The University of Hong Kong
FYP21036 2021-22

Final Report

Supply Chain Management using Blockchain and Non-Fungible Tokens

Increasing Transparency & Traceability

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

Supervisor: Dr. Yuen, John
Date of Submission: 18/04/2022

Abstract

Rapid globalisation and advancing technologies have connected the world, resulting in businesses constantly trying to improve their supply chain for customer satisfaction and financial growth. Current solutions lack traceability and transparency with companies facing a 20% loss in goods every year. Consumers are willing to pay a premium of 2-10% for more information regarding the origin of products. This report offers a potential solution by providing a blockchain-powered platform comprising a web and mobile application that utilises the power of NFTs. The supply chain member will transfer and accept the goods through a mobile application that inherently conducts this process through the tokenisation of shipments into NFTs. Polygon, a layer 2 blockchain built on top of Ethereum, is used because of its features of decentralisation, low transaction fee, high rate of transactions, and support of NFTs. In addition to the CRUD functionality for users and products data, the company management will be able to track and resolve issues in shipments through a web application offered as a part of the platform. The consumers will also be able to view and trace the origin of products at a store through the mobile application. Certain limitations have been identified in the platform, the key one being the requirement for every supply chain member to own a smartphone and use the internet. This report highlights the development schedule, implementation and feasibility of technologies, as well as the working prototypes of the mobile and web applications interacting with the backend and the blockchain. Additionally, future plans include supporting QR codes linked to NFTs for tracking homogenous goods, supply chain analytics for identifying pain points, and integration with Enterprise Resource Planning to create a trusted and traceable supply chain for companies.

Acknowledgement

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

List of Figures

List of Tables

Abbreviations

API: Application Programming Interface

CRUD: Create, Read, Update, Delete

CID: Content Identifier

ERP: Enterprise Resource Planning

IPFS: Interplanetary File System

L1: Layer 1

L2: Layer 2

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

With rising globalisation and advancing technologies, businesses across the world are becoming increasingly connected [1]. Supply chain management (SCM) plays an integral role in this and needs to suffice the needs of the companies for their better growth. Additionally, the end consumer also seems to benefit from the supply chains as the product can reach them faster. Hence, an SCM application that improves on existing solutions and can be trusted by the companies, as well as consumers, is required.

1.1. Background

A supply chain does not solely consist of a producer (company) and consumer (customer). It can be defined as a network of different "entities" that are involved in the transfer or movement of goods and services from a source to the end consumer [2]. This can include different kinds of producers as well as suppliers. For example, Wellcome will need to interact with different kinds of manufacturers and suppliers for its vast variety of products. Hence, for efficient management, all the entities need to collaborate and cooperate with each other.

Successful SCM increases customer satisfaction, reduces operating expenses and improves the financial position of a company [3]. In the current competitive scenario, SCM seems to be becoming increasingly necessary. There has been a paradigm shift from "store versus store" to "supply chain versus supply chain" [4]. According to a research by MIT Sloan School of Management, customers would be satisfied paying an extra 2-10 percent provided there is improved "supply chain transparency" which gives them more information about the origin of the product [5]. Alongside the consumers, businesses also want the goods to be readily available, authentic and cost-effective. Consequently, a robust management system that manages risk, is traceable, and provides "visibility" to all members of the chain is becoming essential [6].

The importance of the SCM market can also be quantified through its market size. From being worth 15.85 billion USD in 2020, it is expected to double in size by 2026 [7]. Any loss of goods incurred in the supply chain can have a drastic impact on the business. According to [8], companies suffer around 20% loss in their inventory every year due to mismanagement and untraceability of their products in the supply chain. Additionally, they are being pressurised by customers, NGOs and governments to reveal more data regarding their supply chains[5]. Therefore, businesses across the world are investing highly in their supply chains to improve their processes.

1.2. Objective

This project aims to develop a substitute for the current SCM systems which provides traceability and transparency, not only to the companies involved but also to the consumers. It will be a blockchain-based application that will utilise Non-Fungible Tokens (NFTs) for tracking the goods being transferred in the supply chain. A web application will be developed to serve the companies specifically as well as a mobile application to cater to the needs of both consumers and companies.

From the perspective of the companies, the application will provide a one-stop shop to improve the SCM. The management can track every product on the application with each shipment being an NFT and present on the blockchain. The consumer will not only be able to view the products in a particular store through the mobile application, but also be able to trace the origin of items to confirm ethical sourcing practices are met.

1.3. Scope and Deliverables

In order to fulfil the objectives, the team aims to deliver the following:

- An implementation of the blockchain so that tokenization and transfer of goods become possible through NFT minting and transfer

- Create an easy to use web interface:
    a. Allow the company management to input and edit information regarding their goods
    b. Allow the company management to input and edit information regarding their employees and stores
    c. Provide tracking of the shipments by utilizing the blockchain developed and NFTs transferred.
    d. Allow the company management to identify shipments with issues and handle them.

- Develop a user-friendly mobile application:

a. Company Employees/Stores: Select the goods for transport; transfer and accept shipments; achieve tracking of shipments; in case of a fault with goods, refuse the transfer which will lead to manual inspection.
b. Consumers: Trace the origin of items for different stores of the company prior to purchasing them.

With a mobile and web application based on blockchain which utilises NFT for certified exchange of goods, the project will provide a trusted alternative to all stakeholders of the supply chain which is transparent and reliable.

1.4. Outline

The purpose of this report is to present a technologically driven solution for SCM. It first specifies the problems in SCM followed by introducing a blockchain-powered application that will be beneficial to both producers and consumers. Having highlighted the scope of the project, it expands upon the technology behind the solution. Then, it outlines the development and implementation of the platform followed by displaying the results. Furthermore, it gives an account of the challenges and limitations of the project. Lastly, it describes the future improvements and plans of the project.

2. Literature Review

To further understand the importance as well as the need for the solution, this section elaborates upon the core technical components of the application followed by an explanation of related works in the field of SCM.

2.1. Technology Review

The project primarily utilises two technical components: Blockchain and NFT. The following section expands upon related findings of these technologies.

2.1.1. Blockchain

Blockchain is a digital ledger that acts as a shared database of all the transactions (data being injected on the blockchain) [9]. There is not just one location where data is stored but rather multiple locations because of which it is known as a "decentralised" database. Furthermore, all these transactions are immutable and non-editable. An additional feature of blockchain is inbuilt security since it makes use of cryptography [10]. The concept can be explained by taking the example of Google Docs. There can be multiple editors on a document who can constantly make changes. However, all these changes are stored in the server and can be reviewed or seen by all the members.

All features can be accredited to the anatomy of the blockchain (Fig. 1). It consists of many blocks, the first one being the genesis block. All blocks have a hash value that represents their uniqueness. Apart from the first block, each of them also contains the stored data and the hash value of the previous block [11]. As a result, if any data is tampered with, the hash value of the block changes, the chain breaks and the participants of the network can identify this issue. Since blockchain is distributed and based on a majority consensus, a hacker will require a massive amount of computational power to modify all instances of the blockchain[10].

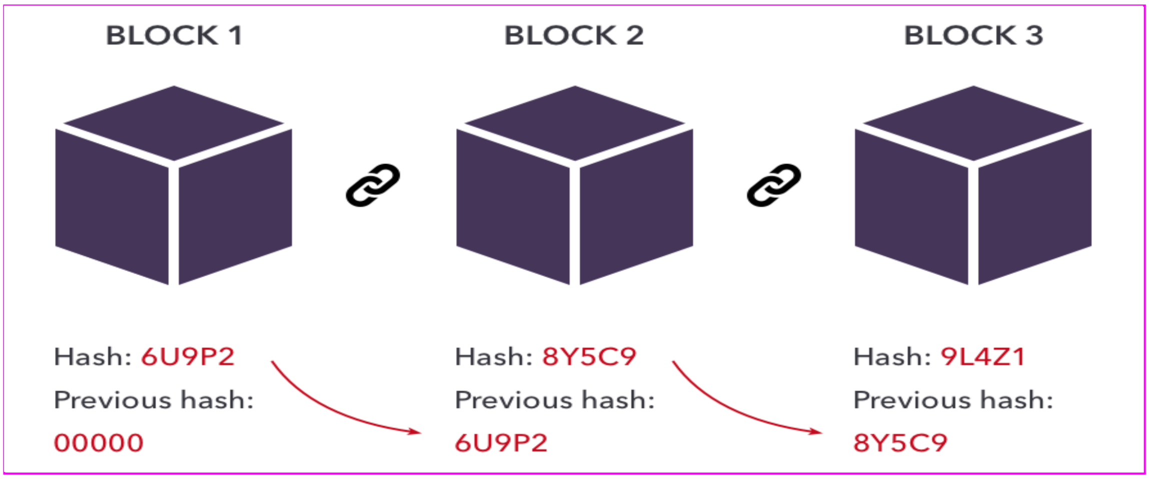


Fig 1. Anatomy of blockchain

Prior to any data (transaction) being added to the blockchain, it is "authenticated" and "authorised". There are various steps involved in this process (Fig. 2). All participants of the network have their private key and a public key which are used together to create a digital identity used for authentication [10]. Once the transaction, represented as a block, is created by a user, it needs to be ratified by a majority of the participants. Post the validation, the block extends the existing blockchain, the entire network is updated with the changes and the transaction is completed.

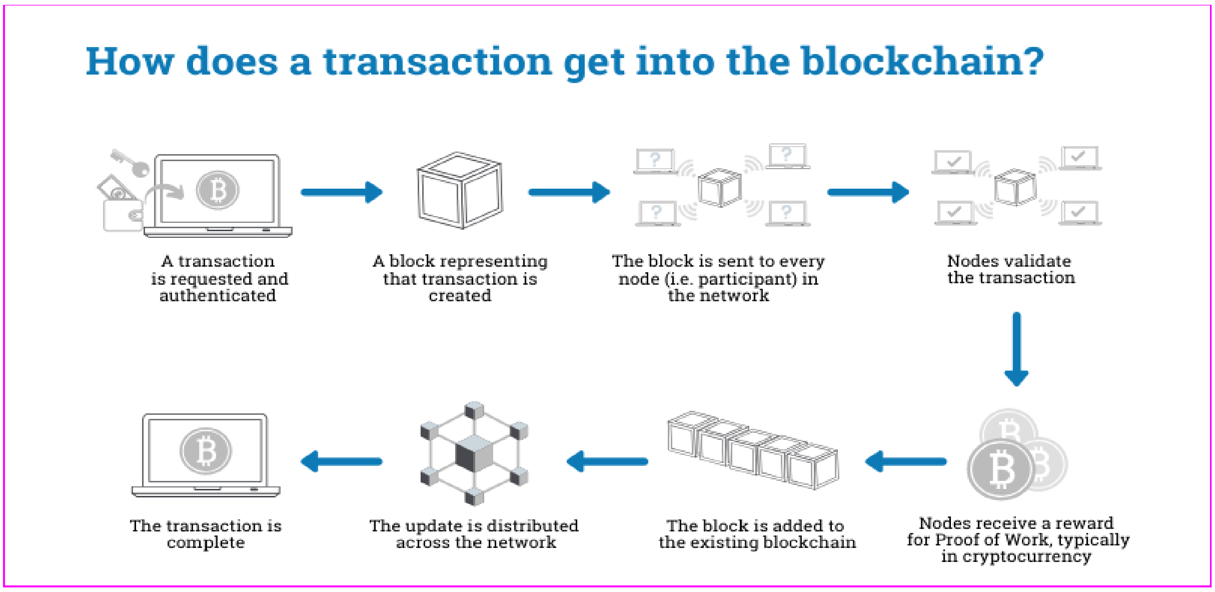

How does a transaction get into the blockchain?
A transaction is requested and authenticated
A block representing that transaction is created
The block is sent to every node (i.e. participant) in the network
Nodes validate the transaction
Nodes receive a reward for Proof of Work, typically in cryptocurrency
The block is added to the existing blockchain
The update is distributed across the network
The transaction is complete

Fig 2. Transaction Entering a Blockchain

In the project, the exchange of goods needs to be recorded and maintained for all members of the supply chain at multiple locations but having the same information. A member should have a record of the previous transactions but not be able to tamper with them. The importance of SCM for a business also demands the system to be secure. All of these requirements are fulfilled by the aforementioned features of the blockchain.

2.1.2. Scalability of Blockchain

Scalability of a blockchain dictates its capacity to sustain a large number of transactions per second (TPS) and support future development. The TPS of a blockchain is essential in its competition with centralised solutions. Since the blockchain infrastructure requires every node to validate a transaction, scalability becomes a cause of concern, even in blockchains like Ethereum and Bitcoin [12]. To tackle the same, Layer 1 and Layer 2 scaling solutions have been developed.

Layer 1 refers to blockchains like Bitcoin and Ethereum where all processes and transactions happen on the network itself. These change the rules of the blockchain protocol to improve scalability such as increasing the block size or changing the consensus mechanisms. However, with increasing transaction history, high transaction fees and a growing number of transactions, this approach is costly and tedious [13].

Layer 2 refers to technologies like Polygon and Polkadot that work on top of an existing blockchain protocol. It essentially involves shifting some parts of the blockchain's transaction data to an additional architecture that processes it and provides the results to the main chain [14]. Therefore, the burden on the base layer is reduced, and the blockchain becomes more scalable whilst maintaining decentralisation and security. Additionally, the transaction fee reduces significantly [15].

This project involves shipments of multiple supply chains of a company being minted as NFTs on the blockchain which are then transferred from one supply chain member to another. Consequently, it will involve a high number of transactions and hence a scalable blockchain. Owing to the aforementioned research, a layer 2 blockchain fulfils these criteria.

2.1.3. NFT - Non-Fungible Tokens

Fungibility of an item means it can be exchanged or replaced (For Example: Gold, currency) [16]. Cryptocurrency is also fungible since 1 ETH holds the same value as any other ETH. NFT is a unique non-fungible token and has a particular owner [17]. It is a digital unit of data stored on a few blockchains like Ethereum and Hyperledger, representing digital or real-life data. Like the concept of blockchain, the record of ownership of NFTs is immutable. Additionally, no two NFTs can be the same and cannot be interchanged. For example, the Mona Lisa is a piece of art that is unique. There can be multiple replicas that can be bought by many people, however, there is only one original owned by one party.

Each NFT has a unique ID and associated metadata that cannot be replicated by another token [17]. The metadata can range from simple details in a JavaScript Object Notation (JSON) format to a video file or an artwork. It is minted through a smart contract on the blockchain which is essential for its creation as well as the transfer. The smart contract provides the linkage to the metadata through a URI of a centralised or decentralised data storage [18] (Fig. 3.).

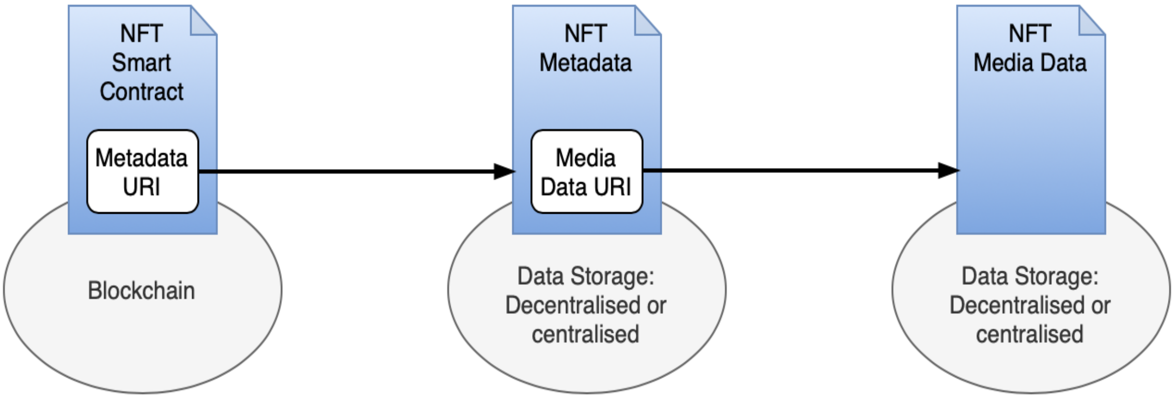

NFT Smart Contract
Metadata URI
Blockchain
NFT Metadata
Media Data URI
Data Storage: Decentralised or centralised
NFT Media Data
Data Storage: Decentralised or centralised

Fig 3. Anatomy of an NFT

NFT enables one to “buy and sell ownership” of items, the record of which is on the blockchain [11]. For the purposes of this project, all the products of the supply chain will be tokenised as NFTs and stored on the blockchain. The metadata linkage is provided through the Interplanetary File System (IPFS) Subsequently, through each step of the chain, the ownership of the product will be transferred by agreement between both parties. Additionally, the companies can trace the ownership of any product easily.

2.1.4. IPFS (Interplanetary File System)

In IPFS, a decentralised network of nodes is created that store and share the data. When data is added to the IPFS through a node, a cryptographic hash (CID) is generated which is used for accessing the data later on in the URL [19]. This prevents it from being corrupted. The data is distributed across multiple nodes connected to the IPFS. The metadata for each NFT in the project will be stored on IPFS and the NFT contains a reference to this data through the CID. IPFS is advantageous for the following reasons [20]:

- Accessible: Decentralised nature allows for public access to data that cannot be tampered
- High Bandwidth: Multiple and nearby nodes can provide the data
- Secure: Free from Distributed Denial of Service Attacks since there is not one server hosting the data. Cryptographic hashes are used to access the data which adds to the security.

