Project Documentation

Diet Recommendations & Health assistant

Submitted to:

DEPI

Submitted by:

Name: Ahmed Hany

Name: Ahmed Farouk

Name: Yousef Badr

Name: Mahmoud Hisham

Digital Egypt Pioneers Initiative

Contents

Chapter No 1: INTRODUCTION

1.1 Introduction:

This project focuses on building a personalized diet and recipe recommendation system to help users make better dietary decisions aligned with their medical conditions and preferences. Using Collaborative Filtering models, we recommend personalized diet plans and recipes. A user-friendly interface, built with Streamlit, allows users to input their medical history and dietary preferences to receive dynamic recommendations.

1.2 Problem Statement:

Managing personalized diets can be challenging, especially for individuals with specific medical conditions (e.g., diabetes, hypertension). Healthcare professionals also need tools to provide diet recommendations, but current solutions lack personalization and adaptability. There's a need for a system that tailors diet plans and recipes to individual needs based on both medical history and personal tastes.

1.3 Proposed Solution:

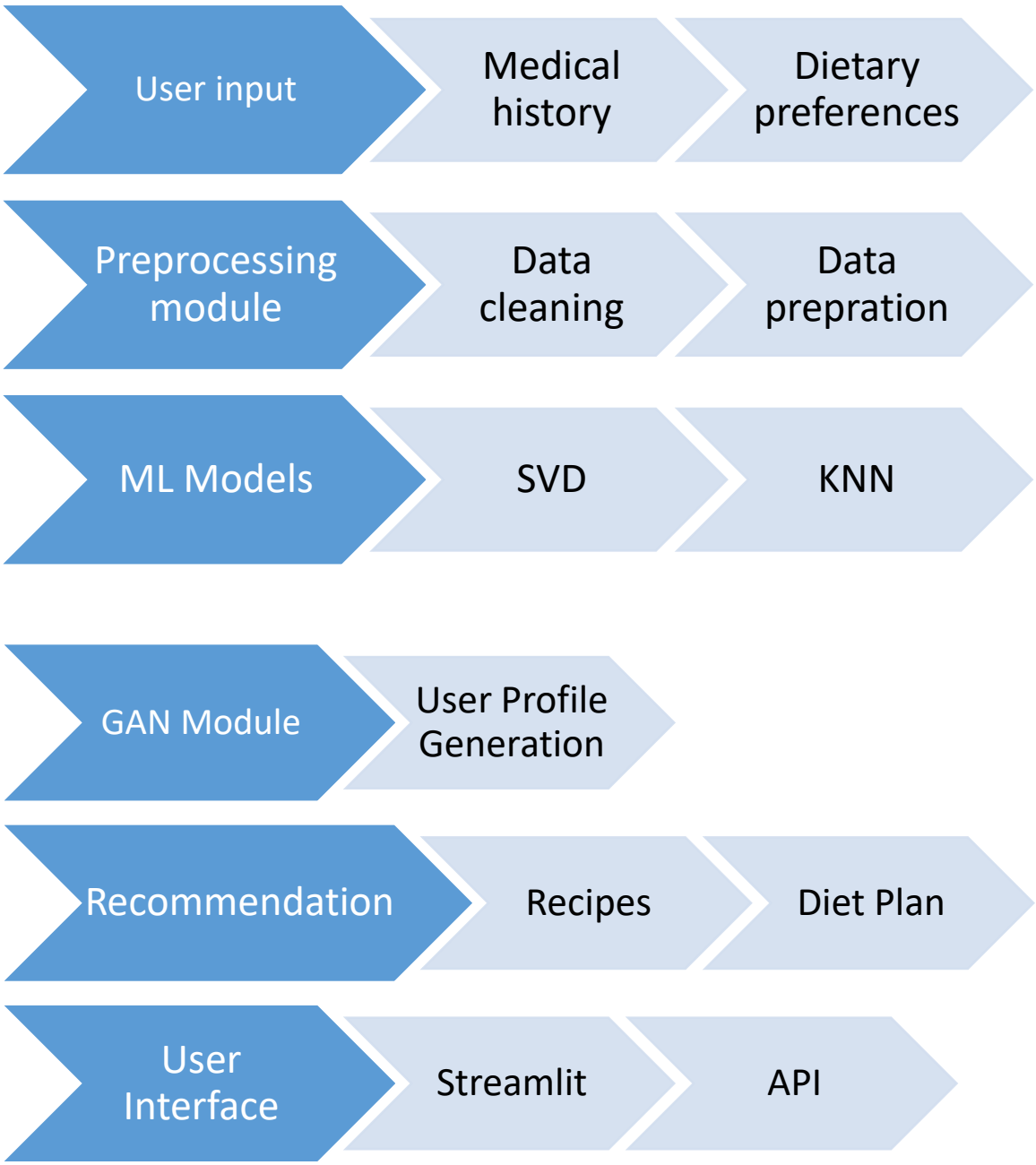
Our system provides an easy way for users to enter personal data (e.g., health conditions, preferences) and receive a customized diet plan with recipe suggestions. The recommendation engine uses advanced Collaborative Filtering models, such as SVD and KNN-based filtering, to predict and recommend recipes. Additionally, GANs help create dynamic user profiles, refining the recommendations as the user interacts with the platform.

