Healthsphere: Intelligent Health Companion

A PROJECT REPORT

Submitted by,

Ms. K Pavadharani- 20211CSD0127

Ms. Khushi S M- 20211CSD0045

Ms. Akshatha C- 20211CSD0115

Under the guidance of,

Dr. Madhusudhan M V

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

at

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **“Healthsphere: Intelligent Health Companion”** being submitted by “**K Pavadharani, Khushi S M, Akshatha C”** bearing roll number(s) “**20211CSD0127, 20211CSD0045, 20211CSD0115**” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a Bonafede work carried out under my supervision.

Dr. Madhusudhan M V
Associate Professor
School of CSE
Presidency University

Dr. Saira Banu Atham,
HOD
School of CSE
Presidency University

Dr. L. Shakkeera
Associate Dean
School of CSE&IS
Presidency University

Dr. Mydhili K Nair
Associate Dean
School of CSE&IS
Presidency University

Dr. Md. Sameeruddin Khan
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **“Healthsphere”** in partial fulfilment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Madhusudhan M V, Associate Professor, School of Computer Science Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

ROLL NUMBERS	NAMES	SIGNATURE
20211CSD0127	**K Pavadharani**	
20211CSD0045	**Khushi S M**	
20211CSD0115	**Akshatha C**	

ABSTRACT

The "Heathsphere" is a comprehensive web-based solution designed to help individuals monitor and improve their overall health through personalized insights and recommendations. By leveraging advanced technologies such as machine learning and API integrations, the application enables users to track their daily nutrition, hydration, and physical activities effortlessly. Users can input details about their meals, water intake, and physical data, such as height and weight, to receive detailed analyses of their nutritional intake, including calories, proteins, carbohydrates, vitamins, and minerals.

The application identifies potential deficiencies in essential nutrients, predicts related health risks, and offers tailored suggestions for dietary and lifestyle improvements. Additionally, it tracks water consumption, provides hydration reminders, and monitors physical activities using Google Fit API to estimate calories burned. By integrating diverse metrics into a unified platform, the Healthsphere simplifies complex health data into actionable insights, empowering users to achieve fitness goals, maintain balanced diets, and prevent diseases caused by nutritional imbalances. This innovative tool addresses the challenges of busy schedules, lack of nutritional awareness, and data accessibility, making it a valuable asset for promoting healthier lifestyles.

This project addresses the growing need for an efficient and user-friendly platform that simplifies health tracking and promotes well-being. By analyzing data over time, the app identifies patterns in user behaviour, such as recurring nutrient deficiencies or irregular hydration habits, and provides proactive recommendations to mitigate potential health risks. It integrates machine learning to deliver precise, real-time insights, enhancing its capability to offer personalized dietary suggestions and exercise plans tailored to individual needs.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected Dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science and Engineering and School of Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science and Engineering and School of Information Science, Presidency University and **Dr. Saira Banu Atham,** Head of the Department, School of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinator **Dr. Manjula H M** and also the department Project Coordinators.

We are greatly indebted to our guide **Dr. Madhusudhan M V, Associate Professor**, School of Computer Science & Engineering, Presidency University for her inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

K Pavadharani- 20211CSD0127
Khushi SM- 20211CSD0045
Akshatha C- 20211CSD0115

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

The digital transformation in healthcare has brought forward the chatbots as effective tools for improving patient interaction, answering routine questions, and increasing access to healthcare. However, the shortcomings include limited emotional intelligence and complex diagnostic understanding, necessitating further improvements to accomplish comprehensive support for patients. In response to these, Healthsphere is the AI-driven healthcare chatbot framework designed for advanced patient interactions and personalized health management. It employs technologies like machine learning and Firebase, and provides functionalities ranging from nutritional analyses to predicting possible deficiencies, which then offers tailored recommendations withto real-time updates using an embedded broadcast module.

1.1 Problem Statement

Many people simply cannot track or monitor their consumption and possible shortcomings in nutrition; hence, many turn out to experience problems with overweight, dehydration, or deficiency in particular vitamins and minerals. Furthermore, users' time schedules do not allow them the luxury of making a record about their physical exercises, water use, and calorific intake. Such a solution needs to be efficient, tracking this data, yet it should provide actionable insights and personalized recommendations toward a healthier lifestyle.

Expectation: The "Healthsphere" app promises to be a comprehensive tool for recording nutrition and hydration intake along with the physical activity. On being fed daily meals, water intake, and personal health information by the user, it predicts the nutritional values for them, shows the deficiency, and recommends improvement. In all, it is the application that helps its users to meet their fitness target and keep themselves hydrated because of constant reminders for intake and activity. Furthermore, it leverages machine learning to predict potential health risks, ensuring users receive early warnings and recommendations for preventive measures.

1.2 Objective of the Project

The primary objective of this project is to develop a comprehensive web-based health analyser that empowers users to:

- Assess their daily nutrition and detect potential deficiencies in vitamins, proteins, and other key nutrients.
- Calculate and track their BMI to understand their physical health category.
- Monitor water intake and provide hydration reminders.
- Suggest personalized exercise routines and lifestyle changes to improve overall health.
- Deliver actionable insights to prevent diseases caused by nutritional deficiencies.

1.3 Project Introduction:

The Healthsphere integrates advanced technologies, including machine learning and Firebase, to create a seamless experience for users. It allows individuals to input their daily food consumption and physical data, such as height, weight, and water intake. Using these inputs, the app generates precise nutritional analyses, predicts possible deficiencies, and offers tailored recommendations. The application also tracks water intake and physical activities, encouraging users to stay hydrated and active. Through machine learning and Google Fit API integration, the platform enhances user experience by providing real-time insights and actionable recommendations.

Key Features:

- **Nutritional Analysis:** Predict the content of calories, proteins, carbohydrates, vitamins, and minerals based on food inputs.
- **Deficiency Detection:** Identify potential vitamin or mineral deficiencies that may lead to health risks.
- **Hydration Monitoring:** Track daily water intake and offer personalized hydration advice.
- **Activity Tracking:** Monitor physical activity and estimate calories burned using Google Fit API.
- **Health Recommendations:** Suggest dietary improvements, exercises, and lifestyle changes based on user data.

1.4 Architecture

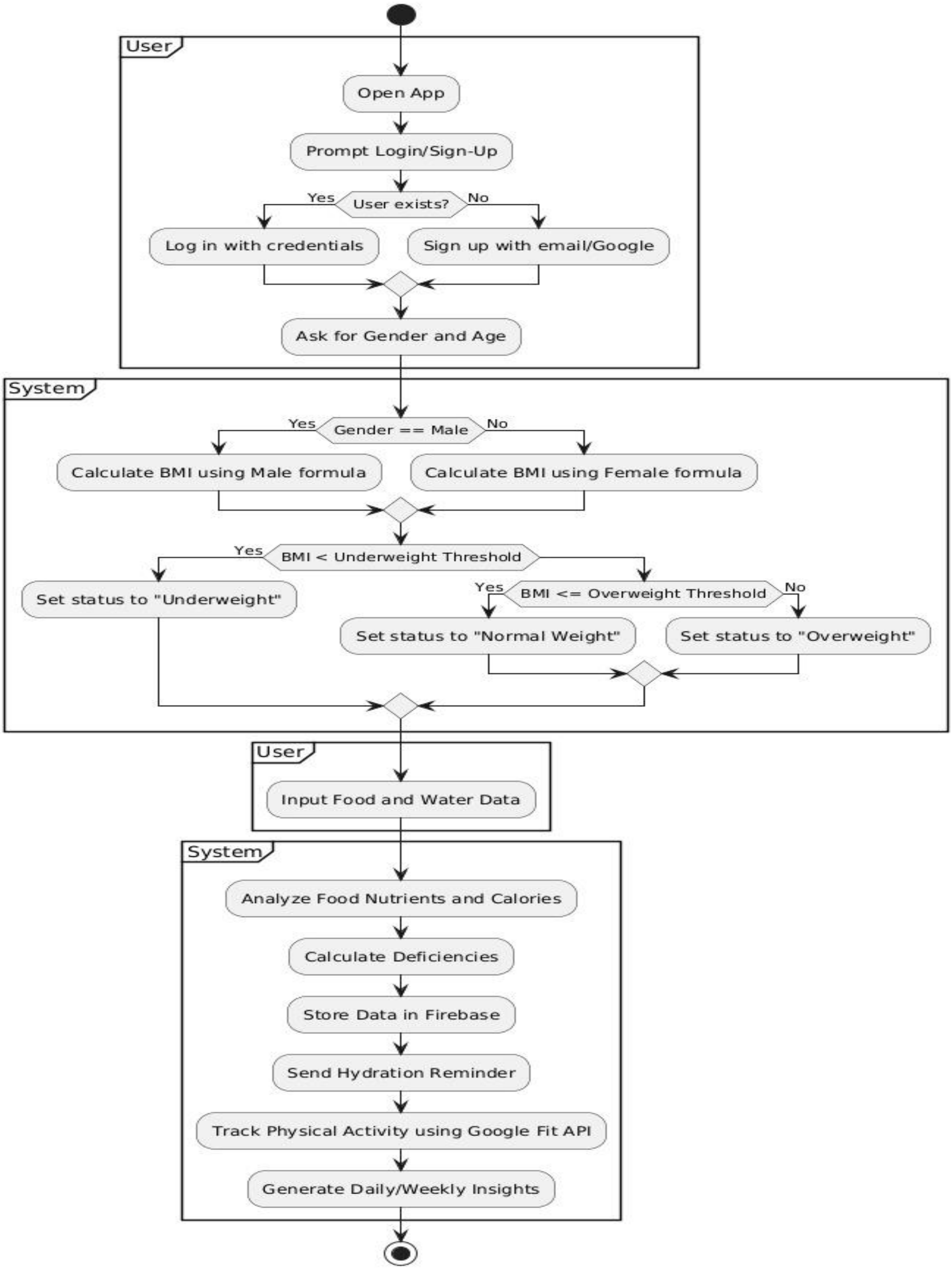


Fig 1. Architecture of Healthsphere

1.5 Challenges Addressed

- **Data Accessibility:** Simplifies complex nutritional information into user-friendly insights, making health tracking accessible to all.
- **Deficiency Awareness:** Educates users about potential health risks stemming from dietary imbalances and water intake.
- **Personalized Insights:** Provides tailored recommendations, ensuring the advice is relevant to individual health profiles.
- **Integration of Diverse Metrics:** Combines nutritional analysis, hydration tracking, and physical activity monitoring into a single platform.
- **Real-Time Analysis:** Utilizes machine learning models to process user data efficiently and deliver accurate results promptly.

CHAPTER-2

LITERATURE SURVEY

AI and machine learning in mobile health apps have revolutionized personalized wellness management. Studies show ML's effectiveness in analyzing diet and predicting nutrient deficiencies. Additionally, fitness tracking APIs, like Google Fit, enable comprehensive health monitoring across nutrition, hydration, and activity:

Mistry et al. [1] developed a Smart Health application for patient record management, prescription, and appointment scheduling using Flutter and Firebase. The system allows doctors to access patients' medical histories and manage appointments and prescriptions via an intuitive interface. Patients can register, log in, book appointments, and view their medical records, including lab results. The application employs Cloud Firestore for a scalable database and email-password authentication for secure access. Despite its user-friendly design and convenience, its dependence on internet connectivity and limitations to specific functionalities, such as video consultancy and graphical file support, present areas for improvement. Future enhancements may include expanded features for privacy and multimedia support.

Negi et al. [2] conducted a comprehensive analysis of various health and nutrition management apps to evaluate their features, user benefits, and limitations. The study categorized apps into health monitoring, telemedicine, and health management, identifying specific functionalities like tracking sleep, calories, and stress, as well as facilitating doctor-patient communication. Apps like Cure.fit, Fitelo, and HealthifyMe were compared for usability and unique features. However, the study noted limitations such as a lack of focus on teenage users, high costs, and dependency on device ecosystems. The analysis emphasized the rapid growth of fitness apps, particularly in India, driven by post-COVID health awareness. Future recommendations include improving accessibility, personalization, and affordability to cater to diverse user groups.

Palička et al. [3] explored the role of mobile applications in supporting physical activities and their integration into physical education at schools. The study surveyed students and teachers in the Czech Republic to assess the adoption and impact of such technologies. These apps, leveraging features like GPS, accelerometers, and gyroscopes, allow for tracking activities such as running and cycling, providing real-time feedback. While the potential benefits include enhanced engagement and personalized activity monitoring, the study noted challenges like inconsistent data reliability, risks of data misuse, and limited academic research on their educational impact. Recommendations included the development of a vetted database of apps tailored to educational contexts to mitigate risks and maximize utility.

Adam Palanica et al. [4] conducted a study titled "Physicians’ Perceptions of Chatbots in Health Care: Cross-Sectional Web-Based Survey," published in the Journal of Medical Internet Research. The research explored physicians' views on healthcare chatbots through a survey of 100 general practitioners in the U.S. The study employed descriptive statistics to analyze responses on benefits, challenges, and risks. Physicians identified logistical uses, such as scheduling appointments and medication reminders, as key advantages, while challenges included chatbots’ inability to display human emotions and provide nuanced medical assessments. The findings suggest chatbots may assist in administrative tasks but are currently inadequate for complex decision-making roles. However, limitations include a focus on general practitioners in the U.S., leaving gaps in perspectives from specialists or other regions.

Gohatre et al. [5] developed FITNOMIC, a fitness app designed to support users in building fitness through personalized workout plans, diet tracking, and motivational features. The app integrates data-driven recommendations and user-friendly tools for managing health and fitness goals. It was evaluated through user feedback surveys conducted during March-April 2023, focusing on usability and effectiveness. FITNOMIC offers accessible fitness solutions and aims to engage users through interactive features and progress tracking. However, its reliance on consistent user input and potential compatibility issues across different devices may limit its usability. Further enhancements could improve its scope for diverse fitness needs and long-term user engagement.

Sobti and Grover [6] conducted a bibliometric analysis of health and fitness apps, emphasizing their increased usage during and after the COVID-19 pandemic. Using R Studio for bibliometric evaluation of 645 documents from the Scopus database, the study highlighted key trends, influential authors, and most cited works. Results showed a significant rise in fitness app downloads during the pandemic, linked to social distancing and gym closures, with a 46% increase globally in Q1 and Q2 of 2020. The study emphasized the role of fitness apps in promoting physical activity, nutrition tracking, and mental well-being, especially during periods of restricted movement. However, it identified challenges such as high attrition rates and the need for improved customer engagement through personalized features. Despite limitations in scope and reliance on secondary data, the study provides valuable insights for researchers and developers aiming to enhance fitness app utility and retention.

Comendador et al. [7] developed Pharmabot, a pediatric generic medicine consultant chatbot, using Visual C# and MS Access as a standalone application. The system employs a Left-Right Parsing Algorithm to process user queries and was evaluated through surveys of pharmacy students and pediatricians, utilizing statistical analysis (T-test) for comparison. Pharmabot provides easy access to pediatric medicine information and features a user-friendly interface with clear instructions, validated by professionals for reliability. However, its focus on pediatric medicines limits its scope, and being a standalone application restricts access. Additionally, specific input formats may frustrate users if not followed correctly.

