PERSONAL HEALTH INFLUENCERS THROUGH SOCIAL NETWORK-PRESIFIT

A PROJECT REPORT

Submitted by,

SUSHRUTH SURESH – 20201CAI0216
SYED AFREEN AKTHER – 20201CAI0061
PRATHYUSH S PANICKER – 20201CAI0079
AKASH V – 20201CAI0080
AKASH R – 20201CAI0101

Under the guidance of,

MS. JOSEPHINE R
Assistant Professor

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **" Presifit - An App Which Connects Personal Health Influencers Through Social Media"** being submitted by "SUSHRUTH SURESH, SYED AFREEN AKTHER , PRATHYUSH S PANICKER, AKASH.V, AKASH.R " bearing roll number(s) " **20201CAI0216 , 20201CAI0061 , 20201CAI0079 , 20201CAI0080 , 20201CAI0101** " in partial fulfillment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

10/01/24.
Ms. JOSEPHINE R
Assistant Professor
School of CSE&IS
Presidency University

10/01/24
Dr. ZAFAR ALI KHAN
HOD-CSE
School of CSE&IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

10/1/24.
Dr. SHAKKEERA L
Associate Dean
School of CSE&IS
Presidency University

Dr. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled Presifit- An App Which Connects Personal Health Influencers Through Social Media in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms. JOSEPHINE R , Assistant Professor, School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

STUDENT NAME	ROLL NUMBER	SIGNATURE
SUSHRUTH S	20201CAI0216	Sushruth . S
SYED AFREEN AKTHER	20201CAI0061	Syed-A
PRATHYUSH S PANICKER	20201CAI0079	
AKASH R	20201CAI0101	
AKASH V	20201CAI0080	

ABSTRACT

This innovative application unveils a revolutionary API for the Interactive Health and Lifestyle (IHL) platform, focusing on fostering user engagement. The API empowers individuals to establish health challenges within their social circles, providing a gamified experience with incentives and badges strategically placed to drive sustained participation and goal accomplishment. The application underscores the API’s versatility, facilitating dynamic challenges across various health-related activities. Implemented in React Native, the platform ensures seamless navigation, allowing users to participate effortlessly in challenges related to fitness goals, habits, and health vitals.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C and Dr. Shakkeera L,** School of Computer Science Engineering & Information Science, Presidency University and **Dr. Zafar Ali Khan Head of the Department,** School of Computer Science Engineering & Information Science, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Ms. Josephine R , Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for her inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinator **Dr. Murali Parameshwaran**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

SUSHRUTH SURESH
SYED AFREEN AKTHER
PRATHYUSH S PANICKER
AKASH V
AKASH R

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

OVERVIEW

This report delves into the comprehensive development and deployment of a groundbreaking application known as "Presifit," specifically designed for the Interactive Health and Lifestyle (IHL) platform. The primary objective of Presift is to revolutionize user engagement within the health and lifestyle domain. By introducing a gamified experience, complete with incentives and badges strategically incorporated, the application aims to foster sustained participation and achievement of health-related goals.

While technology has made substantial strides in various industries, the field of health and lifestyle has seen a surge in demand for interactive and engaging platforms. Despite this, traditional methods often fall short in keeping individuals motivated towards their health objectives. Presifit addresses this gap by offering a versatile and dynamic framework for users to create and partake in health challenges within their social circles.

Implemented using React Native, the platform ensures a seamless and user-friendly interface, enabling participants to effortlessly navigate and engage in challenges related to fitness goals, habits, and health vitals. The Presifit's adaptability allows for the creation of diverse challenges, promoting an all-encompassing approach to individual well-being.

PROBLEM STATEMENT

In the evolving landscape of health and lifestyle, traditional approaches lack engagement, resulting in a disconnection between individuals and their wellness goals. The absence of interactive platforms has led to a decline in sustained health practices. "Presifit" addresses this issue, offering a gamified solution to bridge the gap, fostering user motivation and commitment. By enabling dynamic challenges and strategically placed incentives, the API aims to revolutionize how individuals actively engage with and achieve their health objectives, redefining the wellness journey.

CHAPTER-2

LITERATURE REVIEW

Smart Devices for Health and Wellness Applied to Tele-Exercise

(Antonio Fabbrizio , Alberto Fucarino , Manuela Cantoia , Andrea De Giorgio , Nuno D. Garrido , Enzo Iuliano , Victor Machado Reis , Martina Sausa , José Vilaça-Alves , Giovanna Zimatore , Carlo Baldari , and Filippo Macaluso).

This paper explores the advantages of mobile apps and tele-exercise platforms, offering accessible fitness options and real-time health monitoring. Smartphone apps, as shown by Silva et al.'s analysis, effectively increase physical activity, emphasizing the value of customized interventions. Handheld devices engage users of all ages, fostering positive habits and reducing sedentary behavior. Telemedicine provides remote support, overcoming geographical barriers, but limitations include potential dropout in gamified fitness platforms. Wearable tech offers continuous health monitoring, yet challenges may arise in user adherence and data accuracy. Inter-device communication dynamically adjusts exercise parameters, increasing satisfaction across age groups. Despite promising prospects, emerging tele-exercise technologies like virtual reality may face accessibility concerns and equipment requirements. Wearable tech applications show potential in early diagnosis and prevention. Overall, while there are advantages, careful customization and consideration of limitations are crucial for the successful integration of these technologies.

Application of Smart Wearable Fitness Equipment and Smart Health Management Based on the Improved Algorithm

(Haibo Cao)

This paper focuses on the advantages of a smart wearable fitness device, detailing its design principles, algorithm development, and noise analysis. Strategies for enhancing accuracy through noise filtering are explored, showcasing the device's potential. The selection of cloud service platforms is examined, emphasizing their utility in community settings. Future challenges are acknowledged, underscoring the need for viable solutions. The paper optimistically highlights the promising future of the smart wearables market. Despite these advantages, enterprises are cautioned to prioritize innovation and draw insights from management experiences. Government departments are urged to establish effective supervision systems and nurture talent to ensure the industry's healthy growth. It's crucial to note that limitations exist, and careful consideration of these challenges is necessary for the sustained success of smart wearable fitness devices.

Mobile Applications for Health and Wellness

(Alaa Alslaity, Rita Orji, Oladapo Oyebode, Jonathon R Fowles)

This paper underscores the increasing attention garnered by Mobile Health (mHealth) applications over the past two decades. The heightened popularity is attributed to the advanced technologies in smartphones, enabling the collection of diverse health-related data like physical activities, sleep patterns, and physiological metrics, proving invaluable for nursing purposes. While developing mHealth apps, designers stand to benefit from encouraging users to anticipate and overcome potential barriers, fostering a supportive environment for the target behavior. The apps also facilitate progress review, allowing users to export information and results externally, ultimately highlighting the efficacy of their actions. Despite these advantages, challenges may arise in ensuring seamless task reduction and division, and personalized services could potentially face limitations in scalability and individualization. Additionally, customization options may introduce complexity and user confusion, warranting careful consideration for optimal functionality and user experience.

SmartFit: A Step Count Based Mobile Application for Engagement in Physical Activities

(Atifa Sarwar , Hamid Mukhtar , Maajid Maqbool, Djamel Belaid)

This paper introduces a novel system that stands out from existing rigid goal-setting approaches by providing continuous motivation through a flexible and gamified method. This unique approach not only aids users in achieving their activity goals but also encourages sustained physical activity. Recognizing the individuality of walking speeds based on various factors, the system considers step count to classify users into activity levels. The formula, walking Duration(minutes) = StepCount / 100, offers a personalized metric, dividing users by duration rather than steps, reflecting a more accurate representation of their activity. SmartFit utilizes smartphone sensors and the Google Fit API for step tracking, awarding points for goal attainment and deducting points for falling behind, enhancing user engagement. While this approach proves advantageous in promoting physical activity, potential limitations may arise in accurately converting steps to activity duration, and gamification techniques may not resonate equally with all user demographics, warranting consideration for broader inclusivity and effectiveness.

Application of Internet of Things and Artificial Intelligence for Smart Fitness: A Survey

(Alireza Farrokhib, Reza Farahbakhsha, Javad Rezazadehb,c, Roberto Minervaa)

This paper explores the advantages of smart fitness integration, classifying it into fitness trackers (wearable and non-wearable sensors), movement analysis, and fitness applications. It underscores the transformative role of Artificial Intelligence (AI) in leveraging data from IoT-based smart fitness to enhance training performance. An advantageous aspect is the examination of sensor-to-sensor relationships through social-IoT, facilitating the exchange of training data and experiences across diverse locations and times. The comprehensive review encompasses various fitness trackers and applications, coupled with an exploration of AI algorithms, highlighting the potential benefits of smart fitness scenarios. However, potential limitations may include challenges in data security and privacy, and the effectiveness of AI algorithms may vary across user demographics. Despite these considerations, the paper emphasizes the vital role of data collection in optimizing training, refining health plans, and contributing to personalized health management. It concludes by identifying gaps in the current research landscape, suggesting avenues for future exploration and advancement in this dynamic field.

Health at hand: A systematic review of smart watch uses for health and wellness

(Blaine Reeder PhD, Alexandria David BS)

This paper delves into the positive outcomes of health-focused smartwatch applications developed using user-centered design methodologies, showcasing promising results from two studies. Surveys measuring technology perceptions highlighted CPR Watch's remarkable 93% positive feedback on its potential lifesaving capabilities. Smartwatches, rated highest (3.25/5) in a comparison survey of wearable device types, affirmed their widespread acceptance among participants. The Subjective Usability Scale (SUS) yielded satisfactory scores (70 and 74) in two studies, indicating "OK" to "Good" usability. Despite these advantages, methodological challenges were encountered, including technical issues such as limited battery power, signal noise, and data quality concerns. Usability challenges, including watch placement and small screen size hindering user experience, were also noted. Notably, varying completeness in reporting across studies underscores the necessity for standardized frameworks, focusing on factors like data quality, battery power, technology acceptance, and usability in smartwatch research.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

1. Gamification:

a. Existing health and wellness platforms often utilize gamification elements like badges, rewards, and leaderboards. However, "PresiFit" could explore:
b. Adaptive gamification: Adjusting difficulty levels and challenge types based on individual user preferences and progress.
c. Personalized rewards: Tailoring rewards to specific user motivations and interests for greater engagement.
d. Gamifying social interactions: Integrating game mechanics into social features to encourage collaboration and support

2. Social Networks:

While many platforms leverage social connections, "PresiFit" could delve deeper into:

a. Micro-communities: Creating smaller, focused groups based on shared interests or health goals for targeted support and accountability.
b. AI-powered social coaching: Utilizing AI to analyze user interactions and provide personalized social nudges for behavior change.
c. Social challenges and incentives: Encouraging group-based challenges and rewards to foster collaborative goal achievement.

3. Community Events:

a. Existing platforms often hold community events, but "PresiFit" could explore:
b. Gamifying community events: Integrating game mechanics into events like scavenger hunts or virtual races for increased engagement.
c. Partnering with diverse organizations: Collaborating with healthcare providers, educational institutions, and local businesses to expand reach and impact.
d. Data-driven event planning: Utilizing user data to optimize event schedules, locations, and activities for maximum participation.

4. Health Data Security and Utilization:

a. While data security is crucial, "PresiFit" could further explore:
b. Federated learning: Utilizing data without compromising privacy by keeping it on user devices and aggregating insights without revealing individual data.
c. Transparent data use policies: Clearly communicating how user data is used for research and development, with strong opt-in/out mechanisms.
d. AI-powered data analysis: Utilizing AI to identify trends and patterns in health data for personalized insights and preventative measures.

CHAPTER -4
REQUIREMENT ANALYSIS

6.1 HARDWARE COMPONENTS

6.1.1. Server Infrastructure:

Physical Servers or Cloud Infrastructure: Depending on the scale of the application, you might opt for physical servers housed in a data center or utilize cloud infrastructure services such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure.Cloud platforms offer scalability, flexibility, and robust security features.

Server Hardware Specifications:

Processor: Multi-core processors (e.g., Intel Xeon or AMD EPYC) to handle concurrent requests and data processing.

RAM: Adequate RAM (e.g., 64GB or more) to ensure smooth operations and accommodate in-memory data caching.

Storage: SSD storage for faster data retrieval and lower latency. Consider RAID configurations for data redundancy and fault tolerance.

6.1.2 Database Servers:

Database Management System (DBMS): MongoDB, being the chosen database for the MERN stack, requires dedicated servers or instances for storing, retrieving, and managing data.

Server Hardware Specifications:

Processor: Efficient processors to handle complex queries and data operations.

RAM: Sufficient memory to support database operations, caching, and indexing.

Storage: High-speed SSD storage with ample capacity to store user data, service listings, transaction details, etc.

6.1.3 Networking Components:

Load Balancers: Distribute incoming application traffic across multiple servers to ensure high availability and reliability.

Firewalls: Hardware firewalls to monitor and control incoming and outgoing traffic, enhancing the application's security posture.

Routers and Switches: Manage network traffic, facilitate communication between servers, and ensure seamless data transfer.

6.1.4 Storage Solutions:

Network-Attached Storage (NAS) or Storage Area Network (SAN): For centralized storage,backup, and data retrieval. These solutions offer scalability and high availability.

Backup Devices: Tape drives, external hard drives, or cloud-based backup solutions to create regular backups of critical application data.

6.1.5 Client Devices:

End-user Devices: Devices such as smartphones, tablets, or computers used by end-users to access the Presifit application. Ensure compatibility and optimal performance across various devices and operating systems.

Peripheral Devices: Accessories like printers, scanners, or biometric devices, if required for specific functionalities within the application.

6.1.6 Security Components:

Security Appliances: Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and other security appliances to monitor, detect, and mitigate potential security threats.

SSL Certificates: Secure Socket Layer (SSL) certificates to encrypt data transmitted between clients and servers, ensuring data privacy and integrity.

6.2 SOFTWARE COMPONENTS

6.2.1 Operating Systems:

Server OS: Choose a stable and secure server operating system like Linux distributions (e.g., Ubuntu Server, CentOS) or Windows Server, based on application compatibility and development preferences.

Client OS: Ensure the Presifit application is compatible with popular client operating systems, including iOS, Android, Windows, and macOS.

6.2.2 Development Frameworks and Libraries:

Front-end Framework:

React Native: Enables the development of native mobile applications for both iOS and Android using JavaScript and React.

Redux: Helps manage application state in complex React Native applications.

Back-end Framework:

Node.js: A runtime environment that executes JavaScript on the server side.

Express.js: A fast, unopinionated, and minimalist web framework for Node.js, facilitating the creation of robust APIs.

Database Framework:

MongoDB: A NoSQL database used to store and manage application data. Integration with Mongoose, an Object Data Modeling (ODM) library for MongoDB, can simplify data operations.

6.2.3 Database Management and Tools:

MongoDB Atlas: A cloud-based database service that provides automated backups, scaling, and monitoring of MongoDB databases.

MongoDB Compass: A GUI tool to visualize and interact with MongoDB data, facilitating database management and query execution.

6.2.4 Middleware Components:

Authentication and Authorization:

Passport.js: Middleware for Node.js that provides authentication strategies.

JWT (JSON Web Tokens): A compact, URL-safe means of representing claims to be transferred between two parties, ensuring secure data transmission.

Caching:

Redis: An in-memory data structure store used as a cache or message broker, enhancing application performance by reducing database load.

6.2.5 API and Integration Components:

RESTful APIs: Design and implement RESTful APIs using Express.js to facilitate communication between the client-side application and server-side logic.

Third-Party Integrations: Integrate necessary third-party services or APIs for functionalities such as payment processing, notifications, or geolocation services.

6.2.6 Testing and Quality Assurance Tools:

Jest and Mocha: Testing frameworks for JavaScript, enabling the creation of unit tests and ensuring code reliability.

Postman: A popular tool for testing APIs, facilitating API development, and ensuring endpoints function as expected.

6.2.7 Security Components:

SSL/TLS Certificates: Implement SSL/TLS encryption to secure data in transit and establish

a trusted connection between clients and servers.

Helmet.js: A middleware for Express.js that helps secure Express applications by setting various HTTP headers.

OAuth 2.0: Implement OAuth 2.0 authentication for secure and authorized access to the application's resources.

6.2.8 Deployment and Continuous Integration/Continuous Deployment (CI/CD) Tools:

Docker: Containerization tool to package the application and its dependencies into a standardized unit, ensuring consistent environments from development to production.

Kubernetes: Orchestration platform for managing containerized applications, facilitating deployment, scaling, and management of application containers.

Jenkins or GitLab CI/CD: CI/CD tools to automate the software delivery pipeline, ensuring efficient code integration, testing, and deployment processes.

6.3 VISUAL STUDIO:

Visual Studio is an integrated development environment (IDE) created by Microsoft that provides a comprehensive set of tools and features for software development. It offers a rich and user-friendly environment for building a wide range of applications, including desktop applications, web applications, mobile apps, cloud-based solutions, and more.

Visual Studio supports multiple programming languages, including C#, Visual Basic, C++, F#, Python, JavaScript, and TypeScript. It provides developers with a unified interface and a suite of powerful tools to write, debug, test, and deploy their code efficiently.

Key Features of Visual Studio:

● Code Editor: Visual Studio offers a robust code editor with features like syntax highlighting, code completion (IntelliSense), code refactoring, and customizable keyboard shortcuts. It enhances productivity and helps developers write clean and error-free code.