2.2. Related Works

Traditional SCM systems involved the use of physical paper for the purpose of any transactions and exchange of products which included details regarding the products. These systems are time-consuming, expensive, prone to manual errors and require extensive paperwork [4]. The data is not consolidated in one place for the company and even the physical paper is untraceable. Covid-19 also had a drastic impact on these traditional systems due to various operational failures [21]. Hence, there has been the advent of many technological solutions for SCM such as Enterprise Resource Planning (ERP) and use of blockchain technologies.

ERP systems are used by businesses for a variety of features such as procurement of inventory, accounting as well as data analytics. However, they have a number of drawbacks that make them unsuitable to be used standalone. According to the Harvard Business Review [21], ERP is a tedious and expensive process with each member of the supply chain requiring a separate system. It further states how the members can only trace the goods with whom they have immediately interacted rather than the entire chain. Therefore, ERP systems lack standardization and need to be supplemented by other technologies.

Blockchain has the power to solve the problems of ERP systems for it is a “decentralised system” which can provide traceability and a uniform solution [22]. There are giants like IBM who have developed a solution for SCM using blockchain. Their partnership with TradeLens (a company trading globally) wherein all trade transactions are continuously updated on the blockchain successfully reduced the companies’ shipping time by 40% [23]. Although additional projects in the field have also been carried out by IBM, blockchain solutions still have limitations. Firstly, they are powered by “smart contracts” which provide traceability across all members but it is a difficult process [24]. Secondly, data stored on the blockchain is immutable but it can also be false data that has been added by one member of the supply chain without ratification from the other member.

Previous works in the field of SCM do highlight the benefits of blockchain, however, it is not sufficient. Hence, this project aims to use NFTs along with blockchain technology wherein each product acts as an NFT. As mentioned earlier, since NFTs dictate ownership, they will be easily traceable. Moreover, the exchange of NFTs will require the participation of two members (sending and receiving) preventing the issue of false data.

3. Methodology

This section details the workflow of the application followed by explaining and justifying the technology implementation.

3.1. Platform Workflow

The workflow of the platform (Fig. 4) is based upon using both the web dashboard and the mobile application. The web dashboard is accessible solely by the company administrator. The first step requires the administrator to create users or stores through a simple form who will be using the mobile application for creating, transferring and receiving shipments. Through this form, each user is linked to a cryptocurrency wallet and given a username and password. The details of the wallet are not visible to the user but are essential for the server to mint and transfer NFTs. Once the users are created, the management inputs goods that will be a part of the supply chain. The administrator can edit, add or delete products at any time through the web dashboard.

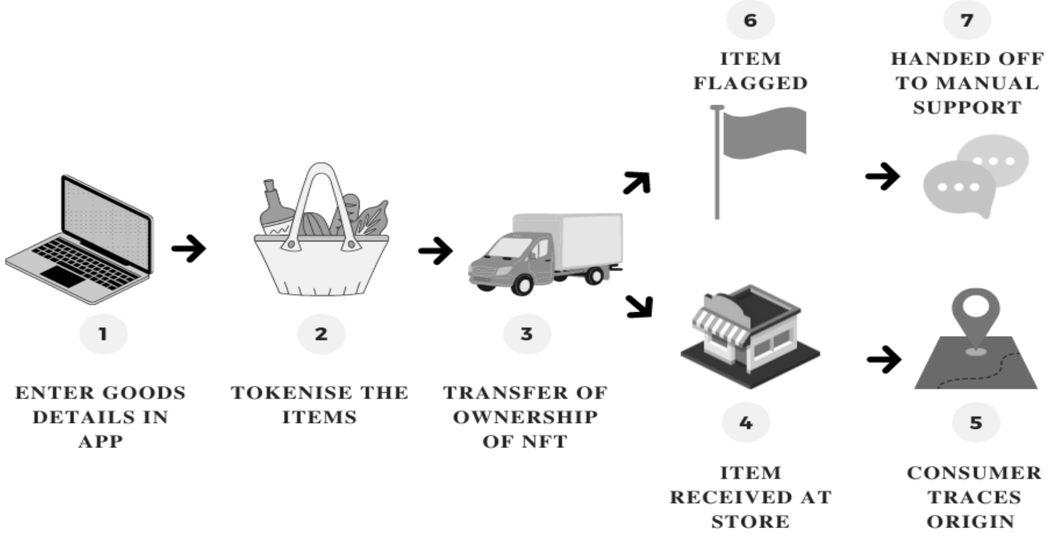

6
ITEM FLAGGED
7
HANDED OFF TO MANUAL SUPPORT
1
ENTER GOODS DETAILS IN APP
2
TOKENISE THE ITEMS
3
TRANSFER OF OWNERSHIP OF NFT
4
ITEM RECEIVED AT STORE
5
CONSUMER TRACES ORIGIN

Fig 4. High-level overview of the product workflow

Post this initial setup, members of the supply chain (users), can start using the mobile application and login using their assigned username and password. An overview of the steps is as follows:

- During the transfer of goods, the first member of the supply chain needs to select the goods and their quantity on the mobile application. The user hence creates a shipment consisting of one or more goods.

- The application, linked to the blockchain, will tokenize this shipment into an NFT using that user's wallet information. That member is now the owner of that shipment (NFT).

- In the entire supply chain, as physical goods (shipments) are transferred, the relevant members will transfer or accept the NFT through the application, thereby relieving or taking ownership of the goods.

- The users need not worry about the blockchain since all the minting and transfer of NFTs is conducted on the backend. They have to simply use the mobile application.

- If there is any loss of goods, the relevant member can refuse to accept the shipment and the company is immediately informed of this event. Consequently, the company will be aware of the loss of goods at the actual step of the supply chain.

- Having manually inspected the shipment, the administrator can resolve the issue in the shipment through the web dashboard. The resolution involves editing the products and their quantity in the shipment to account for any loss.

- Post this resolution, a new NFT is minted with the updated shipment details which can now be transferred by the relevant supply chain member. The metadata of the NFT contains linkage to the previous NFT while maintaining the same origin of the products.

- Provided there is no issue in a shipment, it finally reaches a store of the supply chain. The store becomes the owner and the entire NFT ownership history is recorded on the blockchain.

- The consumers are able to access the mobile application as guests. They can search for a store and choose a particular product in that store. This allows them to see the origin of that product as well as trace its supply chain. The consumer can view all the locations that the product was shipped and transferred from.

- Additionally, the company administrator is able to view and trace each shipment through the web application. Every shipment is linked to OpenSea (an NFT marketplace) which displays the details such as the products and the origin. The entire transaction history of that shipment is also visible on the platform showing each username alongside the wallet's public address.

3.2. Implementation & Feasibility

The following section provides an overview of different technologies used in the system infrastructure (Fig. 5). These are presented in an orderly manner discussing each technology's implementation and feasibility.

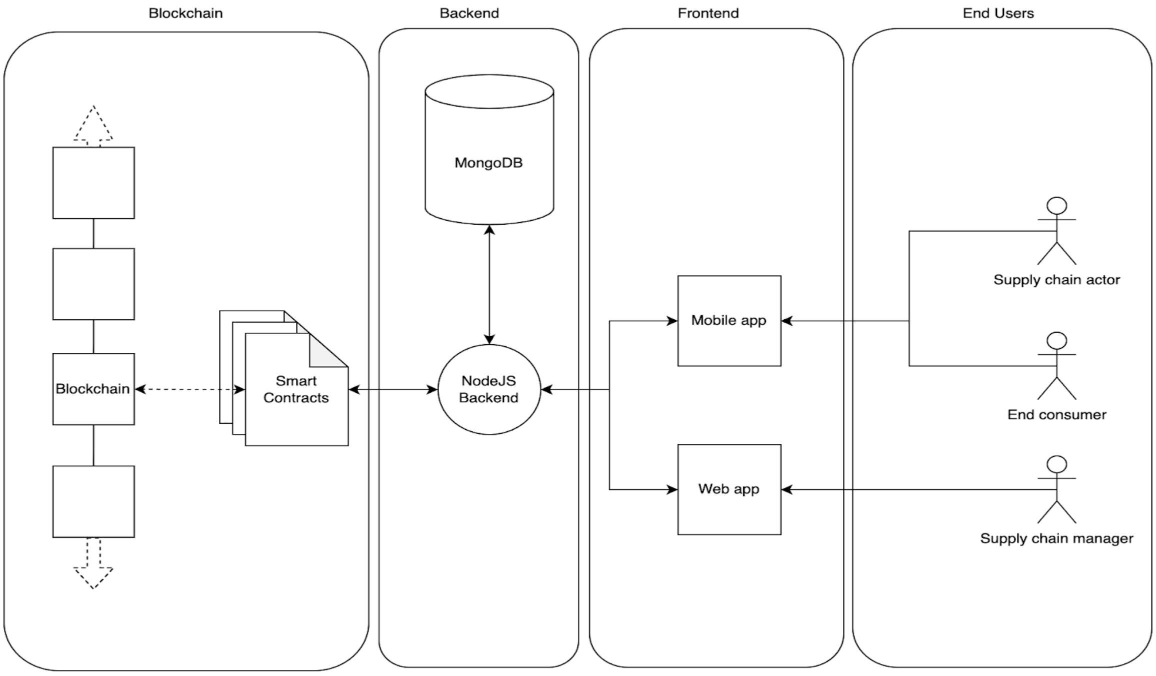


Fig 5. High-level overview of the system infrastructure

3.2.1. Blockchain

Blockchain forms the base of the application. The team needed to decide between a custom or an existing blockchain.

3.2.1.1. Ethereum (Layer 1)

After extensive research, Ethereum was chosen as the layer 1 blockchain. NFTs constitute a major part of the project and Ethereum has extensive support for NFTs, library functions and built-in features of smart contracts and state database. Furthermore, it has an established and growing developer community. However, due to a high transaction fee and low TPS as mentioned in Section 2.1.2, a layer 2 solution is required.

3.2.1.2. Polygon (Layer 2)

Polygon, a Layer-2 scaling solution built on top of Ethereum solves the above issues of Ethereum. Polygon can support up to 7000 TPS on a single sidechain which can further be increased by creating multiple chains as opposed to 12-15 TPS of Ethereum [25]. It supports the same virtual machine as Ethereum for developing contracts allowing users to use Solidity, the smart contract programming language for Ethereum. The gas fee is in its native token MATIC which reduces the average fee for any transaction from 20-50USD on Ethereum to 0.01USD on Polygon [25]. Hence, Polygon not only provides all features of Ethereum including a massive developer community but also mitigates the transaction fee and TPS.

3.2.1.3. Solidity

Solidity, an Ethereum supported high-level language, is used to create the contract which will execute on the blockchain for NFT creation and transfer. Metadata of the shipment will be stored on the NFT. These will include the various product names and their quantity as well as the origin of the shipment. The contracts written in solidity will be called using NodeJS files from the backend to provide details such as the public address of users, private address, and the metadata.

3.2.1.4. Tokenisation of Goods

The Hardhat development environment is used to develop and compile the smart contracts for the creation and transfer of tokenised shipments. It assists developers in handling and automating repetitive tasks while building smart contracts, and has a rapidly growing developer community [26].

Every shipment on the platform is represented by an NFT which is transferred from one supply member to another. A solidity-based smart contract compiled on Hardhat is used for the minting process. Alchemy, a microservice that supplies APIs for developers to interface with the blockchain (L2 or L1), is used to deploy the compiled contract [27].

The minting infrastructure (Fig. 6) requires the following two steps:

- Creating NFT metadata regarding the shipment products and origin followed by uploading it on IPFS using Pinata (a pinning service that uploads data to the IPFS such that it is never deleted)
- Call the mint function of the deployed contract and provide it with the CID from Pinata. This NFT can be visualised on NFT marketplaces like OpenSea.

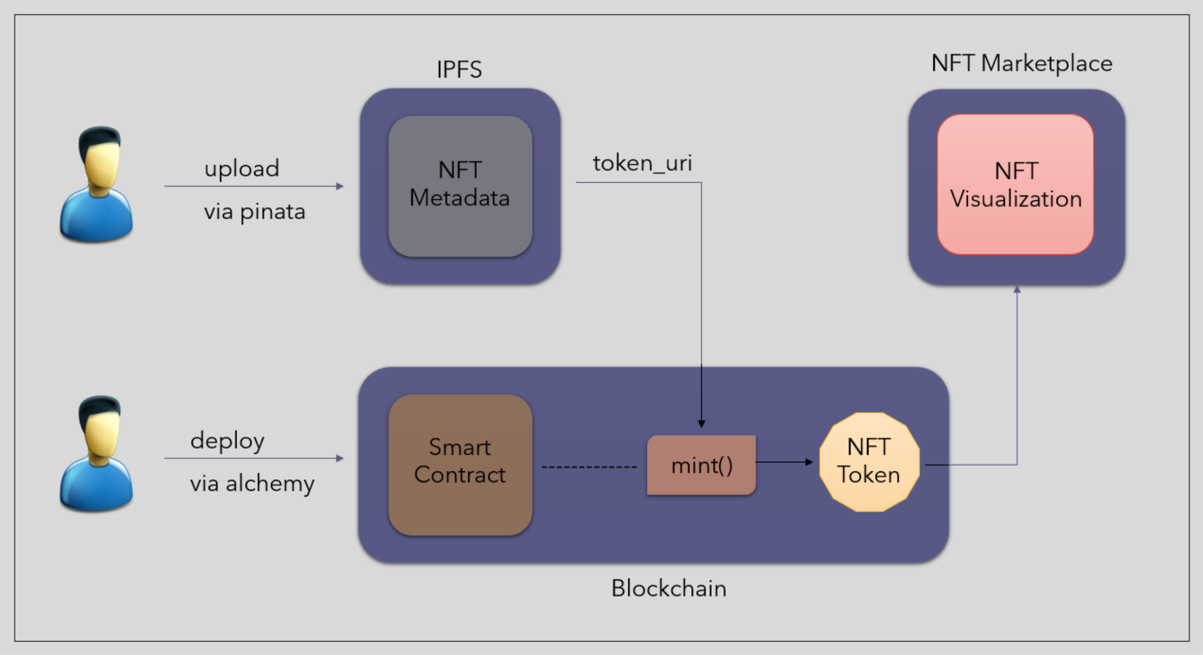


Fig. 6. NFT Minting Architecture

3.2.2. Backend:

The backend of the application is used for interfacing with both the blockchain as well as the mobile and web application. The following are key components of the backend.

3.2.2.1. Express

The backend server will be Express, a Javascript-based framework based in a server-side programming language with vast built-in functions. It is lightweight and assists in setting up an MVC (Model-View-Controller) architecture for the application [28]. It is used for effortlessly managing routing, error handling and HTTP requests, which is otherwise a time-consuming activity. Lastly, since it is written in Javascript, it provides seamless integration with NodeJS, the frontend as well as the database, MongoDB.

3.2.2.2. NodeJS

NodeJS is an open-source JavaScript based language used to create server-side tools and applications. Not only does it have built-in modules but its package manager, NPM (node package manager), provides a large number of installable libraries and components. Additionally, it follows the asynchronous nature of JavaScript and hence maintains "non-blocking" code execution [29]. It also has an extensive developer community which makes it an ideal choice for development.

3.2.3. Database: MongoDB

The platform supports a web as well as a mobile application, each requiring different kinds of information. The metadata of each shipment stored on the blockchain consists primarily of the details of the products inside the shipment. Each shipment is an NFT belonging to a particular wallet (public address, private address). As a result, this public address needs to be mapped with users of the application. Furthermore, every supply chain member's present, incoming and outgoing shipments need to be stored. The platform also needs a database of all the products which will be a part of the supply chain. Consequently, an efficient storage solution is essential.

MongoDB is a non-relational database that will be used to store information regarding users of the applications. It will be populated by the backend for each company and used for verified access to the mobile application. The products of a supply chain will also be stored on it. It stores data as compressed Binary JSON format (BSON) but provides the results as JSON making it developer-friendly [30]. BSON provides multiple searching and indexing options. Due to the complexity and sensitive data of the platform, every company requires a separate MongoDB database. For project development, the team has used MongoDB Atlas, a database-as-a-service on the cloud provided by MongoDB itself, that assists in streamlined deployment and management of databases [30]. MongoDB is thus extremely flexible as opposed to using an SQL database with tables, and has seamless integration with NodeJS and Express.

3.2.4. Frontend

With the backend being javascript driven, the frontend was also chosen of the same language. The mobile application will be created in React Native whereas the web application will be developed in ReactJS.

3.2.4.1. Mobile Application

The mobile application will be used by members of the supply chain to create, transfer and receive shipments. Consumers will have a separate interface to view the list of products and trace their origin. Thus, the application needs to be user-friendly with an easy to use interface. However, it also needs to call the backend API for facilitating the NFT tokenisation and exchange.

