PantryCraft

Senior Project by **Winton Gee**

California Polytechnic State University, San Luis Obispo

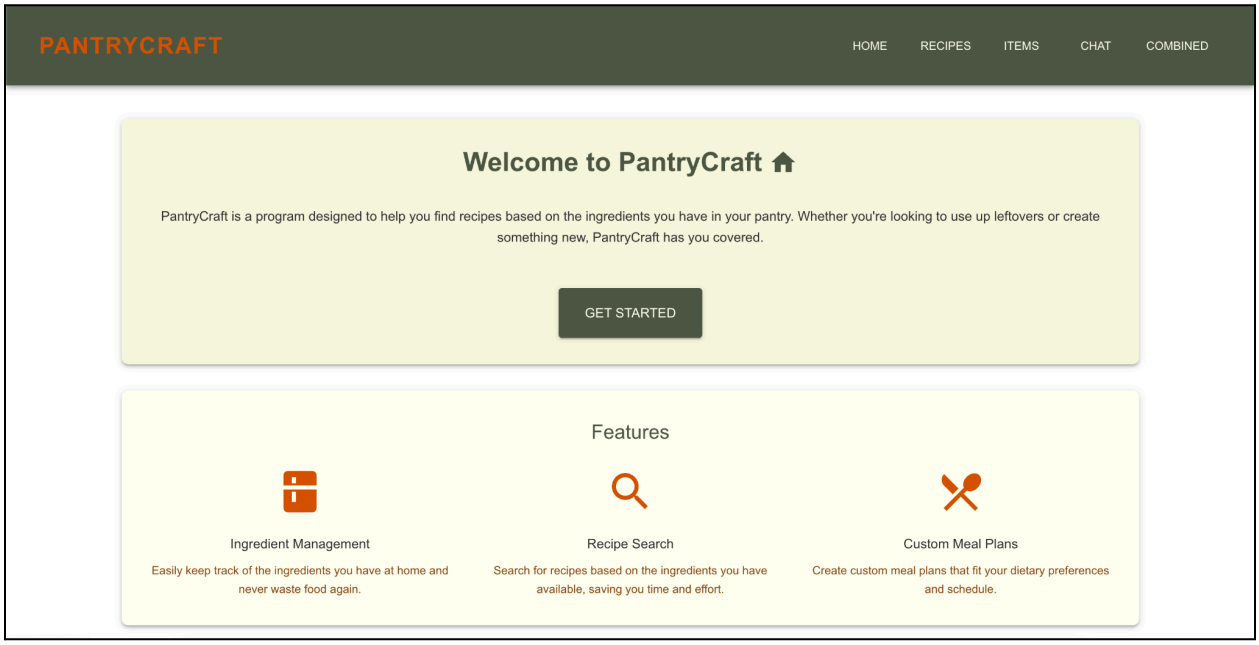
Computer Science & Software Engineering Department

Advised by **Joydeep Mukherjee**

Abstract

PantryCraft is a web application designed to address common problems related to meal planning and ingredient management. Many individuals and families struggle with deciding what to cook and keeping track of their ingredients, often leading to food spoilage and waste. PantryCraft aims to solve these issues by helping users manage their pantry, fridge, and other storage areas efficiently, offering recipe suggestions based on available ingredients. The application leverages MongoDB for backend storage, React.js for the frontend interface, and the OpenAI API for generating recipe suggestions through an interactive chat interface.

The development of PantryCraft was structured over two quarters. The first quarter focused on building the core functionalities, integrating essential technologies, and experimenting with computer vision for ingredient recognition.The second quarter emphasized refining the application, improving code quality, and enhancing the user interface to ensure a seamless and intuitive experience. Several challenges were encountered, including selecting appropriate tools, balancing ambitious goals with project timelines, and developing frontend skills. Despite these challenges, PantryCraft successfully achieved its primary objectives. The application provides a practical solution for ingredient management and meal planning while offering a unique and engaging user experience through its AI-driven recipe suggestions.