Unique Features:

- *GAN-based User Profiles for improved personalization*
- *SVD and KNN Filtering for accurate recipe recommendations*
- *Streamlit Interface*

CHAPTER 2: System Design

2.1 Architecture Diagram



Recommendation Pipeline

1.Data Input: User provides health conditions, preferences, and feedback.

2.SVD Model: Predicts the rating for recipes based on matrix factorization.

3.KNNBasic Filtering: Recommends recipes by comparing similar users.

4.GANs: Refine the user profile over time for better future recommendations.

5.Streamlit UI: Displays the results with easy navigation.

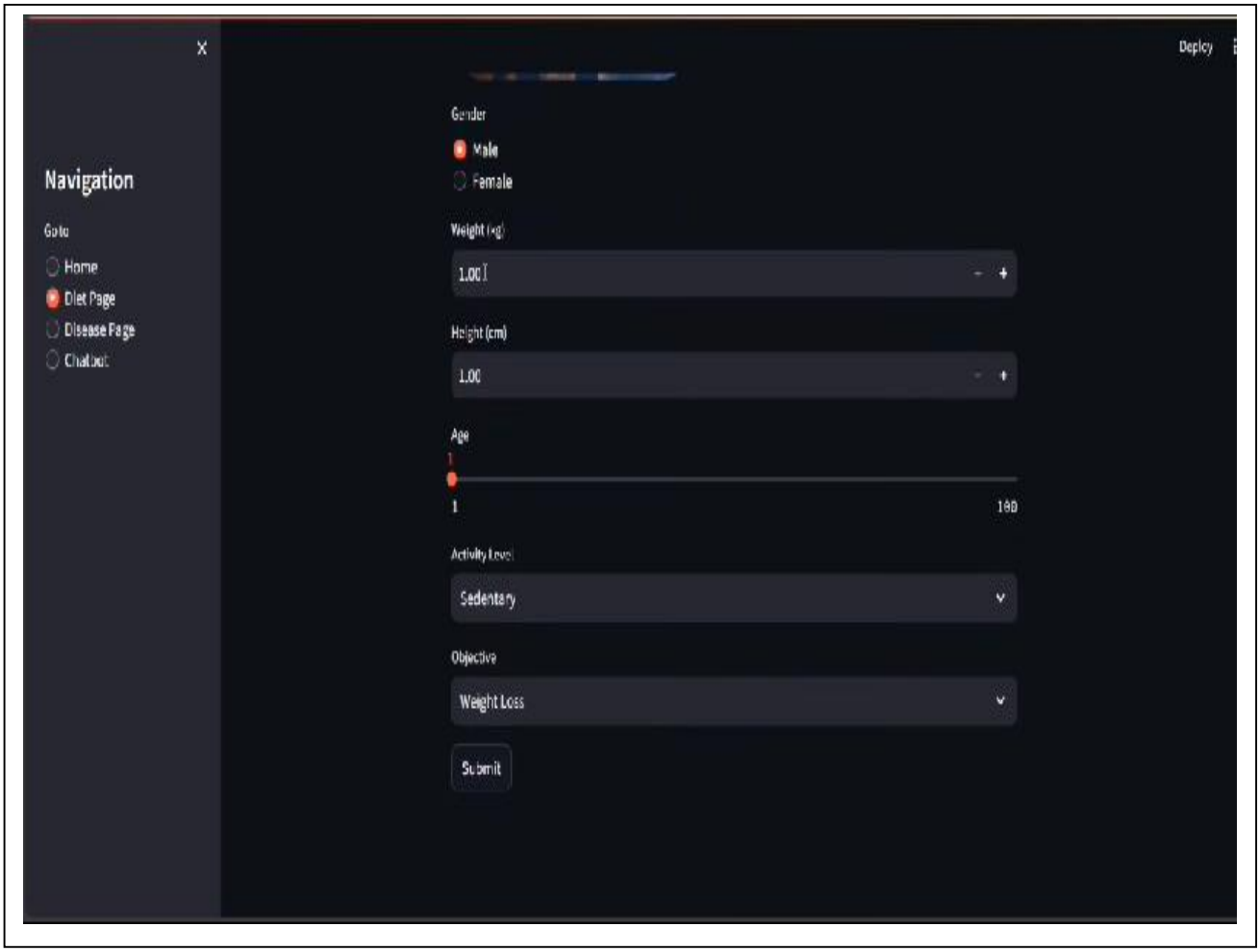
Chapter 3: Wireframe and Interface Design

3.1 Home Page:

Navigation contains 3 more pages you can navigate:

1. **Diet Page**
2. **Disease Page**
3. **Chatbot Page**

3.2 Diet Page:



***Diet Page**:*

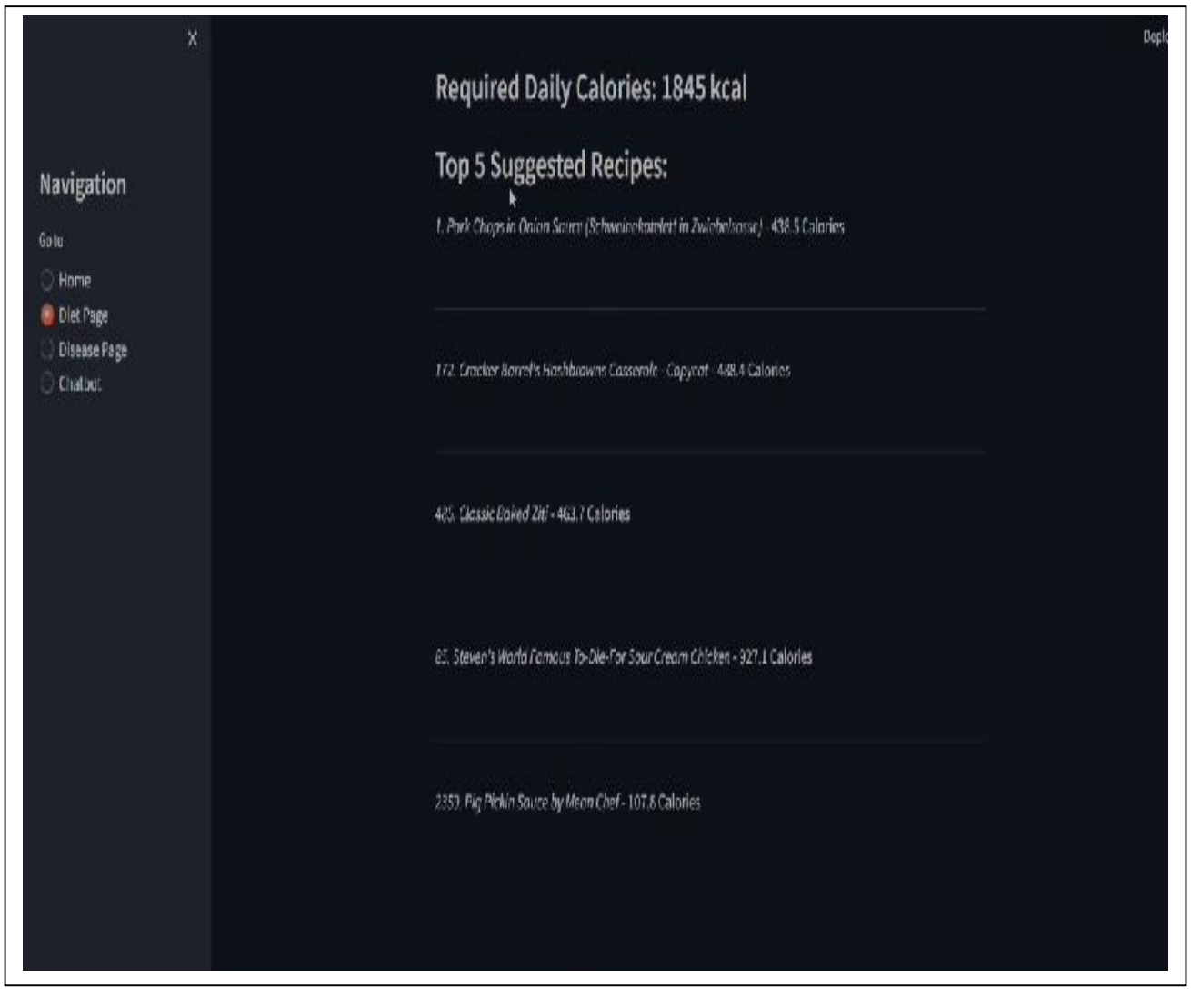
Here the user can enter some personal info for our model

Such:

1. *Gender: (male, female)*
2. *Weight (num)*
3. *Height (num)*
4. *Age (num[0,100])*

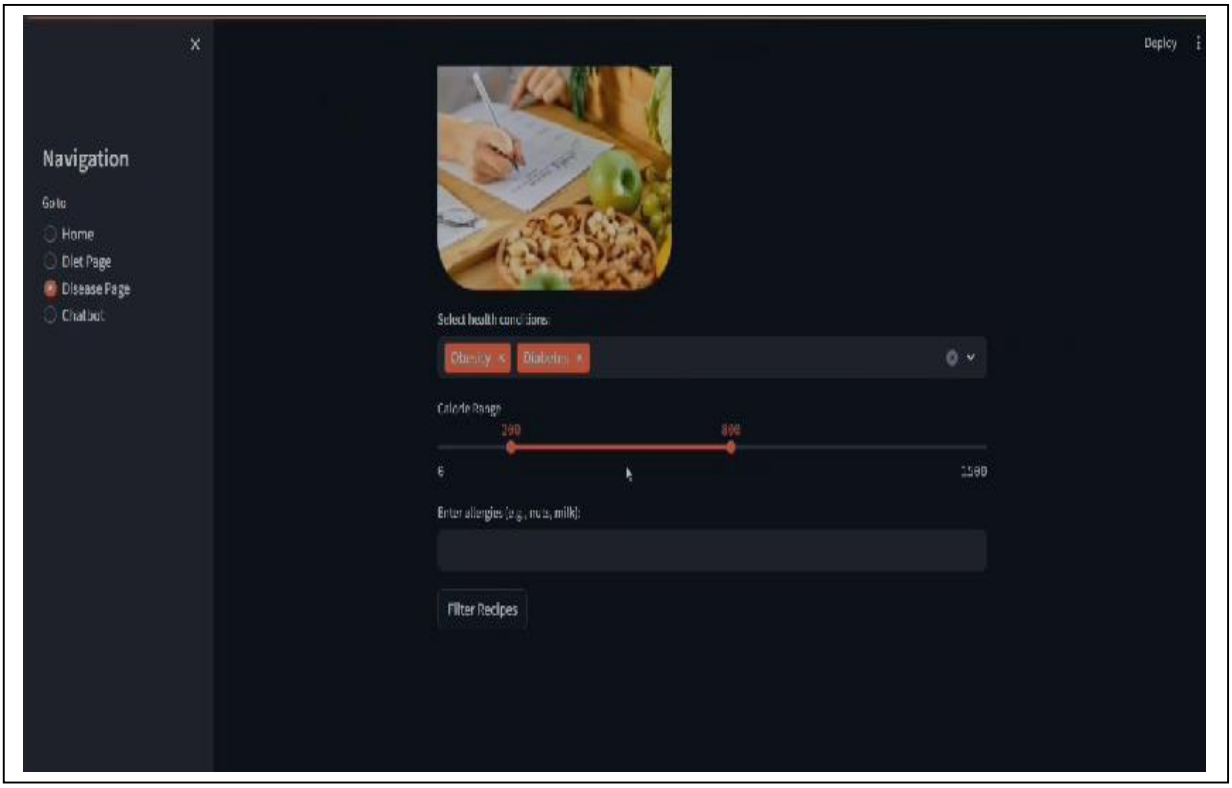
5. *Activity Level (Lightly active, moderately active, Extra active, Very active, Sedentary)*
6. *Objective Adjustment (Weight loss, Muscle gain, Health maintenance)*

Our model Takes these inputs from the user and recommends 5 best recipes and the suitable calories consumption for the user case based on his physical info.



Output

3.4 Disease Page:



The

Disease page:

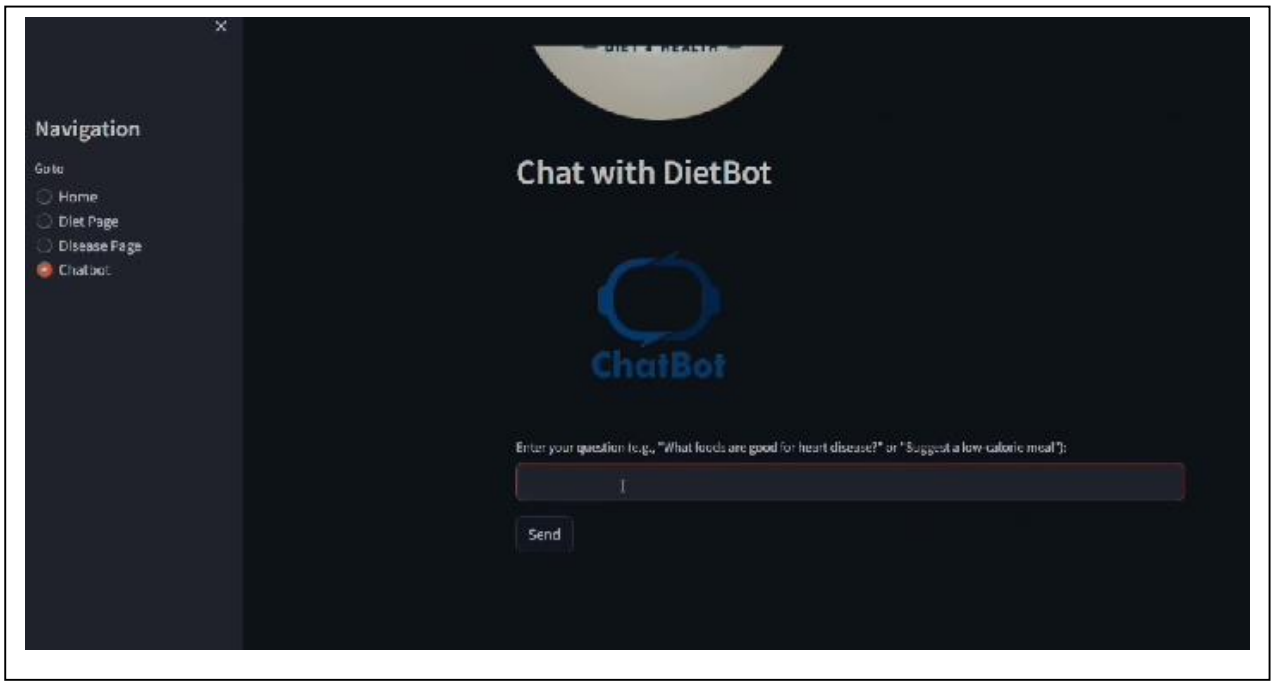
Contains inputs such:

1. *Health Condition (if the user had any diseases (Obesity, Diabestes, ets…))*
2. *Calories rang to make it more optimized*
3. *Allergies (soy, milk, nuts, diary, eggs, shelfish)*

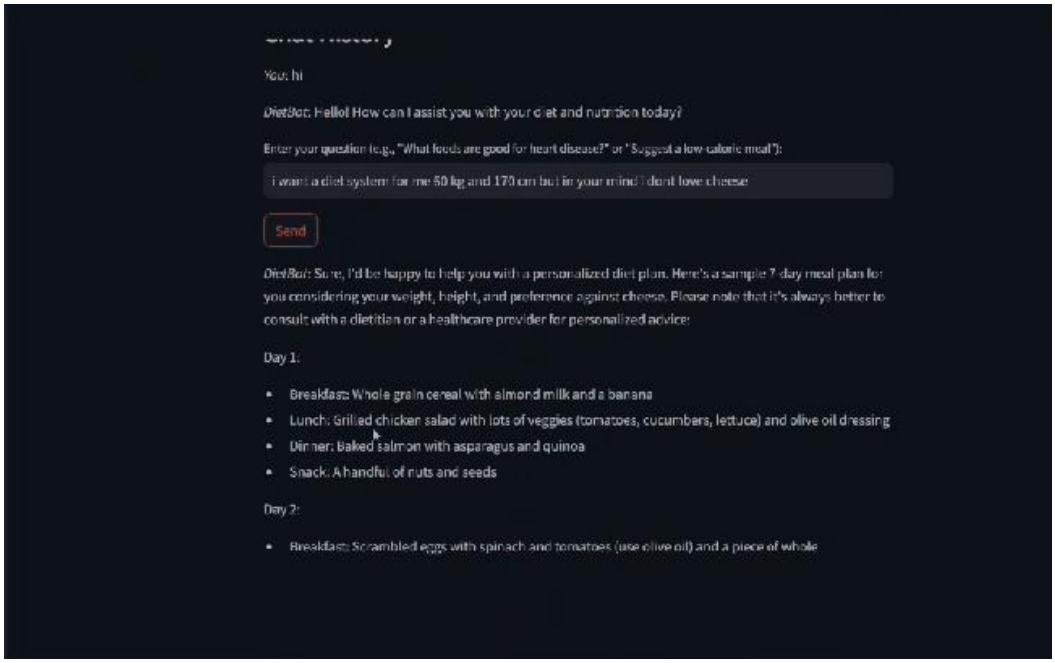
In this page the model recommends the best 5 recipes based on his medical codition.

Output

3.3 Chatbot:



A chat bot to help the user with any question related to the diet or the recipes and help him to understand what type of diet or recipes suits his condition.



Output

CHAPTER 4: Models and Recommendation System Overview

4.1 Introduction to Recommendation Systems

Recommendation systems are algorithms designed to suggest relevant items to users based on their preferences or behavior. They are commonly used in platforms like Netflix, Amazon, or Spotify to recommend movies, products, or songs. There are three major types of recommendation systems:

1. *Content-Based Filtering*

- *Uses information about the items (e.g., recipe ingredients or nutritional values) to recommend similar content.*
- *If a user likes "Vegetable Stir Fry," the system may suggest other healthy vegetable-based recipes.*

2. *Collaborative Filtering*

- *Focuses on the relationship between users and items.*
- *It identifies patterns in user interactions (such as ratings) to recommend items based on the behavior of similar users.*

3. *Hybrid Systems*

- *Combines content-based and collaborative approaches to improve recommendation accuracy.*

4.2 Models We Used

In our project, we implemented a hybrid recommendation system by integrating multiple models:

1. Collaborative Filtering

• Matrix Factorization using SVD (Singular Value Decomposition):

• This algorithm predicts a user's rating for a recipe by decomposing the user-recipe interaction matrix.

• Mathematical Intuition:

*R = U * Sigma * V^T*

- *R: User-Recipe matrix (ratings)*
- *U and V: Matrices representing latent factors for users and items.*
- *Sigma: Diagonal matrix of singular values.*

• This helps predict missing values by identifying hidden patterns between users and recipes.

• User-Based Collaborative Filtering with KNNBasic

The algorithm finds users with similar rating patterns using K-Nearest Neighbors (KNN) and recommends recipes they liked.

- *It compares users by calculating similarity metrics like cosine similarity or Pearson correlation.*

- *Item-Based Collaborative Filtering*

- *Recommends recipes similar to the ones a user has highly rated, based on shared ratings with other users.*

2. Content-Based Filtering

- *Uses metadata from recipes (e.g., ingredients, calories) to recommend similar dishes.*

- *If a user selects a high-protein meal, the system suggests other protein-rich recipes.*

3. Generative Adversarial Network (GAN) for User Profiles

- *To improve personalization, we employed GANs to generate enriched user profiles.*

- *GANs simulate potential user behaviors by generating synthetic data, helping the system better understand individual preferences even with limited real-world data.*

4.3 Why this approach?

• Collaborative Filtering captures relationships between users and their past interactions, enabling personalized recommendations.

