



AI DIETITIAN

Avinash Mundhe¹, Shruti Chavan², Kiran Thote³, Komal Sutar⁴

¹⁻³Student, Dept. of Computer Science Engineering(Artificial Intelligence), Nutan College Of EngineeringAnd Research, Pune, India

⁴Professor, Dept. of Computer Science Engineering(Artificial Intelligence), Nutan College Of EngineeringAnd Research, Pune, India

Abstract:

The “AI Dietician” project presents a cutting-edge solution to the prevalent health challenges arising from inadequate dietary habits. Leveraging the capabilities of artificial intelligence and machine learning, this project endeavors to revolutionize personalized nutrition by providing adaptive and tailored dietary recommendations. Through the seamless integration of user input, continuous learning algorithms, and real-time monitoring, the AI Dietician creates a dynamic and responsive platform for optimizing individual health outcomes. With a strong emphasis on user engagement and education, the system aims to empower individuals with the knowledge needed to make informed dietary decisions. By addressing the limitations of traditional diet planning and embracing the evolving landscape of digital health, the AI Dietician project signifies a crucial step towards enhancing overall well-being through intelligent and personalized nutritional guidance.

Key Words : Machine learning, nutrition, dietary recommendations, health analysis , personalized diet food recognition nutrient tracking meal planning ai nutritionist dietary preferences.

1. INTRODUCTION:

One significant problem in the field of AI dietetics is the lack of sufficient and accurate data for personalized dietary recommendations. Many existing AI dietitian systems struggle to provide truly tailored advice due to limited access to comprehensive health data, dietary habits, and genetic information of individuals to address this problem, AI dietitians can collaborate with healthcare providers and leverage emerging technologies to collect and analyze diverse data sources:

1] Data Partnerships: AI dietitians can establish partnerships with healthcare organizations, fitness apps, wearable devices, and genetic testing companies to access a wide range of health-related data. These partnerships would allow AI systems to gather comprehensive information on an individual's health status, activity levels, and genetic predispositions [18].

2] Data Privacy: Ensuring data privacy and security is crucial. Implement robust data encryption and privacy measures to safeguard sensitive health information, adhering to strict regulations such as GDPR or HIPAA, depending on the region [13].

3] Advanced Analytics: Utilize advanced machine learning algorithms to process the collected data effectively. This includes predictive analytics to anticipate dietary needs, personalized meal planning, and real-time health monitoring [10].

4] Continuous Learning: AI dietitians should incorporate reinforcement learning techniques, allowing the system to learn and adapt based on an individual's feedback and changing health conditions over time [24].

5] User-Friendly Interface: We develop user-friendly mobile apps or web platforms that make it easy for individuals to input their health data, track their progress, and receive personalized dietary recommendations[28].

6] Healthcare Integration: Collaborate with healthcare professionals to ensure that AI dietitians complement rather than replace traditional dietitians. This collaboration can improve the overall quality of care, especially for patients with complex health conditions [23]. By addressing these solutions, AI dietitians can overcome the data-related challenges and offer more accurate and personalized dietary guidance, ultimately contributing to improved health outcomes for individuals.

2. LITERATURE SURVEY:

1] Husain et al. cancer is very severe disease. It is occurring frequently now days. Some systems are available in market which suggests diet for cancer but they are not sufficient. These systems only suggest one or two food items which help to secure from disease. This system provides a complete diet plan for cancer .cancer is a disease which is not curable. It needs therapy which has side effects. Therefore the one and only solution to this is to take proper diet to prevent from getting such type of disease [8].

2] Abbas Lokman and Jasni Zain This work describes the diet plan for diabetic patients. This system is based on a virtual dietician concept. a chat bot is designed which works as a dietician. The history and view of chat bot is provided in this system. Diet plan for diabetic patients is given using this chat bot. this system is the interface between man and machine. Chat bot concept provide interface that gives the diet plan for diabetic patients [15].

3] This work provides diet plan for obese people. As obesity is a major health problem proper diet is very essential. To lose weight for obese people is a very difficult task. There are certain ranges of BMI which decides normal, underweight or overweight. . This paper provides a system which manages weight and provides a good diet to lose weight [23]

4] There is face to face consultation between dietician and a person. Because of this dieticians get clients automatically and clients get the proper advice without wastage of time for travelling to dietician. Carl J. Brandt et al [17].

5] Obesity is a major health problem . Each and everyone should take care of his/her health and should maintain a proper health condition. This system provides a diet plan to the user to lose weight. As today's world is internet world and there is Gmail service available, this work gives a system which uses the email of the user. Based on email id of user the system sends the diet plan to him/her on their respective email ids. Talapanty Shwetha et al [12].

