A

PROJECT REPORT

On

Health Tracker

Submitted by

Beauty Pankajbhai Bechara (IU2141230010)
Dharita Jamadagni Patel (IU2141230164)

In fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING

INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ

AHMEDABAD-382115, GUJARAT, INDIA,

WEB: www.indusuni.ac.in

APRIL 2025

A

PROJECT REPORT

On

Health Tracker

At

In the partial fulfillment of the requirement
for the degree of
Bachelor of Technology
in
Computer Science & Engineering

PREPARED BY

Beauty Pankajbhai Bechara (IU2141230010)
Dharita Jamadagni Patel (IU2141230164)

UNDER GUIDANCE OF

Internal Guide
Dr. Kaushal Jani
Associate Professor,
Department of Computer Science & Engineering,
I.I.T.E, Indus University, Ahmedabad

SUBMITTED TO

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING
INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ
AHMEDABAD-382115, GUJARAT, INDIA,
WEB: www.indusuni.ac.in
APRIL 2025

CANDIDATE'S DECLARATION

I declare that the final semester report entitled "**Health Tracker**" is my own work conducted under the supervision of the guide **Dr. Kaushal Jani**.

I further declare that to the best of my knowledge, the report for B.Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

_____

Candidate's Signature
Beauty Pankajbhai Bechara (IU2141230010)
Dharita Jamadagni Patel (IU2141230164)

_____

Guide: Dr. Kaushal Jani
Associate Professor,
Department of Computer Science & Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING
COMPUTER SCIENCE & ENGINEERING
2024 -2025

CERTIFICATE

Date: 22-04-2025

This is to certify that the project work entitled "**Health Tracker"** has been carried out by **Beauty Pankajbhai Bechara** under my guidance in partial fulfillment of degree of Bachelor of Technology in **COMPUTER SCIENCE & ENGINEERING (Final Year)** of Indus University, Ahmedabad during the academic year 2024 - 2025

Dr. Kaushal Jani Associate Professor, Department of Computer Science & Engineering, IITE, Indus University, Ahmedabad	Dr. Sheetal Pandya Head of Department, Department of Computer Science & Engineering, IITE, Indus University, Ahmedabad	Prof. Zalak Vyas Head of Department, Department of Computer Science & Engineering, IITE, Indus University, Ahmedabad

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING
COMPUTER SCIENCE & ENGINEERING
2024 -2025

CERTIFICATE

Date: 22-04-2025

This is to certify that the project work entitled "**Health Tracker"** has been carried out by **Dharita Jamadagni Patel** under my guidance in partial fulfillment of degree of Bachelor of Technology in **COMPUTER SCIENCE & ENGINEERING (Final Year)** of Indus University, Ahmedabad during the academic year 2024 – 2025.

Dr. Kaushal Jani Associate Professor, Department of Computer Science & Engineering, IITE, Indus University, Ahmedabad	Dr. Sheetal Pandya Head of Department, Department of Computer Science & Engineering, IITE, Indus University, Ahmedabad	Prof. Zalak Vyas Head of Department, Department of Computer Science & Engineering, IITE, Indus University, Ahmedabad

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Dr. Kaushal Jani, Associate Professor in the Department of Computer Science & Engineering, I.I.T.E., Indus University, for his invaluable guidance and support throughout the course of our final year project, Health Tracker. His insightful feedback, technical expertise, and constant motivation have played a crucial role in shaping our work and enhancing our understanding of the MERN Stack development.

We are also grateful to all faculties of Indus University, Ahmedabad for their kind cooperation and able guidance. Without their constant support and teaching this project would only be a dream.

We acknowledge the contribution of various open-source communities, documentation resources, and online learning platform that helped us overcome technical challenges during development.

This project has been an incredible learning journey, and we are grateful to everyone who has been part of it in any capacity.

- Beauty Pankajbhai Bechara
IU2141230010
Computer Science & Engineering

-Dharita Jamadagni Patel
IU2141230164
Computer Science & Engineering

TABLE OF CONTENT

ABSTRACT

*The **Health Tracker** is an intelligent, user-centric platform designed to help iindividuals monitor and manage their health through detailed, data-driven insights. This web application offers a centralized dashboard that consolidates vital health metrics such as steps, calories burned, workout duration, diabetes level, blood pressure, blood oxygen, stress, sleep, and more. It features intuitive navigation with dedicated pages for each metric, including visual summaries through progress bars, graphs, and detailed cards. Users are categorized as either new or existing—while new users gain limited access to features like the medical report analyzer, profile, and contact pages, existing users enjoy full access to comprehensive tools such as health data tracking, dashboard visualization, and personalized recommendations. This report thoroughly documents the development of the website, outlining its architecture, key functionalities, data flow, and user interaction design.*

The frontend of the website is developed using React.js and CSS, providing a responsive and interactive user interface. The backend is built using Node.js and Express.js, handling authentication, user data processing, and secure communication. MongoDB serves as the primary database for storing user information and health records, ensuring a scalable and flexible data structure. The application also integrates OpenAI's API for generating intelligent insights and summaries from uploaded medical reports, and Email.js for communication services. The system follows the Agile Development Process Model, ensuring iterative development with regular feedback, testing, and enhancements throughout the lifecycle.

In conclusion, the Health Tracker Website is a comprehensive digital health companion that empowers users with actionable insights derived from real-time data analysis. By integrating machine learning-driven health prediction, graphical representation of trends, and personalized health advice, the system bridges the gap between raw medical data and user-friendly interpretation. The platform not only promotes proactive health management but also enhances user awareness through simplified medical reporting and tailored lifestyle recommendations, making it a valuable tool for both daily monitoring and long-term wellness planning

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

Abbreviations used throughout this whole document are:

- HTML - Hypertext Markup Language
- UML -Unified Modeling Language
- CSS - Cascading Style Sheet
- AI-Artificial Intelligence
- UI-User Interface
- UX-User Experience
- API-Application Programming Interface
- CRUD- Create, Read, Update, Delete
- SDLC-Software Development Life Cycle
- SRS-Software Requirement Specifications
- FR-Functional Requirement
- TS -Task Set
- VS Code – Visual Studio Code
- HTTP – Hypertext Transfer Protocol
- URL – Uniform resource Locator
- SQL – Structured Query Language
- DD – Data Dictionary
- MERN - MongoDB, Express.js, React.js, Node.js
- JS – JavaScript
- IoT - Internet of Things
- Node.js - Node JavaScript
- Express.js - Express JavaScript
- MongoDB - Mongo Database
- Git - Global Information Tracker
- GitHub - Git + Hub (Web-based Git Repository Hosting Service)
- Chart.js - Chart JavaScript Library
- Recharts - React-based Charting Library
- ER – Entity -relationship

- SSL- Secure Sockets Layer
- WBS- Work Breakdown Structure
- OpenAI- Open Artificial Intelligence
- GDPR – General Data Protection Regulation
- HIPAA – Health Insurance Portability and Accountability Act
- RBAC – Role based Access Control
- FA- Functional Analysis
- JWT – JSON Web Token
- CSRF – Cross-Site Request Forgery
- XSS - Cross-Site Scripting
- DPDP – Digital Personal Data Protection
- AMD – Advanced Micro Devices
- SSD- Solid State Drive
- AWS – Amazon Web Services
- OCR – Optical Character Recognition
- UAT – User Acceptance Testing
- CI – Continuous Integration
- CD - Continuous Deployment
- DOB – Date of Birth
- MUI - Material-UI
- BPM – Beats Per Minute
- BMI – Body Mass Index
- CTA- Call to Actions
- PDF – Portable Document Format
- MB – Megabyte
- GIF- Graphics Interchange Format

CHAPTER 1

INTRODUCTION

- **Project Summary**
- **Project Purpose**
- **Project Scope**
- **Objective**
- **Technology and Literature Overview**
- **Synopsis**

1.1 PROJECT SUMMARY

The Health Tracker Website is a modern, web-based platform developed as part of a final-year engineering project using the MERN stack (MongoDB, Express.js, React.js, Node.js). The website is designed to offer a seamless health monitoring experience, providing users with an organized way to track various physical health metrics through an interactive and visually rich digital dashboard.

The core aim of this project is to empower users with personalized, data-driven insights to support better lifestyle decisions and consistent health monitoring. By integrating real-time tracking, medical report analysis, and health parameter-based diagnostics, the platform encourages users to take preventive measures and stay informed about their well-being.

The platform allows registered users full access to multiple health tracking pages, while non-registered users are given limited access to essential tools like the Medical Report Analyzer, Profile, Data Analysis, and Contact pages. Key modules include the Diabetes Analysis, Blood Pressure Monitoring, and Medical Report Analyzer—each offering personalized assessments and recommendations. The dashboard provides an overview of user data with circular progress bars and health cards, while individual pages offer detailed graphs, trend summaries, and daily/weekly/monthly comparisons. Insights are generated based on user inputs combined with standard medical parameters, making health interpretation more accessible.

Key Features of the Platform Include:

A. Dashboard Page: Displays an overview of key metrics like steps, calories burned, and workout duration using circular progress bars, plus 4 health cards (Blood Pressure, Diabetes, Blood Oxygen, Weight/Height) with visual graphs and a summary section.

B. Step Tracker Page: Shows user's step data for daily, weekly, and monthly periods with a summary graph, maximum/minimum step count, and insightful tracking metrics.

C. Heartbeat Rate Page: Monitors pulse readings with daily/weekly/monthly graphs and a breakdown of average, highest, and lowest heartbeat rates.

D. Sleep Hours Page: Tracks sleep duration trends with detailed analysis of sleep quality across days, weeks, and months, and provides recovery suggestions.

E. Stress Level Page: Displays stress patterns and critical highs using visual graphs; offers insights on stress management with data-driven suggestions.

F. Calorie Burn Page: Shows calories burned over time, highlights workout effectiveness, and includes recommendations for improved fitness planning.

G. Blood Pressure Page: Accepts systolic/diastolic input with contextual data like sleep and stress; provides BP classification, risk identification, and health advice per category (e.g., Stage 1, Hypertensive Crisis).

H. Diabetes Page: Takes age, height, weight, and sugar levels to calculate BMI and categorize diabetes stage; generates lifestyle recommendations and diet/exercise tips.

I. Weight & Height Page: Tracks BMI, weight fluctuation, and health classification (underweight, normal, overweight, obese) using bar and line graphs for trend comparison.

J. Blood Oxygen Page: Analyses oxygen saturation levels with visual patterns and alerts if readings fall outside the healthy range.

K. Data Analysis Page: Provides combined summary of all tracked metrics with graphs, maximum and minimum values, and health trend visualization.

L. Upload Page – Parameter Input Upload: Allows users to manually upload text-based or structured input data (e.g., blood pressure, sugar, sleep, stress levels) in CSV or form format. This data is directly used for analysing health conditions like BP or Diabetes through graphs and personalized insights.

M. Medical Report Upload & Analyzer Page: Users can upload scanned medical reports in image or PDF format. The system uses OCR (Optical Character Recognition) to extract relevant health information and provides user-friendly, summarized results with medical insights and improvement suggestions.

N. Profile Page: Displays user's personal, general, and medical details with editable sections to update health records.

O. Contact Us Page: Offers direct communication for user queries, website feedback, or issues through an in-built contact form.

P. Authentication System (Sign-Up/Sign-In): Only registered users can access all features; new users can access limited pages (Medical Report Analyzer, Profile, Contact, Data Analysis).

Q. Logout Feature: Registered users can securely log out to end their session and protect data access.

1.2 PROJECT PURPOSE

The primary purpose of the Health Tracker website is to provide users with a platform that helps them manage and monitor their physical health more effectively. With the growing emphasis on personal health and wellness, there is a pressing need for a tool that allows individuals to track vital health metrics in an easily accessible and user-friendly manner. The project aims to address this need by offering a digital solution that empowers users to gain insights into their health status and make informed decisions for a healthier lifestyle.

The main goal of the project is to create a web-based health dashboard that enables users to track critical health indicators such as steps taken, calories burned, heart rate, sleep hours, blood pressure, and weight, among others. By providing personalized insights based on individual health data and medical reports, the platform encourages users to monitor their health trends over time and take proactive steps toward maintaining or improving their well-being.

The target audience for the Health Tracker website includes health-conscious individuals, fitness enthusiasts, and people who want to keep track of their health metrics and improve their lifestyle. The website will benefit users by providing them with a comprehensive view of their health, making it easier for them to manage their fitness goals, track their progress, and make data-driven decisions to stay on top of their health.

The project will offer features such as a step tracker, blood pressure monitoring, diabetes analysis, and a medical report analyser, which will give users personalized recommendations based on their health data. These features are designed to enhance user engagement with health data, offering a seamless experience for both registered and non-registered users.

The anticipated impact of this project is to foster preventive healthcare by helping users identify health risks early, track improvements over time, and make informed lifestyle choices. Additionally, the future vision includes expanding the platform by integrating IoT devices and wearable data to offer even more accurate and real-time health monitoring capabilities.

Ultimately, this project seeks to make health management simpler and more accessible, allowing users to take control of their well-being with just a few clicks.

Key Purposes of the Health Tracker Website:

A. Simplified Health Monitoring: To offer users a convenient, web-based platform where they can regularly track essential health metrics like heart rate, sleep, steps, blood pressure, and more — all in one place.

B. Personalized Health Insights: To analyze user input and uploaded medical reports, and provide personalized health assessments and actionable recommendations for improving lifestyle and well-being.

C. Preventive Health Awareness: To encourage early detection of potential health risks through trend analysis and visual summaries, supporting users in taking proactive health measures.

D. Accessible Health Data Visualization: To present complex health information in an intuitive, visual format (graphs, progress bars, health cards), making it easier for users to understand and respond to their health status.

E. Support for Fitness Goals: To assist users in achieving their fitness and wellness objectives by tracking progress over time and adjusting goals based on actual data and system feedback.

F. Inclusive User Experience: To ensure both registered and non-registered users can benefit — with limited access for guests and full access for signed-in users — thereby promoting broader health engagement.

G. Future Integration with IoT Devices: To lay the foundation for future enhancements where real-time data from wearable health trackers and IoT devices can be integrated for continuous and more accurate health monitoring.

1.3 PROJECT SCOPE

The scope of the Health Tracker Website project defines the boundaries and key objectives of developing a full-stack web application that allows users to track, manage, and understand their physical health metrics. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), this project aims to deliver a fully functional, interactive, and accessible platform that supports features such as step tracking, heartbeat monitoring, sleep analysis, stress level tracking, calories burned, BMI calculation, blood oxygen levels, and detailed data analysis. The platform is designed for users who wish to engage with their health data regularly and make informed decisions to improve their lifestyle and well-being.

The core deliverables of the project include a dynamic user interface with individual modules for different health indicators, a secure login and registration system, a medical report analyzer with OCR capabilities, and a dashboard that visualizes user data in the form of graphs and progress cards. The application will support both registered and non-registered users, offering tiered access to maintain user engagement and privacy. Registered users will have access to all features, while non-registered users can explore limited tools such as data analysis and medical report uploads.

While the platform will provide personalized insights based on uploaded medical reports and input data, it does not include real-time integration with wearable devices or live health consultations in its initial release. These components are considered out of scope for the current development phase but are intended as future enhancements. The project assumes a stable internet connection and basic technical understanding on the part of the users.

This scope outlines a three-month development window with clear phases including planning, design, development, testing, and final deployment. The website will be

optimized for responsiveness, security, and user-friendliness, ensuring that users can monitor their health conveniently and effectively through a reliable digital platform.

A. Objectives

The primary objectives of the project are:

- To develop a responsive, full-stack health tracking website using the MERN stack.
- To provide users with tools to track physical health metrics such as steps, heart rate, sleep, stress, BMI, calories, BP, and oxygen levels.
- To offer visual, data-driven health insights to help users make informed lifestyle decisions.
- To integrate an OCR-based Medical Report Analyzer for extracting health data from user-uploaded documents.
- To implement secure user authentication and tier-based feature access (registered vs non-registered).

B. Deliverables

The project will produce the following deliverables:

- A fully functional web application built with MongoDB, Express.js, React.js, and Node.js.
- Interactive dashboard with graphical health data summaries.
- Individual pages/modules for:
    - Step tracking
    - Sleep and stress monitoring
    - Calorie and heartbeat tracking
    - BMI and blood oxygen analysis
    - Blood pressure and diabetes tracking
    - Medical Report Analyzer with OCR integration.
    - Secure login/signup system with logout functionality.
    - User profile management module.
    - Tiered access levels for registered and non-registered users.
    - Contact form for user feedback or queries.

C. Tasks and Activities of the Project

The project will involve the following tasks and activities:

- Requirement gathering and scope finalization.
- UI/UX design of all pages.
- Backend API development and database design.
- Frontend development with responsive design.
- Integration of OCR functionality for report analysis.
- Implementation of user authentication and access control.
- Testing and debugging (unit, integration, UI testing).
- Final deployment and user manual preparation.

D. Stakeholders of the Project

The key stakeholders involve in the project include:

- Primary Developer(s) – Responsible for development and deployment.
- Project Guide/Mentor – Provides guidance and supervision throughout.
- End Users – Health-conscious individuals, fitness enthusiasts, patients, or casual users.
- Academic Evaluators – Review and assess project quality and functionality.

E. Resources Required for the Project

The resources required for the project include:

- Hardware – Laptop/PC with internet connectivity.
- Software & Tools**:**
    - Visual Studio Code (IDE)
    - MongoDB Atlas (Database)
    - Node.js and Express.js (Backend)
    - React.js (Frontend)
    - GitHub for version control
- Development Timeframe – 3 months
- Team Collaboration Tools – Google Meet, Trello, or Notion (optional)

F. Constraints of the Project

The project is constraints to the following constraints:

- No real-time integration with external wearable devices (e.g., Fitbit, Apple Watch) in the current version.
- No live medical consultation or diagnosis features.
- Platform assumes basic digital literacy from users.
- Requires a stable internet connection for access.
- Limited budget and timeline due to academic constraints.

G. Exclusions (To Prevent Scope Creep)

- Integration with Government/Insurance Databases – Out of current scope.
- Mobile App Version – Currently limited to a web-based platform.
- Wearable Device Support – Scheduled for future implementation phases.
- Multilingual Support – Limited to English in this version.

H. Project Scope Statement

The Health Tracker Website aims to deliver a web-based, interactive, and data-driven health monitoring system using the MERN stack. It will empower users to monitor vital health parameters, gain insights through personalized analytics, and manage health data effectively. The project will be completed over a 3-month period, including planning, development, testing, and deployment. The platform will support tiered access levels, OCR-based medical report analysis, and visual health data representation, while excluding real-time device integration and live medical consultations in its initial scope.

1.4 OBJECTIVES

The objective of the Health Tracker Website is to create an accessible and reliable platform that assists users in actively monitoring their physical well-being through a variety of

health-related features. The website is designed to offer interactive and informative tools that allow users to visualize trends, analyze medical data, and make informed decisions about their health. The project intends to enhance user awareness about their fitness and wellness by combining basic health tracking functionalities with meaningful data interpretation, all wrapped in a user-friendly interface. By delivering key health insights in a structured and responsive web environment, the system encourages a more preventive approach to personal healthcare.

1.4.1 Main Objectives

A. To develop a web-based health monitoring platform that allows users to track key health indicators such as heart rate, sleep duration, step count, calories burned, blood pressure, oxygen saturation, and BMI.
B. To design and implement personalized dashboards for registered users, offering visually appealing charts, graphs, and data cards that reflect individual health trends and performance.
C. To build a dynamic health analysis system capable of processing both manually entered health data and medical report uploads, with OCR (Optical Character Recognition) technology for automatic extraction and interpretation of health indicators.
D. To provide contextual health insights and recommendations based on collected user data, helping users make informed decisions for healthier lifestyle management.
E. To ensure a tier-based access structure, allowing full access to registered users while offering limited tool exploration (e.g., data analysis and report upload) to non-registered visitors, increasing engagement and encouraging registration.
F. To foster self-awareness and preventive health care by enabling users to regularly monitor changes in their health metrics and identify potential risks over time.
G. To ensure the system is scalable, responsive, and user-friendly, enabling convenient usage across devices and ensuring accessibility to a wide range of users with basic digital literacy.

1.4.2 Secondary Objectives

A. To ensure a responsive and seamless user experience across various devices and screen sizes, optimizing accessibility and usability.

B. To implement secure user authentication and authorization mechanisms to protect sensitive health data and maintain user privacy.

C. To incorporate modular components such as a goal tracker, educational health cards, and a contact portal, enhancing user engagement and support.

D. To design the system for scalability, allowing smooth integration of future enhancements like data collection from smartwatches or IoT health devices.

E. To establish a foundation for future features such as real-time health alerts, consultation tools, and automated health suggestions, making the platform more robust and interactive.

F. To promote user interaction and long-term engagement by offering helpful tools and features that support consistent health monitoring behaviour.

1.5 TECHNOLOGY AND LITERATURE OVERVIEW

The Health Tracker Website is developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), providing a scalable and efficient structure for building data-driven health applications. The React.js frontend enables a clean, interactive UI with modular components for tracking health metrics like steps, sleep, and heart rate. On the backend, Node.js and Express.js handle APIs and data processing, while MongoDB stores user profiles and health stats.

Unlike platforms like Google Fit or Fitbit, which rely on premium features or devices, this browser-based system uses self-reported data to deliver accessible, personalized insights. The project also integrates data visualization and AI-powered report analysis, offering a unified platform for proactive health management without subscriptions or wearables.

Technology Overview:

- The frontend (client-side) is developed using React.js with CSS and SCSS for styling. It ensures a responsive and user-friendly interface where users can track health metrics, receive insights, and view visualizations.
- The backend (server-side) is handled using Node.js and Express.js to manage routes, logic, and APIs.

- MongoDB is used for storing large sets of health and user data efficiently.
- Additional tools such as Email.js (for contact communication) and OpenAI's GPT.js (for medical report analysis) are integrated to enhance functionality and intelligence.

1.5.1 Frontend Technologies

A. React.js
   - Used to build a component-based UI, making it easy to manage different health modules like sleep tracking, heart rate, calorie burn, etc.
   - Allows dynamic rendering of user-specific data through props and state management.
   - Supports integration of data visualization libraries to show health insights in the form of graphs and progress bars.

B. CSS (Cascading stylesheets)
   - CSS was used for global styling of layouts and interactive elements like buttons, cards, modals, and graphs.
   - CSS was implemented for modular and reusable styling, with nested rules and variables improving the efficiency of design updates.
   - Ensured that the interface is responsive across devices such as desktops, tablets, and smartphones.

C. Email.js
   - Implemented for sending contact emails and user queries directly from the contact portal.

1.5.2 Backend Technologies

A. Node.js
   - Used to create a non-blocking, event-driven server that efficiently handles API requests and logic.
   - Manages the connection between the frontend and the database.

B. Express.js
   - Built RESTful APIs to handle user login, health data processing, and report uploading.
   - Provides middleware integration for security and input validation.

1.5.3 Database – MongoDB

MongoDB was used to store, retrieve, and update user and health tracking data**.** The flexible schema supports complex nested data.

1.5.4 APIs and Integrations

1. gpt.js
    - Integrated with OpenAI's API for medical report analysis using OCR and natural language processing.
    - Uploads the report, extracts text, and returns insight-based responses using GPT.
2. user.js
    - API endpoint to register, login, and fetch user profile data**.**
    - Used for handling secured user data and providing personalized access to dashboards.
3. healthdatamodel.js
    - Handles CRUD operations on health metrics**.**
    - Fetches user-specific health statistics such as steps, BP, heart rate, and updates the dashboard frontend accordingly.

1.6 SYPNOSIS

The Health Tracker Website is an interactive, browser-based application tailored to simplify how individuals observe and reflect on their physical well-being. Designed and developed as part of an academic capstone project, this system brings together intuitive design with practical health-monitoring capabilities, offering a seamless experience for users aiming to maintain a healthier routine through digital means.

In recent years, health awareness has significantly increased, prompting the need for platforms that are not only informative but also personalized and consistent. This project responds to that demand by creating an organized digital environment where users can view their progress, analyze their health inputs, and adjust their habits accordingly—all without relying on third-party applications or physical fitness journals.

The application provides a structured interface where data is visually represented using dynamic charts and progress indicators. Each feature of the system—from daily steps and calories burned to stress levels and sleep hours—works cohesively to deliver a comprehensive health snapshot. What sets this project apart is its multi-dimensional layout, allowing users to explore various metrics in depth while tracking improvements over different time frames.

Built using the MERN stack, the project emphasizes scalability, responsiveness, and maintainability. The separation between frontend and backend ensures that future upgrades, including IoT integration or mobile app support, can be managed with minimal restructuring.

This project also focuses on user interaction and accessibility. Whether a user logs in to monitor their blood pressure or reviews their weekly sleep trends, the platform ensures data is displayed in an understandable and actionable format. Through this, the system encourages self-awareness and timely decision-making.

By bridging everyday technology with personal wellness, the Health Tracker Website sets a foundation for a proactive, informed, and consistent approach to health monitoring for users of all age groups.

1.6.1 Methodology:

A. The project follows the Agile Development Model, emphasizing flexibility, iterative improvement, and user-centric features. The methodology includes:
B. Sprint Planning: Each phase was divided into short sprints focusing on key modules (e.g., login, dashboard, analytics).
C. Daily Stand-ups (informal team check-ins) to monitor progress and roadblocks.
D. Incremental Development: Features were integrated and tested in phases to ensure stability.

E. User Feedback: Inputs from mentors and sample users helped refine UI and functionality.
F. Continuous Integration: Regular updates with version control ensured seamless collaboration.
G. Testing: Each sprint ended with testing (unit + integration) before moving to the next phase.

1.6.2 Expected Outcomes

A. A fully functional, browser-based health tracking dashboard.
B. Real-time analytics for health metrics like steps, calories, sleep, heart rate, etc.
C. Personalized feedback using integrated GPT-based medical report analysis.
D. User authentication system with secure data storage in MongoDB.
E. Scalable architecture for future integration with mobile apps or wearables.
F. Accessible UI encouraging daily health monitoring without premium subscriptions.

CHAPTER 2

LITERATURE SURVEY

- **Introduction of Survey**
- **Purpose of the Survey**
- **Area Covered in the Survey**

2.1 INTRODUCTION OF SURVEY

Introduction of the Survey

The survey conducted for the *Health Tracker* project serves as a foundational step in understanding user needs, identifying potential gaps in current health monitoring tools, and aligning the system's functionalities with real-world expectations. As part of the research and development phase, the survey was designed to gather critical data from a targeted group of individuals, including students, working professionals, and individuals with specific health concerns.

This user-centered approach ensures that the system not only incorporates essential features but also remains relevant, accessible, and practical for daily use. The survey emphasizes individual preferences regarding health data monitoring, feature usability, and expectations from a browser-based health tracking solution. By analyzing feedback from this survey, the project aims to refine its design and functionality to better cater to user habits and health priorities.

2.2 PURPOSE OF THE SURVEY

The primary purposes of the survey are as follows:

A. User-Centric Development**:** To understand user expectations and behaviour patterns related to health tracking and self-monitoring tools.
B. Feature Prioritization**:** To identify which features (e.g., step tracker, sleep monitor, diabetes analysis, blood pressure tracker, calorie burn chart) are considered most valuable by users.
C. Platform Preference**:** To analyse user inclination towards browser-based applications versus mobile apps or physical devices.
D. Technical Direction**:** To inform technological choices for frontend, backend, and database solutions based on practical use cases.
E. Accessibility and Usability**:** To ensure the platform is user-friendly for a wide audience including tech-savvy individuals and those new to digital health tools.
F. Customization and Personalization**:** To gather insight on how users would like the platform to offer personalized feedback based on their data.

2.3 AREA COVERED IN THE SURVEY

The survey comprehensively covered the following domains that align with the full-stack development and deployment of the Health Tracker Website:

A. User Interface & Frontend Design (React.js)

- Preferences for layout design**:** Whether users preferred a card-based, chart-driven, or minimal layout.
- Ease of navigation**:** How users felt about navigating between different health modules.
- Responsiveness**:** Expectations regarding accessibility from desktop, tablet, and mobile browsers.

B. Backend Functionalities (Node.js & Express.js)

- API usage**:** Preferences for real-time updates, quick loading, and fast data retrieval.
- User authentication**:** Opinion on login/signup flows, password security, and session handling.
- Data processing**:** Interest in systems that can compute health trends based on manually entered or real-time data.

C. Database Storage & Security (MongoDB)

- User trust in cloud-based data storage**.**
- Interest in viewing past health records or trends over weeks or months.
- Feedback on the safety and privacy of storing sensitive health data in the cloud**.**

D. Health Feature Modules

- Step tracker
- Sleep hours analysis
- Blood pressure and diabetes insights
- Heart rate monitor
- Stress level overview
- Calorie burn visualization
- Medical report analyzer using AI

Participants were asked to rate the usefulness and clarity of each feature, along with suggestions for improvement or personalization.

A. Accessibility & Device Independence

- Browser-only access**:** How convenient users found a solution that did not depend on wearables or app installations.
- Cross-platform compatibility: Whether the tool should extend to mobile web or remain desktop-first.

B. Future Enhancement Suggestions

- Integration with fitness wearables (optional IoT support).
- Health reminders or alert systems.
- AI-powered prediction for potential health risks.
- Dark mode and theme customization.

CHAPTER 3

HEALTH TRACKER

- **Introduction**
- **Why Health Tracker?**
- **Features**
- **Technology and Tools Used**

3.1 INTRODUCTION

In an age where health awareness and data-driven lifestyles are increasingly gaining prominence, the Health Tracker Website stands out as a versatile, user-centric digital platform designed to support proactive health monitoring. This web-based solution enables individuals to record, visualize, and evaluate key health indicators through a secure and intuitive interface. Developed with a strong focus on preventive healthcare, the platform empowers users with actionable insights derived from their self-entered data, promoting consistent tracking and early identification of potential health concerns.

The core objective of this project is to provide an accessible, efficient, and scalable tool for individuals seeking better control over their health. Unlike many digital solutions that are tied to specific mobile ecosystems or complex infrastructures, the Health Tracker Website is entirely browser-based, ensuring broad accessibility across devices without requiring any installations or integrations.

Users can manually input essential health metrics such as blood pressure, glucose levels, heart rate, and weight, and instantly view the results in an engaging visual format. The system processes these inputs to create clear, informative visualizations—including graphs, charts, and progress indicators—that help users monitor trends, detect irregularities, and make more informed lifestyle choices.

A major design principle behind the platform is clarity and ease of use. The interface has been built to eliminate technical barriers and provide a smooth user experience regardless of medical or technological background. Whether it's daily health management or long-term progress tracking, the website facilitates personal health awareness with minimal complexity.

By offering a flexible, device-independent health tracking solution, the Health Tracker Website ensures that users from diverse backgrounds—be it health-conscious individuals, patients managing chronic conditions, or those pursuing wellness goals—can benefit from a single, unified platform.

In essence, this project serves as a bridge between digital innovation and healthcare needs, delivering a robust, visually engaging, and easy-to-use application that aligns with the vision of modern preventive healthcare.

3.2 WHY HEALTH TRACKER?

The increasing complexity of modern lifestyles, coupled with a rise in lifestyle-related diseases such as diabetes, hypertension, and cardiovascular conditions, underscores the importance of regular health monitoring. In this context, the Health Tracker Website emerges as a purposeful and timely solution that addresses the need for a simple, centralized, and user-friendly platform for individuals to actively manage their health data. Many people today seek convenient tools to track their health metrics, observe long-term patterns, and take informed actions based on data. However, most existing solutions are either device-dependent, subscription-based, or restricted by platform limitations. The Health Tracker Website addresses these gaps by providing an open-access, browser-based solution that allows users to input and visualize health-related data without the need for additional devices, applications, or technical expertise.

This project is driven by the belief that health management should be inclusive, accessible, and insightful. By enabling users to manually enter key health parameters and receive immediate visual feedback, the website encourages proactive engagement and promotes preventive healthcare behaviour. Individuals can take timely measures to improve or stabilize their health conditions before they escalate, contributing to reduced healthcare costs and better overall well-being.

Moreover, in an era where health data plays a crucial role in personal decision-making, the ability to securely log, analyse, and reflect on one’s own health journey becomes essential. This platform caters to that need by offering not only a digital health log but also smart visual analytics that make it easier for users to interpret their metrics and observe meaningful trends over time.

The Health Tracker Website is not just a tool—it is a step towards health empowerment. It allows individuals to break free from passive health tracking and adopt a more engaged and

responsible approach to their well-being. Whether used by students, working professionals, the elderly, or individuals managing chronic conditions, the platform provides a reliable and flexible environment for continuous health management.

In summary, the Health Tracker Website exists to bridge the gap between health awareness and health action. Its intuitive design, data visualization capabilities, and accessibility make it a valuable companion in the journey toward a healthier lifestyle.

Manual Health Data Entry: Users can input vital health statistics such as blood pressure, blood glucose, heart rate, weight, and more.
Dynamic Data Visualization: The website offers intuitive visual representations of the user's health data, including bar charts, line graphs, and progress circles, helping users better understand trends over time.
Personalized Recommendations: Based on the entered data, the system provides basic health suggestions that guide the user toward maintaining or improving their current health condition.
Responsive Dashboard Interface: The interface is clean, modern, and mobile-friendly. It includes dashboard cards and categorized views of different health parameters.
Authentication and Secure Login: Users are required to register and log in securely before accessing their data, ensuring privacy and personalized access.
Historical Tracking and Analysis: Users can view historical data logs that help in identifying patterns and changes in their health metrics.
Doctor Suggestion Page (Optional): The system can also offer a feature that recommends specialist doctors based on the user's health data trends and inputs.
User-Friendly Navigation: Designed with user experience in mind, the website features an intuitive layout and clear navigation, ensuring even non-tech-savvy individuals can use it with ease.

3.3 KEY FEATURES OF THE HEALTH TRACKER WEBSITE

- Manual Health Data Entry: Users can input vital health statistics such as blood pressure, blood glucose, heart rate, weight, and more.

