his picky habits and asks for advice from the community

- Gets motivated through his badges and the monthly challenges like “adding a new veggie this month”
  - Gets various recipes from community members on easy, quick meals and healthy snack ideas
- Value: Carlos gains low-effort and easy-to-make inspiration tailored for his situation, with support from community members. The community helps his struggles and helps him try new ideas that would work for his specific situation.

## 7. Persona 7: Lily, the Culinary Student

- Background: 21 years old, enrolled in a hospitality and cooking program at community college. Passionate about both cooking and teaching, she wants opportunities to practice instructional skills beyond the classroom.
- Goal: To gain more practice outside formal coursework and experiment with different teaching methods in a supportive, low-stakes environment.
- Interaction with App:
  - Creates a “Cooking 101: Easy Meals” micro-community.
  - Posts step-by-step guides and short videos of her cooking process.
  - Answers questions from beginners and offers personalized feedback.
- Value: Builds confidence and practical experience as an instructor, while learning what types of explanations and approaches resonate most with learners.

Sprint 1:

- **Solution Approach 1 use cases:**
  1. A post-graduate working a corporate job connects the app to their Fitbit. Every time they log a home-cooked meal by taking a photo, the calories and nutrition sync with their fitness tracker. Cooking would contribute to their “healthy habits” ring, and they would get a notification celebrating their cooking, which would reinforce the motivation to cook.
  2. A college student struggles with eating out too much and sets a goal to cook 3 dinners a week in their app. Every time they prepare a meal and take a photo of it, their streak grows and their badges and rewards show up in their normal health app with any other exercise achievements. This keeps them motivated to continue cooking.
  3. A friend group in post-graduate school use the app to cook separately but share their progress with each other to keep each other motivated to cook at home. When one friend logs their meal, it appears in the shared streak

through a connected dashboard, encouraging the others to cook as well. This creates a community with cooking and encourages cooking as an everyday practice.

- **Solution Approach 2 Use Cases:**

1. A college student joins a “budget cooking” group to find easy, affordable recipes. They share their weekly meal-prep photos, earn streaks for consistency, and use their Cooking Passport as a fun record of how they’ve improved over the semester.
2. A young professional focused on fitness joins a “high-protein meals” group, where they discover new recipes tailored to their dietary needs. By attaching links, they get AI-generated summaries with nutritional info and substitution ideas, saving them time.
3. A group of friends creates their own private microcommunity, where they upload dishes they’ve made, react to each other’s Cooking Passports, and run polls like “Which dish should we try together next week?”—turning cooking into a fun, shared routine instead of an isolated chore.

- **Solution Approach 3 Use Cases:**

1. A college student on a tight budget often forgets what they already have and overspends on groceries. The Pantry Management Assistant helps by tracking pantry items, generating low-cost shopping lists, and suggesting affordable alternatives. This allows the student to stretch their budget while still maintaining balanced meals.
2. A graduate student with a busy schedule struggles to plan meals and often defaults to quick but unhealthy options. The Pantry Management Assistant suggests recipes based on ingredients already in their pantry and automatically creates organized shopping lists for missing items. This saves the student time while making it easier to eat healthier and more varied meals.
3. A student with a packed schedule wants to avoid daily cooking stress and last-minute grocery runs. The Pantry Management Assistant reviews their pantry, suggests recipes that maximize existing ingredients, and generates a shopping list for the rest. This enables the student to meal prep for the week efficiently, saving both time and money.

## Technical Discussion

### Sprint 5:

In Sprint 5, we expanded our technical architecture beyond the initial habit-tracking experiment to support a more mature, mobile-ready system capable of handling notifications, richer community structures, expanded badges, and a second-platform admin dashboard. With the transition to a fully deployed mobile app, we implemented Expo-based push notifications, enabling real-time streak reminders, log prompts, community alerts, and milestone updates. On the frontend, our React Native interface now includes the dedicated Logging tab, redesigned Profile pages with expanded badge and milestone components, friend connections, and a more structured community experience with goal visibility, progress indicators, and interaction histories. We also introduced admin-facing screens as part of our second-platform approach, allowing administrators to view aggregated activity metrics, streak distributions, community participation levels, and high-level engagement trends within a lightweight mobile dashboard. Conditional routing remains central to our testing: while A/B experiments are still supported, Sprint 5 routing now also determines access to notifications, community goals, and platform-level features based on user group and role.

On the backend, Firestore's data model was significantly expanded to support logging events, notification tokens, badge tiers, profile metadata, friend relationships, and admin analytics collections. We optimized our schema to reduce nesting and avoid relational overload, following feedback that complex subcollection trees in Firebase can become difficult to maintain. New Cloud Functions now handle push notification scheduling, interaction logging, updated streak and badge evaluation logic, and admin analytics aggregation. We also strengthened our progress pipeline: each check-in now triggers a chain of functions that update streaks, refresh milestone status, evaluate badge criteria, and write anonymized behavioral metrics for retention and notification-response analysis. Additional collections were introduced for community goals, challenge assignments, and logging-history snapshots, enabling the more advanced community interactions built this sprint. Pantry Admin is a separate dashboard application that uses the same Firebase APIs as the main app but is logically separate from the backend. It lets us both monitor user activity and experiment results by filtering by both user type (students vs. working professionals) and user groups in A/B testing (group A vs. group B).

We also expanded our REST/HTTPS Function layer, adding endpoints for mobile deployment and advanced interactions:

- POST /api/v1/logs.create – Creates log entries (meals, journals), updates streak progression, and records behavior analytics.
- POST /api/v1/notifications.register – Registers push tokens for device-level reminders.
- POST /api/v1/notifications.send – Sends scheduled or triggered notifications (streak alerts, encouragement, community updates).
- GET /api/v1/admin.metrics – Returns aggregate streak, logging, and activity trends for admin dashboard users.
- POST /api/v1/friends.add – Adds users as friends and initializes shared activity tracking.
- GET /api/v1/community.goals – Returns current goals and progress across each micro-community.
- POST /api/v1/challenges.progress – Updates multi-step challenge progress for individual users or groups.
- GET /api/v1/logs.history – Retrieves longitudinal logging data for user dashboards.

All endpoints use JSON payloads, follow consistent authentication rules via Firebase Auth, and rely on Cloud Functions for secure, server-side evaluation. The mobile app now supports timezone detection, push token handling, offline caching for logs, and media uploads for cooking posts.

Looking ahead, future plans would center on scaling and stabilizing the systems introduced in this sprint. We plan to build multi-tiered badge progressions, implement adaptive notification intelligence (e.g., streak-rescue alerts, personalized reminders, weekend cooking nudges), and deepen community features with shared milestones, rotating weekly challenges, and friend-group cooking goals. We will also continue stress-testing Firestore's scalability and evaluate whether certain modules, such as analytics, challenge logic, or feed ranking, should migrate to a more structured database layer. Finally, we intend to expand the admin dashboard into a more powerful behavioral insights tool with automated reports, participation heatmaps, and segmentation by group type. Sprint 5 laid the foundation for a two-sided ecosystem; the next iterations will focus on optimizing reliability, reducing technical debt, and strengthening the motivational architecture revealed through our longitudinal testing.

**Link to Code Repository:** <https://github.com/navyagupta2804/First-Prototype>

**Link to 2nd Platform (Admin Dashboard) Code Repository:**  
<https://github.com/navyagupta2804/pantry-admin1.git>

## Sprint 4:

- In this sprint, we began validating individual habit-tracking and running an A/B experiment that tests how much streaks, goals, badges, and milestones actually influence consistency. Group A sees the full motivational layer (streaks, goals, milestones, badges, community challenges), while Group B sees a stripped-down journaling-only version. On the frontend, our React Native app now incorporates expanded Profile views, badge and milestone displays, richer goal-tracking modules, A/B routing logic that conditionally hides motivational elements from Group B, and more developed community and challenge interfaces. On the backend, we extended our Firebase architecture to support variant assignment, new habit-tracking logic, community memberships, and richer progress models. Firebase Authentication handles user identity, Firestore stores all core entities (users, variants, check-ins, streaks, badges, milestones, goals, communities, challenges), Firebase Storage manages media, and Cloud Functions process streak computation, badge evaluation, and experiment logging. We also implemented new REST/HTTPS endpoints including:
  - **POST /api/v1/users.initialize** (initializes users and assigns variants)
  - **GET /api/v1/users.variant** (returns “A” or “B” for UI routing)
  - **POST /api/v1/checkins.create** (logs cooking activity, triggers streak/badge updates, records experiment data)
  - **POST /api/v1/streaks.recompute** (timezone-aware streak recalculation)
  - **POST /api/v1/goals.update** (updates goal progress for Group A users)
  - **POST /api/v1/badges.evaluate** (modular badge awarding)
  - **GET /api/v1/milestones.list** (returns milestone states)
  - **POST /api/v1/communities.join** (adds a user to a community and initializes challenge progress)
  - **POST /api/v1/challenges.update** (updates challenge task completion)
  - **GET /api/v1/community.feed** (returns community-specific check-ins).

All endpoints use JSON payloads and are implemented as Firebase HTTPS Functions, with on-device context such as timezone and camera access supporting accurate check-ins.

Looking ahead, our future planning centers on deeper versions of the systems introduced here. We plan to expand badges into multi-tiered progressions, build notification systems for streak rescues, recaps, and challenge reminders as well as making goals more customizable and community-driven. We intend to enhance community features with shared milestones, rotating challenges, and richer social accountability structures. While this sprint implemented variant assignment, streaks, goals, milestones, badges, and community/challenge foundations, the next iterations will focus on scaling these systems,

automating feedback loops, and refining the motivational architecture revealed by our A/B experiment.

**Link to Code Repository:** <https://github.com/navyagupta2804/First-Prototype>

Sprint 3:

We built a mobile app using React Native for the frontend and Firebase for the backend, focused on implementing the core flows needed to support cooking habit formation. The frontend currently includes a Home tab with journal prompts and recent check-ins, a Post interface for uploading photos and captions, and a Profile tab displaying streaks and basic user stats. The design prioritizes a smooth, low-friction user experience, allowing users to quickly log their cooking activity without extensive setup. This initial implementation provides the groundwork for testing whether lightweight personal tracking features can effectively drive motivation and accountability.

On the backend, we integrated Firebase's services to power real-time interaction and habit-tracking logic. Firebase Authentication manages user sign-in, Cloud Firestore stores structured data for users, posts, check-ins, and streaks, Firebase Storage handles image uploads, and Cloud Functions run background jobs such as streak calculations. We also defined a set of REST / HTTPS function endpoints that connect the frontend to the backend which include:

- POST /api/v1/users.initialize initializes a user profile after authentication.
- POST /api/v1/posts.create accepts { uid, caption, imagePath }, validates and stores the post, and uploads the image to Firebase Storage.
- POST /api/v1/checkins.create accepts { uid, note?, photoPath?, cookedAt? } to log a check-in and trigger streak recomputation.
- POST /api/v1/streaks.recompute recalculates daily and weekly streaks in the user's time zone based on recent check-ins, triggered both by scheduled jobs and on-demand.

Our data storage and processing rely on Firestore (users, posts, check-ins, streaks), Storage (image uploads), and Cloud Functions (streak recomputation and validation), while on-device context like the camera and timezone ensures accurate and user-friendly interactions.

Looking ahead, future sprints will extend this foundation to include communities and memberships (community creation, joining, and group feeds), challenges and milestones (structured prompts and goals), and notifications (automated nudges and weekly recaps). We also plan to add advanced moderation and analytics to support scaling and creator

features. While this sprint focused on users, posts, check-ins, and streaks, these planned additions will enable richer social accountability and deeper engagement.

**Link to Code Repository:** <https://github.com/navyagupta2804/First-Prototype>

Sprint 2:

Technical solutions that we could use to complete our application include a range of different interconnected platforms.

Our first option is more Firebase-focused. The client would be created via React Native. The client is where users can post or check in (with features such as photos and captions), view/join micro-communities, and complete milestones and prompts. These will be implemented via React Native, allowing for both Android and Apple development. In the backend, we would use Firebase Authentication for logging in and saving user information, Firestore for realtime and group check-ins, Cloud Functions for computations like streak calculations and challenge scoring, Firebase Storage for images, and FCM for push. This works as it incorporates real-time counters, lightweight interactions, and a low difficulty level. We also all have experience in this platform. The cloud functions will host the streak jobs, scoring challenges, community stats, and moderation checks. In Firestore, we will have the users, communities, membership, check-ins, milestones, challenges, and prompts. In storage we will host image uploads and CDN through Firebase.

