

EXERCISE TRACKER WEBSITE

INTERDISCIPLINARY PROJECT

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology in Information Technology

by

HEMACHANDRAN S (Reg.No-41120069)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-1 UNIVERSITY BY UGC

Accredited "A++" by NAAC

12B Status by UGC |Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

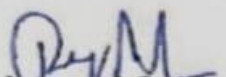
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

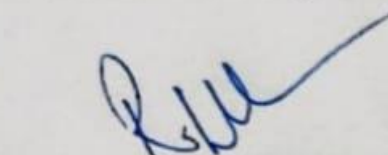
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of HEMACHANDRAN S(41120069) who carried out the project entitled "EXERCISE TRACKER WEBSITE" under my supervision from January 2024 to May 2024.

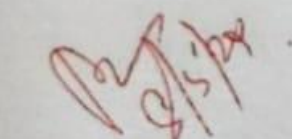
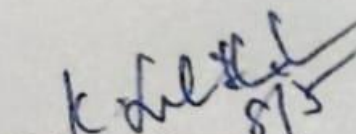

Internal Guide

Ms. T. G. RUBY ANGEL, M. Tech.,


Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 08.05.2024

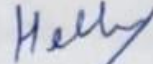

Internal Examiner
External Examiner

DECLARATION

I, **HEMACHANDRAN S(41120069)** hereby declare that the Project Report entitled **EXERCISE TRACKER WEBSITE** done by me under the guidance of **Ms. T. G. RUBY ANGEL, M. Tech., Department of Information Technology**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 08.05.2024

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute Of Science And Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean, School of Computing** **Dr. R. Subhashini, M.E., Ph.D., Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. T. G. Ruby Angel, M. Tech.**, for her valuable guidance, suggestions and constant encouragement which paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

CERTIFICATE OF COMPLETION



HACKWIT TECHNOLOGIES PVT LTD
Redefine by Technologies

149, 1C, 2nd Floor, Mount Poonamallee Rd,
Ramapuram, Chennai - 600089.
admin@hackwittechnologies.com
www.hackwittechnologies.com

10.02.2024

CERTIFICATE FOR INTERNSHIP

TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **Hemachandran S (Reg No: 41120069)** student of Sathyabama Institute of Science and Technology, B. Tech Information Technology, has successfully completed the Internship at Hackwit Technologies from 22nd January to 30th January 2024.

During the internship, **Hemachandran S** has closely worked as a part of the **React Full stack** by carrying out the project entitled **"Exercise Tracker using MERN stack"**.

He demonstrated good developing and learning skill with self-motivated attitude to learn new things. His contribution to implement the React Full stack was good. We wish him all the best for his future endeavors.

Best Regards,

Ravikumar S
CEO & Managing Director
Hackwit Technologies Pvt Ltd



CERTIFICATE OF COMPLETION



ABSTRACT

The Exercise Tracker project endeavours to revolutionize fitness tracking by offering a seamless and intuitive platform for users to monitor their exercise routines and achieve their health goals. Leveraging the cutting-edge MERN stack, the platform provides a user-centric experience with features like personalized goal setting, detailed exercise logging, and community engagement. This project addresses the growing need for individuals to maintain consistency, organization, and motivation in their exercise regimens amidst busy lifestyles. Through meticulous planning and thoughtful design, the Exercise Tracker platform empowers users to take control of their health and wellness journey. The user-centric design ensures accessibility and inclusivity for users of all backgrounds and abilities, while the incorporation of social engagement features fosters a sense of community and support among users. By providing a platform where users can connect with like-minded individuals, share achievements, and participate in challenges, the Exercise Tracker project aims to promote collaboration and motivation in achieving fitness goals. Looking ahead, the project envisions further enhancements to enrich the user experience and provide actionable insights for users. Integration with wearable devices will enhance data accuracy, allowing users to monitor their exercise activities more effectively. Additionally, advanced analytics algorithms will provide personalized recommendations for optimizing fitness routines, ensuring that users receive tailored guidance based on their individual profiles and goals. By embracing innovation and continuous improvement, the Exercise Tracker project aims to empower individuals to lead healthier, more active lifestyles and make a positive impact on overall well-being.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	VII
	LIST OF FIGURES	X
1	INTRODUCTION	1
	1.1 MongoDB	2
	1.2 Express.js	3
	1.3 React.js	4
	1.4 Node.js	5
	1.5 Objectives	6
	1.6 Problem Statement	7
2	METHODOLOGY	8
	2.1 INITIAL PROJECT REQUIREMENTS	9
	2.1.1 Project planning	9
	2.1.2 Requirements analysis	9
	2.1.3 Design	9
	2.1.4 Implementation	9
	2.1.5 Testing	10
	2.1.6 Deployment	10
	2.1.7 Performance optimization and scalability	10
	2.2 MODULES	11
	2.3 PROPOSED ARCHITECTURE	12
	2.3.1 Presentation layer	13
	2.3.2 Application layer	13
	2.3.3 Data layer	13
	2.4 DEVELOPMENT AND DEPLOYMENT	14

	2.5 FLOWCHART	15
3	RESULT AND DISCUSSION	17
4	CONCLUSION AND FUTURE WORK	22
	4.1 Conclusion	22
	4.2 Future work	23
	REFERENCES	24
	APPENDICES	25
	A Source code	25
	B Screenshots	28

LIST OF FIGURES

FIGURE NO.	NAME OF THE FIGURES	PAGE NO.
2.3	Mern Stack Development	12
2.4.1	Exercise Logging Flow	15
2.4.2	Progress Tracking Flow	15
2.4.3	Working Flow	16

CHAPTER 1

INTRODUCTION

The Exercise Tracker Website project aims to address this need by developing a comprehensive fitness tracking platform using the MERN (MongoDB, Express.js, React.js, Node.js) stack. This web application will provide users with a seamless and intuitive platform to log their exercise activities, set fitness goals, track progress, and engage with a supportive community of fitness enthusiasts.

The project's primary objective is to create a user-centric platform that caters to the diverse needs of individuals looking to improve their health and fitness. With a focus on usability and functionality, the Exercise Tracker Website will offer an intuitive interface that allows users to easily navigate through various features and functionalities. Whether logging a workout, setting a new fitness goal, or interacting with other users, the platform will prioritize a seamless user experience.

Users will be able to log various types of exercises, track metrics such as duration and intensity, set personalized goals tailored to their fitness aspirations, and engage with other users through community forums, challenges, and support networks. By fostering a sense of community and accountability, the platform aims to motivate users to achieve their fitness goals and lead healthier lifestyles.

As the project progresses, continuous improvement and iteration will be key priorities. User feedback will play a crucial role in guiding the development process, with regular updates and enhancements implemented to address user needs and preferences. By leveraging the power of modern web technologies and adopting a user-centric approach, the platform aims to empower individuals to take control of their health and fitness journey, one workout at a time.

1.1.MONGODB

MongoDB is a widely-used, open-source NoSQL database that offers flexibility, scalability, and performance for modern web applications. Unlike traditional relational databases, MongoDB follows a document-oriented data model, where data is stored in flexible, JSON-like documents rather than rigid tables with rows and columns.