- Dynamic Data Visualization: The website offers intuitive visual representations of the user's health data, including bar charts, line graphs, and progress circles, helping users better understand trends over time.
- Personalized Recommendations: Based on the entered data, the system provides basic health suggestions that guide the user toward maintaining or improving their current health condition.
- Responsive Dashboard Interface: The interface is clean, modern, and mobile-friendly. It includes dashboard cards and categorizes views of different health parameters.
- Authentication and Secure Login: Users are required to register and log in securely before accessing their data, ensuring privacy and personalized access.
- Historical Tracking and Analysis: Users can view historical data logs that help in identifying patterns and changes in their health metrics.
- Doctor Suggestion Page (Optional): The system can also offer a feature that recommends specialist doctors based on the user's health data trends and inputs.
- User-Friendly Navigation: Designed with user experience in mind, the website features an intuitive layout and clear navigation, ensuring even non-tech-savvy individuals can use it with ease.

3.4 TECHNOLOGY AND TOOLS USED

The Health Tracker Website is built using the MERN stack, a powerful and flexible technology suite ideal for dynamic and scalable web applications. Below is a breakdown of the technologies used:

- MongoDB: A NoSQL database used to store user health data in a structured, flexible, and scalable manner. It supports fast read/write operations, making it suitable for real-time updates.
- Express.js: A lightweight web framework for Node.js, used to build the backend server. It handles routing, middleware, and API endpoints securely and efficiently.
- React.js: A JavaScript library for building interactive user interfaces. React enables the creation of reusable components, real-time updates, and a responsive frontend experience.

- Node.js: A JavaScript runtime environment used to execute backend code. It provides seamless communication between the frontend and backend layers.
- Chart.js / Recharts: Libraries used to generate graphs and charts that visually represent user health data.
- Git & GitHub: Version control tools used for project management and collaboration during development.

CHAPTER 4

PROJECT MANAGEMENT

- **Project Planning Objectives**
- **Project Scheduling**
- **Risk Management**

4.1 PROJECT PLANNING OBJECTIVES

Project planning is a foundational aspect of software development, ensuring that all elements required for successful execution are identified and managed effectively. For the Health Tracker Website, project planning serves as a critical framework that guides development from conceptualization to deployment. The primary objective is to build a robust, scalable, and user-friendly platform that allows users to manually input and analyse key health parameters. The project also aims to ensure security, responsiveness, and interactive data visualization to encourage proactive health management.

Planning objectives include identifying all deliverables, defining the scope, determining resources, estimating time and cost, managing risks, and establishing timelines. It ensures alignment between the development team and project stakeholders, enabling realistic expectations and coordinated execution. Proper planning also provides contingency measures, ensures efficient resource utilization, and helps in early detection of issues. This project particularly emphasizes the importance of a modular design and a user-centric interface to make health tracking accessible to everyone.

4.1.1 Software Scope

A. Home Page

- Overview of the platform.
- Limited feature previews for new users.
- Navigation to login/signup.

B. User Dashboard

- Centralized view displaying health stats.
- Dynamic charts for steps, calories, heart rate, etc.
- Notifications and health recommendations.

C. Step Tracker Page

- Allows users to view and track daily and hourly steps.
- Visual charts to display movement trends.
- Goal setting and comparison with previous weeks.

D. Heart Rate Monitoring Page

- Tracks and displays heart rate trends.
- Zone-based analysis (resting, moderate, intense).
- Alerts for irregular heart rates.

E. Sleep Analysis Page

- User input or smartwatch integration for sleep duration.
- Displays sleep quality, hours, and restfulness score.

F. Stress & Hydration Page

- Displays stress levels from smartwatch or user input.
- Tracks hydration level daily.
- Provides suggestions based on hydration deficit or stress overload.

G. Blood Pressure and Diabetes Analysis Page

- Input and display of systolic, diastolic, and blood sugar values.
- Categorizes values into normal/high/low.
- Diabetes status analysis with visual alerts.

H. Calorie Burn Page

- Tracks calories burned vis activity or input.
- Compares calorie intake vs. burn for goal management

I. Lab Report Analysis Page

- Upload lab reports(PDF/images).
- Uses OCR and ML models to extract, read, and analyze results.
- Summarizes reports and provides explanations.

J. Profile Page

- User details, device connection status, data preferences.

- Settings for notifications, language, theme.

K. Admin/Doctor Panel (optional)

- Doctor login to view user summaries .
- Admin can monitor activity logs and issues.

4.1.2 Resource

Efficient resource planning is vital to ensure successful and timely project completion. For the Health Tracker Website, resources are categorized into three types: human resources, reusable software components, and environmental resources. Each of these plays a crucial role in project implementation and delivery.

4.1.2.1 Human Resource

Human resources form the backbone of this project. Key roles include a project manager, frontend developer, backend developer, UI/UX designer, and database administrator. Each individual is responsible for specific aspects of the development cycle. The frontend developer handles user interface implementation using React.js, while the backend developer manages server-side logic using Node.js and Express.js. The database administrator oversees MongoDB setup and optimization. UI/UX designers ensure intuitive and engaging user experience.

These professionals must work in coordination to ensure timely completion of tasks, integration of modules, and thorough testing. Communication, accountability, and skillsets are essential factors in the efficient use of human resources.

4.1.2.2 Reusable Software Resources

Reusable software resources reduce redundancy and accelerate development. For this project, reusable assets include pre-built UI components, authentication libraries (like JWT), React charting libraries for data visualization, and utility functions for validation and formatting.

By reusing reliable and well-tested components, the team can ensure higher quality, lower error rates, and faster development cycles. Open-source libraries like React Bootstrap and chart.js offer additional tools that are easily integrated into the project.

4.1.2.3 Environment Resource

Environmental resources include all the hardware, software, and network infrastructure required for development, testing, and deployment. For the Health Tracker Website, development environments include local machines configured with necessary IDEs (e.g., VS Code), Node.js runtime, MongoDB servers, and version control via Git.

The testing environment replicates production-like conditions to detect bugs before deployment. Deployment environments involve cloud servers such as Render or Heroku for hosting the backend, and Netlify or Vercel for frontend hosting. Proper environmental setup ensures smooth workflow and seamless application performance.

Project Development Approach

The Health Tracker Website adopts an Agile development approach. Agile methodology allows for iterative development, regular feedback loops, and continuous improvement. The project is divided into small, manageable sprints, each focusing on specific modules like user authentication, data input forms, visualization dashboards, and analytics.

Agile promotes adaptability, quick prototyping, and frequent testing. This approach facilitates easy incorporation of changes and helps in maintaining transparency between the development team and stakeholders. It is especially useful for web-based projects where user feedback and functionality enhancement play a significant role in overall system success.

The project is systematically divided into six key phases: Requirement Analysis, System Design, Implementation, Testing, Deployment, and Maintenance & Evaluation. Each phase contributes significantly to building an effective health dashboard ecosystem that delivers precise medical insights, user personalization, and seamless accessibility.

1) PHASES OF DEVELOPMENT

Phase 1: Requirement Analysis Phase

This is the foundation phase of the development cycle. In this phase, detailed discussions and brainstorming sessions were conducted to identify the needs of two categories of users: registered users and new/guest users**.** The core requirements, such as secure login/signup functionality, personalized dashboards, data analysis features, and medical report analysis, were documented. User expectations for visualizations (graphs, cards, tips), personalized recommendations, and health indicators (BP, diabetes, BMI) were captured through flow diagrams and use case analysis.

Phase 2: System Design Phase

This phase focused on the architectural design of the website, including frontend layout, backend architecture, database schema, and API

integrations. For the frontend, design wireframes were developed to visualize each page—Dashboard, Diabetes, Blood Pressure, Report Analyzer, etc. Backend design involved defining data models to store user health metrics, goals, and analysis results. Technologies such as React.js for the frontend and Node.js with Express.js for the backend were finalized. The database was designed using MongoDB to store user records, medical history, and activity logs.

Phase 3: Implementation Phase

During implementation, the system was built module-wise. Features such as authentication (signup/sign in), dashboard functionalities, health cards, data analysis, and file upload systems were implemented in successive sprints. Each module was integrated with corresponding API endpoints. Circular progress bars, summary graphs, user tips, and condition-specific analysis cards (BP, diabetes) were developed to ensure a rich and interactive user experience. Conditional access features were also implemented where new users could access only a limited number of pages.

Phase 4: Testing Phase

A thorough testing strategy was followed to ensure the reliability and security of the application. Unit testing was conducted for individual components and services. Integration testing ensured seamless interaction between modules like login and dashboard, or data analysis and user profile. User acceptance testing (UAT) involved simulating real-time scenarios like uploading incorrect medical reports, inputting extreme health values, and verifying output accuracy. Bugs were tracked using project management tools and were resolved in iterative cycles.

Phase 5: Deployment Phase

Once testing was successful, the application was deployed on a secure hosting platform. Configuration files were managed to ensure that both

frontend and backend services were correctly linked. SSL certification was added for secure transmission of medical data. The database was configured to scale according to the number of users and data size. Authentication and session management were thoroughly tested post-deployment.

Phase 6: Maintenance and Evaluation Phase

The final phase focuses on real-time monitoring of the application and regular updates. Feedback mechanisms were incorporated through the "Contact Us" page, allowing users to report issues or give suggestions. Regular log analysis and server health checks are scheduled to prevent any service disruption. Feature enhancement plans such as integration with wearables, real-time health alerts, and mobile app deployment are evaluated in this phase.

This phased approach ensures clarity of objectives, systematic progress, modular testing, and continual enhancement. It not only minimizes risk but also enables the development team to deliver a feature-rich, user-centric Health Tracker Website that serves its intended purpose of promoting digital wellness and providing insightful health analysis.

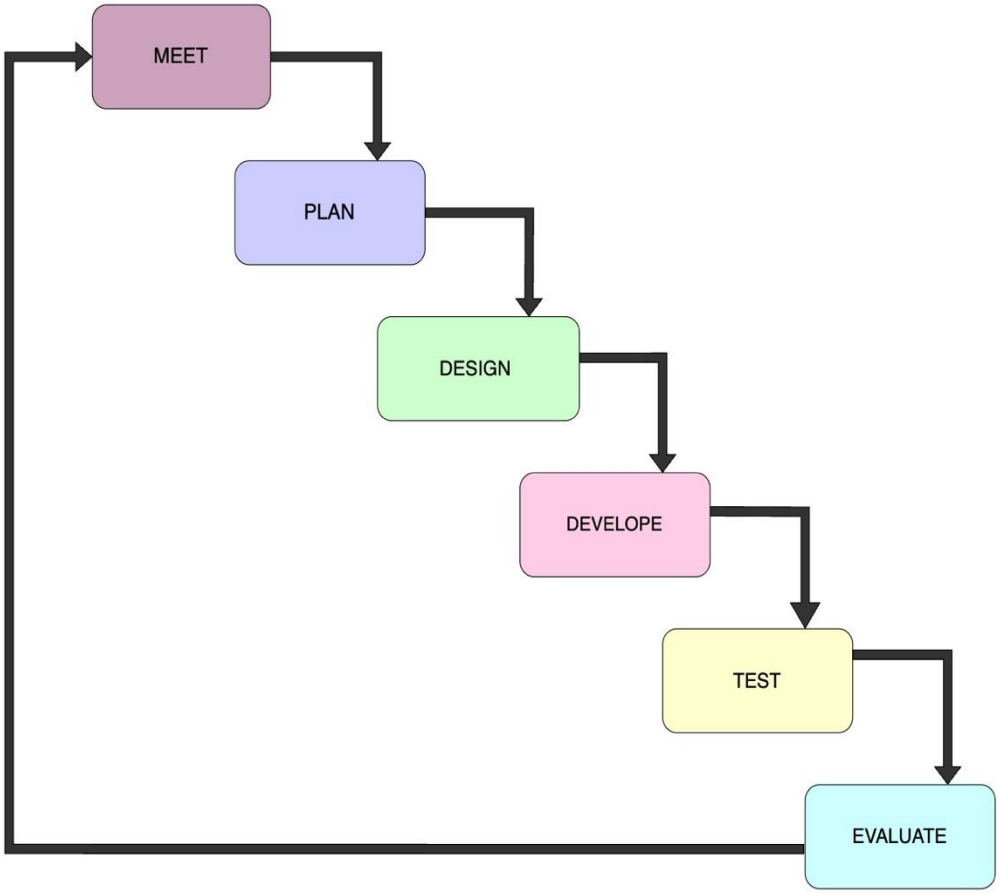

Figure 4.1.3.1:Agile SDLC

4.2 PROJECT SCHEDULING

Project scheduling is a crucial component of project planning that ensures timely delivery, efficient resource allocation, and smooth progression across all development stages. For the Health Tracker Website, scheduling revolves around structured timelines and task dependencies to meet goals like robust health data tracking, user interactivity, security, and insightful analytics.

4.2.1 Basic Principle

Define Clear Objectives: Establish what needs to be achieved by the end of the project.

In your case: a fully functional Health Tracker Website with modules like user dashboard, health data analysis, and lab report integration.

1. Break Down the Work (WBS – Work Breakdown Structure): Divide the project into smaller, manageable tasks and modules.
   Example: Authentication → Dashboard → Step Tracker → Health Pages → Analytics → Testing.
2. Determine Task Dependencies: Identify which tasks rely on the completion of others before they can start.
   E.g., UI design must precede frontend development; database setup must come before backend APIs.
3. Estimate Time and Resources: Assign realistic durations and required resources (people, tools, skills) for each task.
   Don't forget to include time for testing, feedback, and potential rework.
4. Allocate Responsibilities: Clearly assign tasks to team members based on skillset and availability.
   Promotes accountability and prevents overlaps.
5. Create a Timeline: Develop a timeline or schedule using tools like Gantt Charts, Trello, or Jira. Set start and end dates for each task and milestone.
6. Set Milestones and Deadlines: Use key checkpoints to measure progress.
7. Monitor Progress Regularly: Continuously track progress against the schedule.
8. Adjust the timeline or resources if things fall behind or change.
9. Incorporate Buffer Time: Always include extra time for unexpected delays, especially for complex tasks like analytics or ML integration.
10. Review and Revise Schedule: Stay flexible—review schedules periodically and make adjustments as needed.
    Use feedback loops from testing or stakeholder input.

4.2.2 Compartmentalization

Compartmentalization in software development refers to the process of dividing a large project into smaller, well-defined sections or modules, each responsible for specific functionality. This approach enhances manageability,

parallel development, maintainability, and scalability by isolating concerns and assigning focused tasks to different teams or components.

A. Frontend Development
    - User-friendly UI design (React.js).
    - Responsive layout for all devices
    - Interactive dashboard with charts and health data
    - Page components: Login/Register, Step Tracker, Sleep Monitor, BP & Sugar Analysis, Report Viewer.

B. Backend Development
    - RESTful API creation (Node.js + Express.js).
    - Database integration (MongoDB).
    - User authentication and authorization.
    - Data handling for health parameters (steps, heart rate, blood pressure, sugar levels, etc.)

C. AI Integration (Using OpenAI API)
    - Integration with OpenAI for lab report analysis.
    - Generation of health insights and summarization based on lab data.
    - Recommendation system for personalized health tips based on analyzed data.

D. Testing and Debugging
    - Unit testing for frontend and backend modules.
    - API testing using Postman or Swagger.
    - UI/UX testing for responsiveness and performance.
    - Bug tracking and resolution using tools like Jira or GitHub Issues.

E. Deployment

- Hosting on platforms like Vercel (Frontend) and Render/Heroku (Backend).
- Environment setup and variable management.
- Integration of CI/CD pipelines for smooth deployment.

F. Optimization

- Performance tuning (lazy loading, database indexing, image compression).
- Code optimization for faster load time.
- Scalability enhancements to handle user load.
- Security hardening (data encryption, HTTPS, JWT token security

4.2.3 Work Breakdown Structure (WBS)

The Work Breakdown Structure for your Health Tracker Website breaks down the major modules and tasks as follows:

Level 1: Project — Health Tracker Website

Phase	Task	Subtasks	Responsible Team
Sprint 1: Core Setup	Project Initialization	Repo setup, Tech stack setup, Folder structure	All Members
	MongoDB Integration	Schema design for users, reports, analysis, goals	Backend Developer, DB Admin
	EmailJS Integration	Integrate EmailJS, Setup contact/feedback form	Frontend Developer

	Authentication	Login, Register, JWT, Email validation (optional)	Backend Developer
	User Profile Management	Profile editing, Health goals setup	Frontend + Backend Dev
Sprint 2: Dashboard Dev	Dashboard	Sidebar layout, Chart placeholders, Dashboard page	Frontend Developer
	Health Data Tracking	Step Tracker, Calories, Heartbeat, Sleep Monitor, Stress Level	Frontend + Backend Developer
	Backend APIs	CRUD for steps, health data, dashboard info	Backend Developer
	UI Routing	Protected routes, route guards, navigation setup	Frontend Developer
Sprint 3: Report Analyzer	Report Upload & Analysis	Upload feature, File parsing, Send to OpenAI API	Backend + AI Integration Lead
	AI Integration (OpenAI)	Prompt design, Response handling, Display AI-generated output	AI Integration Lead
	Data Storage	Save analysis results in MongoDB	Backend Developer
Sprint 4: Analytics	Data Visualization	Chart.js / Recharts integration, Graphs, Trends, Insights	Frontend Developer

	Health Comparison View	Compare current vs previous metrics, trends	Frontend Developer
	Health Tips	Display insights or suggestions from OpenAI	AI Integration + Frontend Dev
Sprint 5: QA & Testing	Testing & Debugging	Unit Testing (Jest), API Testing (Postman), UI/UX checks	Tester
	Bug Fixes	Resolve frontend/backend bugs, optimize API calls	All Developers
Sprint 6: Deployment	Deployment	Host Frontend on Vercel, Backend on Render, Connect with Env Keys	Deployment Lead
	CI/CD Pipeline	Setup GitHub Actions or similar tool for auto-deploy	Deployment Lead
	Final Review + Feedback	Internal review, feedback incorporation, polish UI	All Members

Table 4.2.3.1 Work Breakdown Structure

4.2.4 Project Organization

The project is organized into specific roles, with each team member responsible for well-defined tasks to ensure smooth execution of the Health Tracker Website.

A. Frontend Developer

- Designs and implements UI using ReactJS.
- Integrates health monitoring pages like Step Tracker, Sleep Monitor, etc.
- Implements interactive charts and responsive layout.

B. Backend Developer

- Develops RESTful APIs using NodeJS and ExpressJS.
- Handles routing, authentication, and data manipulation logic.
- Connects backend services with MongoDB.

C. Database Designer

- Designs and structures MongoDB schemas (e.g., User, Reports, Analysis, Goals).
- Ensures efficient indexing and optimized database queries.

D. AI Integration Lead

- Integrates OpenAI API for automated medical report analysis.
- Designs custom prompts and handles AI-generated results.Works on displaying and interpreting AI insights in the frontend.

E. Tester

- Performs unit and integration testing across frontend and backend modules.
- Uses tools like Postman for API validation.
- Identifies and tracks bugs for resolution.

F. Deployment Manager

- Manages deployment of the frontend (Vercel) and backend (Render).
- Sets up CI/CD pipelines (e.g., GitHub Actions).
- Maintains environment variables and production readiness.

G. Documentation Lead

- Prepares all project-related documents.
- Includes Data Dictionary, ER Diagrams, Class Diagrams, and User Manuals.
- Ensures version control and clarity for future reference.

4.2.5 Timeline Chart

The Timeline Chart for the Health Tracker Website outlines the phased, time-bound progression of the project, reflecting key activities mapped to Agile sprints. It helps track progress, allocate time efficiently, and ensures each component—from authentication to deployment—is completed within the project duration.

Week 1 (Jan 1 – Jan 7): Requirements & Planning

- Finalize project scope and modules.
- Gather user and system requirements.
- Design basic UI wireframes and mockups.
- Define project roles and responsibilities.

Week 2 (Jan 8 – Jan 14): Authentication & Database Setup

- Design basic UI wireframes and mockups.
- Define project roles and responsibilities.
- Implement user registration/login system.
- Setup JWT-based authentication.
- Design MongoDB schemas (User, Health Data, Reports, etc.).
- Configure backend connection to MongoDB.

Week 3 (Jan 15 – Jan 28): Dashboard & Health Tracker

- Add real-time health tracking modules:
- Step tracker
    - Calories burned
    - Heartbeat rate
    - Sleep monitor
    - Stress level monitor

Week 4 (Jan 29 – Feb 11): Medical Report Analyzer

- Enable medical report upload (PDF/Images).

- Connect OpenAI API for lab report analysis.
- Display AI-generated interpretations.
- Save analysis data in MongoDB.

Week 5 (Feb 12 – Feb 25): Testing & Polish

- Conduct unit and API testing using Jest/Postman.
- Fix bugs and optimize performance.
- Improve UI/UX (responsive layout, interactivity).
- Implement error handling and feedback loops.

Week 6 (Feb 26 – Mar 10): Deployment & Documentation

- Deploy frontend on Vercel and backend on Render.
- Configure environment variables and secure endpoints.
- Prepare final project documentation:
    - Data Dictionary
    - ER/Class Diagrams
    - User Manual
- Connect OpenAI API for lab report analysis.

Week 7 (Mar 11 – Apr 10): Buffer & Feedback Phase

- Reserve time for:
    - Final review from mentors/supervisors.
    - Incorporating feedback and changes.
    - Final testing and user acceptance.
    - Preparing for viva or presentation.

4.2.5.1 Time Allocation

The development of the Health Tracker Website spanned approximately 4 months (14 weeks), structured in agile sprints**,** with time distributed among core development, testing, integration, and documentation tasks. The time allocation reflects realistic effort spent per activity, considering the iterative improvements, debugging cycles, and AI integration complexities.

Here's the breakdown of estimated time allocation:

Task	Estimated Duration	Time Period (Approx.)	Description
UI/UX Design	10 days	Jan 1 – Jan 10	Designed wireframes, mockups, and responsive layouts for all pages/modules.
Authentication Module	6 days	Jan 11 – Jan 16	Implemented login, registration, JWT auth, form validation, and route guards.

Backend & DB Setup	8 days	Jan 17 – Jan 24	Created Express.js routes, MongoDB schemas, and connected backend to frontend.
Dashboard & Health Modules	12 days	Jan 25 – Feb 5	Developed step tracker, calories, stress, sleep, heart rate, BP, and diabetes.
Report Analyzer Integration	8 days	Feb 6 – Feb 13	Implemented upload logic, parsed PDF, integrated OpenAI API, stored results.
Data Visualization	6 days	Feb 14 – Feb 20	Created bar, line, and pie charts for analysis insights using Chart.js.
User Profile Management	4 days	Feb 21 – Feb 24	Enabled profile editing, goals, and emergency contact management.
Testing (Unit/API/UI)	8 days	Feb 25 – Mar 5	Conducted frontend tests (Jest), backend API tests (Postman), and UI validation.
EmailJS Integration	2 days	Mar 6 – Mar 7	Connected contact form to EmailJS and validated

			submission feedback loop.
Documentation	5 days	Mar 8 – Mar 13	Prepared data dictionary, diagrams, WBS, user manual, project reports.
Deployment	4 days	Mar 14 – Mar 17	Hosted frontend (Vercel), backend (Render), setup environment variables.
Final Review & Polish	4 days	Mar 18 – Mar 21	Resolved bugs, polished UI, ensured mobile responsiveness, finalized details.
Buffer & Feedback Loop	12 days	Mar 22 – Apr 3	Incorporated internal/external feedback, fixed critical issues if any.
Final Submission Prep	7 days	Apr 4 – Apr 10	Last round of review, ZIP packaging, submission, presentation prep.

Table 4.2.5.1.1 Time Allocation Chart

4.2.5.2 Task Sets

Grouped into logical sets for better tracking:

Set A: Core Setup

- Setup Git repository
- Create folder structure (Frontend, Backend)
- Setup React, NodeJS, MongoDB

Set B: Authentication Module

- Register/Login
- Email verification (optional)
- Token-based security (JWT)

Set C: UI & Dashboard

- Create Sidebar and Routing
- Health Widgets (Steps, Sleep, Stress, Heartbeat)
- Data Cards + Line/Bar charts

Set D: Report Analysis Module

- Report upload (PDF/Image/Text)
- Send content to OpenAI API
- Display interpreted results

Set E: Backend APIs + DB

- Create RESTful routes (user, report, health)
- Build MongoDB collections
- Connect APIs to Frontend

Set F: Testing & Optimization

- Postman test cases
- Mobile responsiveness checks

- Error handling and alerts

Set G: Deployment

- Deploy React app (Vercel)
- Deploy backend (Render/Heroku)
- Link both with correct environment keys

4.3 RISK MANAGEMENT

4.3.1 Risk Identification

Risk Identification involves recognizing potential issues that could impact the project's success. For your Health Tracker Website, consider the following risks:

1. Data Security and Privacy Risks:
    - Unauthorized Access: Risk of unauthorized users accessing sensitive health data.
    - Data Breaches: Potential for data breaches that could compromise user information.
    - Compliance Issues: Non-compliance with data protection regulations like GDPR or HIPAA.
2. Technical Risks:
    - System Failures: Downtime or crashes of the website.
    - Software Bugs: Errors in the code that could affect functionality.
    - Scalability Issues: Difficulty in handling a large number of users or data.
3. User Experience Risks:
    - Usability Issues: Difficulty for users to navigate or understand the website.
    - Accessibility Problems: Lack of accessibility features for users with disabilities.
    - User Engagement: Low user engagement or retention rates.
4. Data Accuracy Risks:

- Inaccurate Data: Incorrect data entry or analysis leading to wrong health recommendations.
- Data Integration: Issues with integrating data from different sources (e.g., wearable devices, manual entries).

5. Operational Risks:
    - Team Coordination: Lack of coordination among team members.
    - Resource Availability: Shortage of resources (e.g., time, budget, personnel).
    - Project Delays: Delays in development or deployment phases.

4.3.1.1 Risk Identification Artifacts

- Risk Register: A document listing all identified risks, their potential impact, and likelihood.
- Stakeholder Feedback: Input from users, developers, and other stakeholders about potential risks.
- Historical Data: Analysis of past projects to identify recurring risks

4.3.1.3 Risk Projection

Risk Projection involves estimating the potential impact and likelihood of identified risks. For your Health Tracker Website, consider the following projections:

1. Data Security and Privacy Risks:
    - Impact: High (compromise of sensitive health data).
    - Likelihood: Medium to High (depending on security measures in place).
    - Mitigation: Implement robust encryption, regular security audits, and compliance checks.
2. Technical Risks:
    - Impact: High (system downtime or functionality issues).
    - Likelihood: Medium (depending on testing and maintenance).
    - Mitigation: Regular testing, backup systems, and scalable architecture.

3. User Experience Risks:
    - Impact: Medium to High (user dissatisfaction or abandonment).
    - Likelihood: Medium (depending on usability testing).
    - Mitigation: Conduct usability testing, gather user feedback, and iterate on design.
4. Data Accuracy Risks:
    - Impact: High (incorrect health recommendations).
    - Likelihood: Medium (depending on data validation processes).
    - Mitigation: Implement data validation checks, user training, and clear instructions.
5. Operational Risks:
    - Impact: Medium to High (project delays or failures).
    - Likelihood: Medium (depending on project management).
    - Mitigation: Use project management tools, regular team meetings, and resource planning.

Additional Considerations

- Regular Risk Reviews: Conduct periodic risk reviews to update the risk register and mitigation strategies.
- Communication Plan: Ensure clear communication channels for reporting and addressing risks.
- Contingency Plans: Develop contingency plans for high-impact risks to minimize disruption.

CHAPTER 5

SYSTEM REQUIREMENTS

- **User Characteristics**
- **Functional Requirements**
- **Non-Functional Requirements**
- **Hardware and Software Requirements**

5.1 USER CHARACHTERISTICS

User Characteristics refer to the attributes, behaviours, and needs of the individuals who will be using the Health Tracker Website. Understanding these characteristics helps in designing a user-centric platform that meets their expectations and requirements.

A. **Target Audience:**
   - Health-Conscious Individuals**:** Users who actively monitor their health metrics to maintain a healthy lifestyle.
   - Patients with Chronic Conditions: Users with conditions like diabetes or hypertension who need to regularly track their health parameters.
   - General Users: Individuals who occasionally check their health metrics and reports.

B. **User Roles:**
   - Registered Users: Users who have created an account and have full access to all features and pages of the website.
   - New Users: Users who have not created an account yet and have limited access to specific pages such as the medical report analyzer, data analysis, profile, and contact us pages.

C. **User Behaviour:**
   - Frequency of Use: Regular users who track their health metrics daily or weekly.
   - Purpose of Use: Monitoring health parameters, analysing medical reports, and getting personalized health recommendations.
   - Engagement Level: High engagement for health-conscious individuals and patients with chronic conditions, moderate engagement for general users.

5.2 FUNCTIONAL REQUIREMENTS

Functional Requirements describe the specific behaviours and functions that the Health Tracker Website must perform to meet user needs and achieve project goals.

5.2.1 Activity and Proposed System

- Dashboard Page
- Set Your Goal
- Labels for steps, calories burned, and workout duration with fixed goals.
- Circular progress bars displaying daily data for steps, calories, and workout duration.

- **Cards:**
  - Blood Pressure: Graphical representation of weekly data with line and bar graphs.
  - Diabetes: Parameters like age group, height, weight, blood sugar level, and analysis risk.
  - Blood Oxygen: Weekly data representation.
  - Weight and Height: Weekly data representation.
  - Data Analysis Page: Access to detailed health data analysis.
  - Medical Report Analyzer Page:
  - Upload Data: Instructions and upload button for analyzing medical reports.
  - Detailed Information: How the medical report analyser works and unlocks critical information.
  - Health Goals Section: Various health-related issues and factors.
- **Profile Page:**
  - Medical Information: User's medical data.
  - General Information: User's personal details.
- **Contact Us Page:**
  - **Contact Information:** Details for user queries and feedback.
- **Other Pages:**
  - **Step Tracker Page:** Daily, weekly, and monthly data with summary graphs.
  - **Heartbeat Rate Page:** Daily, weekly, and monthly data with summary graphs.

- **Sleep Hours Page:** Daily, weekly, and monthly data with summary graphs.
- **Stress Level Page:** Daily, weekly, and monthly data with summary graphs.
- **Calory Burn Page:** Daily, weekly, and monthly data with summary graphs.
- **Weight and Height Page:** Daily, weekly, and monthly data with summary graphs.
- **Blood Oxygen Page:** Daily, weekly, and monthly data with summary graphs.

5.3 NON- FUNCTIONAL REQUIREMENTS

Non-Functional Requirements describe the system's qualities and constraints, such as performance, security, usability, and reliability, which are essential for the overall user experience and system effectiveness.

A. Performance

- Fast Response Time**:** All dashboard pages (e.g., Heart Rate, Steps, Sleep Hours) and backend endpoints for user data fetching or report analysis must load in under 2 seconds to ensure a fluid health monitoring experience.
- High Scalability**:** As users continuously upload health data and lab reports, the platform must efficiently scale to support thousands of concurrent users without crashing or slowing down.
- Efficient Data Handling: Health metrics such as calorie burn, blood pressure, and stress levels should be processed and visualized in real time using charts or indicators to give immediate feedback to the user.
- Low Latency Communication: When a user uploads a report or checks their health stats, API calls should return data instantly to maintain an interactive and responsive UI.

B. Security

- End-to-End Data Encryption**:** All sensitive health records, login credentials, and uploaded reports must be encrypted during transmission (HTTPS) and securely stored using encryption at rest, especially given the sensitive nature of medical information.

- Role-Based Access Control (RBAC)**:** Only authenticated users should access full features like personalized analytics and report history. Limited access (e.g., viewing basic health tips) can be provided for guests.

- Secure Authentication**:** Login and sign-up should include secure password hashing, optional 2FA, and token-based (JWT) authentication to protect user sessions.

- Protection Against Attacks: Since health data is sensitive, all input forms (e.g., report upload, manual data entry) should be protected from injection attacks, CSRF, and XSS vulnerabilities.

C. Usability

- Intuitive User Interface**:** The dashboard, health metric pages, and suggestions should follow a clean layout that even non-technical users (like elderly patients) can navigate effortlessly.

- Responsive Design**:** The website must adapt smoothly to different devices (smartphones, tablets, desktops), enabling users to track their health anytime, anywhere.

- Accessibility Features**:** Implement screen-reader compatibility, keyboard navigation, and contrast themes so that users with visual or motor impairments can use the site effectively.

- Guided Interactions: Provide user-friendly modals and tooltips (e.g., explaining what a specific blood pressure value means) to assist users in understanding their health data**.**

D. Reliability

- High Uptime**:** As users might want to check their health status regularly or even urgently, the platform should maintain 99.9% uptime to ensure 24/7 availability.
- Automatic Failover**:** In case the main server fails, backup servers should take over immediately to avoid data loss and service interruption.
- Frequent Backups**:** All user data and uploaded lab reports should be backed up daily to secure cloud storage, ensuring they can be restored in case of accidental deletion or system failure.
- Error Logging and Monitoring**:** Real-time backend monitoring should be implemented to capture and resolve issues like failed logins, API errors, or broken components.

E. Maintainability

- Modular Codebase**:** Each module (dashboard, step tracker, report analyzer, etc.) should be isolated and independently maintainable to support bug fixing and feature upgrades.
- Documentation**:** Well-documented frontend components and backend APIs will allow easy onboarding for new developers or contributors in future project extensions.
- Code Versioning**:** Git repositories should be used with meaningful commits and branching strategies (e.g., feature branches, main/deploy branches) to manage collaborative development.
- Testing Coverage**:** Use unit and integration tests especially for the report analysis logic and user authentication to avoid introducing bugs during updates.

F. Scalability

- Database Optimization**:** Health data (like step count logs, report results, stress levels) should be stored using indexed collections or relational tables to ensure fast queries.

- Modular Backend**:** For better scaling, you can break functionalities like "User Health Summary", "Lab Report Processing", and "Doctor Suggestions" into microservices or separate modules**.**

- Horizontal Scaling**:** As more users sign up and upload data, your application should be able to handle increased load by adding more servers or containers.