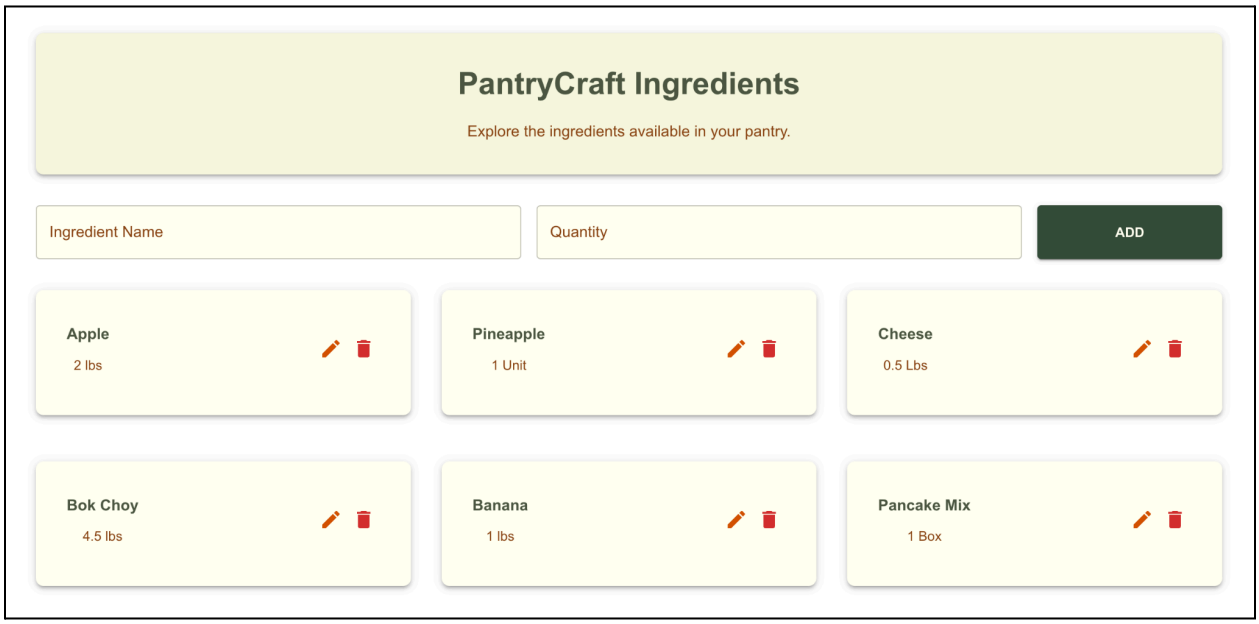


***Home** page for PantryCraft*

Introduction

PantryCraft was conceived out of a personal necessity and a desire to solve a widespread problem. Often, I found myself struggling with deciding what to cook for dinner, which was further complicated by my inability to keep track of the ingredients I had at home. There were countless instances where I discovered spoiled food in my fridge or pantry, simply because I had forgotten about it. This not only led to unnecessary food waste but also added to my frustration during meal planning.

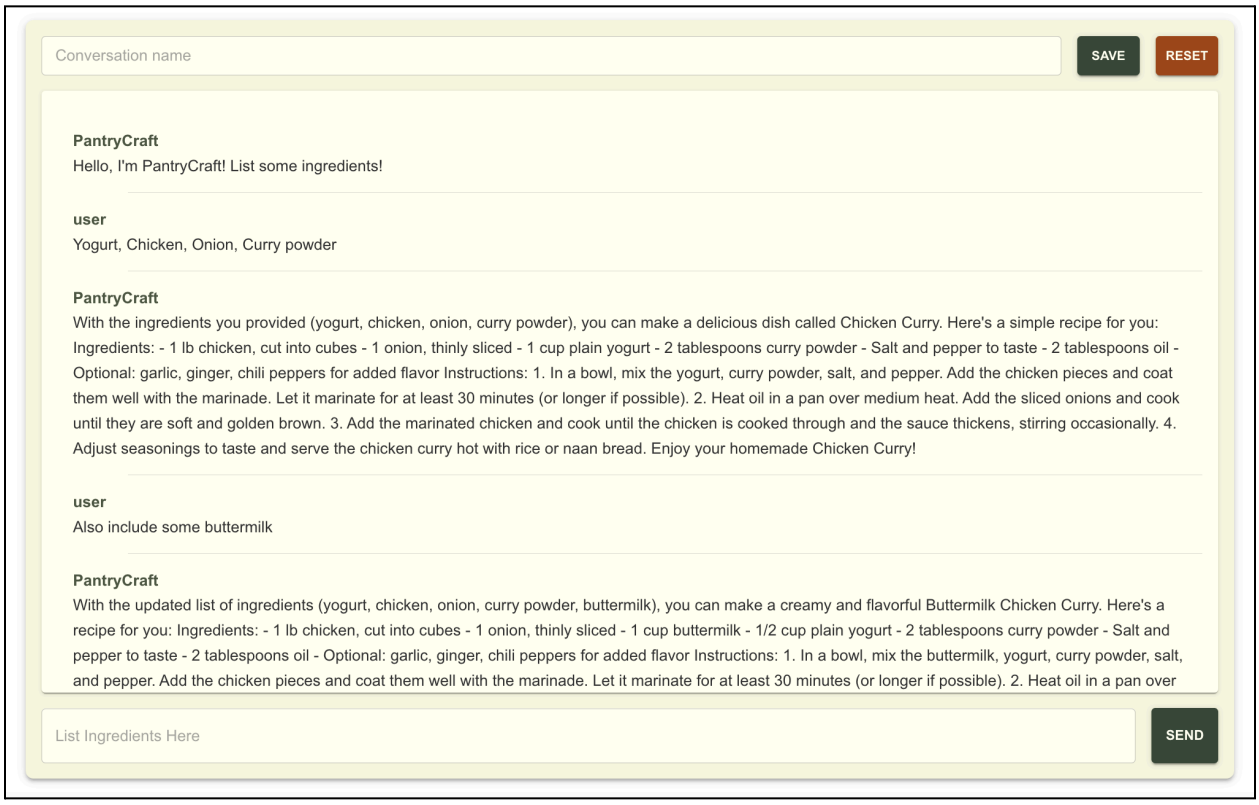
Recognizing that this is a common issue for many people, I envisioned PantryCraft as a solution that could benefit a wide range of users, from individuals and families to even small restaurants. The primary goal of this project is to develop a system that can efficiently track ingredients and suggest recipes based on the available items. By doing so, PantryCraft aims to simplify meal planning and reduce food waste.



