iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70549

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

AI Powered Weight Based Meal Recommendation QnA Chatbot using Pre-Trained Language Model

Yashraj Mishra[1], Ankita Jaiswal[2], Anany Shukla[3], Abhishek Verma[4], Humesh Verma[5], Dr. Goldi Soni[6]
Amity School of Engineering and Technology, Amity University Chhattisgarh

Abstract: ***In recent years, artificial intelligence (AI) has revolutionized personalized health and nutrition systems by integrating machine learning and natural language processing (NLP). This research introduces an AI-powered, weight-based meal recommendation and question-answering (QnA) chatbot system, developed using pre-trained transformer models (T5-base and T5-large). The system aims to assist users in receiving personalized dietary recommendations based on gender and BMI-based weight categories (Underweight, Normal weight, Overweight, Obesity) and offers interactive responses to common health-related queries. The chatbot engine is trained using a custom dataset of 30 question-answer pairs for each of the 8 user categories (based on gender and weight classification), stored and retrieved dynamically using MongoDB. The model leverages Transfer Learning with T5-base and T5-large, both fine-tuned for sequence-to-sequence tasks. Despite its lighter structure, T5-base remains suitable for low-resource devices with as little as 8GB RAM and integrated GPU, offering scalability and accessibility for broader deployment. The system is served through a Flask backend integrated with a responsive front-end developed using HTML, CSS, and JavaScript, enabling real-time user interaction. The final solution demonstrates the viability of transformer-based language models in healthcare QnA and recommendation systems with minimal computational overhead. This framework provides a blueprint for future developments in AI-powered nutrition systems and intelligent health assistants.***

Keywords: T5 Transformer, Question Answering Chatbot, Meal Recommendation System, Pre-trained Language Model, Weight-Based Personalization, Natural Language Processing (NLP), Flask Web Application, MongoDB Integration.

I. INTRODUCTION

In today's fast-paced world, maintaining a healthy lifestyle has become a growing concern, especially with the rise of diet-related diseases such as obesity, diabetes, and cardiovascular disorders. A well-balanced diet tailored to individual needs plays a vital role in health management. However, most individuals struggle with identifying the right type of meal based on their body type, gender, and lifestyle. This gap has created a pressing need for personalized dietary guidance that is not only accurate but also easily accessible. Leveraging Artificial Intelligence (AI) and Natural Language Processing (NLP) offers a promising solution to bridge this gap by providing intelligent, interactive, and customized meal recommendations. This research presents an innovative approach to this problem through the development of an AI-powered chatbot system that performs two key tasks:

1) Weight-based meal recommendations
2) Interactive QnA (Question and Answering) system

The system classifies users into eight distinct categories based on gender (male/female) and weight status (underweight, normal weight, overweight, obesity) and provides tailored recommendations accordingly. It allows users to ask health-related questions, to which the chatbot provides accurate, context-aware answers using a transformer-based language model. To build this system, we employed T5 (Text-To-Text Transfer Transformer), a pre-trained sequence-to-sequence model developed by Google Research. Two variants were used in the experimentation:

- T5-base (220 million parameters): Lightweight, efficient, and suitable for devices with limited resources.
- T5-large (770 million parameters): More accurate but computationally heavier.

Each model was fine-tuned on a custom dataset of 30 question-answer pairs per user category, resulting in a total of 240 high-quality QnA examples stored in MongoDB for efficient retrieval and processing.

The backend of the application is powered by Flask, which acts as a server interface for model prediction and data handling. The frontend is designed using HTML, CSS, and JavaScript, offering a user-friendly interface where users can select their weight category and interact with the chatbot seamlessly. The significance of this work lies in its scalability, low hardware dependency, and domain-specific applicability, making it feasible for deployment even on low-spec machines (as low as 8GB RAM and an integrated GPU). This project thus demonstrates the power of pre-trained language models in developing smart, interactive healthcare solutions, and sets the foundation for further exploration in personalized AI assistants for nutrition, fitness, and wellness domains.

