Introduction:

Obesity, malnutrition, and heart disorders are rising due to poor diet choices and lack of activity. One-size-fits-all diets often fail. This project develops an intelligent nutritional planner using AI to provide personalized diet recommendations based on BMI, height, age, weight, and gender. Integrating a nutritional textbook and local food availability, it offers tailored suggestions to help users manage weight. This approach aims to improve health, reduce diet-related diseases, and enhance well-being, particularly for Africans facing fitness and dietary challenges.

Background

The global burden of diet-related diseases has been increasing, with obesity nearly tripling worldwide since 1975. Malnutrition, including both undernutrition and overweight/obesity, is a significant issue, particularly in developing countries like Africa. Traditional dietary guidelines often fail to account for individual differences, leading to the emergence of personalized nutrition as a promising field. Artificial intelligence and machine learning have great potential to revolutionize personalized nutrition by processing vast amounts of data to generate highly personalized dietary recommendations. This project aims to develop an AI-powered nutritional planner tailored explicitly for African contexts, addressing this population's unique dietary challenges and food preferences.

Methods:

1. Data Loading: The project uses a TextLoader from langchain to load a Kenya Recipe Book from 2018.
2. Text Embedding: OpenAIEmbeddings is used to convert the loaded text data into vector representations.
3. Vector Storage: FAISS is employed to create a searchable vector store of the embedded text data.
4. Conversational AI: A ConversationalRetrievalChain is set up using ChatOpenAI and ConversationBufferMemory.
5. BMI Calculation: A function is implemented to calculate BMI based on weight and height.
6. Personalized Diet Generation: A function is created to generate personalized diet plans using OpenAI's language model, considering the user's BMI, age, height, weight, and gender and many others

Results:

The main result of this project is a functional system that can generate personalized diet plans. An example output for a 25-year-old female with a height of 175 cm and weight of 70 kg, that wants to reduce to 50 kg in 12 weeks and does not eat an oat meal. is provided. The system calculates the BMI (30.1 in this case) and generates a one-day meal plan that includes:

- Breakfast :2 pancakes made with 1 cup of wheat flour, 1 egg, 1/2 cup of cow milk, 1/8 cup of sugar, and 1/8 tsp of salt
- Lunch:Vegetable Samosa

  - 2 vegetable samosas made with cabbage, garlic, onions, carrots, ginger, peas, wheat flour, salt, and cooking oil

- Snack:Tosti Mayai(1 slice of Tosti Mayai made with 1 egg, 1/8 tsp of salt, and 1/8 cup of cooking oil)
- Dinner: Mkate Kuta(1 piece of Mkate Kuta made with wheat flour, self-rising flour, sugar, butter, mixed spices, garlic, and cooking oil)

This meal plan is designed to provide a balanced diet with less protein, as recommended for individuals with a BMI over 30.

Discussion:

The implemented system demonstrates the ability to generate personalized diet plans based on individual metrics and local food options. Here are some key points from the code:

1. BMI Calculation:

```
# Function to calculate BMI
def calculate_bmi(weight, height):
    height_in_meters = height / 100  # Convert height to meters
    bmi = weight / (height_in_meters ** 2)
    return bmi
```

This function accurately calculates BMI, which is crucial for determining the appropriate diet plan.

2.

```
# Function to get completion from OpenAI
def get_completion(prompt, model="gpt-3.5-turbo"):
    messages = [{"role": "user", "content": prompt}]
    response = openai.ChatCompletion.create(
        model=model,# Specifies the language model to use (using the value
passed as an argument to the function)
        messages=messages, #ends the formatted messages list containing
your prompt.
```

```
        temperature=0,  # this is the degree of randomness of the model's
output
    )
    return response.choices[0].message["content"]
```
This code sends a prompt to OpenAI's API to get a response from a specified language model. It sets up the message, makes an API call with defined parameters, and returns the generated content. The temperature is set to 0 for deterministic outputs, meaning no randomness.

