VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI - 590018

DevOps Project Report
(22CSE1663)
On

Nutrition Tracking and Diet Management

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

1BG22CS043 Dumpa Revanth Venkata Sai

1BG22CS178 Venkateshwar Reddy H

1BG23CS401 Ashray V B

Internal Guide

Shrikanth Hampiholi.

Dept. of CSE,

BNMIT, Bengaluru

Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025

URL: www.bnmit.org

Department of Computer Science and Engineering

2024 - 2025

B.N.M. Institute of Technology

An Autonomous Institution under VTU
Approved by AICTE, Accredited as grade A Institution by NAAC. All eligible branches – CSE, ECE, EEE, ISE & Mech. Engg. are Accredited by NBA for academic years 2018-19 to 2024-25 & valid upto 30.06.2025
URL: www.bnmit.org

Department of Computer Science and Engineering

2024 – 2025

Vidyayāmruthamashnuthe

CERTIFICATE

Certified that the DevOps project entitled **Nutrition Tracking and Diet Management** carried out by **Dumpa Revanth Venkata Sai (1BG22CS043),Venkateshwar Reddy H(1BG22CS178), Ashray V B (1BG23CS401),** are Bonafide students of VI Semester, BNM Institute of Technology in partial fulfillment for the award of Bachelor of Engineering in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the year 2024-25. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the project report deposited in the departmental library. The Cloud Computing project report (22CSE1664) has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Shrikanth Hampiholi. **Department of CSE** **BNMIT, Bengaluru**	**Dr. Chayadevi M L** **Professor & HOD** **Department of CSE** **BNMIT, Bengaluru**	**Dr. S. Y. Kulkarni** **Additional Director and Principal** **BNMIT, Bengaluru**

	Name	**Signature with Date**
Examiner 1:		
Examiner 2:		

ACKNOWLEDGEMENT

We would like to place on record our sincere thanks and gratitude to the concerned people, whose suggestions and words of encouragement has been valuable.

We express our heartfelt gratitude to the management of **BNM Institute of Technology**, for giving us the opportunity to pursue Degree of Computer Science and Engineering and helping us to shape our career. We take this opportunity to thank **Shri. Narayan Rao Maanay**, Secretary, **Prof. T. J. Rama Murthy**, Director, **Dr. S. Y. Kulkarni**, Additional Director and Principal, **Prof. Eishwar N Maanay**, Dean and **Dr. Krishnamurthy G. N.**, Deputy Director for their support and encouragement to pursue this project. We would like to thank **Dr. Chayadevi M L**, Professor and Head, Dept. of Computer Science and Engineering, for her support and encouragement.

We would like to thank our guide **Prof. Shrikanth Hampiholi.** Dept. of Computer Science and Engineering, who has been the source of inspiration throughout our project work and has provided us with useful information at every stage of our project.

Finally, we are thankful to all the teaching and non-teaching staff of Department of Computer Science and Engineering for their help in the successful completion of our project. Last but not the least we would like to extend our sincere gratitude to our parents and all our friends who were a constant source of inspiration.

Dumpa Revanth Venkata Sai (1BG22CS043)
Venkateshwar Reddy H (1BG22CS178)
Ashray VB (1BG23CS401)

ABSTRACT

This project, titled “Nutrition Tracking and Diet Management”, is a full-stack web application designed to empower users in managing their dietary habits and achieving their health goals through data-driven insights. The system allows users to log meals, monitor their intake of calories, proteins, carbohydrates, and fats, and set personalized nutrition targets based on their lifestyle needs. The frontend is developed using React.js and Tailwind CSS for a clean, responsive user interface, while the backend is built with Node.js and Express.js to handle data processing and API interactions. MongoDB is used to store user data securely, and integration with external Nutrition APIs ensures the accuracy of food and nutrient information. From a DevOps standpoint, the project adopts fundamental practices including Source Code Version Control using Git and GitHub for collaborative development, and the use of Docker for containerizing backend services to ensure consistency across environments. Although full Continuous Integration and Deployment pipelines are not implemented, Docker simplifies testing and potential deployment workflows, enabling easier scalability and portability of services. The frontend is deployed through Netlify, which supports instant updates on code changes, facilitating a smoother deployment experience. Overall, the project reflects an effective fusion of modern web development with foundational DevOps principles to deliver a scalable, maintainable, and user-focused nutrition tracking solution.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER – I

INTRODUCTION

CHAPTER-I

INTRODUCTION

Proper nutrition plays a vital role in maintaining a healthy and productive life. With increasing awareness about fitness, well-being, and chronic disease prevention, individuals are becoming more conscious of their dietary habits. However, tracking and managing nutritional intake remains a challenge for most people due to lack of knowledge, time constraints, and insufficient access to accurate nutritional information. The modern lifestyle, characterized by sedentary habits and fast food consumption, has led to rising health issues such as obesity, diabetes, and cardiovascular diseases. Although mobile applications and wearable devices provide some health insights, many of them are either too generalized, overly complex, or not tailored to individual needs. Moreover, most of these applications are built without consistent software development practices, making them difficult to maintain, scale, or enhance over time. Therefore, there is a growing need for a user-centric platform that not only provides nutritional tracking features but is also built using modern DevOps methodologies for better reliability, scalability, and maintainability.