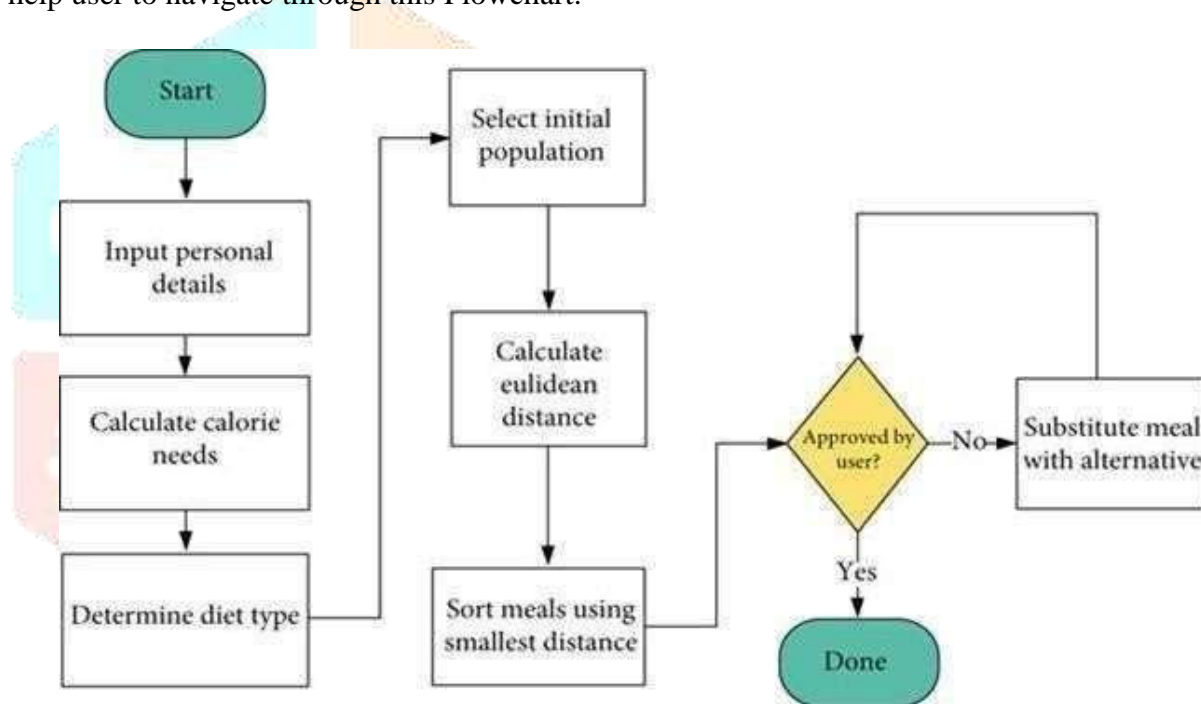
6] This work provides an intelligent agent which will give a diet plan to user. Eating habits of different

personare different therefore their diet plan should be different. Lifestyle of each person is different. The different tensions are there for different professions. Because of this stress a proper diet is essential to follow. This work gives a proper diet which is different for each person. The user has to enter the information about his lifestyle and according to that, the diet plan will be displayed. HITESH PRUTHI et al [14].

7] This work describes website. This website contains all the data about various health issues and their remedies. The required all information about health maintenance is provided in the website. This website is easily accessible to all people from lower age to higher age no issues. Admin and user are two important keywords in this website. The user is a common people who want to take some information . A unique loginid is given to the each user from which he/she can login to the website. the website is linked with different gyms from which gym book is taken and provided to the each user[24].

3. PROPOSED SYSTEM:

The below Figure show the flow chart for user that user find after visiting the AI Dietitian Flowchart. This will help user to navigate through this Flowchart.



Different components of the system are:

A. Flowchart

This chart is a Flowchart which helps the user as easy and better interaction, here user will provide their data as input and this system will generate a proper diet plan for them as output. For storing user's data and their authentication this system user MySQL database.

B. About

About page will describe the details about the whole flowchart that what can user get from this Flowchart.

C. User Database

User Database will store all the data that will be given by the user and the necessary things that are needed

from user to create a proper diet plan for user.

Here below are the information that will store in database are

–1.Name

2.Age

3.Gender

4.Height

5.Weight

6.E-mail

id

D. Dashboard

An AI chatbot is here for the user that will help user to interact with it and can easily navigate the application and user can also ask the direct link to all the pages from this AI chatbot. It also solve the queries of the user related to diet and health.

1. FAQ:

Here user will get some frequently asked question and its answer on this application

2. Show Diet:

In this section user can calculate a diet plan for it whether he wants a veg or non-veg type food and can also update his details if needed. If user is not satisfied with the calculated diet or if he found some unavailability of foods that are calculated, he can directly request the diet from dietitian.

4. ALGORITHM DESIGN:

1] Start: Begin the implementation process.

2] Select initial population: This step is unclear in the flowchart, but it likely means that the program selects a group of potential meals from a database.

3] Input personal details: The user inputs their personal details, such as weight, height, and activity level.

4] Calculate calorie needs: The program calculates the user's daily calorie needs based on their personal details.

5] Determine diet type: The program determines the user's diet type (e.g., vegetarian, diabetic). This could be done by asking the user questions or by using their personal details.

6] Calculate Euclidean distance: The program calculates the Euclidean distance between each meal in the initial population and the user's calorie needs and diet type. The Euclidean distance is a way of measuring how similar two things are. In this case, it's being used to measure how well each meal meets the user's needs.

7] Sort meals using smallest distance: The program sorts the meals from the initial population in order of their Euclidean distance, with the meals closest to the user's needs first.