- Caching Strategies**:** Use Redis or like cache frequently accessed data (e.g., user profile, daily step count) to reduce server load and enhance response speed.

G. Compliance & Legal Considerations

- Data Privacy Laws: Since your site handles personal health records, it must align with privacy regulations like HIPAA (if deployed in the US) **or** India's Digital Personal Data Protection Act (DPDP Act) to ensure user rights and consent.

- User Consent: Before uploading reports or tracking personal data, users must accept the privacy policy and consent to data usage via clear checkboxes or modals.

- Data Retention Policy**:** Define how long user health data and uploaded reports are stored (e.g., 2 years), and provide a mechanism for users to delete their data upon request.

- Audit Trails**:** Maintain logs of who accessed, modified, or deleted health records to maintain transparency and accountability.

- Terms & Conditions and Privacy Policy**:** Make both documents available on the website footer and ensure users acknowledge them during account creation.

5.4 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware and Software Requirements specify the physical and technological infrastructure needed to develop, deploy, and maintain the Health Tracker Website.

5.4.1 Hardware Requirement

Server Requirements (For Hosting the Website):

- **High-Performance-Server**:
  Required to handle concurrent user access, real-time health data analytics (like heart rate, blood pressure, calorie burn), and report uploads. A multi-core CPU (e.g., Intel Xeon or AMD EPYC), 16GB+ RAM, and SSD storage are ideal.

- **Storage**:
  Minimum 1TB SSD storage to efficiently manage user data, medical reports, images, and daily health logs. Scalable cloud storage (AWS S3, Google Cloud Storage) is recommended for flexibility.

- **Network-Infrastructure**:
  A reliable internet connection (1 Gbps or more bandwidth) to

support fast data transfer, remote accessibility, and report visualization without lag.

User Device Requirements (End Users):

- **Device**:
  Smartphone, tablet, or desktop with modern browsers like Chrome, Firefox, or Safari.

- **Internet-Connection**:
  Minimum 4G connection or broadband for smooth user experience when uploading reports or interacting with the dashboard.

- **Browser-Compatibility**:
  The site must support latest versions of Chrome, Firefox, Safari, and Edge.
- **Developer Requirements (Project Team):**
- **Development-Machine**:
  - Laptop/PC with at least:
  - 8GB RAM (16GB recommended)
  - Multi-core processor (i5 or better)
  - SSD storage (for faster local development)
- **Testing Devices**:
  Access to multiple screen sizes and devices (mobile, tablet, desktop) for responsive testing.
- **Local Server Environment**:
  Docker or local Node.js + MongoDB instance for backend testing and development.

5.4.1.1.1.1 Software Requirement

- For the Health Tracker Website (System Software):

- Operating System-Compatibility:
  Backend and frontend development should be compatible with:
- Windows 10+
- macOS Monterey+
- Popular Linux distributions (Ubuntu, Debian)
- Frontend Technologies:
- HTML5, CSS3, JavaScript (ES6+)
- React.js for component-based user interfaces
- Tailwind CSS for responsive design
- Backend Technologies:
- Node.js with Express.js for API creation and server-side logic
- Optionally integrate Python for AI-based health recommendations

Database:

- MongoDB Atlas for scalable cloud-based NoSQL database to manage user profiles, health stats, and report metadata

Authentication and Security:

- JSON Web Tokens (JWT), encrypt for password hashing.

- Middleware for secure routes and access control

5.4.2 Server Hosting Requirement

Server Hosting Requirements outline the necessary infrastructure and services needed to host the Health Tracker Website and ensure its availability and performance.

- **Hosting Provider:** Reliable hosting service with high uptime and security features.

- **Domain Name:** Secure a relevant domain name for the website.
- **SSL Certificate:** Ensure secure communication between the server and users.
- **Backup and Recovery:** Regular backups and a recovery plan in case of data loss.

CHAPTER 6

SYSTEM ANALYSIS

- **Need of Health Tracker**
- **Study of Current System**
- **Problems in Current system**
- **Requirement of New System**
- **Process Model**
- **Feasibility Study**
- **Features**

6.1 NEED OF HEALTH TRACKER

In today’s fast-paced digital world, maintaining a healthy lifestyle has become increasingly challenging due to sedentary habits, irregular routines, and lack of timely medical intervention. With the rise of lifestyle-related diseases such as diabetes, hypertension, and stress disorders, it has become imperative to adopt proactive health monitoring practices. This growing demand for continuous health awareness and self-monitoring gives rise to the need for a Health Tracker Website — a centralized digital platform that empowers individuals to track, manage, and analyze their health parameters in real time.

A. Rising Health Concerns and Lifestyle Diseases

The increasing prevalence of chronic diseases and health complications due to modern lifestyle choices has highlighted the importance of regular health monitoring. According to global health studies, conditions like high blood pressure, elevated stress levels, poor sleep quality, and diabetes are becoming more common, even among younger populations. A health tracker website helps in identifying early warning signs and trends through consistent tracking, allowing users to take preventive actions.

B. Personalized Health Management

Every individual’s health journey is unique. A one-size-fits-all approach often falls short when it comes to addressing personalized wellness goals. The Health Tracker Website provides tailored insights based on each user's data — including heart rate, step count, sleep hours, calorie burn, stress levels, blood pressure, and blood sugar levels. This customization encourages users to take ownership of their health and follow routines that align with their specific needs and medical conditions.

C. Integration of Technology in Healthcare

With the integration of smart technologies like wearables and IoT devices, it is now possible to collect health-related data in real-time. The Health Tracker Website utilizes these data streams to provide live insights and meaningful analytics. This promotes a shift from reactive to preventive healthcare, reducing hospital visits and long-term treatment costs. Furthermore, by leveraging machine learning models and data analysis techniques, the platform can offer predictive insights and early alerts.

D. Accessibility and Convenience

Unlike traditional methods of tracking health through physical records or standalone apps, a web-based platform offers seamless accessibility from any device connected to the internet. Whether on mobile, tablet, or desktop, users can log in to view their health data, upload reports, and access personalized recommendations anytime, anywhere. This enhances user engagement and promotes consistent health monitoring practices.

E. Secure Storage and Analysis of Medical Reports

The platform allows users to upload and analyze lab reports securely. It extracts essential information and provides a summary or recommendation based on the report's data. This simplifies the process for users who may not understand complex medical terminology and bridges the gap between clinical data and user comprehension. Secure authentication and encryption ensure that users' sensitive medical information remains private and protected.

F. Real-Time Feedback and Recommendations

One of the key features of the Health Tracker Website is its ability to provide real-time suggestions based on the user's health data. For example, if a user's stress levels are consistently high, the platform might recommend meditation practices, sleep improvement strategies, or suggest consulting a healthcare provider. These actionable insights contribute to a holistic approach to health and wellness.

G. Health Awareness and Preventive Care

Educating users about their health is just as important as monitoring it. The Health Tracker Website includes informational content, usage guidelines, and health tips that raise awareness about wellness and encourage users to make healthier lifestyle choices. The platform acts as a virtual companion, guiding users toward sustained health improvement through knowledge and regular engagement.

6.2 STUDY OF CURRENT SYSTEM

In the present healthcare landscape, multiple standalone applications and devices exist to monitor specific aspects of health — such as fitness apps for tracking steps, smartwatches for heart rate, or glucose meters for blood sugar. However, these solutions often operate in

silos, lack integration, and do not offer a unified interface for holistic health monitoring. Most of them require users to manually input data, lack real-time analysis, or fail to provide actionable insights and recommendations.

Furthermore, many users struggle with fragmented data across different apps and are often unable to interpret medical reports effectively. As a result, timely medical decisions are delayed, and users miss opportunities for preventive care and early intervention.

This disjointed ecosystem of health tools indicates a strong need for a centralized, intelligent, and user-friendly system — which is precisely what the Health Tracker Website aims to solve.

The traditional and existing digital health monitoring systems mainly rely on isolated tools or apps, such as

- Fitness Trackers & Wearables: Devices like smartwatches that monitor step count, heart rate, and sleep patterns.
- Lab Portals: Users upload their test reports to hospital portals, but these portals do not offer personalized interpretation or suggestions.
- Manual Journaling Apps: Apps where users enter their calorie intake, stress levels, and symptoms manually — which can be tedious and inaccurate.
- Basic Health Dashboards: Some websites offer basic dashboards but lack user-specific recommendations, interactivity, or real-time feedback.

These systems face limitations like:

- Lack of integration between various health parameters.
- Poor user experience for non-tech-savvy individuals.
- Minimal focus on preventive care or intelligent health recommendations.
- No centralized platform that covers all major health metrics in one place.

- **Features of the Current System (Health Tracker Website**

- The Health Tracker Website is designed to provide users with a comprehensive yet user-friendly platform to manage and monitor their health. Focusing on manual input, secure report uploads, and AI-powered health insights, the system ensures a holistic health tracking experience. While the platform does not integrate with wearable devices or use live data visualization, it offers essential tools to help users make informed decisions about their health through structured, browser-based access.

1. Step Tracking

- Users can manually enter their daily step count or upload data sourced from external applications. This allows them to consistently track physical activity and stay on top of daily movement goals.

2. Heart Rate Monitoring

- The system enables users to input their heart rate values, either as single readings or trends over time, obtained from home-use monitors or health check-ups. These readings are stored securely and help users monitor cardiovascular patterns.

3. Sleep Analysis

- Users can log their sleep timings, duration, and quality manually. The system records this data and maintains a historical view to help users recognize sleeping patterns and improve sleep hygiene.

4. Calorie Burn Estimation

- The platform provides an option to enter physical activities manually (like walking, running, yoga, etc.). Based on the duration and type of activity, an estimated calorie burn is calculated using predefined formulas.

5. Report Upload Functionality

    - Users can upload their medical reports (PDF or image format) such as blood tests, sugar levels, ECG, etc. These reports are stored securely and used by the system's backend to offer personalized feedback or alerts.

6. Health Dashboard (Static Overview)
    - A clean, static dashboard displays key health metrics like step count, heart rate, calorie burn, and sleep history. This allows users to view their progress at a glance, without advanced data visualizations or graphs.
7. Basic Health Records Management
    - Users can enter and maintain basic health profile information like age, gender, height, weight, medical history, allergies, and chronic conditions. This personalization helps the system tailor health suggestions appropriately.
8. Lab Report Checker with AI Feedback
    - When users upload lab reports, the system extracts key health indicators using OCR and compares them against healthy thresholds. Based on this, it provides simplified summaries and suggestions, making lab results easy to understand for non-medical users.
9. Stress Level Monitoring
    - Users can input their daily stress levels based on how they feel throughout the day (using options like low, moderate, or high stress). The system keeps a log and, if needed, recommends relaxation exercises or breaks through simple suggestions.

10. Blood Oxygen Level Tracking

    - The system allows manual input of SpO_2 (blood oxygen saturation) levels typically measured by a fingertip pulse oximeter. If abnormal

values are detected (e.g., below 95%), the system highlights this and advises the user to rest, hydrate, or seek medical attention.

11. Diabetes Analysis

- Users can input their fasting, random, or postprandial blood sugar readings, which the system stores and analyzes. The platform:
- Identifies trends in blood glucose levels over time.
- Flags critical or fluctuating values.
- Offers lifestyle recommendations such as dietary changes, suggested physical activity, and hydration advice.
- Reminds users to take medications or perform routine checks.

12. Blood Pressure Analysis

- Users are prompted to enter systolic and diastolic blood pressure readings. The system:
- Categorizes readings (normal, elevated, stage 1/2 hypertension).
- Issues alerts for abnormal readings.
- Offers suggestions such as salt reduction, hydration tips, stress management, and breathing exercises.
- Stores historical data to observe progress and improvement over time.

13. Personalized Recommendations

- Across all features—be it step tracking, sleep, blood sugar, or BP—the system uses basic rule-based AI logic to offer daily health tips based on user input. These include:
    - "Drink 8 glasses of water today."
    - "Try a 20-minute walk to improve circulation."
    - "Your sleep pattern suggests fatigue—consider adjusting your bedtime."
    - "Blood sugar is fluctuating—reduce refined carbs in your meals."

14. User Authentication and Access Control

- To ensure privacy, every user must register and log in to access the full feature set. Guest users may have limited access. All personal data, reports, and health entries are stored securely in a protected environment with authentication mechanisms.

6.3 PROBLEMS IN CURRENT SYSTEM

While the Health Tracker Website provides essential health monitoring features and a user-friendly interface, several limitations exist in its current version. These limitations hinder the full potential of personalized healthcare and digital health management. Identifying these issues is crucial for planning future upgrades and enhancing the overall user experience. Below are the key problems in the current system:

1. Lack of Real-Time Data Integration

The current system relies entirely on manual input from users. It does not support real-time data fetching from wearable devices such as smartwatches, fitness bands, or health monitors. As a result:

- Continuous health monitoring is not possible.
- There is a higher chance of data inaccuracy due to manual entry.
- Users may find it tedious to regularly input data, leading to inconsistency.

2. No Data Visualization or Trend Analytics

The health tracker lacks advanced visualization tools such as interactive graphs, trend charts, and dynamic dashboards. This creates several problems:

- Users cannot easily identify changes or patterns in their health over time.
- There is no visual feedback to motivate improvement or track progress.
- Medical professionals cannot quickly interpret user data when presented visually.

3. Limited AI Capabilities

Although the system includes basic recommendations, it does not have sophisticated AI or machine learning features for deeper insights. This means:

- No personalized predictive health alerts based on long-term data.
- The system cannot detect potential health risks early.
- Recommendations are static and not adaptive to a user's changing health profile.

4. Manual Data Entry is Time-Consuming and Error-Prone

Since users must enter each health parameter manually:

- There is a high probability of incorrect or missed data.
- Users may find the process repetitive and disengaging.
- It reduces the efficiency and user-friendliness of the platform.

5. No Emergency Alert System

In the case of abnormal readings (e.g., high blood pressure or low oxygen level), the system does not:

- Send real-time alerts or emergency notifications.
- Provide any emergency contact or support options. This could delay critical medical attention.

6. Absence of Community or Health Professional Integration

The current system is limited to individual users without any provision for:

- Doctor consultations or feedback based on tracked health data.
- Support groups or communities for users dealing with similar health issues (e.g., diabetes, hypertension).
- Remote healthcare monitoring by assigned medical professionals.

7. Incomplete Health Coverage

While the system monitors essential parameters like steps, heart rate, and sleep, it still lacks:

- Mental health tracking beyond stress input.
- Nutritional tracking such as meal logs or water intake.
- Immunization records or medication schedules.

6.4 REQUIREMENT OF NEW SYSTEM

As health monitoring becomes an essential part of everyday life, especially in a post-pandemic world, there is a growing need for a more intelligent, efficient, and user-centric health tracking system. While the current system offers basic features like step tracking, manual heart rate input, and lab report uploads, it lacks real-time monitoring, personalized insights, and AI-driven analytics. To bridge these gaps and improve user health outcomes, a new and improved version of the Health Tracker Website is required. The following outlines the core requirements of the new system:

1. **Integration with Wearable Devices and IoT Sensors**
    - To enable real-time, automated health tracking
    - The system must support integration with smartwatches, fitness bands, pulse oximeters, and other wearable devices.
    - Automatic syncing of data such as steps, heart rate, sleep, and blood oxygen levels should be enabled.
    - Users should no longer need to manually enter routine data.
2. **Advanced Data Visualization and Trend Analysis**
    - To enhance data interpretation and user engagement:
    - The system must include dynamic charts, graphs, and visual timelines.
    - Users should be able to view historical data trends (daily, weekly, monthly).
    - Visualization of metrics like blood pressure, calorie burn, stress levels, and diabetes risk will help users understand progress.
3. **AI-Based Personalized Health Recommendations**

    To provide smart, adaptive guidance:

- The system should analyze user data using machine learning to offer personalized advice.
- Predictive insights for conditions such as hypertension or diabetes based on historical patterns.
- Suggest lifestyle changes, exercises, and diet plans tailored to individual health metrics.

4. Automated Health Risk Prediction and Alert System

To detect risks early and enhance user safety:

- The system must evaluate user input and generate alerts for critical health conditions (e.g., low SpO2, abnormal heart rate).
- Emergency alert features should notify users or their emergency contacts when concerning patterns are detected.
- Risk scores for chronic diseases should be displayed clearly.

5. Improved User Experience and Accessibility

To ensure inclusivity and ease of use:

- Interfaces must be optimized for all age groups, especially elderly users.
- Accessibility features like voice commands, larger font sizes, and screen reader support should be integrated.
- Multilingual support could enhance user reach.

6. Interactive and Personalized Health Dashboard

To provide a central overview of all health activities:

- A real-time dashboard must present updated metrics from all tracked categories.
- Users should be able to customize dashboard views based on priorities (e.g., stress focus, blood pressure overview).
- Include motivational badges, health score meters, and daily summaries.

7. **Enhanced Lab Report Analysis**

   To transform static reports into actionable data:

   - The system should not only accept report uploads but also analyze and interpret lab values using predefined health standards.
   - Highlight abnormal values, suggest possible causes, and recommend follow-ups or lifestyle tips.
   - Store medical history in a searchable, structured format for future comparisons.

8. **Mental Health and Stress Monitoring**

   To offer holistic health tracking:

   - Include tools for stress and mood tracking via self-assessment or device data (e.g., heart rate variability).
   - Provide calming techniques, stress-relief exercises, and mental wellness resources.
   - Correlate mental and physical health metrics for deeper insights.

9. **Goal-Setting and Progress Monitoring**

   To motivate users and encourage consistency:

   - Allow users to set health goals (e.g., sleep 8 hours, walk 10,000 steps).
   - Provide daily/weekly feedback on goal achievement.
   - Introduce progress reports and suggestions to meet targets more effectively.

6.5 PROCESS MODEL

For the development of the Health Tracker Website, the Agile software development methodology was chosen due to its flexibility, iterative nature, and ability to accommodate evolving user requirements. Agile promotes continuous feedback, adaptability, and incremental delivery—making it ideal for dynamic, user-centered health tracking applications where changes and improvements are common based on user interactions and health data analytics.

Why Agile?

User-centric: Health tracking features need to adapt based on user preferences, feedback, and evolving health technology trends.
Faster feedback loop: Each feature (e.g., blood pressure tracking, step counter, stress analysis) can be developed, tested, and improved in quick cycles.
Continuous delivery: Important modules like lab report upload, dashboard views, and AI-based suggestions can be rolled out in phases.
Team collaboration: Agile ensures constant communication between developers, designers, testers, and stakeholders.
Agile methodology divides the development process into multiple iterations or sprints, each typically lasting 1–3 weeks. Here are the detailed steps:

1. Requirement Gathering and Analysis

- In the initial phase, stakeholders (developers, project guide, end-users) collaborate to gather core requirements.
- Functional needs (like step tracking, report upload, etc.) and non-functional needs (security, performance, UI design) are documented.
- User personas and use cases are identified, especially considering different types of users like regular users and doctors.

2. Sprint Planning

- The project is divided into small modules or **user stories**.
- A backlog is created, listing tasks such as “Create user registration page,” “Implement step tracker,” “Enable report upload.”
- Tasks are prioritized, and a plan is made for what will be achieved in the next sprint.

3. Design

- During this phase, the wireframes and UI/UX designs of the website are developed.
- Decisions are made about the frontend (React.js), backend (Node.js/Express.js), and database (MongoDB).
- Each feature (like sleep tracking, stress level input) has a dedicated design module created using a minimalistic, accessible approach.

4. Development (Sprint Execution)

- Development takes place in sprints. At the end of each sprint, a potentially shippable product increment is delivered.
- Features are coded, integrated, and tested iteratively. For example:
    - Sprint 1: User login, registration, and authentication
    - Sprint 2: Dashboard setup and manual data entry pages
    - Sprint 3: Report upload and basic AI-based analysis
    - Sprint 4: Stress, sleep, blood pressure modules
    - Sprint 5: Personalized health recommendation system

5. Testing

- Each module is tested immediately after development using unit testing**,** integration testing**, and** user acceptance testing (UAT)**.**
- Bugs or issues found are recorded in a bug tracker and fixed before the sprint ends.
- Ensures the website is responsive, secure, and functional across devices.

6. Deployment

- After passing all tests in a sprint, the code is pushed to the deployment environment.
- Partial deployment is done to production (or demo servers) so stakeholders can see live progress.
- Continuous integration/continuous deployment (CI/CD) practices are applied for smooth updates.

7. Review and Feedback (Sprint Review)

- At the end of every sprint, the working module is demonstrated to stakeholders or mentors.
- Feedback is gathered on what's working well, what needs improvement, or what should be added next.
- Suggestions are added to the backlog for future sprints.

8. Retrospective

- The team discusses:
  - What went well?
  - What could have been better?
  - What actions will be taken to improve the next sprint?
- Helps the team evolve their working process and resolve any blockers.

9. Maintenance and Continuous Improvement

- Even after full deployment, Agile supports continuous improvements.
- New features (like device integration, advanced charts, or multilingual support) can be added in future sprints.
- Bugs, user requests, or system updates are addressed in short feedback loops.

6.6 FEASIBILITY STUDY

6.6.1 Technical Feasibility

Technical feasibility assesses whether the existing technical resources — such as hardware, software, and skilled personnel — are sufficient to support the development and deployment of the system.

For the Health Tracker Website:

The technologies used, including HTML, CSS, JavaScript, React.js for frontend, and Node.js with MongoDB for backend, are all widely supported and easily accessible.

The project does not require specialized hardware or high-end computation, making it suitable for standard web servers and hosting environments.

The development team possesses the necessary technical knowledge in web development, database integration, and basic machine learning for lab report analysis.

The system is browser-accessible, requiring only internet connectivity from the user's side — no installation needed.

6.6.2 Operational Feasibility

Operational feasibility evaluates whether the proposed system will function effectively within the current environment and if end users will accept and adopt it.

For the Health Tracker Website:

The system is designed with a simple, user-friendly interface that is accessible to all age groups, including elderly users.

Manual data input allows flexibility for users who do not own wearables or fitness devices.

The project includes personalized recommendations, lab report interpretation, and reminders, which increases user engagement and improves health outcomes.

Features like step tracking, sleep tracking, stress monitoring, BP and diabetes analysis offer direct value to the user.

6.6.3 Economic Feasibility

Economic feasibility analyzes the cost-benefit aspect of the system, ensuring that the project's financial investment is justified by its benefits.

For the Health Tracker Website:

The development cost is relatively low as it relies on open-source technologies and can be hosted on affordable cloud platforms.

Once deployed, maintenance costs are minimal since the system has no third-party API costs and uses basic CRUD operations and AI-based logic.

The benefits — such as early detection of health risks, AI-based lab analysis, personalized recommendations, and increased health awareness — far outweigh the initial development and hosting expenses.

6.6.4 Scheduling Feasibility

Definition:

Scheduling feasibility checks whether the project can be completed in the required or estimated timeframe with the available resources.

For the Health Tracker Website:

The system was planned and developed using the Agile methodology, allowing the team to divide the project into smaller sprints and prioritize features based on importance and complexity.

Most of the core features like user authentication, dashboard, health record tracking, and lab report checking can be developed within a realistic 2–3 month window, depending on team size.

Since it doesn't require extensive integrations or complex AI modelling at the beginning, time constraints can be effectively managed.

6.7 FEATURES

Features refer to the distinct functionalities and capabilities that define what a system can do for its users. For the Health Tracker Website, features represent the full set of modules, user interactions, backend services, and AI integrations that provide a seamless, intelligent, and personalized health monitoring experience.

These features cover both functional aspects (like authentication, report uploads, and trackers) and non-functional aspects (like responsiveness, security, and performance). This section highlights all the major and minor features developed during the project lifecycle.

A. User Authentication & Access Control

- User registration (sign-up) and secure login system.
- JWT-based authentication with session storage.
- Password hashing and validation.
- Role-based access control (new vs registered users).
- Form validations with error prompts.

B. User Profile Management

- Input of user data: name, gender, DOB, email, address, phone number.
- Health metadata: height, weight, blood group.
- Wake-up and bedtime scheduling.
- Emergency contact information.
- Editable profile interface with live validation.

C. Dashboard & Health Overview

- Modular dashboard displaying summarized health stats.
- Live updates on:
    - Step count
    - Calories burned
    - Stress level
    - Sleep hours
    - Heartbeat rate
    - Blood pressure
    - Blood sugar level
- Visual trend charts using Chart.js / Recharts.
- "Last seen" indicator and quick access to insights.

D. Health Data Trackers

Each module allows real-time entry and backend logging of:

- Step Tracker
- Calorie Burn Monitor
- Heart Rate Tracker
- Sleep Hours Logger
- Stress Level Monitor
- Blood Pressure Monitor (Systolic/Diastolic)
- Blood Oxygen Level Tracker
- Diabetes Monitoring (BMI, Sugar Levels)

E. Health Goals Management

- Goal creation: steps, sleep, weight, stress, etc.
- Set deadlines and target values.
- Visual progress indicators (bar, status color codes).
- Goal status: In Progress, Achieved, Missed.

F. AI-Powered Medical Report Analyze

- Upload PDF lab reports for analysis.

- Integration with OpenAI API to extract insights.
- AI-generated content includes:
    - Medical conditions detected
    - Key markers flagged
    - Suggestions for improvement
- Report sections: “Conditions Detected,” “Markers to Monitor,” and “Suggestions.”

G. Data Visualization & Analytical Insights

- Bar, line, and pie charts to visualize:
    - Stress trends
    - Sleep patterns
    - Blood sugar levels
- Graphs with comparative views and summaries.

H. Upload Module

- Upload .json smartwatch data or lab reports.
- Validates file size and type.
- Pre-upload file preview and confirmation.
- Submission handled via Axios and FormData.
- Redirects to analysis page post-upload.

I. Contact & Feedback System

- Contact form with fields: Name, Email, Subject, Message.
- Integrated with EmailJS to send messages directly to admin.
- Real-time status updates and loader animations.

J. Static Informational Pages

- Home Page – Introduction to the website.
- How It Works – Guide on how to use each module.
- Critical Information – Explanation of health terms and tracker usage

K. Backend System Feature

- Modular RESTful APIs with Express.js.
- MongoDB schemas for:
  - Users
  - Health data
  - Reports
  - Goals
- Input validation and secure routing.
- Structured API controllers for maintainability.

L. Technical Implementation Highlights

- Use of React Hooks (useState, useEffect, useContext).
- Route protection via react-router-dom.
- Real-time toast alerts using react-toastify.
- Responsive design with Flexbox and CSS Grid.
- Axios-based frontend-backend communication.

M. Testing Strategy

- Manual unit testing for React components.
- Backend API testing using Postman.
- UI testing for input validations, responsiveness, and flow.
- Console and network debugging during development.

N. Deployment

- Frontend hosted on Vercel.
- Backend hosted on Render.
- Environment variable management via .env.
- Integration of domain (optional) and HTTPS.

O. Project Documentation

- Data Dictionary for all database tables.
- Class Diagram and ER Diagram for system structure.
- Architecture Diagram showing component interaction.
- Work Breakdown Structure (WBS) and Timeline Chart.
- Agile Sprint Table with task distribution and team roles.

CHAPTER 7

MODULE DESCRIPTION

- **Dashboard Module**
- **Step Tracker Module**
- **Heartbeat Rate Module**
- **Sleep Hours Module**
- **Stress Level Module**
- **Calorie Burn Module**
- **Blood Oxygen Module**
- **Data Analysis Module**
- **Data Upload Module**
- **Medical Report Analyzer Module**
- **User Profile Module**
- **Contact Us Module**
- **Security and Privacy Module**

7.1 DASHBOARD MODULE

The Dashboard Page serves as the central hub and the most crucial component of your Health Tracker Website. It is designed with the primary objective of offering users a comprehensive, real-time snapshot of their overall health and fitness status. This page consolidates all key health parameters and user goals into a single, easily accessible location, enabling users to monitor their wellness journey with convenience and clarity.

A. Purpose and Role in the System

- The Dashboard acts as the first point of interaction after a user logs in, providing a personalized view of their health data.
- It bridges multiple modules of the system such as blood pressure tracking, diabetes analysis, blood oxygen levels, and more — presenting their summaries in a clean, intuitive interface.
- The page is built to deliver instant visual feedback on the user's daily and weekly health performance, promoting self-awareness and motivation to stay on track.
- It transforms raw health data into meaningful, actionable insights, aligning with your project's core goal: to empower users to understand and improve their health through technology**.**
- The dashboard not only simplifies complex medical information but also offers early visibility of potential health issues**,** encouraging preventive care.
- As the most frequently used interface, it ensures that all core functionalities of the Health Tracker Website are integrated, engaging, and easily navigable **—** enhancing the overall effectiveness of your system.

B. Progress Circle (Multi-Layered)

- **What it shows:** A layered circular progress bar displaying:
  - Outer Circle: Calories Burned
  - Middle Circle: Step Count
  - Inner Circle: Workout Duration
- **Goal Tracking:** Dynamically compares real-time values with user-defined goals.

- **Purpose:** Motivates users by showing real-time progress in a compact visual format.

C. Step, Calorie, and Workout Stat Boxes

- **Steps Card:** Shows current step count and user goal.
- **Calories Card:** Displays calories burned today and goal.
- **Workout Duration Card:** Shows time spent on physical activity.
- These cards help users quickly glance at their daily activity metrics**.**

D. Blood Pressure Card

- **What it shows on dashboard:**
  - A **Line Chart** plotting systolic and diastolic pressure over the past 7 days.
  - Tooltip shows exact readings per day.

- **Inside BP Page Features:**

  - Detailed trends (weekly/monthly)
  - Health category (Normal, Elevated, Hypertension)
  - Recommendations (hydration, salt control, doctor consult)

E. Diabetes Card

- **What it shows on dashboard:**
  - A Bar Chart of blood sugar levels (past 7–8 days).
  - Visual comparison of daily sugar readings.
- **Inside Diabetes Page Features:**
  - Sugar type categorization (fasting/postprandial/random)
  - Health condition (Normal/Prediabetic/Diabetic)
  - Suggestions (diet control, exercise tips)

F. Blood Oxygen Card

- **What it shows on dashboard:**
  - A Line Chart displaying blood oxygen levels for the last 7 days.

  - Ideal range (95–100%) is reflected visually.
- **Inside Blood Oxygen Page Features:**
  - Weekly/monthly patterns
  - Alerts for SpO_2 below 94%
  - Suggestions like breathing exercises, air quality awareness

G. Weight and Height Card

- **What it shows on dashboard:**
  - A Line Chart showing both weight and height trends across the week.
  - Dual line graph for direct comparison.
- **Inside Weight Page Features:**
  - BMI calculation
  - Weight gain/loss trends
  - Personalized diet or exercise advice

H. Data Analysis Card

- **What it shows on dashboard:**
  - A simple card encouraging users to "Uncover hidden patterns, trends, and opportunities."
- **Inside Data Analysis Page Features:**
  - Aggregated view of all key metrics (BP, diabetes, oxygen, etc.)
  - AI-generated insights (if integrated)

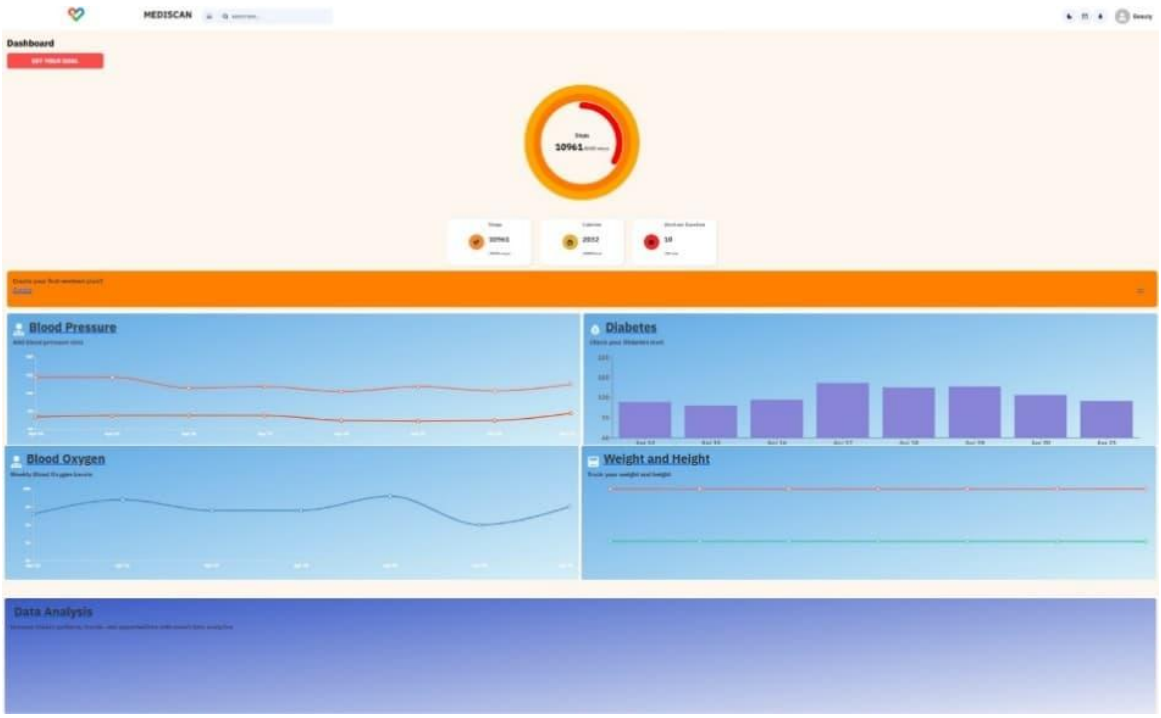

Figure 7.1.1 Dashboard section

7.2 STEP TRACKER MODULE

The Step Tracker Page is a significant sub-module of the Health Tracker Website. It plays a valuable role in monitoring physical activity, which is a foundational element of personal health and fitness.

This page provides users with a dedicated space to manually input and visualize their step count data across different time frames — daily, weekly, and monthly. Walking and general mobility are key to managing weight, blood pressure, blood sugar, and mental well-being, making this page an important component of the user's health journey.

