FridgeChef

Team Name: Dreyfoos Children

Team Members: Remington Ewing, Gabriela Taboada

Course: CIS 4914 Senior Project

Advisor: Michael Link

Advisor Email: Michael.Link@ufl.edu

YouTube Link to Final Presentation: LINK

Keywords: AI, Python, FastAPI, React, OpenAI, Prompting, SQLite, Web App

Project Overview

Project Name and Description

FridgeChef is an AI-powered web application that assists users in creating personalized recipes based on ingredients, nutrition, and kitchen tools they have readily available to them in college environments.It leverages OpenAI's *GPT 3.5-turbo* model to provide health-conscious, nutritionally informative recipes that consider the user's available ingredients, nonperishables, and kitchen tools. Our tool allows the user to input their own customizations and preferences into a prompt which allows an infinite amount of different outputs and personalization.

Goals and Objectives

- Reduce food waste for college students and promote healthier eating habits

- Make cooking more accessible to college students
- Build a recipe generation system powered by GPT-3.5.
- Enable user authentication and persistent ingredient/tool storage.
- Allow user experience to be personalized and data security for each account made.
- Implement ingredient tracking and automatic removal upon recipe acceptance.
- Support user recipe saving and review with interactive interface.
- Create a user-friendly experience that students feel comfortable using

Scope

FridgeChef focuses on frontend-to-backend integration using React and FastAPI, local SQLite-based storage, and OpenAI GPT for AI logic. Its scope includes recipe generation, ingredient/non-consumable inventory, and multi-user support via HTTPBasicAuth.

Team Member Roles

- **Remington Ewing:** Lead Developer (Backend Developer, API design, Database modeling, OpenAI integration)
- **Gabriela Taboada:** Designer and Front-End Developer (CSS design, Modeling, Wireframes)

Deliverables

- FastAPI backend with full CRUD functionality.
- React frontend with interactive chat UI.
- Ingredient and tool management interfaces.
- GPT-based recipe generation and acceptance flow.
- SQLite-based persistent user and recipe storage.

Current Project Status

Our project is fully functional as of 4/23/2025. We have fully integrated and finished AI integration alongside full support for multiple users, saving recipes, sharing recipes, ingredient and non-perishable management, and ensured security vulnerabilities have been taken care of.

Remaining Work & Known Issues

The project currently does not have anything that wasn't met within the bounds of the initial project proposal. We would have liked to reach some of our further stretch goals such as implementing a personalized exercise schedule and recommendations for users but due to the confines of the semester we were unable to reach this stretch.

Additionally, for the future it would be nice to make an additional machine learning implementation to this project where the AI learns the users preferences over time - giving them better initial recommendations. We believe our chatbot is currently ideal as-is though, because it gives the user the ability to put in their preferences each time ensuring the user is in full control.

Technical Details

Codebase Overview

- **Repository:** https://github.com/gabrielatabb/FridgeChef
- **Version Control:** Git
- **Branching Model:** Main branch is final version; also used throughout development.

Dependencies and Libraries

- **Backend:** FastAPI, SQLAlchemy, Uvicorn, python-dotenv
- **Frontend:** React, React Router, Vanilla CSS
- **API:** OpenAI GPT-3.5 API

Deployment Instructions

1. Clone repo and create *.env* with *OPENAI_API_KEY*.
2. Install Python packages: pip install fastapi uvicorn sqlalchemy python-dotenv openai
3. Run backend with *uvicorn main:app --reload*
4. Run frontend with *npm start* inside React project.

Database Schema

- Tables: *users, ingredients, non_consumables, generated_recipes, saved_recipes*
- Each linked by *user_id*

Software Architecture

Frontend: The application uses a single-page React interface that interacts with the backend through RESTful API calls. The React components include functionality for managing ingredients and tools, interacting with a recipe chatbot, and saving generated recipes. All interactions are state-managed using React Hooks.

Backend: The backend is developed using FastAPI, a modern Python web framework. It exposes various HTTP endpoints for authentication, storing and retrieving user data (ingredients and non-consumables), generating recipes using OpenAI's GPT API, and accepting or rejecting recipes. It also handles data validation, database sessions, and secure user authentication using HTTP Basic Auth.

Database: The application uses SQLite for local, lightweight relational data storage. The schema includes normalized tables: users, ingredients, non_consumables, generated_recipes, and saved_recipes, each connected by a user_id foreign key. This schema supports persistence and multi-user data segregation effectively.

Testing Information

Summary

Testing was conducted both manually through the frontend UI and using API testing tools like Postman and browser developer tools. The team tested the full cycle of user interactions including account registration, authentication, ingredient/tool management, recipe generation, and data persistence.

A significant portion of testing focused on edge cases, such as:

- Submitting empty fields
- Attempting to generate a recipe with no ingredients
- Testing unauthorized API access

In addition to manual UI testing, we performed backend-specific testing via:

- **Postman**: Used to validate each REST API endpoint with proper headers and body formats.
- **Logging and error tracing**: FastAPI logging was configured to monitor backend responses, errors, and exception handling.
- **Database integrity checks**: Verified creation, retrieval, and deletion for rows of SQLite database across tables (users, ingredients, non_consumables, generated_recipes, and saved_recipes).