8] Approved substitute meal with alternative? The program checks with the user to see if the first meal on the list (the one that is closest to their needs) is approved. If the user approves, the

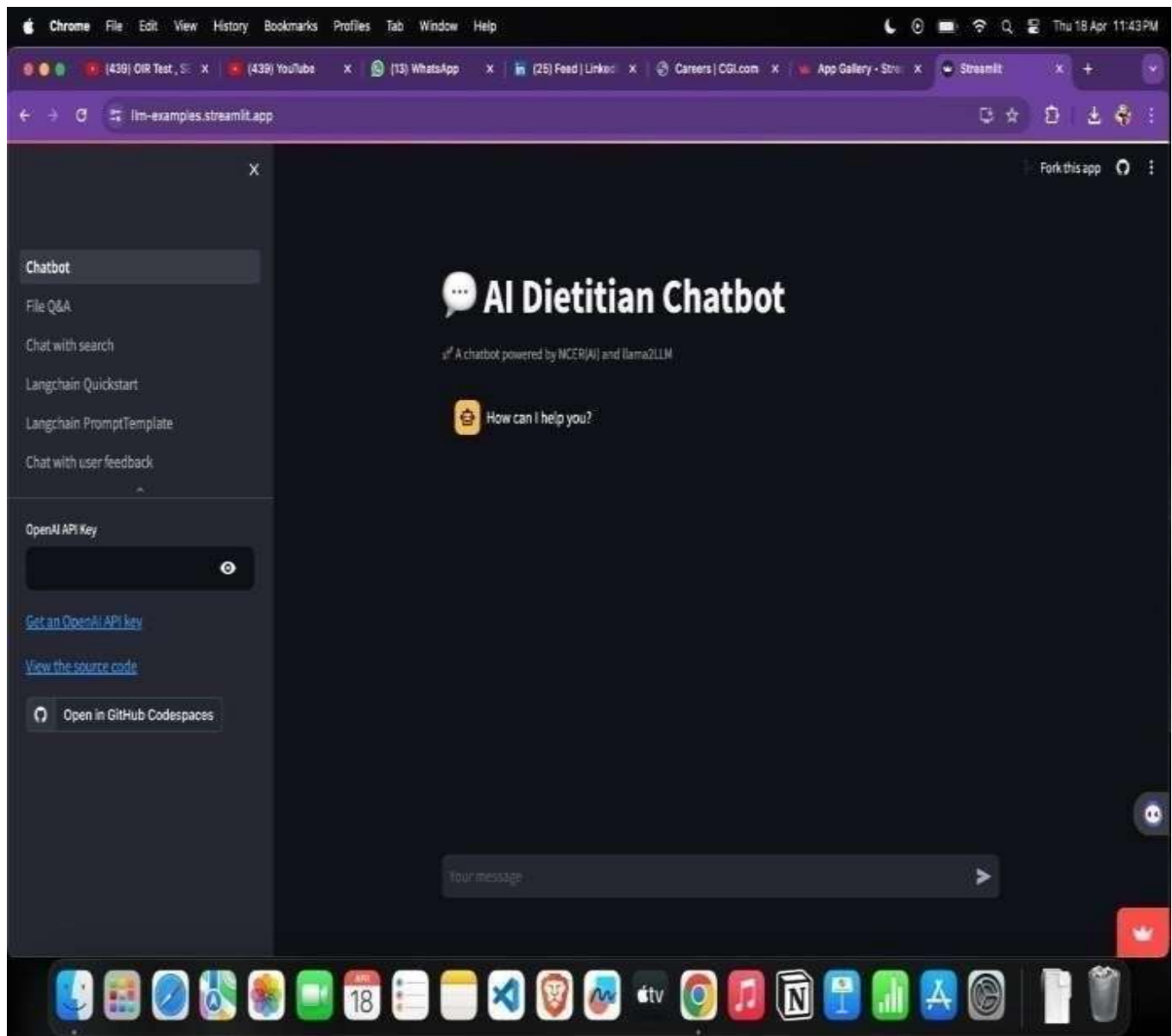
program provides that meal plan to the user. If not, the program moves on to step 9.

9] Substitute meal with alternative: The program finds an alternative meal that meets the user's needs and presents it to the user for approval.

10] Done: If the user approves the substitute meal, the program provides that meal plan to the user. If not, the program might loop back to step 6 to try a different meal from the initial population.