Another option we have is Supabase and Postgres. In this case, the client would again be React Native, but the backend would be Supabase Auth, Postgres, and row-level security, realtime channels, storage, and edge functions for jobs. In this case, we would have great relational modeling through SQL with very flexible relations. However, the syncing will be harder to implement in these platforms, and real-time functions may lag. Features will be implemented in the client the same way as the first option. The edge functions will hold streak computations, challenge results, and image moderation. The database will hold tables for communities, memberships, check-ins, reactions, and milestones. The storage will have user photos. This option, at the moment, does not seem too ideal for our platform.

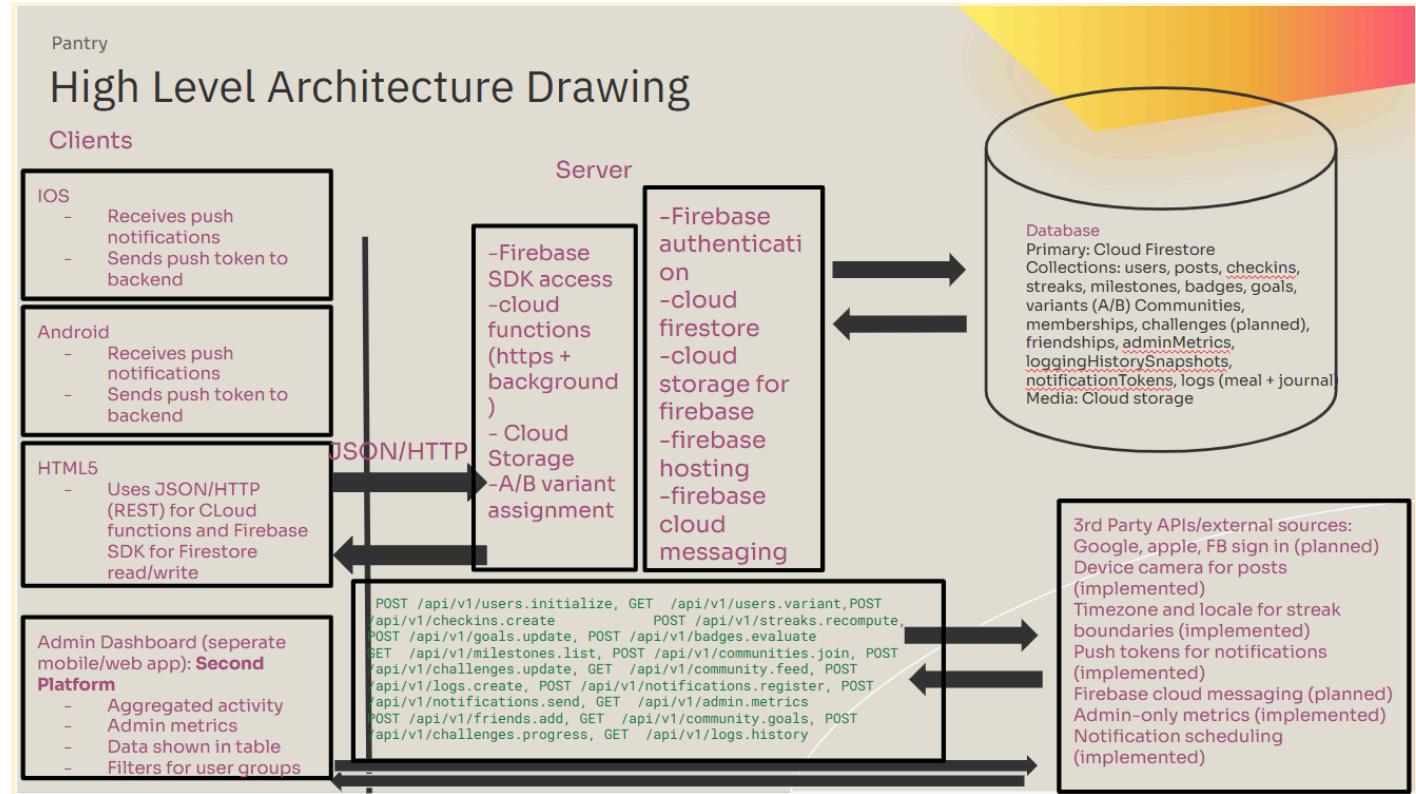
A third option would be AWS serverless, which would include Cognito, AppSync, DynamoDB, S3, and Lambda. This is a cloud-focused solution, which would be highly scalable and easy to control as well as hold a lot of data. However, it would be very slow due to a large overhead and much more code to write for counters, real-time check-ins, and photo uploads. Features will be implemented in React Native for the client as with the

previous two options. AppSync would be real-time subscriptions for community feeds, and Lambda would be used for streaks, analytics, and moderation checks. S3 would be used for media with a CloudFront hosting CDN. SNS would hold push notifications.

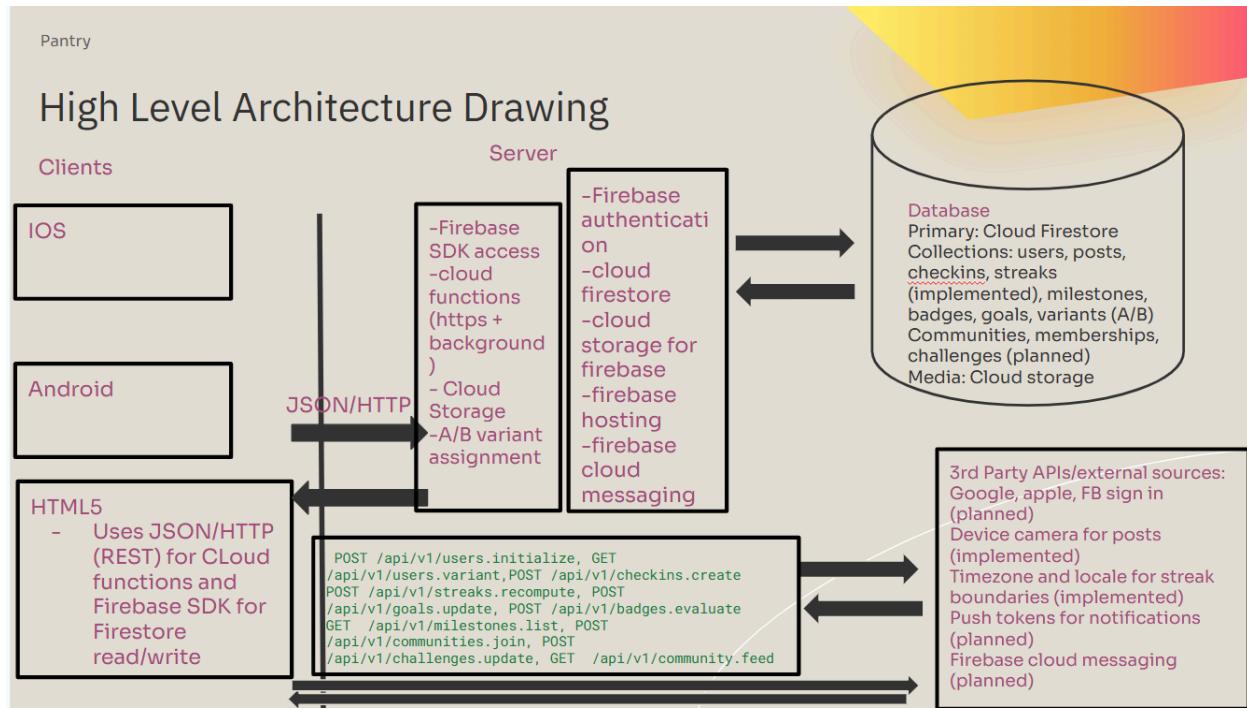
Initially, we think the Firebase option would be the best way to implement the app.

## High Level Architecture Drawing

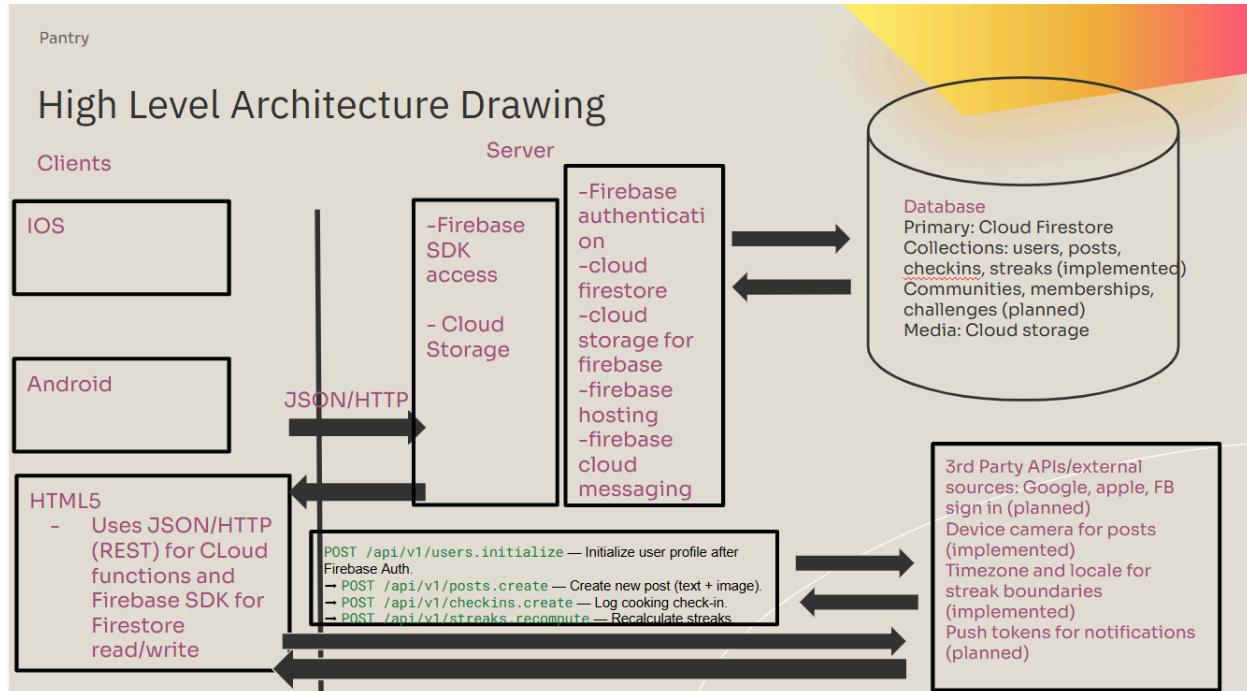
Sprint 5:



Sprint 4:

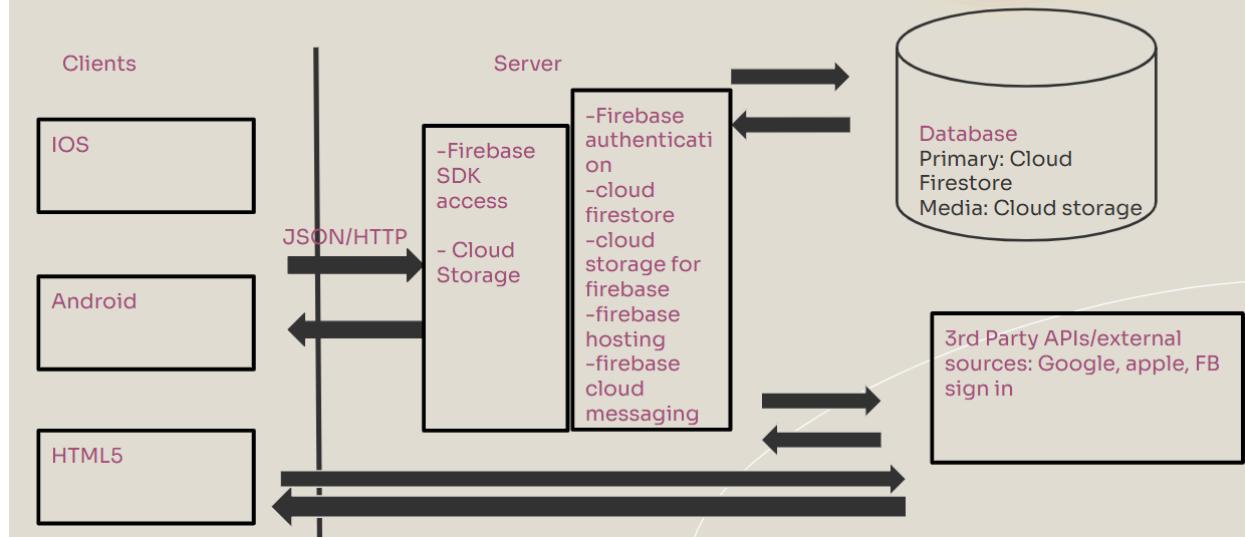


Sprint 3:



Sprint 2:

# High Level Architecture Drawing



# Value Proposition Canvas

Sprint 5:

<b>Key Partners</b>		<b>Key Activities</b>		<b>Value Proposition</b>		<b>Customer Relationships</b>		<b>Customer Segments</b>	
<ul style="list-style-type: none"> <li>• Recipe/Food Creators – Provide fresh, quality content to inspire community activity (possibly through sponsored challenges)</li> <li>• Payment Processors – Handling premium fees</li> <li>• Partnerships for Native Ads/Sponsorships (Grocery chains, Meal Prep services like HelloFresh)</li> </ul>		<ul style="list-style-type: none"> <li>• Moderating groups/communities</li> <li>• Platform maintenance &amp; development (UX, mobile, performance) <ul style="list-style-type: none"> <li>◦ Current – Building the React Native app, A/B routing logic, Firebase backend implementation (streaks, goals, community feed)</li> <li>◦ Future – Building more robust features like refreshed journal prompts, badges for goal completion, and sophisticated community functionality (discussion threads and community goals)</li> </ul> </li> <li>• Marketing &amp; user acquisition</li> <li>• Customer support &amp; retention</li> <li>• Data analysis for tracking user engagement</li> <li>• Admin Dashboard development <ul style="list-style-type: none"> <li>◦ Future – More robust data analytics</li> </ul> </li> <li>• Partner/Sponsor Acquisition and Management</li> </ul>		<p><b>Community Users:</b></p> <p>Transforms cooking from a solo chore into a lightweight, social habit through themed micro-communities, gentle accountability, and immediate social reinforcement (reactions/comments).</p> <p><b>Solo Users:</b></p> <p>Provides immediate value and habit-building tools (streaks, milestones, logging) without pressure.</p> <p><b>Future MVP Plans:</b> The core feed is implemented, but future priority is to ensure robust community features (community goals, discussion threads).</p> <p><b>Learning Prototype (Current):</b> Streaks, goals, milestones, and personal dashboards implemented and validated as driving more engagement.</p> <p><b>All Users:</b></p> <p>Enables passive browsing for inspiration and saving ideas for later.</p> <p><b>Learning Prototype (Current):</b> Streaks, goals, milestones, and personal dashboards implemented and validated as driving more engagement.</p> <p><b>Rewards effort over perfection;</b> provides consistency and belonging without stress. Solves the core behavioral challenge of motivation and accountability.</p> <p><b>Learning Prototype (Current):</b> Implementing nudge/notification logic to increase cooking frequency without feeling intrusive</p>		<ul style="list-style-type: none"> <li>• Self-managed community – users manage their own interactions and provide peer encouragement</li> <li>• Automated encouragement (through journal prompt, community goals, lightweight notifications, etc.)</li> <li>• Customer support (online communication channels)</li> <li>• Personalized &amp; Adaptive Nudges (notifications)</li> <li>• Personalized Goal Tracking (guided weekly goals approach)</li> </ul>		<p><b>The Unmotivated Cook:</b></p> <p>Struggling to cook consistently due to lack of accountability, not lack of recipes (core target).</p> <p><b>Solo Habit Builders:</b></p> <p>Users focused purely on personal growth (streaks, milestones) who prefer a private dashboard.</p> <p><b>Micro-Community Seekers:</b></p> <p>Users who thrive on peer motivation, positive feedback, and small, themed social groups.</p>	
<b>Key Resources</b>						<b>Channels</b>			
<ul style="list-style-type: none"> <li>• Reliable cloud storage for retrieving user-generated content <ul style="list-style-type: none"> <li>◦ Current – User check-in/post data, streaks, badges, milestones, and goals data (used for A/B testing)</li> </ul> </li> <li>• Technology <ul style="list-style-type: none"> <li>◦ Current – React Native mobile front-end, Firebase architecture (Auth, Firestore for data, Storage for media)</li> <li>◦ Future – Potential migration to a more structured backend to handle complex relational community data</li> </ul> </li> <li>• Engaged user base – active users in micro-communities and consistent solo users</li> <li>• Established Partner Network (Future) – for native ads</li> <li>• User Data – for personalization and Admin Insights</li> </ul>						<ul style="list-style-type: none"> <li>• Mobile app stores</li> <li>• Social media marketing</li> <li>• Content marketing – sharing cooking tips and habit-building insights from public communities, linking back to the app</li> <li>• Word of mouth – groups of friends join the app together to have a private friend community</li> </ul>			
<b>Cost Structure</b>						<b>Revenue Streams</b>			
<ul style="list-style-type: none"> <li>• Product development</li> <li>• Technology development (server hosting, cloud hosting and storage)</li> <li>• Marketing &amp; customer acquisition</li> </ul>						<ul style="list-style-type: none"> <li>• Free</li> <li>• Subscription – ad-free experience, personalized goal tracking</li> <li>• In-app advertising &amp; sponsorship – sponsored recipes, in-feed product recommendations, ads that will actually be helpful (like for grocery stores)</li> </ul>			

Sprint 4:

<b>Key Partners</b> 	<b>Key Activities</b> 	<b>Value Proposition</b> 	<b>Customer Relationships</b> 	<b>Customer Segments</b> 
	<ul style="list-style-type: none"> <li>• Recipe/Food Creators – Provide fresh, quality content to inspire community activity (possibly through sponsored challenges)</li> <li>• Payment Processors – Handling subscription fees</li> </ul>	<ul style="list-style-type: none"> <li>Moderating groups/communities</li> <li>Platform maintenance &amp; development (UX, mobile, personalization)               <ul style="list-style-type: none"> <li>• Current – Building the React Native app, A/B routing logic, Firebase backend implementation (users, auth, community feed)</li> <li>• Future – Building more robust features like refreshed journal prompts, badges for goal completion, and sophisticated community functionality</li> </ul> </li> <li>Marketing &amp; user acquisition</li> <li>Customer support &amp; retention</li> <li>Data analysis for tracking user engagement</li> </ul>	<p><b>Community Users:</b></p> <p>Transforms cooking from a solo chore into a lightweight, social habit through themed micro-communities, gentle accountability, and immediate social reinforcement (reactions/comments).</p> <p><b>Future MVP Plans:</b> The core feed is implemented, but future priority is to ensure robust community features (shared content, community goals).</p> <p><b>Solo Users:</b></p> <p>Provides immediate value and habit-building tools (streaks, milestones, logging) without pressure.</p> <p><b>Learning Prototype (Current):</b> Streaks, goals, milestones, and personal dashboards implemented and validated as driving more engagement.</p> <p><b>All Users:</b></p> <p>Rewards effort over perfection; provides consistency and belonging without stress. Solves the core behavioral challenge of motivation and accountability.</p> <p><b>Future MVP Plans:</b> Implementing nudge/notification logic to increase cooking frequency without feeling intrusive</p>	<ul style="list-style-type: none"> <li>• Self-managed community – users manage their own interactions and provide peer encouragement</li> <li>• Automated encouragement (through journal prompt, community goals, lightweight notifications, etc.)</li> <li>• Customer support (online communication channels)</li> </ul>
<b>Key Resources</b> 			<b>Channels</b> 	
	<ul style="list-style-type: none"> <li>Reliable cloud storage for retrieving user-generated content               <ul style="list-style-type: none"> <li>◦ Current – User check-in/post data, streaks, badges, milestones, and goals data (used for A/B testing)</li> </ul> </li> <li>Technology               <ul style="list-style-type: none"> <li>◦ Current – React Native mobile front-end, Firebase architecture (Auth, Firestore for data, Storage for media)</li> <li>◦ Future – Potential migration to a more structured backend to handle complex relational community data</li> </ul> </li> <li>Engaged user base – active users in micro-communities and consistent solo users</li> </ul>	<p><b>Community Users:</b></p> <p>Transform cooking from a solo chore into a lightweight, social habit through themed micro-communities, gentle accountability, and immediate social reinforcement (reactions/comments).</p> <p><b>Solo Users:</b></p> <p>Provides immediate value and habit-building tools (streaks, milestones, logging) without pressure. Enables passive browsing for inspiration and saving ideas for later.</p> <p><b>All Users:</b></p> <p>Rewards effort over perfection; provides consistency and belonging without stress. Solves the core behavioral challenge of motivation and accountability.</p>	<ul style="list-style-type: none"> <li>• Mobile app stores</li> <li>• Social media marketing</li> <li>• Content marketing – sharing cooking tips and habit-building insights from public communities, linking back to the app</li> <li>• Word of mouth – groups of friends join the app together to have a private friend community</li> </ul>	
<b>Cost Structure</b> 	<b>Revenue Streams</b> 			
<ul style="list-style-type: none"> <li>• Product development</li> <li>• Technology development (server hosting, cloud hosting and storage)</li> <li>• Marketing &amp; customer acquisition</li> </ul>	<ul style="list-style-type: none"> <li>• Free</li> <li>• Subscription – pay for more premium features like advanced analytics or no ads</li> <li>• In-app advertising &amp; sponsorship</li> </ul>			

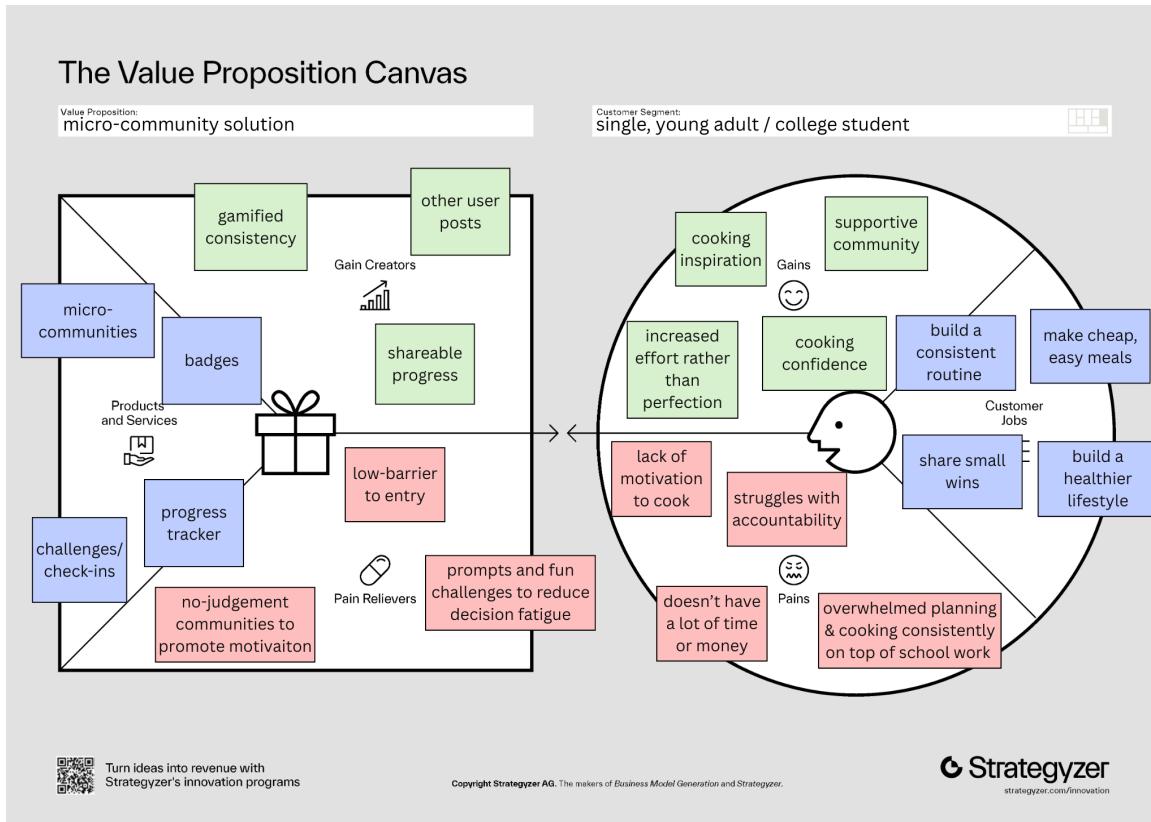
## Sprint 3:

<b>Key Partners</b> 	<b>Key Activities</b> 	<b>Value Proposition</b> 	<b>Customer Relationships</b> 	<b>Customer Segments</b> 
	<ul style="list-style-type: none"> <li>• Recipe/Food Creators – Provide fresh, quality content to inspire community activity (possibly through sponsored challenges)</li> <li>• Payment Processors – Handling subscription fees</li> </ul>	<p><b>Community Users:</b></p> <p>Transform cooking from a solo chore into a lightweight, social habit through themed micro-communities, gentle accountability, and immediate social reinforcement (reactions/comments).</p> <p><b>Solo Users:</b></p> <p>Provides immediate value and habit-building tools (streaks, milestones, logging) without pressure. Enables passive browsing for inspiration and saving ideas for later.</p> <p><b>All Users:</b></p> <p>Rewards effort over perfection; provides consistency and belonging without stress. Solves the core behavioral challenge of motivation and accountability.</p>	<ul style="list-style-type: none"> <li>• Self-managed community – users manage their own interactions</li> <li>• Automated encouragement (through journal prompt, community goals, etc.)</li> <li>• Customer support (online communication channels)</li> </ul>	<p><b>The Unmotivated Cook:</b></p> <p>Struggling to cook consistently due to lack of accountability, not lack of recipes (core target).</p> <p><b>Solo Habit Builders:</b></p> <p>Users focused purely on personal growth (streaks, milestones) who prefer a private dashboard.</p> <p><b>Micro-Community Seekers:</b></p> <p>Users who thrive on peer motivation, positive feedback, and small, themed social groups.</p>
<b>Key Resources</b> 			<b>Channels</b> 	
	<ul style="list-style-type: none"> <li>Photo, challenge, &amp; journal infrastructure</li> <li>Community infrastructure</li> <li>Reliable cloud storage for retrieving user-generated content</li> <li>Technology (app, website, streaks, goals)</li> <li>Engaged user base – active users in micro-communities and consistent solo users</li> </ul>	<p><b>Community Users:</b></p> <p>Transform cooking from a solo chore into a lightweight, social habit through themed micro-communities, gentle accountability, and immediate social reinforcement (reactions/comments).</p> <p><b>Solo Users:</b></p> <p>Provides immediate value and habit-building tools (streaks, milestones, logging) without pressure. Enables passive browsing for inspiration and saving ideas for later.</p> <p><b>All Users:</b></p> <p>Rewards effort over perfection; provides consistency and belonging without stress. Solves the core behavioral challenge of motivation and accountability.</p>	<ul style="list-style-type: none"> <li>• Mobile app stores</li> <li>• Social media marketing</li> <li>• Content marketing – sharing cooking tips and habit-building insights from public communities, linking back to the app</li> <li>• Word of mouth – groups of friends join the app together to have a private friend community</li> </ul>	
<b>Cost Structure</b> 	<b>Revenue Streams</b> 			
<ul style="list-style-type: none"> <li>• Product development</li> <li>• Technology development (server hosting, cloud storage)</li> <li>• Marketing &amp; customer acquisition</li> </ul>	<ul style="list-style-type: none"> <li>• Free</li> <li>• Subscription – pay for more premium features like advanced analytics or no ads</li> <li>• In-app advertising &amp; sponsorship</li> </ul>			

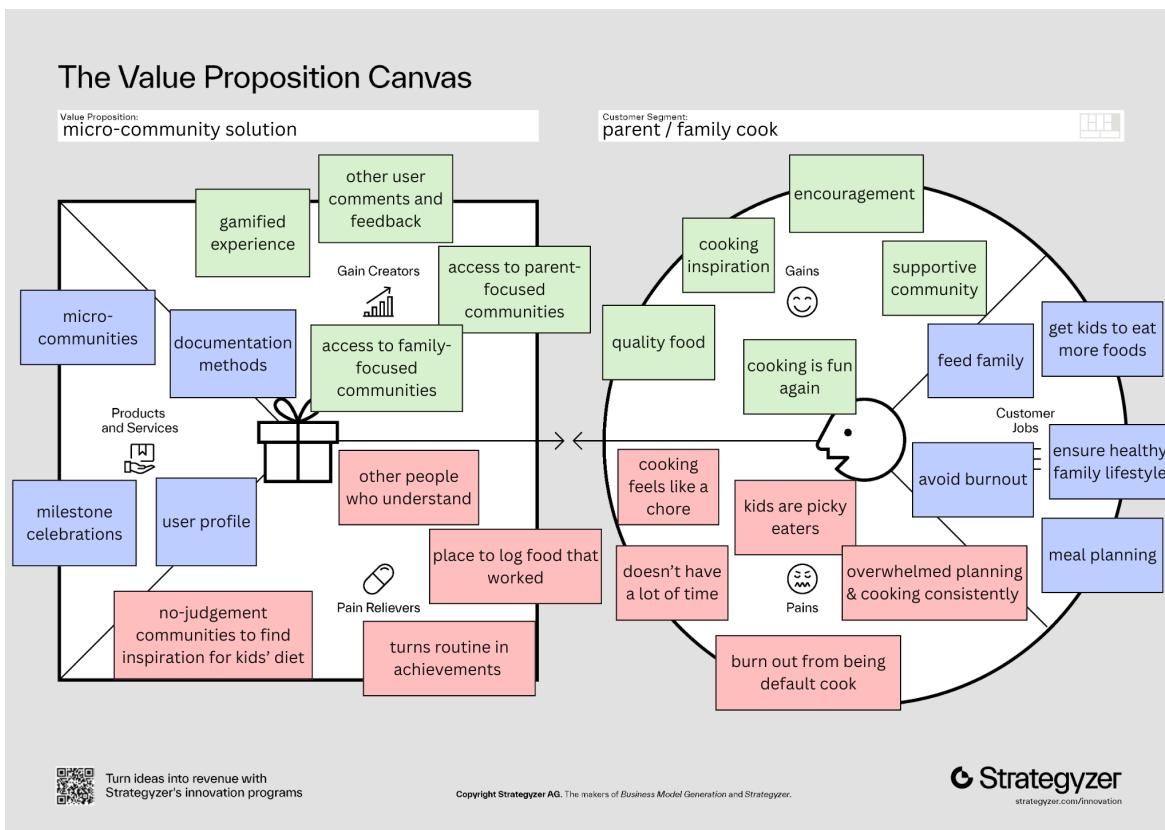
In our last sprint, we focused on different types of users who were looking for cooking communities for motivations. However, through user testing and feedback, we recognized that there are many users who want less interaction with communities and more solo experiences. Additionally, those who did want more interactions with the communities still really valued the solo interactions. Because of this, the BMC for this sprint adds a “Solo Habit Builder” customer segment who prefers a more private dashboard.

Moreover, after Wednesday’s class discussion, we thought more about how we would get revenue streams to offset the cost of product development, server hosting, cloud storage, and marketing. Our initial plan is to have a subscription service for more premium features, in-app advertising (also so we can advertise elsewhere too), and perhaps some sponsorship opportunities.

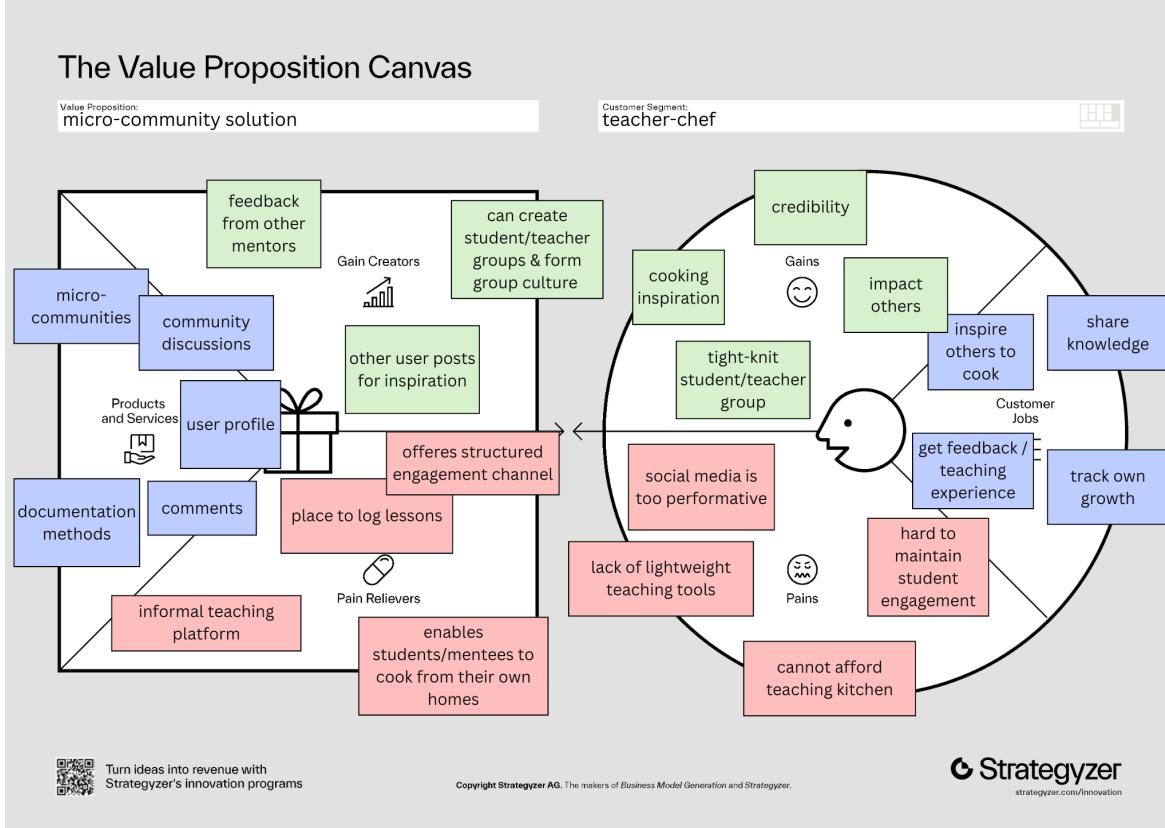
## Sprint 2:



## The Value Proposition Canvas



## The Value Proposition Canvas



## Feature Analysis

### Sprint 5:

#### Micro-Communities

- Progress: Implemented
- Users Testing and Feedback:
  - Users showed strong motivation to post when they were able to view others' cooking activities. Social visibility was identified as a key factor encouraging participation.

#### Milestones & Streaks

- Progress:
  - Streaks: implemented
  - Milestones: Implemented
- Users Testing and Feedback:
  - Users who actively maintained streaks demonstrated higher posting frequency. The presence of visible progress indicators supported continued engagement.

#### Journal Entries

- Progress: partially implemented → Need larger question sets
- User Testing and Feedback:
  - Users utilized the journal feature consistently, but some reported limited variety in prompts, indicating the need for expanded question sets to sustain long-term use.

#### Personal Dashboard

- Progress: implemented
- User Testing and Feedback:
  - Users expressed appreciation for being able to track streaks, journals, and milestones in a single view, though overall feedback volume was limited.

## Personal Challenges

- Progress: Implemented
- User Testing and Feedback :
  - Challenge participation was strongly correlated with higher posting activity. Group B users, who used self-directed challenges, showed higher challenge participation (71.4%) compared to Group A (44.4%).

## Community Challenges

- Priority: HIGH
- Progress: Future implementation
- User Testing and Feedback :
  - Users expressed interest in participating in shared, community-wide challenges, particularly in relation to seasonal or event-based activities.

## Community Feed with Reactions & Comments

- Progress: implemented
- User Testing and Feedback :
  - Users frequently interacted with posts through reactions and comments, and social interaction was identified as one of the major drivers of engagement.

