Recipe Curator

Introduction

The **Recipe Curator** is an advanced, AI-powered recipe recommendation platform designed to revolutionize how users discover and explore culinary options. By leveraging cutting-edge technologies such as Pinecone, OpenAI, Hugging Face, and Gemini, the platform provides personalized recipe suggestions tailored to user preferences and inputs. The project integrates natural language processing (NLP) to handle complex queries like "spicy Indian curries" or "Butter Pecan Cookies" and delivers contextually relevant results.

The system is built on a robust architecture combining a Streamlit frontend, a FastAPI backend, and cloud-based tools like AWS S3 for storage and Snowflake for structured data querying. With its scalable and maintainable design, the Recipe Curator not only enhances the user experience but also provides a seamless interface for users to explore, save, and manage recipes.

Problem Statement

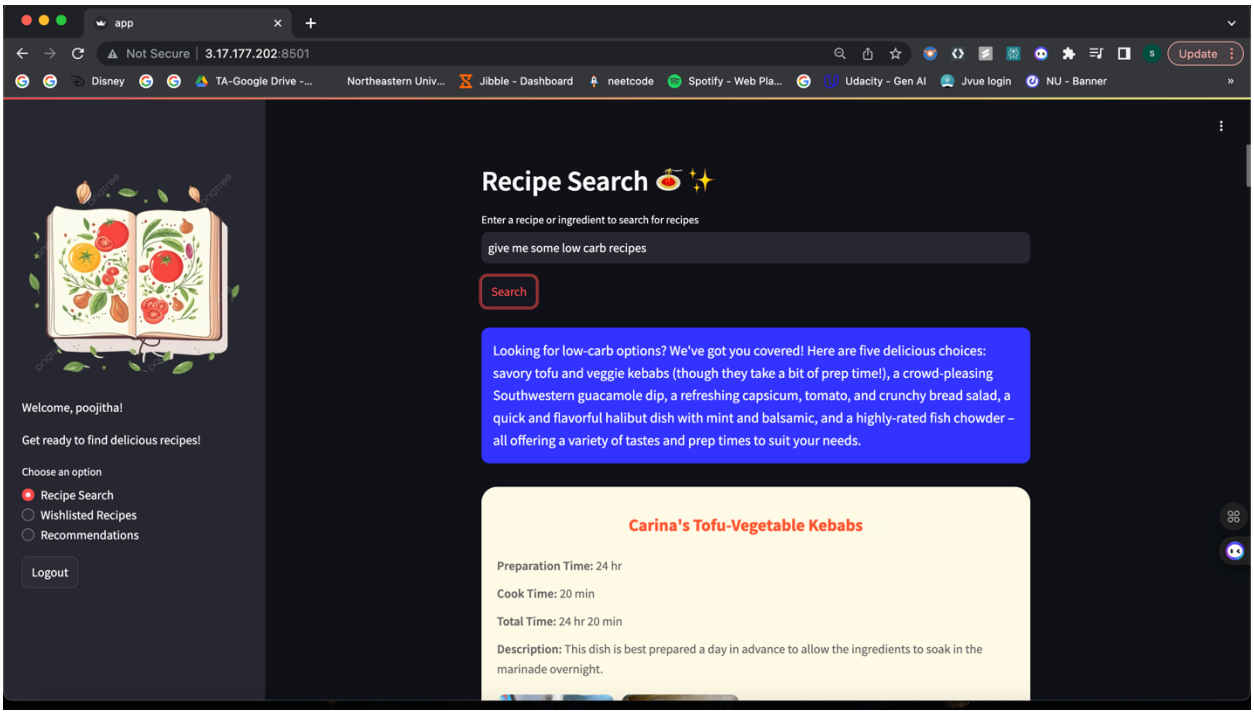
With the vast array of recipes available online, finding a recipe that matches specific preferences or available ingredients can be overwhelming. Existing platforms often lack the personalization and efficiency needed to cater to individual tastes, dietary restrictions, or ingredient availability. Additionally, users face challenges in discovering new cuisines or recipes that align with their unique culinary interests.

The Recipe Curator project addresses these issues by:

- Utilizing NLP and embedding-based vector search to understand and respond to natural language queries effectively.
- Providing tailored recipe recommendations based on user inputs, dietary preferences, and ingredient availability.
- Offering a seamless and intuitive interface for exploring, managing, and saving recipes.

- Leveraging scalable cloud solutions to ensure quick and efficient access to a vast repository of culinary data.

This platform aims to simplify and personalize the process of discovering recipes, transforming it into an engaging and enjoyable experience.



Features of Recipe Curator

1. **Natural Language Query Support**
    - Users can search for recipes using conversational queries like "spicy Indian curries" or "Butter Pecan Cookies."

    - Powered by advanced NLP models from Gemini AI and Hugging Face for precise query understanding.
2. **Personalized Recipe Recommendations**
    - Provides tailored recipe suggestions based on user preferences and query context.
    - Recommendations are enhanced by embeddings and similarity searches using Pinecone.
3. **Wishlist Management**
    - Users can create, update, and manage a wishlist of their favorite recipes.
    - The feature offers a personalized space to save and revisit preferred recipes effortlessly.
4. **Efficient Data Storage and Retrieval**
    - Utilizes **AWS S3** for efficient storage of raw datasets and recipe images.
    - Employs **Snowflake** for structured querying and retrieval of cleaned and preprocessed recipe data.
5. **Advanced Embedding-Based Search**
    - Generates semantic embeddings for user queries and recipes using Hugging Face models.
    - Embedding-based similarity searches in Pinecone ensure accurate and relevant results.
6. **Scalable and Reliable Architecture**
    - Built with scalability in mind, leveraging Docker to containerize the application.
    - Deployed on **Amazon EC2**, enabling high availability and efficient handling of concurrent user requests.
7. **Session Management**
    - Implements robust session management to track user preferences and maintain session states.
    - Enhances user experience by ensuring secure and consistent personalization across sessions using JWT tokens.
8. **End-to-End Dockerization**
    - The entire application, including frontend (Streamlit) and backend (FastAPI), is containerized for streamlined deployments.
    - Dockerized components ensure seamless development, testing, and production setups.

These features collectively make Recipe Curator a powerful, user-friendly, and scalable platform, providing a personalized and intuitive experience for culinary enthusiasts.