React Native is a cross-platform language that will allow any consumer or supply chain member to use the application irrespective of the operating system. It facilitates access to a number of third party libraries and is written in JavaScript [31]. Lastly, it provides a fast interface and has a vast developer community.

3.2.4.2. Web Application

The web application will use the backend API to provide CRUD (Create, Read, Update, Delete) options for products, users and stores. It will also provide additional features for tracking shipments and goods.

React is an open-source JavaScript library that provides reusable components [32]. Thus, a CRUD for two separate entities (users and products) is easily supported. Furthermore, React is extremely fast and extensive for it is able to render the page with new data without reloading. The dashboard requires products and users to be edited whereas the shipments are updated in real-time whenever there is any transfer or creation. Like React Native, it has extensive community support and plugins which can be used to provide a better user interface.

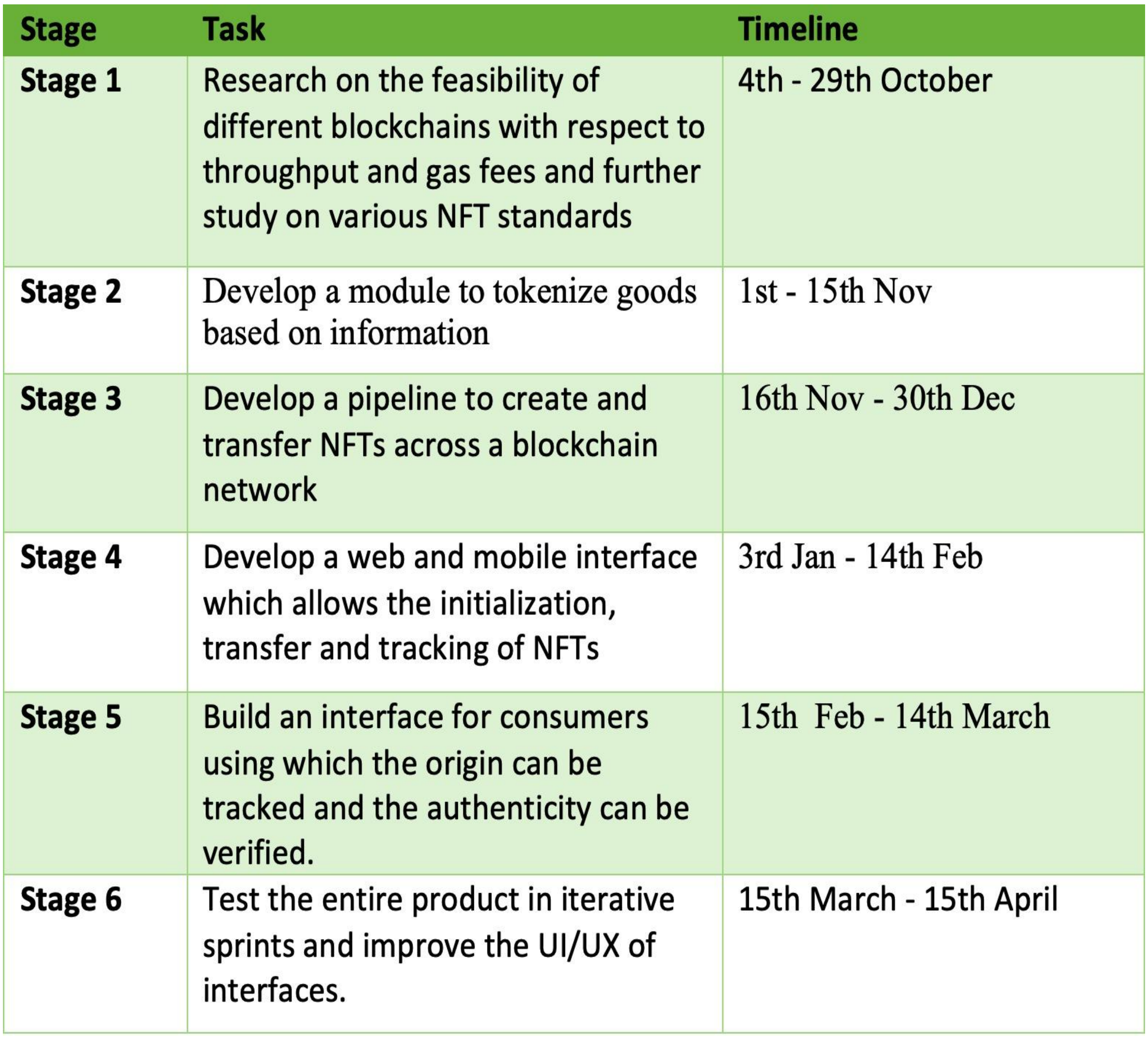
4. Project Development

This section introduces the development schedule of the project. It then expands upon and showcases the results of the implementation of the blockchain, mobile application and web application.

4.1. Schedule

The project is divided into six different stages where each stage is considered an iteration to adhere to an Agile development process (Table 1). Every stage marks the completion of an important task essential for starting future tasks.

Table 1. Project Schedule



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

As shown in Fig. 5, the platform infrastructure is composed of 4 major parts: Blockchain, Backend, Database and Frontend. This section expands upon the above aspects followed by a brief description of the project's codebase and version control.

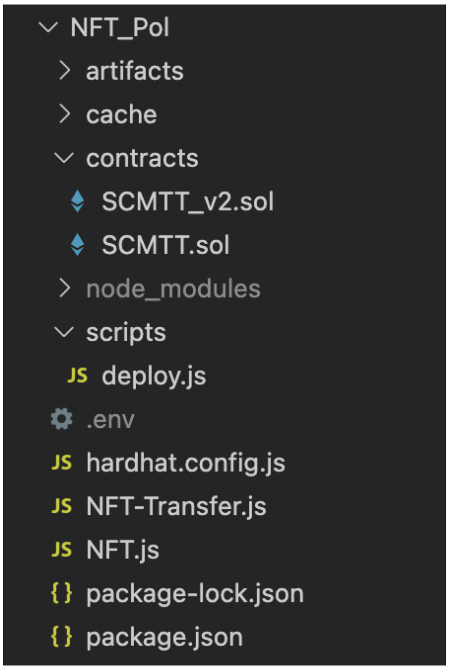
4.2.1. Blockchain Development

While the contracts for blockchain are written in Solidity, the entire development is on a NodeJS environment for ease of integration with the frontend and the backend of the application. The major dependencies of this phase of development are:

- "@openzeppelin/contracts": Smart contract repository with multiple contracts including the ERC-721 contract which provides the template for creating an NFT
- "@alch/alchemy-web3": Assists in setting up the blockchain connection and infrastructure as explained in Section 3.2.1.4

4.2.1.1. Project Structure:

The structure involves multiple sections essential in the entire lifecycle of converting a shipment into an NFT (Fig. 7). "scripts" folder contains the file for deployment using alchemy whereas the solidity contract is stored under the "contracts" folder. "NFT.js" and "NFT-Transfer.js" provide functions for minting and transferring an NFT respectively.

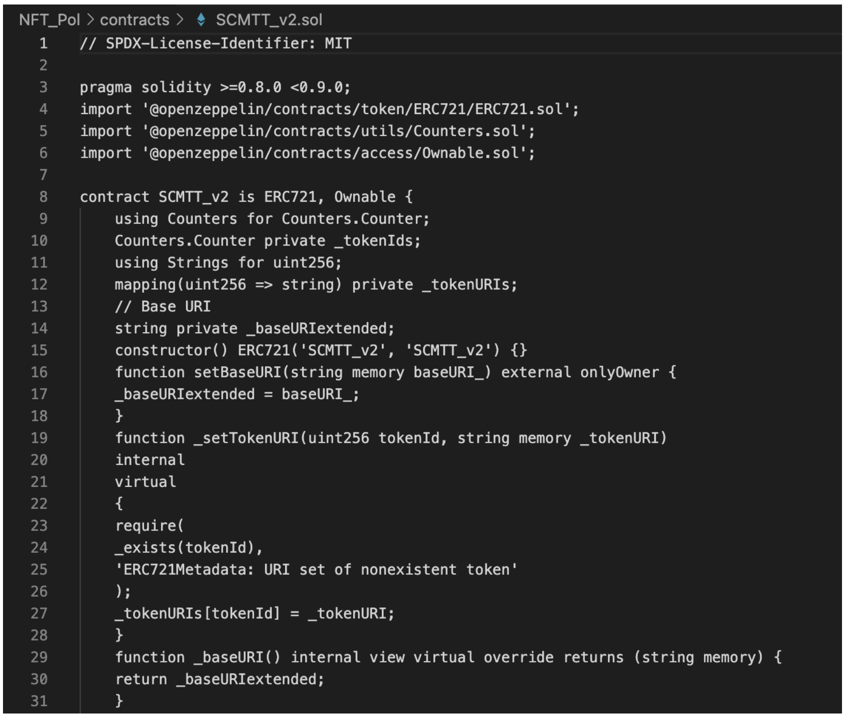


```
NFT_PoI
  artifacts
  cache
  contracts
    SCMTT_v2.sol
    SCMTT.sol
  node_modules
  scripts
    deploy.js
  .env
  hardhat.config.js
  NFT-Transfer.js
  NFT.js
  package-lock.json
  package.json
```

Fig. 7. Blockchain Project Structure

4.2.1.2. Smart Contract Deployment

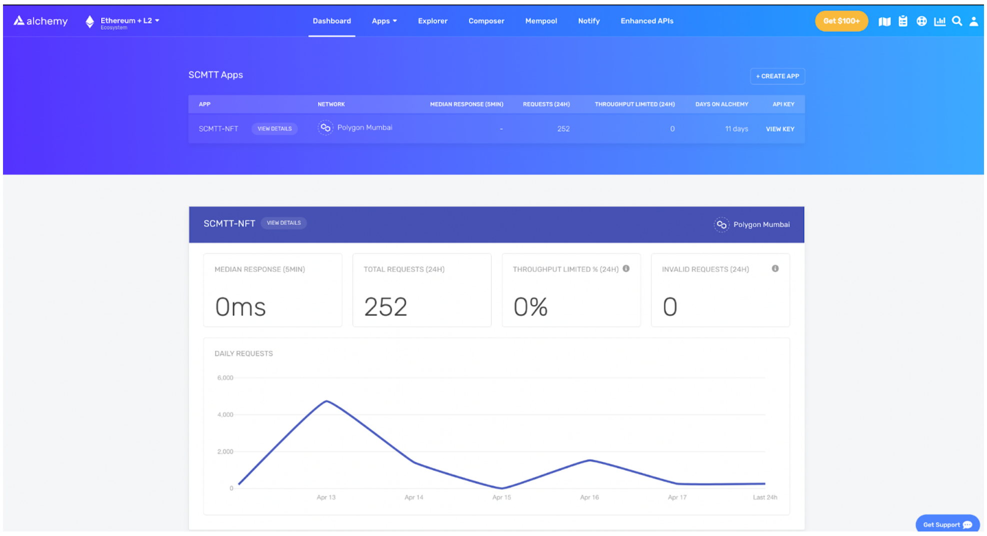
The solidity-based smart contract (Fig. 8) is deployed on the polygon-mumbai blockchain network using the “deploy.js” file supported by the Hardhat runtime environment as well as Alchemy. On successful deployment, a transaction hash is received which can be used to attain the contract address.



```
// 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. NFT token contract (ERC-721)

As all interactions with the blockchain are processed through Alchemy, the Alchemy dashboard also visualises each API call (Fig. 9).



SCMTT Apps

APP	NETWORK	MEDIAN RESPONSE (5MIN)	REQUESTS (24H)	THROUGHPUT LIMITED (24H)	DAYS ON ALCHEMY	API KEY
SCMTT-NFT	Polygon Mumbai	-	252	0	11 days	VIEW KEY

SCMTT-NFT Polygon Mumbai

MEDIAN RESPONSE (5MIN) 0ms

TOTAL REQUESTS (24H) 252

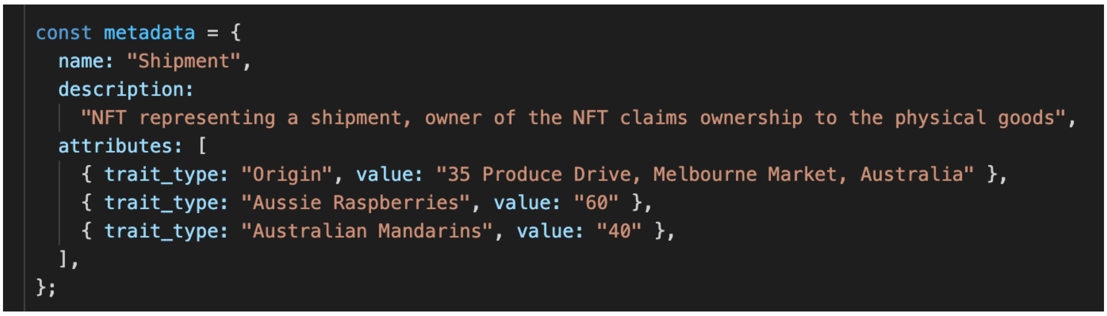
THROUGHPUT LIMITED % (24H) 0%

INVALID REQUESTS (24H) 0

Fig. 9. Alchemy Dashboard

4.2.1.3. NFT Minting

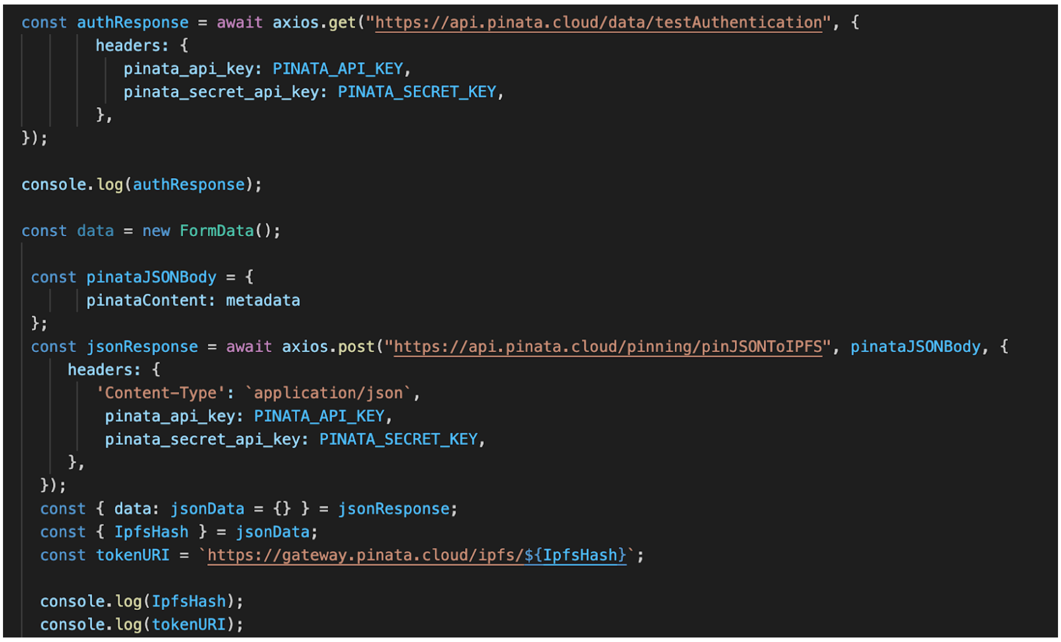
To initiate the minting process, the NFT metadata needs to be prepared and stored. As mentioned in Section 3.2.1.4, the metadata is stored on IPFS using Pinata and the CID generated is used in the token URI. The NFT metadata is in JSON and follows a specific format (Fig. 10). Each NFT represents a shipment and hence, the metadata contains information regarding the products in the shipment as well as the origin of the shipment. These product details and the origin (the geolocation of that user) are supplied to the backend from the supply chain member's mobile application via an API.



```
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. NFT Metadata Example

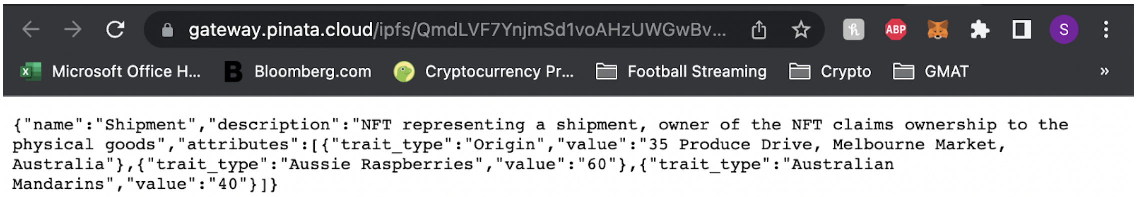
"NFT.js" is called by the backend to mint an NFT for a shipment. It receives the metadata for the shipment and the wallet credentials of the supply chain member from the database. The first step involves uploading the metadata to IPFS and retrieving the CID (cryptographic hash). This is achieved through this code snippet inside NFT.js (Fig. 11).



```
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. Metadata uploading script

The uploaded metadata can be viewed by using the generated CID along with the ipfs Pinata gateway URL which together form the token URI (Fig. 12).



