CHAINSAFE - INSURANCE MANAGEMENT SYSTEM USING BLOCKCHAIN

PROJECT REPORT

submitted by

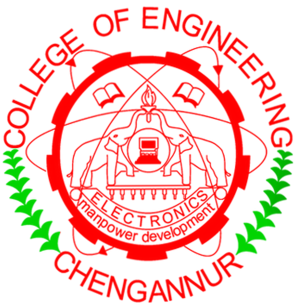
AROMAL S

CHN19CS027

to

APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of B.Tech Degree in Computer Science & Engineering



DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ENGINEERING CHENGANNUR, ALAPPUZHA

JUNE 2023

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ENGINEERING CHENGANNUR

ALAPPUZHA

CERTIFICATE

This is to certify that, the project report titled ***CHAINSAFE - INSURANCE MANAGEMENT SYSTEM USING BLOCKCHAIN*** *is a bonafide record of the* ***CSD416 PROJECT*** *presented by* ***AROMAL S*** *(CHN19CS027) Eighth Semester B. Tech. Computer Science and Engineering student, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree,* ***B. Tech. Computer Science and Engineering*** *of* ***APJ Abdul Kalam Technological University****.*

Smt. Sreelekshmi K R	**Prof. Vinod P R**	**Dr. Manju S Nair**
Guide	Project Coordinator	Head of the Dept.
Assistant Professor	Assistant Professor	Associate Professor
Dept.of CS	Dept.of CS	Dept.of CS

DECLARATION

I undersigned hereby declare that the project report "**Chainsafe - Insurance Management System Using Blockchain**" , submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of **Smt. Sreelekshmi K R**. This submission represents our ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Chengannur **Aromal S**

Date : 19/06/2023 **CHN19CS027**

ACKNOWLEDGEMENT

This work would not have been possible without the support of many people. First and foremost, I give thanks to Almighty God who gave us the inner strength, resources, and ability to complete our project successfully.

I would like to thank **Dr. Smitha Dharan**, The Principal, has provided with the best facilities and atmosphere for the project completion and presentation. I would also like to thank HOD **Dr. Manju S Nair** (Associate Professor, Computer Science and Engineering), our project coordinator **Prof. Vinod P R** (Assistant Professor, Computer Science and Engineering), our project guide **Smt. Sreelekshmi K R** (Assistant Professor, Computer Science and Engineering) for the extended help and the encouragement and support given to me while doing the project.

I would like to thank our dear friends and faculties for extending their cooperation and encouragement throughout the project work, without which I would never have completed the project this well. Thank you all for your love and also for being very understanding.

AROMAL S

CHN19CS027

ABSTRACT

The insurance industry acknowledges the significance of accurate and secure record-keeping and has now embraced the potential of blockchain technology. Through the utilization of Ethereum, a well-known blockchain platform, we can develop a decentralized insurance management system that guarantees the authenticity and traceability of insurance-related data. Our solution aims to securely store and manage policy information, claims data, and relevant documentation for insurers, using smart contracts written in Solidity, a programming language designed for Ethereum. By automating processes like claims processing and policy verification, our system creates a tamper-evident record of all transactions, capitalizing on the transparent and immutable nature of blockchain technology. Through these functionalities, our solution strives to enhance efficiency, minimize the risk of fraud, and foster increased customer trust in insurers.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Area

The field of insurance is a huge and complicated sector that plays an essential part in the process of shielding individuals and organisations from the adverse effects of financial uncertainty. The insurance sector is a highly regulated sector, with stringent laws and guidelines in place to protect the market's fairness and integrity. This is the case since the insurance industry is so important to consumers. Insurers are required to abide by regulations and underwriting criteria that are particularly stringent in order to protect their policyholders. Actuaries are another important part of the industry. These professionals utilise statistical analysis to evaluate and manage the risks that are involved with the many kinds of insurance contracts. The insurance sector may experience a sea change as a result of the advent of blockchain technology, which has the potential to improve transparency, efficiency, and safety. Purchasing, administering, and making insurance claims can all be made more efficient with the help of an insurance management system that is built on blockchain technology.

A blockchain system is able to produce a record of all insurance transactions that is both tamper-proof and transparent since it utilises a decentralised network and employs secure record-keeping. By making it simple and fast to obtain policy details, this has the potential to both cut down on fraudulent activity and enhance the overall customer experience. Additionally, the utilisation of smart contracts can automate a number of manually performed operations, hence eliminating the requirement for intermediaries and resulting in cost savings for both insurers and policyholders. A blockchain-based insurance management system is a digital platform that streamlines and automates a variety of operations that are used in the insurance industry. Blockchain technology is the underlying technology behind the system. The utilisation of digital contracts, also known as smart contracts, is a crucial component of such a system. Contracts that are self-executing are referred to as smart contracts. These kind of contracts have the terms of the agreement between the buyer and seller encoded straight into lines of code. A blockchain-based insurance management system is a digital platform that streamlines and automates a variety of operations that are used in the insurance

sector. The underlying technology for this system is Blockchain. Using an insurance management system that is based on blockchain technology could have a number of potential benefits, including improved security. Blockchain technology is secure and tamper-evident, which can help protect users from fraudulent activity and errors. There are many different areas within the insurance industry, such as policy issurance, that could benefit from the implementation of a management system that is based on blockchain technology.

The process of providing insurance policies, which includes validating the information provided by the policyholder and calculating premiums, can be automated with the help of smart contracts, which can be used to automate the process. A system that is based on blockchain technology has the potential to simplify the process of filing claims by employing smart contracts to automate the validation and payment of claims. The data that is recorded on the blockchain can be used by insurers to more properly assess risks and price those risks. The immutability and transparency of the blockchain make it a useful tool for detecting and preventing fraudulent behaviour in the insurance industry. A system that is based on blockchain technology has the potential to make the process of reinsurance more effective while also easing the transfer of risks between insurers. The installation of an insurance management system that is based on blockchain technology has the potential to provide major benefits to the insurance sector as a whole.

1.2 Objectives

- Demonstrate the potential of blockchain technology in streamlining the claims process in the insurance industry.
- To automate manual processes and reduce errors through the use of smart contracts.
- Improve operating efficiencies and lower transaction costs through the use of blockchain technology.
- To increase trust among parties by providing a secure and transparent record of all transactions.
- Improve the customer experience by providing quick and easy access to policy information and streamlining the claims process.

CHAPTER 2

LITERATURE SURVEY

2.1 Implementation of Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Insurance Services by V. Aleksieva, H. Valchanov and A. Huliyan

Presents the potential of using smart contracts and blockchain technology in the insurance industry. Smart contracts are self-executing programming code that runs on top of a blockchain network and are used to manage complex business logic. They are often used in the insurance industry to streamline the claims process and reduce operational costs by automating tasks and eliminating the need for extrinsic enforcement of legal agreements.

Hyperledger Fabric [1] is a private blockchain platform that is well-suited for insurance applications due to its trusted nodes, quick access to information, cheap transactions, and ability to control privacy levels. The authors in this worrk have implemented smart contracts for insurance services on the Hyperledger Fabric platform and present their experimental results in the work.

Smart contracts based on Hyperledger Fabric blockchain technology can be implemented in the insurance industry to improve the efficiency and transparency of insurance services. Hyperledger Fabric is an open-source blockchain platform that is designed specifically for enterprise use cases. One way in which smart contracts based on Hyperledger Fabric could be used in the insurance industry is for policy issuance. Smart contracts can be used to automate the process of verifying the information provided by the policyholder and calculating the premium. This can help reduce the time and resources needed to issue policies and make it easier for customers to purchase insurance.

Another potential use case for Hyperledger Fabric-based smart contracts in the insurance industry is claims processing. Customers can file claims through the platform and provide any necessary supporting documentation. The smart contract can then automatically validate the claim and release the payout to the policyholder if it is approved. This can help streamline the claims process and re-

duce the time and resources needed to handle claims. Overall, the implementation of smart contracts based on Hyperledger Fabric blockchain technology in the insurance industry has the potential to improve the efficiency and transparency of insurance services. It can also help reduce operational costs and increase customer satisfaction.

2.2 Smart Contracts based on Private and Public Blockchains for the Purpose of Insurance Services by by V. Aleksieva, H. Valchanov and A. Huliyan

Presents about the potential of private and public blockchains to improve the efficiency and transparency of the insurance process. Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. They are used in various industries, including insurance, to automate processes, reduce the risk of fraud, and improve efficiency. In the insurance industry, smart contracts can be used to automate the process of calculating premiums and processing claims.

There are several blockchain platforms that can be used to implement smart contracts, including Ethereum, Hyperledger Fabric, Stellar, Waves, and NXT. Each platform has its own specific features for developing smart contracts.

Private blockchain networks, which are restricted to a group of trusted nodes, are typically used in the insurance industry due to their ability to provide quick access to information, cheaper transactions, and privacy control. Public blockchain networks, on the other hand, are open to anyone and use consensus protocols to validate transactions. While they offer increased transparency, they can be slower and more expensive to use.

In this worrk[2], the authors present two experimental implementations of smart insurance contracts, one based on the public blockchain Ethereum and the other based on the private blockchain Hyperledger Fabric. The results of the experiments are presented and compared to evaluate the suitability of each platform for use in the insurance industry.

On a private blockchain, smart contracts can be used by an insurance company to automate the process of issuing policies and handling claims. For example, when a customer purchases a policy,

a smart contract can be used to verify their information, calculate the premium, and issue the policy. If the policyholder needs to file a claim, they can do so through the platform and provide any necessary supporting documentation. The smart contract can then automatically validate the claim and release the payout to the policyholder if it is approved.

2.3 A Blockchain-Based Decentralized Insurance Platform by S. Alwis and T. M. K. K. Jinasena

This work [3] explores the application of blockchain technology to the development of a decentralised insurance platform with the aim of addressing issues of trust and transparency that are prevalent in the insurance industry. Specifically, the work focuses on how blockchain technology could be used to create a platform that would allow insurance companies to interact directly with policyholders. Because of this, the platform is in a position to be able to automate a variety of insurance processes, such as the processing of claims and the verification of policies. In addition to this, it creates an immutable record of all the transactions that can never be changed in any way.

The authors are also aware of some of the constraints that are connected to the platform that they have proposed in their work. One of the potential disadvantages is the fact that there needs to be a substantial amount of trust between the various parties that are involved in the platform. This is one of the potential drawbacks. The dependability of those who use the platform, in addition to the veracity of the statements made by those who use the platform, is essential to the successful operation of the platform.

One key feature of a decentralized insurance platform is the use of smart contracts. Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. They can be used to automate tasks such as policy issuance, claims processing, and risk assessment.On a decentralized insurance platform, customers can purchase insurance policies directly from insurers or reinsurers without the need for intermediaries. This can help reduce costs and make the process more efficient. Customers can also file claims and receive payouts directly through the platform, using smart contracts to automate the claims process. Overall, a blockchain-based decentralized insurance platform has the potential to transform the way that insurance is bought, sold, and managed by enabling more efficient and cost-effective transactions, improving security and transparency, and increasing accessibility.

2.4 Review of Existing Blockchain-Based Insurance Solutions by A. Averin, E. Musaev and P. Rukhlov

Analyses a number of existing blockchain-based insurance solutions and evaluate their strengths and weaknesses. These solutions include Medishares, Etherisc, PAL Network, Teambrella, VouchForMe, D3i, Insurwave, Dynamis, and Consortium W3c.

The authors discuss the potential benefits of using blockchain technology in the insurance industry [4], such as the ability to automate routine operations and improve production processes, as well as the potential for decentralization and increased transparency for customers. They also examine the potential drawbacks of these solutions, such as the need for a high level of trust among participants and the potential costs and technical challenges of implementation. Overall, the authors conclude that blockchain technology has the potential to significantly modernize and improve the efficiency of the insurance industry.

A review of existing blockchain-based insurance solutions reveals that the technology is being actively explored and implemented in various areas of the insurance industry. For example, some insurers are using blockchain to automate the claims process, using smart contracts to automatically validate and pay out claims. Other insurers are using blockchain to issue policies, using smart contracts to verify the information provided by policyholders and calculate premiums. In addition, some insurers are using blockchain to enhance the security and transparency of their operations, by providing a secure and tamper-evident record of all transactions and data.

Overall, it is clear that the insurance industry is actively exploring the use of blockchain technology and is finding various ways to incorporate it into their operations. It is likely that we will see even more widespread adoption of blockchain-based insurance solutions in the future as the technology continues to mature and become more widely accepted.

2.5 ClaimChain: Secure Blockchain Platform for Handling Insurance Claims Processing by Naga Ramya Bhamidipati, Khaza Anuarul Hoque, and Prasad Calyam

Proposes a blockchain-based platform called ClaimChain[5] for improving the efficiency and scale of insurance claims processing. ClaimChain aims to automate the claims process using a consortium blockchain and to secure the platform against fraud and data integrity attacks through the use of attack tree formalism and machine learning models.

The authors evaluate the scalability of ClaimChain by simulating large numbers of transactions and show that data integrity attacks can be mitigated through the implementation of security design principles. They also demonstrate the effectiveness of ClaimChain's fraud detection capabilities by testing machine learning models on an open dataset. Overall, the authors argue that ClaimChain has the potential to significantly improve the efficiency and security of insurance claims processing.

This platform utilizes smart contracts to automate the claims process, reducing the need for manual handling and the risk of errors or fraud. Customers can file claims on the platform and provide supporting documentation, which is then automatically validated by the smart contracts. If the claim is approved, the customer is paid out according to the terms of their policy. The transparent and immutable nature of the blockchain ensures that all parties have access to the same, up-to-date information and helps to reduce disputes and increase trust. Overall, ClaimChain provides a secure, efficient, and transparent platform for handling insurance claims processing.

2.6 Data Exchange Platform to Fight Insurance Fraud on Blockchain by I. Nath

This work explores the possible applications of blockchain technology in the insurance business to enhance the speed and accuracy of insurance claim processing and settlements. The authors explain how blockchain technology can be utilised to build a decentralised and safe platform for the storage and management of data connected to insurance, such as information on policies, data on claims, and other documentation that is pertinent to the industry. It is possible for insurers to lower the risk of fraud by automating various procedures[6], like as the processing of claims and the verification of policies, with the assistance of smart contracts and the nature of blockchain technology.

The authors also examine the possibilities for utilising machine learning models and risk scoring to detect fraudulent activity and enhance the accuracy of insurance claims processing. This topic is covered in both books. Overall, the research suggests that blockchain technology may have the ability to greatly modernise the insurance business by offering a platform that is both secure and transparent for the management of data linked to insurance.

2.7 PRIDE: A Private and Decentralized Usage-Based Insurance Using Blockchain by Z. Wan, Z. Guan and X. Cheng

Proposes Usage-based insurance (UBI) [7], a type of car insurance where premiums are calculated based on the actual usage and driving patterns of the insured vehicle. It can potentially reduce insurance costs for safe drivers, but it requires detailed driving data to determine premiums, which can be a privacy concern for drivers. The traditional approach to UBI requires a centralized insurance company as an intermediary to manage the insurance, which can be costly and time-consuming.

In this work, the authors propose PRIDE, a privacy-preserving and decentralized UBI scheme using blockchain technology to record encrypted driving data and smart contracts to calculate insurance premiums. PRIDE is designed to provide security and privacy without relying on any centralized party or tamper-proof hardware. The authors analyze the security of PRIDE and evaluate its performance, finding that it is efficient in processing UBI insurance requests with a processing time of around 898ms per request.

In a traditional insurance model, policyholders pay a fixed premium regardless of how much or how little they use their insurance coverage. With UBI, policyholders only pay for the coverage they actually use. This is made possible by using sensors and other technologies to track and record the usage of a particular asset or service. The data is then recorded on a blockchain, which allows for an accurate and immutable record of the usage.

One key benefit of private and decentralized UBI is that it can provide a more customized and cost-effective insurance experience for policyholders. Rather than paying for coverage they may not use, policyholders only pay for the coverage they actually need. This can also help insurers more accurately assess and price risks, leading to more competitive premiums. Additionally, because the data is recorded on a blockchain, it is secure and transparent.

2.8 A Preliminary Study of the Impact of Blockchain Technology on the Application Level of Insurance Industry by T. -W. Yu, A. -P. Wang, L. -M. Tseng and W. -M. Tsao [8]

This work discusses the potential for the distributed ledger technology known as blockchain [8] to have a substantial impact on the insurance sector by improving operational efficiencies and cutting costs via the implementation of smart contracts and other forms of automation. It can also increase data integrity and security by providing a decentralised and secure database for storing and sharing information. This database can be used for both storing and sharing information.

In addition, the technology behind blockchain can be used to increase individuals' privacy by enabling them to disclose only the data that is absolutely essential with their insurers while still retaining control over their own personal information. This is made possible by the distributed ledger. However, the use of blockchain technology in the insurance industry still faces obstacles, such as regulatory issues and the requirement for uniformity. These obstacles must be overcome. Additional study is required to fully understand the possibilities of blockchain technology in the insurance business as well as to design applications that are both effective and secure.

An investigation into the effects that blockchain technology is having on the insurance sector is most likely going to centre on the ways in which this technology is being implemented to enhance a variety of procedures at the application level. The process of policy issuance is one area where blockchain technology may have a substantial impact. Insurers may be able to save time and costs by using smart contracts to automate the process of verifying information provided by policyholders and calculating premiums. This might be accomplished through the use of smart contracts. The handling of insurance claims is an additional domain where blockchain technology might have an effect. Insurers may be able to streamline the claims process and minimise the amount of time it takes to process claims by automating the validation and payment of claims using smart contracts.

In addition, the technology behind blockchain has the potential to be utilised to enhance risk assessment by giving insurers access to data that is both more accurate and up to date. This might make it easier for them to precisely identify and price risks, which might result in premiums that are more competitively priced for policyholders. An investigation of the effects of blockchain tech-

nology on the insurance sector is expected to conclude, on the whole, that this technology has the ability to considerably improve a variety of processes at the application level, which would result in increased efficacy, transparency, and trust.

2.9 Design of Blockchain Application Framework for Claims Platform of Flight Delay Insurance by W. Zhou and Q. Wei

In this work the overall architecture, technical architecture and data architecture of the flight delay insurance claim settlement platform based on blockchain are proposed. Customers purchase flight delay insurance from an insurance company[9].The insurance company records the policy on a blockchain.If a flight is delayed, the customer can file a claim on the platform.he customer provides proof of the flight delay (e.g. a copy of their flight itinerary and a statement from the airline confirming the delay).The claim is validated by the insurance company using smart contracts on the blockchain.If the claim is approved, the customer is paid out according to the terms of their policy.

Using a blockchain to design this platform has several benefits such as It provides a secure and transparent record of all policies and claims.It automates the claims process using smart contracts, reducing the need for manual processing and the risk of errors or fraud.It enables customers to file claims and receive payouts quickly and efficiently.

Overall, a blockchain-based platform for flight delay insurance can improve the customer experience and increase the efficiency and integrity of the claims process.One of the limitation of this model is personal and sensitive insurance data must be kept private, and it can be difficult to ensure the privacy of this data on a public blockchain.

2.10 Regulating Blockchain: Techno-Social and Legal Challenges by Philipp Hacker and Ioannis Lianos

The insurance industry's interest in blockchain is recording insurance entitlements transparently. Smart contracts could decide whether insurance claims should be paid without judicial intervention. Because blockchain and smart contracts are decentralised, they're hard to regulate. Insurance companies must follow complex legal rules, and litigation is costly. Government regulation of insurance purchased with a cryptocurrency smart contract would be difficult.

A government should target individuals or businesses buying blockchain-based insurance, not in-

surers. Using a cryptocurrency as a country's national currency could provide enough stability for smart contracts. A pool of contracts could name a loss-determining arbitrator. The arbitrator's fee may be proportional to the claim. Only high-risk insureds may join a blockchain-based insurance pool.

