**GROUP 15’S HEALTH WEBSITE DOCUMENTATION**

Requirements Analyst/ Documenter: Caleb taiwo toluwase vug/csc/22/6904

Programmer/ Designer: Duke, Odo Orok Vug/csc/22/7476

# Health Website Documentation

## Table of Contents

1. Introduction

- 1.1 Purpose

- 1.2 Scope

- 1.3 Intended Audience

2. System Overview

- 2.1 Functionality

- 2.2 Development Life Cycle

- 2.3 Technologies Used

3. Installation and Setup

- 3.1 System Requirements

- 3.2 Installation Process

4. User Guide

- 4.1 Fitness Tracker

- 4.2 Health Chatbot

- 4.3 Community Tab

- 4.4 Logging Page

- 4.5 Password Page

- 4.6 Profile Settings

5. Developer Guide

- 5.1 Architecture Overview

- 5.2 Folder Structure

- 5.3 Database Schema

- 5.4 API Documentation

- 5.5 Deployment Process

6. Troubleshooting

7. Frequently Asked Questions (FAQ)

8. Glossary

9. References

## 1. Introduction

### 1.1 Purpose

The purpose of this documentation is to provide a comprehensive guide for the health website, describing its features, installation process, usage instructions, and development details. It aims to assist users and developers in understanding and utilizing the functionalities offered by the website.

### 1.2 Scope

This documentation covers the features and functionalities of the health website, including the fitness tracker, health chatbot, community tab, logging page, password page, and profile settings. It does not include information on specific medical advice or treatment.

### 1.3 Intended Audience

This documentation is intended for both end-users and developers. End-users will find instructions on how to use the various features of the website, while developers will gain insights into the system architecture, database schema, and deployment process.

## 2. System Overview

### 2.1 Functionality

The health website provides the following key functionalities:

1. Fitness Tracker: Allows users to track their fitness activities, such as steps taken, calories burned, and distance covered.

2. Health Chatbot: Provides a chatbot interface to answer common health-related questions and provide basic health advice.

3. Community Tab: Enables users to communicate with each other through a chat interface, fostering a supportive community.

4. Login Page: Allows users to log their daily activities, including exercise routines, meals, and overall health status.

5. Password Page: Provides a secure password management system, allowing users to update their passwords.

6. Profile Settings: Allows users to customize their profile information, such as name, profile picture, and notification preferences.

### 2.2 Development Life Cycle

The development of the health website follows an iterative and incremental development life cycle, which includes the following stages:

1. Requirements Gathering: Collecting and documenting the functional and non-functional requirements of the website.

2. Design: Creating a high-level architecture and user interface (UI) design based on the requirements.

3. Development: Implementing the back-end logic, front-end components, and database integration.

4. Testing: Conducting unit tests, integration tests, and user acceptance tests to ensure the system works as expected.

5. Deployment: Deploying the website to a production environment for users to access.

6. Maintenance: Monitoring the website's performance, addressing issues, and continuously improving the system based on user feedback.

### 2.3 Technologies Used

The health website is developed using a combination of technologies and frameworks, including:

- Programming Languages: HTML, CSS, JavaScript,

- Frontend Framework: html, css and java

- Backend Framework: xxxxx

- Database: xxxxx

- Authentication:xxxxx

: JSON Web Tokens (JWT)

- External APIs: Health-related APIs for fitness tracking and chatbot functionality(what are we using guys)

## 3. Installation and Setup

### 3.1 System Requirements

Before installing the health website, ensure that your system meets the following requirements:

- Operating System: Windows, macOS, or Linux

- Web Browser: Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge

- Node.js (version X.X.X) and npm (version X.X.X)

- Python (version X.X.X) and pip (version X.X.X)

- PostgreSQL (version X.X.X) database server

### 3.2 Installation Process

To install and set up the health website, follow these steps:

1. Clone the project repository from the [GitHub repository](https://github.com/this is if we a posting it on github/repository).

2. Install the required dependencies using npm and pip.

3. Configure the PostgreSQL database by creating a new database and updating the connection settings in the configuration file.

4. Run the database migrations to set up the necessary tables and schema.

5. Start the back-end server and front-end development server.

6. Access the health website through your web browser.

## 4. User Guide

This section provides a detailed user guide for utilizing the various features of the health website.

### 4.1 Fitness Tracker

The fitness tracker feature allows users to track their fitness activities. To use the fitness tracker, follow these steps:

1. Log in to your account or create a new account if you don't have one.

2. Navigate to the fitness tracker section.

3. Connect your fitness tracker device or mobile app with the website.

4. View your daily activity metrics, such as steps taken, calories burned, and distance covered.

5. Set personal fitness goals and track your progress over time.

6. Analyze your historical fitness data using charts and graphs.

### 4.2 Health Chatbot

The health chatbot feature provides a chat interface for users to ask health-related questions and receive basic health advice. To use the health chatbot, follow these steps:

1. Access the health chatbot section of the website.

2. Enter your health-related question or concern in the chat interface.

3. The chatbot will provide relevant responses and suggestions based on the entered query.

4. Engage in a conversation with the chatbot to gather more information or clarify your queries.

5. Follow the chatbot's recommendations, if applicable, or seek professional medical advice if necessary.

### 4.3 Community Tab

The community tab feature enables users to interact with each other through a chat interface, fostering a supportive community. To use the community tab, follow these steps:

1. Access the community tab section of the website.

2. Join existing chat rooms or create new chat rooms based on your interests or health concerns.

3. Participate in discussions by sending messages and sharing resources.

4. Interact with other users by responding to messages and offering support.

5. Follow community guidelines to ensure respectful and helpful communication.

### 4.4 Logging Page

The logging page allows users to log their daily activities, including exercise routines, meals, and overall health status. To use the logging page, follow these steps:

1. Access the logging page section of the website.

2. Choose the type of activity you want to log, such as exercise, meals, or general health.

3. Enter the relevant details for the selected activity, such as exercise duration, meal contents, or health symptoms.

4. Save the log entry to record your activity for future reference and analysis.

5. View and analyze your historical logs to track progress and identify patterns.

### 4.5 Password Page

The password page provides a secure password management system, allowing users to update their passwords. To manage your password, follow these

steps:

1. Navigate to the password page section of the website.

2. Enter your current password and the new password you want to set.

3. Confirm the new password to ensure accuracy.

4. Click the "Change Password" button to update your password.

5. Ensure you remember the new password and keep it secure.

### 4.6 Profile Settings

The profile settings feature allows users to customize their profile information, such as name, profile picture, and notification preferences. To modify your profile settings, follow these steps:

1. Access the profile settings section of the website.

2. Update your profile picture by uploading a new image or selecting one from the provided options.

3. Modify your name, contact information, and any other relevant details.

4. Customize your notification preferences, such as email notifications or push notifications.

5. Save the changes to update your profile information.

## 5. Developer Guide

This section provides guidance and documentation for developers interested in understanding the health website's technical details.

### 5.1 Architecture Overview

The health website follows a client-server architecture. The front-end is built using html and css, while the back-end utilizes xxxxx for API development and business logic. The database management system used is Firebase.

The architecture consists of the following components:

- Frontend Components: The user interface components responsible for rendering the website and handling user interactions.

- Backend APIs: RESTful APIs implemented using Flask that handle client requests and perform necessary business logic.

- Database: firebase database for storing user information, fitness data, chat logs, and other relevant data.

### 5.2 Folder Structure

The health website project follows a specific folder structure to organize the codebase efficiently. The structure may look as follows:

```

- back-end/

- api/ # Contains API endpoints and business logic

- models/ # Defines database models and schemas

- services/ # Handles external service integrations

- migrations/ # Database migration scripts

- config.py # Configuration file for the back-end

- front-end/

- public/ # Contains static assets and index.html

- src/ # Source code for front-end components and logic

- config.js # Configuration file for the front-end

- database/ # Scripts for database setup and migrations

- docs/ # Documentation files

- README.md # Project overview and instructions

- requirements.txt # Required dependencies for the project

```

### 5.3 Database Schema

The health website's database schema is designed to store user information, fitness data, chat logs, and other relevant data. Here is an example of a simplified schema:

```

- users

- id (primary key)

- name

- email

- password (hashed)

- fitness\_activities

- id (primary key)

- user\_id (foreign key referencing users.id)

- activity\_type

- duration

- calories\_burned

- distance

- timestamp

- chat\_logs

- id (primary key)

- user\_id (foreign key referencing users.id)

- message

- timestamp

- ...

```

### 5.4 API Documentation

The health website provides several APIs for frontend-backend communication. Detailed API documentation can be found in the API documentation file, which describes each endpoint, its purpose, and the expected request and response formats.

### 5.5 Deployment Process

To deploy the health website to a production environment, follow these general steps:

1. Set up a production server with the necessary system requirements, such as operating system, web server, and database server.

2. Configure the production server environment variables, including database connection details, API keys, and security configurations.

3. Build the front-end code

using a production build command, such as `npm run build`.

4. Configure the web server to serve the front-end build files and proxy back-end API requests to the appropriate back-end server.

5. Set up the database in the production environment and run any necessary database migrations.

6. Deploy the back-end code to the production server and start the back-end server.

7. Test the deployed website to ensure all functionalities work correctly in the production environment.

## 6. Troubleshooting

This section provides troubleshooting tips and solutions for common issues that users or developers may encounter while using or setting up the health website. It covers potential error messages, recommended solutions, and references to external resources for further assistance.

## 7. Frequently Asked Questions (FAQ)

This section answers frequently asked questions related to the health website, covering both user-oriented and technical questions. It provides concise and helpful responses to common queries.

## 8. Glossary

This section provides a glossary of terms and acronyms used throughout the documentation. It helps users and developers understand the specific terminology used in the context of the health website.

## 9. References

This section lists the references used in creating the health website documentation, including external resources, libraries, and frameworks. It serves as a credit and citation to the relevant sources.

Note: I will filling some thing and make some correction when I see the app