COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DE-41 EE
PROJECT REPORT

IOT BASED SMART HELMET

Submitted to the Department of Electrical Engineering
in partial fulfillment of the requirements
for the degree of
Bachelor of Engineering
in
Electrical
2023

Submitted By

Hamza Tanweer 288758

Project Supervisor

Miss Sobia Haye

ACKNOWLEDGMENTS

I want to extend my sincere gratitude to everyone who helped finish this thesis by letting them know how much I appreciate it. Above all else, I would like to convey my deepest appreciation to my supervisor, Miss Sobia Haye, for their crucial advice, motivation, and assistance during this project. Their knowledge and opinions have been crucial in determining this project's course.

I am appreciative to the National University of Science and Technology for providing the facilities and resources required for the successful completion of this project. Modern labs, applications, and equipment have been essential for carrying out studies and collecting data.

I want to convey my deep appreciations to my family and friends for their constant support, compassion, and inspiration throughout this project. They have consistently inspired me with their support and confidence in my abilities.

Finally, I'd want to thank everyone who took part in this study and voluntarily gave their time and knowledge. Their participation has enhanced the research process and has been essential in producing insightful findings.

Without their combined efforts and assistance, this thesis wouldn't be possible. I genuinely appreciate their assistance in making this project a success; their efforts were crucial.

ABSTRACT

This project focused on an idea to develop an innovative smart helmet focusing on boosting traffic safety and fostering general well-being. This project is incorporating cutting-edge technologies into a helmet design, such as wireless microphones, and speakers for clear communication. The helmet also consists of a sophisticated accident detection system that makes use of cutting-edge sensors to quickly identify probable collisions and send out instant alerts (via SMS). The main objective of this was to come up with a solution to eliminate the use of mobile phones while driving. The multimodal design of the smart helmet has the potential to significantly lower the number of accidents and fatalities by raising the bar for driving safety and enforcing a more secure, sustainable environment for riders. As it promotes the use of protective gear and technological innovation for the benefit of society.

<u>SUSTAINABLE DEVELOPMENT GOALS</u>

The following Sustainable Development Goals are achieved in this project.

With its capabilities for accident detection and alarm generation, my smart helmet can contribute to increased traffic safety, fewer accidents, and better health and well-being.

My project integrates cutting-edge technologies into a helmet, such as wireless Bluetooth, a microphone, and speakers, advancing infrastructure and encouraging technological innovation.

My smart helmet can help make cities and communities safer and more sustainable by improving road safety through accident detection and alarm functions.

Safety features in my helmet promotes responsible consumption by encouraging people to use protective gear which in result reduces accidents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Acronyms	Abbreviations
IOT	Internet of thing.
CSS	Cascaded Style Sheet.
API	Application Programming Interface.
GPIO	General Purpose Input & Output pins.
NMEA	National Marine Electronics Association.
IDE	Integrated Development Environment.
VSCODE	Visual Studio Code.
.js	Pure Javascript File.
.ts	Pure Typescript File.
.tsx	Typescript file with jsx written in it (HTML).

Chapter-1

INTRODUCTION

1.1 Motivation

Bikes are the most common and cheap source of transportation in developing countries, but it can be risky to travel on bikes as they are keener on accidents. A survey done by[1] found that drivers of 2-wheeler vehicles are 27 times more likely to die in an accident than those who are riding a four-wheeler vehicle. In motorcycle accidents, the passenger death rate is roughly six times higher than in auto accidents. One of the main reasons for this is the use of mobile phones while driving. But using safety gadgets, especially helmets, is very useful for saving many lives, but many people do not wear them because many think helmets are useless apart from the safety features of a helmet. So, I came up with an idea to innovate it bringing most of the necessary features of a phone on the helmet using the Internet of Things. The uniqueness, smart features, and productivity it brings in their journey will motivate them to wear helmets.

This concept of wireless communication, navigation, and monitoring using different sensors is taken as motivation from highly sophisticated astronaut and fighter pilot helmets. Bringing this technology to bike helmets can bring both productivity and safety. In a research done by [2] author came to a conclusion that by 2030, there will be more than 29 billion Internet of Things (IoT) devices in use worldwide, up from 9.7 billion in 2020. Forecasts indicate that there will be more than 75 billion installed IoT devices worldwide by 2025. So, we will see more adaptability toward smart choices in the upcoming years. Some smart helmets are already available in the market with tons of features but are very expensive, over 200 dollars, and come with mostly useless features. So, we must come up with a solution that is affordable and has all the necessary features.

Considering all the above points, I have divided my project into two parts communication system, a control system that I have used to send sensor data to the cloud on a serverless database, which is then connected to an AI-based web application that gives the real-time location and sensor values, it is also used to detect accident based on my sensor values and generate alerts to notify related authority So, they can act in time.

Another reason for opting for this project was the extent of growth and its application in other fields, for example for construction workers and miners. We can add more and more features to it soon on the software side like machine learning-based detection and on the hardware side more health monitoring sensors, and it has great capability to become a unique product in the market.

1.2 Objectives

Our focus is on achieving three main goals:

1. **Seamless communication and navigation:** We must ensure that the journey of the rider is smooth and seamless. In most of the accident cases, we found that one of the main reasons is the use of mobile phones. For a normal rider, it can be avoided but in the case of delivery riders, it cannot be avoided. They must navigate different areas, attend calls, and use mobile phones to reach their destination in time. Our goal here is to eliminate mobile usage from the system. We have to make a system such that all the tasks our rider is performing on mobile phones are automated.

2. **Accident detection and Alert generation:** Another aspect of our project is to detect accidents. We have to make perfect algorithms and combinations of sensors that are cheap and efficient and predict with minimum error. After prediction, we have to develop a proper alert generation system that will generate in-time alerts so emergency aid reaches in time.

3. **Monitoring Application:** Software is an essential part of the Internet of Things systems. The user needs some level of abstraction. We have to develop a user-friendly application that is available on all platforms on desktops, and mobile phones (Android and IOS). The application should show all the sensor values and the location of the rider.

Summary:

In this chapter, our focus was to briefly introduce our project. We have discussed what was the motivation behind our project and what objectives and results we are looking to achieve.

I have described some recent trends in the field of IOT and smart devices. I have discussed why we need smart helmets as compared to normal and conventional helmets. I have discussed some models already available on the market and some working prototypes. I have discussed a simple plan of how we are going to divide and make our helmets. I have discussed our main objectives (Seamless communication, Accident detection, and monitoring application).

CHAPTER-02

BACKGROUND AND LITERATURE REVIEW

2.1 Background:

Safety and monitoring systems have significantly improved because of the Internet of Things (IoT) technologies' incorporation into a variety of fields in recent years. The creation of smart helmets, which make use of connectivity and sensor technologies to improve safety, offer real-time monitoring, and enhance user experience, is one famous example of an IoT application. These intelligent helmets incorporate cutting-edge features including accident detection, communication capabilities, environmental monitoring, and data analytics to alleviate the shortcomings of conventional helmets.

The main goal of IoT-based smart helmets is to reduce hazards and assure users' safety in a variety of situations, including sports, workplace environments, and transportation. These helmets can gather data in real-time by incorporating a variety of sensors, including accelerometers, gyroscopes, temperature sensors, and heart rate monitors. This allows for a proactive approach to safety. The gathered information can be analyzed to spot probable mishaps, follow users' locations, check vital signs, and spot dangerous situations.

Smart helmets powered by the Internet of Things provide many benefits over conventional helmets. In the event of an accident or dangerous circumstance, they can send users and emergency contacts immediate messages and alarms. Through integrated wireless modules, these helmets can also enable two-way communication, enabling users to place calls, send messages, and listen to audio without endangering their safety. Additionally, the incorporation of GPS technology makes it possible to track a user's whereabouts in real-time, which is useful for locating and helping people in situations.

But there are difficulties in creating IoT-based smart helmets. Given that the multiple sensors and communication modules in these helmets must operate continuously, power consumption is an important factor. To achieve longer battery life, effective power management systems and the usage of low-power components are essential. Another major worry is data security, as the helmets broadcast and gather private environmental and personal information. To protect user privacy, strong authentication, and encryption mechanisms should be used.

2.2 Literature Review:

I have gone through different research papers to sort out problems and requirements for my project Here are some significant conclusions from the literature:

1. In this work done by [3], current developments in the design of smart helmet systems are reviewed. The study's Smart Helmet technology has several uses, including preventing motorcycle accidents, quickly identifying bike crashes for human health, protecting miners from dangerous situations in the mining industry, and warning them about dangerous petrol emissions. The study also looks at how Smart Helmet systems have changed over time, especially as new technologies like the Internet of Things (IoT) have been incorporated. The study also discusses the usage of intelligent motorcycle helmet devices that alert motorcyclists to the presence of behind large vehicles or buses to avoid crashes.
2. A research done by [4] refusal of people to wear helmets is a major contributing factor in the current situation's high frequency of deadly two-wheeler accidents. The use of smart helmet systems has become more important to address this problem and guarantee human safety. These devices seek to both foresee and immediately identify motorcycle accidents. When compared to conventional helmets, the use of IoT-based technologies in smart helmets significantly improves two-wheeler rider safety. Globally, numerous improvements and uses for smart helmet technology have been created over time. This study focuses on categorizing and presenting a thorough assessment of the smart helmet technologies already in use, emphasizing their significance in enhancing traffic safety.
3. In a study done by [5] describes the creation of a smart helmet that will facilitate the handling of rescue worker accidents during emergencies. Despite the proliferation of IoT-based products and services, their application and integration have proven difficult due to their development in particular industries. The researchers have created a new software framework that effectively integrates a variety of hardware and services, facilitating resource management, to address this issue. This framework's smart helmet gathers, produces, and transforms data from a variety of sensors, including oxygen and motion sensors, infrared, electro-optical, and drone cameras. A Command Centre and a head-mounted display (HMD) are used to monitor the information that has been gathered.

4. This paper by [6] describes the creation of an intelligent helmet that is intended to help rescue personnel overcome difficulties they encounter in catastrophe situations. Although there are many IoT-based devices and applications available, their integration across several industries makes it challenging to introduce new devices, alter applications, and update services as needed. Our research team has developed a revolutionary software framework to get around this problem by seamlessly integrating a variety of hardware and services while effectively managing resources. We have created a smart helmet that can respond to safety incidents during disasters based on this framework. The smart helmet gathers information from a variety of sensors, including 6-axis inertial sensors, infrared cameras, electro-optical cameras, drone cameras, oxygen residual sensors, and smartwatches. Following processing, this data is shown in the Command Centre and on a head-mounted display (HMD). We rigorously tested our system with a simulator and produced data based on numerous scenarios to validate it. In addition, we tested all gadgets and services in actual settings to assess how well they functioned and performed.

These studies show the various ways that IoT-based smart helmets can be used across different sectors and domains. They draw attention to the helmets' potential to promote safety, real-time monitoring, and efficient communication. To address issues with power consumption, data security, and user acceptance, however, more study is required.

2.3 Summary:

A thorough overview of IoT-based smart helmets is provided in the literature review and background section. The literature review highlights several studies done on smart helmet applications in many fields, such as motorcycling, construction, and industrial safety. These studies demonstrate how smart helmets may enhance communication, real-time monitoring, and safety by integrating IoT technologies. The significance of IoT-based smart helmets in overcoming the shortcomings of conventional helmets by combining functions like accident detection, communication capabilities, environmental monitoring, and data analytics are explained in the background section. The advantages of smart helmets are emphasized, including real-time alerts, two-way communication, GPS tracking, and their potential use in commercial, athletic, and transportation settings. The section also discusses issues with data security and power usage. Overall, the history and literature analysis gives a strong foundation for understanding the function and promise of IoT-based smart helmets in boosting user experience and safety.

CHAPTER-03

METHODOLOGY

In this chapter, I am going to discuss details about what resources and tools are in our project. We are going to discuss detailed implementation. In the end, we will integrate the complete software and hardware part of our project.

3.1 Resources and Tools

In this section, I will tell you what sensors, frameworks, and databases I am going to use. I will explain detailed working and reason for the choice.

3.1.1 Hardware Resources

I have divided the hardware of my project into two parts: the control unit which is used to manage and upload all of the sensor's data and the communication unit which is responsible for wireless communication with your mobile phone. I will be explaining each device used one by one.

3.1.1.1 Controller:

In this project, I have used "**ESP-8266**" a famous WIFI-based microcontroller from the Arduino family as my microcontroller. In this section, I am going to tell you about some specifications and reasons for choice.
It is a 32-bit microcontroller that is decent enough to operate almost all IOT-based tasks. Which provides WIFI connectivity. It provides 17 GPIO pins of which you can use 11 pins at once. It is one of the most efficient controllers and operates only on 3.3 V.
The following are some reasons for opting for this for my project.

1. It is affordable and available almost everywhere.
2. It is based on the Arduino family which means it is open source there is a large community to back it so we can use it easily for implementing any type of complex control using open-sourced and efficient libraries.
3. The size of this module is very small as compared to others available in the market it is smaller them even Arduino UNO.
4. It supports a wide range of networking protocols such as TCP/IP, UDP, HTTP, MQTT, etc making it easy to use for cloud-based applications.

5. We can easily update its firmware without even needing it to be physically connected to a computer.

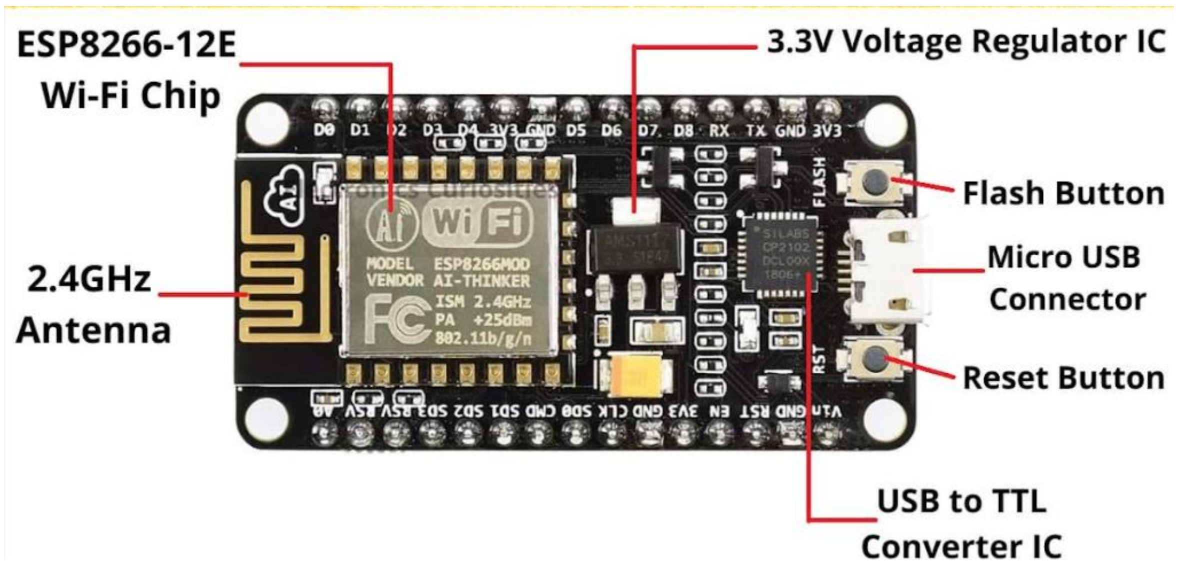


Figure 1:Labeled Diagram of Esp 8266 microcontroller.

3.1.1.2 GPS:

We have used "**U-BLOX Neo 6M**" as our GPS receiver. It is one of the most popular modules available in the market. Here are some reasons why I prefer it over others.

1. The level of accuracy is provided due to its connectivity with multiple satellites which includes the Global Positioning System (American satellite system based on 24 satellites) and Global navigation satellites system (satellite system based on 24 satellites famous for its accuracy of almost 4-6 cm) and Galileo (European union satellite system based on 30 satellites) which ensures the reliability and precision of location data.
2. The level of sensitivity allows it to acquire signals in any sort of challenging, harsh environments, and low signal conditions.
3. The recovery time (Time when the signal is lost due to power off or any other situation and recovered back) of this module is minimum if compared with other sensors.
4. The compact size of this module is also one of the reasons I am using this in my project because it is easy to install and lightweight to carry.
5. Power consumption also makes it one of the most suitable options to be used because

it does not take any external power, the power from ESP 8266 is enough.

Now, I will tell you something about the work of this module. As discussed above, it provides a location from multiple satellite systems in the form of exact time and position. The data received in the form of signals are refined to extract important information.

Then it calculates its most precise location from trilateration, which involves measuring the distance between the satellite and the module based on the signal travel time. As the location is determined it outputs the data using a serial interface to any external device like a microcontroller in our case.

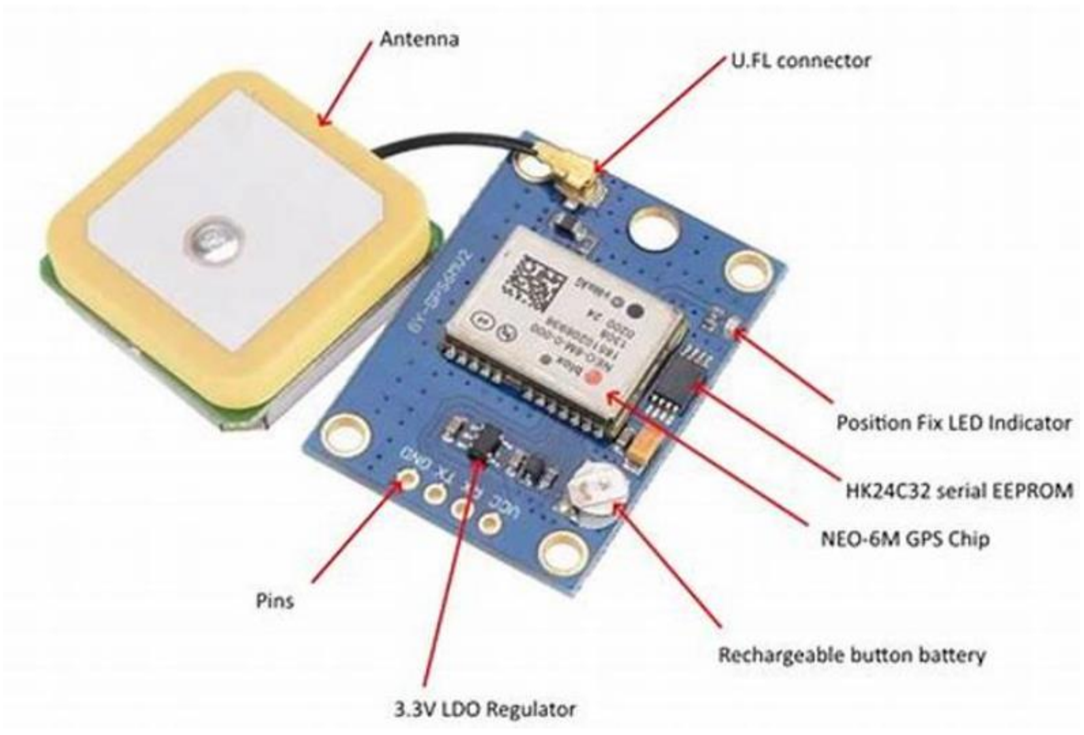


Figure 2:Labeled Diagram of Neo6m GPS module.

3.1.1.3 Vibration Sensor:

This is the main sensor and plays the most vital role in accident detection. We are using SW420 for shock detection. It is the most famous sensor for shock detection because of its level of sensitivity against vibrations. Also, it is easily available in the market at a very affordable price.

A little metal ball is secured in place by a metal spring, which makes up the SW-420 sensor. The metal ball moves when the sensor is subjected to vibration or stress, altering the electrical conductivity between the sensor's two pins.

The sensor outputs a digital signal that is either HIGH or LOW depending on whether there are vibrations or not. The voltage across the two pins is compared using a built-in comparator. The output changes from LOW to HIGH depending on whether the voltage reaches a predetermined threshold.

The sensitivity of some SW-420 sensor versions can also be changed using a potentiometer. You can adjust the sensitivity of the sensor to meet the needs of your particular application by twisting the potentiometer. By connecting the sensor's digital output pin to a microcontroller's digital input pin, the SW-420 sensor can be quickly and simply connected to a microcontroller board, such as an Arduino. The microcontroller may then read the digital signal and, based on the measured vibrations, initiate the necessary steps.

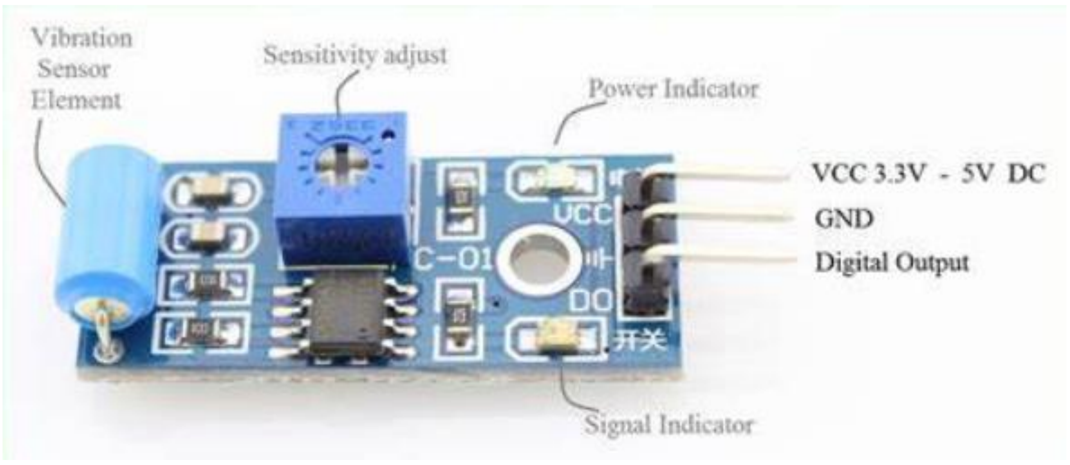


Figure 3:Labeled Diagram of SW-420 (Vibration Sensor).

3.1.1.4 Audio Communication System:

To eliminate the mobile phone, we must first identify what type of task the user is trying to perform using a mobile phone. Mostly they are attending calls, navigating through maps, and sometimes reading messages. You can perform all these by using a wireless headphone system. We can use the voice assistant built into our phone to control our phone we can attend calls, navigate through maps (with maps narration feature), etc. This provides riders with a very enhanced user experience and aesthetic feel.

3.1.2 Software Resources

In this section, I am going to describe all the software development resources I have used in my project resources (web framework, User Interface, APIs, and other services).

3.1.2.1 Web Framework

In this section, I am going to discuss the application side of my project. I have used "**NEXT.js**" as a framework for my application. It is a react-based framework that gives you almost all the support and tools required for my application.
There are many reasons for choosing this as a framework because of its use of new technologies: Number one is Server-side rendering means that it can render pages on the server side and immediately send them to the client side when there is a call request rather than using client-side and resources to render the page which could decrease the rank of a website Number two is static site generation which means that the pages that are rendered on the server side have their static copies already built on build time rather than on runtime which further reduces the server load.
Now you might be wondering why we need these two technologies in this App. At this time there might be no need for these technologies but when the product will grow our App will be heavily loaded servers will be overburdened with client requests So, if we have used both then most of the processing power will be saved and it will not let the server crash and does not make our application slow.

3.1.2.2 User Interface:

For any production application appearance is one of the most important parts. It brings new customers and retains old ones. Like my state-of-the-art framework I have used the latest styling library in it "**Tailwind CSS"** We can use other libraries too, but it is very easy to adapt. My motivation for using this library was the level of customization it provides, and we can write it without leaving our HTML which decreases our burden to maintain CSS files later on while updating the application.

3.1.2.3 API:

In this application, we have used only one API service TWILIO (which is used to send authentication, alert messages, and calls, etc.) for sending alert messages to relevant parties. It is a leading service and is used by many other companies.
There is no obligation or emergency to use this service we can use other services for message sending but it gives student testing packages as compared to others So, I have used it and I recommend it to you for testing if you are a student.

3.1.2.4 Database:

My smart helmet project uses Firebase as its main database since it offers a large number of features and functionalities that simplify data storage, retrieval, and synchronization. Critical sensors data and user data may all be seamlessly synchronized across a variety of platforms and devices thanks to its real-time database. All stakeholders (family members, company, emergency services providers) will have immediate access to the most recent information thanks to this real-time capacity, enabling them to act quickly in case of an emergency.

Additionally, Firebase provides strong security features including integrated authentication systems and granular access management. These features safeguard private user data and guarantee safe database access. Our smart helmet system's functionality and performance are improved thanks to Firebase's integration capabilities with a variety of services and platforms, including Next.js. Additionally, it provides scalability, flexibility, and load balancing for optimum performance to handle growing data volumes and user interactions.

The simple data organizing and querying features of Firebase allow for effective data organization and retrieval, giving us flexibility as our project develops. Overall, Firebase's real-time capabilities, strong security, smooth integration, scalability, and flexibility make it the perfect database resource for our smart helmet project. We can confidently develop a dependable and effective smart helmet system that improves traffic safety and satisfies user needs by utilizing Firebase.

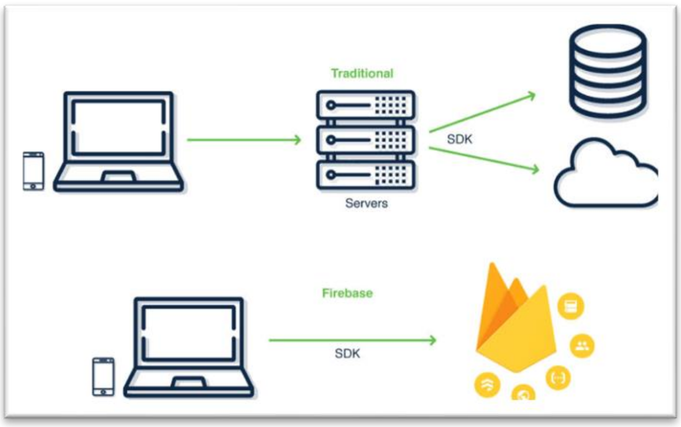


Figure 4: Firebase vs Conventional Architecture

3.2 Implementation

Now I am going to tell you to step by step detailed implementation of my project. Previously, As I have discussed I have divided the project into two parts I will discuss them separately.

3.2.1 Hardware Implementation

In this section, I am going to discuss the hardware implementation of my project. The hardware is divided into two parts a control unit and an audio communication part.

First, I am going to discuss the control unit or rather I should say the heart of my project. It consists of **“ESP-8266”** as a controller, **“Neo 6m”** as my GPS, and a combination of two **“SW-420”** vibration sensors. The specifications of all these modules are explained in section 3.1.1.

First, I will be discussing the circuit diagram of the complete circuit.

I have used **“Node-MCU”** or **“ESP-8266”** due to its Wi-Fi connecting compatibility. I am using Arduino IDE to program the microcontroller and C++ as my programming language. For testing purposes, I am using my laptop to power it will help me power it and monitor all the sensor values using a serial monitor.

For good debugging, I have tested each component one by one. So first connecting vibration sensors I have discussed the detailed working of the sensor in section 3.1.1. I have used a combination of 2 vibration sensors, one will be on the left side, and one will be on the right side to gain more accuracy.

The sensor has three pins (two pins are power (VCC and ground) and the third pin is on when there is vibration and off otherwise) and two LEDs (one of them is a power indicator and the Second is for indicating vibration).

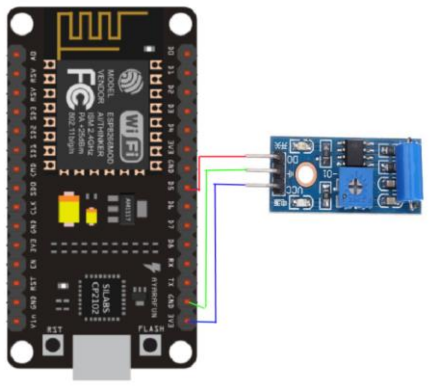


Figure 5:Vibration Sensor and Esp 8266 Interfacing

In the circuit, I have connected both vibration sensors to pins **D0** and **D7,** and in the code I have used the "**digitalRead()**" command to read the pins of both pins. We are reading data and uploading it to the cloud.

Burn the firmware a test your vibration sensor by touching or firmly hitting it. For observing the output, you can see the led it will turn off when there is a possible contact you can also check it on your serial monitor built inside of your IDE.

For the location, I am using "NEO 6M" It has four pins two of which are for power and the other two are for serial communication **"Tx"** and **"Rx"** pins, and an indicator LED which blinks if it receives a signal from the satellite. I have already explained the working of the sensor in section 3.1.1. I powered the sensor using the microcontroller and connected the serial pins to GPIO 4 and 5 of my controller.

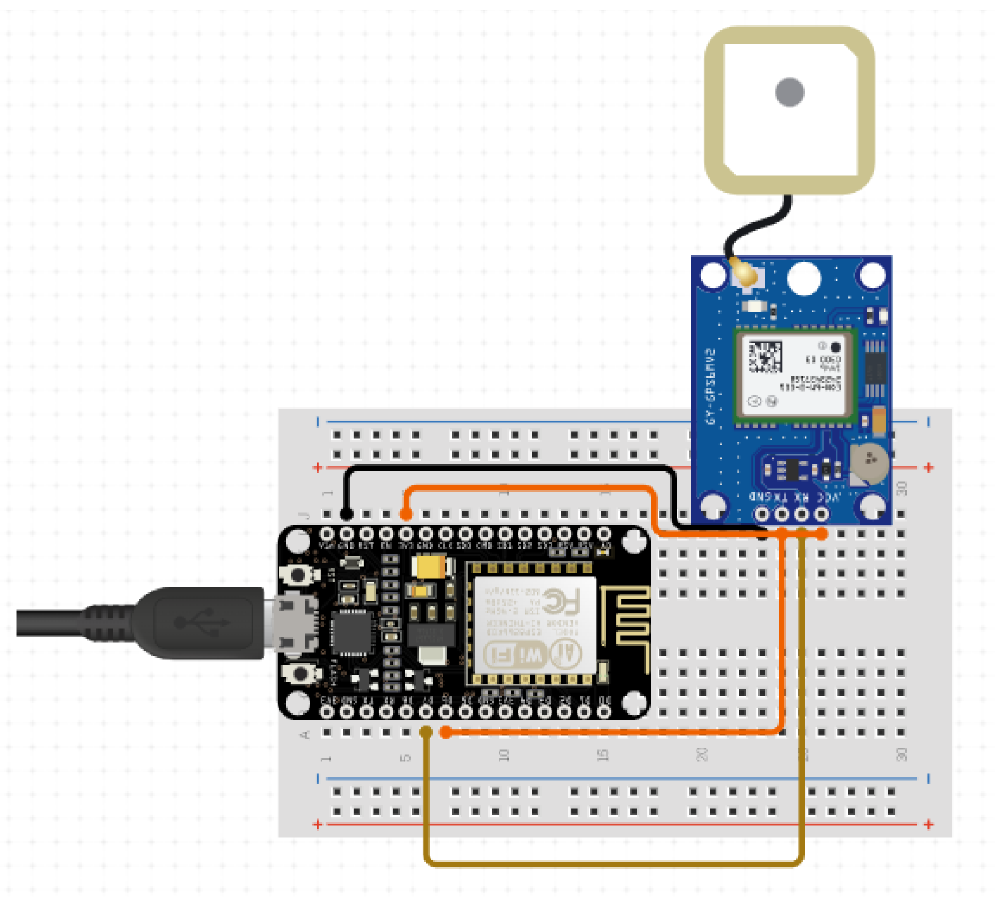
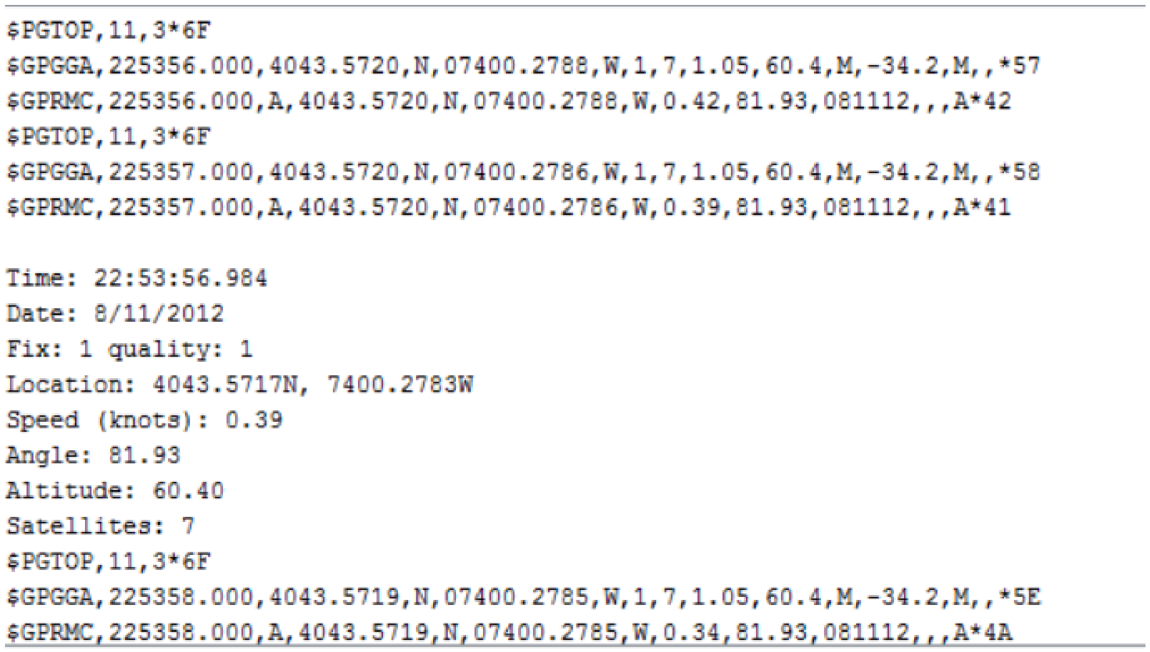


Figure 5:Interfacing of GPS and Microcontroller.

To begin with this sensor or see if it is working you will make connections as mentioned above in the program you should write "Serial. write(GpsSerial.read())". Command:" GpsSerial.read" will be used for reading the input message passed by the GPS. And then write the message on the serial monitor so we can read it.

Brun the firmware on the controller you will see a few lines of message consisting of some numbers. The output message is in a special format called "NMEA". You should see the LED on the module if it is blinking. Because if it is not blinking you might be receiving wrong information or no information.



```
$PGTOP,11,3*6F
$GPGGA,225356.000,4043.5720,N,07400.2788,W,1,7,1.05,60.4,M,-34.2,M,,*57
$GPRMC,225356.000,A,4043.5720,N,07400.2788,W,0.42,81.93,081112,,,A*42
$PGTOP,11,3*6F
$GPGGA,225357.000,4043.5720,N,07400.2786,W,1,7,1.05,60.4,M,-34.2,M,,*58
$GPRMC,225357.000,A,4043.5720,N,07400.2786,W,0.39,81.93,081112,,,A*41

Time: 22:53:56.984
Date: 8/11/2012
Fix: 1 quality: 1
Location: 4043.5717N, 7400.2783W
Speed (knots): 0.39
Angle: 81.93
Altitude: 60.40
Satellites: 7
$PGTOP,11,3*6F
$GPGGA,225358.000,4043.5719,N,07400.2785,W,1,7,1.05,60.4,M,-34.2,M,,*5E
$GPRMC,225358.000,A,4043.5719,N,07400.2785,W,0.34,81.93,081112,,,A*4A
```

Figure 6:NMEA message of NEO6M

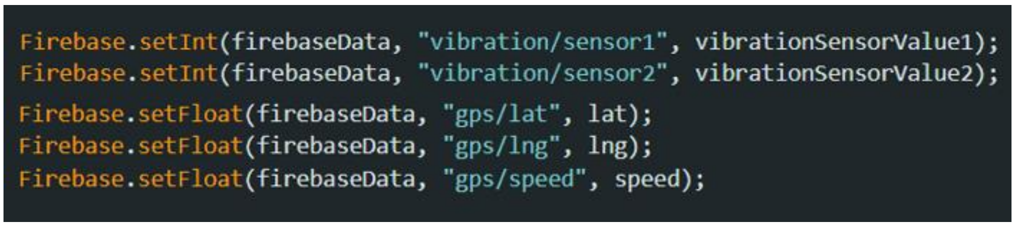
As you will observe the message consists mostly of insignificant information according to the context of our project so I have to extract it by parsing the output message we can do this both manually and by using the library. I recommend that the use of a library is an efficient technique because it will make efficient use of resources as in these cases of application our goal is to use the resources efficiently.

Here I am using the **"TinyGps++"** library for parsing NMEA messages. We just need to extract speed, longitude, and latitude. This library comes up with a vast variety of functions we can calculate everything in different units. Here I am measuring speed in kilometers per hour, longitude, and latitude in degrees up to six decimal accuracies using the following functions:

Now, after doing all this testing I have connected all the components on a single breadboard and merged the firmware code for both vibration sensors and the GPS module. After that, I changed my code a little bit to monitor the GPS and vibration sensor values on the serial monitor.

I have successfully extracted all the information. Now the next phase begins where uploading of data from the cloud takes place. In this project, I am using Firebase for which

I am using the ‘FirebaseESP8266.h’ library. I have passed the credentials of my real-time database project (The creation of our project will be explained later in the software section). And then the Firebase object is created in our project.



```
Firebase.setInt(firebaseData, "vibration/sensor1", vibrationSensorValue1);
Firebase.setInt(firebaseData, "vibration/sensor2", vibrationSensorValue2);
Firebase.setFloat(firebaseData, "gps/lat", lat);
Firebase.setFloat(firebaseData, "gps/lng", lng);
Firebase.setFloat(firebaseData, "gps/speed", speed);
```

Figure 7: Microcontroller Code Snippet to upload data on Firebase.

3.2.2 Software Implementation

As mentioned, the above software is the brain of an IOT system so, I have developed a **“NEXT js”** based web application. The reason for choosing web application was to make cross-platform compatible without getting in the headache of writing and managing different codes for different platforms. In this section, I am going to explain my app design, the libraries used, and working.

3.2.2.1 Folder Structure and Code:

First I will be describing the folder structure in **“NEXT js”** This is important to set up and code your project. A folder structure for your project is provided by the React framework called Next.js. Most of the folders are not for our use the main directory in the framework is the **app directory.** All of my code is written there in this directory. The second directory is the public directory where we save all the resources like images videos etc.
The app directory consists of all the code following folders and files are located there:

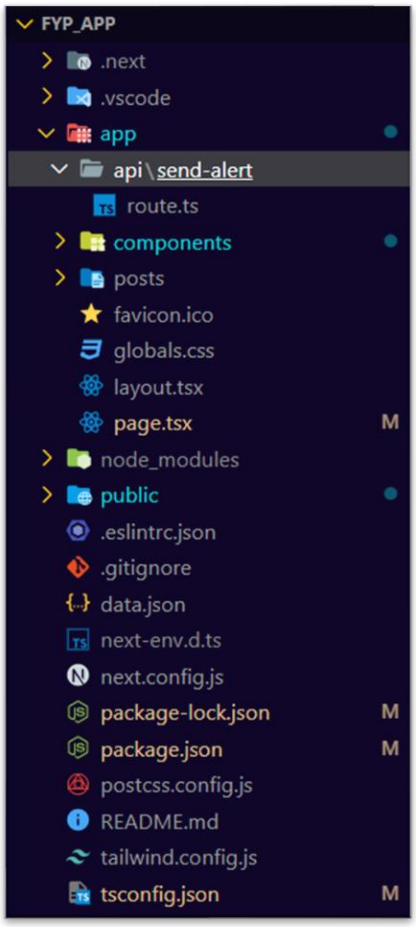
api: This folder consists of all the backend codes you can create APIs in it. I created a folder name **send_alert** and inside it, I created a folder name **route.ts** this is the standard way of creating an API route in this framework**.** In this file, I have created a POST API where I will receive data from my microcontroller.
components: In the components folder I developed all the react components for example my div components the speedometer, the navbar and the sensor grid etc.
posts: This is a custom folder that I have created myself. This is thought of as a good practice so you do not mess up the folder structure if there is a large no of pages in your application. In my case, I wanted to create an About page for my application. I created a

folder named **“About”** in the posts folder and in the About folder I created a file named **page.tsx** in the About folder as this is the standard for creating a page in this framework. Now I will be explaining the coding files available in the app directory.

page.tsx: This is the main file or Home page of the web app when ever the app is launched this is the page that is rendered on your screen. In my case, I created all the components in the components folder and imported them on the home page to keep the code clean and debugging easy. This is also taken as a good practice to make debugging easy and make the code efficient.

layout.tsx: This is the file that contains all the global components of the app for example if a component is imported here then we will see it on all the apps. So, I have used it to import the navbar and set the background on this page.

global.css: In this file, we save all the CSS (it is a language used for styling the page). As mentioned above I am using tailwind css for styling my app. It allows us to write both global and cascaded css (css written within your tsx).

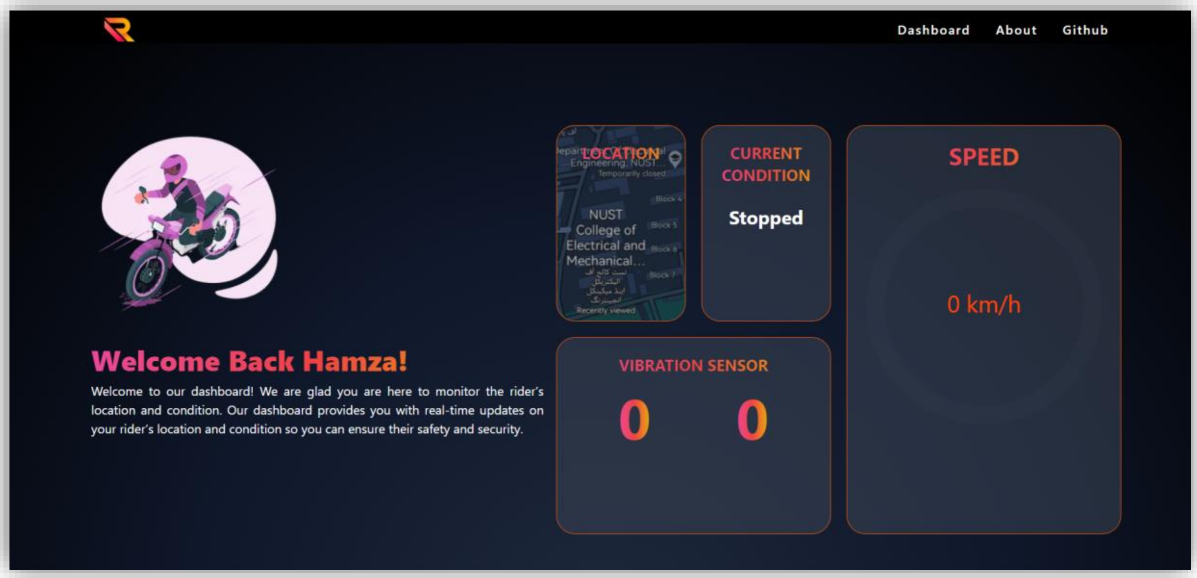


```
FYP_APP
  .next
  .vscode
  app
    api\send-alert
      route.ts
    components
    posts
    favicon.ico
    globals.css
    layout.tsx
    page.tsx                M
  node_modules
  public
  .eslintrc.json
  .gitignore
  data.json
  next-env.d.ts
  next.config.js
  package-lock.json         M
  package.json              M
  postcss.config.js
  README.md
  tailwind.config.js
  tsconfig.json             M
```

Figure 8:Folder structure of our app

3.2.2.2 User Interface:

In this section, I will be telling you about the user interface of my app. I have made a very simple layout at the top I have a navbar with three links of my home, about, GitHub (which has my GitHub link embedded in it), and logo on it. On the home page, I have two sections one giving us all the real-time values of the sensor and the second giving us a welcome message. On the About page, I have explained my app and an acknowledgment message to authorities.



Welcome Back Hamza!

Welcome to our dashboard! We are glad you are here to monitor the rider's location and condition. Our dashboard provides you with real-time updates on your rider's location and condition so you can ensure their safety and security.

Figure 9: User Interface of Web Application

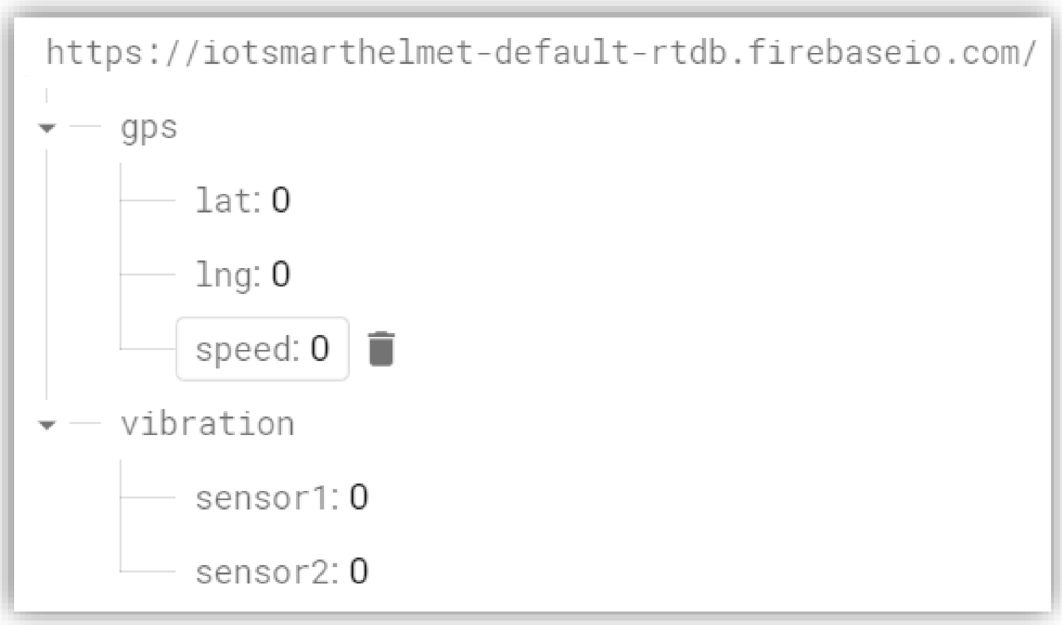
3.2.2.3 Database:

To store, manage, and synchronize crucial data, I am using Firebase as our database resource. The first, step is to connect our Next.js application to the Firebase database by integrating the Firebase JavaScript SDK into our application. Using Firebase's built-in capabilities, I set up the Firebase app, created the real-time database, and implemented user authentication.

The data structure in Firebase was then specifically created to meet the needs of our smart helmet system. In Firebase, we created collections and documents to categorize information like sensor data, and accident notifications. I secured effective data management and retrieval by establishing linkages between various collections, enabling simple access to information.

We made use of Firebase's real-time database feature to provide real-time synchronization.

As a result, we were able to instantly collect and store sensor data from the smart helmet. In response to data changes, Firebase updated all associated platforms and devices instantly. This real-time capability made sure that users and administrators alike had access to the most recent data regarding the status of the helmet.



```
https://iotsmarthelmet-default-rtdb.firebaseio.com/
▾ — gps
      ├── lat: 0
      ├── lng: 0
      └── speed: 0
▾ — vibration
      ├── sensor1: 0
      └── sensor2: 0
```

Figure 10: Data structure on Firebase Realtime Database

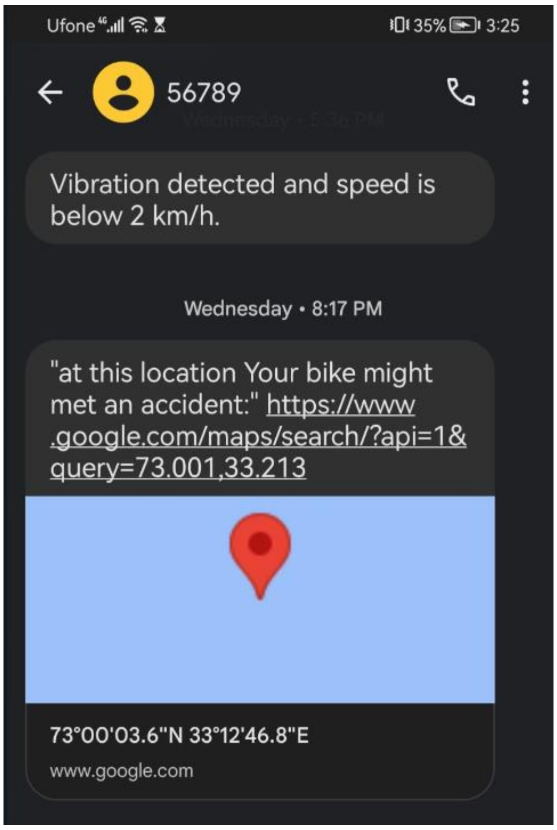
Additionally, Firebase's authentication procedures were extremely important in keeping our smart helmet system secure. Users had to authenticate themselves using the email/password authentication feature of Firebase or another way that was accepted. By doing so, sensitive data was protected, and overall security was improved by ensuring that only people with the proper authorization could access and interact with the system.

We successfully used Firebase's querying features to retrieve data from the database throughout the development. We ran queries to retrieve accident alerts for analysis and notification purposes and retrieve sensor data based on timestamps. These querying capabilities simplified data retrieval and made it possible to quickly access the needed data.

3.2.2.4 Alert System:

In this section, I will be explaining to you the alert system built into my app. I am using the **Twilio** messaging service for the alert system. For this, you will need a Twilio account because you need an authentication token to use it in your app. To get your token you need an account. You can search for it on Google and make an account on it. Account creation is free of cost, and we get fifteen-dollar credits for testing purposes.

After signing up you can generate your own phone number, account sid, and account authentication. This is required when creating an endpoint in your framework. I created a post-request endpoint where I passed the message as input which is sent to the relative body (family members). I later tested this endpoint using Postman.



Vibration detected and speed is below 2 km/h.

Wednesday • 8:17 PM

"at this location Your bike might met an accident:" https://www.google.com/maps/search/?api=1&query=73.001,33.213

73°00'03.6"N 33°12'46.8"E
www.google.com

Figure 11: Test message received on my mobile phone.

To summarize the software implementation of my app. It has increased our smart helmet system's usability and functionality. We created a strong and dynamic web application that smoothly interfaces with the hardware elements of the smart helmet by utilizing the extensive features of Next.js. The Next.js app enables users (companies or families) to monitor riders through real-time data visualization, user-configurable settings, and interaction with external systems. The Next.js app offers customers a seamless and captivating experience with an intuitive user interface and optimized performance, enabling simple interaction with smart helmet data.

3.3 Integration:

Project's hardware components and our Next.js application's successful integration marked a crucial turning point in the development of a complete smart helmet system. I have integrated the hardware and software components flawlessly, allowing them to coexist peacefully, through careful coordination and collaboration. The Next.js app had to be synchronized with the hardware sensors, and microcontroller, into the smart helmet as part of the integration process, which also entailed defining communication protocols, data exchange methods, and synchronization as discussed in section 3.2.2.1 I have created a REST API a post request which is receiving data from the microcontroller. Users can access critical information on the state of their helmets, sensor readings, and alarms thanks to this connection, which enables real-time data collecting, processing, and visualization within the application. Our smart helmet system's total efficacy, usability, and safety are improved by the integrated functionality of the software and hardware parts, making it a useful tool for promoting traffic safety and defending users' well-being.

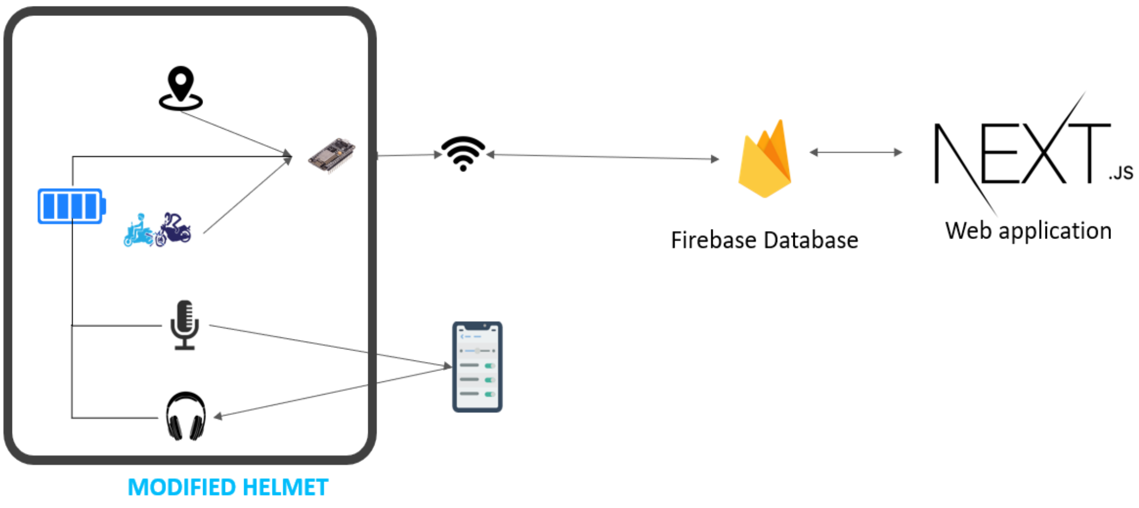


Figure 12:Block Diagram of Complete Integrated Project

3.4 Assembly:

In this section, I will be explaining the positioning of sensors inside the helmet. The main challenge was to come up with a design where sensors can give values with minimum error and maintain the safety of riders. There are two configurations possible either you can mount all the sensors inside the helmet or make a case in which all the sensors are first placed and then mounted on the helmet. In this project, I will be mounting sensors inside the helmet.

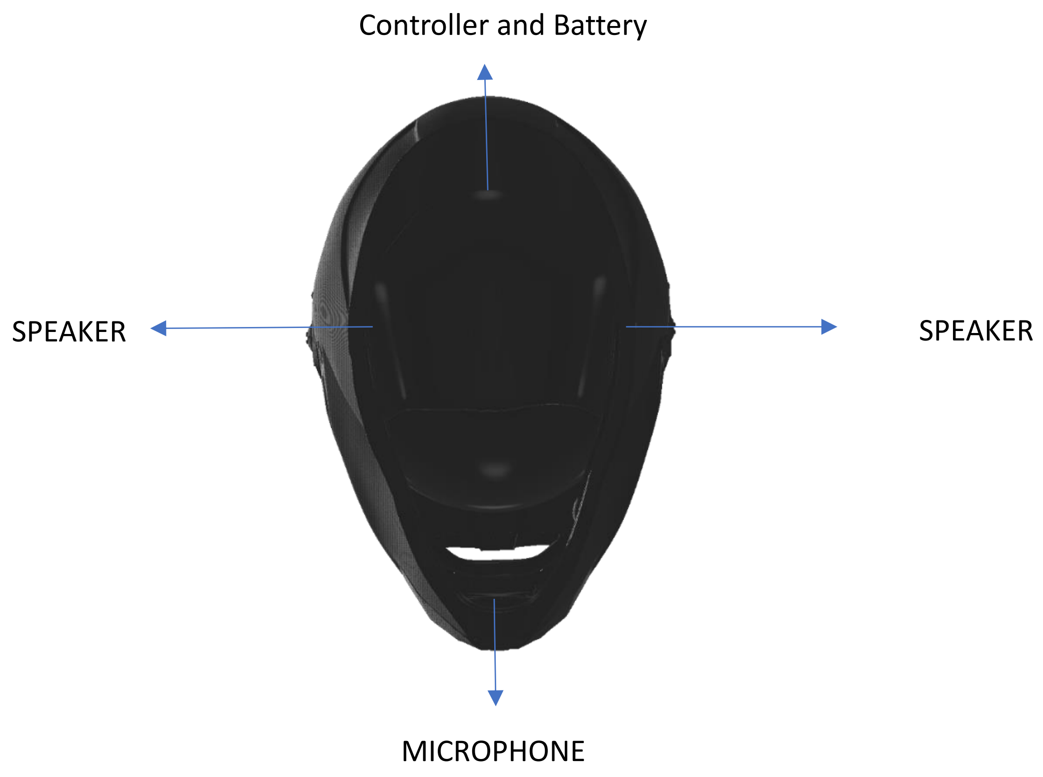


Figure 13:Placement of internal sensors microphone, speakers, controller, and battery

The above figure shows the placement of all the internal equipment placements.

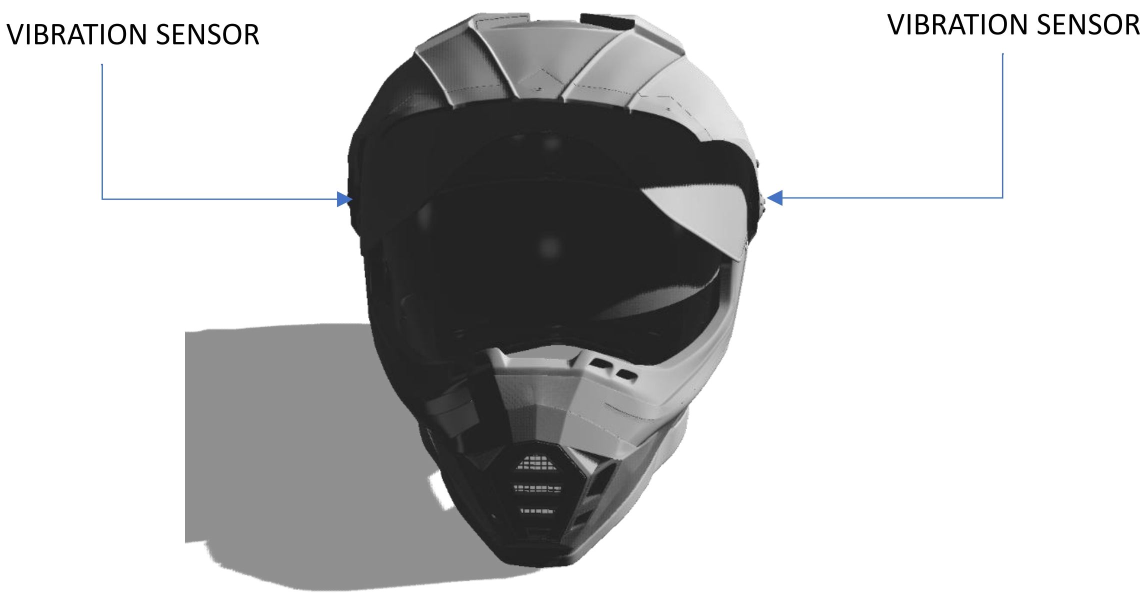


Figure 14: Placement of vibration sensors

Both vibration sensors will be placed on both sides of the helmet where helmet glass fits outside of the main shell. It is because in most cases in accidents, sides are impacted.

3.5 Summary:

The methodology chapter gives a thorough explanation of the strategy and all procedures used to complete this project. At the start, we explained all the resources and tools used for the completion of the project. All parts sensors, and software (GPS, vibration sensor, and microcontroller) are tested and compared with competitors available in the market. These tools and resources are carefully chosen and justified, showing their applicability and relevance to the project's needs. I also discussed why we have selected these tools.
After that implementation phase is covered in-depth, along with the step-by-step procedure for creating the smart helmet system. To do this, different hardware parts, such as sensors, microcontrollers, and communication modules, are separately programmed and then power management systems must be integrated with sensors. The chapter illustrates the logical flow between these elements, emphasizing on what were the factors and choices available during the integration process to guarantee optimal fit, functionality, and user safety.
One of the major parts of our project is describing software implementation. Concentrating on the creation of the Next.js application. In the folder structure and coding section, I

described software architecture, design principles, and coding techniques used to produce a reliable and user-friendly interface are explained. I have also described the application's integration with the hardware parts placing a focus on data synchronization, real-time monitoring, and user interaction. The chapter also describes any difficulties that arose during the implementation phase and the solutions created to address them.

In the integration section, I have explained how the final smart helmet prototype is put together, explaining how the hardware parts and software modules are combined. This comprises information on protocols (HTTP) and software and physical connections to ensure optimum alignment, comfort, and use. The chapter discusses any testing and validation procedures carried out to ensure the integrated system's usability, dependability, and safety.

To conclude, the methodology gives a thorough and organized breakdown of all the materials, equipment, implementation, integration, and assembly processes utilized in the creation of the smart helmet system. The chapter gives insightful information, and a roadmap for comprehending the procedures followed to turn the project concept into an actual, workable solution, highlighting the careful consideration, accuracy, and focus on details.

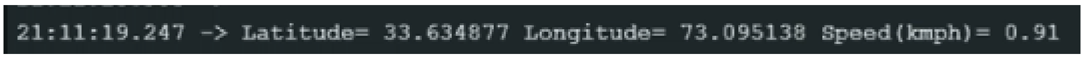
CHAPTER 4

RESULTS

In this chapter, I will be focusing on the tests and the final output of my project. I tested hardware and software separately and then integrated the whole project and then again tested the project.

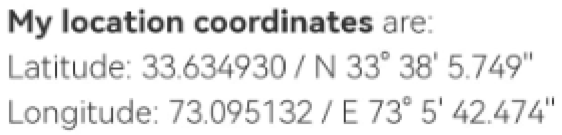
4.1 Outputs:

For hardware testing, I connected all the components separately and tested them with my microcontroller one by one. First I connected the vibration sensor with the microcontroller and read its value using digital read command and then gave it the different extent of shocks to it and observed the value. The value of the vibration sensor is either zero or one (one when vibration is detected).
Then I tested the GPS module and generated its parsed output using the "TinyGps++" library. Here only longitude, latitude, and speed are required. So I have generated only these values. In our GPS module, there is some amount of error greater as compared to mobile. This means GPS has a one to two meters greater radius than the mobile GPS. The below figures show the values of my GPS module and the values of GPS built into my phone.



```
21:11:19.247 -> Latitude= 33.634877 Longitude= 73.095138 Speed(kmph)= 0.91
```

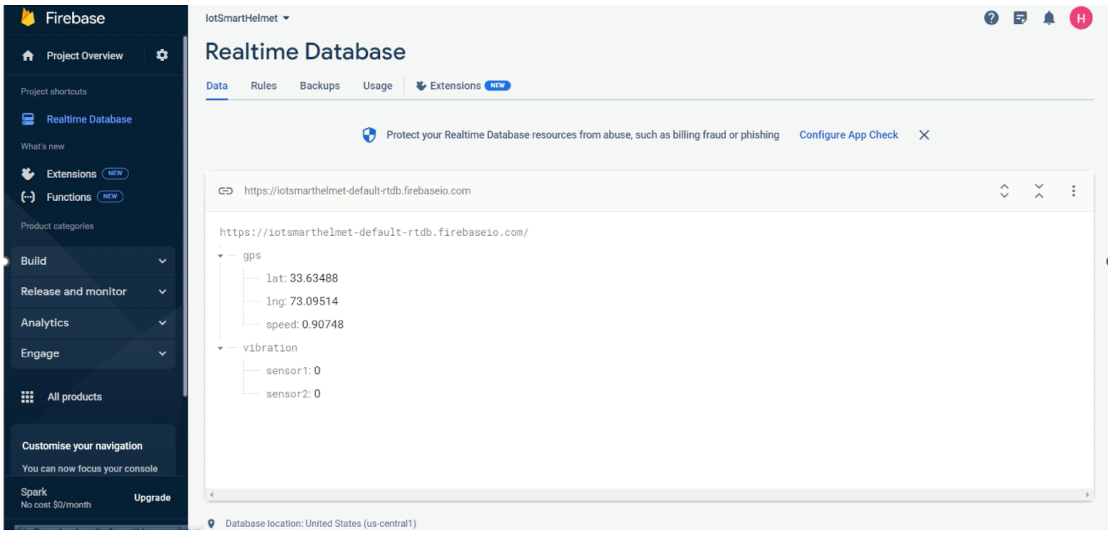
Figure 15: Output Values of the GPS module



My location coordinates are:
Latitude: 33.634930 / N 33° 38' 5.749"
Longitude: 73.095132 / E 73° 5' 42.474"

Figure 16: My Mobile GPS location using google maps.

For application testing, I used Postman for endpoints and VScode for testing, debugging all the code, and then deployed the website on Vercel which was deployed without any errors which means there are no syntax errors while there is a possibility of runtime errors and crashing which are sorted when the application is used by end-users using more scaled servers and customers feedbacks. Till now all the production side bugs are resolved. In my case, I have only one endpoint “send_Alert” the test of which is shown in section 3.2.2.4. Then I integrated the whole project by first uploading the code to send sensor values to Firebase. The procedure of making, connecting, and sending data to a Firebase real-time database is already shown in section 3.2.2.3. Microcontroller was programmed to update this data after every 5 seconds. The below image shows us the uploaded data.



```
https://iotsmarthelmet-default-rtdb.firebaseio.com/
▾ gps
    lat: 33.63488
    lng: 73.09514
    speed: 0.90748
▾ vibration
    sensor1: 0
    sensor2: 0
```

Figure 17: Image of uploaded data at that instant.

This data then gets updated on my app. I have also tested it by giving a hit to one of my vibration sensors and it was showing value on my application, and for the location, we can see it if we click on the location box it will redirect us to a google maps page where we can see the current location of the rider. The below image will show us the uploaded data.

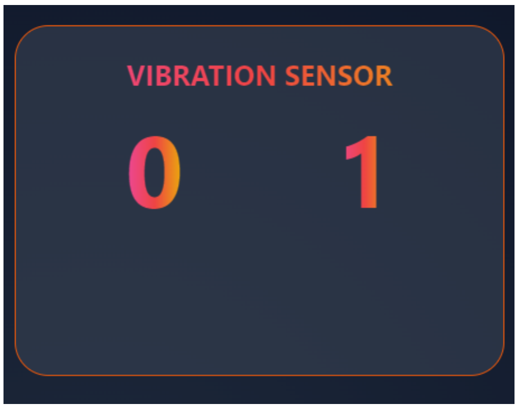


Figure 18:Realtime updated vibration sensor values on the dashboard

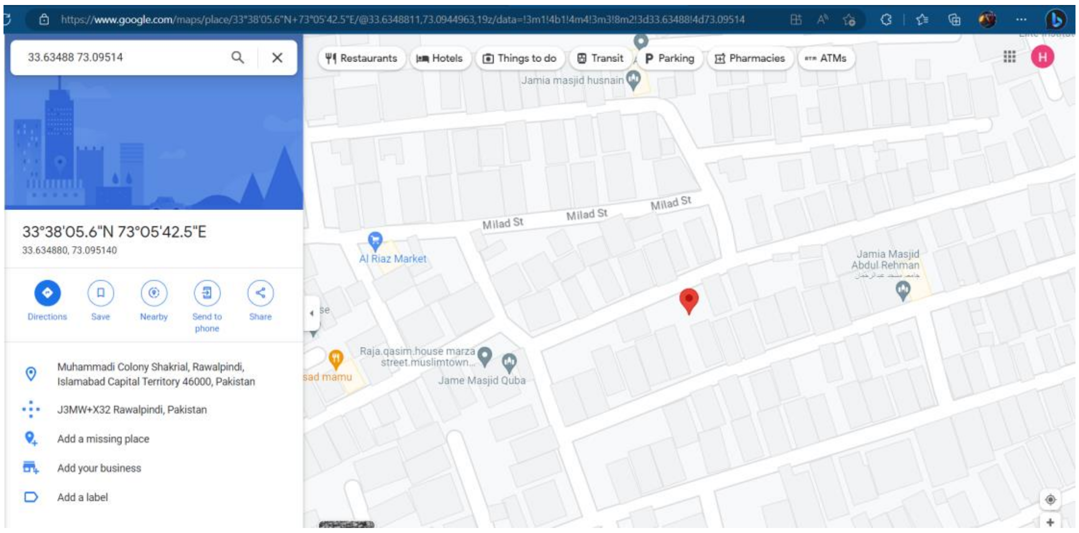
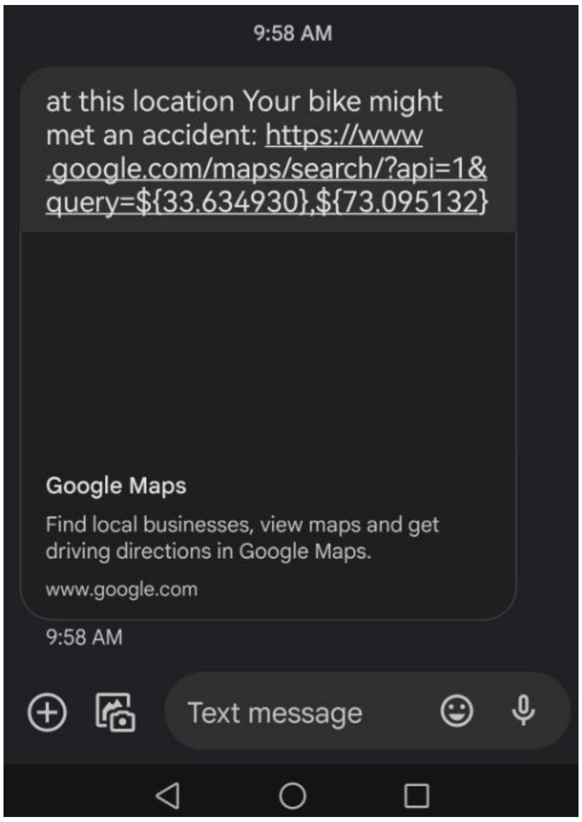


Figure 19: Updated location query

For accident alerts, I wrote conditional statement-based code which will check vibration sensor values and then the speed if any of the vibration sensors are on and the speed is zero it will call a function called send alert which will send a post request to my endpoint and a text message will be received to recipient user want to enter.



9:58 AM

at this location Your bike might met an accident: https://www.google.com/maps/search/?api=1&query=${33.634930},${73.095132}

Google Maps

Find local businesses, view maps and get driving directions in Google Maps.

www.google.com

9:58 AM

Text message

Figure 20:Message received on my phone

4.2 Summary:

In conclusion, the examination of our helmet solution has shown its efficiency in preventing and responding to accidents. The technology continuously tracks accelerometer data to precisely detect accidents, and it quickly creates notifications that are sent to emergency contacts via a cloud-based service. By integrating GPS technology, it is possible to follow a person's exact location and respond to emergencies quickly. User feedback verifies the system's usability and usefulness, and comparative analysis demonstrates that it outperforms competing systems. These outcomes demonstrate how our smart helmet project can improve traffic safety and emergency response. There is still room for improvement in terms of system performance, functionality expansion, and the investigation of new safety and security uses.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this chapter, I will be concluding all of my work and give some future recommendations if someone wants to pursue this project in the future. I was motivated to continue to improve road safety and reduce the dangers connected with accidents. This project has successfully created a comprehensive and creative smart helmet system that incorporates cutting-edge technologies. I developed a robust and efficient solution that offers real-time monitoring, accident detection, and warning production capabilities by integrating hardware parts, such as sensors, microcontrollers, and communication modules with a user-friendly Next.js application.

One of the objectives of my project is to encourage the use of protective gear and lower the likelihood of accidents, for this kind of work smart helmet system has proven to be effective in promoting responsible consumption. The system has demonstrated its capacity to quickly identify probable accidents and promptly warn relevant parties, enabling early aid and involvement in emergency circumstances. These capabilities include accident detection algorithms and SMS alert generating utilizing the Twilio API. This enhances not only the safety of motorbike riders but also people's general well-being and quality of life.

5.1 Future Work:

In this section, I will be giving some future recommendations for this project. This initiative has brought the use of Internet of Things (IoT) technologies one step forward in the area of traffic safety. I have created a scalable and adaptable smart helmet system that may be further improved and expanded in the future by utilizing IoT and cloud computing concepts and integrating various hardware and software components. The system's capacity for real-time data collection and analysis creates opportunities for us to come up with a more advanced Machine Learning or Deep learning-based solution to make correct predictions. Introducing additional features like, live traffic updates, and voice-based navigation, which can give motorcycle riders a more comprehensive and individualized riding experience.

1. First, we can use machine learning techniques and give the system more training data will help to improve the accuracy and dependability of the accident detection algorithm. This would improve the system's ability to distinguish between real

accidents and false alarms, which later helps in decreasing the frequency of pointless notifications.

2. We can also add functions like identification of faces, blood pressure monitoring, and identification of fatigue or any emergency health condition to a smart helmet system that might provide another layer of safety and well-being for motorcyclists. If we can detect these features we can enable a proactive intervention system and advance general health and safety by assisting in the detection of indications of fatigue, anxiety, or other physiological circumstances that may raise the likelihood of accidents.
3. This system can also be made more effective if we link it with smart city structures and response systems for emergencies. So, If there is a case of an emergency or crash, this device would allow seamless communication as well as synchronization between our device, traffic control systems, and rescue personnel, enabling quicker responses and coordinated actions.
4. The aesthetics, functionality, and applicability of the smart helmet system could also be improved if we undertake comprehensive testing for users, surveys, and feedback collection from various stakeholders, such as bikers, first responders, and safety experts. By putting the requirements and preferences of the system's intended users first, this user-centered approach would guarantee that the system is widely adopted and accepted.
5. We can also take the same concepts and apply them to helmets used in other fields such as by construction workers, miners, and firefighters adjusting our sensors according to their requirements

5.2 Summary:

To summarize and conclude, this project has not only created a useful smart helmet system but also provided the groundwork for further investigation and advancement in the area of traffic safety. IoT, artificial intelligence, machine learning, and real-time connectivity and stats monitoring systems are just a few of the cutting-edge technologies that can be integrated to make roads safer and lower the number of accidents. Smart helmet technology can be developed, standardized, and eventually commercialized with sustained work and collaboration, which will have a substantial impact on future road safety and save many lives.

REFERENCES

[1] R. Bodine. "Motorcycle vs. Car Accident Statistics " AutoInsurance.org. https://www.autoinsurance.org/motorcycle-vs-car-accidents/ (accessed.

[2] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020–2030," in *2020 Fourth World Conference on smart trends in systems, security and sustainability (WorldS4)*, 2020: IEEE, pp. 449-453.

[3] N. Divyasudha, P. Arulmozhivarman, and E. Rajkumar, "Analysis of Smart helmets and Designing an IoT based smart helmet: A cost effective solution for Riders," in *2019 1st International Conference on Innovations in Information and communication Technology (ICIICT)*, 2019: IEEE, pp. 1-4.

[4] A. Pangestu, M. Mohammed, S. Al-Zubaidi, S. H. K. Bahrain, and A. Jaenul, "An internet of things toward a novel smart helmet for motorcycle," in *AIP Conference Proceedings*, 2021, vol. 2320, no. 1: AIP Publishing LLC, p. 050026.

[5] M. Jeong, H. Lee, M. Bae, D.-B. Shin, S.-H. Lim, and K. B. Lee, "Development and application of the smart helmet for disaster and safety," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018: IEEE, pp. 1084-1089.

[6] S. Chandran, S. Chandrasekar, and N. E. Elizabeth, "Konnect: An Internet of Things (IoT) based smart helmet for accident detection and notification," in *2016 IEEE Annual India Conference (INDICON)*, 2016: IEEE, pp. 1-4.