Aishwarya Mondal et al. [8] employed a machine learning-based approach to develop a heart disease prediction model utilizing Support Vector Machine (SVM) and Artificial Neural Network (ANN) algorithms. The methodology involves data collection from Kaggle, preprocessing, feature selection via correlation heat maps, and model development. The paper includes achieving high accuracy (86.6% with ANN and 81.6% with SVM), early disease detection, personalized treatment, and efficient feature selection. However, there is dependence on data quality, limited generalizability, complexity, potential overfitting, and difficulty in interpreting ANN decisions.

Seema et al. [9] developed a chatbot named Doctor Chabot to assist users in predicting heart disease by analyzing symptoms with the help of data mining techniques. They used Dialog flow for user interaction and integrated the Support Vector Machine (SVM) algorithm to classify input data and predict heart conditions. This approach enables the chatbot to provide users with a preliminary diagnosis, potentially allowing for early intervention. However, the system's effectiveness depends on the quality of input data, and any inaccuracies in user input could result in misdiagnosis. Nonetheless, Doctor Chat bot offers an accessible, real-time health assessment tool, reducing the need for in-person consultation for initial health concerns.

Huang et al. [10] developed a "Smart Wireless Interactive Healthcare System" (SWITCHes) that utilizes an AI-powered chatbot within a mobile app for health monitoring, specifically targeting weight control and health promotion. By leveraging real-time data collection on dietary intake and physical activity, the chatbot provides personalized feedback, enabling users to make informed lifestyle changes. This approach reduces reliance on self-reporting, which can be inaccurate, and instead offers an objective, data-driven system. However, a limitation lies in its dependency on Android devices, restricting accessibility for iOS users. Despite this, the SWITCHes system supports efficient self-monitoring and delivers tailored health recommendations, enhancing the user experience and engagement.

Honary et al. [11] conducted a study to explore the perceptions and potential harms associated with healthy eating and fitness apps among young people, employing a mixed-methods approach that included surveys, workshops, and expert interviews. This methodology allowed for a comprehensive understanding of user experiences and expert insights, highlighting both the positive and negative impacts of app usage. One advantage of this approach is its ability to capture diverse perspectives, providing a richer context for the findings. However, a disadvantage is the potential for self-selection bias in survey participants and workshop attendees, which may limit the generalizability of the results. Despite these limitations, the study effectively identified key concerns regarding app design and its implications for young users' mental health and well-being.

Wang et al. [12] conducted a study utilizing a mixed-methods approach to evaluate the effectiveness of diet and physical activity apps among users, employing semi-structured focus group discussions and a comprehensive questionnaire to gather qualitative and quantitative data. The methodology allowed for an in-depth understanding of user experiences, motivations, and perceived effectiveness of the apps, which highlighted the positive impact on dietary and physical activity behaviors. However, the reliance on self-reported data may introduce biases, as participants might overestimate their app usage or behavior changes. Despite this limitation, the study provided valuable insights into how tailored app features could enhance user engagement and promote healthier lifestyles.

Khan et al. [13] conducted a review of mobile apps focused on human nutrition, analyzing their roles in tracking diet, calories, and activity to support users in maintaining balanced nutrition and promoting a healthy lifestyle. By providing real-time access to nutritional data and customizable options for tracking food intake and fitness goals, these apps offer a convenient solution for users who may lack access to dietitians or time for in-person consultations. However, the dependency on self-reported data can introduce inaccuracies, potentially affecting the reliability of feedback. Nevertheless, these apps empower users to monitor their health effectively and encourage healthier dietary habits.

Samoggia et al. [14] assessed the impact of the Edo nutrition-information app on consumer behavior by using a two-stage online questionnaire based on the Health Belief Model and Trans-Theoretical Model. Users completed baseline and follow-up surveys over a 12-week period, providing data on health orientation, purchasing habits, and perceived app effectiveness. The app successfully raised awareness of healthy eating and offered personalized food guidance, fostering better dietary decisions. However, the study’s reliance on self-reported data could limit accuracy, and some users reported minor barriers to consistent app usage. Despite these challenges, the Edo app showed promise as an effective tool for enhancing health-conscious behavior in users.

Khaled H. Almotairi [15] presents a comprehensive review of Internet of Things (IoT) technology in healthcare, highlighting its benefits, challenges, and future research trends. The methodology employed involves a thorough analysis of existing literature on IoT-enabled healthcare systems, identifying gaps and areas for improvement. Improved treatment outcomes, enhanced patient experience, and increased efficiency in healthcare delivery are the pros. However, challenges such as security concerns, data management issues, and infrastructure limitations hinder its adoption. Future research trends include integrating machine learning algorithms for predictive analytics, prosthetics sensors, and blockchain technology for secure data storage.

Zečević et al. [16] conducted a study analyzing user reviews of diet-tracking apps through content analysis and text mining techniques, including topical n-grams identification and topic modeling, to uncover user sentiments and preferences. This methodology allowed for the processing of a large dataset of over 72,000 reviews, providing insights into user experiences and app features that enhance usability. However, the study's reliance on user-generated content may introduce biases, as it primarily reflects the opinions of those who choose to leave reviews, potentially overlooking the perspectives of a broader user base. Despite this limitation, the approach effectively highlighted both positive and negative aspects of the apps, offering valuable recommendations for developers to improve user satisfaction and app functionality.

Jörg-Uwe Meyer [17] presents a service-oriented architecture (SOA) health Web platform for mobile medical apps, enabling secure connectivity to electronic health records and medical devices. Methodology involves designing a modular, open platform using established tools (Microsoft .NET Framework, (link unavailable) Web APIs) and standards (WS-, IEC 80001, ISO/AWI TR 80002-2). There is improved interoperability, substitutable medical apps, enhanced patient data security through transient storage, and compliance with regulatory requirements. However, complexity in integrating disparate devices and systems, potential security risks in public cloud usage, dependence on standardized interfaces and semantics, and the need for continuous verification and validation of platform and interface performance has become the limitations.

Vaswani et al. [18] developed a comprehensive healthcare application aimed at easing medical assistance for users like doctors, patients, and medical equipment dealers. The app integrates a symptom analyzer, medicine ordering, doctor consultations via chat, and an SOS button for emergencies. Advantages include ease of access to medical services, live patient monitoring, and secure e-wallet transactions. However, the app’s dependency on internet connectivity and the complexity of enforcing strict data integrity in Firebase pose limitations.

Mondal et al. [19] proposed a chatbot system focused on heart disease prediction, utilizing Support Vector Machine (SVM) and Google’s Dialogflow for natural language processing. This system offers patients the convenience of early diagnosis and appointment scheduling, making healthcare more accessible, especially in areas with limited medical resources. While this chatbot improves self-service in healthcare, its reliance on a pre-defined dataset limits the scope of diagnosis and may affect accuracy for untrained symptoms.

Kazi et al. [20] implemented MedChatBot, a chatbot designed for medical students that utilizes the Unified Medical Language System (UMLS) and Artificial Intelligence Markup Language (AIML) to respond to queries in natural language. This system interprets queries and generates SQL commands to retrieve relevant information from UMLS, aiding medical students in self-directed learning. Although MedChatBot performs well for simple, factual queries, its reliance on UMLS limitations reduces its effectiveness for complex queries, such as symptoms and causal relationships. Despite this, the chatbot improves learning by providing quick answers to common medical questions, enhancing the educational experience for students.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Despite the advancements in healthcare chatbots, several research gaps persist, limiting their full potential. Current chatbots often lack emotional intelligence, making interactions feel impersonal and less supportive. They struggle with complex diagnostic scenarios, leaving patients without the nuanced assistance they might need. Additionally, there's a noticeable gap in seamlessly integrating nutrition and fitness tracking for holistic health management. Many chatbots fail to provide personalized recommendations, leading to generic experiences that lack real impact. Data security and ethical management of sensitive health information remain critical challenges. Sustaining user engagement and promoting long-term behavior changes are areas needing improvement, as is their ability to adapt to real-time contexts dynamically. Finally, scaling these systems effectively for larger user bases while maintaining reliability and performance remains a significant technical hurdle. Addressing these gaps is crucial to creating smarter, more reliable healthcare chatbots. The existing research gaps are:

1. **Limited Emotional Intelligence in Chatbots**

   Current healthcare chatbots, including implementations similar to Healthsphere, lack robust emotional intelligence. While they are proficient in task-oriented interactions, such as symptom tracking and appointment scheduling, they fail to establish a more empathetic or human-like connection with users. This gap limits their effectiveness in handling emotionally sensitive scenarios, such as counselling or providing support for patients with mental health conditions.

2. **Complex Diagnostic Understanding**

   The diagnostic capabilities of existing systems, including AI-based frameworks like Healthsphere, are often constrained by their reliance on predefined data sets and rule-based decision trees. They may struggle with ambiguous symptoms or overlapping conditions, which require nuanced analysis beyond their programmed capabilities. This is a significant gap when compared to the broader diagnostic scope handled by human healthcare providers.

3. **Lack of Comprehensive Integration in Tracking**

   Applications such as the "Healthsphere" provide detailed tracking for nutrition, hydration, and activity, yet they often operate in isolation. There is a lack of seamless integration with larger healthcare ecosystems, such as electronic health records (EHRs) or wearable health devices. This creates fragmented user experiences and limits the utility of the data for healthcare providers.

4. **Data Personalization and Predictive Accuracy**

   While machine learning models can predict potential health risks and suggest improvements, their recommendations may lack the precision required for diverse user demographics. Factors such as cultural dietary preferences, individual metabolic variations, and localized health guidelines are not always adequately incorporated, resulting in generic or less actionable insights.

Fig 2. Impact on System Performance

5. **Challenges in Securing and Managing Sensitive Health Data**

   The use of cloud-based services, such as Firebase and SQLite, raises concerns about data privacy and security. While these frameworks offer robust features, implementing end-to-end encryption, anonymization, and compliance with healthcare regulations like HIPAA and GDPR remains a critical research area. Ensuring secure data storage and transfer without compromising performance or usability is an ongoing challenge.

6. **User Engagement and Long-Term Behaviour Change**

While features such as hydration reminders and exercise recommendations promote short-term engagement, the sustainability of such behavioural changes remains questionable. Research on integrating behavioural psychology principles into these systems is limited. There is a need for features that adapt over time to maintain user motivation and foster long-term healthy habits.

Fig 3. Role Based Access Control

7. **Real-Time Context Awareness**

Current applications lack advanced real-time context-awareness, which could make recommendations more relevant. For instance, nutrition suggestions based on time of day, activity levels, or current location could significantly enhance user experience. Bridging this gap requires further exploration into combining AI with real-time sensors and contextual data processing.

8. **Scalability and Performance Optimization**

As applications like Healthsphere and the "Healthsphere" scale to serve larger user bases, maintaining high performance and real-time responsiveness becomes a challenge. Research on optimizing machine learning models for faster inference without sacrificing accuracy is still evolving and is critical for wider adoption.

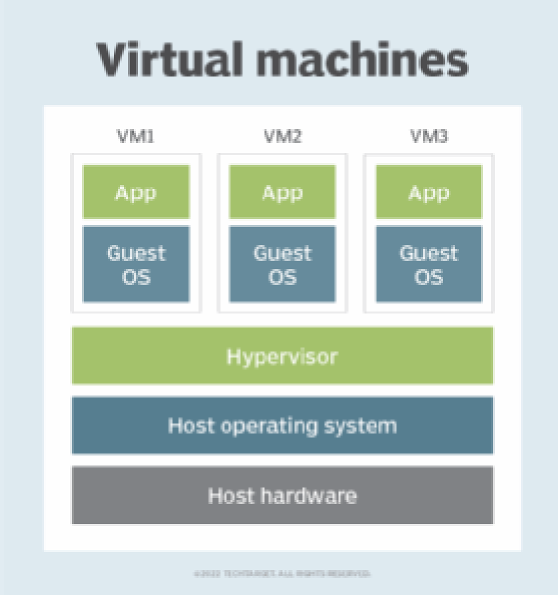


Fig 4. Platform Instance

(In fig 4., a virtual machine (VM) architecture within a virtualization environment involves a collection of several elements harmoniously working to accomplish a singular objective. One can have numerous virtual machines like VM1, VM2, and VM3, running individually on the same physical hardware that supports different applications and a guest operating system differing from the host OS. The hypervisor forms an intermediary layer between the host OS and VMs. Hypervisors, for example, create and manage VMs; allocate physical resources in the form of CPU, memory, and storage to every virtual machine; and ensure isolation without interference. Therefore, the base environment of this hypervisor can be the operating system running on a physical machine. The host hardware is composed of the physical resources, like CPU, memory, and storage, which are shared among VMs through a hypervisor. This architecture ensures isolation between VMs, enables resource sharing, and allows flexibility by supporting different operating systems and applications on a single machine. It is widely used in cloud computing, server consolidation, and development testing environments.)

By addressing these gaps, future research and development in AI-driven healthcare and nutrition applications can significantly improve user experience, broaden diagnostic capabilities, and strengthen integration into existing healthcare systems.

CHAPTER-4
OBJECTIVES

The primary objective of the proposed system, **"Healthsphere"**, is to provide an intelligent and user-friendly platform for monitoring and improving individual health through the analysis of dietary, hydration, and physical activity data. The specific objectives are as follows:

1. **Personalized Health Monitoring**:

    - To enable users to track their daily nutritional intake, hydration levels, and physical activity in a personalized manner based on their unique attributes such as age, weight, height, and activity level.

2. **Nutritional Analysis and Recommendation**:

    - To analyse the nutritional value of meals and identify deficiencies or excesses in caloric and nutrient intake using a robust food database and machine learning algorithms.
    - To provide tailored dietary recommendations for maintaining or improving overall health.

3. **Hydration Management**:

    - To monitor water consumption and ensure users meet their daily hydration requirements based on environmental conditions and physical activity levels.
    - To offer reminders and suggestions to promote optimal hydration habits.

4. **Physical Activity Tracking**:

    - To record and evaluate the user's physical activity, including the type and duration of exercises.
    - To estimate calories burned and incorporate this data into daily caloric balance assessments.

5. **Holistic Health Insights**:

    - To integrate data from the three modules—dietary analysis, hydration monitoring, and physical activity tracking to provide comprehensive overview of health status.
    - To enable users to understand how different factors, such as diet and exercise, interact and influence their overall well-being.

6. **User-Friendly Interface**:

    - To design an intuitive and engaging interface that simplifies data entry and provides clear visualizations of health metrics.
    - To ensure accessibility and ease of use for users of varying technological proficiency.

7. **Promoting Healthy Habits**:

    - To motivate users by offering progress tracking, goal setting, and feedback on their health improvements over time.

8. **Evaluation and Continuous Improvement**:

    - To regularly evaluate the system's performance and accuracy by comparing its outputs with established health guidelines.
    - To incorporate user feedback to enhance functionality and user satisfaction.

The key objectives are:

- Help users track their nutrition, hydration, and activity levels in a way that fits their unique needs, like age, weight, and lifestyle.
- Analyze meals to find nutrient gaps or excesses and provide personalized dietary advice for better health.
- Support daily hydration by monitoring water intake, offering reminders, and adjusting based on activity and environment.
- Track physical activities, estimate calories burned, and factor this into overall health assessments.