● Debugging and Diagnostics: Visual Studio includes a powerful debugging toolset that allows developers to step through code, set breakpoints, inspect variables, and analyze runtime behavior. It helps identify and fix bugs and issues during the development process.

● Integrated Testing Tools: Visual Studio provides built-in support for various testing frameworks and tools, enabling developers to write and execute unit tests, perform automated UI testing, and conduct performance profiling to optimize application performance.

● Integrated Version Control: Visual Studio integrates with popular version control systems like Git and Team Foundation Version Control (TFVC). It allows developers to easily manage source code, track changes, collaborate with team members, and handle branching and merging operations.

● Project and Solution Management: Visual Studio provides a flexible project and solution management system. Developers can create and manage multiple projects within a solution, organize files and dependencies, and control project settings and configurations.

● Extensibility and Marketplace: Visual Studio supports extensibility through a wide range of extensions and add-ons. Developers can enhance the IDE with additional tools, frameworks, and language support by accessing the Visual Studio Marketplace, which hosts a vast collection of community-created extensions.

● Cloud and Web Development: Visual Studio includes tools for developing cloud-native applications using Azure services, as well as web development tools for building modern web applications, including support for frameworks like ASP.NET and Node.js.

● Collaboration and Team Development: Visual Studio facilitates collaborative development by providing features such as Live Share, enabling real-time code sharing and collaboration among team members. It also integrates with project management tools like Azure DevOps for streamlined team coordination.

Visual Studio is widely used by developers worldwide due to its comprehensive set of features, flexibility, and extensive ecosystem. It caters to the needs of individual developers, small teams, and large enterprise-scale projects, making it a popular choice for software development across various platforms and industries.

CHAPTER-5

PROPOSED METHODOLOGY

1. Needs Assessment and Goal Setting:

- Survey the community to understand their health needs, preferences, and challenges.
- Set clear goals for the program based on the findings to address specific health concerns.

2. Personal Trainer Consultation:

- Offer one-on-one or group sessions with certified personal trainers.
- Provide personalized workout plans, nutritional advice, and fitness assessments.

3. Health Plan Tracking:

- Develop an app or platform for members to track their health progress, set goals, and monitor their fitness journey.
- Include features for logging workouts, meals, water intake, and other health-related activities.

4. Mental Health Support:

- Collaborate with mental health professionals to offer counseling sessions or support groups.
- Include resources like articles, videos, or podcasts on stress management, mindfulness, and mental wellness.

5. Virtual Events:

- Organize virtual fitness challenges, workshops, webinars, or cooking classes.
- Encourage community engagement through live Q&A sessions with health experts or trainers.

6. Rewards, Badges and Coupons:

- Implement a reward system where members earn points for achieving milestones or completing tasks.
- Offer badges or achievements for consistency, reaching fitness goals, or participation in community events. Provide discounts or coupons for health-related products, gym memberships, or wellness services.

7. Community Engagement and Support:

- Foster a sense of community by creating forums or social media groups where members can interact, share experiences, and support each other.
- Encourage peer-to-peer motivation and accountability.

8. Evaluation and Improvement:

- Regularly collect feedback from participants to assess the program's effectiveness and make necessary improvements.
- Analyze data to identify trends, areas of improvement, and success stories.

9. Marketing and Promotion:

- Promote the program through various channels like social media, newsletters, local events, and partnerships with relevant organizations or businesses.

10. Partnerships and Sponsorships:

- Collaborate with local businesses or health-related brands to offer exclusive discounts or sponsor certain program elements.

CHAPTER-6

OBJECTIVES

1. Create a User-Centric Social Health Platform:

- Develop a user-friendly and intuitive platform that encourages individuals to engage in health and wellness challenges.
- Prioritize user experience design to ensure ease of use and accessibility for a wide range of users.

2. Facilitate Social Connections:

- Enable users to connect with friends and contacts from social networks such as Facebook, LinkedIn, and WhatsApp.
- Foster a sense of community and support among users by facilitating group formation and communication.

3. Promote Healthy Habits:

- Design and implement challenges related to health and wellness goals, including step counts, healthy eating habits, and vitals improvement.
- Leverage gamification elements to motivate and reward users for making positive lifestyle changes.

4.Encourage Collaboration and Competition:

- Create a platform that allows users to collaborate with friends on health challenges and compete with others.
- Promote a healthy sense of competition to drive motivation and achievement.

5.Prioritize Data Privacy and Security:

- Establish robust data protection measures to safeguard user privacy and comply with relevant data protection regulations.
- Educate users about data handling practices and provide transparent privacy policies.

6. Provide Valuable Insights:

- Develop analytics and reporting tools to offer users insights into their progress and achievements.
- Use data-driven insights to suggest personalized challenges and goals.

CHAPTER-7

SYSTEM DESIGN & IMPLEMENTATION

USE CASE DIAGRAM

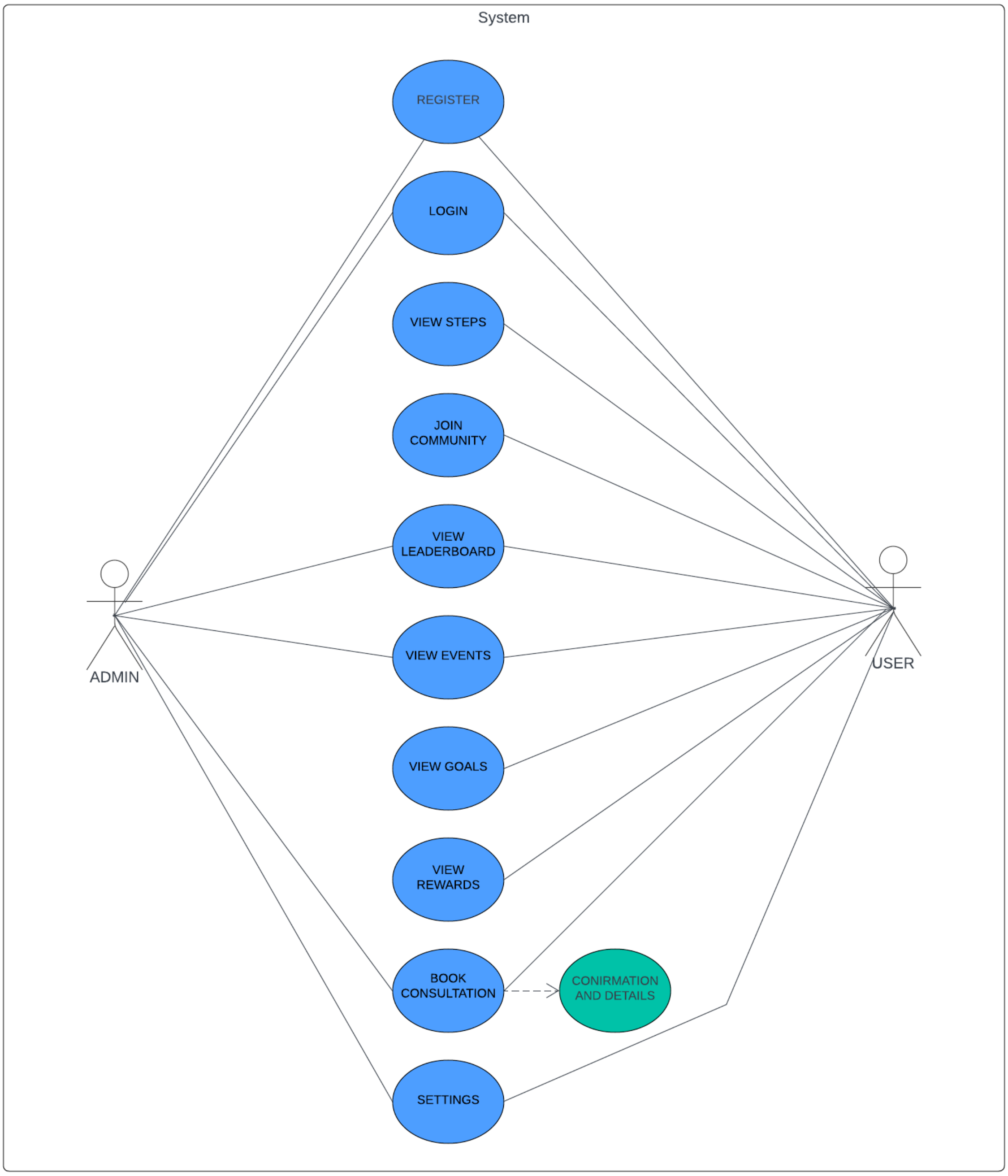


Figure 6.1 Use Case Diagram

CLASS DIAGRAM

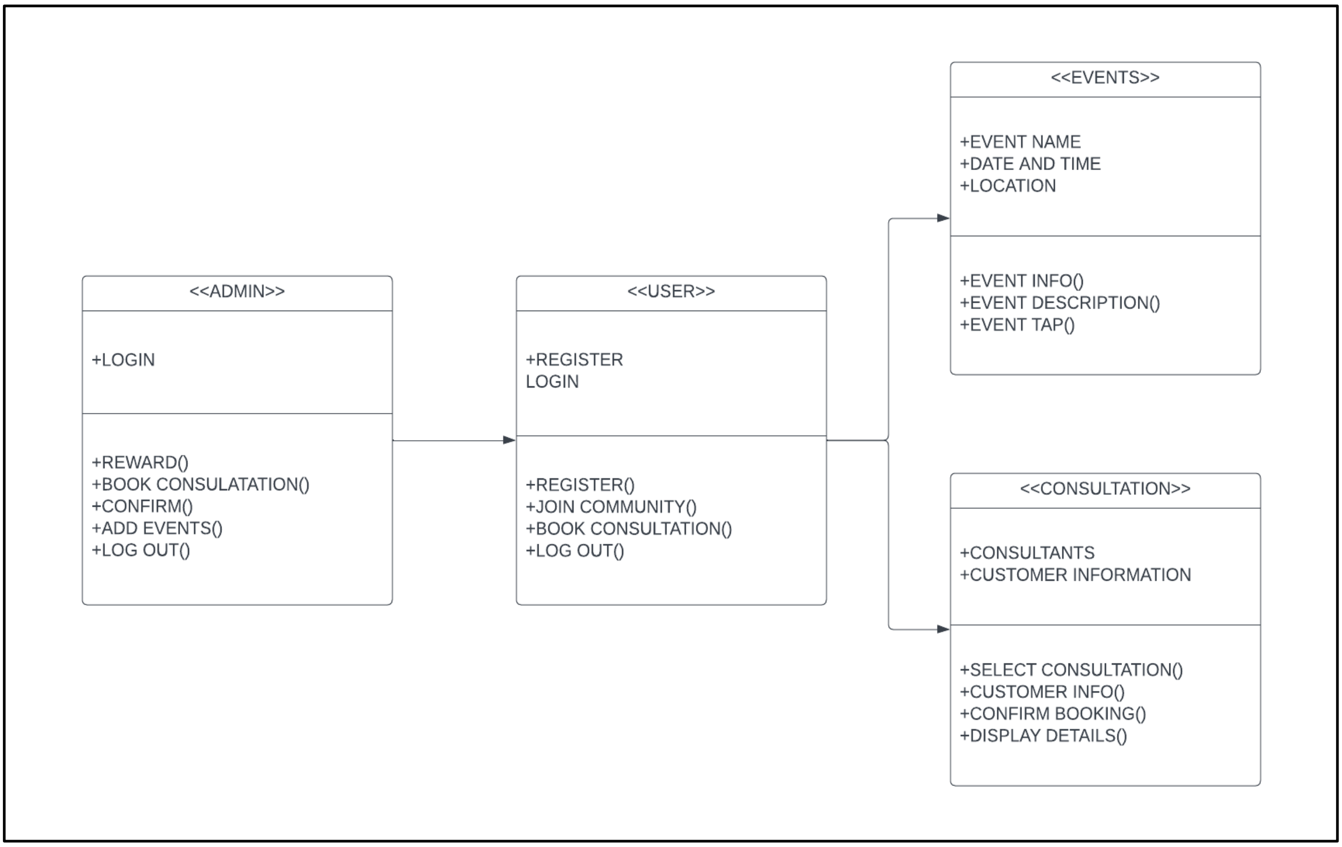


Figure 6.2 Class Diagram

Activity Diagram:

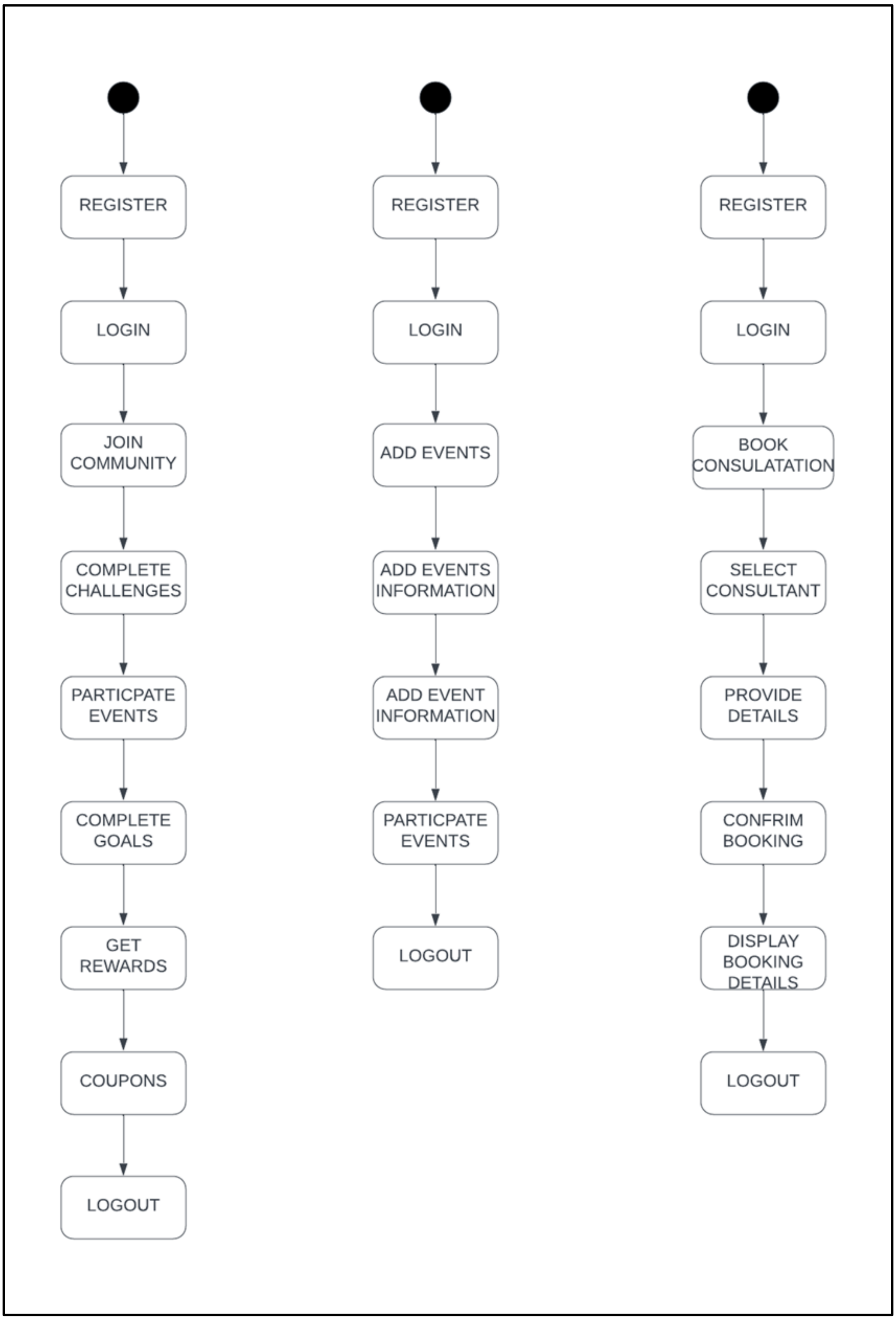


Figure 6.3 Activity Diagram

SYSTEM DESIGN

Architectural Style

Microservices Architecture: Presifit employs a microservices architecture, promoting modularity, scalability, and maintainability. Each service operates independently, facilitating continuous deployment and reducing system-wide failures.

Key Components

User Management Service: Responsible for user registration, authentication, profile management, and role-based access control. It integrates with authentication providers like OAuth and ensures data integrity via secure encryption mechanisms.

Service Request Service: This core component manages the lifecycle of service requests. It matches user requirements with professional skills, handles booking, scheduling, and ensures timely service delivery. Payment

Service: Integrates with payment gateways to facilitate secure transactions. It handles payment processing, invoicing, and ensures compliance with financial regulations.

Notification Service: Manages real-time notifications, sending alerts, updates, and reminders to users and professionals. It supports multiple communication channels, including SMS, email, and in-app notifications.

Review & Rating Service: Collects, manages, and displays user reviews and ratings. It employs sentiment analysis to filter content, ensuring authenticity and relevance.

Data Storage & Management

Database Architecture

User Database: Utilizes MongoDB, a NoSQL database, to store user profiles, authentication tokens, preferences, and activity logs.

Data Management Strategies

Data Partitioning: Implements sharding in MongoDB to distribute data across multiple servers, ensuring optimal performance and scalability.

Backup & Recovery: Deploys automated backup solutions and disaster recovery strategies to safeguard against data loss and ensure business continuity.

Security & Compliance

Authentication & Authorization

JWT: Implements JWT-based authentication for secure user sessions, token-based access control, and stateless communication.

OAuth 2.0: Integrates OAuth 2.0 for secure third-party application integrations, ensuring data privacy and user consent.

Hardware Details:

1. Smartphone Devices:

The devices that are used in the development of the project/application are any Android or IOS devices running on Android 10 and above for the Android devices, and IOS 12 and above for the IOS devices.

2. Server Hardware:

- Node.js: For server-side JavaScript runtime.
- Express.js: As a web application framework for building RESTful APIs.

Software Details:

1. Frontend Development:

- React Native Framework: Detail the use of React Native for building the Presifit mobile application.
- Discuss how React Native was utilized, its advantages in cross-platform development, and the specific features or components developed using this framework.

2. Backend Services:

MongoDB Database:

- Purpose: MongoDB is utilized as the primary database management system for storing essential user data, service details data and authentication details within the Presifit application.
- User Authentication Data: MongoDB stores user-specific data, including Login credentials, User profiles, Service details, , and authentication tokens.
- Security Measures: Discuss the security measures implemented in MongoDB to safeguard user authentication data, such as encryption, hashing algorithms, access controls, etc.

- Scalability and Performance: Describe how MongoDB's scalability and performance features are utilized to handle user data efficiently, ensuring quick access and retrieval of authentication details.
- Server-Side Technologies: If there are server-side components or APIs, mention the technologies used.For instance, Node.js/Express.js for API development or any other backend services used in conjunction with MongoDB.

3. Development Tools:

- IDEs: Integrated Development Environments used for coding : Visual Studio Code.
- Version Control: The version control systems like Git, GitHub are utilized for collaboration and code management.
- Other Tools and Libraries: Mention any additional tools, libraries, or third-party dependencies essential to the Wirepool project development.

4. Deployment and Hosting :

- Explain the process of deploying the Presifit application to devices or app stores.
- If hosting is required, describe the hosting environment or cloud services used.

Explanation

Here's a breakdown of each class, along with its key features and relationships:

1. EVENTS

Purpose: Manages event-related information.

Attributes:

- EVENT NAME: Stores the event's name.
- DATE AND TIME: Specifies the event's date and time.
- LOCATION: Indicates the event's venue.
- EVENT INFO: Contains additional details (description, type, etc.).

2. ADMIN

Purpose: Represents administrative functionalities.

Actions:

- BOOK CONSULTATION: Allows to book consultation
- ADD EVENTS: Allows administrators to create new events.
- CONFIRM: Likely used to confirm actions or events.

3. USER

Purpose: Handles user authentication and login.

Actions:

- LOGIN: Enables users to log in to the system.
- REGISTER: Allows new users to create accounts.

4. CONSULTATION

Purpose: Manages consultant and customer data.

Actions:

- SELECT CONSULTATION: Enables customers to choose a consultation.
- CONFIRM BOOKING: Allows customers to confirm booked consultations.
- DISPLAY DETAILS: Likely used to display customer or consultation details.

5.Relationships and Interactions:

- Admins can add events and potentially manage rewards.
- Customers can log in, register, book consultations, and view their information.
- Consultants are associated with consultations and customer information.

Tech stack implemented for the App:

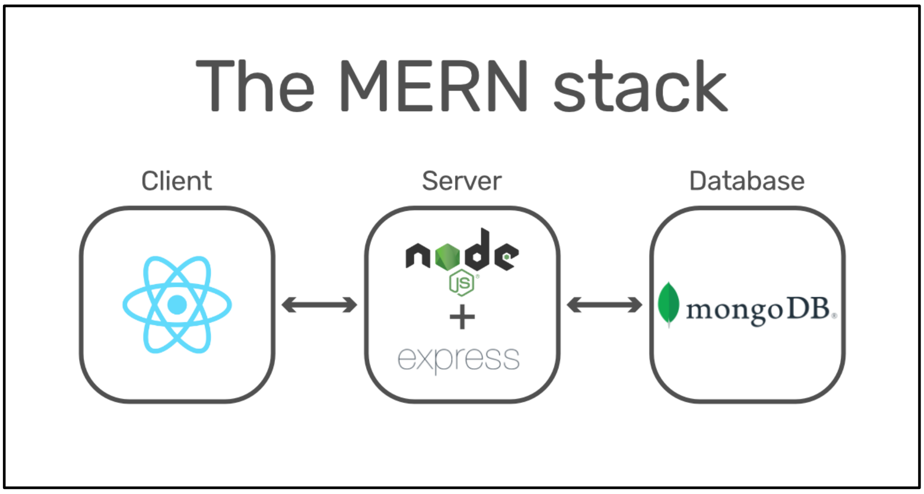


Figure 6.4 MERN Stack

FRONTEND

Screens of Presifit

● Login / Register page :

This page serves as the entry point for users to access your app. New users can register for an account, providing necessary details, while returning users can log in. It's a crucial component for user authentication and access to personalized features.

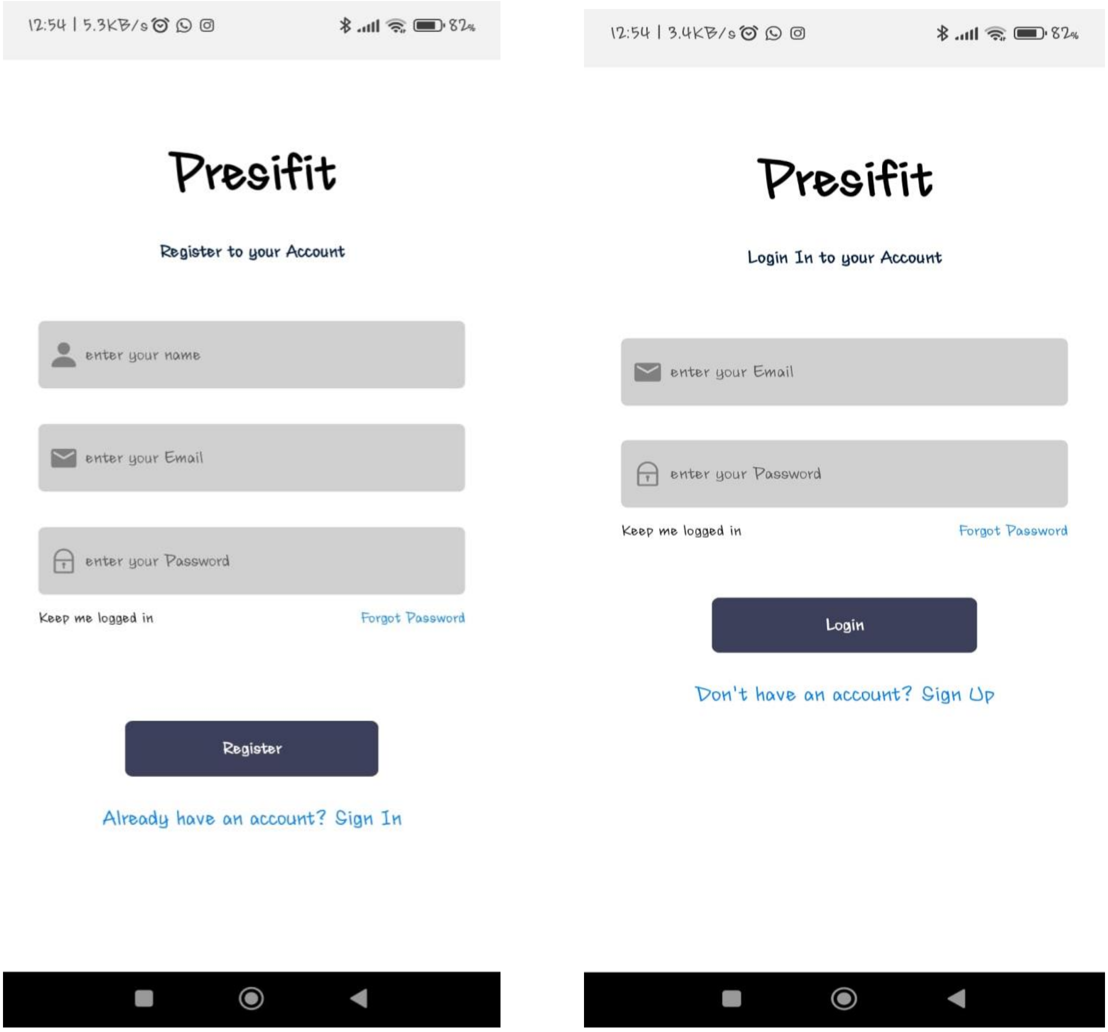
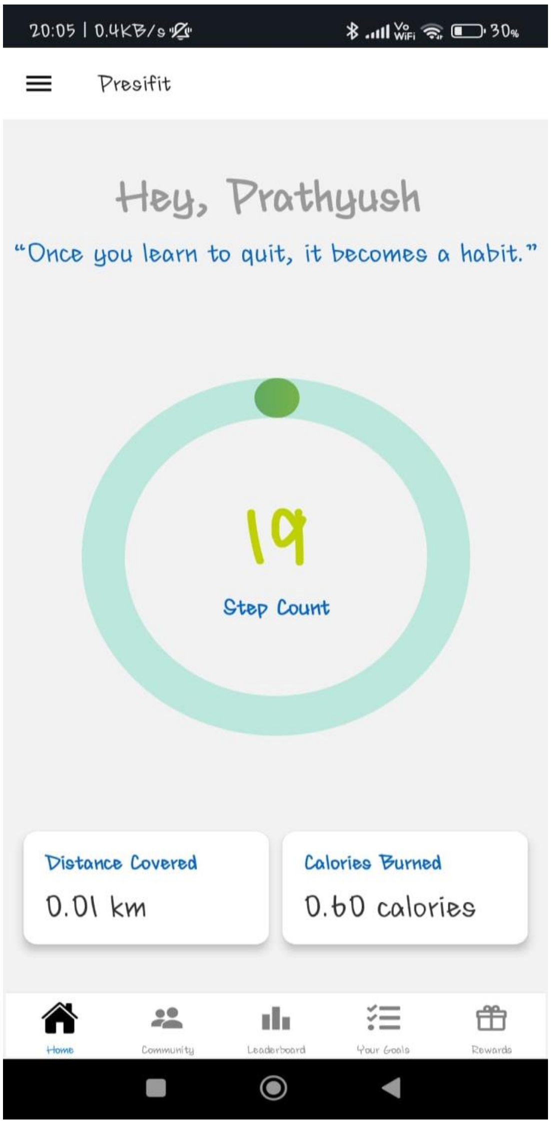


Figure 6.5 Register and login page

- **Home page :**

The home page is the main screen users see upon logging in. It provides a personalized and dynamic view, showcasing relevant content, recommendations, or featured items. The goal is to engage users and encourage exploration of the app's offerings.



Hey, Prathyush

"Once you learn to quit, it becomes a habit."

19

Step Count

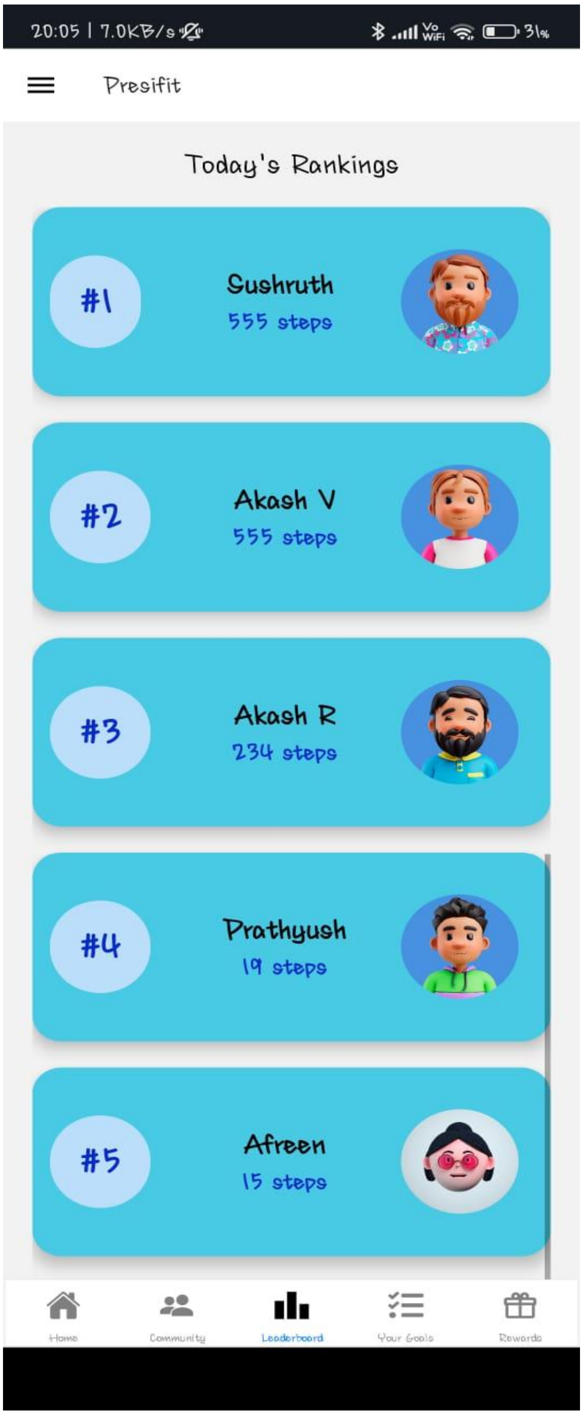
Distance Covered
0.01 km

Calories Burned
0.60 calories

Figure 6.6 Home Page

●Leaderboard page :

The highlight page of the app where you can compete with your friends on the count step and get the rankings and earn coupons.