```
{"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. 12. NFT Metadata on IPFS network

The "NFT.js" makes use of the owner's wallet credentials and token URI of the metadata to call the deployed contract and mints the NFT for the shipment. On successful creation, a transaction hash is generated and this file returns the token ID of the newly minted NFT to the backend. The backend then assigns this token ID and the shipment to the relevant user in the database.

4.2.1.4. NFT Transfer

Once a tokenised shipment is transferred in the mobile application from one supply chain member (sender) and then accepted by another supply chain member (receiver), the function inside "NFT-Transfer.js" is called. This function requires the sender's wallet credentials, the receiver's public address and the token ID of that shipment. As aforementioned, the database stores the token ID of each shipment. The data is thus fed to the function from the backend and the transfer of the NFT is conducted from the sender to the receiver. On a successful transfer, a transaction hash is generated and the function returns a resolved Promise object to the backend. Thereafter, the shipment details alongside the token ID are also transferred to the receiver on the database.

4.2.2. Mobile Application Frontend Development

The mobile application has two customer segments (Fig. 13):

a) Company supply chain members: These include the employees who will be transferring, accepting and creating the shipments. It also incorporates the stores of a company that will only be allowed to accept or reject a shipment. They can login using an assigned username and password.

b) The end consumers: This includes the general public who wish to view the details of the products in a store. Through the application, they can login as a guest and select a particular store of the company to trace the origin of any product of the store.

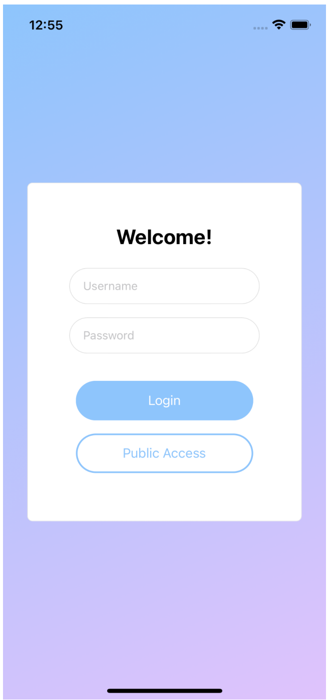
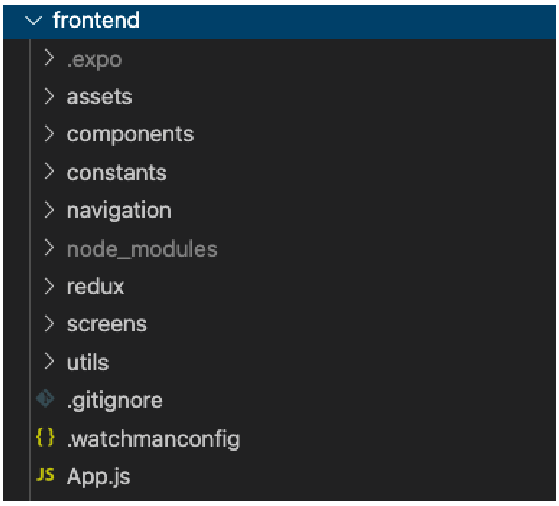


Fig. 13. Login Page

4.2.2.1. Project Structure

The mobile application is developed using React Native. The Expo platform is used to streamline the development across platforms and run the Android/iOS simulators. Redux, a state management plugin, is used to keep track of the logged-in user and details such as username and store location if applicable. A modular structure, as well as self-explanatory naming of the folders, is implemented for ensuring better collaborative development (Fig. 14). ‘App.js’ serves as the entry point, all the routing is maintained inside the ‘navigation’ folder whereas all the major screens are contained inside the ‘screens’ folder.



```
⌄ frontend
  > .expo
  > assets
  > components
  > constants
  > navigation
  > node_modules
  > redux
  > screens
  > utils
    .gitignore
  {} .watchmanconfig
  JS App.js
```

Fig. 14. Mobile App Project Structure

4.2.2.2. Screens: Supply Chain Members

The supply chain members consist of the stores as well as the employees. Apart from the “Login” page, there are six additional screens in the application. Employees have access to all of them whereas Stores have access to three of them since a store is not allowed to create or transfer a shipment. The screens are as follows:

- **Profile Page:** Acts as the homepage for various features of the application. As mentioned above, since there is a slight difference in the features for stores and employees, the profile page differs as well (Fig. 15). An employee, in addition to the features of the stores, can also create and transfer shipments as well as see the shipments that he/she has transferred. The full list of features for the employee can be seen in the side menu bar (Fig. 15).

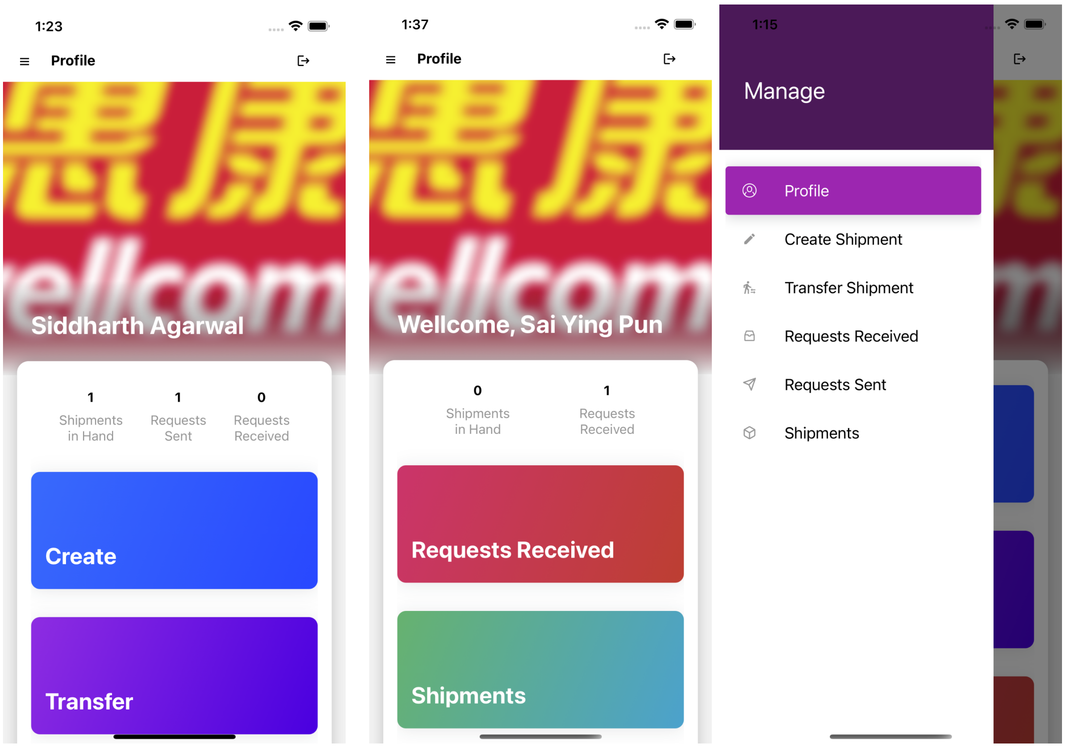
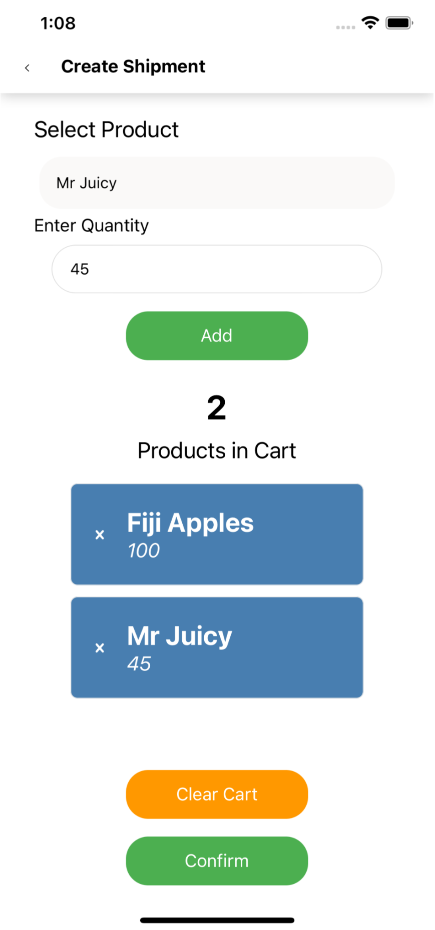


Fig. 15. Employee Profile Page (left); Store Profile Page (center); Menu bar page (right)

- **Create Page:** This page allows employees to create a shipment. The shipment only contains products that the administrator would have added on the web dashboard. The employee can select the product and enter its respective quantity. To make the page user-friendly, the employee can clear the entire cart or delete specific items from the cart (Fig. 16). There is a “<” button on the top left for the user to navigate to the profile page. If the employee is satisfied with the cart, he/she presses the confirm button. On confirmation, the geolocation API is called for the employee to get the current location (Fig. 17). This location alongside the details of the products becomes the metadata for the NFT.



1:08

‹ Create Shipment

Select Product

Mr Juicy

Enter Quantity

45

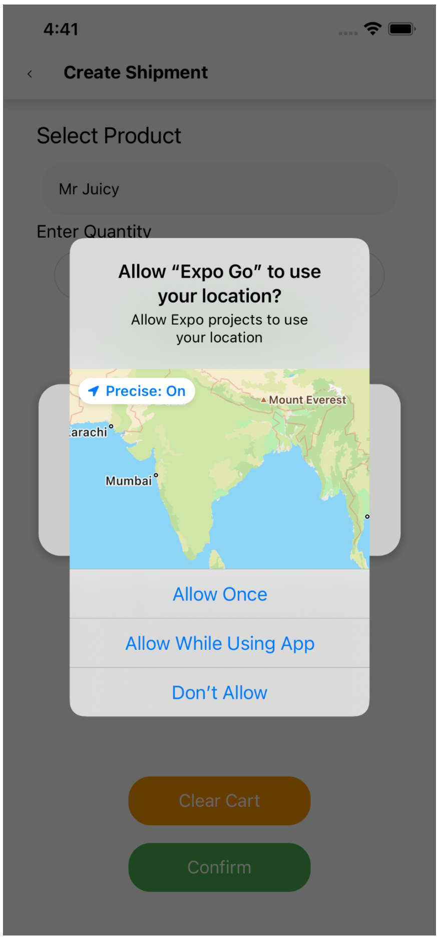
2

Products in Cart

× Fiji Apples
100

× Mr Juicy
45

Fig. 16. Create Page



4:41

‹ Create Shipment

Select Product

Mr Juicy

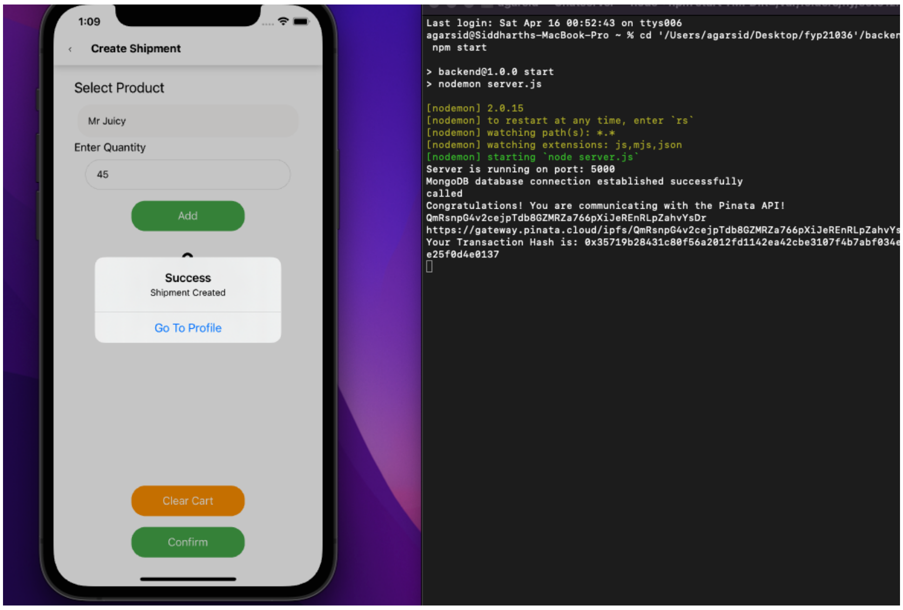
Enter Quantity

Allow “Expo Go” to use your location?

Allow Expo projects to use your location

Fig. 17. Mobile GeoLocation

After providing the location, the NFT is minted for this shipment and on successful creation, the employee is redirected to the profile page (Fig. 18).



1:09

Create Shipment

Select Product

Mr Juicy

Enter Quantity

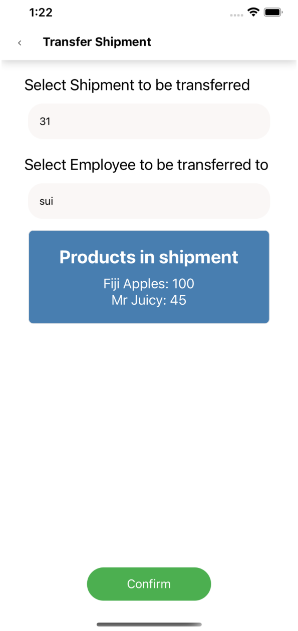
45

Success
Shipment Created

```
Last login: Sat Apr 16 00:52:43 on ttys006
agarsid@Siddharths-MacBook-Pro ~ % cd '/Users/agarsid/Desktop/fyp21036'/backen
 npm start

> backend@1.0.0 start
> nodemon server.js

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Server is running on port: 5000
MongoDB database connection established successfully
called
Congratulations! You are communicating with the Pinata API!
QmRsnpG4v2cejpTdb8GZMRZa766pXiJeREnRLpZahvYsDr
https://gateway.pinata.cloud/ipfs/QmRsnpG4v2cejpTdb8GZMRZa766pXiJeREnRLpZahvYs
Your Transaction Hash is: 0x35719b28431c80f56a2012fd1142ea42cbe3107f4b7abf034e
e25f0d4e0137
```

Fig. 18. NFT Creation through Mobile App

- **Transfer Page:** An employee accesses this page to transfer a shipment he/she currently owns to another employee or store. On selecting the shipment ID, the list of products in that shipment is displayed to prevent any error (Fig. 19).



1:22

Transfer Shipment

Select Shipment to be transferred

31

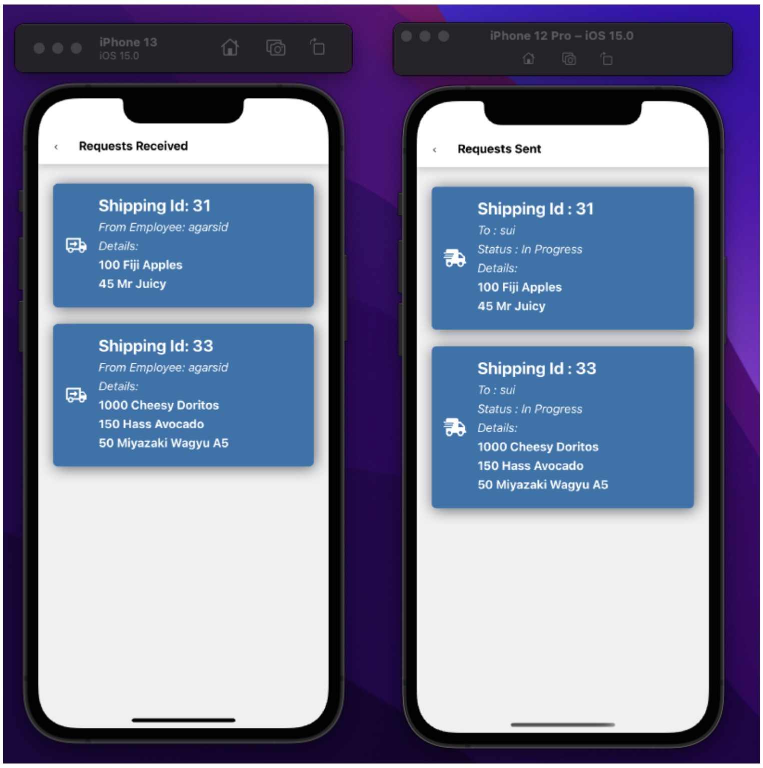
Select Employee to be transferred to

sui

Products in shipment
Fiji Apples: 100
Mr Juicy: 45

Fig. 19. Transfer Page

On setting up a transfer, the NFT is not yet transferred from this employee to another. This screen sets up the transfer initiation process wherein the relevant shipment is shown on the Requests Sent and Requests Received Pages of the sender and receiver respectively (Fig. 20).