- Combine data from nutrition, hydration, and activity to give users a complete view of their well-being.
- Make logging health details simple and enjoyable with a user-friendly interface that's easy for anyone to use.
- Encourage users to set goals, track progress, and celebrate their health achievements over time.
- Keep improving by learning from user feedback and aligning with trusted health standards.

CHAPTER-5

PROPOSED METHODOLOGY

The methodology is about designing and developing an integrated application, "Healthsphere", designed to track and analyze nutritional intake, hydration levels, and physical activity. The subsequent section explains the systematic approach toward building and implementing the system.

5.1 The system has the following modules:

User authentication Module: This process initiates when the app is launched, and a prompt appears for the user to either log in or sign up. If the system finds that the user exists, they are prompted to log in with their credentials. In case the user is new, they are taken through the sign-up process, which can be through email or Google authentication. When the login or signup is successful, access is granted to the app. Finally, the user session is stored in Firebase to ensure that the session information is handled securely for further interactions. The process ends here, and the user can access app functionalities.

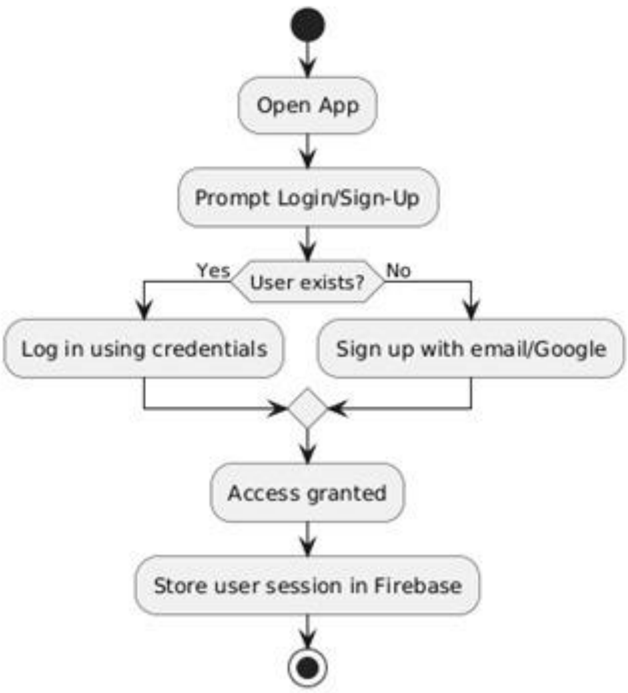


Fig 5. User Authentication Process

- Open App User opens the App.
- Prompt Login / Sign Up, the app makes a prompt Login/Sign-up.
- Check if the User Exists: There are two if conditions: One is for whether the user has already existed he will login. If the User does not have an account will sign up.
- Access Granted: Once the login/sign-up process is successful, access to the application is granted.
- Store User Session in Firebase: The user's session is securely stored in Firebase for future reference and session management.

Dietary Analysis Module: This module is just like having a nutritionist at your disposal. Users can just log in about their meals by providing specific information on ingredients or dish names. The system then analyzes such data using a comprehensive database of nutritional data broken down for fundamental elements like calories, proteins, vitamins, and minerals.

But it doesn't stop at just listing numbers. The module goes further by identifying potential nutrient deficiencies or excesses based on the user’s profile and dietary habits. If you’re missing out on key nutrients like iron or vitamin D, or if you’re consuming too much sugar, the system flags it and suggests actionable changes. For example, it might recommend adding spinach to your diet to boost iron levels or reducing sugary beverages for better balance. This module equips users with tailored nutrition knowledge to make knowledgeable decisions that will align to meet their health objectives.

Hydration Monitoring Module: Calculates and tracks the amount of water one drinks and reminds them to do so regularly. This module computes hydration needs considering user-specific attributes such as weight, age, as well as activity levels. This is often overlooked but is essential for overall well-being. This module makes it easy by tracking water intake throughout the day. Users can log their water consumption, and the system calculates hydration needs based on factors like weight, age, and activity level. For example, if you’re an active individual, your water requirements will be higher, and the system adjusts its recommendations accordingly. It also factors in environmental conditions, such as hot weather, which might increase your hydration needs. What makes this module even more user-friendly is its reminder feature. If you’re falling behind on your water intake, it sends timely alerts to keep you on track. Whether it’s a gentle nudge in the afternoon or an evening reminder to top up your hydration, this module ensures you never forget to take care of your body’s water needs. Physical activity analysis module: It is like carrying a fitness coach in your pocket.

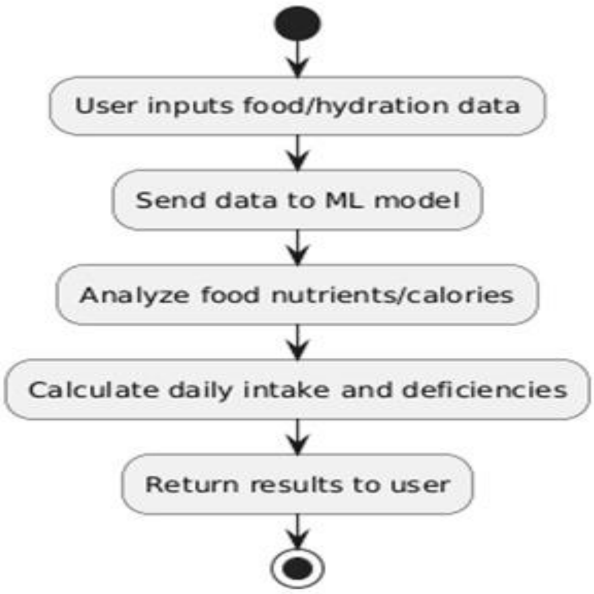


Fig 6. Food/Hydration Data Analysis

The above image (Fig 6.) explains Food/Hydration Data Analysis.

- **User Inputs Food/Hydration Data:** The user inputs information about their food and water intake into the application.
- **Data Input to ML Model:** The machine learning model then receives the data input.
- **Nutritional Value/Calories:** The ML model then analyzes it to find its nutritional values or calorie count.
- **Calculate Total Daily Intake and Deficiency:** The system calculates the user's total intake and deficiency on the nutritional sides.
- **Return to User:** This is then given back to the user for review and action.

The system tracks the physical activities one performs, such as an easy morning walk, an intense session at the gym, or even an activity a user logs in manually. It offers a predefined list of activities along with the calorie burn for each of them. But to make it more flexible and accurate, though, the users can input specific data tailored for certain workouts. The module calculates calories burned in these activities using established formulas. For example, it takes into account the intensity and duration of the workout and the user's weight to give users accurate feedback. In the long run, this enables users to know how their physical activity impacts their health goals. This module also does not only track data but also provides motivational insights and guidance. For example, if your goal is weight loss, it might suggest increasing the duration of your cardio sessions or incorporating strength training to optimize calorie burn. It logs physical activity and calculates calories burned using predefined formulas. It allows users to select activities from a list or input custom data for better accuracy. These modules are interconnected, ensuring that insights from one component influence recommendations from the others.

Input Collection and Preprocessing:

At the time of joining the site, users are requested to provide some basic yet essential details about themselves.

- **Age:** This helps the system factor in age-related nutritional and fitness needs. For example, a teenager's dietary requirements differ significantly from those of an older adult.
- **Weight and Height:** These are inputs that will allow the system to compute Body Mass Index, which is a very important metric indicating whether someone is underweight, within the healthy range, or overweight.
- **Activity Level:** This will determine caloric and hydration needs. A very active person will need much more energy and water than a sedentary person.

From this data collection, the system produces a baseline profile for each user for use as a recommendation mechanism.

The next step is to understand daily habits, and each day, users can log:

- **Food Intake:** A bowl of cereal at breakfast, a salad at lunch, or perhaps something before bedtime at night, the user records everything he or she consumes. It could be done manually, or one could pick the meals from the menu list for ease.
- **Water Intake:** Hydration is an essential factor, and logging water consumption is very easy. It matters not whether it was a glass of water or a bottle of juice; every entry counts.

- **Physical Activity:** From jogging and yoga to a session at the gym, users can log their workouts. The system even allows custom entries for activities not listed, ensuring no part of their routine is missed.

This information gives the system a clear picture of users' daily behaviours, making the insights and recommendations even more accurate and relevant.

Upon entry, it does not store data as is; it processes it for cleaning and standardization to ensure accuracy and compatibility with its algorithms. Here's how,

1. Data Cleaning

If there are incomplete or inconsistent entries, the system flags them and prompts the user to make corrections. This way every data point is usable and meaningful.

2. Standardization

When users input meals or ingredients, the system maps them to standardized nutritional values from a trusted database. For example, if a user logs “apple,” the system will automatically recognize it as a medium-sized apple and assign the appropriate calorie, fiber, and vitamin content. This prevents discrepancies, such as overestimating or underestimating nutritional values due to variations in terminology or portion size.

3. Integration for Precision

Food logs are analysed based on water consumption and exercises tracked. For instance, this system may suggest the need to drink water if there's an entry of a salty meal. Additionally, if there's an entry of a high-calorie meal after engaging in a heavy workout session, the system may notify them that they have recovered adequately.

All data is pre-processed to remove inconsistencies and ensure compatibility with the underlying algorithms. For instance, food entries are mapped to standardized nutritional values from the database to avoid discrepancies.

Machine Learning Model: At the core of Healthsphere is a powerful machine learning engine designed for highly personalized health recommendations. Healthsphere uses a Naive Bayes classifier for analyzing the data provided by the users. It is a very lightweight model, extremely efficient, and very powerful for classification-based prediction tasks in case of multi-input data. Here's how this technology works behind the scenes to simplify health management and enhance user experience:

The Naive Bayes classifier operates by analyzing patterns in user data to make educated predictions. It’s called “Naive” because it assumes that all inputs (like meals, activity, and water intake) are independent of one another—an assumption that makes it both fast and efficient while still delivering reliable results.

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

- $P(C|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(C)$ is the prior probability of *class*.
- $P(x|C)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Since P(X) is constant for all classes, we can omit it for classification and the equation will be simplified as:

$P(C|x) \propto P(x|C) * P(C)$

The ability to process vast amounts of complex data in little time makes it great for health-related insights and prediction based on the input from the user.

A primary task for the model is to calculate the caloric requirement depending on the user's activity levels. It analyzes the following:

- Logged physical activities such as walking, running, gym workout
- The intensity and duration of those activities
- User's weight, height, and age

The model estimates the calories burned daily by the user and adjusts the calorie recommendation accordingly. For example, on a sedentary day when there is not much activity, the system may suggest fewer calories so that the user does not consume too many. Conversely, when there is high exercise, it recommends more calories so that energy levels are met. The system will adjust itself to suit each user's lifestyle and hence deliver accurate recommendations that are precise and relevant for each user.

5.2 Identifying Nutrient Deficiencies

It looks into the nutritional content of meals logged. By referring to data in a solid nutritional database, it scans through the food composition logged every day for any deficiencies:

- **Deficiencies:** The model finds the area of deficiency if the user is not consuming enough of one of the nutrients, for example, iron, calcium, or vitamin D, and then can provide the user with specific recommendations on how to fill that gap. For example, it could recommend spinach or fortified cereals to increase iron intake.
- **Excesses:** Conversely, if it detects there is excessive uptake of unhealthy nutrients such as sodium or saturated fats, it alerts the consumer and provides other healthier options.

5.3 Predicting Hydration Needs

Hydration is fundamental, and through the Naive Bayes classifier, users get to stay hydrated by predicting how much water one needs in a day based on:

1. **Activity Intensity:** The more intense the physical activity, the more water is needed, and the model adjusts its recommendations.
2. **Environmental Influences:** The system also takes into account environmental factors such as temperature and humidity. On a hotter day, for instance, the model may recommend drinking more water so that one does not end up dehydrated.
3. **Nutritional Inputs:** The model will also take into account diet, such as excessive sodium and caffeine, to determine the needs for water. Knowing these subtle differences leads the system to provide personalized hydration advice based on the user's specific situation for optimal fluid balance.

   Track All User Entries and Reminders: The application tracks all the user's input data, including entries and reminders.

   Log Data in Firebase: The recorded data is logged into Firebase for safe storage and retrieval.

Generate Insights Based on History: The application analyzes the stored data and generates insights or patterns based on the user's history, such as reminders and past actions.

Fig 7. Tracking and Logging User Data

5.4 Increased Accuracy with Varied Training Data

A machine learning model only is as good as its training data. In this case, the dataset used to train the Naive Bayes classifier includes:

- **Diverse Dietary Patterns:** The dataset contains data from a wide range of cultural cuisines and eating habits to ensure that the model recognizes the widest possible spectrum of meals and ingredients.
- **Activity Levels:** Sedentary, professional athletes, and everything in between, the training data range ensures that the system can accurately predict for anyone.
- **Hydration Needs:** Data from different climates and lifestyles will ensure that the model can account for varying hydration requirements.

This is the rich and diverse dataset through which the model will learn to adapt to each user's unique profile and preferences and make recommendations that are as inclusive and accurate as possible.

5.5 User Interface Development

Healthsphere is basically built around a seamless and user-friendly interface in which users can interact easily with their health data to work towards wellness goals. Simplicity and practicality guided the design of the interface as a personal health assistant with intuitive progress tracking, insights, and motivation tools. Here is how the interface transforms the user experience:

- A summary dashboard of the daily caloric intake, water consumption, and calories burned.
- Graphical illustration of progress with time.
- Alerts and reminders for hydration and activity goals

Beyond functionality, the interface feels warm and approachable:

- **Visual Appeal:** Bright colours and clean layout make the application fun to use.
- **Positive Reinforcement:** Achievements are celebrated with messages like, “Great job hitting your hydration goal!” or “You’re halfway to your weekly step goal—keep going!”
- **Accessibility:** The design is intuitive enough for users of all ages, ensuring everyone from teens to seniors can navigate it effortlessly.

5.6 Integration of Features

This application really stands out in how its three main modules-Dietary Analysis, Hydration Monitoring, and Physical Activity Analysis-interconnect and integrate with one another to offer the complete health image of the user. The system will not act like separate tools for the different modules but interlink them in a way that ensures the highly personal, actionable recommendations made to the user will adjust dynamically with respect to their lifestyle. This simplifies the user experience further, as it removes the management of several apps or the piecing together of insights manually.

Everything in One Dashboard: See everything from your physical activity to your dietary intake and hydration status and actionable recommendations tying everything together.

Streamlined Data Entry: When an activity or meal is logged, the system automatically updates all related modules, thus removing the burden of repetitive inputs. For instance, when a long hike is logged, calorie expenditure is instantly updated, hydration needs are adjusted, and daily calorie targets are recalculated.

This interconnected design saves time but ensures that users get recommendations based on a complete understanding of their lifestyle.

5.7 Evaluation and Testing

There is a very significant testing phase which will ensure it is effective and reliable in helping users manage health. It involves not just functionality but an intuitive, accurate, and engaging user experience. These key metrics about prediction accuracy, user satisfaction, and performance allow the development team to perfect this application according to the best criteria. At the core of the Healthsphere application is its ability to provide personalized recommendations.