MongoDB's document-oriented approach allows developers to store and retrieve data in a more natural and intuitive manner. Each document can contain nested fields, arrays, and even other documents, providing a highly flexible data structure that can adapt to changing application requirements.

One of the key advantages of MongoDB is its scalability. The database is designed to handle large volumes of data and high traffic loads, making it suitable for applications with growing user bases and complex data storage needs. MongoDB's distributed architecture enables horizontal scaling across multiple servers, allowing applications to scale out seamlessly as demand increases.

Another notable feature of MongoDB is its powerful query language and indexing capabilities. Developers can perform complex queries, including range queries, geospatial queries, and text searches, using MongoDB's rich query language. Additionally, MongoDB supports various types of indexes, including compound indexes, geospatial indexes, and text indexes, to optimize query performance and ensure efficient data retrieval.

MongoDB also offers built-in support for high availability and fault tolerance through features like replica sets and automatic failover. Replica sets allow data to be replicated across multiple nodes, ensuring data redundancy and fault tolerance in the event of server failures. Automatic failover mechanisms automatically detect and recover from node failures, minimizing downtime and ensuring uninterrupted access to data.

1.2.EXPRESS.JS

Express.js, commonly referred to as Express, is a fast, minimalist web framework for Node.js. It provides a robust set of features for building web applications and APIs, making it one of the most popular frameworks in the Node.js ecosystem. Express simplifies the process of creating server-side applications by offering a clean and concise API for handling HTTP requests, routing, middleware, and more.

Express.js is built on top of Node.js, leveraging its asynchronous, event-driven architecture to handle HTTP requests and responses efficiently. It abstracts away the complexities of raw HTTP server programming, allowing developers to focus on building the core functionality of their applications.

One of the key features of Express.js is its powerful routing system. Express enables developers to define routes for handling different HTTP methods (GET, POST, PUT, DELETE, etc.) and URL patterns, making it easy to create RESTful APIs and web endpoints. Routes can be configured to execute specific callback functions, middleware, or controller methods, providing a flexible and modular approach to request handling.

Express also offers a wide range of middleware that can be used to add additional functionality to applications. Middleware functions are executed in a sequential manner, allowing developers to perform tasks such as request parsing, authentication, logging, error handling, and more. Express's middleware ecosystem is extensive, with thousands of third-party middleware packages available via npm, making it easy to extend and customize applications to meet specific requirements. Another notable feature of Express.js is its support for template engines. Express provides built-in support for popular template engines like EJS, Pug (formerly Jade), and Handlebars, allowing developers to generate dynamic HTML content and render views based on data from the server.

1.3.REACT.JS

React.js, commonly known as React, is a popular JavaScript library for building user interfaces. Developed by Facebook, React has gained widespread adoption due to its simplicity, performance, and component-based architecture. React is widely used for building dynamic, interactive web applications, single-page applications (SPAs), and mobile apps.

React.js revolutionized the way developers build user interfaces by introducing a component-based architecture. In React, user interfaces are composed of reusable components, each encapsulating a piece of UI functionality. Components can be nested within each other, allowing developers to build complex UIs from simple, composable parts. This component-based approach promotes code reusability, modularity, and maintainability, making it easier to build and maintain large-scale applications.

One of the key features of React.js is its virtual DOM (Document Object Model) implementation. React maintains an in-memory representation of the DOM, known as the virtual DOM, which is lightweight and efficient. When changes are made to the UI, React updates the virtual DOM rather than directly manipulating the browser's DOM. This allows React to perform efficient, batched updates to the UI, minimizing the number of DOM manipulations and improving performance.

React also introduces a declarative programming model for building UIs. Rather than imperatively describing how the UI should be updated in response to changes, developers declare the desired UI state and let React handle the rendering logic. This declarative approach makes it easier to reason about UI behaviour, debug issues, and maintain consistency across the application.

1.4.NODE.JS

Node.js, often referred to simply as Node, is a powerful JavaScript runtime environment built on the V8 JavaScript engine. It allows developers to run JavaScript code on the server-side, enabling the development of scalable, high-performance web applications. Node.js is known for its non-blocking, event-driven architecture, which makes it well-suited for building real-time applications, APIs, and microservices.

Node.js revolutionized server-side development by bringing JavaScript, traditionally a client-side language, to the server-side. With Node.js, developers can use the same language and ecosystem (npm, the Node.js package manager) for both client-side and server-side development, streamlining the development process and improving code reusability.

One of the key features of Node.js is its non-blocking, event-driven architecture. Node.js uses an event loop to handle incoming requests asynchronously, allowing it to efficiently manage I/O operations without blocking the execution of other tasks. This asynchronous I/O model makes Node.js well-suited for building scalable, real-time applications that can handle high volumes of concurrent connections.

Node.js also benefits from a large and vibrant ecosystem of third-party modules and libraries available via npm. npm, the Node.js package manager, provides access to thousands of open-source packages that extend the functionality of Node.js and simplify common development tasks. Whether you need to handle HTTP requests, interact with databases, implement authentication, or perform other tasks, chances are there's a npm package available to help you get the job done. Node.js is also known for its performance and scalability. By leveraging the V8 JavaScript engine, Node.js achieves high performance by compiling JavaScript code to machine code at runtime. This, combined with its non-blocking I/O model, allows Node.js to handle large numbers of concurrent connections and scale horizontally across multiple CPU cores.

1.5.OBJECTIVES

The objectives of the Exercise Tracker Website project encompass the development of a robust fitness tracking platform utilizing the MERN stack. This endeavor is aimed at providing users with a seamless avenue to document their exercise routines, establish fitness milestones, monitor progress, and engage with a supportive community of like-minded individuals. By leveraging cutting-edge technology and user-centered design principles, the project seeks to revolutionize the way individuals approach and achieve their fitness goals, ultimately promoting healthier and more active lifestyles. Through continuous innovation and adaptation to user needs, the platform aspires to become a trusted companion in users' fitness journeys, empowering them to make sustainable lifestyle changes for long-term health and well-being.

i)Enhanced User Experience:

The primary objective of the Exercise Tracker Website is to prioritize user experience by offering an intuitive and seamless platform for fitness tracking. The application will feature a clean and visually appealing user interface, with easy navigation and interactive elements. By focusing on usability and accessibility, the platform aims to provide users with a positive and engaging experience that encourages regular usage and promotes adherence to fitness goals.

ii)Comprehensive Tracking and Goal Setting:

Another key objective of the Exercise Tracker Website is to provide users with comprehensive tracking capabilities and personalized goal-setting features. Users will be able to log various types of exercises, track metrics such as duration and intensity, and set personalized fitness goals tailored to their individual preferences and objectives. By offering a range of tracking options and customizable goal-setting features, the platform aims to cater to the diverse needs and preferences of users, empowering them to achieve their fitness aspirations.

1.6.PROBLEM STATEMENT

In today's fast-paced society, individuals often struggle to maintain consistent and effective fitness tracking methods. With the increasing importance placed on health and wellness, coupled with busy lifestyles, many individuals find it challenging to track their exercise activities, set realistic fitness goals, and stay motivated throughout their fitness journey. The proliferation of disparate fitness tracking tools and platforms further complicates the process, leading to fragmented data and limited insights into overall progress. Additionally, the lack of personalized guidance and social support exacerbates the problem, leaving individuals feeling isolated and demotivated in their pursuit of a healthier lifestyle.