Results

- 100% success rate across normal user workflows.
- GPT integration returned accurate and relevant responses with *valid* inputs.
- Edge cases such as blank input fields or missing credentials are caught and return descriptive error messages.
- Non-consumables are stored independently from ingredients and are not removed upon recipe acceptance, as intended.

Known Bugs & Issues

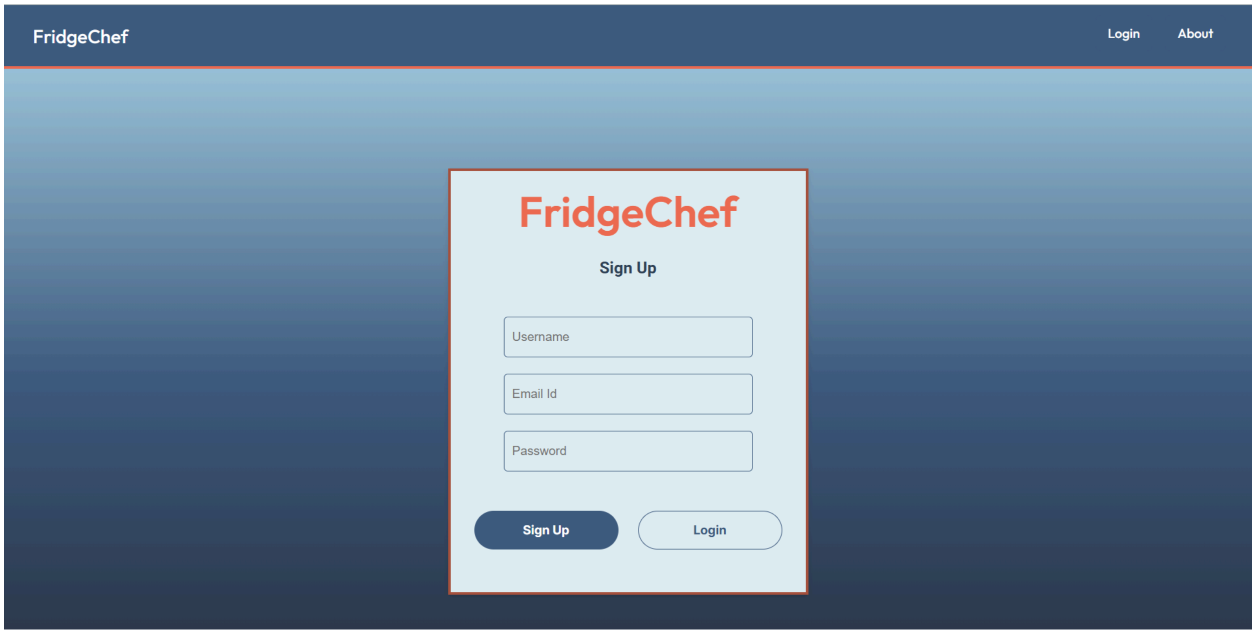
- Minor UI overlap in the saved recipe preview box when recipes are long.
- No JWT token/session management, meaning users are re-authenticated on each session.

- React desync can occur if tools or ingredients are added and deleted rapidly before the database has the chance to respond.

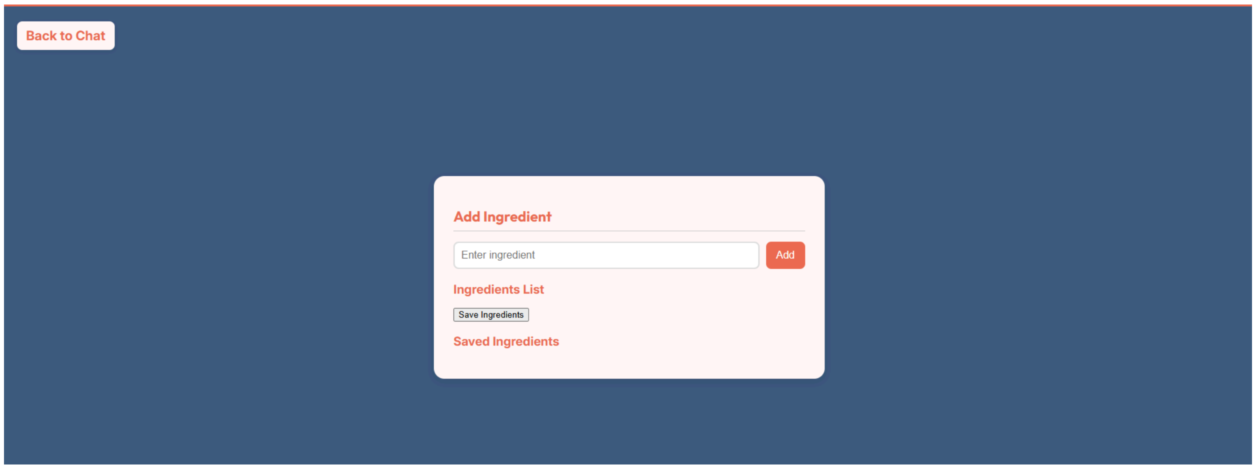
User Documentation

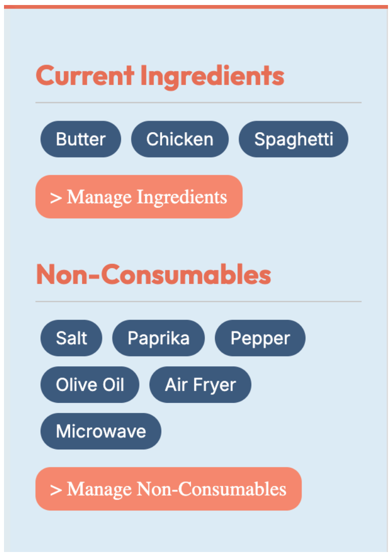
User Guide

FridgeChef is designed to offer an intuitive user experience while interacting with AI-powered recipe generation. Below is a step-by-step walkthrough of typical usage:

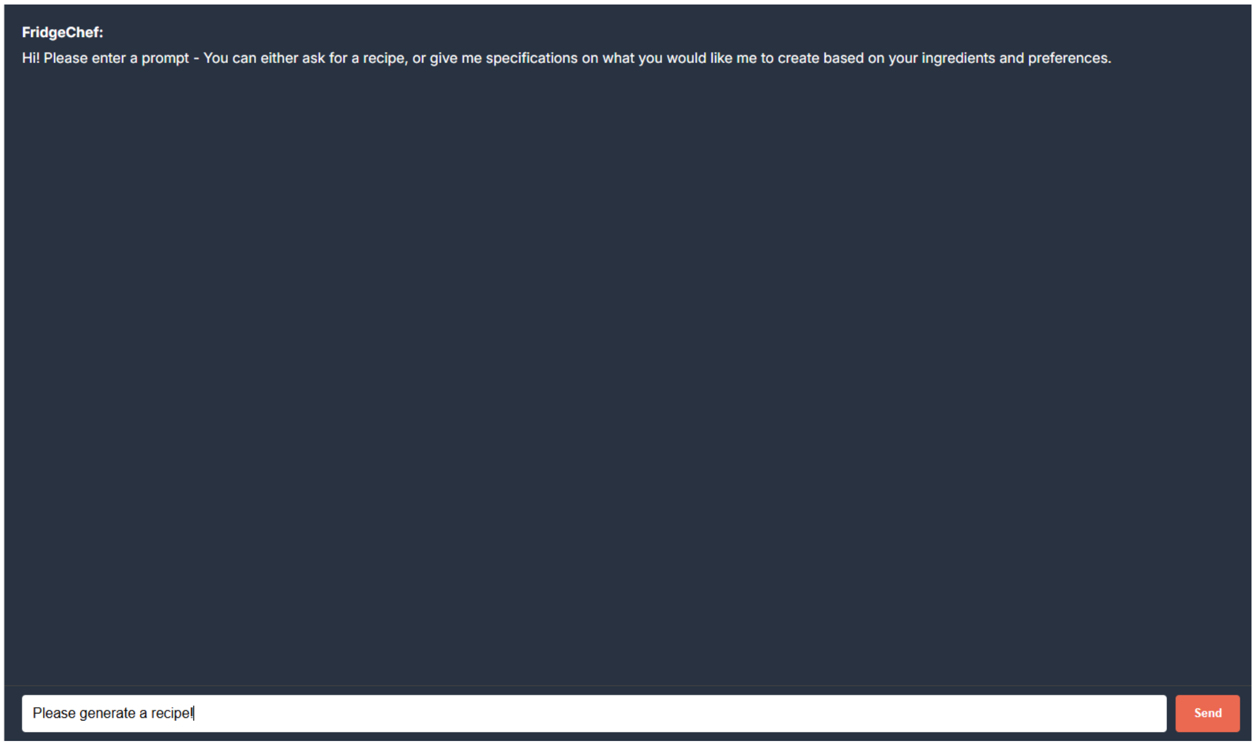


Account Registration: Users begin by creating an account with a unique username, email, and secure password. This enables personalized storage of ingredients, tools, and saved recipes.



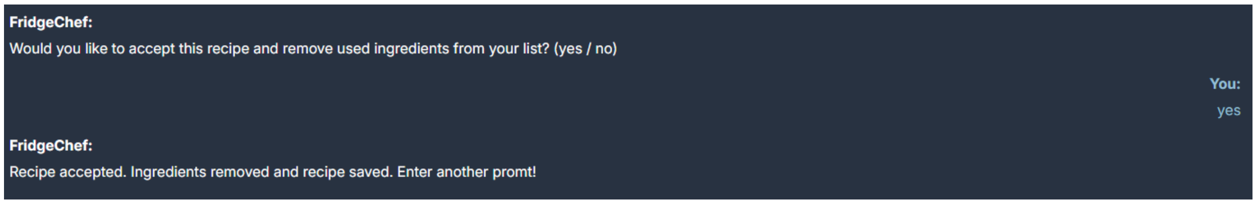


Ingredient and Tool Management: After logging in, users can add ingredients (e.g., chicken, rice) and non-consumables (e.g., frying pan, blender, salt and pepper) through dedicated interfaces. These items are stored in the database and are essential inputs for the recipe generator. These items can also be removed if they expire or are no longer available in the kitchen. After adding ingredients to inventory they will be generated on the chat page in the inventory column.


FridgeChef:
Hi! Please enter a prompt - You can either ask for a recipe, or give me specifications on what you would like me to create based on your ingredients and preferences.

Please generate a recipe!

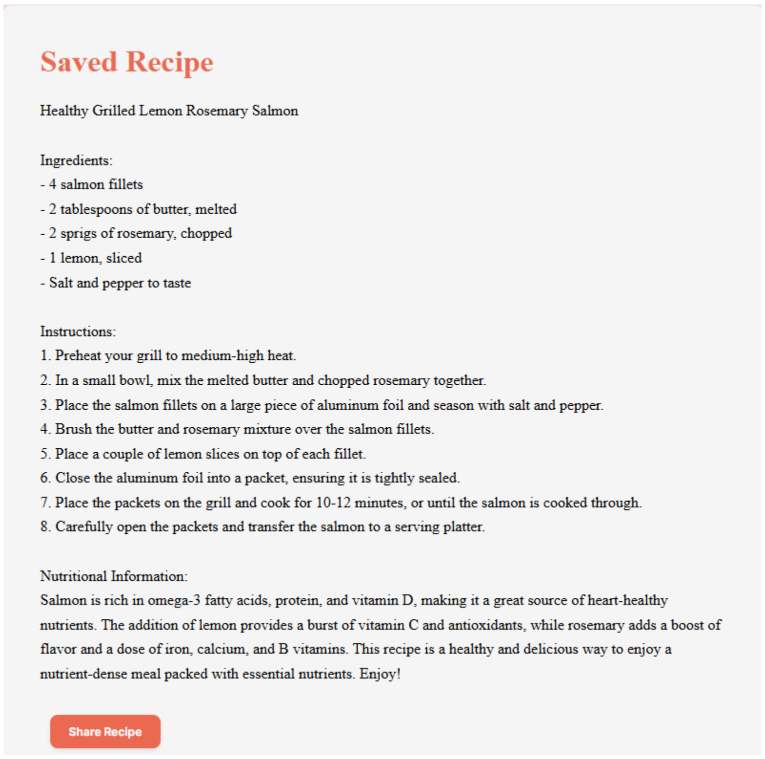
Generating a Recipe: Users can enter a meal idea or request (e.g., "Dinner with garlic and salmon") into the chat input. The system then invokes the GPT API to generate a recipe using only the saved ingredients and tools.


FridgeChef:
Would you like to accept this recipe and remove used ingredients from your list? (yes / no)

You:
yes

FridgeChef:
Recipe accepted. Ingredients removed and recipe saved. Enter another promt!

Accepting or Rejecting Recipes: Once a recipe is returned, users have the option to accept or reject it. If accepted, any ingredients mentioned in the recipe are automatically removed from the list, simulating real-life usage. The recipe is then added to the "Saved Recipes" panel for future access. Otherwise, the user is asked to request another recipe.

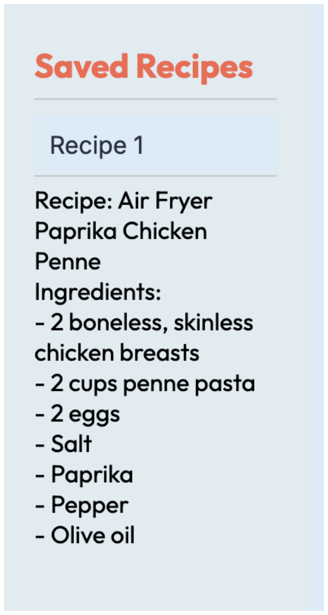

Saved Recipe

Healthy Grilled Lemon Rosemary Salmon

Ingredients:
- 4 salmon fillets
- 2 tablespoons of butter, melted
- 2 sprigs of rosemary, chopped
- 1 lemon, sliced
- Salt and pepper to taste

Instructions:
1. Preheat your grill to medium-high heat.
2. In a small bowl, mix the melted butter and chopped rosemary together.
3. Place the salmon fillets on a large piece of aluminum foil and season with salt and pepper.
4. Brush the butter and rosemary mixture over the salmon fillets.
5. Place a couple of lemon slices on top of each fillet.
6. Close the aluminum foil into a packet, ensuring it is tightly sealed.
7. Place the packets on the grill and cook for 10-12 minutes, or until the salmon is cooked through.
8. Carefully open the packets and transfer the salmon to a serving platter.

Nutritional Information:
Salmon is rich in omega-3 fatty acids, protein, and vitamin D, making it a great source of heart-healthy nutrients. The addition of lemon provides a burst of vitamin C and antioxidants, while rosemary adds a boost of flavor and a dose of iron, calcium, and B vitamins. This recipe is a healthy and delicious way to enjoy a nutrient-dense meal packed with essential nutrients. Enjoy!