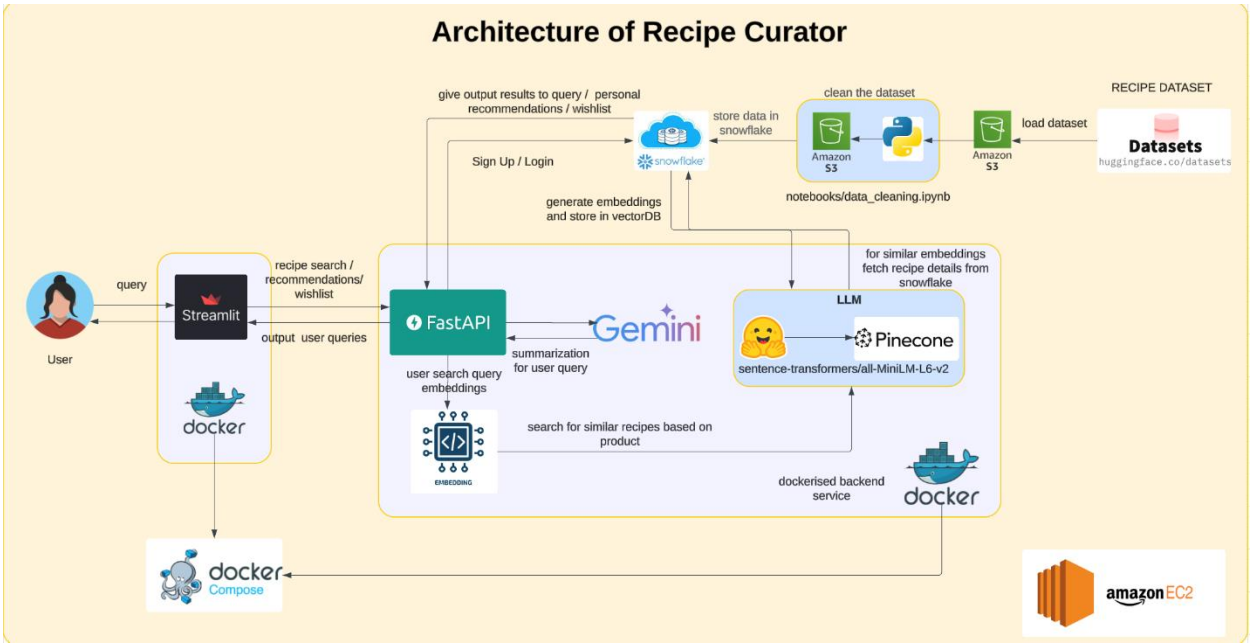
Tools Used

1. **Streamlit**
    - **Purpose:** Frontend development for creating an intuitive and interactive user interface.
    - **Usage:** Displays recipe recommendations, handles user interactions like wishlist management, and provides a seamless browsing experience.
2. **FastAPI**
    - **Purpose:** Backend development for building a high-performance RESTful API.
    - **Usage:** Processes user queries, handles requests and responses, and integrates with various services like Snowflake, Pinecone, and OpenAI.
3. **AWS S3**
    - **Purpose:** Cloud-based object storage.
    - **Usage:** Stores raw datasets, processed recipe images, and other static resources required by the application.
4. **Snowflake**
    - **Purpose:** Cloud-based data warehouse.
    - **Usage:** Stores and retrieves cleaned and preprocessed recipe data, enabling structured querying for recipe recommendations.
5. **Pinecone**
    - **Purpose:** Vector database for similarity search.
    - **Usage:** Stores recipe embeddings and performs embedding-based similarity searches to match user queries with relevant recipes.
6. **OpenAI (GPT Models)**
    - **Purpose:** Natural Language Processing and query understanding.
    - **Usage:** Enhances the ability to process and interpret user queries expressed in natural language.
7. **Hugging Face**
    - **Purpose:** Pre-trained models for embedding generation.
    - **Usage:** Creates semantic embeddings for user queries and recipes, ensuring state-of-the-art NLP capabilities for similarity matching.
8. **Docker**

- **Purpose:** Containerization.
- **Usage:** Ensures seamless deployment and scalability by packaging the application, including its dependencies, into portable containers.

9. **Amazon EC2**
   - **Purpose:** Hosting the application.
   - **Usage:** Deployed on EC2 instances to ensure scalability, reliability, and robust handling of concurrent user traffic.
10. **Gemini**
    - **Purpose:** Query summarization and enhancement.
    - **Usage:** Summarizes complex user queries for better understanding and processing by the system.
11. **Hugging Face Sentence Transformers**
    - **Purpose:** Embedding generation.
    - **Usage:** Generates high-quality embeddings for semantic understanding of recipes and user queries.
12. **Python**
    - **Purpose:** Core programming language.
    - **Usage:** Powers the backend logic, API development, and integration of various machine learning and database components.
13. **Docker Compose**
    - **Purpose:** Multi-container orchestration.
    - **Usage:** Simplifies the management of multiple Docker containers, including frontend, backend, and database components.

Architecture of Recipe Curator:



The **Recipe Curator** follows a structured workflow that seamlessly integrates various tools and services to deliver personalized recipe recommendations. The process begins with the **user interacting through the Streamlit-based frontend**, where they can submit queries, manage wishlists, and explore recommendations. Once a user submits a query (e.g., "spicy Indian curry"), it is passed to the **FastAPI backend**, which serves as the core service for handling requests and orchestrating interactions with other components.

The **query is processed by FastAPI**, where embeddings are generated using Hugging Face's **sentence-transformers/all-MiniLM-L6-v2** model. These embeddings enable the system to understand the semantic intent of the user query. The embeddings are then sent to **Pinecone**, a vector database, where a similarity search is performed to find recipes that closely match the user's query based on the embeddings. Pinecone efficiently retrieves matching recipes' IDs.

Once the relevant recipe IDs are identified, the system queries **Snowflake**, a structured data warehouse, to fetch detailed recipe information, such as ingredients, steps, and nutritional facts. If the query is verbose or complex, it is first summarized by **Gemini AI**, which simplifies the input for better processing by the backend.

The application also incorporates a **data cleaning pipeline** where raw recipe datasets are stored in **AWS S3** and preprocessed using Python scripts in a Jupyter Notebook. The cleaned data is then ingested into Snowflake, ensuring that the database contains only high-quality, structured data. The entire application stack, including the frontend, backend, and data processing components, is **containerized using Docker** and orchestrated with Docker Compose for consistent and scalable deployment on **Amazon EC2**.

Finally, the results are displayed on the **Streamlit frontend**, providing the user with a list of personalized recipe recommendations. To enhance user experience, **session management** is implemented using JSON Web Tokens (JWT), allowing users to maintain their preferences, wishlists, and search history across sessions. This streamlined workflow ensures that Recipe Curator delivers an intuitive, efficient, and engaging experience for its users.