- Acts as a focused tracker for physical movement, specifically steps taken by the user.
- Offers detailed analysis and insights based on user-entered step data.
- Encourages users to build and maintain a more active lifestyle.
- Complements other modules (like BP, Diabetes, Weight) by supporting preventive care through physical activity tracking.

A. Timeframe Switch Buttons (DAY / WEEK / MONTH)

- Users can toggle between **three views**:
  - **Day:** Last 30 days
  - **Week:** Last 7 days
  - **Month:** Last 3 months
- The selected view dynamically updates the chart and summary section.

B. Bar Chart (MUI X Bar Chart)

- Dynamically displays step data for the selected view using a bar chart:
  - X-axis: Date labels (e.g., 21/02 for Day, Feb 21 for Week, Mar 2025 for Month)
  - Y-axis: Step counts
- The height of the bars visually represents activity levels.
- Helps users easily identify active vs. inactive days and overall trends.

C. Summary Card (Steps Statistics)

Displays detailed step metrics based on the current view:

- **Total Steps:** Total recorded steps in the selected time range.
- **Average Steps Per Day:** Helps users assess consistency.
- **Maximum Steps in a Day:** Encourages users to beat their best.
- **Minimum Steps in a Day:** Identifies sedentary days.
- **Days Recorded:** Shows how many days had step data (for accuracy).

D. Informational Card – About Step Tracking

This section educates and motivates users:

- Highlights why walking is important**.**
- Mentions the standard goal of 10,000 steps/day for maintaining health.
- Encourages users to take more steps daily.
- Includes an engaging UI element: arrow icon + "Learn more" prompt.

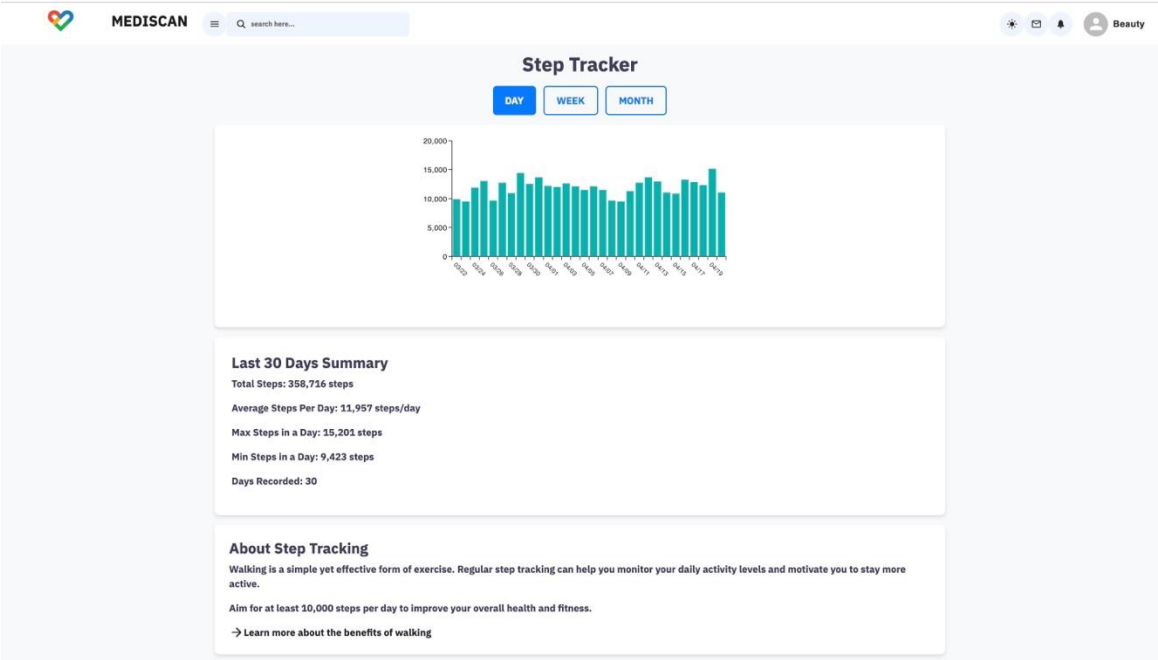

Figure 7.2.1 Daily Step Tracker Section

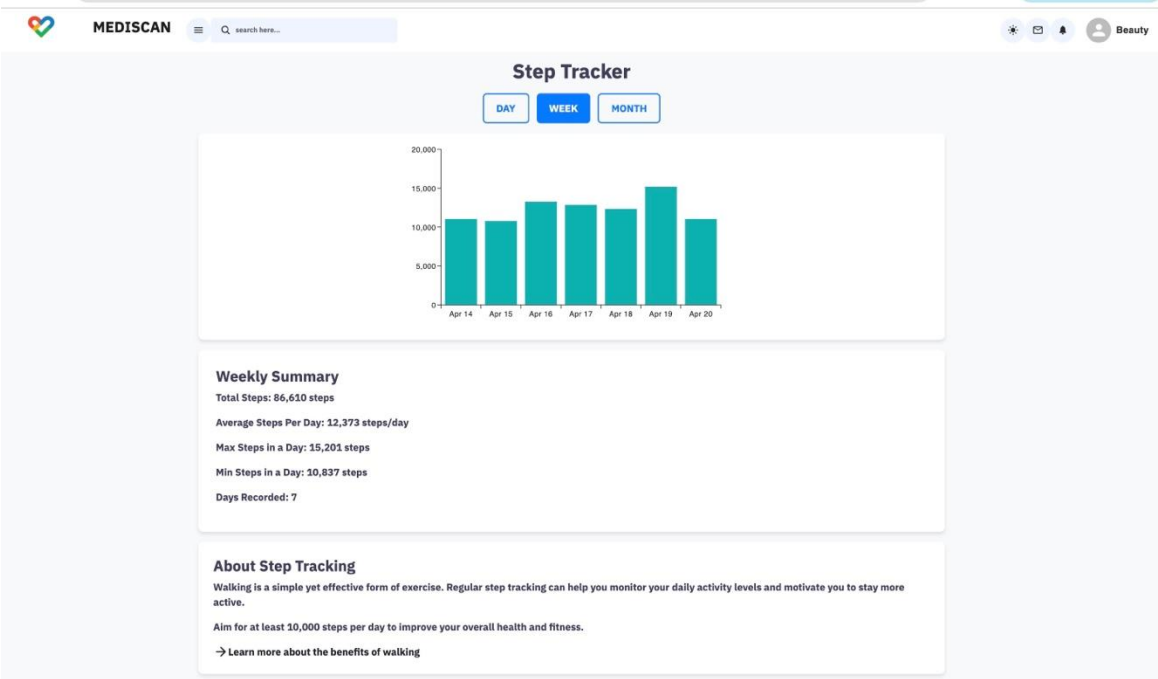

Figure 7.2.2 Weekly Step Tracker Section

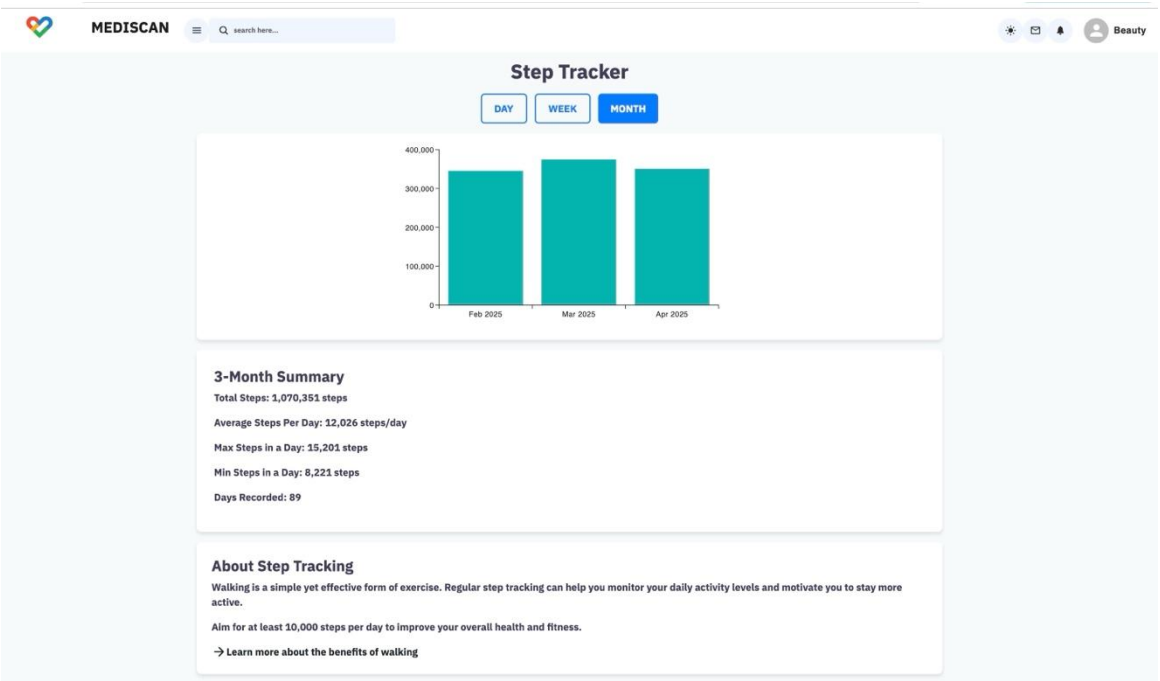

Figure 7.2.3 Monthly Step Tracker Section

7.3 HEARTBEAT RATE MODULE

The Heartbeat Tracker Page is a dedicated sub-module within your Health Tracker Website, designed to allow users to track, monitor, and analyse their heart rate over time**.** While it isn't the main dashboard, this page plays a vital role in monitoring cardiovascular health**,** providing visual insights into the user's average, maximum, and minimum heart rate across different time frames — day, week, and month.

This feature supports the project's mission to enable users to proactively manage their health by understanding heart health trends through clean visualizations, statistics, and educational content.

Purpose of the Page

- To present a detailed breakdown of heart rate data logged by the user.
- To help users visualize their heart activity through interactive bar charts.
- To provide summarized insights like average heart rate, peak values, and consistency.
- To educate users about healthy heart rate ranges and the importance of cardiovascular fitness.

A. Functional Components

1. Time View Toggle (DAY / WEEK / MONTH)

- Three toggle buttons allow users to switch between:
  - DAY View: Last 30 days
  - WEEK View: Last 7 days
  - MONTH View: Last 3 months
- Data updates instantly based on the selected view.
- Visual and textual summaries adjust accordingly.

2. Bar Chart Visualization

- Uses MUI Bar Chart to plot two metrics:
  - Avg BPM (average beats per minute) – Blue bars
  - Max BPM – Red bars
- Dynamic y-axis scaling (up to 180 BPM or more) based on the highest value.
- Data labels:
  - Day: Date (e.g., 21/03)
  - Week: Short format (e.g., Mar 21)
  - Month: Month-Year (e.g., Feb 2025)

Purpose: Help users see how their heart rate fluctuates daily, weekly, or monthly — aiding in trend detection and heart performance awareness.

3. Summary Statistics Card

Displays summarized data based on the current time frame:

- Average Heart Rate – Shows user's average resting or daily BPM.
- Max Heart Rate – Highest recorded BPM.
- Min Heart Rate – Lowest recorded BPM.
- Measurements Recorded – Total entries contributing to the summary.

These insights allow users to monitor cardiovascular performance, detect abnormalities, and track improvements over time.

4. Informational Card – “About Heart Rate”

- Educates the user on heart rate ranges and why it matters:
  - Healthy resting rate: 60–100 BPM.
  - Max heart rate during exercise: ~220 minus user's age.
- Encourages daily monitoring with motivational messaging.
- Styled with icon and soft tone to maintain user engagement.

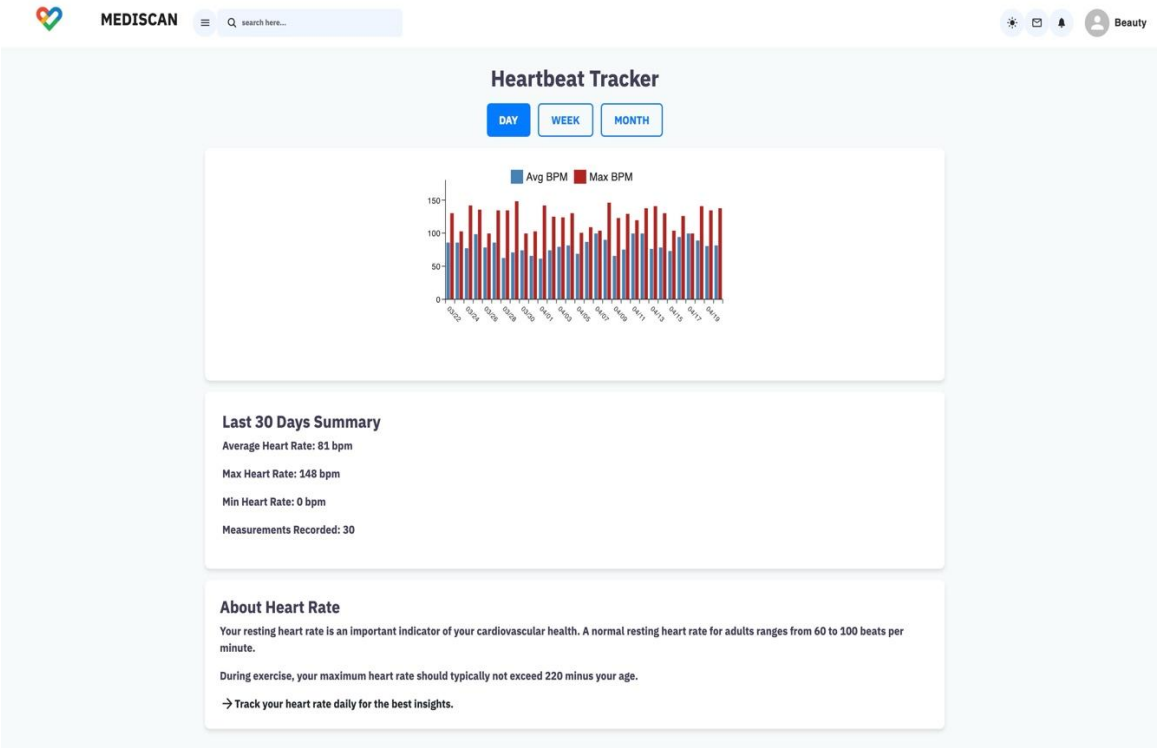

Figure 7.3.1 Daily Heartbeat rate Section

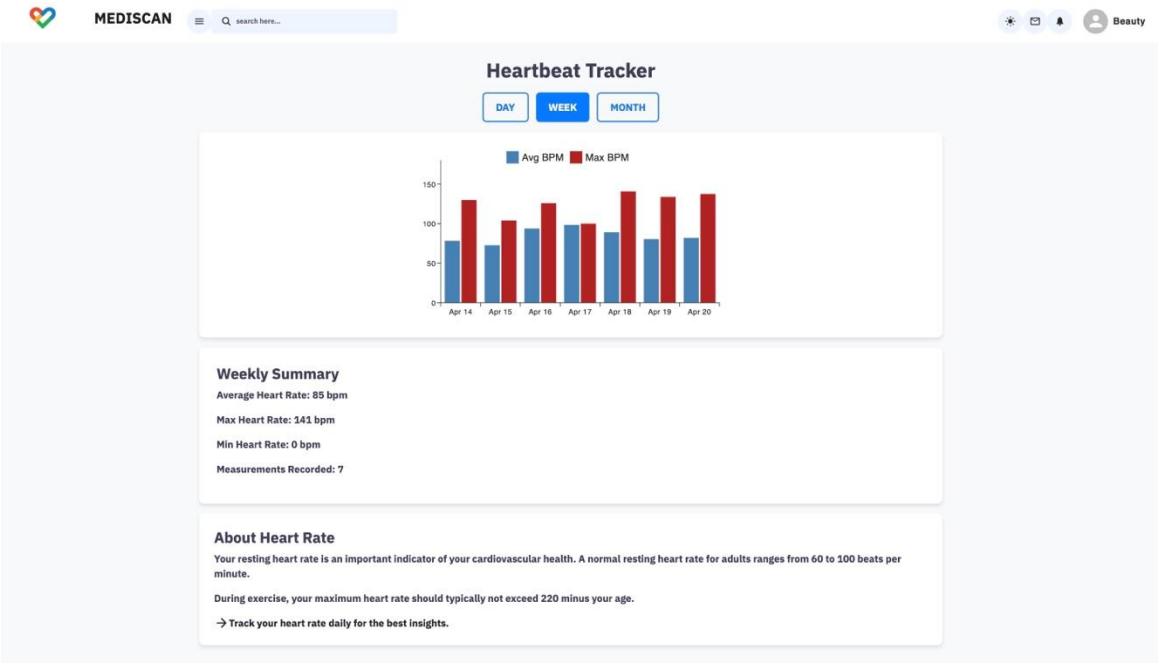

Figure 7.3.2 Weekly Heartbeat rate Section

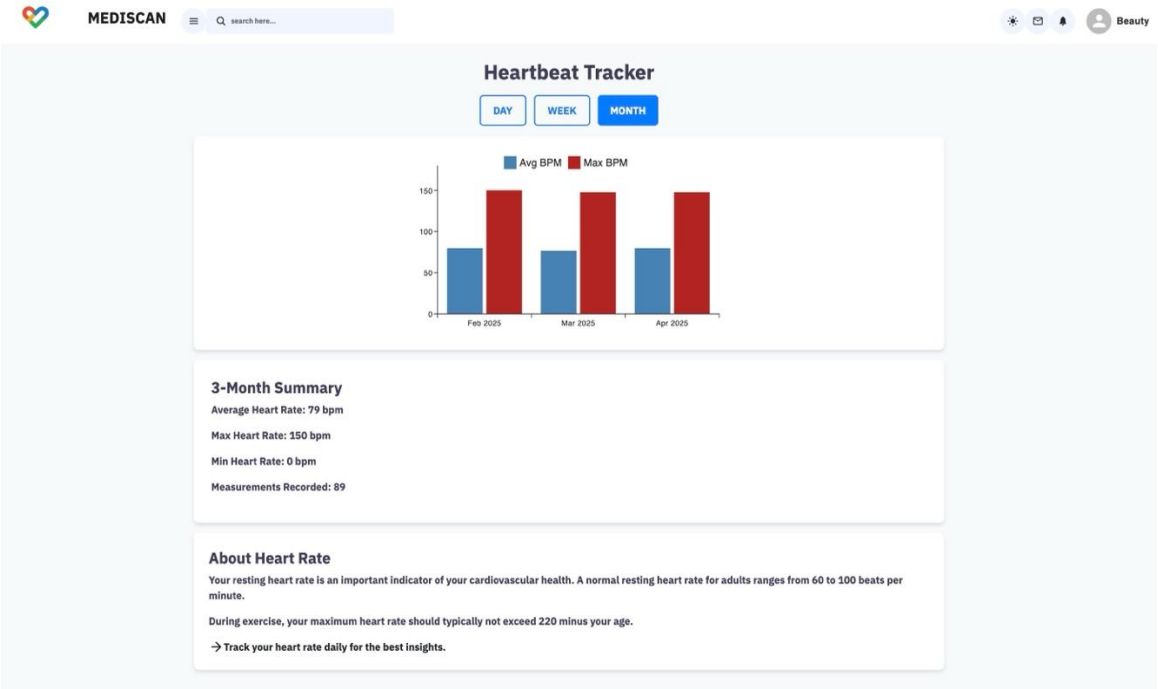

Figure 7.3.3 Monthly Heartbeat rate Section

7.4 SLEEP HOURS MODULE

The Sleep Hours Tracker Page is a specialized sub-module in your Health Tracker Website that provides users with a focused space to track, review, and analyse their sleep patterns.

While not part of the main dashboard, this module holds substantial value in promoting mental clarity, physical health, and emotional well-being**,** all of which are influenced by the quality and consistency of sleep.

By manually recording sleep hours and visualizing them through intuitive graphs and summaries, users are empowered to identify patterns and make informed adjustments to improve their sleep hygiene.

- **Purpose of the Sleep Tracker Page**

- To allow users to log and analyze sleep durations over a range of days.
- To visualize trends through interactive bar charts for Day, Week, and Month views.
- To offer statistical summaries like total, average, minimum, and maximum sleep hours.
- To educate users about the importance of consistent and adequate sleep.
- To support overall health analysis when viewed alongside other modules (e.g., stress, heart rate, BP).

1. Timeframe Toggle (Day / Week / Month)

- Users can switch between:
  - Day View → Shows data for the last 30 days.
  - Week View → Displays the past 7 days.
  - Month View → Aggregates sleep data over the last 3 months.
- Each view dynamically updates both the chart and the statistics summary.

2. Sleep Duration Bar Chart

- A clean, readable bar chart represents daily sleep hours across the selected time range.
- X-axis: Dates or months depending on the view.
- Y-axis: Sleep duration (in hours), scaled dynamically (usually up to 10 hours).
- Users can visually compare active vs. low-sleep periods, making it easier to improve sleeping habits.

3. Sleep Summary Card

Shows key insights depending on the selected time range:

- Total Sleep Hours → Combined hours over the time frame.
- Average Sleep Per Day → Helps track if the user meets recommended 7–9 hours/day.
- Max Sleep in a Day → Highest recorded duration.
- Min Sleep in a Day → Shortest duration, signalling sleep debt.
- Days Recorded → Days where data was available, providing transparency.

4. Informational Card – "About Sleep Hours"

- Offers educational tips on healthy sleep:
  - Reinforces the importance of 7–9 hours of sleep.
  - Explains how sleep supports mental, emotional, and physical health.

5. User Benefits

- Helps users assess whether they're meeting their sleep goals.
- Assists in identifying irregular sleep routines or disruptions.
- Promotes better sleep hygiene through awareness and education.
- Allows users to correlate poor sleep with other health concerns (e.g., fatigue, high stress).

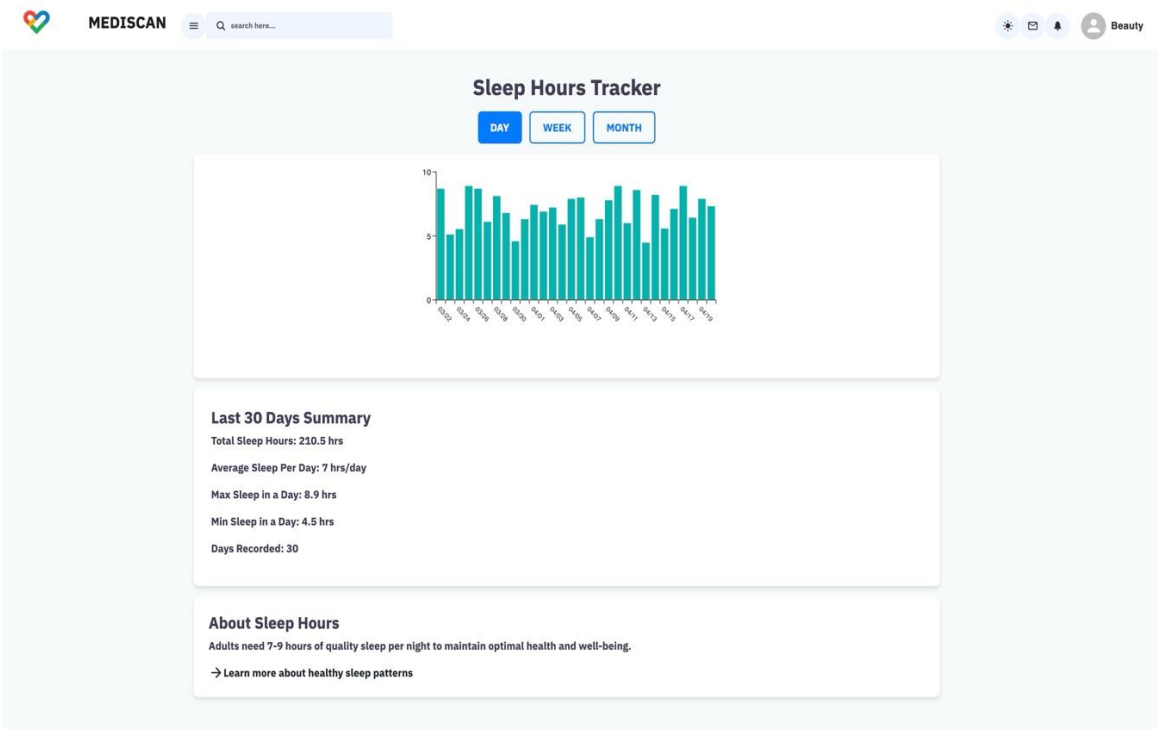

Figure 7.4.1 Daily Sleep Hours Section

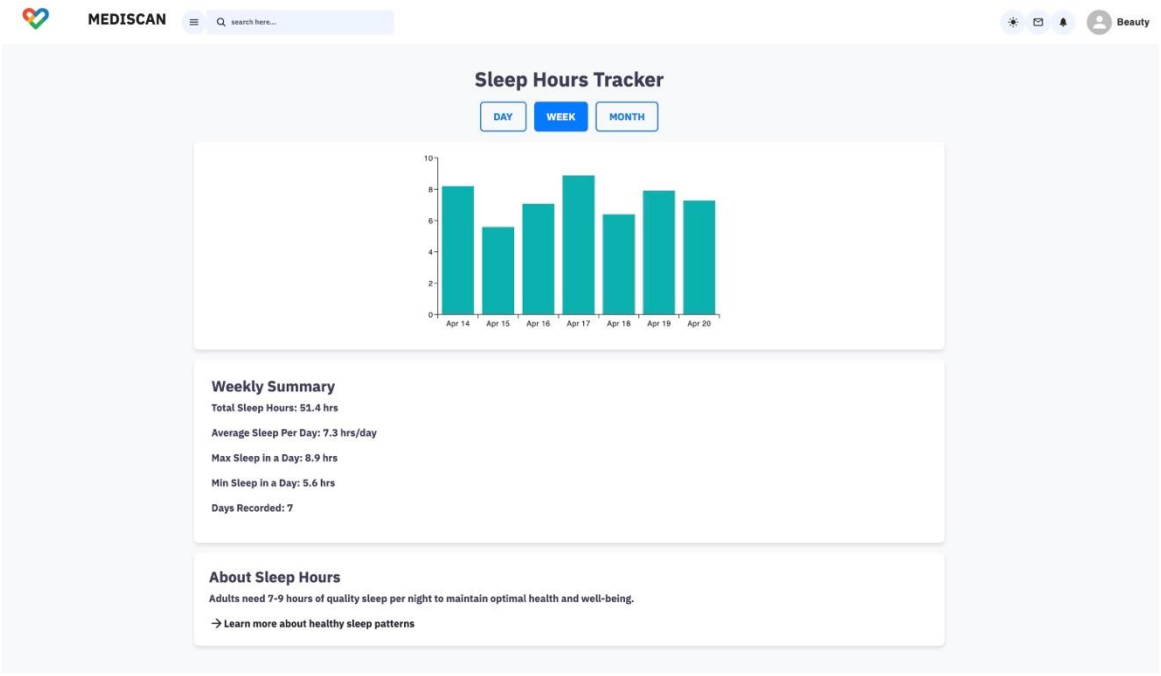

Figure 7.4.2 Weekly Sleep Hours Section

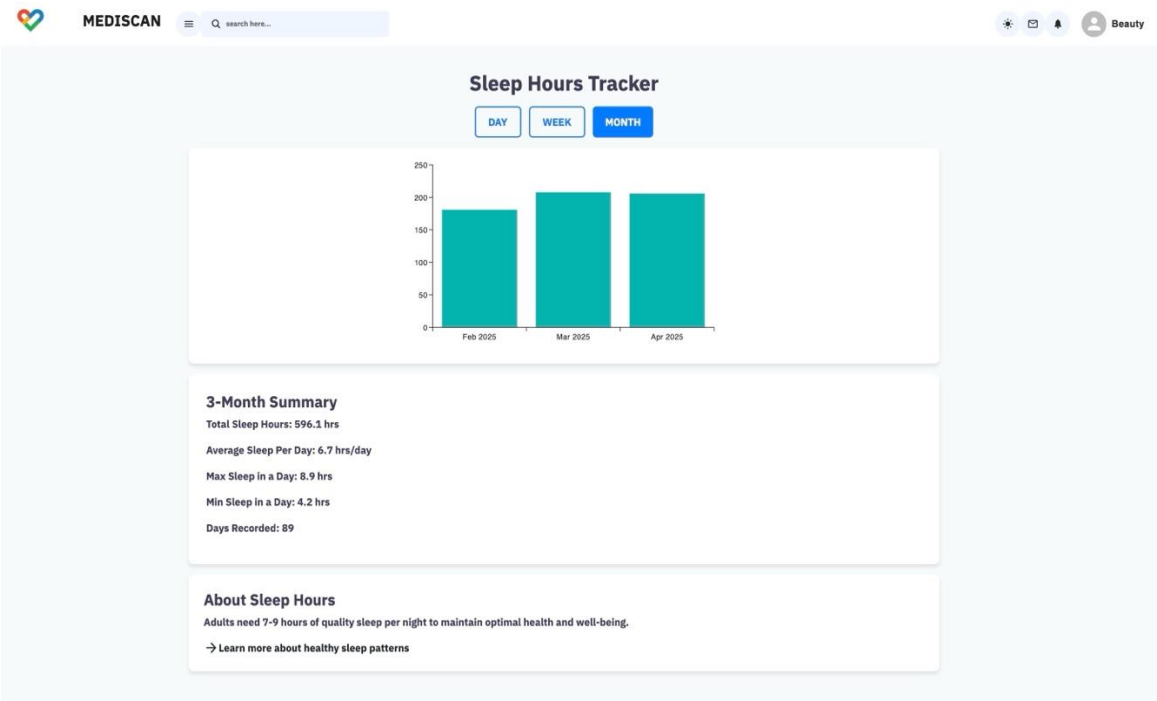

Figure 7.4.3 Monthly Sleep Hours Section

7.5 STRESS LEVEL MODULE

The Stress Level Tracker Page is a specialized sub-module of the Health Tracker Website, designed to help users monitor, track, and analyse their stress levels over a set period. Though not part of the main dashboard, this page plays a crucial role in promoting mental wellness by visualizing fluctuations in stress patterns through intuitive charts and helpful insights.

This feature aligns with the project's mission to provide users with proactive tools for better health management, offering a clear overview of their psychological well-being through data-driven metrics and educational support.

A. Purpose of the Page

- To offer a visual and statistical breakdown of the user's stress level data logged over time.
- To provide interactive bar charts for day, week, and month-wise stress tracking.
- To summarize key insights like average, maximum, and minimum stress levels for better understanding.

- To educate users about stress, its health implications, and the importance of regular self-assessment.

B. Functional Components

1. Time View Toggle (DAY / WEEK / MONTH)

- Three dedicated toggle buttons allow users to switch between:
  - DAY View: Displays data for the last 30 days.
  - WEEK View: Aggregates average stress levels for the past 7 days.
  - MONTH View: Summarizes average stress over the last 3 months.
- On switching views:
  - The bar chart updates instantly.
  - Corresponding statistics also refresh to reflect the selected time frame.

2. Bar Chart Visualization

- Built using MUI Bar Chart, showcasing:
  - Average Stress Level – Highlighted in shades of blue.
- The chart adjusts its Y-axis range dynamically based on the maximum value from the dataset.
- Labelling format:
  - Day: Standard date (e.g., 22/03)
  - Week: Abbreviated format (e.g., Mar 22)
  - Month: Month-Year (e.g., Feb 2025)
- Purpose: Allows users to detect stress trends, track peaks and dips, and analyse behavioural or lifestyle patterns.

3. Summary Statistics Card

Summarizes the user's mental wellness data with values calculated for the selected view:

- Average Stress Level – Mean value over the period.
- Max Stress Level – Highest recorded level.
- Min Stress Level – Lowest recorded level.
- Total Entries – Number of data points logged.

These values empower users to assess their mental state over time, identify potential stress triggers, and seek timely intervention or relaxation strategies.

4. Informational Card – “About Stress Levels”

- Briefly educates users on what stress levels signify and how they affect the body.
- Encourages regular tracking to enhance mental health and self-awareness.
- Uses friendly design and iconography to maintain readability and emotional engagement.

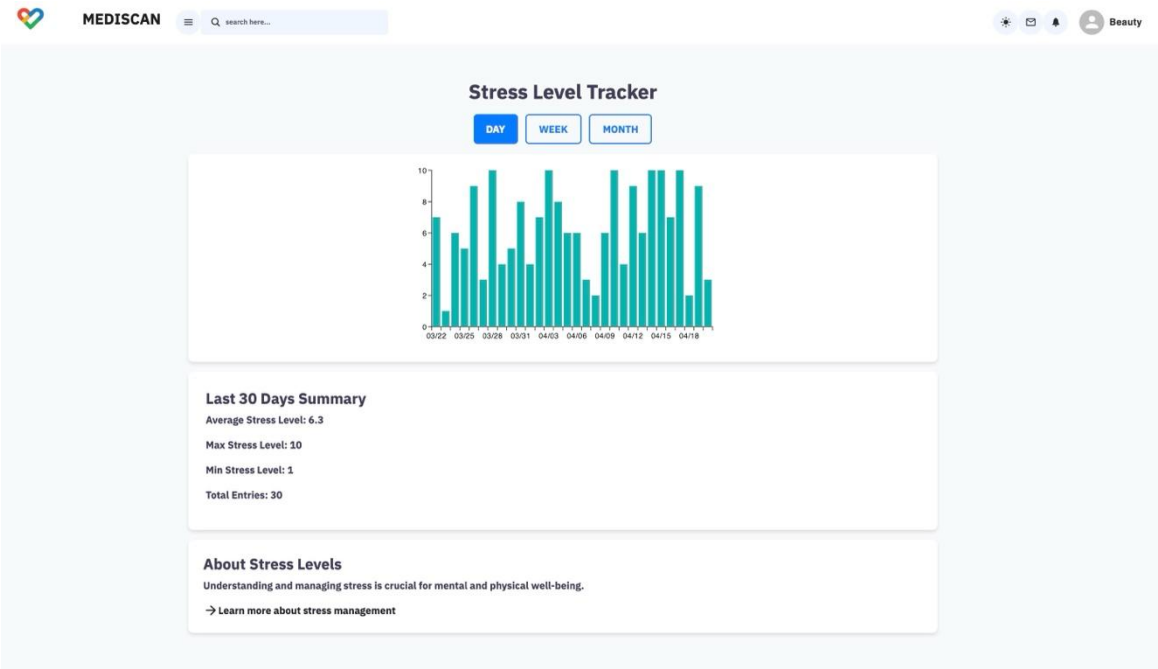

Figure 7.5.1 Daily Stress Level Section

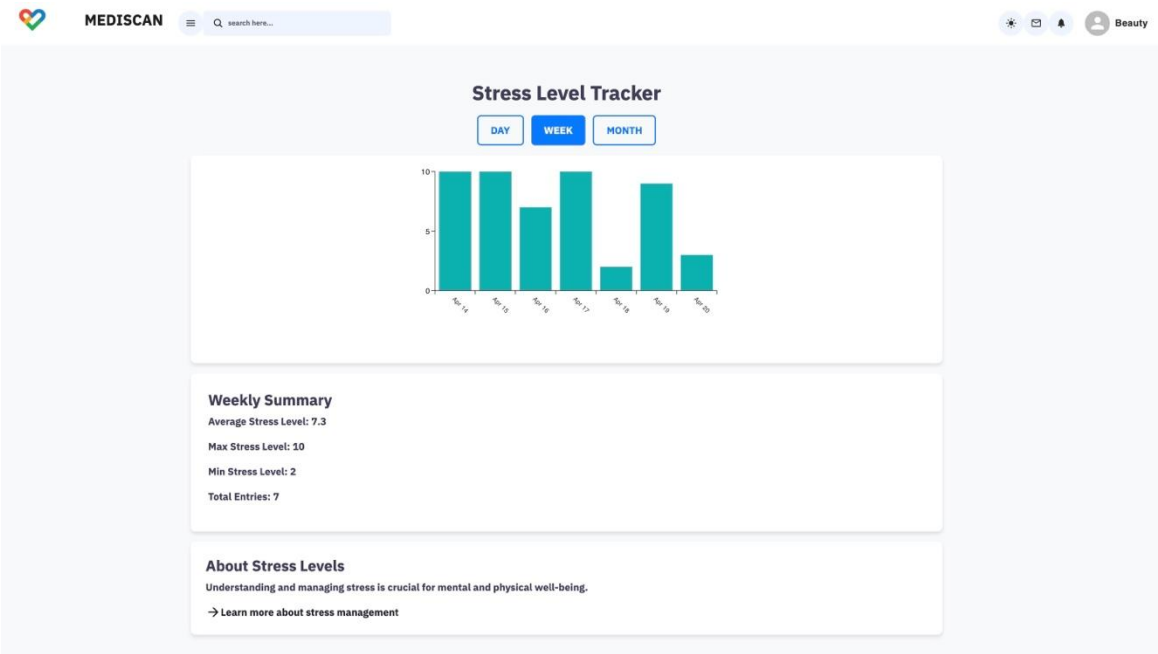

Figure 7.5.2 Weekly Stress Level Section

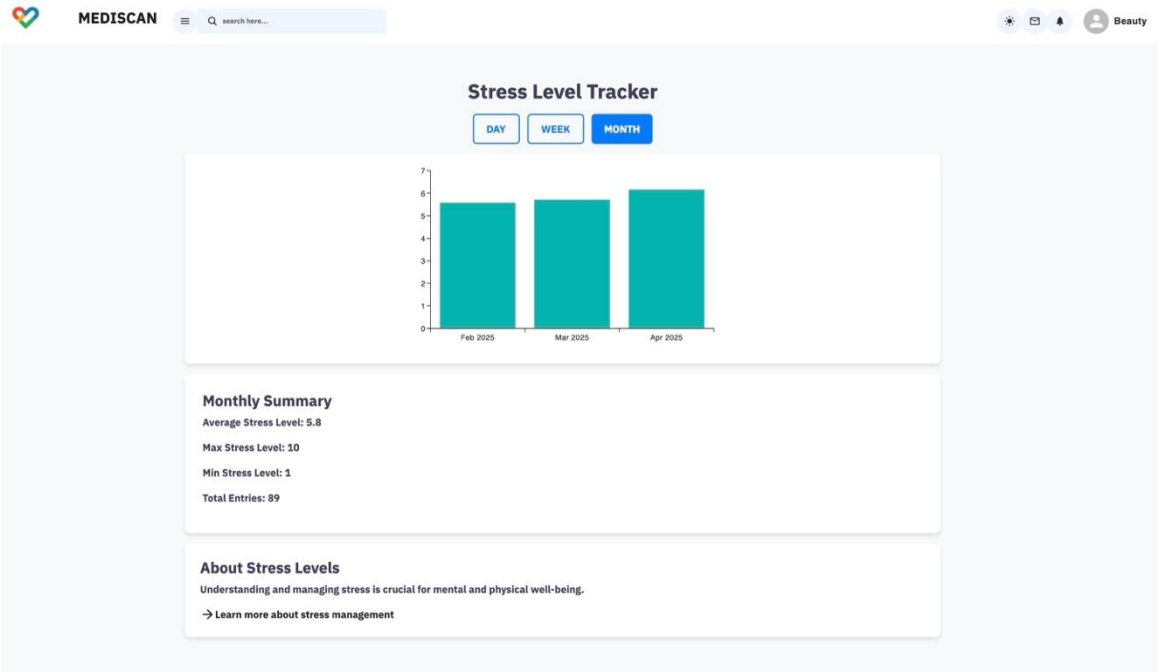

Figure 7.5.3 Monthly Stress Level Section

7.6 BLOOD OXYGEN MODULE

The Blood Oxygen Tracker Page is a vital sub-module of the Health Tracker Website, developed to help users monitor and understand their blood oxygen saturation over time.

While not featured on the main dashboard, this page plays a crucial role in assessing respiratory health by presenting daily data trends and summarizing key statistics in an easy-to-read format.

This feature supports the platform's mission to promote proactive health management by turning physiological data into meaningful insights for better breathing and cardiovascular awareness.

A. Purpose of the Page

- To offer a visual and statistical breakdown of the user's blood oxygen levels.
- To provide interactive bar charts for daily, weekly, and monthly blood oxygen tracking.
- To highlight average, max, min, and total data entries for better evaluation.
- To educate users on the importance of maintaining optimal oxygen saturation.

B. Functional Components

1. Time View Toggle (DAY / WEEK / MONTH)

- Toggle buttons let users switch between:
    - DAY View: Displays blood oxygen data for the last 30 days.
    - WEEK View: Aggregates average levels over the last 7 days.
    - MONTH View: Shows monthly average trends over the last 3 months.
- On view change:
    - The bar chart dynamically updates.
    - Summary statistics refresh according to selected view

2. Bar Chart Visualization

- Built with MUI Bar Chart for real-time feedback.
    - Blood oxygen values visualized in percentage scale.
- Adaptive Y-axis scaling for better clarity.
- Label formats:
    - Day: e.g., 22/03
    - Week: e.g., Mar 22

- Month: e.g., Feb 2025

- Purpose: To help users observe oxygen level trends and spot health concerns.

3. Summary Statistics Card

Summarizes blood oxygen saturation using key metrics:

- Average Blood Oxygen Level
- Maximum Level
- Minimum Level
- Total Entries Logged These values assist in identifying irregularities and maintaining optimal respiratory performance.

4. Informational Card – "About Blood Oxygen Levels"

- Brief explanation of why tracking oxygen saturation matters.
- Reinforces the importance of respiratory health monitoring.
- Uses clean design and subtle icons for user engagement.

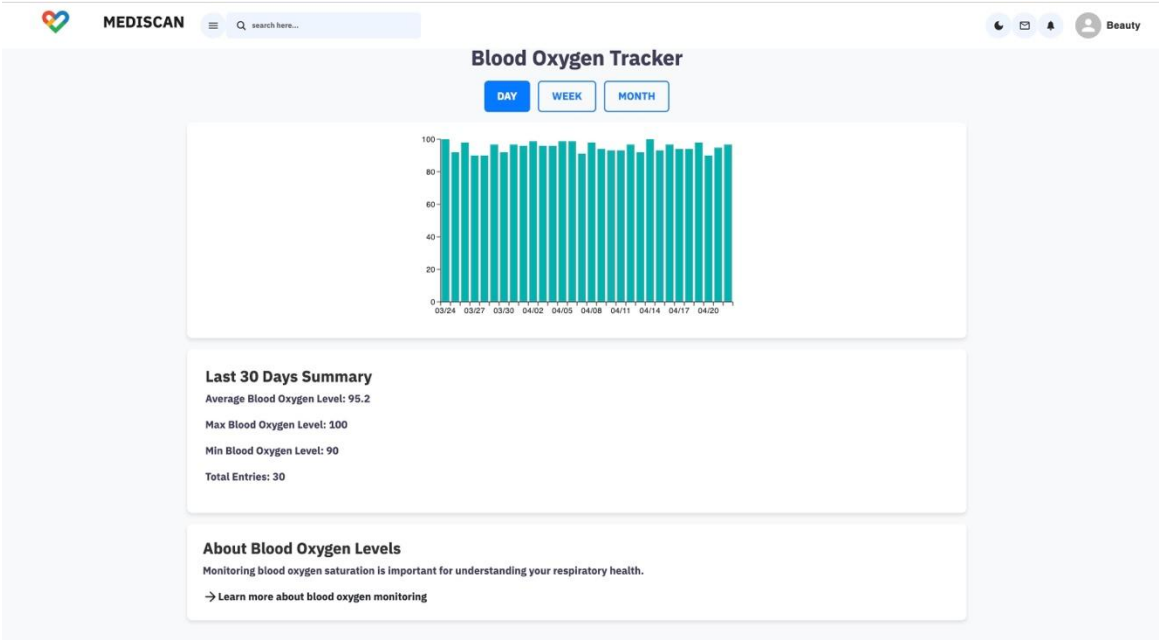

Figure 7.6.1 Daily Blood Oxygen Section

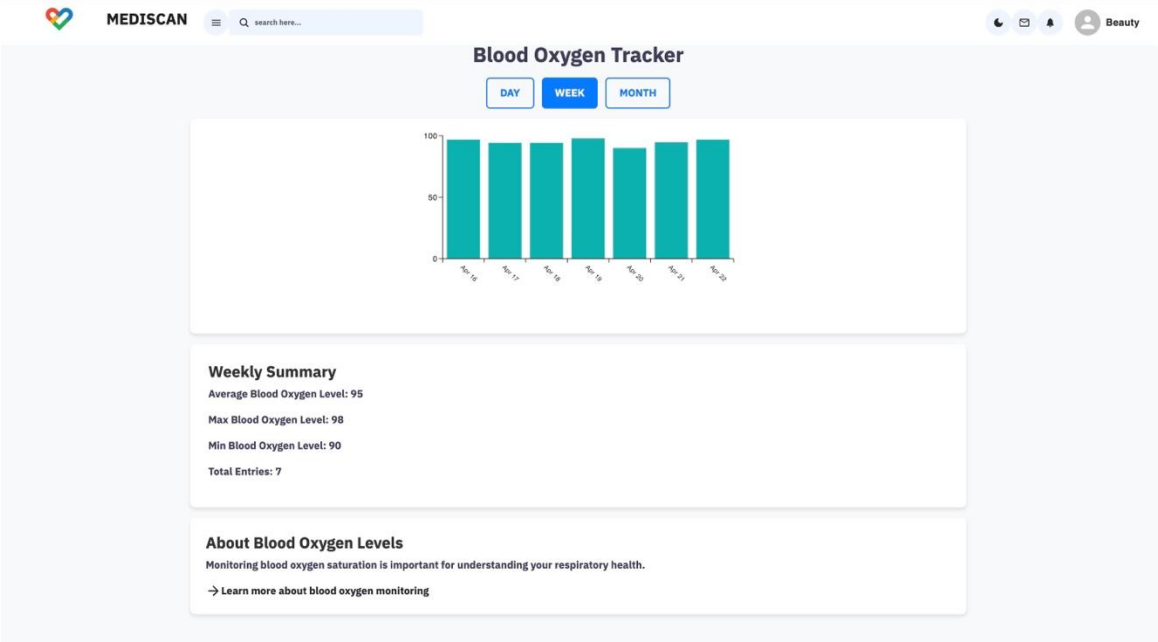

Figure 7.6.2 Weekly Blood Oxygen Section

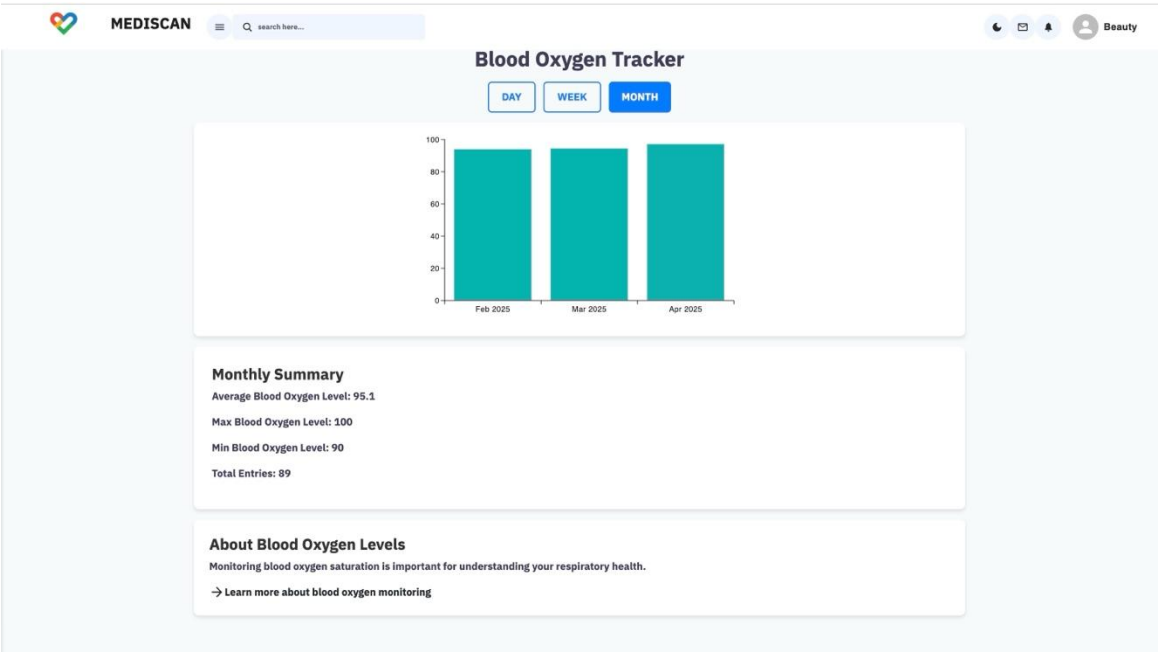

Figure 7.6.3 Monthly Blood Oxygen Section

7.7 CALORIE BURN MODULE

The Calorie Burn Tracker Page is a vital component of the Health Tracker Website, created to give users visibility into their daily energy expenditure. This page provides an intuitive

way to monitor calorie burn through various physical activities—ranging from casual steps to more intensive workouts—across different time views.

It supports the project's goal of promoting an active lifestyle by helping users stay informed about their metabolism and energy balance through interactive charts and insightful summaries.

A. Purpose of the Page

- To offer users a graphical and statistical breakdown of calories burned over time.
- To help users visualize their daily, weekly, and monthly calorie expenditure trends.
- To assist in goal-setting for fitness or weight management by showing active periods.
- To educate users about the importance of tracking calories for balanced health.

B. Functional Components

1. Time View Toggle (DAY / WEEK / MONTH)

- Includes three toggle buttons for switching between:
    - DAY View: Displays hourly calorie burn for the current day (24 hourly segments).
    - WEEK View: Displays total daily calories burned over the last 7 days.
    - MONTH View: Shows aggregate calorie burn for the past 3 months, grouped by month.
- The page dynamically updates the chart and summary statistics based on the selected view.

2. Bar Chart Visualization

- Powered by MUI Bar Chart, the visualization provides:
    - A bar for each hour/day/month, based on selected view.
    - Dynamic y-axis scaling to accommodate peak burn values.
- X-axis labelling:
    - Day View: 0:00 to 23:00 (hourly intervals)
    - Week View: Days of the week (Mon to Sun)

    - Month View: Month-Year format (e.g., Jan 2025)
- Purpose: Enables users to spot active hours, rest days, and monthly movement trends, helping them optimize their physical activity.

3. Summary Statistics Card

This card shows key numerical insights based on the selected view:

- Total Calories Burned – Sum of calories over the time period.
- Average Calories Per Day – Helpful for monitoring consistency.
- Max Calories Burned in a Day – Highest recorded value (activity peak).
- Min Calories Burned in a Day – Lowest recorded burn value.
- Days Recorded – Indicates how many days had valid data entries.

These metrics provide users with feedback on how active they've been and whether their calorie output aligns with their health goals.

4. Informational Card – "About Calorie Burning"

- Offers guidance and facts about energy expenditure and metabolism:
    - Calories are burned through walking, exercise, basic activities, and resting metabolism.
    - Monitoring calorie burn supports fitness, weight management, and energy balance.
- The section encourages consistent activity and healthy habits through motivational copy and a soft visual tone.

- Additional Notes

- Data Conversion Logic:
    - Calories are estimated using steps * 0.05 conversion for both hourly and daily values.
- Data Sources:
    - hourlySteps: Used for hourly (day) analysis.
    - steps + createdAt: Used for daily/weekly/monthly views.

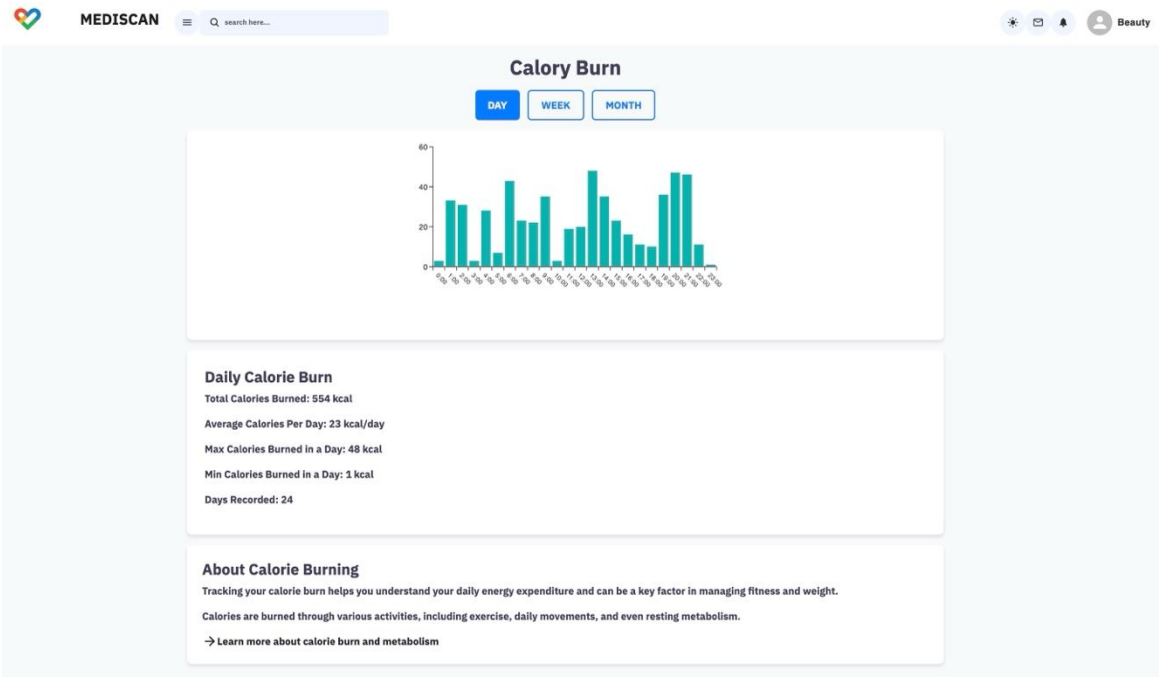

Figure 7.7.1 Daily Calorie Burn Section

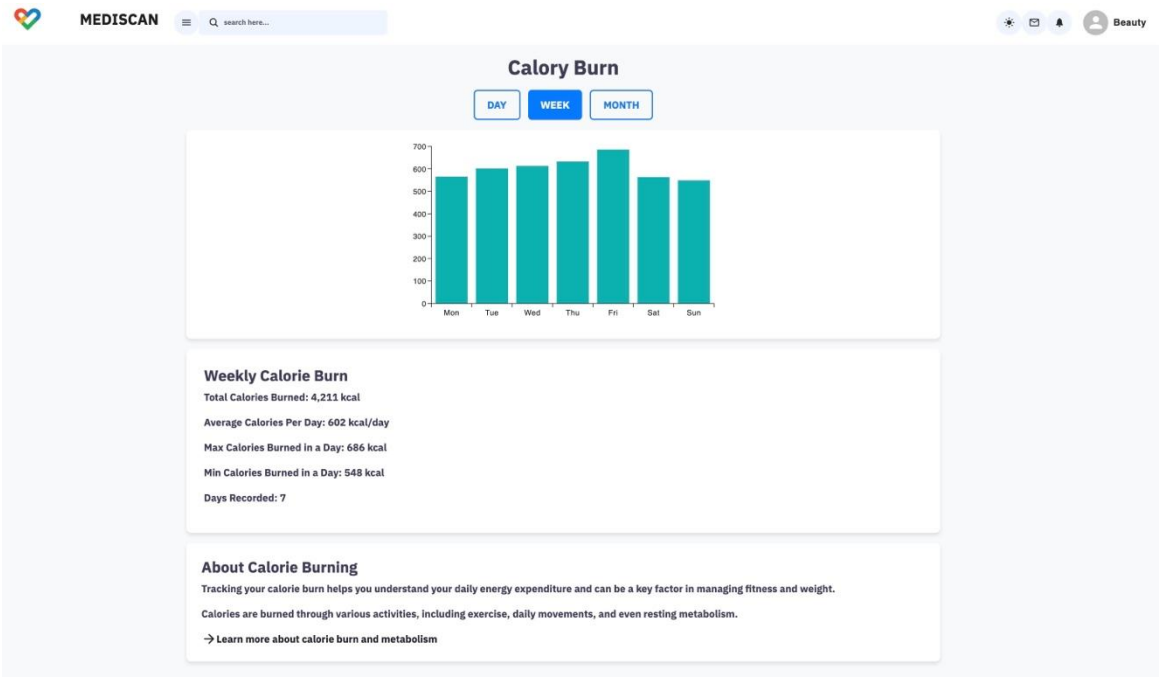

Figure 7.7.2 Weekly Calorie Burn Section

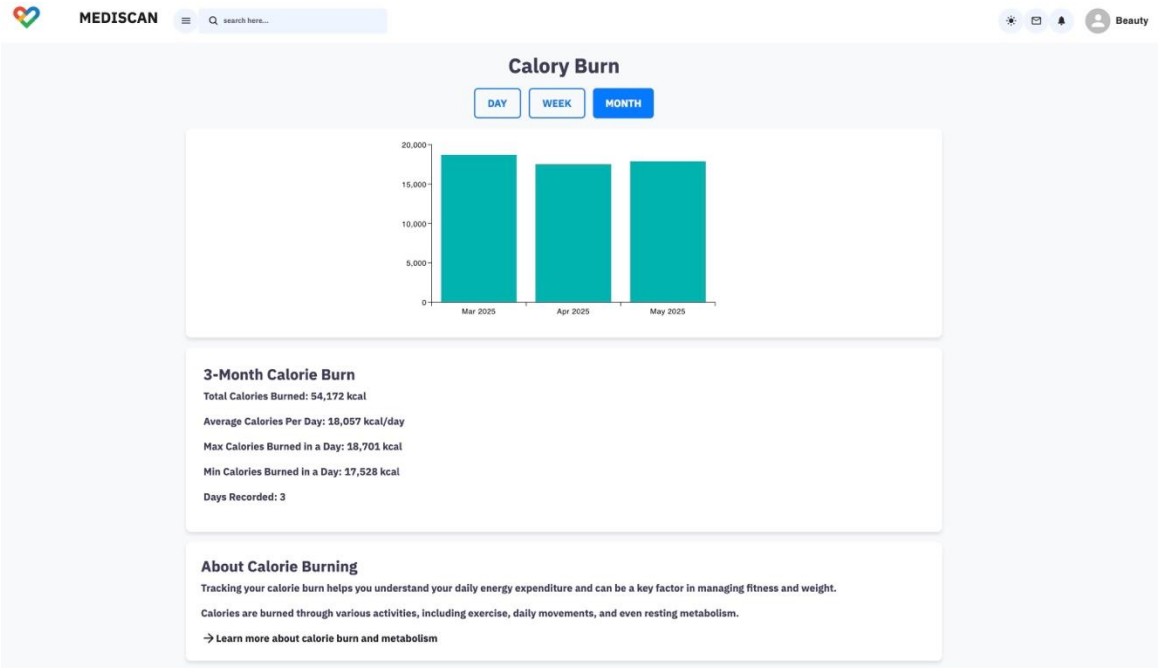

Figure 7.7.3 Monthly Calorie Burn Section

7.8 DATA ANALYSIS MODULE

The Data Analysis Page acts as the intelligent core of the Health Tracker Website, offering a dynamic recommendation engine that translates complex health data into meaningful insights. By integrating real-time smartwatch data with user-provided inputs, this page empowers users to analyze essential health metrics such as blood pressure, BMI, blood sugar levels**,** and diabetes risk—all from one centralized location.

Unlike traditional tracking interfaces that merely display data, this module goes a step further by evaluating**,** interpreting, and personalizing health information. It helps users understand how their physical attributes, activity levels, and glucose readings interact with risk factors for chronic conditions like diabetes and hypertension. It then provides custom suggestions**,** such as daily step goals**,** to mitigate those risks.

More than just a feature, the Data Analysis Page serves as a decision-making companion—enabling early detection, promoting proactive behaviour, and guiding users toward preventive healthcare**.** With its intuitive design, educational support, and visually rich results, it becomes an essential pillar of the overall project, driving both user engagement and health awareness.

A. Purpose of the Page

- To help users analyze and interpret core health parameters using their historical and current data.
- To calculate BMI based on user-provided height and weight (with unit conversion support).
- To assess diabetes risk by combining age, BMI, and blood sugar levels.
- To offer custom step-based activity recommendations to reduce health risks.
- To educate users about healthy ranges for BMI, blood pressure, and blood sugar levels.

B. Functional Components

1. Tab Switcher: Blood Pressure and Diabetes Views

- At the top of the page, two interactive tabs let users switch between:
    - Blood Pressure Analysis Tab
    - Diabetes Risk Assessment Tab
- The default tab is “Blood Pressure,” but routing state allows dynamic pre-selection.

2. Diabetes Risk Assessment (Tab 2)

- Input Fields

- Age Group Selector: Dropdown to classify users into one of four age brackets (e.g., 6–12, 13–19, 20+).
- Height & Unit Selector: Accepts height in cm, inch, or meters and auto-converts internally.
- Weight (kg): Numeric input for body weight.
- Blood Sugar Level (mg/dL): Captures fasting glucose level for analysis.

- **Automated Calculations**

- BMI Calculation:
    - Auto-calculated from height and weight.
    - Classified into categories: Underweight, Normal, Overweight, Obese.
    - Visually displayed with a dynamic BMI bar chart and indicator.

- **Diabetes Risk Evaluation**:
    - Calculates a risk score based on:
        - Age
        - BMI
        - Blood Sugar
    - Classifies users as Low, Moderate, or High Risk.
- **Step Recommendations**:
    - Based on risk level, BMI, and glucose levels, a personalized daily step goal is computed.
    - Example: High-risk users with obesity and elevated glucose are encouraged to reach 12,000–15,000 steps/day.
    - The UI displays:
        - Current steps
        - Recommended steps
        - Steps needed
        - A motivational message

- **Result Summary Section**

- Conditionally displayed after form submission.
- Highlights:
    - Selected Age Group
    - Height & Weight
    - BMI & Category
    - Blood Sugar Level & Status
    - Diabetes Risk Level
    - Custom Steps Suggestion with a progress bar

3. Blood Pressure Section (Tab 1)

- Renders a modular Blood Pressure component.
- Provides:
    - Live BP values and evaluation
    - Classifications (Normal, Elevated, Stage 1, Stage 2, Crisis)
    - Tips for maintaining healthy blood pressure

4. Informational Panel

- Dynamically changes based on selected tab:
    - Blood Pressure Tab:
        - Details healthy ranges (e.g., <120/80 mmHg)
        - Lists risk factors and complications
        - Management tips (e.g., sodium control, regular exercise)
    - Diabetes Tab:
        - BMI categories and formulas
        - Blood sugar classifications (Normal, Prediabetic, Diabetic)
        - Lifestyle tips for diabetes prevention and control:
            - Daily steps, fibre-rich diet, hydration, glucose checks, etc.
        - Ends with a motivational message and icon section for professional advice

- **Technical Highlights**

- React Hooks: Used for state management and effect handling.
- Context API: Pulls in real-time smartwatch data (e.g., steps, glucose levels).
- Toast Notifications: Instant feedback on form validation and risk computation.
- Responsive CSS Styling: Ensures full compatibility across desktop and mobile devices.
- Dynamic Visualization: BMI scale with coloured markers and real-time indicator.

- **Value to the User**

- Transforms raw data into personalized health recommendations.
- Encourages daily activity goals to reduce diabetes risk.
- Promotes awareness of blood pressure and sugar levels in an interactive and user-friendly format.
- Combines educational insights with real-time monitoring to promote holistic wellness.

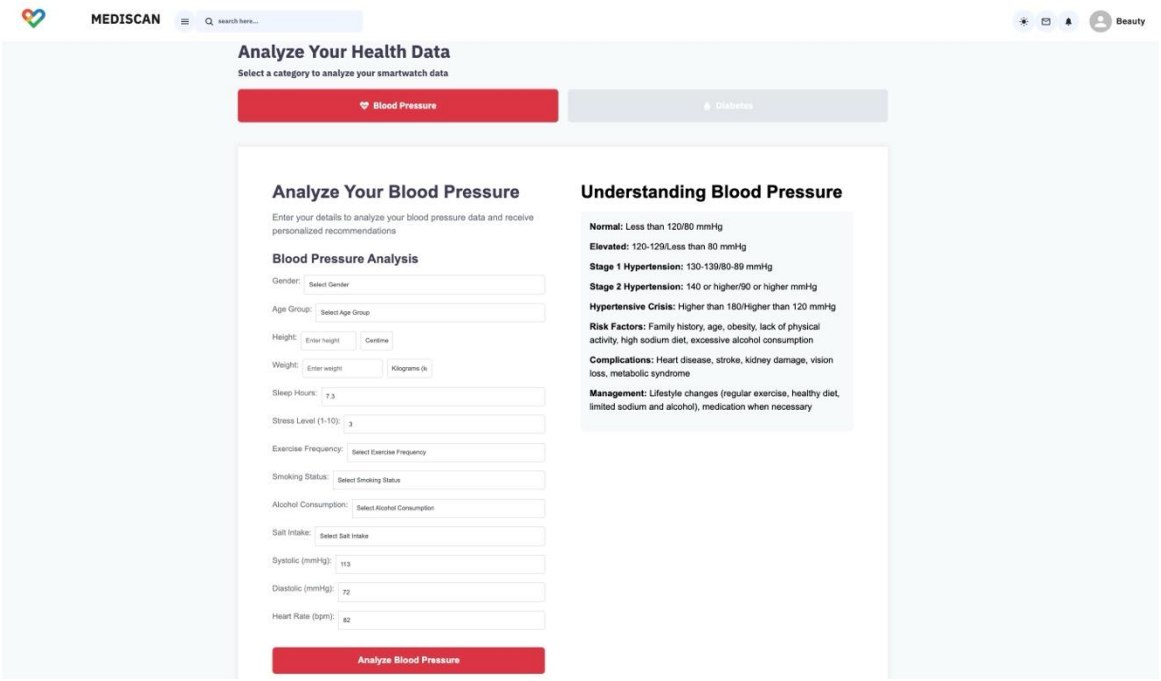

Figure 7.8.1 Data Analysis Section-BP

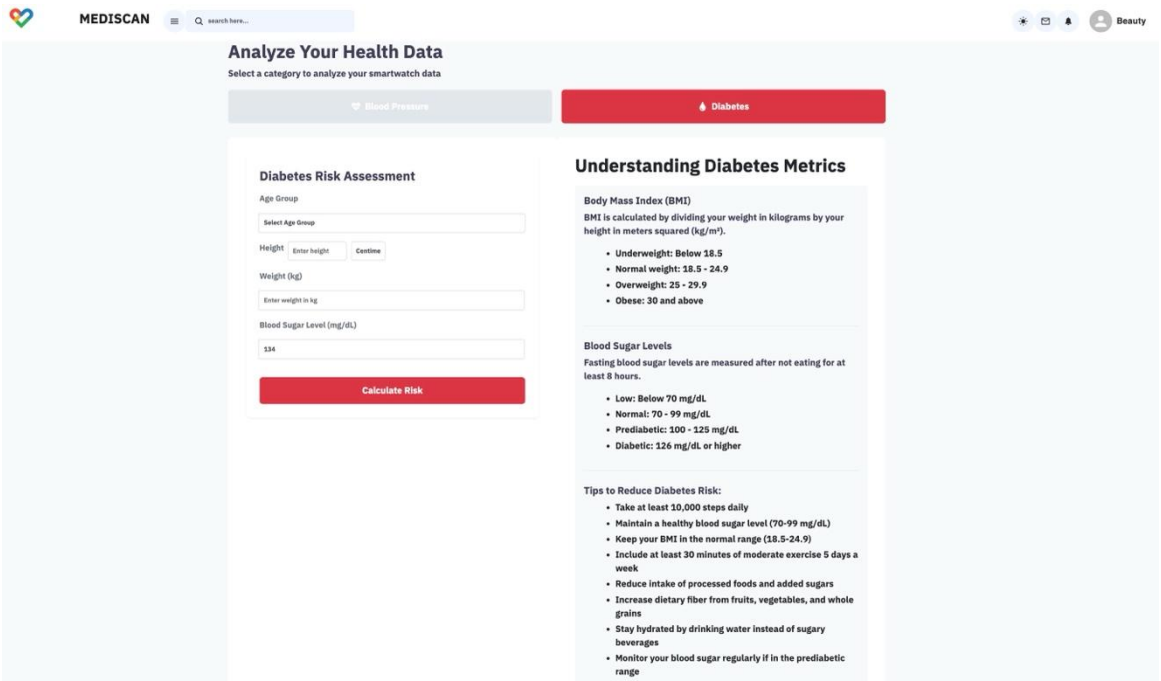

Figure 7.8.2 Data Analysis Section-Diabetes

7.9 DATA UPLOAD MODULE

The Smartwatch Data Upload Page is a foundational entry point in the Health Tracker Website, designed to allow users to securely upload health data in .json format generated by their smartwatch devices. This page enables all other modules—including data visualization, report analysis, and personalized recommendations—to function accurately by providing the necessary raw input.

Serving as the first interaction stage in the application workflow, this page plays a crucial role in initializing the backend analysis pipeline and ensures data integrity and format correctness through validation before submission. Its intuitive interface and real-time feedback provide a user-friendly experience, guiding users step-by-step in preparing their data for detailed health assessment.

A. Purpose of the Page

- To collect smartwatch health data from the user in a secure and controlled format (.json).
- To serve as a gateway for sending user data to the backend for AI-powered analysis.
- To validate file type and size before initiating further medical insight processing.
- To educate the user on the required file format and how to correctly upload it.

B. Functional Components

1. Upload Interface

- Users are prompted to upload their smartwatch-generated .json files.
- Upload is done via a clearly labelled button:
    - "Upload SMARTWATCH DATA", accompanied by an upload icon.
- On click, the hidden file input is triggered, allowing file selection.
- The uploaded file name is displayed for confirmation before continuing.

2. File Validation

- Only .json and .pdf files are accepted, though analysis is only available for .pdf in the current version.

- If the file is not supported (e.g., .docx or .txt), an error message is displayed via react-toastify.
- Feedback includes:
    - Success for valid .json
    - Error for unsupported types
    - Warning if no file is selected
    - Info if a .json file is uploaded but not used for analysis

3. Continue to Analysis

- Clicking “Continue” uploads the file to the backend using axios and a FormData object.
- Upon successful upload:
    - The file is analyzed.
    - The user is redirected to the Visualize Page (/Visualize) to view insights generated using GPT-based analysis.
- Backend endpoint used: POST http://localhost:3000/api/upload-pdf

C. Informational Card – Upload Guide

- A visual guide is provided on the right-hand side, showing:
    - File type supported: .json
    - Max file size: 2MB
- A relevant illustration enhances understanding and improves engagement.
- Designed with clear typography and spacing to avoid information overload

Technical Features

- React Hooks: Used for file handling (useState, useRef)
- Navigation: useNavigate() handles redirection after upload
- Feedback System: react-toastify for real-time validation and status updates
- Axios Integration: For backend communication with appropriate headers
- Responsive Styling: All elements styled using upload.css with Flexbox layout and card-based UI

Role in the Project

Although simple in design, the Smartwatch Upload Page is essential for initializing the health monitoring workflow. Without this input stage, advanced modules such as diabetes prediction, stress tracking, and calorie burn visualization would have no data to operate on.

It ensures:

- Data quality
- User understanding of supported formats
- Smooth entry into AI-powered analysis

As such, it plays a pivotal role in the success of the entire Health Tracker project.

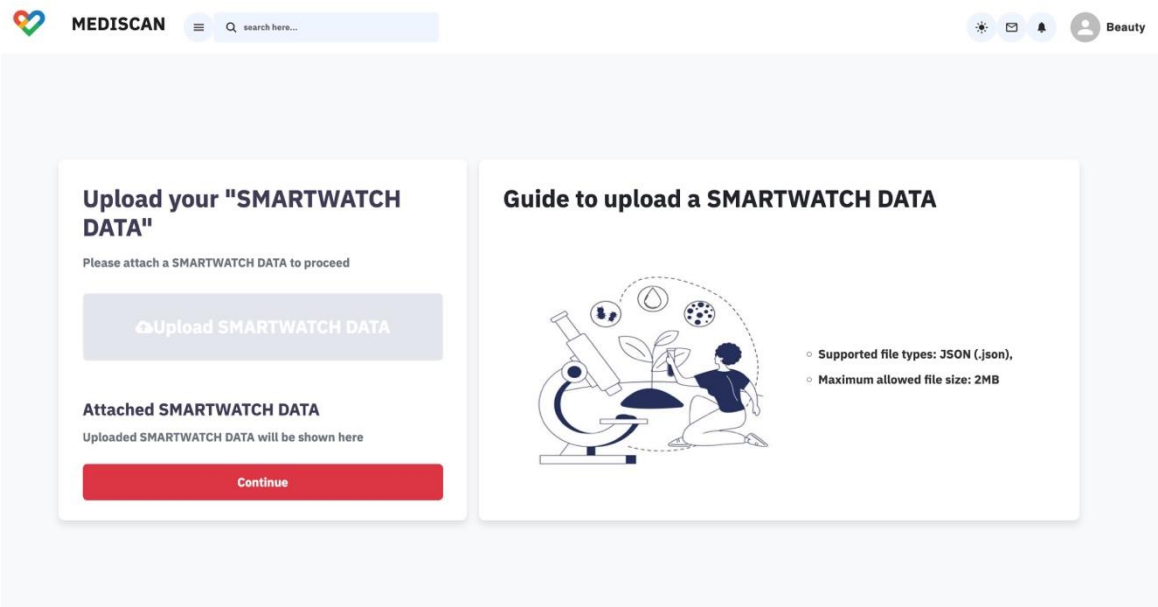

Figure 7.9.1 Data Analysis Section

7.10 MEDICAL REPORT ANALYZER MODULE

The Medical Report Analyzer Page is the heart of the Health Tracker Website. It's where the uploaded reports are parsed, understood, and converted into meaningful, readable, and visual health insights using AI. It allows users to view AI-generated summaries of their uploaded PDF-based lab reports, along with relevant health data visualizations and

educational content. While the core analytical power comes from the OpenAI API, the rest of the page is fully developed using ReactJS**,** NodeJS, and ExpressJS**,** ensuring a seamless and responsive user experience.

A. Purpose of the Page

- To analyze uploaded medical reports using the OpenAI API.
- To detect medical markers, abnormalities, and potential risks based on report content.
- To present results in a simple, easy-to-understand text and graphical format.
- To promote user health awareness by linking findings to health goals and definitions.

B. Functional Components

1. Hero Section
    - Includes a tagline like *"Track Your Health Intelligently"*.
    - Call-to-Action (CTA) button: “Get Started” redirects to the Upload Page.
2. Feature Highlights
    - Cards for major features: Smart Upload, GPT-based Health Insight, Visualization Dashboard.
3. Login/Sign Up Prompt
    - For new users to register and returning users to log in.
4. Tech Stack or Trust Indicators
    - Mentions of AI, GPT, medical-grade accuracy, and data privacy.

- ## How It Works Page

This page educates users about the workflow of the system—from report upload to AI-driven health analysis.

A. Purpose of the Page

- To demystify the backend processes for the end user.
- To increase user trust through transparency.
- To provide a step-by-step usage guide.

B. Functional Components

1. **Workflow Timeline**
    - A horizontal/vertical visual guide:
        1. Upload Report
        2. File Validation
        3. AI-Powered Analysis
        4. Visualization
        5. Health Insights
2. **Technical Summary**
    - Explains use of GPT-4, smart categorization, and health metric tracking.
3. **Illustrations/Icons**
    - Help reinforce understanding visually.

- ## Critical Information Page

This page emphasizes the system's disclaimers, data handling, and usage boundaries to maintain ethical and responsible use.

A. Purpose of the Page

- To inform users about limitations and privacy.
- To prevent misuse and ensure ethical compliance.
- To advise users to consult healthcare professionals for decisions.

B. Functional Components

1. **Disclaimers**
    - "This platform does not replace professional medical advice."
2. **Data Privacy Statement**
    - Explains that no personal data is stored after analysis (unless account-linked).
3. **Terms of Use Summary**
    - Mentions acceptable file types and system responsibilities.

- ## Health Goals Page

This section is focused on long-term health improvement. Users can view goals like maintaining blood pressure, sugar levels, etc., and set personal objectives.

A. Purpose of the Page

- To track and motivate health progress.
- To allow users to set wellness goals.

B. Functional Components

1. **Predefined Goals**
    - Templates for goals like:
        - Keep BP under 130/80 mmHg
        - Weekly step goal: 70,000 steps
2. **Custom Goal Input**
    - Users can add a name, value, target date, and category (e.g., Fitness, Nutrition).
3. **Progress Visuals**
    - Circle progress bars or line charts for goals.
4. **Goal Reminders (Optional Future Feature)**
    - Notify users when they're nearing targets or falling behind.

- ## Upload report Page

The Upload Reports Page is a critical component of the Health Tracker Website, designed specifically for users to upload their clinical or diagnostic medical reports (PDF format) for AI-based health analysis. Unlike the smartwatch upload page, this page handles reports directly from labs, hospitals, or diagnostic centres and uses them as the foundation for in-depth medical insights.

A. Purpose of the Page

- To collect PDF-based medical reports from users.
- To perform AI analysis (via GPT) on structured/unstructured medical text.

- To generate insights such as potential diagnoses, biomarker anomalies, and action suggestions.
- To guide users directly into the Medical Report Analyzer module.

B. Functional Components

1. Upload Interface
    - Uploads PDF reports via a clearly labelled button: Upload Medical Report (.pdf)
    - File validation ensures only .pdf format is accepted.
    - Displays file name once selected.
2. Feedback System
    - Success: File accepted and uploaded.
    - Error: If file type is unsupported or missing.
    - Uses react-toastify for dynamic alerts.
3. Continue to Analysis
    - After upload, file is sent to the backend endpoint: POST http://localhost:3000/api/upload-pdf
    - File is forwarded using Axios + Form Data.
    - Redirects user to /analysis (Medical Report Analyzer Page) after upload.
4. Upload Guidelines Section
    - Side card showing:
        - Supported: .pdf
        - Unsupported: .json, .docx
        - Max size: 2MB
5. Illustration/Instructions
    - Visual mock-up of a medical report to guide formatting.
    - Simple steps: Upload → Analyze → View Insights.

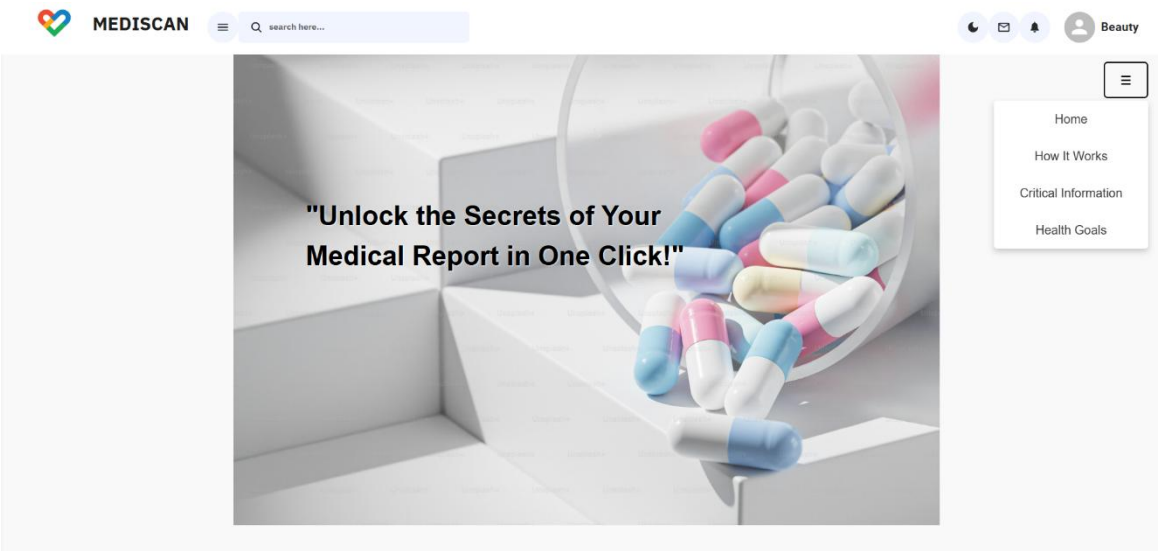

Figure 7.10.1 Medical report analyzer section

7.11 USER PROFILE MODULE

The User Profile Page allows individuals to input and manage their basic health information, activity preferences, and emergency contact details. This page provides a centralized form where data like height, weight, blood pressure, and bedtime schedules can be stored to personalize health insights and goal recommendations across the platform.

It is fully built using ReactJS and styled via custom CSS, ensuring responsiveness and accessibility across all devices.

A. Purpose of the Page

- To collect essential personal and medical information for the user.
- To allow users to set personal activity goals like step counts and heart points.
- To enable tracking of routine health metrics (e.g., BP, sugar, heart rate).
- To provide a safe contact reference through emergency contact fields.
- To enhance the context and accuracy of personalized health suggestions across the Health Tracker platform.

B. Functional Components

1. Activity Goals Section

- Let's users set **daily step targets** (e.g., 5,000 to 20,000 steps).
- Configures **heart points**—a fitness metric used to quantify cardio efforts.
- Drop-down menus provide intuitive selection of fitness goals.

2. Bedtime Schedule Section

- Users can toggle their bedtime tracking **on or off** using a stylish switch button.
- When enabled, users can set:
  - **Bedtime Hour** (e.g., 11:00 PM)
  - **Wake-up Hour** (e.g., 7:00 AM)
- This data helps future modules track sleep duration and patterns.

3. About You Section

- Inputs for gender, height, and weight (with unit options):
  - Height: cm, m, ft, in
  - Weight: kg, g, lb, oz
- Helps adjust BMI calculations or caloric needs for future analysis.

4. Medical Information Section

- Collects vital health metrics:
  - **Email**, **Date of Birth**
  - **Blood Group**
  - **Blood Pressure** (systolic & diastolic)
  - **Heart Rate**
  - **Blood Sugar** (mg/dL)
  - **Current Medications**
- Required fields (like email and blood group) include form validation and error handling.

5. Emergency Contact Section

- Collects:
  - **Name**
  - **Phone Number**

  - **Country Code Selector** (e.g., +91, +1, +44)
- Ensures a contact is on file in case of health emergencies.

6. Submit Button

- Final "**Save Changes**" button submits the form.
- Triggers validation:
  - Highlights errors dynamically using form Errors state.
  - Scrolls smoothly to the first error using scrollIntoView.

C. Technical Features

- **React State Management**
  - Uses use State to manage form data and field-specific errors.
  - Toggle states (e.g., bedtime schedule) handled via custom logic.
- **Form Validation**
  - Required field checks for email, DOB, blood group, emergency contact.
  - Input-level error messages displayed using CSS (.error-message class).
- **Responsive Design**
  - Fully responsive layout using **flexbox** and **media queries**.
  - Automatically stacks form inputs vertically on mobile screens.
- **Custom Input Components**
  - Multi-part input groups for units (e.g., weight + kg).
  - Phone number and bedtime inputs are built with **multi-select controls**.
- **Stylized UI**
  - Switch toggle uses custom CSS animation.
  - Drop-down menus feature icons and alignment via .dropdown-container.
- **User Feedback**
  - Onsuccessfulsubmission:

    alert("Profile updated successfully!")
  - Onerror:

    Scrolls to first invalid field with a red border and message.

Role in the Project

The User Profile Page acts as a user-centric configuration hub. All personal data provided here can enhance the system's interpretation of:

- Smartwatch activity data (steps, heart points)
- Medical report analysis context (age, weight, blood pressure)
- Health goal tailoring (bedtime, medications, blood sugar).

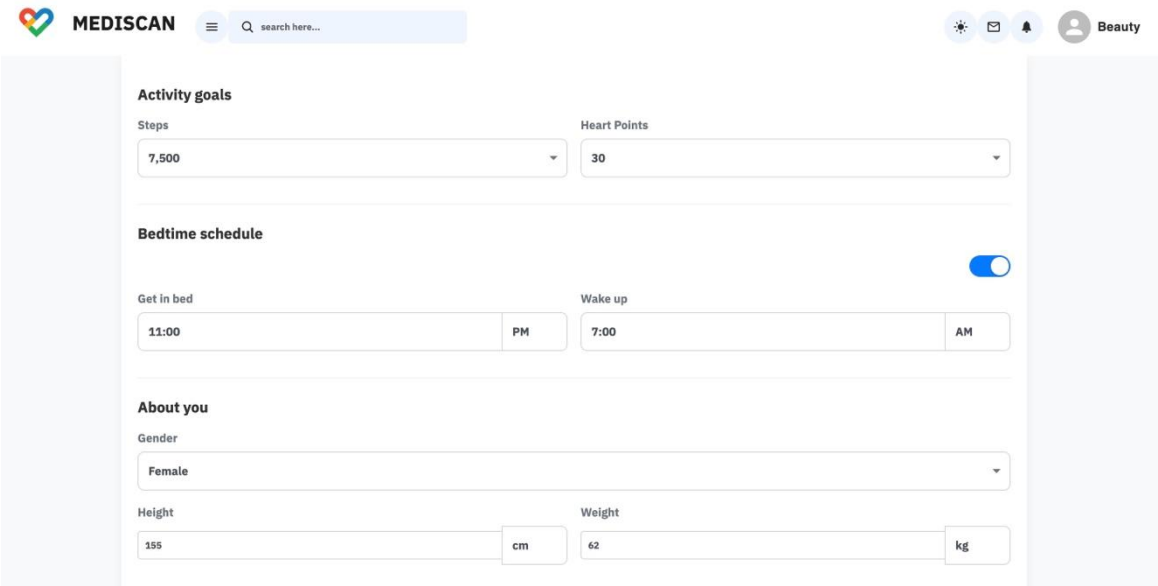

Figure 7.11.1 User Profile Secti

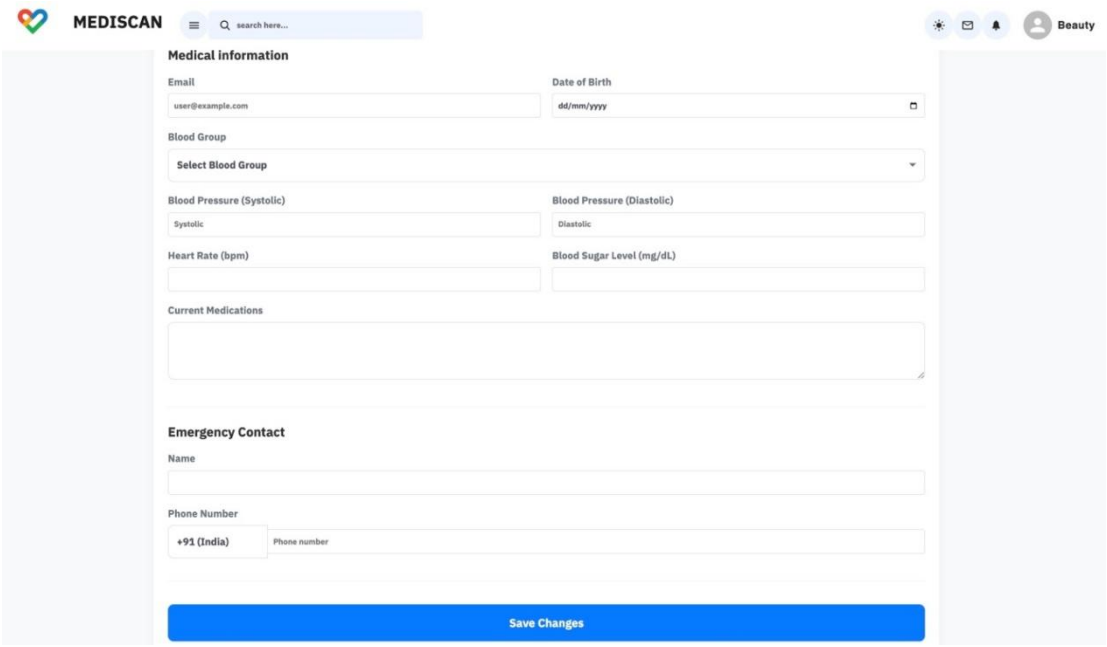

Figure 7.11.1 User Profile Section

7.12 CONTACT US MODULE

The Contact Us Page acts as a communication bridge between the Health Tracker platform and its users. It allows users to send inquiries, feedback, or support requests directly to the developers/admin via an email service. This component enhances user engagement, support responsiveness, and trust by offering an always-available point of contact.

Built using ReactJS and styled with custom CSS, it uses Emailjs API for real-time email dispatching without backend email setup.

A. Purpose of the Page

- To provide users with a direct channel for contacting the platform team.
- To collect structured queries (e.g., inquiries, support issues, feedback).
- To send messages using EmailJS API—no server-side email logic required.
- To build user trust and satisfaction by being approachable.

B. Functional Components

1. Hero Image Section

- A full-width, responsive banner image (contact_img.jpg) adds visual depth.
- Positioned at the top to set the tone and context of the page.

2. Form Container

A neatly designed floating form block styled with padding, drop shadow, and border radius.

Title Block

- Main heading: ″Contact Us Anytime″
- Subtext: ″Reach Out at Your Convenience″

Form Fields

Each field is validated, styled, and aligned in a responsive layout:

Row 1: Basic Info (Three columns)

- Name: Required text input.
- Email: Required email input with built-in HTML validation.
- Phone Number: Optional numeric input.

Row 2: Subject Selection

- Drop-down menu with:
  - “Inquiry”
  - “Support”
  - “Feedback”

Row 3: Message Box

- Multi-line text area for users to input their query or comments.

Submit Button

- Large, red button styled with hover transition.
- Triggers handle Submit() function on click.

C. Submission Process

When the user submits the form:

1. Loading Animation
   - A loading overlay is conditionally shown (placeholder GIF) while the request is being processed.
2. EmailJS Integration
   - Form data is sent to EmailJS using:
     - service_s78vi5i (Service ID)
     - template_pc1m6mo (Template ID)
     - mZjSepw1zsONPA4Ib (User ID)
3. Response Handling
   - On success:
     - Displays a “Submitted Successfully” animation (Done.gif)
     - Form resets after 10 seconds automatically.

- On failure:
  - Logs error (console-based for now; could be improved with alert or toastify).
- Loading state handled with set Loading(true/false).

D. Technical Features

- Frontend Only Email Delivery using EmailJS API (no backend email server needed).
- React State Management
  - use State for form fields, loading, success message toggle.
- Timed Reset Logic
  - set Timeout() used to reset UI and form after successful submission.
- Form Validation
  - Uses HTML required attribute and default validation for email.
- Responsive Design
  - CSS Grid/Flex layout ensures responsiveness across desktop and mobile.
- Styled with Custom CSS
  - Clean layout using .form-group, .form-row, .submit-btn, etc.
  - Colour palette and form shadows align with overall theme.

E. UX Enhancements

- Visual feedback via:
  - Loader (when submitting)
  - Done animation (after success)
- Aesthetic:
  - Soft shadows, light theme, rounded edges
  - Simple and welcoming interface
- Form resets automatically, no page reloads
- Designed for easy expansion (e.g., chatbot embed, FAQ link, live chat support)

- Role in the Project

The Contact Us Page ensures that users always feel supported and connected. While the platform handles complex medical data and analytics, this page ensures that no technical complexity ever becomes a barrier to communication.

It adds a human touch to a data-heavy project, bridging the gap between tech and trust.

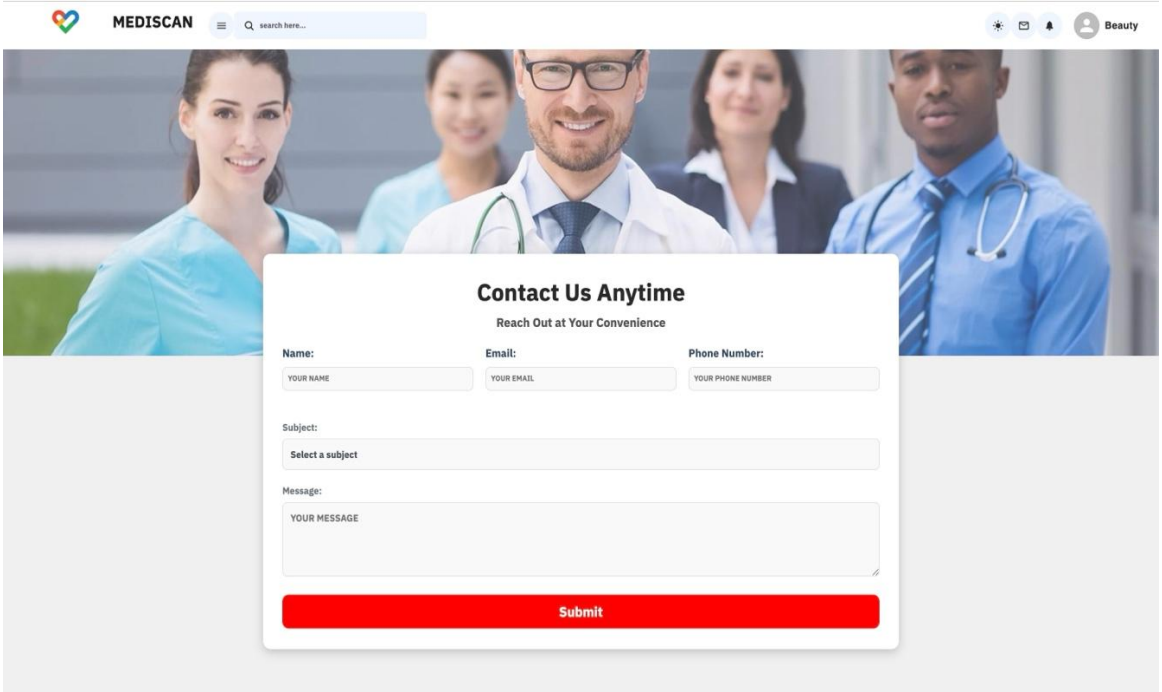

Figure 7.12.1 Contact Us Section

7.13 SECURITY & PRIVACY MODULE

7.13.1 LOGIN

The Login Module is the primary access gateway for registered users of the Health Tracker Website. It provides a clean, responsive interface for logging in via email/password, with options for social sign-in through Google and Facebook.

Built with ReactJS**,** styled using custom CSS**,** and connected to a NodeJS/Express backend**,** it ensures smooth authentication, form handling, and navigation.

A. Purpose of the Page

- To provide a secure, user-friendly interface for signing in to the platform.
- To authenticate users and redirect them to their personalized dashboard.
- To support both traditional (email/password) and social logins (Google, Facebook).
- To establish user session context using React's global state via UserContext.

B. Functional Components

1. Branding Section

- Displays the logo and app title: *"Login to Medi scan"*
- Positioned prominently for instant recognition and trust.

2. Login Form Block

Styled as a centred card with shadow and rounded edges for a professional look.

Input Fields

- Email or Phone input with icon (MdEmail)
- Password input with toggle visibility (IoEye, IoEyeOff)
    - Dynamic type switch between text and password
    - Icon (FaUnlock) placed inside the input

Validation & Submission

- Fields are required before submission.
- On form submission:
    - Sends POST request to:
      http://localhost:3001/api/auth/signin
    - Passes { email, password } as request body via Axios.

Successful Login

- Stores user in global state using UserContext.
- Redirects to /dashboard using useNavigate().

Unsuccessful Login

- Triggers a user-friendly alert:
  "Login failed. Please check your credentials."

3. Extra Options

- Forgot Password link (currently static placeholder).
- Signup redirect to /Signup for new users.

C. Social Login Options

Although mocked for now, the UI supports:

- Sign in withh Google
  Displays Google logo (google.webp) + styled button.
- Sign in with Facebook
  Displays Facebook icon (facebook.png) + branded button.

Note: Social logins can be integrated using libraries like Firebase Auth, OAuth 2.0, or Passport.js.

D. Technical Features

- ReactJS
  - Uses useState for form control and password visibility.
  - useContext(UserContext) for maintaining authenticated user globally.
- Routing
  - useNavigate() from react-router-dom redirects users post-login.
  - Link to redirect users to register page.
- Axios Integration
  - Sends secure POST requests to backend auth service.
  - Error handling via try/catch.
- User Context
  - Centralized user session management across the platform.
  - Prevents unauthorized access to protected routes.

E. UI/UX Design Features

- Visual Clarity
  - Icon-assisted inputs improve usability.
  - Password visibility toggle enhances user experience.
- Responsive Layout
  - Login card and form fields adapt to screen sizes.
  - Mobile-friendly button sizing and spacing.
- Custom Styling
  - Styled using login.css:
    - .login Section, .login Box, .form-control, .toggleShowPassword
  - Button hover states, shadows, and branded color schemes included.
- Form Feedback
  - Immediate validation errors upon missing input.
  - Alert-based feedback on login failure.

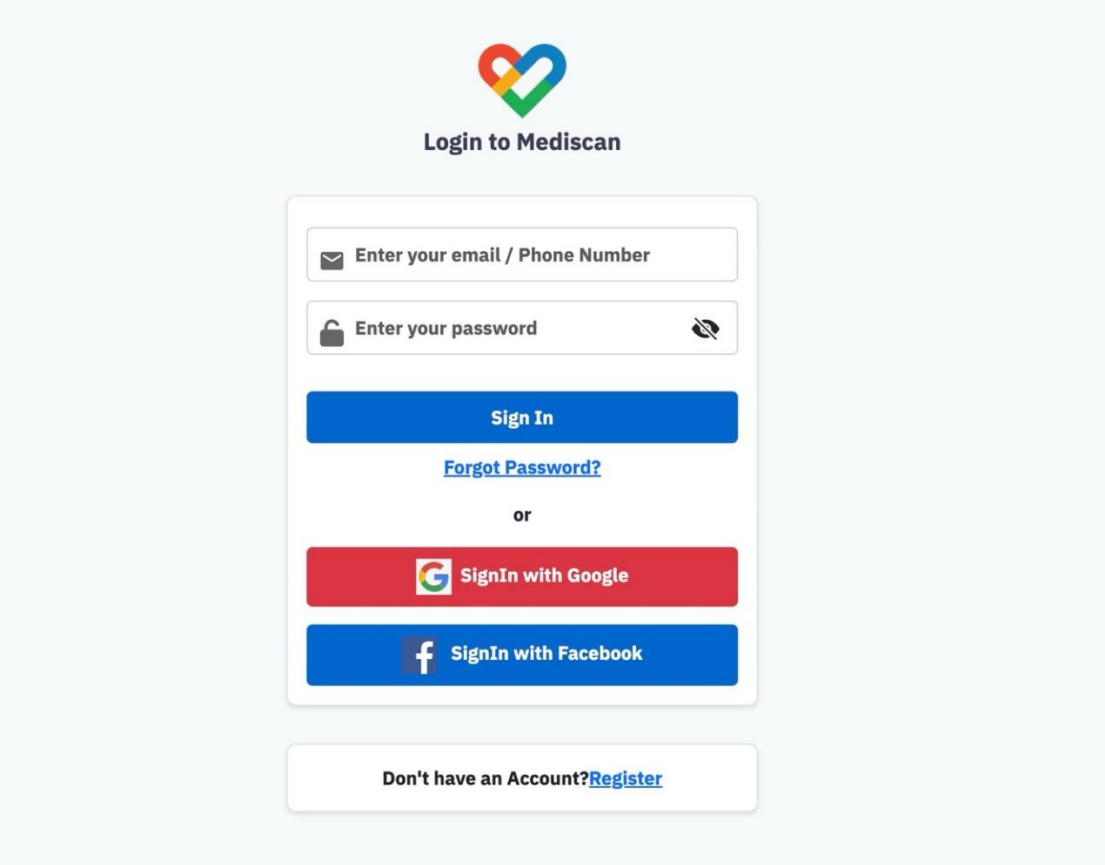

Figure 7.13.1.1 Login Section

7.13.2 SIGNUP

The Signup Page allows new users to create an account on the Health Tracker Website. Designed with a visually appealing two-column layout and secure form validation, it ensures smooth onboarding for users who wish to personalize their health tracking experience.

Built using ReactJS and styled with custom CSS, this page connects to a NodeJS/ExpressJS backend for account creation and supports real-time form interaction.

A. Purpose of the Page

- To allow users to register securely with email, phone, and password.
- To enable access to personalized features like dashboards and health analytics.
- To validate form inputs and confirm password matching before submission.
- To redirect successfully registered users to the login page.

Technical Overview

- React Hooks: use State, use Navigate
- Axios: For form submission to backend
- Form Validations: Client-side checks with alerts for:
    - Matching passwords
    - Terms acceptance
- Custom Styling:
    - .signup Section, .login Box, .form-group, .btn-big
    - Two-column responsive layout with branding and informative text

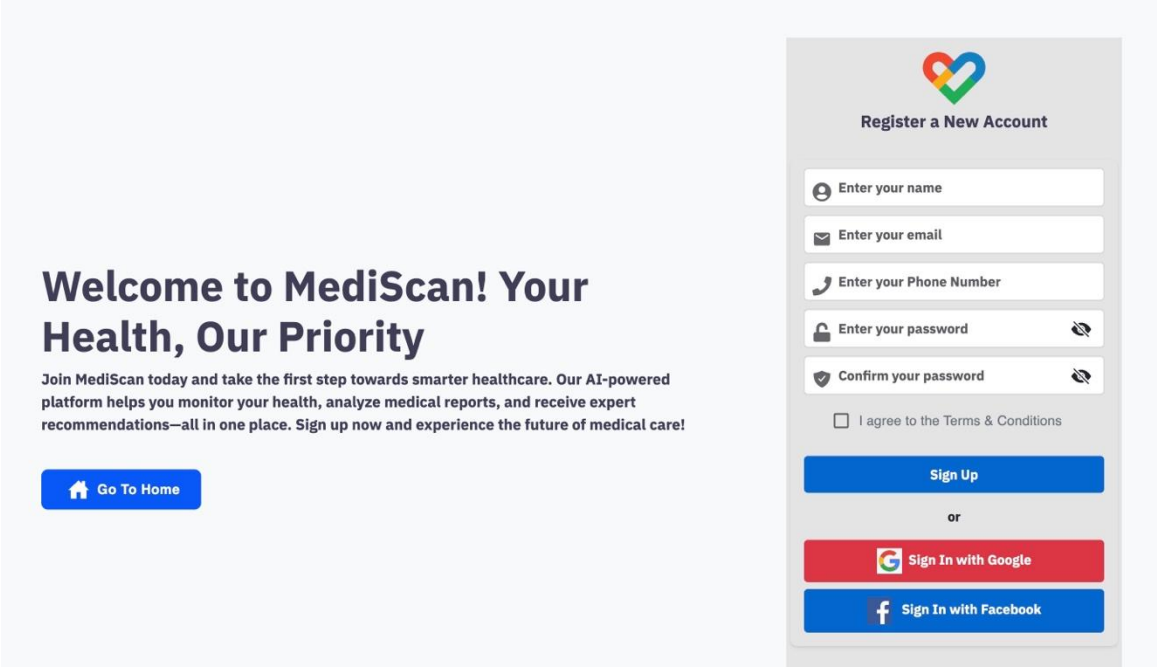

Figure 7.13.2.2 Sign Up Section

CHAPTER 8

TESTING

- **Black -Box Testing**
- **White-Box Testing**
- **Test Cases**

8.1 BLACK-BOX TESTING

Testing is a critical phase in the development of the Health Tracker Website. It ensures that the system functions as expected, is free of bugs, and meets the requirements specified. This chapter covers the testing strategies and methodologies used to validate the functionality, performance, and security of the Health Tracker Website.

Definition: Black-Box Testing is a testing methodology where the internal structure, design, or implementation of the software is not considered. The focus is on the functionality of the software, and tests are designed based on the software's specifications and requirements.

Details:

- **Functional Testing:**
    A. Dashboard Module:
        - Verify that the "Set Your Goal" button allows users to set goals for steps, calories burned, and workout duration.
        - Check that the circular progress bars accurately display the user's daily data for steps, calories, and workout duration.
        - Ensure that the health cards (blood pressure, diabetes, blood oxygen, weight, and height) display the correct data and graphical representations.

    B. Data Analysis Module:
        - Verify that the data analysis page provides accurate and comprehensive analysis of the user's health metrics.
        - Check that the personalized recommendations are generated based on the user's data.

    C. Medical Report Analyzer Module:
        - Verify that users can upload medical reports and receive detailed analysis.
        - Ensure that the instructions and upload button function correctly.

    D. Profile Module:

- Check that the user's medical and general information is accurately displayed.

E. Contact Us Module:

- Verify that the contact information is correct and that users can submit queries and feedback.

- **Usability Testing:**
  - Ensure that the website is user-friendly and easy to navigate.
  - Check that all features are accessible and that the interface is intuitive.
- **Regression Testing:**
  - Perform regression testing after each update or modification to ensure that existing functionality is not affected.

8.2 WHITE-BOX TESTING

Definition: White-Box Testing is a testing methodology where the internal structure, design, and implementation of the software are considered. The focus is on the internal workings of the software, and tests are designed to verify the flow of data, the functionality of internal components, and the overall integrity of the system.

Details:

A. **Unit Testing:**

- Test individual components and modules of the website to ensure they function correctly.
- Verify that the data processing algorithms for health metrics (e.g., BMI calculation, blood sugar level analysis) produce accurate results.
- Ensure that the data integration mechanisms (e.g., uploading medical reports) function as expected.

B. **Integration Testing:**

- Test the interaction between different modules and components of the website.
- Verify that data flows correctly between the dashboard, data analysis, medical report analyzer, profile, and contact us pages.

- Ensure that the system can handle data from various sources (e.g., wearable devices, manual entries) seamlessly.

C. **Security Testing:**

- Perform security testing to identify vulnerabilities and ensure that the system is secure.
- Verify that data encryption, access control, and other security measures are implemented correctly.
- Conduct penetration testing to identify and fix potential security weaknesses.

D. **Database Testing (CRUD Operations)**

- **Create:**

  Test the creation of new user accounts, health goals, and medical records. Ensure all new entries are stored accurately in **MongoDB**.
- **Read:** Validate data retrieval from the database for:
- Dashboard cards
- Health trend graphs
- Past 6-month tracking data for sleep, stress, and calories
- **Update:** Ensure updates to user goals, profile information, or re-uploaded medical reports reflect correctly and consistently across all pages.
- **Delete:** Test deletion of outdated or incorrect data and ensure it is removed cleanly without affecting other linked data.
- **Data-Integrity:** Check relationships between collections (if any) and validate that data does not break due to missing or corrupted entries.

E. Performance Testing

- **Load-Handling:**

  Test how the system performs when multiple users log in, set goals, or view charts simultaneously.
- **Database-Response-Time:**

  Measure response speed when fetching user health data across different modules, especially during dashboard load and data analysis visualization.

- **Real-Time-Chart-Updates:**
  Verify the smooth rendering and responsiveness of dynamic elements such as graphs and health status indicators.
- **Stress-Testing:**
  Evaluate how the system behaves under extreme data loads, such as uploading large medical files or simultaneous real-time updates.

8.3 TEST-CASES

Definition: Test Cases are specific scenarios and conditions used to validate the functionality and performance of the software. They are designed to cover various aspects of the system and ensure that it meets the specified requirements.

Details:

1. **Test Case 1: User Registration and Login**
   - **Description:** Verify that new users can create an account and log in successfully.
   - **Steps:**
     1. Navigate to the signup page.
     2. Enter valid user details (name, email, password).
     3. Submit the registration form.
     4. Navigate to the sign in page.
     5. Enter the registered email and password.
     6. Verify that the user is logged in successfully.
   - **Expected Result:** The user should be able to create an account and log in without any issues.
2. **Test Case 2: Dashboard Functionality**
   - **Description:** Verify that the dashboard page displays the correct data and functionality.
   - **Steps:**
     1. Log in as a registered user.
     2. Navigate to the dashboard page.
     3. Verify that the "Set Your Goal" button allows users to set goals.
     4. Check that the circular progress bars display the correct daily data.

5. Ensure that the health cards display the correct data and graphical representations.

- **Expected Result:** The dashboard should display accurate data and functionality.

3. **Test Case 3: Data Analysis Page**
    - **Description:** Verify that the data analysis page provides accurate and comprehensive analysis.
    - **Steps:**
        1. Log in as a registered user.
        2. Navigate to the data analysis page.
        3. Verify that the page displays the correct health metrics and analysis.
        4. Check that the personalized recommendations are generated based on the user's data.
    - **Expected Result:** The data analysis page should provide accurate and comprehensive analysis.
4. **Test Case 4: Medical Report Analyzer**
    - **Description:** Verify that users can upload medical reports and receive detailed analysis.
    - **Steps:**
        1. Log in as a registered user.
        2. Navigate to the medical report analyzer page.
        3. Follow the instructions to upload a medical report.
        4. Verify that the upload button functions correctly.
        5. Check that the detailed analysis is provided.
    - **Expected Result:** The medical report analyzer should function correctly and provide detailed analysis.
5. **Test Case 5: Profile Page**
    - **Description:** Verify that the profile page displays the correct medical and general information.
    - **Steps:**
        1. Log in as a registered user.
        2. Navigate to the profile page.

3. Verify that the user's medical and general information is accurately displayed.

- **Expected Result:** The profile page should display the correct medical and general information.

6. **Test Case 6: Contact Us Page**
    - **Description:** Verify that the contact us page provides the correct contact information and handles user queries and feedback.
    - **Steps:**
        1. Navigate to the contact us page.
        2. Verify that the contact information is correct.
        3. Submit a query or feedback.
        4. Check that the query or feedback is handled correctly.
    - **Expected Result:** The contact us page should provide the correct contact information and handle user queries and feedback.

CHAPTER 9

SYSTEM DESIGN

- **Website Architecture**
- **Class Diagram**
- **Use-Case Diagram**
- **Sequence Diagram**
- **Activity Diagram**
- **Data Flow Diagram**
- **Entity Relationship Diagram**
- **Website Control Flow**
- **Communication Flow Diagram**
- **Data Dictionery**

9.1 WEBSITE ARCHITECTURE

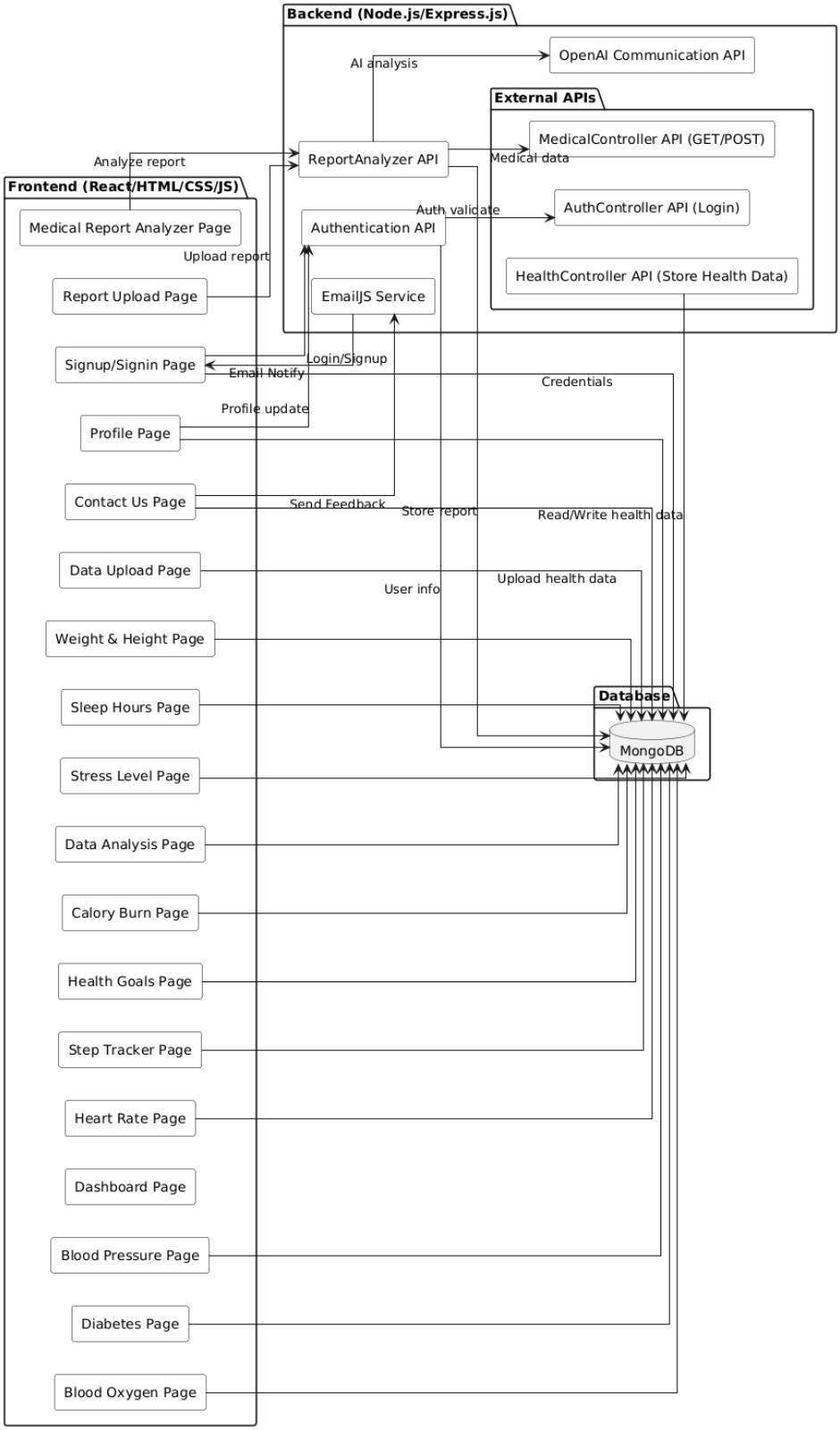

Figure 9.1.1 Website Architecture

9.2 CLASS DIAGRAM

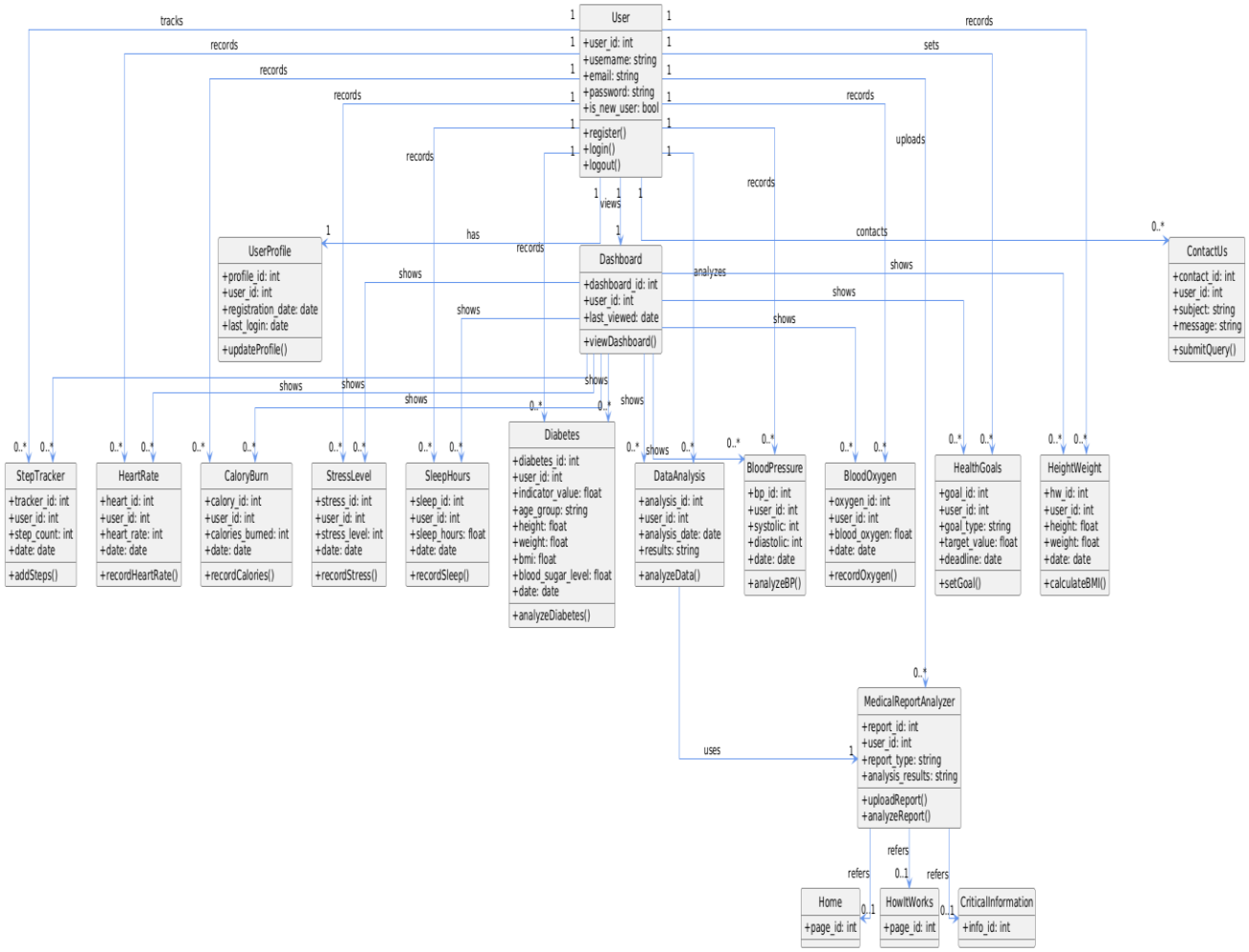

Figure 9.2.1 Class Diagram

9.3 USE CASE DIAGRAM

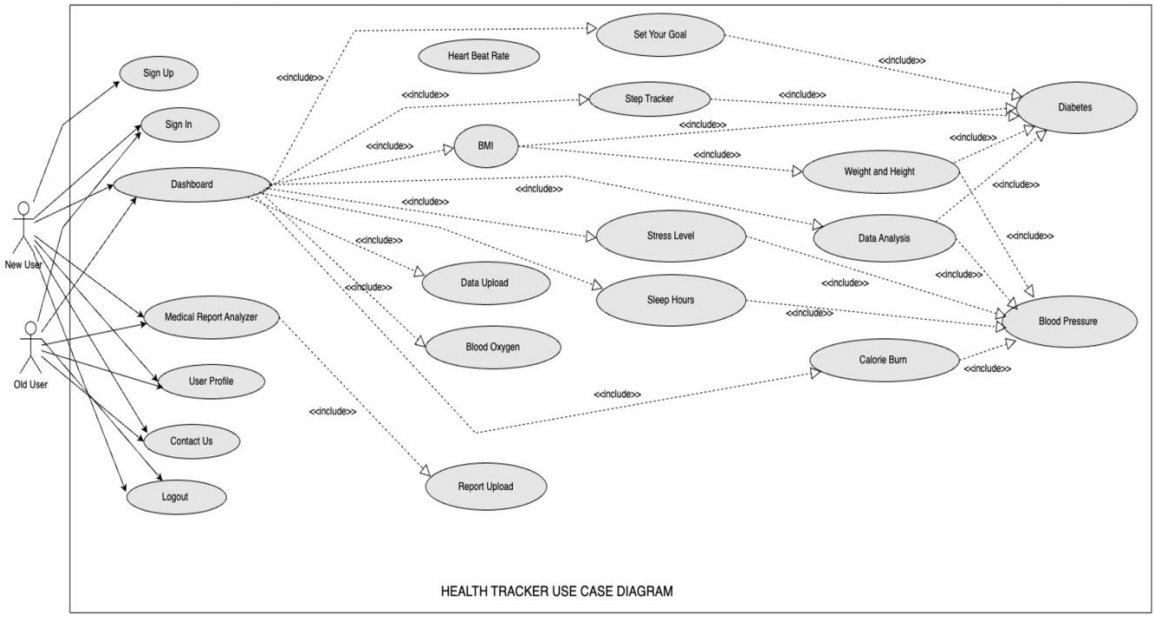

Figure 9.3.1 Use Case Diagram

9.4 SEQUENCE DIAGRAM

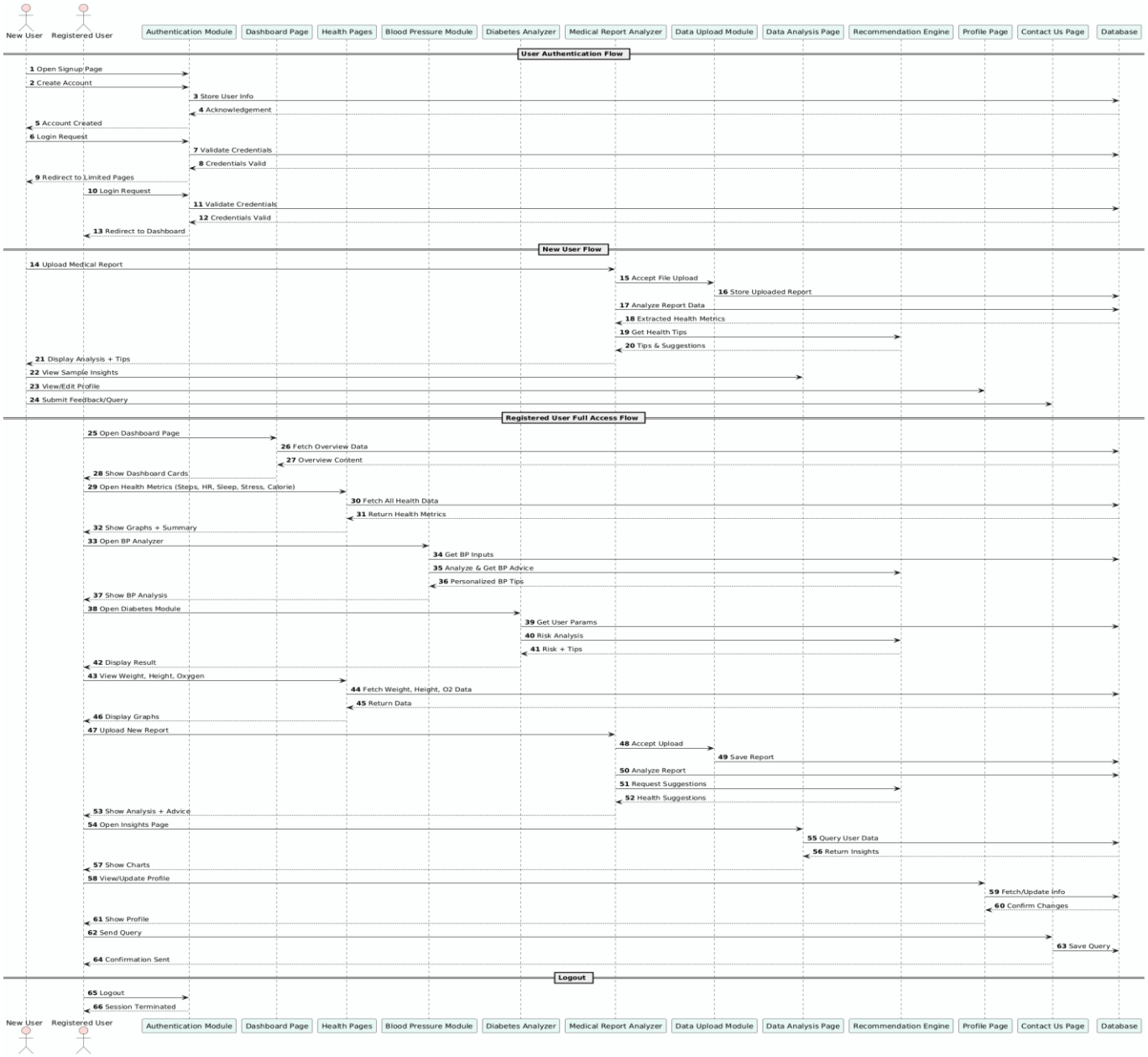

Figure 9.4.1 Sequence Diagram

9.5 ACTIVITY DIAGRAM

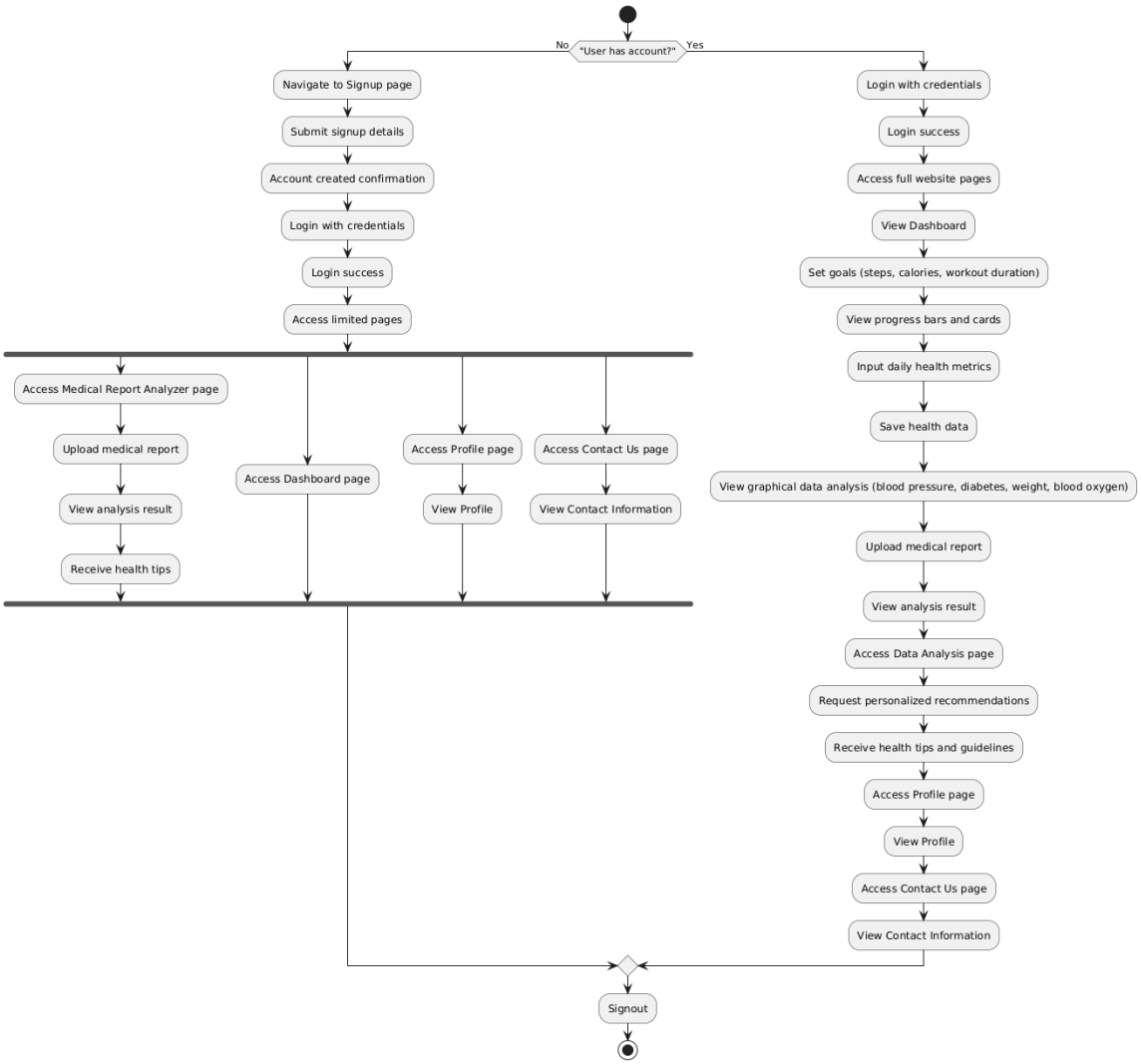

Figure 9.5.1 Activity Diagram

9.6 DATA FLOW DIAGRAM

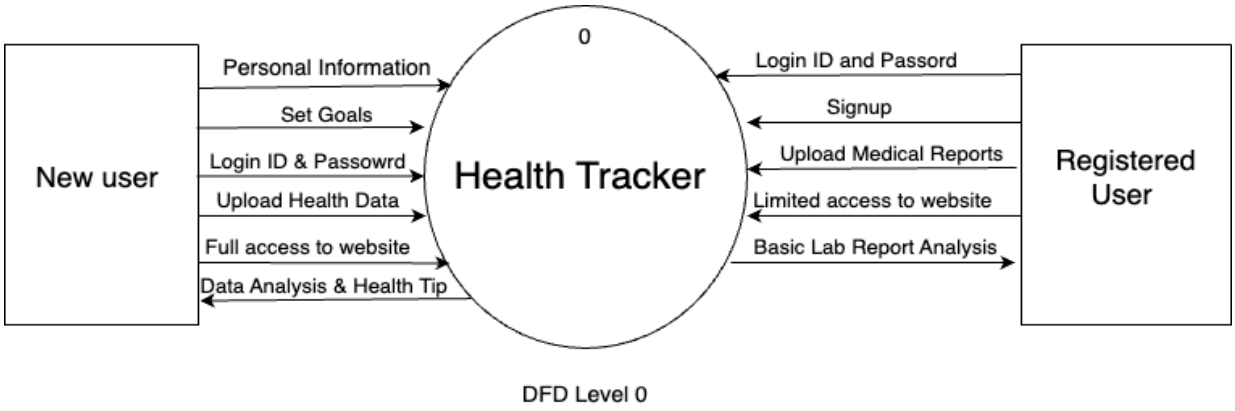

Figure 9.6.1 Data Flow Diagram-level 0

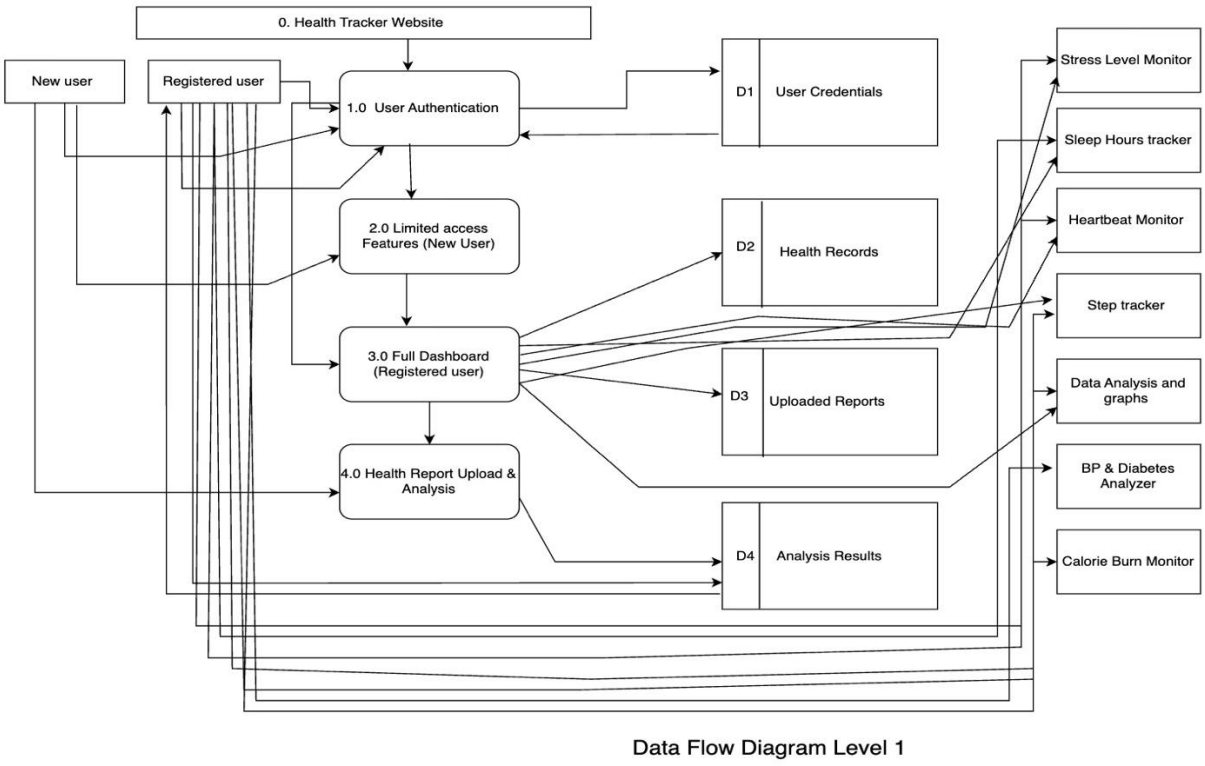

Figure 9.6.2 Data Flow Diagram-Level 1

9.7 ENTITY RELATIONSHIP DIAGRAM

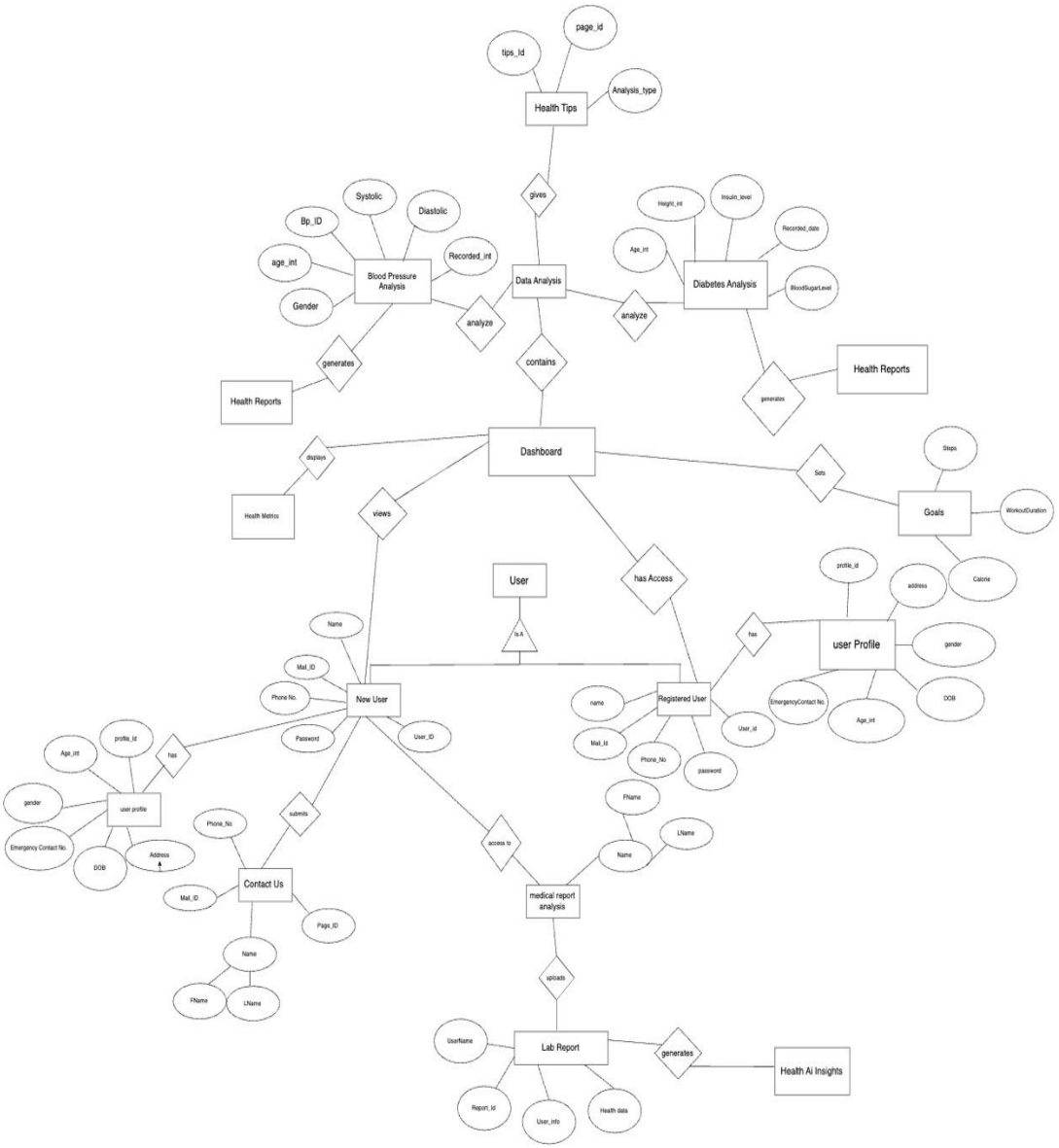

Figure 9.7.1 Entity Relationship

9.8 WEBSITE CONTROL FLOW DIAGRAM

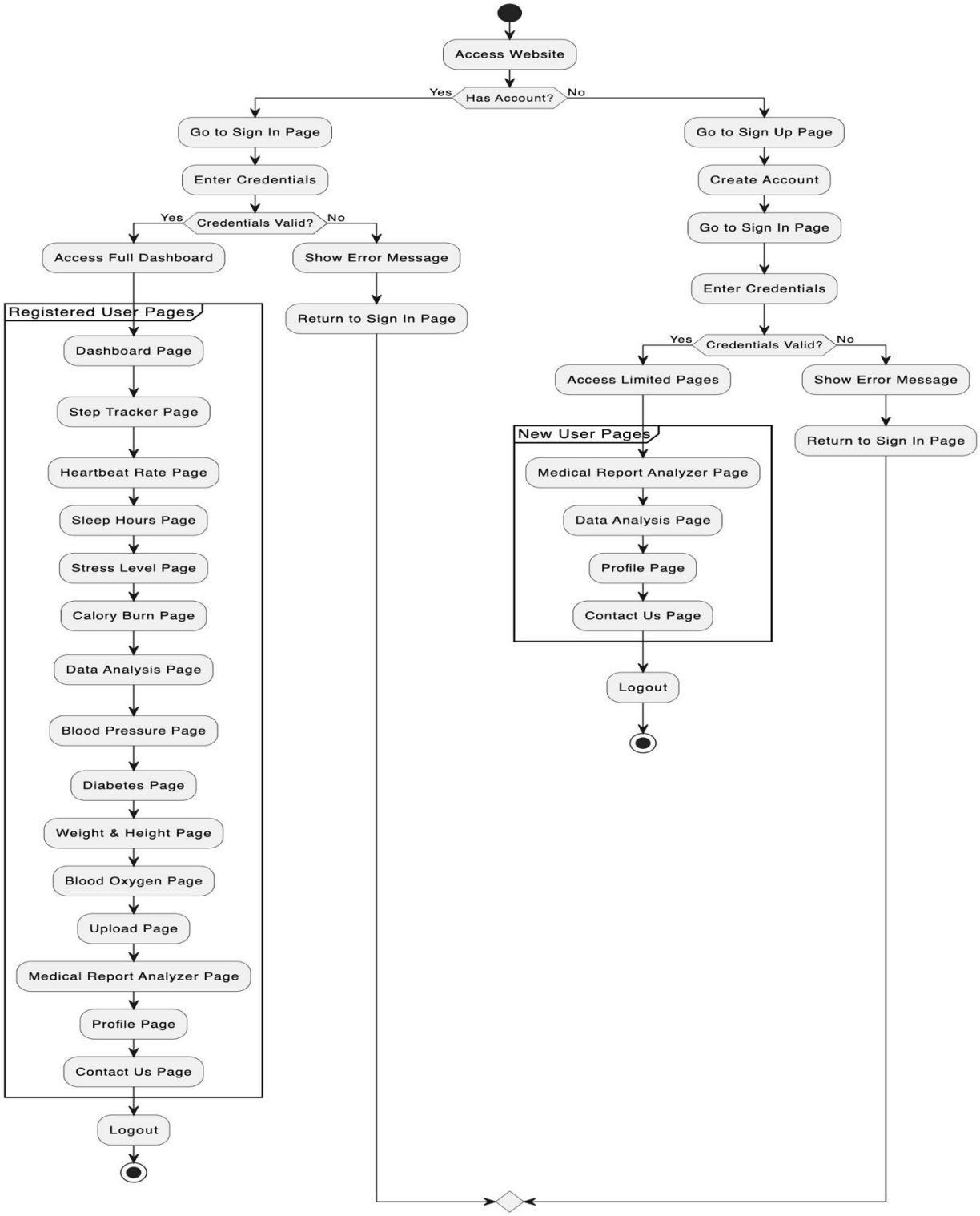

Figure 9.8.1 Website Control Flow

9.9 COMMUNCATION FLOW DIAGRAM

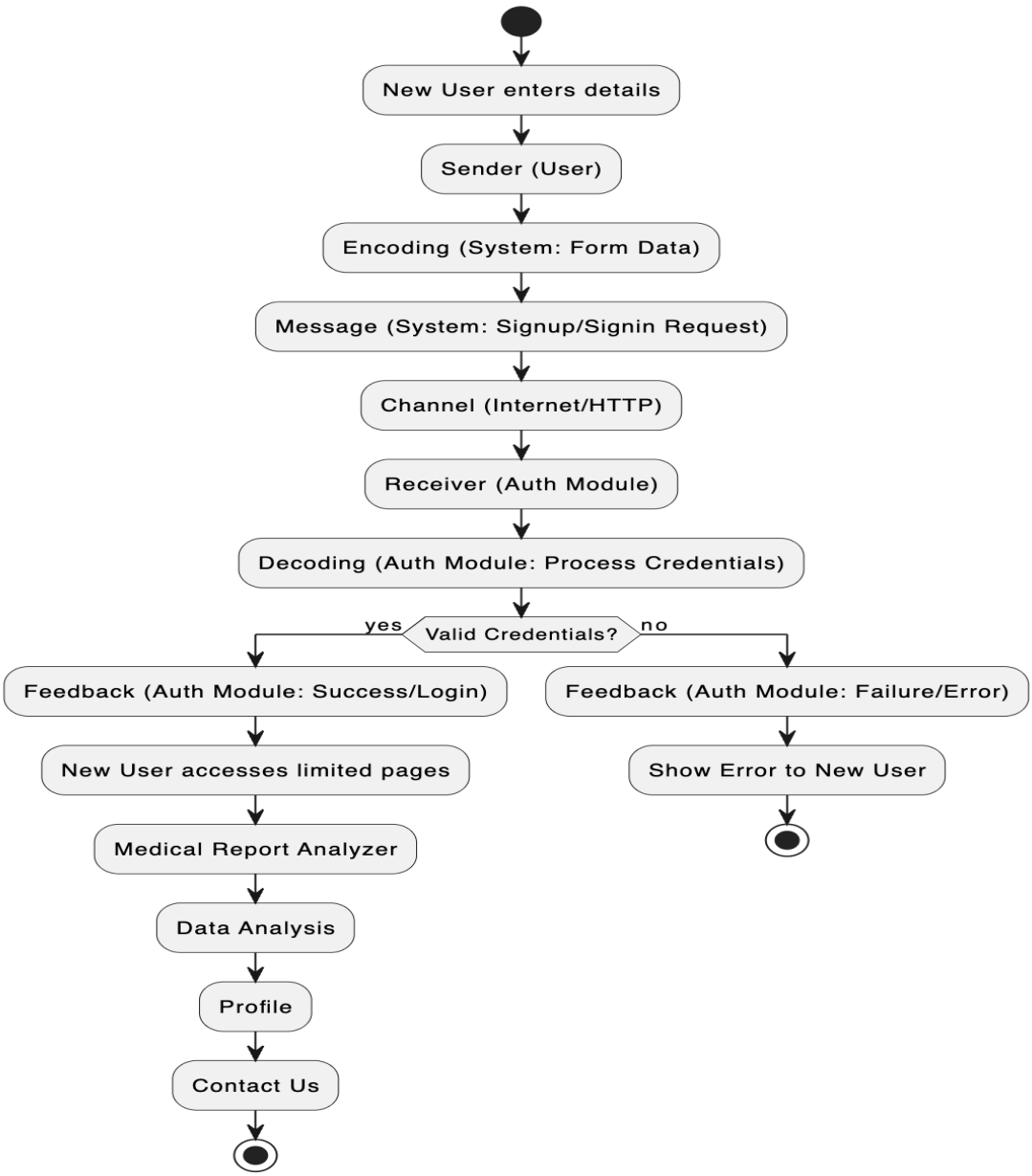

Figure 9.9.1 Communication Flow-1

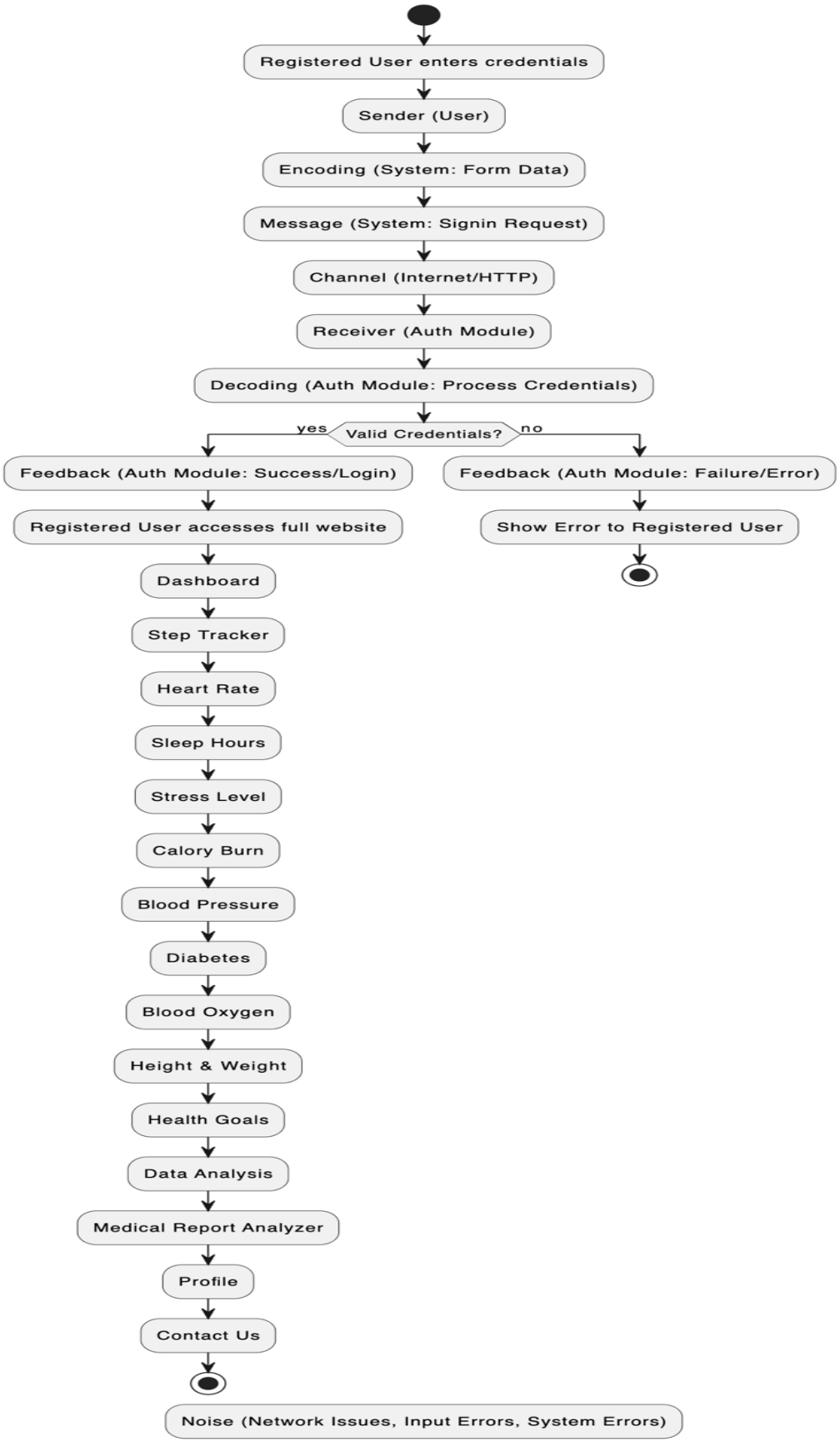

Figure 9.9.2 Communication Flow-2

9.10 DATA DICTIONERY

A Data Dictionary is a structured document or table that defines and describes all the data elements used within a system, database, or software application. It serves as a reference guide for developers, database designers, testers, and project stakeholders to understand the structure, format, and usage of data across the project.

A. User Table Data Dictionary

Field Name	Data Type	Size	Default Value	Constraints	Description
user_id	INT	11	Auto Increment	Primary Key, Not Null	Unique identifier for each user
username	VARCHAR	50	NULL	Not Null	Unique display name used for login
name	VARCHAR	100	NULL	Not Null	Full name of the user
email	VARCHAR	100	NULL	Unique, Not Null	User's email address
password	VARCHAR	255	NULL	Not Null	Encrypted password used for authentication
is_new_user	BOOLEAN	1	TRUE	Not Null	Flags whether the user is newly registered (true/false)

address	TEXT	-	NULL	Optional	User's residential address
gender	ENUM	-	NULL	CHECK (M/F/Other), Optional	Gender of the user
phone_number	VARCHAR	15	NULL	Unique, Optional	User's mobile number (with country code if needed)

Table 9.10.1 User Table DD

B. UserProfile Table

Field Name	Data Type	Size	Default Value	Constraints	Description
profile_id	INT	11	Auto Increment	Primary Key, Not Null	Unique identifier for each user profile
user_id	INT	11	NULL	Foreign Key, Not Null	Links the profile to the corresponding user in User table
name	VARCHAR	100	NULL	Not Null	Full name of the user
email	VARCHAR	100	NULL	Unique, Not Null	Email address of the user

phone_number	VARCHAR	15	NULL	Unique, Optional	Mobile number of the user (with country code if applicable)
password	VARCHAR	255	NULL	Not Null	Encrypted password (mirrors main user password field)
gender	ENUM	-	NULL	CHECK (M/F/Other), Optional	Gender of the user
dob	DATE	-	NULL	Optional	Date of birth of the user
message	TEXT	-	NULL	Optional	Optional user message or feedback
registration_date	DATE	-	CURRENT_DATE	Not Null	Date when the user registered
last_login	DATE	-	NULL	Optional	Timestamp of the most recent login
created_at	TIMESTAMP	-	CURRENT_TIMESTAMP	Not Null	Record creation timestamp
last_updated	TIMESTAMP	-	CURRENT_TIMESTAMP	Not Null	Last time the profile was updated

Table 9.10.2 User Profile DD

C. Dashboard Table

Field Name	Data Type	Size	Default Value	Constraints	Description
dashboard_id	INT	11	Auto Increment	Primary Key, Not Null	Unique identifier for each dashboard record
user_id	INT	11	NULL	Foreign Key, Not Null	Links the dashboard to a specific user
last_viewed	DATE	-	CURRENT_DATE	Optional	Last time the dashboard was accessed
steps_count	INT	6	0	Optional	Number of steps taken by the user
calories_burned	FLOAT	6,2	0.00	Optional	Total calories burned
workout_duration	INT	3	0	Optional	Duration of workout in minutes
sleep_hours	FLOAT	4,2	0.00	Optional	Total hours of sleep logged
stress_level	INT	3	NULL	Optional	Stress level (e.g., on a scale of 1–10)
heart_rate	INT	3	NULL	Optional	Heartbeat rate in beats per minute

blood_pressure_sys	INT	3	NULL	Optional	Systolic blood pressure (upper value)
blood_pressure_dia	INT	3	NULL	Optional	Diastolic blood pressure (lower value)
diabetes_level	FLOAT	5,2	NULL	Optional	Blood sugar level in mg/dL
height_cm	FLOAT	5,2	NULL	Optional	Height of the user in centimeters
weight_kg	FLOAT	5,2	NULL	Optional	Weight of the user in kilograms

Table 9.10.3 Dashboard DD

D. Medical Report Analyzer table

Field Name	Data Type	Size	Default Value	Constraints	Description
report_id	INT	11	Auto Increment	Primary Key, Not Null	Unique identifier for each medical report
user_id	INT	11	NULL	Foreign Key, Not Null	Links the report to a specific user
report_type	VARCHAR	50	NULL	Not Null	Type of medical report (e.g., Blood, Urine, ECG, X-ray)

report_file_name	VARCHAR	255	NULL	Not Null	Name of the uploaded report file
file_format	VARCHAR	10	'pdf'	CHECK (pdf)	Format of the uploaded file (supports only PDF in current version)
upload_date	DATETIME	-	CURRENT_TIMESTAMP	Not Null	Date and time when the report was uploaded
analysis_status analysis_results	ENUM	-	'pending'	CHECK (pending/completed/failed)	Status of the AI report analysis (pending, completed, failed)
keywords_detected	TEXT	-	NULL	Optional	Medical keywords or conditions identified from the report
summary_insights	TEXT	-	NULL	Optional	Summary of key insights and health recommendations
confidence_score	FLOAT	3,2	NULL	Optional	AI model's confidence level in interpreting the report
last_updated	TIMESTAMP	-	CURRENT_TIMESTAMP	Not Null	Last time the report or

					result was updated

Table 9.10.4 Medical Report Analyzer DD

E. Data Analysis Table

Field Name	Data Type	Size	Default Value	Constraints	Description
analysis_id	INT	11	Auto Increment	Primary Key, Not Null	Unique identifier for each data analysis record
user_id	INT	11	NULL	Foreign Key, Not Null	Links the analysis to a specific user
analysis_date	DATE	-	CURRENT_DATE	Not Null	Date when the analysis was conducted
results	TEXT	-	NULL	Optional	Summary or insight generated by analysis (possibly via OpenAI)
gender	ENUM	-	NULL	CHECK (M/F/Other), Optional	Gender of the user

age_group	VARCHAR	20	NULL	Optional	Age range (e.g., "18–25", "26–35")
height_cm	FLOAT	5,2	NULL	Optional	Height of the user in centimeters
weight_kg	FLOAT	5,2	NULL	Optional	Weight of the user in kilograms
sleep_hours	FLOAT	4,2	NULL	Optional	Average sleep duration per day
stress_level	INT	2	NULL	Optional	Self-reported stress level (scale 1–10)
exercise_frequency	VARCHAR	50	NULL	Optional	e.g., "Daily", "3 times/week", "Rarely"
smoking_status	ENUM	-	'Non-smoker'	CHECK (Smoker/Non-smoker)	Indicates if the user is a smoker
alcohol_consumption	ENUM	-	'None'	CHECK (None/Moderate/High)	Alcohol intake level

salt_intake	ENUM	-	'Moderate'	CHECK (Low/Moderate/High)	Dietary salt consumption level
systolic_bp	INT	3	NULL	Optional	Systolic blood pressure (upper value)
diastolic_bp	INT	3	NULL	Optional	Diastolic blood pressure (lower value)
heart_rate	INT	3	NULL	Optional	Heart rate in beats per minute
blood_sugar_level	FLOAT	5,2	NULL	Optional	Blood glucose level (mg/dL)
last_updated	TIMESTAMP	-	CURRENT_TIMESTAMP	Not Null	Last time the record was modified

Table 9.10.5 Data Analysis DD

A. Health Goals Table

Field Name	Data Type	Size	Default Value	Constraints	Description
goal_id	INT	11	Auto Increment	Primary Key, Not Null	Unique identifier for each health goal
user_id	INT	11	NULL	Foreign Key, Not Null	Links the goal to a specific user
goal_type	VARCHAR	50	NULL	Not Null	Type of goal (e.g., steps, calories, weight, sleep, stress)
target_value	FLOAT	6,2	NULL	Not Null	The target value user aims to reach
current_value	FLOAT	6,2	0.00	Optional	The current progress toward the goal
unit	VARCHAR	20	NULL	Optional	Unit of measurement (e.g., steps, kg, hours, bpm)
deadline	DATE	-	NULL	Optional	Target completion date for the goal

progress_status	ENUM	-	'In Progress'	CHECK (In Progress, Achieved, Missed)	Current status of the goal
created_at	TIMESTAMP	-	CURRENT_TIMESTAMP	Not Null	Timestamp when the goal was set
last_updated	TIMESTAMP	-	CURRENT_TIMESTAMP	Not Null	Timestamp of the most recent update to the goal

Table 9.10.6 Health Goals DD

CHAPTER 10

LIMITATION AND FUTURE ENHANCEMENT

- **Limitations**
- **Future Enhancement**

10.1 LIMITATION

While our Health Tracker Website has been thoughtfully designed and carefully developed to meet user needs in monitoring personal health data, there are certain constraints that currently limit the platform's overall scope and performance. These limitations stem from a combination of technical boundaries, feature availability, external dependencies, and design choices made during the current development phase.

The website successfully enables users to track vital health parameters, analyze historical trends, and receive basic recommendations. However, due to factors such as manual data entry requirements, limited automation, and lack of integration with external devices, the system may not fully replicate the functionality of professional-grade health management tools.

Moreover, scalability challenges, data accuracy issues, and security considerations pose barriers to extending the platform to a broader or clinical user base at this stage. By acknowledging these limitations, we aim to provide a clear understanding of the areas that require enhancement, optimization, and future development efforts to elevate the platform to its full potential.

A. Dependence on Manual Data Entry
    - Users without wearable devices must enter health data manually (e.g., blood pressure, weight, sugar levels).
    - This can lead to inaccurate tracking due to human error or irregular updates
    - Automated device integration is limited, reducing the overall efficiency of real-time tracking.

B. Limited Medical Report Understanding
    - The Medical Report Analyzer uses basic data extraction logic and keyword-based analysis.

- It does not yet support complex medical terminologies or varying report formats (e.g., handwritten scans, PDFs with tables, etc.).
- This limits its accuracy in interpreting non-standard or multilingual reports.

C. No Real-Time Health Monitoring

- The system currently does not offer live health monitoring from wearable devices or sensors.
- Data visualization is based on previously submitted or stored information, which may not reflect the user's current health status.

D. Basic Recommendation Engine

- The personalized health recommendations are rule-based (e.g., if blood pressure is high, suggest rest).
- There is no integration of AI-based prediction or adaptive suggestions based on long-term trends or user behavior.
- Users with complex medical conditions may not receive context-specific advice.

10.2 FUTURE ENHANCEMENT

To extend the functionality, improve the user experience, and increase the overall impact of the Health Tracker Website, several future enhancements have been identified. These improvements aim to address the current limitations while introducing advanced features to align the platform with modern digital healthcare standards.

A. Integration with Wearable Devices

- Enhancement: Connect the platform with fitness trackers like Fitbit, Mi Band, or Apple Watch for automatic data input.
- Benefit: Real-time and accurate health data collection without manual effort, improving user experience.
- Implementation: Use APIs provided by wearable device manufacturers to fetch data and sync with the user's profile on the platform.

B AI-Powered Health Insights

- Enhancement: Use machine learning models to provide smart, personalized health suggestions and early warnings.
- Benefit: Helps users make informed decisions and receive alerts about potential health risks.
- Implementation: Train models using user history data (e.g., diabetes, BP trends) and integrate them into the backend for prediction and alert generation.

C. NLP-Based Medical Report Analyzer

- Enhancement: Upgrade analyzer to process handwritten or scanned reports using OCR and Natural Language Processing.
- Benefit: Users can upload non-digital lab reports, making the platform more flexible and inclusive.

- Implementation: Integrate Tesseract.js (OCR) with NLP libraries like spaCy or OpenAI API to extract meaningful health data from uploaded documents.

D. Offline Data Entry and Sync

- Enhancement: Allow users to input data offline, which syncs once they regain internet access.
- Benefit: Improves accessibility in rural or low-network areas, ensuring continuous data entry.
- Implementation: Store data locally using Indexed DB or localStorage and sync with the server when connectivity is detected.

E. Mobile Application Development

- Enhancement: Launch a cross-platform mobile app version of the health tracker.
- Benefit: Better mobility and convenience, increases user engagement on the go.
- Implementation: Use React Native or Flutter to build apps for Android and iOS, reusing current APIs for data handling.

F. Advanced Health Data Visualization

- Enhancement: Introduce interactive graphs, health trend heatmaps, and correlation charts.
- Benefit: Helps users easily understand their health patterns and draw insights from visual data.
- Implementation: Use libraries like Chart.js or Recharts with React to create dynamic, customizable charts and dashboards.

G. Role-Based Access Control (RBAC)

- Enhancement: Provide access controls for users, admins, and doctors with custom permissions.
- Benefit: Ensures secure and specific access to sensitive health data while allowing sharing with medical professionals.
- Implementation: Add user roles in MongoDB schema and restrict page access based on authentication middleware in Express.js.

H. Security & Compliance Upgrades

- Enhancement: Strengthen security with two-factor authentication and comply with global data privacy laws.
- Benefit: Builds user trust and ensures safe handling of personal health data.
- Implementation: Use libraries like node-2fa for two-step login, add encryption for sensitive fields, and map features against GDPR/HIPAA checklists.

I. Diet and Lifestyle Tracking Module

- Enhancement: Add features to log diet, water intake, exercise, and mental wellness.
- Benefit: Offers users a 360-degree view of their health, not just vitals.
- Implementation: Extend existing database schema and UI to capture and analyze these additional parameters.

J. Smart Goal Recommendation System

- Enhancement: Suggest user-specific health goals based on age, history, and current data trends.
- Benefit: Personalized targets encourage better compliance and outcomes.
- Implementation: Use rule-based algorithms or regression models to suggest daily/weekly goals and update the "Set Your Goal" section dynamically.

K. . Multi-User & Family Profile Management

- Enhancement: Allow managing multiple family members' health profiles under one account.
- Benefit: Useful for caregivers, parents, and elderly management in one platform.
- Implementation: Add a relational structure in MongoDB for sub-profiles and update frontend logic to toggle between different users.

L. . Cloud-Based Deployment and Scalability

- Enhancement: Host the backend on scalable cloud platforms like AWS or Azure.

- Benefit: Improves performance, uptime, and scalability for more users and larger data volumes.
- Implementation: Use cloud services like AWS EC2, S3, and MongoDB Atlas for deployment, storage, and database hosting.

CHAPTER 11

CONCLUSION

- **Conclusion**

11.1 CONCLUSION

As we reach the final phase of our Health Tracker Website project, it is important to reflect on the journey, the progress made, and the impact of our work. This project aimed to create a comprehensive, user-friendly platform that empowers users to monitor and understand their health metrics in a more informed and accessible way. Throughout the development lifecycle, from ideation and design to implementation and testing, we encountered various technical and logical challenges that not only shaped the outcome of our system but also significantly enhanced our learning experience.

The conclusion section outlines our key achievements, the challenges we faced and overcame, the valuable lessons we gathered along the way, as well as the possibilities that lie ahead for further development. This reflective summary not only highlights the effectiveness of our solution but also sets the foundation for future improvements and innovation in health tech solutions.

A. Key Achievements

- Throughout the development of the Health Tracker Website, we successfully achieved several milestones that reflect the core objectives of our project:
- Built a full-stack responsive health monitoring platform with modern technologies like React.js, SCSS, Node.js, Express.js, MongoDB, and OpenAI API.
- Developed a dynamic Dashboard that presents real-time health metrics such as blood pressure, blood oxygen, weight, height, diabetes levels, and more — all visualized through card components and interactive charts.
- Implemented a Medical Report Analyzer using Generative AI, helping users understand complex medical terms without needing expert assistance.
- Introduced features like Set Your Goal, User Authentication, Personalized Recommendations, and Data Trend Analysis to promote a healthier lifestyle through informed decisions.
- Achieved a clean, user-friendly interface with responsiveness and accessibility across devices.

B. Challenges and Learnings

- The journey of building this platform brought valuable experiences and lessons:
- Faced integration issues between frontend and backend during real-time data flow, which taught us deeper debugging and API handling techniques.
- Ensured proper data accuracy in health analytics, which required validation techniques and the application of logic in data processing functions.
- Learnt how to manage and manipulate health data across multiple modules with the help of NoSQL databases (MongoDB) and RESTful APIs.
- Understood the importance of user privacy and security, especially in handling sensitive health data, which led to deeper exploration of authentication and encryption techniques.
- Implemented Agile Development practices, learning how to break down complex modules into achievable sprints and deliver results efficiently.

C. Future Scope

- There is vast potential to expand and enhance this platform:
- Integration with smart wearable devices for automatic real-time health tracking.
- AI-based prediction models to provide early alerts for potential health risks.
- A mobile app version for better portability and real-time notifications.
- Role-based access control for sharing health data with doctors or caregivers securely.
- Mental health and lifestyle tracking modules to provide a holistic view of a user's well-being.

D. Final Thoughts

This project allowed us to not only explore our technical skills in full-stack web development but also understand the real-world impact of health technology in improving everyday lives. It was an opportunity to bridge the gap between data and health awareness, offering a solution that is practical, informative, and engaging.

We've also realized the value of building user-centric applications — where simplicity, reliability, and relevance of information can make a meaningful difference in people's health journeys.

In conclusion, the Health Tracker Website stands as a testament to our technical growth, collaborative effort, and passion to create something valuable in the healthcare domain. While there is room for improvement and future advancement, the foundation laid by this project sets a strong base for future development and innovation. We are proud of what we have achieved and excited for the potential that lies ahead.

Bibliography

REFERENCES

A. React App (n.d.). A JavaScript library for building user interfaces.

Link: https://react.dev/learn/creating-a-react-app

React is an open-source JavaScript library maintained by Meta (formerly Facebook).

It allows developers to build fast, interactive UIs using a component-based architecture.

React's virtual DOM improves performance by minimizing updates to the real DOM.

B. Visual Studio Code (n.d.) A source-code editor developed by Microsoft.

Link: https://code.visualstudio.com/

VS Code is a free, lightweight, and powerful code editor that supports multiple programming languages.

It features built-in Git support, syntax highlighting, debugging, and an extensive marketplace of extensions.

It is widely used by developers for web and software development.

C. Postman (n.d.) A platform for API development and testing.

Link: https://www.postman.com/

Postman provides tools for building, testing, and documenting APIs efficiently.

It helps developers simulate HTTP requests and inspect responses, making API development easier.

It's essential for backend testing and integration with frontends.

D. SmartDraw (n.d.) A diagram and flowchart creation software.

Link:

https://www.smartdraw.com/?srsltid=AfmBOop81w6v5Lvt_XIQbJj6kX8ElhJQDd-BMxxSKiMzud5e4XPzY3o9

SmartDraw helps users create diagrams, organizational charts, mind maps, and network designs.

It provides templates and symbols to easily visualize workflows or architectures.
Useful for planning and designing system architectures and workflows.

E. **Diagrams.net (n.d.)** A free, online diagramming tool (formerly Draw.io).
Link: https://app.diagrams.net/
Diagrams.net allows users to create and export flowcharts, wireframes, UML diagrams, and more.
It can integrate with cloud storage like Google Drive and OneDrive.
It is commonly used for software architecture and planning documentation.

F. **MongoDB (n.d.)** A NoSQL database program.
Link: https://www.mongodb.com/
MongoDB stores data in JSON-like documents, offering flexibility and scalability.
It is widely used in modern web applications due to its performance and ease of use.
Developers prefer it for working with large-scale data and real-time applications.

G. OpenAI ChatGPT (n.d.) A conversational AI model developed by OpenAI.
Link: https://openai.com/index/chatgpt/
ChatGPT is designed to generate human-like text responses in conversational contexts.
It can answer questions, write essays, code, summarize content, and more.
It's used in customer service, writing assistance, education, and development.

H. W3Schools (n.d.) A web development learning platform.
Link: https://www.w3schools.com/
W3Schools offers tutorials and references on web development technologies such as HTML, CSS, JavaScript, etc.
It is beginner-friendly and includes interactive examples.
It’s one of the most popular resources for learning front-end web development.

I. **Stack Overflow (n.d.)** A question-and-answer site for programmers.
Link: https://stackoverflow.com/questions

Stack Overflow allows developers to ask and answer coding-related questions. It has a vast repository of solutions, bug fixes, and advice from professionals. It's a go-to platform for solving programming challenges.

J. GeeksforGeeks (n.d.) A computer science portal for learning and practice.
Link: https://www.geeksforgeeks.org/
GeeksforGeeks provides tutorials, interview prep, competitive programming guides, and practice problems.
It is widely used by students and professionals preparing for coding interviews.
It covers data structures, algorithms, and system design in depth.

K. MyMeds247 (n.d.) An online medical consultation and pharmacy platform.
Link: https://mymeds247.com/
MyMeds247 provides healthcare services such as online doctor consultations and medicine delivery.
Users can also access health resources and manage prescriptions digitally.
It helps bridge the gap between patients and medical providers online.

L. Pexels (n.d.) A website offering free stock photos and videos.
Link: https://www.pexels.com/
Pexels offers high-quality, royalty-free images and videos for personal or commercial use.
It's commonly used by designers, developers, and marketers.
The content is contributed by a global community of photographers.

M. Chart.js (n.d.) A JavaScript library for creating charts.
Link: https://www.chartjs.org/docs/latest/
Chart.js allows developers to visualize data with simple yet flexible charts like line, bar, pie, etc.
It is built on HTML5 canvas and integrates easily with frontend frameworks like React.
Perfect for dashboards and data visualization in web applications.

N. React Icons (n.d.) A library providing popular icons for React applications.

Link: https://react-icons.github.io/react-icons/

React Icons brings various icon packs like Font Awesome, Material Icons, and more into React projects.

Developers can easily add icons with a consistent interface and custom styling.

It improves UI/UX and enhances the visual appeal of applications.