University of Wollongong

Faculty of Engineering and Information Sciences
School of Computing and Information Technology

CSIT321
Group Project: Blockchain-based Traceable Anti-counterfeiting System

Assignment 5: Final Submission

Group 7

Letter of Transmittal

***Date**: October 11, 2024*
***To**:*
Dr. John Le
Yudi Zhang
University of Wollongong
Northfields Avenue
Wollongong NSW 2500

Dear Dr. John Le,

*We are pleased to formally submit the report for CSIT321 Assignment 5, **Final Product and Documentation**, by Group 7 at the University of Wollongong. This report provides a comprehensive overview of our project, the **Blockchain-based Traceable Anti-counterfeiting System**, detailing its development and final outcomes.*

The report covers the following key areas:

- ***System Requirements***
- ***Project Summary***
- ***System Design***
- ***Project Closeout***
- ***System Installation Process***

This submission marks the culmination of our development efforts and presents a clear evaluation of how the system meets the original project objectives.

Please direct any questions regarding this report to the document's authors and researchers. We hope you find this report satisfactory.

Sincerely,
Group 7
Hein Khant Zaw (7692870)
Nay Oo Lwin (7691713)
Joel Joseph (7447218)
Alwin Basil Shah (7451775)

EtherWeave Technical Documentation Report

Acknowledgment

We would like to express our gratitude to our supervisor, Yudi Zhang and subject coordinator Dr. John Le for their invaluable guidance and support throughout this project. We are also thankful for the support of various colleagues, and fellow students who provided insightful feedback during development.

Introduction

Project Overview

The **Etherweave** project, developed by Group 7 at the University of Wollongong, is a blockchain-based solution to address the growing global issue of product counterfeiting. Counterfeiting has profound effects on industries worldwide, leading to billions in financial losses, reputational damage, and even safety risks due to counterfeit pharmaceuticals or electronic goods in the market.

Etherweave leverages the immutable and transparent nature of blockchain technology to provide a tamper-proof system where the authenticity of a product can be verified at every stage of the supply chain. From the manufacturer to the end consumer, all transactions and ownership transfers are recorded securely, making it nearly impossible to introduce counterfeit products into the system.

Objective

The primary objective of Etherweave is to develop a decentralised, tamper-proof ledger that logs transactions and ownership transfers for products. By using the Ethereum blockchain and smart contracts, Etherweave ensures that product data remains secure, immutable, and verifiable. Additionally, the system offers a user-friendly interface, enabling manufacturers, suppliers, distributors, and consumers to interact with the blockchain, verify products, and track their authenticity in real-time.

Scope

Etherweave addresses multiple critical challenges across various industries, focusing on combating counterfeiting and ensuring product authenticity through blockchain technology.

Targeted Industries:

Etherweave will initially focus on industries where counterfeiting poses a significant threat to both businesses and consumers. These include:

- **Luxury Goods:** Counterfeiting in the luxury sector affects brand reputation and customer trust. Etherweave enables brands to register unique product data on the blockchain, providing consumers with an easy method to verify authenticity, whether it's high-end fashion, watches, or other exclusive items.
- **Pharmaceuticals:** Fake medications not only cost the industry billions but also pose severe health risks to consumers. Etherweave helps pharmaceutical companies track the journey of their products from manufacturing to the pharmacy, reducing the risk of counterfeit drugs entering the supply chain.
- **Electronics:** Electronics are often duplicated or tampered with, especially in global markets. Etherweave offers real-time tracking for electronic goods, ensuring that consumers and retailers can verify a product’s origin and authenticity with confidence.

Scalability:

Etherweave is built with scalability in mind to ensure that it can accommodate the evolving needs of growing industries. As transaction volumes increase with market adoption, Etherweave’s infrastructure can handle:

- **Large Transaction Volumes:** The platform is capable of processing large-scale data transactions, including product registration, updates, and tracking logs, while maintaining high efficiency and performance.
- **Real-Time Data Access:** Users across the supply chain, including manufacturers, retailers, and consumers, will have real-time access to product data, enabling instant verification of product authenticity and movement. This improves trust and transparency across all parties involved in the supply chain.

User Interfaces:

To ensure broad adoption and ease of use, Etherweave is developing accessible and user-friendly platforms. This includes:

- **Web-Based Application:** A responsive web interface designed for manufacturers, distributors, and retailers to register products on the blockchain, manage inventory, and monitor supply chain data in real-time.
- **Mobile:** The mobile platform is aimed at both businesses and consumers. Business users can track product movements and verify authenticity on the go, while consumers can scan QR codes or RFID tags to check product legitimacy before purchasing.
- **Product Registration and Tracking:** Through QR codes, RFID tags, and other identification technologies, Etherweave integrates these into the blockchain system. This allows users to scan a product and instantly access its history, including manufacturing details, ownership transfers, and movement across the supply chain.