To ensure that the system's advice is in line with established health standards:

- **Comparison with Dietary Guidelines:** The nutrient and hydration advisories of this system will also be compared and contrasted by using globally applicable guidelines, such as those of an organization like WHO or the Guidelines for Americans regarding diet. Example: If it suggests that a particular user should ingest 60 grams of protein for the day, it will consider whether this amounts to the applicable guideline based on that user's age, weight, and activity level.

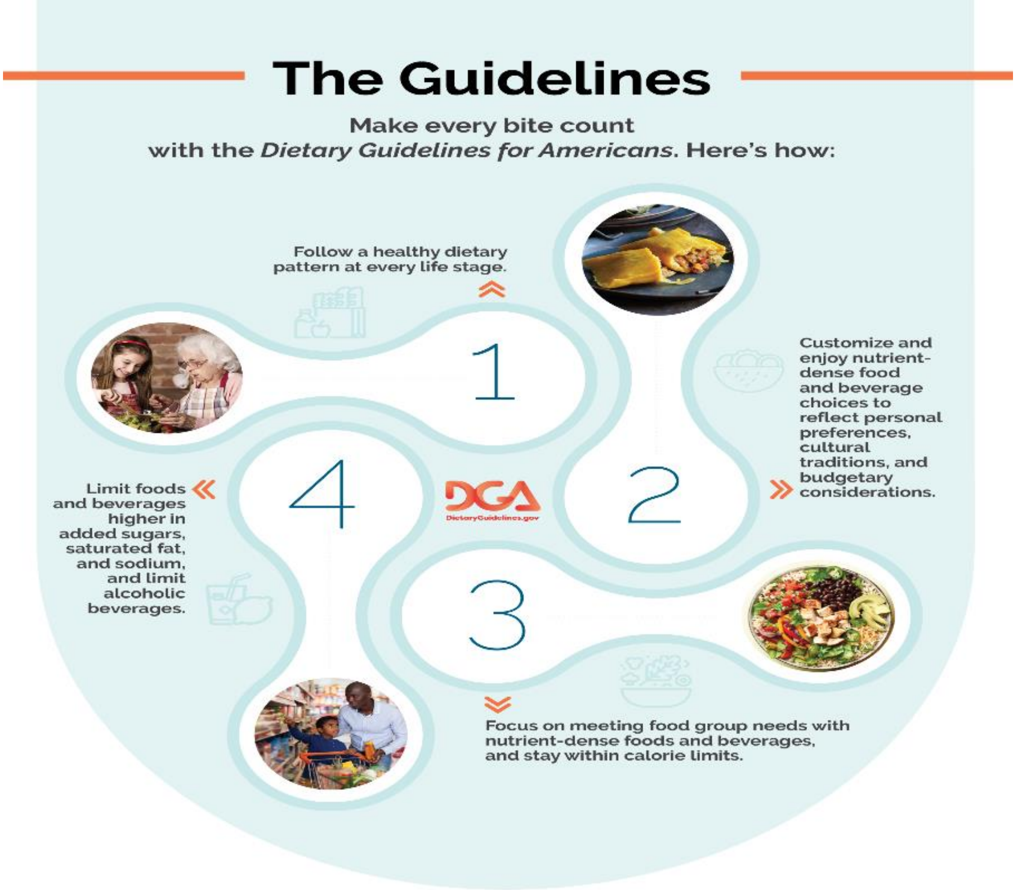


Fig 8. Dietary Guidelines for Americans

- **Iterative Model Training:** A large dataset of various dietary patterns and activity levels will be used to train the machine learning model so that prediction errors are minimized. Real-world usage testing scenarios include varied meal types, hydration needs based on climate, and fitness routines.
- **Continuous Improvement:** Feedback loops will be set up to update the system using user data and new nutritional research emerging in the long term to maintain accuracy.

This is not a one-time process; it is a constant process of testing and improvement. The application, after its release, would involve feedback from the users, usage patterns, and emerging trends in health. The constant updates fine-tune the features, make it more usable, and put the app at the forefront of the expectations of users.

Through rigorous testing and iteration, Healthsphere endeavours to build an application that people can rely on for making sound, confident health decisions. It's about an experience that's precise, engaging, and empowering - helping people easily take control of their journey toward wellness. This intense focus on prediction accuracy means that recommendations made to the users are credible because they are built upon science and uniquely tailored to a person's specific needs.

5.8 Deployment

The final application is going to be released to a diverse and extensive user base, making it accessible to people with different needs, preferences, and levels of expertise. This broad reach underscores the importance of ensuring the application's initial design is intuitive, reliable, and inclusive to cater to its diverse audience effectively.

After the launch, instead of being static, the application would come under a dynamic model of development. Updates will be on a regular basis and will depend on user feedback so that the platform evolves in alignment with the expectations and requirements of users.

DATA FLOW DIAGRAM:

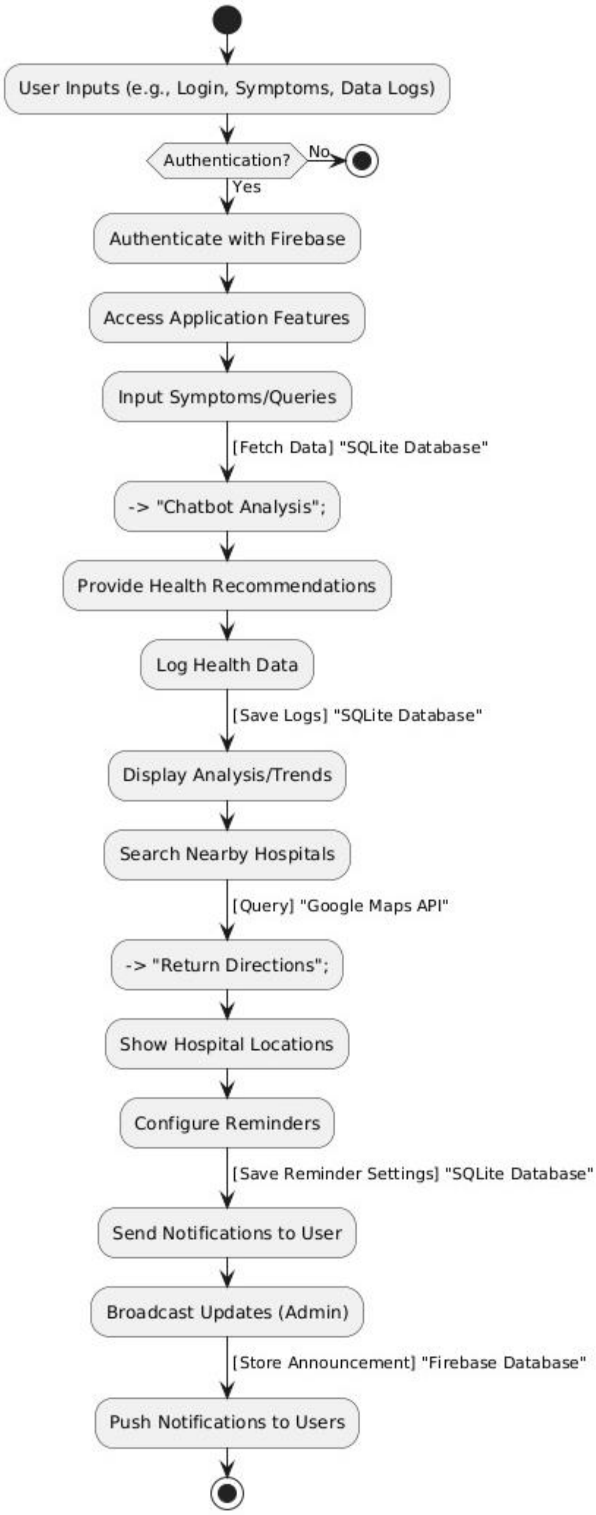


Fig 9. Dataflow Diagram of Healthsphere

1. User Inputs: User enters input such as login credentials, symptoms, or health data logs.

2. Authentication: System verifies if the user is authenticated

- Yes: Proceed to application features.
- No: System asks the user to authenticate via Firebase.

3. Input Symptoms/Queries: The user enters health-related symptoms or queries. Then the system retrieves the relevant data from a SQLite database. It then processes the input with a chatbot for analysis.

4. Provide Health Recommendations: Based on the chatbot analysis, the system provides health suggestions to the user.

5. Log Health Data: system stores the user's health information in the SQLite database for monitoring purposes.

6. Display Trends/Analysis: The application plots the logged data to display health trends or patterns.

7. Configure Reminders:

- Users create reminders for health activities.
- These reminders are saved in the SQLite database and trigger notifications at the scheduled time.

8. Broadcast Updates (Admin)

•Admins send health announcements stored in the Firebase database.

•Notifications are broadcast to users.

The system ensures: User authentication; Health recommendations based on input; Reminders, hospital searches, and personalized notifications; Admin updates to keep users informed.

ALGORITHM

Step1: Water reminder function

Step2: Track Physical Activities and Calories Burned

Step3: Naive bayes for prediction

Step4: Data Collection and Preprocessing

Step5: Train Naive Bayes Models

Step6: Train a Naive Bayes model with the given features and labels

Step7: Prediction Functions

Step8: Predict vitamin deficiencies using the trained Naive Bayes model

Step9: Map deficiencies to possible diseases

Step10: Predict if a hydration reminder is needed

Step11: Predict insights based on physical activity

Step12: Workflow Integration

Step13: Collect user data

Step14: Preprocess data

Step15: Train model

Step16: Make predictions

Step17: Display results

Fig 10. Working of Healthsphere

The algorithm represents an interactive health management process between the user and the system. The user opens the app and either logs in or signs up using credentials. After authentication, the user provides gender and age data, which allows the system to calculate BMI using gender-specific formulas. According to the BMI, the system classifies the user as underweight, normal weight, or overweight. The user then inputs data about food and water consumption, which the system analyzes to determine nutrient and caloric deficiencies. The system stores this data in Firebase, sends hydration reminders, tracks physical activity via the Google Fit API, and generates daily or weekly insights to help the user maintain or improve their health.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

The design and implementation of the "Healthsphere" system follow a modular, user-centric approach, encompassing input, processing, and output components. The **Input Module** gathers user data (e.g., age, weight, meals, hydration, and activities) through a user-friendly interface with secure authentication. The **Processing Module** uses a food database and machine learning algorithms to analyze nutrition, hydration, and physical activity, integrating these data points for a comprehensive health assessment. The **Output Module** provides personalized recommendations, visual insights, and goal-tracking features. Built using Python, tools like tkinter, pandas, matplotlib, and scikit-learn enable efficient data processing, visualization, and machine learning. Testing ensures accurate feedback and a seamless user experience, with ongoing improvements from user feedback. Below is an outline of the system design and the steps taken for its implementation.

System Design

The system is divided into three main components:

1. **Input Module**
   - **User Authentication**:
     - Includes account creation and login functionality for personalized health tracking.

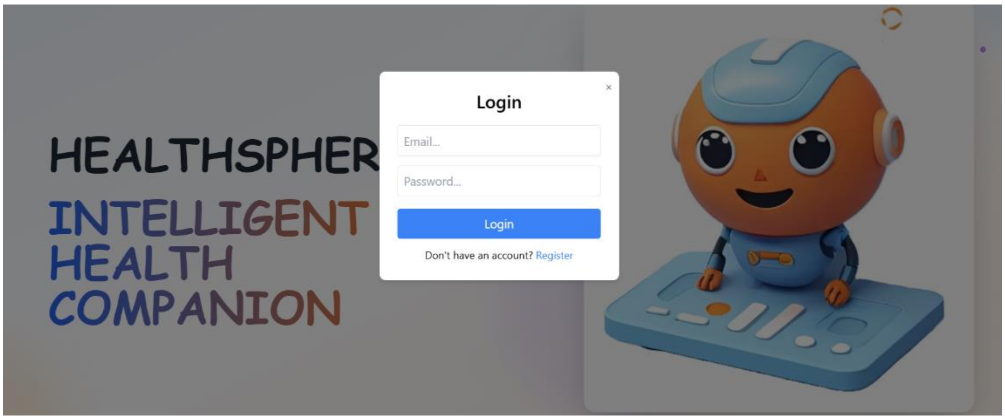


Fig 11. Secure User Authentication

➢ **Data Collection**:

- The input module collects user information such as age, weight, height, gender, and activity level.
- Users input details of their meals, water intake, and physical activities through a user-friendly interface.

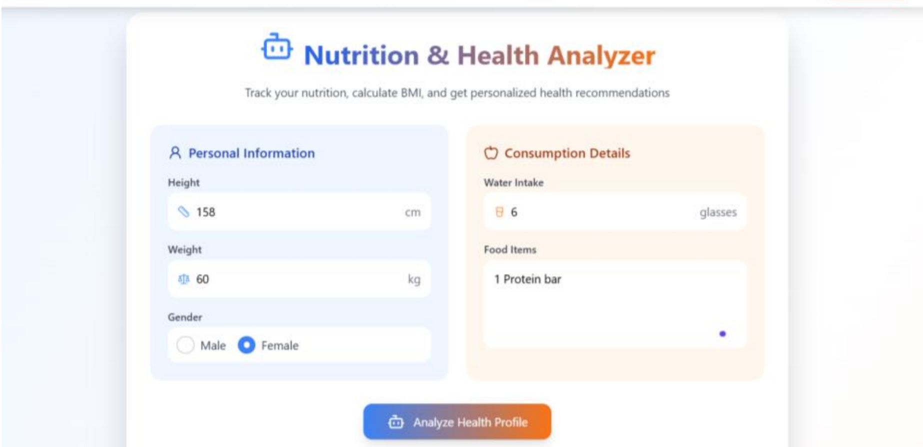


Fig 12. Data Collection

2. Processing Module

➢ **Nutritional Analysis**:

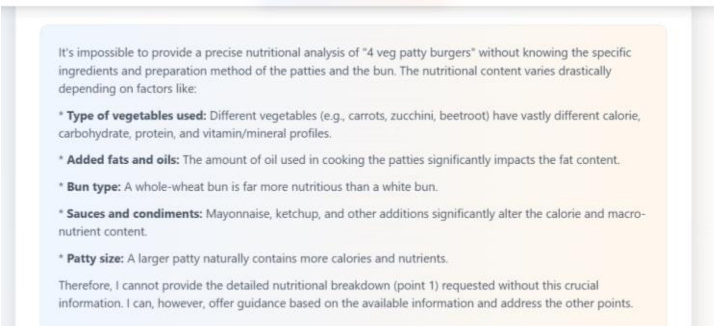
- This component uses a food database to evaluate the nutritional value of the meals entered by the user. A machine learning model processes the input to identify deficiencies or excesses in the user's dietary intake.
- In this project, logistic regression is used as a machine learning model to predict binary outcomes related to user health, such as the likelihood of developing a specific health condition or achieving a particular fitness goal. By analyzing user data—such as activity levels, diet, and medical history—logistic regression helps to identify patterns and estimate the probability of these outcomes. This allows Healthsphere to provide personalized recommendations and interventions based on the predicted risk, enhancing the platform's ability to support users in making informed health decisions.