Dataset Details for Recipe Curator

Source: Hugging Face Dataset - [AkashPS11/recipes_data_food.com]

This dataset is a comprehensive collection of recipes sourced from **Food.com**, making it highly suitable for building an intelligent recipe recommendation platform. It contains detailed information for each recipe, allowing advanced processing and personalized recommendations. Below are the key attributes and their significance:

Key Features of the Dataset:

1. **RecipeId**: A unique identifier for each recipe, essential for indexing and retrieval.
2. **Name**: The title of the recipe, providing a clear description for users.
3. **AuthorId & AuthorName**: Identifies the creator of the recipe, which helps track and credit content creators.
4. **CookTime, PrepTime, and TotalTime**: Detailed time specifications for cooking, preparing, and completing the recipe, enabling users to filter recipes based on available time.
5. **DatePublished**: The date the recipe was published, which can be used to find trendy or seasonal recipes.
6. **Description**: A summary or introduction to the recipe, enhancing user engagement.
7. **Images**: Links to high-quality images associated with the recipe, providing a visual appeal.
8. **RecipeCategory**: Categories such as "Chicken Breast" or "Frozen Desserts," enabling easy filtering.
9. **Keywords**: Tags like "Indian," "Stove Top," or "Healthy," which help with semantic search and relevance.
10. **RecipeIngredientQuantities & Parts**: Detailed quantities and parts of ingredients, crucial for accurate recommendations and user convenience.
11. **AggregatedRating & ReviewCount**: User-generated ratings and reviews, ensuring users get insights about recipe popularity and quality.
12. **Nutrition Information**: Detailed breakdown of calories, fat, cholesterol, sodium, carbohydrates, and proteins, helping health-conscious users make informed decisions.
13. **RecipeInstructions**: Step-by-step instructions for preparing the recipe, ensuring a seamless cooking experience.

How to Run the Application?

1.Clone the Repository:

```
git clone https://github.com/sripoojitha-mandali/Recipe_Curator_GenAI.git
cd Recipe_Curator_GenAI
```

2. Create a .env file in the project directory with the following variables:

```
SNOWFLAKE_USER=xxxx
SNOWFLAKE_PASSWORD=xxxx
SNOWFLAKE_ACCOUNT=xxxx
SNOWFLAKE_WAREHOUSE=xxxx
SNOWFLAKE_SCHEMA=xxxx
SNOWFLAKE_ROLE=xxxx
SNOWFLAKE_DATABASE=xxxx
AWS_ACCESS_KEY_ID=xxxx
AWS_SECRET_ACCESS_KEY=xxxx
pinecone_key=xxxx
gemini_key=xxxx
huggingface_key=xxxx
JWT_SECRET=xxxx
```

3. Build the Containers
Run the following command to build the Docker containers:

```
docker-compose build
```

4. Run the Application
Start the application by running:

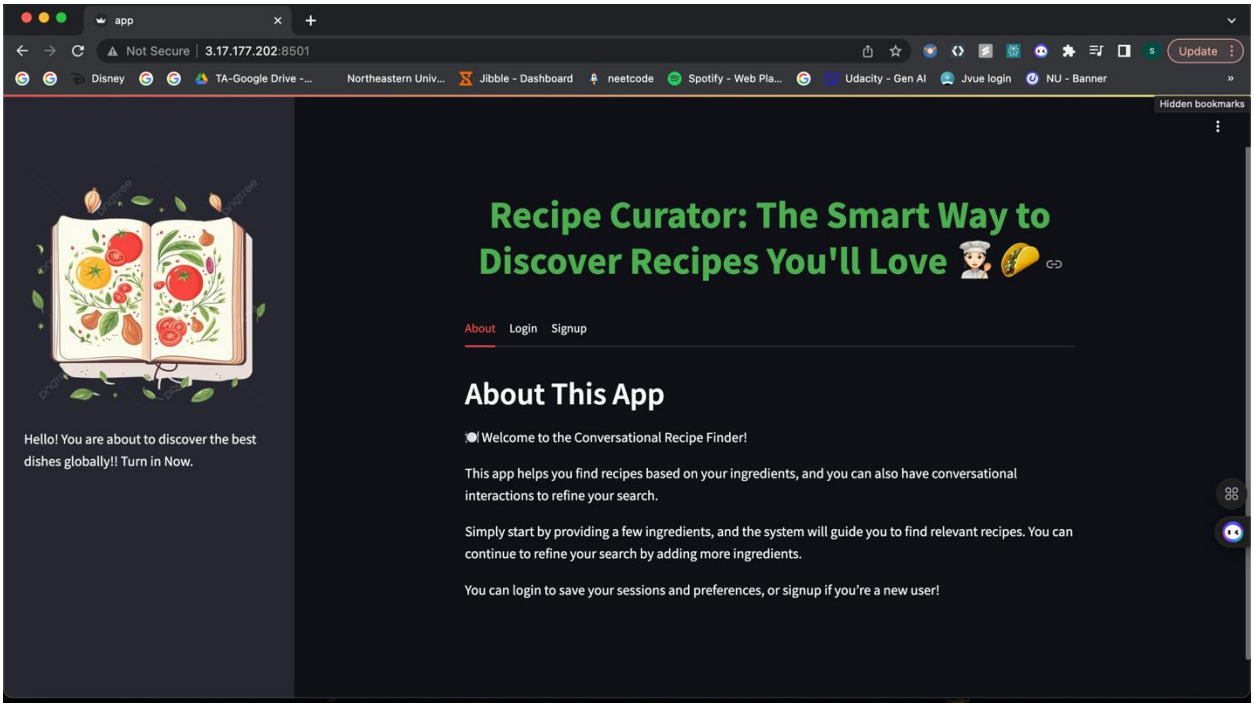
```
docker-compose up
```

Now the application is up and running. Navigate to the following link to check the application in your web browser:

http://0.0.0.0:8501

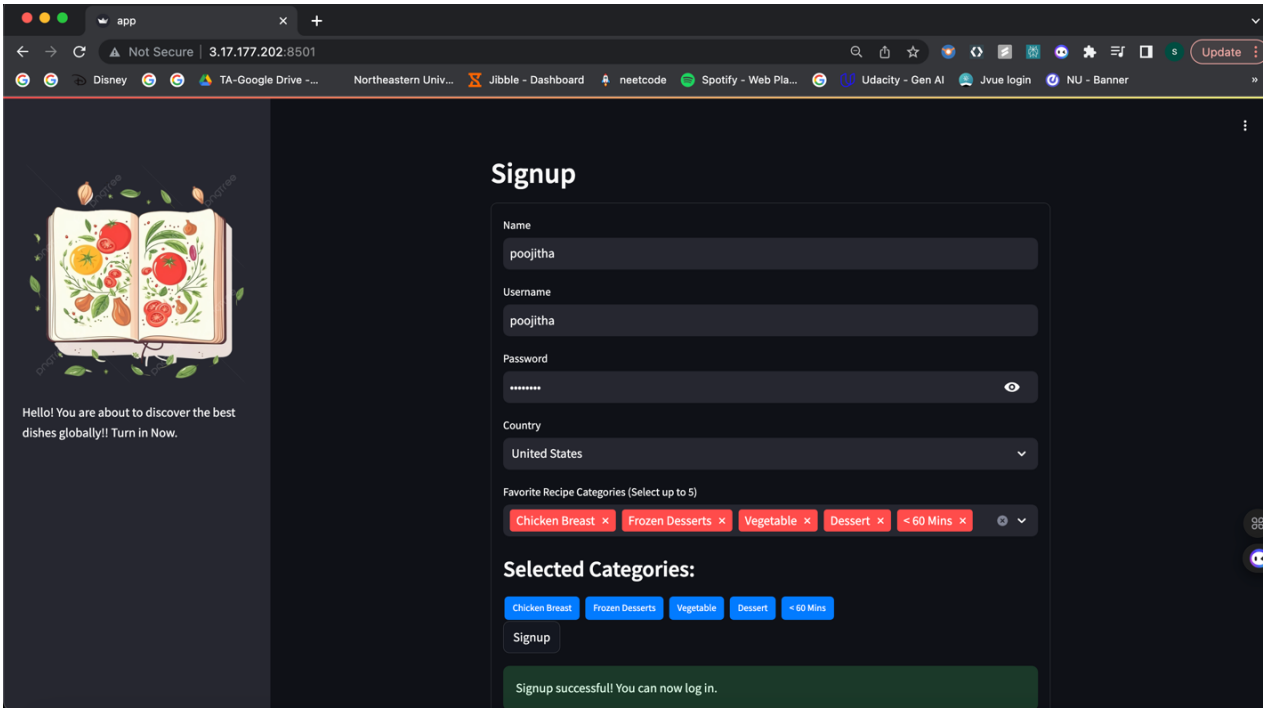
User Manual

1. This is the homepage of the recipe curator application.

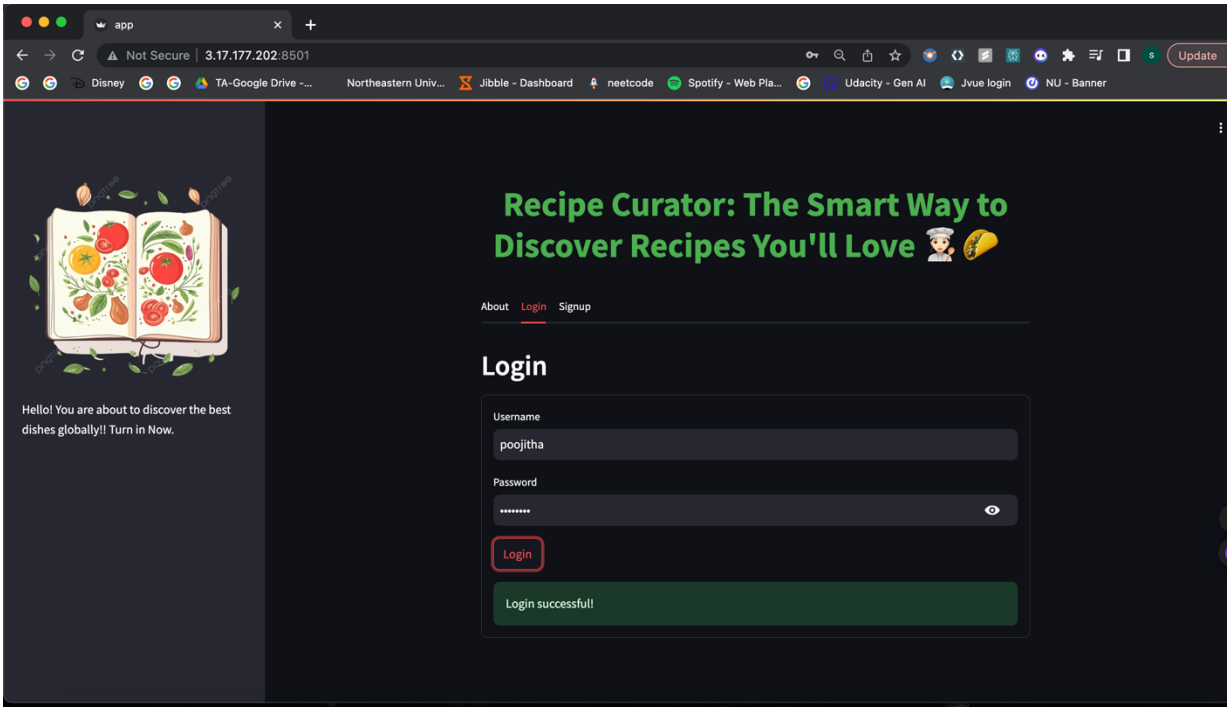


2. Log In or Sign Up: Create an account to unlock a wide range of resources in our application.

Sign-up Page: In the signup page all the fields are mandatory and the fields Favourite Category is vital for a better user experience and personalized recommendations.

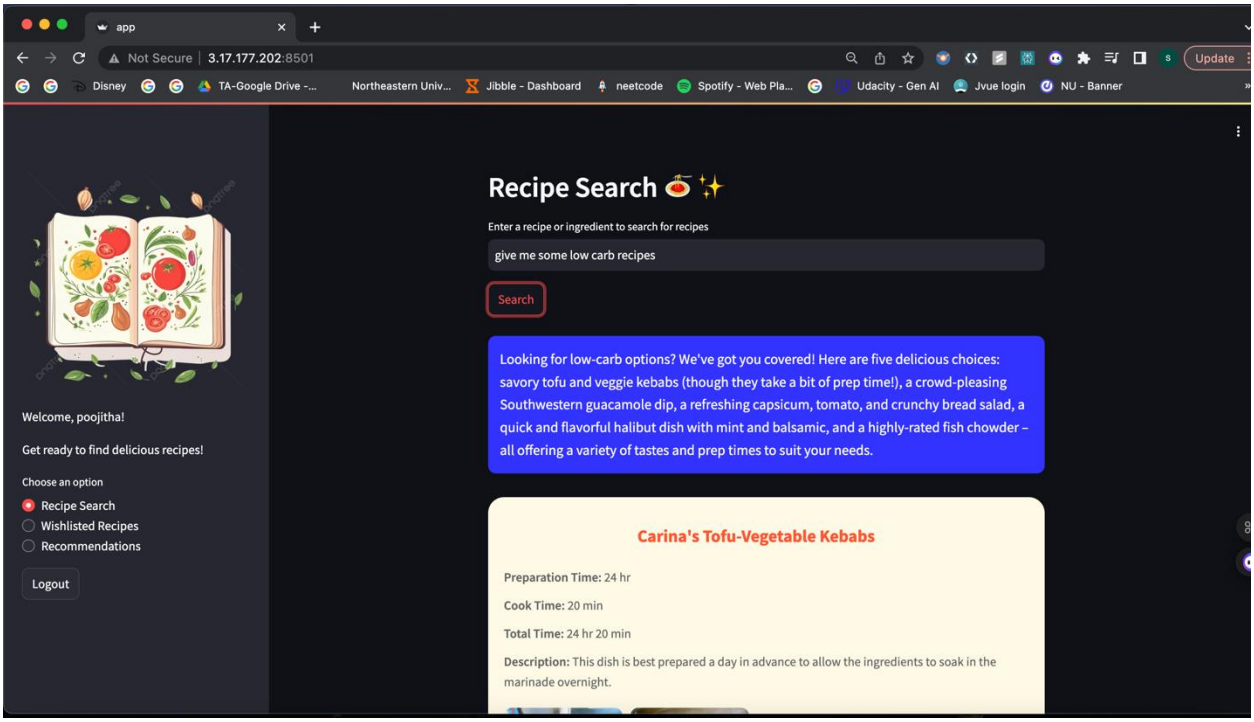


Login Page:

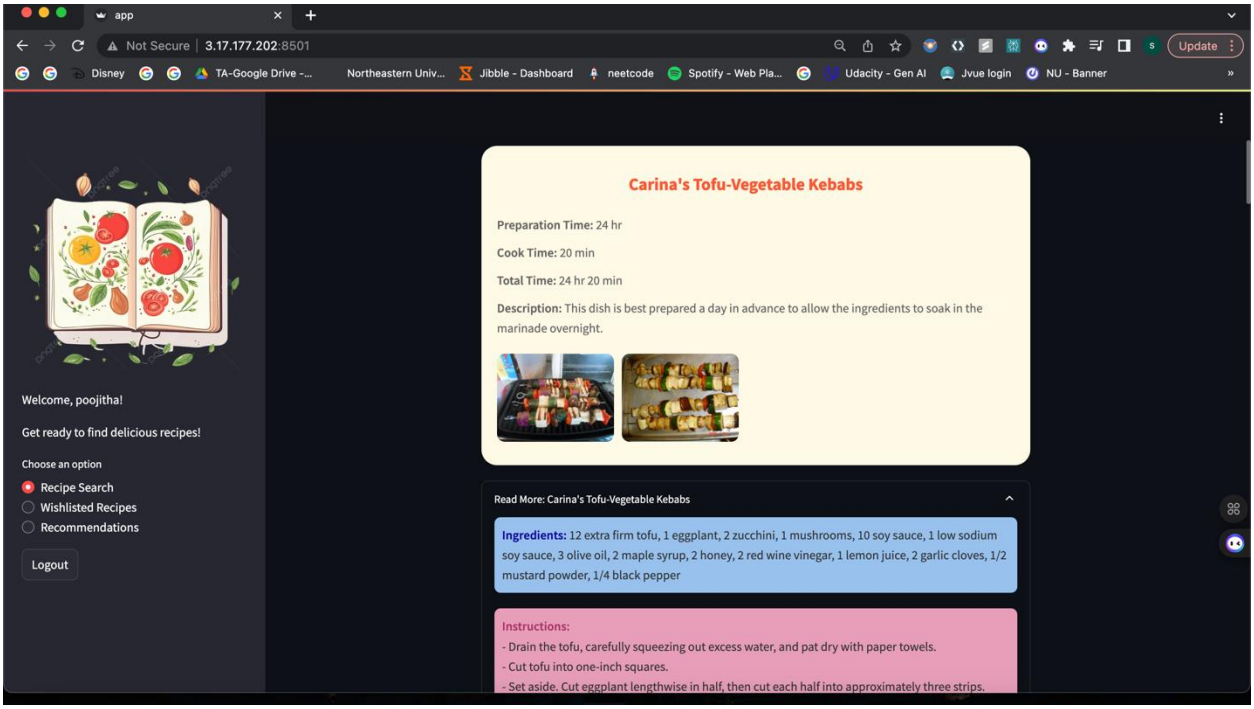


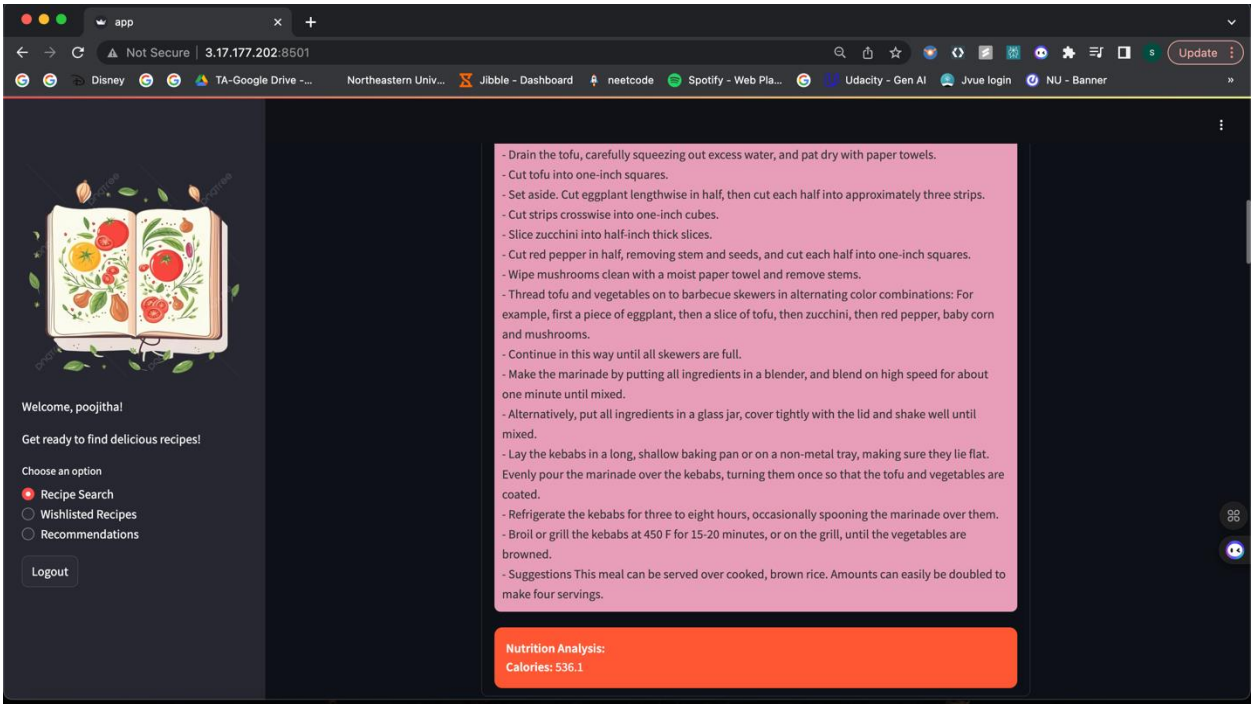
After successful login, the user will be navigated to the main page of the application.

3. Here in recipe search page, you can enter any input queries based on cuisine type , ingredients or search recipes and it will output top 10 recipes for your query and also it will give the ingredients , recipe procedure , time it will take and also nutritional analysis.

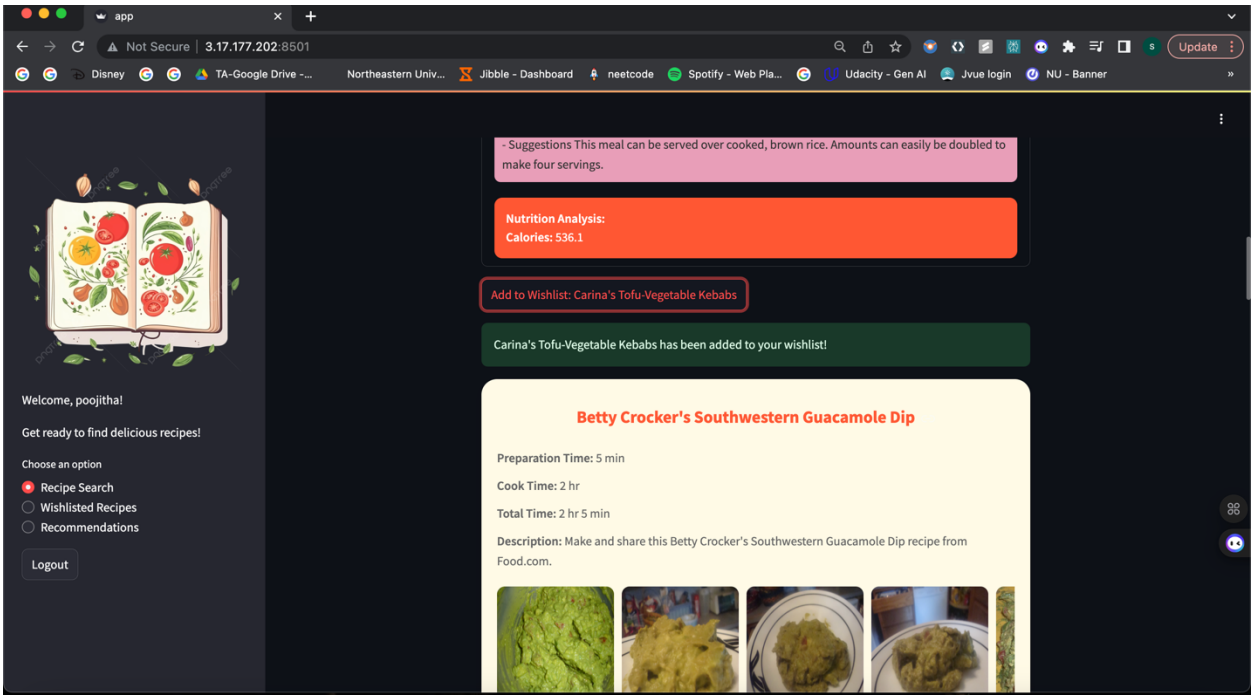


Once you click on read more you can see ingredients, instructions and nutritional analysis.

