MEDIGUARD: REAL-TIME HEALTH MONITORING AND AMBULANCE DISPATCH SYSTEM

Lakshmi. T[1], Ganesh.V[2,*], Gobi. L[3]

[1]Assistant Professor, Department of Artificial Intelligence and Data Science Karpaga Vinayaga College ofEngineering and Technology, Chengalpattu, Tamil Nadu, India.

[2,*]Assistant Professor, Department of Bio Chemistry, Karpaga Vinayaga Institute of Medical Science and Research Center, Chengalpattu, Tamil Nadu, India.

[3]Student Department, of Artificial Intelligence and Data Science Karpaga Vinayaga College ofEngineering and Technology, Chengalpattu, Tamil Nadu, India.

Corresponding Author

ganeshlak85@gmail.com

ABSTRACT

This project represents an innovative approach to revolutionizing crisis medical services (EMS) through the integration of advanced technology and streamlined processes. Mediguard is a instantaneous health tracking and ambulance dispatch framework leveraging wearable devices and sophisticated information analytics. It provides proactive health tracking for individuals and swift ambulance dispatch during emergencies. The AARS enhances EMS efficiency by utilizing instantaneous information from GPS, medical sensors, and user input to detect emergencies. Upon alert activation, the framework identifies the nearest available ambulance equipped with necessary medical facilities and optimally routes it to the crisis location. Key features include live tracking, communication between responders and medical professionals, and automated patient health tracking during transit. Both frameworks utilize Wi-Fi and cellular networks for seamless information transmission, addressing critical requirements to improve EMS efficiency and patient care. The framework aims to significantly reduce reaction times, save lives, and improve overall medical services outcomes.

I. INTRODUCTION

A. MEDIGUARD

Mediguard represents a transformative leap in medical services technology, offering a comprehensive wearable health tracking framework with automatic ambulance alert capabilities. Seamlessly blending cutting-edge sensors and sophisticated algorithms, Mediguard provides users with continuous insights into their vital signs, including heart rate, SpO2 levels, body temperature, and precise GPS location.

The framework's proactive approach to health management shines in crisis situations. When abnormal health parameters are detected or the wearer activates the crisis button, Mediguard springs into action. Leveraging its intelligent alert framework and Firebase Cloud integration, it instantly notifies crisis responders, facilitating swift ambulance dispatch to the wearer's exact location. This rapid reaction mechanism is invaluable in critical moments, potentially saving lives and minimizing medical complications.

Beyond crisis scenarios, Mediguard fosters a sense of security and peace of mind for users and their families. Its user-friendly interface and reliable performance empower individuals to lead active lives while knowing that help is readily available when needed. By merging advanced technology with compassionate care, Mediguard sets a new standard in personalized medical services, ensuring that every individual receives the attention and support they deserve, precisely when they need it most.

B. EXISTING SYSTEM

Existing work in wearable devices already offer extensive health tracking capabilities, including heart rate, physical activity tracking, and sleep analysis. Furthermore, cloud-based platforms are commonly used for storing and analyzing information collected by wearable devices, enabling remote tracking and seamless integration with crisis reaction protocols.

C. PROPOSED SYSTEM

By building upon these existing technologies and incorporating innovative features, "Mediguard" aims to provide a comprehensive solution for instantaneous health tracking and rapid crisis assistance, ultimately enhancing the safety and well-being of users.

The proposed framework operates under the control of an ESP-32 microcontroller, serving as the central processing unit. Sensor modules including the MAX30105 for heart rate and SpO2 tracking, and the LM35 for body temperature sensing, are integrated into the framework. These sensors continuously collect physiological information. Upon surpassing predefined threshold values, the ESP-32 triggers alerts. Data packets containing the wearer's location, along with the alert status, are transmitted to the Firebase Cloud via the ESP-32's communication capabilities. Furthermore, a dedicated push button is incorporated into the framework, allowing the wearer to manually initiate crisis alerts. When pressed, the push button triggers the ESP-32 to send immediate location and crisis status updates to the Firebase Cloud. This comprehensive framework provides instantaneous health tracking and enables prompt crisis reaction, enhancing the wearer's safety and facilitating timely assistance when needed.