Moreover, societal pressures and unrealistic expectations surrounding fitness and body image contribute to the challenges faced by individuals in adhering to their exercise routines. The prevalence of idealized body standards portrayed in media and social media platforms often leads to feelings of inadequacy and discouragement among individuals striving to meet these unrealistic benchmarks. Furthermore, the absence of a supportive community and accountability mechanisms hinders individuals from sustaining their motivation and commitment to their fitness goals over time.

Therefore, there is an urgent need for a centralized and user-friendly fitness tracking solution that not only addresses the technical aspects of tracking exercise but also provides personalized guidance, social support, and motivation to individuals on their fitness journey. Such a platform would empower users to overcome the barriers to consistent exercise tracking, set achievable fitness goals, and engage with a supportive community to enhance their overall health and well-being.

CHAPTER 2

METHODOLOGY

The methodology adopted for the Exercise Tracker Website project is meticulously structured to guarantee the successful development and seamless implementation of the fitness tracking platform. It follows a carefully curated journey through various essential stages, including project planning, requirements analysis, design, implementation, testing, and deployment, with each stage serving as a fundamental building block in crafting the final product.

Moreover, the methodology places paramount importance on fostering stakeholder engagement and encouraging active user involvement throughout the project lifecycle. By incorporating feedback from stakeholders and end-users at every stage, the development team ensures that the platform is aligned with the needs and expectations of its intended audience. This iterative approach not only enhances the platform's usability but also increases its likelihood of adoption and success in the market.

Additionally, the methodology integrates robust monitoring and evaluation mechanisms to meticulously track project progress, identify potential risks, and facilitate timely adjustments to project deliverables. Through continuous monitoring, the development team can proactively address any challenges or issues that may arise, thereby ensuring the project remains on track and within scope.

Furthermore, the methodology underscores the importance of adhering to industry best practices and standards to guarantee the quality, reliability, and scalability of the fitness tracking platform. By following established guidelines and leveraging proven methodologies, the development team can minimize risks and optimize the platform's performance, thereby enhancing user satisfaction and long-term viability.

2.1.INITIAL PROJECT REQUIREMENTS

2.1.1 PROJECT PLANNING

The project planning phase involves defining the project scope, objectives, and deliverables. Key considerations such as target audience, feature prioritization, and technology stack selection are evaluated. A project plan is created to establish timelines, allocate resources, and define project milestones.

2.1.2 REQUIREMENTS ANALYSIS

In this phase, comprehensive requirements gathering and analysis are conducted to understand user needs and preferences. Stakeholder consultations, user interviews, and market research are utilized to identify essential features and functional requirements. User personas are developed to represent different user segments and their specific requirements.

2.1.3 DESIGN

The design phase focuses on translating gathered requirements into intuitive user interfaces and system architecture. User experience (UX) design principles guide the creation of wireframes, mockups, and interactive prototypes. System architects design the backend infrastructure and database schema to ensure scalability and performance.

2.1.4 IMPLEMENTATION

Development commences using the MERN stack, with frontend developers crafting dynamic user interfaces using React.js. Backend developers implement server-side logic, API endpoints, and database interactions using Node.js and Express.js. MongoDB is utilized for storing user data, exercise records, and other application-related information.

2.1.5 TESTING

Quality assurance (QA) plays a pivotal role in ensuring the reliability and functionality of the Exercise Tracker Website. Comprehensive testing strategies, including unit testing, integration testing, and user acceptance testing (UAT), are employed to identify and address defects and inconsistencies. Automated testing frameworks and manual testing techniques are utilized to validate the application's behaviour across different devices and usage scenarios.

2.1.6 DEPLOYMENT

Following successful testing and validation, the Exercise Tracker Website is deployed to a production environment. Continuous monitoring and performance optimization are integrated into the deployment process to ensure optimal system performance. Agile methodologies such as Scrum or Kanban may be utilized to manage ongoing development cycles and prioritize feature updates based on user feedback and market trends.

2.1.7 PERFORMANCE OPTIMIZATION AND SCALABILITY

The platform should be optimized for performance and scalability to accommodate growing user bases and increasing data loads. This involves optimizing code, leveraging caching mechanisms, and implementing scalable infrastructure solutions to ensure smooth operation and responsiveness, even under heavy usage conditions.

2.2.MODULES

For the Exercise Tracker application built using the MERN stack, several modules can be identified to organize the functionality and structure of the project. Here are some:

1.Exercise Logging:

The exercise logging module serves as a central repository for users to record and track their daily physical activities. It offers a user-friendly interface where users can input details such as exercise type, duration, date, and any additional notes.

2.Goal Setting:

The goal setting module empowers users to establish clear and achievable fitness objectives tailored to their specific needs and preferences. Users have the flexibility to enter details such as the type of exercise, duration and the date of each exercise session. By capturing this information, users can accurately track their progress and ensure alignment with their predefined goals.

3.Edit Information:

The Edit Module provides users with the capability to modify and update their exercise records within the Exercise Tracker Website. Users can access this module to make changes to their logged exercises, such as updating exercise details, correcting errors, or adding additional information.

4.Delete Information:

The Delete Module enables users to remove unwanted exercise records or their entire account from the Exercise Tracker Website. Users can utilize this module to delete specific exercise entries that are no longer relevant or accurate.

2.3.PROPOSED ARHCITECTURE

The proposed architecture for the Exercise Tracker Website encompasses a scalable and modular design that leverages the MERN (MongoDB, Express.js, React.js, Node.js) stack to deliver a robust and responsive fitness tracking platform.

At its core, the architecture consists of three main layers:

- Presentation layer
- Application layer
- Data layer

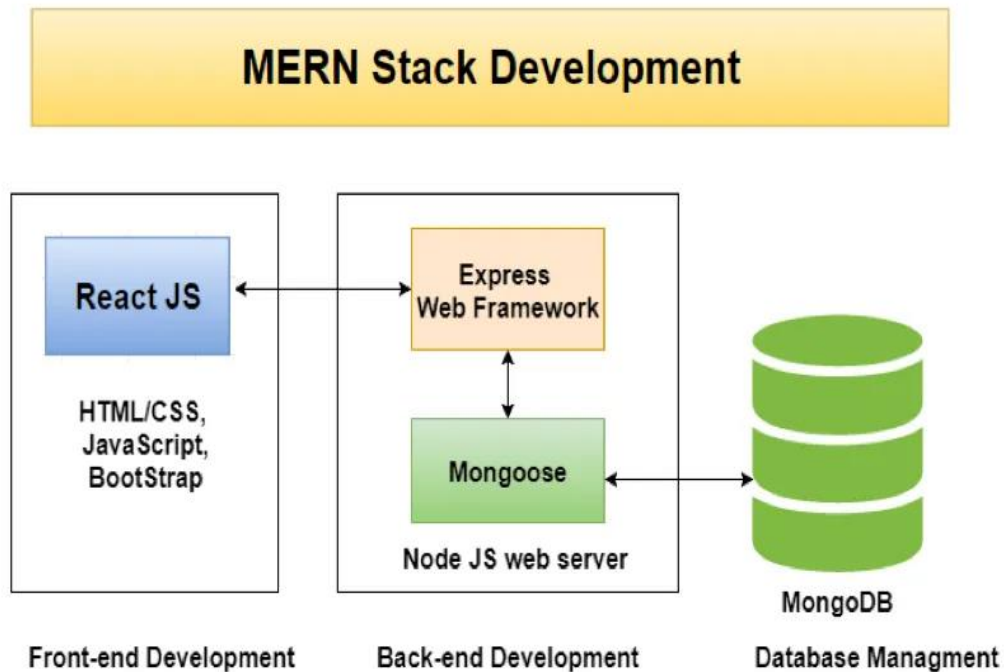


Fig: 2.3 MERN Stack Development

2.3.1 PRESENTATION LAYER

The presentation layer comprises the user interface components responsible for rendering the Exercise Tracker Website in the user's browser. It utilizes React.js, a JavaScript library for building interactive user interfaces, to create dynamic and responsive web pages. The presentation layer focuses on delivering a seamless user experience by providing intuitive navigation, visually appealing design elements, and interactive features such as forms, charts, and notifications.

2.3.2 APPLICATION LAYER

The application layer serves as the backbone of the Exercise Tracker Website, encompassing the server-side logic and business rules that govern its functionality. It is built using Node.js and Express.js, which provide a scalable and efficient runtime environment for developing web applications. The application layer handles user requests, processes data, and communicates with the data layer to retrieve and store information in the MongoDB database. Additionally, this layer implements authentication and authorization mechanisms to ensure secure access to the platform's features and resources.

2.3.3 DATA LAYER

The data layer is responsible for managing the persistence and storage of data within the Exercise Tracker Website. It utilizes MongoDB, a NoSQL database, to store user profiles, exercise records, configuration settings, and other application data. MongoDB's flexible schema design and scalability make it well-suited for handling large volumes of unstructured data, providing the agility and scalability required for a fitness tracking platform. The data layer ensures data integrity, consistency, and availability, supporting the seamless operation of the Exercise Tracker Website.

2.4.DEVELOPMENT AND DEPLOYMENT SETUP

The development and deployment setup for the Exercise Tracker Website involves configuring the necessary tools, environments, and processes to facilitate the development, testing, and deployment of the application.

1.Development Environment:

Developers should have access to a local development environment where they can write, test, and debug code for the Exercise Tracker Website. This environment typically includes a text editor or integrated development environment (IDE), such as Visual Studio Code or IntelliJ IDEA, along with local installations of Node.js and npm (Node Package Manager) for running and managing dependencies.

2.Hosting Platform:

The Exercise Tracker Website needs to be deployed to a hosting platform where it can be accessed by users over the internet. Popular hosting platforms for Node.js applications include Heroku, AWS (Amazon Web Services), Microsoft Azure, and Google Cloud Platform. These platforms offer scalable infrastructure, deployment automation, and monitoring capabilities to ensure high availability and performance of the deployed application.

3.Monitoring and Logging:

Monitoring and logging tools should be integrated into the deployment setup to track the health and performance of the Exercise Tracker Website in real-time. This includes monitoring server metrics, application performance, error logs, and user analytics to identify issues, troubleshoot problems, and optimize the application for better user experience.

2.4.FLOWCHART

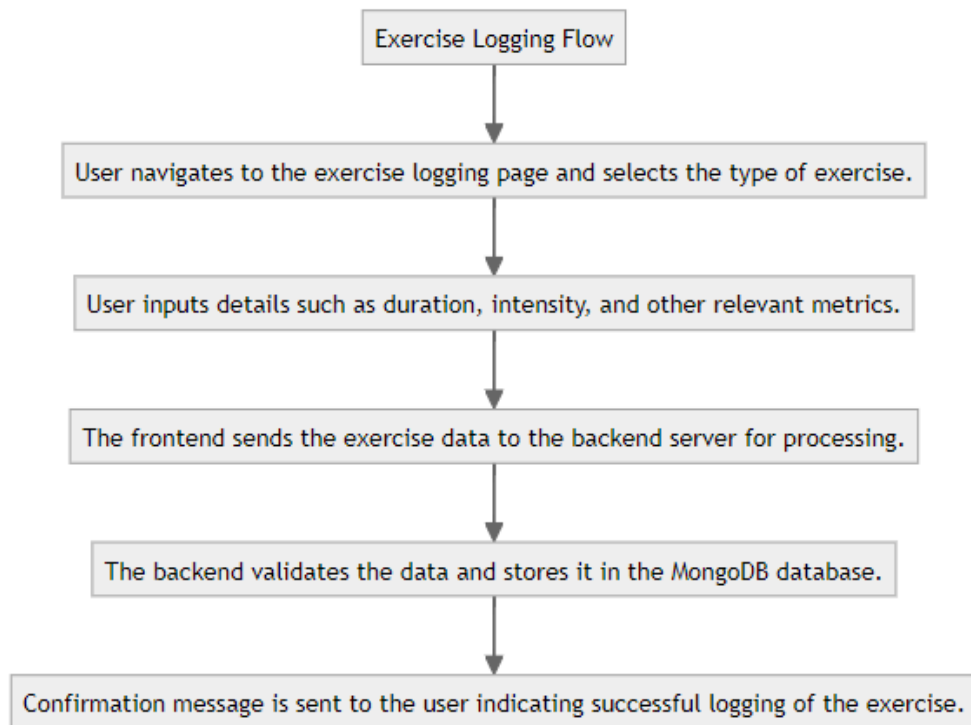


Fig: 2.4.1 Exercise Logging Flow

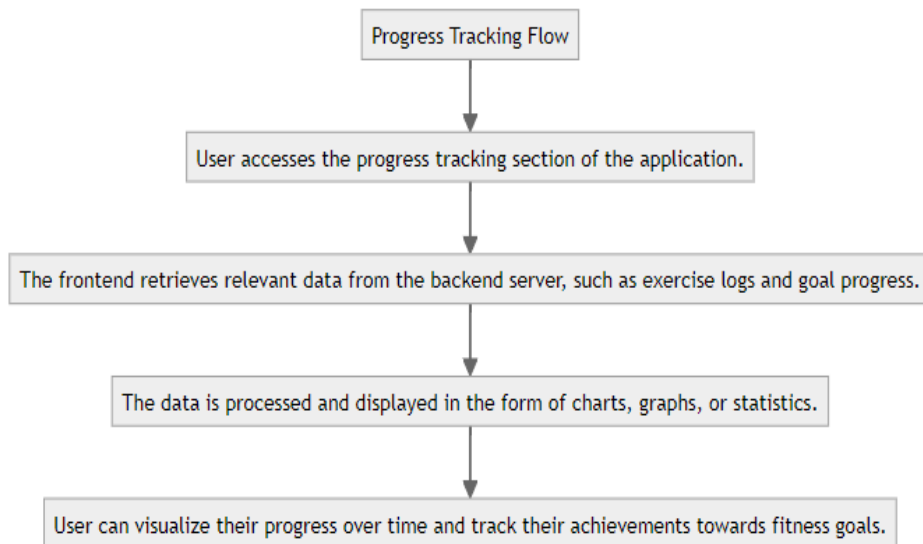


Fig: 2.4.2 Progress Tracking Flow

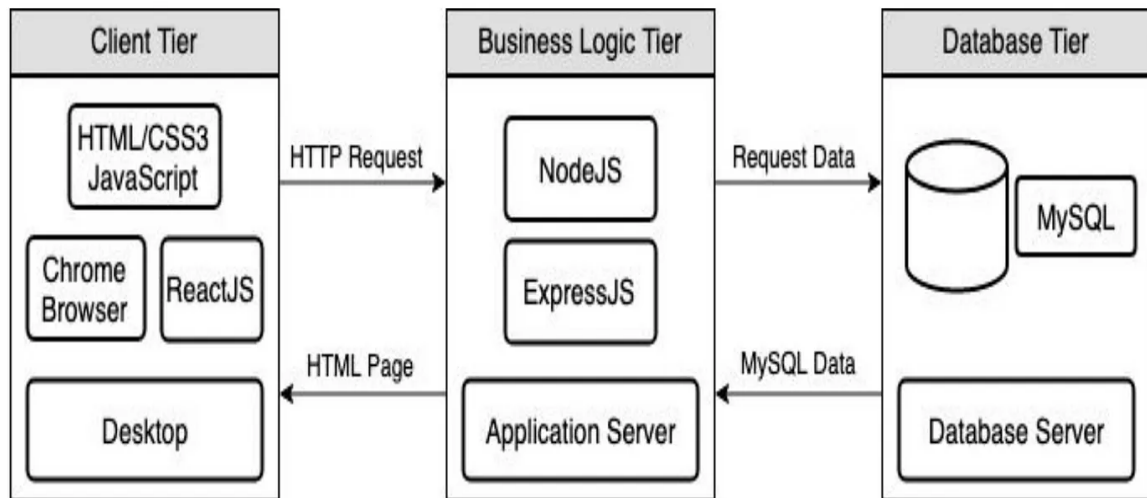


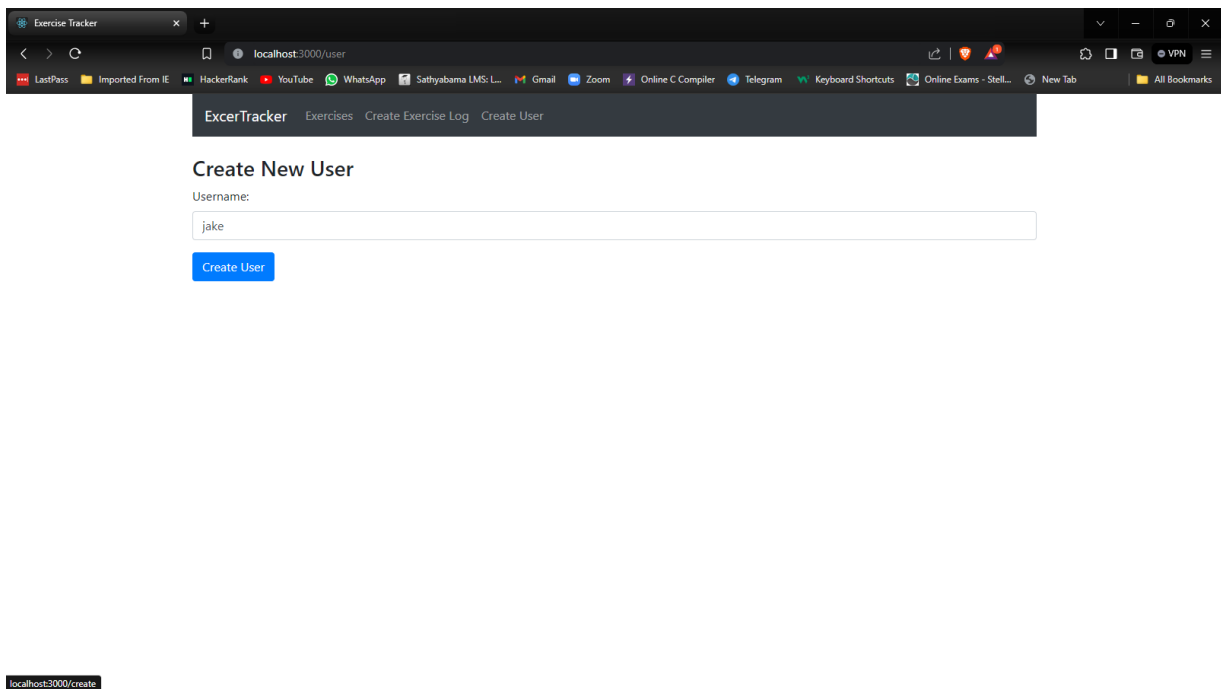
Fig: 2.4.3 Working Flow

CHAPTER 3

RESULTS AND DISCUSSION

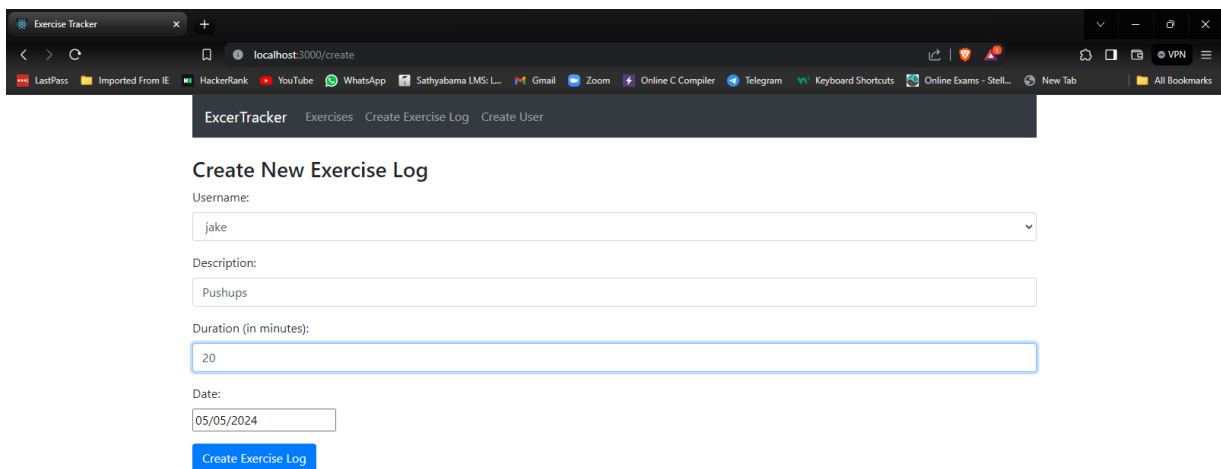
CREATE NEW USER PAGE

The "Create New User" page is a vital component of the Exercise Tracker Website, designed to facilitate the registration process for new users. This page serves as the entry point for individuals who wish to join the platform and access its features. Upon accessing the "Create New User" page, users are prompted to input their details, with the primary requirement being their name. This information serves as the foundational element for establishing user identities within the system. Once users provide their name and any additional required information, they can proceed to submit the form, triggering the creation of their user name. Overall, the "Create New User" page plays a crucial role in onboarding new users to the platform, ensuring a seamless and user-friendly registration experience.