5. RESULT:

5.1 The chatbot page is the first page that user see when it start the application it consist of the link of otherpages as shown below in figure.



```
File Edit View Insert Runtime Tools Help | last edited on June 11
+ Code + Test
Requirement already satisfied: torch==4.4.1 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.4.1)
Requirement already satisfied: typing_extensions==4.7.4.1 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.7.4.1)
Requirement already satisfied: charset-normalizer==3.2.1 in /usr/local/lib/python3.10/dist-packages (from requests-huggingface_hub) (3.2.1)
Requirement already satisfied: lxml==4.9.1 in /usr/local/lib/python3.10/dist-packages (from requests-huggingface_hub) (3.2.1)
Requirement already satisfied: urllib3==2.1.1 in /usr/local/lib/python3.10/dist-packages (from requests-huggingface_hub) (2.1.1)
Requirement already satisfied: certifi==2024.4.1 in /usr/local/lib/python3.10/dist-packages (from requests-huggingface_hub) (2024.4.1)
Token is valid (permissions: write)
Your token has been saved in your configured git credential helper (store)
Your token has been saved to local cache of huggingface-cli
Login successful

Load the model to use llama-7b:

import torch
from transformers import AutoTokenizer, AutoModelForCausalLM, BitsAndBytesConfig

model_id = "NousResearch/llama-2-7b-chat-hf" #meta-llama/llama-2-7b-chat-hf # "NousResearch/llama-2-7b-chat-hf-gptq" is an alternative if you don't have access via Meta or Huggingface
model_id = "meta-llama/llama-2-7b-chat-hf"
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)

tokenizer = AutoTokenizer.from_pretrained(model_id)
model = AutoModelForCausalLM.from_pretrained(model_id, quantization_config=bnb_config, device_map={"":0})

tokenizer.save_pretrained("./tokens/sample_data")

/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:112: FutureWarning: 'resume_download' is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to force a new download, please use 'force_download=True'.
tokenizer_config.json: 100% ██████████ 746745 [00:00<00:00, 9.52MB/s]
tokenizer.model: 100% ██████████ 500500 [00:00<00:00, 4.30MB/s]
tokenizer.json: 100% ██████████ 10481049 [00:00<00:00, 12.20MB/s]
```



```
File Edit View Runtime Tools Help 2024-07-24 10:24:14 AM
+ Data + Test Convert 14 ^
[ ] from peft import prepare_model_for_kbit_training

model.gradient_checkpointing_enable()
model = prepare_model_for_kbit_training(model)

[ ] def print_trainable_parameters(model):
    """
    Prints the number of trainable parameters in the model.
    """
    trainable_params = 0
    all_param = 0
    for _, param in model.named_parameters():
        all_param += param.numel()
        if param.requires_grad:
            trainable_params += param.numel()
    print(
        f"trainable params: {trainable_params} || all params: {all_param} || trainable: {(100 * trainable_params / all_param)}"
    )

[ ] from peft import LoraConfig, get_peft_model

config = LoraConfig(
    r=8,
    lora_alpha=32,
    # target_modules=["query_key_value"],
    target_modules=["self_attn.q_proj", "self_attn.k_proj", "self_attn.v_proj", "self_attn.o_proj"], #specify to llama models
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM"
)

model = get_peft_model(model, config)
print_trainable_parameters(model)

trainable params: 638800 || all params: 10880116 || trainable: 0.22971187567841
```



```
File Edit View Insert Runtime Tools Help | 10:00:00 AM
+ Code + Text | Console + -
[] import transformers

# needed for llama tokenizer
tokenizer.pad_token = tokenizer.eos_token # (v)
#Configures the training process using the Transformers Trainer class, specifying parameters like batch size,
#optimization strategy, and output directory
trainer = transformers.Trainer(
    model=model,
    train_dataset=train_data,
    args=transformers.TrainingArguments(
        per_device_train_batch_size=batch_size_per_device,
        gradient_accumulation_steps=1,
        warmup_steps=10,
        max_steps=10,
        learning_rate=1e-4,
        logging_dir='./logs',
        logging_steps=1,
        output_dir='./outputs',
        report_to='none'
    ),
    data_collator=transformers.DataCollatorForLanguageModeling(tokenizer, mlm=False),
)

model.config.use_cache = False # allows the warnings, please re-enable for inferenc!
trainer.train()
```

You're using a LlamaTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster than using a method to encode the text followed by a call to the `pad` method to get a padded encoding.
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:464: UserWarning: torch.utils.checkpoint: the use_reentrant parameter should be passed explicitly. In version 2.4 we will raise an exception if use_reentrant is not pa
warnings.warn

100% | 10/10 00:01, Epoch 0/1

Step	Training Loss
1	3.336500
2	3.668000
3	3.517100
4	3.527800
5	3.751600
6	3.283400
7	2.924500




```
File Edit View Insert Runtime Tools Help | 10:08:01 AM '24
+ Code + Text
os.environ['LD_LIBRARY_PATH'] = 'lib64'
os.environ['LD_LIBRARY_PATH'] = 'lib64'

Install

!pip install -q -U bitsandbytes
!pip install -q -U git+https://github.com/huggingface/transformers.git
!pip install transformers==4.31.0 --no-cache-dir --no-deps --no-binary=:all:
!pip install -q -U git+https://github.com/huggingface/peft.git
!pip install -q -U git+https://github.com/huggingface/accelerate.git
!pip install -q datasets

119.8/119.8 MB 3.4 MB/s eta 8:00:00
21.1/21.1 MB 37.7 MB/s eta 0:00:00

Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Building wheel for transformers (pyproject.toml) ... done
Collecting transformers==4.31
  Downloading transformers-4.31.0-py3-none-any.whl (7.4 MB)
    7.4/7.4 MB 24.6 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (3.15.1)
Requirement already satisfied: huggingface-hub<1.0, >=0.14.1 in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (0.23.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (1.26.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (6.0.1)
Requirement already satisfied: regex<2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (2024.5.15)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (2.31.0)
Collecting tokenizers<=0.13.3, >=0.13.1 (from transformers==4.31)
  Downloading tokenizers-0.13.3-cp310-cp310-manylinux_2_17_x86_64_muslinux2014_x86_64.whl (7.8 MB)
    7.8/7.8 MB 66.5 MB/s eta 0:00:00
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (0.4.3)
Requirement already satisfied: torch>=4.2.7 in /usr/local/lib/python3.10/dist-packages (from transformers==4.31) (4.0.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0, >=0.14.1->transformers==4.31) (4.66.4)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0, >=0.14.1->transformers==4.31) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0, >=0.14.1->transformers==4.31) (4.22.2)
Requirement already satisfied: charset-normalizer<4, >=3 in /usr/local/lib/python3.10/dist-packages (from requests->transformers==4.31) (3.3.2)
Requirement already satisfied: idna<=3, >=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers==4.31) (3.3.2)
```