The Nutrition Tracking and Diet Management project addresses this problem by providing a complete web-based solution that combines the benefits of full-stack development with DevOps best practices. The application helps users monitor their diet, set health goals, and track macro- and micronutrients such as calories, proteins, fats, and carbohydrates. Visualizations and analytics are incorporated to provide real-time feedback and motivation. From a development operations standpoint, the project adopts containerization through Docker to ensure consistent environments, Git and GitHub for source code management and collaboration, and Netlify for seamless frontend deployment. This project not only solves a relevant real-world problem but also serves as a practical implementation of a DevOps-enabled software development lifecycle.

1.1 Statement of the Problem

There is a lack of accessible, personalized, and reliable tools to help individuals track their daily nutrition and dietary habits effectively.

In today's health-conscious world, individuals increasingly seek tools to manage their diet and stay informed about what they consume. However, many existing applications fall short in providing accurate nutritional data, customizable goals, and engaging user experiences.Users often find it difficult to track calories, macronutrients, and overall progress due to either the complexity of tools or limited food databases.

From the development perspective, these applications are frequently built without DevOps practices, leading to challenges in scalability, maintainability, and consistent deployments. Without standardized development and deployment pipelines, such systems often face issues related to versioning, environment mismatch, and downtime. Therefore, there is a need for a solution that combines a user-friendly nutrition tracking platform with a DevOps-oriented development and deployment pipeline to ensure both usability and technical robustness..

1.2 Objective of the Project

The main objective of this project is to build a comprehensive, scalable, and user-friendly nutrition tracking system using modern full-stack technologies and DevOps practices. The specific goals include:

- Allow users to log meals and snacks with nutritional breakdowns including calories, proteins, fats, and carbohydrates.
- Enable personalized dietary goals, based on individual user preferences or health objectives.
- Visualize nutrition trends and progress using graphs, charts, and dashboards to aid user motivation.
- Integrate verified Nutrition APIs for real-time and accurate food data sourcing.
- Implement Git and GitHub for source code version control to ensure collaborative, trackable development.
- Use Docker for containerizing the backend to enable consistency across development, testing, and production environments.
- Deploy the frontend using Netlify, supporting continuous updates with minimal manual intervention.
- Lay the foundation for future CI/CD automation, allowing the application to evolve into a fully DevOps-compliant product pipeline.

CHAPTER – II

LITERATURE SURVEY

CHAPTER-II

LITERATURE SURVEY

The field of digital health and nutrition tracking has seen rapid growth due to increasing public awareness of personal wellness and the integration of modern technology in daily life. Multiple studies and tools have been developed that align with the goals of NutriTrack, offering valuable insights into both technical implementations and user behavior. One foundational work in the DevOps domain is The Phoenix Project by Kim et al., which provides a narrative-driven explanation of how DevOps practices can transform software delivery, streamline workflows, and reduce deployment risks [1]. The emphasis on communication, automation, and continuous feedback in this model informs the DevOps integration strategy used in NutriTrack.

Continuous Integration (CI), a key aspect of modern software development, is critical to ensuring code quality and reliability. As Fowler discusses, CI involves regularly merging code changes into a shared repository, followed by automated builds and tests. This process minimizes integration issues and accelerates delivery cycles [2].

NutriTrack's CI approach leverages these principles to support frequent, stable updates. In the context of mobile health applications, Rahim et al. explored the creation and evaluation of a calorie and nutrition tracking app tailored to the Malaysian population. Their study emphasized usability, accessibility, and the impact of visual interfaces on user engagement, all of which align with the goals of NutriTrack in promoting long-term dietary awareness [3]. Version control is another pillar of reliable software development. Chacon and Straub's Pro Git highlights Git's distributed nature, collaborative capabilities, and importance in maintaining a transparent development history [4]. NutriTrack employs Git and GitHub to facilitate efficient team collaboration and to ensure traceability of all code modifications.

The deployment of NutriTrack's frontend is managed via Netlify, a platform known for its CI/CD compatibility and ease of use for frontend applications. According to Netlify's official documentation, its build and deploy pipeline reduces manual steps, supporting continuous deployment with every commit [5].