***Items** page for PantryCraft*

One of the unique features of PantryCraft is its integration with the OpenAI API, which enables the application to generate recipe suggestions through a chat interface. This interactive component adds a layer of convenience and personalization, making the application more user-friendly. Additionally, the use of MongoDB for backend storage allows for efficient data management, while React.js provides a dynamic and responsive frontend experience.

The choice of these technologies was driven by their robustness and suitability for the project's needs. MongoDB offers flexibility in handling diverse data types, which is crucial for managing ingredient lists and recipes. React.js, known for its component-based architecture, facilitates the development of a scalable and maintainable user interface. OpenAI's powerful natural language processing capabilities enhance the recipe suggestion feature, providing users with relevant and creative meal ideas.

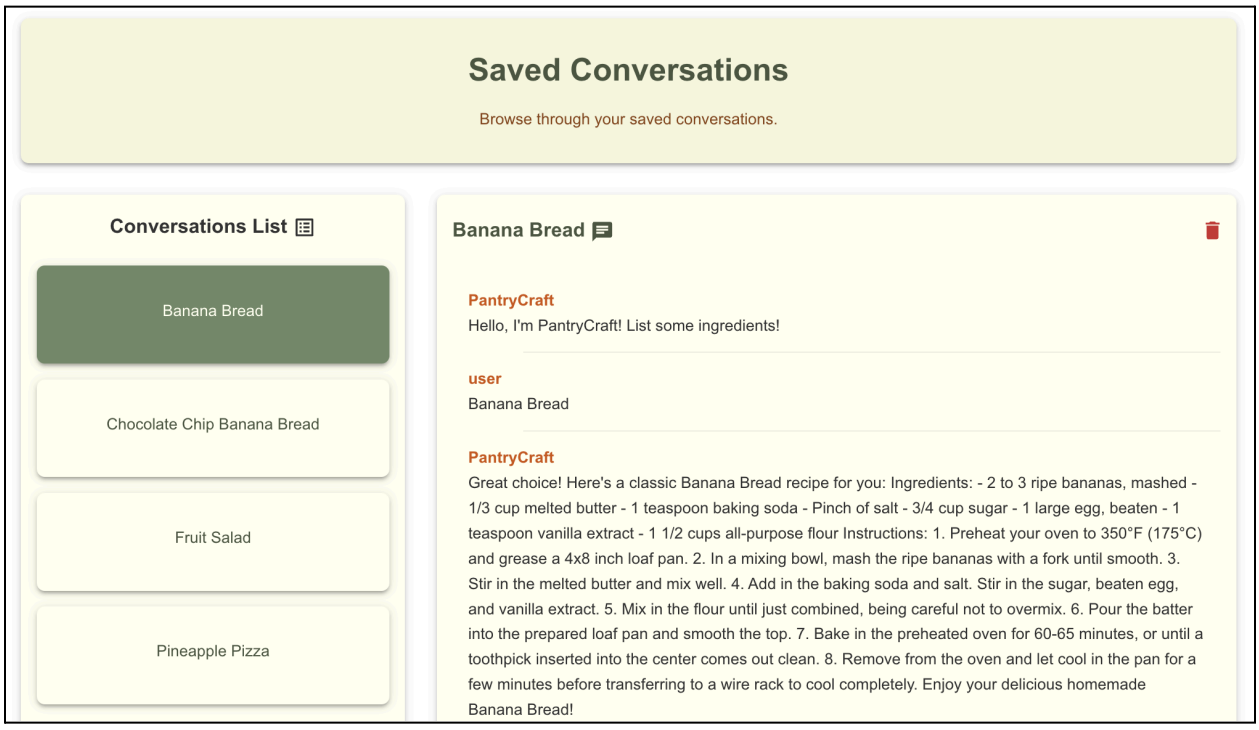


Chat *component for PantryCraft*

The project idea evolved significantly from its initial concept. Originally, I considered incorporating computer vision technology to automatically recognize ingredients using TensorFlow. However, due to the limitations of open-source datasets and the extensive time required for training custom models, this feature was deferred for future development. Instead, the focus shifted to perfecting the core functionalities and ensuring a seamless user experience.

Throughout the development process, I sought feedback from peers and friends, which was instrumental in refining the application. Their insights highlighted the importance of an intuitive user interface and the need for reliable ingredient tracking. Incorporating their suggestions, I made several adjustments to enhance the overall usability of PantryCraft.

My personal motivation for this project stemmed from a desire to create a practical solution to a recurring problem. The moment that sparked the idea was when I had to discard a significant amount of food that had gone bad. This experience made me realize the need for a system that could help people manage their ingredients better and make informed decisions about their meals.



Recipes *page for PantryCraft*

Project in detail

Work Done in Quarter 1

In the first quarter, PantryCraft's development was driven by a strong focus on laying a solid foundation for its core functionalities. The initial phase involved brainstorming and testing a variety of tools to identify the best technologies for the project. The main objective was to utilize modern and cutting-edge tools to create a robust and efficient system. During this period, I dedicated significant effort to building the application's backend and frontend.

For the backend, I chose MongoDB due to its flexibility in handling diverse data types, which is essential for managing ingredient lists and recipes. MongoDB's document-oriented approach provided the necessary scalability and performance for the application. The frontend was developed using React.js, a popular JavaScript library known for its component-based architecture. This choice allowed for the creation of a dynamic and responsive user interface that can be easily maintained and scaled.

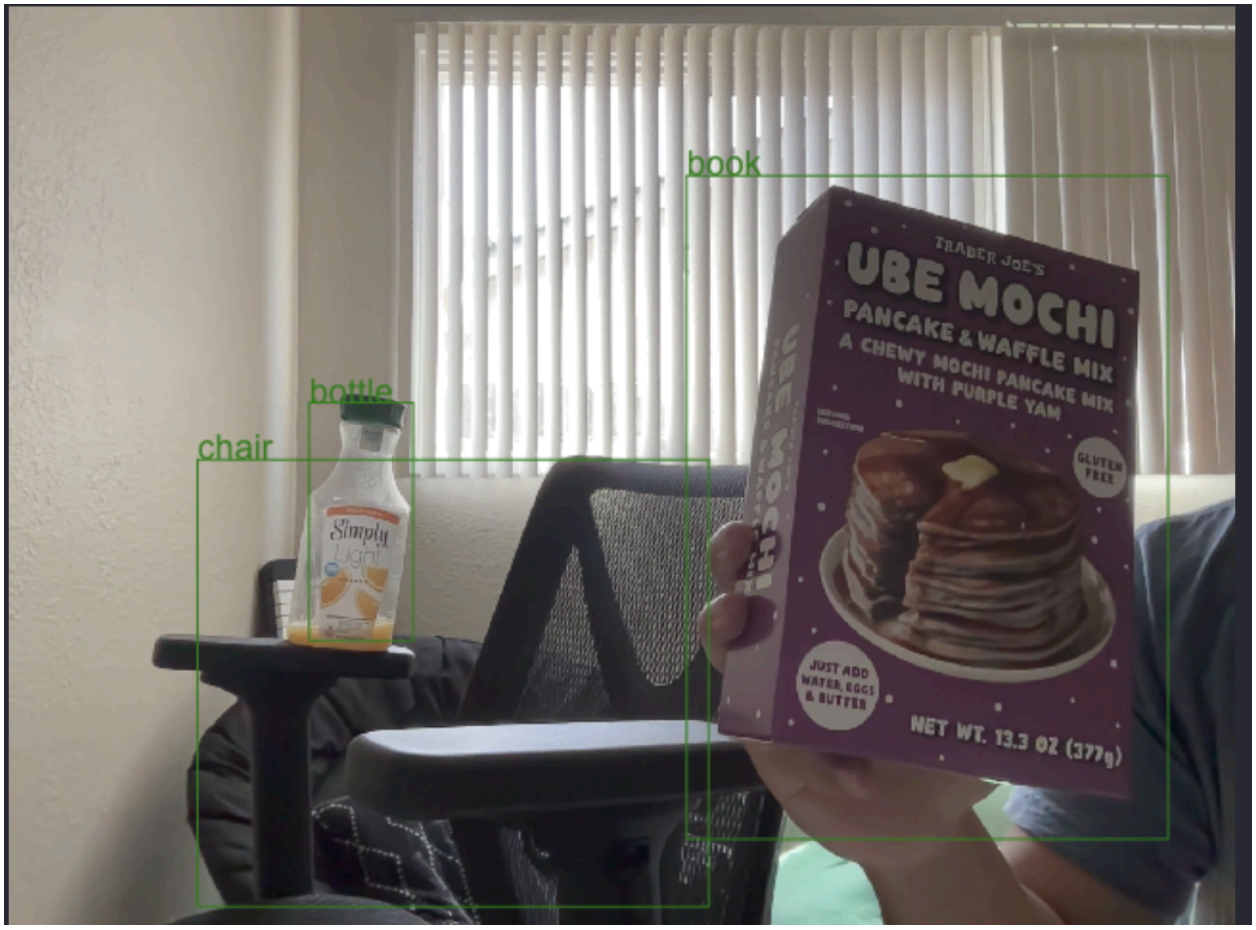
```
// Get a client object for the ingredient item collection
const ingredientItemCollection = useCollection({
  cluster: dataSourceName,
  db: "ingredient",
  collection: "Item",
});
```