Saved Recipes

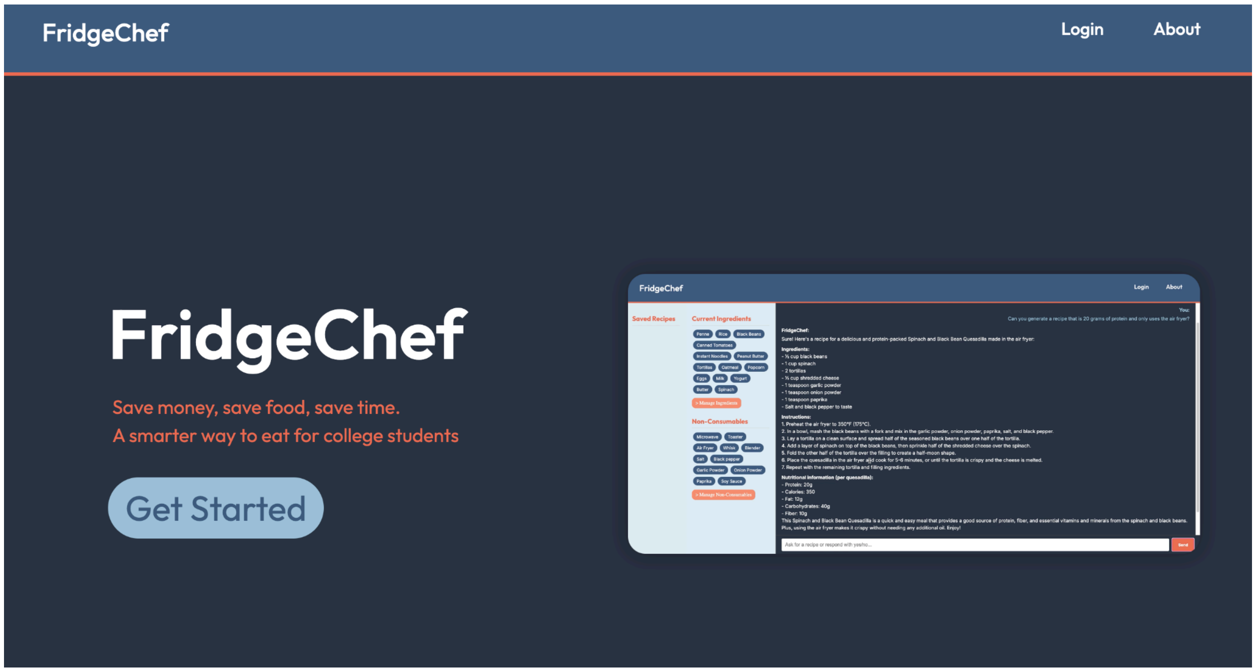
Recipe 1

Recipe: Air Fryer Paprika Chicken Penne
Ingredients:
- 2 boneless, skinless chicken breasts
- 2 cups penne pasta
- 2 eggs
- Salt
- Paprika
- Pepper
- Olive oil

Saved Recipe Review: Users can view previously accepted recipes in the sidebar. Clicking on a saved recipe opens a full preview. If a recipe is clicked, the user can view it on an easy-to-read webpage and can also share it.

Screenshots & Workflow Overview

- **Landing page:** Showcases the overview of the project, mission statement, and allows the user to get started. Additionally shows a video of the project, how-to-use and is the first page the user sees upon access.



You:
Can you generate a recipe that is 20 grams of protein and only uses the air fryer?

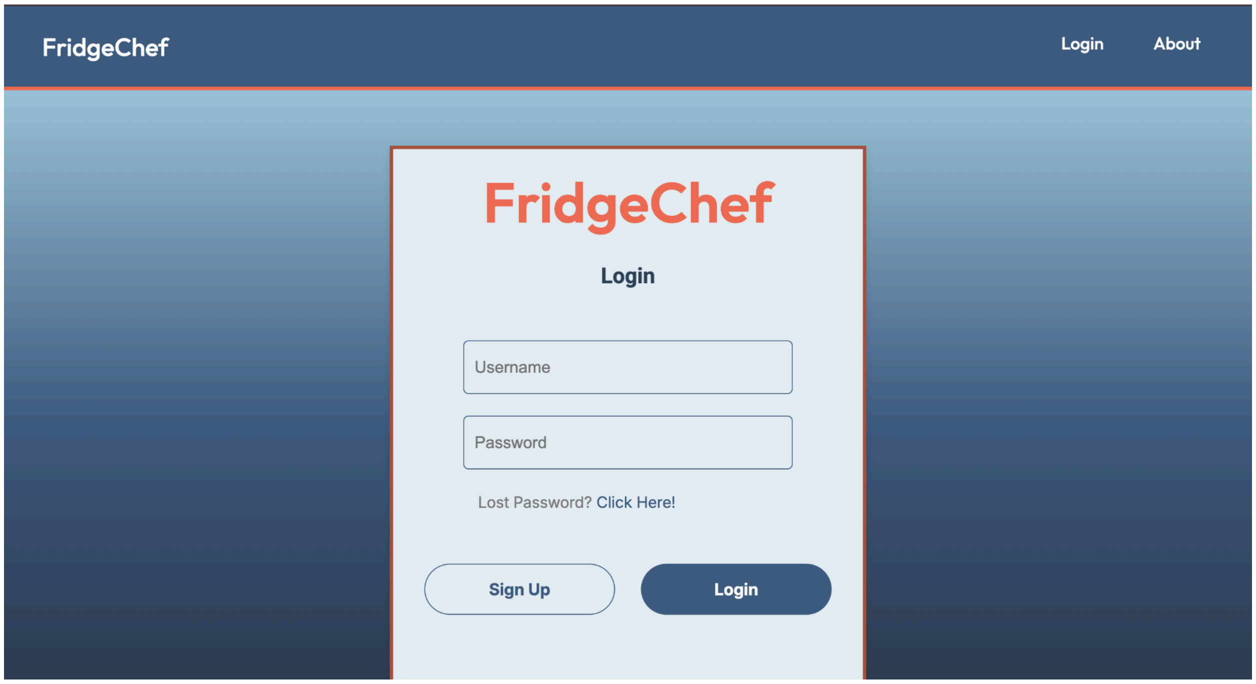
FridgeChef:
Sure! Here's a recipe for a delicious and protein-packed Spinach and Black Bean Quesadilla made in the air fryer:

Ingredients:
- ½ cup black beans
- 1 cup spinach
- 2 tortillas
- ½ cup shredded cheese
- 1 teaspoon garlic powder
- 1 teaspoon onion powder
- 1 teaspoon paprika
- Salt and black pepper to taste

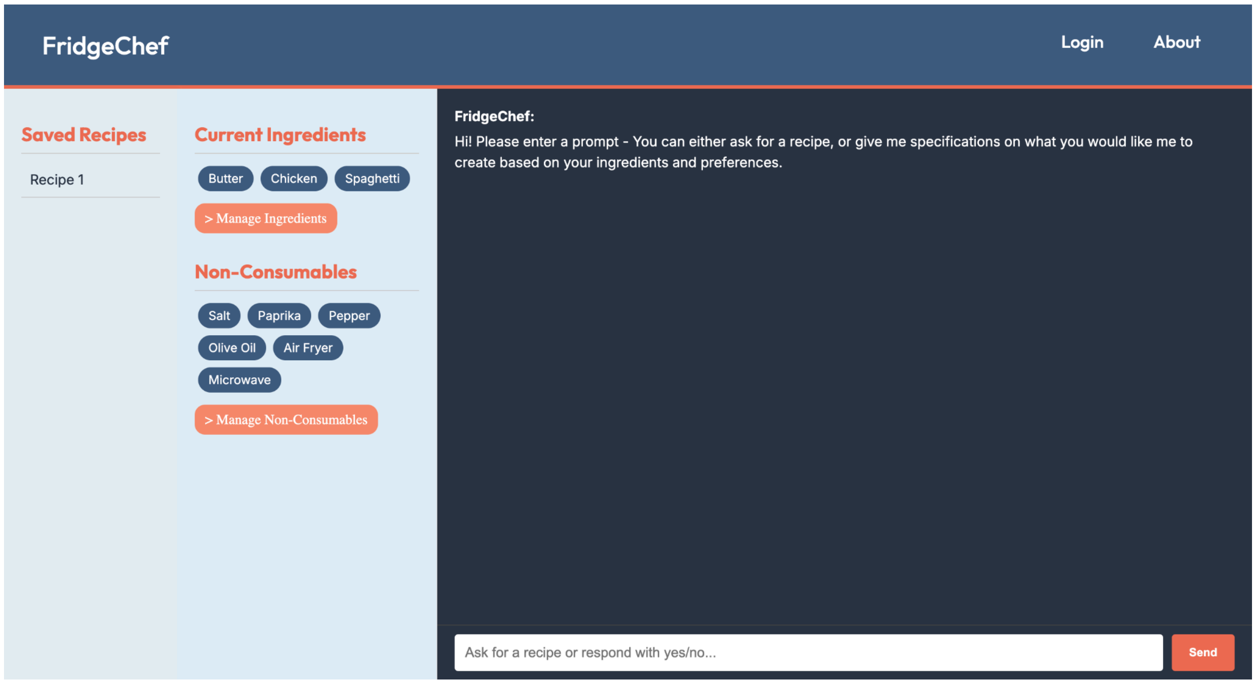
Instructions:
1. Preheat the air fryer to 350°F (175°C).
2. In a bowl, mash the black beans with a fork and mix in the garlic powder, onion powder, paprika, salt, and black pepper.
3. Lay a tortilla on a clean surface and spread half of the seasoned black beans over one half of the tortilla.
4. Add a layer of spinach on top of the black beans, then sprinkle half of the shredded cheese over the spinach.
5. Fold the other half of the tortilla over the filling to create a half-moon shape.
6. Place the quesadilla in the air fryer and cook for 5-6 minutes, or until the tortilla is crispy and the cheese is melted.
7. Repeat with the remaining tortilla and filling ingredients.

