 **ACCIDENT ALERT SYSTEM USING**

**BLOCKCHAIN TECHNOLOGY**

### PROJECT REPORT

#### Submitted by

#### Bharathwaj J 181031016

#### Karthikeyan R 181031047

#### Keerthivasan K 181031052

#### Kumaresan S 181031056

#### 

#### in partial fulfillment for the award of the degree of

## BACHELOR OF ENGINEERING

## IN

## COMPUTER SCIENCE AND ENGINEERING

## 

## Department of Computer Science and Engineering

**MAHENDRA ENGINEERING COLLEGE**

**(Autonomous)**

**Mahendhirapuri, Mallasamudram, Namakkal Dt -637 503**

**MAY-2022**

**MAHENDRA ENGINEERING COLLEGE**

**(Autonomous)**

**Mahendhirapuri, Mallasamudram, Namakkal DT -637 503**

**Department of Computer Science and Engineering**

**BONAFIDE CERTIFICATE**

Certified that this Main project report **“ACCIDENT ALERT SYSTEM USING BLOCKCHAIN TECHNOLOGY”** is the bonafide work of **BHARATHWAJ J (181031016),**

**KARTHIKEYAN R(181031047), KEERTHIVASAN K(181031052), KUMARESAN S(181031057)** who carried out our main project work under my supervision.

**Dr M.KANNAN M.E.,M.S.,Ph.D., Dr M.KANNAN M.E.,M.S.,Ph.D.,**

Professor & Head of the Department, Professor & Head of the Department,

Computer Science and Engineering, Computer Science and Engineering,

Mahendra Engineering College. Mahendra Engineering College.

**MAHENDRA ENGINEERING COLLEGE**

**(Autonomous)**

**Mahendhirapuri, Mallasamudram, Namakkal Dt -637 503**

**Department of Computer Science and Engineering**

**CERTIFICATE OF PROJECT APPROVAL**

This is to certify that the Main project report titled **“ACCIDENT ALERT SYSTEM USING BLOCKCHAIN TECHNOLOGY”** is the approved record of work done by **BHARATHWAJ J (181031016), KARTHIKEYAN R (181031047), KEERTHIVASAN K (181031052), KUMARESAN S (181031055)** in partial fulfillment for the award of the Degree of B.E (Computer Science and Engineering) during the academic year 2021- 2022.

**SUPERVISOR**  **HEAD OF THE DEPARTMENT**

**Date:**

Submitted for the end semester viva voce examination held on \_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER**  **EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We wish our heartful thanks to our respected Chairman **Shri.M.G.Bharath Kumar M.A., B.Ed.,** for the blessing and constant support over our final year project period.

We wish to express our sincere thanks to our respected Principal **Dr R.V.Mahendra Gowda M.Tech.,Ph.D.,** Mahendra Engineering College, for all the blessings and help provided during the period of final year project work.

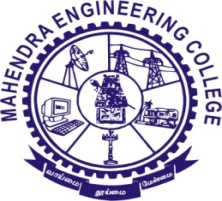
We wish to express our sincere thanks to **Dr M.Kannan M.E.,M.S.,Ph.D.,** Professor and Head, Department of Computer Science and Engineering, for the continuous work and excellent support over the period of final year project work.

We are indebted to our internal guide **Dr M.Kannan M.E.,M.S.,Ph.D.,** Professor and Head, Department of Computer Science and Engineering, for his constant help and creative ideas over the period of final year project work.

We are immensely indebted to our Project coordinator **Mr.R.Venkatesh.,M.E.,** **Associate Professor** for the valuable guidance, support and motivation during the entire course of the project, their timely assistance and reviews went a long way towards the successful completion of this project.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER NO.** | | **TITLE** | **PAGE NO.** |
|  | | **DEPARTMENT VISION & MISSION** | i |
|  | | **POs, PEOs &amp; PSOs** | ii,iii,iv |
|  | | **ABSTRACT** | v |
|  | | **PO ATTAINMENT** | vi |
|  | | **LIST OF FIGURES** | vii |
|  | | **LIST OF ABBREVIATIONS** | viii |
| **1** | | **INTRODUCTION** | 1 |
| **2** | | **LITERATURE SURVEY** | 8 |
| **3** | | **SYSTEM ANALYSIS** | 13 |
| 3.1 | | EXISTING SYSTEM | 13 |
| 3.2 | | DISADVANTAGES | 13 |
| 3.3 | | PROPOSED SYSTEM | 14 |
| 3.4 | | ADVANTAGES | 14 |
| 3.5 | | SYSTEM ARCHITECTURE | 14 |
| **4** | | **MODULE IMPLEMENTATION** | 17 |
| 4.1 | | MODULE LIST | 17 |
| 4.2 | | IMPLEMENTATION | 17 |
| **5** | | **SYSTEM SPECIFICATION** | 19 |
| 5.1 | | HARDWARE SYSTEM CONFIGURATION | 19 |
| 5.2 | | SOFTWARE SYSTEM CONFIGURATION | 19 |
| 5.3 | | SOFTWARE ENVIRONMENT | 19 |
| **6** | | **SYSTEM DESIGN** | 26 |
| 6.1 | | UML DIAGRAM | 26 |
| 6.2 | | USECASE DIAGRAM | 26 |
| 6.3 | | CLASS DIAGRAM | 27 |
| 6.4 | | SEQUENCE DIAGRAM | 27 |
| 6.5 | | DATA FLOW DIAGRAM | 28 |
| **7** | | **SYSTEM TESTING** | 30 |
| 7.1 | | TYPES OF TESTING | 30 |
| **8** | | **SCREENSHOTS** | 32 |
| **9** | | **CONCLUSION AND FUTURE ENHANCEMENT** | 35 |
| 9.1 | | CONCLUSION | 35 |
| 9.2 | | FUTURE WORK | 35 |
| **10** | **SAMPLE SOURCE CODE** | 36 |
|  | **REFERENCE** | 52 |

 **MAHENDRA ENGINEERING COLLEGE**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VISION**

* To produce competent computer engineers with state-of-the-art technologies

**MISSION**

* To impart technical education through effective teaching- learning process
* To enhance the student’s employability through mentoring to various skill activities
* To promote research activities with analytical skills to face global challenges
* To imbibe ethical and enterprising characters to become socially responsible engineers

**PROGRAM OUTCOMES (POs)**

**At the time of graduation, students from the Computer Science and Engineering program will possess:**

**Engineering Graduates will be able to**

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning inform informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solution societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member of indiverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

The graduates of Computer Science and Engineering will be able to,

**PEO1** - Good communication, leadership and entrepreneurship skills

**PEO2** - Expertise on advanced computer technologies to become competitive

**PEO3** - The habit of learning and nurture the research attitude

**PEO4** - The ability to work in a team with professional ethics

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

**Engineering Graduates will be able to**

1. Ability to understand the basic concepts and methodologies of computing techniques.
2. Ability to apply engineering knowledge to design and develop computerized solutions by selecting appropriate technology to solve the problems.
3. Ability to use engineering practices and standard strategies in various domains by providing different approaches towards career success.

**ABSTRACT**

In this era of rapid growth of vehicles, the ratio of road accident increases day by day. Nowadays, accident are persistent problems in both developed and developing countries which result in huge loss of life and property. No one in this world is ready to gaze what's happening around them. Nobody cares even when an accident occurs. The accident hazards have been increased by the emergence of technology and also it leads to the rise of a frequency of road accidents. If an accident occurs, an immediate rescue can be considered as a tightrope walk between life and death. In this work, thus endeavor to resolve these two issues through proposing an effective announcement network called Tron Coin, a novel privacy-preserving incentive announcement network based on Block-chainvia an efficient arriving medical help to the respective places. On the one hand, Tron-Coin allows nondeterministic different signers (i.e., users) to generate the signatures and to send announcements anonymously in the trusted environment. On the other hand, with Block-chain, Tron-Coin motivates users with incentives to share accidental information. In addition, transactions and account information in Tron-Coin are tamper-resistant. Tron-Coin thus is able to motivate users to forward announcements anonymously and reliably. Extensive experimental results show that Tron-Coin is efficient and practical in simulations of smart accident detection.Thus the proposed accident alert system Web application with privacy preserving and non-repudiation Block chain based incentive mechanism provides emergency responders with crucial information at the earliest possible time.

**Keywords**:Blockchain,SmartContract,TronCoin,Decentralizedapp,Cryptocurrency, Bitcoin,Incentive,Accident Alert System.

**PO ATTAINMENT**

✓ Relevance of POs and PSOs towards project

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| KEY WORD | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| Blockchain | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Smart-Contract | ✓ | ✓ | ✓ |  |  | ✓ |  |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tron-coin | ✓ | ✓ | ✓ |  | ✓ | ✓ |  |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Accident Alert System | ✓ | ✓ | ✓ |  | ✓ | ✓ |  |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| 3.1 | SYSTEM ARCHITECTURE | 15 |
| 5.1 | DAPP ARCHITECTURE | 20 |
| 6.1 | USECASE DIAGRAM | 27 |
| 6.2 | CLASS DIAGRAM | 27 |
| 6.3 | SEQUENCE DIAGRAM | 28 |
| 6.4  8.1  8.2  8.3  8.4  8.5  8.6 | DATA FLOW DIAGRAM  HOME PAGE OF ACCIDENT ALERT SYSTEM  USER MANDATORY SIGNUP DETAILS  UPDATE USER INFORMATION  EMERGENCY CONTACT INFORMATION  REPORT AN ACCIDENT  RESULT OF AN ACCIDENT ALERT SYSTEM | 28 |
|  |  |  |
|  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ACRONYM** | **ABBREVIATIONS** |
| GPS | GLOBAL POSITIONING SYSTEM |
| GSM | GLOBAL SYSTEM FOR MOBILE COMMUNICATION |
| Dapp  SDK  TRX  API  HTML  JS | DECENTRALIZED APPLICATION  SOFTWARE DEVELOPMENT KIT  TRON  APPLICATION PROGRAMMING INTERFACE  HYPERTEXT MARKUP LANGUAGE  JAVA SCRIPT |
| UML | UNIFIED MODELING LANGUAGE |
| DFD | DATA FLOW DIAGRAM |

**CHAPTER 1**

**INTRODUCTION**

Transportation is a basic need of society. With the help of transportation, human life becomes more easy and comfortable. A vehicle is the chief mode of any kind of transportation. At the same time, as the number of vehicles increase the threat of accidents is also increasing. Road traffic accidents are a human tragedy. They encompass high human suffering and socioeconomic costs in terms of premature deaths, injuries, loss of productivity and so on. By reducing the time between when an accident happens and when the first responders are moved to the scene reduces mortality rates, it can save lives.

By reducing the time between when an accident happens and when the first responders are moved to the scene reduces mortality rates, thus can save lives. Efficient accident alert message notification to the emergency service with the accident location is an essential need to save precious human life. In India, The Global Status Report on Road Safety by the World Health Organization (WHO) revealed that more number of people die in road accidents in India compared to the more populous China. The delay in the information of the accident being reached to the ambulance or to the hospital leads to the unavailability of the first aid provision and it is the most conspicuous reason for a person’s death during accidents. The investigation shows that reducing accident response time by one minute corresponds to the six percent difference in the number of lives saved. In order to provide a solution to this problem, the researchers found that smart phone application could the best way in the development of car accident detection and notification systems. The software applications that are designed to run on the smart phones, tablets and other advanced Web devices are the smart Web applications or the Web apps.

There are different kinds of applications that are available purposely for traffic safety, Navigation, transportation data collection and transportation emission quantification. The Web applications that provide the real time travel information to the travelling public have already been implemented by the Department of Transportation (DOT) agencies. To access and submit information about important vehicle safety concerns, the Web application named “safecar” has been created by the National Highway Traffic Safety Administration (NHTSA). To monitor the traffic operating speeds and incident notification, Tennessee Department ofTransportation (TDOT) developed its Web application “TDOT Smart Way Web app”.

The need for daily financial transactions has led to the development of several electronic payment systems that have made exchange of values relatively easier compared to the traditional means. But, doing so requires financial institutions and presence of heavy communication equipment to do this efficiently. The transactions using payment cards expose a lot of personal information to all the entities that participate in the transaction. As Web phones are taking most of the functions like complimentary card, identification, and Tron/debit card to name a few, it still has many limitations in its current form for global adoption. The beauty of traditional cash transactions is the freedom to spend money anywhere, at any time, with whoever wants the legal tender, and this is something the phone is yet to achieve. The concept of Blockchain has been invented to remove restrictions posed on financial transactions by financial insituations. This Blockchain paradigm is especially popularfor financial transactions, but is also becoming popular for non-financial purposes, popularly for wireless networks. Cryptocurrency has emerged to create a decentralized banking system, where no financial institution controls the money. Cryptocurrency is a kind of virtual currency that uses cryptography to protect transactions. Before the introduction of cryptocurrency, two factors determine the success of an electronic transaction namely; financial institution and Internet connection. With the introduction of cryptographic payment like Bitcoin and Ethereum, the role of financial institutions has been successfully eliminated. But the need for instantaneous connection to Internet to enable Blockchain transaction is still a limitation of the current architecture. The existing Blockchain methods require Internet connection at the moment of transaction, limiting for global adoption.

Researchers have made efforts to create an offline cryptocurrency. Though cryptocurrency is protected with cryptographic encryption, a Blockchain device is prone to unauthorized hacking. In authors proposed BlueWallet which uses hardware token for completing transactions. This approach succeeded in keeping the private key (in hardware) offline, making the wallet secure, but it still requires point ofsale (POS) Internet connection to the Blockchain network to make transactions. This approach uses Digital Signature for verification and signing of transactions. The transaction is introduced into the Bitcoin network if observed as valid by the POS. Other peers in the network in due course will verify and confirm the transaction before it is added to the Bitcoin network. The ledger of a Bit coin network is the end point of every confirmed Blockchain transaction. It takes between 10 and 40 min to complete this process.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 TITLE: Automatic Vehicle Accident Detection and Messaging System**

**AUTHOR:** Jayati Routh, Arshiya das, Piyashi Kundu, Madhubasha Thakur

**YEAR:** 2019

**DESCRIPTION:**

The advent of technology has also increased the traffic hazards and the road accidents take place frequently which causes huge loss of life and property because of the poor emergency facilities. According to this project, when a vehicle meets with an accident, the Micro electro mechanical system (MEMS) sensor will detect the signal and this signal will be analyzed by Arduino. The Arduino sends the alert message through the GSM Module including the location to police control room or a rescue team. So, the police can immediately trace the location through the GPS Module, after receiving the information. Then after confirming the location necessary action will be taken.

**2.2 TITLE: GSM Based Accident Alert System**

**AUTHOR:** Boopathi Raja, Keerthika A, Keerthika S G,Nandhini A, 2Pranitha K J

**YEAR:** 2020

**DESCRIPTION:**

If emergency service could get accident reports and reach it in time, more lives could have been saved.. GPS has become an integral part of a vehicle system nowadays. The accelerometer senses a sudden shift in the vehicle's axles.. The Arduino sends the warning message via the GSM module to the police control room or a rescue team, including the location. So, after receiving the information, the police can automatically track the location via the GPS module. Then, the appropriate action will be taken after verifying the venue

**2.3 TITLE: Accident Detection and Reporting System**

**AUTHOR:** Aboli Ravindra Wakure, Apurva Rajendra Patkar, Manisha Vitthal

**YEAR:** 2020

**DESCRIPTION:**

In highly populated Countries like India, everyday people lose their lives because of accidents and poor emergency facilities. These lives could have been saved if medical facilities are provided at the right time. Acceleratometer sensor can be used in car security system to sense vibrations in vehicle and GPS to give location of vehicle, so dangerous driving can be detected. When accident occurs, Acceleratometer will detect signal and will send signal to AVR controller, microcontroller will enable airbag to blow and message with accident location is sent to preprogrammed numbers such as ambulance, police station, etc via GSM.

**2.4 TITLE: Web Application for Detecting Vehicle Accident and Tracking System**

**AUTHOR:** Raheem Ajetola, Ogunrinde, Mutiat Adebukola, Adeleye Sakirullah

**YEAR:** 2021

**DESCRIPTION:**

STRACT Nigeria is a country where we experience one of the highest incidences of road accidents in Africa. This is largely due to the deplorable conditions of our road network, incidences of armed robbery attack on the roads as well as lack of proper reporting or alert system for such accidents. The system provides an efficient and proper reporting as well as tracking of vehicles involved in an accident immediately after the accident occurrence i. e. The tracker and the reporter. The tracking module was devised with the use of a Simulated Sensor Program implemented in Visual Basic and a GPS/GPRS/GSM modem to sense accidents occurrences, track the location of accidents and to send alert and report to concerned agencies. Further reporting could also be done on the website through appropriate documentation interested members of the public.

**2.5 TITLE: Vehicle Tracking System Using GPS-GSM Accident Detection And Theft Security**

**AUTHOR:** Dr. Pradnya Mathurkar, Akansha B. Somkuwar, Ashwini R

**YEAR:** 2021

**DESCRIPTION:**

A vehicle tracking system is very useful for tracking the movement of a vehicle from any location at any time. An efficient vehicle tracking system is designed and implemented for tracking the movement of any equipped vehicle from any location at any time. The device is embedded inside a vehicle those positions is to be determined and tracked in real time. A vehicle tracking system is an electronic device installed in a vehicle to enable the owner or a third party to track the vehicle's location. An Arduino UNO is used to control the GPS receiver and GSM module. The vehicle tracking system uses the GPS module to get geographic coordinates at regular time interval. The GSM module is used to transmit and update the vehicle location to a database. This paper gives minute by minute update about vehicle location by sending SMS through GSM modem. This SMS contain latitude and longitude of the location of vehicle.. Thus, user able to continuously monitor a moving vehicle on demand using Smart phone and determine the estimated distance and time for the vehicle to arrive at a given destination.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

A system which used Android smart-phones and ODB-II(Operational Data Book) connection in a vehicle. When the system detects an accident, will sends an SMS to emergency contacts specified by the user, SMS will contain information about the accident and also a call to the emergency services is made automatically. All modern vehicles have ODB-II connection installed which transmits data about the vehicle in real-time such as acceleration, oil pressure, speed, etc. For the system to work a vehicle must support OBD-II standard. In US and this standard is necessary since, European countries have also implemented a version of this standard, so vehicles in the US and in Europe can use this system and is not available to all vehicle in other countries. Other than that, upgrading and maintenance of this system is very expensive process. In existing developed a system which made use of the accelerometer, GPS and microphone to detect accidents. Upon detection of an accident sends an emergency notification to the web server and also sends an SMS to the emergency contacts, emergency responders have to access the web server to find out about an accident.Their system made use of the same sensors and hardware that the algorithm presented in this research work makes use of, except for a few features. The main issue with this system is that the notifications are sent to a web server and responders needs to check the web server for accident notification, there is no system for individual responder that responds to the emergency to track victim’s location and also the system lacks the functionality to send emergency notification to the nearest emergency center in case there is more than one emergency center in the area.

**3.2 DISADVANTAGE**

* Upgrading and maintenance of this system is very expensive process.
* There is no system for individual responder that responds to the emergency to track victim’s location.
* The system lacks the functionality to send emergency notification to the nearest emergency center

**3.3 PROPOSED SYSTEM**

A large number of deaths are caused by Traffic accidents worldwide. The global crisis of road safety can be seen by observing the significant number of deaths and injuries that are caused by road traffic accidents. In many situations the family members or emergency services are not informed in time. This results in delayed emergency service response time, which can lead to an individual’s death or cause severe injury. The purpose of this work is to reduce the response time of emergency services in situations like traffic accidents or other emergencies like medical emergencies.In this case, Users usually lack the enthusiasm to forward accident alert messages.The accident hazards have been increased by the emergence of technology and also it leads to the rise of a frequency of road accidents. If an accident occurs, an immediate rescue can be considered as a tightrope walk between life and death.In this work, they endeavor to resolve these two issues through proposing an effective announcement network called Tron Coin, a novel privacy-preserving incentive announcement network based on Block-chain via an efficient arriving medical help to the respective places. On the one hand, Tron-Coin allows nondeterministic different signers (i.e., users) to generate the signatures and to send announcements anonymously in the non-fully trusted environment. On the other hand, with Block-chain, Tron-Coin motivates users with incentives to share accidental information.Thus the proposed accident alert system Web application with privacy preserving and non-repudiation Blockchain based incentive mechanism provides emergency responders with crucial information at the earliest possible time.

**3.4 ADVANTAGE**

* If an accident takes place, quick transmission of a message to the emergency services.
* To provide maximum assistance even in the unpopulated area.
* The ultimate goal of our system is to provide immediate location tracking of where the accident occurs.
* No need of any admin, system runs anonymously.

**3.5 SYSTEM ARCHITECTURE**

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

Application design

User registration

Achieving accident location

Alert message generation

Share accident location

Alert message to emergency service

Incentive mechanism

Block-chain Database

FIG 3.1 SYSTEM ARCHITECTURE

**CHAPTER 4**

**MODULE IMPLEMENTATION**

**4.1 MODULE LIST**

* **USER REGISTRATION**
* **BLOCKCHAIN-WALLET CREATION**
* **ACCIDENT ZONE**
* **USER SHARING ACCIDENT ZONE**
* **INCENTIVE**
  1. **IMPLEMENTATION**
     1. **USER REGISTRATION**
* The new user needs to register their details by providing some basic information.Whenever an accident happens, the user who witnesses an accident called initiator who is having an accident alert system application account may send that accident alert message through the application to the emergency services as well as all registered vehicle users in the application.
  + 1. **BLOCKCHAIN-WALLET CREATION**
* It is a kind of distributed database containing a number of blocks. The timestamp and cryptographic link can be contained by the blocks and that is connected with the earlier block. Besides that, it also contains a database that cannot be updated by any one user but it is visible to all.Wallet proposes a solution not as a replacement to the existing Internet based architecture, but as an extension to enable offline crypto currency transaction, for a more rapid adoption and user’s convenience.
  + 1. **ACCIDENT ZONE**
* The most likely reason for an individual's death in an accident is lack of the first aid provision that is because of emergency services not receiving information about accident in time. Emergency response time is extremely vital when it involves incidents involving vehicle accidents. Analysis shows that if they decrease just 1-minute in accident response time that can increase chances of saving an individual’s life up to six percent. In order to reduce response time, implementation of technologies would be necessary, which will help scale back response time and therefore reduce fatalities.

**4.2.4 USER SHARING ACCIDENT ZONE**

* Users usually lack the motivation to forward announcements.Proposing an effective announcement network called Tron Coin, a novel privacy-preserving incentive announcement network based on Block-chain via an efficient anonymous vehicular announcement aggregation protocol. With Blockchain, TronCoin motivates users with incentives to share traffic information. In addition, transactions and account information in TronCoin are tamper-resistant.

**4.2.5 INCENTIVE**

* Here propose a novel privacy-preserving incentive announcement network based on Blockchain, named TronCoin, which contains two parts, the announcement protocol and incentive mechanism. After the alert message containing the accident location received by the emergency services, they provide some incentives to the initiator. Transactions among users are forwarded based on the Blockchain technology. The valid transactions get added to the blocks on the chain.

**CHAPTER 5**

**SOFTWARE SYSTEM SPECIFICATION**

**5.1 HARDWARE SYSTEM CONFIGURATION:-**

* System : Core i3 2.4 GHz.
* Hard Disk : 500 GB.
* Monitor : 17 “ TFT
* Mouse : Logitech.
* Ram : 4 GB.

**5.2 SOFTWARE SYSTEM CONFIGURATION:-**

* Operating system : Windows 8
* Coding Language : Solidity
* IDE : Remix
* Block Chain : TRON
* Script : Vue.Js

**5.3 SOFTWARE ENVIRONMENT**

**5.3.1 DECENTRALIZED APPLICATIONS (DAPPS)**

## What Are Decentralized Applications (dApps)?

Decentralized applications (dApps) are digital applications or programs that exist and run on a [blockchain](https://www.investopedia.com/terms/b/blockchain.asp) or [peer-to-peer](https://www.investopedia.com/terms/p/peertopeer-p2p-service.asp) (P2P) network of computers instead of a single computer. DApps (also called "dapps") are outside the purview and control of a single authority. DApps—which are often built on the [Tron](https://www.investopedia.com/terms/e/ethereum.asp) platform—can be developed for a variety of purposes including gaming, finance, and social media.

### KEY TAKEAWAYS

* Decentralized applications—also known as "dApps" or "dapps"—are digital applications that run on a blockchain network of computers instead of relying on a single computer.
* Because dApps are decentralized, they are free from the control and interference of a single authority.
* Benefits of dApps include the safeguarding of user privacy, the lack of censorship, and the flexibility of development.
* Drawbacks include the potential inability to scale, challenges in developing a user interface, and difficulties in making code modifications.

## Understanding Decentralized Applications (dApps)

A standard web app, such as [Uber](https://www.investopedia.com/articles/personal-finance/111015/story-uber.asp) or Twitter, runs on a computer system that is owned and operated by an organization, giving it full authority over the app and its workings. There may be multiple users on one side, but the backend is controlled by a single organization.

DApps can run on a P2P network or a blockchain network. For example, BitTorrent, [Tor](https://www.investopedia.com/terms/t/tor.asp), and Popcorn Time are applications that run on computers that are part of a P2P network, whereby multiple participants are consuming content, feeding or seeding content, or simultaneously performing both functions.

In the context of cryptocurrencies, dApps run on a blockchain network in a public, [open-source](https://www.investopedia.com/terms/o/open-source.asp), decentralized environment and are free from control and interference by any single authority. For example, a developer can create a Twitter-like dApp and put it on a blockchain where any user can publish messages. Once posted, no one—including the app creators—can delete the messages.

## Advantages and Disadvantages of dApps

### Advantages

Many of the advantages of dApps center around the program's ability to safeguard user privacy. With decentralized apps, users do not need to submit their personal information to use the function the app provides. DApps use [smart contracts](https://www.investopedia.com/terms/s/smart-contracts.asp) to complete the transaction between two anonymous parties without the need to rely on a central authority.

Proponents interested in free speech point out that dApps can be developed as alternative social media platforms. A decentralized social media platform would be resistant to censorship because no single participant on the blockchain can delete messages or block messages from being posted.

Tron is a flexible platform for creating new dApps, providing the infrastructure needed for developers to focus their efforts on finding innovative uses for digital applications. This could enable rapid deployment of dApps in a variety of industries including banking and finance, gaming, social media, and online shopping.

### Disadvantages

The use of dApps is still in the early stages, and thus it is experimental and prone to certain problems and unknowns. There are questions as to whether the applications will be [able to scale](https://www.investopedia.com/terms/s/scalability.asp) effectively, particularly in the event that an app requires significant computations and overloads a network, causing network congestion.

The ability to develop a user-friendly interface is another concern. Most users of apps developed by traditional centralized institutions have an ease-of-use expectation that encourages them to use and interact with the app. Getting people to transition to dApps will require developers to create an [end-user experience](https://www.investopedia.com/terms/e/end-user.asp) and level of performance that rivals already popular and established programs.

The challenge of doing code modifications is another limitation of dApps. Once deployed, a dApp will likely need ongoing changes for the purposes of making enhancements or to correct bugs or security risks. According to Tron, it can be challenging for developers to make needed updates to dApps because the data and code published to the blockchain are hard to modify.2

Pros

* Promotes user privacy
* Resists censorship
* Flexible platform enables dApp development

Cons

* Experimental, may not be able to scale
* Challenges in developing a user-friendly interface
* Difficult to make needed code modifications

## What Are Tron dApps?

These are decentralized applications that are powered and developed using the Tron platform. Tron dApps use smart contracts for their logic. They are deployed on the Tron network and use the platform's blockchain for data storage.

## What Is the Difference Between a Centralized and Decentralized App?

A centralized app is owned by a single company. The application software for a centralized app resides on one or more servers controlled by the company. As a user, you'll interact with the app by downloading a copy of the app and then sending and receiving data back and forth from the company's server.

A decentralized app (also known as a dApp or dapp) operates on a blockchain or peer-to-peer network of computers. It enables users to engage in transactions directly with one another as opposed to relying on a central authority. The user of a dApp will pay the developer an amount of [cryptocurrency](https://www.investopedia.com/terms/c/cryptocurrency.asp) to download and use the program's source code. The source code is known as a smart contract, which allows users to complete transactions without revealing personal information.

## What Are Examples of Centralized and Decentralized Apps?

Well-known examples of centralized apps are Twitter, Facebook, Instagram, and Netflix. Banks and other financial institutions use centralized apps to allow their customers online access to their accounts.

Peepeth, a social network alternative to Twitter, is an example of a decentralized app. [Cryptokitties](https://www.investopedia.com/news/cryptokitties-are-still-thing-heres-why/) is a dApp game that allows users to buy and sell virtual cats. MakerDAO is a decentralized Tron service supporting the [stablecoin](https://www.investopedia.com/terms/s/stablecoin.asp) Dai and allows users to open a collateralized debt position (CDP).

**5.3.2 SMART CONTRACT:**

## What Is a Smart Contract?

A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized [blockchain](https://www.investopedia.com/terms/b/blockchain.asp) network. The code controls the execution, and transactions are trackable and irreversible.

Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism.

While blockchain technology has come to be thought of primarily as the foundation for bitcoin​, it has evolved far beyond underpinning the [virtual currency](https://www.investopedia.com/articles/forex/091013/future-cryptocurrency.asp).

### WHAT YOU NEED TO KNOW

* Smart contracts are self-executing lines of code with the terms of an agreement between buyer and seller automatically verified and executed via a computer network.
* Nick Szabo, an American computer scientist who invented a virtual currency called "Bit Gold" in 1998,1 defined smart contracts as computerized transaction protocols that execute terms of a contract.2
* Smart contracts deployed to blockchains render transactions traceable, transparent, and irreversible.Watch Now: What Is a Smart Contract?

## How Smart Contracts Work

Smart contracts were first proposed in 1994 by Nick Szabo, an American computer scientist who invented a virtual currency called "Bit Gold" in 1998, fully 10 years before the invention of bitcoin. In fact, Szabo is often rumored to be the real Satoshi Nakamoto, the anonymous inventor of bitcoin, which he has denied.1

Szabo defined smart contracts as computerized transaction protocols that execute terms of a contract.2 He wanted to extend the functionality of electronic transaction methods, such as POS (point of sale), to the digital realm.

In his paper, Szabo also proposed the execution of a contract for synthetic assets, such as derivatives and bonds. Szabo wrote: "These new securities are formed by combining securities (such as bonds) and derivatives (options and futures) in a wide variety of ways. Very complex term structures for payments can now be built into standardized contracts and traded with low transaction costs, due to computerized analysis of these complex term structures."2

**5.3.3 TRON**

## What Is Tron?

Tron is a [blockchain](https://www.investopedia.com/terms/b/blockchain.asp)-based decentralized digital platform with its own cryptocurrency, called Tronix or TRX. Founded in 2017 by a Singapore non-profit organization, the Tron Foundation, Tron aims to host a global entertainment system for the cost-effective sharing of digital content.

Initially marketed primarily in Asia, Tron had now gone global. The platform had more than 50 million accounts as of August 2021.1

Founded by Justin Sun, now its CEO, Tron has offices in Singapore and San Francisco. Born in 1990, Sun also is the CEO of BitTorrent, the file-sharing program.

* Tron is a blockchain-based digital platform that primarily hosts entertainment applications.
* It has its own in-house cryptocurrency, called Tronix or TRX.
* TRX ranks 31st among cryptocurrencies in market capitalization.

## Understanding Tron

Tron uses the features of the blockchain and [peer-to-peer](https://www.investopedia.com/terms/p/peertopeer-p2p-service.asp) (P2P) network technology to eliminate the middleman and allow content creators to sell their work directly to consumers. Software developers use the Solidity programming language to create apps that are hosted on the Tron platform.2

The currency used on the network is Tronix, or TRX. Users of the network use TRX to directly pay the content creators to access their applications. Content creators do not pay a transaction fee to Tron. (TRX transactions also are free on the platform.)

Users can store their crypto-assets in a desktop, Web, or hardware wallet.

### Technical

TRON is one of the largest blockchain-based operating systems in the world.

High-throughput

High throughput is achieved by improving the TPS in TRON, which has surpassed Bitcoin and Tron, to a daily-use practical degree.

High-scalability

Applications are given a wider variety of ways to be deployed in TRON because of its scalability and highly effective smart contract. It can support enormous numbers of users.

High-availability

More reliable network structure, user asset, intrinsic value and a higher degree of decentralization consensus come with an improved rewards distribution mechanism.

### Wallets

Because Tron is a popular [blockchain](https://www.thebalance.com/best-blockchain-etfs-5220329), it’s widely supported by many software and hardware wallets. That includes the hardware wallet Ledger and the Tron wallet made by the Tron Foundation.

As with all cryptocurrency wallets, it’s extremely important to keep your passwords and access information private and somewhere safe. If you lose access to your wallet or make a mistake when sending, you could lose access to your TRX and any other Tron-compatible assets held in your wallet.

### Special Features of Tron

Tron utilizes a delegated [proof-of-stake](https://www.thebalance.com/proof-of-stake-pos-5196135) system, which means it uses far less power than competing currencies like Bitcoin. Tron’s architecture gives the Tron network the ability to handle far more transactions at a time than proof-of-work systems such as Bitcoin, which rely on a massive network of cryptocurrency miners.

The higher throughput isn’t just a small improvement over older, larger cryptocurrency networks. Tron claims the ability to handle 10,000 transactions per second with no transaction fees.4

In its 2018 white paper, Tron claimed its delegated proof-of-work mechanism allowed for 2,000 transactions per second compared to three for Bitcoin and 15 for Tron.5

Higher throughput and lower costs make it attractive for smart contracts and Ðapp

**5.3.4 TRON-WEB**

## What is Tron-Web?

TronWeb is a Javascript library of TRON full node's API functions. To include the TronWeb library in your Dapp.

## Providers

Providers are used to provide methods of transport for requests. At the moment, Tron-Web only supports a single transport - HTTP.

### HTTP Provider

If you wish to provide more options to the node endpoints, such as user authentication or custom headers, you can pass a TronWeb.providers.HttpProvider instance.

## Promises vs. Callbacks

Tron-Web has been designed to be as compatible with Tron's Web3 implementation as possible. This means we support callbacks in the majority of our functions.

That doesn't mean Tron-Web is limited in any shape or form. This means that we also support promises. By default, calling a function will return a promise. If you provide a callback function, then a promise will not be returned and the callback will act as the response trigger instead.

**5.3.5 VUE JS**

**VueJS** is an open source progressive JavaScript framework used to develop interactive web interfaces. It is one of the famous frameworks used to simplify web development. VueJS focusses on the view layer. It can be easily integrated into big projects for front-end development without any issues.

The installation for VueJS is very easy to start with. Any developer can easily understand and build interactive web interfaces in a matter of time. VueJS is created by Evan You, an ex-employee from Google. The first version of VueJS was released in Feb 2014. It recently has clocked to 64,828 stars on GitHub, making it very popular.

## Features

Following are the features available with VueJS.

### Virtual DOM

VueJS makes the use of virtual DOM, which is also used by other frameworks such as React, Ember, etc. The changes are not made to the DOM, instead a replica of the DOM is created which is present in the form of JavaScript data structures. Whenever any changes are to be made, they are made to the JavaScript data structures and the latter is compared with the original data structure. The final changes are then updated to the real DOM, which the user will see changing. This is good in terms of optimization, it is less expensive and the changes can be made at a faster rate.

### Data Binding

The data binding feature helps manipulate or assign values to HTML attributes, change the style, assign classes with the help of binding directive called **v-bind** available with VueJS.

### Components

Components are one of the important features of VueJS that helps create custom elements, which can be reused in HTML.

### Event Handling

**v-on** is the attribute added to the DOM elements to listen to the events in VueJS.

### Animation/Transition

VueJS provides various ways to apply transition to HTML elements when they are added/updated or removed from the DOM. VueJS has a built-in transition component that needs to be wrapped around the element for transition effect. We can easily add third party animation libraries and also add more interactivity to the interface.

### Computed Properties

This is one of the important features of VueJS. It helps to listen to the changes made to the UI elements and performs the necessary calculations. There is no need of additional coding for this.

### Templates

VueJS provides HTML-based templates that bind the DOM with the Vue instance data. Vue compiles the templates into virtual DOM Render functions. We can make use of the template of the render functions and to do so we have to replace the template with the render function.

### Directives

VueJS has built-in directives such as v-if, v-else, v-show, v-on, v-bind, and v-model, which are used to perform various actions on the frontend.

### Watchers

Watchers are applied to data that changes. For example, form input elements. Here, we don’t have to add any additional events. Watcher takes care of handling any data changes making the code simple and fast.

### Routing

Navigation between pages is performed with the help of vue-router.

### Lightweight

VueJS script is very lightweight and the performance is also very fast.

### Vue-CLI

VueJS can be installed at the command line using the vue-cli command line interface. It helps to build and compile the project easily using vue-cli.

# **5.3.6 WEB3 .JS TRON JAVASCRIPT API**

web3.js is a collection of libraries which allow you to interact with a local or remote Tron node, using a HTTP or IPC connection. The web3 JavaScript library interacts with the Tron blockchain. It can retrieve user accounts, send transactions, interact with smart contracts, and more.

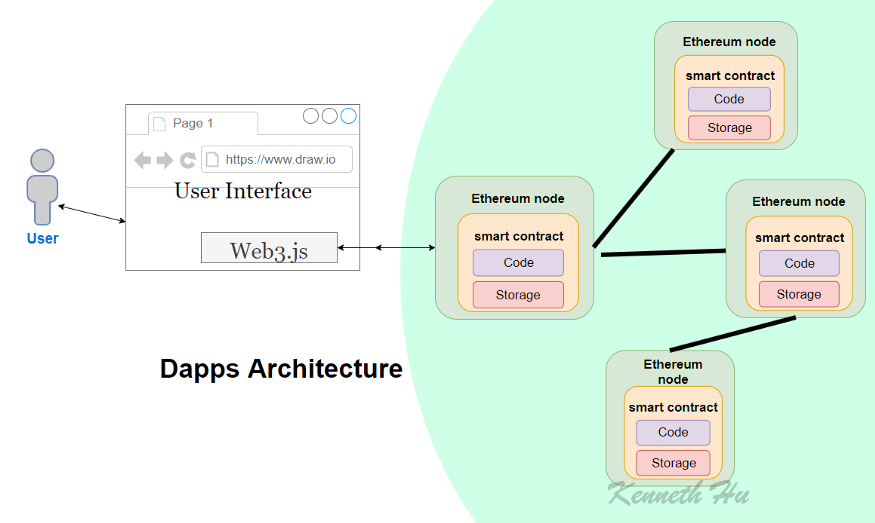


FIG 5.1 DAPP ARCHITECTURE

Web3.js API Type

eth: Tron blockchain related methods

net: Node’s network status

personal: Account functions and sending

db: Get/put for local LevelDB

shh: P2P messaging using Whisper

First, we check if we are using modern dapp browsers or the more recent versions of [MetaMask](https://github.com/MetaMask" \t "_blank) where an Tron provider is injected into the window object. If so, we use it to create our web3 object, but we also need to explicitly request access to the accounts with Tron.enable().

If the **Tron** object does not exist, we then check for an injected **web3**instance. If it exists, this indicates that we are using an older dapp browser (like [Mist](https://github.com/ethereum/mist) or an older version of MetaMask). If so, we get its provider and use it to create our web3 object.

If no injected web3 instance is present, we create our web3 object based on our local provider. (This fallback is fine for development environments, but insecure and not suitable for production.)

**5.3.7 TRONLINK WALLET**

Most SecureAsset Management .Decentralized wallet: data security ensured through local storage of private key and multi-layer algorithm encryption.Hot and cold wallets: sign in your hot wallet and store in your cold wallet to achieve physical isolation and double protection.Node stability and reliability: free switching among nodes and auto reconnection supported.Most Well-RoundedFeatures of TRON Full coverage of tokens: we support TRX, TRC-10, TRC-20 and TRC-721 tokens while serving global TRON users. Unique on-chain mechanism: Stake TRX to gain resources and voting rights, consume resources instead of gas fees and vote for SRs to win considerable TRX.

Available SDK and testnet: Serve developers who can help expand TRON's ecosystem through joint efforts. Most ConvenientUser Experience .Simple and user-friendly: we support creating/importing wallets with one click, HD wallets, importing Ledger wallets via bluetooth and creating muiltiple wallet accounts. Multisignature feature: multiple accounts to manage one set of assets with various use cases and a new way of signing. Switch and manage with ease: TronLink is available in the forms of Web app and browser extension, catering to your needs in different use cases. Come and enjoy TRON DApps whenever and wherever you want!

**CHAPTER 6**

**SYSTEM DESIGN**

**6.1 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

**GOALS:**

The Primary goals in the design of the UML are as follows:

* Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
* Provide extendibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development process.

**6.2 USE CASE DIAGRAM:**

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating. So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour.

In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships.

The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used. A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the case and will often be accompanied by other types of diagrams as well.

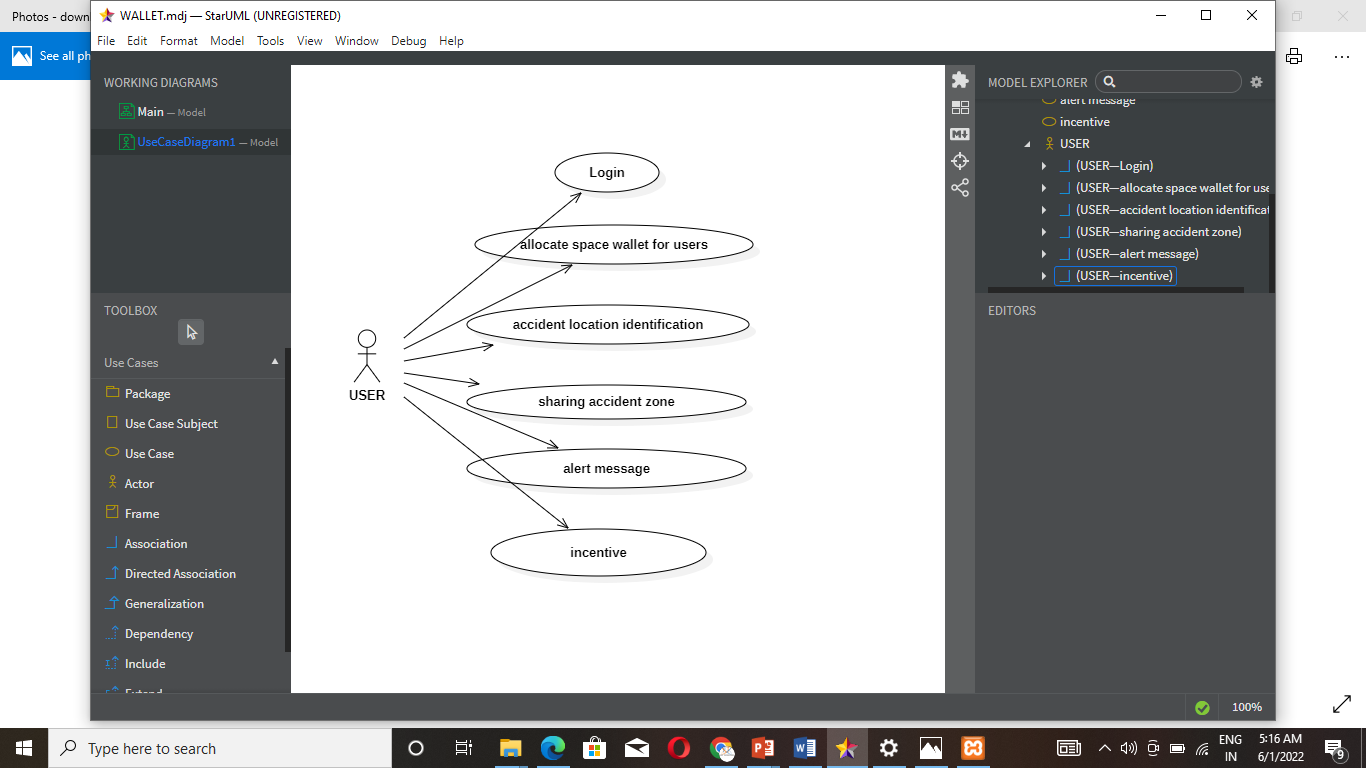


FIG 6.1 USE CASE DIAGRAM

**6.3 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class

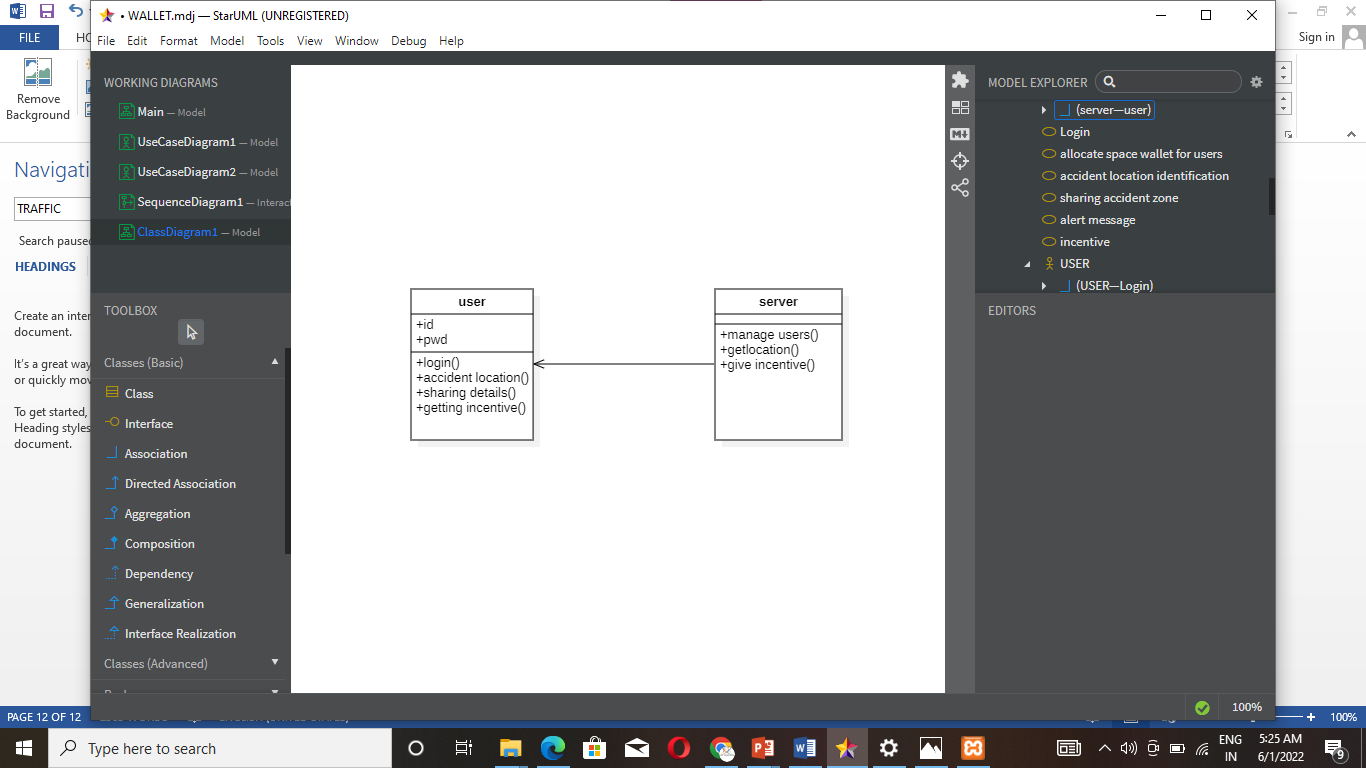


FIG 6.2 CLASS DIAGRAM

**6.4 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

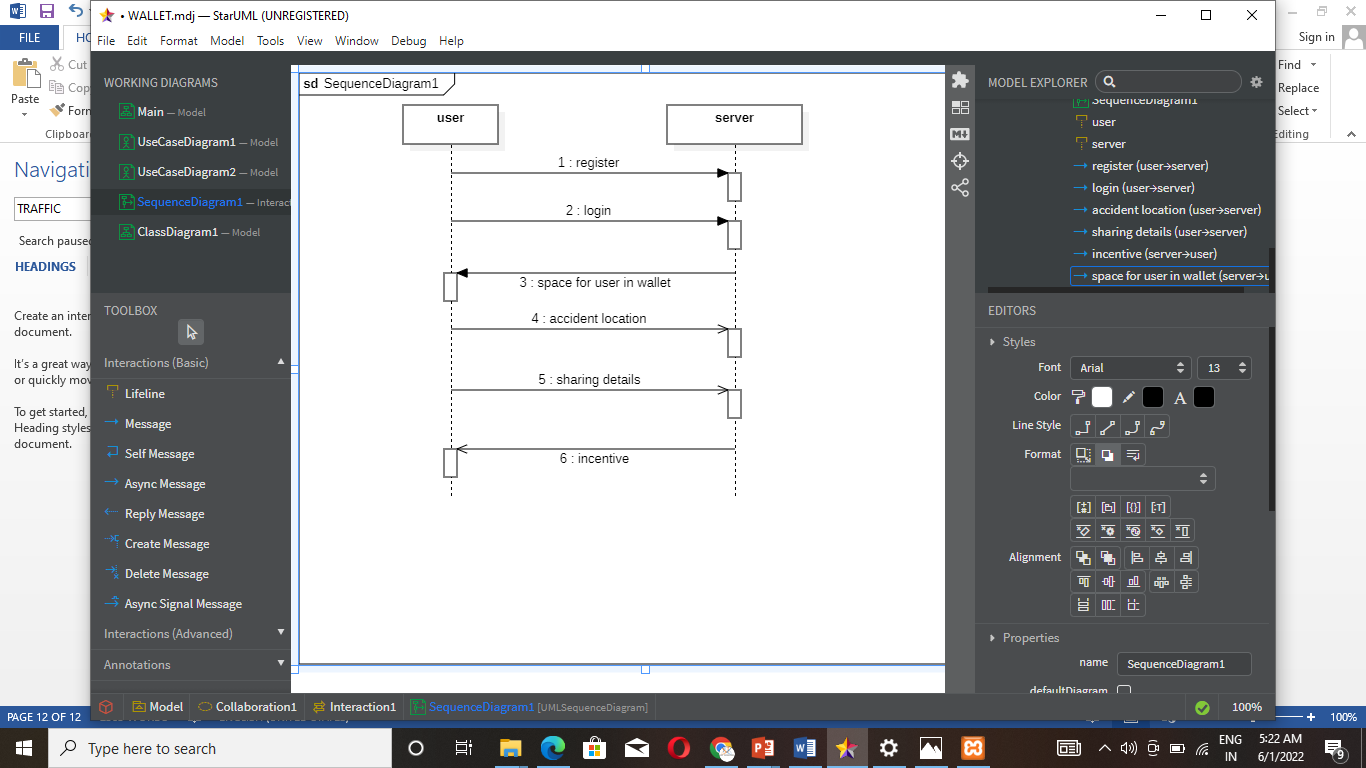


FIG 6.3 SEQUENCE DIAGRAM

**6.5 DATA FLOW DIAGRAM:**

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software.

Data objects represented by labeled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

Register and login

Block chain wallet creation

Accident location

User sharing accident zone

Provide incentive using bit coin

FIG 6.4 DATA FLOW DIAGRAM

### SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**7.1 TYPES OF TESTING**

**7.1.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results

**7.1.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**7.1.3 FUNCTIONAL TESTING**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.1.4 SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**7.1.5 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.1.6 BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**7.1.7 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**CHAPTER 8**

**SCREENSHOTS**

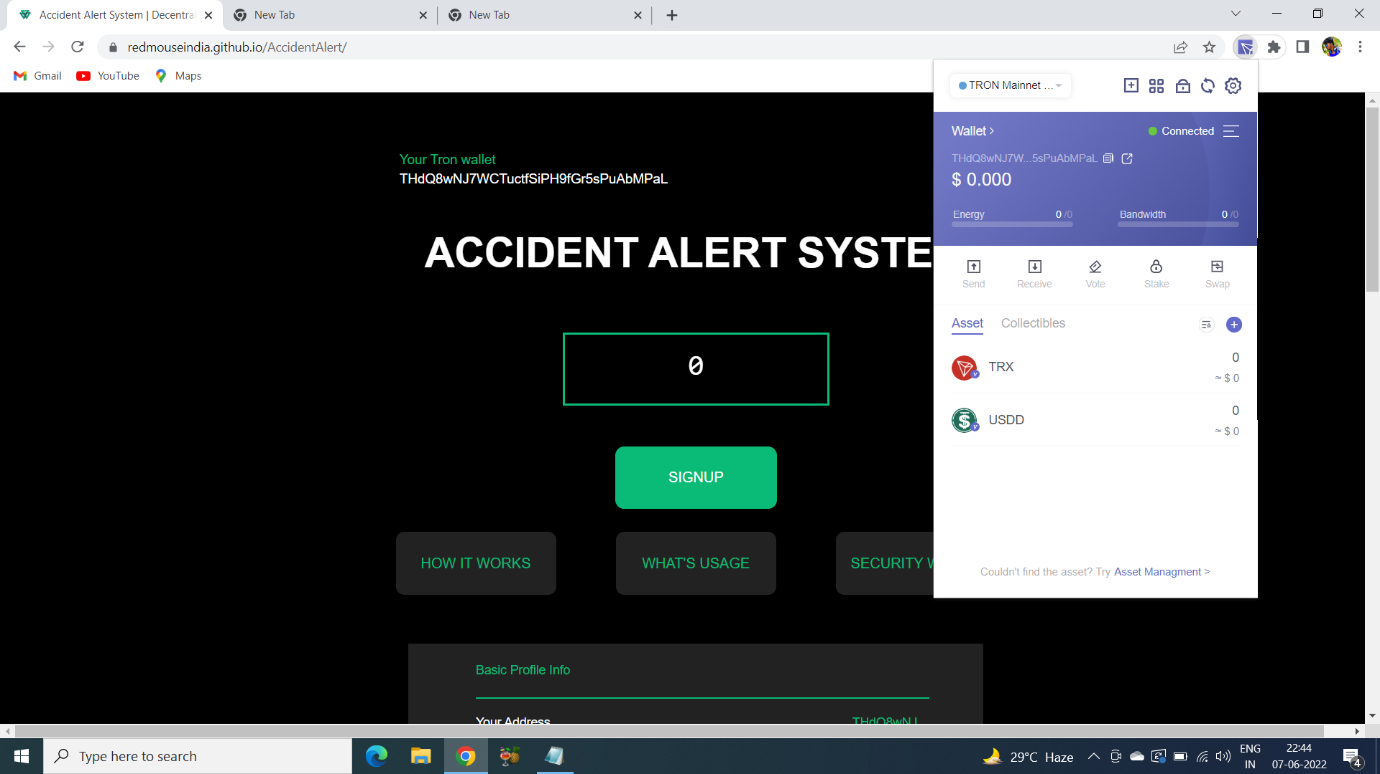


FIG 8.1 HOME PAGE OF ACCIDENT ALERT SYSTEM

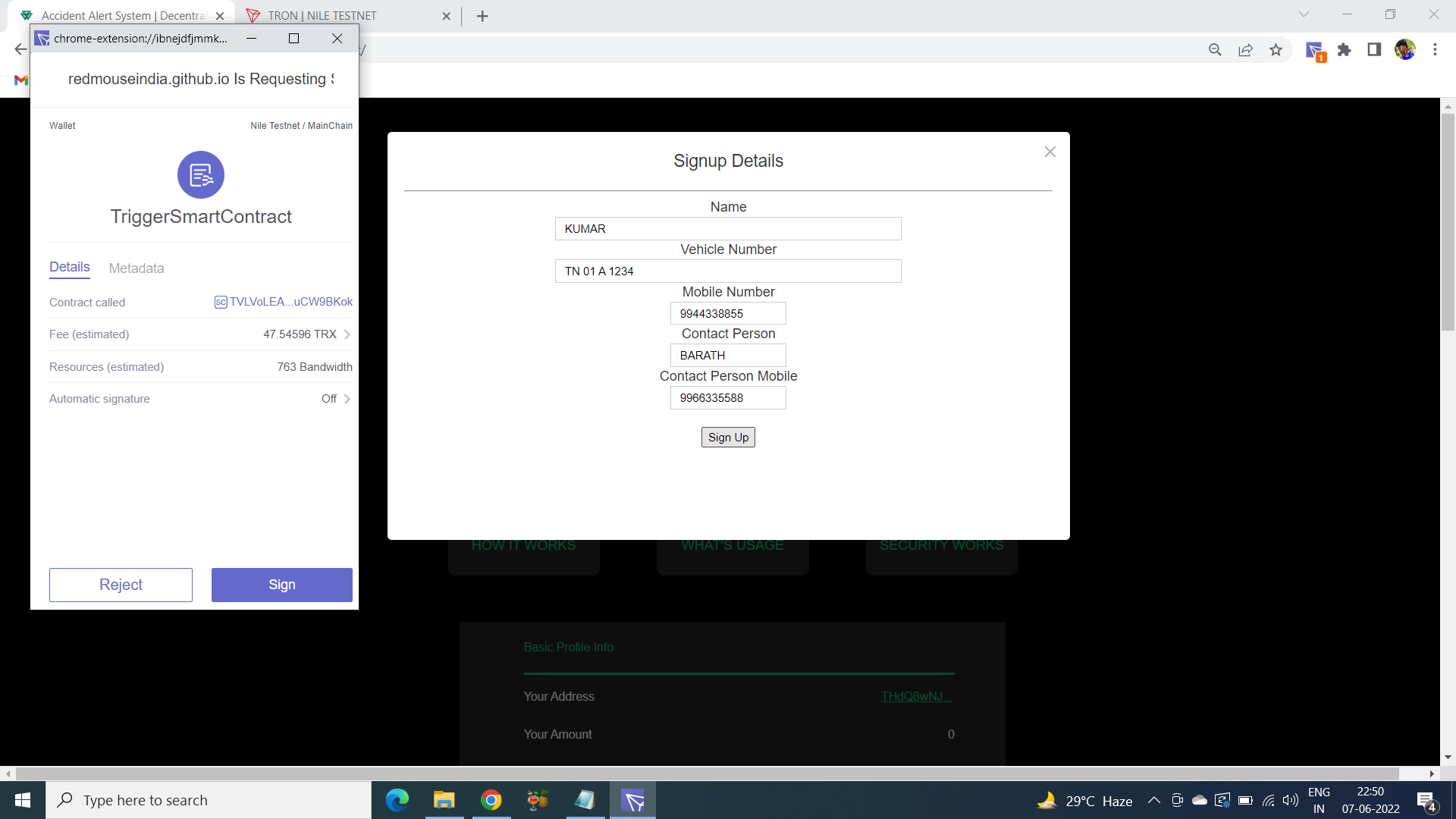


FIG 8.2 USER MANDATORY SIGNUP DETAILS

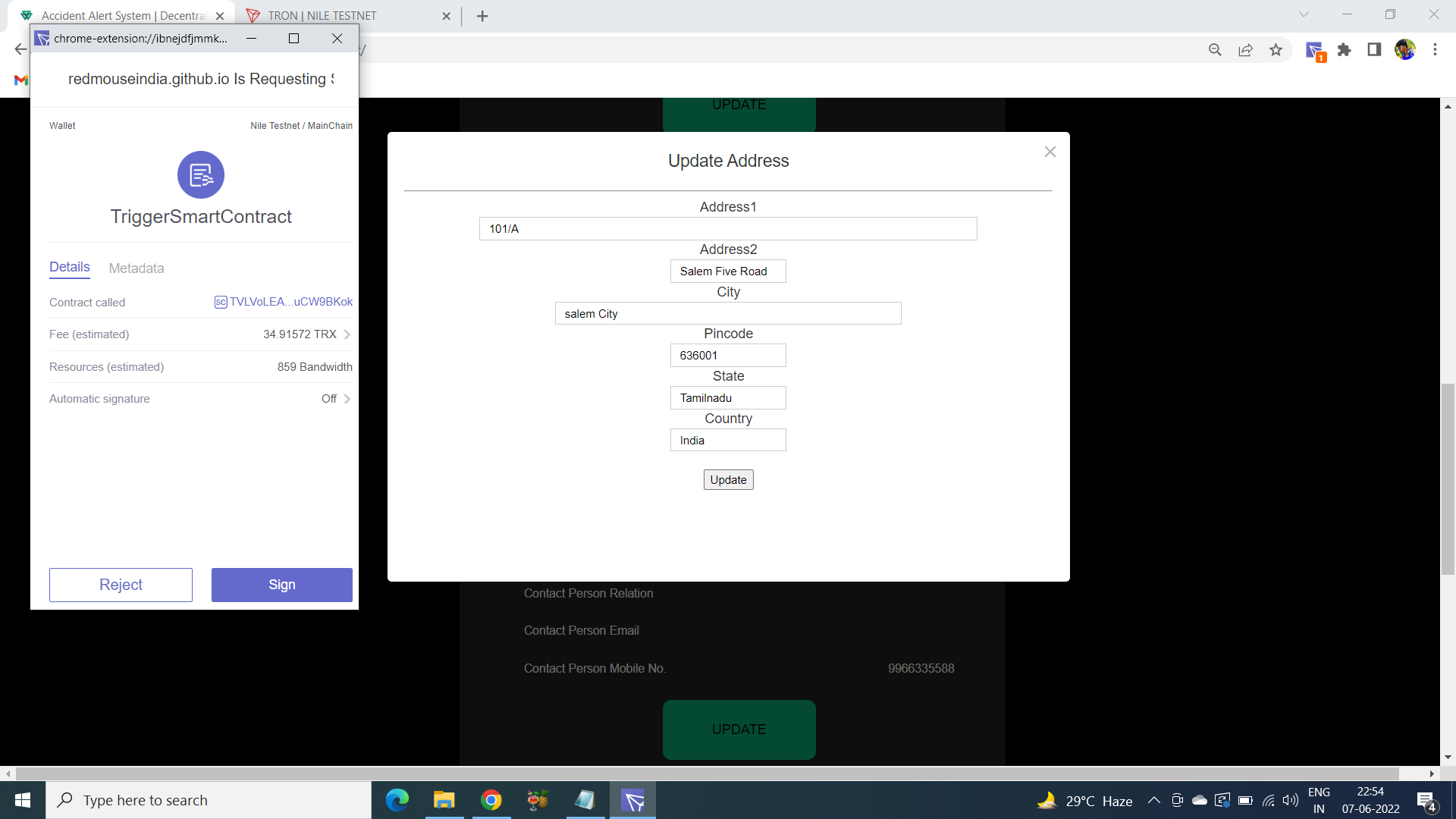
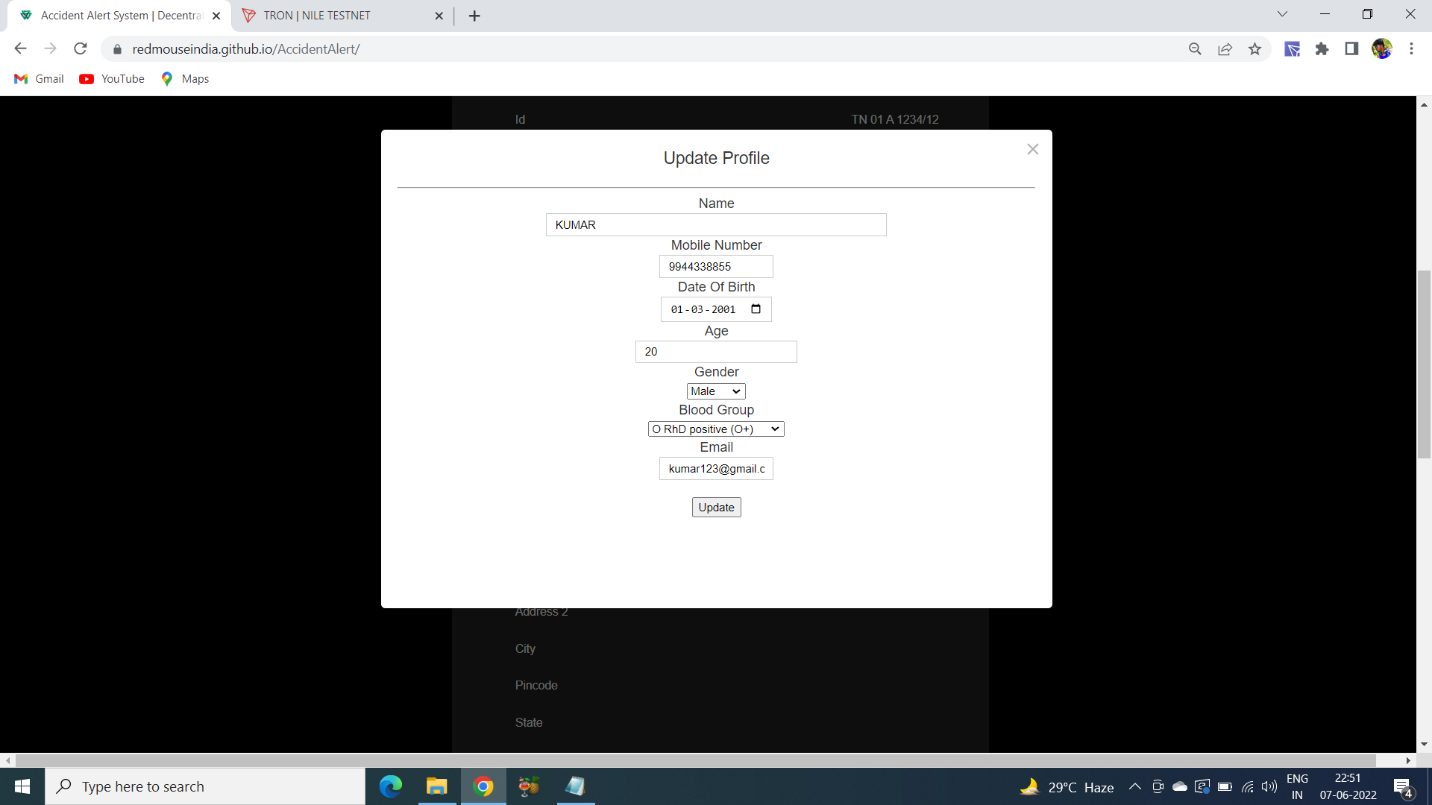


FIG 8.3 UPDATE USER INFORMATION

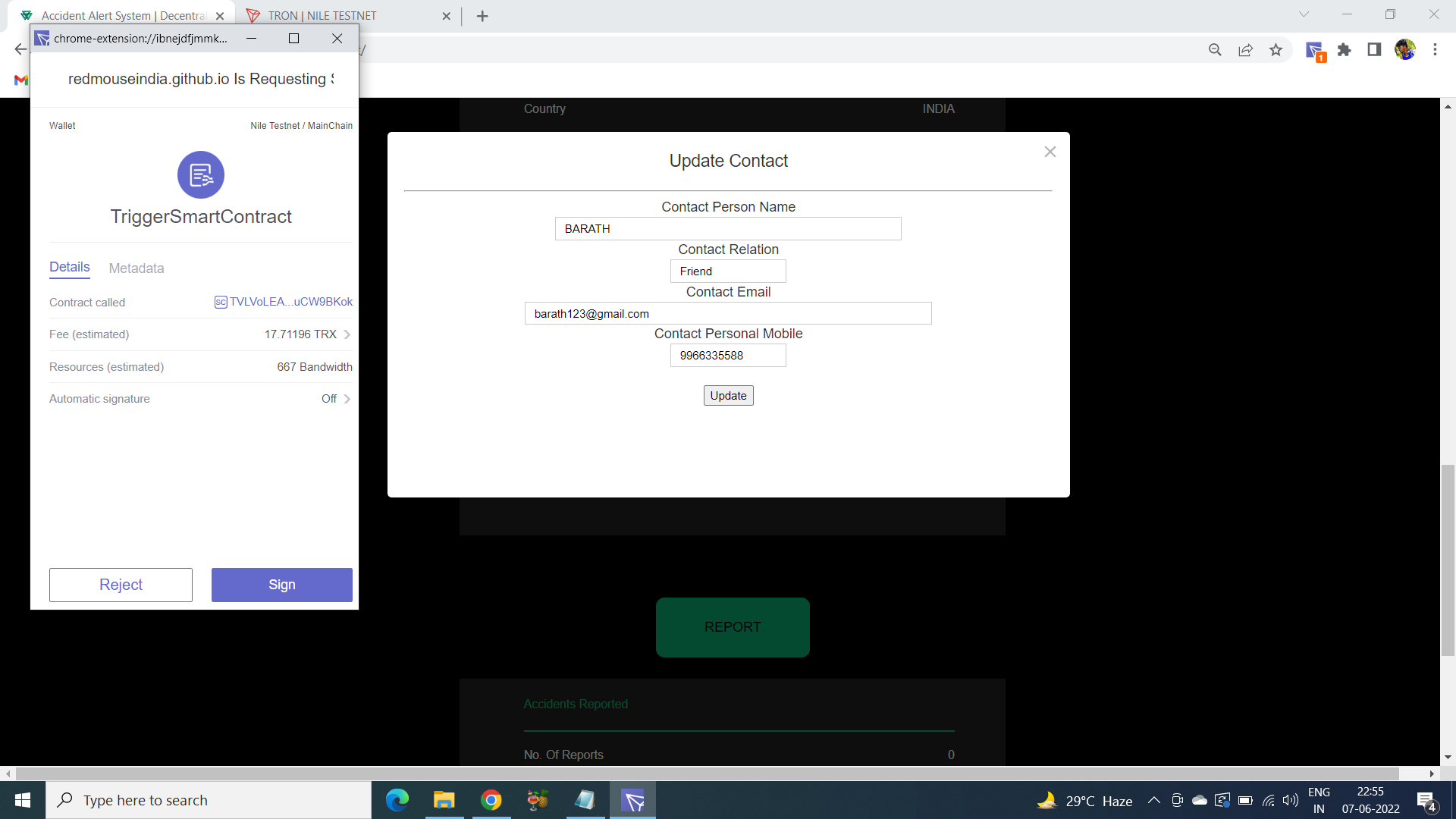


FIG 8.4 EMERGENCY CONTACT INFORMATION

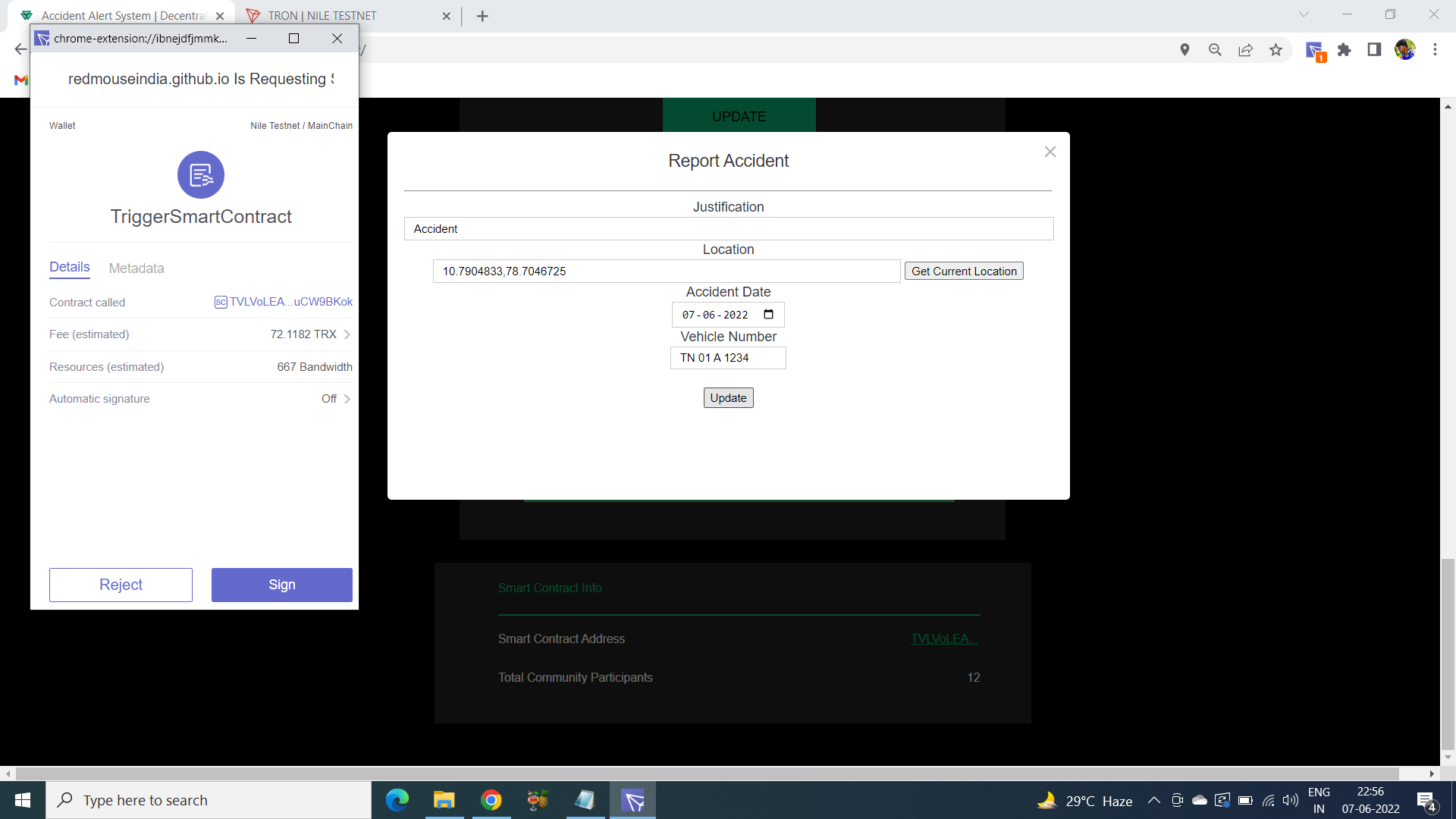


FIG 8.5 REPORT AN ACCIDENT

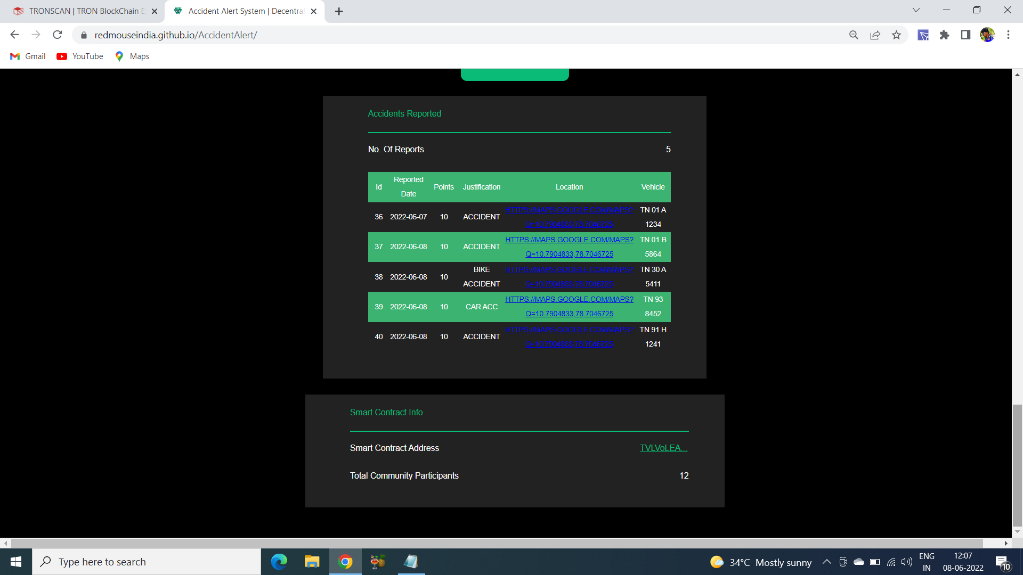
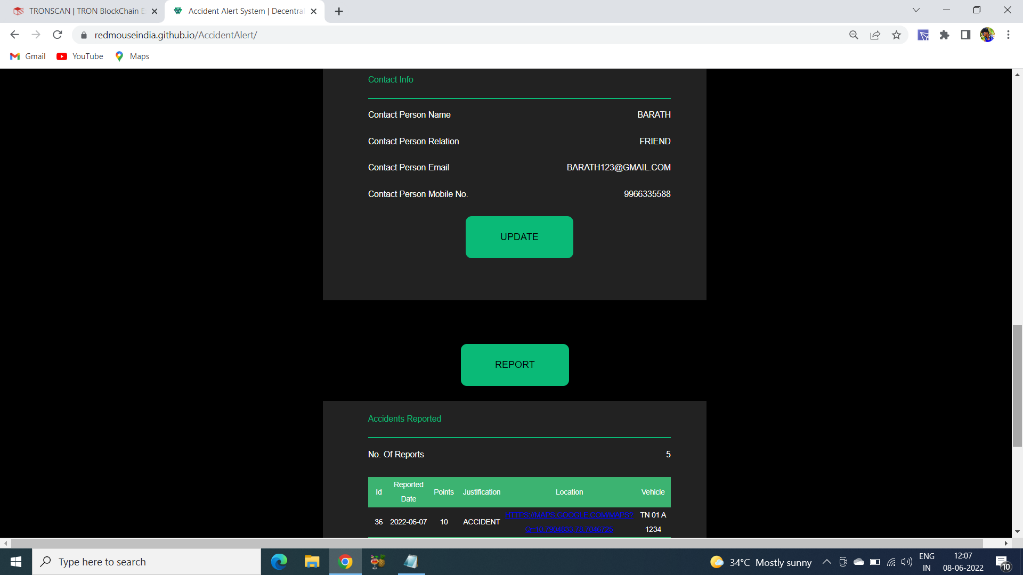
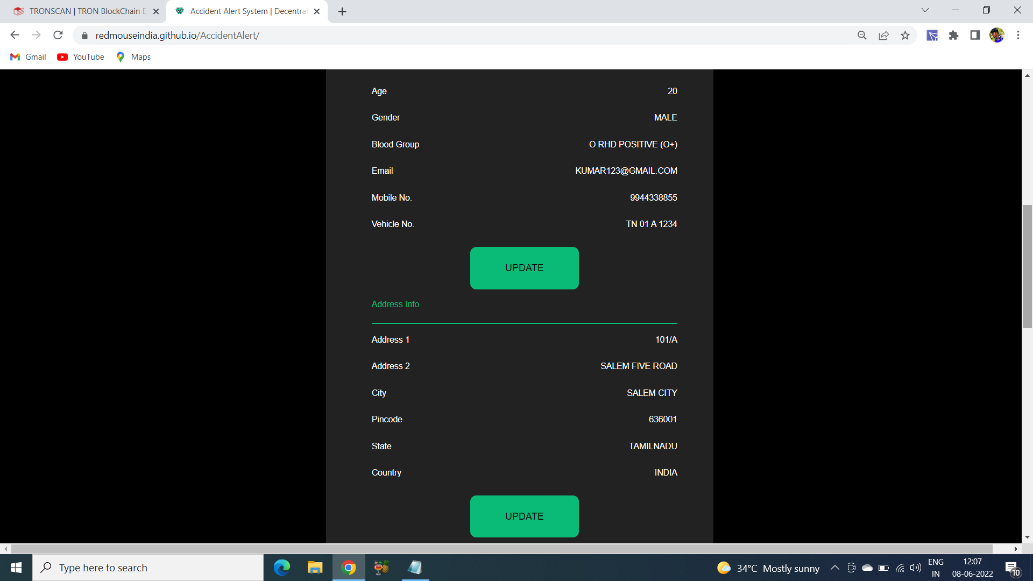
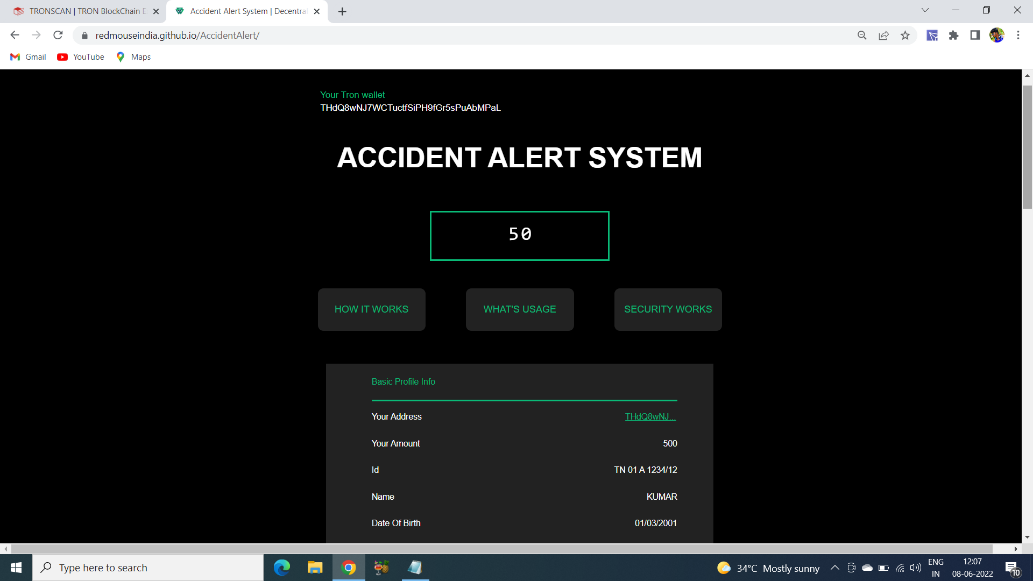


FIG 8.6 RESULT OF AN ACCIDENT ALERT SYSTEM

**CHAPTER 9**

**CONCLUSION AND FUTURE ENHANCEMENT**

**9.1 CONCLUSION**

The proposed accident alert system Web application with privacy-preserving and no repudiation Block chain based incentive mechanism maintains the reliability of alert messages without revealing users’ privacy and is reliable and efficient in the non-fully-trusted environment. This application ensures the fast arrival of emergency services to the accident location. Due to its low cost and easy installation, the application proves to be better efficiency. Undoubtedly, Block chain offers high security to the Web applications and helps application developers to add feature confidentiality in the application. By introducing a Block chain technology into the Web application to secure transactions and it is a practical solution for Web devices to achieve security, efficiency, and scalability of data collection, processing, and sharing. Analysis and performance evaluation shown that, the proposed system is feasible and adequate for a secure accident alert message system allowing application users to retain complete privacy while at the same time offering non repudiation security.

**9.2 FUTURE WORK**

In future work will plan to improve the key management and the coin balance in Tron Coin. Designing more effective trading propositions is also being investigated. In future work will strengthen the data collection of real applications.

**CHAPTER 10**

**SAMPLE SOURCE CODE**

**Front-end :**

<!DOCTYPE html>

<html lang="en">

<meta http-equiv="content-type" content="text/html;charset=UTF-8" />

<head> <meta charset="UTF-8">

<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Accident Alert System | Decentralized | Smart Contract based Project</title>

<link rel="stylesheet" href="modules/site/main/assets/styles/odometer-theme-digital.css">

<link rel="stylesheet" href="modules/site/main/assets/styles/styles23b7.css?1599046343">

<link rel="icon" href="modules/site/main/assets/images/favicon.png" type="image/png" /> <meta name="sc.balance" content="" />

<script>

function alphaOnly(event) {

var key = event.keyCode;

return ((key >= 65 && key <= 90) || key == 8 || key == 9);

}; </script>

<style>

[v-cloak] {

display: none;

}

</style>

<style>

table, th, td {

border: 0px solid #FFF;

border-collapse: collapse;

line-height: 2;

font-weight: 500;

}

td {

color: #fff;

}

.style-0 {

background-color: transparent;

}

.style-1 {

background-color: #3CB371;

}

a:link, a:visited {

background-color: transparent;

color: #0000EE;

padding: 0px 2px;

text-align: center;

text-decoration: underline;

display: inline-block;

}

a:hover, a:active {

background-color: transparent;

}

/\* Chrome, Safari, Edge, Opera \*/

input::-webkit-outer-spin-button,

input::-webkit-inner-spin-button {

-webkit-appearance: none;

margin: 0;

}

/\* Firefox \*/

input[type=number] {

-moz-appearance: textfield;

}

form.errors {

:invalid {

outline: 2px solid red;

}

} </style>

<link rel="stylesheet" href="style.css">

<script>

function getLocation() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition);

} else {

alert("Geolocation is not supported by this browser.");

}

}

function showPosition(position) {

//var embedLink = "https://www.google.com/maps/embed/v1/place?key=AIzaSyDUROz7BPJt\_FrMYvc6YgrSLZY\_NlyU5Sg&q=" + position.coords.latitude + "," + position.coords.longitude;

//var srcLink = "https://maps.google.com/maps?q=" + position.coords.latitude + "," + position.coords.longitude;

var srcLinkValue = position.coords.latitude + "," + position.coords.longitude;

document.getElementById("txtLocation").value = srcLinkValue;

//document.getElementById("iframe1").src=embedLink;

//document.getElementById("alink").href = srcLinkValue;

//return false;

}

function ValidateAge(el)

{

var result;

var res = Number(e1.value);

if (res >= 1 || res <=120 )

{

result= 0;

} else if (el.value.length == 0)

{

result = 0;

} else {

result= -1;

}

if( result == -1)

{

alert("Invalid Age");

var onblurevent = el.onblur;

el.onblur = "";

setTimeout(function() {

el.focus();

el.onblur = onblurevent;

},0);

}

return false;

}

function ValidateDate(el)

{

var result;

if (/^([0-9]{2})\/([0-9]{2})\/([0-9]{4})$/.test(el.value))

{

result= 0;

} else if (el.value.length == 0)

{

result = 0;

} else {

result= -1;

}

if( result == -1)

{

alert("Invalid Date");

var onblurevent = el.onblur;

el.onblur = "";

setTimeout(function() {

el.focus();

el.onblur = onblurevent;

},0);

}

return false;

}

function ValidatePincode(el)

{

var result;

if (/^[1-9][0-9]{5}$/gi.test(el.value))

{

result= 0;

} else if (el.value.length == 0)

{

result = 0;

} else {

result= -1;

}

if( result == -1)

{

alert("Invalid Pincode");

var onblurevent = el.onblur;

el.onblur = "";

setTimeout(function() {

el.focus();

el.onblur = onblurevent;

},0);

}

return false;

}

function ValidateMobile(el)

{

var result;

if (/^[6-9]\d{9}$/gi.test(el.value))

{

result= 0;

} else if (el.value.length == 0)

{

result = 0;

} else {

result= -1;

}

if( result == -1)

{

alert("Invalid Mobile Number");

var onblurevent = el.onblur;

el.onblur = "";

setTimeout(function() {

el.focus();

el.onblur = onblurevent;

},0);

}

return false;

}

function ValidateEmail(el)

{

var result;

if (/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/.test(el.value))

{

result= 0;

} else if (el.value.length == 0)

{

result = 0;

} else {

result= -1;

}

if( result == -1)

{

alert("Invalid Email Address");

var onblurevent = el.onblur;

el.onblur = "";

setTimeout(function() {

el.focus();

el.onblur = onblurevent;

},0);

}

return false;

}

//document.getElementById("txtMobile1").addEventListener("blur", function () { alert('Hi'); });

</script>

</head>

<body bgcolor="#000">

<section class="fxt-template-animation fxt-template-layout27">

<main id="App" v-cloak>

<center>

<div class="header">

<div class="header\_\_wrap">

<div class="header\_\_item" v-if="tron.account">

<div class="header\_\_label">Your Tron wallet</div>

<div class="header\_\_value"><span v-text="tron.account"></span>

</div>

</div>

<div class="header\_\_grow"></div>

</div>

</div>

</center>

<section >

<!--

<img class="top-logo" src="modules/site/main/assets/images/top.png?2" width="125" style=" padding-top: 0px; padding-bottom: 0px;" /><br>

<img class="top-logo" src="modules/site/main/assets/images/logoc81e-green.png?2" width="300" style=" padding-top: 0px; padding-bottom: 0px;" />

* <br/> <br/> <br/>

<h2 class="join" style=" text-align:center; color: #FFF">Accident Alert System</h2>

<br/><br/>

<!-- <img class="top-heading-mobile" width="50%" style=" margin-bottom: 50px;" src="modules/site/main/assets/images/logo-mobile.png" /> -->

<div class="container" v-show="tab == 'main'">

<div class="counter-wrapper">

<div class="odometer" ref="odometer">{{ contract.total\_users }}</div>

</div>

<center>

<div>

<button class="button" @click="signup()" v-if="user.id == 0">

<div class="button-content" style=" color: #fff; background-color: #0aba77; font-weight: 500">Signup</div>

</button>

</div>

</center>

<div class="menu">

<div>

<button class="button" @click.prevent="tab = 'how'">

<div class="button-content" style=" color: #0aba77; border: 1px solid #000;">How it Works</div>

</button>

</div>

<div>

<button class="button" @click.prevent="tab = 'usage'">

<div class="button-content" style=" color: #0aba77; border: 1px solid #000;">What's Usage</div>

</button>

</div>

<div>

<button class="button" @click.prevent="tab = 'security'">

<div class="button-content" style=" color: #0aba77; border: 1px solid #000;">Security Works</div>

</button>

</div>

</div>

</div>

</section>

<section class="section calc bg-hidden" v-show="tab == 'main'">

<div class="container">

<div class="panel-info" style="border: 0px solid #000; background-color: #222; margin: 0;">

<div>

<div class="panel-info-item" style=" line-height:0;"><span style=" color: #0aba77"> Basic Profile Info </span> </div>

<hr style=" line-height:1; " color="#0aba77" /><br/>

<div class="panel-info-item" style=" line-height:1;" ><span>Your Address</span><a v-if="tron.account" :href="'https://shasta.tronscan.org/#/address/' + tron.account" target="\_blank" style=" color: #0aba77;">{{ tron.account.substr(0, 8) }}...</a><span v-if="!tron.account" class="ether-value">---</span></div>

<div class="panel-info-item" style=" line-height:1;" ><span>Your Amount</span> <span>{{ user.total\_amount }} </span></div>

<div class="panel-info-item" style=" line-height:1;" ><span>Id</span><span> {{ user.vehicle\_no }}/{{ user.id }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Name</span><span>{{ user.username }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Date of Birth</span><span>{{ user.dob }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Age</span><span>{{ user.age }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Gender</span><span>{{ user.gender }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Blood Group</span><span>{{ user.blood\_group }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Email</span><span>{{ user.email }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Mobile No.</span><span>{{ user.mobile\_no }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Vehicle No.</span><span>{{ user.vehicle\_no }} </span> </div>

<!--

<div class="panel-info-item" style=" line-height:1;" ><span>RC Book</span><span>{{ user.rc\_book }} </span> </div>

-->

<button class="button" @click="updateprofile()" v-if="user.id > 0">

<div class="button-content" style=" color: #000; background-color: #0aba77; font-weight: 500">Update</div>

</button>

<div class="panel-info-item" style=" line-height:0;"><span style=" color: #0aba77"> Address Info </span> </div> <hr style=" line-height:1; " color="#0aba77" /><br/>

<div class="panel-info-item" style=" line-height:1;" ><span>Address 1</span><span>{{ user.address1 }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Address 2</span><span>{{ user.address2 }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>City</span><span>{{ user.city }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Pincode</span><span>{{ user.pincode }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>State</span><span>{{ user.state }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Country</span><span>{{ user.country }} </span> </div>

<button class="button" @click="updateaddress()" v-if="user.id > 0">

<div class="button-content" style=" color: #000; background-color: #0aba77; font-weight: 500">Update</div>

</button>

<div class="panel-info-item" style=" line-height:0;"><span style=" color: #0aba77"> Contact Info </span> </div> <hr style=" line-height:1; " color="#0aba77" /><br/>

<div class="panel-info-item" style=" line-height:1;" ><span>Contact Person Name</span><span>{{ user.cp\_name }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Contact Person Relation</span><span>{{ user.cp\_relation }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Contact Person Email</span><span>{{ user.cp\_email }} </span> </div>

<div class="panel-info-item" style=" line-height:1;" ><span>Contact Person Mobile No.</span><span>{{ user.cp\_mobile\_no }} </span> </div>

<button class="button" @click="updatecontact()" v-if="user.id > 0">

<div class="button-content" style=" color: #000; background-color: #0aba77; font-weight: 500">Update</div>

</button> </div> </div> <br/> </div>

</section>

<section class="section" v-show="tab == 'main'">

<div class="container">

<center>

<div>

<button class="button" @click="report()" v-if="user.id > 0">

<div class="button-content" style=" color: #000; background-color: #0aba77; font-weight: 500">Report</div>

</button> </div> </center>

<div class="panel-info" style="border: 0px solid #000; background-color: #222; margin: 0;"> <div>

<div class="panel-info-item" style=" line-height:0;"><span style=" color: #0aba77"> Accidents Reported </span> </div> <hr style=" line-height:1;" color="#0aba77" /><br/>

<div class="panel-info-item"><span>No. of Reports</span><span class="ether-value">{{ safe( parseInt(user.total\_reports).toLocaleString()) }} </span> </div>

<table width="100%" >

<tr style=" background-color: #3CB371">

<th width="14%" align="center" style="text-transform: capitalize;"> Id </th>

<th width="17%" align="center" style="text-transform: capitalize;"> Reported <br/> Date</th>

<th width="17%" align="center" style="text-transform: capitalize;"> Points <br/> </th>

<th width="17%" align="center" style="text-transform: capitalize;"> Justification <br/> </th>

<th width="17%" align="center" style="text-transform: capitalize;"> Location </th>

<th width="18%" align="center" style="text-transform: capitalize;"> Vehicle <br/> </th> </tr>

<tr v-for="(report,index) in user.report\_info" :key="index" :class="'style-' + (index % 2)">

<td align="center" height="18"> <span> {{report[0]}}</span> </td>

<td align="center" height="18"> <span> {{report[1]}}</span> </td>

<td align="center" height="18"> <span> {{report[2]}}</span> </td>

<td align="center" height="18"> <span> {{report[3]}}</span> </td>

<td align="center" height="18"> <span> <a v-bind:href="'https://maps.google.com/maps?q=' + report[4]" > https://maps.google.com/maps?q={{report[4]}} </a> </span> </td>

<td align="center" height="18"> <span> {{report[5]}}</span> </td> </tr> </table> </div> </div> <br/> </div> <br/>

<div class="panel-info" style="border: 0px solid #000; background-color: #222; margin: 0;">

<div> <div class="panel-info-item" style=" line-height:0;"><span style=" color: #0aba77"> Smart Contract Info </span> </div> <hr style=" line-height:1;" color="#0aba77" /><br/>

<div class="panel-info-item"><span>Smart Contract address</span><a :href="'https://nile.tronscan.org/#/contract/' + contract\_address" target="\_blank" style=" color: #0aba77;">{{ contract\_address.substr(0, 8) }}...</a> </div>

<div class="panel-info-item"><span>Total Community participants</span><span class="ether-value" v-text="safe(contract.total\_users)">?</span>

</div> </div> </div>

</section>

<section class="section" v-show="tab == 'how'">

<div class="page">

<h3>Heading</h3>

<p></p>

<p></p>

<br><a href="index.html" @click.prevent="tab = 'main'">Back</a>

</div>

</section>

<section class="section" v-show="tab == 'usage'">

<div class="page">

<h3>Heading</h3>

<p></p>

<p></p>

<br/><a href="index.html" @click.prevent="tab = 'main'">Back</a>

</div>

</section>

<section class="section" v-show="tab == 'security'">

<div class="page">

<h3>Heading</h3>

<p></p>

<p></p>

<br/><a href="index.html" @click.prevent="tab = 'main'">Back</a>

</div>

</section>

<div class="i-modal" v-if="upmodal.show" >

<form>

<div class="i-modal\_\_wrap">

<div class="i-modal\_\_close" @click="upmodal.show = false"></div>

<div class="i-modal\_\_body">

<div class="i-modal\_\_title" style=" text-transform: capitalize;" >Signup Details</div> <hr/>

<div class="i-modal\_\_text" style=" text-transform: capitalize;" >

<span> Name </span> <br/> <input type="text" onkeydown="return alphaOnly(event);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="50" onblur v-model="upmodal.username"> <br/>

<span> Vehicle Number </span> <br/> <input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="50" v-model="upmodal.vehicleno"> <br/>

<span> Mobile Number </span> <br/> <input type="number" onblur="ValidateMobile(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="50" v-model="upmodal.mobileno"> <br/>

<span> Contact Person </span> <br/> <input type="text" onkeydown="return alphaOnly(event);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="50" v-model="upmodal.cpname" > <br/>

<span> Contact Person Mobile </span> <br/> <input type="number" onblur="ValidateMobile(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="50" v-model="upmodal.cpmobileno" > <br/> </div> <br/>

<button type="button" style=" height:24px;" @click="signup(upmodal.username, upmodal.vehicleno, upmodal.mobileno, upmodal.cpname, upmodal.cpmobileno)">Sign Up</button> <br/> <br/>

<p style="text-transform:none">Check your signup details!

<br/>You can change it before proceeding with signup.</p>

</div>

</div>

</div>

</form>

</div>

<div class="i-modal" v-if="upmodal1.show" >

<div class="i-modal\_\_wrap">

<div class="i-modal\_\_close" @click="upmodal1.show = false"></div>

<div class="i-modal\_\_body">

<div class="i-modal\_\_title" style=" text-transform: capitalize;" >Update Profile</div>

<hr/>

<div class="i-modal\_\_text" style=" text-transform: capitalize;" >

<span> Name </span> <br/> <input type="text" onkeydown="return alphaOnly(event);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="50" v-model="upmodal1.username"> <br/>

<span> Mobile Number </span> <br/> <input type="number" onblur="ValidateMobile(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="30" v-model="upmodal1.mobileno" > <br/>

<span> Date of Birth</span> <br/> <input type="date" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="40" v-model="upmodal1.dob" > <br/>

<span> Age </span> <br/> <input type="number" onblur="ValidateAge(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="20" v-model="upmodal1.age"> <br/>

<span> Gender </span> <br/>

<select v-model="upmodal1.gender">

<option>Male</option>

<option>Female</option> </select>

<!--<input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="30" v-model="upmodal1.gender" >--> <br/>

<span> Blood Group</span> <br/>

<select v-model="upmodal1.blood\_group">

<option>A RhD positive (A+)</option>

<option>A RhD negative (A-)</option>

<option>B RhD positive (B+)</option>

<option>B RhD negative (B-)</option>

<option>O RhD positive (O+)</option>

<option>O RhD negative (O-)</option>

<option>AB RhD positive (AB+)</option>

<option>AB RhD negative (AB-)</option>

</select>

<!-- <input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="30" v-model="upmodal1.blood\_group" > --> <br/>

<span> Email</span> <br/> <input type="text" onblur="ValidateEmail(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="50" v-model="upmodal1.email" > <br/> </div> <br/>

<buttonstyle="height:24px;" @click="updateprofile(upmodal1.username, upmodal1.mobileno, upmodal1.dob, upmodal1.age, upmodal1.gender, upmodal1.blood\_group, upmodal1.email)">Update</button>

<br/> <br/>

<p style="text-transform:none">Check your details.

<br/>You can change it before proceeding.</p>

</div> </div> </div> </div>

<div class="i-modal" v-if="upmodal2.show" >

<div class="i-modal\_\_wrap">

<div class="i-modal\_\_close" @click="upmodal2.show = false"></div>

<div class="i-modal\_\_body">

<div class="i-modal\_\_title" style=" text-transform: capitalize;" >Update Address</div> <hr/>

<div class="i-modal\_\_text" style=" text-transform: capitalize;" >

<span> Address1 </span> <br/> <input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="75" v-model="upmodal2.address1"> <br/>

<span> Address2 </span> <br/> <input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="75" v-model="upmodal2.address2" > <br/>

<span> City </span> <br/> <input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="50" v-model="upmodal2.city"> <br/>

<span> Pincode </span> <br/> <input type="number" onblur="return ValidatePincode(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="30" v-model="upmodal2.pincode" > <br/>

<span> State </span> <br/> <input type="text" onkeydown="return alphaOnly(event);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="50" v-model="upmodal2.state" > <br/>

<span> Country </span> <br/> <input type="text" onkeydown="return alphaOnly(event);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="50" v-model="upmodal2.country" > <br/> </div> <br/>

<buttonstyle="height:24px;" @click="updateaddress(upmodal2.address1, upmodal2.address2, upmodal2.city, upmodal2.pincode, upmodal2.state, upmodal2.country)">Update</button> <br/> <br/>

<p style="text-transform:none">Check your details.

<br/>You can change it before proceeding.</p> </div> </div> </div> </div>

<div class="i-modal" v-if="upmodal3.show" >

<div class="i-modal\_\_wrap">

<div class="i-modal\_\_close" @click="upmodal3.show = false"></div>

<div class="i-modal\_\_body">

<div class="i-modal\_\_title" style=" text-transform: capitalize;" >Update Contact</div>

<hr/>

<div class="i-modal\_\_text" style=" text-transform: capitalize;" >

<span> Contact Person Name </span> <br/> <input type="text" onkeydown="return alphaOnly(event);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="50" v-model="upmodal3.cp\_name"> <br/>

<span> Contact Relation </span> <br/> <input type="text" onkeydown="return alphaOnly(event);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="50" v-model="upmodal3.cp\_relation" > <br/>

<span> Contact Email </span> <br/> <input type="text" onblur="ValidateEmail(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="60" v-model="upmodal3.cp\_email"> <br/>

<span> Contact Personal Mobile </span> <br/> <input type="number" onblur="ValidateMobile(this);" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="30" v-model="upmodal3.cp\_mobileno" > <br/> </div> <br/>

<button style=" height:24px;" @click="updatecontact(upmodal3.cp\_name, upmodal3.cp\_relation,upmodal3.cp\_email,upmodal3.cp\_mobileno)">Update</button> <br/> <br/>

<p style="text-transform:none">Check your details.

<br/>You can change it before proceeding.</p>

</div> </div> </div> </div>

<div class="i-modal" v-if="upmodal4.show" >

<div class="i-modal\_\_wrap">

<div class="i-modal\_\_close" @click="upmodal4.show = false"></div>

<div class="i-modal\_\_body">

<div class="i-modal\_\_title" style=" text-transform: capitalize;" >Report Accident</div> <hr/>

<div class="i-modal\_\_text" style=" text-transform: capitalize;" >

<span> Justification</span> <br/> <input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="100" v-model="upmodal4.justification"> <br/>

<span> Location </span> <br/>

<input type="text" id="txtLocation" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="70" >

<input type="button" onClick="getLocation();" value="Get Current Location"/> <br/>

<span> Accident Date </span> <br/> <input type="date" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:100%;" size="40" v-model="upmodal4.accident\_date"> <br/>

<span> Vehicle Number </span> <br/> <input type="text" style="color:#000; padding:5px 10px; font:14px; border:1px solid #ccc; max-width:15%;" size="30" v-model="upmodal4.vehicle\_no" > <br/>

</div>

<br/>

<buttonstyle="height:24px;"@click="report(upmodal4.justification, $('#txtLocation').val(),upmodal4.accident\_date,upmodal4.vehicle\_no)">Update</button> <br/> <br/>

<p style="text-transform:none">Check your details.

<br/>You can change it before proceeding.</p>

</div> </div> </div> </div>

</main>

</section>

<footer>

<!--

<div class="footer-content"><a href="#" target="\_blank" class="btn" style=" color: #000; background-color: #0aba77">Official channel</a>

</div>

-->

</footer>

<script src="cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/vue@2.6.11"></script>

<script src="modules/site/main/assets/scripts/odometer.min.js"></script>

<script src="modules/site/main/assets/scripts/common7620.js?1602063002"></script>

<script type="text/javascript">

(function(m, e, t, r, i, k, a) {

m[i] = m[i] || function() {

(m[i].a = m[i].a || []).push(arguments)

};

m[i].l = 1 \* new Date();

k = e.createElement(t), a = e.getElementsByTagName(t)[0], k.async = 1, k.src = r, a.parentNode.insertBefore(k, a)

})

(window, document, "script", "mc.yandex.ru/metrika/tag.js", "ym");

ym(66417985, "init", {

clickmap: true,

trackLinks: true,

accurateTrackBounce: true

});

</script>

<!-- jquery for background animation-->

<script src="js/imagesloaded.pkgd.min.js"></script>

<script src="js/particles.min.js"></script>

<script src="js/particles-1.js"></script>

<script src="js/main.js"></script>

<noscript>

<div>

<img src="https://mc.yandex.ru/watch/66417985" style="position:absolute; left:-9999px;" alt="" /> </div> </noscript> </body> </html>

**SMART CONTRACT**

pragma solidity 0.5.10;

contract AccidentAlert {

struct User {

uint256 id;

mapping(uint256 => uint256) accidents;

mapping(uint256 => uint256) reports;

mapping(uint8 => PersonalDetails) personaldetails;

mapping(uint8 => VehicleDetails) vehicledetails;

mapping(uint8 => ContactDetails) contactdetails;

uint256 points;

uint256 total\_accidents;

uint256 total\_reports;

}

struct PersonalDetails

{

string newname;

string name;

uint40 dob;

uint8 age;

string gender;

string blood\_group;

string email;

string mobile\_no;

string address1;

string address2;

string city;

string pincode;

string state;

string country;

string profile\_pic;

}

struct VehicleDetails

{

string vehicle\_no;

string rc\_book;

}

struct ContactDetails

{

string cp\_name;

string cp\_relation;

string cp\_email;

string cp\_mobile\_no;

}

struct Accident {

uint256 id;

string justification;

string location;

string pic1;

string accident\_date;

string vehicle\_no;

uint256 points;

address reportedby;

}

uint256 public accident\_points = 10;

address payable public owner;

mapping(address => User) public users;

mapping(uint256 => Accident) public accidents;

mapping(uint256 => address) public addresses;

mapping(string => address) public vehicles;

uint256 public total\_users = 1;

uint256 public total\_accidents = 0;

event AccidentReported(address indexed reportedby, string vehicle\_no, string accident\_date, string loc);

event AccidentReportedWithContact(address indexed reportedby, string vehicle\_no, string accident\_date, string loc,

string cpname, string cpmobileno);

constructor(address payable \_owner) public {

owner = \_owner;

users[\_owner].id=total\_users;

addresses[total\_users]=\_owner;

}

function signup(string calldata \_name, string calldata \_vehicle\_no, string calldata \_mobile\_no,

string calldata \_cp\_name, string calldata \_cp\_mobile\_no) external {

require(users[msg.sender].id == 0, "Address Already Registered" );

require(vehicles[\_vehicle\_no] == address(0),"Vehicle Already Registered");

users[msg.sender].personaldetails[0].name = \_name;

users[msg.sender].vehicledetails[0].vehicle\_no = \_vehicle\_no;

users[msg.sender].personaldetails[0].mobile\_no = \_mobile\_no;

users[msg.sender].contactdetails[0].cp\_name = \_cp\_name;

users[msg.sender].contactdetails[0].cp\_mobile\_no = \_cp\_mobile\_no;

total\_users++;

users[msg.sender].id = total\_users;

vehicles[\_vehicle\_no] = msg.sender;

addresses[users[msg.sender].id] = msg.sender;

}

function signup1(string calldata \_newname) external {

require(users[msg.sender].id == 0, "Address Already Registered" );

users[msg.sender].personaldetails[0].newname = \_newname;

total\_users++;

users[msg.sender].id = total\_users;

addresses[users[msg.sender].id] = msg.sender;

}

function updateprofile(string calldata \_name, string calldata \_mobile\_no, uint40 \_dob,

uint8 \_age, string calldata \_gender, string calldata \_blood\_group, string calldata \_email, string calldata \_rc\_book

) external {

require(users[msg.sender].id > 0, "User Not Registered" );

users[msg.sender].personaldetails[0].name = \_name;

users[msg.sender].personaldetails[0].mobile\_no = \_mobile\_no;

users[msg.sender].personaldetails[0].dob = \_dob;

users[msg.sender].personaldetails[0].age = \_age;

users[msg.sender].personaldetails[0].gender = \_gender;

users[msg.sender].personaldetails[0].blood\_group = \_blood\_group;

users[msg.sender].personaldetails[0].email = \_email;

users[msg.sender].vehicledetails[0].rc\_book = \_rc\_book;

}

function updateaddress(string calldata \_address1, string calldata \_address2, string calldata \_city,

string calldata \_pincode, string calldata \_state, string calldata \_country

) external {

require(users[msg.sender].id > 0, "User Not Registered" );

users[msg.sender].personaldetails[0].address1 = \_address1;

users[msg.sender].personaldetails[0].address2 = \_address2;

users[msg.sender].personaldetails[0].city = \_city;

users[msg.sender].personaldetails[0].pincode = \_pincode;

users[msg.sender].personaldetails[0].state = \_state;

users[msg.sender].personaldetails[0].country = \_country;

}

function updatecontact(string calldata \_cp\_name,

string calldata \_cp\_relation,

string calldata \_cp\_email, string calldata \_cp\_mobile\_no

) external {

require(users[msg.sender].id > 0, "User Not Registered" );

users[msg.sender].contactdetails[0].cp\_name = \_cp\_name;

users[msg.sender].contactdetails[0].cp\_relation = \_cp\_relation;

users[msg.sender].contactdetails[0].cp\_email = \_cp\_email;

users[msg.sender].contactdetails[0].cp\_mobile\_no = \_cp\_mobile\_no;

}

function report( string calldata \_justification, string calldata \_location,

string calldata \_accident\_date, string calldata \_vehicle\_no

) external {

require(users[msg.sender].id > 0, "User Not Registered" );

//require(users[vehicles[\_vehicle\_no]].id > 0, "Vehicle Not Registered" );

total\_accidents++;

accidents[total\_accidents].id = total\_accidents;

accidents[total\_accidents].justification = \_justification;

accidents[total\_accidents].location = \_location;

accidents[total\_accidents].accident\_date = \_accident\_date;

accidents[total\_accidents].vehicle\_no = \_vehicle\_no;

accidents[total\_accidents].reportedby = msg.sender;

accidents[total\_accidents].points = accident\_points;

users[msg.sender].total\_reports++;

users[msg.sender].reports[users[msg.sender].total\_reports] = total\_accidents;

users[msg.sender].points += accident\_points;

if (users[vehicles[\_vehicle\_no]].id > 0)

{

users[vehicles[\_vehicle\_no]].total\_accidents++;

users[vehicles[\_vehicle\_no]].accidents[users[vehicles[\_vehicle\_no]].total\_accidents] = total\_accidents;

emit AccidentReportedWithContact(msg.sender, \_vehicle\_no, \_accident\_date, \_location,

users[vehicles[\_vehicle\_no]].contactdetails[0].cp\_name,

users[vehicles[\_vehicle\_no]].contactdetails[0].cp\_mobile\_no );

}

else

{

emit AccidentReported(msg.sender, \_vehicle\_no, \_accident\_date, \_location);

}

}

function stringToBytes32(string memory source) public pure returns (bytes32 result) {

bytes memory tempEmptyStringTest = bytes(source);

if (tempEmptyStringTest.length == 0) {

return 0x0;

}

assembly {

result := mload(add(source, 32))

}

}

/\*

Only external call

\*/

function userInfo(address \_addr) view external returns(

uint256 id, uint40 dob, uint8 age,

bytes32[] memory user\_details

)

{

bytes32[] memory \_user\_details = new bytes32[](18);

\_user\_details[0] = stringToBytes32(users[\_addr].personaldetails[0].name);

\_user\_details[1] = stringToBytes32(users[\_addr].personaldetails[0].gender);

\_user\_details[2] = stringToBytes32(users[\_addr].personaldetails[0].blood\_group);

\_user\_details[3] = stringToBytes32(users[\_addr].personaldetails[0].email);

\_user\_details[4] = stringToBytes32(users[\_addr].personaldetails[0].mobile\_no);

\_user\_details[5] = stringToBytes32(users[\_addr].personaldetails[0].address1);

\_user\_details[6] = stringToBytes32(users[\_addr].personaldetails[0].address2);

\_user\_details[7] = stringToBytes32(users[\_addr].personaldetails[0].city);

\_user\_details[8] = stringToBytes32(users[\_addr].personaldetails[0].pincode);

\_user\_details[9] = stringToBytes32(users[\_addr].personaldetails[0].state);

\_user\_details[10] = stringToBytes32(users[\_addr].personaldetails[0].country);

\_user\_details[11] = stringToBytes32(users[\_addr].personaldetails[0].profile\_pic);

\_user\_details[12] = stringToBytes32(users[\_addr].vehicledetails[0].vehicle\_no);

\_user\_details[13] = stringToBytes32(users[\_addr].vehicledetails[0].rc\_book);

\_user\_details[14] = stringToBytes32(users[\_addr].contactdetails[0].cp\_name);

\_user\_details[15] = stringToBytes32(users[\_addr].contactdetails[0].cp\_relation);

\_user\_details[16] = stringToBytes32(users[\_addr].contactdetails[0].cp\_email);

\_user\_details[17] = stringToBytes32(users[\_addr].contactdetails[0].cp\_mobile\_no);

return (

users[\_addr].id, users[\_addr].personaldetails[0].dob, users[\_addr].personaldetails[0].age,

\_user\_details

);

}

function userTotalInfo(address \_addr) view external returns( uint256 id, string memory name, uint256 points, uint256 total\_accident, uint256 total\_report) {

return (users[\_addr].id, users[\_addr].personaldetails[0].name, users[\_addr].points, users[\_addr].total\_accidents, users[\_addr].total\_reports);

}

function contractInfo() view external returns(uint256 \_total\_users, uint256 \_total\_reports) {

return (total\_users, total\_accidents);

}

function userReportInfo(address \_addr) view external returns(

uint256[] memory id,

uint256[] memory points,

bytes32[5][] memory others

)

{

uint256[] memory \_id = new uint256[](users[\_addr].total\_reports+1);

uint256[] memory \_points = new uint256[](users[\_addr].total\_reports+1);

bytes32[5][] memory \_others = new bytes32[5][](users[\_addr].total\_reports+1);

for (uint8 i = 1; i <= users[\_addr].total\_reports; i++) {

\_id[i] = accidents[users[\_addr].reports[i]].id;

\_points[i] =accidents[users[\_addr].reports[i]].points;

\_others[i][0] = stringToBytes32(accidents[users[\_addr].reports[i]].justification);

\_others[i][1] = stringToBytes32(accidents[users[\_addr].reports[i]].location);

\_others[i][2] = stringToBytes32(accidents[users[\_addr].reports[i]].pic1);

\_others[i][3] = stringToBytes32(accidents[users[\_addr].reports[i]].vehicle\_no);

\_others[i][4] = stringToBytes32(accidents[users[\_addr].reports[i]].accident\_date);

}

return ( \_id, \_points, \_others );

}

}

**REFERENCE**

1] L. Chen, S.-L. Ng, and G. Wang, "Threshold anonymous announcement in VANETs," IEEE J. Sel. Areas Commun., vol. 29, no. 3, pp. 605-615, Feb. 2011.

[2] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for VANETs," IEEE Trans. Veh. Technol., vol. 65, no. 3, pp. 1711-1720, Feb. 2016.

[3] S. Nakamoto. (2008, Bitcoin: A peer-to-peer electronic cash system. Available: <https://bitcoin.org/bitcoin.pdf>

[4] H. Hartenstein and L. Laberteaux, "A tutorial survey on vehicular ad hoc networks," IEEE Commun. Mag., vol. 46, no. 6, June 2008.

[5] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in Proc. Workshop on hot topics in networks (HotNetsIV), Maryland, USA, Nov. 2005, pp. 1-6.

[6] F. Dötzer, "Privacy issues in vehicular ad hoc networks," in Proc. Int. Workshop on Privacy Enhancing Technologies, May 2005, pp. 197-209.

[7] J. R. Douceur, "The sybil attack," in Proc. Int. Workshop on Peerto-Peer Syst. 2002, pp. 251-260.

[8] E. Bresson, J. Stern, and M. Szydlo, "Threshold ring signatures and applications to ad-hoc groups," in Annu. Int. Cryptology Conference, Aug. 2002, pp. 465-480.

[9] J. Ren and L. Harn, "An efficient threshold anonymous authentication scheme for privacy-preserving communications," IEEE Trans. Wireless Commun., vol. 12, no. 3, pp. 1018-1025, Jan. 2013.

[10] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient secure aggregation in VANETs," in Proc. of the 3rd Int. workshop on Veh. ad hoc networks, Sep. 2006, pp. 67-75.