## Nudges & Notifications

- Progress: Implemented
- User Testing and Feedback:
  - Users indicated that reminder notifications would be helpful in maintaining consistent app usage and posting behavior.

Sprint 4:

## Micro-Communities

- Priority: HIGH

- Progress: future implementation (next learning prototype)
- Users Testing and Feedback:
  - There's a strong pattern across multiple users saying that seeing what others cooked lead them to feeling inspired or motivated to cook and that leads them to wanting to post themselves
  - Overall, our 10 users posted a total of 16 times in 2.5 days, suggesting that the social aspect of the app is motivating

## Milestones & Streaks

- Priority: HIGH
- Progress:
  - Streaks: implemented
  - Milestones: in progress → badges are yet to be fully implemented and a variety of different milestones also needs to be developed
- Users Testing and Feedback:
  - Group A, which was the group that had access to all the features, posted more than group B, suggesting that personal interactions further encourage users to use the app/cook more

## Journal Entries

- Progress: implemented → needs to be stored somewhere
- User Testing and Feedback:
  - There were 15 journal entries made in the 2.5 days,
  - However, there was confusion around the prompt staying up when you enter your response and users entries not showing up anywhere to be seen later

## Personal Dashboard

- Progress: implemented
- User Testing and Feedback: not much feedback on this particular feature, users did like seeing their streaks and journals and milestones as was mentioned above

## **Community Challenges**

- Priority: HIGH
- Progress: future implementation (next learning prototype)
- User Testing and Feedback : again, users really liked the community and milestone feature

## **Personal Challenges**

- Progress: implemented → could add different types of personal challenges in the future
- User Testing and Feedback :
  - Group A, which was the group that had access to all the features, posted more than group B, suggesting that personal interactions further encourage users to use the app/cook more
  - Users in Group B also later stated that they would have preferred to have personal goals as well

## **Community Feed with Reactions & Comments**

- Progress: implemented
- User Testing and Feedback :
  - There were 16 posts made by users in the 2.5 days
  - Users state that they like being able to interact with the content they saw

## **Nudges & Notifications**

- Priority: HIGH
- Progress: future implementation (next learning prototype)
- User Testing and Feedback:
  - Notifications and reminders came up as a suggestion from users so that people do not forget to use the app

Sprint 3:

## **Micro-Communities**

- Progress: future implementation (next learning prototype)
- Users Testing and Feedback:
  - Users state that friends being on the app or small casual groups would keep them consistent long-term
  - Users wanted to interact with the different posts that they saw and positive peer pressure
  - 55% of users said that they'd enjoy occasional group accountability if it's low-pressure

## Milestones & Streaks

- Progress:
  - Streaks: implemented
  - Milestones: in progress
- Users Testing and Feedback:
  - 66.7% of users said that seeing streaks/progress helped to motivate them
  - Users were excited about actually being able to complete milestones and earn badges

## Journal Entries

- Progress: implemented
- User Testing and Feedback:
  - User testing actually showed a pretty low reaction to Journal prompt, with only 22.2% of users saying that it helped them feel more motivated

## Personal Dashboard

- Progress: implemented
- User Testing and Feedback: not much feedback on this particular feature, users did like seeing their streaks and journals and milestones as was mentioned above

## Community Challenges

- Progress: future implementation (next learning prototype)

- User Testing and Feedback : again, users wanted positive peer pressure to “catch up on streaks or share the pictures”

## Personal Challenges (new)

- Progress: in-progress
- User Testing and Feedback :
  - 66.7% of users said that the cooking challenges helped them to feel more motivated
  - Users want personal motivators, not just community goals

## Community Feed with Reactions & Comments

- Progress: implemented
- User Testing and Feedback :
  - Users like logging what they cooked (77.8%) and seeing other people's post (66.7%)
  - Users wanted to interact more with the posts they saw

## Nudges & Notifications

- Progress: future implementation (next learning prototype)
- User Testing and Feedback:
  - Notifications and reminders came up multiple times as a suggestion from users

Sprint 2:

## Feature: Micro-Communities

- Description: Small, themed groups (e.g., “30-Minute Dinners,” “Parents with Picky Eaters”) designed for accountability and belonging. Unlike large forums, they are lightweight, personal, and goal-oriented.
- Persona / Use Cases:
  - Emily: Joins a quick-dinner group to stay motivated with peers.
  - David: Joins “Beginner Bakers” to practice new skills.
  - Stephen: Share experiences and get advice in “Parents with Picky Eaters.”
  - Izzy: Creates “Learn to Cook with Me” for beginners.
  - Lily: Starts “Cooking 101: Easy Meals” to practice teaching.
- Priority: Must Have

### **Feature: Milestones & Streaks**

- Description: Positive reinforcement via streaks and milestones (e.g., “5 home dinners”) that celebrate progress without punishing lapses.
- Persona / Use Cases:
  - Emily: Celebrates a “First Week of Home Dinners.”
  - David: Tracks consistency in experimenting with recipes.
  - Maria: Reaches “10 family dinners cooked” with pride.
  - Carlos: Motivated by badges for adding healthier meals.
  - Izzy: Builds confidence as her streak grows.
- Priority: Must Have

### **Feature: Personal Dashboard**

- Description: A private space to log meals, view streaks, and celebrate milestones. Provides solo value for those less interested in social posting.
- Persona / Use Cases:
  - Maria: Uses it as her main way of tracking family meals.
  - David: Journals progress and documents recipes.
  - Izzy: Stores feedback on what to improve for future attempts.
  - Michael-type users (future): Can value it for personal recipe archiving.
- Priority: Must Have

### **Feature: Community Challenges**

- Description: Weekly or monthly prompts (e.g., “Try 2 new veggies this month,” “Cook twice this week”).
- Personas:

- David: Motivated by skill-based challenges.
- Stephen & Carlos: Engage in “Introduce a new food for kids this month.”
- Izzy: Gains structure and encouragement from themed beginner challenges.
- Priority: Good-to-Have

### Feature: Community Feed with Reactions & Comments

- Description: A group feed where members can share their check-ins and interact with each other's posts. Users can react with quick gestures like “Like” or “Cheer,” and leave short comments for encouragement.
- Personas / Use Cases:
  - Emily: Feels motivated when roommates and peers cheer her posts.
  - David: Receives supportive comments when he tries a new recipe.
  - Izzy: Gets constructive feedback on her beginner cooking process.
  - Lily: Shares mini-lessons and engages with learners through comments.
- Priority: Good-to-Have

### Feature: Nudges & Notifications

- Description: Configurable reminders and weekly recaps (e.g., “You cooked twice this week!”).
- Personas:
  - Emily: Gets nudged if she hasn't cooked in a few days.
  - Maria: Weekly summary without heavy social pressure.
  - Carlos: Reminded about ongoing challenges.
- Priority: Good-to-Have

## Learning Prototype Plans

### Sprint 5:

Following the results of our Sprint 5 testing, our next phase will focus on scaling Pantry beyond core habit-building features and moving toward platform sustainability, broader engagement mechanisms, and deeper behavioral reinforcement. With notifications, enhanced logging tools, micro-communities, and badges now validated across a longer testing window, we are ready to examine how these features can support more advanced interactions and more diverse user journeys. One of the major priorities ahead is exploring monetization strategies that feel aligned with user needs rather than disruptive. Early user feedback raised questions about premium features, optional boosts for streak recovery, and the potential value of an ad-supported free tier. To test these ideas safely,

we plan to introduce lightweight experiments on non-intrusive ad placement and tiered pricing models, measuring whether users remain engaged and whether these additions enhance or compromise the overall experience.

A second area of expansion involves building out a more robust challenges system that integrates seamlessly with both individual streaks and community structures. Sprint 5 testing showed that users responded positively to timed prompts and gentle accountability, so our next step is designing multi-step challenges with detailed progress milestones, optional collaborative goals, and adaptive notifications that adjust to user behavior over time. These advanced challenges will help us evaluate whether deeper gamification, such as weekly cooking quests, ingredient-based goals, or streak-recovery bonuses, strengthens long-term retention across different user segments. In parallel, we plan to refine the notification system to be more context-aware, experimenting with dynamic triggers based on patterns such as previous logging times, streak vulnerability, or group activity spikes.

Finally, with the introduction of the admin dashboard in Sprint 5, we are entering a stage where Pantry can operate as a two-sided platform. Our next steps will focus on expanding administrator capabilities to support larger communities such as clubs, residence halls, fitness groups, or workplace wellness programs. This includes testing features like scheduled group challenges, community-wide announcements, analytics reports, and optional incentive structures. These refinements will help us evaluate whether Pantry can scale from a personal cooking-motivation tool into a broader habit-forming ecosystem that supports individuals, friend groups, and organized communities. As these new layers roll out, we will continue using a mixed-methods evaluation strategy, pairing behavioral analytics with qualitative interviews, to assess how monetization, expanded challenges, and platform-level tools influence engagement, satisfaction, and long-term cooking behavior.

#### Sprint 4:

For our next learning prototype, we again need to measure user engagement and consistency over a longer period of time. In Sprint 4 our results show that users are motivated to use the app (and in turn cook) when interacting with its social features. However, also having access to the personal interactions provided even more motivation and users in this group ended up posting more in the days of the test. However, this was done over only a couple of days, and we know that habits take time to build. Additionally, it is possible that the results are not as accurate as they could be due to the application only being available on the browser and users forgetting to use it. Thus, what we need to know next is if notifications, more robust functionality, and a longer time to get familiar with the

app would affect use activity. Moreover, much of the functionality of the MVP is very basic. There are no refreshed journal prompts, badges for goal completion, or other motivational feedback loops. Community features are also pretty limited, with likes and comments being implemented, but there is no way for the user to know if someone had interacted with it without going to the post directly.

Thus, a key priority for Sprint 5 is to ensure more robust features are built out so that engagement with the app on a daily basis doesn't get stale. Additionally, it is important that we deploy the application to mobile so that we can implement mobile notifications and nudges. This way, we can rule out forgetfulness as a reason for low usage.

In addition to well-developed individual features, we really want to put a strong focus on micro-communities. We are interested in how small-group accountability influences cooking motivation and engagement. Thus, improving cross-user interactivity is important for this stage. This includes allowing users to add friends and see other people's profile pages. Additionally, we want to implement community goals to see if a shared sense of responsibility will further motivate users to cook. Finally, peer feedback from Sprint 3 suggested comparing user engagement over different types of groups. So, our next learning prototype will aim to test if user-defined groups (group A) or groups predefined by the app (group B) garner more engagement.

Again, we intend to gather feedback through a more longitudinal study, where we hope to give users (individuals and groups of friends) the chance to use the application for about a week or so. From there, we will either conduct follow-up interviews or send out a follow-up survey to get users' thoughts. Additionally, we plan to track engagement analytics through the applications.

### Sprint 3:

For our next learning prototype, we need to measure user engagement and consistency over a longer period of time. In the feedback we received from this sprint, users projected they would likely use the application regularly. However, this reflects human sentiment, not human behavior. What we really need to know is if someone were to use the app for a full week, would this sentiment be the same? Do these features genuinely keep users engaged over the long term? Or do they simply spark initial excitement?

To answer this, a key priority for the next sprint is to ensure that all solo-interaction features are fully functional and easy to engage with on a daily basis. This includes refreshed journal prompts, badges for goal completion, and other motivational feedback

loops. With this, we will be able to properly evaluate how well the solo-interaction features drive continued motivation.

In addition to well developed individual features, we really want to put a strong focus on micro-communities. We are interested in how small-group accountability influences cooking motivation and engagement. Thus, improving cross-user interactivity is important for this stage. Much of the Sprint 2 feedback we got from peer reviews noted the difficulty of adopting a community-based app when there's no existing activity - the "empty room" problem - and suggested that we recruit groups of friends to use the app together. So to solve this, we plan to recruit pre-existing groups of friends to onboard together.

We intend to gather feedback through a more longitudinal study, where we hope to give users (individuals and groups of friends), the chance to use the application for about a week or so. From there, we will either conduct follow-up interviews or send out a follow-up survey to get users' thoughts. Additionally, we plan to track engagement analytics through the applications.