Nutritional information (per quesadilla):
- Protein: 20g
- Calories: 350
- Fat: 12g
- Carbohydrates: 40g
- Fiber: 10g
This Spinach and Black Bean Quesadilla is a quick and easy meal that provides a good source of protein, fiber, and essential vitamins and minerals from the spinach and black beans. Plus, using the air fryer makes it crispy without needing any additional oil. Enjoy!

- **Login page:** Allows the user to sign-up or login to their account.

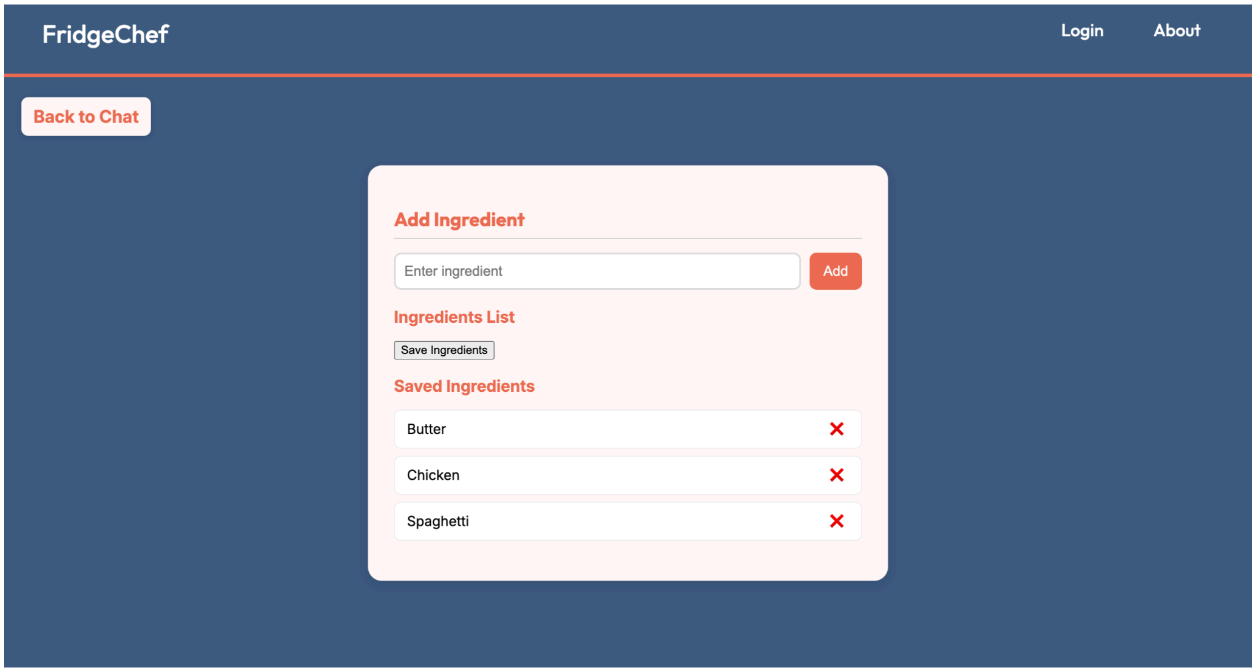


- **Chat Panel**: Presents a clean conversational interface between the user and the FridgeChef assistant, including GPT-generated recipe responses and system prompts.