System Requirements

Hardware Requirements:

To ensure Etherweave operates efficiently across its various components, from blockchain interactions to mobile applications, specific hardware specifications are necessary:

- ➔ **Modern Computer:**
  - ◆ **Processor:** An Intel i5 processor or higher (or AMD equivalent) is recommended to handle the computational needs of interacting with the Ethereum blockchain and running smart contracts. This processor will also efficiently manage local development environments and simulation tasks during testing.
  - ◆ **RAM:** A minimum of **8 GB of RAM** is essential for managing multiple blockchain processes simultaneously. This includes running a full node, working with development environments like Hardhat and Ganache, and efficiently handling smart contract deployment, execution, and product data logging.
  - ◆ **Storage:** At least **500 GB of storage** is required, primarily to store blockchain node data, local development files, product tracking logs, and interactions with smart contracts. Blockchain nodes can grow significantly in size as more transactions are processed, making ample storage necessary.
  - ◆ **Reliable Internet Connection:** Etherweave requires stable and fast internet access to connect with blockchain nodes, cloud services (such as AWS), and Ethereum test networks. Real-time interaction with the blockchain is vital for product verification and tracking.

- ➔ **Smartphone Devices:**
  - ◆ **Mobile Scanning Capabilities:** Mobile users must have smartphones equipped with **QR code** and **NFC scanning capabilities**. These technologies enable users to interact with products registered on the blockchain, verifying authenticity and viewing product data by scanning embedded codes or tags.
  - ◆ For example, consumers can scan a product's QR code or use NFC-enabled devices to retrieve product information instantly from the blockchain, ensuring the product's authenticity before purchase.

Software Requirements:

Etherweave leverages a combination of blockchain and cloud technologies to manage product registration, tracking, and verification. The following software technologies form the backbone of the system:

- **Ethereum Blockchain:**

The **Ethereum blockchain** serves as the foundation for Etherweave's decentralised platform. It is used to store immutable product records, ensuring that data related to product creation, registration, and transfer is transparent and cannot be altered. By leveraging Ethereum's decentralised nature, the system ensures that no single entity can manipulate the data, increasing trust across the supply chain.

- **Solidity:**

Solidity is the primary programming language for developing the smart contracts that power Etherweave. These smart contracts manage the core functions of product lifecycle management, including:

- ➔ Product creation and registration.
- ➔ Ownership transfer between supply chain entities (manufacturers, suppliers, retailers).
- ➔ Verification of authenticity by consumers. Solidity's object-oriented nature and compatibility with Ethereum make it ideal for securely executing these processes on the blockchain.

- **Hardhat:**

Hard Hat is a flexible development environment designed for Ethereum smart contract development. It simplifies tasks such as compiling Solidity code, running local blockchain simulations, testing contract functionality, and deploying contracts to the Ethereum network. Etherweave developers use Hardhat to build, test, and refine the system's smart contracts before deployment to live blockchain environments. Key features include debugging tools and network management, allowing developers to interact seamlessly with the Ethereum blockchain.

- **Ganache:**

Ganache is a personal blockchain used for local development, enabling developers to simulate the Ethereum network on their machines. Ganache is essential for testing smart contracts in a controlled environment before pushing them to the live blockchain. Etherweave uses Ganache to:

  - Simulate product registration and verification processes.
  - Test the integrity and functionality of smart contracts.
  - Identify potential vulnerabilities or bugs in the contract logic before public deployment.

- **Visual Studio Code (VSCode):**

  VSCode is the primary code editor used for Etherweave development. It offers robust support for Solidity and Ethereum development through extensions that integrate smart contract coding, testing, and debugging. Developers use VSCode to write, compile, and deploy smart contracts, as well as to build the web-based and mobile applications for Etherweave's user interfaces.

- **PostgreSQL:**

  PostgreSQL serves as the primary database for Etherweave, complementing the blockchain by storing essential auxiliary data such as user credentials, session data, and access logs, while product-related information remains securely stored on the Ethereum blockchain.

Security Requirements

Security is a paramount concern for Etherweave, given the sensitive nature of product data and the high stakes of ensuring authenticity.

End-to-End Encryption: To protect communications between the blockchain and Etherweave's users, **end-to-end encryption** is enforced across the system. This ensures that all data exchanges, whether they are related to product registration, ownership transfers, or verification, are encrypted and protected from external threats such as data interception or man-in-the-middle attacks.

