HEALTH HAVEN

A Minor Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Artificial Intelligence and Data Science

By

Aayushi Kar (1601-21-771-001)

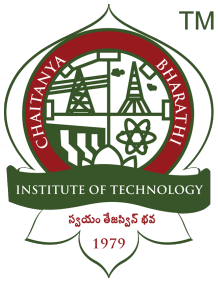
G.Sai Akshitha (1601-21-771-008)

Rimsha Fatima (1601-21-771-017)

Under the esteemed guidance of

Ms. V.Krishna Aravinda

Assistant Professor



DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY

HYDERABAD – 500075

DECEMBER 2023

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
HYDERABAD – 500075

INSTITUTE VISION

"To be the center of excellence in technical education and research".

INSTITUTE MISSION

"To address the emerging needs through quality technical education and advanced research".

DEPARTMENT VISION

"To be a globally recognized center of excellence in the field of Artificial Intelligence and Data Science that produces innovative pioneers and research experts capable of addressing complex real-world challenges and contributing to the socio-economic development of the nation."

DEPARTMENT MISSION

1. To provide cutting-edge education in the field of Artificial Intelligence and Data Science that is rooted in ethical and moral values.
2. To establish strong partnerships with industries and research organizations in the field of Artificial Intelligence and Data Science, and to excel in the emerging areas of research by creating innovative solutions.
3. To cultivate a strong sense of social responsibility among students, fostering their inclination to utilize their knowledge and skills for the betterment of society.
4. To motivate and mentor students to become trailblazers in Artificial Intelligence and Data Science, and develop an entrepreneurial mindset that nurtures innovation and creativity.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
HYDERABAD – 500075

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates of AI & DS will be able to:

1. Adapt emerging technologies of Artificial Intelligence & Data Science and develop state of the art solutions in the fields of Manufacturing, Agriculture, Health-care, Education, and Cyber Security.
2. Exhibit professional leadership qualities to excel in interdisciplinary domains.
3. Possess human values, professional ethics, application-oriented skills, and engage in lifelong learning.
4. Contribute to the research community to meet the needs of public and private sectors.

PROGRAM SPECIFIC OUTCOMES (PSOs)

After successful completion of the program, students will be able to:

1. Exhibit proficiency of Artificial Intelligence and Data Science in providing sustainable solutions by adapting to societal, environmental and ethical concerns to real world problems.
2. Develop professional skills in the thrust areas like ANN and Deep learning, Robotics, Internet of Things and Big Data Analytics.
3. Pursue higher studies in Artificial Intelligence and Data Science in reputed Universities and to work in research establishments.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
HYDERABAD – 500075

PROGRAM OUTCOMES

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems

2. **Problem analysis:** Identify, formulate, review, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
HYDERABAD – 500075

MINOR PROJECT-I

COURSE OBJECTIVES

1. To enable students learning by doing.
2. To develop capability to analyse and solve real world problems.
3. To inculcate innovative ideas of the students.
4. To impart team building and management skills among students.
5. To instill writing and presentation skills for completing the project.

COURSE OUTCOMES

Upon successful completion of this course, students will be able to:

1. Interpret Literature with the purpose of formulating a project proposal.
2. Plan, Analyse, Design and Implement a project using SDLC model.
3. Find the solution of identified problem with the help of modern Technology and give priority to real time scenarios.
4. Plan to work as a team and to focus on getting a working project done and submit a report within astipulated period of time.
5. Prepare and submit the Report and deliver presentation before the Departmental Committee.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY

HYDERABAD – 500075

CO-PO MAPPING

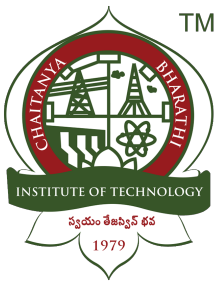
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	3	3	3	3	2	1	2	3	3
CO2	3	3	3	3	3	3	3	2	1	2	3	3
CO3	3	3	3	3	3	3	3	2	-	2	3	3
CO4	2	2	2	3	3	3	3	2	3	3	2	3
CO5	1	2	1	2	3	3	-	-	2	3	-	-

Mapping of Course Outcomes with Program Outcomes

CO-PSO MAPPING

	PSO1	PSO2	PSO3
CO1	2	3	3
CO2	3	3	3
CO3	3	3	3
CO4	2	3	3
CO5	-	3	-

Mapping of Course Outcomes with Program Specific Outcomes



DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
HYDERABAD – 500075

DECLARATION CERTIFICATE

We hereby declare that the project titled **HEALTH HAVEN** submitted by us to the **Artificial Intelligence and Data Science CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY, HYDERABAD** in partial fulfillment of the requirements for the award of **Bachelor of Engineering** is a bona-fide record of the work carried out by us under the supervision of **Ms.V.Krishna Aravinda** . We further declare that the work reported in this project, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Project Associates

Aayushi Kar (1601-21-771-001)

G.Sai Akshitha (1601-21-771-008)

Rimsha Fatima (1601-21-771-017)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
HYDERABAD – 500075

BONAFIDE CERTIFICATE

This is to certify that the project titled **HEALTH HAVEN** is a bonafide record of the work done by

Aayushi Kar (1601-21-771-001)

G.Sai Akshitha (1601-21-771-008)

Rimsha Fatima (1601-21-771-017)

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Engineering** in **Artificial Intelligence and Data Science** to the **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY, HYDERABAD** carried out under my guidance and supervision during the year 2022-23. The results presented in this project report have not been submitted to any other university or Institute for the award of any degree.

Ms. V.Krishna Aravinda — Guide

Dr.K.Ramana — Head of the Department

Submitted for Semester Minor-Project viva-voce examination held on _____

Examiner-1 **Examiner-2**

ABSTRACT

”Health Haven is a comprehensive web platform designed to empower individuals in their journey towards improved health and fitness. Leveraging the MERN (MongoDB, Express.js, React, Node.js) stack, this digital health ecosystem offers users a personalized dashboard where they can effortlessly track and monitor essential health metrics such as calories intake, sleep, steps, water intake, weight, and workout routines. The platform goes beyond mere data recording by providing tailored exercise recommendations, nutritional guidance, and the ability to set and monitor individual health and fitness targets. With a user-centric approach, Health Haven fosters a sense of community through social features, allowing users to connect, share achievements, and participate in challenges. The integration of e-commerce functionality enhances the holistic approach by enabling users to conveniently purchase fitness products and select healthy food options. This innovative platform aims to address the multifaceted needs of individuals seeking a seamless, secure, and engaging solution for enhancing their overall well-being.”

Keywords : MERN

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to the following people for guiding us through this course and without whom this project and the results achieved from it would not have reached completion.

Ms. V.Krishna Aravinda, Assistant Professor, Department of Artificial Intelligence and Data Science, for helping us and guiding us in the course of this project. Without his/her guidance, we would not have been able to successfully complete this project. His/Her patience and genial attitude is and always will be a source of inspiration to us.

Dr.K.Ramana, the Head of the Department, Department of Artificial Intelligence and Data Science, for allowing us to avail the facilities at the department.

We are also thankful to the faculty and staff members of the Department of Artificial Intelligence and Data Science, our individual parents and our friends for their constant support and help.

TABLE OF CONTENTS

Title	Page No.
ABSTRACT	i
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	vi
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Applications	1
1.3 Problem Statement	2
1.4 Organization of Project	2
CHAPTER 2 LITERATURE SURVEY	4
2.1 Basics of HTML	4
2.2 Basics of CSS	4
2.3 Basics of JavaScript	4
2.4 Basics of Node	5
2.5 Basics of Express	5
2.6 Basics of MongoDB	5
2.7 Basics of React	6
CHAPTER 3 SYSTEM REQUIREMENTS AND SPECIFICATIONS	7
3.1 Functional Requirements	7
3.2 Non-Functional Requirements	8
3.3 Software Requirements	10
3.4 Hardware Requirements	10

CHAPTER 4 METHODOLOGY 11
4.1 Requirement Analysis: 11
4.2 System Design and Architecture: 11
4.3 Technology Selection and Planning: 11
4.4 Development and Implementation: 12
4.5 Testing and Quality Assurance: 12
4.6 Deployment and Configuration: 12
4.7 Documentation and Training: 12
4.8 Launch and Post-launch Support: 12
4.9 Monitoring and Optimization: 13
CHAPTER 5 IMPLEMENTATION 14
CHAPTER 6 TESTING AND RESULTS 15
6.1 Login page 15
6.2 Sign up page 15
6.3 Home Page 16
6.4 Targets and Tracking 16
6.5 Database Login and Sign Up Credentials 17
6.6 Database Targets (User Input Values) 17
6.7 Exercises page 18
6.8 Showing Results on Searching by calling of RAPID API 18
6.9 Showing Similar Exercises 19
6.10 Showing the Links to Videos of Top Fitness Instructors 19
6.11 Adding of Items to the Shopping Cart 20
6.12 Checkout Page and Total Price Page 20
CHAPTER 7 FUTURE SCOPE 21
CHAPTER 8 CONCLUSION 22
APPENDIX A CODE ATTACHMENTS 23
A.1 Login Page 23
A.1.1 index.js 23

A.2 SignUp Page.jsx 25
A.2.1 index.js of server side for connecting to database 25
A.2.2 Validation of schemas used for login and signup 25
1.3 App.js 26
1.3.1 Cart.js 26
1.3.2 Header.jsx 27
1.4 Exercises.jsx 29
1.5 ShoppingCart.jsx 30
1.5.1 Iteams.jsx 31
1.6 Targets.jsx 33
1.6.1 Targets storing on backend 33

LIST OF FIGURES

6.1 Login page screenshot 15

6.2 Signup page screenshot 15

6.3 Home page screenshot 16

6.4 Inputs taken screenshot 16

6.5 Database screenshot 17

6.6 Database screenshot 17

6.7 Explore exercises page 18

6.8 Showing results of the search 18

6.9 Similar exercises page 19

6.10 YouTube links of specific exercise 19

6.11 Shopping Cart 20

6.12 Checkout Page 20

CHAPTER 1

INTRODUCTION

1.1 Overview

The project ”Health Haven” is designed for the development of a user-friendly and highly personalized website that caters to individuals seeking to improve their overall health and fitness. The current challenge is to create a platform where users can seamlessly navigate to their personalized dashboard after entering their login credentials. Within this personal dashboard, users should be able to input and track essential health metrics, including ’Calories intake,’ ’Sleep,’ ’Steps,’ ’Water intake,’ ’Weight,’ and ’Workout.’ Additionally, users must be able to set and monitor their individual health and fitness targets. The primary objective is to develop a website that not only records these metrics but also provides tailored exercise recommendations to help users achieve their health and fitness goals. Furthermore, the platform should feature an e-commerce cart where users can conveniently purchase fitness products and select healthy food options, thereby promoting a holistic approach to wellbeing. The challenge lies in creating a seamless and effective system that addresses these multifaceted health and fitness needs, while ensuring the website’s usability and personalized user experience.

1.2 Applications

- **Real-time Updates:** Node.js, part of the MERN stack, is known for its event-driven architecture, making it well-suited for real-time applications. This can be beneficial for providing users with instant updates on their health metrics, progress, and any personalized recommendations.
- **Responsive User Interface:**React, a JavaScript library for building user interfaces, allows for the creation of interactive and responsive UI components. This is essential for providing a seamless and user-friendly experience on the Health Haven website.
- **Goal Setting and Monitoring:** Users can set personalized health and fitness goals, and the platform will track their progress. Regular updates and insights can be provided to motivate users and help them stay on track.

- **Scalability:** MongoDB, a NoSQL database used in the MERN stack, is designed for scalability and flexibility. As your user base grows and the amount of data increases, MongoDB can easily handle the scalability requirements of your health and fitness platform.
- **E-commerce Integration:** Using the MERN stack, you can easily integrate an e-commerce system into your platform, allowing users to purchase fitness products and healthy food options directly through the website.
- **Data Security:** aImplementing secure authentication and authorization mechanisms with the help of Node.js and Express.js can help ensure the privacy and security of user data.

1.3 Problem Statement

In today's fast-paced world, where individuals are increasingly prioritizing their health and well-being, there exists a need for a comprehensive and user-friendly digital platform. This platform, named 'Health Haven,' aims to address the challenge of seamlessly integrating health and fitness tracking, personalized goal setting, and holistic well-being support. The problem at hand is to design and develop a MERN stack-based web application that not only allows users to effortlessly track and monitor their health metrics but also provides tailored exercise recommendations, nutritional guidance, and a seamless e-commerce experience for fitness products and healthy food options. The primary objective is to create a centralized, intuitive, and secure space where users can actively engage in improving their overall health, backed by data-driven insights, social interaction, and expert guidance. The challenge lies in creating a cohesive system that balances usability, personalization, and the diverse needs of individuals seeking a holistic approach to their health and fitness journey.

1.4 Organization of Project

While structuring our project, we listed out the main functionalities of it. They follows:

- **Frontend Development (React):**

  User Interface (UI): Designing a user-friendly and captivating UI to enable seamless interaction between viewers and video content.

- **Backend Development (Node.js, Express):**

Logic on the server side: constructing the backend to process requests, control data flow, and communicate with the front end.

- **Data Storage (MongoDB):**

  Database design is the process of organizing databases to hold links to online retailers, user data, blog posts, and video metadata.

  Data management: CRUD operations should be implemented for effective data handling.

- **E-commerce Integration:**

  Product Linking:Product linking allows viewers to direct purchasing links within video content.

- **User Authentication and Authorization:**

  User management: Putting authentication systems in place for account management, login, and registration.

  Role-based Access Control: Managing various user roles, such as viewers and content producers, is known as role-based access control.

  Technologies like HTML, CSS, JAVASCRIPT, NODE, EXPRESS, REACT, MONGODB, will be able to effectively fulfil functionalities/requirements

CHAPTER 2

LITERATURE SURVEY

2.1 Basics of HTML

HTML, which stands for HyperText Markup Language, is the standard language used to create and design web pages. It provides the structure and layout of a webpage by using various tags and elements to define different components such as text, images, links, forms, and more. HTML works together with CSS (Cascading Style Sheets) and JavaScript to create visually appealing, interactive, and functional websites.

2.2 Basics of CSS

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation of a document written in HTML or XML. It is used to separate the presentation of a document from its structure and content. With CSS, you can control the layout, colours, fonts, and other visual elements of a web page. It allows you to create consistent and visually appealing designs across multiple web pages and devices. CSS can be applied to a single HTML element, multiple elements, or even an entire web page. It is a powerful tool for web developers and designers, as it allows them to create visually stunning and responsive websites with minimal effort.

2.3 Basics of JavaScript

JavaScript is a versatile and widely used programming language that plays a pivotal role in web development, enabling dynamic and interactive user experiences across a multitude of platforms. Initially created to enhance static HTML pages, JavaScript has evolved into a full-fledged, multi-paradigm language, featuring event-driven, functional, and object-oriented programming capabilities. As a client-side scripting language, it runs in web browsers, allowing developers to manipulate the Document Object Model (DOM), handle user interactions, and dynamically update content. Its ubiquity on the client

side, coupled with the advent of server-side JavaScript through platforms like Node.js, has solidified JavaScript as an integral part of full-stack development. With a continually growing ecosystem of libraries and frameworks, such as React, Angular, and Vue.js, JavaScript empowers developers to build scalable and responsive web applications, making it an indispensable tool in the modern software development landscape.

2.4 Basics of Node

Node.js, commonly referred to as Node, is an open-source, server-side JavaScript runtime environment. Built on the V8 JavaScript engine, it enables developers to execute JavaScript code outside a web browser. Node.js facilitates scalable and high-performance applications, handling concurrent connections and I/O operations efficiently through its event-driven, non-blocking architecture. It offers an extensive package ecosystem via npm (Node Package Manager), fostering rapid development by providing access to numerous libraries and modules. Node.js is favored for building web servers, APIs, real-time applications, and microservices due to its versatility, speed, and the ability to leverage JavaScript across both client-side and server-side environments, fostering seamless code sharing and development practices.

2.5 Basics of Express

Express.js is a minimalist, flexible Node.js web application framework, designed for building robust and scalable web applications and APIs. It simplifies the creation of web servers in Node.js by providing a robust set of features and middleware. Express.js streamlines routing, handling HTTP requests, defining endpoints, and managing responses. Its modular design allows developers to add functionality through middleware, enabling tasks like parsing incoming requests, handling authentication, and integrating third-party APIs. As a lightweight framework, Express.js offers flexibility, allowing developers to structure applications according to their preferences, making it a popular choice for creating web applications due to its simplicity and extensibility.

2.6 Basics of MongoDB

MongoDB is a popular NoSQL, document-oriented database, known for its flexibility and scalability in handling large volumes of unstructured or semi-

structured data. It stores data in JSON-like documents, allowing for dynamic schemas and easy integration with applications. MongoDB's distributed architecture supports high availability and horizontal scaling, enabling seamless data distribution across clusters. It offers powerful querying through its expressive query language and indexing capabilities, fostering efficient data retrieval. MongoDB is utilized across various industries for its adaptability to diverse data types, making it a preferred choice for applications requiring flexible data modeling, real-time analytics, and agile development due to its schema-less nature and ease of scalability.

2.7 Basics of React

React, developed by Facebook, is a popular JavaScript library for building user interfaces (UIs) in web applications. It follows a component-based architecture, allowing developers to create reusable UI components, simplifying complex UI development. React's virtual DOM efficiently updates and renders only the necessary components when data changes, enhancing performance. It embraces a declarative approach, enabling easier state management and prop handling. React's ecosystem includes tools like Redux for state management, JSX for blending JavaScript with HTML, and a vast community-contributed library of reusable components. Widely adopted for its efficiency, React empowers developers to craft interactive, responsive, and scalable web applications with ease, making it a cornerstone in modern web development.

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATIONS

3.1 Functional Requirements

Functional requirements for the ”Health Haven” project encompass the essential features and functionalities that the platform must offer to meet user needs. Here’s a breakdown of functional requirements:

- **User Authentication and Management:**

  User registration,using email and password login.

  Features for managing accounts (password resets, profile editing, etc.).

- **Targets and Tracking**

  The users can input the various targets like sleep ,calories intake,water intake,steps walked etc.

- **Targeted Exercise Selection**

  Smart Search Bar: Implementing a robust search functionality enabling users to find specific exercises, body parts, or equipment effortlessly.

  Tagged Content: Enhancing discoverability with tags for body parts, equipment, and exercise types.

- **Targeted Exercise Selection**

  Body Part Focus: Customized exercise options allowing users to select specific target areas such as cardio, shoulders, belly, etc.

  Instrument Choice: Empowering users to choose exercise instruments, catering to personal preferences and equipment availability.

- **Interactive Exercise Animations**

  Visual Guidance: Incorporating interactive animations for each exercise, providing users with a visual guide on proper form and technique.

  User Engagement: Enhancing user experience through immersive and interactive exercise demonstrations.

– **E-commerce Integration:**

Direct links to buy the fitness based items featured in the videos.

The ability for users to gather several items for purchase through a shopping cart was inorder to interact with the material.

3.2 Non-Functional Requirements

Non-functional requirements for the ”FoodBlog PlayHub” project outline the qualities, constraints, and characteristics that are essential for the platform's overall performance, security, usability, and other aspects beyond specific functionalities. Here are some non-functional requirements:

– **Performance:**

Response Time: Ensure that the app loads quickly and is responsive; animations and interactive features should initiate promptly..

Scalability: The platform should efficiently handle an increasing number of users, accommodating user growth and content uploads without compromising performance.

Optimized Media Delivery: Implement caching and Content Delivery Network (CDN) integration for seamless and effective delivery of interactive exercise animations.

– **Security:**

Data encryption: To maintain security, encrypt transactional data and sensitive user information.

Authentication and Authorization:Strong mechanisms to stop illegal access to features and content are provided by authentication and authorization.

Payment Security:Payment security refers to processing payments securely by adhering to industry standards (such as PCI-DSS).

– **Reliability:**

High Availability: Reliability and accessibility of the platform are ensured by minimal downtime.

Data Backup and Recovery:Data backup and recovery include routine backups as well as procedures for restoring data in the event of an issue.

– **Usability:**

Accessibility: Compliance with accessibility standards (WCAG) for inclusively across various user needs.

Intuitive Interface: User-friendly design and navigation for seamless interaction and content discovery.

- **Compatibility:**

  Cross-Browser Compatibility: Compatibility with major web browsers (Chrome, Firefox, Safari, etc.).

  Device Compatibility: Responsive design ensuring functionality across different devices and screen sizes.

- **Maintainability:**

  Code Maintainability: A clean, modular, and well-documented codebase makes updates and maintenance simpler.

  Update and Patch Management: Simplified procedure for updating and patching software to improve functionality and security.

- **Scalability:**

  Horizontal Scalability: Ability to scale horizontally across servers to accommodate growing user bases.

  Load Balancing: Implement load balancing to evenly distribute traffic across servers for optimal performance.

3.3 Software Requirements

- Operating System: Windows 7 or above
- Vs code
- Any Browser

3.4 Hardware Requirements

- 64-bit CPU (Intel / AMD architecture)
- 4 GB or above RAM
- Min 500 mb free disk space

CHAPTER 4

METHODOLOGY

The system methodology for the ”Health Haven” project involves a structured approach that encompasses various phases from planning to deployment. Here’s a detailed breakdown:

4.1 Requirement Analysis:

- Gather stakeholder requirements through meetings, questionnaires, and interviews.
- Define functional and non-functional requirements to guide the development process.
- Create personas, use cases, and user stories to capture a range of user demands and scenarios.

4.2 System Design and Architecture:

- Describe the system architecture in terms of the database (MongoDB), frontend (React), backend (Node.js, Express), and cloud integration.
- Create entity-relationship diagrams (ERDs), data flow diagrams, and database schemas to facilitate effective data management.

4.3 Technology Selection and Planning:

- Assess and choose the best tools, frameworks, in accordance with the demands of the project and its scalability.
- Plan the deployment strategy, including local testing, staging, and production environments.
- Establish version control systems and development environments.

4.4 Development and Implementation:

- Implement the frontend using React to achieve an interactive user interface with features like exercise animations, user interaction, and content discovery.
- Develop the backend using Node.js and Express for content management, API integrations, and authentication. .

4.5 Testing and Quality Assurance:

- Conduct comprehensive testing, including unit, integration, and end-to-end testing, to ensure functionality, security, and performance,accessibility, compatibility, and usability testing on various browsers and devices.
- Implement bug tracking and resolution procedures to address and fix identified problems and defects.

4.6 Deployment and Configuration:

- Deploy the application to staging environments for user acceptance testing (UAT) and validation.
- Configure servers, databases, and other infrastructure components for scalability and performance optimization.
- Set up continuous integration and deployment (CI/CD) pipelines for automated testing and deployment.

4.7 Documentation and Training:

- Create comprehensive documentation to ensure functionality, security, and performance.
- Conduct training sessions for users and stakeholders.

4.8 Launch and Post-launch Support:

- Release the platform to production following extensive testing and verification.

– Provide post-launch assistance, resolving any problems, installing updates, and doing continuous upkeep to guarantee system functionality and stability.

4.9 Monitoring and Optimization:

– Use monitoring tools for system health, user analytics, and real-time performance monitoring.

– To find areas that need improvement and optimization, analyze user data and system metrics.

– Continuously enhance the platform's usability, security, and performance based on user input and observations.

CHAPTER 5

IMPLEMENTATION

Health Haven, a dynamic web application, seamlessly integrates essential features to provide users with a holistic approach to health and fitness. The home page serves as an informative gateway, detailing the platform's mission and services. Upon navigating to the register page, users can easily create accounts, leveraging robust authentication and authorization processes to ensure data security. The registration process also collects user inputs, storing them in a MongoDB database, where the platform's backend performs CRUD operations, maintaining a comprehensive user profile.

The exercises page takes user engagement to the next level by incorporating CSS animations rendered through an API. Users receive exercise suggestions tailored to their needs, with the platform intelligently prompting workouts based on individual preferences. To enhance the user experience, each exercise recommendation includes informative YouTube video links, guiding users on proper execution. This not only adds a visual element to the routine but also ensures that users can confidently perform exercises with accurate form.

The shopping cart functionality enhances Health Haven's offering, with a dynamic products page and cart page. Users can seamlessly browse through a curated selection of fitness products, and as items are added or removed from the cart, the platform dynamically calculates the total price. This real-time price adjustment ensures transparency and convenience, allowing users to make informed decisions.

Underpinning these features is the MERN stack, with MongoDB handling the persistent storage of user data and facilitating CRUD operations. The platform's user-friendly interface, secure authentication processes, and dynamic content generation make Health Haven a comprehensive solution for individuals seeking a personalized and engaging journey toward improved health and fitness. Whether it's tracking metrics, receiving exercise recommendations, or managing fitness-related purchases, Health Haven stands as a versatile and user-centric platform in the digital health and wellness landscape.

CHAPTER 6

TESTING AND RESULTS

6.1 Login page

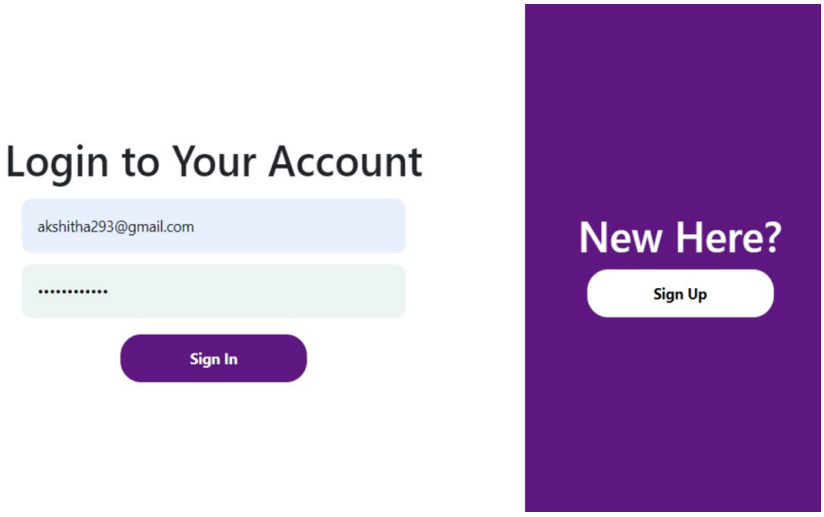


Figure 6.1: Login page screenshot

6.2 Sign up page

Figure 6.2: Signup page screenshot

6.3 Home Page

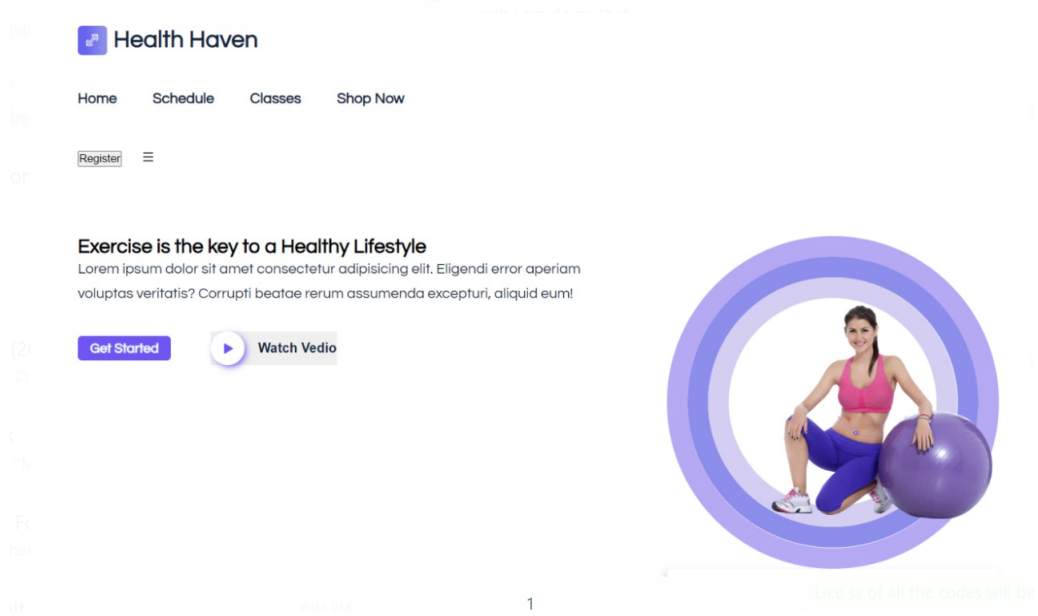


Figure 6.3: Home page screenshot

6.4 Targets and Tracking

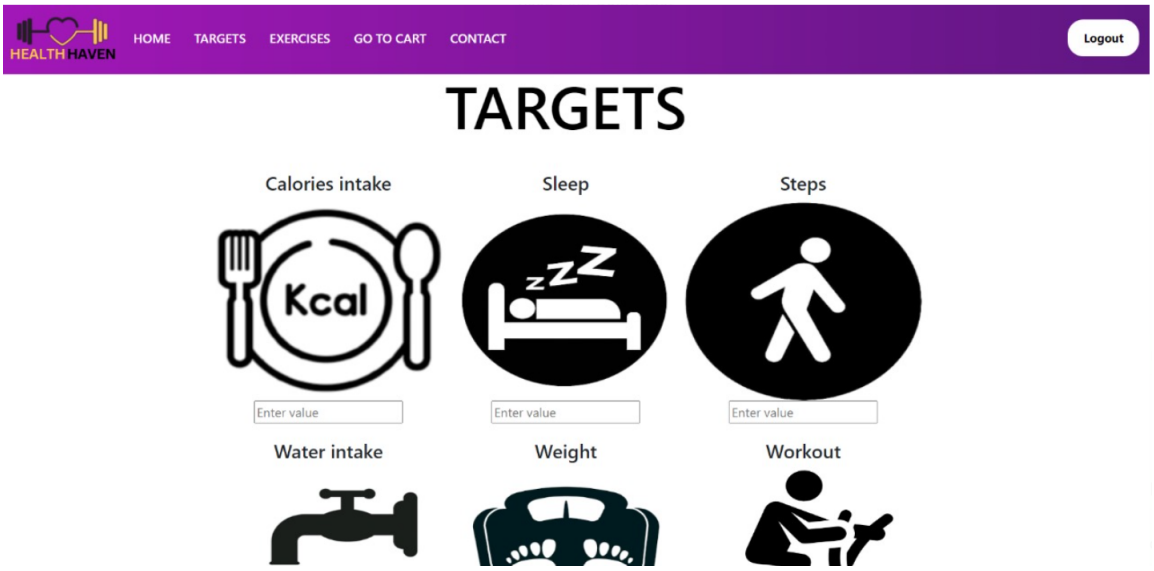


Figure 6.4: Inputs taken screenshot

6.5 Database Login and Sign Up Credentials

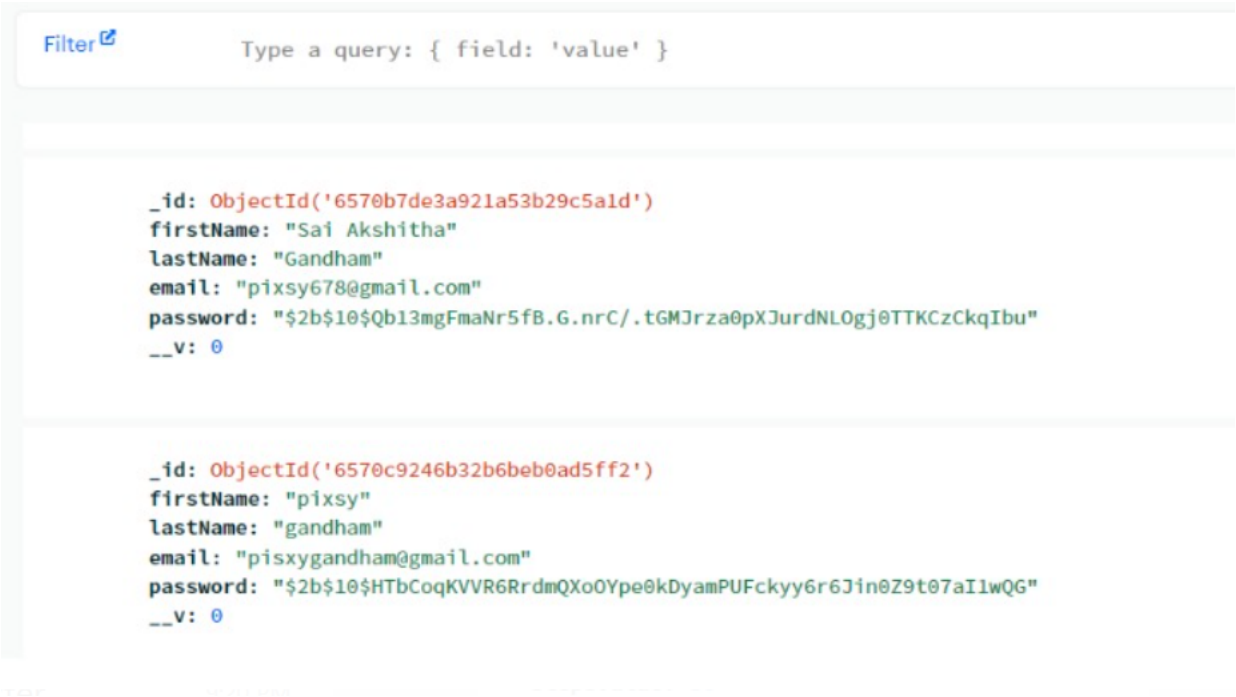


Figure 6.5: Database screenshot

6.6 Database Targets (User Input Values)

Figure 6.6: Database screenshot

6.7 Exercises page

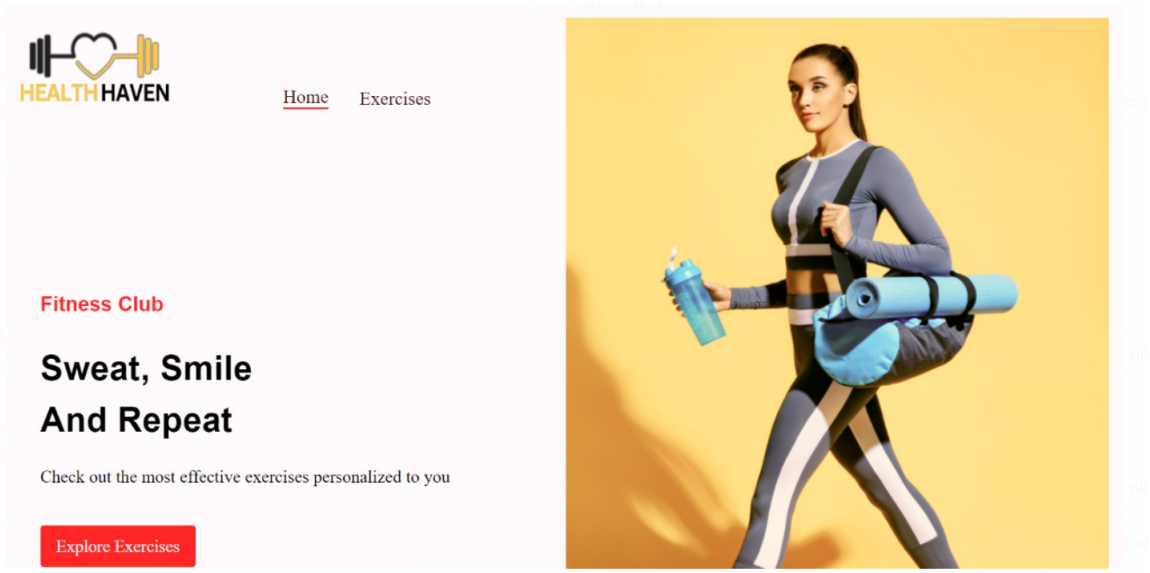


Figure 6.7: Explore exercises page

6.8 Showing Results on Searching by calling of RAPID API

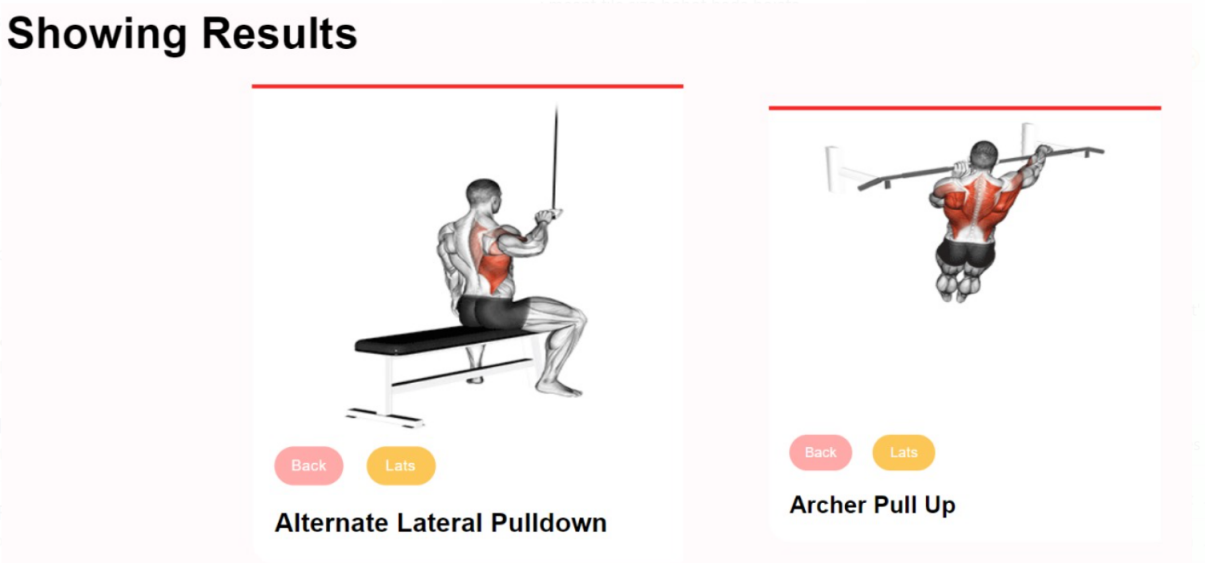


Figure 6.8: Showing results of the search

6.9 Showing Similar Exercises

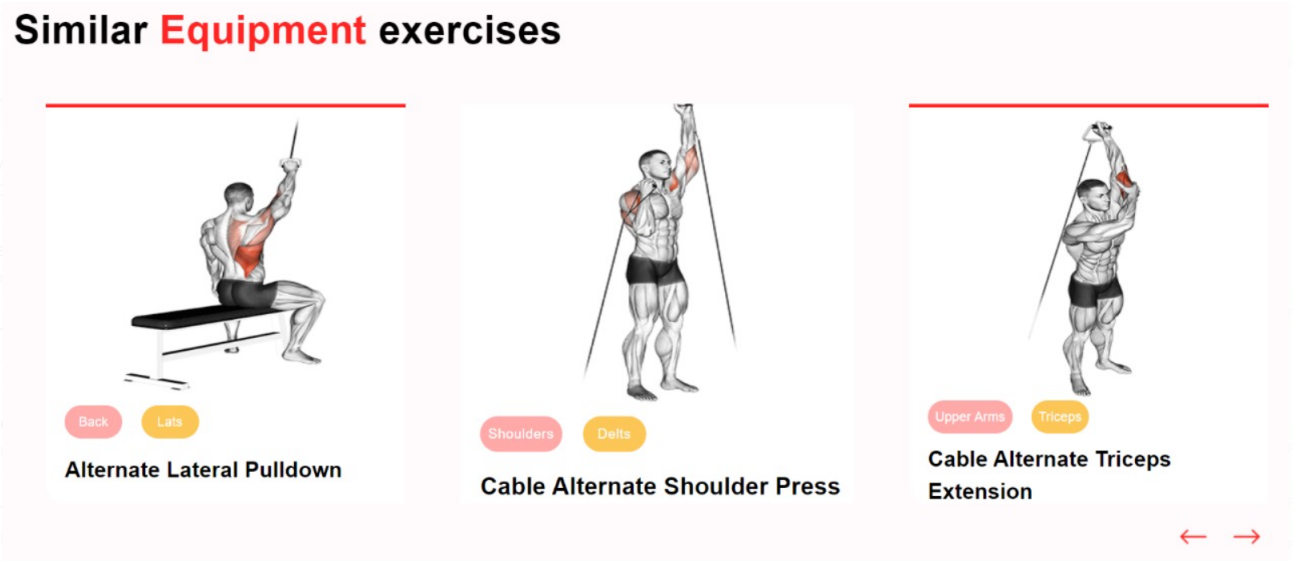


Figure 6.9: Similar exercises page

6.10 Showing the Links to Videos of Top Fitness Instructors

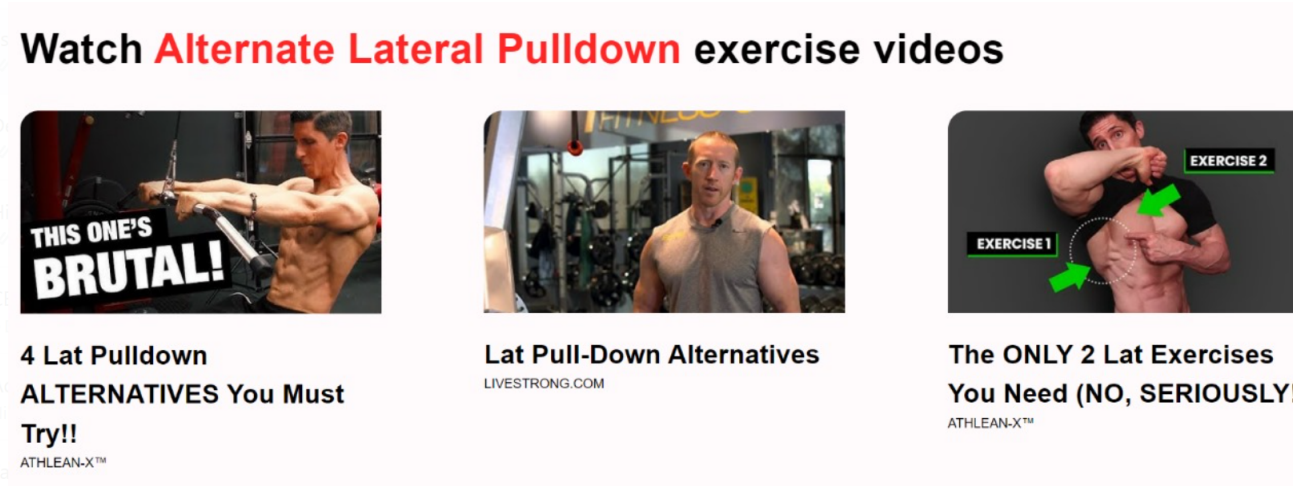


Figure 6.10: YouTube links of specific exercise

6.11 Adding of Items to the Shopping Cart

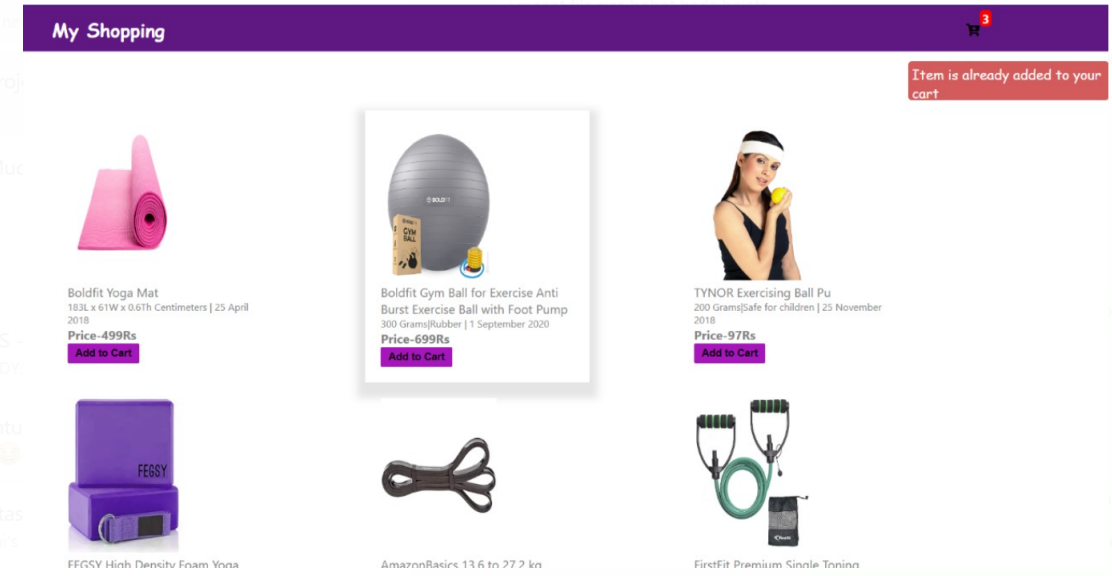