Requests Received

Shipping Id: 31
From Employee: agarsid
Details:
100 Fiji Apples
45 Mr Juicy

Shipping Id: 33
From Employee: agarsid
Details:
1000 Cheesy Doritos
150 Hass Avocado
50 Miyazaki Wagyu A5

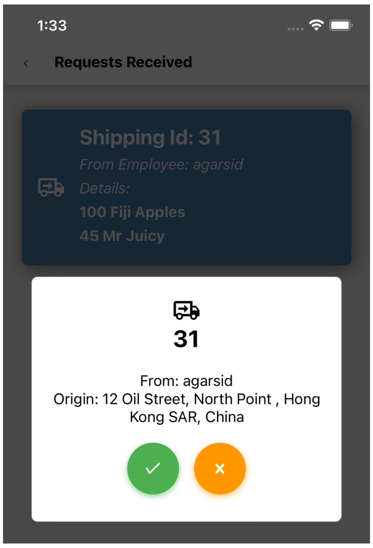
Requests Sent

Shipping Id : 31
To : sui
Status : In Progress
Details:
100 Fiji Apples
45 Mr Juicy

Shipping Id : 33
To : sui
Status : In Progress
Details:
1000 Cheesy Doritos
150 Hass Avocado
50 Miyazaki Wagyu A5

Fig. 20. Requests Received (Left; user: sui) & Requests Sent (Right, user: agarsid)

- **Requests Received:** As can be seen in Fig. 20, the Requests received page displays all the shipments that an employee has received from other employees. Continuing with an easy to use UI, the user sees all details of the shipment including the sender of the shipment. The user gets an option to accept or reject the shipment by clicking on the card (Fig. 21). If the shipment is accepted, the NFT is transferred from the sender to the receiver (current user). However, if it is rejected, the shipment remains with the sender. The same page and functionality are also available to the stores.



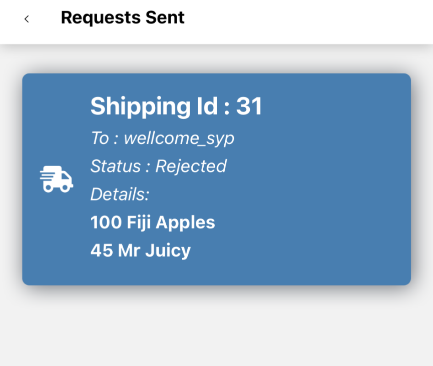
Shipping Id: 31
From Employee: agarsid
Details:
100 Fiji Apples
45 Mr Juicy

31

From: agarsid
Origin: 12 Oil Street, North Point , Hong Kong SAR, China

Fig. 21. Accept or Reject a Shipment

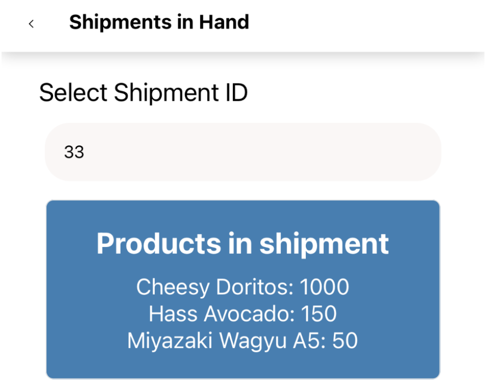
- **Requests Sent:** Similar to the Requests Received page, this page displays all the shipments that an employee has sent to other employees. The user can view all the details of the shipments he has transferred including the receiver of the shipment. A "Status" field exists for each shipment on this page as well. This signifies whether a shipment is in progress or has been rejected. If the shipment is accepted, it implies that the NFT has been transferred from the sender (current user) to the receiver. As a result, that shipment is cleared from this page. However, if it is rejected, the shipment remains with the sender and the status changes to "Rejected" (Fig. 22). Manual inspection is then conducted by the company. Having solved the issue, the administrator can resolve this issue from the web dashboard (Section 4.2.3.2).



Shipping Id : 31
To : wellcome_syp
Status : Rejected
Details:
100 Fiji Apples
45 Mr Juicy

Fig. 22. Rejected Requests Sent Page

- **Shipments:** This is the second major page common to employees and stores. This page allows the supply chain member to select a shipment ID from a dropdown list and view the products in the shipment (Fig. 23). This page is part of making the user experience better wherein each member is aware of what products he/she owns.


‹ Shipments in Hand

Select Shipment ID

33

Products in shipment

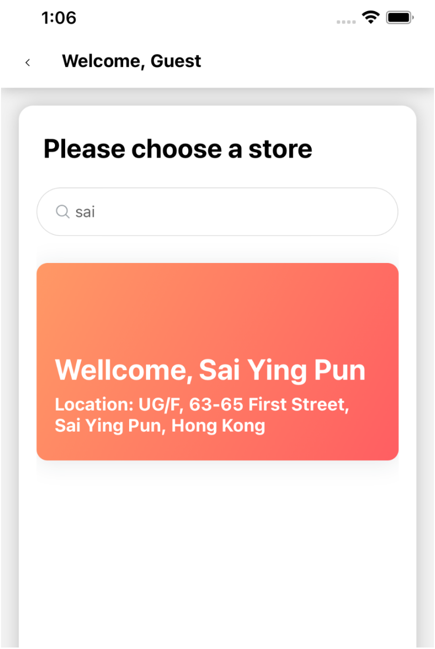
Cheesy Doritos: 1000
Hass Avocado: 150
Miyazaki Wagyu A5: 50

Fig. 23. Shipments in Hand Page

4.2.2.3. Screens: Consumers

Consumers are the end of the supply chain. The mobile application provides the consumers with three basic screens to trace the origin of items.

- **Stores:** On accessing the application as a guest, the consumer can choose a particular store of the company (Fig. 24). The stores can be searched by name and location as well for faster and easier access.



‹ Welcome, Guest

Please choose a store

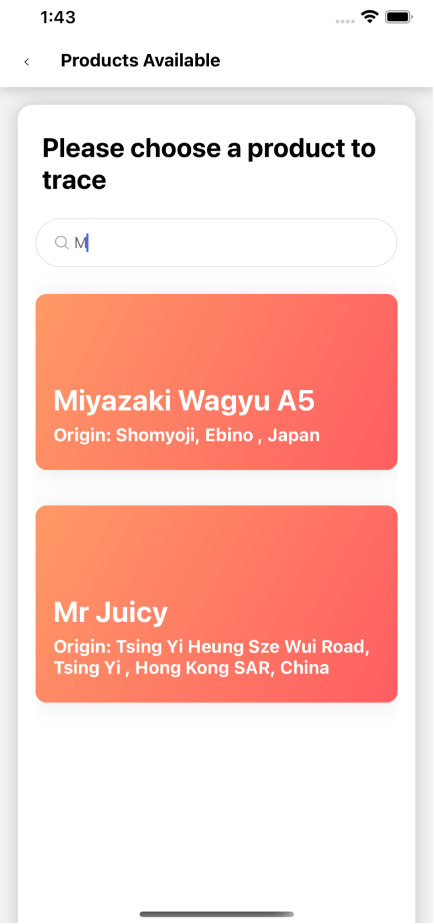
sai

Wellcome, Sai Ying Pun

Location: UG/F, 63-65 First Street, Sai Ying Pun, Hong Kong

Fig. 24. Select Store Page

- **Products:** Having chosen a store, the consumer can choose a particular product of the company (Fig. 25). Like the "stores" page, the products can also be searched by name and location. Each product has an associated origin with it to distinguish between homogeneous products.



1:43

Products Available

Please choose a product to trace

M

Miyazaki Wagyu A5

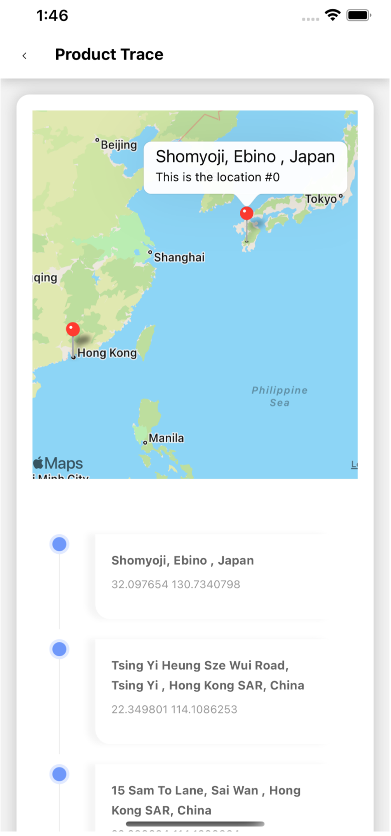
Origin: Shomyoji, Ebino , Japan

Mr Juicy

Origin: Tsing Yi Heung Sze Wui Road, Tsing Yi , Hong Kong SAR, China

Fig. 25. Select Products Page

- **Product Trace:** The final and most important page for a consumer is the product tracing page. Having selected a product of a particular store, the user is now able to view the origin of the product alongside the various locations it has travelled from to reach the store (Fig. 26). All this data has been recorded on the blockchain with the NFT transfer happening simultaneously with the transfer of the shipment. Since this page serves the end consumer, its layout is simple yet efficient. As shown in the figures, a timeline of the locations alongside a map with the exact coordinates of all the locations is provided. The customer can tap on a location pin to see the entire address. Furthermore, the user can select any location from the timeline and shift to that location on the map as well (Fig. 27). This page serves the purpose of providing complete supply chain transparency to the end consumer.

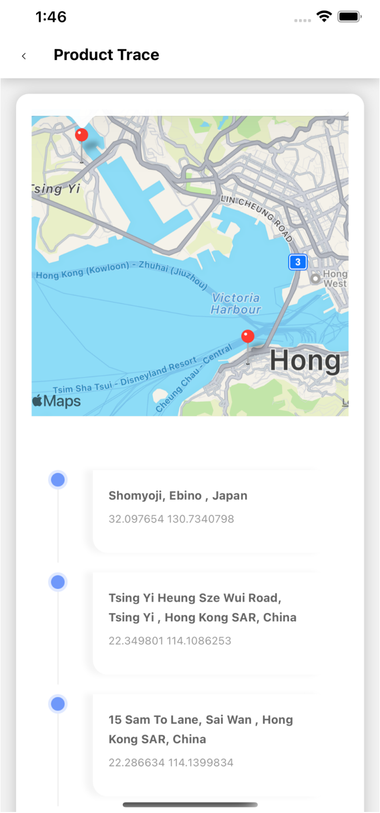

1:46

Product Trace

Shomyoji, Ebino , Japan
32.097654 130.7340798

Tsing Yi Heung Sze Wui Road, Tsing Yi , Hong Kong SAR, China
22.349801 114.1086253

15 Sam To Lane, Sai Wan , Hong Kong SAR, China


1:46

Product Trace

Tsing Yi
LIN CHEUNG ROAD
3
Hong Kong (Kowloon) - Zhuhai (Jiuzhou)
Hong West
Victoria Harbour
Tsim Sha Tsui - Disneyland Resort - Central
Cheung Chau
Hong
Maps

Shomyoji, Ebino , Japan
32.097654 130.7340798

Tsing Yi Heung Sze Wui Road, Tsing Yi , Hong Kong SAR, China
22.349801 114.1086253

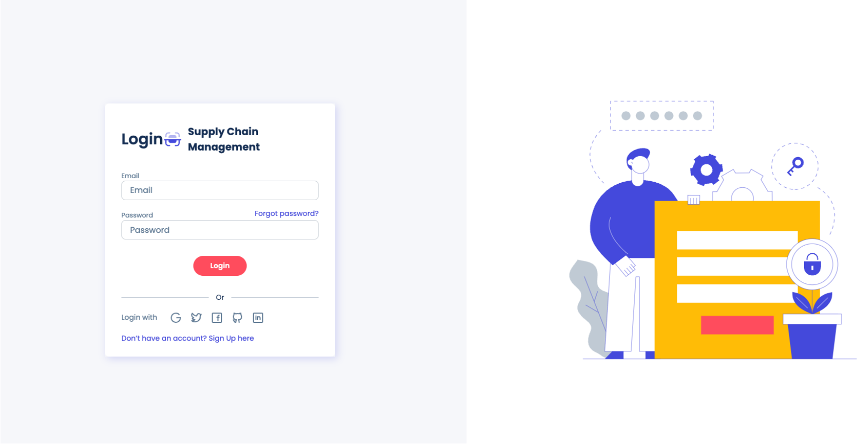
15 Sam To Lane, Sai Wan , Hong Kong SAR, China
22.286634 114.1399834

Fig. 26. Tracing Page 1

Fig. 27. Tracing Page 2

4.2.3. Web Application Frontend Development

The web application provides a dashboard to the company administrator. The three components of the mobile application, that is, users, shipments and products can be handled in this dashboard. It allows the administrator to login and use several features (Fig. 28).


Login

Supply Chain Management

Email: Email

Password: Password

Forgot password?

Login

Or

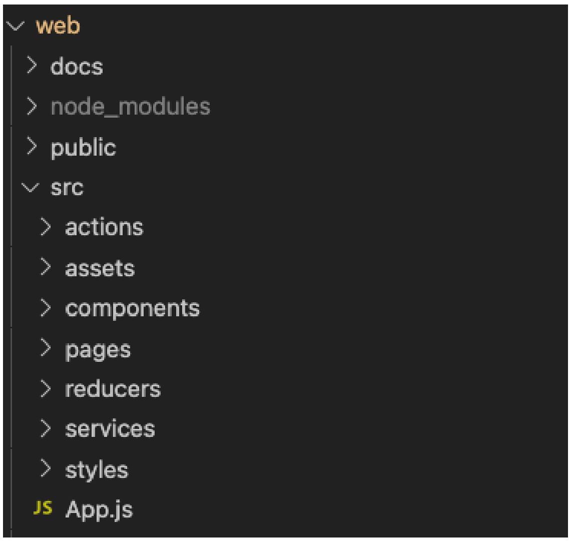
Login with

Don't have an account? Sign Up here

Fig. 28. Web Login Page

4.2.3.1. Project Structure

JavaScript being the base language of the entire project, ReactJS was used in the web application. Similar to the mobile application, a modular structure, with self-explanatory naming of the folders, is implemented (Fig 29.). ‘App.js’ serves as the entry point, all the reusable components such as the sidebar and notification popups are maintained inside the “components” folder whereas all the major pages are contained inside the “pages” folder.



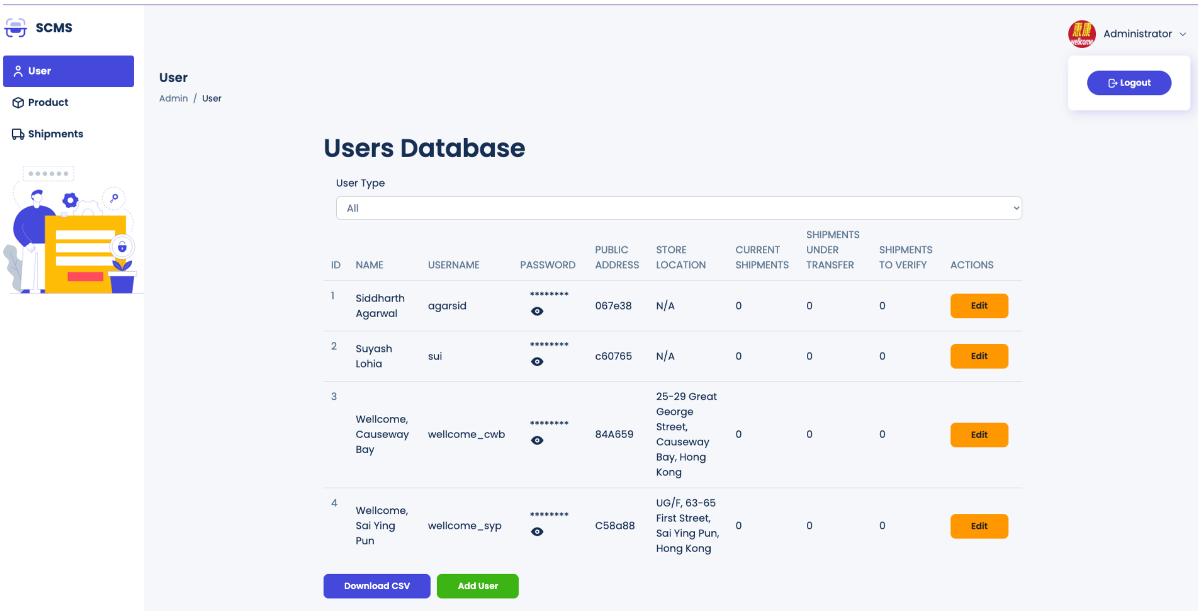
```
⌄ web
  > docs
  > node_modules
  > public
  ⌄ src
    > actions
    > assets
    > components
    > pages
    > reducers
    > services
    > styles
    JS App.js
```

Fig. 29. Web App Project Structure

4.2.3.2. Major Pages

There are 3 major pages in the application, each of which corresponds to a specific use case for the company administrator.

- **User Management (Home):** Post login, the application displays a list of users to the administrator (Fig. 30). This page is essential in setting up the supply chain members of the company apart from the end consumers who use the mobile application to create and transfer shipments. All the details are displayed in a table sorted by the username. Every existing user can be edited to change their display name and passwords. The information can also be downloaded as a CSV for internal records.



User

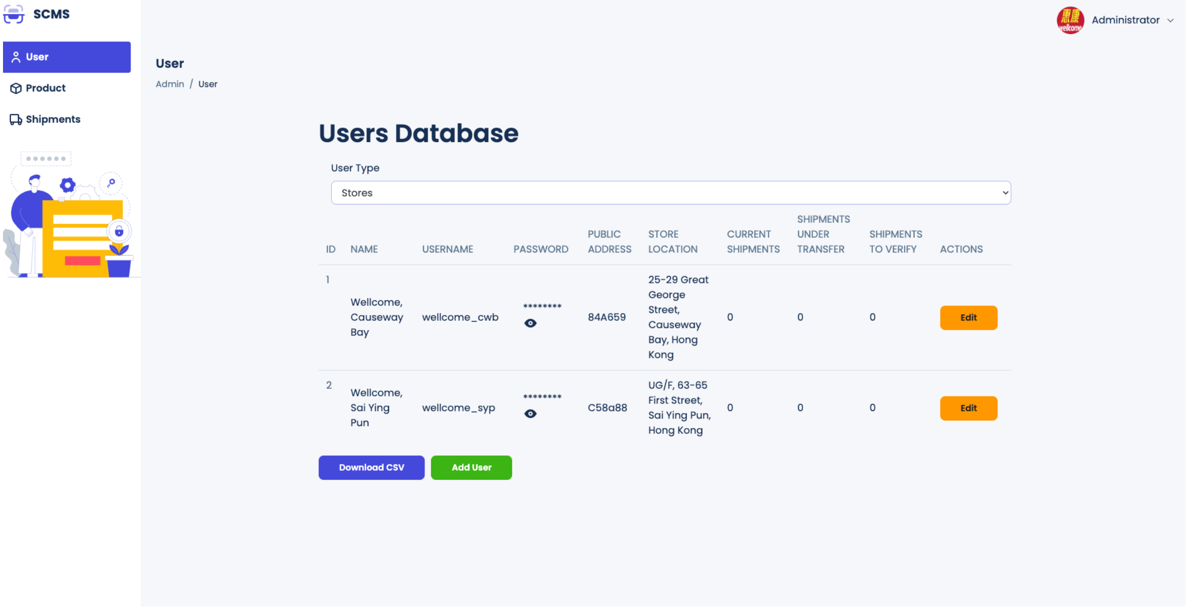
Users Database

User Type: All

ID	NAME	USERNAME	PASSWORD	PUBLIC ADDRESS	STORE LOCATION	CURRENT SHIPMENTS	SHIPMENTS UNDER TRANSFER	SHIPMENTS TO VERIFY	ACTIONS
1	Siddharth Agarwal	agarsid	********	067e38	N/A	0	0	0	Edit
2	Suyash Lohia	sui	********	c60765	N/A	0	0	0	Edit
3	Wellcome, Causeway Bay	wellcome_cwb	********	84A659	25-29 Great George Street, Causeway Bay, Hong Kong	0	0	0	Edit
4	Wellcome, Sai Ying Pun	wellcome_syp	********	C58a88	UG/F, 63-65 First Street, Sai Ying Pun, Hong Kong	0	0	0	Edit

Fig. 30. *Web App Home Page*

For ease of access, the admin can choose to filter the table by employee or store type and gain insight into each entity separately (Fig. 31).



User

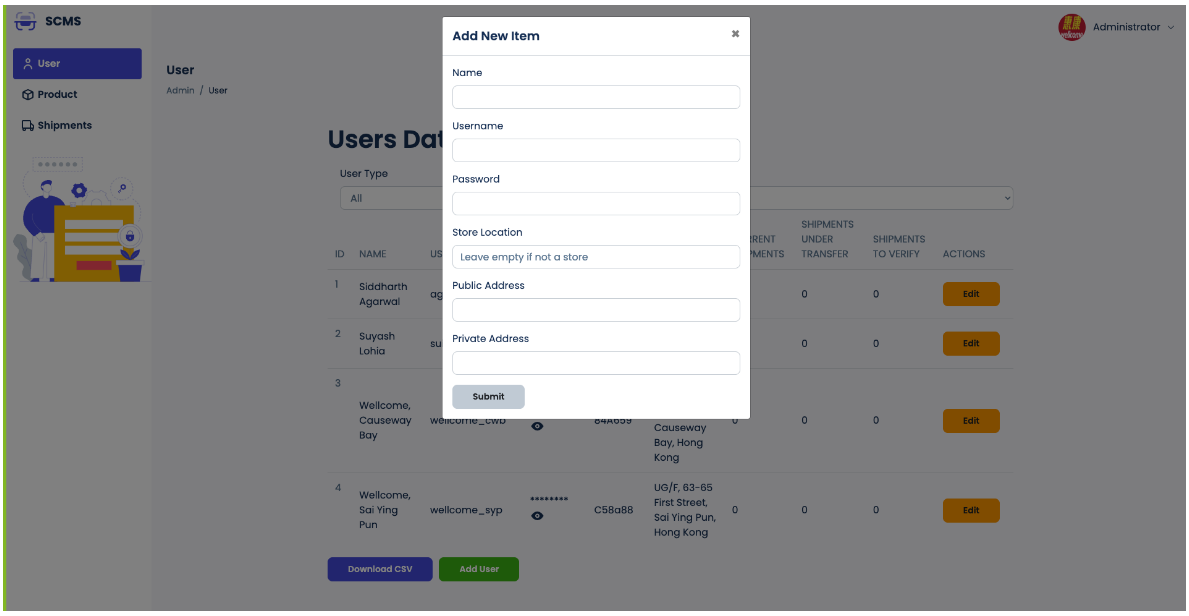
Users Database

User Type: Stores

ID	NAME	USERNAME	PASSWORD	PUBLIC ADDRESS	STORE LOCATION	CURRENT SHIPMENTS	SHIPMENTS UNDER TRANSFER	SHIPMENTS TO VERIFY	ACTIONS
1	Wellcome, Causeway Bay	wellcome_cwb	********	84A659	25-29 Great George Street, Causeway Bay, Hong Kong	0	0	0	Edit
2	Wellcome, Sai Ying Pun	wellcome_syp	********	C58a88	UG/F, 63-65 First Street, Sai Ying Pun, Hong Kong	0	0	0	Edit

Fig. 31. *Web App Users Filter*

Lastly, the admin can press the “add user” button to create a new user (employee or store) and provide the details through a simple modal popup (Fig. 32). Every user is linked to a cryptocurrency wallet by the admin that is used in the creation and transfer of shipments as NFTs. Consequently, the supply chain member has no access to the wallet.



User Type: All

Add New Item

Name

Username

Password

Store Location: Leave empty if not a store

Public Address

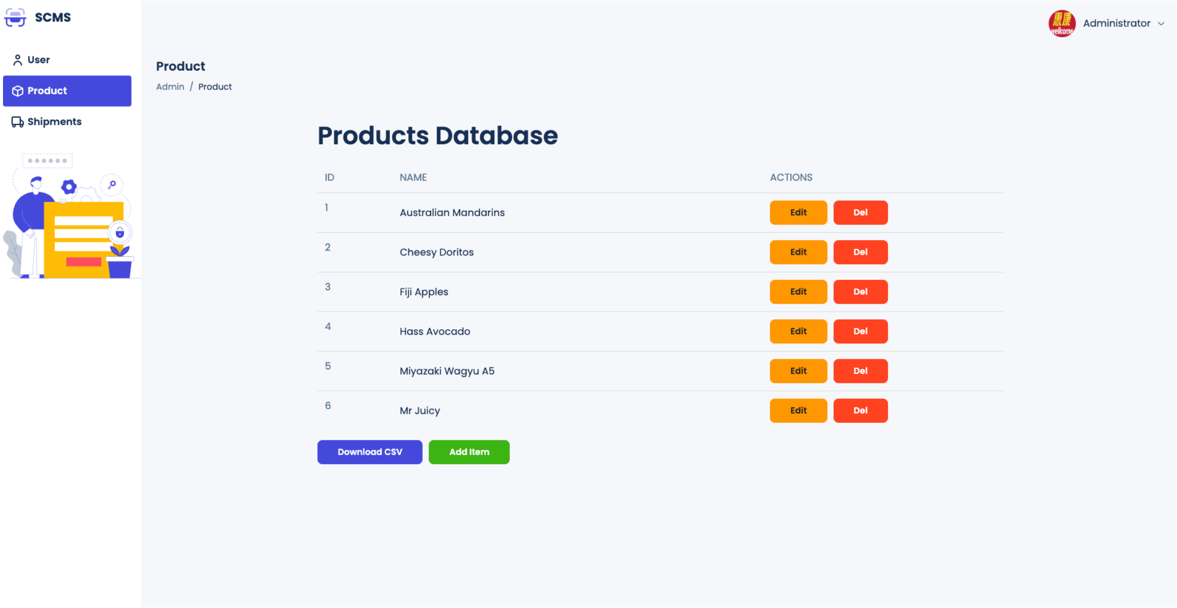
Private Address

Submit

ID	NAME	US				RENT MENTS	SHIPMENTS UNDER TRANSFER	SHIPMENTS TO VERIFY	ACTIONS
1	Siddharth Agarwal	ag					0	0	Edit
2	Suyash Lohia	su					0	0	Edit
3	Wellcome, Causeway Bay	wellcome_cwb		84A659	Causeway Bay, Hong Kong	0	0	0	Edit
4	Wellcome, Sai Ying Pun	wellcome_syp	********	C58a88	UG/F, 63-65 First Street, Sai Ying Pun, Hong Kong	0	0	0	Edit

Fig. 32. Web App Users Add

- **Products Management:** This page contains all the unique products which can be transported in the supply chain. Each new tokenised shipment created can only contain products that are present on this page. The application provides CRUD functionality to the admin for the products page (Fig. 33). All the details are displayed in a table sorted by the product name. The system prevents the admin from adding or editing data that already exists in the database as well.

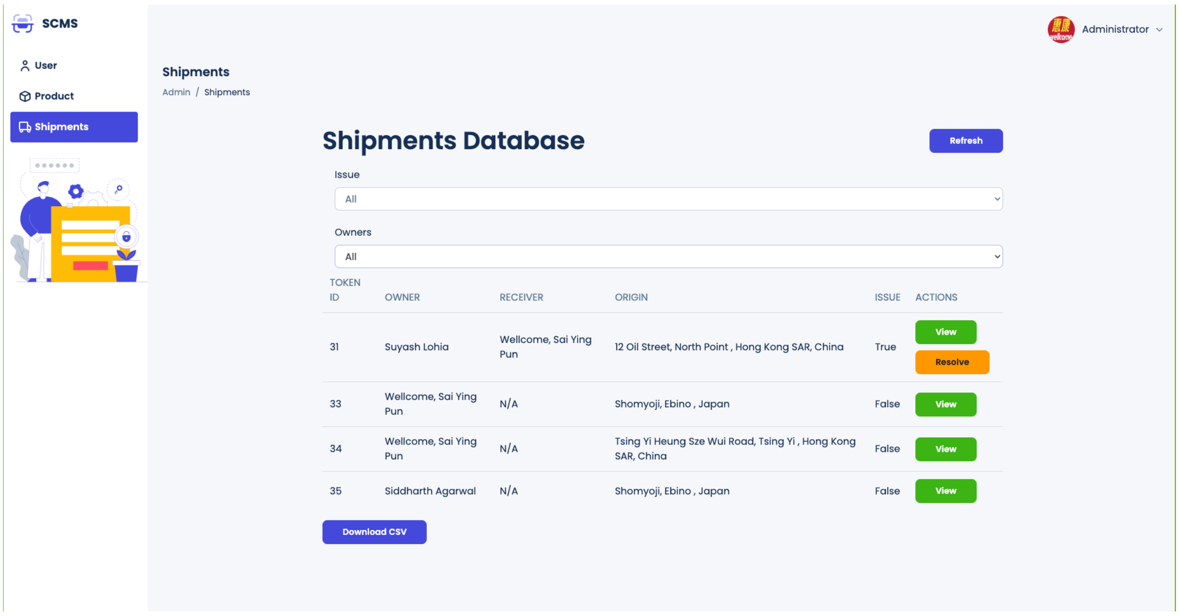


Products Database

ID	NAME	ACTIONS
1	Australian Mandarins	Edit Del
2	Cheesy Doritos	Edit Del
3	Fiji Apples	Edit Del
4	Hass Avocado	Edit Del
5	Miyazaki Wagyu A5	Edit Del
6	Mr Juicy	Edit Del

Fig. 33. Web App Products

- **Shipments:** The shipments page provides supply chain visibility to the admin. It displays a table of shipments sorted by the token id with each row containing information about the sender, receiver and the origin location (Fig. 34). These are not editable since every shipment is linked to an NFT.



Shipments
Admin / Shipments

Shipments Database

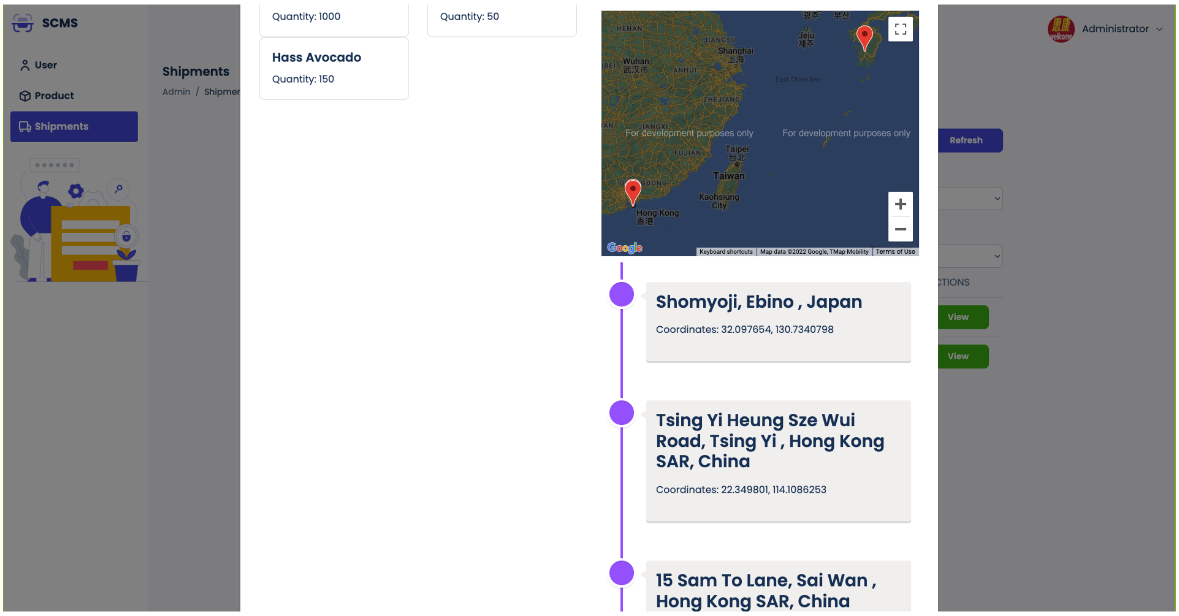
Issue: All

Owners: All

TOKEN ID	OWNER	RECEIVER	ORIGIN	ISSUE	ACTIONS
31	Suyash Lohia	Wellcome, Sai Ying Pun	12 Oil Street, North Point , Hong Kong SAR, China	True	View Resolve
33	Wellcome, Sai Ying Pun	N/A	Shomyoji, Ebino , Japan	False	View
34	Wellcome, Sai Ying Pun	N/A	Tsing Yi Heung Sze Wui Road, Tsing Yi , Hong Kong SAR, China	False	View
35	Siddharth Agarwal	N/A	Shomyoji, Ebino , Japan	False	View

Fig. 34. Web App Shipments

The actions column allows the admin to view further details about the shipment. This "view" button opens a modal that showcases the products in the shipment, location tracing of the shipment (Fig. 35) as well as the ownership history.