II. LITERATURE SURVEY

Artificial Intelligence and Natural Language Processing have become foundational in building smart systems across healthcare, nutrition, and fitness domains. This section reviews prior work related to dietary recommendation systems, QnA systems using pre-trained models, and transformer-based language models, particularly in context with healthcare applications.

Traditional rule-based diet recommendation systems used fixed parameters such as BMI, calorie count, and macronutrient distribution to provide user-specific suggestions [1]. While effective in structured environments, these systems lack the flexibility and adaptability of AI-powered approaches. Recent advancements have shown that integrating machine learning models can enhance recommendation personalization and user satisfaction [2].

In parallel, chatbot systems have seen a revolution with the adoption of pre-trained transformer models. Models such as BERT, GPT, and T5 have significantly improved the ability of systems to understand and generate human-like responses. Specifically, T5 (Text-to-Text Transfer Transformer) introduced by Raffel et al. reformulates all NLP tasks into a unified text-to-text format, which simplifies task modeling and improves generalization [3].

For health and nutrition-based question-answering systems, domain-specific fine-tuning has proven highly effective. For instance, studies have used BERT-based QA models trained on medical datasets like SQuAD-Med or HealthQA to achieve high semantic accuracy [4]. However, such models often require extensive computational resources. This makes lightweight alternatives like T5-base valuable, particularly for deployment on consumer-grade hardware.

A related study by Saha et al. developed a BERT-based chatbot for dietary assistance and found that semantic fluency and contextual relevance improve with larger models, albeit with increased computational costs [5].

Moreover, the use of MongoDB in NLP applications is growing due to its flexibility in storing semi-structured QnA datasets, as demonstrated in recent chatbot implementations for healthcare and e-commerce domains [6].

These developments collectively underline the value of transformer-based models in building efficient, scalable, and interactive dietary recommendation systems, especially when integrated with real-time chat interfaces and categorized health data.

III.PROPOSED METHODOLOGY

The proposed system is designed to deliver personalized meal recommendations and question-answering capabilities based on a user's gender and BMI category (Underweight, Normal weight, Overweight, Obesity). This hybrid framework leverages the power of T5-base and T5-large models for natural language understanding and generation. The methodology is divided into the following stages: Data Collection & Preprocessing, Model Training, Evaluation & Validation, and Deployment via Flask Web Interface.

A. Data Collection and Categorization

The QnA data is structured and stored in MongoDB, a NoSQL database suitable for flexible document-based storage. The database consists of 8 collections, categorized by gender and BMI class:

- Male_Underweight, Male_Normalweight, Male_Overweight, Male_Obesity
- Female_Underweight, Female_Normalweight, Female_Overweight, Female_Obesity

Each document within the collection includes a question and its corresponding answer, formatted as:

```
{
  "question": "What should I eat in the morning?",
  "answer": "Oatmeal with fruits and nuts is a good start for your day."
}
```

These are transformed into a sequence-to-sequence format:

Input → "Question: <user_question>"

Output → "Answer: <corresponding_answer>"

This format aligns with the T5 architecture that treats every NLP problem as a text-to-text task.

B. Model Architecture and Training

Two pre-trained models from Hugging Face — T5-base (220M parameters) and T5-large (770M parameters) — were fine-tuned on the domain-specific dataset.

Training Details:

- Tokenization: Using AutoTokenizer from HuggingFace Transformers
- Max sequence length: 32 tokens (both input and output)
- Teacher forcing: Padding tokens set to -100 to be ignored in loss computation
- Training epochs: 10
- Batch size: 2
- Optimizer: AdamW (default via Trainer class)
- Framework: HuggingFace Trainer API on PyTorch backend

The fine-tuning code follows a typical supervised training paradigm where tokenized input_text and output_text pairs are fed into the model. The Trainer handles gradient updates and checkpointing. The model and tokenizer are saved locally.
This process was repeated for both T5-base and T5-large. The larger model required more memory but showed improved semantic coherence.

C. Response Generation and Manual Evaluation

Post-training, the saved models were validated manually using a custom test.py script. This script loads the model and tokenizer, encodes a sample input, and decodes the generated output using beam search with 4 beams. Example input:

```
"Question: What should I eat in the morning?
Category: Male_Overweight"
```

This ensures that meal recommendations are not generic but instead tailored to a user's gender and BMI group, matching the correct MongoDB collection the model was trained on.

D. Frontend and Backend Integration (Deployment)

The trained model was deployed using a Flask backend, providing a lightweight web server to serve inference results. Key deployment steps:

- Frontend: HTML, CSS, JavaScript to build the user interface where users input questions and view responses.
- Backend: Flask handles HTTP requests, loads the saved T5 model, and passes the user's query for inference.
- User Profile: Stored in MongoDB, including attributes such as name, gender, height, weight, and BMI.
- BMI Calculation: Done at registration or login, and used to route user queries to the appropriate model response context (e.g., Male_Obesity).

The application runs smoothly on systems with 8GB RAM and an integrated GPU, due to the model's optimization and compact inference footprint.

IV. FLOW DIAGRAM

Our proposed methodology pipeline represents:

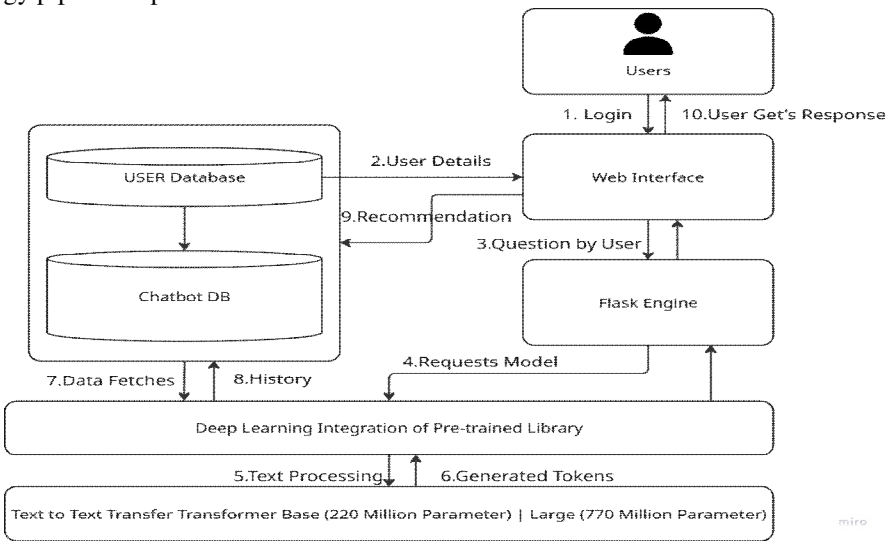


Fig 1 shows the architecture flowchart of model

1) User Login: The process begins when the user logs in to the system via a web interface. User credentials and BMI-related information are either retrieved or updated in the USER Database.
2) User Details Accessed: Upon successful login, the system fetches the user's BMI classification (Underweight, Normal, Overweight, Obese) and gender (Male/Female) from the USER Database. This information is used to personalize the recommendation and chatbot behavior.
3) Question by User: The user interacts with the chatbot by asking a question (e.g., *"What should I eat for breakfast?"*). This query is submitted through the Web Interface and routed to the Flask Engine.
4) Request Model: The Flask Engine constructs a model request using the user's question and category information. This request is then sent to the Deep Learning Integration Module.
5) Text Processing: The T5-base or T5-large model (depending on the chosen configuration) tokenizes and processes the input question. This step includes semantic encoding, token generation, and sequence formatting using the pre-trained transformer.
6) Generated Tokens: The input is transformed into generated tokens which are decoded into a human-readable answer. These tokens represent the model's prediction/response for the user's query.
7) Data Fetches: In parallel, relevant data is fetched from the Chatbot Database (Chatbot DB) which includes historical QnA, meal suggestions, and previously stored conversation patterns.
8) History: The system maintains a history of the user's previous interactions and recommendations for contextual learning or personalization in future responses.
9) Recommendation Delivery: Based on the user's category and question, the response or meal recommendation is compiled. The Flask engine returns this final response back to the Web Interface.
10) User Gets Response: The user receives the final, AI-generated response or suggestion via the web frontend. This completes one full interaction cycle.

V. RESULTS AND EVALUATION

The performance of the proposed meal recommendation and QnA chatbot system was evaluated based on the semantic correctness and accuracy of the generated answers. Two different transformer-based models — T5-base and T5-large — were fine-tuned and compared.

A. *Evaluation Metrics*

For the evaluation, we considered the following:

- Accuracy (%): The proportion of correctly generated answers that match expected semantic output.
- Manual Evaluation: Human judgment was used to evaluate whether generated responses were contextually and semantically accurate, even in cases of minor lexical differences.
- Repetition Rate: Observation of word redundancy or phrase repetition in model outputs.
- Resource Utilization: Time, memory, and computational resources required for training and inference.

B. *Model Comparison*

Model	Parameters	Accuracy (%)	Observations
T5-base	220M	90%	Occasional repetition of phrases but semantically correct. Lightweight and easy to deploy.
T5-large	770M	93%	More fluent responses with improved coherence. Requires higher memory and training time.

- The T5-large model achieved 3% higher accuracy, benefiting from a greater number of parameters and deeper contextual understanding.
- However, T5-base remains a favourable choice for deployment on lower-end hardware due to its smaller size and comparable performance.

C. *Example Outputs*

Sample Input:
Question: What should I eat in the morning?
Category: Male_Overweight
T5-base Output:
Answer: You can eat oats with fruits and green tea for energy.
T5-large Output:
Answer: A bowl of oatmeal topped with berries and a boiled egg is ideal for overweight males in the morning.

D. *Graphical Representation*

To better visualize the performance, the following graphs were plotted:

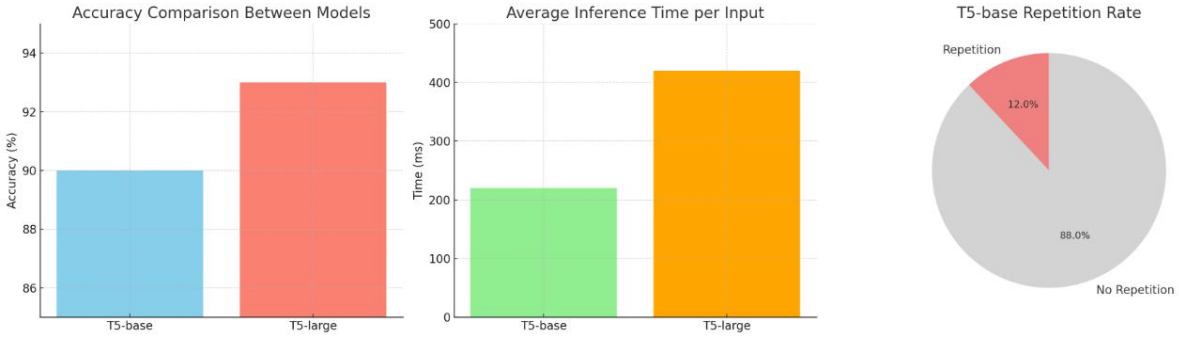


Figure 2: Graph illustrating the accuracy comparison between models and average inference time per input

Here are the graphical representations:

- Accuracy Comparison: T5-base achieved 90% accuracy. T5-large improved upon this with 93% accuracy.
- Inference Time per Input: T5-base model averaged around 220ms per input. T5-large model required 420ms, reflecting its larger size and complexity.
- Word Repetition Rate (T5-base): Roughly 12% of generated responses included repeated words, although they retained correct semantic meaning.