D. BLOCK DIAGRAM

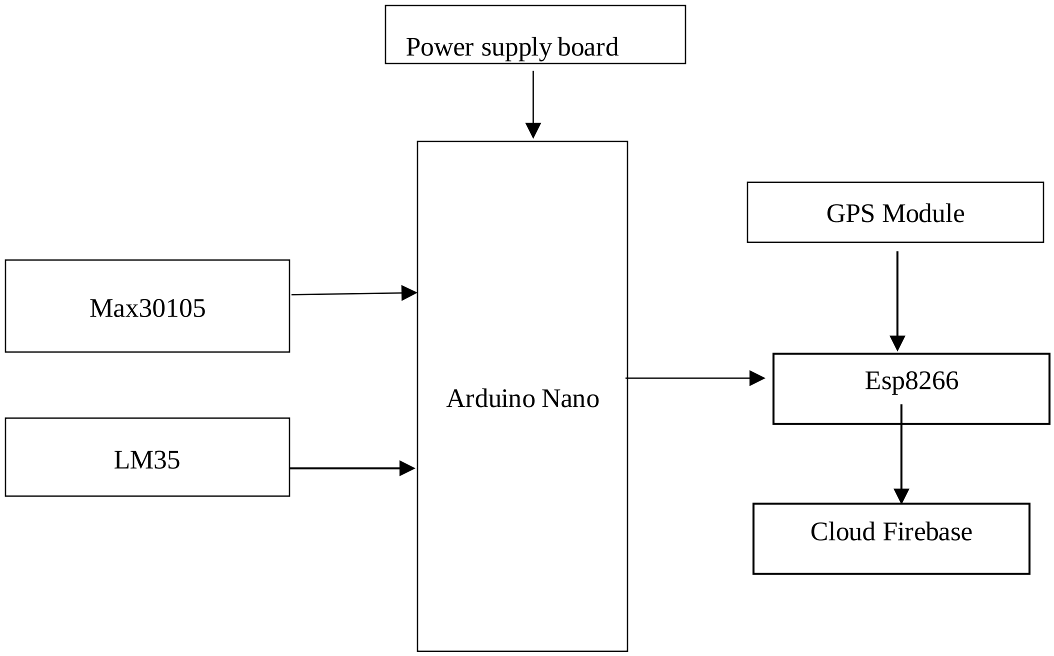

Fig.1 Block Diagram

E. INTRODUCTION TO IOT

IoT refers to an Internet of Things (IoT). Connecting any device (including everything from cell phones, vehicles, home appliances and other wearable embedded with sensors and actuators) with Internet so that these objects can exchange information with each other on a network. It is interesting to note that there is a difference between IoT and the Internet; it is the absence of Human role. The IoT devices can create information about individual's behaviors, analyze it, and take action.

Smart System and the Internet of the Things are driven by a combination for:

1. Sensors & Actuators
2. Connectivity
3. People & Process

F. INTEROPERABILITY IN IOT

The Internet of Things (IoT) is an incredibly diverse space, encompassing a large variety of hardware form factors and software ecoframeworks unlike anything we have seen in technology. Smart watches, connected cameras, drones, thermostats, voice-enabled speakers, smart appliances and more—they all live together within the IoT.The diversity and innovation that excites many IoT fans is a big challenge not just for manufacturers and developers, but also (and most importantly) consumers. Which technology options should be used when designing or deploying IoT devices? How do they keep up with updated or new operating frameworks? What about new software and connectivity technologies coming up? Those are just some of today's challenges. Having a single, unified communication and software framework for the IoT seems like an ideal solution, but the diverse and fast paced nature of the IoT makes this utopia a big challenge. Diversity in the IoT is not something to be solved, but an aspect that must be embraced and managed.

G. APPLICATION AREAS FOR THE INTERNET OF THINGS

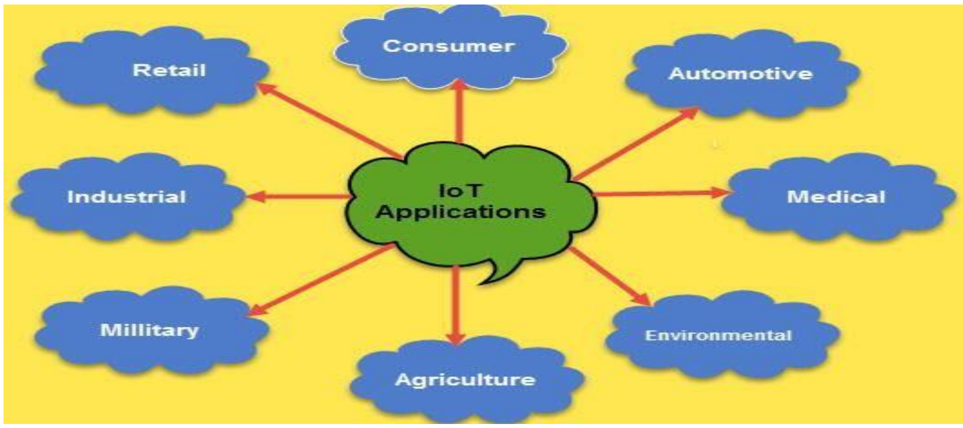

Fig.2 Application of IOT

a. Smart Home

The concept of Smart Home is brought up to save time, energy and money. With the introduction of Smart Homes, we would be able to switch on air conditioning before reaching home or switch off lights even after leaving home or unlock the doors to friends for temporary access even when you are not at home.