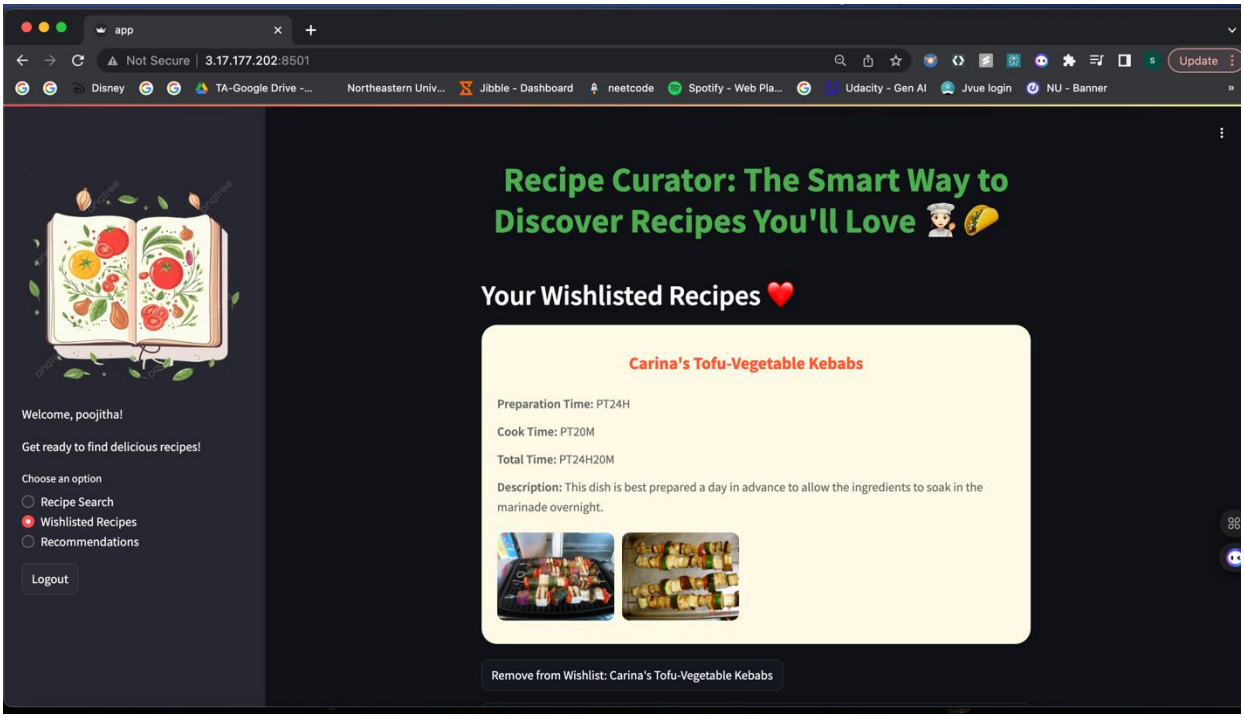




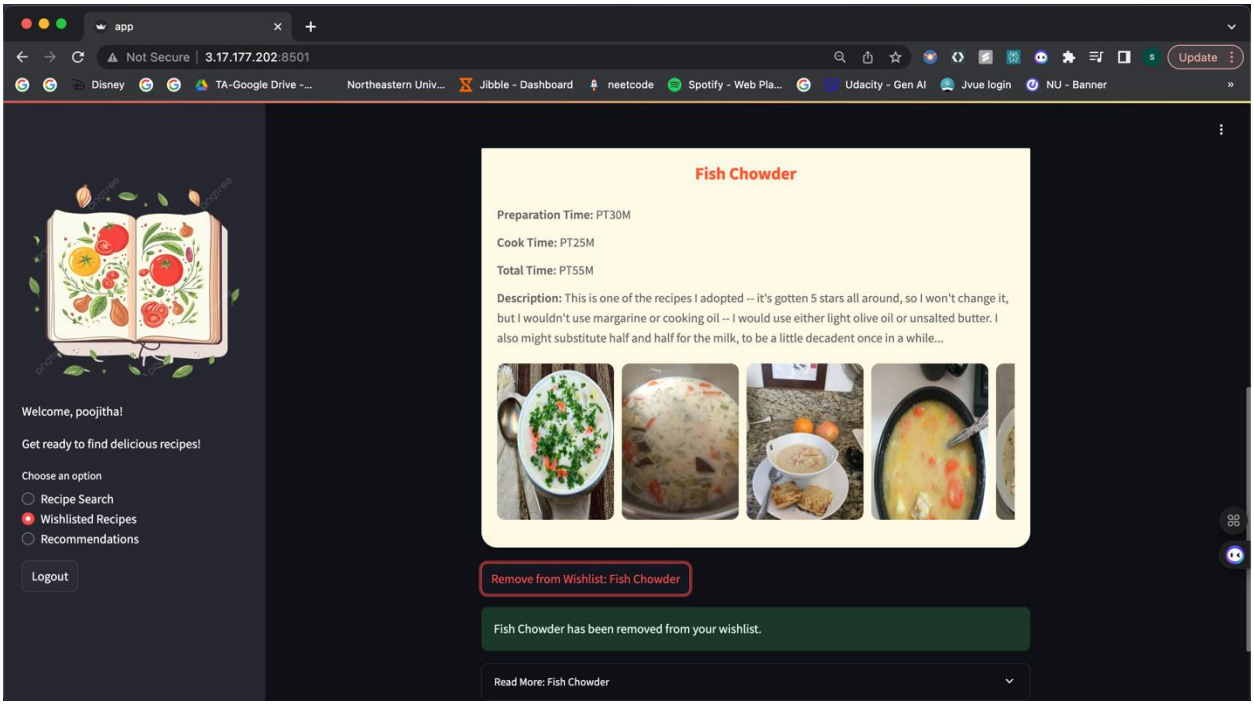
4. You can Wishlist interested recipes to your Wishlist cart.



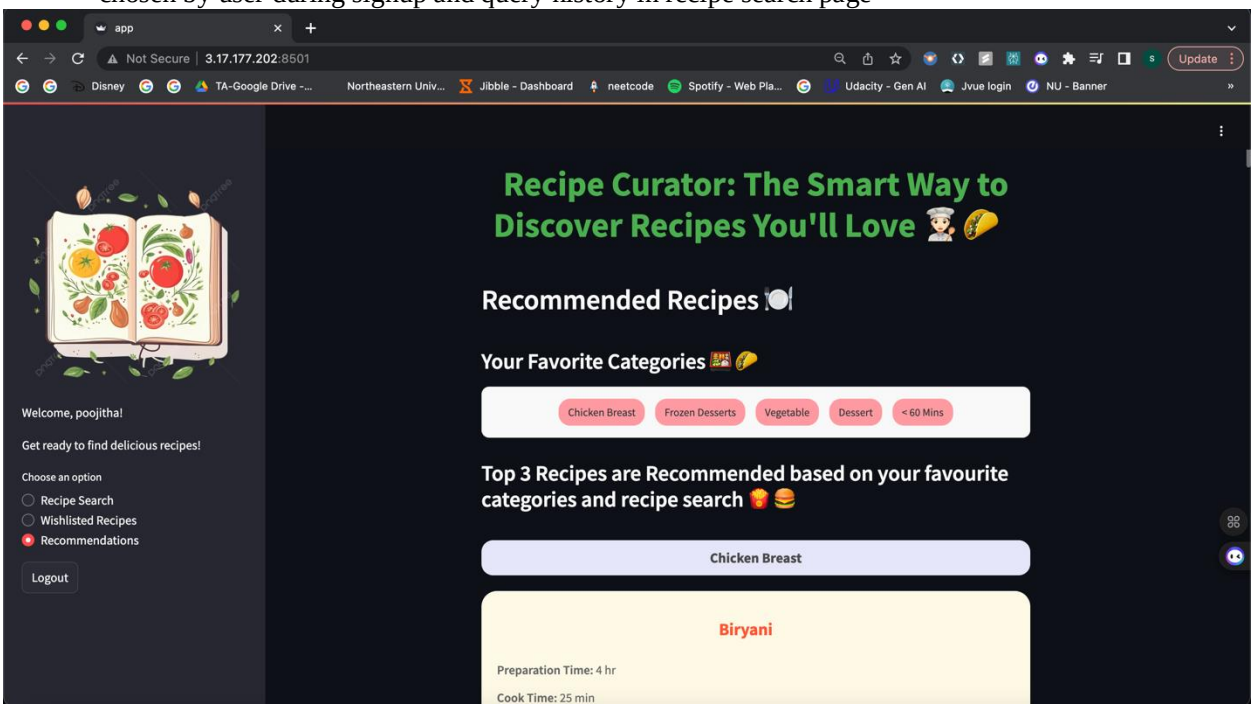
5. You can see your wishlisted recipes in wishlist cart