***Ingredient** collection*

```
_id: ObjectId('663329438377f469aa3d5ed9')
summary : "Bok choy"
owner_id : "65d910075abb21f072acb302"
quantity : "4.5 lbs"
```

Data stored in MongoDB

During this quarter, I also experimented with implementing computer vision technology using TensorFlow. The goal was to enable users to add ingredients to the system by simply scanning them, thereby eliminating the need for manual entry. However, I encountered several challenges with the accuracy of open-source datasets and the extensive time required for training custom models. Despite these setbacks, the experience provided valuable insights, and the feature was earmarked for future development.



Computer Vision using Tensorflow with an open-source model

One of the most innovative aspects of PantryCraft is its integration with the OpenAI API. This feature allows users to interact with a chat interface to receive recipe suggestions based on the ingredients they have on hand. The AI-driven chat component adds a layer of personalization and convenience, making meal planning more enjoyable and less stressful.

```
// Handles sending the message to OpenAI
async function processMessageToChatGPT(chatMessages) {
  let apiMessages = chatMessages.map((messageObject) => {
    let role = messageObject.sender === "PantryCraft" ? "assistant" : "user";
    return { role, content: messageObject.message };
  });

  const apiRequestBody = {
    model: "gpt-3.5-turbo",
    messages: [systemMessage, ...apiMessages],
  };

  try {
    const response = await fetch(
      "https://api.openai.com/v1/chat/completions",
      {
        method: "POST",
        headers: {
          Authorization: "Bearer " + API_KEY,
          "Content-Type": "application/json",
        },
        body: JSON.stringify(apiRequestBody),
      }
    );

    const data = await response.json();
    setMessages([
      ...chatMessages,
      { message: data.choices[0].message.content, sender: "PantryCraft" },
    ]);
    setIsTyping(false);
  } catch (error) {
    console.error("Error processing message to OpenAI:", error);
    setError("Error sending message. Please try again.");
    setIsTyping(false);
  }
}
```

Core Functionality of PantryCraft: ***Chat Component***

The first quarter was primarily about building the functionality of the project. This involved setting up the backend and frontend, integrating the OpenAI API, and experimenting with computer vision. Each of these tasks required a deep understanding of the chosen technologies and a commitment to overcoming the challenges encountered along the way.