Figure 6.11: Shopping Cart

6.12 Checkout Page and Total Price Page

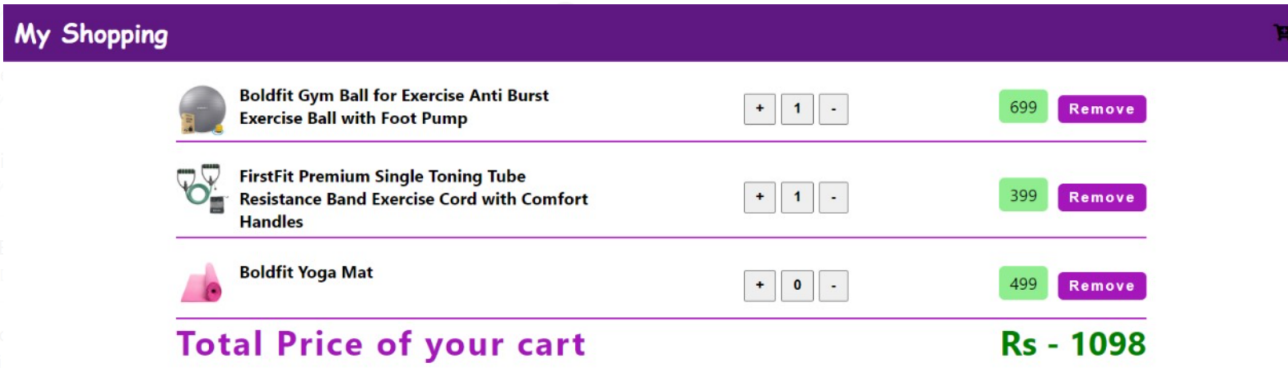


Figure 6.12: Checkout Page

CHAPTER 7

FUTURE SCOPE

The ”Health Haven” project holds significant potential for future expansion and enhancement, paving the way for several promising developments:

- **Advanced Personalization and Analytics:** Leverage AI-driven recommendation systems to offer users personalized workout routines based on goals and preferences. Implement analytics to track and provide data-driven insights into user performance, fostering a tailored fitness experience.
- **Augmented Reality (AR) and Virtual Training:** Enhance user engagement by integrating AR features for visualizing correct exercise techniques and introducing virtual training sessions led by experienced fitness trainers, creating an immersive and interactive workout environment.
- **Global Expansion and Localization:** Ensure inclusivity by adding multilingual support and incorporating region-specific fitness routines. Implement location-based features to recommend outdoor workouts based on local weather conditions, catering to a diverse global user base.
- **Enhanced Social and Community Features:** Cultivate a vibrant community by enabling users to share achievements, progress, and tips. Encourage live discussions with fitness experts, fostering a supportive environment for users to connect and interact.
- **Integration with Emerging Technologies:**Improve user engagement with voice-assisted interactions for hands-free navigation. Explore IoT for seamless integration with smart fitness equipment and consider blockchain for secure and transparent recording of user achievements and transactions.