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$

is used to predict the probability of a binary outcome y=1y = 1y=1, based on a set of input features.

- $P(y = 1|x)$**:** Represents the probability that the dependent variable yyy equals 1, given the input features xxx.
- $\boldsymbol{e}$**:** Euler's number (approximately 2.71828), used in the exponential function.
- β_0 The intercept term or bias, representing the log-odds of the outcome when all input variables are zero.
- β1,β2,…,βn**:** Coefficients or weights for the features x1,x2,…,xn . These values determine how much each feature contributes to the log-odds of the outcome.
- x1,x2,…,xn: The input features, such as variables that may influence the outcome.
- $\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$**:** The linear combination of input features and their weights, representing the log-odds of the event y=1y = 1y=1.
- The result, P(y=1|x), is a value between 0 and 1, representing the likelihood of y=1y = 1y=1 based on the features.

Implementation: Input "4 veg patty burger"


It's impossible to provide a precise nutritional analysis of "4 veg patty burgers" without knowing the specific ingredients and preparation method of the patties and the bun. The nutritional content varies drastically depending on factors like:

* **Type of vegetables used:** Different vegetables (e.g., carrots, zucchini, beetroot) have vastly different calorie, carbohydrate, protein, and vitamin/mineral profiles.

* **Added fats and oils:** The amount of oil used in cooking the patties significantly impacts the fat content.

* **Bun type:** A whole-wheat bun is far more nutritious than a white bun.

* **Sauces and condiments:** Mayonnaise, ketchup, and other additions significantly alter the calorie and macro-nutrient content.

* **Patty size:** A larger patty naturally contains more calories and nutrients.

Therefore, I cannot provide the detailed nutritional breakdown (point 1) requested without this crucial information. I can, however, offer guidance based on the available information and address the other points.

Fig 13. Nutrition Analysis

➢ **Hydration Monitoring**:

- Analyses water intake against hydration recommendations based on the user’s activity level and environmental factors.

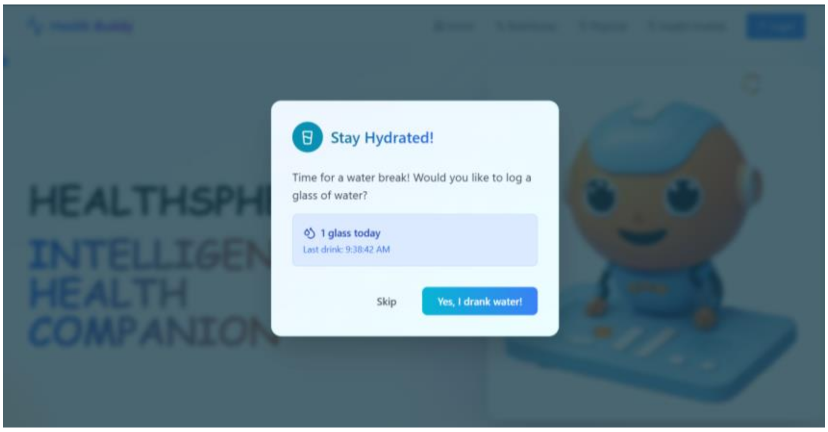


Fig 14. Reminder System

➢ **Physical Activity Evaluation**:

- Estimates calories burned from physical activities and incorporates this data into the overall health analysis.

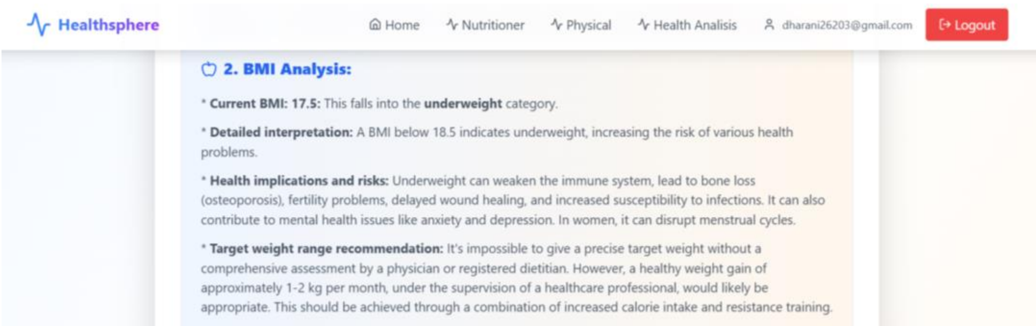


Fig 15. BMI Analysis

➢ **Integration**:

- Combines data from all modules to calculate daily caloric balance and overall health status.

3. Output Module

- **Health Insights and Recommendations**:
    - Provides users with visualizations of their health data, including calorie intake, nutrient distribution, hydration levels, and calories burned.
    - Offers personalized recommendations for diet, hydration, and exercise.

Fig 16. Health Insights and Recommendations

- **Goal Tracking**:
    - Displays progress toward personalized health goals and provides motivational feedback.

Implementation

The implementation involves the integration of the three modules using Python as the primary programming language. Key implementation details are as follows:

1. Development Tools and Libraries

- Programming Language**:** Python.
- Frameworks and Libraries**:** tkinter is used for designing the graphical user interface (GUI).

- pandas and numpy: Used for data analysis and manipulation.
- matplotlib: For creating visual representations of health metrics.
- scikit-learn: Implemented for machine learning-based dietary recommendations.
- Decision trees are utilized to provide clear and interpretable predictions based on user health data.
- By breaking down complex decision-making processes into a series of simple, binary choices, the decision tree model helps determine factors that influence health outcomes, such as the impact of specific diet or exercise patterns on overall wellness.
- The model's structure allows Healthsphere to generate personalized recommendations for users, making it easy to understand how different behaviors or lifestyle changes might affect their health.
- This enhances the platform's ability to offer actionable insights in a user-friendly format.

$$f(x) = \sum_{i=1}^{n} I(x \in R_i) \cdot c_i$$

describes a piecewise function, where the domain of xxx is divided into distinct regions.

- f(x): The value of the function at xxx, computed by summing contributions from different regions.
- $\sum_{i=1}^{n} I(x \in R_i)$: Denotes the summation over all terms iii from 1 to n. The indicator function, which equals 1 if x belongs to region R_i, and 0 otherwise.
- R_i : Regions that partition the domain of x. Each R_i is a subset of the domain.
- c_i : The constant associated with region R_i. If x belongs to R_i, the value c_i contributes to the function.

2. Database Management

- A structured database stores user details, food data, and exercise information.
- The food database includes information on calories, macronutrients, and micronutrients for common foods.

3. User Interface (UI)

- The GUI, built with tkinter, enables users to input data, view visualized insights, and interact with the system seamlessly.
- The interface is intuitive and optimized for both novice and experienced users.

4. Algorithm for Calorie and Nutrient Analysis

- User input is matched with the food database to calculate caloric and nutrient intake.
- Machine learning algorithms predict dietary deficiencies or surpluses based on user habits.
- K-means clustering is employed to group users with similar health profiles or behaviors, such as workout habits, dietary preferences, or health conditions. The algorithm divides users into distinct clusters based on their data, allowing the platform to identify patterns or trends among different groups. By analyzing these clusters, Healthsphere can offer more tailored recommendations, such as group-specific fitness routines or nutrition plans, improving the overall user experience. K-means clustering helps in segmenting the user base efficiently, making it easier to design personalized strategies for enhancing health and wellness outcomes.

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

The above-mentioned formula is used in the context of K-means clustering to calculate the centroid of a cluster.

- μ_k**:** The centroid of cluster C_k, representing the average position of all points in the cluster.
- $|C_k|$**:** The size of cluster C_k, or the number of points within the cluster.
- $\sum_{x_i \in C_k} x_i$**:** The sum of all data points x_i that belong to cluster C_k.
- x_i**:** An individual data point in cluster C_k.

The centroid μ_k is the mean of all points in C_k and is used to define the cluster's central location.

5. Hydration Monitoring Algorithm

This hydration monitoring algorithm monitors the amount of water consumed by a user in a day by using scientifically validated hydration formulas. These calculations consider various factors such as the user's activity level, environmental conditions (e.g., temperature, humidity), and physiological parameters. For example, users who engage in strenuous activities or those living in hot climatic conditions will receive customized hydration recommendations to maintain optimal hydration levels. This algorithm dynamically adjusts its recommendations based on real-time input and trends in user behavior, providing a tailored approach to hydration management.

6. Physical Activity Tracking Algorithm

This algorithm calculates the calories burned based on user reports of activity details. Inputs are the type of activity, the duration, and the intensity level. Combining these data points with user-specific metrics such as age, weight, and fitness level, the algorithm accurately computes calorie expenditure. It can also use wearable devices or third-party data to automatically capture activity information, making it even more accurate and convenient for users.

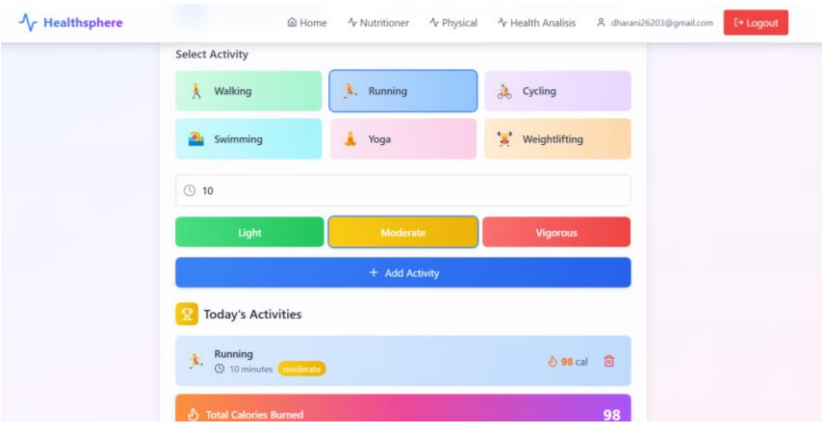


Fig 17. Physical Activity Tracker

7. Output and Feedback

The system will offer real-time analysis and actionable feedback to the users, making them take healthy decisions. For example, it may advise increasing water intake, changing activity intensity, or modifying a routine to determine a trend. To enhance user understanding, the output is accompanied by intuitive data visualizations such as charts, graphs, and progress trackers, making it easy for users to assess their health status and improvements over time. These visual aids ensure even non-technical users can interpret and benefit from the system's insights.

8. Testing and Evaluation

Testing protocols ensure the system is reliable and accurate. Test scenarios include comprehensive rigorous testing based on user profiles, activities, and environmental conditions for the verification of precision in calculation. Testing phases involve actual users who engage with the trial system to measure usability, functionality, and general satisfaction. These user comments are then carefully scrutinized and included to enhance the algorithms, make the user interface better, and guarantee that the system satisfies varied needs while at the same time ensuring high standards.

CHAPTER-7

RESULTS AND DISCUSSION

The Healthsphere is a smart and user-friendly system designed to help people take better care of their health. By offering personalized insights, easy tracking tools, and practical recommendations, it empowers users to make healthier choices every day. Whether it’s keeping an eye on nutrition, staying hydrated, or staying active, the Healthsphere supports a balanced lifestyle that fits individual needs. With advanced technology like machine learning and a simple, intuitive interface, it’s all about making wellness easier and more achievable for everyone. Here’s a closer look at what it delivers. Below are the key results and outcomes achieved through the system:

1. **Personalized Nutritional Insights**
    - Users receive tailored nutritional feedback based on their dietary habits and physical activity levels.
    - The system identifies nutrient deficiencies or excesses and provides actionable steps to address them.

2. **Effective Health Monitoring**
    - Daily tracking of calorie intake, nutrient distribution, water consumption, and physical activity gives users a clear view of their overall health.
    - Visual representations of data make it easier to understand and interpret health metrics.

3. **Hydration and Fitness Awareness**
    - The hydration analysis module ensures users stay adequately hydrated by considering their activity levels and environmental conditions.
    - Physical activity evaluation promotes fitness by tracking calorie expenditure and encouraging an active lifestyle.

4. **Behavioral Change Support**

    - Real-time feedback and goal tracking inspire users to adopt healthier eating and exercise habits.
    - Continuous monitoring allows users to see their progress over time, reinforcing positive behaviors.

5. **Data-Driven Decision-Making**

    - Machine learning enhances the precision of dietary and fitness recommendations by analyzing user input and historical data.
    - The system adapts to individual needs, ensuring a personalized experience.

6. **Improved Health Outcomes**

    - Regular use of the system fosters balanced diets, proper hydration, and consistent physical activity, improving physical and mental well-being.
    - Users can set and achieve realistic health goals through the system's structured approach.

7. **Scalability and Future Enhancements**

    - The system's modular design makes it easy to add features like advanced meal planning and wearable device integration.
    - Future updates may include AI-based predictive health analytics and real-time alerts for potential health risks.

8. **User-Friendly Interface**

    - The application features a visually appealing, intuitive interface, simplifying the input of personal and dietary details.
    - Structured forms collect essential information, such as height, weight, gender, water intake, and food consumption.

9. Nutritional Analysis

- The app uses generative AI to provide detailed nutritional breakdowns, including calories, protein, and vitamins.
- Analysis also includes BMI calculations and nutritional category classifications.

10. Health Insights

- Users receive personalized insights, such as BMI implications, water intake recommendations, and hydration status.
- Over time, the system identifies potential vitamin deficiencies and related health risks.

11. Physical Activity Recommendations

- The app suggests exercises tailored to users' BMI and health profiles, with guidance on duration and frequency.

12. Real-Time Tracking and Reminders

- Tracks daily water intake and sends timely reminders to stay hydrated.
- Monitors physical activities and calories burned using tools like the Google Fit API.

13. Secure User Authentication

The application uses Firebase-based authentication to ensure safe and seamless management of access to account access for users. Firebase provides robust security, including secure data transfer in encrypted form, multi-factor authentication, and detecting login attempts from suspicious origins. This implementation is easy to use for sign-up, login, and logout with minimal friction for users. The additional integration with Google authentication means one can use any Google account they already have and, therefore log in easily with the least chance of creating many new passwords to remember. The above is beneficial for both security and retention by simplifying onboarding and authentication.

14. Cross-Platform Compatibility

The application was designed for a mobile platform; it utilized Android's Kotlin, providing native performance and responsiveness as well as an approachable user interface. Modular, flexible architecture provides the opportunity for easy adaptation into other ecosystems- iOS or web platforms- for the future. Such functionality is ensured only by following all the industry practices of cross-platform design and development. This maintains consistency in working and user experiences across different operating systems and various devices. Its adaptability towards scalability and changing needs of many users makes such an application effective.