Work Done in Quarter 2

The second quarter of PantryCraft's development was dedicated to refining and enhancing the project. This phase was crucial for transforming the initial functional prototype into a polished and user-friendly application. The focus was on code cleanup, bug fixing, and frontend development. I began by improving the readability and maintainability of the code. This involved refactoring the existing codebase, fixing bugs, and optimizing various components. Ensuring that the code was clean and well-documented was essential for future scalability and ease of maintenance.

```jsx
function App() {
  return (
    <div className="App">
      <AppBar position="sticky" className="app-bar">
        <Toolbar>
          <AppName />
          <Tabs>
            <Tab label="Home" to="/" component={Link} />
            <Tab label="Recipes" to="/recipes" component={Link} />
            <Tab label="Items" to="/items" component={Link} />
            <Tab label="Chat" to="/chat" component={Link} />
            <Tab label="Combined" to="/combined" component={Link} />
          </Tabs>
        </Toolbar>
      </AppBar>

      <Routes>
        <Route path="/" element={<HomePage />} />
        <Route path="/recipes" element={<RecipePage />} />
        <Route path="/items" element={<ItemsPage />} />
        <Route path="/chat" element={<ChatPage />} />
        <Route path="/combined" element={<CombinedPage />} />
      </Routes>
      <Typography
        variant="body2"
        sx={{
          textAlign: "center",
          p: 2,
          color: "text.secondary",
          mt: 4,
        }}
      >
        &copy; {new Date().getFullYear()} PantryCraft. All rights reserved.
      </Typography>
    </div>
  );
}
```

App.js: *Main React Component*

A significant portion of this quarter was devoted to enhancing the frontend of the application. Given my limited experience with frontend development, this was a challenging yet rewarding task. I spent considerable time learning and implementing best practices for creating a responsive and intuitive user interface. This involved exploring various tutorials, experimenting with different design patterns, and integrating feedback from initial users.

The frontend development process included refining the layout, improving the navigation, and ensuring that the application was visually appealing. I aimed to create a seamless user experience that would make it easy for users to track their ingredients and receive recipe suggestions. The interface was designed to be intuitive, with clear visual cues and easy-to-use features.

```
import { createTheme } from "@mui/material/styles";

// Define color constants
const DEEP_FOREST_GREEN = "#4B5842";
const RUSTIC_ORANGE = "#D35400";
const LIGHT_BEIGE = "#F5F5DC";
const IVORY = "#FFFFF0";
const CHARCOAL = "#333333";
const SADDLE_BROWN = "#8B4513";
const GOLDENROD = "#DAA520";
const LIGHT_GREY = "#E0E0E0";
const DARK_GREY = "#616161";

const theme = createTheme({
  palette: {
    primary: {
      main: DEEP_FOREST_GREEN,
      contrastText: IVORY,
      dark: "#354B3A",
      light: "#6F8A67",
    },
    secondary: {
      main: RUSTIC_ORANGE,
      contrastText: IVORY,
      dark: "#A84300",
      light: "#FF7F33",
    },
    background: {
      default: LIGHT_BEIGE,
      paper: IVORY,
    },
    text: {
      primary: CHARCOAL,
      secondary: SADDLE_BROWN,
    },
```

Theme.js: *Utilized for frontend consistency*

Sensitivity Analysis

PantryCraft, while innovative and functional, operates under a set of assumptions and faces several challenges and vulnerabilities that were carefully considered during its development. One of the primary assumptions is that users will manually input their ingredients into the system. This manual entry process is straightforward but relies heavily on user diligence and accuracy. Although attempts were made to integrate computer vision technology to automate this process, the current implementation assumes that users are willing to take the time to input their ingredients.

*Snippet of **Ingredient Input***

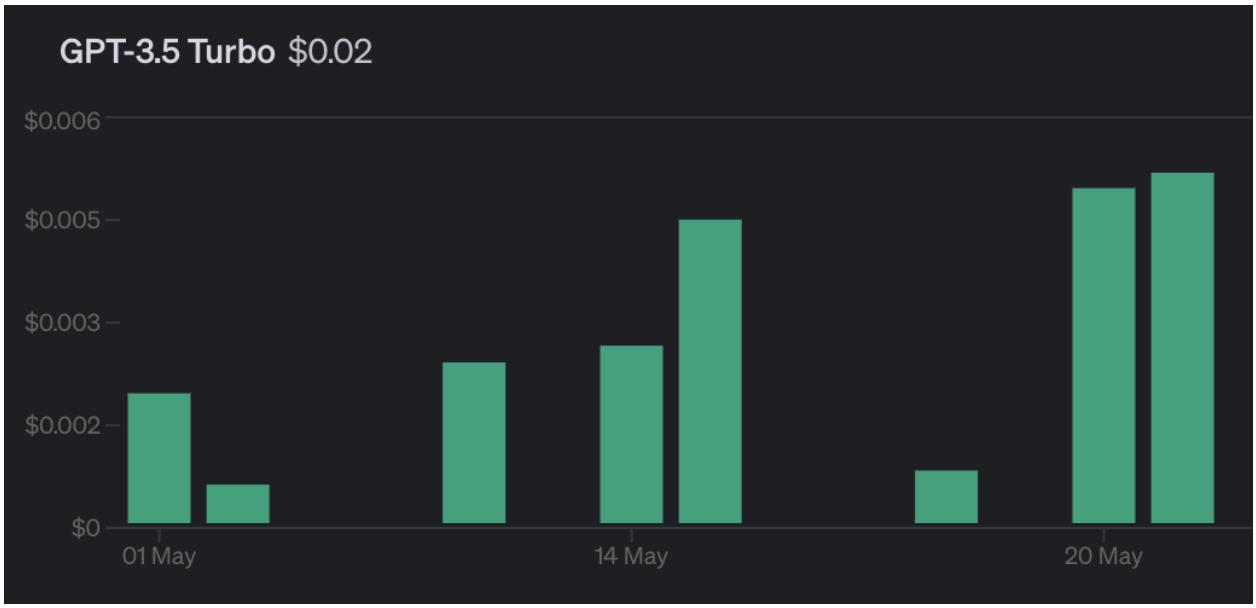
A notable vulnerability of PantryCraft lies in its reliance on the OpenAI API for generating recipe suggestions. While the API provides powerful and creative outputs, each call incurs a cost, making it potentially expensive to operate, especially if scaled for public use. This cost factor poses a significant challenge for widespread adoption and sustainability. Balancing the cost while maintaining the quality and responsiveness of the recipe suggestions was a constant consideration throughout the development.