• Content-Based Filtering ensures recommendations remain relevant to the user's preferences based on the recipe's nutritional content.

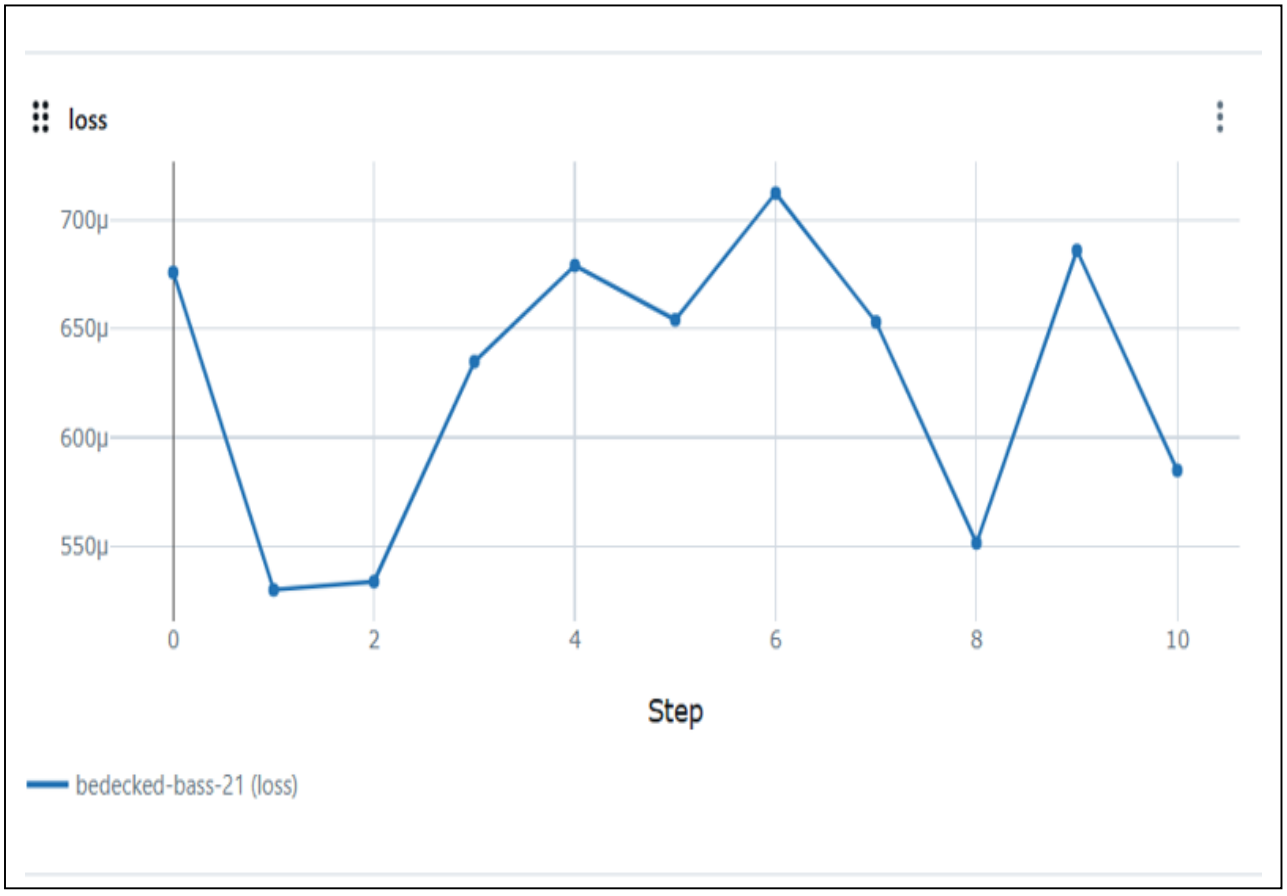
• GANs allow us to model nuanced preferences, even with sparse data, enhancing recommendation quality.