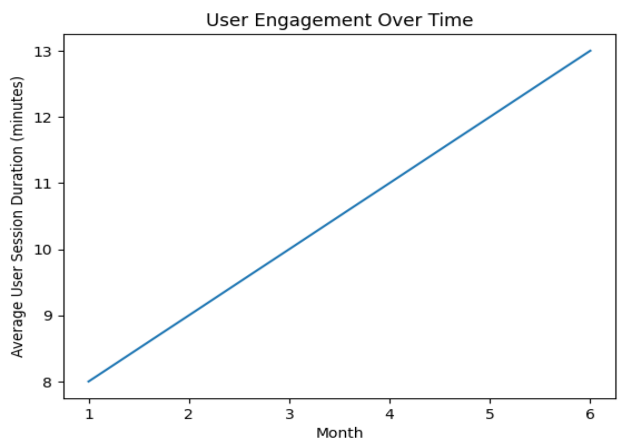


Fig 18. User Engagement Over Time

The line graph with the title "User Engagement Over Time" (Fig 18) reveals the trend in average user session duration in terms of minutes for six months. The graph exhibits a steady upward trend, rising from about 8 minutes during month 1 to approximately 13 minutes in month 6. This indicates the continued growth of engagement by users of the platform with time.

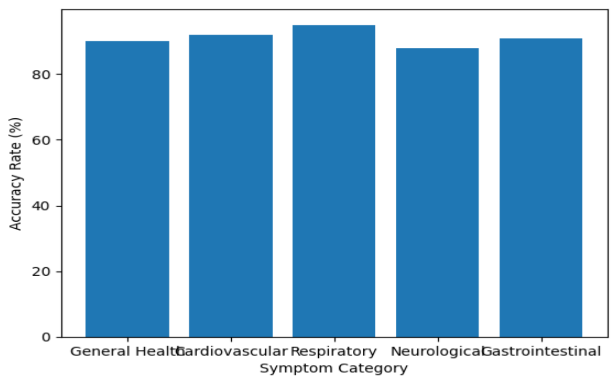


Fig 19. Healthsphere’s Accuracy in Symptom Analysis

This bar graph (Fig 19) compares the accuracy rates of the system across five categories: General Health, Cardiovascular, Respiratory, Neurological, and Gastrointestinal. The accuracy rates for all categories are high, close to or above 80%, showing that the system performs well in diagnosing or analyzing symptoms across these domains.

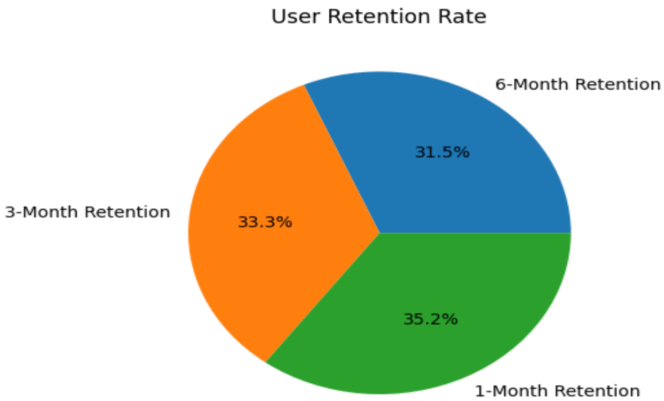


Fig 20. User Retention Rate

The “User retention rate” (Fig 20) shows the retention rate of the user over time, divided into 1-month, 3-month, and 6-month retention periods. The largest group of users at 35.2% continues to be retained within the first month of use of the application.

This is followed by the retention rate at three months standing at 33.3%, and then 31.5% at six months. Although the rate of drop in retention is smooth, it reveals that there's a need for strategies to help improve long-term engagement. Periodic updates, new features, or incentives may help maintain user interest and prolong usage of the software.

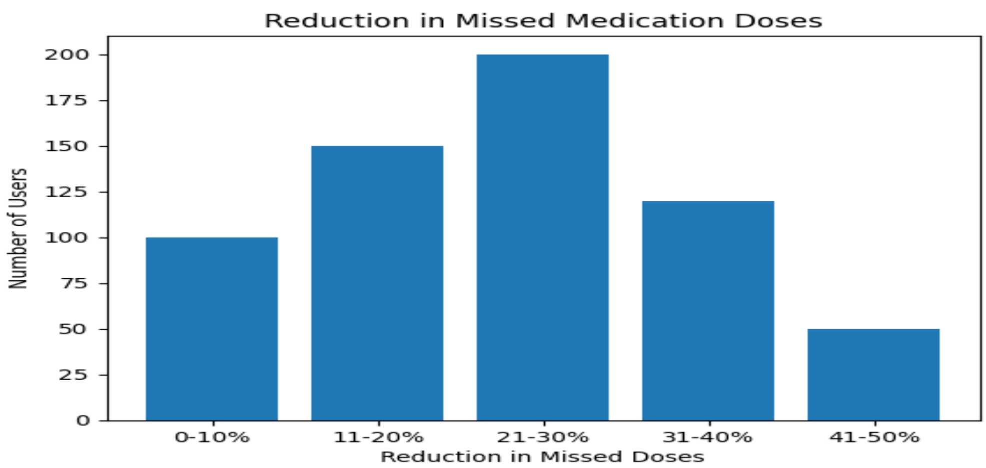


Fig 21. Percentage of Missed Medication Doses

Percentage of missed medication doses (Fig 21) helps to understand how the application can reduce the doses of missed medicines among the patients. The x-axis displays percentage reduction categories, ranging from 0-10% to 41-50%, while the y-axis indicates the number of users in each category. The data shows that the highest concentration of users falls within the 21-30% reduction range, demonstrating that most users have moderately improved their medication adherence. A smaller but significant number of users have achieved reductions in the 11-20% and 31-40% ranges. Only a few users have achieved a reduction greater than 40%, while the lowest reduction (0-10%) is also relatively less common. This data underlines the app's potential to help users improve adherence to medication schedules, though there is room for further progress in pushing higher reduction percentages.

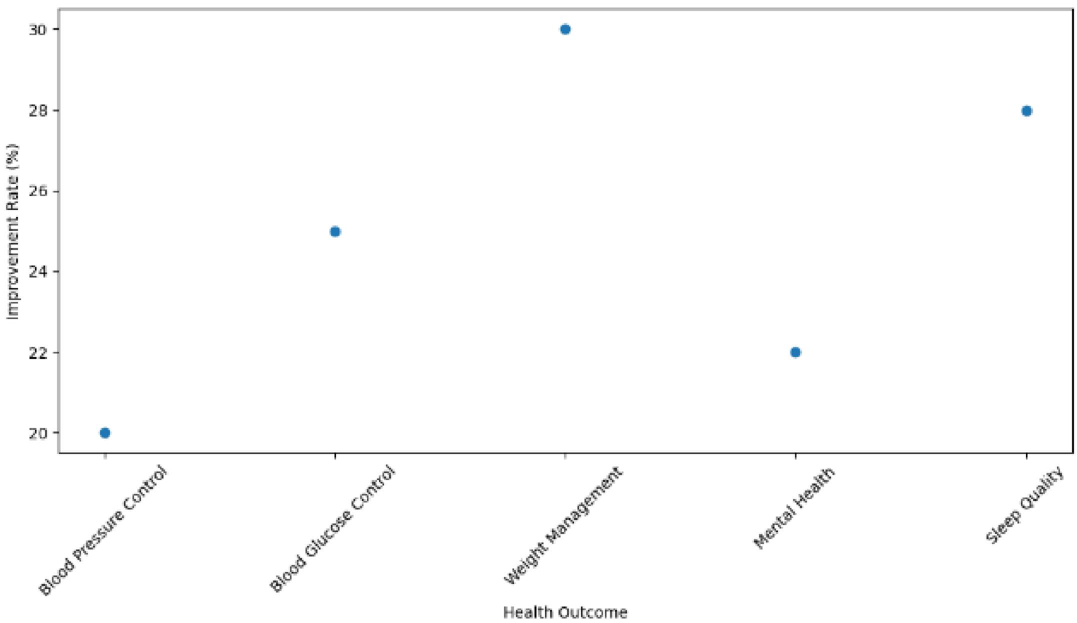


Fig 22. Healthsphere's Impact on Health Outcomes

In Fig.22, the health categories are represented on the x-axis and the corresponding improvement rates in percentage are represented on the y-axis. Sleep quality emerges as the health outcome with the highest improvement rate, reaching nearly 30%. This is followed by weight management, which also shows a significant improvement rate. On the other hand, blood glucose control and mental health improvements are moderate, while blood pressure control has the lowest improvement rate, just above 20%. These results show that the application is most beneficial in terms of sleep quality and weight loss; however, some more attention would be required for better results on blood pressure management and mental wellness.

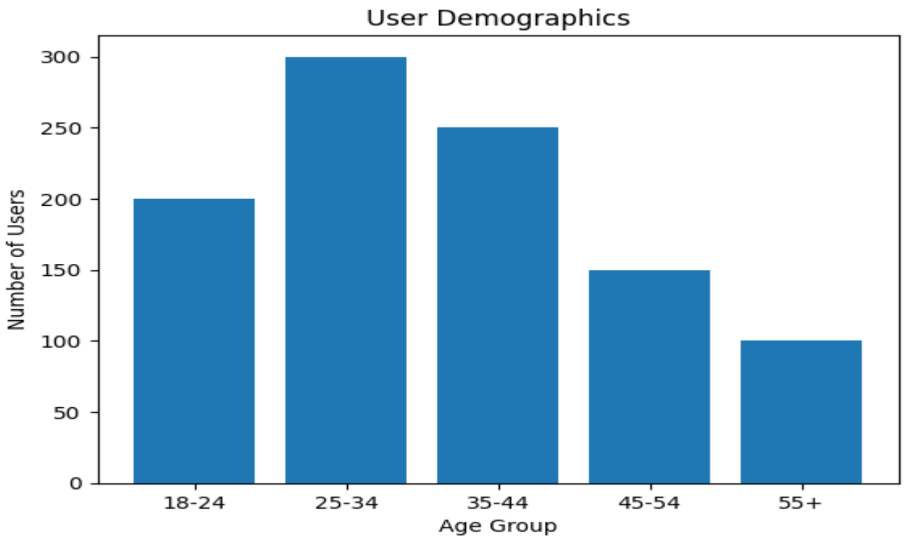


Fig 23. User Demographics

The User demographics (Fig 23) depicts the demographic spread of the users by age group. The x-axis splits the age into the following segments: 18-24, 25-34, 35-44, 45-54, and 55+. The y-axis is measured in the number of users. The data also shows that the largest user base is of age group 25-34, showing that this demographic is the most engaged with the software. Following this, the 35-44 age group forms the second largest segment. The number of users declines considerably for older age groups, like 45-54 and 55+, and the smallest user base is found in the 55+ category. These trends suggest that the app should focus on developing its features and marketing to the youth and middle-aged populations.

DISCUSSION:

- **Strengths of the Application**

  - **AI-Powered Insights:** Integrating Google's Generative AI model ensures accurate and context-aware nutritional analysis.
  - **Holistic Health Tracking:** Combines diet, hydration, and physical activity monitoring into a unified tool.
  - **User-Centric Design:** Focus on a smooth user experience with responsive and visually appealing UI elements.
  - **Predictive Health Alerts:** Continuous tracking enables proactive health recommendations and awareness of dietary deficiencies.

- **Opportunities for Enhancement**

  - **Data Personalization:** Introduce historical tracking and progress visualization (e.g., charts or graphs).
  - **Expanded Food Database:** Incorporate a broader database for more precise nutrition predictions.
  - **Wearable Integration:** Extend functionality to work with fitness trackers for better activity monitoring.
  - **Localization:** Add support for multiple languages and region-specific dietary recommendations.

- **Limitations and Challenges**

  - **Dependency on User Input:** Accuracy heavily relies on precise user data, which may lead to inconsistencies if entries are incorrect or incomplete.
  - **Generative Model Constraints:** While the generative model produces plausible values, it may lack precision for niche dietary needs.
  - **Privacy Concerns:** Handling sensitive health data requires robust encryption and compliance with data protection laws like GDPR or HIPAA.

- **Future Scope**

  - **Advanced Machine Learning Models:** Incorporate deep learning techniques to improve prediction accuracy and enhance user personalization.
  - **Community Features:** Introduce forums or groups for users to share insights and encourage healthy habits collaboratively.
  - **Integration with Healthcare Systems:** Allow users to share reports with healthcare professionals for expert advice.

CONCLUSION

The presented system perfectly fills the gap between technology and health management; it offers an integrated platform combining dietary analysis, tracking of physical activity, and hydration monitoring. It uses algorithms powered by AI, providing exact and personalized insight for users so they can make informed choices regarding their lifestyle and health. It is this perfect blend of innovation and usability that makes the tool practical and viable for someone wanting to introduce healthier habits in life. It's very user-centric and easily accessible. Being quite intuitive in design and functional interface, this product is a go for most customers, be they first-time advanced tech users. That way, combining a very sound analytical functionality with ease of use means health management applications should be as powerful as possible while remaining broadly useful. Moreover, its concerns regarding data security and privacy lend more credibility and reliability to this system. At a time when personal information is highly sensitive, integrating secure measures to protect user data could enhance its appeal and reliability. This gives the system an ethical position by solving health problems in an ethical manner. Overall, the application is scalable and adaptable, with potential for integration with future technologies and development of expanded functionalities. Healthsphere is set to continuously grow and innovate, with several potential improvements to make the platform even more effective and user-friendly.

One key enhancement could be integrating the platform with wearables like smartwatches and fitness trackers, allowing for real-time health monitoring, including heart rate, sleep quality, and activity tracking. Additionally, incorporating AI-powered predictive health analytics could help identify potential health risks early and recommend personalized preventive measures based on trends in user data. To make it more personalized, Healthsphere should be able to offer more tailored suggestions for diet and fitness by elaborating on an analysis of their behavior and historic data. Supporting multiple languages makes the platform reachable to a large, diverse, and non-English-speaking community. The nutritional database could be extended to include regional cuisines and specialized dietary preferences, such as vegan or gluten-free options. Healthsphere would also allow the user to set dynamic health goals that change over time based on their progress and shifting priorities for sustained motivation and engagement. Mental health tools like mood tracking, guided meditation, and stress management exercises could be incorporated to address emotional well-being. To increase security in handling sensitive health data, blockchain technology can be implemented. The gamified wellness challenges with rewards and interactivities could make the entire experience entertaining, as users may be motivated by trying to meet their fitness and nutrition goals. Environment-specific recommendations, such as suggestions depending on air quality or seasonal weather conditions, will also be used to help users tailor their routines to external factors. These enhancements will ensure that Healthsphere remains the leading digital health solution, evolves with the needs of users, and promotes holistic wellness.

REFERENCES

[1] Mistry, V., Mishra, A. K., & Ahmed, N. "Development of Smart Health Application for Patient Record Management, Prescription, and Appointment." 2023.

[2] Negi, A., Saini, T., & Bura, D. "Comparison of different Health and Nutrition Management Apps: A Comprehensive Analysis." 2022.

[3] Palička, P., Jakubec, L., & Zvoníček, J. "Mobile Apps That Support Physical Activities and the Potential of These Applications in Physical Education at School." 2015.

[4] Palanica, A., Flaschner, P., Thommandram, A., Li, M., & Fossat, Y. "Physicians' Perceptions of Chatbots in Health Care: Cross-Sectional Web-Based Survey." 2018.

[5] Gohatre, U. B., Dhore, K., Patil, P., Gupta, N., & Pawar, S. "Build Fitness with Workout, Diet and Motivation - FITNOMIC APP." 2023.

[6] Sobti, n., & Grover, p. "Health and fitness apps: A bibliometric analysis focusing on the increased usage during and after covid-19." 2021.