Shipments
Admin / Shipments

Quantity: 1000

Quantity: 50

Hass Avocado
Quantity: 150

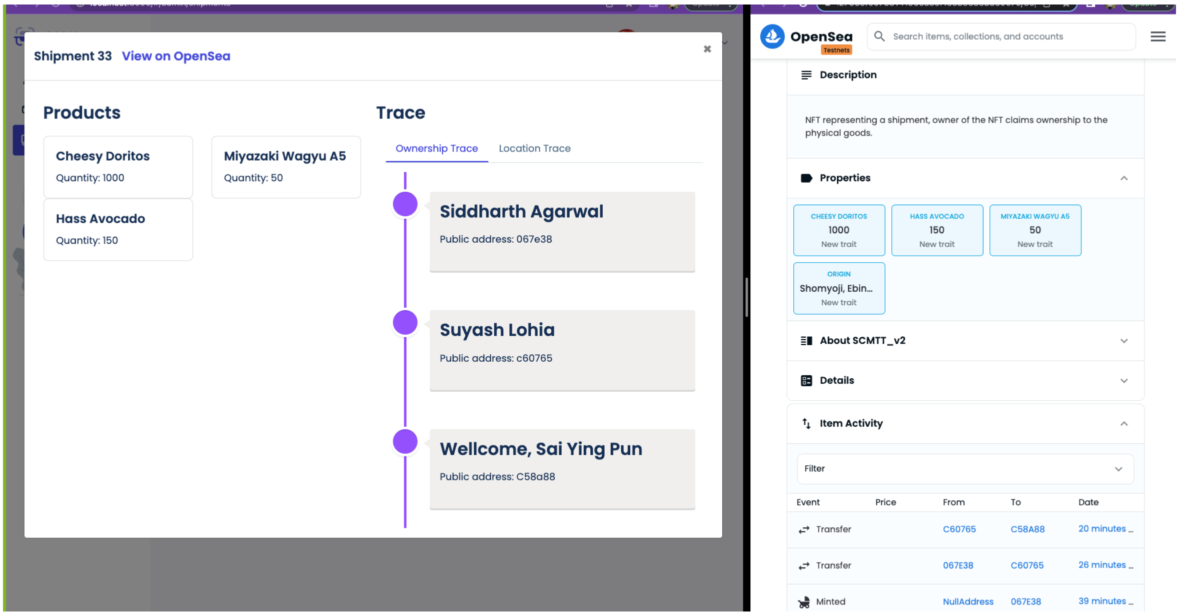
Shomyoji, Ebino , Japan
Coordinates: 32.097654, 130.7340798

Tsing Yi Heung Sze Wui Road, Tsing Yi , Hong Kong SAR, China
Coordinates: 22.349801, 114.1086253

15 Sam To Lane, Sai Wan , Hong Kong SAR, China

Fig. 35. Web App Shipments Modal Location

As aforementioned, each shipment is linked to an NFT. The ownership history displayed in the shipment modal can also be seen on OpenSea by simply clicking the link present in the modal. Apart from the products and origin of the shipment, the entire transfer process is also visible on OpenSea which shows the public addresses of the users (Fig. 36).



Cheesy Doritos
Quantity: 1000

Miyazaki Wagyu A5
Quantity: 50

Hass Avocado
Quantity: 150

Siddharth Agarwal
Public address: 067e38

Suyash Lohia
Public address: c60765

Wellcome, Sai Ying Pun
Public address: C58a88

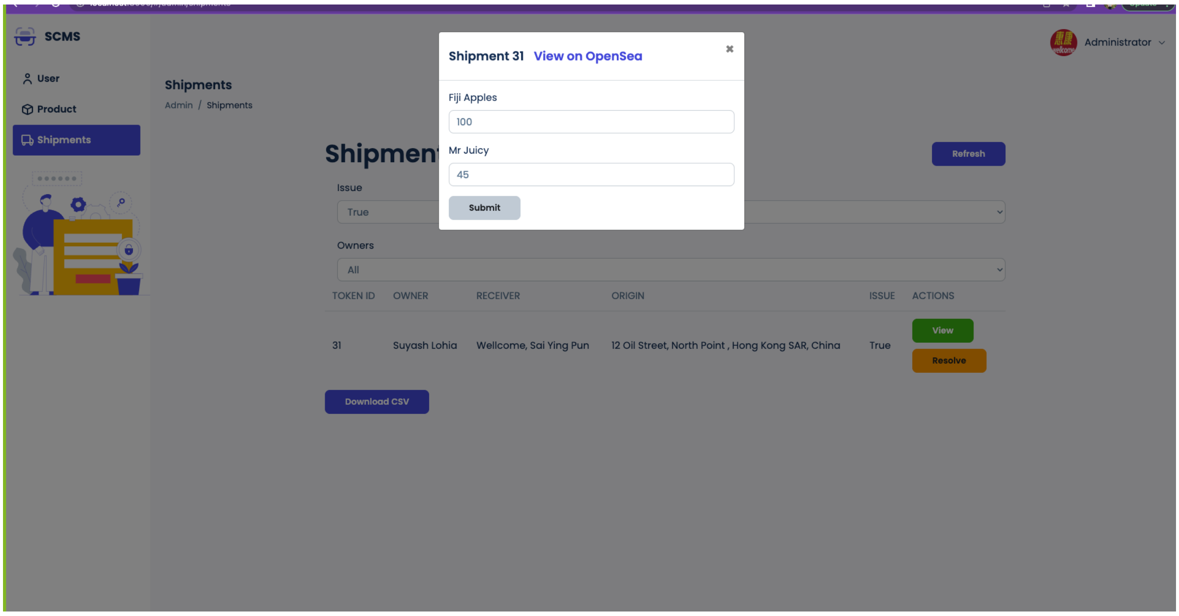
NFT representing a shipment, owner of the NFT claims ownership to the physical goods.

CHEESY DORITOS 1000 New trait
HASS AVOCADO 150 New trait
MIYAZAKI WAGYU A5 50 New trait
ORIGIN Shomyoji, Ebin... New trait

Event	Price	From	To	Date
Transfer		C60765	C58A88	20 minutes ...
Transfer		067E38	C60765	26 minutes ...
Minted		NullAddress	067E38	39 minutes ...

Fig. 36 Web App Shipments Modal Ownership (left); Shipment on OpenSea (right)

Additionally, the admin can filter the shipments table based on owners or shipments having issues. Shipments with issues will undergo manual inspection. If there has been a loss of goods and it has been inspected, the admin can choose to resolve that particular shipment by reducing the quantity of the relevant products (Fig. 37).



Shipment 31 View on OpenSea

Fiji Apples: 100

Mr Juicy: 45

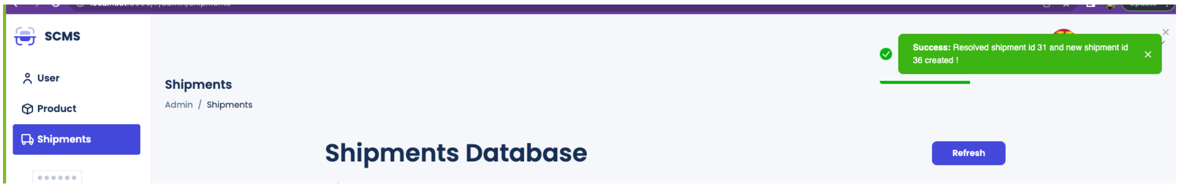
Issue: True

Owners: All

TOKEN ID	OWNER	RECEIVER	ORIGIN	ISSUE	ACTIONS
31	Suyash Lohia	Wellcome, Sai Ying Pun	12 Oil Street, North Point , Hong Kong SAR, China	True	View Resolve

Fig. 37. Web App Shipments Resolution Modal

Having changed the quantity of the products, a new NFT is minted for the modified shipment which contains an attribute linked to the old shipment token (Fig. 38). The old shipment is removed from the database and the new shipment is assigned to the owner.



Shipments

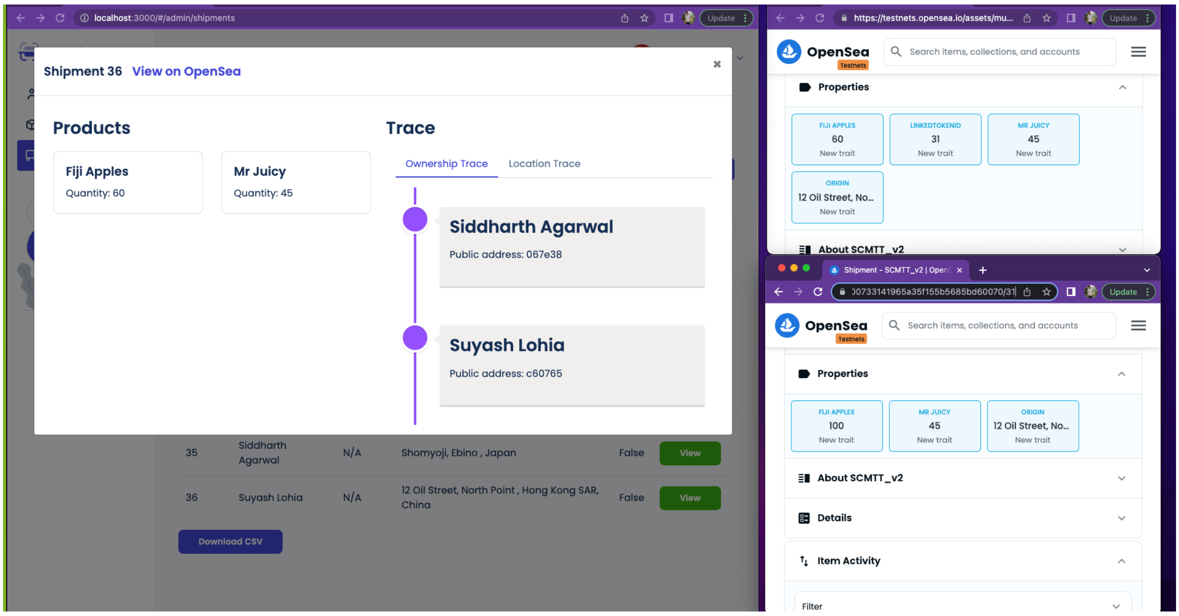
Admin / Shipments

Success: Resolved shipment id 31 and new shipment id 36 created !

Shipments Database

Fig. 38. Web App Shipments Resolution Notification

The new shipment can be inspected on OpenSea to view the changes and linkage with the previous shipment (Fig. 39). The left column displays the new shipment created. The right column displays two NFTs on OpenSea. The top one is of the new shipment which contains the token ID of the NFT displayed at the bottom. The quantity of apples was changed from 100 to 60.


localhost:3000/#/admin/shipments
Update
View on OpenSea
Ownership Trace
Location Trace
View
View
Download CSV
https://testnets.opensea.io/assets/mu...
OpenSea
Testnets
Search items, collections, and accounts
Properties
About SCMTT_v2
Shipment - SCMTT_v2 | OpenS
ↄ0733141965a35f155b5685bd60070/31
Details
Item Activity
Filter

Shipment 36

Products

Fiji Apples
Quantity: 60

Mr Juicy
Quantity: 45

Trace

Siddharth Agarwal
Public address: 067e38

Suyash Lohia
Public address: c60765

35	Siddharth Agarwal	N/A	Shomyoji, Ebino , Japan	False
36	Suyash Lohia	N/A	12 Oil Street, North Point , Hong Kong SAR, China	False

FIJI APPLES	LINKEDTOKENID	MR JUICY	ORIGIN
60	31	45	12 Oil Street, No...
New trait	New trait	New trait	New trait

FIJI APPLES	MR JUICY	ORIGIN
100	45	12 Oil Street, No...
New trait	New trait	New trait

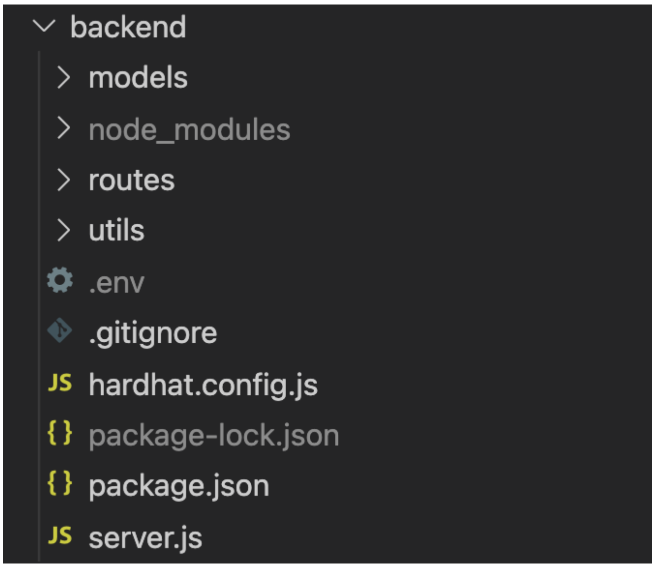
Fig. 39. Web App Shipments Resolution Result

4.2.4. Backend Development

As mentioned in the methodology the backend was developed using NodeJS and Express to connect with the blockchain along with providing a robust API to the mobile and web application.

4.2.4.1. Project Structure

The backend developed using the Express framework uses a modularised structure for better development (Fig. 40). The 'server.js' file sets up the process and connects with the database. The 'models' folder contains the schema of the database written in "mongoose" since MongoDB is the chosen database. Lastly, the 'routes' folder houses different routes for all API calls made from the frontend.



```
∨ backend
  > models
  > node_modules
  > routes
  > utils
    .env
    .gitignore
  JS hardhat.config.js
  {} package-lock.json
  {} package.json
  JS server.js
```

Fig. 40. Backend Project Structure

4.2.4.2. RESTful API

The mobile and web applications require an API to provide functionality to all the users. The NFT creation and transfer are also initiated through API calls to the backend. As a result, the Express routing was used for setting up the API for different use cases. There are three major use cases for which different endpoints have been created: Products data (Table 2); User data (Table 3); Shipments data of every user (Table 4). Postman was used for testing the routes during the backend development process.

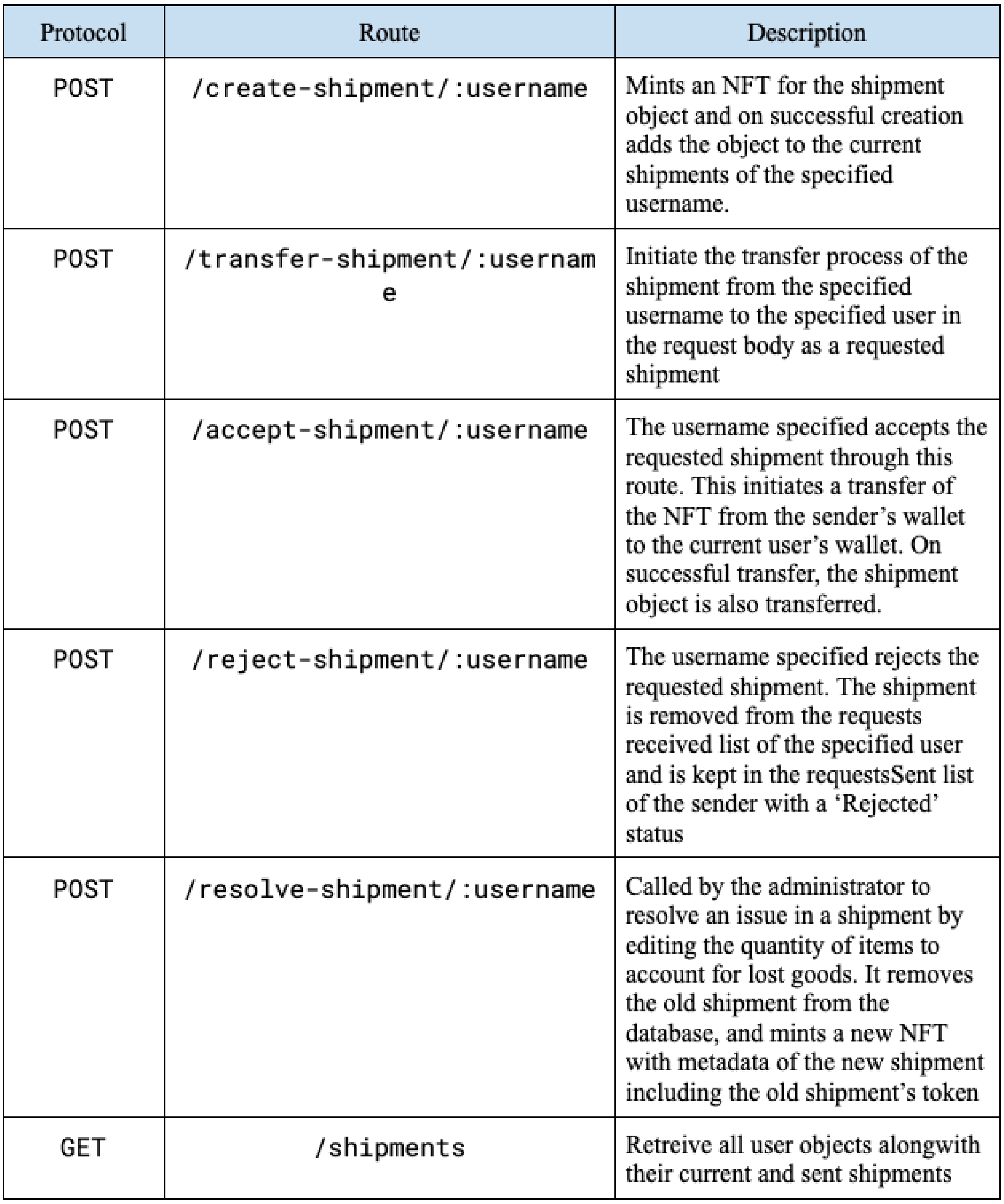
Table 2. Products Endpoints

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. Users Endpoints

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. Shipments Endpoints

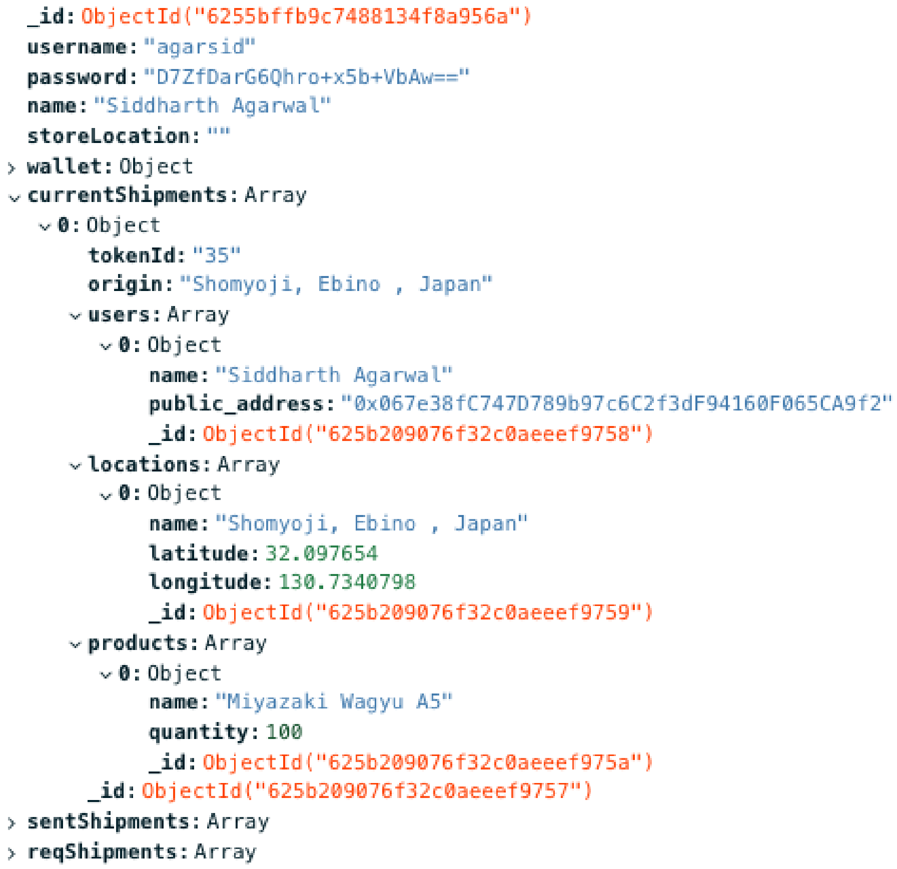


Protocol	Route	Description
POST	/create-shipment/:username	Mints an NFT for the shipment object and on successful creation adds the object to the current shipments of the specified username.
POST	/transfer-shipment/:username	Initiate the transfer process of the shipment from the specified username to 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

MongoDB Atlas, a cloud MongoDB solution serves as the database provider. It provides access to an easy to use interface and fast connection with Express. Every user in the system contains a variety of information. Each user's shipments can increase or decrease. Consequently, flexible and scalable models were required in the database.

Two models have been created on the database. The ‘products’ model contains a list of the names of all products in a supply chain. This model is the one with CRUD functionality on the web dashboard and all the new tokenised shipments can only contain products that exist in the database. The ‘users’ model maintains user data, cryptocurrency wallet details, and details of all shipments in hand or in the transfer process (Fig. 41). Each of them contains every product's name and quantity. The ‘sentShipments’ are shipments pending to be approved by the receiving user. The ‘reqShipments’ refers to the shipments that the current user needs to either accept or reject and continue the transfer process.



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

Fig. 41. Database User Model

4.2.6. Codebase, Git and GitHub

Owing to the number of components of the platform, the entire development was supported through git for version control and the repository was hosted on GitHub. The repository consisted of 4 folders, one each for the backend, mobile application, web application and the blockchain. Different branches were maintained for development and testing of each aspect. Once each component was functional, the entire process was tested again and then pushed to the main branch.

5. Discussion

This section discusses the limitations and challenges faced by the application and lays out the future plans for the project.

5.1. Limitations

The team has identified the following major limitations in the platform:

5.1.1. Requirement of Smartphone and Internet

The mobile application for both consumers and supply chain members requires an internet connection for the functionality of creating, transferring, accepting and viewing the shipments and products. Consequently, an internet connection and a smartphone become a requirement for all supply chain members for the platform to work. This limits widespread adoption of the platform, especially in third world and developing countries which have less access to the same.

5.1.2. Handling loss in the quality and quantity of Goods

The current system allows the receiver to reject a shipment if there is any issue in it. The company management can manually inspect the issue further and resolve it by changing the quantity of each product in that shipment. This also leads to the creation of a new token assigned to the sender of the original shipment. This however limits the product in the following ways:

- Firstly, the system can only inform the company management of an issue in the shipment. It does not know if the number of products has been reduced due to actual loss in quantity or loss due to degradation of quality. This could be due to poor management by the supply chain member or external factors such as the weather.
- Secondly, once the old shipment is resolved, the new tokenised shipment does not have the entire ownership history on the blockchain or in its metadata. To tackle this, the ownership history is also stored on the database and the old shipment's token ID is added to the metadata of the new shipment. However, this is not the best practice and goes against the concept of traceability through NFT ownership. In the future, the team hopes this could be resolved by the introduction of fractional ownership in NFTs as proposed by Ethereum.

5.1.3. Environmental Impact

Both the creation and transfer of NFTs require electricity thereby impacting the carbon footprint. Although the team has tried to reduce this limitation by switching to a layer 2 blockchain, the environmental impact is still greater than that of other centralised solutions. With blockchain networks like Ethereum working towards reducing their carbon footprint, the team hopes to mitigate this limitation further in the future.

5.2. Challenges

The team has addressed the following challenges during development to present a cohesive platform:

5.2.1. Human Error

Even though every shipment is tokenised on the blockchain, the data is fed to it by users of the mobile and web application. The administrator inputs the important wallet information of users and creates the products on the dashboard. The mobile application will be used by multiple supply chain members for creating and transferring shipments. Hence, there was a need to make the application user-friendly. Consequently, the blockchain connection is kept independent of the mobile application. For the user, it is an uncomplicated mobile application that allows them to review their shipment, add or delete products and transfer shipments only to the available dropdown list. The administrator also cannot create repetitive products or users. Additionally, error handling is conducted both on the backend as well as the frontend level of the application to show informational alerts to the users.

5.2.2. Blockchain: TPS and Gas Fee

The platform requires a large number of transactions on the blockchain due to the creation and transfer of tokenised shipments by multiple users. Initially, the team had decided to work on the Ethereum blockchain. However, it has a high transaction fee and the TPS was very low as compared to centralised solutions. Through further research, this was solved by using Polygon, a layer 2 blockchain built on Ethereum. It provides the advantages of Ethereum at a high TPS and minimal transaction fee in its own token MATIC.

5.2.3. Tracing homogenous products

From a technical perspective, tracing homogenous products in a store arriving from different supply chains was the biggest challenge. This creates a problem for both consumers and businesses and can lead to the misrepresentation of information. Currently, the system stores the origin of each shipment in the NFT metadata and the mobile application accordingly separates homogenous products. Hence, different supply chains for homogenous products can be displayed. Despite this, the end consumer needs to be able to differentiate between the same product kept on the same shelf and view its origin in the application. Moreover, a product could have the same origin but different supply chains. To resolve this challenge, the team proposes the use of QR codes on every product, with each product linked to the tokenised shipment's ID.

5.3. Future Plan

Apart from the current objective and deliverables, the team believes in a wide scope of the application by solving challenges and improving the platform.

5.3.1. Supporting QR codes in the application

As stated in the challenges, homogenous goods tracing can be improved using QR codes. The team plans that every product will have a QR code that is linked to the token ID of the shipment. This code will be attached after the shipment has been tokenised by the system. As a result, every product will be unique and its entire ownership and location history can be traced. To support this, the mobile application will need to support QR code scanning. The backend will extract the token ID from the QR code, retrieve the product's details and display them to the consumer. Even the supply chain members like employees and stores can use this feature.

5.3.2. Supply Chain Analytics

The team plans to provide smart analytics of different supply chains to the company administration. These include analytics on the performance of each supply chain and of each supply chain member to identify the pain points. Furthermore, analytics could highlight the products which are lost the most and at what location in the supply chain. Correlational analysis could be performed on the loss of goods against external conditions like humidity and temperature during different times of the year. This could help narrow down if there has been a loss of goods due to poor management or external factors.

5.3.3. ERP systems and SaaS

Integrating the supply chain management platform with existing ERP solutions to provide accounting, inventory management, and additional smart analytics could provide a one-stop solution whilst also decentralising the existing ERP solutions The team’s vision is to provide the same as a SaaS (Software As A Service) platform which can help revolutionize the supply chain.

6. Conclusion

With ever-growing interconnectivity, businesses have had a massive outreach resulting in a severe impact on the supply chain. SCM has developed into an industry worth billions of dollars that needs to improve every day to handle the rising complexity. Consumers also demand supply chain transparency from companies to gain more information about the products. Hence, a successful SCM leads to customer satisfaction and a better financial position for firms. Current solutions are either tedious or lack traceability or both. This report presents a potential solution to this problem in the form of a blockchain-powered platform.

The application aims to provide transparency and traceability in the supply chain through the use of NFTs and blockchain. The web dashboard, accessible solely by the company administrator, is used to create a products and users database. Every supply chain member is linked to a cryptocurrency wallet which is essential for the server to mint and transfer NFTs. In the supply chain, the first member selects the products to be transferred through the mobile application. These are tokenised into NFTs and stored on the blockchain. Every successive member will transfer or accept the NFT through the application, thereby relieving or taking ownership of the goods. The business management can review and track each step through the web application which can detect any inconsistency. Once it reaches the final store, the consumers can view all the products in that store and trace the origin of each product. Polygon, a layer 2 blockchain built on top of Ethereum, is used for its compatibility with NFTs, low transaction fee, and a high TPS while the MERN stack is used for smooth internal and external integration for all applications.

The project's major limitation is widespread adoption by supply chain members since it requires them to use the internet and a smartphone. Additionally, the creation and transfer of NFTs impact the carbon footprint which has been reduced by switching from a level 1 to a level 2 blockchain. Lastly, the traceability of homogeneous goods at a store arriving through different supply chains presents a challenge.

Owing to the rising demand for technologically advanced SCM, future work will involve mitigating the challenges by introducing QR codes linked to the tokenised shipment on every product. Additionally, smart analytics and integration with ERP systems would provide a one-stop solution in a trusted and decentralised manner.

7. References

[1] “How globalization impacts supply chain management,” *Flowspace Blog*, 23-Sep-2021. [Online]. Available: https://www.flow.space/blog/globalization-supply-chain/. [Accessed: 27-Oct-2021].

[2] J. T. Mentzer et al, "Defining supply chain management," Journal of Business Logistics, vol. 22, (2), pp. 1-26, 2001. Available: http://eproxy.lib.hku.hk/login?url=https://www.proquest.com/scholarly-journals/defining-supply-chain-management/docview/212663354/se-2?accountid=14548.

[3] “The Importance of Supply Chain Management,” *The importance of Supply Chain Management*. [Online]. Available: https://cscmp.org/CSCMP/Develop/Starting_Your_Career/Importance_of_Supply_Chain_Management.aspx. [Accessed: 27-Oct-2021].

[4] D. M. Lambert and M. C. Cooper, “Issues in Supply Chain Management,” *Industrial Marketing Management*, 24-Jan-2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019850199001133. [Accessed: 27-Oct-2021].

[5] A. Bateman and L. Bonanni, “What supply chain transparency really means,” Harvard Business Review, 20-Jan-2021. [Online]. Available: https://hbr.org/2019/08/what-supply-chain-transparency-really-means. [Accessed: 17-Apr-2022].

[6] B. Cooksey, “Why is supply chain visibility so important?,” *C.H. Robinson Blog*, 08-Dec-2017. [Online]. Available: https://www.chrobinson.com/blog/why-is-supply-chain-visibility-so-important/. [Accessed: 27-Oct-2021].

[7] E. Mazareanu, “Topic: Global Supply Chain Management,” Statista. [Online]. Available: https://www.statista.com/topics/7150/global-supply-chain-management/. [Accessed: 03-Oct-2021].

[8] Y. Lemma, “Loss in perishable food supply chain: An Optimization Approach Literature Review,” Academia.edu. [Online]. Available: https://www.academia.edu/45463716/Loss_in_Perishable_Food_Supply_Chain_An_Optimization_Approach_Literature_Review. [Accessed: 03-Oct-2021].

[9] T. K. Sharma, “Is blockchain comparable to ‘the new internet’?,” *Is Blockchain comparable to "the new internet"? | Blockchain Council*, 10-Sep-2018. [Online]. Available: https://www.blockchain-council.org/blockchain/is-blockchain-comparable-new-internet/. [Accessed: 27-Oct-2021].

[10] C. Miles, “Blockchain security: What keeps your transaction data safe?,” *IBM Supply Chain and Blockchain Blog*, 18-Dec-2020. [Online]. Available: https://www.ibm.com/blogs/blockchain/2017/12/blockchain-security-what-keeps-your-transaction-data-safe/. [Accessed: 27-Oct-2021].

[11] “The anatomy of a block in a Blockchain,” *Burr & Forman*, 17-May-2018. [Online]. Available: https://www.burr.com/blogs/blockchain-law/2017/11/07/the-anatomy-of-a-block-in-a-blockchain/. [Accessed: 27-Oct-2021].

[12] D. Khan, L. T. Jung, and M. A. Hashmani, “Systematic literature review of challenges in blockchain scalability,” MDPI, 09-Oct-2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/20/9372. [Accessed: 18-Apr-2022].

[13] D. Geroni, “Blockchain scalability problem - why is it difficult to scale blockchain,” 101 Blockchains, 30-Sep-2021. [Online]. Available: https://101blockchains.com/blockchain-scalability-challenges/. [Accessed: 18-Apr-2022].

[14] “Blockchain technology: Layer-1 and layer-2 networks,” Cryptopedia, 09-Mar-2022. [Online]. Available:

https://www.gemini.com/cryptopedia/blockchain-layer-2-network-layer-1-network#section-layer-2-scaling-solutions. [Accessed: 18-Apr-2022].

[15] A. W. 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].

[16] A. Armaan, “NFT: Explained in simple language‐non-fungible tokens,” *Medium*, 19-Mar-2021. [Online]. Available: https://medium.datadriveninvestor.com/nft-explained-in-simple-language-non-fungible-tokens-7996a9c605ff. [Accessed: 27-Oct-2021].

[17] “Non-fungible tokens (NFT),” *ethereum.org*. [Online]. Available: https://ethereum.org/en/nft/. [Accessed: 27-Oct-2021].

[18] M. Valenta, “The anatomy of an NFT‐and questionable real-world implementations,” Medium, 25-Apr-2021. [Online]. Available: https://thevalenta.medium.com/the-anatomy-of-an-nft-and-questionable-real-world-implementations-d93a0a18002b. [Accessed: 18-Apr-2022].

[19] S. Rowlands, “HTTP or ipfs - centralised vs decentralised server,” Novoda Insights, 23-Feb-2021. [Online]. Available: https://blog.novoda.com/how-ipfs-is-taking-on-http/. [Accessed: 18-Apr-2022].

[20] 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].

[21] Harvard Business Review, “Blockchain and ERP systems: Future of Supply Chain Management,” *Atom Insights | Blockchain And ERP Systems: Future Of Supply Chain Management*. [Online]. Available:

https://www.atominc.com/post/blockchain-and-erp-systems-the-integrated-future-of-supply-chain-management. [Accessed: 27-Oct-2021].

[22] V. Gaur and A. Gaiha, “Building a transparent supply chain,” *Harvard Business Review*, 14-Apr-2020. [Online]. Available: https://hbr.org/2020/05/building-a-transparent-supply-chain. [Accessed: 27-Oct-2021].

[23] 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].

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

[25] A. S. Rawat, “What is Polygon (MATIC)? features and advantages,” Analytics Steps, 16-Feb-2022. [Online]. Available: https://www.analyticssteps.com/blogs/what-polygon-matic-features-and-advantages. [Accessed: 18-Apr-2022].

[26] “Ethereum development environment for professionals by Nomic Foundation,” Hardhat. [Online]. Available: https://hardhat.org/getting-started/. [Accessed: 18-Apr-2022].

[27] E. Jones, “How to deploy NFT smart contracts,” Medium, 21-Jan-2022. [Online]. Available: https://betterprogramming.pub/how-to-deploy-nft-smart-contracts-9271ce5e91c0. [Accessed: 18-Apr-2022].

[28] “What is express.js?: Why should use express.js?: Features of express.js,” Besant Technologies, 17-Sep-2020. [Online]. Available: https://www.besanttechnologies.com/what-is-expressjs#:~:text=It%20is%20used%20for%20designing,and%20API%20without%20any%20effort. [Accessed: 18-Apr-2022].

[29] “Express/node introduction,” MDN, 12-Apr-2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction. [Accessed: 18-Apr-2022].

[30] “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].

[31] J. Grajcar, “Why use react native for your mobile app?,” Python Development Company, 17-Mar-2022. [Online]. Available: https://www.stxnext.com/blog/why-use-react-native-your-mobile-app/#:~:text=React%20Native%20is%20great%20for,compromise%20on%20quality%20and%20functionality. [Accessed: 18-Apr-2022].

[32] N. Pandit, “What and why react.js,” C# Corner, 10-Feb-2021. [Online]. Available: https://www.c-sharpcorner.com/article/what-and-why-reactjs/#:~:text=React%20allows%20developers%20to%20create,view%20in%20the%20MVC%20template. [Accessed: 18-Apr-2022].