

SmartCart: An AI Agent for Busy Home Cooks

Luke Lin **Yi Zhang** **Jessica Zhang** **Matteo Perona**
lul018@ucsd.edu yiz142@ucsd.edu liz035@ucsd.edu mperona@ucsd.edu

Jingbo Shang
jshang@ucsd.edu

Abstract

Students often have enough time to cook, but not enough time to organize and plan their meals. In the past, home cooks have primarily relied on cookbooks, social media, and websites to discover what to cook, and they have used web searches, trial and error, or word of mouth to discover where to buy their ingredients. These solutions, while effective, require the cook to invest lots of time searching through digital and physical media as well as exploring the stores in their area which makes daily meal planning very difficult for most people. SmartCart simplifies the process of meal planning by solving the problems of recipe discovery and store search in one integrated solution. This project introduces a specialized large language model (LLM) agent which engages with the user to identify a dish to cook based on their meal preferences. Once a dish is chosen, the agent generates a comprehensive shopping list of required ingredients with estimated prices and a short list of stores where the user can purchase the items based on their location. SmartCart creates an integrated solution for meal selection and ingredient sourcing, improving time efficiency over existing solutions, enabling busy people to cook at home without investing.

Website: https://jz024.github.io/grocery_website/

Code: <https://github.com/jz024/grocery>

<https://github.com/lukelin15/REACT-NATIVE>

| | | |
|---|-------------------------------|---|
| 1 | Introduction | 2 |
| 2 | Methods | 5 |
| 3 | Results | 7 |
| 4 | Discussion | 8 |
| 5 | Conclusion | 9 |

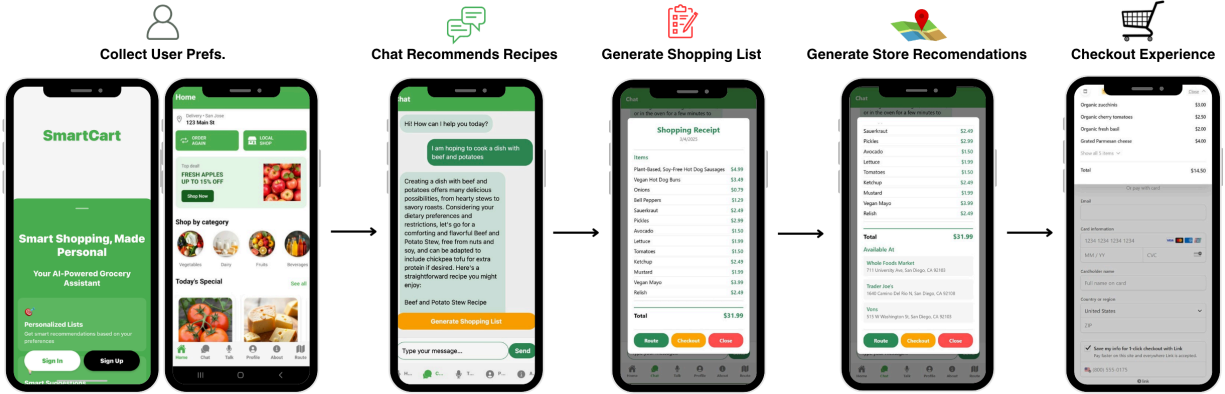


Figure 1: Use flow of SmartCart (1) We collect user data including dietary restrictions, culinary preferences, and location; (2) User chats with our agent to determine what dish they want to cook; (3) Agent generates a shopping list and (4) store recommendations; finally, (5) the user either navigates to the stores or uses our tailored checkout experience.

1 Introduction

Meal planning, which involves selecting recipes, preparing shopping lists, and sourcing ingredients, is a time-consuming and often inefficient process for home cooks. As modern lifestyles become increasingly fast-paced, individuals seek solutions that simplify meal preparation while accommodating their dietary preferences, budget constraints, and time limitations. Despite the vast availability of online recipes and meal-planning applications, existing solutions often lack personalization and seamless grocery store integration. As a result, users often spent too much time sifting through recipes and stores to find options that meet their needs.