Lower-risk parties either forgo insurance or choose traditional insurance, where premiums are based on risk. A decentralised risk-assessment system is needed. Third-parties may certify insureds for a fee. Investors would consider the reputation of these third parties when covering an insured's risk. Some forms of insurance may be beyond regulators' reach, but existing carriers may benefit from stricter regulation.[10]

Participants are rewarded for announcing a resolution closer to the later valuation than the prior valuation, while those who push it the other way lose currency. Hybrid insurance approaches could use smart contracts' distributed nature. Smart contract sponsors may seek out smart contract holders from many locations to reduce the risk of total claims exceeding total premiums.

CHAPTER 3

PROBLEM DEFINITION

3.1 Existing System

In the traditional insurance industry, insurance management is typically handled through a centralized approach, with a single company or organization serving as the central authority. To improve the efficiency and security of their operations, many insurance companies have implemented technology solutions such as electronic claims processing and automated underwriting.

One such solution is the use of electronic claims processing, which allows policyholders to submit claims and supporting documentation electronically. This can speed up the process and reduce the risk of errors, as the information is automatically verified and processed. In addition, automated underwriting systems can analyze and assess the risk of a policy based on predetermined criteria, helping to reduce the time and cost of underwriting.

Another technology solution that is commonly used in the insurance industry is the use of data analytics. By analyzing large amounts of data, insurance companies can identify trends and patterns that can help them better understand their customers and the risks they face. This can help insurers to better target their products and services, and to more accurately assess the risk of a policy.

While these technology solutions have brought significant benefits to the insurance
industry, they have not completely eliminated the challenges faced by insurers.The insurance industry is often slow and manual, with many processes being completed manually or through outdated systems. This can lead to long turnaround times and high operational costs. Another problem is lack of transparency. There is often a lack of transparency in the insurance industry, which can lead to disputes between insurers and policyholders and reduce trust. Fraud is also a significant problem in the insurance industry, with false or exaggerated claims costing insurers billions of dollars each year. The current insurance management system also has high overhead costs due to the manual nature of many processes and the need to maintain large staffs to handle these tasks. Additionally, insurance products can be difficult for some people to access, particularly in developing countries or

underserved communities. A blockchain-based insurance management system could help address these problems by improving efficiency, transparency, and accessibility while helping to reduce fraud and lower overhead costs.

Centralized insurance management systems rely on a central authority to manage and oversee all processes. While this can be effective in some cases, it can also lead to issues of trust and transparency. Because the central authority has complete control over the system, there is a risk of bias or corruption. There may also be concerns about the security of the system, as it is vulnerable to hacking or other types of cyber attacks. Additionally, centralized systems can be inflexible and slow to adapt to changing market conditions or customer needs. In contrast, a blockchain-based insurance management system is decentralized and relies on a distributed network of computers to validate transactions. This can increase trust and transparency because all parties have access to the same, up-to-date information on the blockchain. It can also enhance security because the blockchain is secure and tamper-evident. Overall, a blockchain-based insurance management system has the potential to address many of the trust and transparency issues that are present in centralized systems.

3.2 Limitations

- Centralized insurance management technology solutions rely on a central authority, which can lead to issues of trust and transparency
- These systems can be costly to implement and maintain, which can be a burden for smaller insurers
- There is a potential concern for data privacy and security with large amounts of sensitive personal and financial information being stored and processed.
- Interoperability between different systems can be a challenge, leading to inefficiencies and delays
- Inefficiency: The insurance industry is often slow and manual, with many processes being completed manually or through outdated systems. This can lead to long turnaround times and high operational costs.
- Fraud: Fraud is a significant problem in the insurance industry, with false or exaggerated claims costing insurers billions of dollars each year.
- High overhead costs: The insurance industry has high overhead costs due to the manual nature of many processes and the need to maintain large staffs to handle these tasks.

- Inaccessibility: Insurance products can be difficult for some people to access, particularly in developing countries or underserved communities.

3.3 Problem Statement

To develop a decentralized insurance management system using blockchain technology, addressing trust, revolutionizing the industry and promoting security and reliability.

3.4 Proposed Model

The proposed insurance app using Ethereum and Solidity will be designed to improve the efficiency, transparency, and security of the insurance process. The app will use blockchain technology to create a decentralized, immutable ledger of all insurance transactions.

To develop a decentralized insurance management system using blockchain technology, the app will enable shops to create and manage policies, store policy information on the ledger, and utilize smart contracts for automated actions. Policyholders will have an intuitive interface to view and manage their policies, make premium payments, and initiate claims. The system's decentralized nature ensures secure and tamper-proof data, reducing fraud risk. The smart contract can automatically trigger the disbursement of the claim amount based on the predefined conditions, ensuring faster and smoother claims settlement for policyholders. Smart contracts automate processes, such as payment reminders and claim disbursements, increasing efficiency and minimizing errors. Claims processing becomes more efficient with quick verification and validation through the blockchain, leading to faster settlements. Overall, this solution revolutionizes insurance by enhancing trust, improving efficiency, and providing greater security in policy creation, management, and processing.

In addition to policy management, the app will also enable clients to easily access and manage their insurance policies. The app will also provide real-time updates on the status of claims, allowing clients to track the progress of their claims in real-time. To ensure the security and integrity of the platform, the app will utilize Ethereum's built-in security features, such as identity and access management, as well as various network-level security measures. Overall, the app will provide a more secure and transparent experience for both insurance companies and clients, thanks to the immutable nature of the Ethereum blockchain.

A decentralised insurance management system built on blockchain technology might offer a number of viable solutions to the challenges that the insurance sector is now facing. Enhanced productivity is one possible option. A system that is built on blockchain technology can cut down on the amount of time and resources needed to complete tasks like as the issuance of policies and the processing of claims by automating manual operations and making use of smart contracts. Because of this, turnaround times and operational expenses can end up being reduced as a result. Increased openness is another another potential option. A system that is built on blockchain technology can provide a safe and transparent record of all policies and claims, which can help minimise the number of disputes that occur between insurers and policyholders and improve the level of confidence that exists between the two parties.

A system that is based on blockchain technology can also assist decrease fraud since it provides a record of all transactions that is impossible to alter. In addition, a system based on blockchain technology has the potential to reduce overhead expenses by automating a large number of processes that are currently carried out manually. In conclusion, a system that is based on blockchain technology has the potential to improve accessibility by enabling clients to access insurance goods and services from any location in the world that has access to the internet. In general, an insurance management system that is built on blockchain technology has the potential to bring solutions to a significant number of the issues that are currently confronting the insurance business.

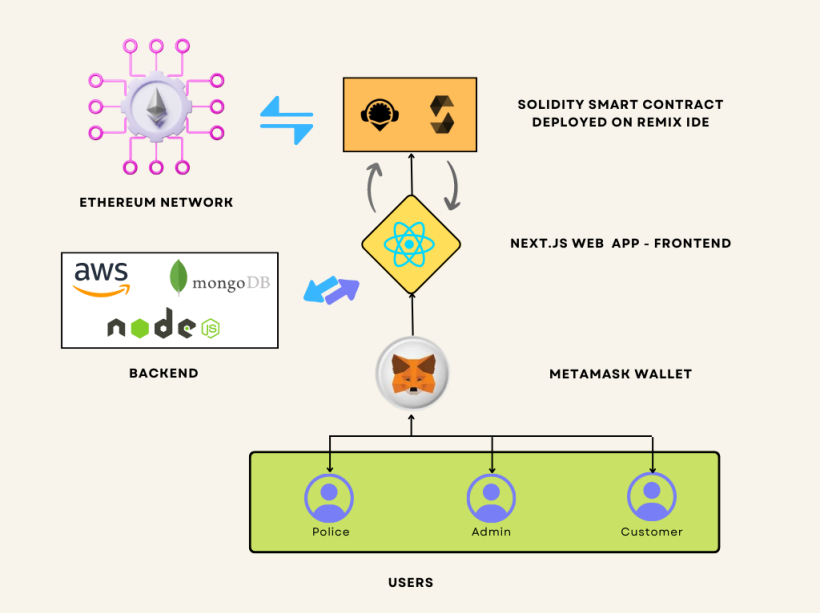


Figure 3.1: Architecture of the proposed system

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 Hardware Requirements

The minimum hardware configuration required for the proper functioning of the system can be outlined below:

- **CPU**: Intel Core i3 or above
- **Operation Speed**: 2.0 GHz or above
- **Ram**: 4GB or above

4.2 Software Requirements

The minimum software configuration required for the proper functioning of the system can be outlined below:

- **Operating System (OS)**: Windows 8 or above, or the latest version of any Linux distros.
- **Development Languages**: JavaScript for frontend development and Solidity for smart contract development.
- **Integrated Development Environment (IDE)**: Visual Studio Code (with relevant options for JavaScript and Solidity), and Remix IDE (a web-based IDE specifically designed for Solidity development).
- **Web Browsers**:
  - **Google Chrome**: The latest version of Google Chrome browser is recommended for optimal compatibility.
  - **Mozilla Firefox**: The latest version of Mozilla Firefox browser can also be used.
  - **Apple Safari**: The latest version of Safari browser for macOS users.
- **Metamask Extension**: Metamask is a popular Ethereum wallet and browser extension that allows users to interact with Ethereum-based applications. It should be installed and configured in the chosen web browser.

4.3 Languages

4.3.1 Javascript

JavaScript is a programming language that is commonly used in web development to add interactivity to websites and web-based applications. It is a client-side language, meaning that it is executed by the user's web browser rather than on the web server.

JavaScript has become an integral part of the modern web and is often used in conjunction with HTML and CSS to build dynamic and responsive web pages. It is also used to create interactive elements on web pages such as forms, menus, and games.

In a blockchain-based insurance project using Hyperledger Fabric, JavaScript can be used in several ways. For example, it can be used to build the user interface of the insurance application, allowing policyholders to easily interact with the system and perform tasks such as purchasing insurance policies or making claims.

JavaScript can also be used to implement business logic in the insurance application, such as calculating premiums or processing claims. This can be done by writing JavaScript functions that are triggered by certain events, such as the submission of a claim form or the expiration of an insurance policy.

Additionally, JavaScript can be used to integrate the insurance application with other systems and services, such as payment gateways or external databases. This can be done through the use of APIs (Application Programming Interfaces) and libraries that allow JS to communicate with these systems.

4.3.2 Solidity

Solidity is a high-level programming language used to write smart contracts on blockchain platforms, particularly on the Ethereum network. It is an object-oriented language with a syntax similar to JavaScript and C++, making it accessible to developers with programming experience in those languages. Solidity supports a range of data types, including integers, strings, booleans, and arrays, as well as complex types like structs and mappings.

Smart contracts written in Solidity are executed on the Ethereum Virtual Machine (EVM), which is a decentralized, tamper-proof virtual machine that executes the code exactly as it is written. Solidity is designed to be secure, reliable, and efficient, and it includes a range of features to support this, such as error handling, contract-level access control, and gas. One of the key benefits of Solidity is its ability to facilitate the creation of decentralized applications (dApps). Smart contracts written in Solidity can be used to build a wide range of dApps, including decentralized finance (DeFi) platforms, digital identity solutions, and supply chain management systems. Solidity also supports a number of development tools and frameworks, such as Truffle and Remix, that make it easier to write and test smart contracts.

Solidity is used to write smart contracts, which are programs that run on the blockchain and can execute transactions automatically. These contracts are immutable, meaning that they cannot be changed once they are deployed to the blockchain. Solidity includes features such as inheritance, libraries, and interfaces, which make it easy to write complex smart contracts.

Overall, Solidity is an important tool for developers looking to build decentralized apps on the Ethereum network. While it has a relatively steep learning curve, it offers significant advantages in terms of security, reliability, and efficiency, making it a popular choice for smart contract development.

4.4 Frameworks

4.4.1 Ethereum

Ethereum is a decentralized blockchain platform that enables the creation and execution of smart contracts and decentralized applications (dApps). It was launched in 2015 by Vitalik Buterin and is the second-largest cryptocurrency by market capitalization, after Bitcoin.

The Ethereum blockchain operates on a decentralized, peer-to-peer network, which means that it is not controlled by any central authority or government. This network is maintained by nodes, which are computers that run Ethereum software and participate in verifying transactions and maintaining the blockchain.

Ethereum uses its own cryptocurrency, Ether (ETH), as its native currency. Ether is used to pay for transaction fees on the Ethereum network, as well as to incentivize nodes to participate in the net-

work by providing computational power to process transactions and execute smart contracts.Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into code. These contracts can be used for a variety of purposes, such as digital identity verification, supply chain management, and decentralized finance (DeFi) applications.

Overall, Ethereum is an important platform for the development of decentralized apps and the execution of smart contracts. Its decentralized nature and use of smart contracts have opened up new possibilities for a wide range of industries, from finance to supply chain management to social media.

4.4.2 Tailwind CSS

Tailwind CSS is a highly customizable, utility-first CSS framework for building modern web interfaces. It focuses on providing a comprehensive set of pre-designed utility classes that can be directly applied to HTML elements to style and layout them effectively.

Tailwind CSS takes a different approach compared to frameworks like Bootstrap. Rather than relying on predefined components, Tailwind CSS provides a large collection of low-level utility classes that can be combined to create custom designs. These utility classes handle everything from spacing and sizing to typography and color, allowing developers to rapidly prototype and build unique user interfaces.

One of the key advantages of Tailwind CSS is its flexibility and customizability. It provides an extensive configuration file that enables developers to customize the framework according to their project's specific needs. This allows for a highly optimized output, reducing unnecessary CSS and keeping the file size minimal.

Another notable feature of Tailwind CSS is its focus on responsive design. It includes responsive utility classes that make it easy to create layouts that adapt to different screen sizes and devices. By using these classes, developers can build fully responsive websites and applications without the need for media queries or additional CSS.

Tailwind CSS also offers a plugin ecosystem, which extends its capabilities further. These plugins provide additional features, such as dark mode support, form validation, and more, making Tailwind CSS even more powerful and versatile.

Overall, Tailwind CSS provides developers with a flexible and customizable CSS that enables rapid development and consistent styling. Its utility-first approach, extensive config options, and responsive design capabilities make it a popular choice for building modern and visually appealing web interfaces.

4.4.3 Next.js

Next.js is a popular React framework for building server-side rendered and static websites. It simplifies the development process by providing features like server-side rendering (SSR), static site generation (SSG), automatic code splitting, and API routes.

Next.js supports server-side rendering, generating initial HTML on the server and improve website performance. It also offers static site generation, where HTML pages are pre-generated at build time, resulting in faster page loads. Additionally, Next.js automatically splits JavaScript code into smaller chunks and lazy loads them, reducing bundle sizes and improving initial load times.

API routes in Next.js enable developers to create serverless API endpoints within their application, simplifying backend integration. Next.js has a robust ecosystem of plugins and extensions that enhance its capabilities, including features like internationalization, auth, and styling libraries.

With comprehensive documentation and an active community, Next.js provides an web framework for developers to build modern and high-performing web applications.

In summary, Next.js is a powerful React framework that streamlines the development of server-side rendered and static websites. Its features, including server-side rendering, static site generation, automatic code splitting, and API routes, along with its rich ecosystem, make it a popular choice for creating modern web applications.

4.4.4 NextUI

NextUI is a component library designed for Next.js applications. It offers pre-designed and customizable UI components, including buttons, forms, menus, modals, and cards. These components are optimized for Next.js projects, ensuring smooth performance and compatibility.

NextUI promotes a consistent design language with cohesive components, saving time and pro-

viding a unified user experience. The components are responsive by default, adapting to different screen sizes and devices without the need for custom CSS or media queries.With extensive documentation and examples, NextUI simplifies the development process. It provides clear usage guidelines, customization options, and integration instructions, allowing developers to quickly implement the components.

In summary, NextUI is a valuable component library for Next.js applications, offering ready-to-use UI components, seamless integration, responsive design, and documentation.

4.4.5 Node.js and Express

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine that allows developers to run JavaScript code outside of a web browser. It provides an event-driven, non-blocking I/O model that makes it lightweight and efficient for building scalable network applications.

Express.js, commonly referred to as Express, is a minimalist web application framework for Node.js. It provides a set of features and tools that simplify the process of building web applications and APIs. Express offers a robust routing system, middleware support, and various utilities to handle requests and responses.

Together, Node.js and Express enable developers to build server-side applications using JavaScript, allowing them to utilize their existing JavaScript skills and libraries. Express provides a structured and organized approach to building web applications, making it easier to handle routing, middleware, and HTTP requests.

With Express, developers can define routes and handle various HTTP methods (GET, POST, PUT, DELETE) to respond to client requests. It also supports middleware, which allows developers to add custom functionality and handle common tasks such as logging, authentication, and error handling.

The combination of Node.js and Express provides a scalable and efficient platform for building web applications and APIs. Node.js' non-blocking, event-driven architecture and Express.js

4.5 Tools

4.5.1 Vercel

Vercel is a cloud platform that provides a variety of tools and services for developing, deploying, and managing web applications. It is a popular platform for deploying static websites, frontend applications, and serverless functions. Vercel is designed to be developer-friendly and focuses on providing a simple and streamlined user experience.

One of the key features of Vercel is its serverless deployment platform, which allows developers to deploy websites and applications without the need for a dedicated server or infrastructure. Vercel supports a range of programming languages and frameworks, including React, Angular, Vue, and Node.js.

Vercel is integrated with Git, which makes it easy to deploy and manage web applications directly from a Git repository. Developers can set up automatic deployments based on code changes, create custom deployment configurations, and manage multiple environments (such as staging and production) from a single dashboard.

Overall, Vercel is a powerful platform for web application development and deployment, with a range of features and tools that can help streamline the development process and improve the performance and security of web applications. Its focus on simplicity and ease of use make it a popular choice for developers of all skill levels.

4.5.2 MetaMask Wallet

MetaMask is a popular software wallet that allows users to securely store, manage, and interact with cryptocurrencies and decentralized applications (dApps) on the Ethereum blockchain. It is a browser extension that can be used with major web browsers such as Chrome, Firefox, and Brave. MetaMask provides a simple and easy-to-use interface for managing Ethereum wallets, which are used to store and send Ether (ETH), the native currency of the Ethereum blockchain. It also supports a range of other Ethereum-based tokens, such as ERC-20 tokens, which are used in a variety of dApps and decentralized finance (DeFi) applications.

MetaMask uses advanced security features, such as encrypted data storage, password protection,

and multi-factor authentication, to help protect user funds and personal information. It also provides users with full control over their private keys, which are used to sign transactions and access wallet funds. This means that users have complete ownership and control over their funds, without having to rely on a third-party custodian.

Overall, MetaMask is a powerful and user-friendly software wallet that provides a secure and convenient way to manage and interact with cryptocurrencies and dApps on the Ethereum blockchain. Its built-in dApp browser and advanced security features make it a popular choice for users looking to explore the world of decentralized finance and blockchain applications.

4.5.3 Remix IDE

Remix is a popular web-based Integrated Development Environment (IDE) for developing and testing smart contracts on the Ethereum blockchain. It is a free and open-source tool that is designed to be user-friendly and accessible for developers of all skill levels.

Remix provides a range of features and tools for developing and testing smart contracts, including a code editor, a debugger, a compiler, and a deployment manager. It also includes a built-in Ethereum Virtual Machine (EVM), which allows developers to test their smart contracts on a local blockchain before deploying them to the main Ethereum network.

One of the key benefits of Remix is its integration with the Solidity programming language, which is used to write smart contracts on the Ethereum blockchain. Remix provides a range of Solidity-specific features, such as syntax highlighting, autocompletion, and error highlighting, to help developers write clean and efficient code.

Remix also provides a range of plugins and extensions that can be used to extend its functionality and integrate with other tools and services. For example, there are plugins for integrating Remix with Metamask, a popular Ethereum wallet, and for connecting to external APIs and data sources.