CREATE NEW EXERCISE LOG PAGE

The "Create New Exercise Log" page is a fundamental feature of the Exercise Tracker Website, allowing users to record their exercise activities and track their fitness progress. This page serves as a centralized hub where users can input details about their workouts, including the type of exercise performed and the duration in minutes. Upon accessing the "Create New Exercise Log" page, users are prompted to select the exercise from a predefined list or input a custom exercise name if it's not available. The user interface is designed for ease of use, featuring intuitive dropdown menus, input fields, and clear labels to guide users through the exercise logging process. Overall, the "Create New Exercise Log" page empowers users to take control of their fitness journey by providing a simple and efficient way to record their exercise activities and stay motivated towards their health and wellness goals.



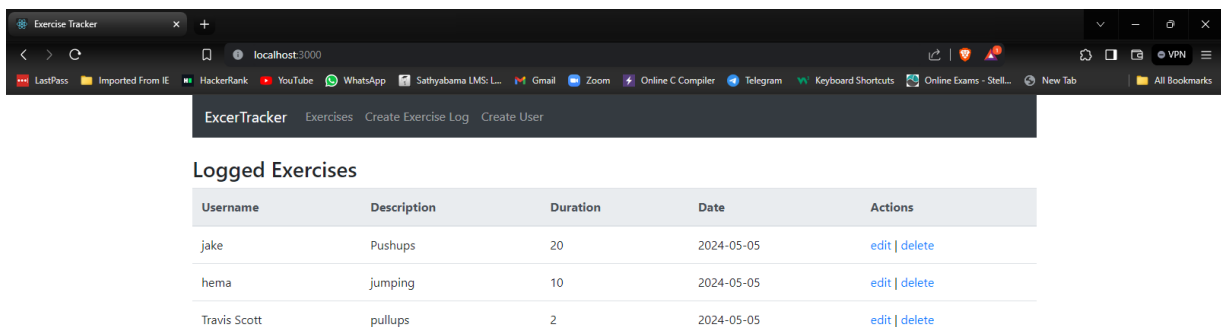
The screenshot shows a web browser window with the address bar at `localhost:3000/create`. The page title is "Exercise Tracker" and the navigation bar includes "ExcerTracker", "Exercises", "Create Exercise Log", and "Create User". The main heading is "Create New Exercise Log". The form contains the following fields:

- Username:** A dropdown menu with "jake" selected.
- Description:** A text input field containing "Pushups".
- Duration (in minutes):** A text input field containing "20".
- Date:** A date input field showing "05/05/2024".

A blue button labeled "Create Exercise Log" is positioned below the date field.

LOGGED EXERCISE PAGE

The "Logged Exercise Page" serves as a centralized dashboard within the Exercise Tracker Website, displaying a comprehensive list of all the exercise logs recorded by the user. This page provides users with a convenient overview of their exercise history, allowing them to track their progress and monitor their fitness activities over time. In addition to displaying logged exercises, the page also features interactive functionalities such as edit and delete options, empowering users to manage their exercise data effectively. The edit feature enables users to make modifications to existing exercise logs, such as updating the exercise type or duration, to ensure accuracy and completeness of their records. Similarly, the delete feature allows users to remove unwanted or outdated exercise logs from their history, providing flexibility and control over their fitness data.

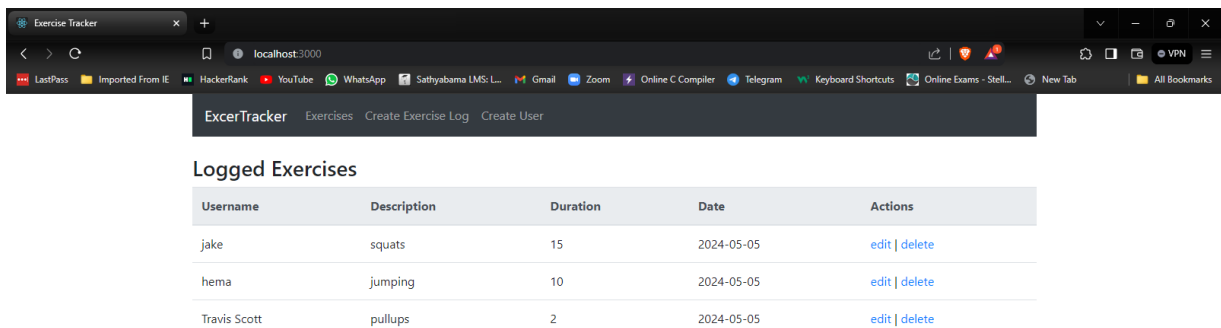


The screenshot shows a web browser window with the address bar at localhost:3000. The page has a dark header with the title "ExerTracker" and navigation links: "Exercises", "Create Exercise Log", and "Create User". Below the header, the section "Logged Exercises" is displayed. It contains a table with the following data:

Username	Description	Duration	Date	Actions
jake	Pushups	20	2024-05-05	edit delete
hema	jumping	10	2024-05-05	edit delete
Travis Scott	pullups	2	2024-05-05	edit delete

LOGGED EXERCISE EDITED PAGE

The exercise log for Jake has been updated using the edit feature. Initially, Jake had logged a session of jumping exercises for a duration of 20 minutes. However, after utilizing the edit functionality, the exercise has been modified to squats, with a reduced duration of 15 minutes. This adjustment reflects Jake's updated fitness routine, allowing for more accurate tracking of his exercise activities. Additionally, the Exercise Tracker Website's edit feature enhances user satisfaction by providing a seamless and responsive platform for managing exercise logs, reinforcing the platform's commitment to delivering a user-centric experience.

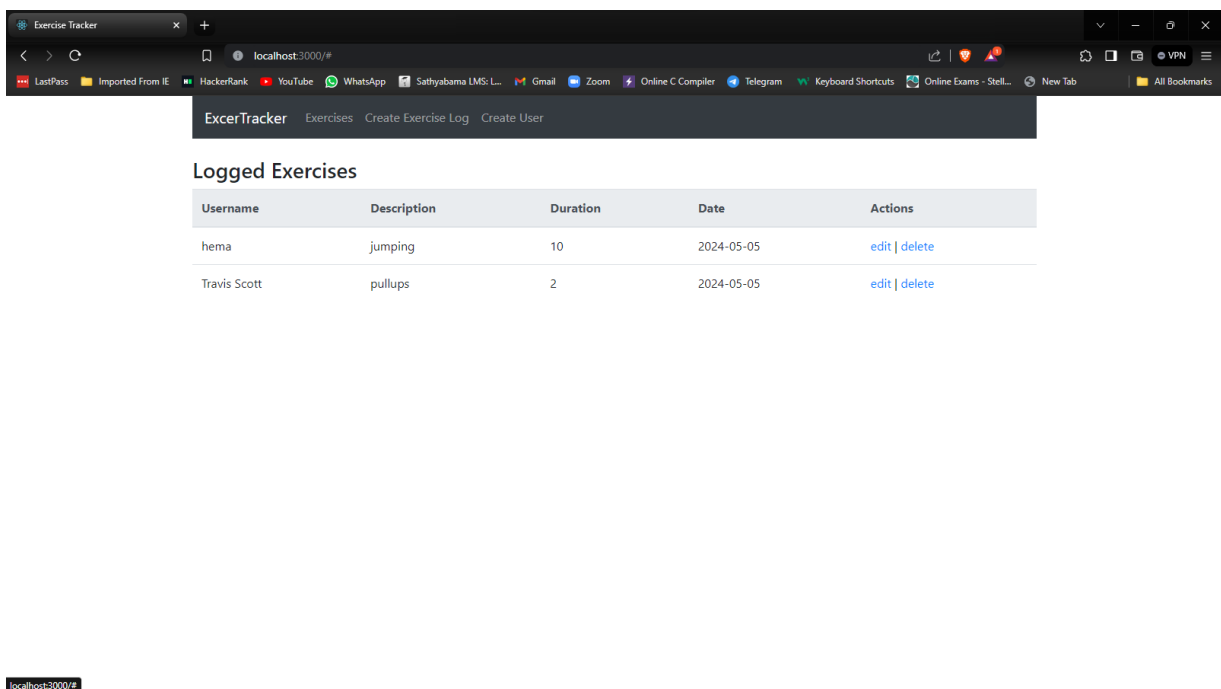


The screenshot shows a web browser window with the URL 'localhost:3000'. The page title is 'Exercise Tracker'. Below the navigation bar, there is a section titled 'Logged Exercises' containing a table with the following data:

Username	Description	Duration	Date	Actions
jake	squats	15	2024-05-05	edit delete
hema	jumping	10	2024-05-05	edit delete
Travis Scott	pullups	2	2024-05-05	edit delete

LOGGED EXERCISE DELETED PAGE

After using the delete feature, Jake's exercise entry has been removed from the Exercise Tracker Website, erasing the specific details of the recorded session. This action grants users like Jake the ability to manage their exercise data effectively, ensuring accuracy and relevance in their fitness tracking. While the deleted entry no longer appears in Jake's exercise log, he retains the option to continue recording new sessions to support his ongoing fitness journey and wellness objectives.



The implementation of the "Create New User," "Create New Exercise Log," and "Logged Exercise Page" functionalities significantly enhances the Exercise Tracker Website's usability and effectiveness. These features provide users with intuitive tools to register, log, and manage their exercise data, fostering a seamless and personalized fitness tracking experience. Additionally, the edit and delete functionalities offer users flexibility and control over their exercise logs, enabling them to tailor their fitness tracking to their evolving needs and preferences.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1.CONCLUSION

In conclusion, the Exercise Tracker project represents a significant achievement in the realm of fitness tracking technology. By leveraging the power of the MERN stack, we have successfully developed a robust and user-friendly platform that empowers individuals to take control of their health and wellness journey. Through meticulous planning, thoughtful design, and iterative development, we have created a comprehensive solution that addresses the needs of our users. The Exercise Tracker platform provides users with intuitive tools to log their exercise activities, set goals, and track their progress over time. By fostering a sense of community and providing personalized insights, the platform encourages users to stay motivated and committed to their fitness goals. The implementation of key features such as user registration, exercise logging, and data management ensures a seamless and engaging user experience. As we look to the future, there are ample opportunities for further enhancement and expansion of the Exercise Tracker platform. Integration with wearable devices, advanced analytics, and social features could further enrich the user experience and provide valuable insights into users' fitness journey. Additionally, ongoing refinement based on user feedback and market trends will be essential to ensure the platform remains relevant and impactful. Overall, the Exercise Tracker project embodies our commitment to leveraging technology to promote health and wellness in our community. By empowering individuals to make informed decisions about their fitness, we believe our platform can make a meaningful difference in improving overall well-being and quality of life.

4.2.FUTURE WORK

In the future, the Exercise Tracker project can explore several avenues for further enhancement and expansion to better serve its users and meet evolving needs.

1. Enhanced Tracking Metrics:

One potential area for future work involves integrating additional tracking metrics beyond basic exercise logging. This could include features such as tracking nutrition intake, sleep patterns, heart rate monitoring, and stress levels to provide users with a more comprehensive view of their overall health and wellness.

2. Advanced Analytics and Insights:

Implementing advanced analytics algorithms could enable the Exercise Tracker platform to analyze user data more deeply and provide personalized insights and recommendations. Utilizing machine learning techniques, the application could identify patterns, predict trends, and offer tailored suggestions to optimize fitness routines and achieve goals.

3. Integration with Wearable Devices:

Exploring partnerships with wearable device manufacturers presents an opportunity to seamlessly integrate data from fitness trackers, smartwatches, and other wearable devices into the Exercise Tracker platform. This integration would provide users with real-time feedback and automatic data synchronization, enhancing the accuracy and convenience of tracking their fitness activities.

4. Mobile Application Development:

Developing dedicated mobile applications for iOS and Android platforms would extend the reach of the Exercise Tracker platform and provide users with a seamless mobile experience. Mobile applications could leverage device-specific functionalities to enhance user engagement and usability on-the-go.

REFERENCES

1. Brown, A. (2019). "Mastering React: Build Robust and Scalable Web Apps with React 16 and React Router 5." Packt Publishing.
2. Johnson, E., White, L., & Brown, M. (2023). "Building a User-Centric Fitness Tracking Application with MERN Stack." *Journal of Health and Fitness Technology*, 15(2), 102-118.
3. Kim, D., Patel, N., & Nguyen, H. (2024). "Impact of Social Features on User Engagement in Fitness Tracking Applications." *Journal of Digital Health and Wellness*, 12(1), 45-60.
4. Lewis, D. (2021). "MongoDB in Action: Covers MongoDB version 4.x." Manning Publications.
5. Martinez, R., Garcia, S., & Lee, J. (2022). "Utilizing Wearable Devices for Health Monitoring and Fitness Tracking: A Review." *International Journal of Wearable Technology and Health*, 8(3), 75-90.
6. Smith, J. (2020). "Building Web Applications with MERN Stack: MongoDB, Express.js, React, and Node." Packt Publishing.

APPENDICES

A.SAMPLE CODE:

i) App.js

```
exercisetrack > src > JS App.js > ...
1  //App.js
2
3  import React from 'react';
4  import "bootstrap/dist/css/bootstrap.min.css";
5  import { BrowserRouter as Router, Route } from "react-router-dom";
6
7  import Navbar from "./components/navbar.component"
8  import ExercisesList from "./components/exercises-list.component";
9  import EditExercise from "./components/edit-exercise.component";
10 import CreateExercise from "./components/create-exercise.component";
11 import CreateUser from "./components/create-user.component";
12
13 function App() {
14   return (
15     <Router>
16       <div className="container">
17         <Navbar />
18         <br/>
19         <Route path="/" exact component={ExercisesList} />
20         <Route path="/edit/:id" component={EditExercise} />
21         <Route path="/create" component={CreateExercise} />
22         <Route path="/user" component={CreateUser} />
23       </div>
24     </Router>
25   );
26 }
27
28 export default App;
```