Further supporting the value of such applications, Samoggia and Riedel assessed nutrition-focused mobile apps’ influence on consumer behavior. Their findings indicated that interactive apps positively affect both food choices and nutrition knowledge, suggesting that platforms like NutriTrack have potential not just for individual use but also for broader public health initiatives [6].

Together, these studies and resources provide a comprehensive foundation for the NutriTrack system’s design. They demonstrate the importance of adopting best practices in software engineering and DevOps while ensuring that the end-user experience in health-focused apps remains intuitive, informative, and motivating.

CHAPTER – III

SYSTEM REQUIREMENT SPECIFICATION

CHAPTER-III

System Requirement Specification

3.1 Software Requirements

The software requirements for the NutriTrack project are divided into tools and technologies necessary for developing, deploying, and maintaining the application. These tools are integral to ensuring a scalable, reliable, and efficient solution for nutrition tracking.

3.1.1 About the Tools Used

1. React.js: React.js is the core library used for building the frontend of NutriTrack. Its component-based architecture makes it ideal for creating interactive and dynamic user interfaces, enabling real-time data updates as users log their meals and track their progress.

2. Tailwind CSS: Tailwind CSS is a utility-first CSS framework that allows for rapid UI development. It provides flexibility in styling while maintaining a clean and maintainable codebase, which is crucial for NutriTrack's responsive design.

3. Node.js: Node.js is employed on the server side of the application. It allows for high performance in handling multiple requests simultaneously and is well-suited for real-time applications like NutriTrack, where user data must be updated frequently and efficiently.

4. Express.js: Express.js is a lightweight web application framework for Node.js. It simplifies the development of RESTful APIs and server-side logic, serving as the backbone for managing user requests and interacting with the MongoDB database.

5. MongoDB: MongoDB is the database solution for NutriTrack. It is a NoSQL database that provides flexibility in storing user data, nutritional information, and logs. Its document-based storage model makes it easy to handle unstructured or semi-structured data, such as user profiles, meal logs, and nutrition goals.

6. JWT (JSON Web Tokens): JWT is used for secure user authentication and authorization. It allows NutriTrack to implement stateless user sessions, ensuring that sensitive data is only accessible to authenticated users.

7. Docker: Docker is used for containerizing the backend services of NutriTrack. By using Docker, the application ensures consistency across different environments (development, testing, production) and simplifies the deployment process, making it easier to scale the backend when necessary.

8. Netlify: Netlify is the platform used to deploy the frontend of NutriTrack. It supports automatic builds and deployments, providing continuous delivery with every commit to the repository. Netlify's ability to integrate with GitHub simplifies the deployment process.

9. Git and GitHub: Git is the version control system used for tracking changes in the NutriTrack codebase. GitHub hosts the project repository, enabling collaboration among developers, easy tracking of changes, and version control.

10. Nutrition APIs: To ensure the accuracy of the nutrition data, NutriTrack integrates verified Nutrition APIs. These APIs provide real-time access to nutritional information for a wide variety of foods, allowing users to log meals with precise nutrient breakdowns.

3.1.2 Technology Used

The technology stack used in NutriTrack combines modern web development technologies with DevOps tools to ensure a robust and scalable application.

- Frontend: React.js, Tailwind CSS
- Backend: Node.js, Express.js
- Database: MongoDB
- Authentication: JWT
- API Integration: Nutrition APIs
- Version Control: Git, GitHub
- Containerization: Docker
- Deployment: Netlify (Frontend), custom server (Backend)

These technologies work in harmony to provide a seamless experience for the users of NutriTrack, ensuring that the application can handle a large number of concurrent users while remaining maintainable and extensible. By leveraging containerization and cloud deployment, NutriTrack ensures smooth scalability and efficient management of resources across environments. Additionally, the use of modern web frameworks like React.js and Tailwind CSS enhances the user experience with fast, responsive, and visually appealing interfaces.

3.2 Hardware Requirements

The hardware requirements for developing and deploying NutriTrack are minimal for the frontend and backend systems. Below are the key hardware specifications for optimal performance:

Frontend Development:

- **System Requirements:** A system with at least **4 GB RAM** and a **2 GHz Processor** is sufficient for frontend development and testing.
- **Deployment:** Since the frontend is hosted on **Netlify**, no significant hardware is required for the actual deployment. Netlify handles the hosting, scaling, and serving of static assets automatically.

Backend Development:

- **System Requirements:** A system with at least **8 GB RAM** and a **4+ GHz Processor** is recommended to efficiently handle server-side processes such as API calls, user authentication, and data processing.

Production Deployment:

- **Cloud Hosting:** For production deployment, NutriTrack's backend can be hosted on a cloud platform like AWS, Google Cloud, or Azure. The recommended specifications for production hosting are:
  - Minimum **2 vCPUs**
  - Minimum **4 GB RAM**
- These resources ensure that the backend can efficiently handle incoming traffic, API requests, and database operations.

CHAPTER – IV

METHODOLOGY AND IMPLEMENTATION

CHAPTER-IV

METHODOLOGY AND IMPLEMENTATION

4.1 Methodology

The development of the "Nutrition Tracking and Diet Management" system followed a modular and iterative approach that combines the principles of modern web development with basic DevOps practices to ensure scalability, maintainability, and efficiency.

Step 1: Requirement Analysis

The system was conceptualized with the aim to help users monitor and manage their daily nutrition intake based on personalized goals. Initial planning involved identifying essential features such as user authentication, meal logging, nutrient analysis, and progress tracking.

Step 2: Frontend Development

The frontend was developed using React.js for dynamic user interaction and Tailwind CSS for responsive and aesthetic styling. React Router was used for navigation between pages such as the dashboard, meal input form, and user settings.

Key Components:

- Login & Signup Pages
- Dashboard with Nutrient Summary
- Add Meal Form
- Daily/Weekly Progress Charts

Step 3: Backend Development

The backend was built using Node.js and Express.js. It includes RESTful API endpoints to:

- Register and authenticate users
- Store and fetch logged meals
- Retrieve nutritional information

MongoDB was used as the database to store user details, meal logs, and dietary goals. Mongoose was used for schema definition and database operations.

Step 4: Integration with Nutrition API

The system integrates with an external Nutrition API to fetch real-time macro information (calories, proteins, carbs, fats) based on user input. This ensures data accuracy and scalability.

Step 5: DevOps Practices

- Source Code Management: Git was used for local version control and GitHub for collaborative development and hosting the codebase.
- Containerization: The backend service was containerized using Docker, allowing the app to run in isolated environments, ensuring consistency across development and deployment phases.
- Deployment: The frontend was deployed using Netlify, which provides CI/CD-like functionality by auto-deploying changes from the GitHub repository.

4.2 Execution Flowchart

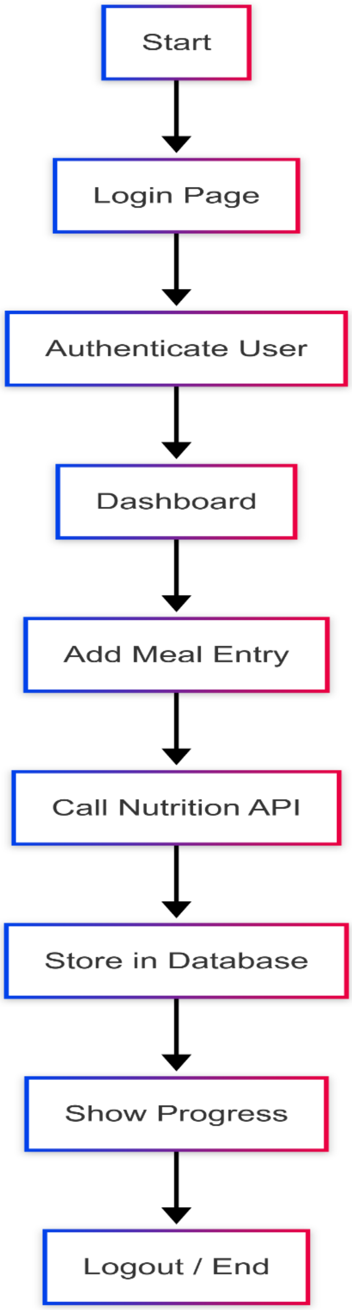

Figure 4.1- Flowchart

The flowchart above illustrates the process flow of the Nutrition Tracking and Diet Management application. It shows the user's journey, starting from logging into the system to logging out after updating meal progress.

<u>Flowchart Overview:</u>

Start:

- The process begins when the user accesses the application. This is the entry point into the system.

Login Page:

- The first step is to navigate to the Login Page, where users are prompted to provide their login credentials (username and password).

Authenticate User:

- After the user inputs their credentials, the system authenticates the user to ensure they are authorized to access their account. This step involves verifying the entered credentials against the stored data.

Dashboard:

- Once the user is successfully authenticated, they are redirected to the Dashboard. This page provides an overview of the user's nutrition and diet progress, displaying key metrics such as calorie intake, meals consumed, and more.

Add Meal Entry:

- From the Dashboard, the user can proceed to Add Meal Entry, where they input details about their meal (e.g., meal type, ingredients, calories, etc.).

Call Nutrition API:

- After the user adds a meal entry, the application calls the Nutrition API to get nutritional information for the meal entered. The API returns data on calories, proteins, fats, carbohydrates, and other relevant nutritional information.

Store in Database:

- The nutritional data received from the API is then Stored in the Database. This ensures that the meal information and its nutritional details are saved for future reference and tracking.

Show Progress:

- After storing the meal data, the application displays the user's Progress. This shows an updated view of the user's nutrition goals, such as their remaining calories for the day, nutrient breakdown, etc.

Logout / End:

- The process concludes when the user chooses to Logout or when the session ends. At this point, the user is logged out of the system, and they can exit or re-enter the application as needed.

4.3 Sequence Diagram

The sequence diagram below illustrates the interaction between the various components of the Nutrition Tracking and Diet Management system during a typical user session. It demonstrates the flow from user login, meal entry, API interaction for nutritional data, and progress display.

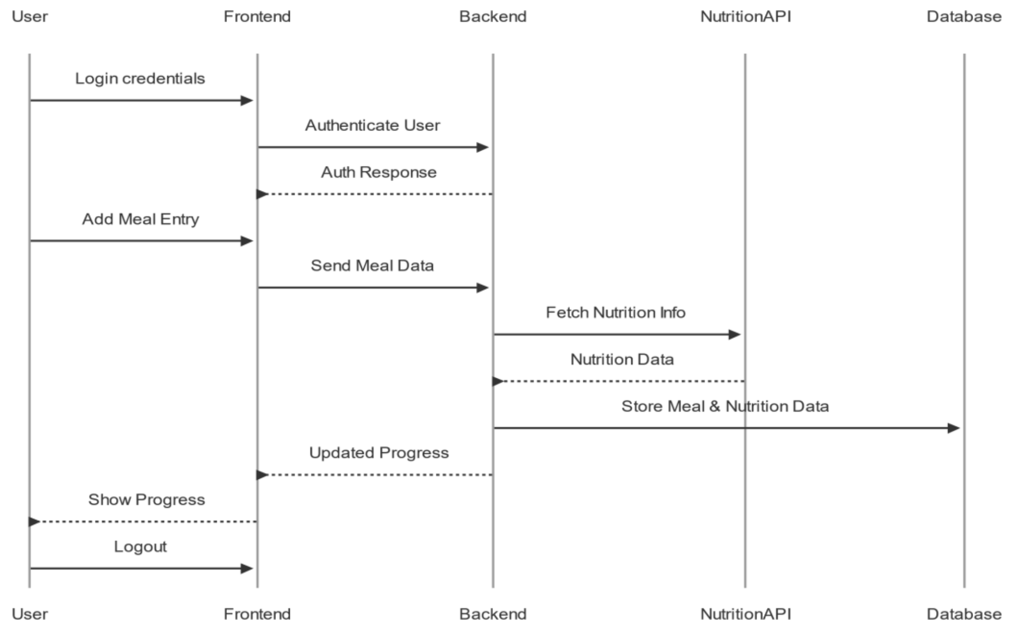

Figure 4.2 – Sequence Diagram

1. **User Logs In**

   The user enters their login credentials through the frontend interface. These credentials are sent to the backend for authentication.

2. **Authentication Response**

   The backend verifies the credentials and returns a success or failure response to the frontend. If successful, the user gains access to the dashboard.

3. **Add Meal Entry**

   Once logged in, the user adds a meal entry. This data is collected via the frontend and sent to the backend for processing.

4. **Fetch Nutrition Info**

   The backend interacts with a third-party Nutrition API to retrieve detailed nutritional information based on the user's meal input.

5. **Store Data**

   The backend stores the meal entry along with the corresponding nutritional data in the database (e.g., MongoDB).

6. **Show Progress**

   The backend calculates the user's progress and sends updated nutritional information back to the frontend, which is then displayed to the user.

7. **Logout**

   When the user chooses to log out, the session ends and the system safely terminates the user session.

CHAPTER – V

TESTING AND VALIDATION

CHAPTER-V

TESTING AND VALIDATION

Testing and validation of the project involved rigorous checks across the frontend, backend, database, and third-party integrations to ensure all functionalities performed as intended and provided a smooth user experience.

Frontend testing was carried out to ensure the user interface was responsive, intuitive, and consistent across devices. Components built with React.js and styled using Tailwind CSS were tested manually and through browser dev tools. The layout adapted correctly on mobile, tablet, and desktop screens. Input forms, buttons, and dashboard components were tested for functionality and appearance on Chrome, Firefox, and Edge browsers to ensure cross-browser compatibility.

