The University of Hong Kong

FYP21036 2021-22

Final Report

Supply Chain Management using Blockchain and NFT

Increasing Transparency & Traceability

Group Member	UID
Agarwal, Siddharth	3035555482
Bera, Navoneel	**3035551735**
Lohia, Suyash	3035550406

Supervisor: Dr. Yuen, John

Date of submission: 18th April 2022

Abstract

Globalization has led to increasingly complex supply chains which are relied upon by a multitude of stakeholders. This has resulted in an increased importance of supply chain management. Optimizing the supply chain can be beneficial not only to business owners but also to consumers. There are various areas that can be optimized in the supply chain, out of which an important one is supply chain visibility. Currently, 57% of firms report poor supply chain visibility and 83% of consumers are willing to pay more for goods that are ethically sourced in comparison to goods whose sourcing practices are opaque. Increasing visibility in supply chains can grant benefits such as improved inventory management to business owners and verification of ethical sourcing practices to consumers. This project aims to offer a solution that increases supply chain visibility for both businesses and consumers in the form of a blockchain and NFT-based platform. It uses Polygon, an L2 network on top of Ethereum, in order to achieve decentralization, high transactional throughput, and low transaction fees. It provides a mobile application for the tokenization and transfer of shipments through NFTs and a web application through which company managers can manage users, products, and shipments. The mobile application also provides consumers the ability to trace the origin of goods that they buy so as to ensure ethical sourcing practices. The platform has a few limitations of which an important issue is the availability of the internet and smartphones for all supply chain actors and this might hinder the adoption of the platform. This report presents a high-level overview of the working of the platform, discusses the methodology behind the implementation, justifies engineering choices for the technologies used, and showcases the final working prototype of the system including the mobile and web applications and the blockchain. The future work for the platform includes implementation of a QR code based system to track homogenous goods, the introduction of analytics that help identify points of disruption and integration with existing ERP systems so as to make them more uniform.

Acknowledgment

I would like to express my gratitude to the supervisor of this final year project, Dr. John Yuen, for his guidance and support. I am also thankful to the Department of Computer Science, HKU for providing me with this opportunity.

List of Figures

List of Tables

Abbreviations

API: Application Programming Interface

CRUD: Create, Read, Update, Delete

ERP: Enterprise Resource Planning

IPFS: Interplanetary File System

MERN: MongoDB, Express, React, Node

NFT: Non-Fungible Tokens

SaaS: Software as a Service

SCM: Supply Chain Management

TPS: Transactions Per Second

UI/UX: User Interface/ User Experience

Table of Contents

1. Introduction

Better supply chain visibility gives significant advantages, not only to businesses but also to consumers. Businesses can manage inventory more efficiently, identity, manage and mitigate risks as well as provide better customer service. Consumers can identify the origins of products to ensure the ethicality of their sourcing practices. Thus, there is a need for a platform that increases visibility while being relatively easy to integrate into current supply chains.

1.1. Background

There have been massive socio-economic shifts in the world over the past few decades. A rapidly growing middle-class and the advent of the information age and the internet alongside rapid globalization have led to the formation of services like Amazon as well as the expansion of retail stores like ParknShop. To the consumer, these businesses provide an accessible and fast method of obtaining goods. However, at their core, they are complicated and intricate networks of suppliers, consumers, and everything in between, working tirelessly across transnational boundaries. These networks are called supply chains.

Techniques that help to manage and optimize supply chains are called supply chain management (SCM) [1]. Streamlined and more transparent supply chains lead to not only reducing delivery times but also help to cut down excess inventory, manage and mitigate risks more efficiently across the whole supply chain [2] and enable faster cash-to-cash cycles (time between paying suppliers and being paid by consumers) [3]. To reflect this business advantage, the SCM industry is estimated to grow from having a market value of 15.58 billion USD in 2019 to almost 37.5 billion USD in 2027 [4]. This growth is fueled by a number of factors, among which one is the rising demand for better technology that can help to improve supply chain visibility.

Supply chain visibility is the ability to track the status of different goods at each point in the supply chain [5]. Having a highly visible supply chain enables businesses to provide better customer service, manage and mitigate risks, and also help in managing inventory in a more efficient manner [5]. So it is surprising that 57% of business owners report poor supply chain visibility [6].

On the other hand, consumers have different concerns when it comes to supply chain visibility. As the world becomes more interconnected and supply chains continue to expand, acquiring goods and products which are produced in a diverse set of countries becomes increasingly easy. However, this ease of purchase has led to increasing concerns about how certain products are sourced. A prime example is the sourcing of avocados. Avocados have seen a recent boom in sales in many countries owing to their nutritional value. The international market for avocados, which was around $12.8 billion in 2019, is estimated to grow up to $17.9 billion representing a growth of 40% in 6 years [7]. This growth, however, is encouraging a range of malpractices by producers across the world. Rapid avocado farming poses concerns about illegal deforestation [7], poisoning of water bodies through pesticides, and exploitation of workers by cartels in Mexico (which is the world's largest producer of avocados). In view of such concerns, consumers have increasingly become conscious of purchasing avocados, and other goods in general, that align with their personal humanitarian and environmental values.

According to an OpenText survey, 88% of global consumers prefer to buy from producers which have implemented ethical sourcing strategies and 83% of consumers are willing to spend more on ethically sourced products [8]. Improved supply chain visibility would help consumers trace the origin and sourcing ethicality of the goods they purchase, bolstering ethical sourcing practices amongst producers.

Given the importance that visibility plays not only for businesses but also for consumers, a platform that enables greater visibility throughout the whole process while maintaining data integrity and transparency is needed.

1.2. Objectives

The main objective of this project is to use blockchain technology and non-fungible tokens to develop a platform that improves visibility for stakeholders in the supply chain. The platform will enable the creation, management, and, tracing of tokenized goods by supply chain managers and actors as well as allow consumers to track the origin of goods before buying them. Thus, the benefits of the platform will be twofold: business owners will benefit from increased supply chain visibility and consumers will be able to buy products that match ethical sourcing practices.

1.3. Scope and deliverables

The team aims to develop and deliver the following in order to meet the set objectives:

1. A smart contract on a blockchain that allows for the creation, storage, and transfer of non-fungible tokens.
2. An intuitive web application to be used by company managers for the following purposes:
    a. Creation and management of user accounts and linking them to cryptographic wallets.
    b. Creation and management of products that can be used to create shipments.
    c. Management of shipments created by viewing location and ownership history and resolution of issues arising from product loss.
3. A mobile application that interfaces with the blockchain and is used by:
    a. Supply chain actors including end stores to create, view, transfer, and accept/reject tokenized shipments
    b. Consumers to trace the origin of goods.

1.4. Outline

The report begins by introducing the background of the project and gives motivations for the objectives formed. The scope and the deliverables of the project are also elaborated upon. Then it presents a literature review of the technologies used and discusses some related work. Next, a high-level overview of the platform workflow and the methodology that has been followed for realizing the proposed system architecture is presented. Following this, the practical implementation of each part of the platform is discussed and the final results are showcased. Finally, the limitations of the platform and challenges faced are presented and future plans for the platform are discussed.

2. Literature Review

In order to have a better understanding of how the different technologies used would interface with each other to create a seamless experience on the platform, a literature review was carried out. The technologies themselves as well as the current state of the technologies used in SCM were surveyed.

2.1. Technologies Used

Two main technologies are to be used extensively in the development of the platform: blockchain and NFTs. This section gives an overview of these technologies.

2.1.1. Blockchain

Sometimes called Distributed Ledger Technology, Blockchain is a system of storing data such that the data remains immutable (non-tamperable) [9]. A blockchain consists of units called blocks that contain data and reference each other in order to create a "chain" (Fig. 1).

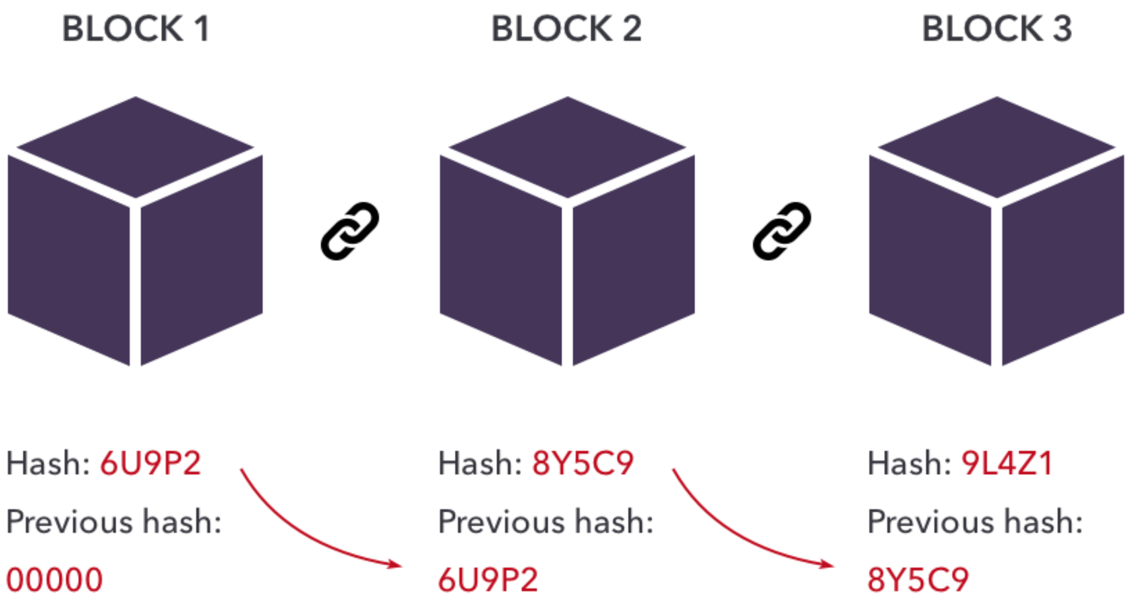

Fig 1. Simplified structure of a blockchain

Each user of the blockchain has a copy of the entire blockchain, thus making it distributed in nature. Immutability and chaining are achieved through a procedure called hashing [9]. Hashing takes data and maps it onto a unique and non-reversible representation. In the context of blockchain, hashing is used to create a hash of a block [9]. This way each block has a unique representation that is representative of the data stored in it. Chaining is achieved through blocks referencing the hash of the previous block (Fig. 1).

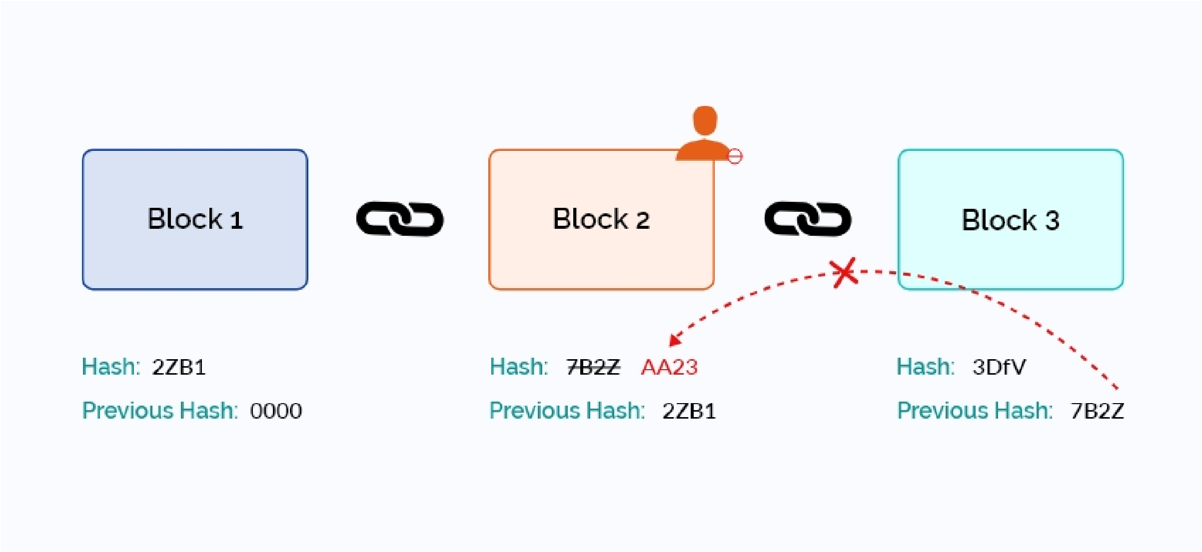

Fig 2. Reference to block destroyed when data changed

So, when the data of a block changes, its hash also changes and this makes all subsequent blocks inaccessible (Fig. 2).

One workaround is to recalculate the hashes of every subsequent block. This workaround, however, is almost impossible in a blockchain on account of its distributed nature and a mechanism called proof-of-work. When blocks are created, before being added to the blockchain two steps need to happen [10]. First, a computationally intensive procedure called proof-of-work needs to be performed and attached to the block. Each task for the proof-of-work is generated dynamically and takes about 10 minutes to complete, making it time-consuming [11]. Second, because of the distributed nature of blockchain, a block has to be "accepted" by more than 50% of the users in order to be added [12]. When an existing block is modified and hashes change the proof-of-work becomes invalid and needs to be redone for all the subsequent blocks. Moreover, a nefarious actor planning on changing data on the blockchain would not only need to redo the proof-of-work (which in itself is time consuming) but also control 51% of the computing power of the blockchain network (called the hash rate) for the block to be accepted by the network [13].

The distributed and immutable nature of blockchains enables secure data storage. In this project, blockchain is a natural solution since data about goods and the transactions taking place in the supply chain needs to be stored in a manner that ensures its access by a multitude of stakeholders while maintaining its integrity.

2.1.2. Scalability of Blockchains: The Need and Solutions

A blockchain network's potential to handle high transaction throughput and increasing the number of nodes is called scalability. Scalability is a major issue that needs to be kept in mind while designing any system with blockchain as a component. Scalability in terms of throughput of transactions can be measured using a metric called Transactions per Second or TPS. The TPS of the popular Ethereum blockchain network currently stands at 12-15 TPS [14]. This is quite low for systems looking to achieve high transactional throughput and might drive them towards using conventional centralized systems. To tackle the problem of low TPS and scalability, solutions have been proposed for both L1 and L2 blockchain networks.

L1, or Layer 1, networks are base blockchain networks that are standalone [15]. They process, verify and finalize transactions on their own network [15]. Ethereum and Bitcoin are examples of L1 blockchains [15]. Solutions such as sharding and different consensus mechanisms have been proposed to solve the scalability issues that L1 networks face [15]. However, these are time-consuming and costly to implement practically.

L2, or Layer 2, networks are separate protocols that are built on top of existing L1 networks. They are mainly used for performing time-consuming L1 transactions by processing them off the L1 network. This helps in tackling the aforementioned scalability issues while still maintaining the aspects of data security and decentralization promised by L1 networks [16]. Polygon and the Bitcoin lightning network are examples of L2 networks.

The proposed platform is intended to serve global supply chains which have a large number of actors and thus support for high transactional throughput is imperative. L1 blockchains, with their low TPS and scalability issues, would therefore not be suitable for such an application. L2 blockchains solve these issues and as an added benefit also reduce the fees required per transaction [16]. Thus L2 networks are an apt solution for the requirements of the platform.

2.1.3. Non-Fungible Tokens

Fungible assets are assets that are interchangeable. For example, all 10 HKD bills are interchangeable. Non-fungible assets on the other hand have unique properties that make them non-interchangeable [17]. For example, there is only one Mona Lisa. Non-fungible tokens (NFTs) are representations of non-fungible real-world or digital assets stored on blockchains such as Ethereum or HyperLedger Fabric that can be bought and sold.

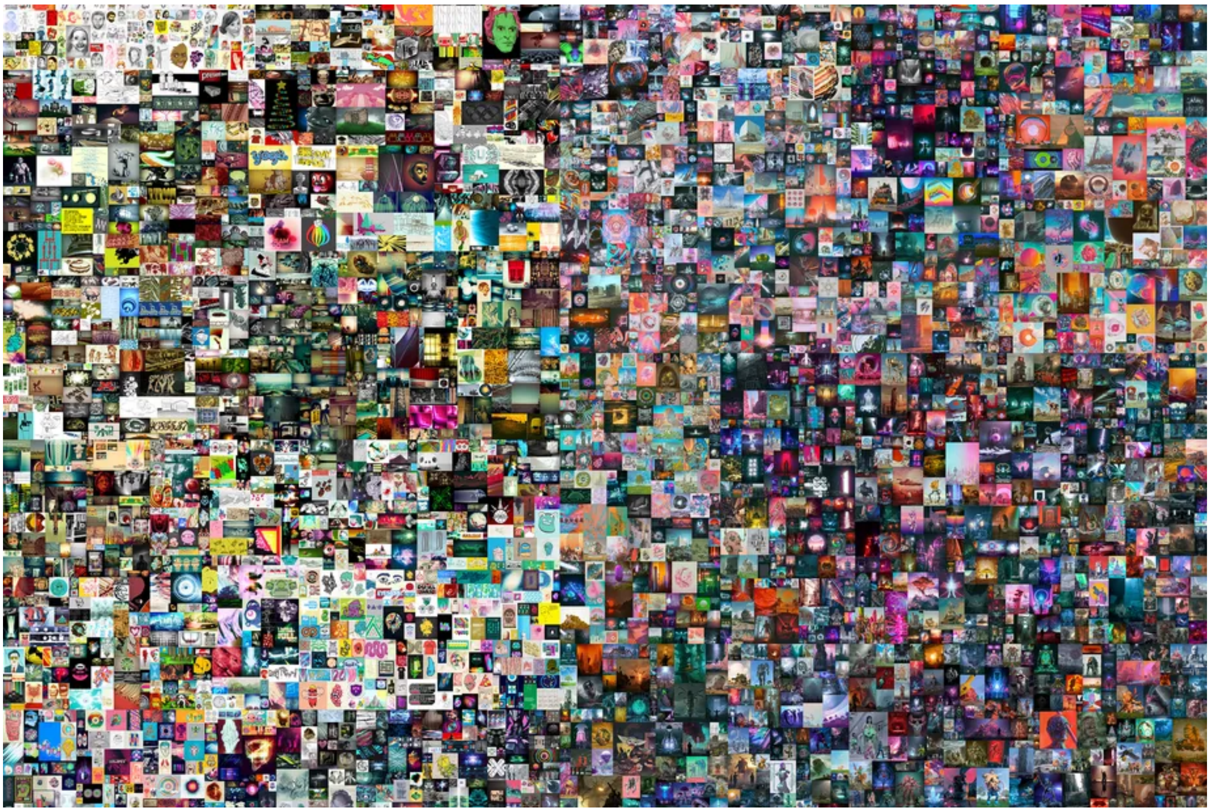

Fig 3. Beeple's record-breaking NFT

NFTs recently came into public view after a few high-value sales of artwork represented as NFTs. The artist Beeple sold his NFT (Fig. 3) for 69 million USD at a Christies's auction [18].

Each NFT has a unique identifier associated with it along with metadata in the form of a JSON object. Since the cost of storing data on-chain is expensive, the actual JSON is stored on a distributed data storage platform such as IPFS, and only a linkage to this data is stored in the actual token.

NFTs offer a convenient and easy way to check and trace the ownership of items since each change of ownership is recorded on the blockchain. It also simplifies the process of transfer of ownership [19]. NFTs, which will have their metadata stored on IPFS, will thus allow for the tokenization of each good in supply chains and offer a streamlined solution for the transfer of ownership of goods by agreement between both parties and for tracing and tracking goods within the supply chain leading to increased supply chain visibility.

2.1.4. Interplanetary File System (IPFS)

IPFS or Interplanetary File System, is a peer-to-peer distributed file system for data storage. A cryptographic identifier, called the CID is generated for each file uploaded to IPFS and the same CID is used to retrieve it. NFTs can utilize IPFS by storing the IPFS CID in them instead of storing the actual data [20]. This offers various advantages [21]:

- The IPFS hashing system in the form of a CID ensures data is tamper-proof
- NFT data can be directly accessed using the CID in a decentralized manner
- Faster data access owing to higher bandwidth

2.2. Related works

Since the introduction of the current form of blockchain technology in 2012 [22], it has been used in a number of industries including finance and healthcare. However, applications in the field of Supply Chain Management have been few so far and the industry still relies on Enterprise Resource Planning (ERP) systems for many tasks. These systems are unable to handle the rising complexity of global supply chains. A lack of uniformity between different ERP systems makes it difficult for two ERP systems to interact amongst themselves leading to decreased visibility in supply chains [23]. Thus, a better solution is needed that is either supplementary to ERPs, can readily integrate with them, or is a replacement.

One of the biggest undertakings of using blockchain in SCM has been by IBM. Their services in this space include increasing visibility, improving container logistics, and creating a safer food supply chain (through IBM Food Trust) among many others [24]. However, the implementation of these services via just the blockchain raises two main concerns:

1. The blockchain cannot ensure non-falsified data is being entered [19]. A user can easily enter falsified data and record it on a blockchain.

2. Smart contracts used in the process of exchange and transfer are not easily traceable through the blockchain [19].

Moreover, these services are not meant for the consumer but for enterprises looking to improve and optimize their supply chain. Even though consumers form an important part of supply chains, solutions that assist them in knowing more about the products they are buying and the supply chain behind them are few and far in between.

The work performed by IBM legitimizes the use of blockchain in supply chain management. However, as previously discussed, they have some issues. The proposed platform overcomes these issues with the application of NFTs. NFTs simplify the process of transfer of goods and traceability of these transfers while maintaining data authenticity and integrity due to their inherent unique and non-fungible property. The proposed platform offers services that can be taken advantage of by both enterprises and consumers alike.

3. Methodology

This section elaborates on the workflow of the platform and gives a brief high-level overview of the architecture of the underlying system.

3.1. Workflow of the platform

The platform consists of two complementary applications: one web and one mobile application. The web application is intended to be used by company administrators while the mobile application is to be used by both supply chain actors as well as consumers. The entire process starting from the creation of users to the delivery of goods as well as tracing of origin by the consumer is as follows (Fig. 4):

- In the initial setup stage, company management creates new users (representing supply chain actors and end stores) through the web application by assigning usernames, and passwords as well as wallet credentials by inputting information into a form. This process allows supply chain actors to gain access to the mobile application as well as links their account to a company-supervised cryptocurrency wallet which will store the tokenized shipments as well as provide money for all NFT-related transactions. The details of the wallet are not revealed to the users of the mobile application in order to keep application workflows simple and secure.

- Company management enters product information on the web interface. These products will be made into shipments. The web interface also allows management to edit and delete products.

- The first point of contact in the supply chain selects products and enters their quantity in the mobile application in order to tokenize them into a shipment consisting of one or more products. The NFT of the shipment will be in the user's linked wallet now, making them the owner of the shipment. Being the owner of the shipment gives the user the responsibility for that shipment.

- The transfer of ownership, as well as responsibility, takes place repeatedly between supply chain actors till the item reaches its destination or faces rejection. This entire process can be tracked on the web application, where managers can view the location and ownership history for each shipment.

- If at any point during the transfer process, an actor discovers a loss of goods, they reject the transfer of ownership. The responsibility of the shipment thus falls upon the last person who was the owner of the associated token.

- Upon rejection, the managers are notified about the issue through the web interface. A company manager then performs a manual check on the item in question.

- Having inspected the faulty shipment, the manager can resolve the issue on the web dashboard by editing the quantity of the items in the shipment. This leads to the creation of a new NFT using the wallet of the user who last held the faulty shipment. This new NFT is based on the old one and thus preserves corresponding metadata such as the origin.

- The new shipment is now with the supply chain actor under whose responsibility the issue occurred and the process of transfer can continue.

- The shipment reaches the final step of the supply chain (say, a store), and the process of transfer stops. Users denoted as stores cannot create or transfer shipments. They can only accept or reject them.

- All products at a store can be tracked by consumers through the mobile application. Consumers can go to a particular store on the application and choose a particular product. The entire location history, as well as the point of origin, will be viewable.

- Company management can use the web interface to view each and every shipment as well as track their geolocation and ownership history. The web interface also provides linkage to OpenSea (a popular NFT marketplace) for each token. The transaction history and metadata of the token can be viewed there as well.

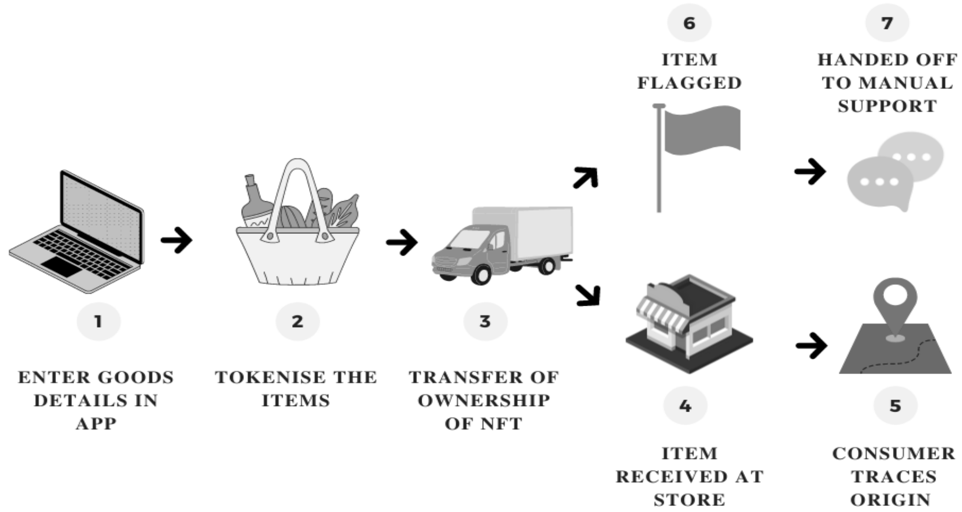

Fig 4. An overview of the workflow of the platform

3.2. Technology Implementation & Feasibility

This section provides an overview of the technology stack utilized to create the system architecture (Fig. 5) of the platform as well as presents justifications for engineering choices.

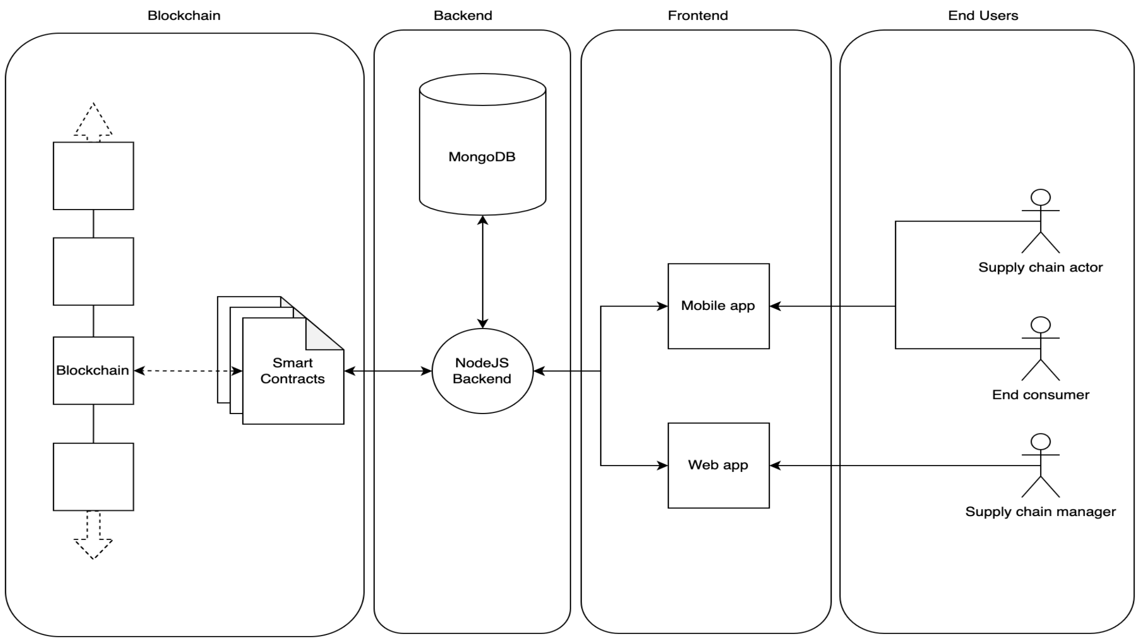

Fig 5. A high-level overview of the system architecture

3.2.1. Blockchain: Ethereum, Polygon and Solidity

After extensive research, Ethereum was chosen as the L1 blockchain of choice as it supports NFTs, has an extensive developer community and is relatively easy to use. The support of NFTs is important as the platform makes heavy use of them. Data (such as quantity and name) about various goods will be tokenized at the first point of contact in the supply chain and the NFTs will be stored on the blockchain. Homogenous goods can be grouped together to be tokenized thereby saving on transaction costs.

However, as discussed in section 2.1.2, the Ethereum main net, being an L1 network suffers from low transactional throughput. In light of the high-frequency nature of the transactions desired by the project, Polygon, an L2 network built over Ethereum was chosen to tackle throughput and transaction cost issues. Polygon, due to its underlying architecture, can support a much higher TPS of 10,000 when compared to the 12-15 achieved by the Ethereum main net [25]. Moreover, transactions on the Polygon network also cost less, averaging around $0.01 per transaction [26]. Historically, the highest transaction cost on the Polygon network has been $15.86 [26].

Solidity has been used to create the contract that executes to mint and transfer NFTs. It is an Object-Oriented Programming (OOP) style language provided by Ethereum which makes the creation of NFTs simple and fast. The backend of our applications will interface with the blockchain via the contract and provide APIs to the frontend.

3.2.1.1. Tokenization of goods

Each shipment is tokenized and represented as an NFT on the blockchain. The tokenization process is achieved through a smart contract, which is a program stored on the blockchain that executes when predetermined conditions are met [27]. The Hardhat development environment was used to compile and test the solidity-based smart contract. Alchemy, a microservice that provides APIs to access L2 networks built on top of Ethereum, was used to deploy the compiled smart contract. The minting process works in two steps (Fig. 6):

1. Upload the metadata supplied to IPFS by using Pinata. Pinata is a pinning service that ensures files uploaded to IPFS are never deleted.

2. Use the mint function of the smart contract along with the CID generated by Pinata in order to mint the NFT. The minted NFT can then be viewed on platforms such as OpenSea.

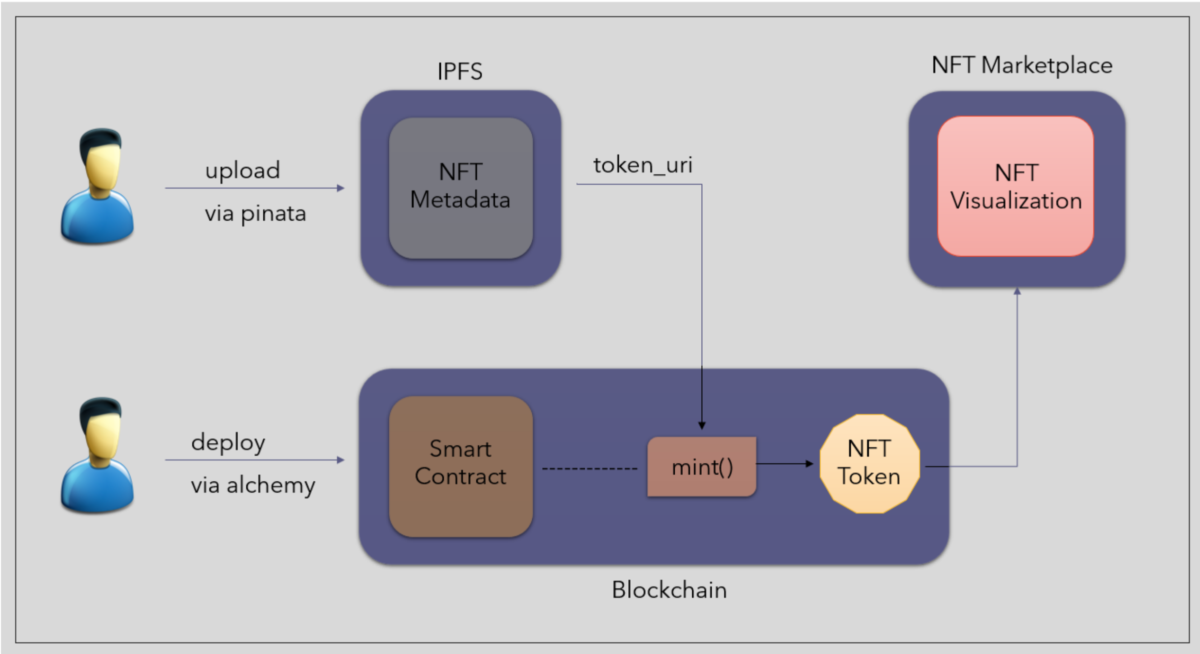

Fig 6. NFT Minting Process

3.2.2. Backend: Express and Node.js

Express, and Node.js have been used to create the backend of the web and mobile apps. Node.js is a cross-platform runtime environment, built on the performant and open-source V8 JavaScript engine, for executing JavaScript on the server-side [28]. Express is a framework built on top of Node.js which is used to develop servers [28]. It introduces additional functionality such as the use of middleware and routing which simplifies the organization and development of servers and RESTful application programming interfaces (APIs) written in Node [28].

Express and Node were primarily chosen because of a few reasons:

- First, they interface seamlessly with various blockchain technologies

- Second, they are highly performant and scalable, being able to handle a large number of concurrent requests through the use of load balancing. This is important as the platform will be used by large transnational supply chains [29]

- Third, being based on JavaScript, they allow for easy development of asynchronous I/O operations through the use of promises and syntactic sugar such as async/await. This is important as operations on the blockchain and database would need to be asynchronous and non-blocking

- Finally, they have an extensive selection of third-party libraries and an active developer community which will make the development process smoother and faster.

3.2.3. Database: MongoDB

Blockchains are ideal for storing minimal amounts of data. This is because, on top of the transaction cost of adding a block to the network, there is a price to be paid per byte of data stored [30]. The proposed platform will need to store large amounts of data and some data might be edited. Thus, there is a clear need for an external storage solution. MongoDB was chosen for storing data off-chain.

MongoDB is a document-oriented NoSQL database. Instead of the traditional model of storing data in tables and creating relationships between them, MongoDB stores the data in the form of documents. This allows for a variety of data to be stored and accessed efficiently in JavaScript Object Notation (JSON) format. JSON is a flexible, powerful, and human-readable way of storing and dealing with data [31]. MongoDB, the backend, and the frontend all communicate using JSON thus making the data representation across these services uniform.

Mongo was chosen mainly due to its native use of the JSON format, its flexible schema representation that can evolve over time, its seamless integration with Node, and the rapid setup and prototyping of data models that it offers.

3.2.4. Frontend: React, React Native

Since users might not be familiar with how blockchains, NFTs, and cryptocurrency work, an easy-to-use frontend with simple flows that abstract away underlying details of the implementation and blockchain is essential to the functioning of the platform. In light of this a simple, but effective UI has been created for both the mobile and the web applications using React and React Native.

React is an open-source front-end javascript library developed by Facebook which is used to make web user interfaces through components. React uses JSX (an extension of JavaScript) to create components that are self-contained and reusable. Components are finally rendered as HTML on the browser [32]. The system of composing smaller components to build larger user interfaces provides developers with a high degree of flexibility [32]. Moreover, React is able to re-render pages with new data without making the browser refresh.

React Native is a javascript frontend framework based on React for creating cross-platform mobile user interfaces. It gives developers the ability to write a single code-base in a syntax they are familiar with (React and JSX) to create user interfaces for both Android and iOS, thus making the development cycle of mobile applications faster, more streamlined, and less esoteric.

The React family has been chosen due to its cross-platform support, reusable components, and great developer community which enables faster development.

Moreover, since the backend is written in JavaScript, and React and Native both use an extension of JavaScript integration is made easier. The mobile and web application will use the APIs provided by the backend to provide users with various functionalities such as tokenization, transfer of ownership, tracing of goods, and more in an easy-to-use interface.

4. Project Development

This section outlines the development timeline of the project that has been followed and elaborates on the implementation of each aspect of the platform i.e the blockchain, the mobile application, and the web application along with the associated backend and database. Finally, it presents the version control system used throughout the project in brief.

4.1. Development Timeline

An iterative approach to research and development was followed in accordance with the following development timeline (Table 1).

Table 1. Development Timeline

Stage	Task	Timeline
Stage 1	Research on the feasibility of different blockchains with respect to throughput and gas fees and further study on various NFT standards	4th - 29th October
Stage 2	Develop a module to tokenize goods based on information	1st - 15th Nov
Stage 3	Develop a pipeline to create and transfer NFTs across a blockchain network	16th Nov - 30th Dec
Stage 4	Develop a web and mobile interface which allows the initialization, transfer and tracking of NFTs	3rd Jan - 14th Feb
Stage 5	Build an interface for consumers using which the origin can be tracked and the authenticity can be verified.	15th Feb - 14th March
Stage 6	Test the entire product in iterative sprints and improve the UI/UX of interfaces.	15th March - 15th April

4.2. Platform Development

The platform comprises a lot of moving parts and technologies that have been integrated seamlessly together in order to create a cohesive system architecture. As described in Fig. 6, the system is vertically divided into four parts: Blockchain, Database, Frontend, and Backend.

4.2.1. Blockchain Development

The smart contract for minting the NFTs was written in solidity. However, for ease of integration with the backend, scripts that could interact with the smart contract to mint and transfer NFTs were written in Node.js. Open Zeppelin provided a base for the smart contract and Alchemy provided functionality for connectivity to Polygon.

4.2.1.1. Project structure

The project has been segregated into a few notable components (Fig. 7): 'contracts' contains Solidity smart contracts. 'scripts' contains a script that utilizes Alchemy in order to deploy compiled Solidity contracts. 'NFT.js' calls the mint function of the deployed contract to create NFTs. 'NFT-Transfer.js' is a script for transferring ownership of a specified NFT.

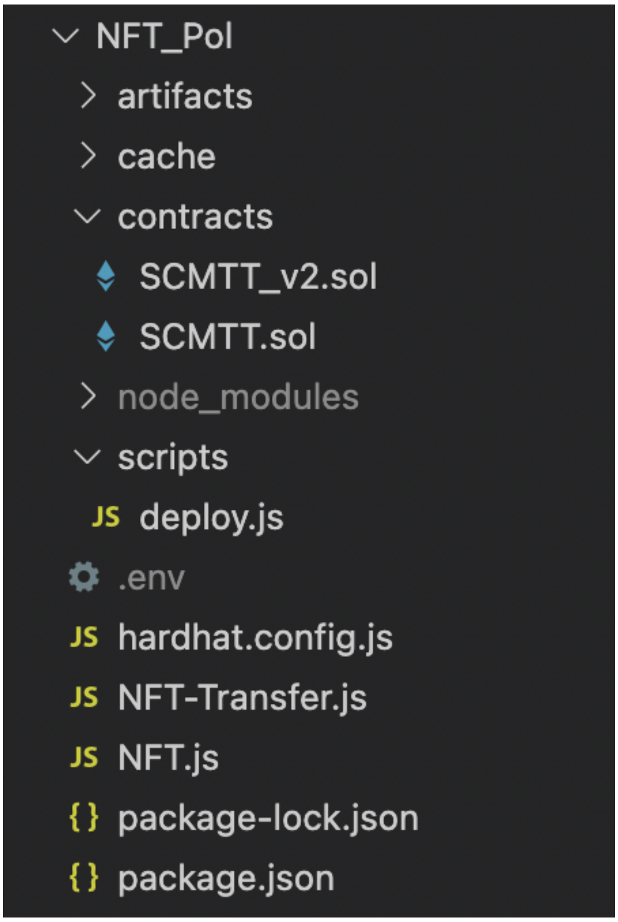

Fig 7. Blockchain development project structure

4.2.1.2. Solidity contract and deployment

As mentioned earlier, the smart contract has been written using Solidity. A snippet of the code is shown in Fig. 8.

NFT_Pol > contracts > SCMTT_v2.sol

```
// SPDX-License-Identifier: MIT

pragma solidity >=0.8.0 <0.9.0;
import '@openzeppelin/contracts/token/ERC721/ERC721.sol';
import '@openzeppelin/contracts/utils/Counters.sol';
import '@openzeppelin/contracts/access/Ownable.sol';

contract SCMTT_v2 is ERC721, Ownable {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIds;
    using Strings for uint256;
    mapping(uint256 => string) private _tokenURIs;
    // Base URI
    string private _baseURIextended;
    constructor() ERC721('SCMTT_v2', 'SCMTT_v2') {}
    function setBaseURI(string memory baseURI_) external onlyOwner {
    _baseURIextended = baseURI_;
    }
    function _setTokenURI(uint256 tokenId, string memory _tokenURI)
    internal
    virtual
    {
    require(
    _exists(tokenId),
    'ERC721Metadata: URI set of nonexistent token'
    );
    _tokenURIs[tokenId] = _tokenURI;
    }
    function _baseURI() internal view virtual override returns (string memory) {
    return _baseURIextended;
    }
```

Fig 8. Smart contract code

After compilation, the smart contract was deployed to the Polygon-Mumbai testnet using APIs provided by Alchemy. Analytics about each API call can be viewed on the Alchemy dashboard (Fig. 9). On deployment, a transaction hash is received.

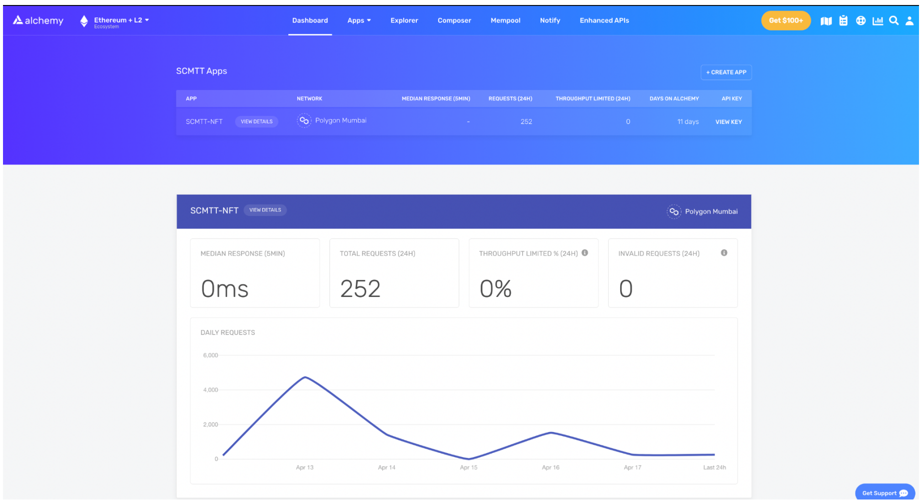

Fig 9. Alchemy dashboard showing analytics

4.2.1.3. NFT Minting

The previously deployed smart contract can be used to mint NFTs. Prior to minting, a uniform format for the metadata needed to be decided for use on the platform. The metadata for an NFT usually contains two fixed fields of name and description, along with an optional list of additional fields called attributes. For the purposes of the platform, each token is treated as a shipment containing multiple products. The attributes list contains 'origin' as a fixed field, which is the geolocation origin of the shipment. Along with 'origin', the attributes list also contains names of products and their quantities contained within the shipment (Fig. 10).

```
const metadata = {
  name: "Shipment",
  description:
    "NFT representing a shipment, owner of the NFT claims ownership to the physical goods",
  attributes: [
    { trait_type: "Origin", value: "35 Produce Drive, Melbourne Market, Australia" },
    { trait_type: "Aussie Raspberries", value: "60" },
    { trait_type: "Australian Mandarins", value: "40" },
  ],
};
```

Fig 10. Example of metadata for the platform

After the metadata has been received from the frontend, APIs on the backend pass this data to the minting function. As discussed in section 3.2.1.1, the minting process happens in two steps. The received metadata is first uploaded to IPFS via Pinata, the code for which is shown in Fig. 11.

```javascript
const authResponse = await axios.get("https://api.pinata.cloud/data/testAuthentication", {
        headers: {
            pinata_api_key: PINATA_API_KEY,
            pinata_secret_api_key: PINATA_SECRET_KEY,
        },
});

console.log(authResponse);

const data = new FormData();

 const pinataJSONBody = {
       pinataContent: metadata
 };
 const jsonResponse = await axios.post("https://api.pinata.cloud/pinning/pinJSONToIPFS", pinataJSONBody, {
     headers: {
         'Content-Type': `application/json`,
          pinata_api_key: PINATA_API_KEY,
          pinata_secret_api_key: PINATA_SECRET_KEY,
     },
  });
  const { data: jsonData = {} } = jsonResponse;
  const { IpfsHash } = jsonData;
  const tokenURI = `https://gateway.pinata.cloud/ipfs/${IpfsHash}`;

  console.log(IpfsHash);
  console.log(tokenURI);
```

Fig 11. Script to upload metadata to IPFS using Pinata

Upon being successfully uploaded, an IPFS CID pointing to the metadata is received. This can be used to view the data (Fig. 12).

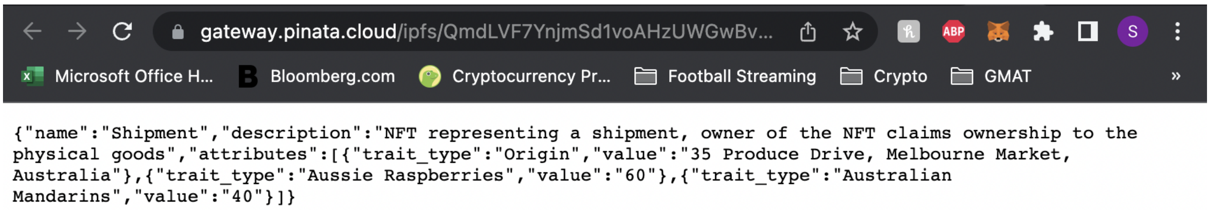

Fig 12. Metadata uploaded to IPFS

After uploading the metadata, the minting process can be initiated. The minting process requires the minter's wallet details as well as the URL (obtained using the base gateway URL of Pinata along with the CID received when the metadata was uploaded) which points to the metadata uploaded to IPFS. After the minting process is successful, a transaction hash and a token ID are received. The tokenized shipment is uploaded to the database and is now available to be transferred by the minter using the front end.

4.2.1.4. NFT Transfer

The transfer process is initiated when a shipment is accepted on the receiver's side on the mobile application. The NFT transfer process is straightforward. It requires the token ID of the NFT to be transferred along with the sender and the receiver's wallet details. After the transfer is successful, a transaction hash is received. The receiver becomes the new owner of the NFT and hence, the shipment.

4.2.2. Mobile Application Frontend Development

The mobile application (Fig. 13) has been developed to serve a dual user base:

1. Intermediate Supply chain actors will use the application mainly for the creation, transfer, and acceptance of tokenized shipments. The actors at the ends of supply chains (stores) will only be able to accept or reject shipments and will not be allowed to create or transfer them.
2. Consumers will log in as guests to the system. They will be able to search for stores and trace the origin and location history of each of the items within a store so as to confirm the meeting of ethical sourcing standards.

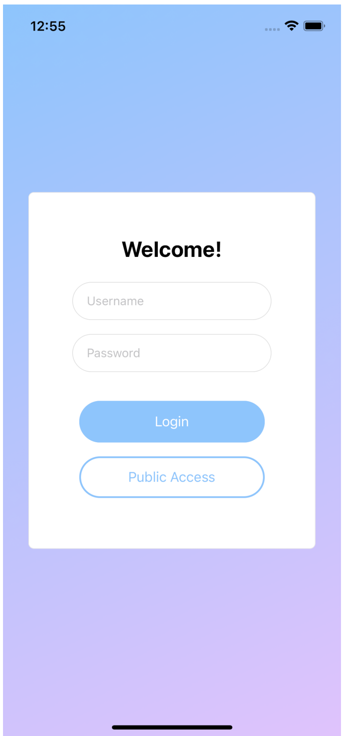

Fig 13. Mobile application login page

4.2.2.1. Project Structure

React Native has been used to create the frontend. Redux, which is a state management library, has been used to keep a global state which contains useful information such as current username and login state. The code is highly modular (Fig. 14) and 'App.js' forms the entry point. Individual elements in the components folder are composed to create screens. The navigation folder contains the structure of the navigation stack.

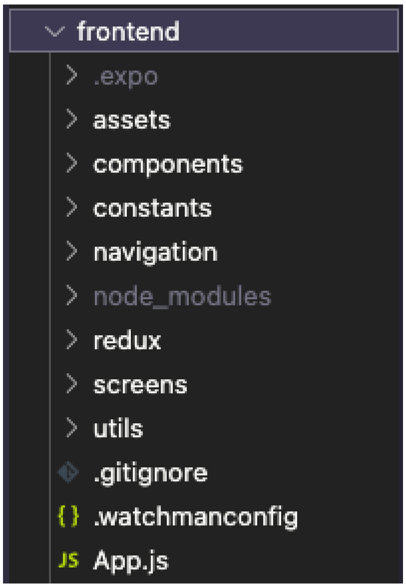

Fig 14. Frontend project structure

4.2.2.2. Screens: Supply Chain Actors

In total there are six major screens and five functions that can be performed using the application:

- **Profile Screen:** As mentioned before, supply chain actors consist of both intermediate actors as well as end stores. Both of them will have different functions available for use and this is reflected by the profile screen of each. Intermediate actors can perform all five functions: creation, transfer, acceptance/rejection, view shipments in-hand, and view shipments sent (Fig. 15). End stores on the other hand can only accept/reject and view shipments in-hand (Fig. 15). A sidebar menu is also provided for ease of navigation (Fig. 15).

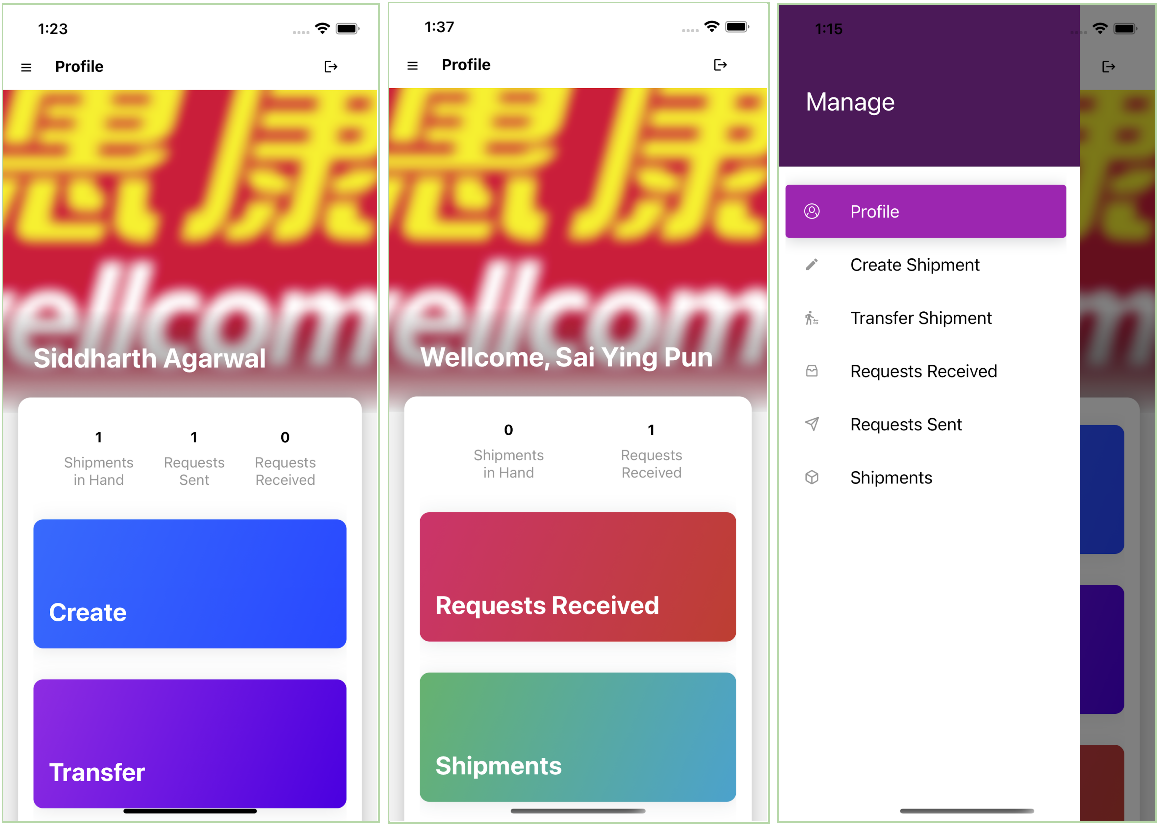

Fig 15. Profile screen for intermediate actors (left); Profile screen for end stores (middle); Menubar (right)

- **Create Screen:** This screen allows intermediate actors to create tokenized shipments containing one or more products (which have been added by the administrator through the web application) in varying quantities. Users select products to be added using the searchable dropdown and enter their quantities. Products are added repeatedly to a cart until the user is satisfied. Users can also choose to remove one or more items from the cart or clear it completely (Fig. 16). When done with the addition of products, users can press the confirm button to initiate the creation process. The application asks for permission to access the user's current geolocation data (Fig. 16). This data along with the cart data constitutes the metadata for the NFT being minted. Upon completion of the minting process, the user is notified (Fig. 17).

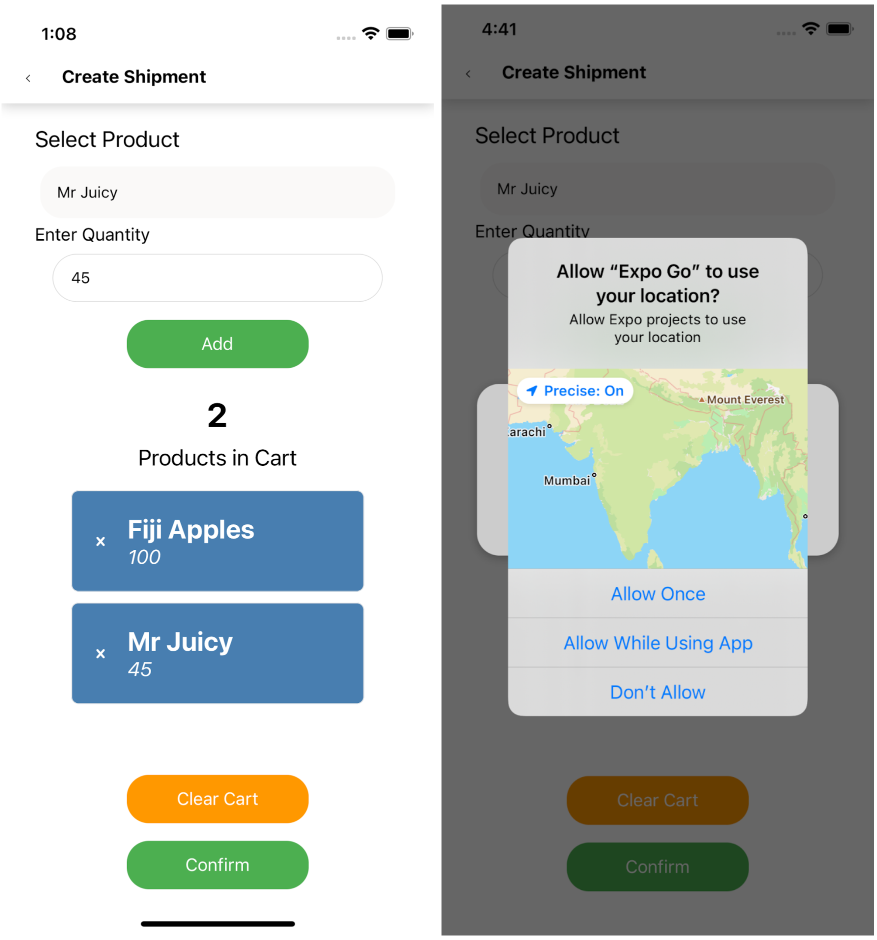

Fig 16. Create screen (left); Geolocation access permission (right)

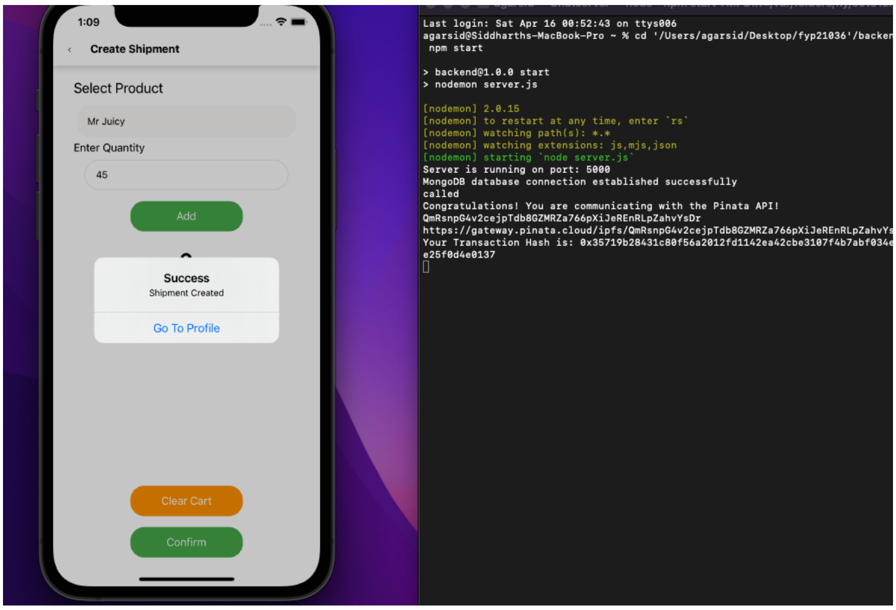

Fig 17. Notification of successful tokenization of shipment (left); NFT minting process in the backend (right)

- **Transfer Screen:** Following the creation of a tokenized shipment, the intermediate actor can then transfer the ownership of the token to the next actor in the supply chain through the transfer screen (Fig. 18). Here, users can select the id of the shipment to be transferred and the person to be transferred to. The details of the shipment selected are also displayed for confirmation purposes.

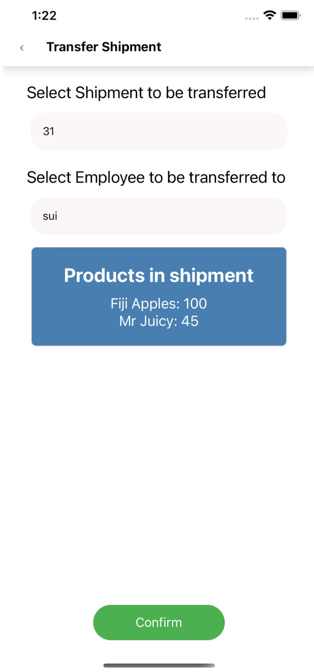

Fig 18. Transfer Screen

Upon pressing the confirm button, the transfer is initiated. Note that the transfer has not yet been done. For this, the receiving user needs to accept/reject the shipment. The shipment is instead put in the requests sent screen of the sender and the requests received screen of the receiver (Fig. 19).

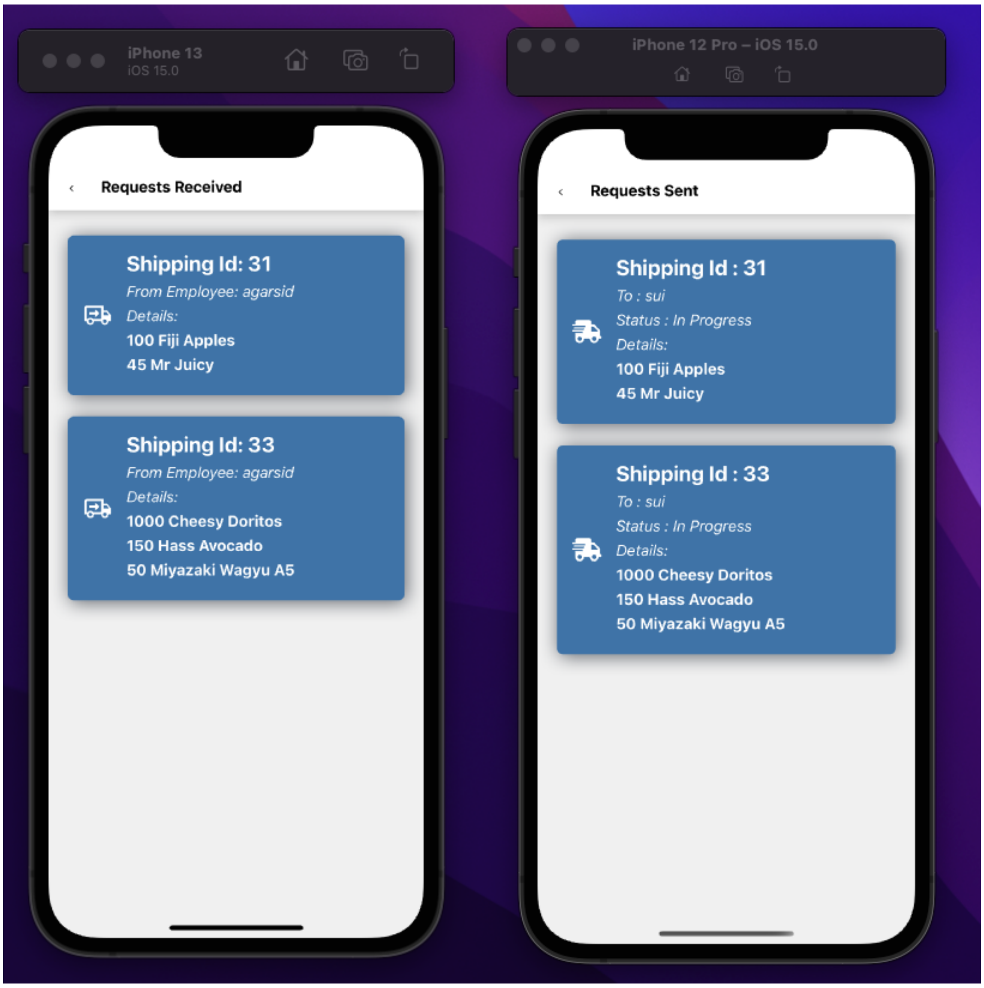

Fig 19. Requests received screen (left); Requests sent screen (right)

- **Requests Received screen:** Both intermediate actors and end-stores can use this screen to view and interact with all the shipments they have received. Details about the transferred shipment such as sender and geolocation origin are shown (Fig. 19). When interacting with a shipment, users are presented with a simple modal using which they can accept or reject a shipment after examining all the data displayed (Fig. 20). If a shipment is rejected, it is removed from the receiver's requests received and stays in the sender's requests sent. Upon acceptance of a shipment, the NFT transfer process starts, and the ownership of the NFT (and thus the tokenized shipment) shifts to the receiver.

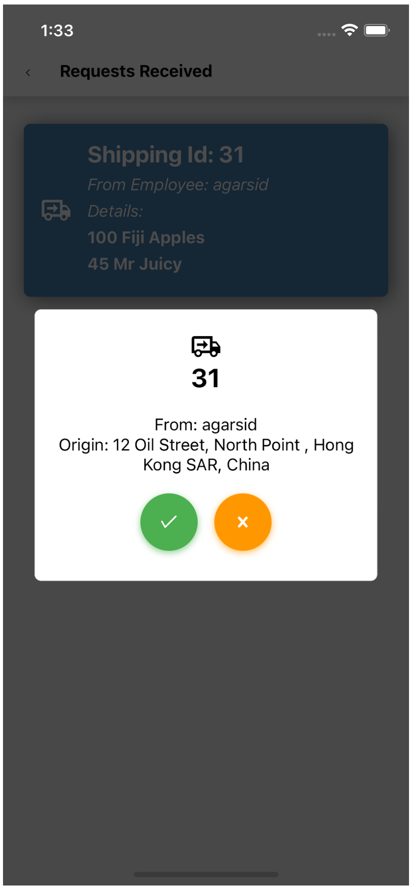

Fig 20. Requests received modal for interacting with a shipment

- **Requests Sent screen:** An intermediate actor can use this page to view a list of all the shipments they have transferred (Fig. 19). It contains detailed information about each shipment as well as the 'status' of the transfer. Transferred shipments that are yet to be accepted/rejected are marked as "In Progress". Shipments that have been accepted no longer belong to the current user and hence aren't displayed in their requests sent screen. Shipments that have been rejected are marked as "rejected" (Fig. 21). These shipments remain on this screen till the resolution process described in section 4.2.3.2 is completed.

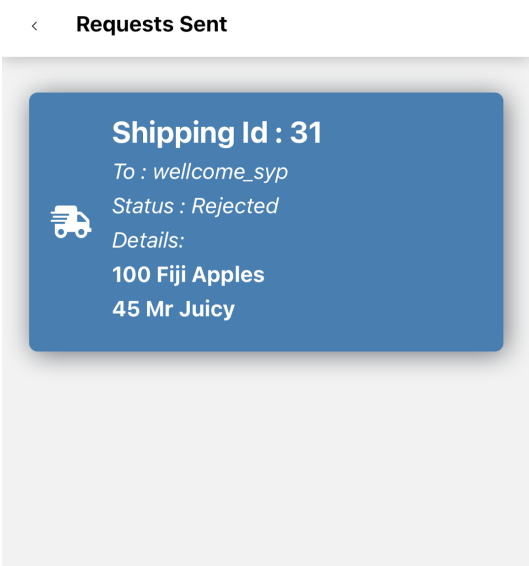

Fig 21. Shipment marked with "rejected" status

- **Shipments screen:** Intermediate actors and end stores can use this screen to view details about shipments they have on hand. They need to select the id of the shipment they want to view and all its details are displayed (Fig. 22).

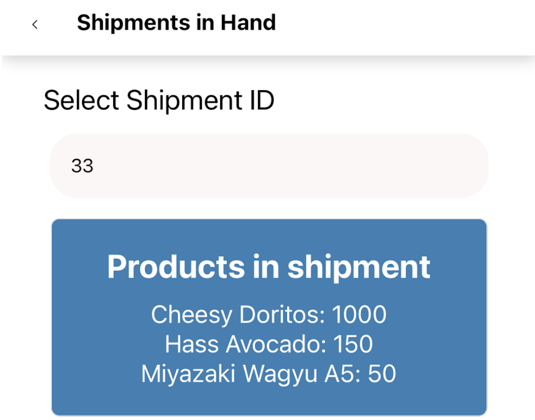

Fig 22. Shipments screen

4.2.2.3. Screens: End Consumers

End consumers form an important part of supply chains and our mobile application caters to them as well by providing highly transparent location tracing for the items they buy:

- **Stores screen (home screen):** Upon logging in as guests, consumers are redirected to the stores screen. Here, they can search for and select stores based on their location and name (Fig. 23).

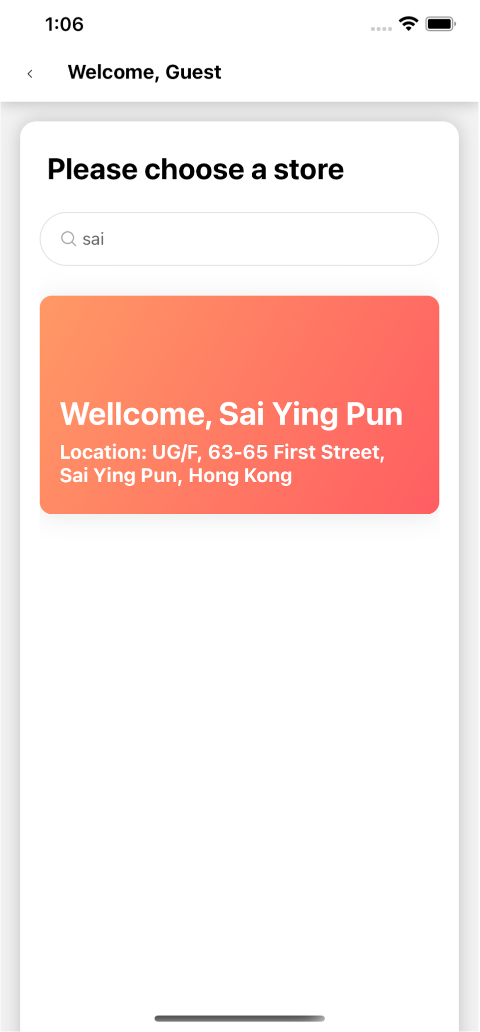

Fig 23. Stores screen for consumers

- **Products screen:** After selecting a store, consumers are directed to a products screen (Fig. 24) that displays all the products in the chosen store. Herein they can search for and choose the product whose geolocation history they are interested in learning more about.

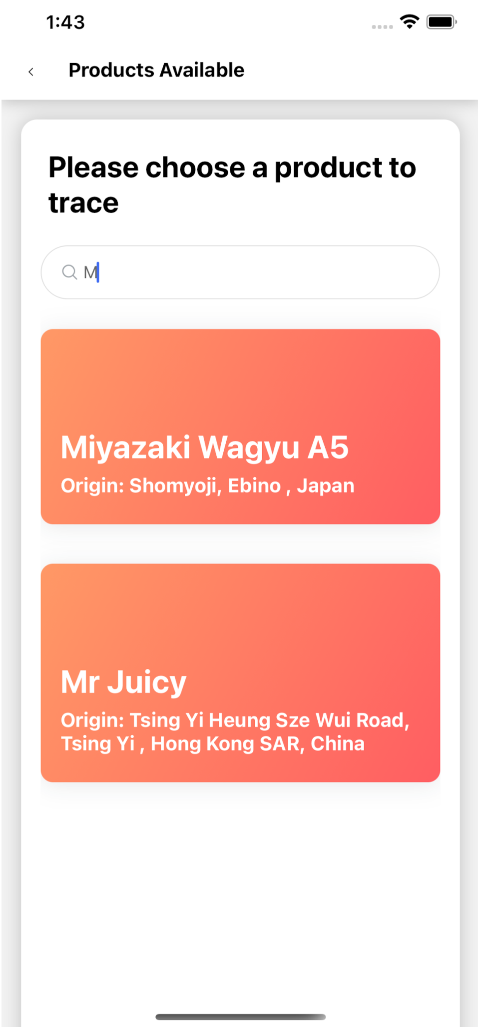

Fig 24. Products screen for a particular store

- **Product Tracing screen:** Upon selection of the desired product, consumers are presented with the product tracing screen. It's a comprehensive history of a product's geolocation and helps consumers identify products that follow ethical sourcing practices by knowing about their origin and their shipping history. The page presents the users with a map that shows all the locations the item has been through the use of pins. Each pin can be clicked to show the name of the location (Fig. 25). Users are also presented with a timeline of the item's geolocation history (Fig. 25). They can use this to glean more information at once than is possible by using the map. They can also click on each individual location on the timeline and the map pans to that location.

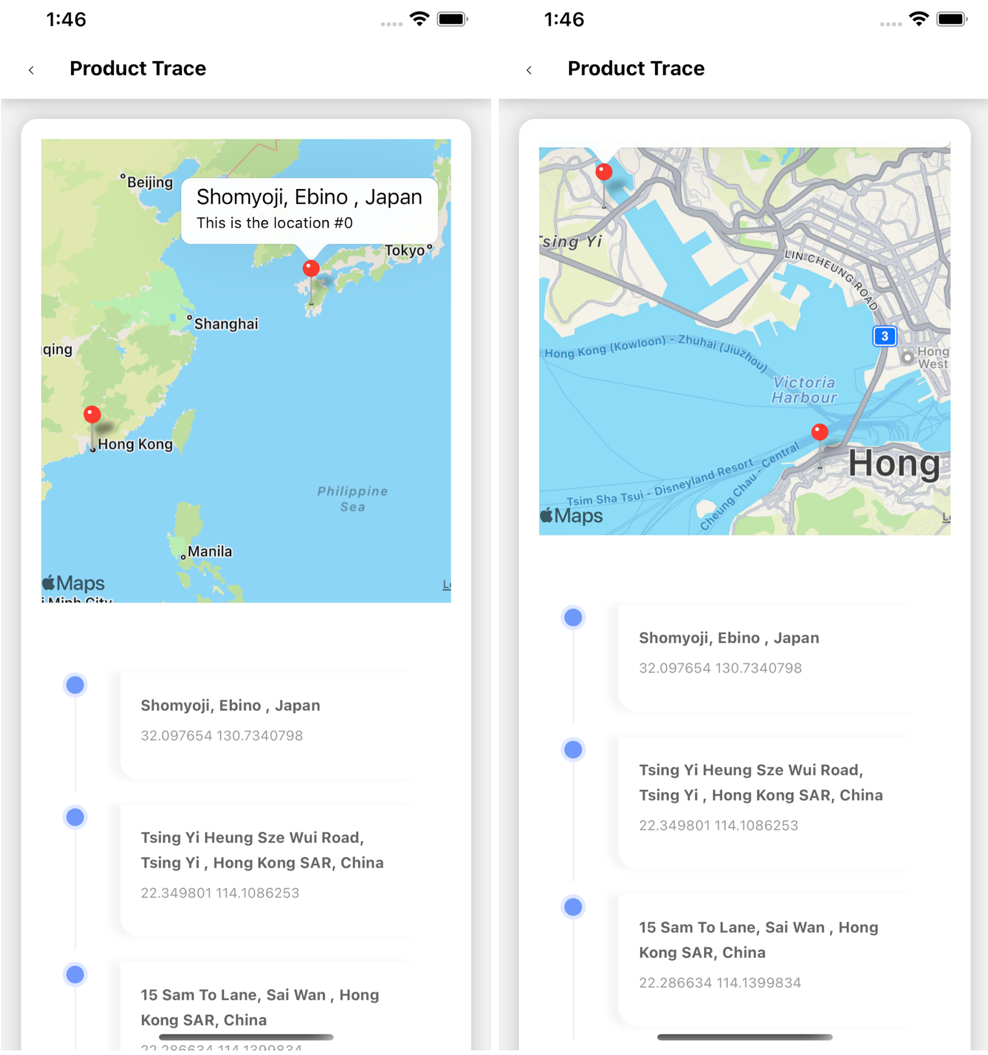

Fig 25. Product tracing screen presented to the consumer. Pin information on the map (left); Timeline information (right)

4.2.3. Web Application Frontend Development

The Web application (Fig. 26) is intended to be used by company managers. It provides managers with several administrative functions in an easy-to-use and intuitive interface.

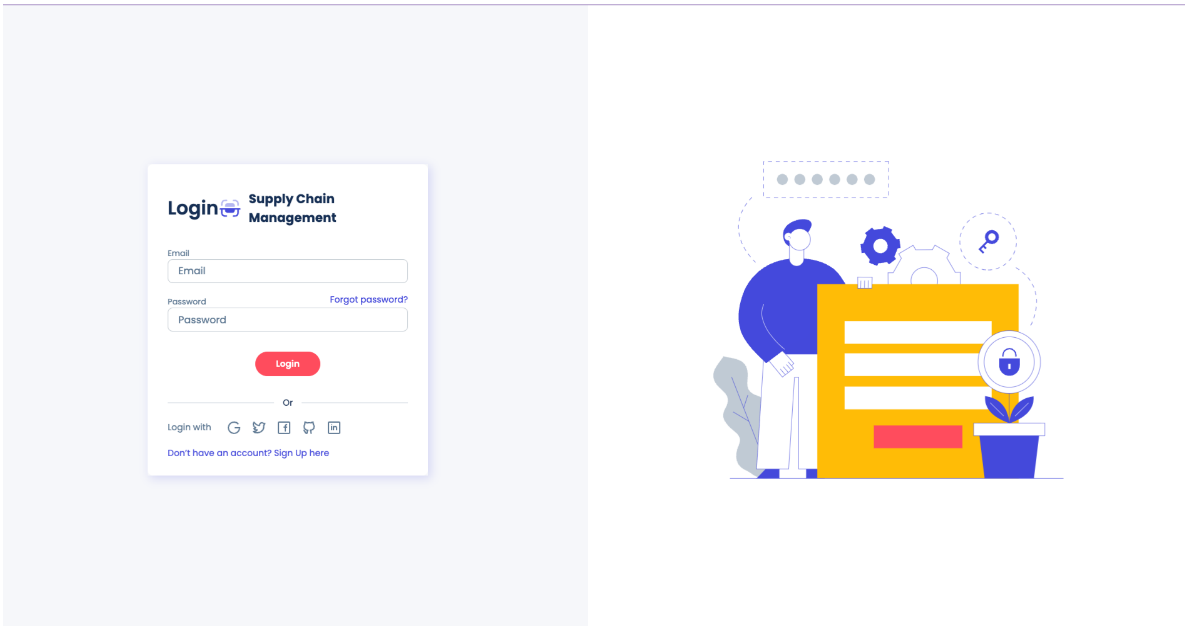

Fig 26. Web Application Login

4.2.3.1. Project Structure

The web application user interface has been created using React. The project structure (Fig. 27) for the web application code has been kept simple and modular in order to streamline collaborative development. The 'components' folder contains various reusable components to be used throughout the application such as the sidebar or the header. The 'pages' folder consists of the different pages in the application such as the user management page. 'App.js' serves as the entry point for the application.

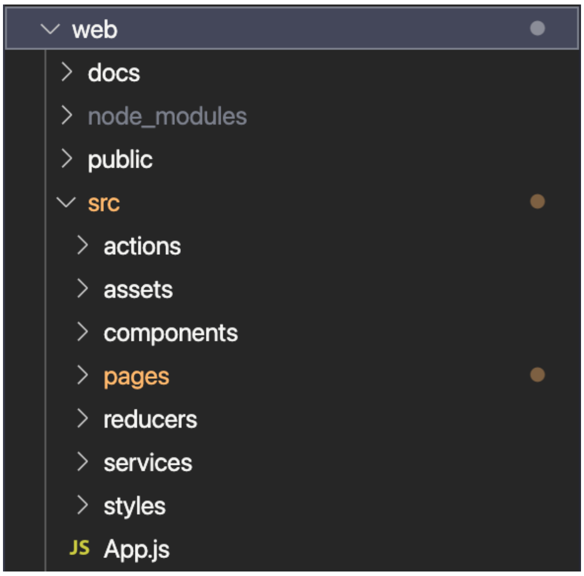

Fig 27. Folder structure for the web application frontend

4.2.3.2. Major Pages

The web application consists of three major pages each corresponding to an administrative function to be used by company managers:

- **User Management Page (Home Page):** This is the home page users are redirected to after logging into the application. It is important not only for the initial setup of the system but also for its continued use. Company managers utilize this page to add new users (supply chain actors and stores) and to manage existing users. Company managers can view details about existing users (Fig. 28) and edit basic information about them such as their names.

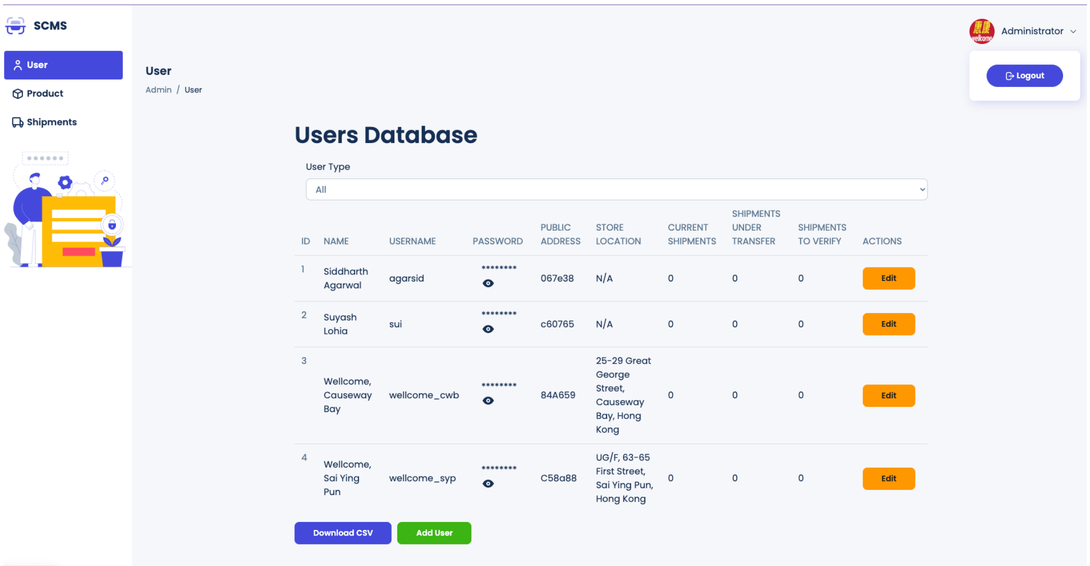

Fig 28. User Management Page / Homepage

They can also filter the view of the existing users according to whether the existing users are stores or supply chain actors (Fig. 29).

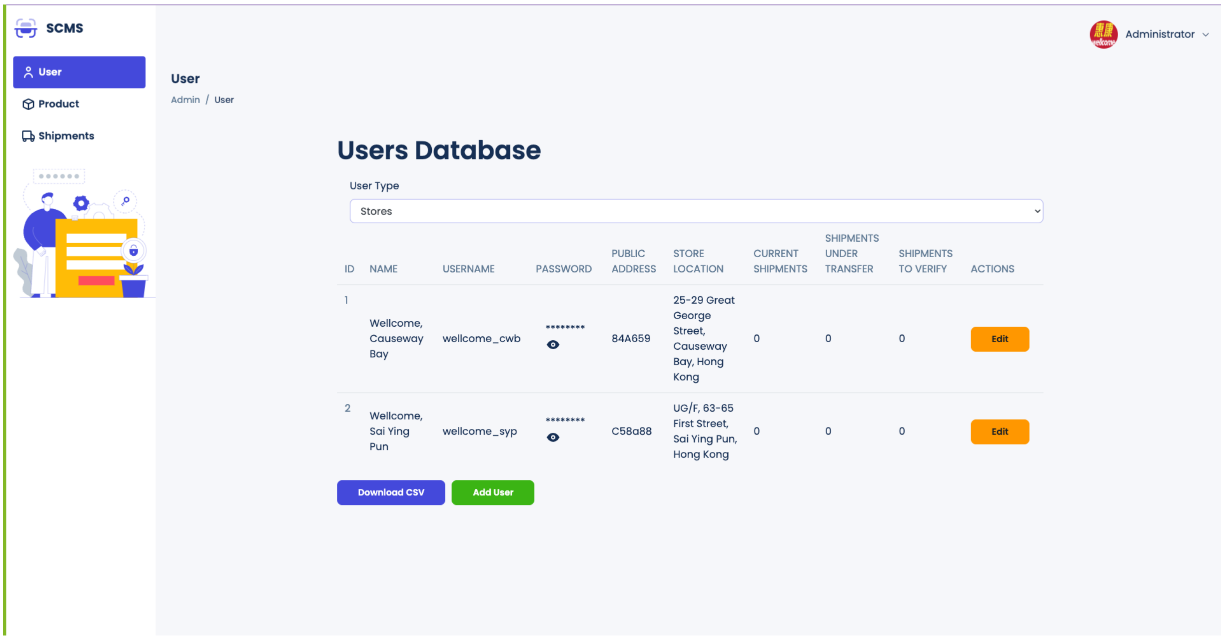

Fig 29. Filtered View of only Stores

New users are added through a modal to give them access credentials for the mobile application and to link their accounts to a company-managed cryptocurrency wallet which will hold all the tokenized shipments they own (Fig. 30).

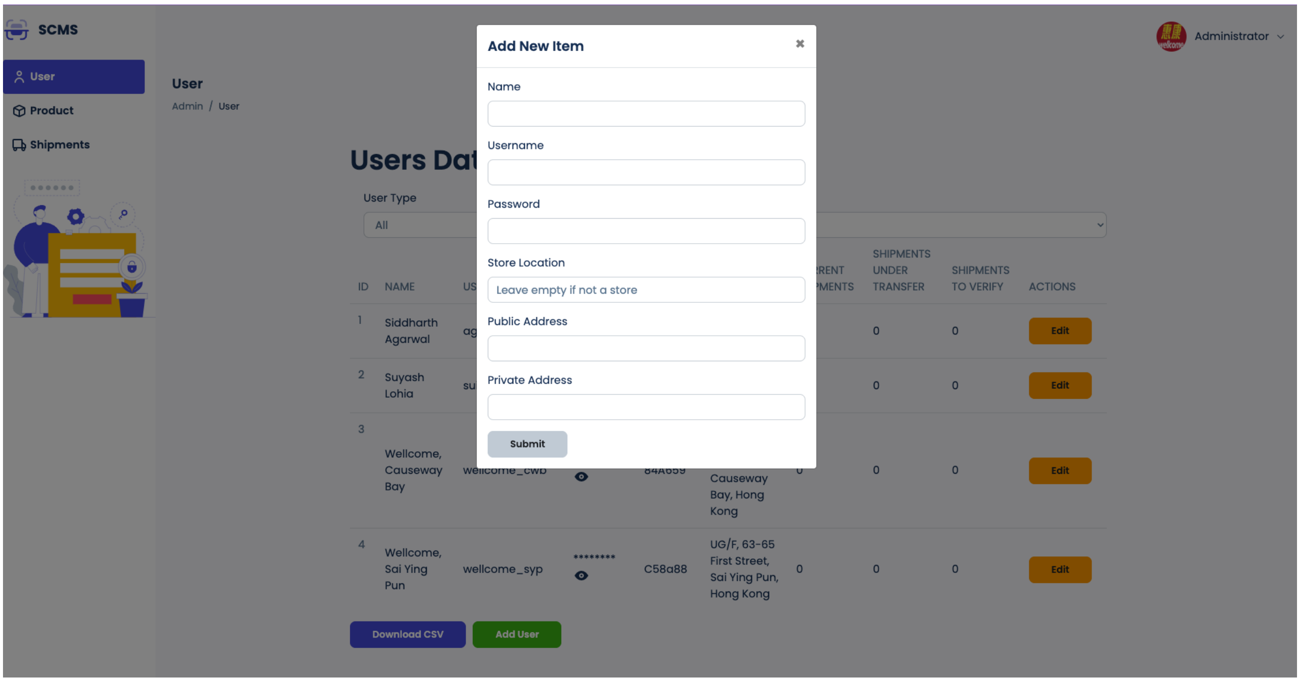

Fig 30. Add a new user

- **Products Management Page:** Each tokenized shipment in the platform consists of one or more products selected by supply chain actors on the mobile application.

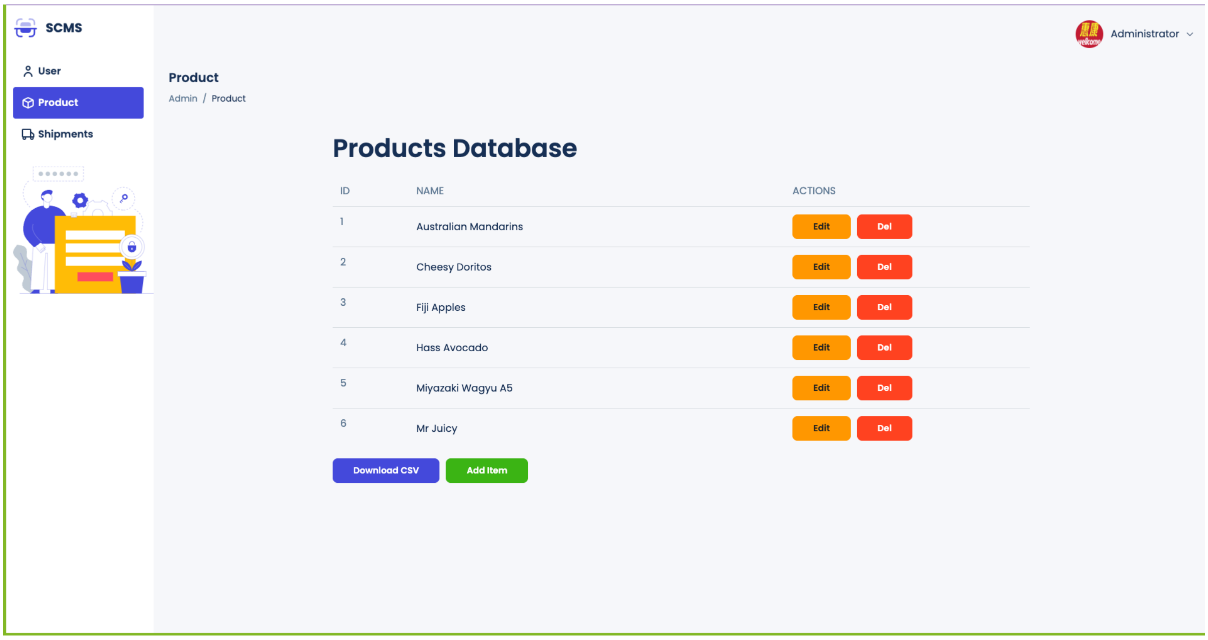

Fig 31. Products Management Page

The products management page (Fig. 31) gives company managers the ability to add new products to the system so that these products can be made into shipments. It also allows managers to edit or delete existing products in the system.

- **Shipments Management Page:** The last, but the most important page for company managers, is the shipments management page (Fig. 32). This page gives them the ability to view details about all the shipments that have been created, check whether a shipment has an issue, and then resolve the issue within the system.

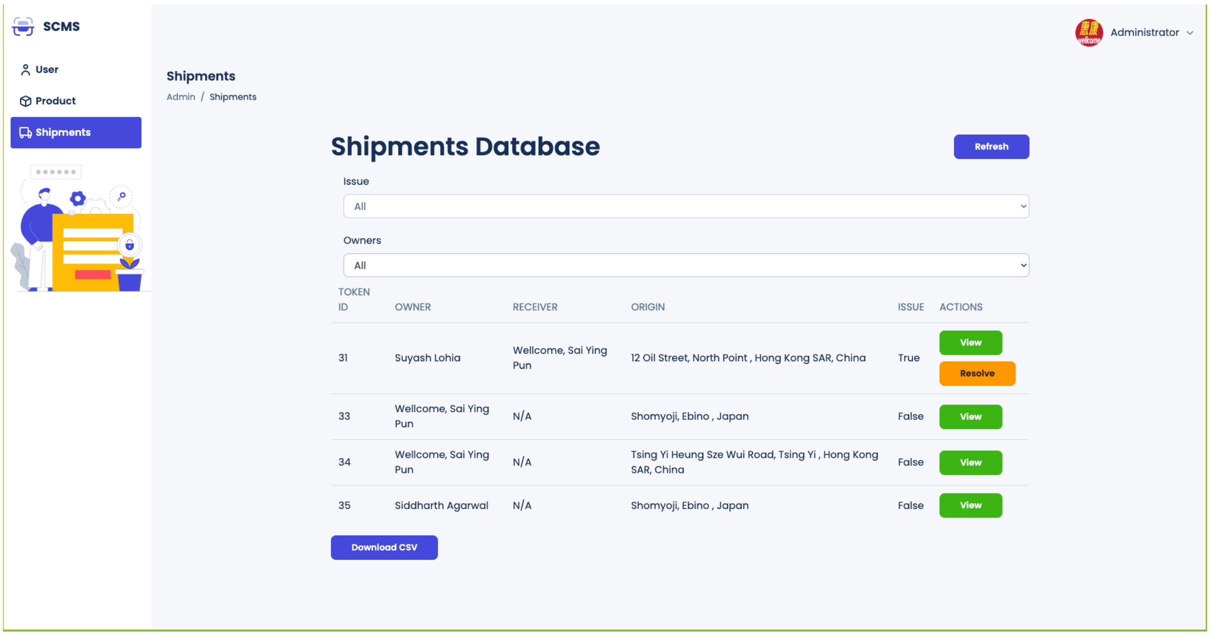

Fig 32. Shipments Management Page

Each shipment can be expanded into a modal to view details about the shipment. Managers can view details about the products in the shipment, trace location history (Fig. 33), and also trace ownership history (Fig. 34). Since the shipment is an NFT on the blockchain, the modal also provides a link to view the token on OpenSea (Fig. 34).

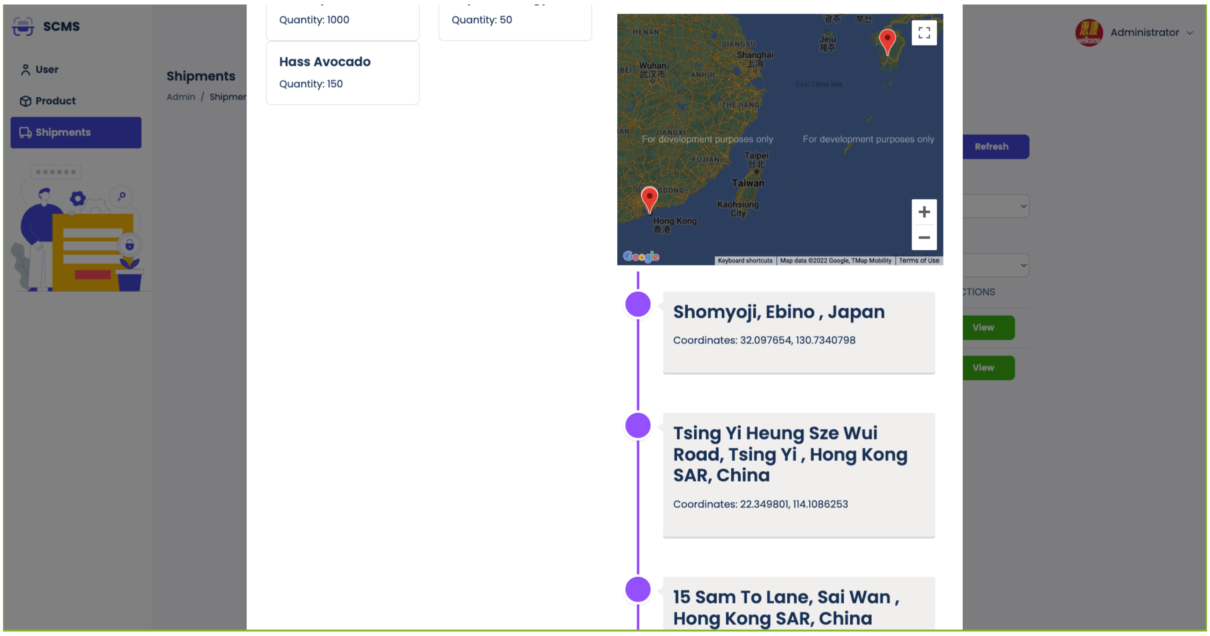

Fig 33. Shipment modal showing products and location history

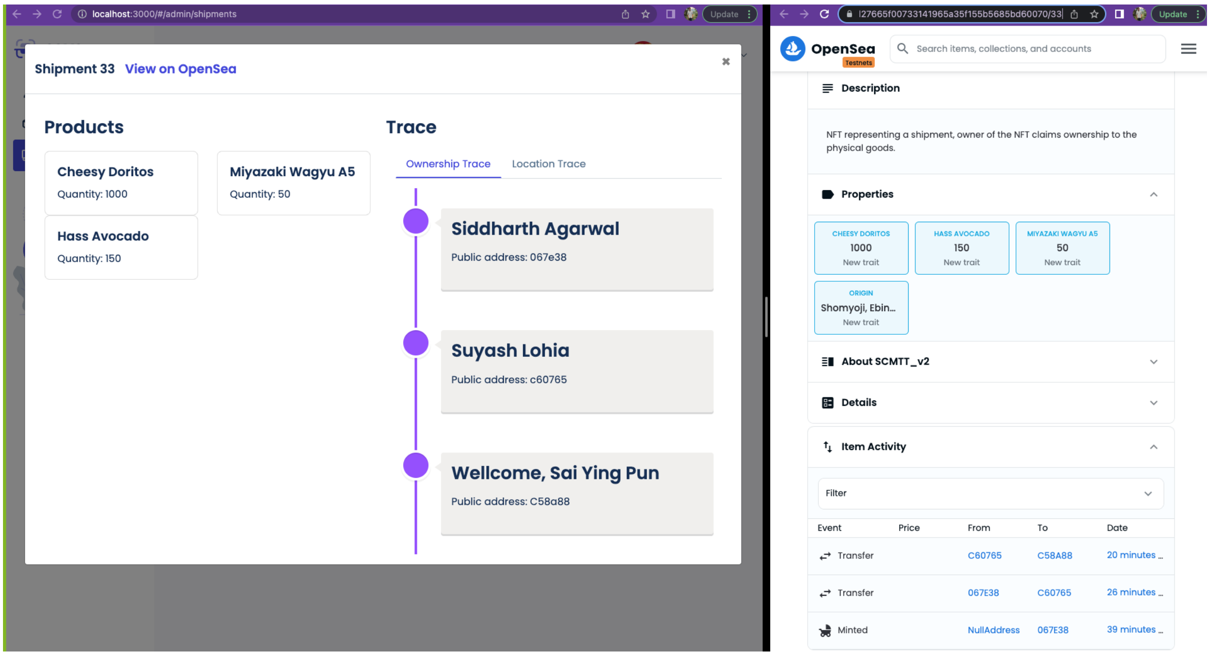

Fig 34. Ownership tracing in the modal (left) and shipment opened in OpenSea (right)

The page also provides filters to view shipments for particular users and view shipments having issues. Managers can resolve issues pertaining to the loss of goods within the platform after inspecting the faulty shipments. They can change the quantities of the products in the shipment (Fig. 35). Upon resolution, a new token is minted for the shipment based on the previous token with the updated product quantities. The old faulty shipment is removed from the database. Data such as origin is preserved (Fig. 35).

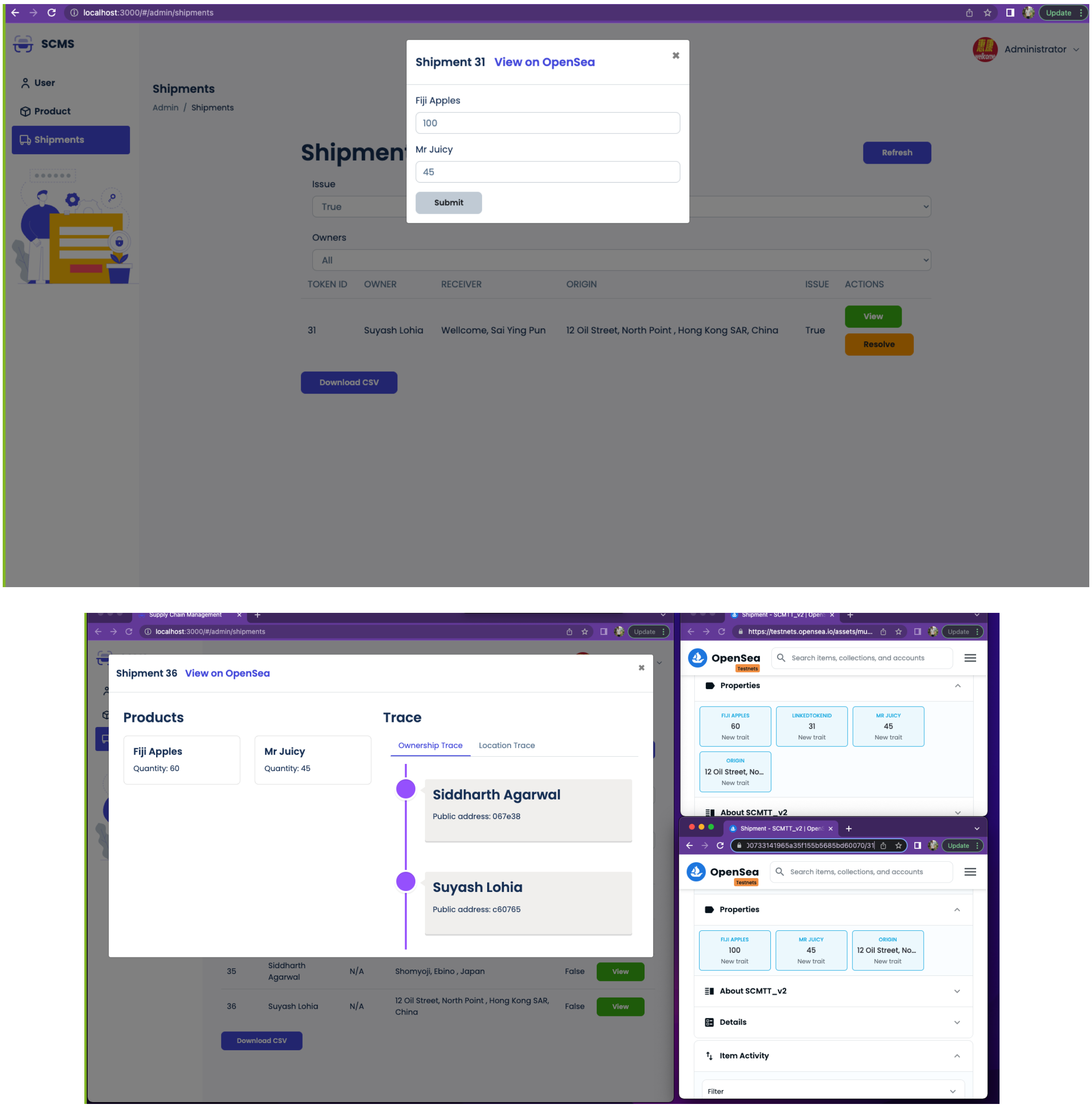

Fig 35. Shipment resolution modal (top) and new tokenized shipment with correct information based on the previous shipment (bottom)

4.2.4. Backend Development

As mentioned in section 3.2.2, Node.js and Express have been used for backend server-side development. An API that helps the blockchain, frontend, and database interface and communicate has been developed.

4.2.4.1. Project structure

The project structure (Fig. 36) has been separated into 'models' and 'routes'. Models contain the schemas that the MongoDB database will follow and the 'routes' expose the API endpoints. The project structure has been streamlined so as to speed up the development process. The entry point for the Express server is ‘app.js’

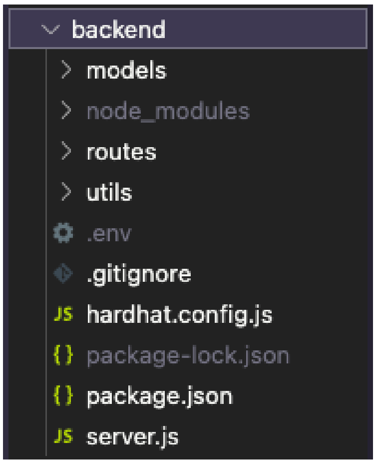

Fig 36. Folder structure of the backend

4.2.4.2. RESTful API

Various endpoints of the RESTful API have been exposed through routes so as to facilitate seamless communication between the major components of the platform. The different routes can be separated into three major groups based on the underlying entity they interact with: Table 2 explains the routes that interact with ‘products’; Table 3 explains the routes that interact and facilitate CRUD operations on ‘users’; Table 4 explains the routes that mainly interact with ‘shipments’.

Table 2. Product-related routes

Protocol	Route	Description
`GET`	`/products/`	Retrieve all product objects
`DELETE`	`/products/`	Delete a product object based on the name supplied in the request body
`POST`	`/products/add`	Add a product object using the name supplied in the body
`PUT`	`/:id`	Update a product object using name supplied in the body

Table 3. User-related routes

Protocol	Route	Description
`GET`	`/users/`	Retrieve the usernames of all users
`GET`	`/data-users/`	Retrieve all user objects
`POST`	`/create-user`	Add a user object using the data supplied in the request body
`PUT`	`/update-user/:id`	Update the user object whose id is in the request parameters using the data in the request body
`POST`	`/login/:username/:password`	Login using the credentials supplied in the request parameters
`GET`	`/stores`	Retrieve all user objects that are stores
`GET`	`/:username`	Retrieve a user using the username specified in the request parameters

Table 4. Shipment-related routes

Protocol	Route	Description
POST	/create-shipment/:username	Mints an NFT for the shipment object and on successful creation adds the object to the current shipments of the specified username.
POST	/transfer-shipment/:usernam e	Initiate the transfer process of the shipment from the specified username to the specified user in the request body as a requested shipment
POST	/accept-shipment/:username	The username specified accepts the requested shipment through this route. This initiates a transfer of the NFT from the sender's wallet to the current user's wallet. On successful transfer, the shipment object is also transferred.
POST	/reject-shipment/:username	The username specified rejects the requested shipment. The shipment is removed from the requests received list of the specified user and is kept in the requestsSent list of the sender with a 'Rejected' status
POST	/resolve-shipment/:username	Called by the administrator to resolve an issue in a shipment by editing the quantity of items to account for lost goods. It removes the old shipment from the database, and mints a new NFT with metadata of the new shipment including the old shipment's token
GET	/shipments	Retreive all user objects alongwith their current and sent shipments

4.2.5. Database Development

As mentioned in section 3.2.3, MongoDB is being used as the off-chain data storage solution. Two models have been created for use by the MongoDB database:

1. **The 'products' model** simply stores the name of the products. It is used by the web application when new products are added or existing products are edited. It is also

used by the mobile application to present a list of available products during the creation of tokenized shipments.

2. **The 'users' model** (Fig. 37) stores user data. It has data like username and wallet details. It maintains lists of current shipments that users have in hand, requested shipments that have been sent to the user but are waiting for acceptance/rejection, and sent shipments for which the transfer process has been initiated by the user but is not complete yet. It is an important model as its use is ubiquitous through the platform.


```
_id: ObjectId("6255bffb9c7488134f8a956a")
username: "agarsid"
password: "D7ZfDarG6Qhro+x5b+VbAw=="
name: "Siddharth Agarwal"
storeLocation: ""
> wallet: Object
v currentShipments: Array
  v 0: Object
      tokenId: "35"
      origin: "Shomyoji, Ebino , Japan"
    v users: Array
      v 0: Object
          name: "Siddharth Agarwal"
          public_address: "0x067e38fC747D789b97c6C2f3dF94160F065CA9f2"
          _id: ObjectId("625b209076f32c0aeeef9758")
    v locations: Array
      v 0: Object
          name: "Shomyoji, Ebino , Japan"
          latitude: 32.097654
          longitude: 130.7340798
          _id: ObjectId("625b209076f32c0aeeef9759")
    v products: Array
      v 0: Object
          name: "Miyazaki Wagyu A5"
          quantity: 100
          _id: ObjectId("625b209076f32c0aeeef975a")
      _id: ObjectId("625b209076f32c0aeeef9757")
> sentShipments: Array
> reqShipments: Array
```


Fig 37. The user data model

4.2.6. Coding Practices: Git and GitHub

Git was used throughout the development process for version control and Github was used to host and manage the git repository online. Different git branches were created to develop different parts of the platform such as the frontend or blockchain. This ensured code written for one part did not affect the others. After working on the frontend, backend, and blockchain code separately, all of these were integrated together using a separate branch.

5. Discussion

This section discusses the limitations and the challenges faced and proposes some potential solutions for these. It further presents future plans the team has for the platform.

5.1. Limitations

The team has identified some limitations of the platform:

- The mobile application not only requires supply chain actors to own a smartphone but also depends on them having a reliable internet connection at all times. Supply chain actors belong to varied socioeconomic and geographic backgrounds and consequently, not every member of the supply chain might have access to the platform, thus limiting the widespread adoption of the platform in its current state.

- The system that informs about and assists to resolve issues with shipments is also limited. Currently, it assumes that issues arise due to the loss of goods. Practically, however, issues could arise from various other factors such as mishandling by supply chain actors or weather damage. Moreover, upon resolution of loss of goods issues, a new NFT is minted to represent the reduction in quantities of the items within the shipment. This new NFT does not preserve the history of the previous NFT. Currently, the location history is stored on the database and transferred to the new NFT and the old token's ID is added to the metadata of the new one to provide a linkage. However, this system is not robust and does not fully leverage the power of the blockchain. Although no natural solution is currently present, however, with the advent of fractional NFTs in the near future, which will allow for users to own stakes on the NFT instead of the whole token, the team hopes this issue can be tackled.

- Another important limitation is the environmental impact associated with the creation and transfer of NFTs on the Ethereum blockchain. Although the team has tried to mitigate this by using Polygon, a Layer 2 network, the energy consumption for transactions is still higher than popular centralized solutions. Environmental impact is a limitation in terms of platform adoption, but more importantly, it is also a limitation in terms of working towards creating more sustainable business solutions on the blockchain.

5.2. Challenges

The team encountered many challenges while developing the platform. The main challenges and their solutions are discussed below:

- Any system with which a multitude of users, each with varying levels of comfort with technology, interact is prone to human error and the proposed platform is no exception. Human error can occur in various forms, such as tokenization of the wrong items, transferring shipments to the wrong person, or raising an issue for a shipment that has none. The team has tried to tackle and minimize the chances of such scenarios occurring by designing an intuitive and simple user interface that constantly asks for confirmations, and permissions and raises appropriate errors.

- Transaction fees for actions on the blockchain such as minting and transfer of NFTs are relatively high while the throughput of these transactions is low. As the platform is targeted toward a large user base and provides minting and transferring of NFTs as core functionalities, a solution that minimized transaction fees while maximizing the throughput was sought. The team finally settled on Polygon to resolve these issues.

- For consumers, one of the key limitations of the platform is the tracing of homogenous goods having varied origins and supply chains but being sold in the same batch at end stores. Although the mobile application provides consumers the ability to search for particular products within stores and items with different origins have different tokens, the consumer is still unable to discern between the same products coming from varied origins. The team proposes the use of QR codes attached to individual units of shipments as a viable solution.

5.3. Future Plans

While tracing and tracking goods is a good first step towards improving supply chain visibility, there is more to be performed in the space.

- One major short-term goal of the team is the implementation of the QR code system for tracing homogenous goods as mentioned in the previous section. In this system, QR codes, that are linked to the underlying token, are to be attached to each individual unit of a homogenous whole at the point of origin after tokenization. End consumers can then scan these codes in-store and trace the origin of a particular item in a batch of homogenous goods with diverse points of origin.

- Beyond the objectives and the scope of this project, there are various long-term goals the team has in mind. The developed platform can provide analytics and actionable insights to business owners to easily optimize supply chains. An example of such analytics could be ones that help businesses to identify points of disruption within their supply chains.

- Another long-term goal is to standardize and decentralize current ERP solutions for all participants of the supply chain by integrating the proposed platform with them. The end goal is to disrupt the supply chain management industry by offering a Software as a Service (SaaS) platform.

6. Conclusion

With the increasing complexity of global supply chains and a burgeoning reliance on them, supply chain visibility has become a key area of importance in SCM. Increased supply chain visibility provides businesses with various benefits such as increased efficiency in inventory management, management and mitigation of risks, and improved customer support. Increased visibility also benefits end consumers by providing them the ability to trace the origin of goods so as to ensure ethical sourcing practices. Therefore, a system that improves supply chain visibility while maintaining the integrity of data is necessary.

This report presents a platform that leverages blockchain and NFTs in order to improve supply chain visibility. The platform has three main components that work in sync: First, the blockchain on which data about goods is stored in the form of NFTs. Specifically, Polygon, an L2 network on top of the Ethereum main net, has been chosen owing to its support of NFTs, high transactional throughput, and low transaction fees. Second, a web application that is to be used by managers for various management purposes such as entering data about goods that will be tokenized, adding new users to the system, and resolving issues related to the loss of goods within shipments in transit. Third, a mobile application that will be used by various participants in the supply chain to tokenize, transfer ownership of, and raise issues about goods. It will also be used by consumers to trace the origin of goods to ensure ethical sourcing practices. Both the web and the mobile application will be developed using the MERN stack because of its ease of use, extensive third-party library support, active developer community, and cross-platform support.

Currently, one of the major limitations of the platform is its requirement for all users to possess capable smartphones and stable internet connections. In an industry as socioeconomically and geographically varied as supply chain management, this might not always be possible. Another major limitation is the carbon footprint associated with transactions on the blockchain. Although the use of Polygon in place of the Ethereum main net reduces the carbon footprint of the application, the energy consumption is still high. The final major limitation is the current issue resolution system of the platform which assumes loss of goods as the root cause of issues and mints new NFTs upon resolution. Minting new NFTs can be avoided upon the introduction of Fractional NFTs in the future.

There are various ways in which the platform can be extended in the future. Implementation of a QR code system to facilitate the tracing of homogenous goods is a short-term goal. In the long term, analytics and actionable insights that help optimize supply chains can be provided. Integration with existing ERPs so as to standardize them and provide the benefits of decentralization can also be implemented.

7. References

[1] D. Daniel, “What is Supply Chain Management (SCM)?,” SearchERP, 10-Nov-2020. [Online]. Available: https://searcherp.techtarget.com/definition/supply-chain-management-SCM. [Accessed: 30-Nov-2021].

[2] J. Gomez, “Important benefits of effective supply chain management systems,” 6 River Systems, 27-Sep-2021. [Online]. Available: https://6river.com/8-key-benefits-of-effective-supply-chain-management/. [Accessed: 30-Nov-2021].

[3] A. Loudin, “Making the most of the cash-to-cash cycle,” Supply Chain Dive, 12-Mar-2020. [Online]. Available: https://www.supplychaindive.com/news/cash-to-cash-cycle/574034/. [Accessed: 30-Nov-2021].

[4] “Global Supply Chain Management Market (2020 to 2027) - by component, user type, and industry vertical - researchandmarkets.com,” *Business Wire*, 13-Aug-2020. [Online]. Available: https://www.businesswire.com/news/home/20200813005569/en/Global-Supply-Chain-Management-Market-2020-to-2027---by-Component-User-Type-and-Industry-Vertical---ResearchAndMarkets.com. [Accessed: 27-Oct-2021].

[5] B. Cooksey, “Why is supply chain visibility so important?,” C.H. Robinson Blog, 08-Jun-2020. [Online]. Available: https://www.chrobinson.com/blog/why-is-supply-chain-visibility-so-important/#:~:text=Supply%20chain%20visibility%20is%20the,limiting%20disruptions%20and%20risk%20mitigation. [Accessed: 30-Nov-2021].

[6] L. Ross, “The state of Supply Chain Management – Statistics and trends...,” *Invespcro*, 28-May-2020. [Online]. Available: https://www.invespcro.com/blog/supply-chain-management/. [Accessed: 27-Oct-2021].

[7] A. Barker, “Ripe with potential: The many benefits of Fairtrade avocados,” Fairtrade Foundation, 07-Jan-2022. [Online]. Available: https://www.fairtrade.org.uk/media-centre/blog/benefits-of-fairtrade-avocados/. [Accessed: 18-Apr-2022].

[8] OpenText, “OpenText Survey Shows Increase in Demand for Ethically Sourced Goods,” OpenText, 29-Sep-2021. [Online]. Available: https://www.opentext.com/about/press-releases?id=5188A6174DE447FCAC930758694BBCB9#:~:text=The%20research%20also%20highlights%20that,premium%20for%20ethically%20produced%20goods. [Accessed: 18-Apr-2022].

[9] L. Conway, “Blockchain explained,” Investopedia, 04-Nov-2021. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp. [Accessed: 30-Nov-2021].

[10] J. Frankenfield, “Proof of work (POW),” Investopedia, 20-Nov-2021. [Online]. Available: https://www.investopedia.com/terms/p/proof-work.asp. [Accessed: 30-Nov-2021].

[11] A. Hertig, “What is proof-of-work?,” *CoinDesk Latest Headlines RSS*, 16-Dec-2020. [Online]. Available: https://www.coindesk.com/learn/2020/12/16/what-is-proof-of-work/#:~:text=A%20new%20block%20is%20accepted,is%20with%20expensive%2C%20specialized%20computers [Accessed: 27-Oct-2021].

[12] Y. Yun, “Proof of work: What is it, and how does it figure into Bitcoin halving?,” Forkast, 03-Aug-2021. [Online]. Available: https://forkast.news/proof-of-work-what-is-it-bitcoin-halving/. [Accessed: 30-Nov-2021].

[13] L. Conway, “Proof-of-work vs. proof-of-stake: Which is better?,” Blockworks, 23-Feb-2022. [Online]. Available: https://blockworks.co/proof-of-work-vs-proof-of-stake-whats-the-difference/#:~:text=Quick%20facts%3A-,Proof%2Dof%2Dwork%20and%20proof%2Dof%2Dstake%20are,of%20transactions%20to%20the%20chain. [Accessed: 18-Apr-2022].

[14] PYMNTS, “Ethereum 2.0 project enters final stretch, promising massive scalability by Summer,” Yahoo! Finance, 17-Mar-2022. [Online]. Available: https://finance.yahoo.com/news/ethereum-2-0-project-enters-174440315.html#:~:text=The%20Ethereum%202.0%20project%20will,TPS%20limit%20to%20100%2C000%20TPS%2C. [Accessed: 18-Apr-2022].

[15] Binance Academy, “What is Layer 1 in Blockchain?,” Binance Academy, 22-Feb-2022. [Online]. Available: https://academy.binance.com/en/articles/what-is-layer-1-in-blockchain. [Accessed: 18-Apr-2022].

[16] A. White-Gomez, “What are Layer 2 solutions and why are they important?,” ONE37pm, 08-Mar-2022. [Online]. Available: https://www.one37pm.com/nft/what-are-layer-2-solutions-and-why-are-they-important. [Accessed: 18-Apr-2022].

[17] S. Srinivasan, “The NFT boom : What are non-fungible tokens ?,” *The Daily Marketing*, 02-Mar-2021. [Online]. Available: https://www.thedailymarketing.in/post/the-nft-boom-what-are-non-fungible-tokens. [Accessed: 27-Oct-2021].

[18] J. Kastrenakes, “Beeple sold an NFT for $69 million,” *The Verge*, 11-Mar-2021. [Online]. Available: https://www.theverge.com/2021/3/11/22325054/beeple-christies-nft-sale-cost-everydays-69-million. [Accessed: 27-Oct-2021].

[19] Q. de Beauchesne, “NFTs for Supplychain,” Medium, 26-May-2021. [Online]. Available: https://medium.com/ownest/nfts-for-supplychain-29eefa395265. [Accessed: 27-Oct-2021].

[20] D. Choi and J. Victor, “What are IPFS and Filecoin and how can they be used for NFTs?,” CoinDesk Latest Headlines RSS, 16-Feb-2022. [Online]. Available: https://www.coindesk.com/layer2/2022/02/15/what-are-ipfs-and-filecoin-and-how-can-they-be-used-for-nfts/#:~:text=IPFS%20enables%20users%20to%20store,than%20a%20brittle%20HTTP%20link. [Accessed: 18-Apr-2022].

[21] J. Smith, “HTTP VS IPFS: Is peer-to-peer sharing the future of the web?,” SitePoint, 01-Mar-2016. [Online]. Available: https://www.sitepoint.com/http-vs-ipfs-is-peer-to-peer-sharing-the-future-of-the-web/. [Accessed: 18-Apr-2022].

[22] A. Narayanan, *Bitcoin and cryptocurrency technologies: A comprehensive introduction*. Princeton, NJ: Princeton University Press, 2016.

[23] A. Kulkarni, “Blockchain and ERP systems: The Integrated Future of Supply Chain Management,” Medium, 20-Apr-2018. [Online]. Available: https://blog.chronicled.com/blockchain-versus-erp-systems-why-one-is-superior-for-supply-chain-management-4486c12d56b2. [Accessed: 18-Apr-2022].

[24] V. Chawla, “IBM blockchain: How big blue is using distributed ledgers to dominate global supply chains,” *Analytics India Magazine*, 01-Jul-2020. [Online]. Available: https://analyticsindiamag.com/ibm-blockchain-supply-chain/. [Accessed: 27-Oct-2021].

[25] CNBCTV18, “What is Sharding and why Polygon co-founder says it may not scale Ethereum Enough?,” cnbctv18.com, 05-Jan-2022. [Online]. Available: https://www.cnbctv18.com/cryptocurrency/what-is-sharding-and-why-polygon-co-founder-says-it-may-not-scale-ethereum-enough-12003272.htm. [Accessed: 18-Apr-2022].

[26] Zipmex, “3 reasons Polygon is a better alternative for NFT market,” Zipmex, 09-Feb-2022. [Online]. Available: https://zipmex.com/learn/polygon-for-nft-market/#:~:text=The%20standard%20gas%20fee%20of,than%20US%240.01%20per%20transaction. [Accessed: 18-Apr-2022].

[27] IBM, “What are smart contracts on Blockchain?,” IBM, 05-Mar-2021. [Online]. Available: https://www.ibm.com/topics/smart-contracts. [Accessed: 18-Apr-2022].

[28] Geeks For Geeks, “Node.js vs express.js,” GeeksforGeeks, 09-Dec-2020. [Online]. Available: https://www.geeksforgeeks.org/node-js-vs-express-js/. [Accessed: 18-Apr-2022].

[29] A. Dziuba, “7 advantages of node.js for startups,” Relevant Software, 27-Aug-2021. [Online]. Available: https://relevant.software/blog/7-benefits-of-node-js-for-startups/. [Accessed: 18-Apr-2022].

[30] L. Marx, “Storing data on the blockchain: The Developers Guide,” malcoded, 05-Jul-2018. [Online]. Available: https://malcoded.com/posts/storing-data-blockchain/. [Accessed: 18-Apr-2022].

[31] MongoDB, “Why use mongodb and when to use it?,” MongoDB. [Online]. Available: https://www.mongodb.com/why-use-mongodb#:~:text=Using%20MongoDB%20can%20provide%20many,modern%20databases%20such%20as%20transactions. [Accessed: 18-Apr-2022].

[32] S. Surve, “Why you should use react.js for web development,” freeCodeCamp.org, 18-Feb-2021. [Online]. Available: https://www.freecodecamp.org/news/why-use-react-for-web-development/#:~:text=React%20is%20an%20excellent%20tool,the%20web%20development%20share%20today. [Accessed: 18-Apr-2022].