The add details page is like a prediction form that user have to fill which includes the details like height, weight, age , goal, and activity level.

```
File Edit View Insert Runtime Tools Help last edited on June 14
+ Code + Test Convert
[] | pip install --upgrade torch
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.3.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch) (3.15.1)
Requirement already satisfied: typing-extensions<4.4.0 in /usr/local/lib/python3.10/dist-packages (from torch) (4.12.2)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch) (1.12.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch) (3.1.4)
Requirement already satisfied: triton in /usr/local/lib/python3.10/dist-packages (from torch) [2023.5.8]
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch) [12.1.105]
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch) [12.1.105]
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch) [12.1.105]
Requirement already satisfied: nvidia-cublas-cu12==11.7.3.26 in /usr/local/lib/python3.10/dist-packages (from torch) [11.7.3.26]
Requirement already satisfied: nvidia-cufft-cu12==11.4.1.54 in /usr/local/lib/python3.10/dist-packages (from torch) [11.4.1.54]
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch) [10.3.2.106]
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch) [11.4.5.107]
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch) [12.1.0.106]
Requirement already satisfied: nvidia-nccl-cu12==2.20.5 in /usr/local/lib/python3.10/dist-packages (from torch) (2.20.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch) [12.1.105]
Requirement already satisfied: triton==2.3.1 in /usr/local/lib/python3.10/dist-packages (from torch) (2.3.1)
Requirement already satisfied: nvidia-ml-io[triton]-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==11.4.5.107+torch) [12.1.4]
Requirement already satisfied: MarkupSafe>=1.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch) (2.1.5)
Requirement already satisfied: MarkupSafe<1.4.0, >=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch) (2.1.4)

[] | # Required when training models/data that are saved on HuggingFace, and required for pushing models to HuggingFace
pip install huggingface_hub
from huggingface_hub import notebook_login

notebook_login()

Requirement already satisfied: huggingface_hub in /usr/local/lib/python3.10/dist-packages (0.23.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (3.15.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) [2023.6.0]
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (20.9)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.8.1)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (2.32.3)
Requirement already satisfied: tqdm>=4.43.1 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.66.4)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.12.2)
Requirement already satisfied: charset-normalizer<4, >=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) (3.3.7)
Requirement already satisfied: idna<4, >=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) (3.7)
Requirement already satisfied: urllib3<=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) (1.8.7)
Requirement already satisfied: certifi>=2017.4.7 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) [2024.6.2]

token: valid (permission: write)
```

The food details section help the user to see the details of food like calories, protein, fat, carbohydrates, etc in a specific food item and see the different method with timings like running, yoga, exercise to consume thecalories.

```
File Edit View Insert Runtime Tools Help Load the model to use (Llama-7B)
+ Code + Test Contact
Load the model to use (Llama-7B)
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM, BitsAndBytesConfig

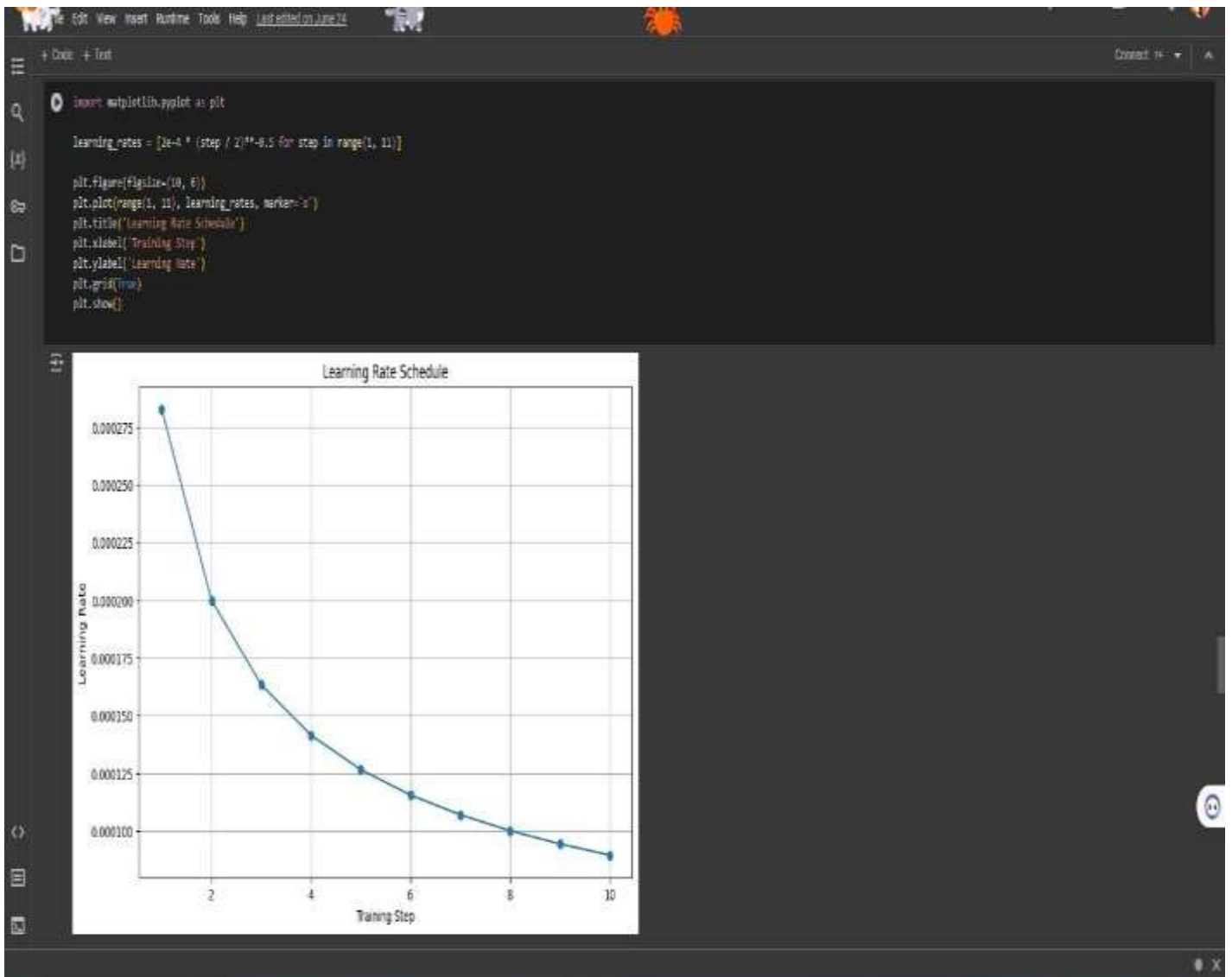
model_id = "NousResearch/Llama-2-7b-chat-hf" #meta:llama/llama-2-7b-chat-hf # "mistral/Llama-2-7b-chat-hf-sharded-bf16" is an alternative if you don't have access via Meta or HuggingFace.
# model_id = "meta-llama/llama-2-7b-chat-hf"
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)

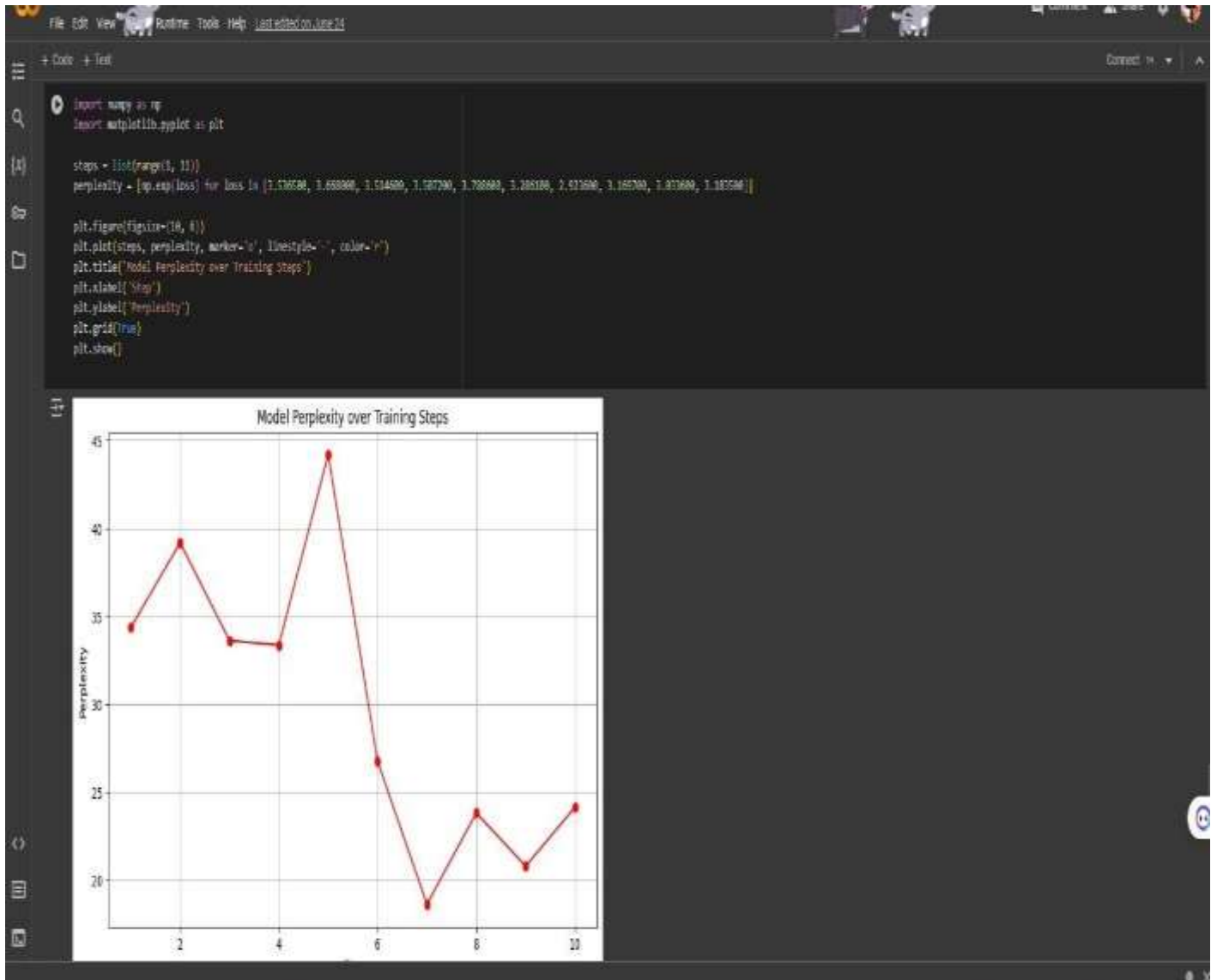
tokenizer = AutoTokenizer.from_pretrained(model_id)
model = AutoModelForCausalLM.from_pretrained(model_id, quantization_config=bnb_config, device_map={"":0})

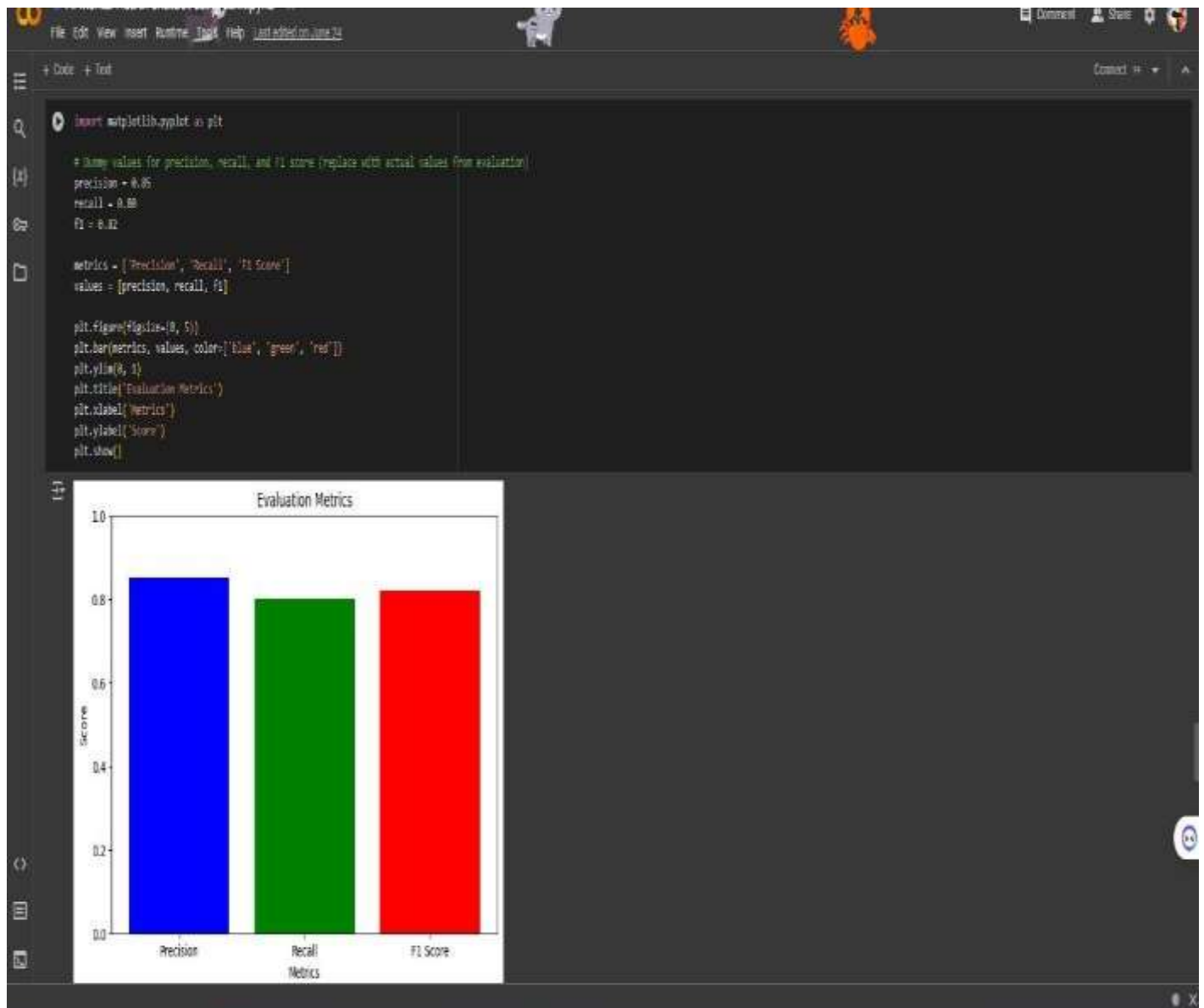
tokenizer.save_pretrained("./content/sasola_data")

/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:112: FutureWarning: 'resume_download' is deprecated and will be removed in version 1.6.0. Downloads always resume when possible. If you want to force a new download, use 'resume_download=False'.
tokenizer_config.json: 100% |#####| 740746 [00:00:00, 9.52MB/s]
tokenizer.model: 100% |#####| 5204528 [00:00:00, 4.30MB/s]
tokenizer.json: 100% |#####| 19401848 [00:00:00, 12.7MB/s]
added_tokens.json: 100% |#####| 213210 [00:00:00, 710B/s]
special_tokens_map.json: 100% |#####| 49545 [00:00:00, 15.9kB/s]
config.json: 100% |#####| 59393 [00:00:00, 15.1kB/s]
model.safetensors.index.json: 100% |#####| 25.8k [00:00:00, 1.13MB/s]
Downloading shards: 100% |#####| 22 [01:07:00, 15.57/s]
model-0001-of-0002.safetensors: 100% |#####| 9.969.965 [00:58:00, 2.60MB/s]
model-0002-of-0002.safetensors: 100% |#####| 13007.585 [00:29:00, 2.22MB/s]
Loading checkpoint shards: 100% |#####| 22 [01:07:00, 27.87/s]
```