In this work, we present **SmartCart**, a system designed to streamline the meal-planning process by unifying recipe discovery, personalized meal recommendations, and grocery sourcing into a single, user-friendly interface. Our solution employs a specialized large language model (LLM) agent that interacts with users to gather their dietary preferences, budget constraints, and location information before suggesting suitable dishes. Once a recipe is selected, the agent generates a tailored shopping list with estimated prices and offers curated store recommendations based on proximity and availability. We evaluated SmartCart through internal testing and user feedback, observing a reduction in total meal-planning time and an increase in user satisfaction compared to conventional approaches. These results suggest that an integrated, AI-driven platform can significantly alleviate the burden of organizing daily meals, thereby encouraging more individuals to cook at home while accommodating their personal and logistical needs.

1.1 Prior Work

Several meal-planning applications and recommendation systems exist, but many have limitations in personalization and grocery store integration. Traditional meal-planning tools, such as *Mealime* and *Paprika*, focus primarily on recipe organization without dynamically adjusting recommendations based on individual dietary needs or store inventory. Other AI-driven recommendation systems, such as *IBM Watson Meal Planner*, employ machine learning for personalized meal suggestions but lack real-time store data integration. Additionally, grocery shopping apps like *Instacart* provide product availability but do not offer intelligent meal planning based on user preferences. Our approach differs by leveraging state-of-the-art natural language processing (NLP) techniques to personalize meal recommendations while integrating real-time store data for seamless ingredient sourcing.

1.2 Data Description

This project does not rely on external datasets. We collect information from the user, store their chat history, and store all shopping lists they generate while using the app. All other data we use comes through the APIs covered in our methods section.

1.2.1 User Preferences

User preferences are stored in Firebase covering (1) the user's culinary preferences, (2) any dietary preferences, (3) food allergies, and (4) the user's location. These preferences are used by the agent to tailor recipe recommendations and find nearby stores. Each time the agent generates a shopping list it revises the user preferences given the latest chat history. Reference the following example of user preferences:

```
{
  "cuisinePreferences": [
    "Jamaican",
    "Italian",
  ],
  "dietaryPreferences": [
    "Gluten-Free",
    "Lactose-Free"
  ],
  "foodAllergies": [
    "Fish"
  ],
  "location": {
    "latitude": 32.8666937,
    "longitude": -117.2213634
  }
}
```

1.2.2 Chat History

The user's complete chat history is stored in MongoDB. Chat history is used to later generate shopping lists and store recommendations. Each user message and response is stored in a chat object like the following:

```
{
  "_id": {
    "$oid": "67c6aef650d929f3f024d484"
  },
  "uid": "Y8401MLwhMRTUISOeKHHJucdCBq2",
  "type": "chat",
  "user_message": "can you generate the shopping list",
  "ai_response": "Certainly! Here's a concise grocery shopping list tailored for
  ↪ making a vegan rendang for your dinner today ...",
  "timestamp": {
    "$date": {
      "$numberLong": "1741045366841"
    }
  }
}
```

1.2.3 Shopping Lists

Shopping list objects consist of the (1) list of items one is shopping for with their estimated prices and (2) a list of recommended stores to purchase those items. After shopping lists are generated they get stored to a MongoDB collection in the following form:

```
{
  "_id": {
    "$oid": "67c6adea50d929f3f024d482"
  },
  "uid": "Y8401MLwhMRTUISOeKHHJucdCBq2",
  "type": "shopping_list",
  "list_data": {
    "items": [
      {
        "name": "Mushrooms (portobello or shiitake)",
        "price": "3.5"
      },
      {
        "name": "Potatoes",
        "price": "2.0"
      },
      ...
    ],
    "store_recommendations": {
      "stores": [
        {
          "name": "Marukai Market",
          "address": "8151 Balboa Ave, San Diego, CA 92111",
          "available_items": [
            "Mushrooms (shiitake)",
            "Garlic",
            ...
          ]
        },
        ...
      ]
    }
  },
  "timestamp": {
    "$date": {
      "$numberLong": "1741045098003"
    }
  }
}
```