- **Expanded E-commerce Capabilities:** Elevate the platform’s value proposition by enhancing the shopping cart with curated fitness products, affiliate marketing, loyalty programs, and discounts. This strategy benefits both users and vendors, creating a more comprehensive and appealing e-commerce integration.

CHAPTER 8

CONCLUSION

In conclusion, the MERN Stack Health Haven App represents a revolutionary approach to reshaping the health and fitness landscape, leveraging cutting-edge technologies to create an immersive and personalized experience for users. By seamlessly integrating AR, AI-driven personalization, and social features, the app aims to not only guide users through their wellness journeys but also foster a vibrant and supportive community.

The comprehensive global expansion strategy, including multilingual support and regional health and wellness traditions, ensures inclusivity and accessibility for users worldwide. The incorporation of emerging technologies such as voice-assisted interactions, IoT-enabled smart devices, and blockchain for secure transactions demonstrates a commitment to staying at the forefront of innovation in the health and wellness tech space.

Furthermore, the platform's expanded e-commerce capabilities, featuring curated products, affiliate marketing, and loyalty programs, aim to enhance the overall user experience and provide valuable monetization opportunities for both users and vendors.

As the MERN Stack Health Haven App evolves, the focus remains on continuous improvement, user feedback integration, and responsiveness to technological advancements. The development methodology, mirroring agile practices and iterative improvements, ensures the app's high performance, security, and usability.

In essence, the MERN Stack Health Haven App strives to redefine the health and wellness app landscape, offering not just health and fitness guidance but a holistic and engaging wellness experience that adapts to users' evolving needs and preferences. The journey ahead involves staying committed to innovation, community building, and user satisfaction to solidify its position as a leading player in the dynamic realm of health and wellness technology.

APPENDIX A

CODE ATTACHMENTS

A.1 Login Page

A.1.1 index.js

```
1  import { useState } from 'react';
2  import { Link, useNavigate } from 'react-router-dom';
3  import axios from 'axios';
4  import styles from './styles.module.css';
5  const Signup = () => {
6    const [data, setData] = useState({
7      firstName: "",
8      lastName: "",
9      email: "",
10     password: ""
11   });
12   const [error, setError] = useState("");
13   const navigate = useNavigate();
14
15   const handleChange = ({ currentTarget: input }) => {
16     setData({ ...data, [input.name]: input.value });
17   };
18   const handleSubmit = async (e) => {
19     e.preventDefault();
20     try {
21       const url = "http://localhost:8080/api/users";
22       const { data: res } = await axios.post(url, data);
23       navigate("/login");
24       console.log(res.message);
25     } catch (error) {
26       if (error.response && error.response.status >= 400 &&
             error.response.status <= 500) {
27         setError(error.response.data.message);
28       }
29     }
30   };
31   return (
32     <div className={styles.signup_container}>
33       <div className={styles.signup_form_container}>
34         <div className={styles.left}>
35           <div className={styles.centerContent}>
36             <h1>Already have an account?</h1>
37             <Link to="/login">
38               <button type='button' className={styles.white_btn
                   }>
39                 Sign in
```

```
                </button>
              </Link>
            </div>
          </div>
          <div className={styles.right}>
            <form className={styles.form_container} onSubmit={
                handleSubmit}>
              <h1>Create Account</h1>
              <input
                type='text'
                placeholder='First Name'
                name='firstName'
                onChange={handleChange}
                value={data.firstName}
                required
                className={styles.input}
              />
              <input
                type='text'
                placeholder='Last Name'
                name='lastName'
                onChange={handleChange}
                value={data.lastName}
                required
                className={styles.input}
              />
              <input
                type='email'
                placeholder='Email'
                name='email'
                onChange={handleChange}
                value={data.email}
                required
                className={styles.input}
              />
              <input
                type='password'
                placeholder='Password'
                name='password'
                onChange={handleChange}
                value={data.password}
                required
                className={styles.input}
              />
              {error && <div className={styles.error_msg}>{error
                  }</div>}
              <button type='submit' className={styles.green_btn}>
                Sign Up
              </button>
            </form>
          </div>
        </div>
      </div>
    );
};
export default Signup;
```

A.2 SignUp Page.jsx

A.2.1 index.js of server side for connecting to database

```
require('dotenv').config();
const express = require('express');
const cors = require('cors');
const app = express();
const connection = require('./db');
const userRoutes = require('./routes/users');
const authRoutes = require('./routes/auth');
const DataModel = require('./models/data');
// Database connection
connection();

// Middlewares
app.use(express.json());
app.use(cors({
    origin: '',  // Allow any origin, you can replace '' with
        your specific frontend URL
    credentials: true,
  }))
// Routes
app.use("/api/users", userRoutes);
app.use("/api/auth", authRoutes);

// Save data to MongoDB
app.post('/save', async (req, res) => {
    try {
      // Access the data sent from the front end
      const dataFromFrontend = req.body;
      // Log the data to the console
      console.log('Data from front end:', dataFromFrontend);
      // Save the data to MongoDB
      const newData = new DataModel(dataFromFrontend);
      await newData.save();
      // Send a response if needed
      res.status(200).json({ message: 'Data received and saved
          successfully' });
    } catch (error) {
      console.error('Error processing data:', error);
      res.status(500).json({ message: 'Internal Server Error' })
          ;
    }
  });
  const port = process.env.PORT || 8080;
  app.listen(port, () => console.log(Listening on port ${port
      }...));
```

A.2.2 Validation of schemas used for login and signup

```
const mongoose = require('mongoose');
const jwt=require('jsonwebtoken');
```

```
const Joi=require('joi');
const passwordComplexity=require("joi-password-complexity");


const userSchema=new mongoose.Schema({
    firstName:{ type:String, required:true},
    lastName:{ type:String, required:true},
    email:{ type:String, required:true},
    password:{ type:String, required:true},
});
userSchema.methods.generateAuthToken=function(){
    const token=jwt.sign({_id:this._id},process.env.
       JWTPRIVATEKEY,{expiresIn:'30d'});
    return token;
};


const User=mongoose.model("User",userSchema);
// const Target=mongoose.model("Target", targetSchema);
const validate=(data)=>{
    const complexityOptions = {
        min: 8,                    // minimum password length
        max: 30,                   // maximum password length
        lowerCase: 1,              // require at least 1 lowercase
            letter
        upperCase: 1,              // require at least 1 uppercase
            letter
        numeric: 1,                // require at least 1 digit
        symbol: 1,                 // require at least 1 special
           character
        requirementCount: 4,       // total number of requirements
            to satisfy
    };
    const schema=Joi.object({
        firstName: Joi.string().required().label("First Name"),
        lastName: Joi.string().required().label("Last Name"),
        email: Joi.string().email().required().label("Email"),
        password: passwordComplexity().required().label("
           Password"),
    });
    return schema.validate(data);
};
module.exports={User, validate};
```

1.3 App.js

1.3.1 Cart.js

```
import React, { useState } from 'react';
import Cart from './components/Cart';
import './styles/amazon.css';
const App = () => {
  const [show, setShow] = useState(true);
```

```
  const [cart, setCart] = useState([]);
  const handleClick = (item) => {
    let isPresent = false;
    cart.forEach((product) => {
      if (item.id === product.id) {
        isPresent = true;
      }
    });
    if (isPresent) {
      setTimeout(() => {
        setWarning(true);
        setTimeout(() => {
          setWarning(false);
        }, 2000);
      }, 2000);

      return;
    }
    setCart([...cart, item]);
  };
  const handleChange = (item, amountDelta) => {
    let index = -1;

    cart.forEach((data, idx) => {
      if (data.id === item.id) {
        index = idx;
      }
    });
    const tempArr = [...cart];

    tempArr[index].amount += amountDelta;

    if (tempArr[index].amount <= 0) {
      tempArr.splice(index, 1);
    }
    setCart(tempArr);
  };
  return (
    <div>
      {/* Your JSX content goes here */}
      <Cart
        cartItems={cart}
        onItemChange={handleChange}
        // Pass any other required props
      />
    </div>
  );
};
export default App;
```

1.3.2 Header.jsx

```
import React from 'react'
import "../../styles/header.css";
```

```
import logo from "../../assets/img/dumble.png"
import { NavItem } from 'react-bootstrap';
const  nav__links=[
    {
        path:'#',
        display:'Home'
    },
    {
        path:'#',
        display:'Exercises'
    },
    {
        path:'#',
        display:'Shop Now'
    },
]
const Header = () => {
  return(
    <header className='header'>
        <div className="container">
            <div className="nav__wrapper">
                {/*==============LOGO============*/}
                <div className="logo">
                    <div className="logo__img">
                        <img src={logo} alt=""/>
                    </div>
                    <h2>Health Haven</h2>
                </div>
                {/* ======= navigation menu========*/}
                <div className="navigation">
                    <ul className="menu">
                        {
                            nav__links.map(item=>(
                                <li className="nav__item"><a
                                    href={item.path}>
                                {item.display}</a></li>
                            ))
                        }
                    </ul>
                </div>
                {/*======nav right=====*/}
                <div className='nav__right'>
                    <button className='register_btn'>Register</
                        button>
                    <span className='mobile-menu'>
                    <i class="ri-menu-line"></i></span>
                </div>
            </div>
        </div>
    </header>
  )
}
export default Header;
```

1.4 Exercises.jsx

```
1  import React from 'react'
2  import "../../styles/exercises.css";
3  import lunges from "../../assets/img/lunges.png";
4  import yoga from "../../assets/img/yoga-pose.png";
5  import extended  from "../../assets/img/extended.png";
6  const Exercises = () => {
7    return <section>
8      <div className="container exercises__container">
9          <div className="exercise__top">
10             <h2 className="section__title">Benefits of <span
                   className="highlights">Exerises </span></h2>
11             <p>Lorem ipsum dolor sit amet consectetur
                   adipisicing elit.<br/>
12              Labore cum, cupiditate dolorem quia similique
                    dolore.</p>
13         </div>
14         {/*===========exercises list ==========*/}
15         <div className="exercise__wrapper">
16          <div className="exercise__item">
17             <span className="exercise__icon"><img src={lunges}
                   alt="" /></span>
18             <div className="exercise__content">
19                 <h4>Healthy Life</h4>
20                 <p>Lorem ipsum dolor sit amet, consectetur
                       adipisicing elit. Aut vero illum vitae
                       quibusdam nemo tempora.</p>
21         </div>
22          </div>
23          <div className="exercise__item">
24             <span className="exercise__icon">
25                 <img src={yoga} alt="" />
26             </span>
27             <div className="exercise__content">
28                 <h4>Increased Flexibility</h4>
29                 <p>Lorem ipsum dolor sit amet, consectetur
                       adipisicing elit. Aut vero illum vitae
                       quibusdam nemo tempora.</p>
30             </div>
31          </div>
32          <div className="exercise__item">
33             <span className="exercise__icon">
34                 <img src={extended} alt="" />
35             </span>
36             <div className="exercise__content">
37                 <h4>Reducing Blood Pressure</h4>
38                 <p>Lorem ipsum dolor sit amet, consectetur
                       adipisicing elit. Aut vero illum vitae
                       quibusdam nemo tempora.</p>
39             </div>
40          </div>
41         </div>
42     </div>
43   </section>
```

```
}
export default Exercises;
```

1.5 ShoppingCart.jsx

```
import React , {useState} from 'react';
import Navbar from "./components/Navbar";
import Amazon from './components/Amazon';
import Cart from './components/Cart';
import './styles/amazon.css';
const App = () => {
    const[show,setShow] = useState(true);
    const[cart,setCart] = useState([]);
    const [warning,setWarning] = useState(false);
    const handleClick = (item)=>{
      let isPresent = false;
      cart.forEach((product)=>{
        if (item.id === product.id)
        isPresent =true;
      })
      if (isPresent){
          setWarning(true);
          setTimeout(()=>{
            setWarning(false);
          },2000);
          return;
      }
      setCart([...cart,item ]);
    }
    const handleChange=(item, d)=>{
      let ind = -1;
      cart.forEach((data,index)=>{
        if (data.id === item.id)
          ind=index;
      });
      const tempArr=cart;
      tempArr[ind].amount+=d;
      if (tempArr[ind]===0)
        tempArr[ind].amount=1;
      setCart([...tempArr])
    }
  return (
    <React.Fragment>
      <Navbar size={cart.length} setShow={setShow}/>
      {
        show ? <Amazon handleClick={handleClick}/> : <Cart cart
            ={cart} setCart={setCart} handleChange={handleChange
            }/>
      }
      {
        warning && <div className='warning'>Item is already
            added to your cart</div>
      }
```

```
46      </React.Fragment>
47    )
48  }
49  export default App
```

1.5.1 Iteams.jsx

```
1   const list = [
2     {
3       id: 1,
4       title: "Boldfit Yoga Mat",
5       author: "183L x 61W x 0.6Th Centimeters | 25 April 2018",
6       price: 499,
7       img: "https://media.istockphoto.com/id/149027826/photo/
            isolated-pink-yoga-mat-slightly-unrolled.jpg?s=612x612&w
            =0&k=20&c=UWIAobquPuXspDPWEUvF19nJ90NihlcSMDHFW3XlqYc=",
8       amount: 1,
9     },
10    {
11      id: 2,
12      title: "Boldfit Gym Ball for Exercise Anti Burst Exercise
            Ball with Foot Pump",
13      author: "300 Grams|Rubber  | 1 September 2020",
14      price: 699,
15      img: "https://m.media-amazon.com/images/I/612c8i-J3dL.
            _SL1500_.jpg",
16      amount: 1,
17    },
18    {
19      id: 3,
20      title: "TYNOR Exercising Ball Pu",
21      author: "200 Grams|Safe for children | 25 November 2018",
22      price: 97,
23      img: "https://m.media-amazon.com/images/I/81zYltWcRQL.
            _SL1500_.jpg",
24      amount: 1,
25    },
26    {
27      id: 4,
28      title: "FEGSY High Density Foam Yoga Block Brick",
29      author: "Set of 2 with Yoga Strap for Improve Strength, Aid
            Balance, and Flexibility (Multicolor) | 25 April 2018",
30      price: 650,
31      img: "https://m.media-amazon.com/images/I/41EOXnSqEZL.
            _SX300_SY300_QL70_FMwebp_.jpg",
32      amount: 1,
33    },
34    {
35      id: 5,
36      title: "AmazonBasics 13.6 to 27.2 kg Resistance Pull Up Band
            ",
37      author: "3/4 Inch, Black(Material: Rubber) | 1 January
            2018",
38      price: 450,
```

```
39      img: "https://m.media-amazon.com/images/I/31MKPIq17fL._SX300_SY300_QL70_FMwebp_.jpg",
40      amount: 1,
41    },
42    {
43      id: 6,
44      title: "FirstFit Premium Single Toning Tube Resistance Band Exercise Cord with Comfort Handles",
45      author: "Material  TPE and Foam with High Quality Thick Latex Tube (Green)|Item Weight  224 Grams| 30 September 2020",
46      price: 399,
47      img: "https://m.media-amazon.com/images/I/71cbmuqZykL._SX679_.jpg",
48      amount: 1,
49    },
50    {
51      id: 7,
52      title: "FitBox Sports 4kg Hexa PVC Dumbbells ",
53      author: "(2kg x 2)  | 1 January 2021",
54      price: 175,
55      img: "https://m.media-amazon.com/images/I/61D-zNTp08L._SX679_.jpg",
56      amount: 1,
57    },
58    {
59      id: 8,
60      title: "Fitness Mantra  Originals Mens Special Design Sneaker Length Socks ",
61      author: "Free Size |Pack of 3 Pairs | Winter Wear Socks| 6 December 2020",
62      price: 289,
63      img: "https://m.media-amazon.com/images/I/71WRc3HoDTL._SX679_.jpg",
64      amount: 1,
65    },
66    {
67      id: 9,
68      title: "Boldfit Adjustable Hand Grip Strengthener",
69      author: "Forearm Exercise",
70      price: 200,
71      img: "https://m.media-amazon.com/images/I/51yu8-ihgNL._SX300_SY300_QL70_FMwebp_.jpg",
72      amount: 1,
73    },
74    {
75      id: 10,
76      title: "Sfane Polyester Shoulder Gym Bag",
77      author: "23cms|Separate Shoe Compartment (Grey)| 1 September 2021",
78      price: 679,
79      img: "https://m.media-amazon.com/images/I/31q--YOSqkS._SY300_SX300_.jpg",
80      amount: 1,
81    },
82    {
```

```
    id: 11,
    title: "Mens Thermal Suit",
    author: "Cotton | 14 December 2021",
    price: 711,
    img: "https://m.media-amazon.com/images/I/51d35byGmPL._SX569_.jpg",
    amount: 1,
  },
  {
    id: 12,
    title: "Boldfit Skipping Rope",
    author: "2 meters (Black)| 1 January 2015",
    price: 157,
    img: "https://m.media-amazon.com/images/I/41cLGNGAQrL._SX300_SY300_QL70_FMwebp_.jpg",
    amount: 1,
  },
];
export default list;
```

1.6 Targets.jsx

1.6.1 Targets storing on backend

```
// models/data.js
const mongoose = require('mongoose');
const dataSchema = new mongoose.Schema({
  // Update this schema based on the actual structure of your data
  caloriesValue: {
    type: Number,
    required: true,
  },
  sleepValue: {
    type: Number,
    required: true,
  },
  stepsValue: {
    type: Number,
    required: true,
  },
  waterValue: {
    type: Number,
    required: true,
  },
  weightValue: {
    type: Number,
    required: true,
  },
  workoutValue: {
    type: Number,
    required: true,
  },
```

```
29    formattedDate: {
30      type: String,
31      required: true,
32    },
33  });
34  const DataModel = mongoose.model('Data', dataSchema);
35  module.exports = DataModel;
```