The project also faced several technical and logistical challenges. One of the earliest challenges was deciding on the appropriate tools and technologies to use. The choice of MongoDB, React.js, and the OpenAI API was driven by their suitability for the project's needs, but each came with its learning curve. Integrating these technologies smoothly required careful planning and execution.

An ambitious aspect of the project was the attempt to incorporate computer vision for ingredient detection using TensorFlow. This feature aimed to simplify the user experience by allowing users to add ingredients through image recognition. However, the challenges associated with this feature, including the accuracy of open-source datasets and the time required for training custom models, proved to be significant. Consequently, this feature was deferred for future development, reflecting the reality of balancing ambitious goals with project timelines.

Personal challenges also played a role in the project's development. As a developer more comfortable with backend technologies, transitioning to frontend development required substantial effort. Learning new frameworks, understanding best practices, and implementing a user-friendly interface was time-consuming but ultimately rewarding. This journey underscored the importance of versatility and the willingness to step out of one's comfort zone to achieve project goals.

Despite these challenges, the development of PantryCraft was a valuable learning experience. It highlighted the importance of adaptability, resourcefulness, and strategic planning in bringing a complex project to fruition. The insights gained from tackling these challenges have not only strengthened the current implementation but have also laid a robust foundation for future enhancements and iterations of PantryCraft.



*Snippet of **OpenAI API Usage Costs***

Conclusion and Future Work

The journey of developing PantryCraft has been both challenging and rewarding, culminating in a functional and user-friendly application that meets its primary objectives. The application successfully tracks ingredients, suggests recipes based on available items, and integrates seamlessly with modern technologies like MongoDB, React.js, and the OpenAI API. The satisfaction of seeing PantryCraft come to life, from a conceptual idea to a tangible product, is immense.

PantryCraft has achieved its core functionality and provides a valuable tool for individuals and families looking to manage their ingredients more effectively and reduce food waste. The intuitive user interface and interactive chat component, powered by OpenAI, offer a unique and engaging user experience. Users can effortlessly track their pantry items and receive creative recipe suggestions, making meal planning simpler and more enjoyable.

Reflecting on the project, I am proud of the progress made and the solutions implemented to overcome various challenges. The decision to defer the computer vision feature was sensible, focusing instead on perfecting the core functionalities. This approach ensured that the application remained robust and reliable within the project timeframe. The experience underscored the importance of prioritizing essential features and being adaptable to change.

Looking ahead, there are several exciting opportunities for future work and enhancements to PantryCraft. One of the primary areas for improvement is the integration of computer vision technology. Starting with receipt detection and eventually moving to object recognition, this feature could significantly streamline the process of adding ingredients. Automating this step would enhance the user experience by reducing manual input and making the application even more convenient to use.

Another future enhancement involves refining the user interface further. While the current UI is functional and user-friendly, there is always room for improvement. Incorporating more user feedback, conducting usability tests, and iterating on the design could lead to a more polished and intuitive interface. Enhancements such as drag-and-drop functionality for organizing ingredients and interactive visualizations for recipe suggestions are potential areas to explore.

Additionally, expanding the functionality of PantryCraft to include more advanced features could increase its utility. For instance, integrating a shopping list feature that automatically updates based on depleted ingredients or missing items for selected recipes would be highly beneficial. Another idea is to incorporate meal planning and scheduling capabilities, allowing users to plan their meals for the week and receive reminders about ingredient usage and expiration dates.