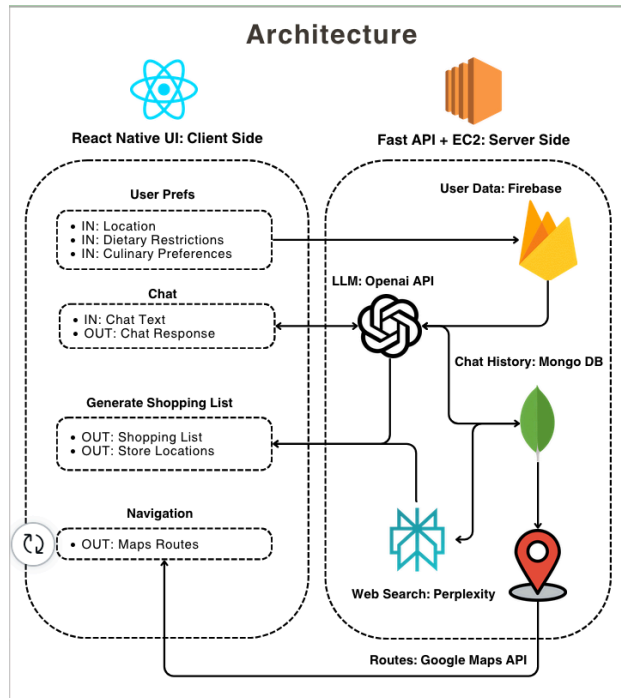


Figure 2: **SmartCart architecture**, API calls from user actions are shown on the left while the backend infrastructure they interact with are on the right. Arrows show how data is passed between different parts of the system.

1.2.4 Data Relevance and Usage

The collected data plays a crucial role in enhancing the system’s functionality:

- **User Preferences** drive meal personalization and ensure recommendations align with dietary needs.
- **Chat History** enables contextual responses and assists in refining shopping list generation.
- **Shopping Lists** provide a structured format for grocery planning and store selection.

By integrating these data sources with external APIs, the system dynamically adapts to user preferences and real-world availability, ensuring an efficient and personalized meal-planning experience.

2 Methods

Our system integrates a React Native front end for mobile, with a backend built in python using FastAPI hosted on AWS EC2. We used the following APIs and services:

1. **OpenAI API:** we used GPT-4o through OpenAI API to implement LLM driving our chat service.
2. **Perplexity API:** Perplexity is an AI based search engine we used for store recom-

mendations.

3. **Firestore:** stores user data collected on sign up.
4. **MongoDB:** we used MongoDB to store chat history, store recommendations, and .
5. **Shopping Lists:** used to provide routes to recommended stores given their addresses.

We start by collecting user preferences on sign up. This includes their culinary preference, dietary preferences, food allergies, and location. These get stored to Firestore. At this point, the user can start interacting with the agent through chat. The user's preferences are fetched from Firestore and added to the context window for OpenAI api each time a chat request is sent.

Chats can look like the following:

User: Hi. I want Japanese udon.

SmartCart: Absolutely! Udon is a delicious and versatile Japanese noodle dish. Since you have dietary preferences and allergies, let's focus on a plant-based, soy-free, and nut-free version. Here's a simple recipe for a homemade Vegan Udon soup that you can customize to your liking:

Vegan Udon Soup Recipe (Soy-Free & Nut-Free)

Ingredients:

- Udon Noodles (ensure they are vegan and soy-free)
- 4 cups Vegetable Broth (low-sodium)
- 2 Carrots, thinly sliced
- 2 stalks Celery, thinly sliced
- 4 Mushrooms, sliced (shiitake or any kind you prefer)
- 1 small Leek, washed and sliced
- 1 Tbsp Ginger, grated
- 2 Garlic Cloves, minced
- 1 Tbsp Olive Oil (instead of sesame oil)
- Salt (to taste)
- Pepper (to taste)
- Optional: Green onions for garnish

Instructions:

1. *Prepare the Vegetables:* Heat the olive oil in a large pot over medium heat. Add the garlic and ginger, sauté for about a minute, then add carrots, celery, mushrooms, and leeks. Cook for 5–7 minutes until softened.
2. *Make the Soup:* Pour in the vegetable broth, bring to a boil, then simmer for 10 minutes. Season with salt and pepper.
3. *Cook the Udon:* In a separate pot, cook the noodles according to package instructions. Avoid overcooking.
4. *Assemble:* Divide the noodles into bowls. Ladle the hot broth and vegetables over them.

5. *Garnish and Serve*: Top with green onions if desired. Enjoy!

Feel free to adjust the vegetables based on what's available or your preferences. This recipe is quite forgiving and customizable. Would you like assistance with finding any of these ingredients or have other questions?

At this point the user may choose to request a shopping list to be generated. Generating the shopping list involves first fetching the user's chat history. We send that history to GPT and prompt it to generate a structured list of ingredients for the user with price estimates for each item. Next, that list is sent to the perplexity API which we use to search the web for stores carrying the items in the list. Finally, the two responses are merged into one object containing a list of ingredients and the stores where they can be found. Each time the user generates a shopping list, we use GPT to dynamically update the user's preferences and write those updates back to Firebase. This makes sure that the user's preferences stay up to date as they use the app.

The final two parts of our system are the Google Maps API integration for navigation and stripe for checkout. We use google maps for routing and navigation in our UI. Addresses are fetched from the shopping list object stored in MongoDB and used for navigation through Google Maps API. On the other hand we have a dummy checkout session built with stripe, which is meant to simulate how future versions of the app would let you checkout from the app itself.

In summary, the frontend is developed using React Native, enabling cross-platform functionality for both Android and iOS devices. It communicates with the backend through a RESTful API hosted on EC2. The backend is built using FastAPI, integrating Firebase Firestore as the main data store for user preferences and MongoDB for storing chat history and shopping lists. Firebase Firestore allows for real-time updates and ensures that user preferences are dynamically updated based on interactions with the system.

float

3 Results

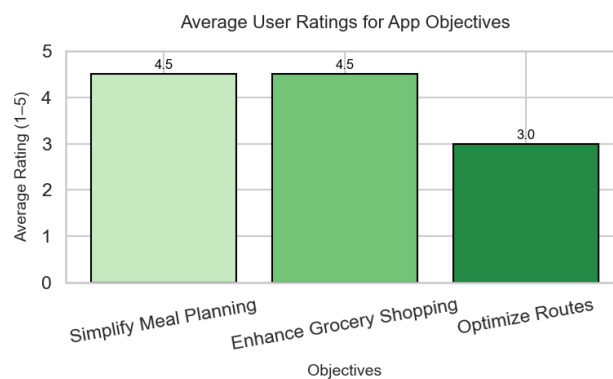


Figure 3: Average results so far from survey responses.

We evaluated SmartCart on 3 objectives by surveying our users on each one's performance.

1. **Simplify Meal Planning:** Did SmartCart make planning meals easier?
2. **Enhance Grocery Shopping:** Did SmartCart enhance or expedite your grocery shopping experience?
3. **Optimize Routes:** Did SmartCart make it easier to plan and execute your shopping trips?

Refer to figure 3 to see our current results. Figure 3 presents the average survey responses evaluating SmartCart on three key objectives: simplifying meal planning, enhancing grocery shopping, and optimizing routes. The survey collected numerical ratings from users on a 1 to 5 scale, with higher scores indicating better performance. The average response shows roughly 4.5 in meal planning, 4.5 in enhanced grocery shopping, and 3.0 in route optimization.

