



Glycobuddy: Building Scalable and User-Friendly AI Applications for Diabetic Dietary Management by Integrating FastAPI, PostgreSQL, and AI-Driven Recommendations

Muhammad Ibrahim Baba^{1*}, Abdullahi Musa², Kamorudeen Hassan Taiwo³, Ajayi Ore-Ofe¹, Nnamani Stanley Wisdom¹, Ibrahim Ibrahim¹

¹Computer Engineering Department, Ahmadu Bello University, Zaria.

²Federal Polytechnic,
Kaura Namoda Zamfara State.

³Senior Registrar, Department of Family Medicine,
Ahmadu Bello University Teaching Hospital, Zaria.

ABSTRACT

Diabetic patients face significant challenges in managing their dietary habits, particularly in regions like Nigeria, where traditional diets are rich in carbohydrates. This paper presents Glycobuddy, a scalable and AI-powered dietary management application designed specifically for diabetic patients in these regions. The app integrates the CLIP model for meal image classification and Gemini Large Language Models (LLMs) for natural language processing, providing personalized dietary recommendations based on glycemic index and patient dietary needs. Built using FastAPI and PostgreSQL for backend scalability and React Native for a cross-platform user interface, the system ensures real-time performance and secure data management. Glycobuddy achieved an 84.37% accuracy in classifying 24 Nigerian dishes, offering tailored nutritional advice to users. This paper discusses the system architecture, design decisions, performance evaluation, and future improvements, including the integration of real-time health data and enhanced personalization for diabetic dietary management.

ARTICLE INFO

Article History

Received: July, 2024

Received in revised form: July, 2024

Accepted: August, 2024

Published online: September, 2024

KEYWORDS

Vocational training, women's Diabetes management, AI-driven dietary recommendations, Mobile health applications, Backend architecture, Frontend interface, React Native, FastAPI, Scalable healthcare solutions, Real-time data processing, and User-centered design

INTRODUCTION

Diabetes mellitus is a chronic condition that affects millions of people globally, with more than 463 million adults living with diabetes as of 2019. This number is projected to rise to 700 million by 2045 [1]. In regions like Nigeria, where traditional diets tend to be carbohydrate-rich, managing diabetes through diet presents unique challenges. Proper nutrition plays a critical role in managing diabetes, yet there is a lack of tools specifically designed to meet the dietary needs of patients in these regions [2].

Artificial intelligence (AI) has emerged as a powerful tool in healthcare, providing personalized recommendations and automating complex tasks. AI-driven applications can help diabetic patients better manage their condition by offering tailored nutritional advice [3]. This paper presents Glycobuddy, an AI-powered

dietary management application designed to assist diabetic patients in Nigeria. The app utilizes the CLIP model for food image classification and Gemini Large Language Models (LLMs) for natural language processing, providing culturally relevant dietary recommendations.

The backend of Glycobuddy is built with FastAPI and PostgreSQL, ensuring scalability, real-time data processing, and secure data management. The frontend, developed with React Native, provides a seamless user experience across Android and iOS devices, allowing users to upload meal images and receive nutritional advice in real time. By integrating these technologies, Glycobuddy aims to fill a gap in the availability of dietary management tools for diabetic patients in Nigeria.

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



This paper explores the system architecture, design decisions, and overall performance of Glycobuddy. It also discusses future improvements, such as the integration of real-time health data and enhanced personalization features, to better assist diabetic patients in managing their dietary needs.

LITERATURE REVIEW

Snap-n-Eat (2015) was an early mobile application designed to automatically recognize food items and calculate their caloric and nutritional values using smartphone images. The app used low-level features like color and texture to achieve over 85% accuracy with a dataset of 2,000 images across 15 classes. However, it did not address the specific needs of diabetic patients or the intricacies of non-Western diets, particularly in regions like Nigeria. Glycobuddy builds on this concept by focusing on culturally relevant dishes and offering personalized dietary advice tailored for diabetic patients [5].

DiaFit (2016) is a mobile application aimed at enhancing patient-provider communication and promoting self-management for individuals with type 2 diabetes (T2D) and obesity. The app integrates physical activity, nutrition, glucose levels, and medication adherence data, supporting real-time monitoring through wearables. While DiaFit provides a comprehensive approach to obesity management, Glycobuddy focuses on diabetes-specific dietary recommendations, particularly for Nigerian dishes [6].

A 2016 review of mobile apps for diabetes management highlighted the potential of mobile health (mHealth) in real-time tracking of nutritional intake. However, many existing apps require manual data input, which limits user engagement. The review called for more advanced AI solutions that automate food classification and provide tailored recommendations. Glycobuddy addresses this gap by using AI models like CLIP and Gemini LLMs to offer culturally relevant dietary advice for diabetic patients in Nigeria [7].

In 2017, Petersen and Hempler developed a mobile app for newly diagnosed T2D patients, focusing on self-management,

goal setting, and psychological health tracking. Although the app was well-received, issues such as manual data entry and lack of Android compatibility hindered its usability. Glycobuddy builds on this approach by offering personalized dietary advice with a specific focus on the cultural and nutritional needs of diabetic patients in Nigeria [8].

mDiabetes (2017) showcased the potential of mobile health tools to improve medication adherence and glucose control. A review of 89 free diabetes management apps in 2017 found that while many were functional, they lacked credible, evidence-based dietary recommendations. Glycobuddy addresses this shortcoming by offering a comprehensive, evidence-based solution tailored to the dietary habits of Nigerian diabetic patients [9,10].

In 2023, M-Healthcare, a mobile app for type 2 diabetes patients, integrated features for tracking blood sugar, physical activity, medication, and sleep. Despite its effectiveness in diabetes management, M-Healthcare lacked region-specific dietary recommendations, a gap that Glycobuddy aims to fill by providing localized, AI-powered dietary advice [14].

METHODOLOGY

The methodology for this paper involved designing, developing, and testing the mobile app's user interface (UI) and backend system. Below is an outline of the phases followed during development, with a focus on creating an efficient and scalable system architecture, developing a user-friendly interface, and integrating backend services.

The system architecture as shown in figure 1 was designed to ensure scalability and seamless communication between the mobile application and backend services. This design facilitated efficient data exchange, secure user interactions, and responsive functionality.

The mobile application and backend system were designed using a multi-tier architecture consisting of the following components:

- a. Frontend Layer (Mobile App): Developed using React Native, this layer handles all user interactions, such as capturing and uploading meal images. The app communicates with the backend through

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



- API calls to retrieve meal-related information, process images, and receive recommendations.
- Backend Layer: The backend, built with FastAPI, processes API requests from the frontend, performs business logic operations, and interacts with the database. This layer manages user authentication, meal data storage, and image upload handling.
 - Database Layer: The PostgreSQL database stores all relevant data, including user profiles, meal metadata, and nutritional information. The database is structured to handle queries related to user preferences, meal history, and dietary analysis.
 - Cloud Storage and Processing: AWS S3 handles the storage of images uploaded by users. Once uploaded, images are securely stored and managed in the cloud, ready for backend processing.

The system follows a simple, user-centric interaction flow to ensure efficient data handling and responsive service:

- User Interaction: A user uploads a meal image through the mobile app either by using the camera feature in the app or uploading the meal image from local storage.
- Image Storage: The image is stored in AWS S3.
- Backend Request: The backend processes the request by interacting with the database and cloud storage.
- Response Delivery: The processed results are returned to the mobile app, which then displays the relevant information or recommendations to the user.

The following technology stack was selected to meet the paper objectives, focusing on scalability, performance, and simplicity:

- React Native: Chosen for its cross-platform capabilities, allowing the development of a unified mobile app for both iOS and Android.
- FastAPI: Selected for backend development due to its high performance, ease of creating RESTful APIs, and

simplicity in managing asynchronous processes.

- PostgreSQL: Used for data storage, providing reliable, structured data handling with efficient query support.
- AWS S3: Chosen for its secure and scalable image storage capabilities, facilitating seamless integration with the backend.
- GitHub: Employed for version control, enabling collaboration and tracking of development progress.

The user interface (UI) of the mobile application was designed to prioritize simplicity, responsiveness, and user accessibility. The app allows users to capture and upload meal images, view their dietary history, and receive recommendations based on their inputs. The frontend development process required setting up an environment conducive to cross-platform mobile development. The following tools and procedures were used:

- React Native and Expo: For cross-platform development of the mobile application.
- Node.js: For managing dependencies and server-side JavaScript execution.
- npm/Yarn: Package managers for handling project dependencies.
- Visual Studio Code: For testing and validating the app on Android and iOS platforms.

The backend of the application serves as the core engine, managing data flow between the user interfaces and underlying systems to ensure seamless user interactions. Built using FastAPI, the backend prioritizes scalability, performance, and security. FastAPI's support for asynchronous capabilities and modern architecture enables real-time data handling, while its automatic generation of OpenAPI documentation simplifies API management and maintenance.

The backend architecture was designed to support the application's core functionalities while ensuring high performance, scalability, and reliability. The technology stack used includes:

- FastAPI: The primary framework for the backend. FastAPI's asynchronous capabilities enable real-time data

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



processing and interaction, making it suitable for handling multiple requests concurrently. Its modular structure and dependency injection system enhance the backend's maintainability and scalability.

- b. PostgreSQL: Chosen for its robustness and ability to handle complex queries, PostgreSQL stores and manages all user data, including meal plans, food scans, and recipes. Its support for JSONB storage and full-text search makes it well-suited for handling structured and unstructured data efficiently.
- c. SQLAlchemy: Used as the Object-Relational Mapping (ORM) tool, SQLAlchemy simplifies database interactions by mapping database tables to Python objects, reducing the need for raw SQL queries.
- d. Alembic: For managing database migrations, Alembic was employed to handle schema changes over time without disrupting existing data. This

ensures smooth implementation of new features and enhancements.

- e. Docker and Kubernetes: Docker was used to containerize the backend application, ensuring consistency across different development and deployment environments. Kubernetes was employed for orchestration, enabling horizontal scaling and robust performance management, particularly during peak usage periods.

The backend employed SQLAlchemy to define model classes representing core entities within the application, such as users, meals, scans, and recipes. These models ensure that the data is organized efficiently and the relationships between entities are optimized for quick retrieval. The database schema illustrated in figure 2 was designed to support fast, reliable queries, ensuring that the application can scale while maintaining performance.

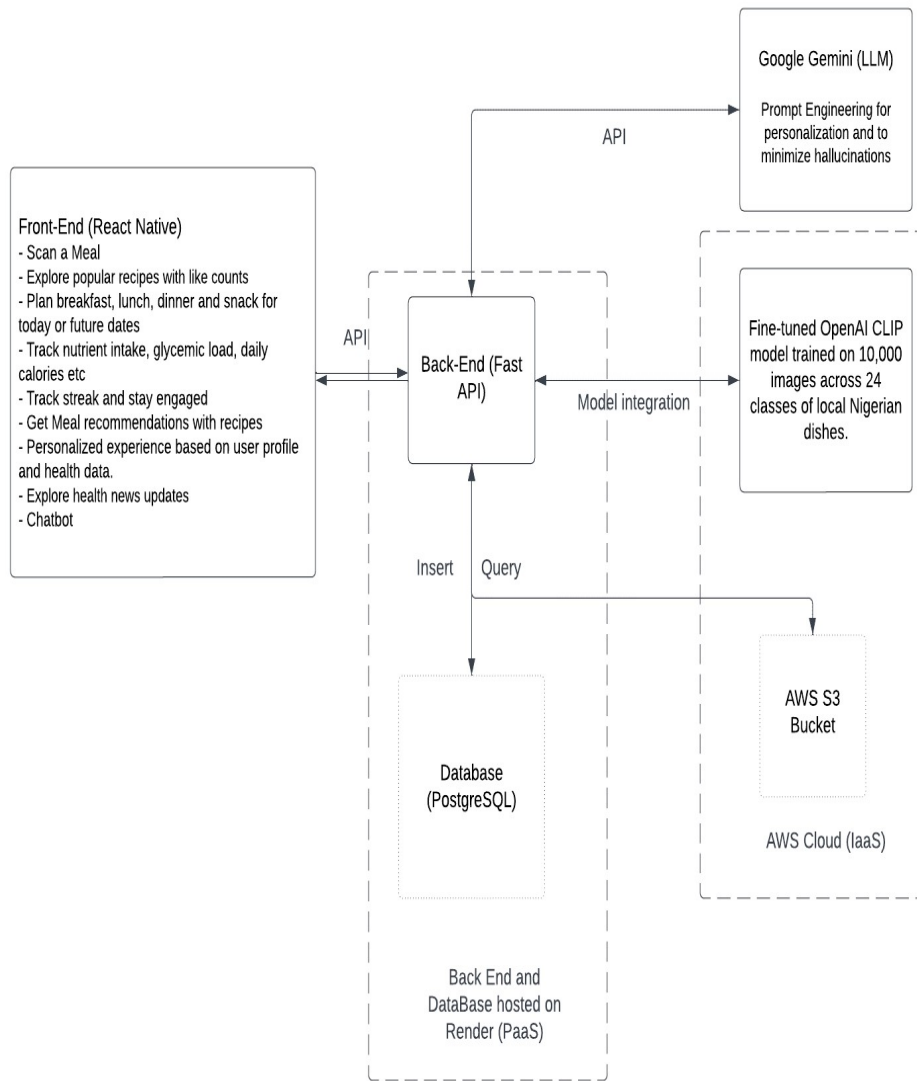


Figure 1: System Architecture

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

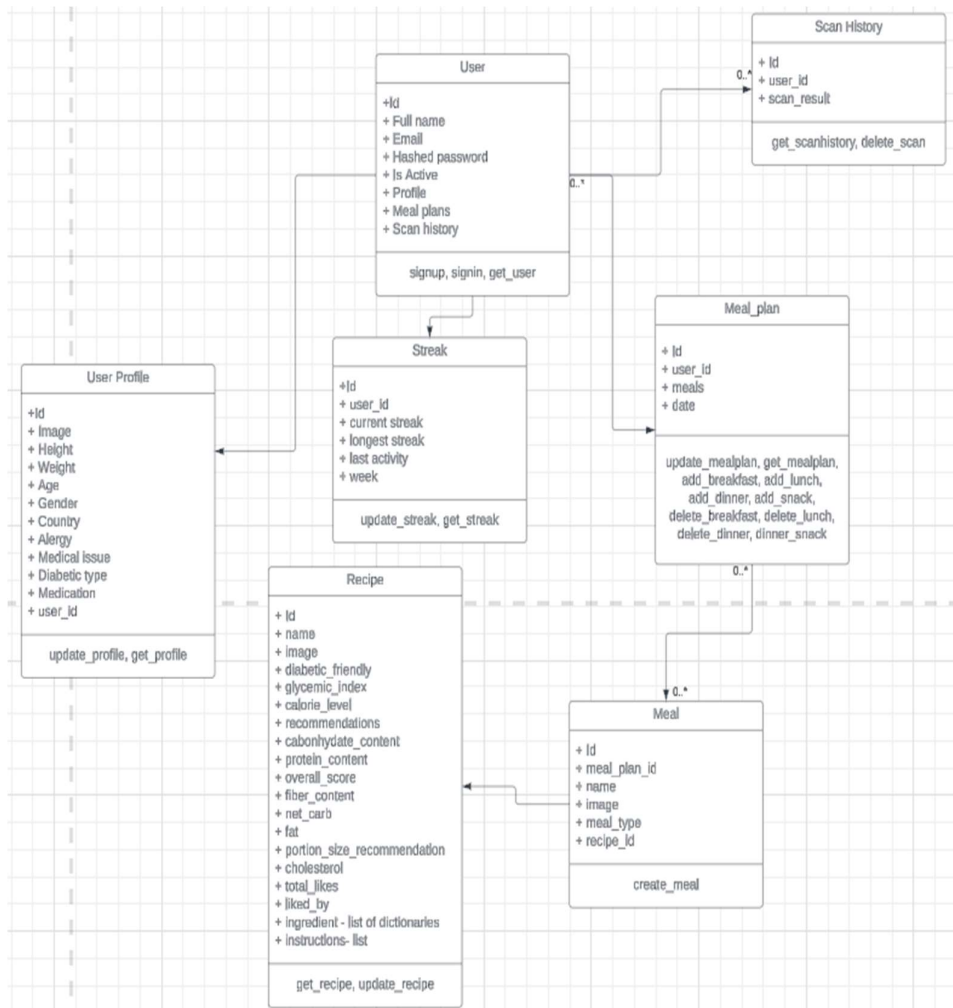


Figure 2: Database Schema

RESULTS AND DISCUSSIONS

The implementation of the dietary management application yielded several noteworthy results, particularly in the areas of user interface functionality and backend performance. The user interface demonstrated a high level of usability, with intuitive navigation and seamless interaction between core features, enhancing the overall user experience. The backend, built using FastAPI and PostgreSQL, effectively handled real-time requests, demonstrating both scalability and

reliability under various load conditions. This section delves into these outcomes, offering a critical analysis of the system's effectiveness in meeting the paper's objectives and addressing the identified challenges.

The dietary management application was successfully developed with a user-friendly interface that integrates backend functionalities which supports a wide range of features, aligning with the application's goal of assisting users in managing their dietary needs,

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

particularly for diabetic patients. The key functionalities include:

- a. **Food Scanning and Nutritional Analysis:** Users can upload images of food, which are processed for nutritional analysis. The backend integrates with AI models (CLIP and Gemini) to identify food items and generate detailed nutritional information, including glycemic index and calorie content, allowing users to make informed dietary choices.
- b. **Meal Planning:** Users can add identified food items to their meal plans, categorizing them by meal type (e.g., breakfast, lunch, dinner). This feature enables users to track and manage their daily nutritional intake, supporting better dietary control.
- c. **Health News Feed:** The home page includes a dynamic news feed that provides up-to-date health-related news, helping users stay informed about the latest developments in health, nutrition, and diabetes management.
- d. **Recipe Section:** The app offers a curated selection of recipes, each accompanied by nutritional information. This helps users prepare meals that align with their dietary restrictions and health goals.
- e. **Streak Feature:** To encourage consistent app use, a streak feature was implemented. It tracks user engagement by logging daily interactions with the app, helping to build and reinforce healthy habits.
- f. **Caloric and Glycemic Tracking:** The app automatically tracks users' daily caloric and glycemic intake, feeding this data into the recommendation system to provide tailored dietary advice based on the user's health profile.
- g. **Meal Recommendation:** Personalized meal recommendations are provided based on users' dietary history, health profiles, and previous meal choices. The recommendations include recipes that users can follow to prepare meals aligned with their nutritional needs.
- h. **Meal Planning Section:** Users can plan meals in advance, up to one month, using a calendar view. This feature displays

past, present, and future meal plans, allowing users to organize their meals effectively.

- i. **Scan History:** The app maintains a history of all scanned meals, allowing users to review their past dietary choices. This feature helps users track eating patterns and make informed adjustments to their diet.

The dietary application features some key UI components which include:

- a. **Home Screen (Figure 3):** Provides main navigation and meal planning (a), giving access to essential features like recommended recipes (b) and health news feed (c).
- b. **Meal Scanning Modal (Figure 4):** Enables users to capture and upload meal images.

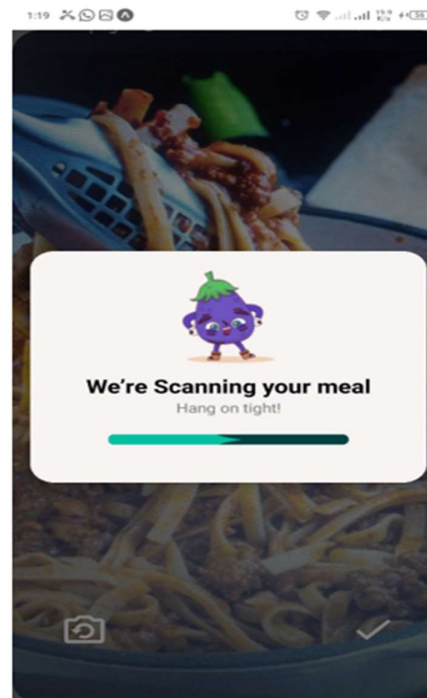


Figure 4: Scanning Meal Modal

- c. **Scan Result Screen (Figure 5):** Displays the results of meal scans, presenting food data retrieved from the backend.

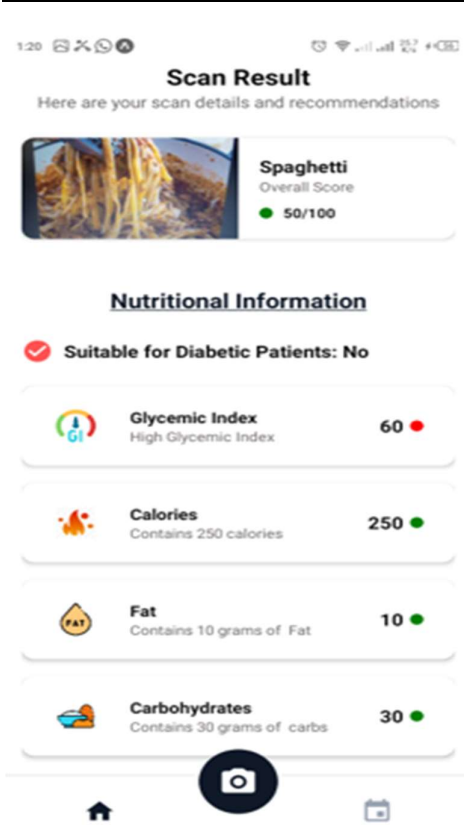


Figure 5: Scan Result Screen

- d. Scan Result Recommendations (Figure 6): Offers personalized nutritional insights and recommendations based on the meal input.



Figure 6: Scan Result Recommendation

- e. Meal Planning Section (Figure 7): Allows users to organize their meals in advance, facilitating daily, weekly, or monthly meal planning.

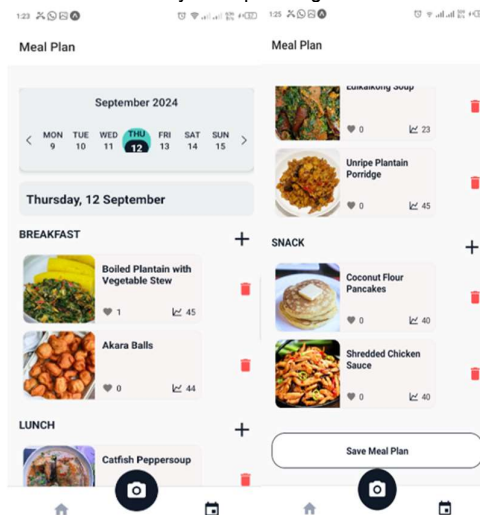
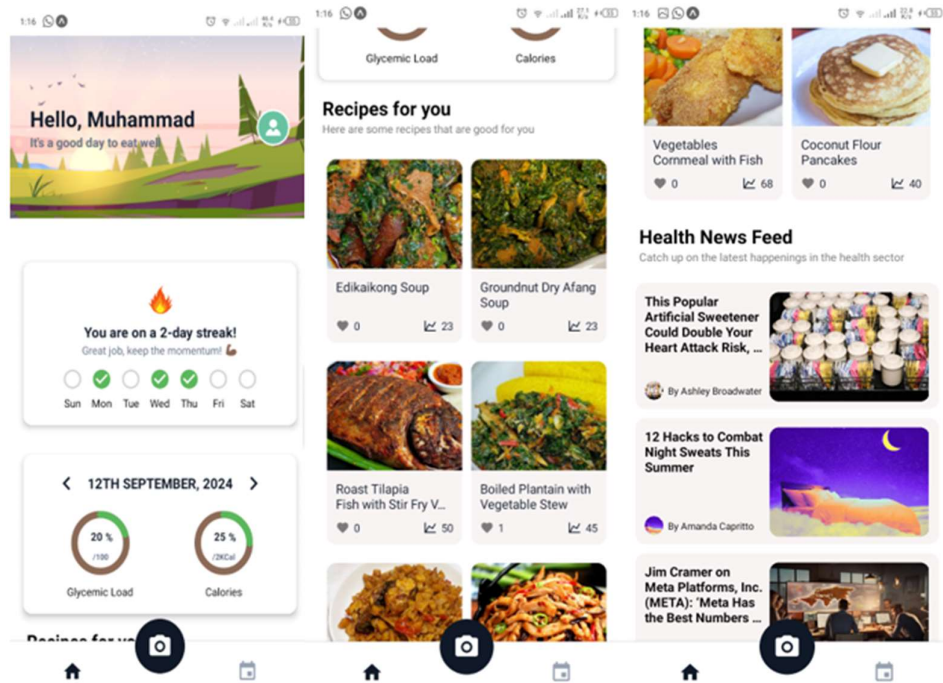


Figure 7: Meal Planning Section



(a) (b) (c)
Figure 3: Home Screen (a) Home Screen: “Recipes for You” Section (b) Home Screen: “Health News Feed” Section (c)

The backend system, developed using FastAPI and PostgreSQL, was evaluated for performance, reliability, and scalability. The system architecture shown in figure 1 illustrates the interaction between the frontend and backend components, showing how the backend processes user requests and manages data storage. The combination of FastAPI, PostgreSQL, and containerization with Docker and Kubernetes facilitated an efficient and scalable backend architecture, ensuring smooth operation during both normal and high-traffic periods. Key performance metrics of the backend include:

- a. Response Time: During testing, the backend consistently achieved low-latency response times, well below the set threshold for real-time interactions, validating the choice of FastAPI as the backend framework.
- b. Scalability: Scalability tests conducted using Docker and Kubernetes indicated that the backend could handle varying

traffic loads without performance degradation. Horizontal scaling was achieved seamlessly during peak usage scenarios, ensuring that the system can accommodate increasing user demands.

- c. Reliability: The backend exhibited stable performance under stress tests, confirming its ability to manage high volumes of requests without sacrificing data integrity or response speed.

User feedback was gathered to assess the overall usability and effectiveness of the application. The feedback was predominantly positive, particularly regarding the intuitive design of the user interface and the smooth navigation experience. Key findings from the user feedback include:

- a. Ease of Use: Users found the interface intuitive and easy to navigate. The straightforward design facilitated efficient access to core features like meal planning, scanning, and recipe browsing.

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



- b. Meal Planning: The meal planning feature was highly valued by users, allowing them to organize and track meals over time. This feature was highlighted as particularly useful for individuals managing specific dietary requirements.

The results demonstrated that the dietary management application effectively meets the paper's primary objectives, particularly in terms of providing a user-friendly interface and robust backend performance.

The user interface was highly appreciated for its simplicity and effectiveness in supporting meal planning and dietary management. Users could easily navigate between the app's various features, and the interface facilitated seamless interaction with the backend services.

The backend's low-latency performance and ability to scale horizontally highlight the efficiency of the chosen technology stack. FastAPI, PostgreSQL, and containerization tools like Docker and Kubernetes proved to be well-suited for handling asynchronous data requests and real-time interactions.

User feedback highlighted key areas for improvement, particularly around expanding customization in meal recommendations and integrating health tracking features. These suggestions represent potential avenues for future enhancements to the application.

CONCLUSION

The primary objective of this paper was to develop a scalable and user-friendly dietary management application tailored to the needs of diabetic patients, with a focus on Nigerian dietary habits. The application successfully integrates modern backend technologies with a clean and intuitive user interface, offering features such as meal planning, recipe browsing, and personalized nutritional insights.

A significant outcome of the paper was the development of a highly functional user interface (UI), designed to ensure ease of use and seamless navigation through core features. The meal planning and personalized recommendation functionalities were particularly well-received, with user feedback

indicating that the interface effectively supported dietary management. Additionally, the backend, built using FastAPI, PostgreSQL, Docker, and Kubernetes, demonstrated strong performance by handling real-time data requests efficiently. Scalability testing confirmed that the backend could manage increased loads, ensuring the system's responsiveness and robustness during peak usage.

This paper contributes to healthcare technology by creating a specialized dietary management solution that integrates a scalable backend with an intuitive, user-centric design. The success of the UI design, coupled with the reliable performance of the backend, highlights the effectiveness of combining modern technologies in supporting diabetic patients with their dietary needs. The application's design and architecture offer a practical, scalable solution that can be further developed to meet the growing demands of dietary management in healthcare.

REFERENCES

- [1] International Diabetes Federation, "Diabetes Facts and Figures," 2019.
- [2] American Diabetes Association, "Standards of Medical Care in Diabetes—2020," *Clinical Diabetes*, vol. 38, no. 1, pp. 10-38, 2020.
- [3] E. J. Topol, *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books, 2019.
- [1] International Diabetes Federation, "Diabetes Facts and Figures," 2019.
- [5] Zhang, J., Li, J., & Lin, Z. (2015). Snap-n-Eat: A mobile food recognition system. *Journal of Medical Internet Research*, 17(8), 234-241.
- [6] Modave, F., Bian, J., Rosenberg, E., Mendoza, T. M., Liang, Z., Bhosale, R., Maeztu, C., Rodriguez, C., & Cardel, M. I. (2016). DiaFit: The development of a smart app for patients with type 2 diabetes and obesity. *JMIR Diabetes*, 1(2), e5.
- [7] Darby, A., Strum, M. W., Holmes, E., & Gatwood, J. (2016). A review of nutritional tracking mobile applications for diabetes

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



- patient use. *Diabetes Technology & Therapeutics*, 18(3), 200-212.
- [8] Petersen, M., & Hempler, N. F. (2017). Development and testing of a mobile application to support diabetes self-management for people with newly diagnosed type 2 diabetes: A design thinking case study. *BMC Medical Informatics and Decision Making*, 17(91). <https://doi.org/10.1186/s12911-017-0493-6>
- [9] Trawley, S., Baptista, S., Browne, J., Pouwer, F., & Speight, J. (2017). The use of mobile applications among adults with type 1 and type 2 diabetes. *Diabetes Technology & Therapeutics*, 19(12), 730-738.
- [10] Chavez, S., Fedele, D., Guo, Y., Bernier, A., Smith, M., Warnick, J., & Modave, F. (2017). Mobile apps for the management of diabetes. *Diabetes Care*, 40(10), e145-e146.
- [11] Schmocker, K., Zwahlen, F., & Denecke, K. (2018). Mobile app for simplifying life with diabetes: Technical description and usability study of GlucoMan. *JMIR Diabetes*, 3(1), e6.
- [12] Weatherall, J., Paprocki, Y., Meyer, T., Kudel, I., & Witt, E. (2018). Sleep tracking and exercise in patients with type 2 diabetes mellitus (Step-D). *JMIR mHealth and uHealth*, 6(6), e131.
- [13] Shan, R., Sarkar, S., & Martin, S. S. (2019). Digital health technology and mobile devices for the management of diabetes mellitus: State of the art. *Diabetologia*, 62(6), 877-887.
- [14] Joshua, S. R., Abbas, W., & Lee, J.-H. (2023). M-Healthcare model: An architecture for a type 2 diabetes mellitus mobile application. *Applied Sciences*, 13(8), 200-212.

Corresponding author: Muhammad Ibrahim Baba.

✉ ibrahimmuhammad271@gmail.com

Computer Engineering Department, Ahmadu Bello University, Zaria, Kaduna State.

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved