DISASTER GUARD FOR DISASTER MANAGEMENT

A PROJECT REPORT

Submitted by

MOHAMMED FOUZAN	(730920104069)
MUHAMMED RAZAL	(730920104072)
PRATHYUSH PAVITHRAN	(730920104084)
VAISHNAV A	(730920104119)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

EXCEL ENGINEERING COLLEGE(AUTONOMOUS)

KOMARAPALAYAM - 637 303

ANNA UNIVERSITY::CHENNAI - 600 025

APRIL 2024

EXCEL ENGINEERING COLLEGE(AUTONOMOUS)

KOMARAPALAYAM - 637 303

ANNA UNIVERSITY::CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**DISASTER GUARD**" is the bonafide work of "**MOHAMMED FOUZAN (730920104069)**, **MUHAMMED RAZAL FARDEEN (730920104072)**, **PRATHYUSH PAVITHRAN (730920104084)**, **VAISHNAV A (730920104119)** "Who carried out the project work under my supervision".

SIGNATURE	**SIGNATURE**
Dr. P. C. SENTHIL MAHESH M.E., Ph.D.,	**Dr.A.T.RAVI M.E., Ph.D.,**
HEAD OF THE DEPARTMENT,	**SUPERVISOR,**
Professor,	Professor,
Department of CSE,	Department of CSE,
Excel Engineering College,	Excel Engineering College,
Komarapalayam-637303.	Komarapalayam-637303.

Submitted for the University Examination held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who actuated it, without them it would never have into existence. To them, we lay word of gratitude imprinted within ourselves.

We wish our heartfelt thanks to our respected Founder and Chairman of Excel Group Institutions **Prof. Dr. A. K. NATESAN M.Com., M.B.A., M.Phil., PhD., FTA** and Vice Chairman **Dr. N. MATHAN KARTHICK M.B.B.S., M. H. Sc (Diabetology)** for allowing us to have the extensive use of the college facilities to do our project effectively.

We express our sincere gratitude and heartfelt thanks to the respected Principal **Dr. K. BOMMANNA RAJA M.Tech., Ph.D.,** for his encouragement and support to complete the project.

We would like to express our profound interest and sincere gratitude to the Head of the Department **Dr.P.C.SENTHIL MAHESH M.E., Ph.D.,** Department of Computer Science and Engineering for his encouragement and support to complete the project.

We are privileged to express our deep sense of gratitude to Project Supervisor **Dr.A.T.RAVI M.E.,Ph.D.,**Professor Department of Computer Science and Engineering who gave guidance and support throughout our work and made this as a successful project.

We would like our sincere gratitude and heartfelt thanks to our Project Coordinator **Mrs.J.OBURADHA M.E,**Assistant Professor, Department of Computer Science and Engineering for continuous help over the period and creative ideas for this phase of our project work.

Finally, we thank the Almighty, all my Staff Members, Parents, Friends and well Wishers for the moral support throughout the project.

ABSTRACT

The "Disaster Guard" project presents a comprehensive disaster management mobile app designed to empower individuals and communities in navigating the challenges of disaster preparedness, response, and recovery. With a focus on fostering resilience, the app integrates cutting-edge features, including real-time disaster alerts, location-based services, and a user-friendly emergency services directory. The inclusion of SOS and distress signaling, coupled with two-way communication channels, ensures swift assistance during critical moments. Leveraging data prediction and early warning systems, the platform enables proactive planning, enhancing the overall preparedness for potential disasters.

An innovative aspect of the "Disaster Guard" app is its offline functionality, ensuring seamless operation in areas with limited internet connectivity. This feature addresses a crucial gap in disaster management technology, allowing users to access vital information even in challenging network conditions. Moreover, the project emphasizes community engagement, encouraging active participation to collectively strengthen disaster resilience.

The documentation provides a detailed exploration of the app's architecture, implementation strategies, and testing procedures. Rigorous testing, incorporating real-world scenarios, guarantees the reliability and accuracy of the system. The iterative feedback loop from users ensures continuous improvements, setting the stage for a dynamic and responsive disaster management tool. The "Disaster Guard" project represents a significant stride towards empowering communities, fostering preparedness, and creating a robust foundation for disaster resilience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OFABBREVIATIONS

SOS	:	Save Our Souls
API	:	Application Programming Interface
GPS	:	Global Positioning System
IOT	:	Internet Of Things
VPN	:	Virtual Private Network
URI	:	Universal Resource Identifier
URL	:	Universal Resource Link
Auth	:	Authentication
Ref	:	Reference
Res	:	Resource

CHAPTER 1

INTRODUCTION

In an age marked by unprecedented environmental uncertainties, the importance of effective disaster management cannot be overstated. The frequency and severity of natural disasters, ranging from earthquakes and floods to wildfires and pandemics, necessitate a paradigm shift in how individuals and communities prepare, respond, and recover. This imperative has propelled the development of the "DisasterGuard" mobile application – a revolutionary tool designed to empower users and communities, enhancing their disaster resilience through a holistic and innovative approach.

The "DisasterGuard" app addresses the multifaceted challenges inherent in disaster management. It goes beyond conventional alert systems by integrating real-time disaster alerts and notifications, thereby providing users with timely and critical information. Informed decision-making is facilitated through location-based services, ensuring that individuals have the contextual information necessary for effective responses.

A key element of the app is its commitment to swift and efficient assistance. The inclusion of SOS and distress signaling capabilities ensures that users can quickly summon help when faced with emergencies. Furthermore, the two-way communication feature fosters a dynamic and collaborative response environment, allowing users to share vital information and coordinate efforts during crises.

1.1 PURPOSE

The "DisasterGuard" project aims to revolutionize disaster management by developing a comprehensive mobile application. With a focus on empowering individuals and communities, the app delivers real-time disaster alerts, location-based services, and an emergency services directory. The inclusion of SOS and distress signaling, along with two-way communication during emergencies, ensures swift assistance and effective coordination. Utilizing data prediction and early warning systems, the app facilitates proactive planning, while offline functionality guarantees accessibility in challenging connectivity scenarios. Emphasizing community engagement, the project seeks to create a resilient framework where collective contributions enhance disaster preparedness and recovery. Through these features, the "DisasterGuard" app strives to be a pivotal tool in transforming the dynamics of disaster response and fostering a culture of resilience.

1.2 SCOPE

The scope of the "DisasterGuard" project is comprehensive, encompassing various dimensions of disaster management to provide a versatile and adaptive tool for individuals and communities. The primary scope includes:

- Individual and Community Empowerment: The project focuses on empowering individuals and communities to take an active role in disaster preparedness, response, and recovery. By providing accessible and user-friendly features, the app aims to enhance the capabilities of users, fostering a sense of self-reliance.
- Real-Time Disaster Alerts and Notifications: The app's scope extends to delivering real-time disaster alerts and notifications. By leveraging timely information, users can make informed decisions and take immediate actions to ensure their safety and that of their communities.
- Location-Based Services for Informed Decision-Making: Incorporating location-based services widens the scope to offer contextual information. This feature ensures that

- users receive data relevant to their immediate surroundings, enhancing the precision of decision-making during disasters.
- Emergency Services Directory for Swift Access: The inclusion of an emergency services directory broadens the scope by providing users with swift access to essential contacts. This feature streamlines communication with relevant authorities and services during critical situations.
- User Feedback and Continuous Improvement: Emphasizing community engagement widens the scope to encourage active participation and contributions. Users are motivated to actively contribute to disaster resilience, creating a collective effort towards preparedness and recovery.

In summary, the "DisasterGuard" project's scope is ambitious, aiming to provide a versatile, user-centric, and community-driven mobile application that comprehensively addresses the challenges posed by disasters. Through a combination of features, the project seeks to establish a robust framework for disaster management that is adaptable to diverse scenarios and user needs.

1.3 PROJECT OBJECTIVE

The primary objectives of the "DisasterGuard" project are to empower individuals and communities for effective disaster preparedness, response, and recovery. The project aims to achieve this by delivering real-time disaster alerts, utilizing location-based services for informed decision-making, creating a streamlined emergency services directory, enabling SOS and distress signaling for swift assistance, and facilitating two-way communication during emergencies. The incorporation of data prediction and early warning systems ensures proactive planning, while offline functionality addresses challenges related to limited internet connectivity. A central objective is to promote active community engagement and contributions, fostering a collective approach to enhance overall disaster resilience. Through these objectives, the "DisasterGuard" project aspires to create a versatile and user-friendly mobile application that significantly improves the landscape of disaster management.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY 1

TITLE: Enhancing Community Resilience through Mobile

Author : Maria Rodriguez and Ahmed Hassan

Year:2017

Description:

Maria Rodriguez and Ahmed Hassan's paper, "Enhancing Community Resilience through Mobile Applications: A Systematic Review," delves into the role of mobile applications in bolstering community resilience, a critical aspect of effective disaster management. The authors systematically review existing literature to understand how mobile applications contribute to community resilience by fostering communication, disseminating critical information, and encouraging active participation. The survey assesses various mobile applications across different disaster scenarios and evaluates their impact on community preparedness and recovery. By synthesizing findings from diverse studies, the paper offers insights into successful strategies and potential areas for improvement in leveraging mobile applications for community resilience. This survey serves as a valuable resource for informing the design and functionalities of the "DisasterGuard" app to maximize its effectiveness in enhancing community resilience.

LITERATURE SURVEY 2

TITLE: Offline use of mobile application

Author : Priya Sharma and Rajan Patel

Year : 2018

Description

Priya Sharma and Rajan Patel's paper, "Offline Functionality in Mobile Applications: A Review of Strategies and Implementation Challenges," provides a comprehensive survey of strategies and challenges associated with enabling offline functionality in mobile applications. Given the importance of maintaining communication and accessing critical information during disasters where internet connectivity may be compromised, this review explores various techniques employed to ensure app functionality in offline modes. The authors analyze the effectiveness of these strategies and highlight potential challenges, offering valuable insights for the implementation of offline features in disaster management apps like "DisasterGuard." This literature survey contributes essential considerations to ensure the app's reliability and utility under adverse connectivity conditions.

LITERATURE SURVEY 3

TITLE: Two-Way Communication in Emergency Situations

Author : Laura Anderson and Vikram Verma

Year : 2019

Description

Laura Anderson and Vikram Verma's paper, "Two-Way Communication in Emergency Situations: A Critical Analysis of Mobile Applications," critically examines the role of two-way communication features in mobile applications during emergency scenarios. The authors conduct an in-depth analysis of existing mobile applications designed for emergency communication, evaluating their effectiveness in facilitating real-time interaction between users and emergency responders. The review emphasizes the importance of seamless and reliable communication channels for swift decision-making and coordination during disasters. Insights from this literature survey contribute valuable considerations for implementing robust two-way communication features in the "DisasterGuard" app, ensuring its efficacy in emergency situations.

CHAPTER 3

SYSTEM ANALYSIS

3.1 DEMERITS OF EXISTING SYSTEM

The existing disaster management systems exhibit several drawbacks, including limited accessibility and outreach, fragmented information dissemination, inadequate offline functionality, and a lack of interactivity and community involvement. These limitations contribute to challenges such as delayed responses, difficulties in accessing emergency services, and insufficient predictive capabilities. Additionally, the complexity of communication channels and the inefficiency of two-way communication further hinder the overall effectiveness of disaster management efforts. Recognizing these demerits underscores the imperative for the development of the "DisasterGuard" app, which aims to systematically address these shortcomings and introduce innovative features to enhance the responsiveness and inclusivity of disaster management processes.

3.2 MERITS OF PROPOSED SYSTEM

The "DisasterGuard" system introduces a set of merits designed to significantly enhance disaster management capabilities. With a focus on accessibility, the system ensures widespread outreach, reaching even remote areas. Information dissemination is streamlined, addressing previous issues of fragmentation and ensuring a cohesive platform for timely communication. Robust offline functionality mitigates connectivity challenges, guaranteeing essential information remains accessible during crises. Active community involvement is fostered through interactive features, empowering users to contribute collectively to disaster resilience. The system incorporates advanced predictive capabilities, enabling proactive planning based on predictive analytics. Emergency service access is streamlined with intuitive interfaces, reducing response times. Efficient two-way communication channels prioritize real-time interactions between users and emergency responders, promoting effective collaboration and coordination. In summary, the "DisasterGuard" system stands poised to revolutionize disaster management, offering a suite of merits that address existing limitations and significantly enhance overall responsiveness and resilience.

3.3 ARCHITECTURE

The "DisasterGuard" system is architecturally structured into three layers: the Presentation Layer, Application Layer, and Data Layer. The Presentation Layer encompasses the user interface, providing intuitive access to real-time alerts, an emergency services directory, and location-based services. The Application Layer houses core functionalities, including alert and notification management, two-way communication, predictive analytics, and offline functionality. The Data Layer manages user profiles, disaster information, emergency service contacts, and communication logs. The system is designed for seamless integration with external databases and APIs, ensuring flexibility, scalability, and widespread accessibility across diverse platforms and devices. This architecture positions "DisasterGuard" to deliver a comprehensive, user-centric disaster management solution while allowing room for future enhancements.

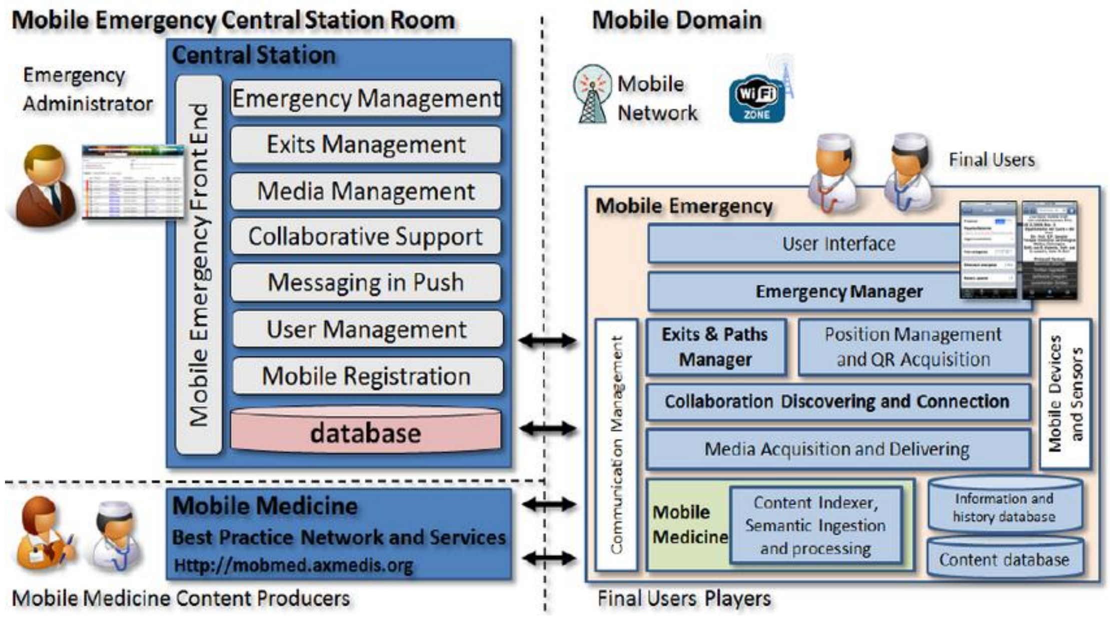

Mobile Emergency Central Station Room
Emergency Administrator
Central Station
Mobile Emergency Front End
Emergency Management
Exits Management
Media Management
Collaborative Support
Messaging in Push
User Management
Mobile Registration
database
Mobile Medicine
Best Practice Network and Services
Http://mobmed.axmedis.org
Mobile Medicine Content Producers
Mobile Domain
Mobile Network
Final Users
Mobile Emergency
User Interface
Emergency Manager
Communication Management
Exits & Paths Manager
Position Management and QR Acquisition
Collaboration Discovering and Connection
Media Acquisition and Delivering
Mobile Devices and Sensors
Mobile Medicine
Content Indexer, Semantic Ingestion and processing
Information and history database
Content database
Final Users Players

Fig : 3.1

3.4 USE-CASE DIAGRAMS

The "DisasterGuard" project introduces a dynamic and user-centric disaster management mobile application designed to revolutionize how individuals and communities prepare for, respond to, and recover from disasters. Through a meticulously crafted architecture, the app addresses existing limitations by enhancing accessibility, providing real-time alerts, fostering community engagement, and incorporating advanced features such as offline functionality and predictive analytics. The use-case diagrams visually depict key interactions, showcasing the seamless integration of critical functionalities. By prioritizing user empowerment, streamlined communication, and community resilience, "DisasterGuard" stands as a pioneering solution, poised to redefine the landscape of disaster management through innovation, adaptability, and a commitment to enhancing overall disaster resilience.

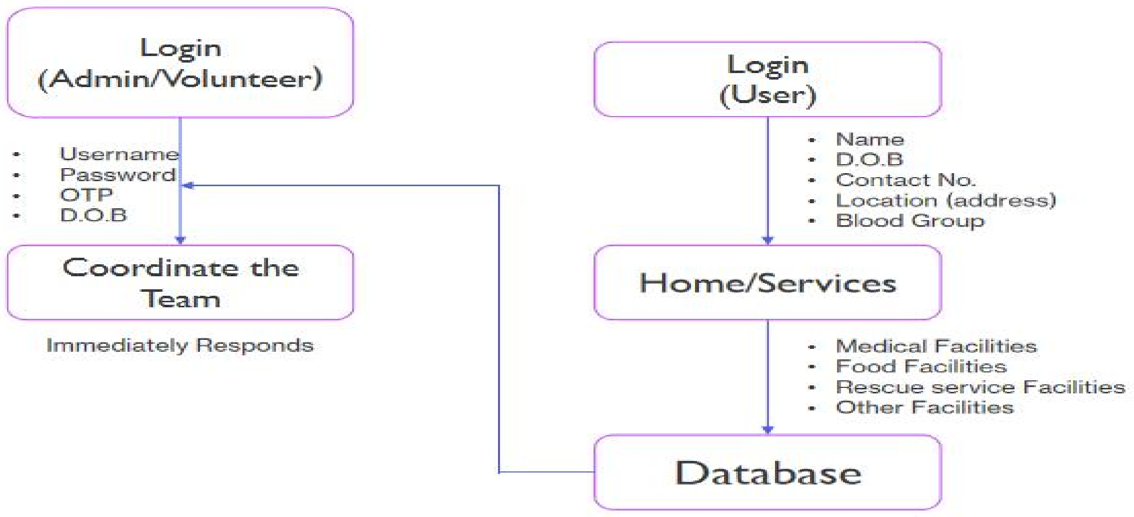


Fig3.2

CHAPTER 4

MODULE DESCRIPTION

4.1 Registration Module

The Registration Module is a fundamental entry point for users, enabling them to establish accounts with a unique username and password. This streamlined process enhances user convenience and expedites registration. The registration page features editable text fields for entering a chosen username and a secure password. By focusing on essential credentials, this module ensures a swift and user-friendly registration process, laying the foundation for users to access and benefit from the comprehensive features of the "DisasterGuard" app.

4.2 Detail Module

The Detail Module plays a crucial role in enriching user profiles within the DisasterGuard app, capturing essential information to enhance disaster management support. Users are prompted to enter key details, such as their geographical location, including state and district, providing valuable context for tailored disaster alerts. Additionally, users can input specific train-related information, such as station and platform numbers, ensuring personalized and relevant disaster information. This module empowers the DisasterGuard app to deliver location-specific alerts, real-time updates, and community engagement features, fostering a more informed and resilient user base. By incorporating these details, the DisasterGuard app ensures a comprehensive understanding of the user's context, enabling more effective disaster preparedness and response.

4.3 Navigation menu Module

The Navigation Module is a cornerstone feature in the DisasterGuard app, designed to optimize user experience and accessibility. This module incorporates a user-friendly menu bar, acting as a central hub for seamless navigation within the application. Through this intuitive menu bar, users can effortlessly access critical features, including live location tracking, emergency services directory, community engagement, and app settings. The Navigation Module enhances the overall user-friendliness of the DisasterGuard app, ensuring that users can efficiently find and utilize essential functionalities for disaster preparedness, response, and recovery. This centralized navigation hub contributes to the app's effectiveness in providing swift and organized access to vital information and features during critical situations.

4.4 Two Way Communication

The Two-Way Communication Module facilitates real-time and efficient communication between users and emergency responders during disasters. Users can send distress signals, request assistance, and communicate their needs directly through the app. Emergency responders can receive and respond to these requests, ensuring a swift and coordinated response to emergencies. This module enhances communication channels, fostering effective collaboration and timely assistance during critical situations.

4.5 Early Warning Module

The Predictive Analytics and Early Warning Module utilize advanced data prediction models to provide users with proactive disaster planning information. The module analyzes historical data and trends to anticipate potential disasters, offering users early warnings and actionable insights. By incorporating predictive analytics, the DisasterGuard app empowers users to take preemptive measures, enhancing overall disaster preparedness and response.

CHAPTER 5

SOFTWARE SPECIFICATIONS

5.1 Visual Studio Code (IDE)

Visual Studio Code (VSCode) is a lightweight, open-source code editor developed by Microsoft. It's designed to be a versatile and extensible tool for various programming languages and development tasks. Here are some key features and aspects of Visual Studio Code.Visual Studio Code (VSCode) is a lightweight and powerful source-code editor developed by Microsoft, renowned for its versatility and extensive features. Released in 2015, VSCode has rapidly become a preferred choice among developers for its user-friendly interface, robust performance, and a rich ecosystem of extensions. Its minimalistic design, coupled with a plethora of functionalities such as built-in Git integration, a powerful debugger, and support for a multitude of programming languages, makes it a highly adaptable tool for various development tasks. The editor's strength lies in its extensibility, allowing developers to tailor their coding environment with a vast array of extensions available through the Visual Studio Code Marketplace, transforming it into a personalized and efficient integrated development environment (IDE). Whether working on web development, cloud services, or any other coding project, VSCode stands out as a flexible and feature-rich code editor, enhancing the development experience for programmers across the globe.

5.2 React Native

React Native is an open-source framework developed by Facebook that enables the development of cross-platform mobile applications using a single codebase. Launched in 2015, React Native has gained immense popularity in the mobile development community for its efficiency and the ability to write code once and deploy it on both iOS and Android platforms. At its core, React Native leverages the principles of React, a JavaScript library for building user interfaces, allowing developers to use a declarative and component-based approach to create intuitive and responsive mobile applications.

One of the standout features of React Native is its "write once, run anywhere" philosophy, which enables developers to use the same codebase for both iOS and Android platforms. This not only streamlines the development process but also ensures a consistent user experience across different devices. React Native achieves this by employing native components rather than web-based components, providing high performance and a native feel to the applications. The framework has been adopted by a myriad of companies and developers globally, making it a go-to choice for building cross-platform mobile apps efficiently and effectively.

5.2.1 React Redux

React Redux is a state management library that seamlessly integrates with React, providing a predictable state container for managing the state of a React application in a scalable and efficient way. Redux itself is a standalone library for managing the state of JavaScript applications, and when combined with React, it becomes React Redux. This integration is particularly powerful for complex applications where managing state can become challenging.

The core idea behind React Redux is the unidirectional data flow. State is stored in a single, centralized store, and changes to that state are communicated through actions. React components can connect to the Redux store to access or update the state. React Redux introduces the concept of "containers," which are components that are connected to the Redux store. These containers subscribe to the store, ensuring they receive updates when the state changes. Actions, representing changes to the state, are dispatched to the store, triggering the appropriate updates in the connected components.

React Redux simplifies the process of managing state in large React applications by promoting a more organized and scalable approach. It's widely used in conjunction with React to handle the complex state management requirements of modern web applications.

5.2.2 React Navigation

React Navigation is a popular library for handling navigation in React Native applications. It simplifies the process of navigating between different screens and managing the navigation stack. With a declarative API, React Navigation allows developers to define navigation flows in a clear and intuitive way. It supports various navigation patterns, including stack navigation, tab navigation, and drawer navigation, providing a flexible and customizable solution for diverse app structures. React Navigation seamlessly integrates with React Native components, ensuring smooth transitions between screens while maintaining a consistent user experience. As an essential tool for mobile app development with React Native, React Navigation empowers developers to create navigational structures that are both efficient and user-friendly.

5.3 MongoDB

MongoDB is a NoSQL database management application. NoSQL database systems offer an alternative to traditional relational databases using SQL (Structured Query Language). Data is stored in tables, rows, and columns in a relational database, with relationships between entities. In MongoDB, the data is stored in documents using JSON-like structure to represent and interact with data.

MongoDB is a popular open-source NoSQL database management system that falls under the category of document-oriented databases. Developed by MongoDB Inc., it is designed to handle large volumes of data and provide high performance for both read and write operations. MongoDB uses a flexible, schema-less document model, where data is stored in BSON (Binary JSON) documents. These documents can contain nested arrays and subdocuments, allowing for the representation of complex hierarchical relationships.

Key features of MongoDB include horizontal scalability through sharding, which enables the distribution of data across multiple servers, and automatic sharding, which facilitates the management of large datasets. MongoDB supports dynamic queries through its rich query language, and its indexing capabilities contribute to efficient data retrieval. As a NoSQL database, MongoDB is particularly well-suited for applications with evolving and unpredictable schemas, making it a popular choice for web development, content management systems, and other scenarios where flexible data models are essential. Its versatility, scalability, and ease of use have contributed to its widespread adoption in the development community.

CHAPTER 6

IMPLEMENTATION AND RESULTS

6.1 Registration Page

A registration module is a fundamental component of many systems, allowing users to create accounts and access the system's features. The registration page consists of user registration field requesting user login and password (Identity) of the user. The next page involves the terms and conditions whether the user wants to be a volunteer.

.

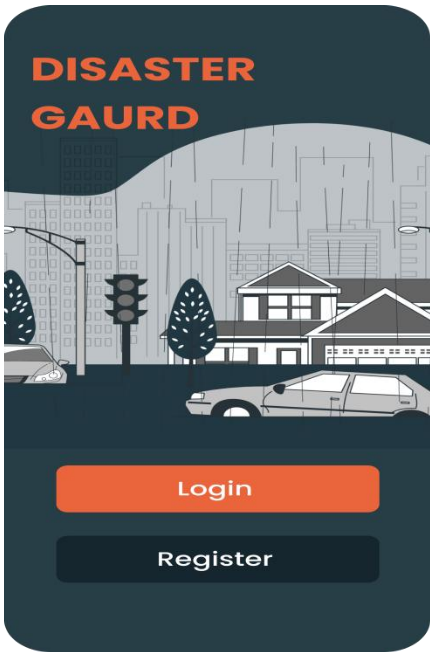


Fig 6.1

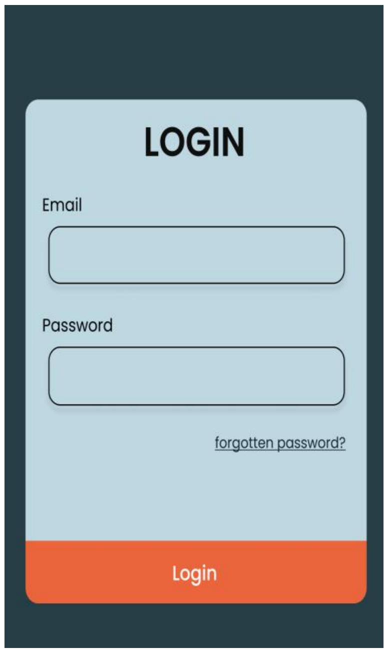


Fig 6.2

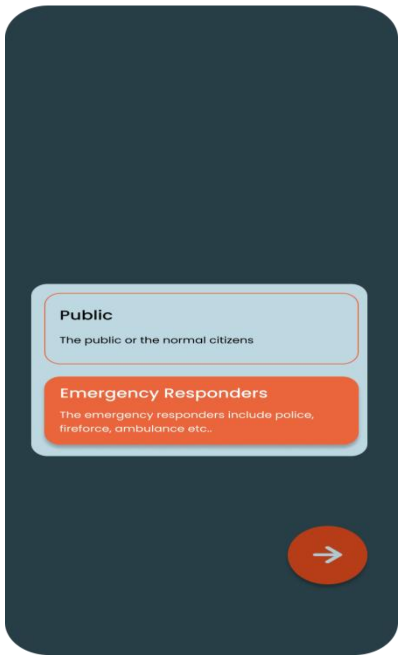

Public
The public or the normal citizens
Emergency Responders
The emergency responders include police, fireforce, ambulance etc..

Fig 6.3

6.2 Home Page

The Home page has several tabs and features for the user to access , it includes a emergency call button incase of in need of immediate evacuation. And emergency call includes the direct contact to safety departments

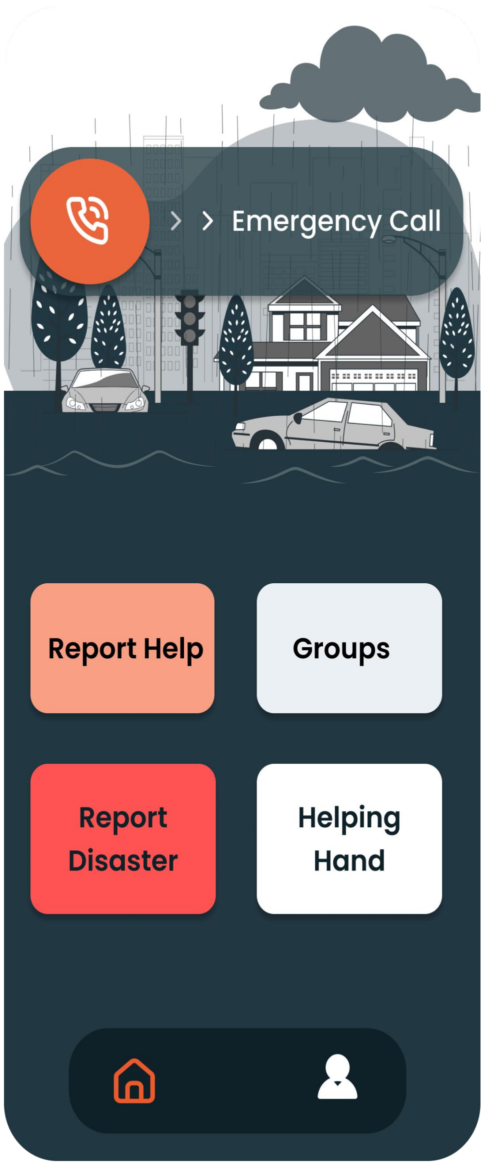


Fig 6.4

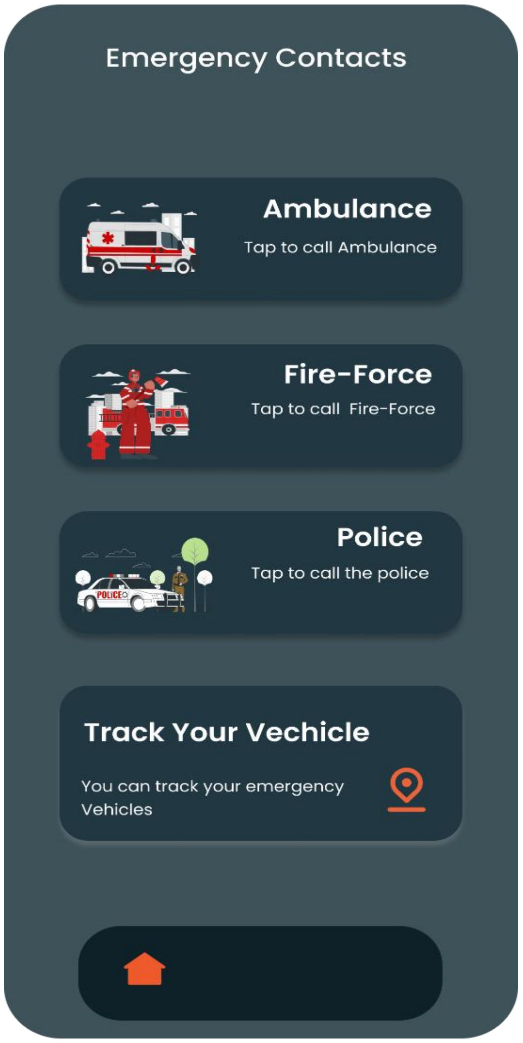


Fig 6.5

6.3 Report Disaster Page

The Report Disaster page within the DisasterGuard app serves as a crucial interface for users to swiftly communicate and document ongoing disasters. This feature empowers users to provide real-time information about the nature and extent of a disaster, contributing to a comprehensive situational awareness. Users can input critical details such as the disaster type, location, and additional notes, fostering a collective effort to enhance emergency response. The Report Disaster page stands as a pivotal component in DisasterGuard, facilitating immediate and accurate reporting for proactive disaster management.

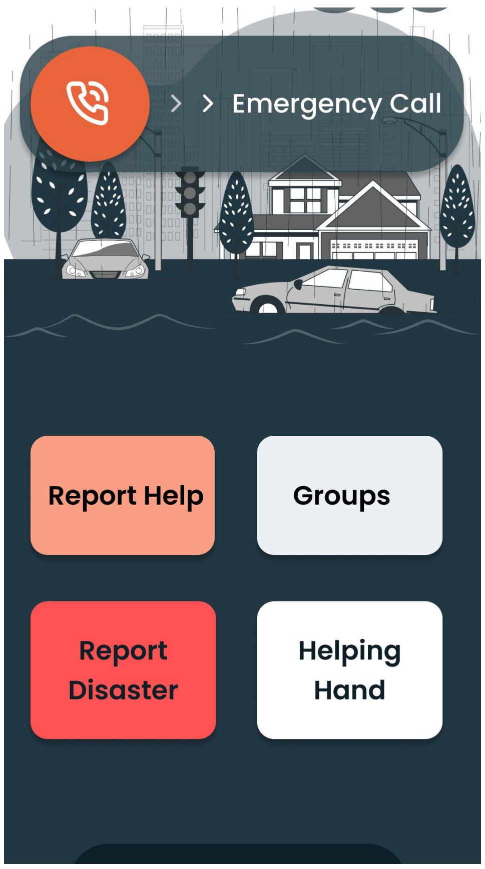
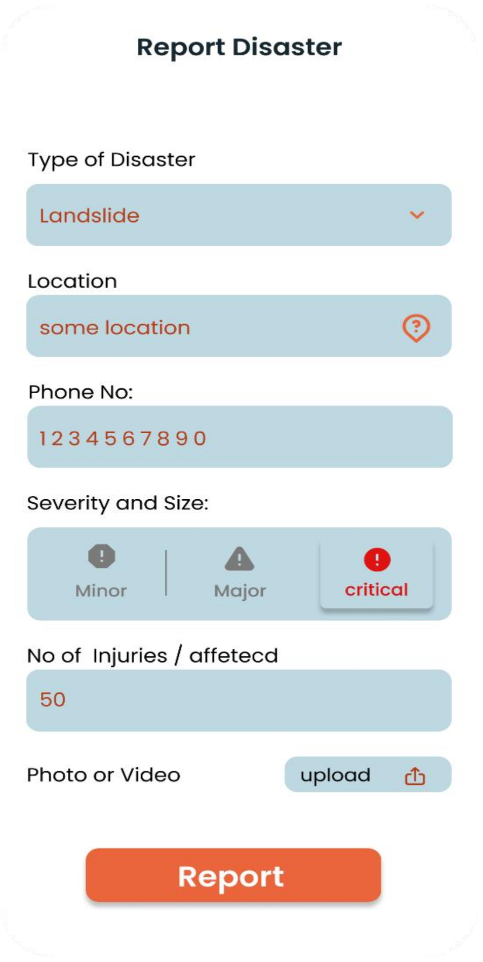


Fig 6.6



Type of Disaster
Landslide

Location
some location

Phone No:
1234567890

Severity and Size:
Minor | Major | critical

No of Injuries / affetecd
50

Photo or Video
upload

Fig 6.7

6.4 Report Help

The "Report Help" module serves as a pivotal component of the system, it lets the users to request for help regarding the basic necessity needs. The user can report the items that is needed for the problem or for the safety measures.

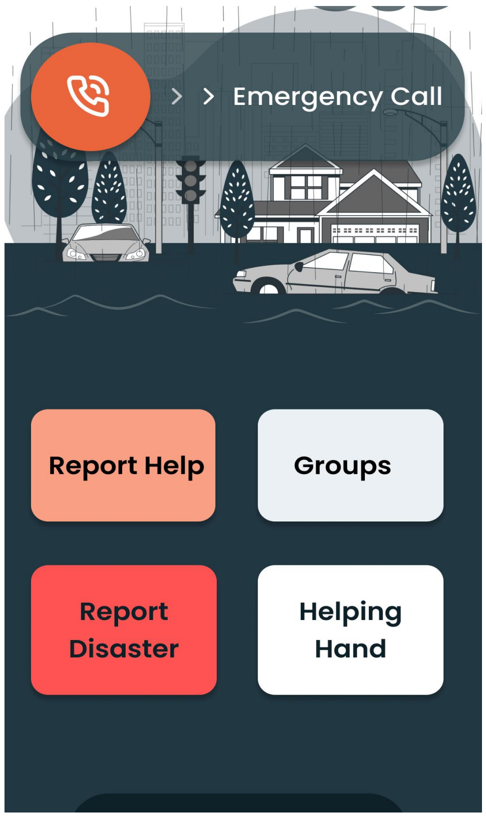
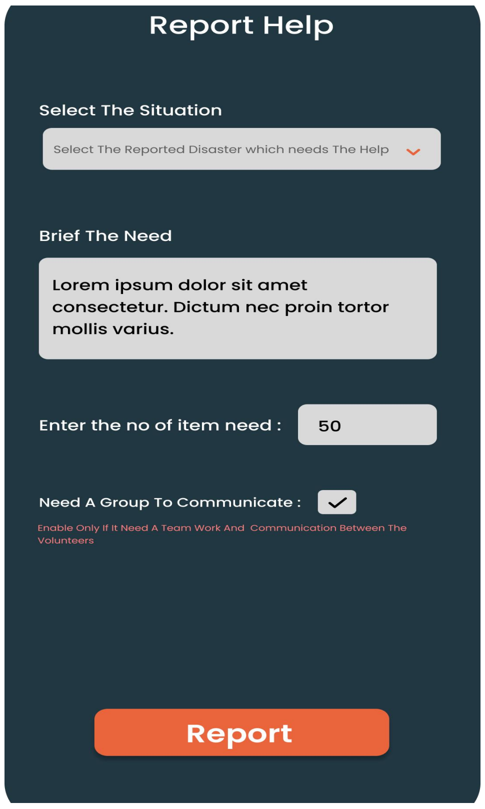


Fig 6.8


Report Help

Select The Situation

Select The Reported Disaster which needs The Help

Brief The Need

Lorem ipsum dolor sit amet consectetur. Dictum nec proin tortor mollis varius.

Enter the no of item need : 50

Need A Group To Communicate : ✓

Enable Only If It Need A Team Work And Communication Between The Volunteers

Fig 6.9

6.5 Group Page

The Group Page module serves as a user-friendly interface, allowing volunteers to engage in real-time text-based conversations. This allows the volunteers to interact with various other volunteers that is present in the group inorder to take tactical decisions .

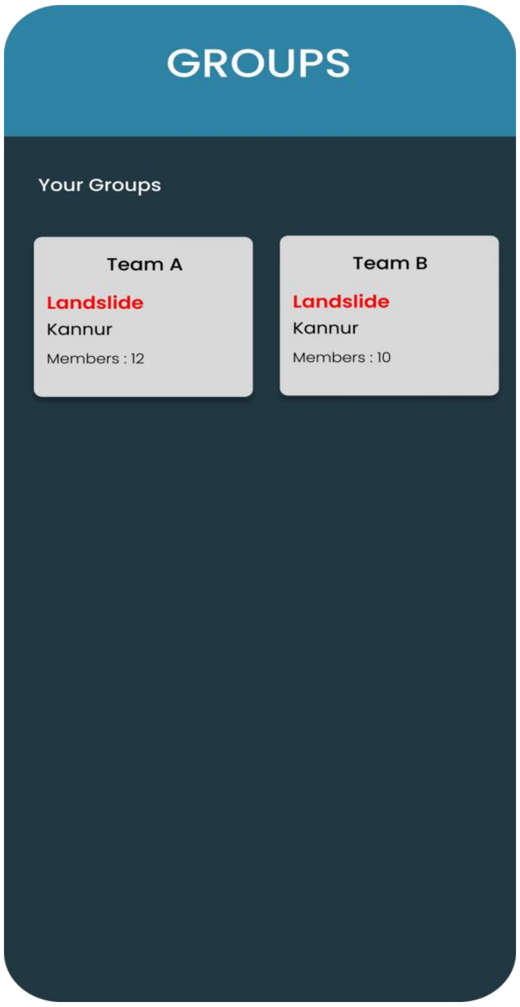
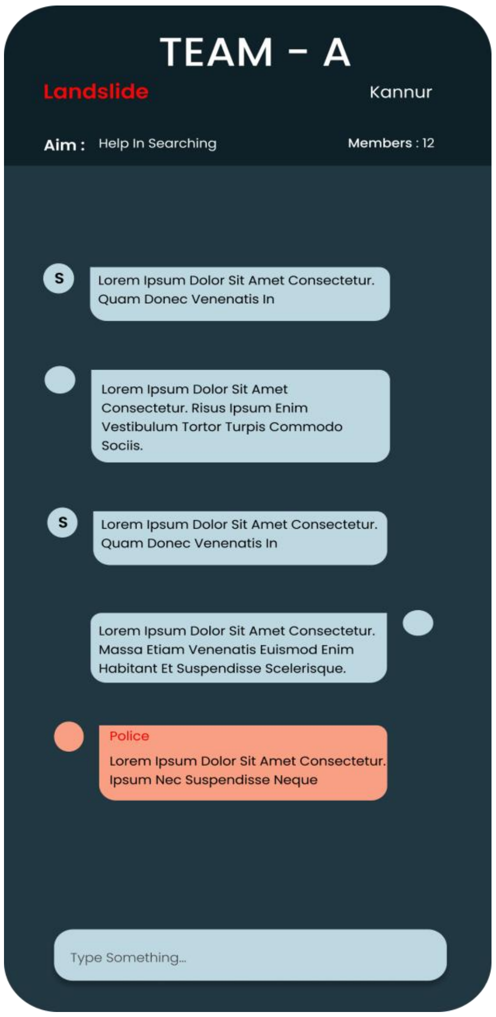


Fig 6.10



Lorem Ipsum Dolor Sit Amet Consectetur. Quam Donec Venenatis In

Lorem Ipsum Dolor Sit Amet Consectetur. Risus Ipsum Enim Vestibulum Tortor Turpis Commodo Sociis.

Lorem Ipsum Dolor Sit Amet Consectetur. Quam Donec Venenatis In

Lorem Ipsum Dolor Sit Amet Consectetur. Massa Etiam Venenatis Euismod Enim Habitant Et Suspendisse Scelerisque.

Lorem Ipsum Dolor Sit Amet Consectetur. Ipsum Nec Suspendisse Neque

Fig 6.11

6.6 Helping Hand Page

The Helping Hand feature is designed to facilitate assistance among app users within affected areas. It allows individuals to offer help or request assistance based on specific needs arising from a disaster.

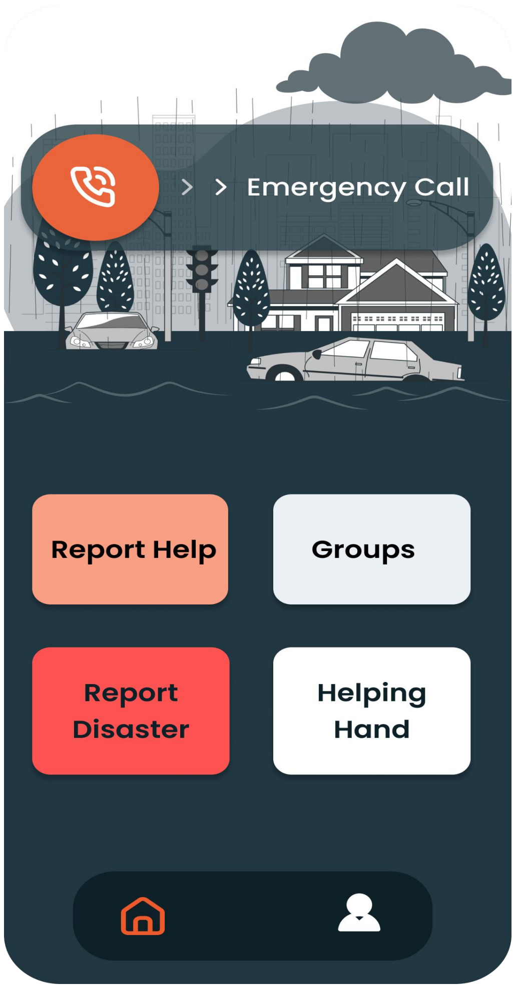
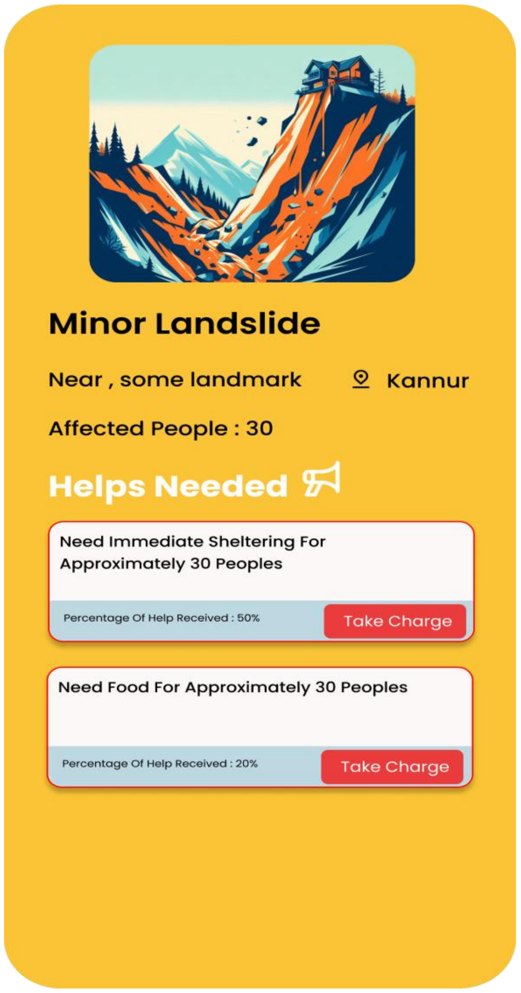


Fig 6.12



Minor Landslide

Near , some landmark Kannur

Affected People : 30

Helps Needed

Need Immediate Sheltering For Approximately 30 Peoples

Percentage Of Help Received : 50%

Need Food For Approximately 30 Peoples

Percentage Of Help Received : 20%

Fig 6.13

6.7 Government Officials Side

The Admin Side gives Access to various features which is related to the safety of the people.the homepage for admins contains features such as location tracking of safety vehicles .direct interaction with volunteers.

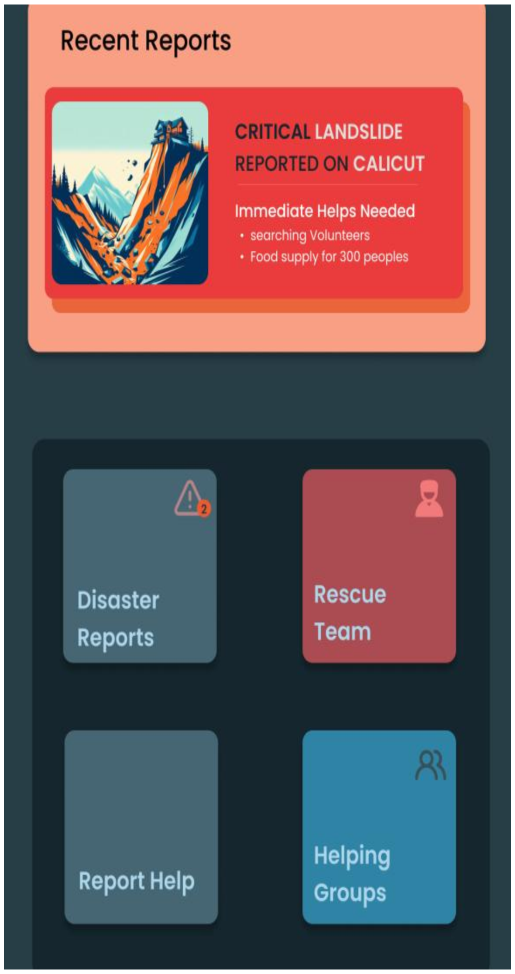


Fig 6.14

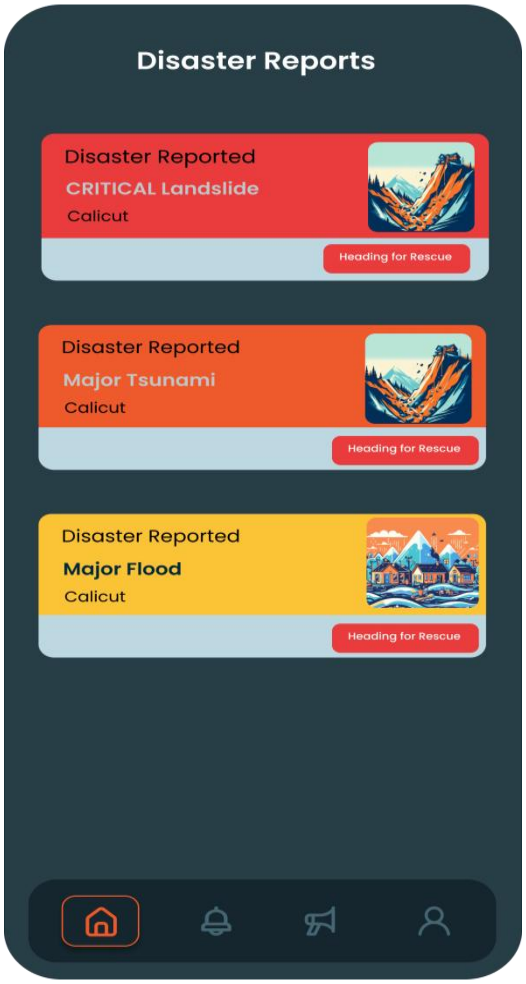
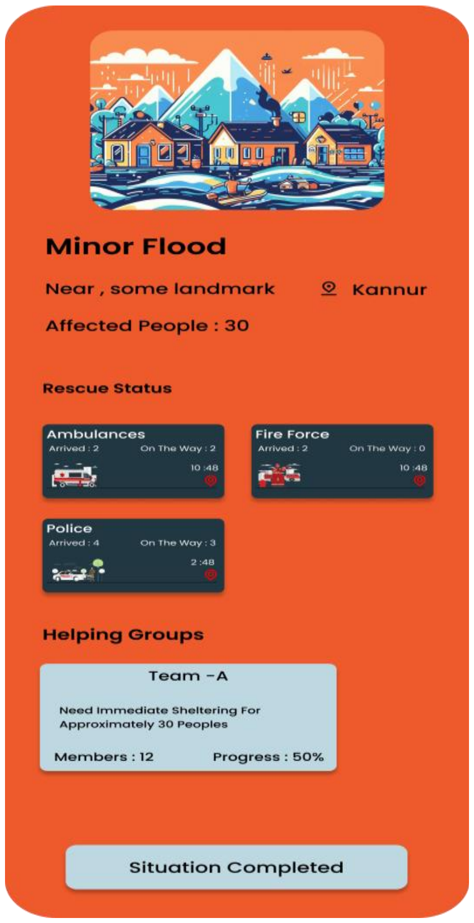


Fig 6.15

6.8 Rescue Team Page

The Admin Side gives Access to various features which is related to the safety of the people.the homepage for admins contains features such as location tracking of safety vehicles .direct interaction with volunteers. Rescue Team gives Access to the ongoing Safety Rescue that is carried out .



Minor Flood

Near , some landmark

Kannur

Affected People : 30

Rescue Status

Ambulances
Arrived : 2 On The Way : 2
10 :48

Fire Force
Arrived : 2 On The Way : 0
10 :48

Police
Arrived : 4 On The Way : 3
2 :48

Helping Groups

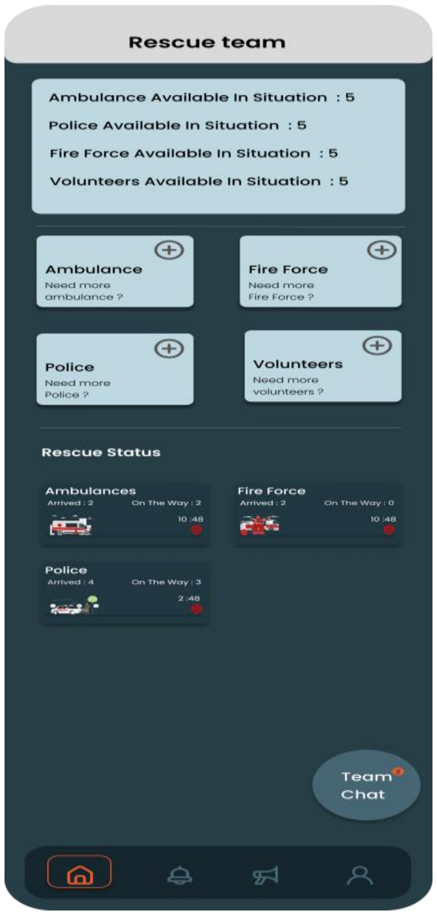
Team -A

Need Immediate Sheltering For Approximately 30 Peoples

Members : 12 Progress : 50%

Situation Completed

Fig 6.16



Rescue team

Ambulance Available In Situation : 5
Police Available In Situation : 5
Fire Force Available In Situation : 5
Volunteers Available In Situation : 5

Ambulance
Need more ambulance ?

Fire Force
Need more Fire Force ?

Police
Need more Police ?

Volunteers
Need more volunteers ?

Rescue Status

Ambulances
Arrived : 2 On The Way : 2
10 :48

Fire Force
Arrived : 2 On The Way : 0
10 :48

Police
Arrived : 4 On The Way : 3
2 :48

Team Chat

Fig 6.17

6.9 Report Help Page

The Safety Department Interface Help Page serves as a comprehensive guide for Safety Department personnel, offering detailed insights and instructions on effectively navigating and utilizing the interface. Covering essential aspects such as interface navigation, safety reporting procedures, document uploads, emergency reporting, and adherence to safety protocols, this guide ensures a thorough understanding of the interface functionalities. It emphasizes the importance of accurate reporting, provides troubleshooting tips, and encourages adherence to safety guidelines, thereby enabling users to efficiently report incidents, maintain safety standards, and contribute to a secure working environment while using the interface.

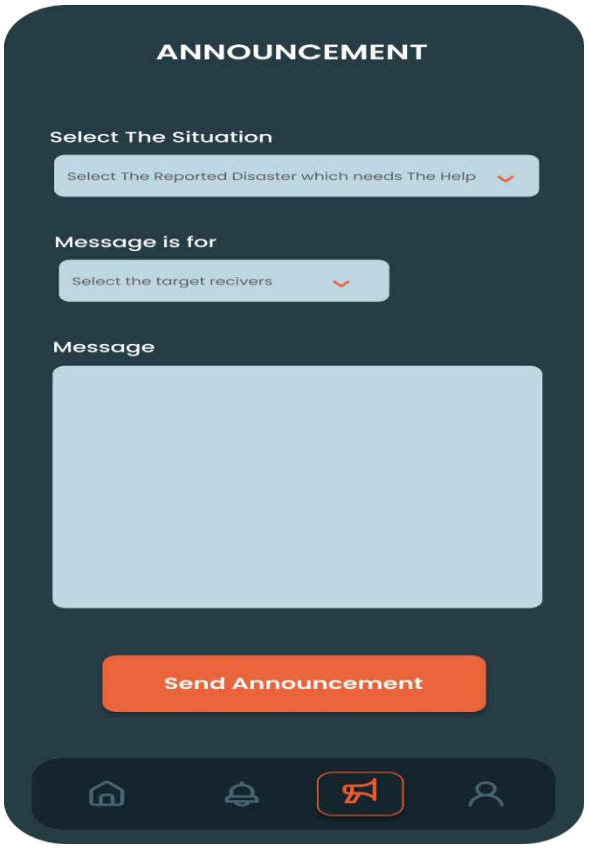
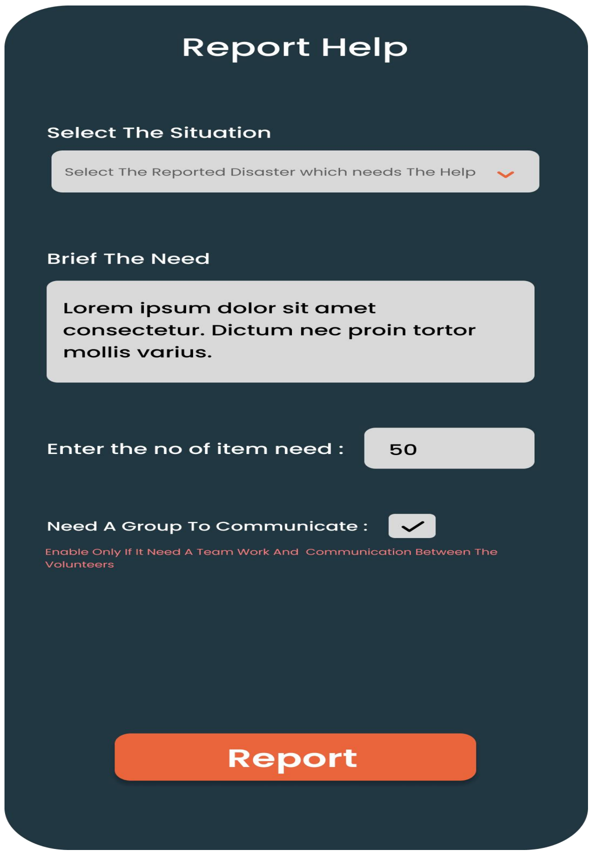

ANNOUNCEMENT
Select The Situation
Select The Reported Disaster which needs The Help
Message is for
Select the target recivers
Message
Send Announcement

Fig 6.18



Lorem ipsum dolor sit amet consectetur. Dictum nec proin tortor mollis varius.

Enter the no of item need : 50

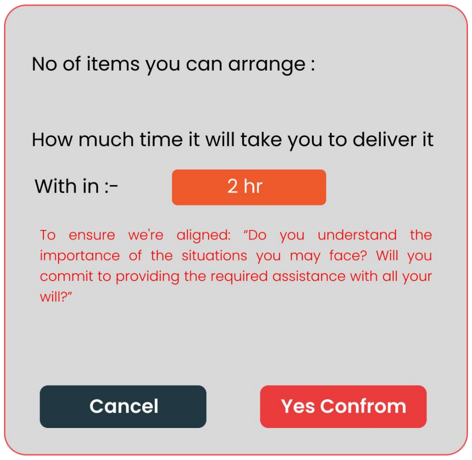
Need A Group To Communicate : ✓

Enable Only If It Need A Team Work And Communication Between The Volunteers

Fig 6.19

6.10 Safety Vehicle Tracking Page

A safety vehicle tracking page aims to enhance operational efficiency, ensure compliance with safety protocols, and improve response times in case of emergencies by providing real-time insights and control over the fleet's safety status and movements. safety vehicle tracking page serves as a centralized platform to monitor and manage the movement and safety status of vehicles, ensuring their secure operation and the safety of passengers or cargo.



No of items you can arrange :

How much time it will take you to deliver it

With in :- 2 hr

To ensure we're aligned: "Do you understand the importance of the situations you may face? Will you commit to providing the required assistance with all your will?"

Fig 6.20

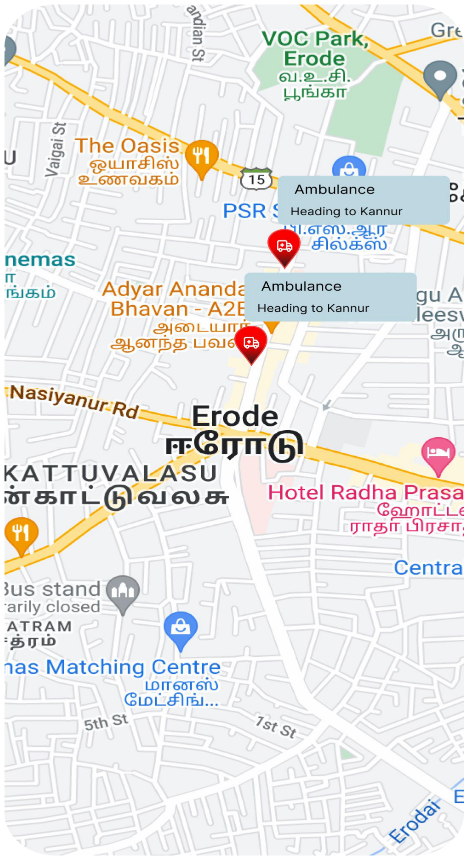


Fig 6.21

CHAPTER 7

CONCLUSION

The In a more detailed elaboration, the DisasterGuard mobile app emerges as a multifaceted and user-centric platform, strategically designed to address the complex challenges of disaster management. The integration of features such as real-time disaster alerts and notifications ensures that users receive timely and critical information, enabling swift decision-making during emergencies. The emergency services directory adds an extra layer of accessibility, providing users with essential contacts for immediate assistance.

Moreover, the app's predictive analytics and early warning systems showcase a forward-thinking approach, utilizing data trends to forecast potential disasters and empower users with proactive planning information. The emphasis on offline functionality is a notable highlight, recognizing the practical constraints of internet connectivity during disasters and ensuring that users can access vital information even in offline scenarios.

Community engagement is not merely a feature but a philosophy embedded in DisasterGuard. The platform facilitates user interactions, sharing of information, and collective contributions to foster a sense of community resilience. The streamlined registration process, focusing on a username and password, simplifies user onboarding, making it easier for individuals to access and utilize the app's array of features.

CHAPTER 8

FUTURE ENHANCEMENT

Future enhancements for the Disaster Guard app could include the integration of advanced technologies to further improve disaster management capabilities. Here are some potential avenues for future development:

- Machine Learning for Prediction: Implementing machine learning algorithms could enhance the app's predictive analytics, allowing for more accurate forecasting of potential disasters based on historical data. This could provide users with even earlier warnings and better preparation.

- Augmented Reality (AR) for Navigation: Integrating AR features into the app could assist users in navigating disaster-stricken areas more effectively. AR overlays could provide real-time information on safe routes, emergency shelters, and points of interest during evacuation.

- IoT Integration for Real-Time Data: Connecting the app with Internet of Things (IoT) devices and sensors could enable the collection of real-time environmental data during disasters. This data could include factors like air quality, temperature, and seismic activity, providing users with more comprehensive situational awareness.

- Blockchain for Secure Information Sharing: Implementing blockchain technology could enhance the security and transparency of information sharing within the app's community engagement features. This could be particularly valuable for securely sharing critical information and resources during and after disasters.

- Customizable Emergency Plans: Providing users with the ability to create and customize their emergency plans within the app could further empower individuals and communities. Users could tailor plans based on specific needs, ensuring a more personalized and effective response during disasters.

- Integration with Emergency Services APIs: Collaborating with official emergency services and agencies to integrate their APIs into the app could enhance the accuracy and immediacy of information, as well as facilitate more seamless communication with relevant authorities.

- Multilingual Support: Expanding language support for the app could make it more accessible to diverse communities, ensuring that crucial information is effectively communicated to a broader user base.

These future enhancements align with the app's commitment to innovation and continuous improvement, aiming to provide an increasingly sophisticated and user-friendly disaster management solution.

APPENDICES

CODING

FrontEnd

```
import { StatusBar, Text, View, Dimensions, StyleSheet, Platform,SafeAreaView } from 'react-
native';
import AuthNav from './Navigation/AuthNav';
import { Provider } from 'react-redux';
import { createStore ,applyMiddleware, compose} from 'redux'
import thunk from 'redux-thunk';
import reducers from './Redux/reducers';
import { AuthProvider } from './context/Authcontext';

const store = createStore(reducers, compose(applyMiddleware(thunk)))

export default function App() {
  return (
    <Provider store={store} >
      <AuthProvider>
        <View style={{ flex: 1 }}>
          <AuthNav />
        </View>
      </AuthProvider>
    </Provider>
  );
}
```

}

HomePage

```
import { useSelector } from "react-redux";
import { useState,useEffect,createContext,} from "react";
import AsyncStorage from "@react-native-async-storage/async-storage";
export const AuthContext = createContext()
export const AuthProvider = ({children})=>{
  const [userData ,setUserData]=useState(null);
  const token = useSelector((state)=> state.authReducer?.userInfo)
  useEffect(()=>{
    if(token !== null) setUserData(token.toString())
  },[token])
  if(userData !== null) AsyncStorage.setItem('UserToken',userData)
  const isLogedIn = async() =>{
    try{
      const userToken =  await AsyncStorage.getItem('UserToken')
      console.dir(userToken)
      // const userToken = JSON.parse(user)
      if(userData !== null ) setUserData(userToken)
    }catch(err){
      console.log(err)
    }
  }
  useEffect(()=>{
   isLogedIn()
  },[])
return(
  <AuthContext.Provider value={{isLogedIn,userData}}>
    {children}
  </AuthContext.Provider>
  )
}
```

MenuPage

```
import { View, Text,ImageBackground,TouchableOpacity,Image,SafeAreaView} from 'react-native';
import React, { useEffect, useState } from 'react'
import styles from '../Style';
import assets from '../../../Components/Assets/assets';
import HostelId from './HostelId';
import SignIn from './SignIn';
import UserRegister from './UserRegister';
import {useDispatch,useSelector} from 'react-redux';
import { validatehostel ,LoginIn } from '../../../Redux/actions/Auth';
const Login = () => {
  let loginData = {
    password:null,
    email:null,
    userType:null
  }
  const [password,setPassword] = useState();
  const [Email,setEmail] = useState();
  const [page ,setPage]=useState(0);
  const [userType ,setUserType]= useState()
  const [hostelID,sethosteID]=useState({})
  const dispatch = useDispatch()
  const hostelStatus =  useSelector( state=> state.authReducer.hostelData)
  useEffect(()=>{
    if(hostelStatus?.hostelExist) setPage(page + 1)
  },[hostelStatus])
  const handleNext = ()=>{
    dispatch(validatehostel(hostelID))
  }
  const handleSignIn =()=>{
    loginData.email=Email,
    loginData.password=password,
    loginData.userType=userType,
    dispatch(LoginIn(loginData))
  }
  const manageLogin =()=>{
    if(page === 0){
      return(
          <HostelId hostelID={hostelID} sethosteID={sethosteID}/>
      )
  }else if(page === 1){
      return(
          <SignIn setPassword={setPassword} setEmail={setEmail} userType={setUserType}/>
      )
  }else{
      return(
          <UserRegister/>
      )
  }
}
  const renderButton =()=>{
       switch(page){
         case 0:return(
           <TouchableOpacity onPress={handleNext}>
             <View
style={[styles.authBtnS,{backgroundColor:"#BA181B",height:50,flexDirection:"row",}]}>
              <Text style={[styles.signUpText,{marginRight:12}]}>Next</Text>
              <Image source={assets.IMAGES.NextIcon}/>
            </View>
           </TouchableOpacity>
         )
         case 1:return(
           <View>
```

```
            <TouchableOpacity onPress={handleSignIn}>
              <View
style={[styles.authBtnS,{backgroundColor:"#BA181B",height:50,flexDirection:"row",}]}>
                <Text style={[styles.signUpText,{marginRight:12}]}>SignIn</Text>
              </View>
            </TouchableOpacity>
            <View style={{marginTop:43,flexDirection:"row",justifyContent:"space-around"}}>
              <Text style={{color:"white"}}>Not yet registered ?</Text>
                <TouchableOpacity onPress={()=>setPage(page+1)}>
                  <Text style={{color:"blue"}}> Register</Text>
                  </TouchableOpacity>
            </View>
            </View>
          )
          default:return null;
        }
    };
    return (
      <ImageBackground source={assets.IMAGES.bg1}style={[styles.registerForm]}>
        <View style={styles.loginTop}>
        <TouchableOpacity style={styles.backBtn} >
              <Image source={assets.IMAGES.BackIcon} />
        </TouchableOpacity>
          <View>
            <ImageBackground source={assets.IMAGES.RegisterIcon} style={styles.registerIcon}/>
          </View>
        </View>
        <View style={[styles.registerBottom]}>
          {
            manageLogin()
          }
          {
```

```
          renderButton()
        }
      </View>
    </ImageBackground>
  )
}
export default Login
```

BackEnd

```
import { View, Text, ImageBackground, TouchableOpacity, StatusBar } from 'react-native';
import styles from './Style';
import assets from '../../Components/Assets/assets';
import Animated, { useSharedValue, useAnimatedStyle, withTiming } from 'react-native-
reanimated';
import { useState } from 'react';
import { useNavigation } from '@react-navigation/native';


const Auth = () => {
  const Navigation = useNavigation()
  return (
    <ImageBackground source={assets.IMAGES.bg1} style={styles.authContainer}>
      <View style={styles.hostelSignUpContainer}>
        <View style={[{ flex: 1 }]}>
          <ImageBackground style={[styles.shape]} source={assets.IMAGES.shapes}>
            <ImageBackground source={assets.IMAGES.Authimg} style={styles.authimg} />
            <View style={[styles.authBtnContainer]}>
              <View style={styles.authBtnCover}>
                <TouchableOpacity onPress={() => Navigation.navigate('Register')}>
                  <View style={[styles.authBtnR]}>
                    <Text style={styles.regText}>Register</Text>
                  </View>
                </TouchableOpacity>
                <TouchableOpacity onPress={() => Navigation.navigate('Login')}>
                  <View style={[styles.authBtnS]}>
                    <Text style={styles.signUpText}>SignIn</Text>
                  </View>
                </TouchableOpacity>
              </View>
            </View>
          </ImageBackground>
        </View>
      </View>
    </ImageBackground>
  )
}
export default Auth;
```

Style

```
import { Dimensions, StyleSheet } from 'react-native';
import assets from '../../Components/Assets/assets';
const { width } = Dimensions.get('window')
const OtpInputWidth = Math.round(width / 6)
const styles = StyleSheet.create({
  authContainer: {
    width: "100%",
    height: "100%",
    display: "flex",
  },
shape: {
   width: "100%",
   height: "100%",
   position: "absolute",
},
authimg: {
   width: "100%",
   height: "82%",
},
authBtnContainer: {
   width: "100%",
   height: "20%",
   justifyContent: "center",
   alignItems: "center",
},
authBtnCover: {
   display: "flex",
   flexDirection: "row",
   justifyContent: "space-between",
   backgroundColor: "black",
   borderRadius: 20,
},
authBtnR: {
   width: 150,
   height: 60,
   backgroundColor: "white",
   borderRadius: 20,
   alignItems: "center",
   justifyContent: 'center',
},
authBtnS: {
   width: 150,
    height: 60,
    backgroundColor: "black",
    borderRadius: 20,
    alignItems: "center",
    justifyContent: 'center',
    textAlign: "center"
},
signUpText: {
    fontFamily: assets.FONTS.fontFamily,
    color: "white",
    fontSize: assets.FONTS.Btn,
    fontWeight: assets.FONTS.Medium,
},
regText: {
    fontFamily: assets.FONTS.fontFamily,
    color: "black",
    fontSize: assets.FONTS.Btn,
    fontWeight: assets.FONTS.Medium,
},
hostelSignUpContainer: {
    width: "100%",
    height: "100%",
},
registerForm: {
    width: "100%",
    height: "100%",
    position: "absolute",
    justifyContent: "space-between",
},
registerTop: {
    width: "100%",
    height: "35%",
  display: "flex",
  flexDirection: "column",
  justifyContent: "flex-end",
  alignItems: "center",
},
registerBottom: {
  width: "100%",
  height: "65%",
  alignItems: "center",
  paddingTop: 10,
  flexDirection: "column",
  marginBottom: 12
},
registerIcon: {
  width: 150,
  height: 150,
  borderRadius: 50,
  overflow: "hidden",
  shadowColor: "black",
  elevation: 25,
},
registerInputs: {
  backgroundColor: "white",
  width: "75%",
  shadowColor: "black",
  elevation: 20,
  height: 50,
  borderRadius: 15,
  padding: 10,
  margin: 6,
  fontSize: 18
},
roomDetails: {
   width: "90%",
   height: 80,
   justifyContent: "space-between",
   flexDirection: "row",
   marginBottom: 8,
   alignItems: "center",
   // backgroundColor:"yellow"
},
roomTypes: {
   width: "90%",
   height: 80,
   justifyContent: "space-between",
   flexDirection: "row",
   marginBottom: 8,
   alignItems: "center",
   borderColor: "red",
   borderWidth: 1,
   borderRadius: 15,
   padding: 5
},
loginTop: {
   width: "100%",
   height: "35%",
   justifyContent: "flex-end",
   alignItems: "center",
   position: "relative"
},
backBtn: {
   width: 50,
   height: 50,
   top: 45,
    left: 30,
    position: "absolute",
    justifyContent: "center",
    alignItems: "center"
},
hostelId: {
    width: "100%",
    height: "65%",
    alignItems: "center",
    justifyContent: "center",
    flexDirection: "column",
    position: "relative",
},
OtpInputContainer: {
    width: OtpInputWidth,
    height: OtpInputWidth,
    borderWidth: 2,
    borderColor: "red",
    backgroundColor: "white",
    justifyContent: "center",
    alignItems: "center",
    opacity: .5,
    borderRadius: 15
},
OtpInput: {
    fontSize: assets.FONTS.Btn,
    paddingHorizontal: 15
},
userTypeContainer: {
    backgroundColor: "black",
    flexDirection: "row",
    width: "75%",
        shadowColor: "black",
        height: 50,
        borderRadius: 15,
        fontSize: 16,
        marginBottom: 34,
    },
    userType: {
        width: "50%",
        backgroundColor: "black",
        height: "100%",
        borderRadius: 15,
        alignItems: "center",
        justifyContent: "center"
    }
}
)
export default styles
```

Style2

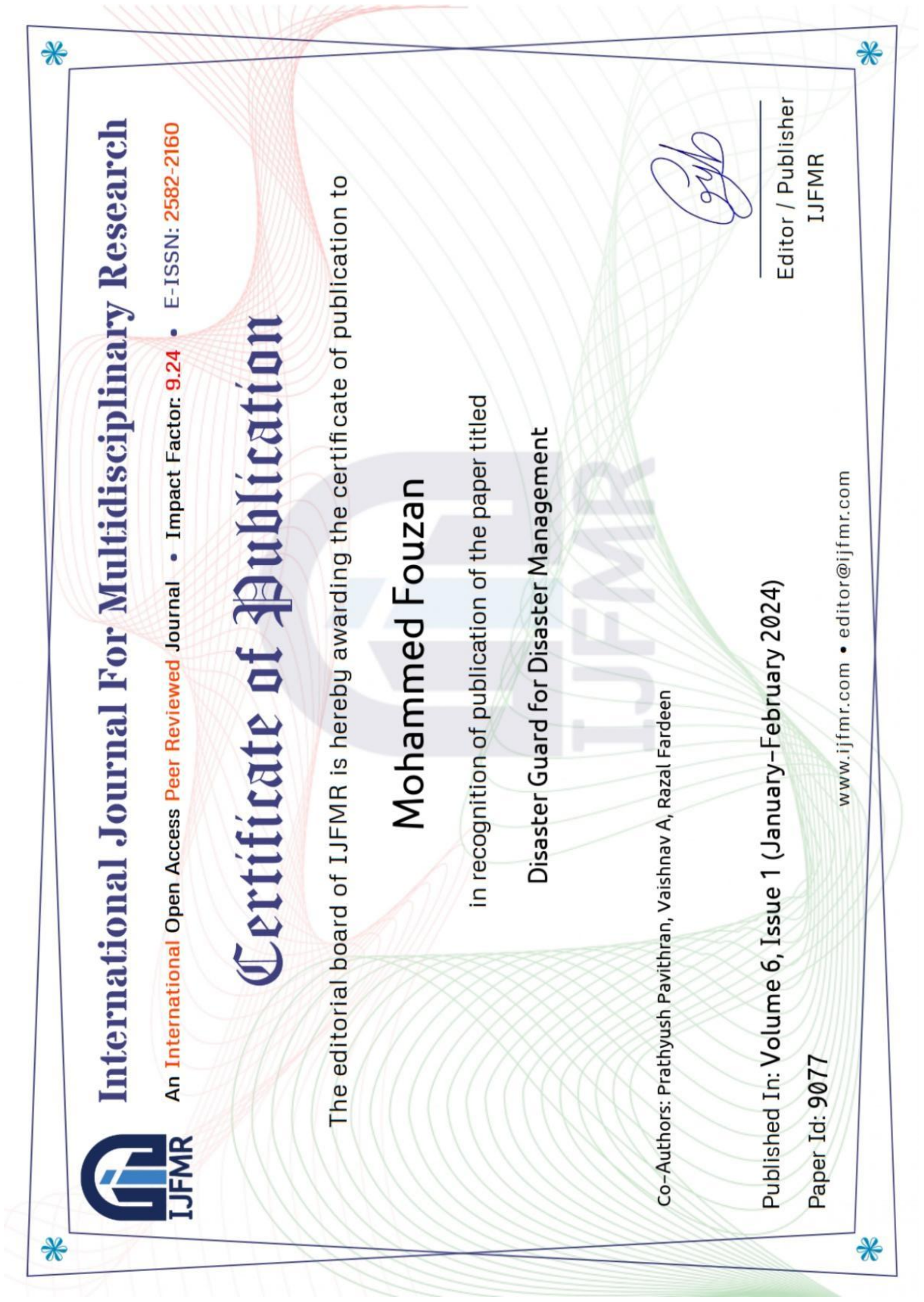
```
import { View, Text,ImageBackground,TouchableOpacity,Image,SafeAreaView} from 'react-native';
import React, { useEffect, useState } from 'react'
import styles from '../Style';
import assets from '../../../Components/Assets/assets';
import HostelId from './HostelId';
import SignIn from './SignIn';
import UserRegister from './UserRegister';
import {useDispatch,useSelector} from 'react-redux';
import { validatehostel ,LoginIn } from '../../../Redux/actions/Auth';
const Login = () => {
  let loginData = {
    password:null,
    email:null,
  userType:null
}
const [password,setPassword] = useState();
const [Email,setEmail] = useState();
const [page ,setPage]=useState(0);
const [userType ,setUserType]= useState()
const [hostelID,sethosteID]=useState({})
const dispatch = useDispatch()
const hostelStatus =  useSelector( state=> state.authReducer.hostelData)
useEffect(()=>{
  if(hostelStatus?.hostelExist) setPage(page + 1)
},[hostelStatus])
const handleNext = ()=>{
  dispatch(validatehostel(hostelID))
}
const handleSignIn =()=>{
  loginData.email=Email,
  loginData.password=password,
  loginData.userType=userType,
  dispatch(LoginIn(loginData))
}
const manageLogin =()=>{
  if(page === 0){
    return(
        <HostelId hostelID={hostelID} sethosteID={sethosteID}/>
    )
}else if(page === 1){
    return(
        <SignIn setPassword={setPassword} setEmail={setEmail} userType={setUserType}/>
    )
}else{
    return(
          <UserRegister/>
      )
  }
}
  const renderButton =()=>{
       switch(page){
         case 0:return(
           <TouchableOpacity onPress={handleNext}>
             <View
style={[styles.authBtnS,{backgroundColor:"#BA181B",height:50,flexDirection:"row",}]}>
              <Text style={[styles.signUpText,{marginRight:12}]}>Next</Text>
              <Image source={assets.IMAGES.NextIcon}/>
            </View>
           </TouchableOpacity>
         )
         case 1:return(
           <View>
           <TouchableOpacity onPress={handleSignIn}>
             <View
style={[styles.authBtnS,{backgroundColor:"#BA181B",height:50,flexDirection:"row",}]}>
               <Text style={[styles.signUpText,{marginRight:12}]}>SignIn</Text>
             </View>
           </TouchableOpacity>
           <View style={{marginTop:43,flexDirection:"row",justifyContent:"space-around"}}>
             <Text style={{color:"white"}}>Not yet registered ?</Text>
                <TouchableOpacity onPress={()=>setPage(page+1)}>
                  <Text style={{color:"blue"}}> Register</Text>
                  </TouchableOpacity>
           </View>
           </View>
         )
         default:return null;
```

```
     }
  };
  return (
    <ImageBackground source={assets.IMAGES.bg1}style={[styles.registerForm]}>
      <View style={styles.loginTop}>
      <TouchableOpacity style={styles.backBtn} >
              <Image source={assets.IMAGES.BackIcon} />
      </TouchableOpacity>
        <View>
          <ImageBackground source={assets.IMAGES.RegisterIcon} style={styles.registerIcon}/>
        </View>
      </View>
      <View style={[styles.registerBottom]}>
        {
          manageLogin()
        }
        {
          renderButton()
        }
      </View>
    </ImageBackground>
  )
}
export default Login
```

REFERENCES

- Bartlett, S. (2008). After the tsunami in Cooks Nagar: The challenges of participatory rebuilding. Children, Youth and Environments, 18(1), 470-484.
- Beck, J. (1998). 100 years of “Just Say No” versus “just say know”: Re-evaluating drug education goals for the coming century. Evaluation Review, 22(1), 15-45.
- Bernardo, L. M., & Veenema, T. G. (2004). Pediatric emergency preparedness for mass gatherings and special events. Disaster Management Response, 2(4), 188-122.
- Center for the Study of Traumatic Stress. (n.d.). Disasters and poverty: Natural disasters disproportionately affect the world’s poor.
- Centers for Disease Control and Prevention. (2009). Coping with a traumatic event. Retrieved from http://emergency.cdc.gov/masscasualties/copingpub.asp
- Corrarino, J. E., Walsh, P. J., & Nadel, E. (2001). Does teaching scald burn prevention to families of young children make a difference? A pilot study. Journal of Pediatric Nursing, 16, 256-262.
- Federal Emergency Management Agency. (1990). Definitions of terms (Instruction 5000.2). Washington DC: FEMA.
- Fuhrmann, S. Stone, L. D., Casey, M. C., Curtis, M. D., Doyle, A. L., Earle, B. D., Jones, D. D., Rodriguez, P., & Schermerhorn, S. M. (2008). Teaching disaster preparedness in geographic education. Journal of Geography, 107(3), 112-120.
- Lauten, A. (2002). Disaster preparedness and safe villages in Central Vietnam. In A. Jabry (Ed.), Children and disasters: After the cameras have gone (p. 34-40). London, England: Plan UK.
- Loar, N., Wolmer, L., & Cohen, D. J. (2001). Mother’s functioning and children’s symptoms 5 years after a SCUD missile attack. American Journal of Psychiatry, 21, 383-390.
- Markenson, D. & Redlener, I. (2004). Pediatric terrorism preparedness national guidelines and recommendations: Findings of an evidence-based consensus process. Biosecurity Bioterrorism, 2(4), 301-319.
- National Advisory Committee on Children and Terrorism. (2004). Schools and terrorism: A supplement to the report of the National Advisory Committee on Children and Terrorism. Journal of School Health, 74(2):39-51.
- Nager, A. L. (2009). Family reunification - Concepts and challenges. Clinical Pediatric Emergency Medicine 10(3), 195-207. Retrieved from http://www.ny2aap.org/pdf/Disaster/195.pdf (PDF, 13 pages)

CONFERENCE CERTIFICATE



IJFMR

International Journal For Multidisciplinary Research

An International Open Access Peer Reviewed Journal • Impact Factor: 9.24 • E-ISSN: 2582-2160

Certificate of Publication

The editorial board of IJFMR is hereby awarding the certificate of publication to

Mohammed Fouzan

in recognition of publication of the paper titled

Disaster Guard for Disaster Management

Co-Authors: Prathyush Pavithran, Vaishnav A, Razal Fardeen

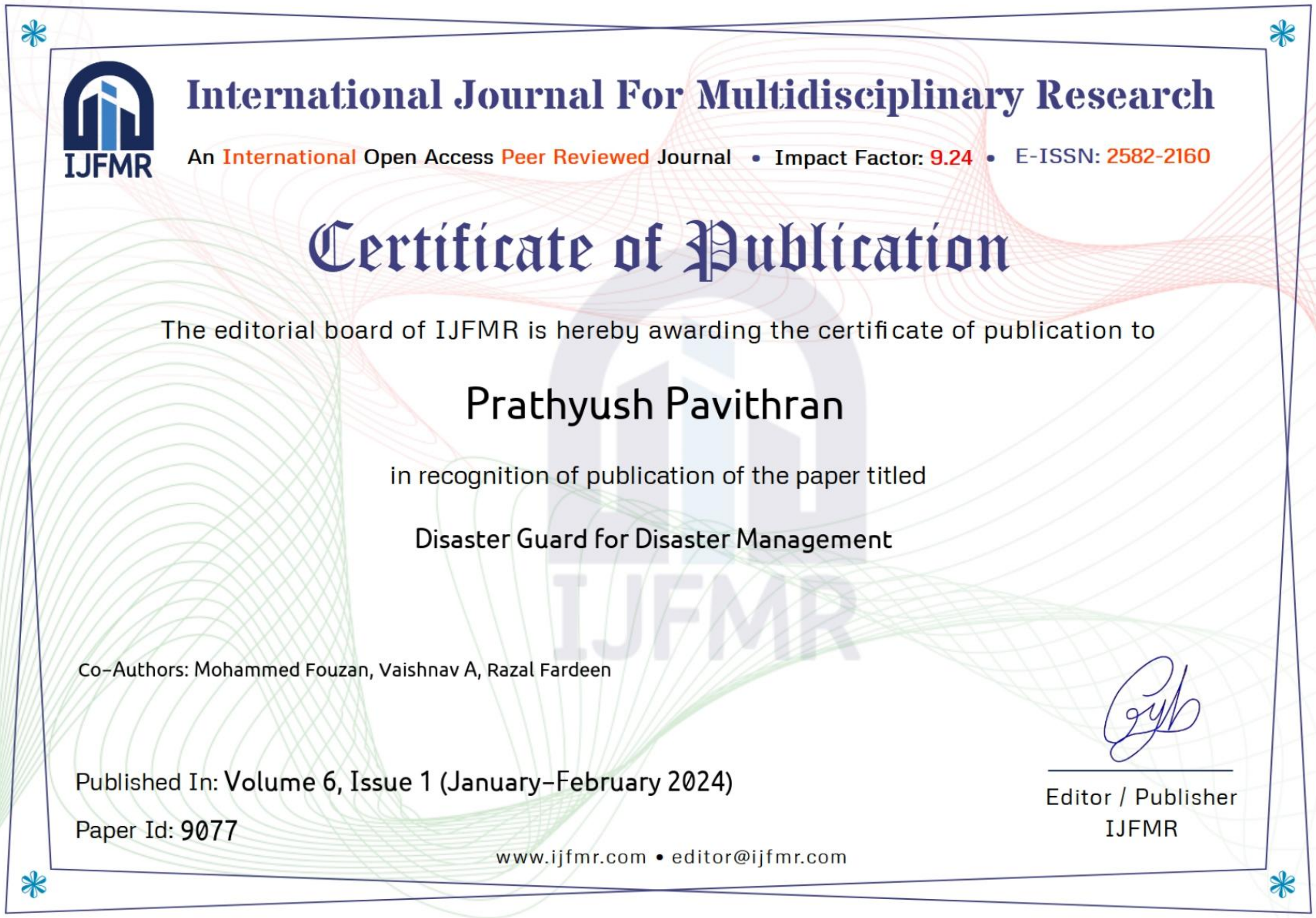
Published In: Volume 6, Issue 1 (January-February 2024)

Paper Id: 9077

[illegible]

Editor / Publisher
IJFMR

www.ijfmr.com • editor@ijfmr.com



International Journal For Multidisciplinary Research

An International Open Access Peer Reviewed Journal • Impact Factor: 9.24 • E-ISSN: 2582-2160

Certificate of Publication

The editorial board of IJFMR is hereby awarding the certificate of publication to

Prathyush Pavithran

in recognition of publication of the paper titled

Disaster Guard for Disaster Management

Co-Authors: Mohammed Fouzan, Vaishnav A, Razal Fardeen

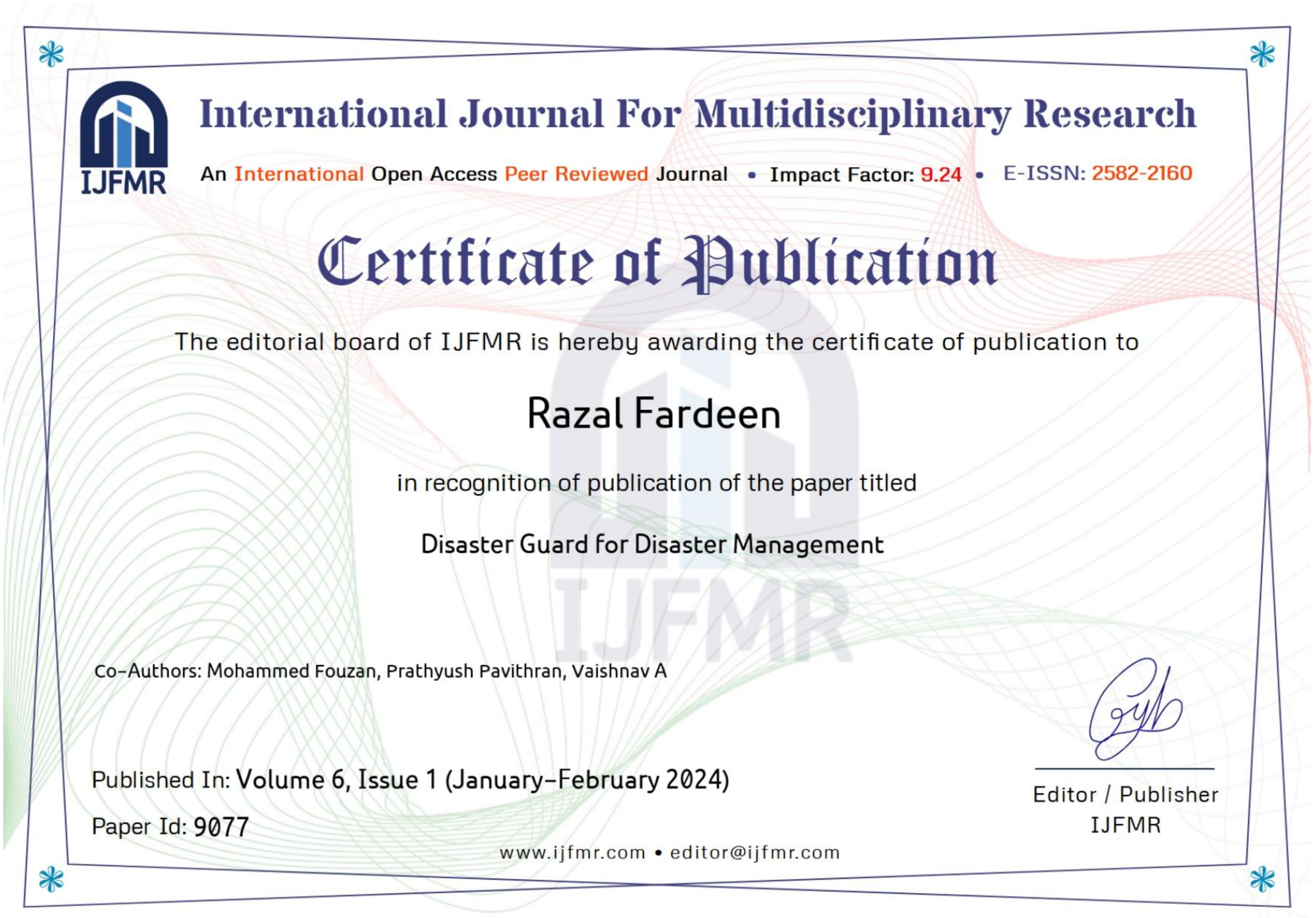
Published In: Volume 6, Issue 1 (January-February 2024)

Paper Id: 9077

www.ijfmr.com • editor@ijfmr.com

[illegible]

Editor / Publisher
IJFMR



International Journal For Multidisciplinary Research

An International Open Access Peer Reviewed Journal • Impact Factor: 9.24 • E-ISSN: 2582-2160

Certificate of Publication

The editorial board of IJFMR is hereby awarding the certificate of publication to

Razal Fardeen

in recognition of publication of the paper titled

Disaster Guard for Disaster Management

Co-Authors: Mohammed Fouzan, Prathyush Pavithran, Vaishnav A

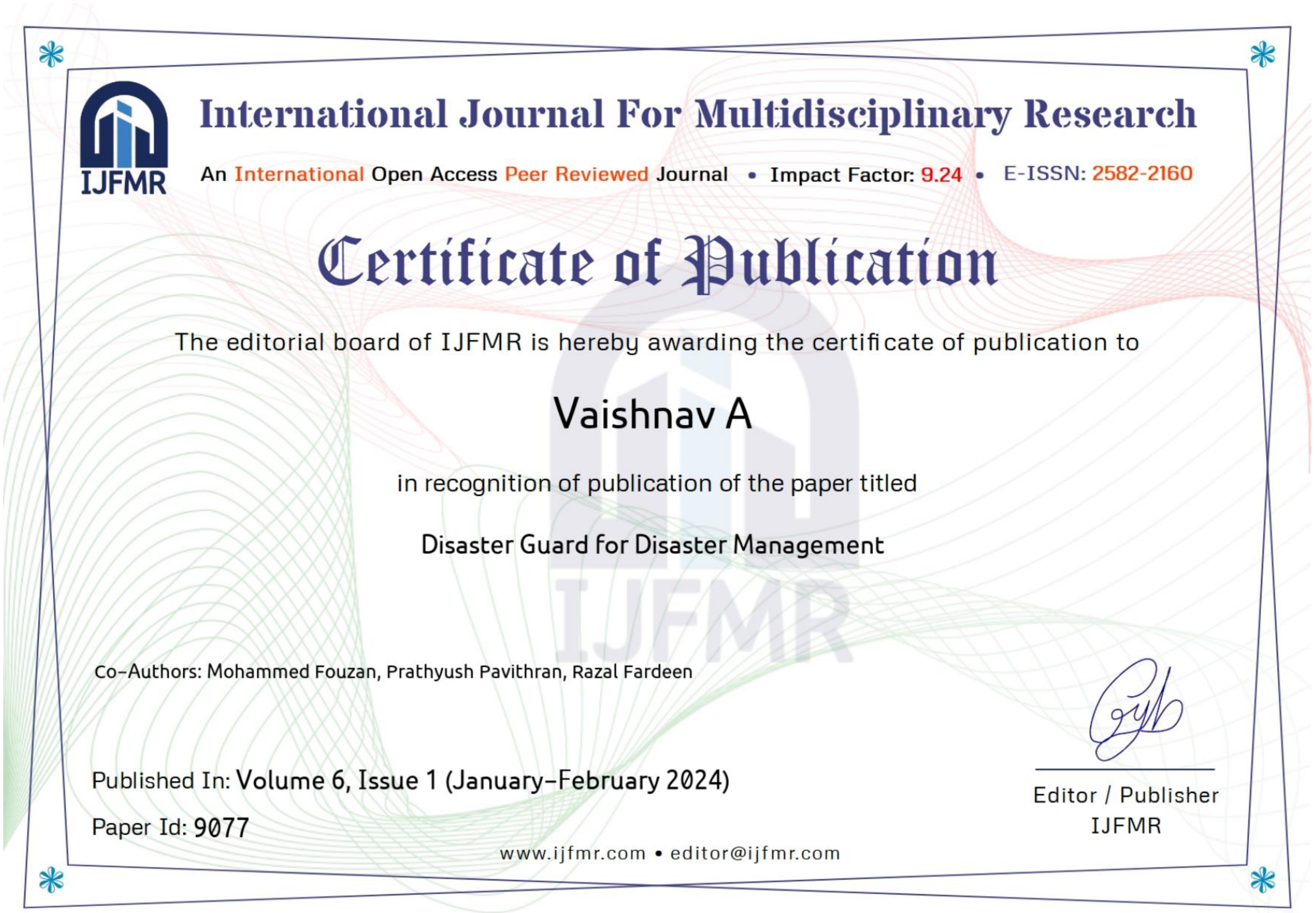
Published In: Volume 6, Issue 1 (January-February 2024)

Paper Id: 9077

[illegible]

Editor / Publisher
IJFMR

www.ijfmr.com • editor@ijfmr.com



International Journal For Multidisciplinary Research

An International Open Access Peer Reviewed Journal • Impact Factor: 9.24 • E-ISSN: 2582-2160

Certificate of Publication

The editorial board of IJFMR is hereby awarding the certificate of publication to

Vaishnav A

in recognition of publication of the paper titled

Disaster Guard for Disaster Management

Co-Authors: Mohammed Fouzan, Prathyush Pavithran, Razal Fardeen

Published In: Volume 6, Issue 1 (January-February 2024)

Paper Id: 9077

[illegible]

Editor / Publisher
IJFMR

www.ijfmr.com • editor@ijfmr.com