Overall, Remix is a powerful and user-friendly IDE that provides a range of features and tools for developing and testing smart contracts on the Ethereum blockchain. Its integration with Solidity and its built-in EVM make it a popular choice for developers looking to create and test smart contracts for a wide range of applications, from decentralized finance (DeFi) to supply chain management to digital identity verification.

4.5.4 Render

Render is a cloud platform that provides a range of tools and services for developing, deploying, and managing web applications and infrastructure. It is designed to be simple and developer-friendly, with a focus on providing a streamlined user experience and powerful automation tools.

One of the key features of Render is its container-based infrastructure, which allows developers to deploy and manage applications and services in a highly scalable and flexible environment. Render supports a range of programming languages and frameworks, including Node.js, Python, Ruby, and Go, and provides built-in support for popular web servers and databases.

Render also provides a range of automation tools to help simplify the deployment and management of web applications. For example, it provides automatic SSL encryption and domain management, automated deployments from Git repositories, and automatic scaling and load balancing based on traffic and demand.

In addition to its infrastructure and automation tools, Render provides a range of best optimization features, such as automatic caching, content delivery network (CDN) integration, and database replication. It also provides a range of security features, such as intrusion detection and prevention, automatic backups and disaster recovery, and compliance certifications.

Overall, Render is a powerful and user-friendly cloud platform that provides a range of tools and services for developing and managing web applications and infrastructure. Its focus on simplicity, automation, and performance optimization make it a popular choice for developers and businesses looking to deploy and scale web applications with ease.

4.5.5 Ganache

Ganache is a popular blockchain development tool used to create and test decentralized applications (DApps) in a local development environment. It is an open-source software that provides a personal Ethereum blockchain for developers to test and deploy smart contracts, develop new features, and simulate different blockchain scenarios.

Developed by the Ethereum development studio Truffle, Ganache provides a graphical user interface (GUI) that makes it easy for developers to create a blockchain network and test their DApps with-

out the need for an external network. Ganache also provides advanced features such as blockchain customization, account management, and transaction management. Ganache supports various development frameworks, including Remix and Web3.js, and can be used with various programming languages, including Solidity, JavaScript, and Vyper. It is available for download on Windows, Mac, and Linux operating systems, and is widely used by blockchain developers and enthusiasts worldwide.

4.5.6 Truffle

Truffle is a popular blockchain development framework used to create and deploy DApps on the Ethereum blockchain. It is an open-source software that provides developers with a suite of tools to manage the entire development lifecycle of a DApp, including smart contract development, testing, deployment, and monitoring.

Truffle simplifies the process of smart contract development by providing a standardized project structure and a set of built-in smart contract templates that developers can use to build their DApps. Truffle also comes with a testing framework that allows developers to write tests for their smart contracts and ensure that they are working as expected.

Truffle integrates with various Ethereum client implementations, including Ganache, Geth, and Parity, and supports various programming languages, including Solidity, Vyper, and LLL. Truffle also provides a user-friendly interface that makes it easy for developers to interact with their smart contracts and monitor their DApps.

Truffle is widely used by blockchain developers and organizations worldwide and has become an essential tool in the Ethereum ecosystem. It is actively maintained and supported by a community of developers and contributors, and new features and updates are regularly released to improve its functionality and performance.

4.5.7 AWS EC2 (Elastic Compute Cloud)

AWS EC2, or Amazon Elastic Compute Cloud, is a flexible and scalable cloud computing service offered by Amazon Web Services (AWS). It enables developers to provision and manage virtual servers, known as instances, in the cloud with ease.

With EC2, developers can select from a wide range of instance types optimized for various work-

loads and performance requirements. These instances can be quickly provisioned and configured to meet the specific needs of their applications, providing necessary computing resources on demand.

Scalability is a key advantage of EC2. Developers can scale the number of instances up or down based on application demand, ensuring optimal resource allocation and cost-efficiency. This elasticity allows developers to handle fluctuating workloads without overprovisioning or experiencing performance bottlenecks.

EC2 supports multiple operating systems, including Linux, Windows, and macOS, ease for application deployment. Additionally, EC2 offers a comprehensive set of features and services, such as load balancing, auto-scaling, and virtual private clouds, which enhance app availability, performance, and security.

Moreover, EC2 seamlessly integrates with other AWS services, empowering developers to build robust and scalable architectures. By combining EC2 with services like Amazon S3 for storage, Amazon RDS for managed databases, and AWS Lambda for serverless computing, developers can create comprehensive and highly efficient cloud-based solutions.In summary, AWS EC2 is a powerful cloud computing service that offers flexibility, scalability, and a rich set of features for deploying applications. Its extensive selection of instance types, seamless integration with other AWS services, and ability to handle varying workloads make it a preferred choice for a wide range of applications, from small-scale projects to large enterprise systems.

4.5.8 MongoDB Atlas

MongoDB Atlas is a fully managed database service provided by MongoDB, offering a cloud-based solution for deploying, managing, and scaling MongoDB databases. It provides automated and seamless scaling, allowing applications to handle growing workloads and high data storage needs. With built-in data replication and automatic failover, MongoDB Atlas ensures high availability and reliability for applications, even in the face of hardware or network failures.

Security is a priority in MongoDB Atlas, with features such as network isolation at rest and encryption in transit. These measures protect data from unauthorized access and ensure compliance with industry standards. The platform also includes powerful monitoring and management tools, providing developers with insights into database performance, alerts, and troubleshooting capability.

MongoDB Atlas seamlessly integrates with popular cloud platforms like AWS, Azure, and Google Cloud. It simplifies the deployment and configuration of databases in the cloud environment of choice. With an easy-to-use interface and integration with development and monitoring tools, MongoDB Atlas streamlines the management of databases, allowing us to focus on building applications without worrying about infrastructure management.

In summary, MongoDB Atlas is a reliable and scalable managed database service. Its automated scaling, high availability, security features, and integration with cloud platforms make it an ideal choice for developers seeking an efficient and hassle-free way to deploy, manage, and scale MongoDB databases in the cloud.

4.6 Methodology

The implementation of the Insurance Management System on the blockchain involves the creation of smart contracts using the Solidity language. The smart contracts handle various processes such as adding products, customer purchases, insurance purchase, and management. Here is the methodology for the smart contract implementation:

4.6.1 Product and Insurance Creation

1. The admin adds product details to a MongoDB database, including product information such as name, price, and specifications.
2. The admin creates insurance options for each product, specifying the coverage, premium amount, and terms.
3. The product and insurance details are stored in the smart contract, associating them with a unique identifier.

4.6.2 Customer Purchase and Insurance Selection

1. Customers browse the available products and select the desired item to purchase.
2. The customer initiates the purchase transaction, providing the product ID and payment details.
3. Upon successful payment, the smart contract verifies the purchase and generates a unique customer ID.
4. The customer selects an insurance option associated with the purchased product.
5. The selected insurance details are linked to the customer's ID and stored in the smart contract.

4.6.3 Claim Management

- Accident, Theft, or Not Working Claims

    1. In the event of an accident, theft, or malfunctioning of the purchased product, the customer files a claim.
    2. The claim details are submitted to the smart contract, including the customer ID, claim type, and relevant evidence.
    3. For accident and theft claims, the claim is verified by the police or relevant authorities outside the blockchain.
    4. Once verified, the claim details are handed over to the admin for further processing.
    5. If the claim is rejected by the verifying authority, it is directly rejected by the admin in the smart contract.
    6. If the claim is verified, the admin evaluates the claim and decides whether to proceed with a refund or replacement.

- Not Working Claims

    1. In the case of a not working claim, the customer directly submits the claim to the admin through the smart contract.
    2. The admin evaluates the claim and determines whether a refund or replacement is genuine.

- Admin Approval or Rejection of Refund

    1. The admin reviews the claim details and decides whether to approve or reject the refund.
    2. If the refund is approved, the smart contract initiates the refund transaction to the customer account.
    3. If the refund is rejected, the claim status is updated in the smart contract to reflect the rejection.

CHAPTER 5

IMPLEMENTATION

5.0.1 Implementation of smart contracts using Solidity

Solidity is a programming language used to implement smart contracts on blockchain platforms like Ethereum. It allows developers to define the rules and behaviors of their contracts, enabling the execution of decentralized applications (dApps) and the automation of trustless transactions with predefined conditions and logic.

The Solidity code represents a smart contract called ”ProductInsurance” for managing product insurance on the Ethereum blockchain. The contract allows users to add products, purchase insurance for the products, file claims, and perform various administrative functions related to insurance status and refunds.

The contract includes the following functionalities:

- Adding a Product: Users can add a product by providing details such as product ID, brand, model, product image, product price, purchase date, and owner address.
- Adding Insurance: Users can add insurance for a product by specifying the product ID, insurance start and end dates, and insurance price. The insurance status is set to ”Valid” for the product, and the insurance amount is transferred to the contract’s admin.
- Adding a Claim: Users can file a claim for a product by providing the product ID. The claim can only be filed if the product has valid insurance and the product’s warranty has not expired. The insurance status is updated to ”Claim Filed” for the product.
- Updating Insurance Status: The contract provides a function to update the insurance status of a product. This function can be called by the admin or police. The status can be set to values such as Invalid, Repair, Replace, Refund Approved, Refund Success, Claim Filed, Under Investigation, or Claim Rejected. The description of the insurance status change is also provided.
- Getting User’s Products: Users can retrieve a list of products they own.

- Getting All Claims: The admin can retrieve a list of all claims filed by users.
- Getting Claims Under Investigation: The police can retrieve a list of claims that are currently under investigation.
- Updating Insurance Status by Police: The police can update the insurance status of a product to either Refund Approved or Claim Rejected. This function is restricted to the police account.
- Sending Refund: The admin can send a refund to the owner of a product if the insurance status is set to Refund Approved, and the correct amount is sent with the function call.

The contract utilizes various modifiers to restrict access to specific functions, such as ”onlyAdmin” and ”onlyPolice,” ensuring that only authorized accounts can perform certain actions.

5.0.2 Next.js React framework for frontend UI development

Next.js is a React framework used for frontend development. It allows developers to build server-side rendered or statically generated React applications. Next.js provides a flexible and efficient environment for building UI components and managing the frontend of web applications.

Next UI is not a widely known or commonly used UI library. Next.js itself doesn't impose any specific UI library, giving you the flexibility to choose the UI library or component framework that best suits your needs.

5.0.3 Using MongoDB database for storing product information

MongoDB is a NoSQL database that is commonly used for storing and managing data. In the context of the implementation described, MongoDB can be used to store product information and map usernames with wallet addresses while storing other data on the blockchain.

To use MongoDB for storing product information:

- Create a MongoDB collection for storing product information.
- Define the necessary fields such as product ID, brand

  , model, image URL, price, purchase date, etc. Each document in the collection represents a unique product.
- Add a field in the MongoDB collection to store the wallet address associated with each product owner. This field can be named ”walletAddress” or similar. When adding a product, you can include the wallet address of the owner in the document.

- Use the blockchain for storing other data. Use the Solidity smart contract to manage insurance-related functionalities and store insurance-related data on the blockchain. Store the product ID, insurance status, insurance status description, and any other insurance-related data in the Solidity contract's mappings and struct variables.

By separating the data storage between MongoDB and the blockchain, you can utilize MongoDB for flexible querying, indexing, and efficient storage of product information and user mappings. The blockchain can then handle the immutability and transparency aspects related to insurance-related data and actions.

5.0.4 Dynamically changing price and product details

Storing product details in a MongoDB database can provide more flexibility for the admin user to dynamically change prices and product details.

To achieve this:

- Create a MongoDB collection for storing product details.
- Implement CRUD operations for product management. The admin user can add new products, retrieve existing product details, update product information such as price and description, and delete products if needed.
- Implement appropriate authentication and authorization mechanisms. Secure the db to ensure only authorized users, such as the admin user, can perform CRUD operations. Implement authentication to validate the admin user's identity before allowing access to the product management functionalities.
- Integrate with the Solidity smart contract. When adding a product or updating product details in MongoDB, you can also trigger corresponding events or function calls in the Solidity smart contract. For example, when the admin user updates the price of a product in MongoDB, you can invoke a function in the Solidity contract to update the corresponding product's price in the blockchain.

This approach allows for more dynamic management of product details while leveraging the immutability and transparency provided by the blockchain for insurance-related data and actions.

5.0.5 Business logic written in Solidity smart contract deployed on Ethereum blockchain network

Deploying a Solidity smart contract on the Ethereum blockchain network for insurance management ensures transparency, immutability, and decentralized execution of insurance-related operations.

To implement this:

- Write the necessary contracts needed for the insurance management system in Solidity.
- Define the desired contract structure, functions, and data structures to represent your insurance management system.
- Implement the desired insurance-related functionalities, such as adding insurance to a product, filing a claim, updating claim status, and handling refunds.
- Compile the Solidity code using a Solidity compiler. The compilation process transforms your contract into a format that can be deployed on the Ethereum network.
- Deploy the contract to the Ethereum network using an Ethereum development framework like Remix. Specify the desired network (e.g., testnet, mainnet) and provide any necessary deployment configurations.

Deploying the contract on the Ethereum blockchain ensures that the insurance system operates on a decentralized network with built-in trust and transparency.

5.0.6 Interacting with MongoDB

To interact with MongoDB in the context of the described implementation, you can set up a Node.js Express server on AWS EC2 to handle the interaction with MongoDB Atlas and design API endpoints for a Next.js frontend.

Here's a high-level overview of the steps involved:

- Create an EC2 instance on AWS and configure it with the necessary resources, such as the desired operating system and network

  settings.
- Install Node.js and other dependencies on the EC2 instance.
- Create a new Node.js project and set up an Express server.
- Obtain the connection string for your MongoDB Atlas cluster.

- Set up models and schemas to define the structure and behavior of your data.
- Define the required API endpoints in your Express server to handle CRUD operations and other functionalities.
- Configure your AWS EC2 instance to listen on the desired port and handle incoming HTTP requests.
- Start the Node.js Express server on your EC2 instance and ensure it is accessible from the internet.

This setup allows your Next.js frontend to communicate with the Express server, which in turn interacts with MongoDB to perform database operations.

5.0.7 Frontend hosting using Vercel

Vercel is a popular hosting service that specializes in deploying frontend applications, particularly those built with Next.js. It is a cloud platform that offers serverless deployment and hosting for web applications. Vercel provides developers with tools to build, deploy, and scale frontend applications and websites easily.

To host your frontend using Vercel:

- Sign up for a Vercel account.
- Connect your Vercel account with your Git repository, such as GitHub.
- Configure your Vercel project settings, such as specifying the framework (Next.js) and build settings.
- Trigger a deployment by pushing your code changes to the connected Git repository.
- Vercel will automatically build and deploy your Next.js application based on the latest code changes.
- Access your deployed application through the provided Vercel domain or set up a custom domain.

Vercel simplifies the deployment process and provides features like auto deployments based on code updates, serverless functions, and edge caching for improved performance.

5.0.8 Using GitHub for version control

GitHub is a popular platform for hosting Git repositories and managing code versions. It provides features for collaborative software development and version control. Integrating GitHub with Vercel allows for seamless deployment and automatic updates whenever changes are pushed to the repository.

To use GitHub for version control:

- Create a GitHub repository for your project.
- Push your code to the GitHub repository, either using Git commands or through a Git client.
- Connect your GitHub repository to your Vercel account.
- Configure the deployment settings in Vercel to specify the GitHub repository and branch to deploy from.
- Vercel will monitor the connected GitHub repository for code changes and automatically trigger deployments whenever changes are pushed.

This integration ensures that your deployed frontend stays up-to-date with the latest code changes in your GitHub repository.

CHAPTER 6

APPLICATION RESULTS

6.0.1 Admin Page

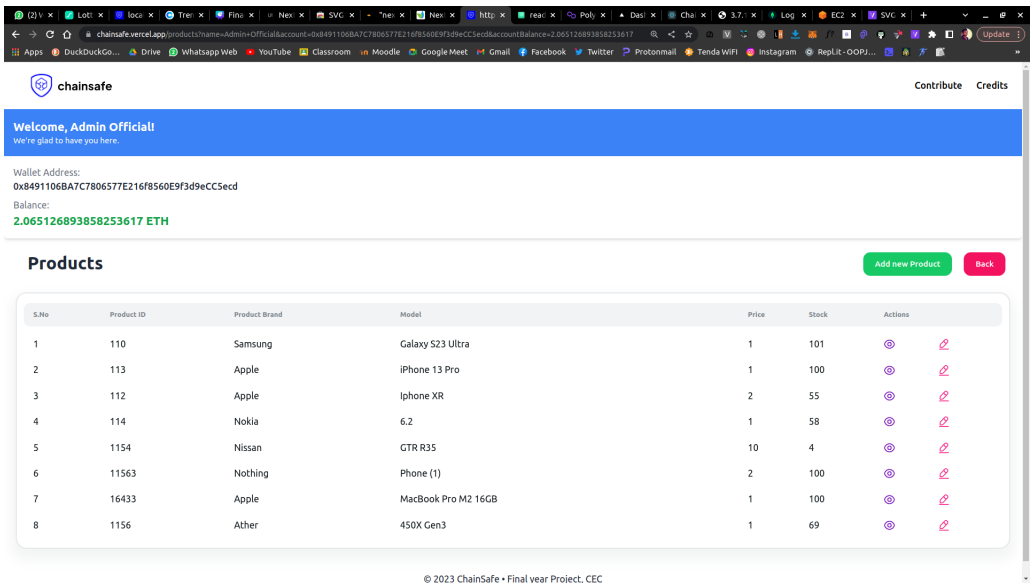


Figure 6.1: Admin View - Options to edit, add products.

In the admin view, users have the ability to edit existing products and add new ones, providing a comprehensive set of options for managing and updating the product inventory.

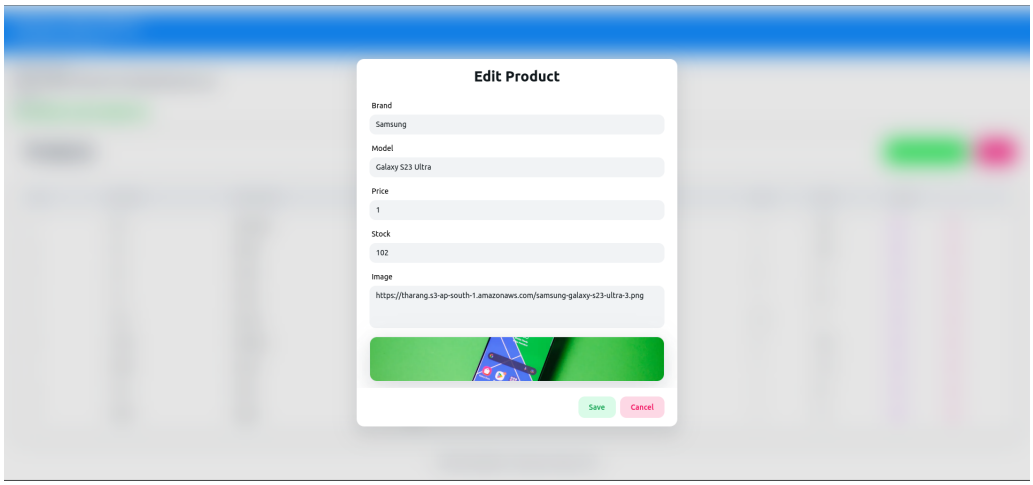


Figure 6.2: Admin View - Options to edit product

Options to edit various product details such as brand, model, stock, image link, and price etc.

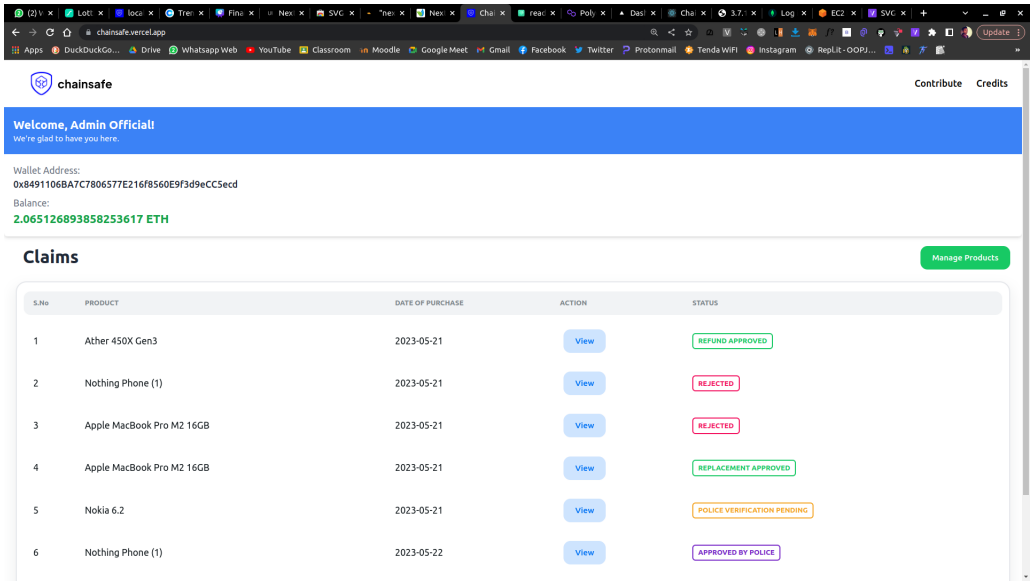


Figure 6.3: Admin view - Status of all insurance claims

This admin view provides an overview of the status of all insurance claims, including pending verification, refund success, and rejection, among others.

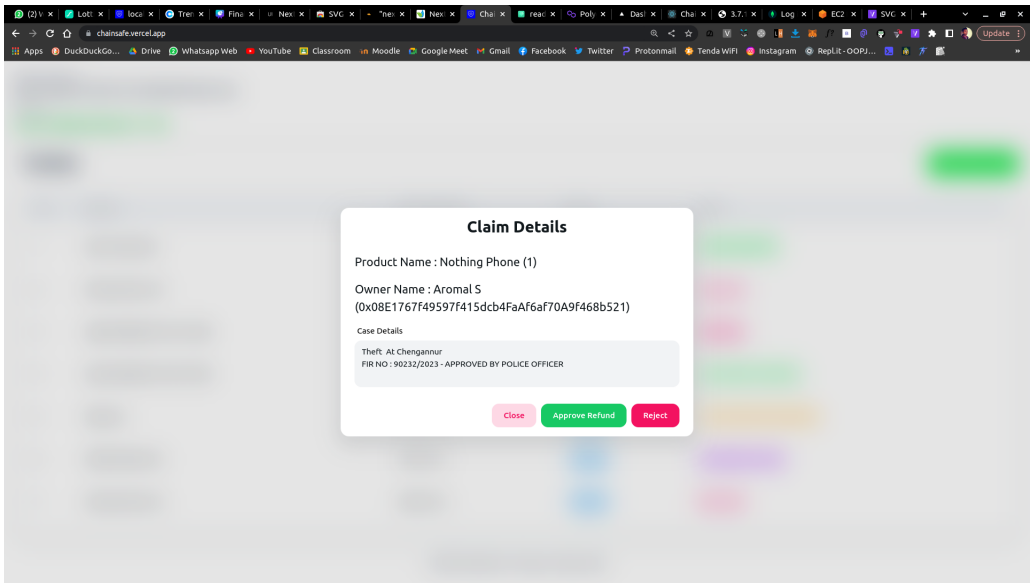


Figure 6.4: Admin view - Verification Option

After reviewing the submitted supporting documents and receiving confirmation from the police, the admin has the option to either verify the claim or reject it.

6.0.2 User Page

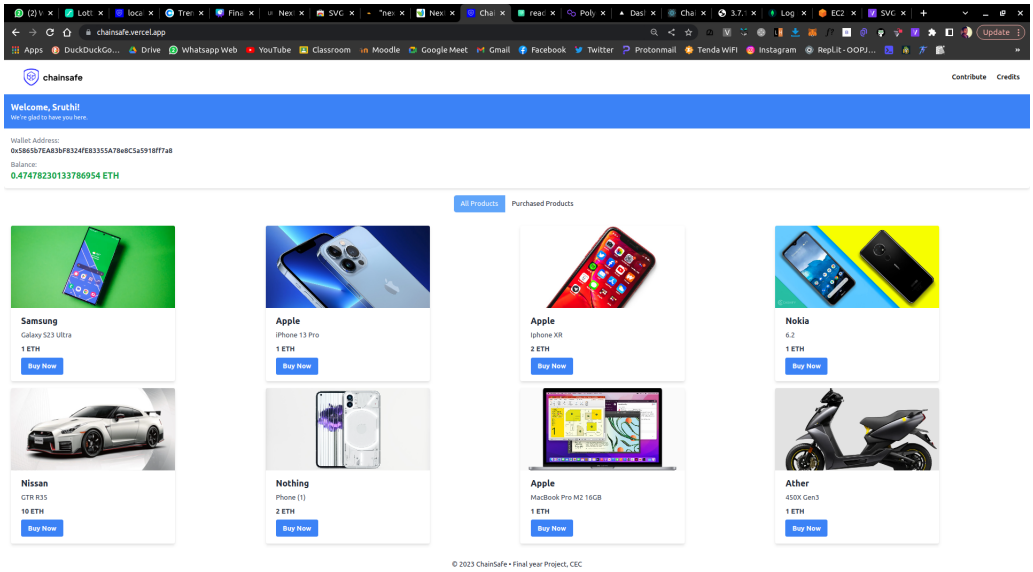


Figure 6.5: User View - List of available products

Shop displays a list of available products, each presented as a product card with a buy option. Users can easily browse through the offerings and make purchases directly from this interface.

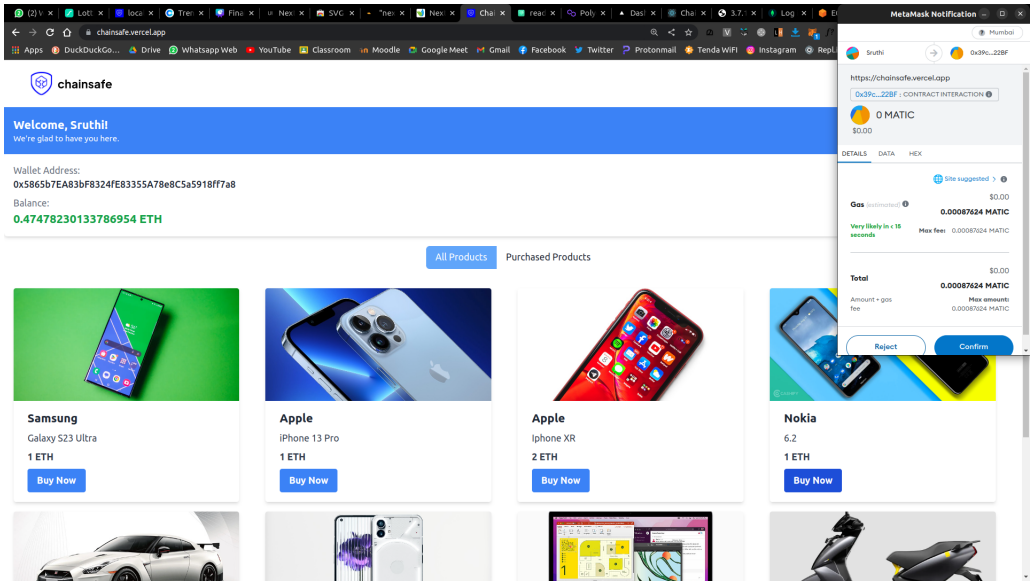


Figure 6.6: User view - purchasing the product by using cryptocurrencies

To make a purchase using payment can be done with MetaMask wallet balance and provide the corresponding wallet address. confirmation for the transaction.

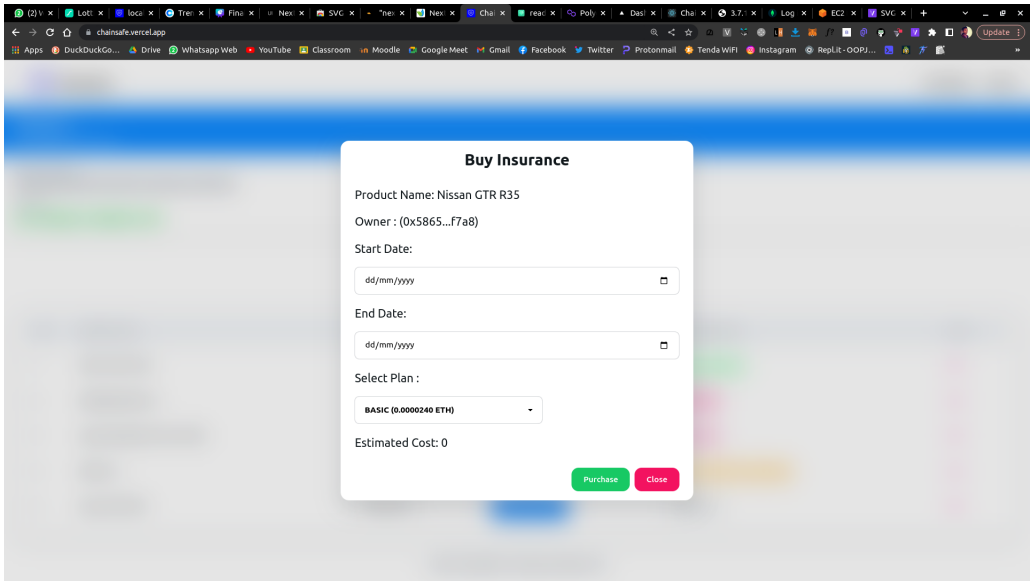


Figure 6.7: User can buy insurance after product purchase

After completing the product purchase, users have the option to select the tenure and an appropriate insurance plan to purchase ensure coverage for their newly acquired item.