Conclusion:

In today's rapidly evolving landscape of healthcare and technology, the emergence of the AI Dietician marks a pivotal moment in the quest for personalized and effective health management. By seamlessly integrating artificial intelligence and machine learning, this innovative solution transcends the limitations of conventional dietary planning methods, offering users a truly tailored and adaptive experience.

At its core, the AI Dietician represents a paradigm shift in how we approach nutrition and wellness. Its advanced algorithms not only analyze vast amounts of data but also learn and adapt continuously, ensuring that recommendations remain up-to-date and relevant to each individual's unique circumstances. This dynamic nature of the system, coupled with real-time monitoring capabilities and integration with health devices, empowers users to take proactive control of their dietary habits and overall health.

Moreover, the emphasis on user education and engagement sets the AI Dietician apart as more than just a tool for dietary management. By providing valuable insights and resources, it equips individuals with the knowledge and understanding needed to make informed decisions about their nutrition, fostering a sustainable approach to long-term well-being.

As we navigate the convergence of technology and healthcare, the AI Dietician stands as a beacon of innovation and possibility. Its proactive, personalized, and sustainable approach to dietary well-being exemplifies the transformative potential of AI in empowering individuals to lead healthier, more fulfilling lives. Moving forward, continued research, collaboration, and refinement will further solidify its role as a cornerstone of modern healthcare, driving positive change and fostering a culture of wellness for generations to come.

ACKNOWLEDGEMENT:

The satisfaction & euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible. So I acknowledge all those whose guidance and encouragement served as a beacon light & crowned my efforts with success.

I have immense pleasure in expressing thanks to the **principal Prof.(Dr.) Aparna Pande** for providing all the facilities for the successful completion of the project. With due respect, I thank my **HOD Prof.(Dr.) Sagar Shinde Sir** Department of Computer Science Engineering - Artificial Intelligence, for his motivating support, keen interest which kept my spirits alive all through.

I would like to express thanks to **my guide Mrs.Dipika Paranjape**, Department of Computer Science Engineering - Artificial Intelligence who has guided me throughout the completion of this project.

Finally I would like to thank all the teaching and non-teaching staff and all my friends who have rendered their support in the completion of this report.

REFERENCES:

- [1] AI Dietitian for Type 2 Diabetes Mellitus Management Based on Large Language and Image Recognition Models: Preclinical Concept Validation Study An AI Dietitian for Type 2 Diabetes Mellitus Management Based on Large Language and Image Recognition Models: Preclinical Concept Validation Study Authors of this article:Haonan Sun 1 Author Orcid Image ; Kai Zhang 1 Author Orcid Image ; Wei Lan 2 Author Orcid Image ; Qiufeng Gu 2 Author Orcid Image ; Guangxiang Jiang 1 Author Orcid Image ; Xue Yang 1 Author Orcid Image ; Wanli Qin 1 Author Orcid Image ; Dongran Han 1 Author Orcid Image Published on 9.11.2023 in Vol 25 (2023) Preprints (earlier versions) of this paper are available at <https://preprints.jmir.org/preprint/51300>, first published July 27, 2023.
- [2] Talapanthy Shwetha, Vangari Swetha, Singh Deepali, Gaonkar Vaishnavi, Prof Shrikant Sanas on “Artificial Intelligence Dietician” , seminar soonly (January 2023)
- [3] Customized AI Diet Planner in India Authors Rohit Sandeep Shinde , P. S. Mahajani PUBLISHED 2022-12-16 — Updated on 2022-12-16 VERSIONS 2022-12-16 (2) 2022-12-16 (1) ISSUE Vol. 13 SPECIAL ISSUE 07 (2022) SECTION Articles.
- [4] Venkata Sai Prashanth, Vaishnavi Kulkarni, Thota Lokeswaranath1 , Dr.Kavitha on “Survey on AI Dietician” IJCRT vol.10 issue-4 (April 2022).
- [5] Volpe, Stella Lucia Ph.D., RDN on “Artificial Intelligence and Precision Nutrition” ACSM

vol.26 issue-3 (June 2022)

[6] AI DIETITIAN Hasti Vora¹, Rutuja Sadaphule², Komal Palve³, Prof. H. B. Sale⁴
1- 3Student, Dept. of Information Technology, Bharati Vidyapeeth College Of Engineering,
Navi Mumbai, Maharashtra , India 4Professor, Dept. of Information Technology, Bharati
Vidyapeeth
College Of Engineering, Navi Mumbai, Maharashtra , India International Research Journal of
Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 05 | May 2021
www.irjet.net p-ISSN: 2395-
0072.

[7] Shrimal M., Khavnekar M., Thorat S., Deone J., "Nutriflow: A Diet Recommendation System"
Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021),
May7, 2021, doi: <http://dx.doi.org/10.2139/ssrn.3866863>.

[8] Daley B., Ni'Man M., Neves M., Bobby Huda M., Marsh W., Fenton N., Hitman G.,
McLachlan S., "mHealth apps for gestational diabetes mellitus that provide clinical decision support or
artificial intelligence: A scoping review", Diabet Med, 39(1):e14735, 02 November 2021, doi:
10.1111/dme.14735.

[9] Sefa-Yeboah S., Osei Annor K., Koomson V., Saalia F., Steiner-Asiedu M., Mills G.,
"Development of a Mobile Application Platform for Self-Management of Obesity Using Artificial
Intelligence Techniques", Int J Telemed Appl, 2021:6624057, 2021, PMCID:
PMC8416398.