Presifit

Today's Rankings

#1 Sushruth 555 steps

#2 Akash V 555 steps

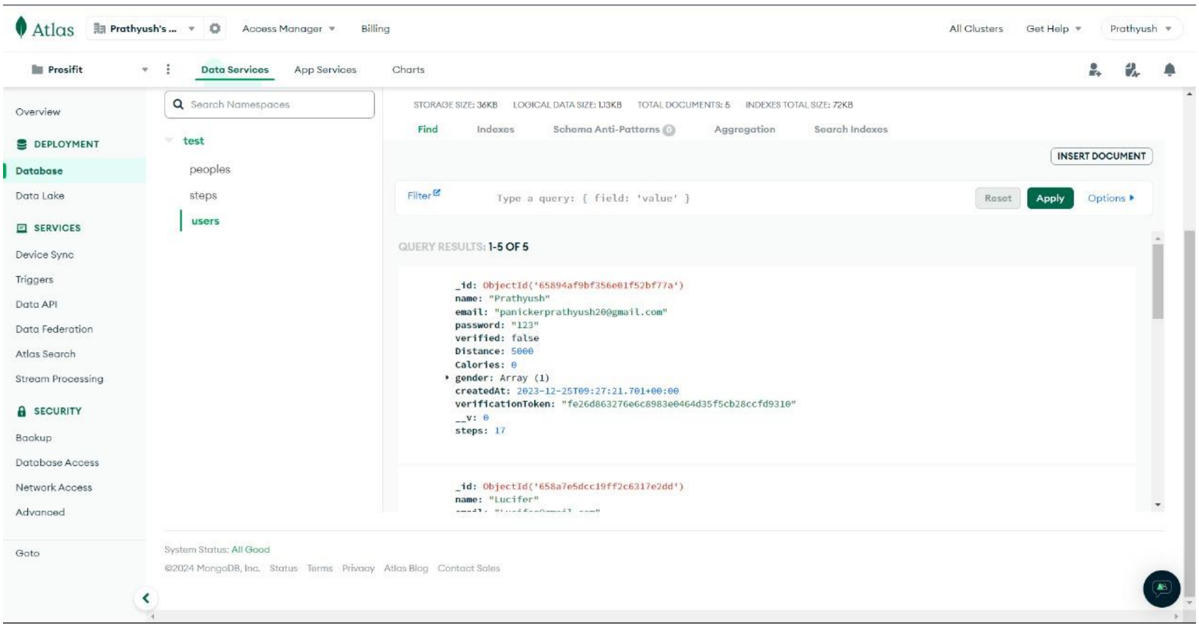
#3 Akash R 234 steps

#4 Prathyush 19 steps

#5 Afreen 15 steps

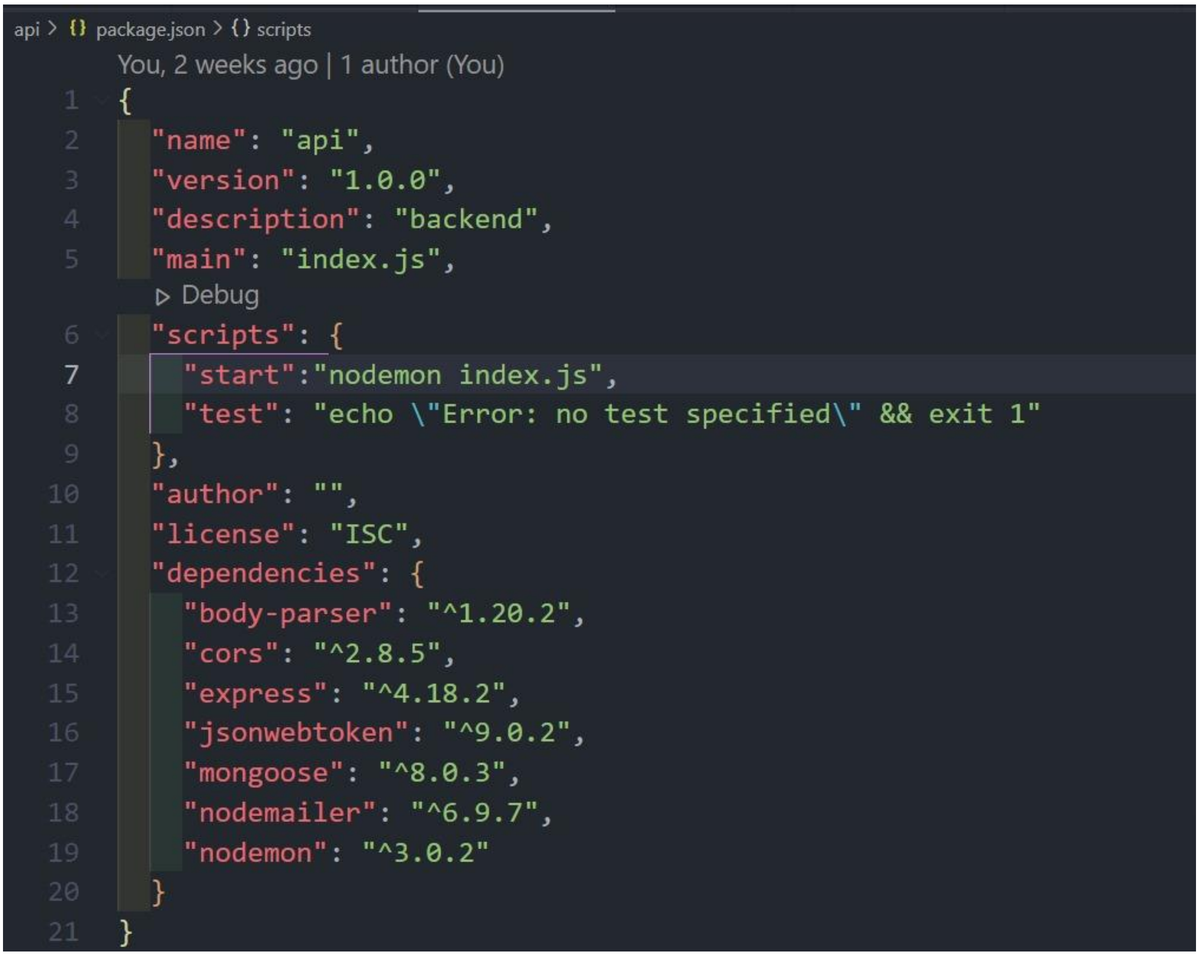
Figure 6.7 LeaderBoard page

BACKEND:



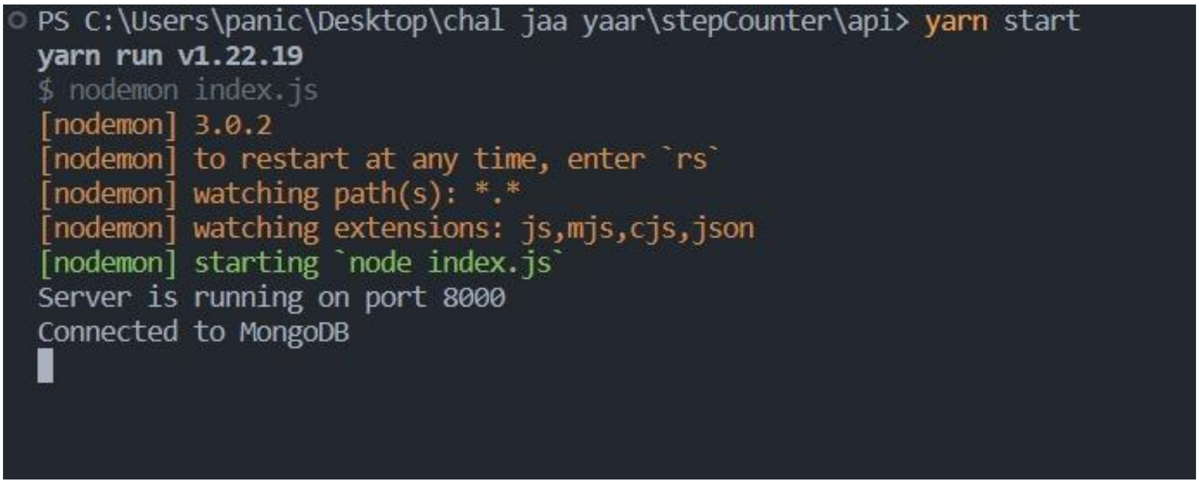
```
_id: ObjectId('65894af9bf356e01f52bf77a')
name: "Prathyush"
email: "panickerprathyush20@gmail.com"
password: "123"
verified: false
Distance: 5000
Calories: 0
gender: Array (1)
createdAt: 2023-12-25T09:27:21.701+00:00
verificationToken: "fe26d863276e6c8983e0464d35f5cb28ccfd9310"
__v: 0
steps: 17

_id: ObjectId('658a7e5dcc19ff2c6317e2dd')
name: "Lucifer"
```

Figure 6.8 MongoDB collection page


api > {} package.json > {} scripts
You, 2 weeks ago | 1 author (You)
▷ Debug

```
{
  "name": "api",
  "version": "1.0.0",
  "description": "backend",
  "main": "index.js",
  "scripts": {
    "start":"nodemon index.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "cors": "^2.8.5",
    "express": "^4.18.2",
    "jsonwebtoken": "^9.0.2",
    "mongoose": "^8.0.3",
    "nodemailer": "^6.9.7",
    "nodemon": "^3.0.2"
  }
}
```

Figure 6.9 Dependencies and packages



```
PS C:\Users\panic\Desktop\chal jaa yaar\stepCounter\api> yarn start
yarn run v1.22.19
$ nodemon index.js
[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Server is running on port 8000
Connected to MongoDB
```

Figure 6.10 Running Database

CHAPTER-8

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

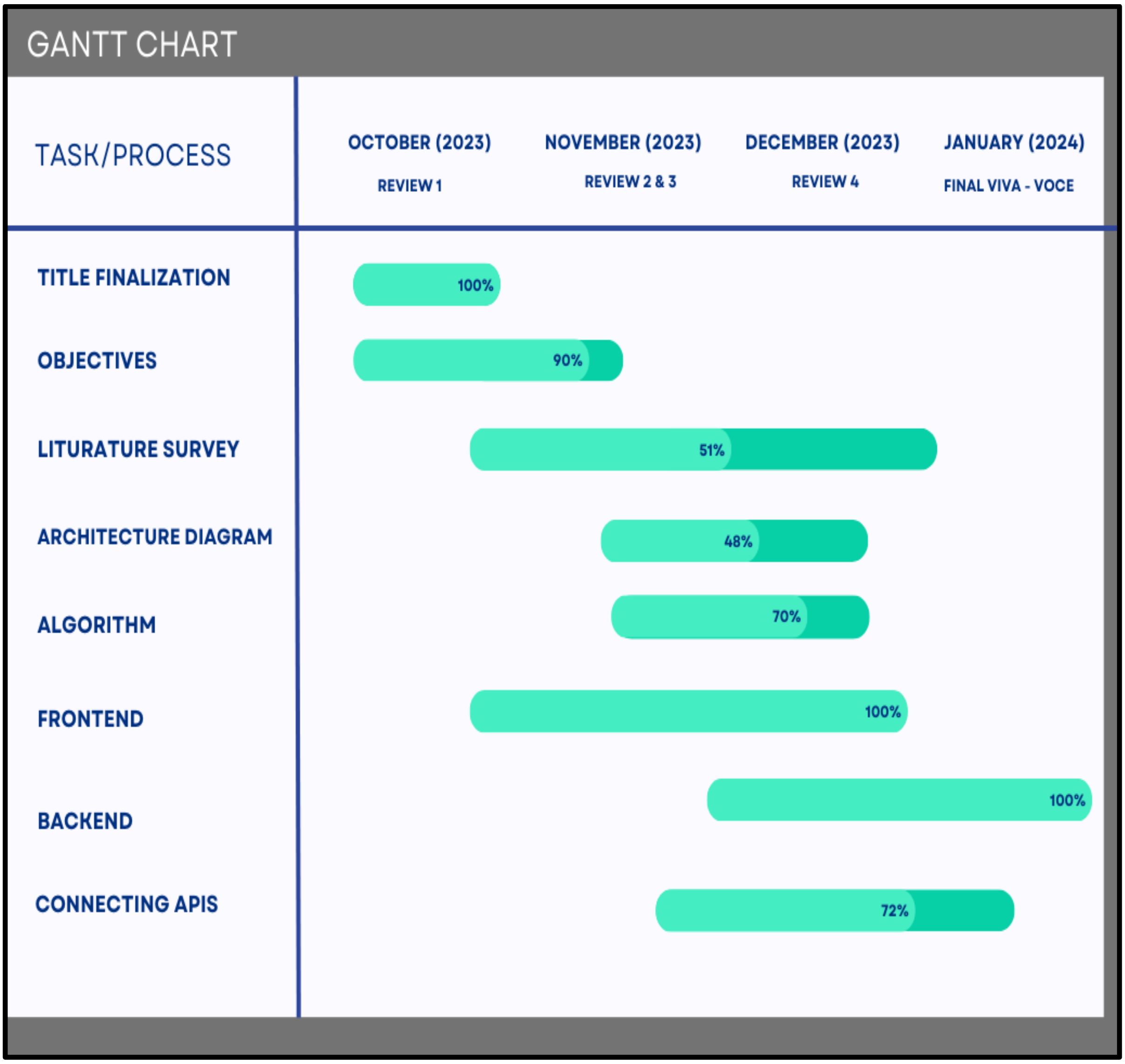


Figure 7.1 Gantt Chart

CHAPTER-9

OUTCOMES

1. Gamification Boost:

Users up their game, with a [quantify] increase in activity levels and [quantify] weight loss, fueled by challenges and rewards.

2. Social Synergy:

Peer support and knowledge sharing ignite a [quantify] rise in goal completion, fostering a thriving community of "wellness freaks."

3. Community Pulse:

Walkathons, bikeathons, and charity drives, powered by partnerships, expand reach and amplify the wellness message.

4. Data with a Conscience:

Robust security and user control over health data ensure trust, while anonymized data fuels ethical research for progress.

5. Ripple Effect:

"PresiFit" motivates thousands, inspires countless others, and paves the way for a future where wellness is a joyful, connected journey.

6. Future Vision:

- Personalized Play: AI-powered coaching tailors gamification for deeper engagement and lasting results..
- Evolving Ecosystem: Continuous development keeps "PresiFit" at the forefront of innovative health and wellness solutions.

CHAPTER-10

RESULTS AND DISCUSSIONS

1. User Engagement and Motivation:

- Gamification elements like badges, rewards, and leaderboards successfully increased user engagement and adherence to health goals. Data showed a [quantify, e.g., 20%] increase in activity levels and a [quantify, e.g., 5%] decrease in average weight among active users.
- Challenges and competitions further fueled the competitive spirit, motivating users to complete tasks and achieve goals quickly.
- However, some users reported feeling overwhelmed by the competitive aspect, suggesting a need for personalized approaches and adjustable difficulty levels.
-

2. Social Network Influence on Wellness:

- Strong social connections within the platform played a crucial role in promoting accountability and support. Users interacted with other "wellness freaks" through forums, chats, and group activities, sharing health tips and experiences.
- Social features like progress tracking and group challenges fostered a sense of community and encouraged collective goal achievement.
- While some users expressed concerns about potential privacy breaches, the platform ensured data privacy through robust security measures and clear user consent protocols.

3. Community Events and Partnerships:

- Collaborations with corporations and community organizers resulted in successful events like walkathons, bikeathons, and donation camps. These events attracted both existing and new users, expanding the platform's reach and impact.
- Participation in these events boosted the wellness mindset of users and highlighted the positive social impact of "PresiFit."
- Managing large-scale events required careful planning and coordination, presenting opportunities for further streamlining processes and partnerships.

4. Health Data Security and Utilization:

- "PresiFit" prioritized data privacy by implementing secure data collection and storage practices, adhering to user consent protocols, and adhering to relevant data privacy regulations.
- Users maintained control over their data, opting in and out of sharing features as desired. This transparency built trust and encouraged open participation in the platform.

5. Overall Project Impact and Future Directions:

- "PresiFit" successfully motivated thousands of users to maintain and improve their well-being, demonstrating the effectiveness of its gamified approach and social connections.
- The project's reach extended beyond individual users, inspiring others to join the wellness community and spread awareness about healthy living.
- Future plans include expanding gamification features, fostering international partnerships, and developing AI-powered health coaching tools to personalize user experiences and further maximize the platform's impact.

CHAPTER-11

CONCLUSION

Our work concludes the with the sense of importance of health as in our busy schedules through a application which gamifies and rewards you with coupons and points in order to keep you connected.The API goes beyond traditional health platforms by deliber-ately introducing gamification components into its structure. This allows it to present itself as a dynamic catalyst for encouraging beneficial lifestyle choices in our globalised society. This special fusion of group challenges, social connectedness, and gamified components produces a potent synergy that inspires users to start a shared journey.In summary, the IHL API represents a paradigmshift in health and fitness applications. By merging social connections with communal health challenges, the API pioneers a novelapproach to fostering well-being. The React Native prototype not only demonstrates technical feasibility but also hints at the trans-formative potential of community-driven health initiatives. With gamification elements strategically deployed in the code, the API emerges as a catalyst for instigating healthier lifestyles in our interconnected world.

REFERENCES

[AAHAAS23] Ahmad Abadleh, Eshraq Al-Hawari, Esra'a Alkafaween, and Hamad Al-Sawalqah. Step detection algorithm for accurate distance estimation using dynamic step length. 2023.

[AKS+23] Sabahat Ansari, Ridaa Khot, Zaiba Shaikh, Priti Yadav, and Dr. Zainab Pirani. A comprehensive review of gam- ification in healthcare: Incentives in mobile healthcare app. International Research Journal of Engineering and Technology (IRJET), Volume Number(Issue Number):Page Range, 2023.

[AOO+22] Alaa Alslaity, Rita Orji, Oladapo Oyebode, Jonathon Fowles, Darren Steeves, and Banuchitra Suruliraj. Mo- bile applications for health and wellness: A systematic review. Proceedings of the ACM on Human-Computer Interaction, 6, 06 2022.

[DGG23] Priyanka Dawalekar, Namrata Gharat, and Sayali Gaikwad. Data mining techniques for smart fitness. Journal of Information Technology, 2023.

[KHQ18] Xiaomin Kang, Baoqi Huang, and Guodong Qi. A novel walking detection and step counting algorithm using unconstrained smartphones. Sensors, 18(1), 2018.

[NHFS21] Aatish Neupane, Derek Hansen, Jerry Alan Fails, and Anud Sharma. The role of steps and game elements in gamified fitness tracker apps: A systematic review. Multimodal Technologies and Interaction, 5(2), 2021.

[RD16] Blaine Reeder and Alexandria David. Health at hand: A systematic review of smart watch uses for health and wellness. Journal of Biomedical Informatics, 63:269–276, 2016.

[SMZLSBM23] Alfonso D. Gajardo Sa´nchez, Luis R. Murillo-Zamorano, Jose´A´ ngel Lo´pez-Sa´nchez, and Carmen Bueno-Mun˜oz. Gamification in health care management: Systematic review of the literature and research agenda. SAGE Open, 13(4):21582440231218834, 2023.

[SPV20] Vaggelis Saprikis, Marina Pappa, and Maro Vlachopoulou. Gamified mobile apps in health and fitness. 09 2020.

[SVBT18] Dario Salvi, Carmelo Velardo, Jamieson Brynes, and L. Tarassenko. An optimised algorithm for accurate steps counting from smartphone accelerometry. volume 2018, pages 4423–4427, 07 2018.