We ran one extra survey to make sure that dietary restrictions were considered accurately in model outputs. Of the responses we received 100% reported success.

4 Discussion

SmartCart seems to successfully simplify meal planning and enhance grocery shopping, while it does not necessarily make route optimization better. The evaluation of our meal-planning assistant highlights its effectiveness in simplifying meal selection and enhancing the grocery shopping experience.

The evaluation results demonstrate that our meal-planning assistant effectively streamlines meal selection and grocery shopping for most users. The integration of Firebase Firestore, FastAPI, and Google Maps API enabled a seamless experience, with meal recommendations and shopping lists tailored to user preferences. Compared to existing meal-planning applications such as *Mealime* and *Paprika*, our system offers a higher degree of personalization and convenience by incorporating real-time store location data. While AI-driven recommendation systems, such as *IBM Watson Meal Planner*, provide personalized meal suggestions, they lack the direct integration with grocery stores and automated shopping list generation that our system provides.

Despite these advantages, some challenges remain. Users in less densely populated areas experienced difficulty finding relevant grocery store recommendations due to limited store data, a limitation also observed in prior studies on location-based retail applications. While the system aimed to optimize cost efficiency, the absence of real-time store pricing sometimes led to inaccurate cost estimates, a common issue in online grocery shopping platforms that rely on static pricing models. Additionally, some users expressed interest in more flexible meal customization options, such as ingredient substitutions and dietary-specific meal plans, indicating a need for greater adaptability in future iterations.

To further improve the system, several enhancements can be considered. Implementing real-time pricing APIs from grocery retailers could improve cost estimation accuracy, addressing a key limitation of the current approach. Expanding the ingredient substitution

feature would provide users with greater flexibility in adjusting their meal plans based on availability and dietary preferences. Additionally, increasing store data coverage in rural areas through partnerships with regional grocery chains could enhance the system’s usability in underserved locations. Another promising direction is improving the transparency of recommendation logic, allowing users to understand why certain meal suggestions are made, which could improve trust and user engagement.

Overall, the SmartCart system demonstrates strong potential as a practical solution for meal planning and grocery shopping. While it effectively personalizes meal recommendations and optimizes grocery store selection, addressing the identified limitations in future iterations will enhance its reliability and user satisfaction. Future work should also explore machine learning-driven optimization techniques for balancing price, distance, and ingredient availability in real time, further refining the system’s ability to meet user needs.

5 Conclusion

This project presents an innovative meal-planning LLM agent that significantly enhances the user experience by streamlining meal selection, shopping list generation, and grocery store recommendations.

The evaluation results demonstrate that the system successfully simplifies meal planning and grocery shopping, with a majority of users reporting improved efficiency in selecting meals, generating shopping lists, and planning store visits. The ability to provide personalized meal recommendations and optimize grocery store selection was particularly well-received. While the majority of users found meal suggestions relevant, some desired greater flexibility in ingredient substitutions and increased transparency in recommendation logic.

Overall, the results validate the system’s potential as a practical and intelligent solution for meal planning and grocery shopping. By addressing the identified limitations and incorporating user-driven improvements, future iterations of the assistant could provide an even more seamless and adaptable experience, catering to a broader range of users and shopping needs.

References

- [1] Brown, T., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. *NeurIPS 2020*.
- [2] FastAPI. (2023). FastAPI Documentation. Retrieved from: <https://fastapi.tiangolo.com/>
- [3] Firebase. (2023). Cloud Firestore Documentation. Retrieved from: <https://firebase.google.com/docs/firestore>
- [4] Google Developers. (2023). Google Maps Platform Documentation. Retrieved from: <https://developers.google.com/maps/documentation>

- [5] Smith, J., & Brown, L. (2021). AI-Driven Meal Planning: Enhancing Efficiency in Food Selection and Shopping. *Journal of Computational Consumer Science*, 14(3), 112-130.