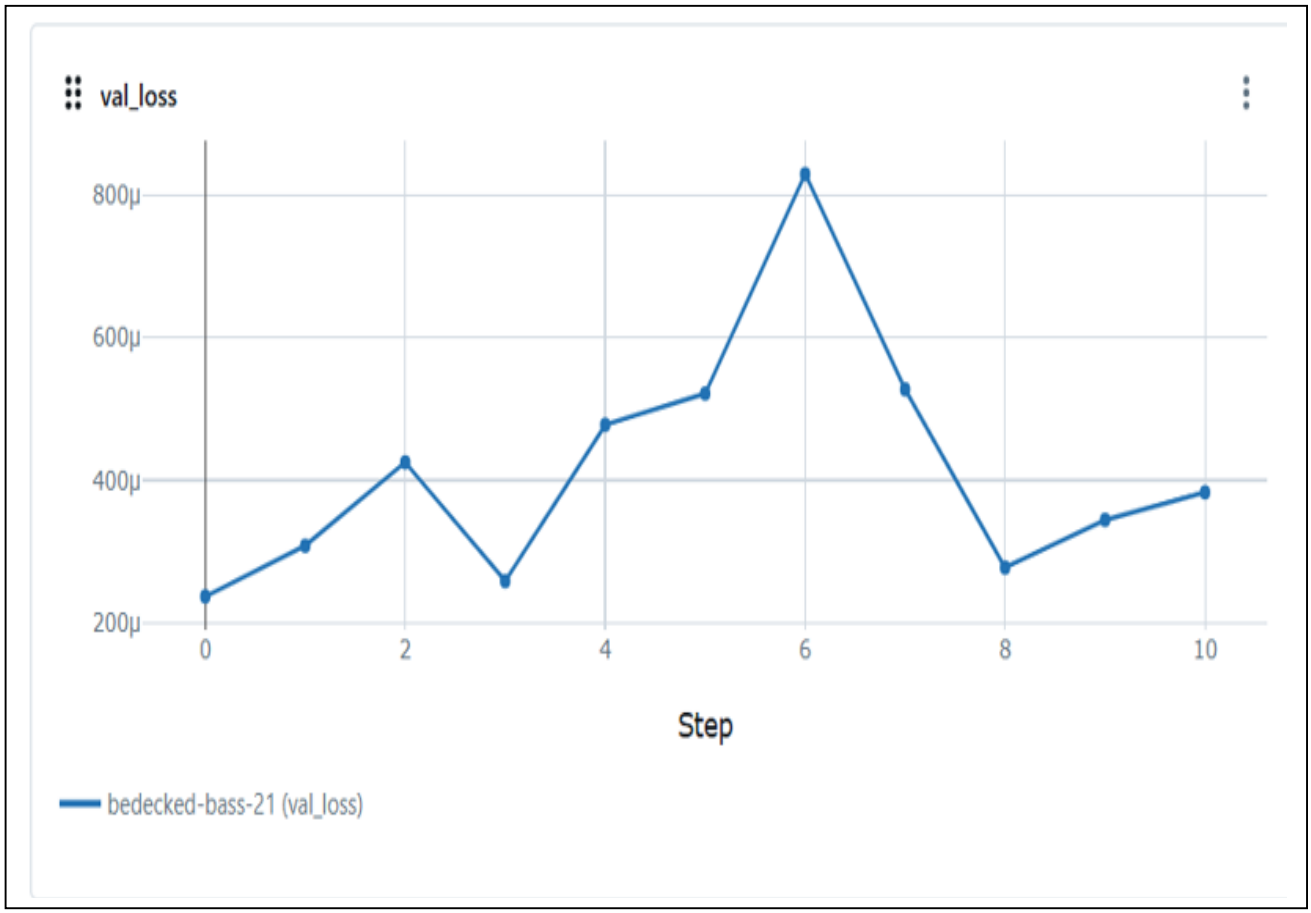
This hybrid approach ensures that our system delivers diverse and highly personalized diet plans and recipes, combining both user behavior and content attributes.

Chapter 5: Visualizations and Results

5.1 Accuracy

Model Performance:





- SVD Accuracy: 88%
- KNN Accuracy: 90%

:

Chapter 6: Technology Stack and Tools

Here's a peek into the tools and technologies we used to build and deploy our personalized diet and recipe recommendation system:

Programming Language

• Python – The backbone of our project, handling everything from data manipulation to machine learning.

Libraries for Data Manipulation and Analysis

• Pandas – For loading, cleaning, and processing datasets efficiently.

• NumPy – Used for numerical operations and managing matrix computations (especially useful in collaborative filtering).

Machine Learning and Deep Learning Libraries

• TensorFlow & Keras – For building and training the GAN models used to enhance user profiling.

• Surprise – Specialized library for collaborative filtering, implementing algorithms like SVD and KNNBasic.

Visualization Tools

• Matplotlib – For creating visualizations to analyze the results and track model performance.

Framework for Deployment

- *Streamlit – A fast and easy way to create a web interface for users to input their data and see recommendations instantly.*
- *Streamlit-Chat – Added a chat feature to engage users more interactively.*

Utilities and Supporting Libraries

- *OpenAI – For enhancing interaction, potentially integrating conversational features or smart inputs.*
- *Joblib – For saving and loading trained machine learning models efficiently.*
- *Requests – To handle HTTP requests, enabling our app to interact with APIs or external data sources.*
- *Python-Dotenv – For managing environment variables securely, keeping API keys and sensitive information safe.*

Why This Stack?

This combination of tools allowed us to build a system that is:

- *Scalable – Leveraging TensorFlow and Keras for deep learning.*
- *Interactive – Streamlit provides a seamless user experience.*
- *Efficient – Pandas, NumPy, and Surprise help with fast data processing and collaborative filtering.*

Together, these technologies ensure that our recommendation system performs well and delivers value to users in real-time.

Chapter 7: User Guide

7.1 How to Use the Diet Recommender Web App

1. Accessing the Web App

• Once the app is running (via Streamlit), open it in your browser using the local link:

http://localhost:8501

• You will see the Home Page as the starting point.