## Sprint 2:

For our next learning prototype, we need to learn about the usability of the application. Are users able to pick up the solution and understand what to do with it right away? Do they understand all the functionality and features they can use? Are they overwhelmed by the application? Additionally, we need to see if we correctly fulfilled user needs and desires for a motivational and accountability tool. Did we correctly implement the feedback given to us in this Sprint? Will this approach actually motivate users to cook more? Can it be motivational for solo users as well?

Our priority in this next Sprint is to have a minimal working prototype where we can implement key features and test to see user reactions. Features that will be important to implement in this next step are micro-communities, prompting, challenges, rewards, user profiles, and connecting with friends. As many interviewees stated, being able to keep up with their friends would be a huge motivator to interact with the app. Implementing these features would also help to understand if the motivation would extend to them actually cooking. We also want to focus on challenges and prompting in order to promote use of the app with users who don't necessarily want to engage much with the communities. Like Strava, we want the solution to still be engaging in this case. Thus, we need to test if the current solo features we plan to implement would work for this.

Less critical at this stage are more advanced features like content creator functionality (as suggested by one of the interviews), live cooking, and recipe saving. While these features would definitely add to the experience of use, our research and interviews suggested that

they're not as important as the features above. They also stray a bit from the problem space, which is motivation and accountability. We first need to evaluate the usefulness of the basic functionality before adding onto it.

We intend to gather feedback by creating a lightweight version of the solution and asking participants to use it for a couple of days. Then, through follow-up interviews and possibly journal entries, evaluate the success of the product and further refine the features to make them as valuable to the user experience as possible.

#### Sprint 1:

The purpose of our next learning prototype is to test if the core features that we have narrowed down to: micro-communities, streak-based tracking, saving recipes, tagging recipes, and using AI to simplify recipes will actually add something of value to users and provide something different from current platforms. Though we also brainstormed many other features, our priority is to focus on the essential functions that tackle the biggest pain points from interviews: saving and organizing recipes, clear and simplified instructions from social media or blogs, and creating motivation through streaks, social sharing, and micro-community support. Less critical at this stage are more advanced features like stylized AI visuals or complex dashboards, since we need to first validate whether the social and habit-building aspects alone are compelling. With this, we want to answer questions like: Do streaks and cooking passports motivate users, or do these feel trivial? Would they prefer smaller groups to broader platforms, and does that improve their consistency in cooking or trying new recipes? Do AI-generated recipe summaries and measurement adjustments save time, or would users prefer original recipe formats? We intend to gather feedback through user testing sessions and follow-up interviews to further refine the features that users will use the most and the ones that add the most value to the user experience.

## Biggest Concerns

#### Sprint 5:

Although Sprint 5 introduced stronger feature maturity: mobile deployment, notifications, enhanced logging, expanded badges, and more structured micro-communities, meaning several key concerns remain as we look beyond this sprint. Our largest challenge continues to be the limited testing window and small sample size. While early results show encouraging engagement across both solo and social features, the short duration still makes it difficult to confirm whether users are developing genuine cooking habits or simply responding to novelty. To address this, longer multi-week testing will be essential, especially now that notifications and daily logging can generate richer behavioral data. A

second concern is the increasing complexity of our Firebase architecture. As we scale into community goals, reactions, friend connections, and admin-level analytics, our current schema risks becoming unwieldy. Several users and peers have flagged that Firebase can struggle with deeply relational data, prompting us to carefully evaluate whether future versions may require schema restructuring or even a long-term migration to a more structured backend.

Another major concern centers on ensuring the right balance between solo value and social motivation. Sprint 5 testing reaffirmed that combining both elements drives higher engagement, but we still need to guarantee that community participation feels additive rather than required. Visibility settings, private logging, and differentiated notifications help mitigate this, but we must continue refining how social features surface so that users who prefer to cook independently still feel fully supported. Feature completeness is another area needing attention; earlier inconsistencies such as disappearing journal entries, lack of interaction notifications, or limited prompt variety, limited the depth of the user experience. While Sprint 5 addressed many of these issues, maintaining freshness over longer usage periods will require continually evolving prompts, challenges, milestones, and community mechanics.

Finally, the most significant long-term concern is proving that our platform fosters sustained behavioral change rather than short bursts of activity. Even with stronger notifications, enhanced logging, and more robust micro-communities, it remains unclear whether engagement will remain stable over weeks. To answer this, we will rely heavily on longitudinal analytics: streak continuation rates, day-7 retention, week-over-week activity, notification response patterns, and comparisons between user-defined versus app-defined group structures. Establishing clear evidence of habit formation will be critical as we begin exploring monetization models, platform expansion, and broader community partnerships in future sprints.

#### Sprint 4:

While our Sprint 4 results show promising engagement patterns with 16 posts and 15 journal entries over 2.5 days across 10 users, several challenges remain. Our testing still relies on a relatively small sample size and short time frame, insufficient to confirm long-term habit formation. To address this, we are planning longitudinal testing for Sprint 5 that tracks user behavior over a full week or more, measuring whether initial enthusiasm translates to sustained engagement. A major technical concern involves scaling our Firebase architecture as we layer in more complex community features like shared content, reactions, community goals, and group progress metrics. Feedback from previous sprints highlighted that Firebase can become difficult to manage with complex relational

data, prompting us to document our database schema more thoroughly and consider whether we might need to migrate to a more structured backend in future development.

From a product perspective, we continue to refine the balance between social motivation and solo value. Our Sprint 4 testing showed that Group A, which had access to both personal and social features, posted more than Group B, suggesting that layering features together drives stronger engagement. However, determining how to make community participation feel optional rather than mandatory remains important, which we are addressing through clear visibility settings and ensuring solo users can access full value without joining groups. Another concern is feature completeness and preventing the app from feeling stale over repeated use. Users identified gaps like journal entries disappearing after submission, no notifications when someone interacts with their posts, and limited variety in prompts and milestones. We are addressing this in Sprint 5 by implementing persistent journal storage, notification systems, refreshed prompts, badges, and more robust community features including friend connections and profile pages. Deploying to mobile will allow push notifications and reduce friction from users forgetting to engage.

Our biggest concern remains proving long-term habit formation rather than novelty-driven use. While users posted frequently during our 2.5-day test and expressed enthusiasm, maintaining this behavior over weeks is untested. We are addressing this through rigorous analytics tracking including streak continuation rates, day seven retention, weekly active users, and feature-specific engagement metrics. Sprint 5 will specifically test whether notifications, more developed features, and longer exposure increase consistent usage, and whether user-defined groups versus app-defined groups generate different levels of engagement. By focusing on measurable behavioral data rather than sentiment alone, we aim to validate whether our core mechanisms genuinely support sustained cooking habits.

### Sprint 3:

While our early results were promising, several challenges emerged that we'll need to address in future sprints. Our sample size was limited to testers, which provided valuable qualitative insights but isn't enough to confirm engagement trends or measure long-term motivation. To strengthen our findings, we'll need broader and more diverse user testing to understand how different users respond to solo vs. community-based experiences. A major implementation challenge we faced involved integrating community features. Conceptually, micro-communities and shared challenges align with our goal of creating social accountability, but technically, they introduced complexity around data structure, permissions, and real-time updates. Within Firebase, managing shared content (like posts,

reactions, and group challenges) requires careful planning to prevent high read/write costs and performance slowdowns as data scales. From a design perspective, we also struggled to find the right balance — we want to add social motivation without overwhelming users who prefer a private, reflective cooking experience. Determining how to layer community features gradually, while keeping the personal dashboard central, remains an open question. Additionally, our analytics setup in Firebase was fairly minimal this sprint. We lacked event tracking for user journeys like challenge participation, journaling frequency, or time spent in the app – metrics that are crucial to understanding what drives sustained engagement. Finally, while short-term engagement looked positive, maintaining motivation over time is still untested. Our next sprint will focus on scalable data tracking, a more intentional rollout of community features, and experiments that measure long-term habit formation through both solo and small-group challenges.

### Sprint 2:

For Sprint 3, our biggest concerns revolve around the sustainability as well as moderation of micro-communities, as ensuring safety, respectful engagement, and consistent activity will be critical to long-term success. We are also cautious about balancing the personal and community tabs because too much emphasis on one could weaken the other, and we need to ensure both feel cohesive rather than fragmented. Another concern is scalability: while micro-communities work well in smaller groups, we need to confirm that the experience remains engaging and personalized as more users join. Finally, we are mindful of differentiation, making sure our platform's community-driven accountability stands out against larger, more generalized social media platforms, instead of being perceived as a duplicate of other existing community features.

### Sprint 1:

For the next stage, our biggest concerns are mainly the cost and difficulty of implementing an AI agent, as we need to ensure the cost-benefit analysis is there and users will actually value having this option or not. Additionally, we are nervous about the accuracy of this AI agent. We are also worried about the scope of the app, as we don't want to overwhelm users with too many features at once without a sense of cohesiveness across the platform. Another concern is determining which features actually provide unique value that users will adopt, rather than creating repetitive platforms similar to tools already available online.

### Sprint 5:

## Code Review

## Part 1: Code Organization Review

Our Pantry application follows a modern mobile-first architecture built with React Native and Expo, using Firebase as our backend-as-a-service platform. The frontend is structured using Expo Router for file-based routing, with a clear separation of concerns across five main directories: /tabs for navigation screens, /components for reusable UI organized by feature domain, /services for platform-specific business logic (particularly our dual notification system), /utils for helper functions, and /assets for static resources including our gamification badge system.

On the backend, we leverage Firebase's ecosystem comprehensively: Firestore for our NoSQL database with collections for users, feed (posts), journals, communities, and notification tokens; Firebase Authentication for user management; Firebase Storage for image uploads; Cloud Functions for server-side logic including 8+ functions handling streak calculations, notifications, denormalized data updates, and A/B test assignment; and Firebase Analytics for user behavior tracking. This architecture enables us to support two platforms—Android (native mobile) and Web (browser-based)—from a single codebase, fulfilling the 2nd platform requirement through platform-specific notification implementations.

The key architectural decision was implementing platform detection at the root level (\_layout.jsx) to initialize the appropriate services. This allows us to maintain shared business logic while accommodating platform-specific APIs, particularly for our notification system which required completely different implementations for mobile (Expo Notifications + FCM) versus web (Firebase Cloud Messaging with service workers).

## Part 2: Platform Examples with Code

### 1) Multi-platform notification system

One of our most significant technical challenges was implementing a unified notification experience across Android and Web platforms, despite these platforms requiring fundamentally different APIs.

[code from app/\_layout.jsx lines 16-29 for platform detection]

```
12 export default function RootLayout() {
13   const [user, setUser] = useState(null);
14   const [loading, setLoading] = useState(true);
15   const router = useRouter();
16   const segments = useSegments();
17   const auth = getAuth();
18
19   // Initialize notifications
20   useEffect(() => {
21     if (Platform.OS === 'web') {
22       // Initialize web notifications
23       webNotificationService.initializeApp(firebaseApp);
24     } else {
25       // Initialize mobile notifications
26       setupNotificationChannels();
27     }
28   }, []);
29
30   // Auth state listener
31   useEffect(() => {
32     const unsubscribe = onAuthStateChanged(auth, async (user) => {
33       setUser(user);
34       setLoading(false);
35
36       // Register for push notifications when user signs in
37       if (user) {
38         await registerForPushNotifications(user.uid);
39       }
40     });
41
42     return unsubscribe;
43   }, []);
44 }
```

```
// Register for push notifications
const registerForPushNotifications = async (userId) => {
  try {
    if (Platform.OS === 'web') {
      // Web notification registration (optional automatic setup)
      // User can manually enable later in settings
      console.log('Web notifications can be enabled in Settings');
    } else {
      // Mobile notification registration
      const { status, token } = await notificationService.requestPermissions();

      if (status === 'granted' && token) {
        await notificationService.registerToken(userId, token);

        // Set default notification settings
        await notificationService.updateSettings(userId, {
          dailyReminders: true,
          reminderTime: '09:00',
          reminderDays: ['monday', 'tuesday', 'wednesday', 'thursday', 'friday'],
          friendActivity: true,
        });

        // Schedule daily reminders (9:00 AM)
        await notificationService.scheduleDailyReminder(9, 0);
      }
    }
  } catch (error) {
    console.error('Error registering for notifications:', error);
  }
};
```