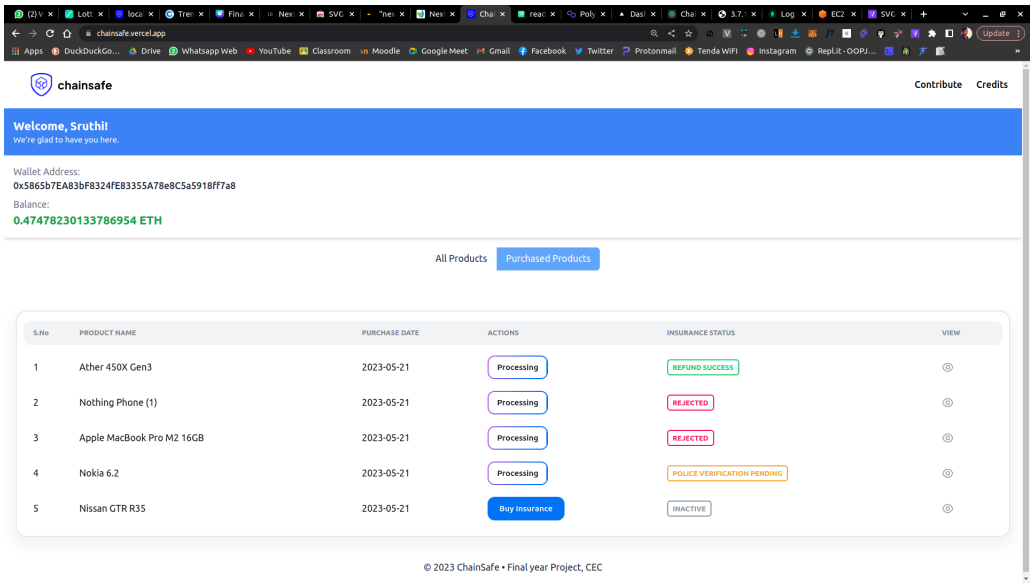


Figure 6.8: User View - Purchased Items Insurance Status

User's view of their purchased items along with the status of their insurance claims. Users are provided with options to submit claims, upload supporting documents, and access relevant information.

6.0.3 Police Page

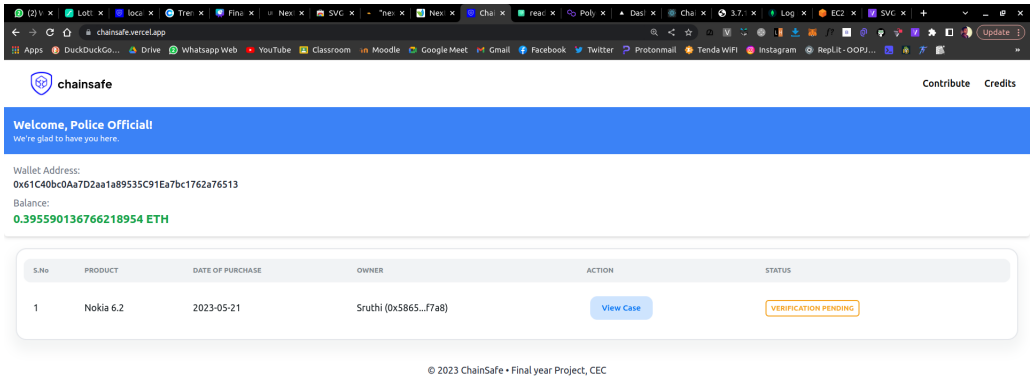


Figure 6.9: Police View - Pending Verifications

Police view of a verification window, where all pending claims are visible upon login. Processed claims are removed from the list as they are completed, ensuring that pending claims are always visible for further action.

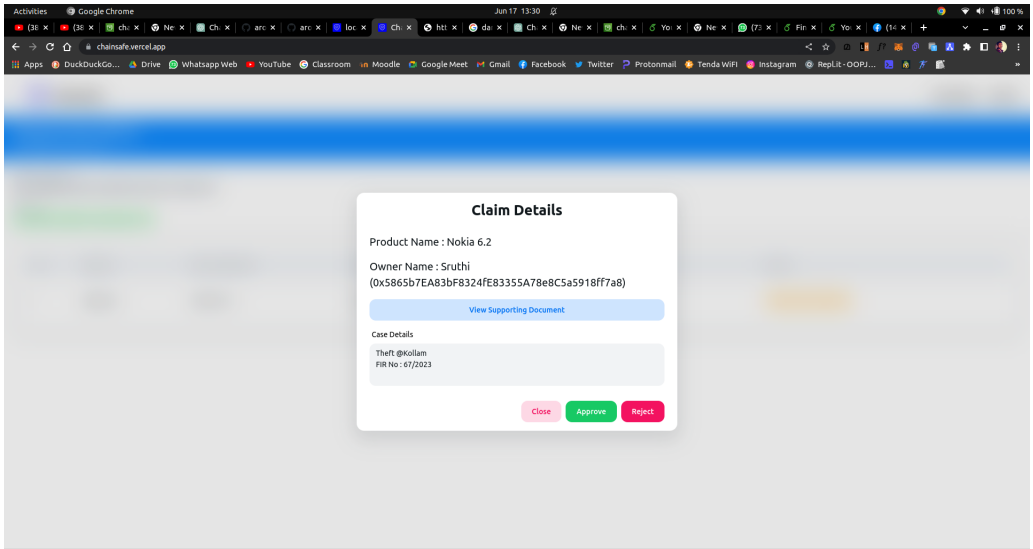


Figure 6.10: Police View - Verification Window, View Supporting Documents

Option to thoroughly review the details and supporting documents for verification purposes, allowing the user to either accept or reject the information based on their evaluation.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.0.1 Conclusion

In conclusion, a blockchain-based insurance management system has the potential to revolutionize the insurance industry by addressing existing challenges and improving efficiency, security, and trust. By leveraging blockchain technology, insurers can eliminate intermediaries, automate processes, reduce errors and fraud, and achieve cost savings. The immutable and transparent nature of blockchain enhances data integrity, regulatory compliance, and accountability. Furthermore, a blockchain-based system streamlines the claims process, improves customer satisfaction, facilitates data sharing for underwriting and risk assessment, and enables real-time fraud detection. While challenges like scalability, regulatory compliance, legacy system integration, and data privacy need to be overcome, the increasing adoption of blockchain in insurance holds promise for a more efficient and secure ecosystem.

7.0.2 Future Scope

Looking ahead, the future of blockchain-based insurance management systems holds immense potential. As the technology continues to evolve, we can anticipate further advancements and adoption in the insurance industry. Efforts should focus on addressing scalability concerns and ensuring regulatory compliance to encourage wider implementation. Additionally, integrating blockchain solutions with existing legacy systems will be crucial for successful implementation. Maintaining a balance between data privacy and transparency will also be essential. Further exploration of use cases and collaborations between insurers, technology providers, and regulatory bodies can drive innovation and foster the growth of blockchain-based insurance management systems. Ultimately, embracing blockchain technology has the potential to create a more efficient, secure, and customer-centric insurance ecosystem in the future.

REFERENCES

[1] **V. Aleksieva**, **H. Valchanov**, and **A. Huliyan**, "Implementation of Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Insurance Services," in *2020 International Conference on Biomedical Innovations and Applications (BIA)*.

[2] **V. Aleksieva**, **H. Valchanov**, and **A. Huliyan**, "Smart Contracts based on Private and Public Blockchains for the Purpose of Insurance Services," in *2020 International Conference Automatics and Informatics (ICAI)*.

[3] **S. Alwis and T. M. K. K. Jinasena** "A Blockchain-Based Decentralized Insurance Platform," in *International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 2022.

[4] **A. Averin, E. Musaev, and P. Rukhlov** "Review of Existing Blockchain-Based Insurance Solutions," in *2021 International Conference on Quality Management, Transport and Information Security*.

[5] **N. R. Bhamidipati** "ClaimChain: Secure Blockchain Platform for Handling Insurance Claims Processing," in *2021 IEEE International Conference on Blockchain (Blockchain)*.

[6] **I. Nath** "Data Exchange Platform to Fight Insurance Fraud on Blockchain," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*.

[7] **Z. Wan, Z. Guan, and X. Cheng,** "PRIDE: A Private and Decentralized Usage-Based Insurance Using Blockchain," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom)*.

[8] **T. -W. Yu, A. -P. Wang, L. -M. Tseng, and W. -M. Tsao** "A Preliminary Study of the Impact of Blockchain Technology on the Application Level of Insurance Industry," in *2021 IEEE International Conference on Social Sciences and Intelligent Management (SSIM)*.

[9] **V. Aleksieva**, **H. Valchanov**, and **A. Huliyan** "Design of Blockchain Application Framework for Claims Platform of Flight Delay Insurance," in *2022 Global Conference on Robotics, Artificial Intelligence and Information Technology*.

[10] **P. Hacker (ed.), I. Lianos (ed.), G. Dimitropoulos (ed.), S. Eich (ed.)** "Regulating Blockchain: Techno-Social and Legal Challenge," *Oxford Academics.*

[11] **”Solidity Documentation”** [Online] : https://docs.soliditylang.org.

[12] **”NextJS Documentation”** [Online] : https://nextjs.org/docs.

[13] **”NextUI Documentation”** [Online] : https://nextui.org/docs/.

[14] **”Metamask Developer Documentation”** [Online] : https://docs.metamask.io/.

[15] **”MongoDB Documentation”** [Online] : https://www.mongodb.com/docs.

[16] **”Node.js Documentation”** [Online] : https://nodejs.org/en/docs.

[17] **”OpenAI - ChatGPT”** [Online] : https://chat.openai.com/.

[18] **”AWS Documentation”** [Online] : https://docs.aws.amazon.com/ec2/index.html.