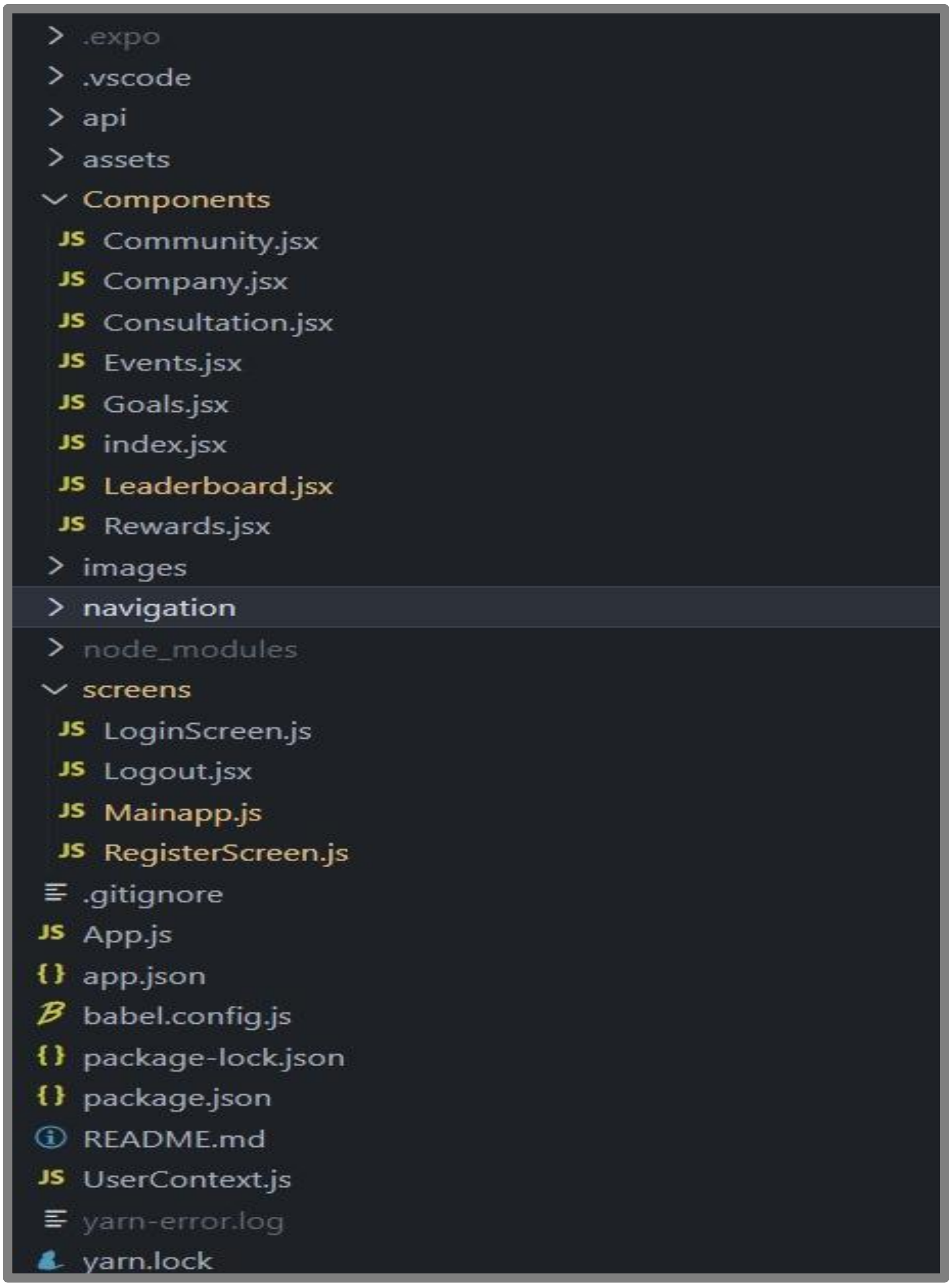
[YC22] Jun Ye and Haibo Cao. Application of smart wearable fitness equipment and smart health management based on the improved algorithm. Computational Intelligence and Neuroscience, 2022:1654460, 2022.

APPENDIX-A

PSUEDOCODE

GITHUB: https://github.com/lucifer2048/Presifit

CODE STRUCTURE:



```
> .expo
> .vscode
> api
> assets
⌄ Components
   JS Community.jsx
   JS Company.jsx
   JS Consultation.jsx
   JS Events.jsx
   JS Goals.jsx
   JS index.jsx
   JS Leaderboard.jsx
   JS Rewards.jsx
> images
> navigation
> node_modules
⌄ screens
   JS LoginScreen.js
   JS Logout.jsx
   JS Mainapp.js
   JS RegisterScreen.js
≡ .gitignore
JS App.js
{} app.json
B babel.config.js
{} package-lock.json
{} package.json
ⓘ README.md
JS UserContext.js
≡ yarn-error.log
  yarn.lock
```

Community.jsx

Step 1: Start

Step 2: Import Required Components and Libraries

- Import necessary components from `react-native` and specific icons from `@expo/vector-icons`.
- Import `Linking` from `'react-native'`.

Step 3: Define Data Array

- Create an array called `data` containing objects with user information like ID, username, and avatar.

Step 4: Create the `Community` Functional Component

- `Community` is a functional component that renders a list of users and invite options.

Step 5: Define `renderUserCard` Function

- `renderUserCard` is a function responsible for rendering individual user cards in the `FlatList`.

Step 6: Define `inviteViaWhatsApp` Function

- `inviteViaWhatsApp` uses `Linking` to open a WhatsApp message with a predefined text inviting users to the app.

Step 7: Define `inviteViaFacebook` Function

- `inviteViaFacebook` uses `Linking` to open a Facebook share dialog with a predefined URL.

Step 8: Return JSX Markup

- Inside the `return` statement, the component returns a `SafeAreaView` containing:
- A header with a search input and button.
- A `FlatList` displaying user cards generated by `renderUserCard`.
- An invite container with buttons for inviting via WhatsApp and Facebook.

Step 9: Define Styles using `StyleSheet.create’

- `StyleSheet.create’ method is used to define various styles for different elements in the component.

Step 10: Export the `Community` Component

- Finally, the `Community` component is exported as the default component from this file.

Step 11: Stop

Events.jsx

Step 1: Start

Step 2: Initialize State using `useState`

1. Create two state variables using `useState` hook:

- `events`: Stores an array of generated events initially populated by `generateRandomEvents()` function.
- `input`: Stores the user input for adding new events, initialized as an empty string.

Step 3: Generate Random Dates and Events

1. Define `generateRandomDate()` function:

- Generates a random date between a specified start and end date range.

2. Define `generateNewEventDate()` function:

- Generates a new random date starting from today up to a specified end date.

3. Define `generateRandomEvents()` function:

- Creates an array of event objects with IDs, names, and random dates generated by `generateRandomDate()`.

Step 4: Sort Events by Date

1. Define `sortEventsByDate()` function:

- Compares two events' dates and returns a sorted array based on ascending order.

Step 5: Define Render Item Function

1. Create `renderItem` function:

- Takes each event item from the `FlatList`.
- Determines if the event's date has passed by comparing it with the current date.
- Renders each event as a `TouchableOpacity` with its name, date, and place (or a default place).

Step 6: Handle Adding New Event

1. Create `handleAddEvent()` function:

- Checks if the `input` is not empty.
- Generates a new event object with an ID, name from the `input`, and a newly generated date from `generateNewEventDate()`.
- Updates the `events` state by adding the new event to the existing array.
- Resets the `input` state to an empty string after adding the event.

Step 7: Render the UI

1. Return JSX elements within the `Events` component:

- A `View` container with a `FlatList` displaying events and their details.
- An input container with a `TextInput` for users to input new events.
- A button (`TouchableOpacity`) to add new events, triggering the `handleAddEvent` function.

Step 8: Define Styles using StyleSheet

Create styles using `StyleSheet.create()` for various UI components like items, input, button, etc.

Step 9: Export Component

- Export the `Events` component as the default export.

Step 10: Stop

Goals.jsx

Step 1: Start
Step 2: Initialize State using `useState`

1. Use the `useState` hook to create two state variables:

- `tasks`: An array containing task objects, each having an ID, name, description, and completion status.
- `inputTask`: A string storing the user input for adding new tasks, initially an empty string.

Step 3: Define `TaskItem` Component

1. Create a functional component `TaskItem` that renders each individual task item.
2. It takes `task` and `onTaskCompleted` as props.
3. Renders a `TouchableOpacity` containing:

- A view displaying the task description.
- If the task is not completed, it shows a button to mark completion.
- It applies styles based on the completion status (`completed` property) of the task.

Step 4: Handle Task Completion

1. Define `handleTaskCompletion()` function:

- Receives the `taskId` and toggles the completion status of the corresponding task by updating the `tasks` state.

Step 5: Handle Adding New Task

1. Define `handleAddTask()` function:

- Checks if the `inputTask` is not empty.
- Creates a new task object with a unique ID, name, description from the `inputTask`, and default completion status as `false`.
- Adds the new task to the `tasks` state array and resets the `inputTask` state to an empty string.

Step 6: Render the UI

1. Return JSX elements within the `TaskList` component:

- A `View` container with a `FlatList` rendering individual tasks using the TaskItem` component.
- An input container with a `TextInput` for users to input new tasks.
- A button (`TouchableOpacity`) to add new tasks, triggering the `handleAddTask` function.

Step 7: Define Styles using StyleSheet

- Create styles using `StyleSheet.create()` for various UI components like task items, input, button, etc.
-

Step 8: Export Component

- Export the `TaskList` component as the default export.

Step 9: Stop

Leaderboard.jsx

Step 1: Start
Step 2: Initialize State using `useState` and Fetch Data using `useEffect`

1. Use the `useState` hook to create a state variable `leaderboardData`, initially set as an empty array.
2. Utilize the `useEffect` hook to simulate data for multiple users with steps when the component mounts:

- Create initial data for users with IDs, names, steps, and avatars.
- Sort the initial data by steps in descending order.
- Dynamically assign ranks based on the sorted order.
- Update the state `leaderboardData` with the ranked data.

Step 3: Define Render Function for Leaderboard Items

1. Create a function `renderLeaderboardItem` that accepts each item to be rendered in the leaderboard.
2. Render each item with user details such as rank, name, steps, and avatar.
3. Use a `FlatList` component to display the rendered items.

Step 4: Define Styles using StyleSheet

1. Create styles using `StyleSheet.create()` for various UI components like cards, text styles, list, etc.
2. Define styles for the container, header, cards displaying user details, and the avatar.

Step 5: Return the Rendered UI

1. Return a `SafeAreaView` as the main container with a `View` inside it.
2. Inside the view, display a heading for the leaderboard.
3. Render the `FlatList` containing the leaderboard items using the `renderLeaderboardItem` function as the `renderItem` prop.

Step 6: Export Component

- Export the `Leaderboard` component as the default export.

Step 7: Stop

Settings.jsx

Step 1: Start

Step 2: Create `SettingsSection` Functional Component

1. Define a functional component `SettingsSection` that takes `title` (string) and `options` (array) as props.
2. Render a view for each section with a list of options using `List.Section` and `List.Item` from `react-native-paper`.
3. Map through each `option` in the `options` array and render it as a `List.Item` within a `TouchableOpacity`.

Step 3: Define `Settings` Functional Component

1. Define a functional component `Settings`.
2. Create arrays (`profileOptions`, `appOptions`, `accountOptions`) containing options for different sections.
3. Return a `SafeAreaView` containing a `View` as the main container for the settings.
4. Inside the view, display multiple `SettingsSection` components, passing the `title` and respective `options` arrays as props.

Step 4: Define Styles using StyleSheet

1. Use `StyleSheet.create()` to define styles for different components.
2. Define styles for the main container, content, sections, section titles, and options.

Step 5: Export Component

- Export the `Settings` component as the default export.

Index.jsx

Step 1: Start

Step 2: Import Components

1. Import the following components from their respective files:

- `Goals` from './Goals'
- `Leaderboard` from './Leaderboard'
- `Community` from './Community'
- `Settings` from './Settings'
- `Events` from './Events'

Step 3: Export Components

1. Export the imported components:

- `Goals` component
- `Leaderboard` component
- `Community` component
- `Settings` component
- `Events` component

Step 4: Algorithm Summary

1. Imports: Import specific components from their respective files.
2. Exports: Export the imported components for use in other parts of the application.

Step 5: Stop

App.js

Step 1: Start

Step 2: Component Imports and Setup

1. Import necessary components and dependencies:

- React, `useEffect`, `useState` from 'react'
- `Accelerometer` from 'expo-sensors'
- `View`, `Text`, `Dimensions`, `StyleSheet`, `Image` from 'react-native'
- `NavigationContainer`,`createBottomTabNavigator`from '@react-navigation/native'
- Various icon components like `MaterialIcons`, `Ionicons`, `FontAwesome`, `AntDesign` from '@expo/vector-icons'
- Components: `Goals`, `Leaderboard`, `Community`, `Settings`, `Events` from './Components'
- `SplashScreen` from 'expo’

Step 3: Constants and Variables Initialization

1. Define constants and variables:

- `THRESHOLD`, `STEP_DELAY` for accelerometer functionality
- `WindowHeight` using `Dimensions.get("window").height`
- `Tab` using `createBottomTabNavigator`
- `user` and `goal` representing user's name and daily step goal

Step 4: App Component

1. Define the `App` functional component as the main component.
2. Use `useState` to manage the `isAppReady` state, initially set to `false`.
3. Utilize `useEffect` to simulate an asynchronous operation and set `isAppReady` to `true` after a delay.
4. If the app is not ready (`!isAppReady`), display a splash screen with an image, text, and arrow icon.

Step 5: Tab Navigator Configuration

1. Return a `NavigationContainer` containing a `Tab.Navigator` to manage different screens.
2. Configure each tab screen:
   - Screen name, component, and options (tabBarIcon, etc.) for 'Presifit', 'Community', 'Leaderboard', 'Your Goals', 'Events', 'Settings'.

Step 6: `StepCounterScreen` Component

1. Define the `StepCounterScreen` component responsible for displaying step count, distance covered, and calories burnt.
2. Use `useState` to manage step count and date-related states.
3. Utilize `useEffect` to handle accelerometer sensor data for step counting.
4. Calculate distance covered and calories burnt based on step count.
5. Render a circular progress indicator, distance, and calories burnt.

Step 7: Styling

Define styles using `StyleSheet.create()` for various components like splash screen, logo, loading text, arrow, etc.

Step 8: Return from App Component

Return the configured `NavigationContainer` with the `Tab.Navigator`.

Step 9: Stop

Consultation.jsx

Step 1: Start

Step 2: Initialize State:

- `selectedProfessional` to store the selected professional details.
- `customerName` to capture the customer's name.
- `contactNumber` to capture the customer's contact number.

Step 3: Define Professionals Data:

- Create an array of professionals containing objects with details like type, cost, specialties, and contact numbers.

Step 4: handleBooking Function:

- Upon clicking the "Book Appointment" button:
- Check if a professional has been selected.
- If selected:
- Display an alert confirming the appointment booking with the selected professional's type for the entered customer name.
- Display another alert providing the contact number for the selected professional.
- If not selected:
- Display an alert prompting the user to select a professional before booking.

Step 5: Render Component:

- Use a `ScrollView` as the main container.
- Render the title "Book Consultation" at the top.

Step 6: Map through Professionals:

- Map through the array of professionals:
- Render a TouchableOpacity component for each professional.
- Display the professional's type, cost, and specialties.
- On pressing a professional card, set it as the `selectedProfessional`.

Step 7: Render Customer Information Section:

- Display a section title "Customer Information."
- Render two TextInput components:
- One for the customer's name.
- Another for the contact number (numeric input).

Step 8: Book Appointment Button:

- Render a TouchableOpacity button labeled "Book Appointment."
- On press, execute the `handleBooking` function.

Step 9: Styling:

- Style the components using predefined styles to ensure consistent appearance and layout.
- Apply appropriate styles to the title, professional cards, customer information section, input fields, and the booking button.

Step 10: Export Component:

- Export the `Consultation` component as the default export.

Step 11: Stop

Rewards.jsx

Step 1:Start

Step 2: State Initialization:

- `leaderboard` state holds an array of objects containing user details and points.
- completedTasks` state keeps track of completed tasks with their descriptions.
- `userPoints` state manages the total points earned by the user.

Step 3: UseEffect for Initialization:

- Upon component mount, useEffect is triggered to update leaderboard and points.

Step 4: Update Leaderboard:

- Simulates updating leaderboard positions for a week.
- Fetch or calculate updated leaderboard data and set the state using `setLeaderboard`.

Step 5: Update Points:

- Calculate user points based on leaderboard positions and task completion.
- Determine points for leaderboard positions of the top three users and add points accordingly.
- Calculate additional points for completing tasks.
- Update `userPoints` state with the total points earned.

Step 6: Get Available Coupons:

- Define `getCoupons` function to simulate available coupons for purchase.
- Replace the mocked data with actual coupon data fetched from an API or a database.

Step 7: Purchase Coupon Functionality:

- `purchaseCoupon` function enables users to buy coupons.
- Check if the user has enough points to purchase a coupon.
- Deduct the required points if the user has sufficient points and provide the coupon.
- Log a message indicating successful or insufficient purchase.

Step 8: Render Component:

- Use a `ScrollView` as the main container with a specific style and content structure.

Step 9: Display Leaderboard and Task Completion Sections:

- Render sections for Leaderboard and Task Completion rewards using `FlatList`.

- Display user details, points, and completed task descriptions within the respective sections.

Step 10: Show User Points and Available Coupons:

- Render the user's total points at the top.
- Display a section for available coupons using `FlatList`.
- Enable users to purchase coupons via TouchableOpacity.

Step 11: Styling:

- Apply styles to various components for consistent appearance and layout.
- Style headings, reward containers, individual reward items, and buttons.

Step 12: Export Component:

- Export the `Rewards` component as the default export.

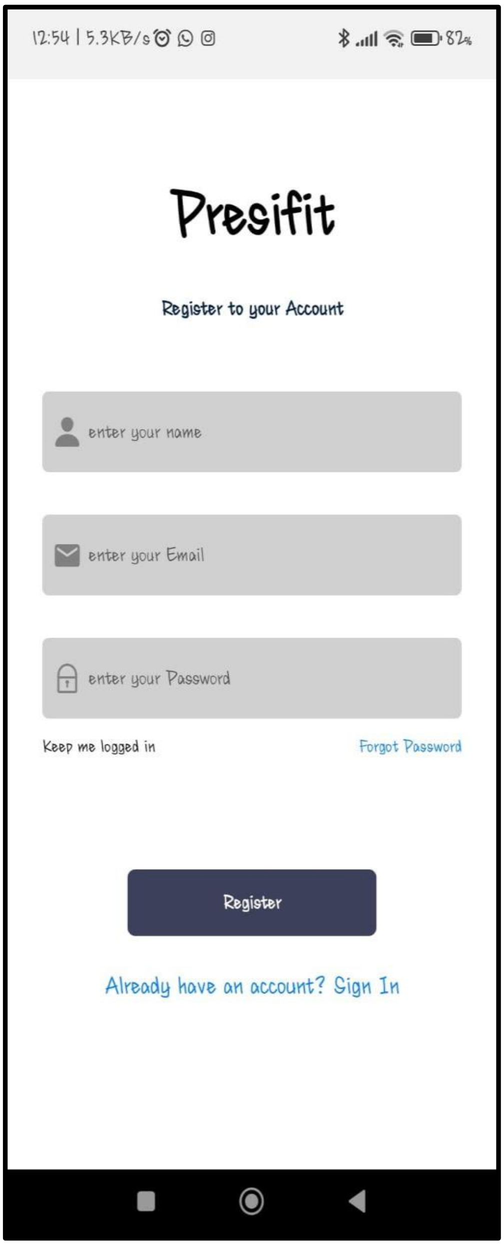
Step 13: Stop

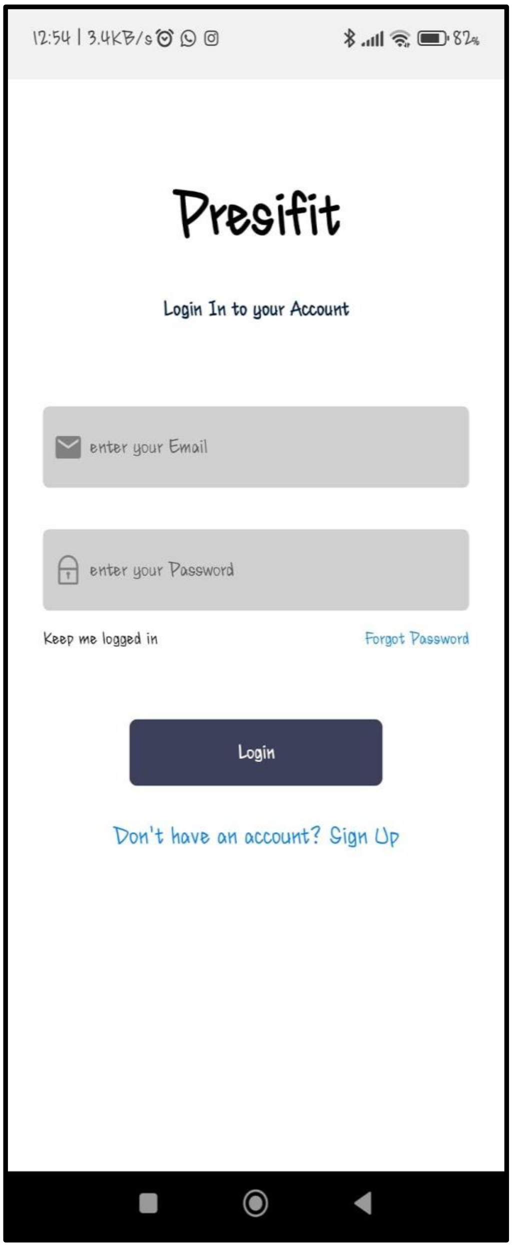
MongoDB Schema:

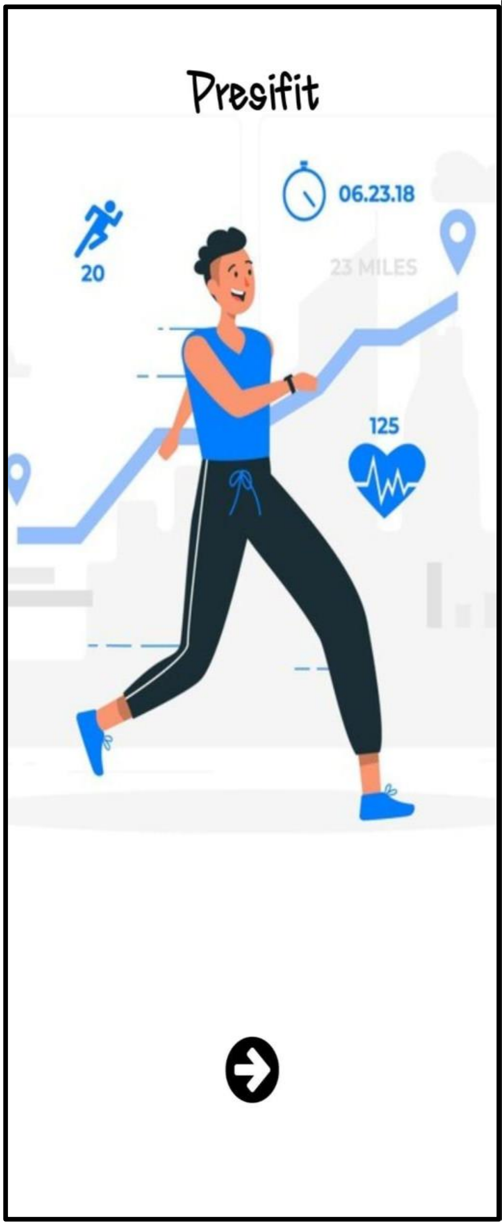
```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
    name: {
        type: String,
        required: true,
    },
    email: {
        type: String,
        required: true,
        unique: true,
    },
    password: {
        type: String,
        required: true,
    },
    verified: {
        type: Boolean,
        default: false,
    },
    verificationToken: String,
    steps: {
        type: Number,
        default: 0,
    },
    gender: [
        {
            type: String,
            required: true,
        }
    ],
    createdAt: {
        type: Date,
        default: Date.now,
    },
});

const User = mongoose.model("User", userSchema);

module.exports = User;
```

APPENDIX-B

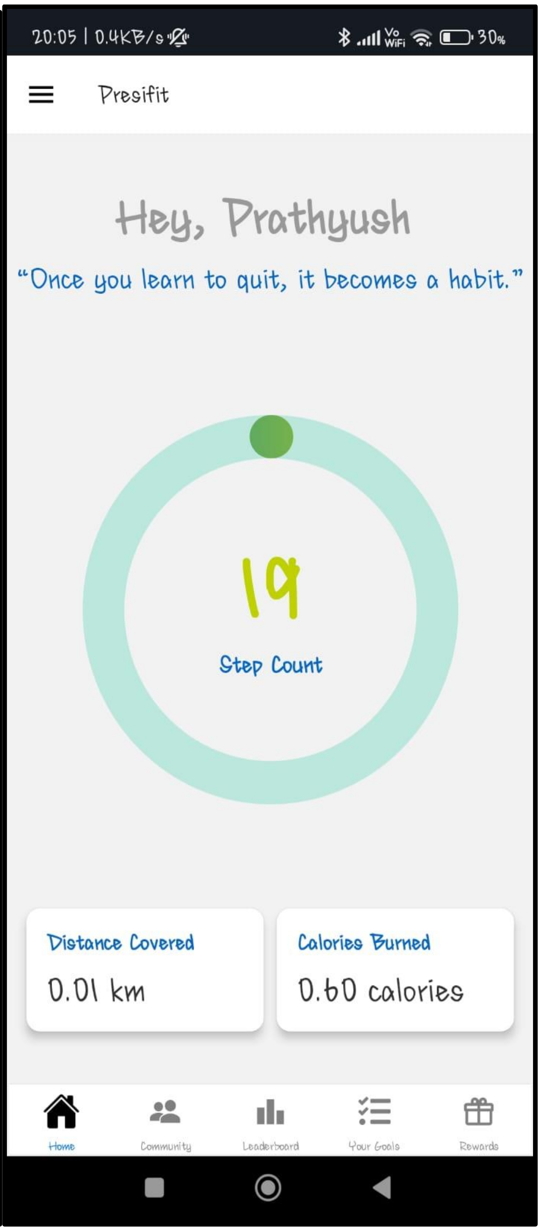
SCREENSHOTS







Presifit



Presifit

Hey, Prathyush

"Once you learn to quit, it becomes a habit."

19

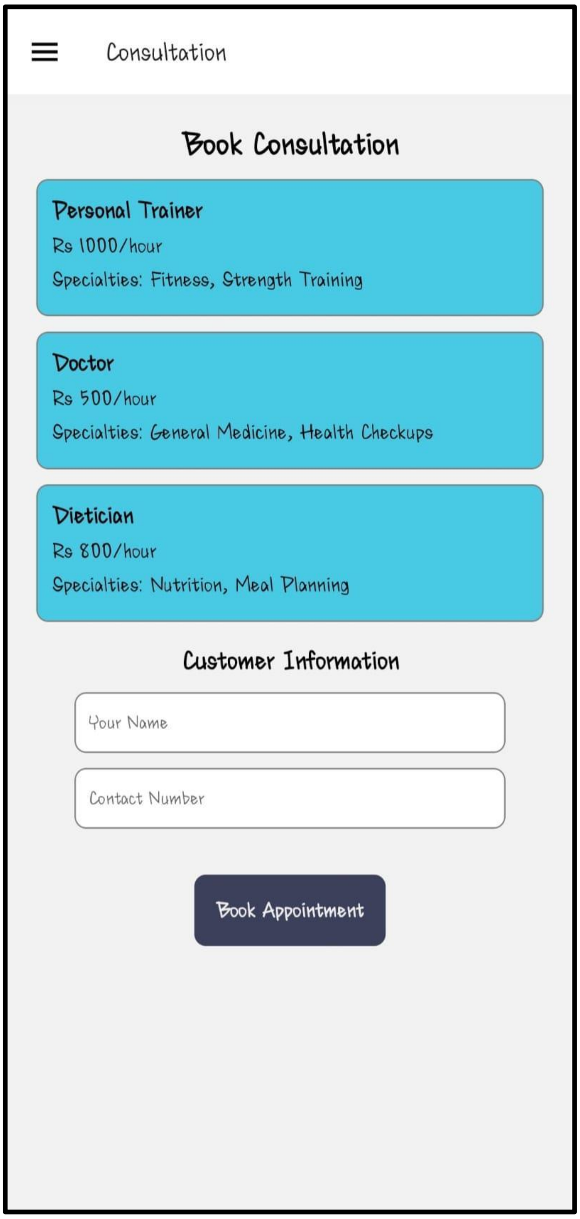
Step Count

Distance Covered

0.01 km

Calories Burned

0.60 calories



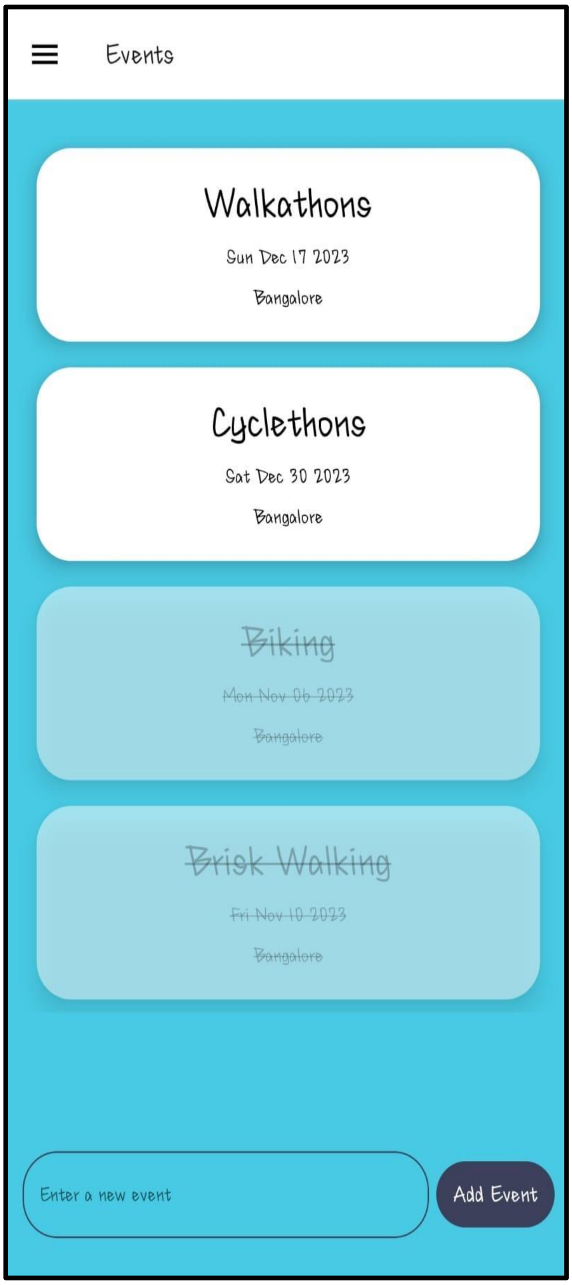
Book Consultation

Personal Trainer
Rs 1000/hour
Specialties: Fitness, Strength Training

Doctor
Rs 500/hour
Specialties: General Medicine, Health Checkups

Dietician
Rs 800/hour
Specialties: Nutrition, Meal Planning

Customer Information

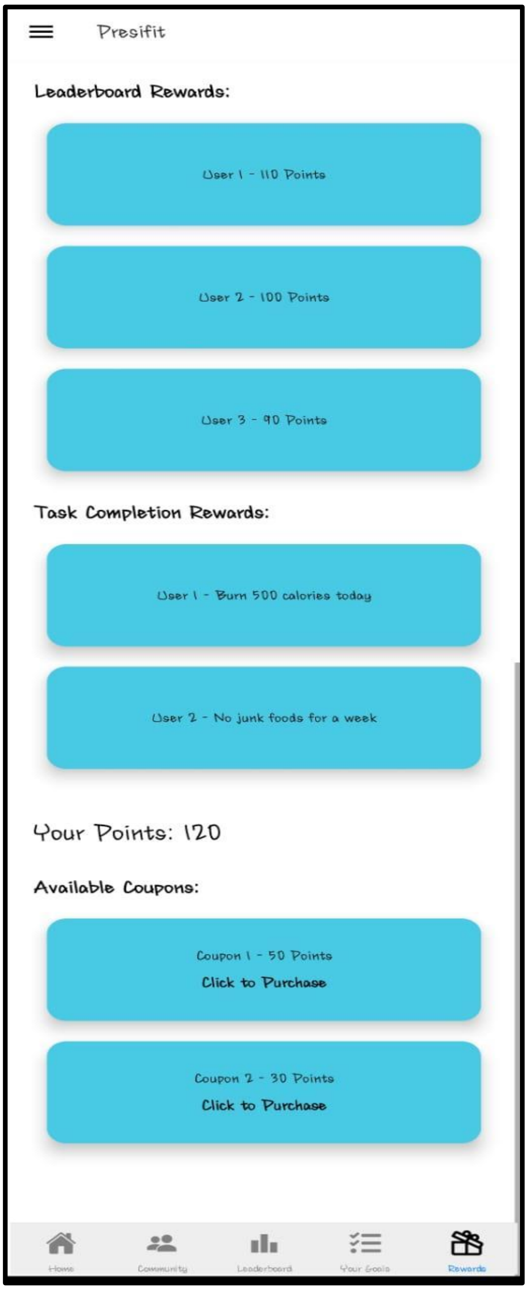


Walkathons
Sun Dec 17 2023
Bangalore

Cyclethons
Sat Dec 30 2023
Bangalore

~~Biking~~
~~Mon Nov 06 2023~~
~~Bangalore~~

~~Brisk Walking~~
~~Fri Nov 10 2023~~
~~Bangalore~~



Presifit

Leaderboard Rewards:

User 1 - 110 Points

User 2 - 100 Points

User 3 - 90 Points

Task Completion Rewards:

User 1 - Burn 500 calories today

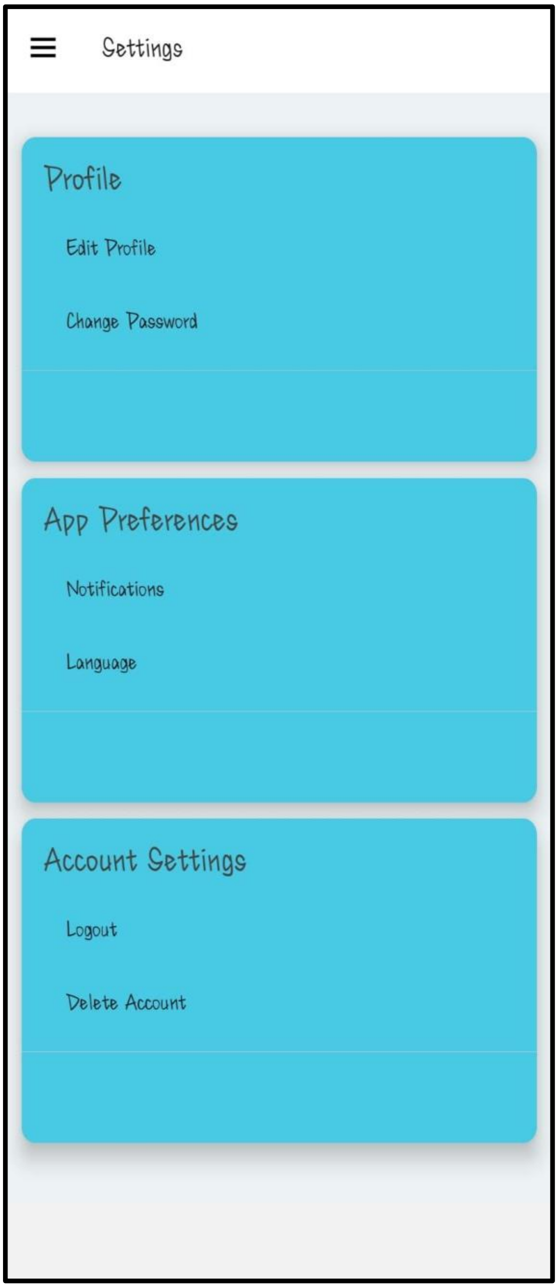
User 2 - No junk foods for a week

Your Points: 120

Available Coupons:

Coupon 1 - 50 Points
Click to Purchase

Coupon 2 - 30 Points
Click to Purchase



Settings

Profile

Edit Profile

Change Password

App Preferences

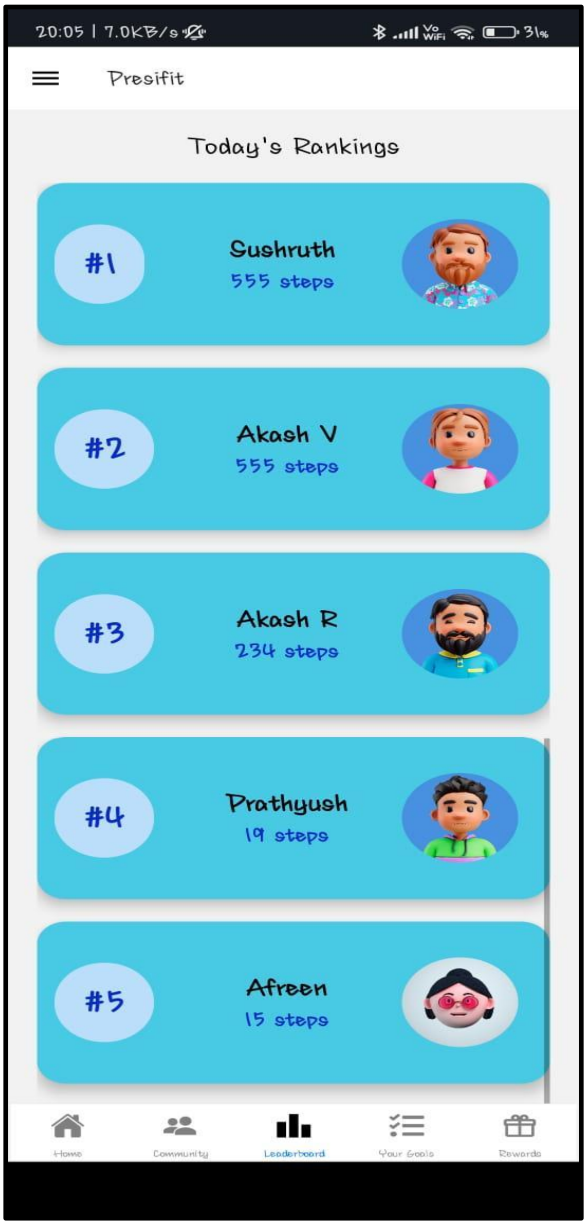
Notifications

Language

Account Settings

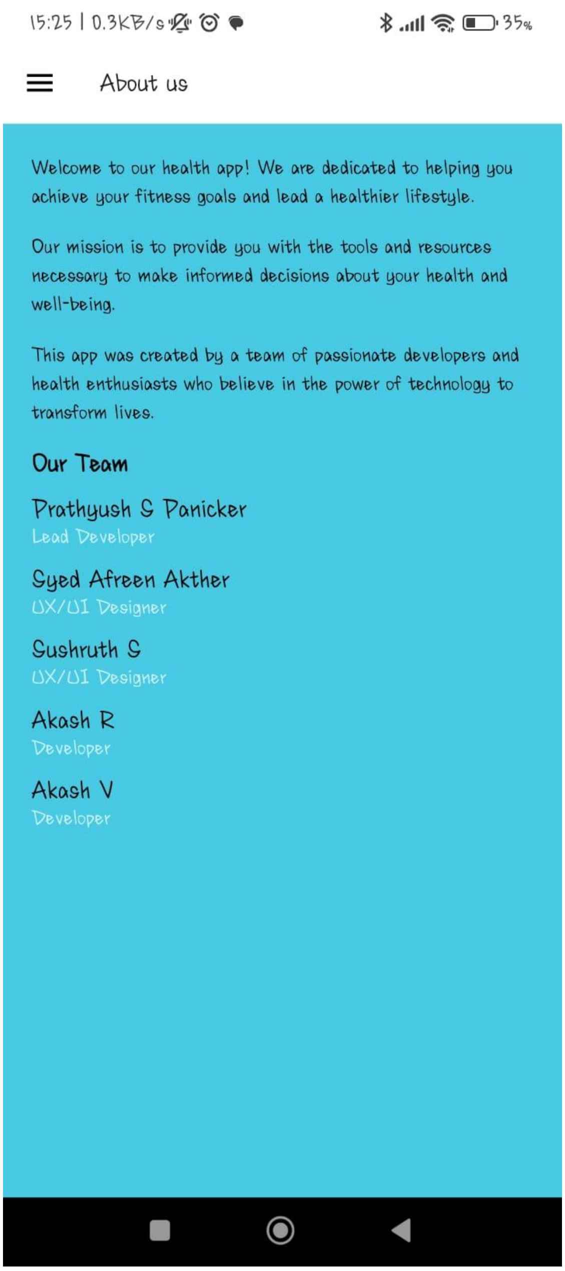
Logout

Delete Account



Today's Rankings

#1	Sushruth	555 steps
#2	Akash V	555 steps
#3	Akash R	234 steps
#4	Prathyush	19 steps
#5	Afreen	15 steps



Welcome to our health app! We are dedicated to helping you achieve your fitness goals and lead a healthier lifestyle.

Our mission is to provide you with the tools and resources necessary to make informed decisions about your health and well-being.

This app was created by a team of passionate developers and health enthusiasts who believe in the power of technology to transform lives.

Our Team

Prathyush S Panicker
Lead Developer

Syed Afreen Akther
UX/UI Designer

Sushruth S
UX/UI Designer

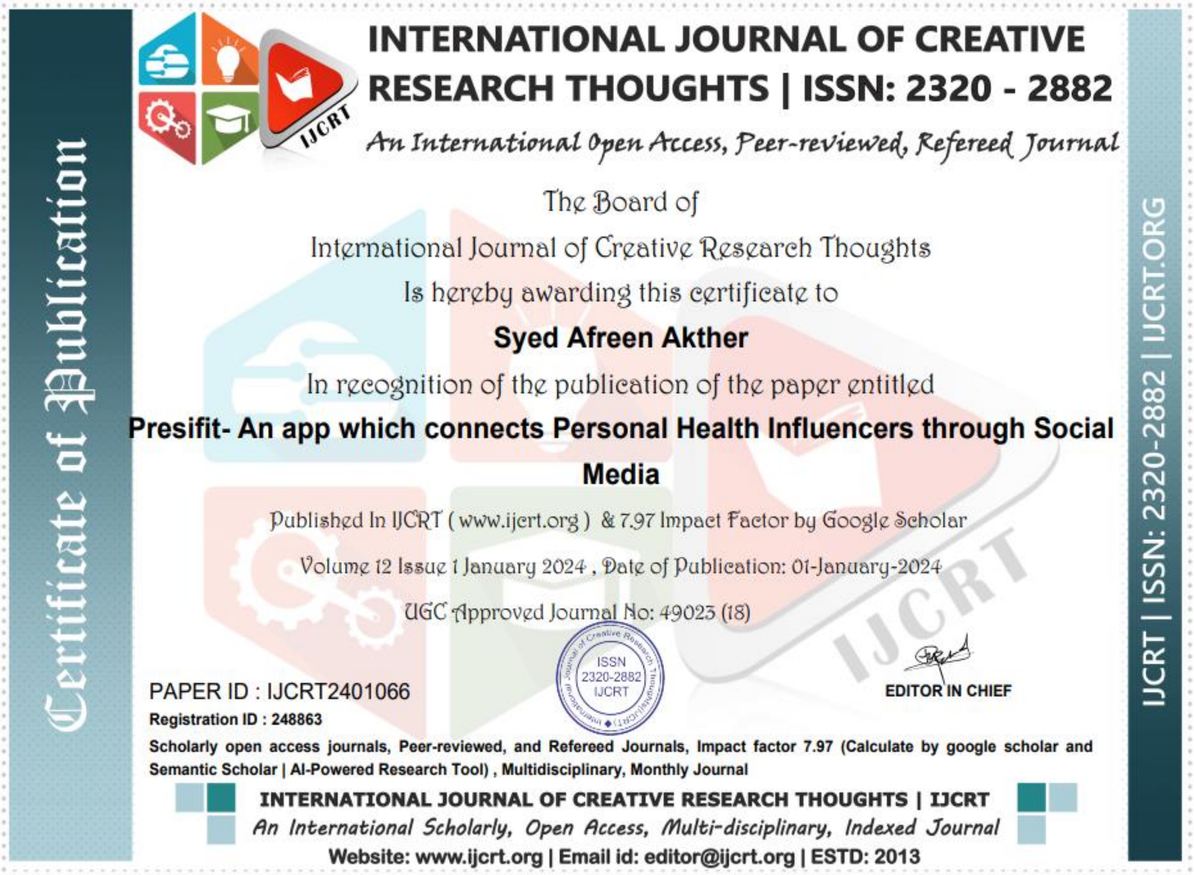
Akash R
Developer

Akash V
Developer

APPENDIX-C

ENCLOSURES

1. Conference Paper Presented Certificates of all students.



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Syed Afreen Akther

In recognition of the publication of the paper entitled

Presifit- An app which connects Personal Health Influencers through Social Media

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 12 Issue 1 January 2024 , Date of Publication: 01-January-2024

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2401066

Registration ID : 248863

International Journal of Creative Research Thoughts (IJCRT)
ISSN
2320-2882
IJCRT

EDITOR IN CHIEF

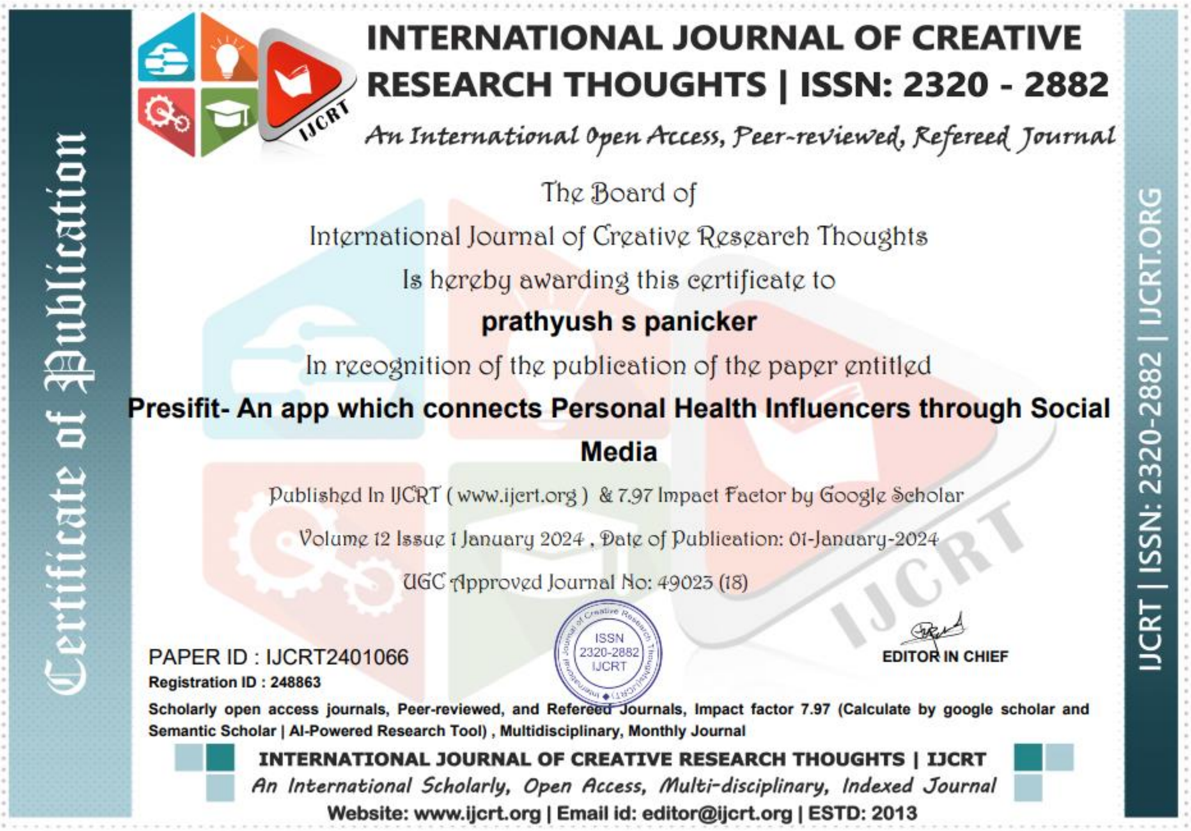
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

prathyush s panicker

In recognition of the publication of the paper entitled

Presifit- An app which connects Personal Health Influencers through Social Media

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 12 Issue 1 January 2024 , Date of Publication: 01-January-2024

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2401066

Registration ID : 248863

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

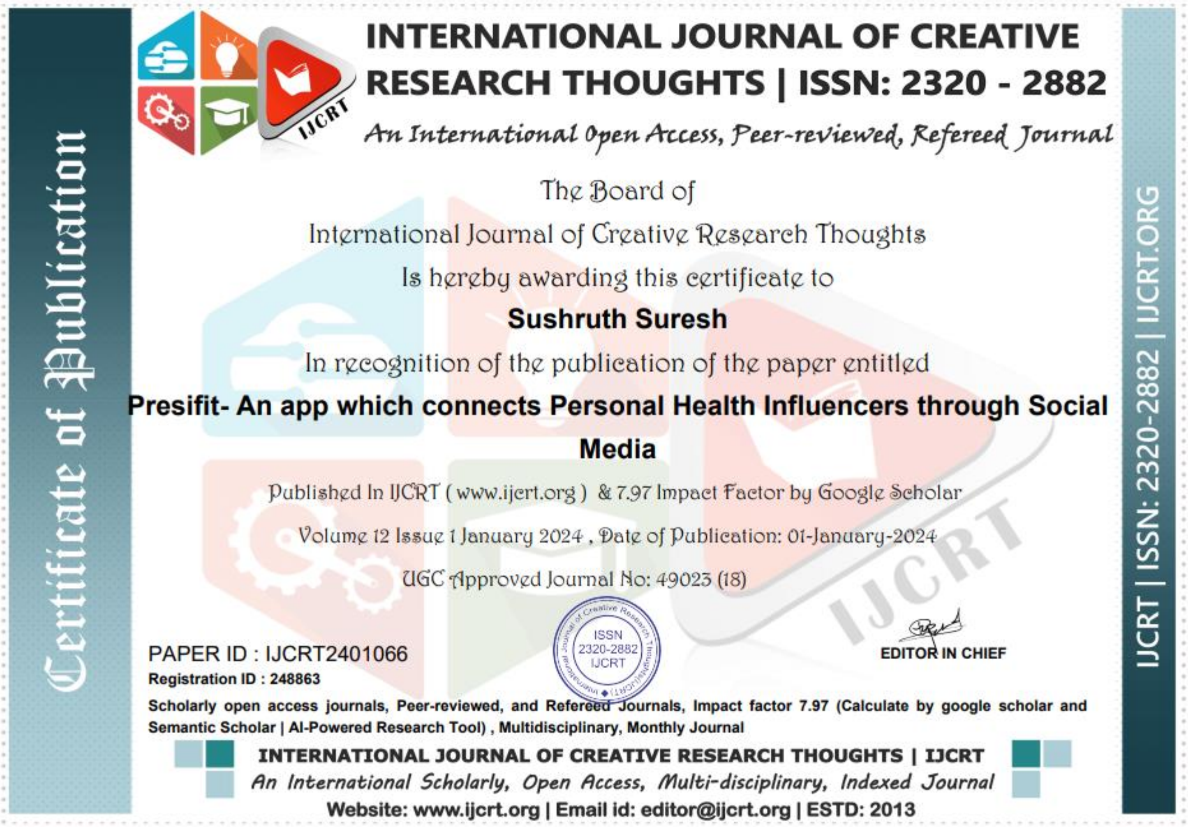
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Sushruth Suresh

In recognition of the publication of the paper entitled

Presifit- An app which connects Personal Health Influencers through Social Media

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 12 Issue 1 January 2024 , Date of Publication: 01-January-2024

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2401066

Registration ID : 248863

ISSN 2320-2882 IJCRT

International Journal of Creative Research Thoughts (IJCRT)

EDITOR IN CHIEF

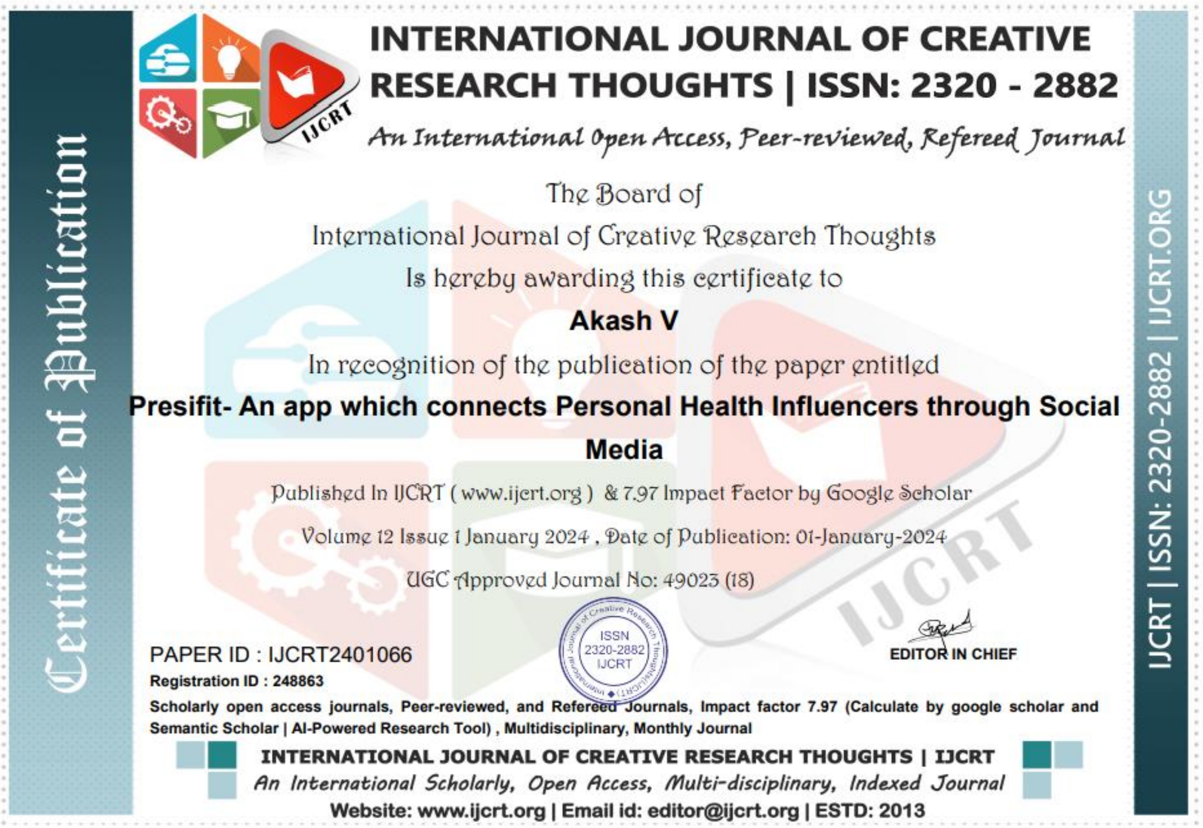
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Akash V

In recognition of the publication of the paper entitled

Presifit- An app which connects Personal Health Influencers through Social Media

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 12 Issue 1 January 2024 , Date of Publication: 01-January-2024

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2401066

Registration ID : 248863

International Journal of Creative Research Thoughts (IJCRT)
ISSN
2320-2882
IJCRT

EDITOR IN CHIEF

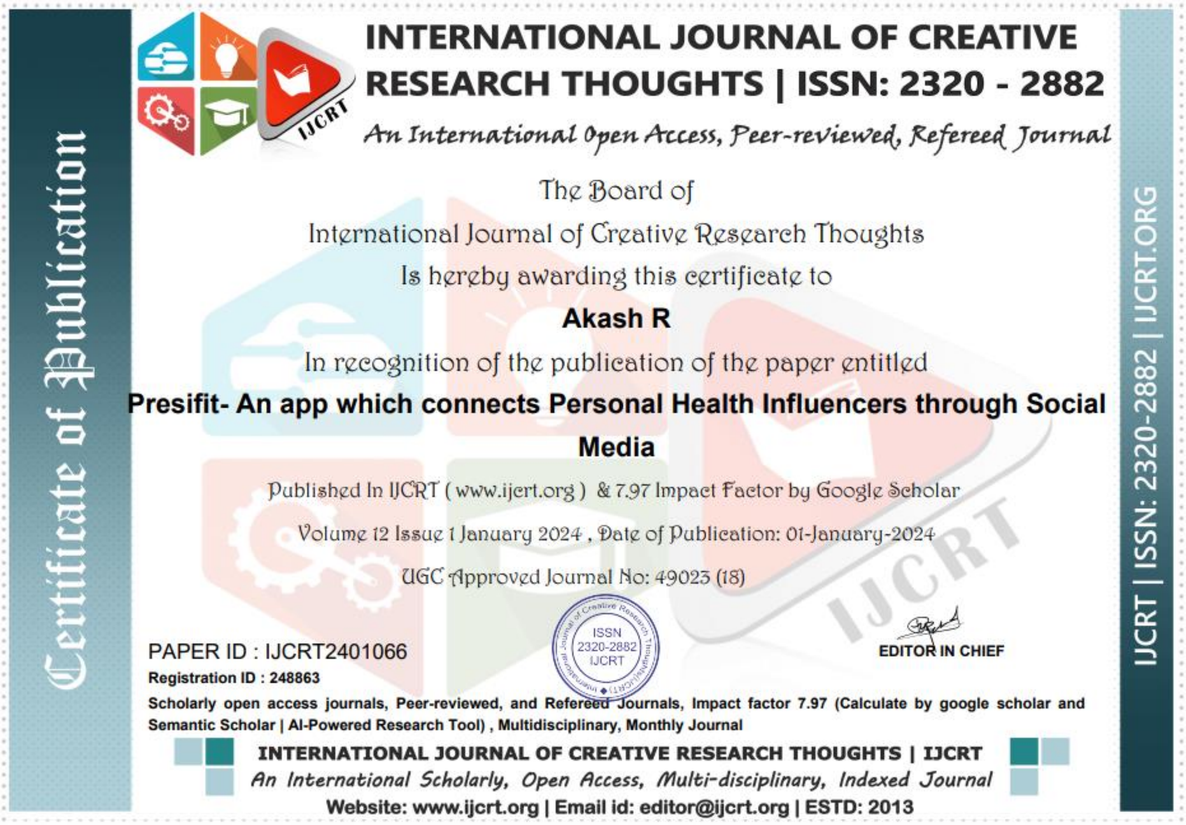
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Akash R

In recognition of the publication of the paper entitled

Presifit- An app which connects Personal Health Influencers through Social Media

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 12 Issue 1 January 2024 , Date of Publication: 01-January-2024

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2401066

Registration ID : 248863

International Journal of Creative Research Thoughts ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

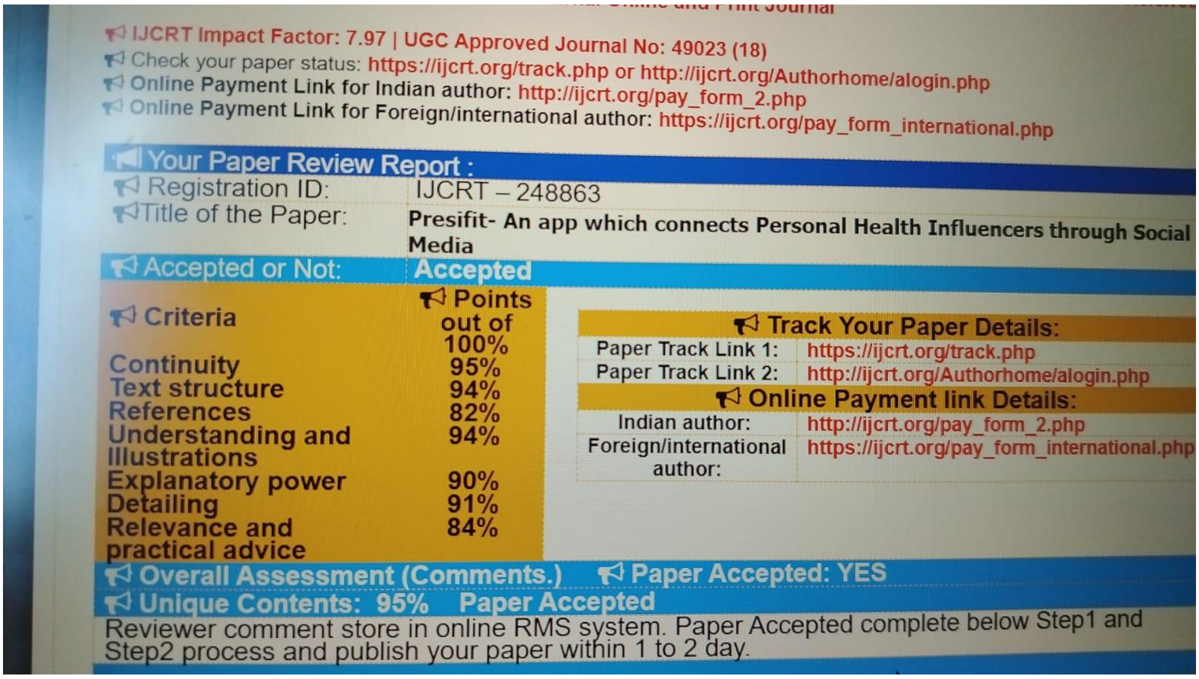
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG

2. Similarity Index / Plagiarism Check report



IJCRT Impact Factor: 7.97 | UGC Approved Journal No: 49023 (18)

Check your paper status: https://ijcrt.org/track.php or http://ijcrt.org/Authorhome/alogin.php

Online Payment Link for Indian author: http://ijcrt.org/pay_form_2.php

Online Payment Link for Foreign/international author: https://ijcrt.org/pay_form_international.php

Your Paper Review Report :

Registration ID:	IJCRT – 248863
Title of the Paper:	Presifit- An app which connects Personal Health Influencers through Social Media
Accepted or Not:	Accepted

Criteria	Points out of 100%
Continuity	95%
Text structure	94%
References	82%
Understanding and Illustrations	94%
Explanatory power	90%
Detailing	91%
Relevance and practical advice	84%

Track Your Paper Details:

Paper Track Link 1:	https://ijcrt.org/track.php
Paper Track Link 2:	http://ijcrt.org/Authorhome/alogin.php

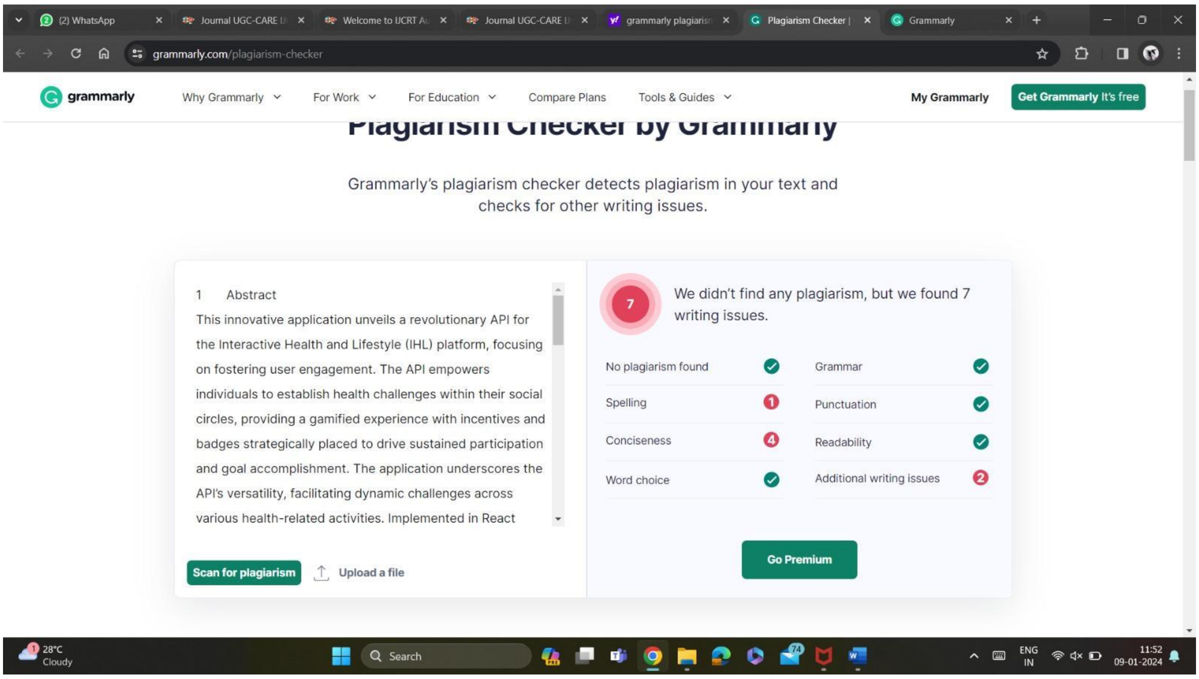
Online Payment link Details:

Indian author:	http://ijcrt.org/pay_form_2.php
Foreign/international author:	https://ijcrt.org/pay_form_international.php

Overall Assessment (Comments.) Paper Accepted: YES

Unique Contents: 95% Paper Accepted

Reviewer comment store in online RMS system. Paper Accepted complete below Step1 and Step2 process and publish your paper within 1 to 2 day.



Plagiarism Checker by Grammarly

Grammarly's plagiarism checker detects plagiarism in your text and checks for other writing issues.

1 Abstract

This innovative application unveils a revolutionary API for the Interactive Health and Lifestyle (IHL) platform, focusing on fostering user engagement. The API empowers individuals to establish health challenges within their social circles, providing a gamified experience with incentives and badges strategically placed to drive sustained participation and goal accomplishment. The application underscores the API's versatility, facilitating dynamic challenges across various health-related activities. Implemented in React

7 We didn't find any plagiarism, but we found 7 writing issues.

No plagiarism found	✓	Grammar	✓
Spelling	1	Punctuation	✓
Conciseness	4	Readability	✓
Word choice	✓	Additional writing issues	2

PERSONAL HEALTH INFLUENCERS THROUGH SOCIAL NETWORK- PRESIFIT

ORIGINALITY REPORT

13%	10%	5%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	3%
2	www.coursehero.com Internet Source	1%
3	B Varshini, HR Yogesh, Syed Danish Pasha, Maaz Suhail, V Madhumitha, Archana Sasi. "IoT-Enabled Smart Doors for Monitoring Body Temperature and Face Mask Detection", Global Transitions Proceedings, 2021 Publication	1%
4	Submitted to University of Greenwich Student Paper	1%
5	Submitted to Monash University Student Paper	1%
6	www.mdpi.com Internet Source	<1%
7	Submitted to Al-Nahrain University Student Paper	<1%

8	Submitted to University of Westminster Student Paper	<1%
9	dev.to Internet Source	<1%
10	Submitted to Victoria University Student Paper	<1%
11	pdfs.semanticscholar.org Internet Source	<1%
12	coek.info Internet Source	<1%
13	Submitted to CSU, San Jose State University Student Paper	<1%
14	djaytechdiary.com Internet Source	<1%
15	surajgyawali.com Internet Source	<1%
16	e-hir.org Internet Source	<1%
17	pubmed.ncbi.nlm.nih.gov Internet Source	<1%
18	Submitted to HCUC Student Paper	<1%
19	Submitted to Trine University Student Paper	<1%

20	escholarship.org Internet Source	<1%
21	www.c-sharpcorner.com Internet Source	<1%
22	www.geeksforgeeks.org Internet Source	<1%
23	Submitted to University of Cape Town Student Paper	<1%
24	Submitted to University of Leicester Student Paper	<1%
25	www.arge-bfw.de Internet Source	<1%
26	www.researchgate.net Internet Source	<1%
27	andywfec454.website2.me Internet Source	<1%
28	artoonsolutions.com Internet Source	<1%
29	cyber.gc.ca Internet Source	<1%
30	www.wisdomlib.org Internet Source	<1%
31	www.change.org Internet Source	<1%

32	www.corporatewellnessmagazine.com Internet Source	<1%
33	Submitted to Southampton Solent University Student Paper	<1%
34	article.sciencepublishinggroup.com Internet Source	<1%
35	iq.opengenus.org Internet Source	<1%
36	slides.com Internet Source	<1%
37	Alaa AlSlaity, Banuchitra Suruliraj, Oladapo Oyebode, Jonathon Fowles, darren steeves, Rita Orji. "Mobile Applications for Health and Wellness: A Systematic Review", Proceedings of the ACM on Human-Computer Interaction, 2022 Publication	<1%
38	Bhanu Sankar Ravi, Chepuri Madhukanthi, P Sivasankar, John Deva Prasanna. "A Novel Technique to Improve Latency and Response Time of AI Models using Serverless Infrastructure", 2023 International Conference on Inventive Computation Technologies (ICICT), 2023 Publication	<1%
39	awplife.com	

Internet Source

<1%

40 mirrors.maine.edu

Internet Source

<1%

41 Antonio Fabbrizio, Alberto Fucarino, Manuela Cantoia, Andrea De Giorgio et al. "Smart Devices for Health and Wellness Applied to Tele-Exercise: An Overview of New Trends and Technologies Such as IoT and AI", Healthcare, 2023

Publication

<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		