Collecting feedback from a broader audience will also be crucial for future iterations. While initial feedback from peers and friends was invaluable, gaining insights from a diverse user base can provide new perspectives and highlight areas for further improvement. Engaging with users through surveys, beta testing, and community forums can help shape the development roadmap and ensure that PantryCraft continues to evolve in line with user needs.

In conclusion, PantryCraft has made significant strides towards its goal of simplifying meal planning and reducing food waste. The project's success is a testament to the careful planning, adaptability, and dedication invested in its development. With a solid foundation in place, the future holds promising opportunities for enhancing PantryCraft and making it an even more indispensable tool for home cooks and food enthusiasts. The journey has been enlightening, and the lessons learned will undoubtedly inform and inspire future projects.

Resources

Demo:
https://drive.google.com/file/d/1K-K9rlCdrVY7Eyx3IUjVMjyGRrRviBVY/view?usp=drive_link

Github Repo: https://github.com/WintonGee/PantryCraft

Backend

MongoDB - Database used for storing ingredients, recipes, and user data

Node.js - JavaScript runtime used for building the backend server and APIs

Express.js - Web application framework for Node.js used to create RESTful APIs for the application

Frontend

React.js - JavaScript library for building the user interface, chosen for its dynamic UIs

Material-UI - React component library used to implement Google's pre-designed components

React Router - Library used for routing in the React application, enabling navigation between different pages

APIs and External Services

OpenAI API - Used for generating recipe suggestions through an interactive chat interface. Provides AI-driven responses based on user-inputted ingredients

Axios - A promise-based HTTP client for making API requests, used for interacting with the OpenAI API and backend services

Experimentation with Computer Vision

TensorFlow - An open-source machine learning platform used for experimenting with computer vision tasks

OpenCV - An open-source computer vision library initially considered for ingredient recognition

Learning Tools

OpenAI

"How To Build Your Own AI With ChatGPT API." YouTube, uploaded by Web Dev Simplified, 2 June 2023, https://www.youtube.com/watch?v=4qNwoAAfnk4

"How To Create Custom GPTs - Build your own ChatGPT." YouTube, uploaded by Skill Leap AI, 10 May 2023, https://www.youtube.com/watch?v=5--JexprHuk

"Using ChatGPT with YOUR OWN Data. This is magical. (LangChain OpenAI API)." YouTube, uploaded by TechLead, 8 June 2023, https://www.youtube.com/watch?v=9AXP7tCI9PI

"The ChatGPT API Beginners Guide." YouTube, uploaded by warpdotdev, 1 April 2023, https://www.youtube.com/watch?v=7p7kJvckrFE

"Build A Chatbot With The ChatGPT API In React (gpt-3.5-turbo Tutorial)." YouTube, uploaded by Cooper Codes, 15 April 2023, https://www.youtube.com/watch?v=Lag9Pj_33hM

"Computer Vision with ChatGPT - Receipt Assistant Plugin." YouTube, uploaded by Andrej Baranovskij, 23 March 2023, https://www.youtube.com/watch?v=JyWQxpt99qo

Frontend

"How to add Icons in HTML Website | Font Awesome Icons | HTML CSS." YouTube, uploaded by Web Tech Knowledge, 24 April 2021, https://www.youtube.com/watch?v=7nJN12cHyv8

"React Tutorial." W3Schools, https://www.w3schools.com/react/

"CSS Grid Layout." W3Schools, https://www.w3schools.com/css/css_grid.asp

"Material-UI: React components for faster and easier web development." Material-UI, https://mui.com/material-ui/

Experimentation with Computer Vision

"Image Recognition AI App w/ REACT.JS and TENSORFLOW.JS | Beginners Javascript AI." YouTube, uploaded by Coder One, 2 July 2020, https://www.youtube.com/watch?v=SksXKCdeWkM

"Building an Object Detection App with Tensorflow.JS and React.JS in 15 Minutes | COCO SSD." YouTube, uploaded by Nicholas Renotte, 11 September 2020, https://www.youtube.com/watch?v=uTdUUpfA83s

"Training Your Own AI Model Is Not As Hard As You (Probably) Think." YouTube, uploaded by Steve (Builder.io), 20 February 2021, https://www.youtube.com/watch?v=fCUkvL0mbxI

"React TensorFlow AI Template and Simple Project." Stackademic Blog, https://blog.stackademic.com/react-tensorflow-ai-template-and-simple-project-6012071e2514