2. Navigation Menu

Located on the left side of the screen, the navigation bar contains the following sections:

1. *Home*
2. *Diet Page*
3. *Disease Page*
4. *Chatbot Page*

To switch between pages:

• Simply click on the page name in the navigation menu.

• The selected page will load immediately on the right side of the screen.

3. Page-by-Page Guide

Home Page

- *This is the welcome screen of the app, featuring the platform's logo and title.*
- *You can navigate to other pages using the menu on the left.*

Diet Page

The Diet Page helps you generate a personalized diet plan based on your information.

1. Fill out your details:

- *Age: Enter your current age.*
- *Height: Input your height (in cm).*
- *Weight: Input your weight (in kg).*
- *Gender: Select your gender from the dropdown menu.*
- *Activity Level: Choose how active you are (e.g., Sedentary, Lightly active).*
- *Objective Adjustment: Choose your goal (e.g., Weight Loss, Muscle Gain).*

2. Submit your details:

- *Click the Submit button to get your customized diet plan.*
- *A list of suitable recipes will be displayed based on your input.*

Disease Page

The Disease Page allows you to customize your diet plan based on specific health conditions.

1. Select your health condition from the dropdown (e.g., Diabetes, Heart Disease).

2. Adjust the calorie range using the slider to set your target.

3. Enter any allergies (e.g., nuts, milk) to exclude certain ingredients.

4. Click Filter Recipes to view a list of tailored recipes that match your dietary needs.

Chatbot Page

Interact with the DietBot to ask questions and get instant recommendations.

1. Enter your query in the chat box (e.g., "What meals are good for diabetes?").

2. Click Send or press Enter.

3. DietBot will respond with personalized suggestions or answers to your diet-related questions.

4. Key Features and Tips

• Real-time interactions: Use the Chatbot for quick recommendations.

• Health-focused options: If you have specific health needs, use the Disease Page for personalized suggestions.

• Diet tracking: Revisit the Diet Page to update your profile and see new recipes as your preferences change.

5. Troubleshooting & Support

• If a page doesn't load, try refreshing the browser.

• For questions, use the Chatbot to get instant help.

8.1 Advantages:

- ***Personalized Recommendations:***

• Uses advanced collaborative filtering (SVD, KNN) to recommend tailored diet plans based on user data and preferences.

- ***User-Friendly Interface****:*

• Built with Streamlit, the system is easy to use, allowing users to input data and receive recommendations in just a few steps.

- ***Adaptive User Profiles with GANs:***

• The use of GANs enhances the model's ability to adapt to changing user behavior and preferences dynamically.

- ***Health-Focused Approach:***

• Unlike generic recommendation systems, this project emphasizes dietary plans aligned with health conditions and restrictions.

- ***Continuous Learning:***

• As users provide feedback and input over time, the system improves its recommendations for better personalization.

8.2 Disadvantages:

- ***Cold Start Problem:***

• Collaborative filtering methods like SVD and KNN may perform poorly for new users or recipes without sufficient data.

- ***Limited Dataset:***

• The recommendations heavily rely on the quality and size of the dataset. With a limited dataset, the recommendations may lack diversity.

- ***Scalability Issues:***

• As the number of users and recipes increases, KNN and similar algorithms might struggle with scalability and response time.

- ***Dependency on User Input Accuracy:***

• The system's recommendations are only as good as the user input, which may result in incorrect diet suggestions if inaccurate data is provided.

8.2 Future work:

1. Integration with Gyms and Fitness Platforms

• Connecting with Gyms:
We plan to expand the system to provide customized meal plans integrated with gym workouts. Users can sync their gym schedules and workout intensity with the platform to receive optimized diet recommendations based on their physical activity levels.
• Example: If a user has a heavy workout day, the system could suggest high-protein meals and recovery foods to aid muscle growth.

• Collaboration with Fitness Apps:
The platform could be linked with popular fitness trackers (like Fitbit or Google Fit) to monitor the user's daily activity and adjust the recommendations dynamically.

2. Adding a Mental Wellness Component

• Nutrition plays a critical role in mental well-being. In future versions, the system can recommend foods that boost mood or reduce stress, aligning meals with mental health goals.
• Example: Suggesting omega-3-rich foods (like fish) to improve cognitive function or magnesium-rich recipes for stress reduction.

Chapter 9: Conclusion:

9.1 Conclusion:

Our personalized diet and recipe recommendation system offers a powerful solution to individuals looking for healthy, customized meal plans that fit their medical history and preferences. By implementing collaborative filtering techniques (SVD, KNNBasic) and leveraging GANs for dynamic user profiling, we ensure accurate, adaptive, and personalized recommendations.

Although we faced challenges, such as the cold-start problem and scalability issues, our system provides significant value through its user-friendly interface and health-centric approach. With plans to integrate with gyms and mental wellness programs, we aim to extend the impact of our platform beyond dietary suggestions to become a comprehensive lifestyle assistant.

By continuously improving through user feedback and expanding into new domains, this project has the potential to revolutionize personalized nutrition and promote long-term healthy habits for its users.