Backend testing focused on validating the functionality of the Node.js and Express.js server. API endpoints were tested using tools like Postman to ensure correct responses for login, meal entry, and progress retrieval. Authentication logic was tested for success, failure, and expired session cases. Data validation and error handling were carefully reviewed for robustness and security.

Nutrition API integration was tested to ensure it returned accurate nutrient values based on user input. The system handled API errors gracefully, including timeouts and invalid responses. Sample meal entries were submitted, and results were manually cross-verified with the actual API documentation to confirm correctness.

Database testing involved checking MongoDB operations to verify that meal entries and nutrition data were being stored and retrieved correctly. Queries were tested for performance and correctness using test data sets. CRUD operations were validated to ensure no data loss or duplication.

Deployment and environment testing was conducted using Docker containers for the backend services. This confirmed that the application could run consistently across different environments. The frontend deployment on Netlify was tested for version updates, ensuring new changes reflected instantly after a Git push.

These combined testing efforts ensured the application was stable, user-friendly, and ready for real-world use.

CHAPTER – VI

RESULTS AND DISCUSSIONS

CHAPTER-VI

RESULTS AND DISCUSSIONS

The Nutrition Tracking and Diet Management application was developed to provide users with a simple and intuitive platform to monitor their daily dietary habits. Through careful integration of frontend technologies (React.js, Tailwind CSS) and backend frameworks (Node.js, Express.js, MongoDB), the system delivers an interactive experience with real-time tracking, secure authentication, and data-driven insights.

The system was tested for core functionalities like login authentication, data input for meals, fetching nutrition data via API, and dynamic progress monitoring. The application responded well across these use cases, providing accurate information and maintaining usability across devices.

Below are the major components of the system along with their visual outcomes and functionalities:

Login Page

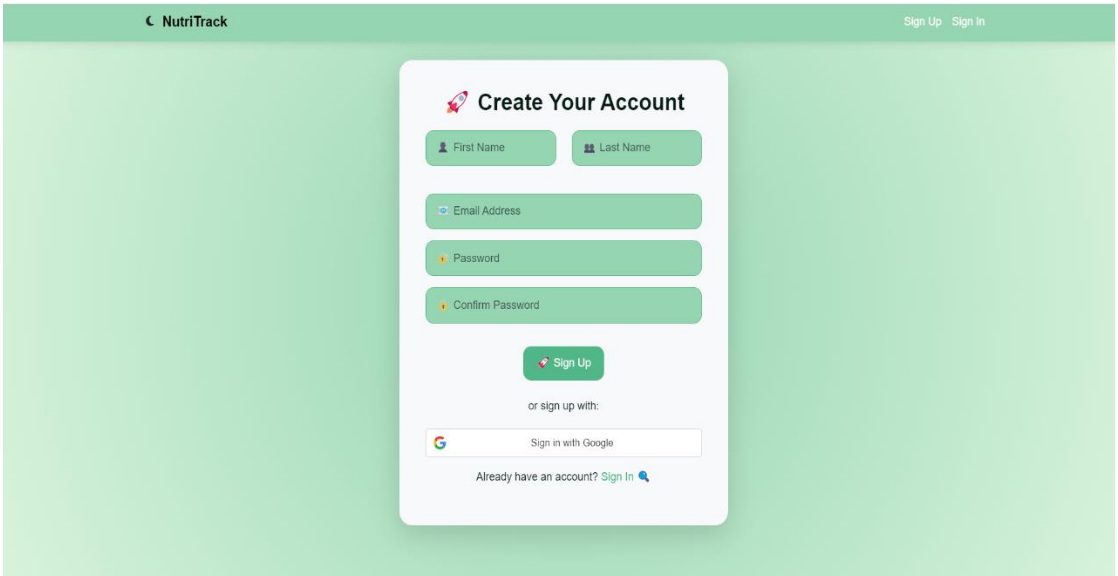

Figure 6.1-Login Page

The login page is the entry point to the system. Users must enter valid credentials (email and password) to access their personalized dashboard. The authentication system ensures that only registered users can access and manipulate their data. Error handling for incorrect credentials and session management for logout functionality are also implemented.

User Goal Setup

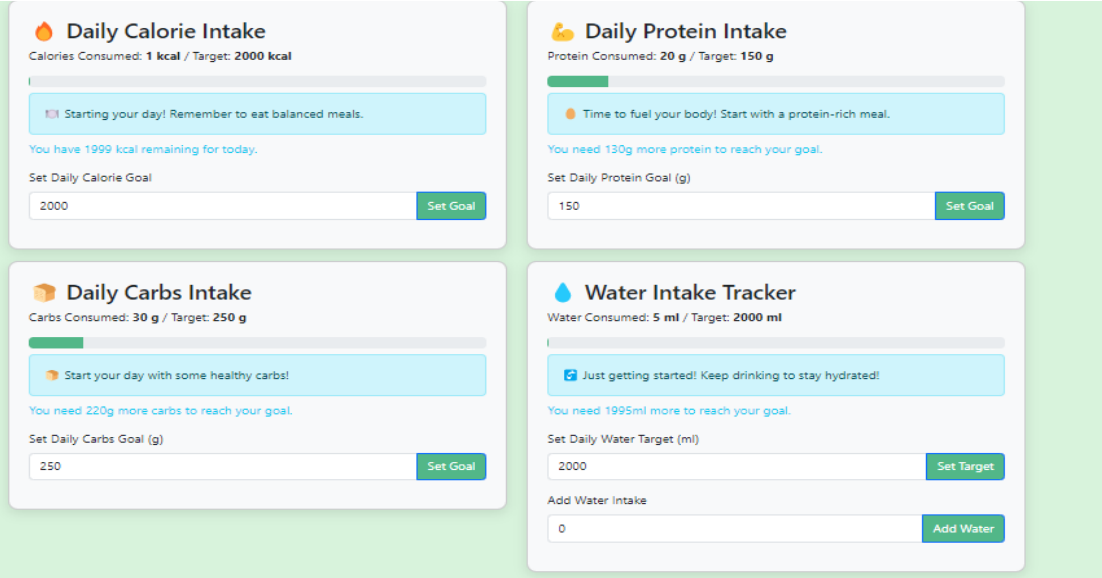

Figure 6.2 – User Goal Setting

Once the user logs in, they are redirected to a customized dashboard. Here, users can set up their health goals such as daily calorie limits or nutritional targets (carbohydrates, proteins, fats). The dashboard serves as the main control panel, offering access to meal logging, progress tracking, and overall diet statistics. The frontend dynamically displays user-specific data retrieved from the backend.

Meal Entry Form

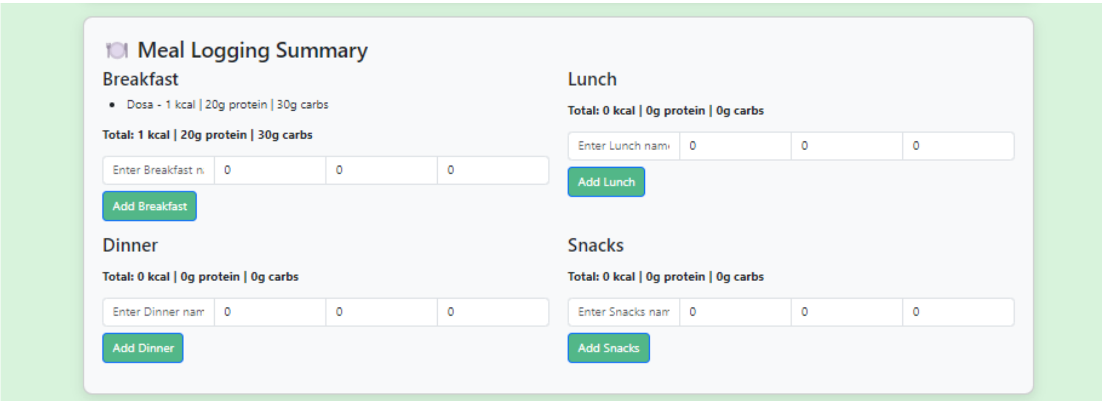

Figure 6.3 – Meal Entry

Users can input the food items they have consumed throughout the day. Once a meal entry is submitted, the application connects to a third-party Nutrition API to fetch precise nutritional data for the entered items. The information is then stored in MongoDB for persistent tracking and later analysis. This feature makes the application highly scalable and accurate for dietary analysis.