b. Smart cities

Smart surveillance, automated transportation, smarter energy management frameworks, water distribution, urban security and environmental tracking all are examples of internet of things applications for smart cities. IoT will solve major problems faced by the people living in cities like pollution, traffic congestion and shortage of energy supplies etc. By installing sensors and using web applications, citizens can find free available parking slots across the city. Also, the sensors can detect meter tampering issues, general malfunctions, and any installation issues in the electricity framework.

c. Wearables

Wearable devices are installed with sensors and software's which collect information and information about the users. This information is later pre-processed to extract essential insights about user. These devices broadly cover fitness, health and entertainment requirements. The pre-requisite from internet of things technology for wearable applications is to be highly energy efficient or ultra-low power and small sized.

d. Healthcare

IoT in medical services is aimed at empowering people to live healthier life and regular checkup by wearing connected devices. The collected information will help in personalized analysis of an individual's health and provide tailor made strategies to combat illness.

H. SECURITY CHALLENGES

Security is a big issue with IoT devices. With billions of devices being connected together over Internet, how can people be sure that their information is secure? These security issues can be of the following kinds.

a. Data Encryption

IoT applications collect tons of information. Data retrieval and processing is integral part of the whole IoT environment. Most of this information is personal and needs to be protected through encryption. Encryption is widely used on the internet to protect user information being sent between a browser and a server, including passwords, payment information and other personal information that should be considered private.

b. Data Authentication

After successful encryption of information chances of device itself being hacked still exist. If there is no way to establish the authenticity of the information being communicated to and from an IoT device, security is compromised. For instance, say you built a temperature sensor for smart homes. Even though you encrypt the information it transfers is there is no way to authenticate the source of information then anyone can make up fake information and send it to your sensor instructing it to cool the room even when its freezing or vice versa.

c. Side-channel Attacks

Encryption and authentication both in place still leave scope for side channel attacks. Such attacks focus less on the information and more on how that information is being presented.

d. Privacy Challenges

Then we have the issue of privacy and information sharing. That is because these devices not only collect personal information like users' names and telephone numbers, but can also monitor user activities (e.g., when users are in their houses and what they had for lunch).

e. Connectivity Challenges—Billions of devices on a centralized server

One of the biggest challenges for IoT in the future is to connect large number of devices and massive amounts of information that all of these devices are going to produce. There will be need to find out a way to store, track, analyze and make sense of the vast amounts of information that will be generated. Presently, we rely upon centralized, server/client model to authorize, authenticate, and connect several nodes present on the network. This model is sufficient for the number of IoT devices that are currently a part of the ecoframework.

f. Compatibility and Longevity Challenges-Extra hardware and software

Different technologies like ZigBee, Z-Wave, Wi-Fi, Bluetooth and Bluetooth Low Energy (BTLE) are all battling to become the dominant transport mechanism between devices and hubs. This becomes a major source of problems when a lot of devices have to be connected; such dense connectivity requires the deployment of extra hardware and software. Conversations about the IoT are taking place all over the world as we are trying to understand how this will impact our lives. We are also trying to understand what the many opportunities and challenges are going to be as more and more devices start to join the IoT. So, all that we can do is educate ourselves about what the IoT is and how it will be after some years.

I. FIREBASE

With a variety of server-side technologies that are on the market today, developers have a tough job of deciding what kind of backend is most suitable for their app. In this post, we will explore one of these choices that go by the name of Firebase, and all the tools and services that it provides.

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

Back in 2011, before Firebase was Firebase, it was a start-up called Envolve. As Envolve, it provided developers with an API that enabled the integration of online chat functionality into their website. What's interesting is that people used Envolve to pass application information that was more than just chat messages. Developers were using Envolve to sync application information such as a game state in real time across their users.

This led the founders of Envolve, James Tamplin and Andrew Lee, to separate the chat framework and the instantaneous architecture. In April 2012, Firebase was created as a separate company that provided Backend-as-a-Service with instantaneous functionality. After it was acquired by Google in 2014, Firebase rapidly evolved into the multifunctional behemoth of a mobile and web platform that it is today.

a. Real-time Database

The Firebase Real-time Database is a cloud-hosted NoSQL information base that lets you store and sync between your users in Real-time.