[Show code from app/services/notificationService.js for mobile implementation]

```
app > services > js notificationService.js >  NotificationService >  requestPermissions
16  class NotificationService {
17
18    /**
19     * Get Expo Push Token (wraps FCM for both platforms)
20     */
21    async getExpoPushToken() {
22      try {
23        const projectId = Constants.expoConfig?.extra?.eas?.projectId ??
24          Constants.easConfig?.projectId;
25
26        if (!projectId) {
27          throw new Error('Project ID not found');
28        }
29
30        const tokenData = await Notifications.getExpoPushTokenAsync({
31          projectId,
32        });
33
34        return tokenData.data; // This is your FCM token
35      } catch (error) {
36        console.error('Error getting push token:', error);
37        return null;
38      }
39    }
40
41    /**
42     * Register device token with Firestore
43     * Stores token in users/{userId}/fcmTokens/{tokenId}
44     */
45    async registerToken(userId, token) {
46      if (!token || !userId) {
47        console.log('Missing token or userId');
48        return false;
49      }
50
51      try {
52        const db = getFirestore();
53        const tokenId = this.generateTokenId(token);
54        const tokenRef = doc(db, `users/${userId}/fcmTokens`, tokenId);
55
56        await setDoc(tokenRef, {
57          token,
58          platform: Platform.OS,
59          deviceName: Device.deviceName || 'Unknown Device',
60          osVersion: Device.osVersion || 'Unknown',
61          createdAt: serverTimestamp(),
62          lastUsed: serverTimestamp(),
63        }, { merge: true });
64
65        console.log('Token registered successfully');
66        return true;
67      } catch (error) {
68        console.error('Error registering token:', error);
69        return false;
70      }
71    }
72  }
```

```
app > services > js notificationService.js > NotificationService > constructor
1 import * as Notifications from 'expo-notifications';
2 import * as Device from 'expo-device';
3 import { Platform } from 'react-native';
4 import { getFirestore, doc, setDoc, collection, serverTimestamp } from 'firebase/firestore';
5 import Constants from 'expo-constants';
6
7 // Configure notification behavior
8 Notifications.setNotificationHandler({
9   handleNotification: async () => ({
10     shouldShowAlert: true,
11     shouldPlaySound: true,
12     shouldSetBadge: true,
13   }),
14 });
15
16 class NotificationService {
17   constructor() {
18     this.notificationListener = null;
19     this.responseListener = null;
20   }
21
22 /**
23  * Request notification permissions
24  * Returns: { status: 'granted' | 'denied', token: string | null }
25  */
26 async requestPermissions() {
27   try {
28     // Check if device is physical (not simulator)
29     if (!Device.isDevice) {
30       console.log('Notifications only work on physical devices');
31       return { status: 'denied', token: null };
32     }
33
34     // Check current permission status
35     const { status: existingStatus } = await Notifications.getPermissionsAsync();
36     let finalStatus = existingStatus;
37
38     // Request permissions if not already granted
39     if (existingStatus !== 'granted') {
40       const { status } = await Notifications.requestPermissionsAsync();
41       finalStatus = status;
42     }
43
44     if (finalStatus !== 'granted') {
45       console.log('Notification permission denied');
46       return { status: 'denied', token: null };
47     }
48
49     // Get Expo push token (this will get FCM token internally)
50     const token = await this.getExpoPushToken();
51
52     return { status: 'granted', token };
53   } catch (error) {
54     console.error('Error requesting notification permissions:', error);
55     return { status: 'denied', token: null };
56   }
57 }
```

[Show code from app/services/webNotificationService.js for web implementation]

```

app > services > js webNotificationService.js > WebNotificationService > getToken > token
1 // Web Push Notification Service using Firebase Cloud Messaging (FCM)
2 // This service is specifically for web browsers (not React Native mobile apps)
3
4 import { getMessaging, getToken, onMessage } from 'firebase/messaging';
5 import { getFirestore, doc, setDoc, serverTimestamp } from 'firebase/firestore';
6
7 class WebNotificationService {
8   constructor() {
9     this.messaging = null;
10    this.currentToken = null;
11    this.unsubscribeOnMessage = null;
12  }
13
14 /**
15  * Initialize Firebase Messaging for web
16  * Must be called after Firebase app is initialized
17  */
18 initialize(firebaseApp) {
19   if (typeof window === 'undefined' || !('serviceWorker' in navigator)) {
20     console.log('Service Worker not supported in this environment');
21     return false;
22   }
23
24   try {
25     this.messaging = getMessaging(firebaseApp);
26     console.log('Firebase Messaging initialized for web');
27     return true;
28   } catch (error) {
29     console.error('Error initializing Firebase Messaging:', error);
30     return false;
31   }
32 }
33
34 /**
35  * Request notification permission from the user
36  * Returns: { status: 'granted' | 'denied' | 'default', token: string | null }
37  */
38 async requestPermission() {
39   if (!this.messaging) {
40     console.error('Firebase Messaging not initialized');
41     return { status: 'denied', token: null };
42   }
43
44   try {
45     // Request permission
46     const permission = await Notification.requestPermission();
47
48     if (permission !== 'granted') {
49       console.log('Notification permission denied');
50       return { status: permission, token: null };
51     }
52
53     // Get FCM token
54     const token = await this.getToken();
55
56     return { status: 'granted', token };
57   } catch (error) {

```

**Challenge:** The Expo Notifications API (mobile) and Firebase Cloud Messaging API (web) have completely different permission models, token generation processes, and message handling patterns. Mobile uses Expo push tokens that map to FCM tokens, while web requires VAPID keys and service worker registration.

**Solution:** We created platform-specific service layers that expose a consistent interface to our components. Platform detection happens once at app initialization, and the appropriate service is loaded. Both services expose the same methods (`requestPermissions`, `registerToken`, `setupListeners`), but implement them differently under the hood.

Learning: Attempting to create a single unified notification service led to complex conditional logic and fragile code. Separating into distinct services with a shared interface pattern produced cleaner, more maintainable code. This also made testing easier since we could test each platform implementation independently.

## 2) Real-time streak calculation

Our streak system is central to user motivation, so accuracy and consistency across all clients was paramount. Initially, we calculated streaks client-side, but quickly encountered issues with timezone differences and race conditions when multiple devices were logged in.

[Show Cloud Function code from functions/index.js - updateStreakOnNewPost]

```
526 exports.updateStreakOnNewPost = onDocumentCreated({
527   database: "pantry1",
528   document: "feed/{postId}",
529   region: "us-central1",
530 }, async (event) => {
531   const snapshot = event.data;
532
533   if (!snapshot) {
534     console.log("No data found in event snapshot.");
535     return null;
536   }
537
538   const postData = snapshot.data();
539   const userId = postData.uid;
540   const userRef = db.collection("users").doc(userId);
541   const userDoc = await userRef.get();
542   const userData = userDoc.data();
543
544   // Basic check to ensure user data exists
545   if (!userDoc.exists || !userData) {
546     console.log(`User ${userId} not found or has no data. Cannot update streak.`);
547     return null;
548   }
549
550   // 1. Get the current post's time (server's time in UTC)
551   const now = new Date();
552
553   // 2. Get the recorded streak start date (the Sunday midnight timestamp set by the client)
554   const lastStartTimestamp = userData.streakStartDate;
555   const lastRecordedStart = lastStartTimestamp ? lastStartTimestamp.toDate() : null;
556
557   // 3. Calculate the Sunday midnight for the time the post was made (in UTC)
558   const currentWeekStart = getStartOfWeek(now).getTime();
559
560   // 4. Calculate the Sunday midnight for the recorded streak start (in UTC)
561   // This value should be the exact Sunday midnight timestamp sent by the client.
562   const lastWeekStart = lastRecordedStart ? getStartOfWeek(lastRecordedStart).getTime() : 0;
563
564   // Check if the current calendar week start is LATER than the recorded week start
565   const isNewWeek = currentWeekStart > lastWeekStart;
```

```

567 // streak calculation
568 let newCurrentWeekPosts = userData.currentWeekPosts || 0;
569 let newStreakCount = userData.streakCount || 0;
570 let newStreakStartDate = userData.streakStartDate;
571
572 let hasMetGoal = userData.hasGoalBeenMetThisWeek || false;
573 const weeklyGoal = userData.weeklyGoal || 0;
574
575 if (isNewWeek) {
576   newCurrentWeekPosts = 1;
577   newStreakStartDate = admin.firestore.Timestamp.fromDate(getStartOfWeek(now));
578   hasMetGoal = false; // Reset the flag for the new week
579 } else {
580   newCurrentWeekPosts += 1; // If NOT a new week, just increment the posts
581 }
582
583 if (!hasMetGoal && newCurrentWeekPosts >= weeklyGoal) {
584   // Goal is being met with this new post, AND the streak hasn't been incremented yet
585   newStreakCount += 1;
586   hasMetGoal = true; // Set the flag to prevent double-counting this week's goal
587   console.log(`GOAL MET! Streak incremented immediately to ${newStreakCount}.`);
588 }
589
590 // 3. Update the user document
591 await userRef.update({
592   currentWeekPosts: newCurrentWeekPosts,
593   streakCount: newStreakCount,
594   streakStartDate: newStreakStartDate,
595   hasGoalBeenMetThisWeek: hasMetGoal,
596 });
597
598 console.log(`Streak updated for user ${userId}.`);
599 console.log(`Posts: ${newCurrentWeekPosts}, Streak: ${newStreakCount}`);
600 return null;
601 });
602

```

**Challenge:** Week boundaries depend on timezone, and users could be in different timezones. Client-calculated streaks led to inconsistencies where a post made at 11pm PST counted as a different day than the same post made by a user in EST. Additionally, rapid successive posts could cause race conditions where the streak count got out of sync.

**Solution:** We moved all streak calculations to server-side Cloud Functions triggered by new posts. Using Firebase's `serverTimestamp()` ensures all calculations happen in a consistent timezone (UTC). The function calculates week boundaries using a standardized `getStartOfWeek` helper, compares against the user's last streak start date, and atomically updates the user's document with the new streak count and progress.

**Learning:** Server-side calculations with Cloud Functions eliminate client-side inconsistencies and race conditions. The trade-off is slightly delayed UI updates (waiting for the function to complete), but we mitigated this with optimistic UI

updates that get corrected if the server calculation differs. This architecture is essential for any feature where data integrity is critical.

### 3) A/B Testing Infrastructure

To validate whether personal habit-building features or community features drove more engagement, we implemented A/B testing infrastructure that randomly assigns users to groups and tracks their behavior through Firebase Analytics.

[Show Cloud Function code from functions/index.js - assignABGroup]

```
661 // Assigns a new user to Group A or Group B based on which group has fewer members.
662 exports.assignABGroup = onCall({
663   region: "us-central1",
664   enforceAppCheck: false, // Set to true if you use App Check
665 }, async (request) => [
666   // 1. Get the authenticated user's ID
667   const userId = request.auth.uid;
668   if (!userId) {
669     throw new functions.https.HttpsError(
670       "unauthenticated",
671       "The function must be called while authenticated.",
672     );
673   }
674
675   const countersRef = db.collection("abTestCounters").doc("userSplit");
676   const userProfileRef = db.collection("users").doc(userId);
677   let assignedGroup = null;
678
679   // --- 2. Run Atomic Transaction to Assign Group ---
680   try {
681     await db.runTransaction(async (transaction) => {
682       const doc = await transaction.get(countersRef);
683
684       // Initialize counters if the document doesn't exist
685       if (!doc.exists) {
686         // Initialize with 0 and assign the very first user to Group A
687         transaction.set(countersRef, {groupA_count: 0, groupB_count: 0});
688         assignedGroup = "Group A";
689         transaction.update(countersRef, {groupA_count: admin.firestore.FieldValue.increment(1)});
690         return;
691       }
692
693       const data = doc.data();
694       const countA = data.groupA_count || 0;
695       const countB = data.groupB_count || 0;
696
697       // Assign to the group with fewer (or equal) members
698       if (countA <= countB) {
699         assignedGroup = "Group A";
700         transaction.update(countersRef, {groupA_count: admin.firestore.FieldValue.increment(1)});
701       } else {
702         assignedGroup = "Group B";
703         transaction.update(countersRef, {groupB_count: admin.firestore.FieldValue.increment(1)});
704       }
705     });
706   }
```

[Show analytics code from app/utils/analyticsHelper.js]

```

6  /**
7  * Logs a custom event for every successful post creation, tagged with the A/B group.
8  * @param {string} abGroup - The user's assigned group ('Group A' or 'Group B').
9  */
10 export const logPostCreation = (abGroup) => {
11   const eventName = 'pantry_post_created'; // A clear event name
12
13   if (analytics) {
14     logEvent(analytics, eventName, {
15       // CRITICAL: Tagging the post with the user's A/B group
16       ab_test_group: abGroup,
17     });
18     console.log(`[Analytics] Logged post creation for Group ${abGroup}`);
19   } else {
20     console.warn("Firebase Analytics is not initialized. Cannot log post event.");
21   }
22 };

```

**Challenge:** Ensuring perfectly balanced group assignment (50/50 split) while preventing users from manipulating their group assignment or being reassigned on subsequent logins. Also needed seamless integration with Firebase Analytics for segmented analysis.