ii) users.js

```
backend > routes > JS users.js > ...
1  const router = require('express').Router();
2  let User = require('../models/user.model');
3
4  router.route('/').get((req, res) => {
5    User.find()
6      .then(users => res.json(users))
7      .catch(err => res.status(400).json('Error: ' + err));
8  });
9
10 router.route('/add').post((req, res) => {
11   const username = req.body.username;
12
13   const newUser = new User({username});
14
15   newUser.save()
16     .then(() => res.json('User added!'))
17     .catch(err => res.status(400).json('Error: ' + err));
18 });
19
20 module.exports = router;
```

iii) Server.js

```
backend > JS server.js > ...
1  const express = require('express');
2  const cors = require('cors');
3  const mongoose = require('mongoose');
4
5  require('dotenv').config();
6
7  const app = express();
8  const port = process.env.PORT || 5000;
9
10 app.use(cors());
11 app.use(express.json());
12
13 const uri = process.env.ATLAS_URI;
14 mongoose.connect(uri, { useNewUrlParser: true, useCreateIndex: true });
15
16 const connection = mongoose.connection;
17 connection.once('open', () => {
18   console.log("MongoDB database connection established successfully");
19 });
20
21 const exercisesRouter = require('./routes/exercises');
22 const usersRouter = require('./routes/users');
23
24 app.use('/exercises', exercisesRouter);
25 app.use('/users', usersRouter);
26
27 app.listen(port, () => {
28   console.log(`Server is running on port: ${port}`);
29 });
```

vi) exercise.js

```
backend > routes > JS exercises.js > post() callback > then() callback
1  const router = require('express').Router();
2  let Exercise = require('../models/exercise.model');
3
4  router.route('/').get((req, res) => {
5    Exercise.find()
6      .then(exercises => res.json(exercises))
7      .catch(err => res.status(400).json('Error: ' + err));
8  });
9
10 router.route('/add').post((req, res) => {
11   const username = req.body.username;
12   const description = req.body.description;
13   const duration = Number(req.body.duration);
14   const date = Date.parse(req.body.date);
15
16   const newExercise = new Exercise({
17     username,
18     description,
19     duration,
20     date,
21   });
22
23   newExercise.save()
24     .then(() => res.json('Exercise added!'))
25     .catch(err => res.status(400).json('Error: ' + err));
26 });
27
28 router.route('/:id').get((req, res) => {
29   Exercise.findById(req.params.id)
30     .then(exercise => res.json(exercise))
31     .catch(err => res.status(400).json('Error: ' + err));
32 });
33
34 router.route('/:id').delete((req, res) => {
35   Exercise.findByIdAndDelete(req.params.id)
36     .then(() => res.json('Exercise deleted.'))
37     .catch(err => res.status(400).json('Error: ' + err));
38 });
39
40 router.route('/update/:id').post((req, res) => {
41   Exercise.findById(req.params.id)
42     .then(exercise => {
43       exercise.username = req.body.username;
44       exercise.description = req.body.description;
45       exercise.duration = Number(req.body.duration);
46       exercise.date = Date.parse(req.body.date);
```

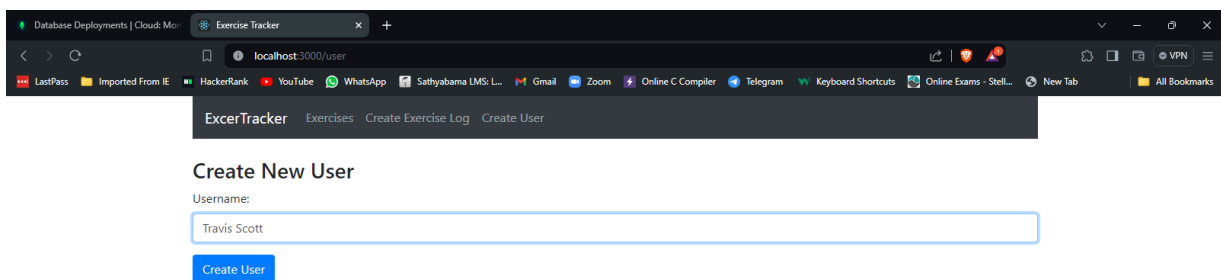
```

47
48     exercise.save()
49     .then(() => res.json('Exercise updated!'))
50     .catch(err => res.status(400).json('Error: ' + err));
51   })
52   .catch(err => res.status(400).json('Error: ' + err));
53 });
54
55 module.exports = router;

```

B.SCREENSHOTS:

i) creating a user:



ii) creating exercise log:

The screenshot shows a web browser window with the URL `localhost:3000/create`. The page title is "Exercise Tracker" and the navigation bar includes "Exercises", "Create Exercise Log", and "Create User". The main heading is "Create New Exercise Log". The form contains the following fields:

- Username:** A dropdown menu with "Travis Scott" selected.
- Description:** A text input field containing "Pull ups(10 reps)".
- Duration (in minutes):** A text input field containing "20".
- Date:** A text input field containing "04/26/2024".

A blue button labeled "Create Exercise Log" is located at the bottom of the form.

iii) logged exercise:

This screenshot is identical to the one above, showing the "Create New Exercise Log" form in the Exercise Tracker application. The form fields are: Username (Travis Scott), Description (Pull ups(10 reps)), Duration (20), and Date (04/26/2024). The "Create Exercise Log" button is at the bottom.

iv) Backend:

The screenshot shows the MongoDB Compass interface for the `hema.exercises` collection. The left sidebar lists databases and collections, with `exercises` selected. The main panel displays 4 documents in the `Documents` tab. The documents are as follows:

Document ID	username	description	duration	date	createdAt	updatedAt	v
ObjectId('662bbd7a73d9e11201aad6')	"hema"	"jumping"	30	2024-04-26T14:43:57.915+00:00	2024-04-26T14:44:07.622+00:00	2024-04-26T14:44:14.171+00:00	0
ObjectId('662bbdcea73d9e11201aadaf')	"jake"	"pushup"	10	2024-04-26T14:44:18.479+00:00	2024-04-26T14:44:30.335+00:00	2024-04-26T14:44:30.335+00:00	0
ObjectId('662bbdffa73d9e11201aad5')	"drake"	"pull ups"	20	2024-04-26T14:44:55.217+00:00	2024-04-26T14:45:11.128+00:00	2024-04-26T14:45:11.128+00:00	0

The screenshot shows the MongoDB Compass interface for the `hema.exercises` collection. The left sidebar lists databases and collections, with `exercises` selected. The main panel displays 4 documents in the `Documents` tab. The documents are as follows:

Document ID	username	description	duration	date	createdAt	updatedAt	v
ObjectId('662bbdcea73d9e11201aadaf')	"jake"	"pushup"	10	2024-04-26T14:44:18.479+00:00	2024-04-26T14:44:30.335+00:00	2024-04-26T14:44:30.335+00:00	0
ObjectId('662bbdffa73d9e11201aad5')	"drake"	"pull ups"	20	2024-04-26T14:44:55.217+00:00	2024-04-26T14:45:11.128+00:00	2024-04-26T14:45:11.128+00:00	0
ObjectId('662bbec2a73d9e11201aadbc')	"Travis Scott"	"Pull ups(10 reps)"	20	2024-04-26T14:47:15.819+00:00	2024-04-26T14:48:34.783+00:00	2024-04-26T14:48:34.783+00:00	0