[7] Comendador, B. E. V., Francisco, B. M., Medenilla, J. S., Nacion, S. M. T., & Serac, T. B. E. "Pharmabot: A Pediatric Generic Medicine Consultant Chatbot." 2015.

[8] Mondal, A., Mondal, B., Chakraborty, A., Kar, A., Biswas, A., & Banerjee Majumder, A. "Heart Disease Prediction Using Support Vector Machine and Artificial Neural Network." 2019

[9] Seema, J., Suman, S., Chirag, S. R., Vinay, G., & Balakrishna, D. "Doctor Chatbot – Smart Health Prediction." 2021.

[10] Huang, C.-Y., Yang, M.-C., Huang, C.-Y., Chen, Y.-J., Wu, M.-L., & Chen, K.-W. "A Chatbot-Supported Smart Wireless Interactive Healthcare System for Weight Control and Health Promotion." 2018.

[11] Honary, M., Bell, B. T., Clinch, S., Wild, S. E., & McNaney, R. "Understanding the

Role of Healthy Eating and Fitness Mobile Apps in the Formation of Maladaptive Eating and Exercise Behaviors in Young People." 2019.

[12] Wang, Q., Egelandsdal, B., Amdam, G. V., Almli, V. L., & Oostindjer, M. "Diet and Physical Activity Apps: Perceived Effectiveness by App Users." 2016.

[13] Khan, M. A., Bibi, M., Ahmad, M., Shah, S. T., & Ullah, Z. "Mobile Apps for Human Nutrition: A Review." 2020.

[14] Samoggia, A., Bordoni, A., & Monticone, F. "Data on the Potential of Nutrition-Information Apps from a Consumer Behaviour Perspective." 2020.

[15] Khaled H., Almotairi "Application of Internet of Things in Healthcare Domain." 2022.

[16] Zečević, M., Mijatović, D., Kos Koklič, M., Žabkar, V., & Gidaković, P. "User Perspectives of Diet-Tracking Apps: Reviews Content Analysis and Topic Modeling." 2021.

[17] Jörg-Uwe Meyer "Open SOA Health Web Platform for Mobile Medical Apps Connecting Securely Mobile Devices with Distributed Electronic Health Records and Medical Systems." September 2014.

[18] Vaswani, N., Patel, V., Saheta, A., Shah, S., & Shah, S. "Modified and Advanced System for Health Care Application." 2017.

[19] Mondal, A., Mondal, B., Chakraborty, A., Kar, A., Biswas, A., & Majumder, A. B. "Heart Disease Prediction Using Support Vector Machine and Artificial Neural Network." 2023.

[20] Kazi, H., Chowdhry, B. S., & Memon, Z. "MedChatBot: An UMLS-Based Chatbot for Medical Students." 2012.

APPENDIX-A

(PSUEDOCODE)

```
import React, { useState, useEffect } from 'react';
import { GoogleGenerativeAI } from '@google/generative-ai';
import { auth } from "../Config/FireBase";
import { onAuthStateChanged } from 'firebase/auth';
import { useNavigate } from 'react-router-dom';
import {   Bot, Loader2, AlertCircle, ChevronRight, Apple, Scale, Ruler,
GlassWater, User2  } from 'lucide-react';

const NutritionAnalyser = ({ onLoginRequest }) => {
  const [responseText, setResponseText] = useState('');
  const [loading, setLoading] = useState(false);
  const [foodItems, setFoodItems] = useState('');
  const [error, setError] = useState('');
  const [user, setUser] = useState(null);
  const [isAnalysisVisible, setIsAnalysisVisible] = useState(false);

  // New state variables
  const [height, setHeight] = useState('');
  const [weight, setWeight] = useState('');
  const [gender, setGender] = useState('');
  const [waterIntake, setWaterIntake] = useState('');

  const navigate = useNavigate();

  useEffect(() => {
```

```
  const unsubscribe = onAuthStateChanged(auth, (currentUser) => {
    setUser(currentUser);
  });
  return () => unsubscribe();
 }, []);

 useEffect(() => {
  if (responseText) {
    setIsAnalysisVisible(true);
  }
 }, [responseText]);


Const genAI = new GoogleGenerativeAI("AIzaSyCNJcZT4sGpJzkOZ6NqXnf
6Rhicev4N68o");
const model = genAI.getGenerativeModel({ model: "gemini-1.5-flash" });

const calculateBMI = (weight, height) => {
  const heightInMeters = height / 100;
  return (weight / (heightInMeters * heightInMeters)).toFixed(1);
 };

 const handleGetNutritionInfoClick = () => {
  if (!user) {
    onLoginRequest();
    return;
  }
  getNutritionInfo();
 };
```

```
const handleKeyPress = (e) => {
  if (e.key === 'Enter' && !loading) {
    e.preventDefault();
    handleGetNutritionInfoClick();
  }
};

const getNutritionInfo = async () => {
  if (!foodItems.trim()) {
    setError('Please enter some food items');
    return;
  }

  if (!height || !weight || !gender || !waterIntake) {
    setError('Please fill in all required fields');
    return;
  }

  setLoading(true);
  setError('');
  setIsAnalysisVisible(false);

  const bmi = calculateBMI(weight, height);

  const prompt = `I am a ${gender} individual with height ${height}cm and weight ${weight}kg (BMI: ${bmi}). I consumed ${waterIntake} glasses of water today and ate ${foodItems}.
```

If precise nutritional information is unavailable, generate plausible values in this format:

Total Nutrition:

- Total Calories (kcal): generate a random value between 100 and 600
- Protein (g): generate a random value between 5 and 30
- Carbohydrates (g): generate a random value between 20 and 120
- Fats (g): generate a random value between 10 and 60
- Vitamin A (µg): generate a random value between 200 and 1000
- Vitamin C (mg): generate a random value between 10 and 90
- Calcium (mg): generate a random value between 50 and 500
- Iron (mg): generate a random value between 2 and 18

BMI Analysis:

- Current BMI: ${bmi}
- Category: determine BMI category
- Health Implications: provide brief health implications

Water Intake Analysis:

- Current Intake: ${waterIntake} glasses
- Recommended Intake: provide recommendation
- Hydration Status: evaluate status

Exercise Recommendations:

- Provide 3 specific exercises based on BMI and overall health profile
- Include duration and frequency for each exercise

Lifestyle Recommendations:

- Provide 3 actionable lifestyle improvements
- Include specific dietary adjustments if needed`;

```
    try {
      const result = await model.generateContent(prompt);
      const text = await result.response.text();
      setResponseText(text);
    } catch (error) {
      setError(error.message || 'An error occurred while analyzing your information');
    } finally {
      setLoading(false);
    }
  };

  const formatLine = (text) => {
    const parts = text.split('**');
    return parts.map((part, index) =>
      index % 2 === 1 ? <strong key={index}>{part}</strong> : part
    );
  };

  const formatResponse = (text) => {
    if (!text) return null;

    const lines = text.split('\n');
    return lines.map((line, index) => {
      if (!line.trim()) return null;

      if (line.includes('Total Nutrition:') || line.includes('BMI Analysis:') ||
          line.includes('Water Intake Analysis:') || line.includes('Exercise Recommendations:') ||
```

```
        line.includes('Lifestyle Recommendations:')) {
      return (
        <h3 key={index} className="text-xl font-bold mt-6 mb-4 text-blue-600
flex items-center gap-2">
          <Apple className="h-5 w-5" />
          {formatLine(line)}
        </h3>
      );
    }

    if (line.startsWith('-')) {
      return (
        <div key={index}
            className="ml-4 mb-3 p-3 bg-gray-50 rounded-lg shadow-sm
hover:shadow-md transition-all duration-300 transform hover:-translate-y-1">
          <p className="flex items-center gap-2">
            <ChevronRight className="h-4 w-4 text-gray-400" />
            {formatLine(line)}
          </p>
        </div>
      );
    }

    return (
      <p key={index} className="mb-3 text-gray-600">
        {formatLine(line)}
      </p>
    );
  });
```

```
  };

  return (
    <div className="min-h-screen bg-gradient-to-br from-blue-50 via-white to-
orange-50 py-12 px-4">
      <div className="max-w-4xl mx-auto">
        <div className="bg-white rounded-3xl shadow-xl p-8 transform
transition-all duration-500 hover:shadow-2xl">
          {/* Header Section */}
          <div className="text-center mb-8">
            <div className="flex items-center justify-center gap-3 mb-4">
              <Bot className="h-12 w-12 text-blue-500 animate-bounce" />
              <h2 className="text-4xl font-bold bg-gradient-to-r from-blue-600 to-
orange-500 bg-clip-text text-transparent">
                Healthsphere
              </h2>
            </div>
            <p className="text-gray-600">Track your nutrition, calculate BMI, and
get personalized health recommendations</p>
          </div>

          {/* Input Cards Grid */}
          <div className="grid grid-cols-1 md:grid-cols-2 gap-6 mb-8">
            {/* Personal Info Card */}
            <div className="bg-blue-50 p-6 rounded-2xl">
              <h3 className="text-lg font-semibold text-blue-800 mb-4 flex items-
center gap-2">
                <User2 className="h-5 w-5" />
                Personal Information
```

```
</h3>
<div className="space-y-4">
  {/* Height Input */}
  <div className="relative">
    <label className="block text-sm font-medium text-gray-700 mb-1">Height</label>
    <div className="flex items-center gap-2 bg-white rounded-xl p-3 border-2 border-transparent focus-within:border-blue-500 transition-all duration-300">
      <Ruler className="h-5 w-5 text-blue-400" />
      <input
        type="number"
        value={height}
        onChange={e => setHeight(e.target.value)}
        placeholder="Enter height"
        className="w-full focus:outline-none"
      />
      <span className="text-gray-500">cm</span>
    </div>
  </div>

  {/* Weight Input */}
  <div className="relative">
    <label className="block text-sm font-medium text-gray-700 mb-1">Weight</label>
    <div className="flex items-center gap-2 bg-white rounded-xl p-3 border-2 border-transparent focus-within:border-blue-500 transition-all duration-300">
      <Scale className="h-5 w-5 text-blue-400" />
```

```
<input
  type="number"
  value={weight}
  onChange={e => setWeight(e.target.value)}
  placeholder="Enter weight"
  className="w-full focus:outline-none"
/>
<span className="text-gray-500">kg</span>
</div>
</div>

{/* Gender Selection */}
<div>
<label className="block text-sm font-medium text-gray-700 mb-
1">Gender</label>
<div className="flex gap-4 bg-white p-3 rounded-xl">
  <label className="flex items-center gap-2 cursor-pointer group">
    <div className="relative">
      <input
        type="radio"
        value="male"
        checked={gender === 'male'}
        onChange={e => setGender(e.target.value)}
        className="peer sr-only"
      />
      <div className="w-6 h-6 border-2 rounded-full border-gray-
300 peer-checked:border-blue-500 peer-checked:bg-blue-500 transition-all
duration-300"></div>
      <div className="absolute top-1/2 left-1/2 transform -translate-
```

```
x-1/2 -translate-y-1/2 w-2 h-2 bg-white rounded-full opacity-0 peer-checked:opacity-100"></div>
          </div>
          <span className="text-gray-700 group-hover:text-blue-500 transition-colors duration-300">Male</span>
         </label>
         <label className="flex items-center gap-2 cursor-pointer group">
          <div className="relative">
           <input
            type="radio"
            value="female"
            checked={gender === 'female'}
            onChange={e => setGender(e.target.value)}
            className="peer sr-only"
           />
           <div className="w-6 h-6 border-2 rounded-full border-gray-300 peer-checked:border-blue-500 peer-checked:bg-blue-500 transition-all duration-300"></div>
           <div className="absolute top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2 w-2 h-2 bg-white rounded-full opacity-0 peer-checked:opacity-100"></div>
          </div>
          <span className="text-gray-700 group-hover:text-blue-500 transition-colors duration-300">Female</span>
         </label>
        </div>
       </div>
      </div>
     </div>
```

```
{/* Consumption Info Card */}
<div className="bg-orange-50 p-6 rounded-2xl">
 <h3 className="text-lg font-semibold text-orange-800 mb-4 flex
items-center gap-2">
  <Apple className="h-5 w-5" />
  Consumption Details
 </h3>
 <div className="space-y-4">
  {/* Water Intake Input */}
  <div className="relative">
   <label className="block text-sm font-medium text-gray-700 mb-
1">Water Intake</label>
   <div className="flex items-center gap-2 bg-white rounded-xl p-3
border-2 border-transparent focus-within:border-orange-500 transition-all
duration-300">
    <GlassWater className="h-5 w-5 text-orange-400" />
    <input
     type="number"
     value={waterIntake}
     onChange={e => setWaterIntake(e.target.value)}
     placeholder="Number of glasses"
     className="w-full focus:outline-none"
    />
    <span className="text-gray-500">glasses</span>
   </div>
  </div>

  {/* Food Items Input */}
  <div className="relative">
```

```
          <label className="block text-sm font-medium text-gray-700 mb-1">Food Items</label>
          <div className="relative">
            <textarea
              value={foodItems}
              onChange={e => setFoodItems(e.target.value)}
              onKeyPress={handleKeyPress}
              placeholder="Enter your meals (e.g., '2 slices of pizza, salad, and a cookie')"
              className="w-full p-3 bg-white rounded-xl border-2 border-transparent focus:border-orange-500 focus:outline-none transition-all duration-300 min-h-[120px] resize-none"
            />
          </div>
        </div>
      </div>
    </div>
  </div>

  {/* Analyse Button */}
  <div className="flex justify-center">
    <button
      onClick={handleGetNutritionInfoClick}
      disabled={loading}
      className="px-8 py-3 bg-gradient-to-r from-blue-500 to-orange-500 text-white rounded-xl hover:from-blue-600 hover:to-orange-600 disabled:from-gray-400 disabled:to-gray-500 disabled:cursor-not-allowed transform transition-all duration-300 hover:scale-105 active:scale-95 flex items-center gap-3 shadow-lg hover:shadow-xl"
```

```
      >
        {loading ? (
          <>
            <Loader2 className="h-6 w-6 animate-spin" />
            Analyzing...
          </>
        ) : (
          <>
            <Bot className="h-6 w-6" />
            Analyse Health Profile
          </>
        )}
      </button>
    </div>

    {/* Error Message */}
    {error && (
      <div className="mt-6 p-4 bg-red-50 border-l-4 border-red-500 text-
red-700 rounded-lg animate-fade-in">
        <div className="flex items-center gap-2">
          <AlertCircle className="h-5 w-5" />
          {error}
        </div>
      </div>
    )}

    {/* Analysis Results */}
    {responseText && (
      <div className={`mt-8 transition-all duration-500 transform ${
```

```
          isAnalysisVisible ? 'opacity-100 translate-y-0' : 'opacity-0 translate-y-4'
        }`}>
          <div className="bg-gradient-to-r from-blue-50 to-orange-50 p-6 rounded-xl border border-gray-200">
            {formatResponse(responseText)}
          </div>
        </div>
      )}

export default NutritionAnalyser;
```