**Solution:** Server-side group assignment using Firestore transactions to maintain atomic counters. When a user signs up, a callable Cloud Function queries the current counts and assigns the user to whichever group has fewer members, then increments that group's counter. All within a transaction to prevent race conditions. The assigned group is permanently stored in the user's profile and used to tag all analytics events.

**Learning:** Client-side random assignment is easily gamed and can lead to unbalanced groups. Server-side transactions ensure perfect balance and prevent manipulation. Integration with Firebase Analytics' custom parameters enabled us to segment all metrics by A/B group without additional infrastructure.

#### 4) Image Upload

Image uploads needed to work across multiple platforms (web, Android) and handle various formats (JPEG, PNG, HEIC) while maintaining quality and reasonable file sizes.

[ code from app/utils/imageUpload.js - uploadImageToFirebase function]

```
app > utils > JS imageUpload.js > uploadImageToFirebase > extension
1 import * as ImagePicker from 'expo-image-picker';
2 import { getDownloadURL, ref, uploadBytes } from 'firebase/storage';
3 import { Alert, Platform } from 'react-native';
4 import { storage } from '../../../../../firebaseConfig'; // Adjust the import path as needed
5
6 // --- Image Picker/Camera Logic ---
7 /**
8  * Requests necessary permissions for the image picker/camera.
9  * @param {'camera' | 'library'} type - The type of permission to request.
10 * @returns {Promise<boolean>} - True if permission is granted, false otherwise.
11 */
12 const requestPermissions = async (type) => {
13   const permissionMethod =
14     type === 'camera'
15       ? ImagePicker.requestCameraPermissionsAsync
16       : ImagePicker.requestMediaLibraryPermissionsAsync;
17
18   const { status } = await permissionMethod();
19   if (status !== 'granted') {
20     Alert.alert('Permission needed', `Please allow access to your ${type} to continue.`);
21     return false;
22   }
23   return true;
24 };
25
26 /**
27  * Launches the image library or camera.
28  * @param {'camera' | 'library'} type - The source to launch.
29  * @returns {Promise<{uri: string, mimeType: string} | null>} - The image asset data or null if cancelled/failed.
30 */
31 export const launchImagePicker = async (type) => {
32   const isGranted = await requestPermissions(type);
33   if (!isGranted) return null;
34
35   const launchMethod =
36     type === 'camera' ? ImagePicker.launchCameraAsync : ImagePicker.launchImageLibraryAsync;
37
38   const result = await launchMethod({
39     mediaTypes: 'Images',
40     allowsEditing: true,
41     aspect: [4, 3], // You might want to make this configurable for profile vs. post
42     quality: 0.7,
43   });
44
45   if (result.canceled || !result.assets || result.assets.length === 0) {
46     return null;
47   }
48
49   const asset = result.assets[0];
50
51   // HEIC/TIFF web check
52   if (Platform.OS === 'web' && asset.mimeType && (asset.mimeType.includes('heic') || asset.mimeType.includes('tiff'))) {
53     Alert.alert(
54       "File Conversion Failed",
55       "This file type cannot be displayed in the web browser. Please convert the image to JPEG or PNG externally before uploading."
56     );
57     return null;
58 }
```

Challenge: We encountered a "could not find MIME for Buffer" error when uploading images to Firebase Storage. The blob created from fetch() didn't always have MIME type metadata attached, especially on web platforms, causing Firebase to reject the upload.

**Solution:** We explicitly pass metadata with the `contentType` field to the `uploadBytes()` function. The `uploadImageToFirebase` function now validates the `mimeType` parameter and uses it as the `contentType` in the upload metadata, with a fallback to 'image/jpeg' if no MIME type is provided.

**Learning:** Firebase Storage requires explicit MIME type specification for blob uploads. Relying on the blob's inherent content type is unreliable across platforms. Always include metadata when uploading to Firebase Storage. This fix also improved our image serving since browsers now receive correct Content-Type headers.

### Part 3: Additional Technical Challenges and Learnings

1. **Denormalized Data Consistency:** When users update their profile (name, photo), these changes must propagate to all their posts and comments across the app.
  - a. **Solution:** Cloud Function with batch writes that updates all documents where the user appears
  - b. **Learning:** Denormalization trades write complexity for read performance; acceptable for infrequently updated data
2. **Real-Time Data Synchronization:** Feed, communities, and profiles all use Firestore snapshots for live updates
  - a. **Challenge:** Managing multiple concurrent subscriptions without memory leaks
  - b. **Solution:** Properly structured `useEffect` hooks with cleanup functions
  - c. **Learning:** Firestore snapshots are powerful but require careful subscription lifecycle management
3. **Scheduled Cloud Functions:** Daily reminders must fire at appropriate times for users across timezones
  - a. **Solution:** Cloud Scheduler + Cloud Function that queries users by timezone stored in their settings

- b. Learning: Cloud Functions scale automatically; storing user preferences enables personalization at scale
- 
- 4. Badge Calculation Performance: Initially recalculated all badges on every activity
    - a. Solution: Incremental evaluation with early exit logic; only check badges that could have been earned
    - b. Learning: Client-side works for simple rules; complex logic should move server-side for consistency

## Future Plans

The next phase of Pantry focuses on strengthening our habit-building mechanics, improving the app's technical foundation, and developing a sustainable business model. Everything builds directly on what we learned and was validated in our previous sprints—especially the power of timely nudges, supportive communities, and visible progress tracking. Below is a detailed future direction plan.

### Technical Improvements

**Adaptive Notification Implementation:** Instead of basic scheduled reminders, we'll move toward adaptive and personalized notifications. This involves using user behavior data (e.g., last log time, streak vulnerability, typical cooking window) to trigger context-aware nudges. For example, a "**Streak Rescue Alert**" could be dynamically sent one hour before a user's cooking "deadline" if they have an active streak but no log for the day.

- **Justification:** Sprint 5 validated that timely prompts are critical to overcome the "temporal deficit" of forgetting to cook.

**Expansion of Offline-First Architecture:** While basic offline logging is implemented, we will expand offline caching to include the recipe library, community feed snapshots, and earned badge/milestone data.

- **Justification:** Given the use of the app in real-world kitchen environments, ensuring access to necessary content (recipes, saved posts) regardless of connectivity is critical for lowering friction and improving the core user experience.

**Performance Monitoring Integration:** We will fully integrate performance monitoring tools (e.g., Firebase Performance Monitoring) to proactively track app load times, query latency, and cold-start times for all Cloud Functions.

- **Justification:** As we scale, performance bottlenecks in high-load areas (such as streak calculation or the Admin Dashboard's analytics aggregation) must be identified and optimized before they impact the user experience at scale.

### Feature Enhancements (based on user feedback)

Feature enhancements will concentrate on providing a more structured community experience and deeper gamification loops.

**Multi-Step Community Challenges:** This is the next major feature. We will implement weekly challenges that mix personal goals with group accountability. These will be structured as multi-step quests (e.g., "Cook with a new ingredient," "Share a photo,") and can be tied to shared milestones and friend-group cooking goals.

- **Justification:** User testing showed that community support and goal-oriented groups are key drivers of long-term engagement, aligning with the "Stay for the Community" model of successful habit-tracking platforms like Strava.

**Flexible Goal Setting:** Based on Sprint 5 A/B test feedback, which showed that some users (especially students) found it challenging to create their own goals, we will implement a flexible challenge system that offers:

- **Guided Structure (Presets):** A pre-defined checklist to reduce cognitive load.
- **Autonomy:** The ability for experienced users to set their own goals.

**Journaling Depth:** We will expand the reflection question sets and introduce daily/weekly rotating prompts to maintain user engagement and support the app's reflective design component.

**Multi-Tiered Badge Progression:** Implement a deeper, more satisfying multi-tiered badge system (e.g., "Novice Chef," "Home Cook," "Culinary Master") to ensure rewards feel meaningful and continue to motivate long-term retention.

### Scalability & Architecture Evolution

The primary architectural concern is the management of relational data in a NoSQL environment (Firebase), which is complicated by the introduction of complex community features.

**Evaluate Firebase Schema + Possible Migration:** We will thoroughly document the current Firebase schema and run stress tests on relational queries required by the new

**Multi-Step Challenges and Feed Ranking logic.** If the current architecture proves difficult to manage or scale with relational data (shared content, group progress, etc.), we will pilot a migration of these complex, highly relational modules (e.g., user profiles, challenge logic) to a more structured, cost-effective backend (e.g., a managed PostgreSQL or similar relational database).

- **Justification:** This addresses the key technical constraint identified in the sprint: Firebase's difficulty with complex relational data. A phased migration is a responsible approach to scaling while minimizing disruption.

**Stronger Data Pipeline:** Clean up the way we aggregate and denormalize data for the Admin Dashboard so we don't slow down the main app. This ensures that app developers (Admins) have access to high-level, real-time engagement patterns (log submissions, streak rates) without compromising the performance or data integrity of the main user application.

## Business Model & Monetization

The focus is to introduce a Sustainable Revenue Model, ensuring monetization is non-intrusive and aligned with motivational value.

**Refined Subscription (Premium Tier):** Formalize a premium subscription tier that includes two key value-adds identified in our competitive/user analysis:

- **Ad-Free Experience:** Removes all advertising to cater to the user segment prioritizing seamless experience.
- **AI-Powered Goal Setting & Personalization:** Offers a highly tailored challenge creation experience, potentially using AI to generate goals based on user progress and dietary needs.

**Non-Intrusive Advertising Model:** Introduce an ad-supported experience that fits naturally into the app or the feed. This includes promoting sponsored ingredients, relevant cooking products, or nearby grocery services directly within the recipe or community feed.

**Monetized Streak-Recovery Boosts:** Explore monetizing "Streak-Recovery Boosts" as a micro-transaction. This feature allows a user who missed a day to maintain their streak for a small fee, providing both a revenue stream and a way to re-engage users who might otherwise be discouraged.

## User Experience & Design Evolution

**Social/Solo Balance Control:** Add clearer notification controls and optional community features so solo users don't feel overwhelmed. This could include managing community alerts separately from streak/personal reminders.

**Guided Onboarding for Challenges:** Design an explicit, streamlined onboarding sequence for the new Multi-Step Challenges feature to clearly communicate the rules, rewards, and how group accountability functions, maximizing feature adoption.

**Admin Dashboard UX/UI:** Improve the Admin Dashboard's visualization and filtering tools to allow us to easily interpret complex behavioral data (e.g., drop-off points in a challenge, high-activity groups) and guide our users effectively.

### Research & Validation Priorities

- A/B test deeper gamification (tiered badges, streak saves).
- Validate adaptive notifications and find the best timing patterns.
- Study which challenge formats drive the most participation and streak continuation.

### Technical Debt & Code Quality

Maintaining a clean, well-tested codebase is essential for supporting the planned architectural evolution and new features.

**Refactor Core Logic:** Conduct a refactoring sprint to optimize and standardize the code for Cloud Functions (e.g., updateStreakOnNewPost, badge calculation).

- **Justification:** Moving complex logic server-side ensures consistency, but requires the code to be highly efficient and well-documented to prevent errors as the user base grows.

**Improve Documentation and Testing:** Increase the level of in-line code documentation for all Firebase backend functions and crucial frontend components. Also expand unit/integration tests for new systems like challenges and notifications.

**Denormalization Management Audit:** Review our batch writes and update logic to make sure they scale without causing data inconsistencies.