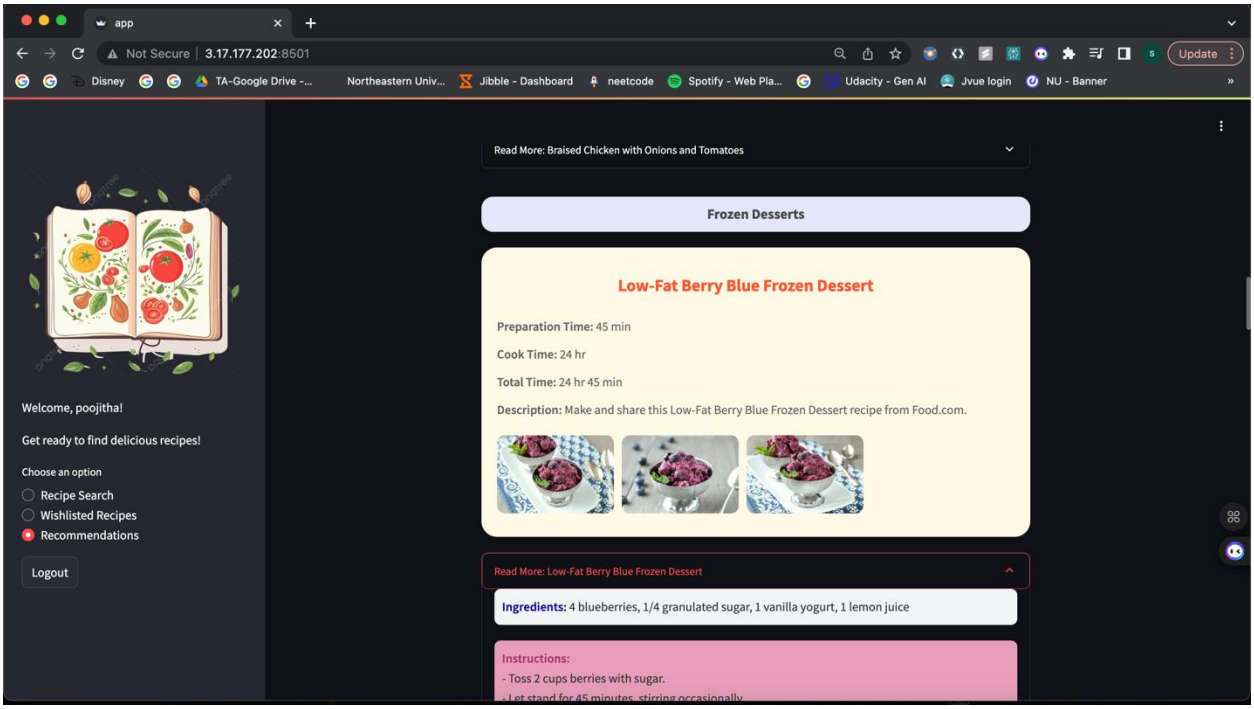


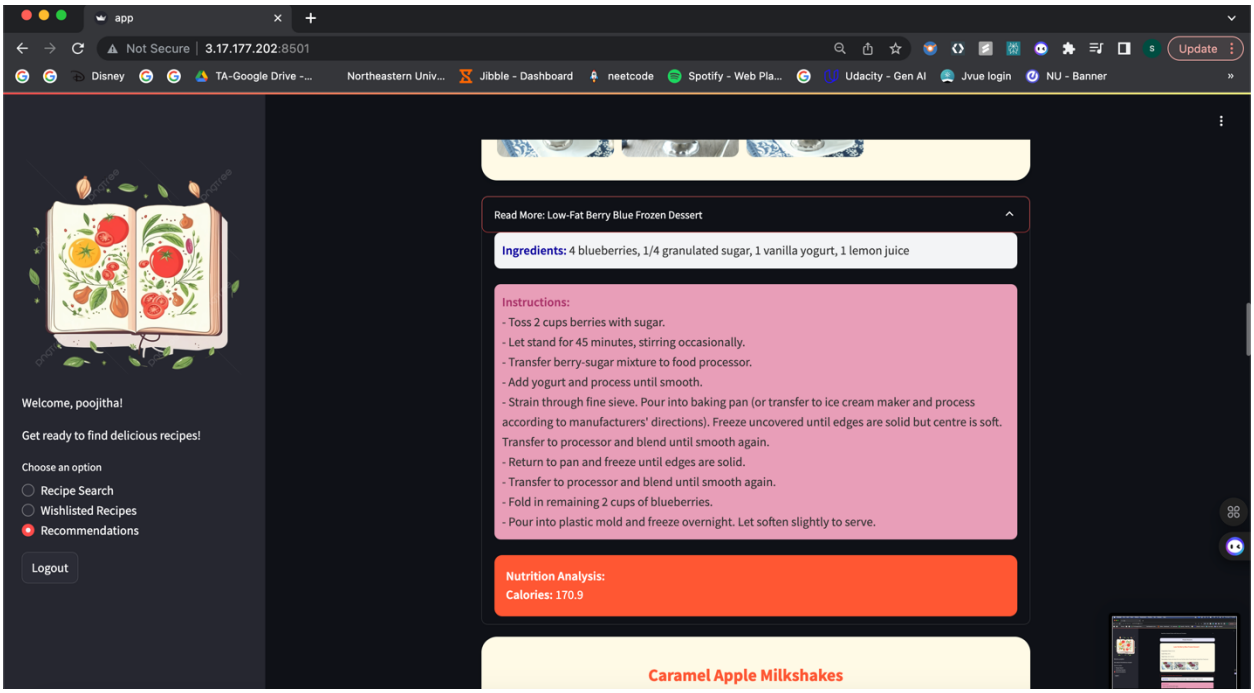
Also, you can remove recipes from your Wishlist cart:



6. The application provides recommendations of recipes to users based over the 5 different categories chosen by user during signup and query history in recipe search page







Challenges Faced in the Recipe Curator Project

1. **Data Cleaning and Preprocessing**
    - The dataset from **Hugging Face** contained inconsistencies such as missing values, unstructured ingredient lists, and inconsistent formats for fields like

`RecipeIngredientQuantities`. Ensuring the data was clean and usable required significant effort and iterative refinement.

2. **Efficient Embedding Generation**
    - Generating high-quality embeddings for thousands of recipes using Hugging Face models like `sentence-transformers/all-MiniLM-L6-v2` required optimization to balance computational cost and processing time while ensuring relevance in similarity searches.
3. **Scalable Deployment on Amazon EC2**
    - Dockerizing and deploying all components, including the **Streamlit frontend**, **FastAPI backend**, and **Pinecone vector search**, on **Amazon EC2** was a complex process. Ensuring scalability to handle growing user traffic and data volume was a key focus area.
4. **Natural Language Understanding**
    - Implementing natural language query understanding, especially for ambiguous queries like "spicy Indian curry," required leveraging OpenAI's LLMs and Gemini for summarization. Tuning these models to provide accurate and context-aware recommendations was a significant challenge.
5. **Frontend-Backend Communication and User Sessions**
    - Managing secure and efficient communication between the **Streamlit frontend** and the **FastAPI backend** posed integration challenges. Additionally, implementing robust session management for personalized recommendations and wishlists required careful handling of user data and authentication mechanisms.

These challenges were addressed through methodical problem-solving, iterative testing, and the integration of state-of-the-art tools and technologies.

Conclusion

The **Recipe Curator** project aims to transform the way users interact with recipe recommendations by leveraging cutting-edge AI technologies such as LLMs and semantic vector databases. The integration of tools like **Streamlit**, **FastAPI**, **Pinecone**, and **Snowflake** has enabled the development of a personalized, interactive, and scalable solution. The project's unique combination of conversational AI and advanced search capabilities enhances the user experience, offering an intuitive way to discover, manage, and plan recipes. Future improvements like multi-language support, expanded datasets, and complex meal planning features will continue to enrich the system, making it more accessible and versatile for a broader audience.

References

1. **Hugging Face Dataset**: AkashPS11/recipes_data_food.com, Hugging Face.
2. **Streamlit Documentation**: Streamlit Official Documentation, streamlit.io
3. **FastAPI Documentation**: FastAPI Official Documentation, fastapi.tiangolo.com.
4. **Pinecone Vector Database**: Pinecone Official Documentation, pinecone.io.
5. **Snowflake Documentation**: Snowflake Official Documentation, snowflake.com.
6. **Hugging Face Sentence Transformers**: all-MiniLM-L6-v2, huggingface.co.
7. **Docker and Docker Compose**: Docker Official Documentation, docker.com.

8. **Amazon EC2**: Amazon EC2 Overview, aws.amazon.com.

License

MIT License

Copyright (c) 2024 Sri Poojitha Mandali, Vasudha Ambre, Rasika Kole

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Team Members

Sri Poojitha Mandali	mandali.s@northeastern.edu	001560891
Vasudha Ambre	ambre.v@northeastern.edu	002676424
Rasika Kole	kole.r@northeastern.edu	002620332