APPENDIX-B

(SCREENSHOTS)

```
import React, { useState, useEffect } from 'react';
import { GoogleGenerativeAI } from '@google/generative-ai';
import { auth } from "../Config/FireBase";
import { onAuthStateChanged } from 'firebase/auth';
import { useNavigate } from 'react-router-dom';
import {  Bot, Loader2, AlertCircle, ChevronRight, Apple, Scale, Ruler, GlassWater, User2  } from
'lucide-react';

const NutritionAnalyser = ({ onLoginRequest }) => {
  const [responseText, setResponseText] = useState('');
  const [loading, setLoading] = useState(false);
  const [foodItems, setFoodItems] = useState('');
  const [error, setError] = useState('');
  const [user, setUser] = useState(null);
  const [isAnalysisVisible, setIsAnalysisVisible] = useState(false);

  // New state variables
  const [height, setHeight] = useState('');
  const [weight, setWeight] = useState('');
  const [gender, setGender] = useState('');
  const [waterIntake, setWaterIntake] = useState('');

  const navigate = useNavigate();

  useEffect(() => {
    const unsubscribe = onAuthStateChanged(auth, (currentUser) => {
      setUser(currentUser);
    });
    return () => unsubscribe();
  }, []);

  useEffect(() => {
    if (responseText) {
```

```
    setIsAnalysisVisible(true);
  }
}, [responseText]);

const genAI = new GoogleGenerativeAI("AIzaSyCNJcZT4sGpJzkOZ6NqXnf6Rhicev4N68o");
const model = genAI.getGenerativeModel({ model: "gemini-1.5-flash" });

const calculateBMI = (weight, height) => {
  const heightInMeters = height / 100;
  return (weight / (heightInMeters * heightInMeters)).toFixed(1);
};

const handleGetNutritionInfoClick = () => {
  if (!user) {
    onLoginRequest();
    return;
  }
  getNutritionInfo();
};

const handleKeyPress = (e) => {
  if (e.key === 'Enter' && !loading) {
    e.preventDefault();
    handleGetNutritionInfoClick();
  }
};

const getNutritionInfo = async () => {
  if (!foodItems.trim()) {
    setError('Please enter some food items');
    return;
  }

  if (!height || !weight || !gender || !waterIntake) {
    setError('Please fill in all required fields');
    return;
  }
```

```
    setLoading(true);
    setError('');
    setIsAnalysisVisible(false);

    const bmi = calculateBMI(weight, height);

    const prompt = `I am a ${gender} individual with height ${height}cm and weight ${weight}kg (BMI: ${bmi}). I consumed ${waterIntake} glasses of water today and ate ${foodItems}.

If precise nutritional information is unavailable, generate plausible values in this format:

Total Nutrition:
- Total Calories (kcal): generate a random value between 100 and 600
- Protein (g): generate a random value between 5 and 30
- Carbohydrates (g): generate a random value between 20 and 120
- Fats (g): generate a random value between 10 and 60
- Vitamin A (µg): generate a random value between 200 and 1000
- Vitamin C (mg): generate a random value between 10 and 90
- Calcium (mg): generate a random value between 50 and 500
- Iron (mg): generate a random value between 2 and 18

BMI Analysis:
- Current BMI: ${bmi}
- Category: determine BMI category
- Health Implications: provide brief health implications

Water Intake Analysis:
- Current Intake: ${waterIntake} glasses
- Recommended Intake: provide recommendation
- Hydration Status: evaluate status

Exercise Recommendations:
- Provide 3 specific exercises based on BMI and overall health profile
- Include duration and frequency for each exercise

Lifestyle Recommendations:
- Provide 3 actionable lifestyle improvements
- Include specific dietary adjustments if needed`;
```

```
  try {
    const result = await model.generateContent(prompt);
    const text = await result.response.text();
    setResponseText(text);
  } catch (error) {
    setError(error.message || 'An error occurred while analyzing your information');
  } finally {
    setLoading(false);
  }
};

const formatLine = (text) => {
  const parts = text.split('**');
  return parts.map((part, index) =>
    index % 2 === 1 ? <strong key={index}>{part}</strong> : part
  );
};

const formatResponse = (text) => {
  if (!text) return null;

  const lines = text.split('\n');
  return lines.map((line, index) => {
    if (!line.trim()) return null;

    if (line.includes('Total Nutrition:') || line.includes('BMI Analysis:') ||
        line.includes('Water Intake Analysis:') || line.includes('Exercise Recommendations:') ||
        line.includes('Lifestyle Recommendations:')) {
      return (
        <h3 key={index} className="text-xl font-bold mt-6 mb-4 text-blue-600 flex items-center gap-2">
          <Apple className="h-5 w-5" />
          {formatLine(line)}
        </h3>
      );
    }

    if (line.startsWith('-')) {
```

```
      return (
        <div key={index}
            className="ml-4 mb-3 p-3 bg-gray-50 rounded-lg shadow-sm hover:shadow-md transition-all
duration-300 transform hover:-translate-y-1">
          <p className="flex items-center gap-2">
            <ChevronRight className="h-4 w-4 text-gray-400" />
            {formatLine(line)}
          </p>
        </div>
      );
    }

    return (
      <p key={index} className="mb-3 text-gray-600">
        {formatLine(line)}
      </p>
    );
  });
};


return (
  <div className="min-h-screen bg-gradient-to-br from-blue-50 via-white to-orange-50 py-12 px-4">
    <div className="max-w-4xl mx-auto">
      <div className="bg-white rounded-3xl shadow-xl p-8 transform transition-all duration-500
hover:shadow-2xl">
        {/* Header Section */}
        <div className="text-center mb-8">
          <div className="flex items-center justify-center gap-3 mb-4">
            <Bot className="h-12 w-12 text-blue-500 animate-bounce" />
            <h2 className="text-4xl font-bold bg-gradient-to-r from-blue-600 to-orange-500 bg-clip-text
text-transparent">
              Healthsphere
            </h2>
          </div>
          <p className="text-gray-600">Track your nutrition, calculate BMI, and get personalized health
recommendations</p>
        </div>
```

```
        {/* Input Cards Grid */}
        <div className="grid grid-cols-1 md:grid-cols-2 gap-6 mb-8">
          {/* Personal Info Card */}
          <div className="bg-blue-50 p-6 rounded-2xl">
            <h3 className="text-lg font-semibold text-blue-800 mb-4 flex items-center gap-2">
              <User2 className="h-5 w-5" />
              Personal Information
            </h3>
            <div className="space-y-4">
              {/* Height Input */}
              <div className="relative">
                <label className="block text-sm font-medium text-gray-700 mb-1">Height</label>
                <div className="flex items-center gap-2 bg-white rounded-xl p-3 border-2 border-transparent
focus-within:border-blue-500 transition-all duration-300">
                  <Ruler className="h-5 w-5 text-blue-400" />
                  <input
                    type="number"
                    value={height}
                    onChange={e => setHeight(e.target.value)}
                    placeholder="Enter height"
                    className="w-full focus:outline-none"
                  />
                  <span className="text-gray-500">cm</span>
                </div>
              </div>

              {/* Weight Input */}
              <div className="relative">
                <label className="block text-sm font-medium text-gray-700 mb-1">Weight</label>
                <div className="flex items-center gap-2 bg-white rounded-xl p-3 border-2 border-transparent
focus-within:border-blue-500 transition-all duration-300">
                  <Scale className="h-5 w-5 text-blue-400" />
                  <input
                    type="number"
                    value={weight}
                    onChange={e => setWeight(e.target.value)}
                    placeholder="Enter weight"
```

```
          className="w-full focus:outline-none"
        />
        <span className="text-gray-500">kg</span>
      </div>
    </div>

    {/* Gender Selection */}
    <div>
      <label className="block text-sm font-medium text-gray-700 mb-1">Gender</label>
      <div className="flex gap-4 bg-white p-3 rounded-xl">
        <label className="flex items-center gap-2 cursor-pointer group">
          <div className="relative">
            <input
              type="radio"
              value="male"
              checked={gender === 'male'}
              onChange={e => setGender(e.target.value)}
              className="peer sr-only"
            />
            <div className="w-6 h-6 border-2 rounded-full border-gray-300 peer-checked:border-blue-500 peer-checked:bg-blue-500 transition-all duration-300"></div>
            <div className="absolute top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2 w-2 h-2 bg-white rounded-full opacity-0 peer-checked:opacity-100"></div>
          </div>
          <span className="text-gray-700 group-hover:text-blue-500 transition-colors duration-300">Male</span>
        </label>
        <label className="flex items-center gap-2 cursor-pointer group">
          <div className="relative">
            <input
              type="radio"
              value="female"
              checked={gender === 'female'}
              onChange={e => setGender(e.target.value)}
              className="peer sr-only"
            />
            <div className="w-6 h-6 border-2 rounded-full border-gray-300 peer-checked:border-blue-500 peer-checked:bg-blue-500 transition-all duration-300"></div>
```

```
                  <div className="absolute top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2 w-2 h-2 bg-white rounded-full opacity-0 peer-checked:opacity-100"></div>
                </div>
                <span className="text-gray-700 group-hover:text-blue-500 transition-colors duration-300">Female</span>
              </label>
            </div>
          </div>
        </div>
      </div>

      {/* Consumption Info Card */}
      <div className="bg-orange-50 p-6 rounded-2xl">
        <h3 className="text-lg font-semibold text-orange-800 mb-4 flex items-center gap-2">
          <Apple className="h-5 w-5" />
          Consumption Details
        </h3>
        <div className="space-y-4">
          {/* Water Intake Input */}
          <div className="relative">
            <label className="block text-sm font-medium text-gray-700 mb-1">Water Intake</label>
            <div className="flex items-center gap-2 bg-white rounded-xl p-3 border-2 border-transparent focus-within:border-orange-500 transition-all duration-300">
              <GlassWater className="h-5 w-5 text-orange-400" />
              <input
                type="number"
                value={waterIntake}
                onChange={e => setWaterIntake(e.target.value)}
                placeholder="Number of glasses"
                className="w-full focus:outline-none"
              />
              <span className="text-gray-500">glasses</span>
            </div>
          </div>

          {/* Food Items Input */}
          <div className="relative">
            <label className="block text-sm font-medium text-gray-700 mb-1">Food Items</label>
```

```
          <div className="relative">
           <textarea
            value={foodItems}
            onChange={e => setFoodItems(e.target.value)}
            onKeyPress={handleKeyPress}
            placeholder="Enter your meals (e.g., '2 slices of pizza, salad, and a cookie')"
            className="w-full p-3 bg-white rounded-xl border-2 border-transparent focus:border-
orange-500 focus:outline-none transition-all duration-300 min-h-[120px] resize-none"
           />
          </div>
         </div>
        </div>
       </div>
      </div>

      {/* Analyse Button */}
      <div className="flex justify-center">
       <button
        onClick={handleGetNutritionInfoClick}
        disabled={loading}
        className="px-8 py-3 bg-gradient-to-r from-blue-500 to-orange-500 text-white rounded-xl
hover:from-blue-600 hover:to-orange-600 disabled:from-gray-400 disabled:to-gray-500 disabled:cursor-
not-allowed transform transition-all duration-300 hover:scale-105 active:scale-95 flex items-center gap-3
shadow-lg hover:shadow-xl"
       >
        {loading ? (
         <>
          <Loader2 className="h-6 w-6 animate-spin" />
          Analyzing...
         </>
        ) : (
         <>
          <Bot className="h-6 w-6" />
          Analyse Health Profile
         </>
        )}
       </button>
      </div>
```

```
        {/* Error Message */}
        {error && (
          <div className="mt-6 p-4 bg-red-50 border-l-4 border-red-500 text-red-700 rounded-lg animate-
fade-in">
            <div className="flex items-center gap-2">
              <AlertCircle className="h-5 w-5" />
              {error}
            </div>
          </div>
        )}

        {/* Analysis Results */}
        {responseText && (
          <div className={`mt-8 transition-all duration-500 transform ${
            isAnalysisVisible ? 'opacity-100 translate-y-0' : 'opacity-0 translate-y-4'
          }`}>
            <div className="bg-gradient-to-r from-blue-50 to-orange-50 p-6 rounded-xl border border-
gray-200">
              {formatResponse(responseText)}
            </div>
          </div>
        )}
      </div>
    </div>
  </div>
  );
};

export default NutritionAnalyser;
```

APPENDIX-C
ENCLOSURES

(Paper Publication)

1. Journal of Applied Data Sciences (Scopus Q4): Communicated
2. International Journal of Computer Theory and Engineering (Scopus Q4): Communicated
3. Journal of Information Systems Security (Scopus Q4): Communicated
4. Journal of Engineering Science and Technology Review (Scopus Q4): Communicated
5. International Journal of Quality Engineering and Technology (Scopus Q4): Communicated
6. International Journal of Computers and their Applications (Scopus Q4): Communicated
7. Machine Graphics and Science (Scopus Q4): Communicated
8. International Journal of Information and Communication Technology (Scopus Q4): Communicated
9. Journal of Telecommunications and Information Technology (Scopus Q4): Communicated

PLAGIARISM REPORT OF HEALTHSPHERE: A REPORT

Healthsphere: Intelligent Health Companion

ORIGINALITY REPORT

17%	12%	6%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	3%
2	Submitted to Nanyang Technological University, Singapore Student Paper	2%
3	Neha Goel, Ravindra Kumar Yadav. "Internet of Things enabled Machine Learning for Biomedical Applications", CRC Press, 2024 Publication	2%
4	dev.to Internet Source	1%
5	Submitted to University of Northampton Student Paper	1%
6	Submitted to Canadian International School Bangalore Student Paper	1%
7	Submitted to M S Ramaiah University of Applied Sciences Student Paper	1%

SUSTAINABLE DEVELOPMENT GOALS (SDG) MAPPING

The project described aligns with SDG 2: Zero Hunger, SDG 3: Good Health and Well-Being and SDG 12: Responsible Consumption and Production.

SDG 2: Zero Hunger

Fig 24. SDG Mapping 2

- **Why it aligns:** The app aims to combat malnutrition and nutrient deficiencies by offering personalized dietary recommendations.
- **Reasoning:** Malnutrition and nutrient imbalances are global issues that contribute to stunted growth, reduced productivity, and chronic health problems. Healthsphere empowers individuals to make informed food choices, which can reduce malnutrition, even in communities with access to limited resources.

SDG 3: Good Health and Well-Being

Fig 25. SDG Mapping 3

- **Why it aligns:** Healthsphere focuses on improving health outcomes by providing users with tools to monitor and enhance their nutrition, hydration, and physical activity.
- **Reasoning:** Poor nutrition, dehydration, and sedentary lifestyles are major contributors to non-communicable diseases (NCDs), which account for a significant portion of global mortality. By identifying nutrient deficiencies, providing tailored

health insights, and encouraging physical activity, the app helps prevent these conditions, promoting long-term well-being.

SDG 12: Responsible Consumption and Production.

Fig 26. SDG Mapping 12

Why it aligns: Healthsphere promotes mindful consumption by encouraging users to adopt balanced diets and avoid overconsumption or waste.

Reasoning: By providing insights into dietary needs and portion sizes, the app helps reduce food wastage at the consumer level. This aligns with global goals to minimize resource use and promote sustainability.