E. *Deployment Feasibility*

Despite the increased performance of T5-large, the T5-base model was selected for final deployment due to the following reasons:

- Achieved 90% accuracy, which meets functional performance thresholds.
- Compatible with systems having minimum 8GB RAM and integrated GPUs, ensuring scalability and accessibility.
- Faster inference time makes it ideal for real-time QnA applications.

F. *Summary of Findings*

- Both models demonstrate strong performance for QnA tasks in the nutrition domain.
- Accuracy improved with model size, but resource trade-offs must be considered.
- Even with a relatively small dataset (30 QnA pairs per category), the T5 architecture generalizes well, indicating robustness for domain-specific fine-tuning.

VI. CONCLUSION AND INSIGHTS

A. *Conclusion*

In this research, we successfully developed an AI-Powered Weight-Based Meal Recommendation and QnA Chatbot using pre-trained T5 language models—specifically T5-base and T5-large. By leveraging the question-answering capabilities of transformer-based models, our system accurately responds to user queries based on their gender and BMI classification (Underweight, Normal Weight, Overweight, and Obesity for both Male and Female categories).

The system was trained on a custom dataset extracted from MongoDB containing approximately 30 QA pairs per category, making it highly focused and domain-specific. The results highlight that:

- T5-base provides a fast and lightweight solution with 90% accuracy, making it ideal for deployment on standard hardware (e.g., 8GB RAM and integrated GPU).
- T5-large, although more resource-intensive, achieved higher performance with 93% accuracy, thanks to its 770 million parameters, offering better contextual understanding and less semantic ambiguity.

B. Key Insights

- Model Efficiency vs. Accuracy Trade-off: T5-base is more suitable for real-time applications on low-resource systems, while T5-large excels in accuracy when higher computational power is available.
- Semantic Robustness: Even when some word repetitions were observed in T5-base outputs, the semantic correctness of answers was retained, which is critical in real-world chatbot usage.
- User-Centric Design: By categorizing users into weight-based groups and tailoring answers accordingly, the chatbot offers personalized meal suggestions, increasing its usefulness in dietary applications.

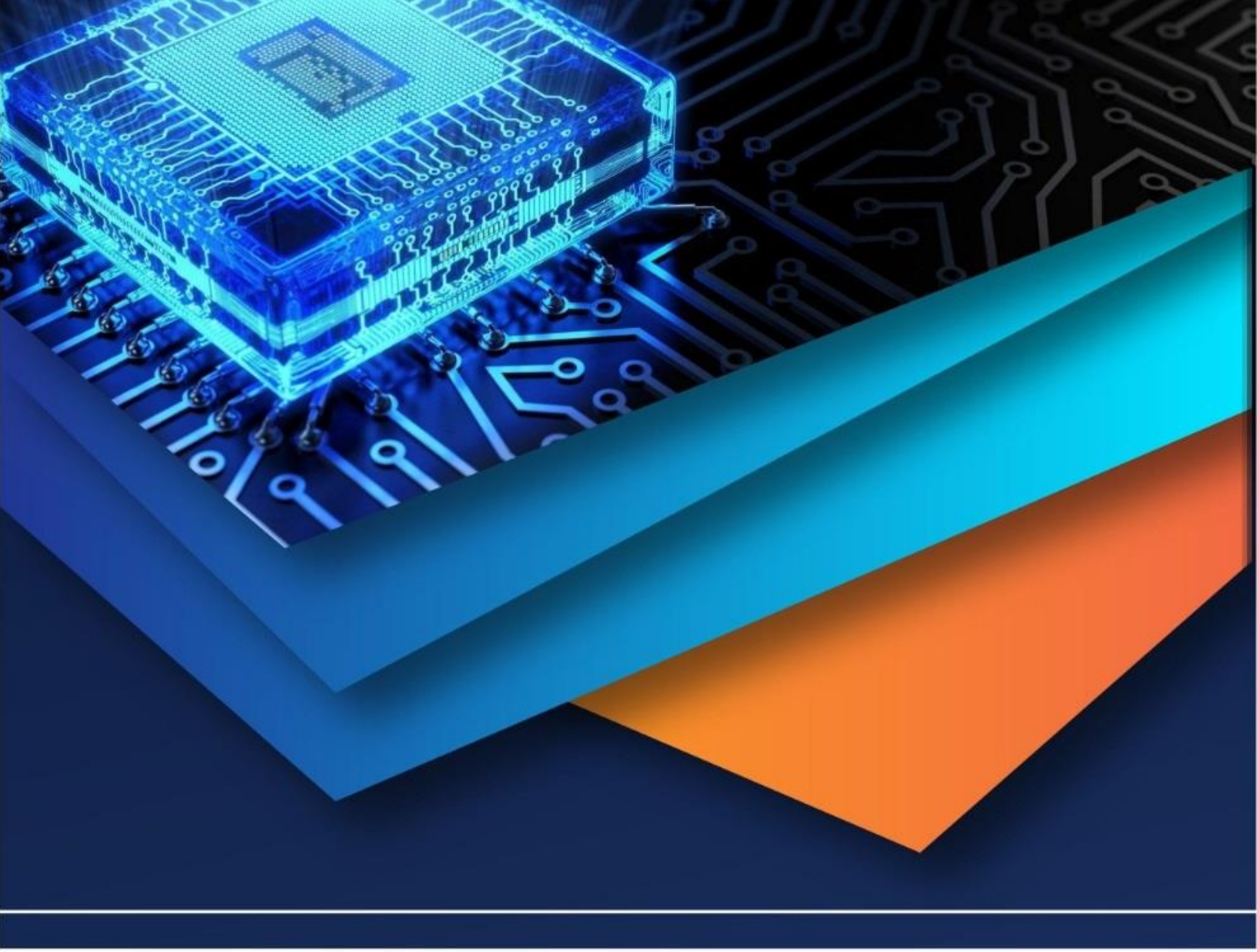
VII. FUTURE SCOPE

In the future, this system can be enhanced by:

1) Integrating voice-based interaction for a hands-free chatbot experience.
2) Expanding the dataset with more diverse and real-world QA pairs to improve robustness.
3) Incorporating user feedback mechanisms to enable dynamic learning and personalization.
4) Deploying on cloud platforms for scalability and real-time access.
5) Exploring domain-specific transformer models like BioBERT to further improve response accuracy in health and nutrition contexts.

REFERENCES

[1] P. Shetty, "Dietary guidelines and recommendations in the prevention and treatment of metabolic syndrome: an evidence-based approach," Nutrition Reviews, vol. 71, no. 3, pp. 122–135, 2013.
[2] S. Dey, A. Biswas, and M. Ghosh, "A machine learning based personalized diet recommendation system," in Proc. 2nd Int. Conf. on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 738–743.
[3] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.
[4] L. Y. Tan and K. S. Ong, "Fine-tuning BERT for Question Answering on Medical Documents," in Proc. 2021 Int. Conf. on Machine Learning and Cybernetics (ICMLC), Shenzhen, China, 2021, pp. 240–245.
[5] A. Saha, S. Paul, and R. Mandal, "BERT Based Interactive Chatbot for Health and Nutrition Guidance," in Proc. 11th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1–6.
[6] H. Lin and M. Gao, "A MongoDB-based Chatbot Architecture for Real-time User Queries," in Proc. 2022 Int. Conf. on Big Data and Smart Computing (BigComp), Jeju Island, South Korea, 2022, pp. 525–529.
[7] Google's T5 (text to text Transfer Transformer), available in https://huggingface.co/docs/transformers/en/model_doc/t5.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY