

COURSE TITLE: SOFTWARE ENGINEERING

COURSE CODE: CSC 216

END OF COURSE CAPSTONE PROJECT

GROUP 16 MEMBERS

|  |  |  |
| --- | --- | --- |
| S/N | NAMES | MATRICULATION NUMBER |
| 1 | Nwankwoala Kelechi | VUG/CSC/22/6834 |
| 2 | Nwankwo Stephine | VUG/CSC/22/ |
| 3 | Alademerin Salma Ore-Oluwa | VUG/CSC/22/7422 |
| 4 | Osagie Daniel Ehirehmen | VUG/CSC/22/7545 |
| 5 | Ajayi Iyanuoluwa Abdulkareem | VUG/CSC/22/6935 |
| 6 | Uhiara Chigozie Patrick | VUG/CSC/22/7517 |
| 7 | Jeol Barnabas | VUG/CSC/22/7602 |
| 8 | Ekele David Oche | VUG/CSC/22/7336 |
| 9 | Oba Abdulkadir | VUG/CSC/22/7335 |
| 10 | Okoh Vincent | VUG/CSC/22/7459 |
| 11 | BAMAS NIARIG EBENEZER | VUG/CSC/22/7560 |

Project Document: Smart Health Assistant

Project Title: Fitness-rings

**Project Description:**

The fitness-rings is a web application that helps users keep track of their health goals and provides personalized recommendations to improve their overall well-being. The application allows users to create a profile and set up goals such as weight loss, increasing physical activity, or reducing stress levels. Users can input data about their desired exercise. The application will use this information to provide customized recommendations, such as healthy exercise routines.

**Project Background:**

The Smart Health Assistant project aims to address the growing need for an innovative solution that helps individuals take control of their health and well-being. With the rise of sedentary lifestyles and an increase in chronic diseases, there is a pressing need for accessible tools that empower individuals to make informed choices about their health.

Traditional approaches to health management often lack personalized guidance and real-time feedback, leading to limited success in achieving health goals. The Smart Health Assistant seeks to bridge this gap by leveraging the power of technology to provide users with personalized recommendations and support.

The project recognizes the importance of goal setting and tracking in driving positive behavioral changes. By allowing users to set health goals tailored to their specific needs, such as weight loss, increased physical activity, or stress reduction, the Smart Health Assistant empowers individuals to take proactive steps toward a healthier lifestyle.

Users can easily log their exercise routines through data input functionalities, enabling the application to analyze and interpret this information. The Smart Health Assistant generates meaningful insights by harnessing this data and presents users with personalized recommendations, including tailored meal plans, exercise routines, and relaxation techniques. These recommendations are designed to align with the user's goals and provide an actionable roadmap toward improved health and well-being.

To ensure a seamless user experience, the project incorporates user registration and authentication features, guaranteeing privacy and security. The application also integrates a chatbot, enhancing user engagement and providing an interactive platform for addressing queries and concerns.

From a technical perspective, the project utilizes modern web development frameworks, such as React or Angular for the front end and Node.js for the back end. The integration of a reliable database solution, such as Firebase, ensures efficient storage and retrieval of user data. Hosting the application on Vercel guarantees scalability, performance, and accessibility for users.

The Smart Health Assistant project represents a valuable opportunity for the student to apply their software engineering, web development, and database design skills. It also encompasses essential aspects of project management, teamwork, and communication, as the successful completion of the project requires effective coordination and collaboration among team members.

Ultimately, the Smart Health Assistant aims to empower individuals in taking charge of their health, providing them with the necessary tools, guidance, and support to make meaningful and sustainable lifestyle changes. By combining technology, personalized recommendations, and user engagement, the project endeavors to improve overall well-being and contribute to a healthier society.

**Project Objectives:**

Functional Requirements:

a. User Registration and Authentication:

- Users should be able to create an account securely with unique credentials.

- The application should provide authentication mechanisms to ensure user privacy and data security.

b. Goal Setting and Tracking:

- Users should be able to set health goals, such as weight loss, and desired physique.

- The application should allow users to track their progress toward their goals and provide visual representations of their achievements.

c. Data Input:

- Users should be able to input data about their desired exercise

- The application should provide an intuitive and user-friendly interface for data input.

d. Data Visualization:

- The application should display graphs and charts to help users visualize their progress toward their health goals.

- The visualization should be clear, and interactive, and provide meaningful insights to users.

e. Personalized Recommendations:

- Based on the data input by the users, the application should provide personalized recommendations, such as healthy meal plans, exercise routines, and relaxation techniques.

- Recommendations should be tailored to each user's specific health goals and preferences.

Non-Functional Requirements:

a. Technology Stack:

- The web application should be developed using a modern web framework like React for the front end and Next.js for the back end.

- The backend should utilize a database system such as MySQL or any of your choice for data storage.

b. Security and Privacy:

- The application should implement robust security measures to protect user data and ensure privacy.

- User data should be encrypted during transmission and storage.

c. User Interface and User Experience:

- The user interface should be intuitive, visually appealing, and responsive across different devices and screen sizes.

- The application should provide clear instructions, guidance, and feedback to users for seamless navigation and interaction.

d. Performance:

- The application should have fast response times, minimal latency, and handle concurrent user requests efficiently.

- Performance optimizations should be implemented

**Our project incorporates the following features:**

**a. User registration and authentication:**

Users can create an account by providing necessary details such as username, email, and password. Authentication mechanisms are implemented to ensure secure access to the application and personalized user experiences.

**b. Goal setting and progress tracking:**

Users can set fitness goals, such as weight loss or muscle gain, and track their progress over time. They can input relevant metrics, such as weight, body measurements, or exercise performance, to monitor their achievements.

**c. Data input for desired exercise:**

Users can log their exercise activities, including the type of exercise, duration, intensity, and any additional notes. This allows them to keep a record of their workouts and monitor their fitness routines.

**d. Data visualization through graphs and charts:**

The application presents visual representations of the user's progress and exercise data using graphs and charts. This helps users analyze their performance, identify trends, and make informed decisions regarding their fitness journey.

**e. Personalized recommendations based on user data:**

Based on the user's goals, preferences, and historical data, the application provides personalized recommendations. These recommendations may include workout routines, nutrition plans, exercise variations, or lifestyle tips tailored to the user's specific needs and objectives.

**f. Integration of a Chatbot:**

The project incorporates a chatbot feature that allows users to interact with an AI-powered assistant. The chatbot can provide instant support, answer frequently asked questions, offer guidance on exercise techniques, and provide general fitness-related information.

These features collectively create a comprehensive fitness application that empowers users to track their progress, set goals, visualize their data, receive personalized recommendations, and access support through the integrated chatbot.

**Technical Requirements:**

In our project, we utilized the following technical requirements:

* Front-end: React

React is a popular JavaScript library for building user interfaces. We used React for the front-end development of our project. React allowed us to create reusable UI components, manage state efficiently, and implement a responsive and interactive user interface.

* Back-end: Next.js

Next.js is a framework built on top of React that enables server-side rendering, routing, and other features for building web applications. We chose Next.js as our back-end framework to enhance the performance and SEO-friendliness of our project. With Next.js, we were able to pre-render pages on the server and optimize the loading speed of our application.

* Database: Firebase

Firebase is a cloud-based platform provided by Google that offers various services, including a NoSQL database. We utilized Firebase as our database solution. Firebase's real-time database allowed us to store and synchronize data in real-time across multiple clients. It provided us with a scalable and reliable storage solution for our application.

* Hosting: Vercel

Vercel is a cloud platform specifically designed for hosting static websites and serverless functions. We chose Vercel as our hosting provider to deploy our React and Next.js application. Vercel provided us with an easy deployment process, seamless integration with our Git repository, and automatic scaling to handle high traffic loads.

* Authentication: Firebase

Firebase also offers authentication services that we utilized for user authentication in our project. Firebase Authentication provided secure and easy-to-implement authentication methods, such as email/password authentication, social logins (e.g., Google, Facebook), and more. It allowed us to authenticate users and manage their access to certain features or content within our application.

By leveraging React for the front end, Next.js for the back end, Firebase for the database and authentication, and hosting our application on Vercel, we were able to build a scalable, performant, and secure web application that meets our project requirements.

**Project Milestones:**

**a. Project planning and requirements gathering:**

- Conduct initial project kickoff meeting to define project goals, objectives, and stakeholders.

- Identify and document project requirements, including functional and non-functional requirements.

- Conduct interviews and workshops with stakeholders to gather detailed project requirements.

- Perform feasibility analysis to determine the project's technical and financial viability.

- Create a project plan, including a timeline, resource allocation, and risk management strategy.

**b. User interface design and prototyping:**

- Conduct user research and analysis to understand user needs and preferences.

- Develop user personas and user journey maps to guide the design process.

- Create wireframes and low-fidelity mockups to visualize the application's layout and flow.

- Iterate on the design based on feedback from stakeholders and usability testing.

- Develop high-fidelity prototypes using design tools and interactive prototyping software.

**c. Backend development and database integration:**

- Set up the development environment for backend development using Next.js.

- Define the server architecture, including routes, middleware, and data models.

- Implement user registration and authentication functionality using Firebase Authentication.

- Integrate Firebase as the backend database, ensuring proper data storage and retrieval.

- Develop APIs and server-side logic to support the frontend features and data interactions.

- Implement data validation, security measures, and error handling in the backend.

**d. Frontend development and integration with the backend:**

- Set up the development environment for frontend development using React.

- Develop the UI components and layouts based on the finalized design.

- Implement frontend functionality for user registration, goal setting, and progress tracking.

- Integrate frontend components with the backend APIs to enable data exchange.

- Implement data visualization using libraries or custom components for graphs and charts.

- Implement personalized recommendation algorithms based on user data analysis.

**e. Testing and debugging:**

- Conduct unit testing to ensure the correctness of individual components and functions.

- Perform integration testing to verify the proper interaction between frontend and backend.

- Conduct system testing to evaluate the application's overall functionality and performance.

- Perform usability testing to gather feedback on the user interface and user experience.

- Debug and fix any issues or bugs identified during testing.

- Conduct user acceptance testing with a group of target users to validate the application.

**f. Deployment and maintenance:**

- Set up hosting on Vercel or other preferred hosting platform.

- Deploy the application to the hosting environment for public access.

- Configure domain settings and SSL certificates, if applicable.

- Set up monitoring and error tracking tools to monitor application performance.

- Plan for future maintenance and updates, including bug fixes and feature enhancements.

- Provide user support and address any issues or inquiries post-deployment.

These expanded milestones provide a more detailed overview of the project's progression, ensuring thorough planning, development, and testing of the application.

**Task Allocation:**

**- Project Manager: [Nwankwoala Kelechi]**

- Oversee the entire project.

- Coordinate team members and monitor progress.

- Ensure project objectives are met.

- Handle communication with stakeholders.

**- Front-end Developer (1): [Osagie Daniel Ehirehmen ]**

- Develop the user interface using React or Angular.

- Implement responsive design and user interactions.

- Integrate UI components with the back end.

**- Back-end Developer (1): [Uhiara Chigozie Patrick , Joel Barnabas]**

- Set up the Next.js server.

- Design and develop the API endpoints for user authentication and registration.

- Implement data storage and retrieval from the database.

**- Back-end Developer (2): [Osagie Daniel Ehirehmen ]**

- Develop the goal-setting and progress-tracking features.

- Handle the logic for storing and updating user goals and progress data.

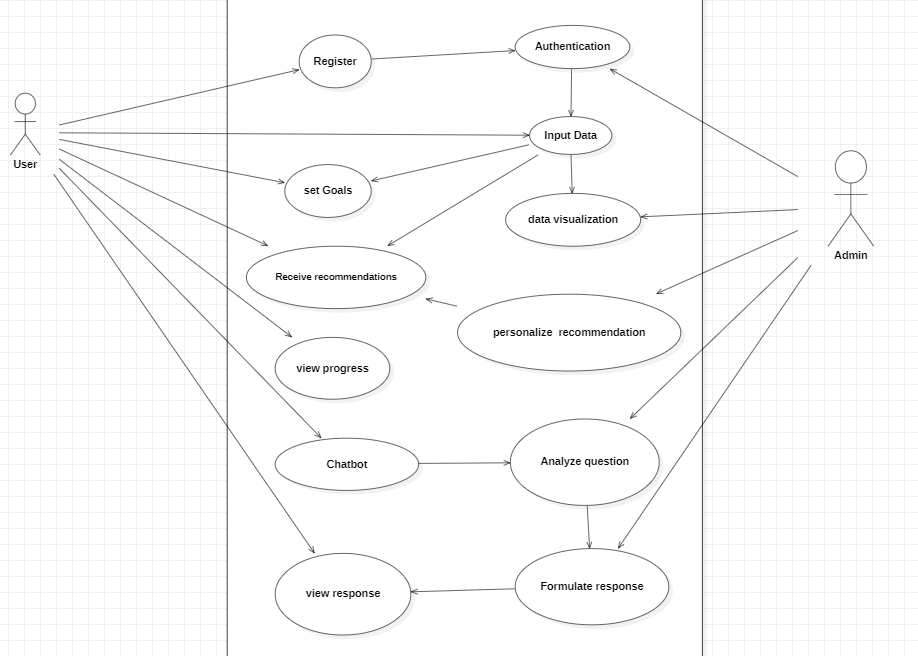
- Collaborate with the front-end developers for seamless integration.

- **Data Analyst: [Nwankwo Stephine, Abdul-malik ,** **Alademerin Salma Ore-Oluwa ]**

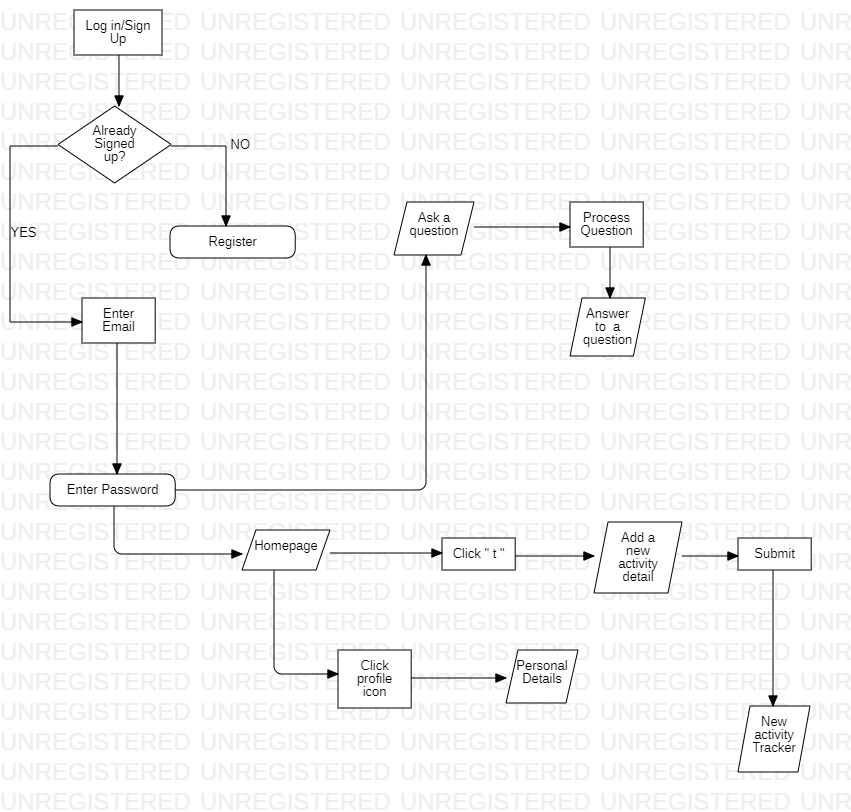
- Perform data flow analysis and develop algorithms for personalized recommendations.

- Analyse user input data to generate tailored recommendations.

- Collaborate with the back-end developers for data integration.



**Use case Diagram**



**Flow Chart Diagram**

**- Quality Assurance Tester: [**Ekele David OcheOba Abdulkadir**]**

- Develop test cases and perform thorough testing of the application.

- Identify and report bugs or issues.

- Ensure the application meets the specified requirements and functions

correctly.

-These are some issues that were identified along the way;

1. Integrating a chatbot:

Yes, integrating a chatbot into a web application can be convenient as it provides a way for users to interact and conversationally get assistance or information. Chatbots can handle common queries, provide guidance, and enhance the overall user experience. It's a useful feature to consider implementing.

2. Adding daily meals and providing a proper diet routine:

This sounds like a valuable feature for a web application, especially if it is related to health, nutrition, or meal planning. Allowing users to input their daily meals and receiving a proper diet routine or suggestions can be beneficial for those seeking to maintain a healthy lifestyle. It may require integration with a nutrition database or algorithm to generate personalized recommendations based on the user's input.

3. Interface stretched out on a laptop or wide screens:

The interface of the web application appears stretched out on larger screens, To ensure a consistent and optimal user experience across different screen sizes, you can implement responsive web design techniques. This involves using CSS media queries and flexible layouts to adapt the interface to various screen sizes, including laptops and wide screens.

But due to limited time, 2 and 3 couldn't be fixed so it's only the chatbot that was added

**- Technical Writer: [Nwankwoala Kelechi, Okoh Vincent]**

- Create user manuals, installation guides, and technical documentation.

- Document the system architecture, APIs, and database structure.

- Assist with any necessary documentation throughout the project.

**Project Outcome:**

The Smart Health Assistant will help users to achieve their health goals by providing personalized recommendations and support. The project will demonstrate the team's skills in software engineering, web development, and database design. The project will also provide valuable experience in project management, teamwork, and communication.

**Project Approval:**

Mr. Thomas Baidoo