The Real-time Database is really just one big JSON object that the developers can manage in Real-time.

Real-time Database => A Tree of Values

With just a single API, the Firebase information base provides your app with both the current value of the information and any updates to that information.

Real-time syncing makes it easy for your users to access their information from any device, be it web or mobile. Real-time Database also helps your users collaborate with one another. Another amazing benefit of Real-time Database is that it ships with mobile and web SDKs, allowing you to build your apps without the need for servers.

When your users go offline, the Real-time Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local information is automatically synchronized. The Real-time Database can also integrate with Firebase Authentication to provide a simple and intuitive authentication process.

b. Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. Normally, it would take you months to set up your own authentication framework. And even after that, you would need to keep a dedicated team to maintain that framework. But if you use Firebase, you can set up the entire framework in under 10 lines of code that will handle everything for you, including complex operations like account merging.

We can authenticate our app's users through the following methods:

- Email & Password
- Phone numbers
- Google
- Facebook
- Twitter
- & more!

Using Firebase Authentication makes building secure authentication frameworks easier, while also improving the sign in and on boarding experience for end users. Firebase Authentication is built by the same people who created Google Sign-in, Smart Lock, and Chrome Password Manager.

c. Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive messages and notifications on iOS, Android, and the web at no cost.

You can send notification messages (2KB limit) and information messages (4KB limit). Using FCM, you can easily target messages using predefined segments or create your own, using demographics and behaviour. You can send messages to a group of devices that are subscribed to specific topics, or you can get as granular as a single device.

FCM can deliver messages instantly, or at a future time in the user's local time zone. You can send custom app information like setting priorities, sounds, and expiration dates, and also track custom conversion events. The best thing about FCM is that there is hardly any coding involved! FCM is completely integrated with Firebase Analytics, giving you detailed engagement and conversion tracking.

d. Firebase Database Query

Firebase has simplified the process of retrieving specific information from the informationbase through queries. Queries are created by chaining together one or more filter methods.

Firebase has 4 ordering functions:

- Order By Key()
- Order By Child('child')
- Order By Value()
- Order By Priority()

II. REVIEW OF LITERATURE

A. REVIEW OF LITERATURE

Literature Survey is a framework tic and thorough search of all types of published literature as well as other sources including dissertation, these to identify as many items as possible that are relevant to a particular topic.

a. Title: "Intelligent Ambulance Dispatch Systems: A Comprehensive Survey "

Author: Sophia Patel

Year: 2023

Overview: This review focuses on recent advancements in intelligent ambulance dispatch frameworks, including the integration of instantaneous information analytics, machine learning algorithms, and crowd-sourced information. It explores how these technologies enhance crisis reaction by improving dispatch accuracy, optimizing routing, and facilitating dynamic resource allocation.

Advantage: Enhanced responsiveness, improved decision-making, and better adaptation to dynamic crisis situations.

Disadvantage: Dependency on accurate and up-to-date information sources, potential biases in algorithmic decision-making, and challenges in integrating heterogeneous information sources.

b. Title: "Next-Generation Automatic Ambulance Routing System: A Comprehensive Survey"

Author: Daniel Wong

Year: 2022

Overview: This survey examines recent trends and innovations in automatic ambulance routing frameworks, including the use of artificial intelligence, predictive analytics, and multi-criteria optimization techniques. It discusses how these advancements enable more efficient routing, faster reaction times, and enhanced patient outcomes in crisis medical services.

Advantage: Enhanced routing accuracy, adaptive reaction to changing traffic conditions, and improve scalability for large- scale crisis events.

Disadvantage: Complexity in algorithm design, potential computational overhead, and the need for continuous validation and refinement.

c. Title: "Advancements in Autonomous Ambulance Systems: A Comprehensive Survey "

Author: Rachel Johnson

Year: 2024

Overview: This review examines recent developments in autonomous ambulance frameworks, including advancements in self-driving vehicle technology, sensor fusion, and artificial intelligence. It explores how these innovations are transforming crisis medical services by enabling autonomous navigation, remote tracking, and telemedicine capabilities onboard ambulances.

Advantage: Reduced reaction times, improved safety through autonomous driving, and enhanced medical care delivery during transit.

Disadvantage: Regulatory challenges, public perception barriers, and technical limitations in autonomous vehicle deployment.

d. Title: "Recent Progress in Real-Time Ambulance Tracking Systems: A Comprehensive Survey"

Author: Alexander Garcia

Year: 2023

Overview: This comprehensive survey provides an overview of recent progress in instantaneous ambulance tracking frameworks, focusing on advancements in GPS technology, information visualization, and integration with crisis dispatch platforms. It discusses how these frameworks improve situational awareness, coordination among crisis responders, and patient care delivery.

Advantage: Enhanced visibility into ambulance locations, improved reaction coordination, and better communication among crisis services.

Disadvantage: Privacy concerns regarding location information, potential vulnerabilities to cyberattacks, and the need for robust information security measures.

e. Title: "Recent Advances in Optimization Techniques for Ambulance Routing: A Comprehensive Survey"

Author: Julia Martinez

Year: 2022

Overview: This state-of-the-art review explores recent advances in optimization techniques for ambulance routing, including metaheuristic algorithms, hybrid approaches, and instantaneous adaptive strategies. It highlights how these advancements address the complexity of crisis medical service logistics, leading to more efficient resource allocation and faster crisis reaction times.

Advantage: Improved route optimization, adaptive reaction to dynamic conditions, and scalability for large-scale crisis events.

Disadvantage: Computational complexity, sensitivity to parameter settings, and challenges in instantaneous implementation on resource-constrained platforms.

III.IMPLEMENTATION

EXPERIMENTAL RESULTS

A. FIRE BASE

Firebase is a mobile and web app development platform that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit.

a. A Brief History

Back in 2011, before Firebase was Firebase, it was a start-up called Envolve. As Envolve, it provided developers with an API that enabled the integration of online chat functionality into their website. What's interesting is that people used Envolve to pass application information that was more than just chat messages. Developers were using Envolve to sync application information such as a game state in real time across their users.

This led the founders of Envolve, and, to separate the chat framework and the instantaneous architecture. In April 2012, Firebase was created as a separate company that provided Backend-as-a-Service with instantaneous functionality. After it was acquired by Google in 2014, Firebase rapidly evolved into the multifunctional behemoth of a mobile and web platform that it is today.

b. Real-time Database

The Firebase Real-time Database is a cloud-hosted NoSQL information base that lets you store and sync between your users in Real-time. The Real-time Database is really just one big JSON object that the developers can manage in Real-time.

Real-time Database => A Tree of Values

With just a single API, the Firebase information base provides your app with both the current value of the information and any updates to that information.

Limit To First (10)

Limit To Last (10) Real-time syncing makes it easy for your users to access their information from any device, be it web or mobile. Real-time Database also helps your users collaborate with one another. Another amazing benefit of Real-time Database is that it ships with mobile and web SDKs, allowing you to build your apps without the need for servers.

c. Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive messages and notifications on iOS, Android, and the web at no cost.

We can send notification messages (2KB limit) and information messages (4KB limit). Using FCM, you can easily target messages using predefined segments or create your own, using demographics and behavior. You can send messages to a group of devices that are subscribed to specific topics, or you can get as granular as a single device.

FCM can deliver messages instantly, or at a future time in the user's local time zone. You can send custom app information like setting priorities, sounds, and expiration dates, and also track custom conversion events.The best thing about FCM is that there is hardly any coding involved! FCM is completely integrated with Firebase Analytics, giving you detailed engagement and conversion tracking.

d. Firebase Database Query

Firebase has simplified the process of retrieving specific information from the information base through queries. Queries are created by chaining together one or more filter methods.

Firebase has 4 ordering functions:

- Order By Key ()
- Order By Child('child')
- Order By Value ()
- Order By Priority ()

Note that we will only receive information from a query if you have used the on () or once () method.

We can also use these advanced querying functions to further restrict information:

- Start At('value')
- End At('value')
- Equal To ('child key')
- Limit To First (10)
- Limit To Last (10)

In SQL, the basics of querying involve two steps. First, you select the columns from your table. Here I am selecting the Users column. Next, you can apply a restriction to your query using the WHERE clause. From the below-given query, I will get a list of Users whose name is Geek Ants.

Fig.3 Firebase Query

We can also use the LIMIT clause, which will restrict the number of results that you will get back from your query.

In Firebase, querying also involves two steps. First, you create a reference to the parent key and then you use an ordering function. Optionally, you can also append a querying function for a more advanced restricting.

```
const db = firebase.database();

const firebaseRef = db.child(`child`);

firebaseRef.orderByChild("user").equalTo("GeekyAnts").on("child_added",
    function(snapshot) {
        console.log(snapshot.key);
    }
);
```

Fig.4 Firebase Query

How to Store Data? => Firebase Storage

Firebase Storage is a standalone solution for uploading user-generated content like images and videos from an iOS and Android device, as well as the Web. Firebase Storage is designed specifically to scale your apps, provide security, and ensure network resiliency. Firebase Storage uses a simple folder/file framework to structure its information.

```
var storageRef = firebase.storage.ref("folderName/file.jpg");
var fileUpload = document.getElementById("fileUpload");
fileUpload.on('change',
    function(evt) {
        var firstFile = evt.target.file[0]; // get the first file uploaded
        var uploadTask = storageRef.put(firstFile);
    }
);
```

Fig.5 Firebase Query

e. Query with Fire store

Imagine that you have created a collection in Fire store that contains a list of Cities. So, before you can send out query, you will have to store the information base inside a variable.

```
var citiesRef = db.collection("cities");
```

Fig.6 Fire store query

Here's another example of queries in Fire store. Say you want to see only 2 of cities from your information base whose population is more than 100,000.

```
citiesRef.where("population", ">", 100000).orderBy("population").limit(2)
```

Fig.7 Fire store query

But Cloud Fire store can make querying even easier! In some cases, Cloud Fire store can automatically search your information base across multiple fields. Fire store will guide you towards automatically building an index that will help Fire store to make querying extremely simple.

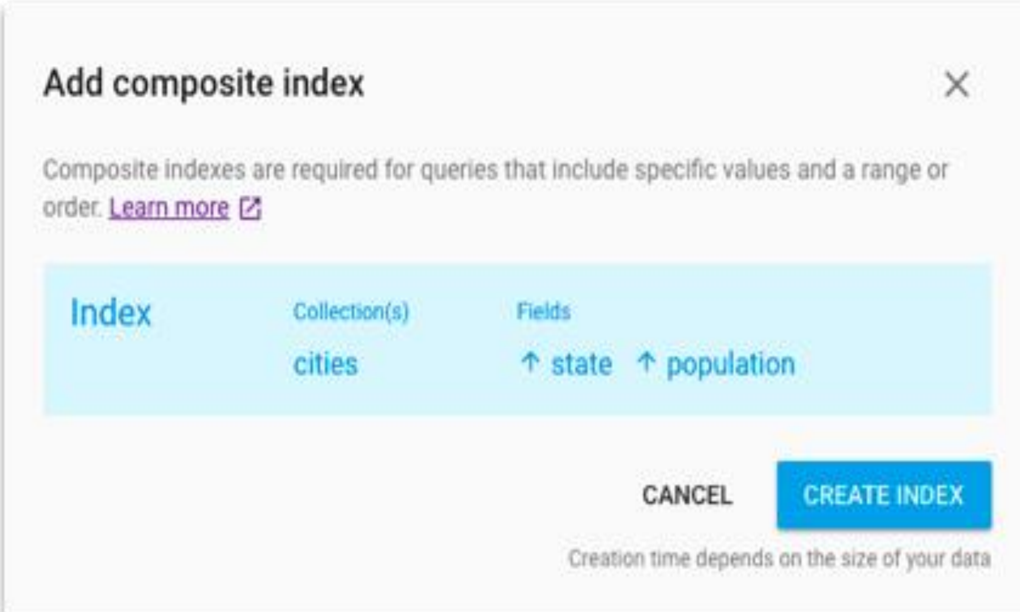

Fig.8 Fire store Index

f. Better Scalability

Though Firebase's Real-time Database is capable of scaling, things will start to get crazy when you app becomes really popular or if your information base becomes really massive. Cloud Fire store is based on Googles Cloud infrastructure. This allows it to scale much more easily and to a greater capacity than the Real-time Database.

A. PSYCHOLOGICAL DATA

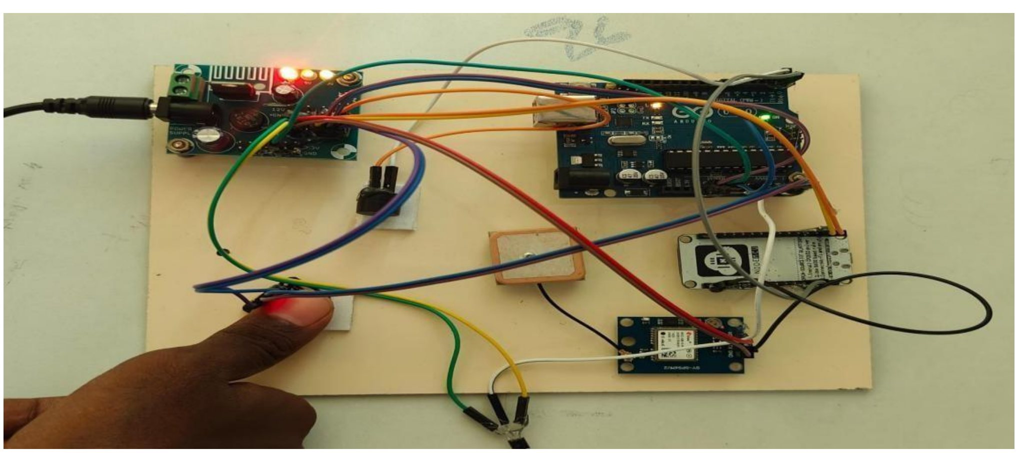

Fig.9 Psychological Data

B. FIREBASE RESULT

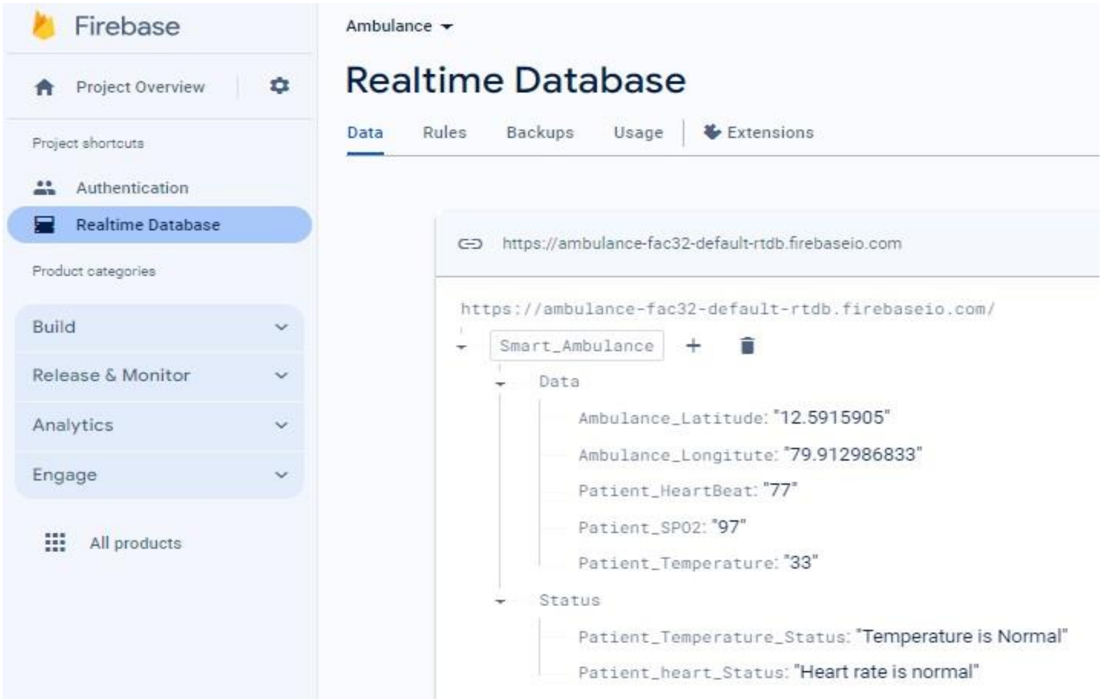

Fig.10 Firebase Result

IV. CONCLUSION

In conclusion, driven by the ESP-32 microcontroller and equipped with sensor modules for continuous physiological tracking, provides a robust solution for instantaneous health tracking and crisis reaction. By integrating sensors like the MAX30105 and LM35, the framework can effectively monitor vital signs and detect abnormalities, triggering alerts when predefined threshold values are surpassed. The seamless transmission of information packets containing wearer information and alert status to the Firebase Cloud ensures remote accessibility and facilitates prompt intervention in crisis situations.

V. FUTURE SCOPE

In Future Scope, by integrating wearable devices and IoT sensors can enable continuous health tracking, fostering a more comprehensive and personalized approach to medical services delivery. Strengthening information encryption and privacy measures will be crucial to ensure compliance with regulations and safeguard sensitive health information, instilling trust and confidence among users.

VI. REFERENCE

[1] Yang, B.; Haghighat, F.; Fung, B.C.; Panchabikesan, K. Season-Based Occupancy Prediction in Residential Buildings Using Machine Learning Models. E-Prime-Adv. Electr. Eng. Electron. Energy 2021, 1, 100003. [] []

[2] Naseem, S.; Alhudhaif, A.; Anwar, M.; Qureshi, K.N.; Jeon, G. Artificial general intelligence-based rational behavior detection using cognitive correlates for tracking online harms. Pers. Ubiquitous Comput. 2022, 17, 1–9. [] []

[3]Rajesh, G.; Benny, A.R.; Harikrishnan, A.; Abraham, J.J.; John, N.P. A Deep Learning based Accident Detection System. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 1322–1325. []

[4] Wang, C.; Dai, Y.; Zhou, W.; Geng, Y. A Vision-Based Video Crash Detection Framework for Mixed Traffic Flow Environment Considering Low-Visibility Condition. J. Adv. Transp. 2020, 2020, 9194028. [] [] []

[5] Bhakat, A.; Chahar, N.; Vijayasherly, V. Vehicle Accident Detection & Alert System using IoT and Artificial Intelligence. In Proceedings of the 2021 Asian Conference on Innovation in Technology (ASIANCON), Pune, India, 27–29 August 2021; pp. 1–7. []

[6] Choi, J.G.; Kong, C.W.; Kim, G.; Lim, S. Car crash detection using ensemble deep learning and multimodal information from dashboard cameras. Expert Syst. Appl. 2021, 183, 115400. [] []

[7] Pour, H.H.; Li, F.; Wegmeth, L.; Trense, C.; Doniec, R.; Grzegorzek, M.; Wismüller, R. A Machine Learning Framework for Automated Accident Detection Based on Multimodal Sensors. Cars. Sens. 2022, 2022, 1–21. []

[8] Comi, A.; Polimeni, A.; Balsamo, C. Road Accident Analysis with Data Mining Approach: Evidence from Rome. Transp. Res. Procedia 2022, 62, 798–805. [] []

[9] Park, E.S.; Fitzpatrick, K.; Das, S.; Avelar, R. Exploration of the relationship among roadway characteristics, operating speed, and crashes for city streets using path analysis. Accid. Anal. Prev. 2021, 150, 105896. [] [] []

[10] Singh, G.; Pal, M.; Yadav, Y.; Singla, T. Deep neural network-based predictive modeling of road accidents. Neural Comput. Appl. 2020, 32, 12417–12426. [] []

[11] Gupta, R.K.; Bharti, S.; Kunhare, N.; Sahu, Y.; Pathik, N. Brain Tumor Detection and Classification Using Cycle Generative Adversarial Networks. Interdiscip. Sci. Comput. Life Sci. 2022, 17, 1–17. []

[12] Xie, Y.; Xie, B.; Wang, Z.; Gupta, R.K.; Baz, M.; AlZain, M.A.; Masud, M. Geological Resource Planning and Environmental Impact Assessments Based on GIS. Sustainability 2022, 14, 906. [] []

[13] Yan, L.; Cengiz, K.; Sharma, A. An improved image processing algorithm for automatic defect inspection in TFT-LCD TCON. Nonlinear Eng. 2021, 10, 293–303. [] []

[14] Zhang, X.; Rane, K.P.; Kakaravada, I.; Shabaz, M. Research on vibration tracking and fault diagnosis of rotating machinery based on internet of things technology. Nonlinear Eng. 2021, 10, 245–254. [] []

[15] Guo, Z.; Xiao, Z. Research on online calibration of lidar and camera for intelligent connected vehicles based on depth-edge matching. Nonlinear Eng. 2021, 10, 469–476. [] []

[16] Xie, H.; Wang, Y.; Gao, Z.; Ganthia, B.P.; Truong, C.V. Research on frequency parameter detection of frequency shifted track circuit based on nonlinear algorithm. Nonlinear Eng. 2021, 10, 592–599. [] []

[17] Liu, J.; Khattak, A.J.; Li, X.; Nie, Q.; Ling, Z. Bicyclist injury severity in traffic crashes: A spatial approach for geo-referenced crash information to uncover non-stationary correlates. J. Saf. Res. 2020, 73, 25–35. [] [].

[18] Dashora, C.; Sudhagar, P.E.; Marietta, J. IoT based framework for the detection of vehicle accident. Cloud Comput. 2019, 2, 1–16. [] []

[19] Yan, L.; Cengiz, K.; Sharma, A. An improved image processing algorithm for automatic defect inspection in TFT-LCD TCON. Nonlinear Eng. 2021, 10, 293–303. [] []

[20] Zhang, X.; Rane, K.P.; Kakaravada, I.; Shabaz, M. Research on vibration tracking and fault diagnosis of rotating machinery based on internet of things technology. Nonlinear Eng. 2021, 10, 245–254. [] []