Elevate your cooking one swipe at a time

Context:

Plated is a mobile app for amaetur and seasoned chef's alike to discover and share recipes from local chefs, friends, and family effortlessly based on the ingredients in your pantry or grocery list. Users can display their creativity while gaining technical skills to be self-expressive and independent in the kitchen. Explore more of the high fidelity implementation of plated below!

Mobile Installation Requirements:

1. If you'd like to interact with our prototype on your phone, download the Expo Go app on your mobile device.
2. Scan this QR code using the camera app on your ios device:

Operating Instructions:

1. Go to our github repo & press the green code button to download the zip file of our source code.
2. Unzip the file, then in your terminal, navigate into the folder platedapp52 that you just unzipped.
3. In the terminal, navigate to this folder using cd (local path) & enter the command npm install. This will install the project dependencies.
4. Start the dev server by running npx expo start into the terminal.
5. Scan the QR code produced in the terminal or type "i" to test on a simulator.
6. Follow the on-screen instructions from Expo CLI to interact with our prototype on your computer or mobile device.
7. Enjoy! Thank you for taking the time to explore Plated.

Design Tools:

We combined several tools to bring our app prototype to life. Starting with Notability, we sketched low-fidelity prototypes to brainstorm ideas and visualize the app's basic flow. Once the layout and task flows were set, we moved to Figma, where we built a medium-fidelity prototype, refining key features and interactions. To create consistent and polished icons and logos, we used SF Symbols, aligning the design with iOS standards. This approach offered several benefits: Notability allowed for quick ideation, Figma streamlined design and testing, and SF Symbols provided a cohesive visual style. From our figma prototype, we implemented our High Fidelity prototype in React Native using Expo Go simulator. We utilized Chat GPT to help us work through parts that were beyond our knowledge scope of react native.

UI & Features Walk-Through:

Our prototype offers a streamlined experience for discovering, sharing, and organizing recipes, with added support from an AI chef assistant. The bottom navigation bar provides easy access to the main sections of the app: *Home*, *Pantry*, *Recipe Posting*, *Chef Su*, and *Profile*.

- Home Screen: Before browsing recipes, users can set their personalized filters using the filter icon in the top left. These filters allow users to customize their feed based on their preferences, cuisine types, and ingredient availability, ensuring that the recommendations align with their unique tastes and needs. Once filters are set, begin swiping through a curated feed of recipes. Swipe down to save a recipe to your recipe box or up to skip to the next one. Tap on a recipe to view detailed information, including ingredients and cooking instructions. The back button in the top left allows you to return to the main feed.
- Pantry Screen: This section helps you manage your ingredients, keeping track of items you have at home and what needs replenishing. The pantry is organized by categories for easy browsing, and you can add or remove items by tapping on the relevant category.
- Recipe Posting Screen: Use the center navigation button to share your own recipes. You can take a photo or select one from your library, then add a title, recipe element descriptions, recipe instructions, and ingredients before posting. This feature encourages sharing creativity within the community.
- Chef Su Screen: Chef Su is our AI-powered cooking assistant, specifically designed to answer common cooking questions in real time. Users can ask Chef Su for guidance on cooking techniques, ingredient substitutions, or specific steps within a recipe. Chef Su's role is to provide helpful, focused answers to enhance the cooking process, rather than offering generalized advice or meal planning suggestions. This makes Chef Su an ideal resource for quick, targeted support when you're in the middle of cooking and need answers right away. If you are unfamiliar with AI powered Chat Agents, the information button in the top right corner will offer information and disclaimers about our chat bot.
- Profile Screen: The profile page shows user preferences and dietary restriction information in the upper left three line dropdown, allowing for a more personalized recipe recommendation experience. Here, you can also access your saved recipes and adjust personal settings.

Throughout the app, interactive hotspots guide the user through each screen, with most actions relying on clicks or swipes. Remember that the back button in the top left corner is available on most screens for easy navigation.

Tasks:
Simple: Swipe to discover and save recipes
Moderate: post a new recipe
Complex: add items to your pantry

Limitations:

The profile is also limited to a single preset user, meaning that personalization options are restricted to predefined choices. These constraints allowed us to focus on crafting a cohesive user experience, though they limit some interactive aspects of the app. Additionally:

- There is one preset test user profile
- You cannot communicate with other real users.

Wizard of Oz:

In our prototype, certain features are represented through Wizard of Oz techniques to illustrate intended functionality without fully building it out. We tell the user that you are able to ask Chef Su to do certain tasks based on your ingredients in your pantry or grocery cart. While chef Su is a fully Gemini AI API powered bot, it doesn't have access to local or supabase user data such as the pantry or grocery cart.

These Wizard of Oz elements help us portray the prototype's vision and user experience goals while keeping development focused on core interface flow.

Hard Coded Items:

Our final prototype includes a few hard-coded elements to streamline the demonstration and focus on the core functionality. For example, only the name and images are viewable for the recreations. Similarly, when users post recipes, only the recipe name and image are uploaded to the database. Ingredients, steps, and other variables are hard coded to ensure formatting stays consistent. Below is an overview of all of the hard coded items:

- Filter Options: When users select the filters from the home page, the options are static. Users cannot add options that we didn't include or take any off. We hard coded these items to ensure output consistency when the filters are saved and the script to render the recipes with the selected filters runs. Similarly, the permanent filter is statically set with "Nut Free" to show users that they can add filters that always apply. To focus on every-day filtering, though, we do not let users change their permanent filters for the demo.
- Servings: To save time when creating the database, we hard coded the number of servings to be 4 for all recipes. We also do not pass in servings into the filters, to ensure that the stack of matching recipes is always large enough to allow swiping.
- Recreations: The Recreations do not have dynamic detail pages like the rest of the recipes. We view this feature as something to implement for future iterations of the app, but for now simply showing a list of recreations in the users page suffices in showing users that they have the option to recreate recipes and add their own personal touch.
- Posting Recipes: Currently, users can post recipes with an image. These are displayed in the profile page, but the recipe details page of posted recipes is not accessible due to string parsing issues we encountered. We instead allow users to scroll through the images and names of their posts, but they cannot access the details.
- Pantry: The data for the pantry is hard coded with two json files in the app that store the information for what is in your pantry. This allows users to simulate adding items, but saves us time from setting up a complete database that handles updating and removing items from your pantry. We view this as a secondary feature to recipe discovery, so simulating the experience with a hard-coded database is enough for users to understand functionality.

- User Profiles + Followers: The profiles of the logged in user (chef you) and of other users are hard coded to specific names with profile images imported into the app. We did not set up user authentication or management tables, but while using the app, the hard-coded profile pictures and names are displayed to match the experience of what it would look like with fully-developed user management systems. Similarly, the number of followers is set to a fixed value.