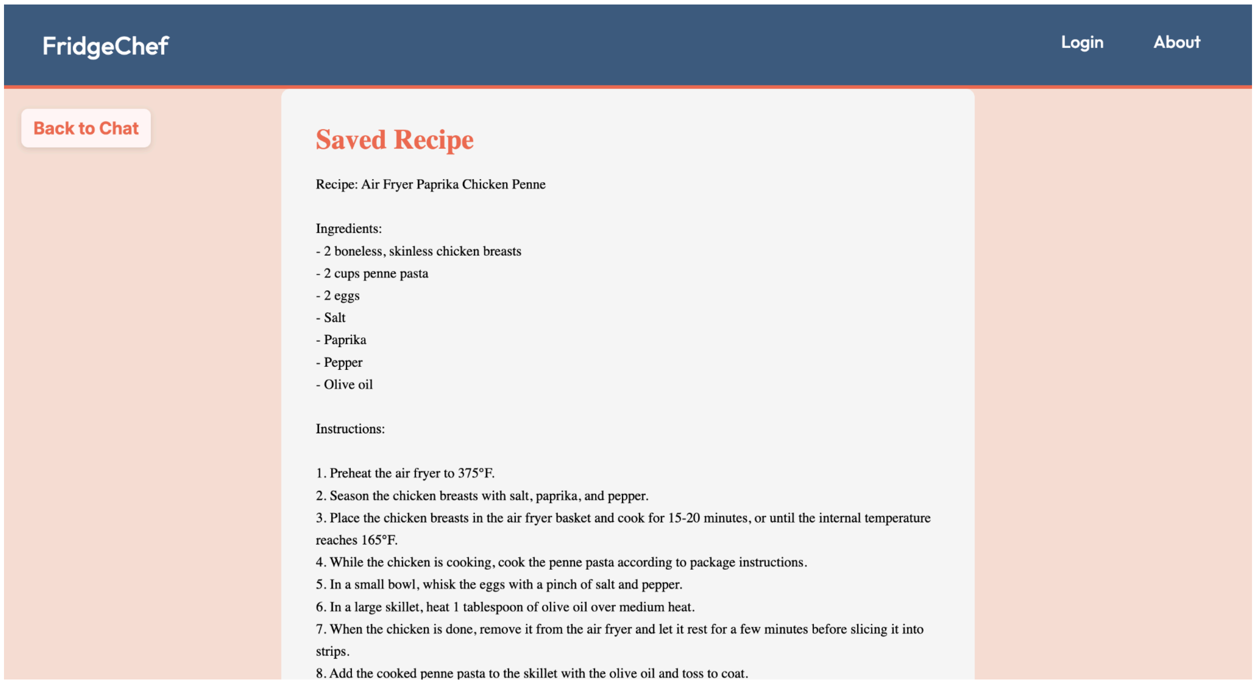


FridgeChef:
Hi! Please enter a prompt - You can either ask for a recipe, or give me specifications on what you would like me to create based on your ingredients and preferences.

- **Ingredient Panel**: Displays all currently stored ingredients and tools. Users can manage these directly from the interface using "Manage Ingredients" and "Manage Non-Consumables" links.



- **Saved Recipes Sidebar**: A persistent sidebar that lists all accepted recipes. Hovering displays a brief preview; clicking opens the full recipe view.



Saved Recipe

Recipe: Air Fryer Paprika Chicken Penne

Ingredients:
- 2 boneless, skinless chicken breasts
- 2 cups penne pasta
- 2 eggs
- Salt
- Paprika
- Pepper
- Olive oil

Instructions:

1. Preheat the air fryer to 375°F.
2. Season the chicken breasts with salt, paprika, and pepper.
3. Place the chicken breasts in the air fryer basket and cook for 15-20 minutes, or until the internal temperature reaches 165°F.
4. While the chicken is cooking, cook the penne pasta according to package instructions.
5. In a small bowl, whisk the eggs with a pinch of salt and pepper.
6. In a large skillet, heat 1 tablespoon of olive oil over medium heat.
7. When the chicken is done, remove it from the air fryer and let it rest for a few minutes before slicing it into strips.
8. Add the cooked penne pasta to the skillet with the olive oil and toss to coat.

Transition Plan / Next Steps

Knowledge Transfer

- Codebase documented with consistent naming and file organization.
- README includes full setup instructions.

Sustainability

- Plan to add login sessions with JWT.

- Explore hosting opportunities.

Acknowledgments

- We would like to give a special thank you to Michael Link, our project advisor. He was very understanding of our circumstances through the semester and was incredibly generous and fair with us. We are excited to see how he continues to grow his career at the University of Florida
- OpenAI for API access and AI generation.
- Python/FastAPI documentation.

Appendices

Appendix A: Code Fragment – GPT Prompt Template

```
prompt = (
    f"User request: {request.prompt}\n"
    f"Available ingredients: {ingredient_list}\n"
    f"Available tools and spices: {tool_list}\n"
    "Respond with a recipe that fits the request..."
)
```

Appendix B: API Endpoint Summary

- POST /register – Create new user
- POST /login – Authenticate user
- POST /store_ingredients – Save ingredient list
- POST /store_non_consumables – Save kitchen tools

- GET /get_ingredients – Retrieve current ingredients
- GET /get_non_consumables – Retrieve kitchen tools
- POST /generate_recipe – Use GPT to create recipe
- POST /accept_recipe – Confirm or reject generated recipe
- GET /get_saved_recipes – List past accepted recipes

Appendix C: Example User Flow

1. Register → Add ingredients → Add tools
2. Input: “I want a dinner with chicken and garlic”
3. GPT generates a recipe using current inputs
4. Accept → used ingredients removed and recipe saved

Appendix D: Environmental Variables Setup

```
OPENAI_API_KEY=your_key_here
```

Stored in .env, loaded via dotenv.

Biography

Remington Houston Ewing

A Computer Science major at the University of Florida, Remington has a strong interest in Backend Development, Automation, AI development, and Cybersecurity. His previous working experience includes working with special-needs students at the Agency for Persons with Disabilities - where he specialized in assisting students obtain Oracle and Google Licenses. Beyond this he also interned at A-1 Industries over the summer, where he worked on automating the company’s IT department and developing software for the roofing industry.

He is interested in seeking opportunities in Software Engineering with a strong interest in building tools that help at-risk groups. His personal interests include distance running, blacksmithing, playing guitar, and learning Italian.

Gabriela Elena Taboada

Gabriela Taboada will soon begin a full-time role in Miramar at Southern Glazer's Wine & Spirits as part of their new graduate IT rotational program. She previously interned at UKG, where she focused on backend software development. More recently, Gabriela has taken a growing interest in frontend development, inspired by her passion for art and design. Outside of work, she enjoys painting and discovering new digital illustration tools.