2. Diet Generation:

```
   # Function to generate personalized diet
def generate_personalized_diet(user_id, date, data_text):
    user_data = users_collection.find_one({'user_id': user_id})
    if not user_data:
        raise ValueError(f"User with ID {user_id} does not exist.")
    age, height, weight, gender, goal, health_issues = user_data['age'],
user_data['height'], user_data['weight'], user_data['gender'],
user_data['goal'], user_data['health_issues']
    bmi = calculate_bmi(weight, height)
    # Calculate daily kilojoules intake
    kilojoules_needed = calculate_kilojoules_intake(age, height, weight,
gender, goal)
    # Get past meals
    past_meals = get_past_meals(user_id)
```

Adds a new user, verifies the addition, prepares data, and generates a daily meal plan for the user. It also includes error handling for potential issues during the recommendation process

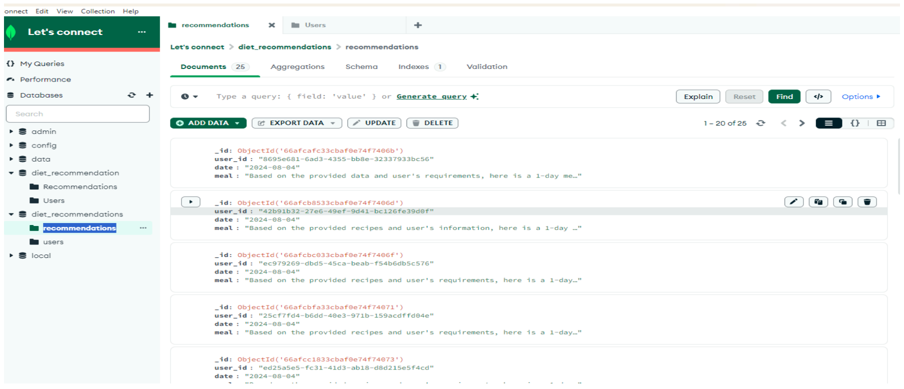
```
# Add a new user

new_user_id = add_user(25, 175, 70, 'male', 50, 12, 'does not eat
oatmeal')

# Verify that the user was added

user_data = users_collection.find_one({'user_id': new_user_id})

if not user_data:
    print(f"User with ID {new_user_id} does not exist after insertion.")
else:
  print(f"User with ID {new_user_id} successfully added.")
# Summarize and prepare the data text
data_text = summarize_data(data)
# Get the daily meal plan for the new user
try:
    daily_meal_plan = daily_recommendation(new_user_id, data_text)
    print(daily_meal_plan)
except ValueError as e:
 print(e
```

This personalized diet plan generator combines pre-processed document content and user-specific metrics (age, height, weight, gender, diet duration) to create tailored nutrition advice. The system consolidates data, defines user parameters, and employs a custom function to generate personalized diet plans. While the actual output isn't shown, it demonstrates a practical application of personalized nutrition technology. By considering individual factors, it offers more relevant dietary advice than generic plans. Future improvements could include displaying the function's logic, incorporating more user preferences, and integrating a larger nutrition database.

DATABASES:

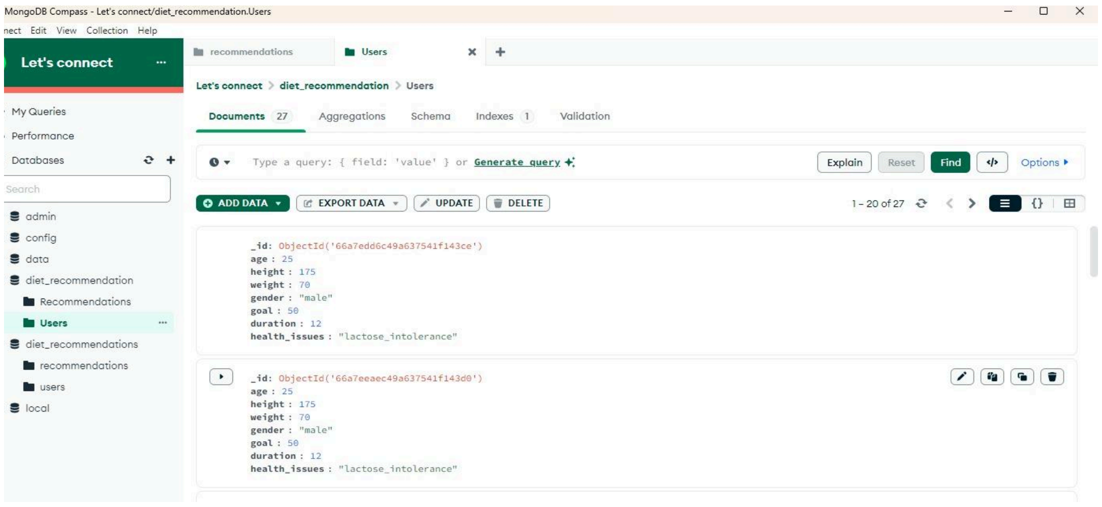
We have databases, one that stores past meal plan histories and the other one that takes in user input.

Recommendations for past history:



```
_id: ObjectId('66afcafc33cbaf0e74f7406b')
user_id : "8695e681-6ad3-4355-bb8e-32337933bc56"
date : "2024-08-04"
meal : "Based on the provided data and user's requirements, here is a 1-day me…"

_id: ObjectId('66afcb8533cbaf0e74f7406d')
user_id : "42b91b32-27e6-49ef-9d41-bc126fe39d0f"
date : "2024-08-04"
meal : "Based on the provided recipes and user's information, here is a 1-day …"

_id: ObjectId('66afcbc033cbaf0e74f7406f')
user_id : "ec979269-dbd5-45ca-beab-f54b6db5c576"
date : "2024-08-04"
meal : "Based on the provided recipes and user's requirements, here is a 1-day…"

_id: ObjectId('66afcbfa33cbaf0e74f74071')
user_id : "25cf7fd4-b6dd-40e3-971b-159acdffd04e"
date : "2024-08-04"
meal : "Based on the provided recipes and user's requirements, here is a 1-day…"

_id: ObjectId('66afcc1833cbaf0e74f74073')
user_id : "ed25a5e5-fc31-41d3-ab18-d8d215e5f4cd"
date : "2024-08-04"
```

User that in stores our user inputs.



```
_id: ObjectId('66a7edd6c49a637541f143ce')
age : 25
height : 175
weight : 70
gender : "male"
goal : 50
duration : 12
health_issues : "lactose_intolerance"

_id: ObjectId('66a7eeaec49a637541f143d0')
age : 25
height : 175
weight : 70
gender : "male"
goal : 50
duration : 12
health_issues : "lactose_intolerance"
```

Conclusion:

The implemented intelligent nutritional planner demonstrates the potential of AI in providing personalized health recommendations. By considering individual metrics and local food options, in the textbook. The system offers tailored diet plans that are both practical and potentially effective. This approach could significantly contribute to addressing obesity, malnutrition, and related health issues, particularly in African contexts. Future work could focus on expanding the food database, incorporating more complex nutritional guidelines, and validating the effectiveness of the generated diet plans through user studies.

REFERENCES:

Popkin, B. M., Adair, L. S., & Ng, S. W. (2012). Global nutrition transition and the pandemic of obesity in developing countries. *Nutrition Reviews*, *70*(1), 3–21. https://doi.org/10.1111/j.1753-4887.2011.00456.x

Steyn, N. P., & Mchiza, Z. J. (2014). Obesity and the nutrition transition in Sub‐Saharan Africa. *Annals of the New York Academy of Sciences*, *1311*(1), 88–101. https://doi.org/10.1111/nyas.12433

World Health Organization: WHO. (2024, March 1). *Obesity and overweight*. https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight