SMART HEALTH MANAGEMENT SYSTEM

A Project Report Submitted in the partial fulfillment of the requirements of the course titled

“Problem Solving Through Programming (JAVA)”

BACHELOR OF TECHNOLOGY

**In**

# DEPARTMENT OF FRESHMAN ENGINEERING

**By**

# B. MANJU BHARGAVI 2520040003

K. HARINI **2520040011**

**P. SRIJA** **2520090053**

**G. SHIVANI 2520090046**

**S. SRIKAR PRATHAP 2520040006**

**P. SRIKAR 2520080038**

**M. ADITH KUMAR REDDY 2520030559**

Under the Esteemed Guidance of

**Dr JAYARAM BOGA**

**Designation**

**Department of Freshman Engineering**

****

**K L (Deemed to be) University**

**DEPARTMENT OF FRESHMAN ENGINEERING**

****

**Declaration**

The Project Report entitled “**Smart Health Management System”** is a record of Bonafide work of **B. Manju Bharghavi - 2520040003, K. Harini – 2520040011 , P. Srija– 2520090053 , G. Shivani – 2520090046 , S. Srikar Prathap – 2520040006 , P. Srikar – 2520080038 , M. Adith Kumar Reddy – 2520030559 ,** submitted in partial fulfillment of the requirements of the course titled “Problem Solving Through Programming (JAVA)” under the B.Tech Ist Year Trimester - I program in Department of Freshman Engineering at K L University. The results presented in this report have not been copied from any other department, university, or institute.

# B. MANJU BHARGHAVI – 2520040003

# K. HARINI – 2520040011

# P. SRIJA – 2520090053

# G. SHIVANI – 2520090046

# S. SRIKAR PRATHAP– 2520040006

# P. SRIKAR – 2520080038

# M. ADITH KUMAR REDDY- 2520030559

**K L (Deemed to be) University**

# DEPARTMENT OF FRESHMAN ENGINEERING

****

**CERTIFICATE**

This is certify that the java project based report entitled **Smart Health Management System** is a bonafide work done and submitted by **B. Manju Bharghavi – 2520040003 , K. Harini – 2520040011, P. Srija– 2520090053 , G. Shivani – 2520090046 , S. Srikar Prathap – 2520040006 , P. Srikar – 2520080038 , M. Adith Kumar Reddy – 2520030559 ,** in partial fulfillment of the requirements of the course titled “Problem Solving Through Programming (JAVA)” under the B.Tech Ist Year Trimester - I program in Department of Freshman Engineering, K L (Deemed to be University), during the academic year **2025-2026.**

**Signature of the Guide**

**Signature of the Course Coordinator Signature of the HOD**

**ACKNOWLEDGEMENT**

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our wish to express my sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my Course Coordinator **Dr Y Ashok**, and my guide **Dr Jayaram Boga,** Department of Freshman Engineering which cannot be expressed in words for his/her tremendous support, encouragement and guidance for this project.

We express our gratitude to **Dr. N. Chaitanya Kumar**, Head of the ***Department for Freshman Engineering*** for providing us with adequate facilities, ways and means by which we are able to complete this project-based Lab.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project. Finally, I sincerely thank my parents, friends and classmates for their kind help and cooperation during my work.

# B. MANJU BHARGHAVI – 2520040003

# K. HARINI – 2520040011

# P. SRIJA – 2520090053

# G. SHIVANI – 2520090046

# S. SRIKAR PRATHAP– 2520040006

# P. SRIKAR – 2520080038

# M. ADITH KUMAR REDDY- 2520030559

**ABSTRACT**

The Smart Health Management System is a console-based Java application designed to help users monitor and evaluate their everyday health habits in a simple and interactive manner. The system collects four essential health parameters like steps walked, water intake, sleep duration, and calories consumed and processes them to generate personalized suggestions for improving overall wellness. The program validates user inputs, stores the data securely in an external file for future reference, and displays a detailed summary along with personalized health recommendations. By combining technology with basic fitness insights, this project offers an easy-to-use digital wellness companion that encourages healthier daily routines**.**

This project shows how Java concepts like classes and objects, constructors, encapsulation, inheritance, polymorphism, an abstract class, interfaces, the collection framework (HashMap), exception handling, file handling, methods, if-else decision making, loops, recursion-based input validation, and basic data types and variables can be used in a real example. Overall, it is a simple and useful tool to help people understand their daily habits better.

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.**  **No.** | **Chapters** | **Topics** | **Page.no** |
|  |  | Acknowledgement | iv |
|  |  | Abstract | v |
| 1 | Introduction | * 1. Background of the project   2. Problem statement | 1  2 |
| 2 | System Architecture | * 1. High-level architecture diagram/Block diagram   2. Class Diagram | 3  4 |
| 3 | CO’s Attainments | * 1. CO1 Attainment   2. CO2 Attainment   3.3 CO3 Attainment  3.4 CO4 Attainment   * 1. CO5 Attainment   3.6 CO6 Attainment | 5  6  7  8  9  10 |
| 4 | Screen Shots | * 1. Screen Shots | 11-16 |
| 5 | Testing | * 1. Test cases and results | 17-21 |
| 6 | Future Enhancements | * 1. Planned features   2. Possible integrations or optimizations | 22  23 |
| 7 | Conclusion | * 1. Summary of the project   2. What was achieved   7.3 Skills learned during development | 24  25  26 |
| 8 | References | - Books, tutorials, documentation sites used | 27-28 |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | Appendices | * Installation/setup instructions * User manual or guide   -Geo Tag photos with guide  -Review forms with guide signatures | 29 |

# CHAPTER -1 INTRODUCTION

## Background of the Project

Today, many people want to stay healthy but because of busy schedules they often forget to track simple daily habits like how much they walked, how much water they drank, how long they slept or how many calories they consumed. These small habits may look simple, but they have a huge impact on our overall well-being. Most health apps are complicated, require internet, or come with too many features that confuse regular users. So, I wanted to create something simple, quick and useful, something that helps people understand their daily health in just a few seconds.

That’s how the idea of my project ‘Smart Health Management System’ was born.  
It is a straightforward Java based console application where a person enters their basic daily health values and immediately receives meaningful suggestions. It’s like having a small digital health guide that motivates you to maintain good habits every day. The main aim behind developing this project is to show how programming, especially Java concepts like OOP, collections, exception handling, and file handling, can be used to build something practical that helps people in their everyday life.

**1.2 Problem statement**

In today’s busy lifestyle, most people neglect basic daily health habits. They often forget to walk enough, do not track how much water they drink, sleep irregularly, and consume calories without any awareness. Because they don’t monitor these activities, they fail to understand their health patterns and cannot improve them effectively .There is a clear need for a simple, offline, and easy-to-use tool that allows people to quickly record their daily health data and receive meaningful suggestions instantly.

The core problem is:  
How can we make daily health tracking simple, fast, and accessible for everyone using basic technology? My project solves this by offering a clean console-based Java application that collects daily health values, provides smart suggestions, and saves the data for future reference.

# CHAPTER -2 SYSTEM ARCHITECTURE

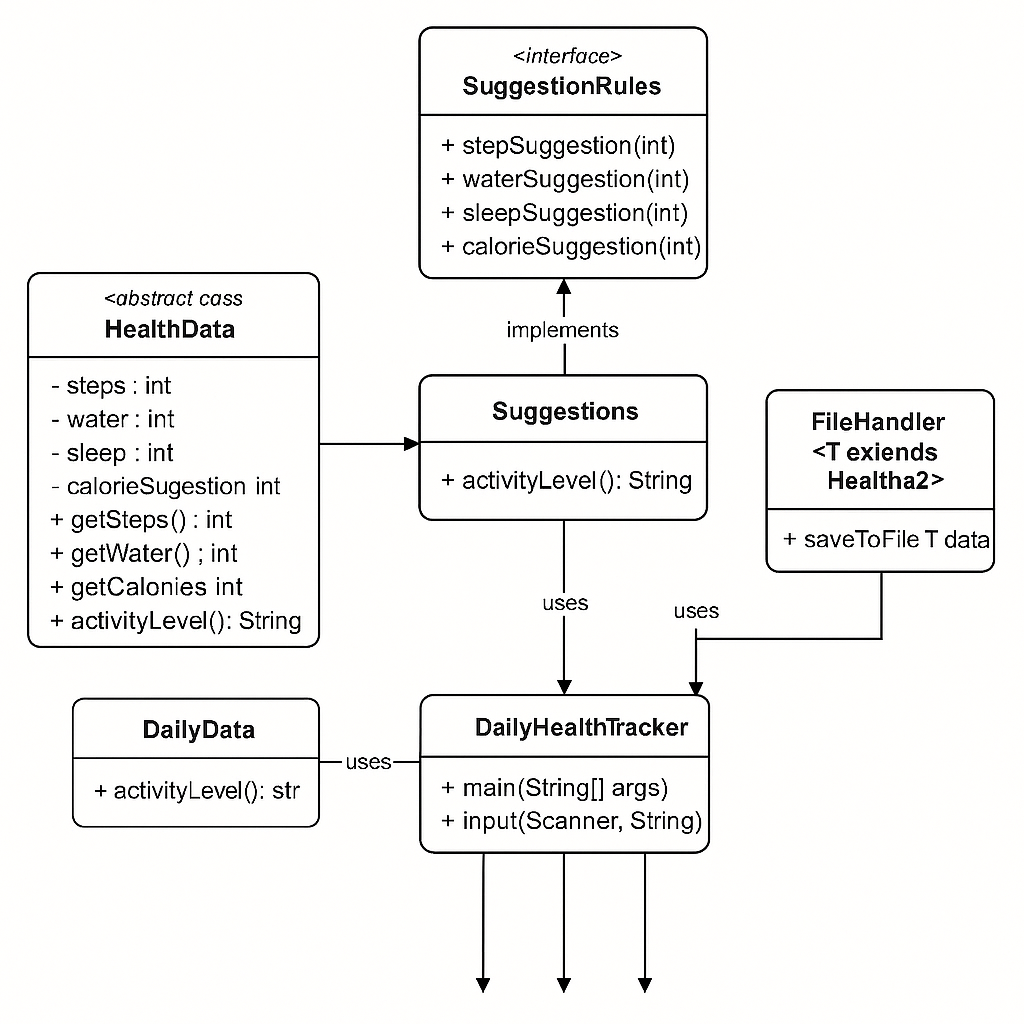
## High-level architecture diagram

A diagram of a software system

AI-generated content may be incorrect.

In my project, users interact with the Daily Health Tracker through a console-based menu to enter their health data or use demo values. The data is managed by the Daily Data object, which extends the abstract Health Data class. Suggestions are generated by the Suggestions Module and displayed alongside the user data, including a health score and recommendations. Finally, all information is saved persistently in **healthdata.txt** using a generic File Handler.

* 1. **Class Diagram**

****

This class diagram explains how the Daily Health Tracker system is organized using OOP concepts. Health data is an abstract class that stores core health information and is inherited by daily data which provides actual daily records. The suggestion rules interface defines suggestion methods, and **Suggestions** implements those rules to give personalized health advice. The main class daily health tracker creates and connects all objects while file handler stores the recorded data into a file for future use.

# CHAPTER -3 CO’s ATTAINMENT

**3.1 CO1 Attainment**

|  |  |
| --- | --- |
| **CO1 Syllabus** | **CO1 Concepts Included in Project** |
| Apply fundamental programming constructs such as data types, operators, conditional and iterative statements in Java to develop logic-based solutions for basic computational problems. Students will learn to design simple algorithms, trace execution, and validate logic through hands-on coding tasks**.** | Data types, operators, conditions, loops. |

**3.1.1 Scenario’s for CO1 implementation.**

In my project, I used int data types to store steps, water, sleep, and calories. I used relational operators like < and > to compare the values. The suggestions are decided using if–else conditions. For input validation, I used a while loop that keeps asking until the user enters a valid number.

**3.1.2 CO1 code screen shot.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**3.2 CO2 Attainment**

|  |  |
| --- | --- |
| **CO2 Syllabus** | **CO2 Concepts Included in Project** |
| Design, trace, and optimize algorithms using one-dimensional and two-dimensional arrays to solve mathematical, quantitative, and real-world problems efficiently through search, sort, and matrix manipulation techniques. | One-dimensional Array |

**3.2.1 Scenario’s for CO2 implementation.**

In my project, I used a HashMap to temporarily store the user’s inputs like steps, water, sleep, and calories. This shows the use of collections, which work like advanced arrays for storing and accessing multiple values easily.

**3.2.2 CO2 code screen shot.**

A computer screen shot of text

AI-generated content may be incorrect.

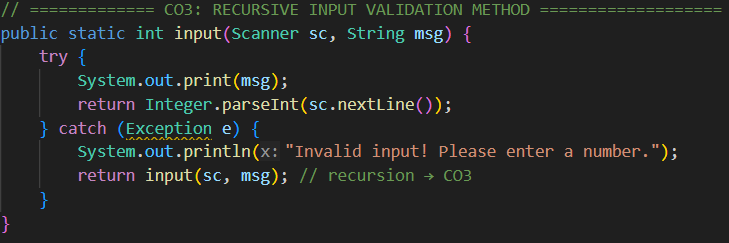
**3.3 CO3 Attainment**

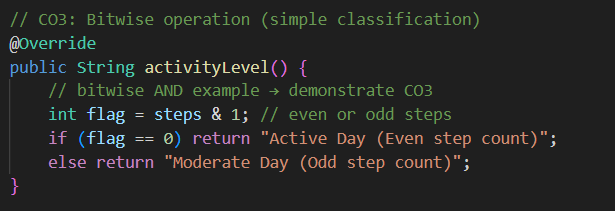
|  |  |
| --- | --- |
| **CO3 Syllabus** | **CO3 Concepts Included in Project** |
| Construct and evaluate advanced problem-solving logic using strings, recursion, and bitwise operations for solving complex mathematical, pattern-based, and combinatorial problems relevant to competitive coding platforms. | Recursive method , bitwise operation ,  string operations (trim, substring, etc.) |

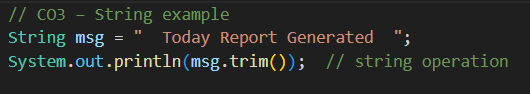
* + 1. **Scenario’s for CO3 implementation.**

In my project, I used String handling to create clear output messages and suggestions for the user. I also used String concatenation to neatly display the final summary and health report. These string operations help make the overall output more readable and user-friendly.

* + 1. **CO3 code screen shot.**





****

* 1. **CO4 Attainment**

|  |  |
| --- | --- |
| **CO4 Syllabus** | **CO4 Concepts Included in Project** |
| Develop structured and modular programs by applying object-oriented programming principles such as encapsulation, abstraction, and modularization using Java classes, methods, and constructors. | Classes, objects , constructors, encapsulation, abstraction. |

* + 1. **Scenario’s for CO4 implementation.**

In my project, I used classes, objects, constructors, and methods to keep the program structured and organized. Each class has a specific role, like storing data, giving suggestions, or handling files. This makes the entire system modular, easy to understand, and easier to maintain**.**

**3.4.2 CO4 code screen shot.**

A screen shot of a computer program

AI-generated content may be incorrect.

**3.5 C05 Attainment**

|  |  |
| --- | --- |
| **CO5 Syllabus** | **CO5 Concepts Included in Project** |
| Design extensible and reusable Java programs employing inheritance, polymorphism, abstract classes, interfaces, and reflection API to solve domain-oriented problems with clarity and maintainability. | Inheritance, polymorphism, interface, abstract class. |

**3.5.1 Scenario’s for CO5 implementation.**

In my project, I used inheritance by extending a base class and applied polymorphism through method overriding. I also created an abstract class to define a common structure and used an interface for suggestion related methods. These concepts make my project more flexible, reusable and easier to expand with new features**.**

**3.5.2 CO5 code screen shot.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A computer screen with text

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.**

**3.6 C06 Attainment**

|  |  |
| --- | --- |
| **CO6 Syllabus** | **CO6 Concepts Included in Project** |
| Implement robust, scalable, and generic Java applications integrating exception handling, file I/O, generics, and collections framework, along with functional programming constructs to handle real-world data-driven tasks. | Exception handling, generics, file I/O |

**3.6.1 Scenario’s for CO6 implementation.**

In my project, I used exception handling to catch wrong inputs and prevent the program from crashing. I used File writer for saving daily health data into a text file and a HashMap from the collections framework to temporarily store values. These features make the program more reliable, practical, and user-friendly.

**3.6.2 CO6 code screen shot.**

**A screen shot of a computer program

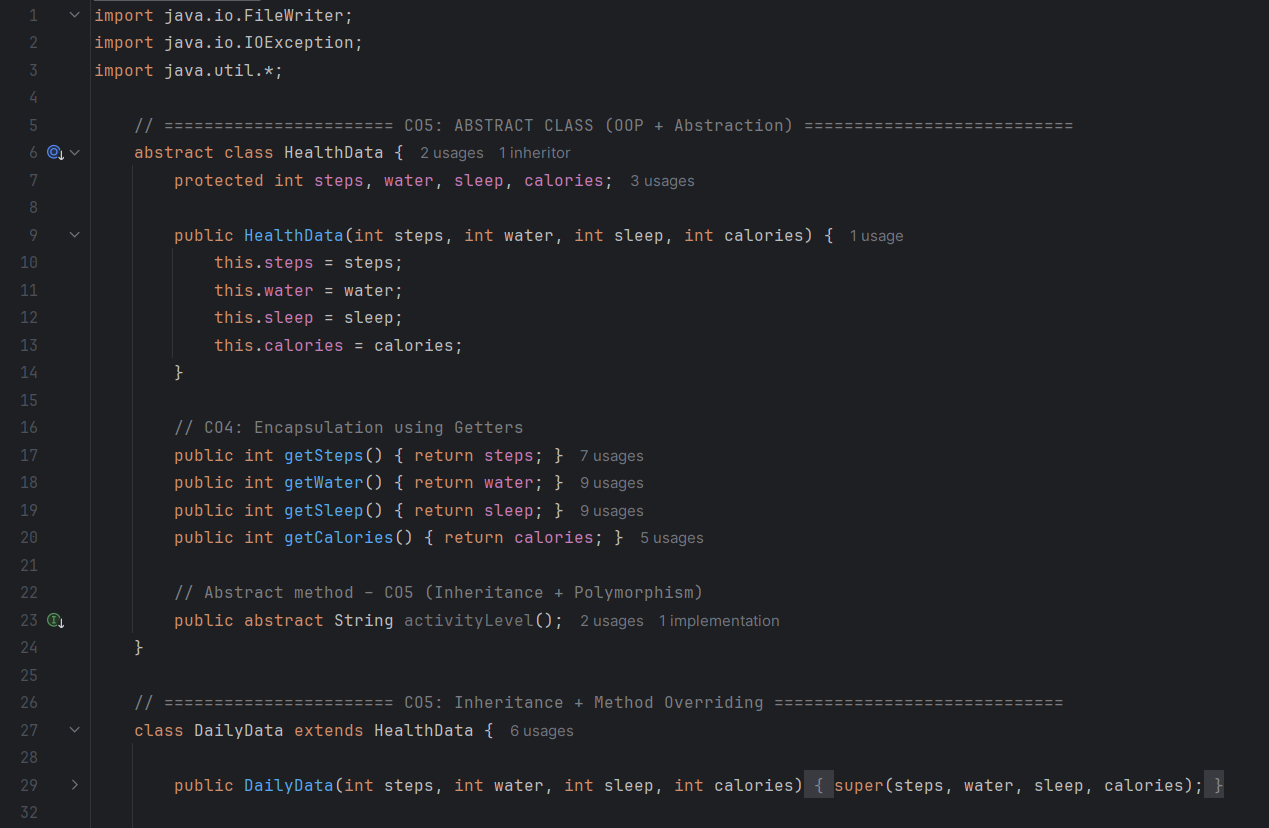
AI-generated content may be incorrect.**

**A computer screen shot of a program

AI-generated content may be incorrect.**

**CHAPTER -4 SCREEN SHOTS**

## 4.1 Screen Shots



A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of code

AI-generated content may be incorrect.

**OUTPUT:**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# CHAPTER -5 TESTING

## Test Cases and Results

## A screenshot of a computer program AI-generated content may be incorrect.

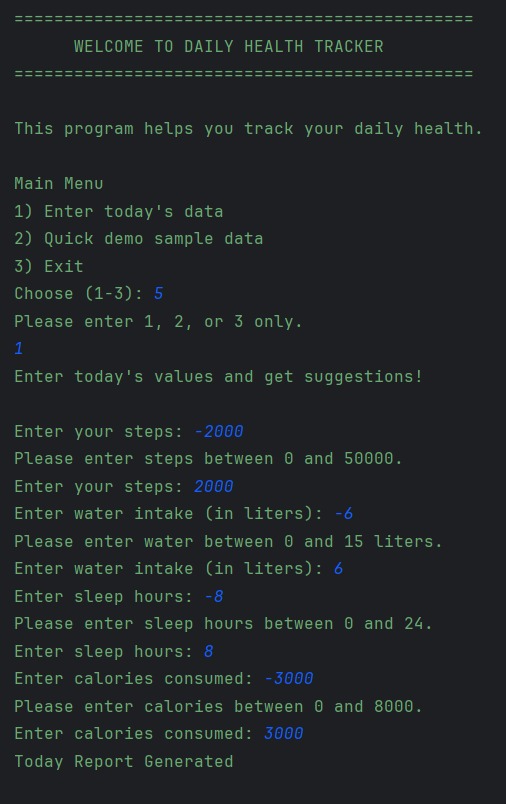
## A screenshot of a computer AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

****

**A screenshot of a phone

AI-generated content may be incorrect.**

**A screenshot of a phone

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

# CHAPTER -6 FUTURE ENHANCEMENTS

## Planned Features

In the future, my project can grow into a much smarter and more convenient health companion. Some ideas I plan to add include AI food recognition, where the user can simply click a photo of their meal and the system will estimate calories automatically. I also want to include mood tracking, voice input, and a small daily health-tips section to support users in a simple way. Another idea is to add weekly challenges and reward badges to keep users motivated along with health alerts when something looks unusual like very low sleep or high calories for many days.

As the system improves, it can be upgraded to work with smartwatches and fitness bands, so the user doesn’t have to enter everything manually. I also plan to add PDF report downloads, customizable themes like dark mode and water-reminder timers to help users stay hydrated. Features like sleep-quality scoring, trend graphs, and long-term insights can make the tracker even more helpful. Overall, all these ideas aim to transform this simple console project into a smart, interactive and supportive health-tracking system

**6.2Possible integrations or optimizations**

## In the future, my project can become even more powerful by connecting it with smarter

## systems and improving the way it works internally. One useful integration would be linking

## the tracker with wearable devices or step counters so that the data is collected automatically

## without manual input. It can also be connected with cloud storage, allowing users to save and

## access their health history from any device.

## On the optimization side, the program can be enhanced to run faster, use memory more

## efficiently and analyze user habits more intelligently. Features like weekly trend detection,

## smarter suggestion algorithms, and machine-learning-based pattern prediction can make the

## system more accurate and personalized. By combining these integrations and optimizations,

## the project can grow from a simple console app into a smart, connected, and highly efficient

## personal health assistant.

## CHAPTER -7 CONCLUSION

* 1. **Summary of the Project**

My project began with a simple question: How can we help people understand their daily health habits in an easy and accessible way? Through this console-based Java application, I designed a system that records essential metrics like steps, water intake, sleep hours and calories, and instantly provides personalized suggestions to help users improve their lifestyle. The architecture, class design, and features work together to make health tracking simple, organized, and user-friendly, even for someone who has never used a fitness app before.

This project may look small on the surface, but it brings together important real-world concepts: structured data handling, smart decision-making, user guidance, and clean program design through classes, interfaces and abstraction. Most importantly, it shows that even a small, simple tool can remind people to take better care of themselves and build healthier habits.As it continues to evolve with future enhancements like AI-based recognition, automation, reports, and smart integrations, this system has the potential to grow into a complete personal health assistant.

In summary, this project is not just a program, it’s a meaningful step toward using simple technology to improve everyday life. And that is the purpose that drives the entire work.

**7.2 What was achieved**

Through this project, I successfully developed a fully functional health-tracking system that meets all the initial objectives. The program captures daily health inputs, processes them intelligently, and presents useful recommendations, all within a simple console interface. It demonstrates how basic tools, when used effectively, can create a meaningful and helpful user experience. The application also stores the data for later reference, which adds real value beyond single-use interaction.

Along with making the system work smoothly, this project also helped me achieve important technical goals. I was able to integrate core Java concepts such as abstraction, inheritance, interfaces, loops, conditions, exception handling and file operations in a real, working environment. Because the code is structured in a clear and flexible way, the project can grow easily with more advanced features later on. Overall, the project delivered both practical usefulness and strong technical depth.

**7.3 Skills learned during development**

Throughout the development process, I learned how to apply core Java concepts in a real-world scenario instead of just studying them in theory. Working with classes, interfaces, file handling and exception management helped me understand how different pieces of code communicate and work together. I also learned how to plan the structure of a program before writing code, which improved the overall clarity and flow of the project.

Beyond technical skills, this project helped me grow as a developer. I improved my logical thinking, learned how to solve unexpected bugs, and understood the importance of writing clean, readable code. I also strengthened my skills in writing organized documentation, building structured system designs and presenting complex ideas in a straightforward way. These skills will be valuable not just for this project, but for all future software development work I take up.

# CHAPTER -8 REFERENCES

## Books

Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner’s Approach* (8th ed.). McGraw- Hill Education.

Freeman, E., & Freeman, E. (2020). *Head First HTML and CSS* (2nd ed.). O’Reilly Media. Duckett, J. (2014). *HTML & CSS: Design and Build Websites*.

Duckett, J. (2015). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley. Sommerville, I. (2016). *Software Engineering* (10th ed.).

## Tutorials and Learning Resources

W3Schools. (2025). *HTML, CSS, and JavaScript Tutorials.* Retrieved from https://[www.w3schools.com](http://www.w3schools.com/)

MDN Web Docs. (2025). *Web Technologies Documentation.* Retrieved from https://developer.mozilla.org

FreeCodeCamp. (2025). *Responsive Web Design Certification.* Retrieved from https://[www.freecodecamp.org](http://www.freecodecamp.org/)

GeeksforGeeks. (2025). *Front-end Development and Web Technologies.* Retrieved from https://[www.geeksforgeeks.org](http://www.geeksforgeeks.org/)

Tutorialspoint. (2025). *HTML, CSS, JavaScript, and Web Development Tutorials.* Retrieved from https://[www.tutorialspoint.com](http://www.tutorialspoint.com/)

## APIs and Tools Used

Google Maps Platform. (2025). *Maps JavaScript API Documentation.* Retrieved from https://developers.google.com/maps

Unsplash API. (2025). *Free High-Resolution Image Access.* Retrieved from https://unsplash.com/developers

OpenWeatherMap. (2025). *Weather API for Travel Assistance.* Retrieved from https://openweathermap.org/api

RapidAPI Hub. (2025). *Travel and Tourism APIs.* Retrieved from https://rapidapi.com/hub

Firebase by Google. (2025). *Authentication and Database Services.* Retrieved from https://firebase.google.com

## Documentation and Research References

World Tourism Organization (UNWTO). (2024). *Tourism Data Dashboard.* Retrieved from https://[www.unwto.org](http://www.unwto.org/)

Booking.com. (2025). *Accommodation and Destination Insights.* Retrieved from https://[www.booking.com](http://www.booking.com/)

TripAdvisor. (2025). *Tourism and Destination Review Data.* Retrieved from https://[www.tripadvisor.com](http://www.tripadvisor.com/)

Travel + Leisure. (2025). *Top Global Destinations and Travel Trends.* Retrieved from https://[www.travelandleisure.com](http://www.travelandleisure.com/)

Tourism Review. (2025). *Digital Transformation in the Tourism Industry.* Retrieved from https://www.tourism- review.com

# CHAPTER -9 APPENDICES

**Check if Java is installed:**

* Open Command Prompt/Terminal and type java -version.

**Download Java JDK:**

* Go to Oracle Java Downloads and download the JDK for your OS.

**Install Java:**

* Run the installer and follow instructions (Windows/Mac).
* On Linux, use sudo apt install openjdk-20-jdk.

**Set environment variables (Windows only):**

* Create JAVA\_HOME pointing to JDK folder.
* Add %JAVA\_HOME%\bin to Path.

**Verify installation:**

* Type java -version and javac -version in Command Prompt/Terminal.

**Geo Tag photos with guide**