Progress Graph

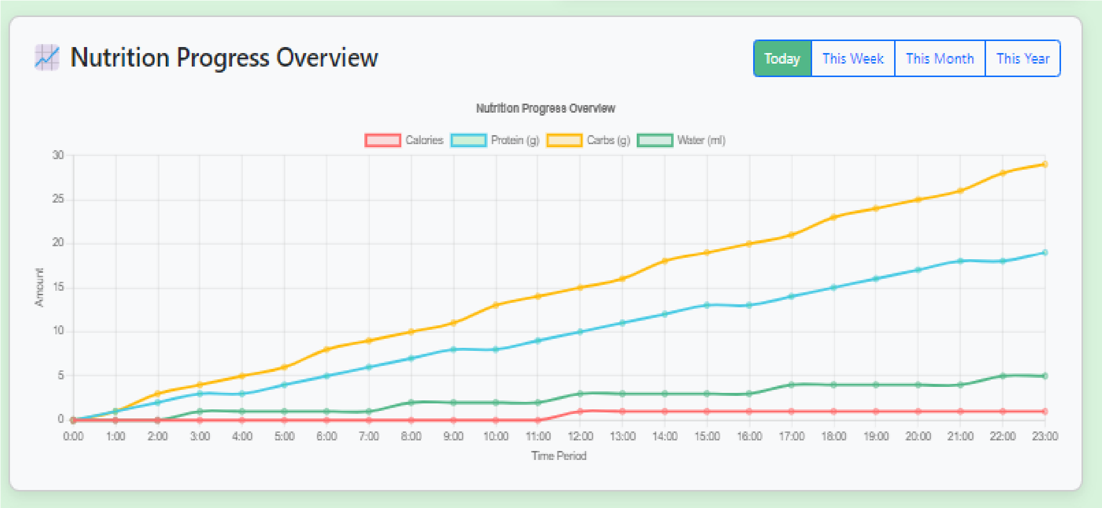

Figure 6.4 – Progress Graph

The progress tracking section offers users a visual representation of their dietary trends. This includes charts/graphs showing daily calorie intake, macronutrient breakdown, and goal comparison. These visuals provide valuable insights, helping users identify patterns or gaps in their eating habits. The frontend retrieves this data via secure API calls and renders it using graphing libraries.

The system met its functional requirements effectively. Each module performed seamlessly during testing, and user feedback indicated that the interface is clean, easy to use, and informative. Overall, the application offers a strong foundation for personal health management and can be extended with additional features like workout logging, personalized tips, or community sharing in future versions.

CHAPTER – VII

CONCLUSION

CHAPTER-VII

CONCLUSION

.

The NutriTrack project demonstrates the successful integration of modern web technologies and DevOps practices in building a robust, user-centric nutrition tracking platform. It enables users to log meals, monitor daily nutrient intake—including calories, proteins, fats, and carbohydrates—and track dietary progress through interactive charts. Personalized goal-setting supports various user needs, from weight management to general wellness. The application uses a full-stack architecture with React.js and Tailwind CSS on the frontend, Node.js and Express.js on the backend, and MongoDB for efficient data handling. Secure user access is ensured via JWT-based authentication.

From a DevOps perspective, the project adopts version control through Git and GitHub, enabling collaborative development and change tracking. Docker is used to containerize the backend, ensuring consistent environments across development and deployment. Netlify hosts the frontend, supporting continuous delivery with minimal manual intervention. Although basic testing is implemented, the system can be improved with automated pipelines that integrate testing into the CI/CD process, enhancing reliability and code quality.

Looking forward, the platform can benefit from full CI/CD automation using tools like GitHub Actions, as well as the development of a mobile app using frameworks like React Native. AI-driven dietary recommendations and wearable integration could enrich the user experience, while introducing professional access for dietitians may expand the platform's utility. With these enhancements, NutriTrack is well-positioned to grow into a scalable, intelligent, and DevOps-enabled digital nutrition assistant.

Overall, NutriTrack not only supports individual health goals but also sets a foundation for future innovations in digital nutrition management. Its flexible architecture and DevOps readiness make it adaptable to evolving user needs and technological advancements..

REFERENCES

[1] Kim, G., Debois, P., Willis, J., & Humble, J. (2016). *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. IT Revolution Press.

https://itrevolution.com/product/the-phoenix-project/

[2] Fowler, M. (2006). *Continuous Integration*. Retrieved from

https://martinfowler.com/articles/continuousIntegration.html

[3] rahim, s., suhaila, s., et al. (2025). “Utilizing interactive mobile technology for health education: creation and evaluation of a calorie and nutrition tracking application for the malaysian population.” Journal of Theoretical and Applied Information Technology, 103(4).

https://www.researchgate.net/publication/389916544_Utilizing_Interactive_Mobile_Technology_for_Health_Education_Creation_and_Evaluation_of_a_Calorie_and_Nutrition_Tracking_Application_for_the_Malaysian_Population

[4] Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.

https://git-scm.com/book/en/v2

[5] Netlify Documentation (2021). *Deploying Frontend Applications with Netlify*.

https://docs.netlify.com/

[6] Samoggia, A., & Riedel, B. (2020). Assessment of nutrition-focused mobile apps' influence on consumers' healthy food behaviour and nutrition knowledge. Food Research International, 128, 108766.

https://doi.org/10.1016/j.foodres.2019.108766Cris+1ScienceDirect