Role-Based Access Control (RBAC): Etherweave implements **Role-Based Access Control (RBAC)** to manage user permissions based on their role in the supply chain. The system defines distinct roles for manufacturers, suppliers, retailers, and consumers, ensuring that only authorised users can view or modify specific data. For example:

- **Manufacturers** can register new products and track their distribution.
- **Suppliers** can update product movement records.
- **Retailers** can verify product authenticity and track sales.
- **Consumers** can only verify the authenticity of products and view limited product history. This granular control over data access ensures that sensitive information remains protected and that the system is compliant with privacy and security standards.

System Design

Information Architecture:

The Etherweave system is designed to ensure transparency, immutability, and traceability throughout the product life cycle, leveraging the Ethereum blockchain as the foundation for maintaining data integrity and security. The system is structured to support seamless product tracking and authentication by various users, including manufacturers, suppliers, retailers, and consumers. The core components of Etherweave's information architecture include:

Ethereum Blockchain:
At the heart of Etherweave is the **Ethereum blockchain**, a decentralised, distributed ledger that guarantees the immutability of data. The blockchain records all transactions related to product creation, verification, and ownership transfers, ensuring that every action is securely documented. The decentralised nature of Ethereum ensures that no single entity has control over the data, making it tamper-proof and resistant to unauthorised alterations.

Smart Contracts:
Etherweave's key operations are governed by **smart contracts** written in **Solidity**, the programming language designed for Ethereum. These contracts automate critical processes such as product creation, verification, and ownership transfer:

- **Product Registration:** When a manufacturer registers a new product on Etherweave, details like the product's name, batch number, production date, and other relevant metadata are stored on the blockchain via a smart contract.
- **Ownership Transfers:** As products move through the supply chain, ownership transfers are recorded on the blockchain. Each transfer is executed as a new blockchain transaction, creating an auditable record of every product's journey, from manufacturer to consumer.
- **Verification:** Consumers and other stakeholders can verify the authenticity of a product by accessing the blockchain records. The immutable nature of blockchain ensures that this data cannot be tampered with, providing confidence in the legitimacy of the product.

Each transaction, whether it's the registration of a new product or a transfer of ownership, is added to the blockchain as a new block. This guarantees **immutability**—once a block is created, it cannot be altered or deleted, providing a permanent and auditable history of each product's lifecycle.

Frontend and Backend Architecture:

To deliver a seamless user experience, Etherweave employs a robust frontend and backend architecture that supports real-time interaction with the blockchain and efficient data management.

1. **Frontend:**

The frontend of Etherweave is designed to be intuitive and user-friendly, providing responsive web interfaces for interacting with the system. These interfaces are built using modern web technologies such as:

- **HTML/CSS:** The foundational structure and styling of the web interface, ensuring responsive and visually appealing design across different devices.
- **React:** A powerful JavaScript library used for building dynamic user interfaces. React allows the Etherweave platform to provide fast, responsive interactions, such as product searches, verifications, and ownership transfers, while minimising page reloads.
- **Node.js:** Although primarily a backend technology, Node.js is leveraged in parts of the frontend architecture to handle asynchronous requests and support dynamic content updates without compromising performance.
- **JavaScript:** The core programming language for creating dynamic elements on the frontend. It powers the interactivity of Etherweave's web and mobile interfaces, such as validating user input, updating the interface in real-time, and handling communication with the backend.

The **web-based interface** allows manufacturers, suppliers, and retailers to register products, track product movements, and verify authenticity easily. These frontends provide real-time access to the data stored on the blockchain, ensuring transparency across the product lifecycle.

2. **Backend:**

Etherweave's backend is primarily responsible for handling interactions with the Ethereum blockchain. The backend processes are tightly integrated with the blockchain to manage product data, execute smart contracts, and ensure secure communication between the frontend and the decentralised ledger. The key components of the backend include:

- **Smart Contract Execution:** The backend interacts directly with the Ethereum blockchain, executing smart contracts that govern all product-related activities. When a user registers a product, initiates an ownership transfer, or verifies authenticity, the backend triggers the corresponding smart contract to perform the required action on the blockchain.
- **Blockchain Interaction:** Etherweave's backend interacts with Ethereum nodes to read and write data to the blockchain. Whether it's retrieving product information or recording a new transaction, the backend ensures that all interactions with the blockchain are secure, reliable, and fast.

- **APIs:** The backend provides an API layer that connects the frontend with the smart contracts on the Ethereum blockchain. These APIs enable the web and mobile interfaces to interact with the blockchain without requiring direct access, ensuring a seamless and secure user experience.
- **Database:** While the blockchain stores immutable product data, the backend may also utilise traditional databases (such as PostgresQL) for faster access to non-critical or auxiliary data, such as user session data, access logs, and caching for product images or metadata. This ensures efficient performance without compromising the security and immutability of core product records.

Integration of Frontend and Backend:

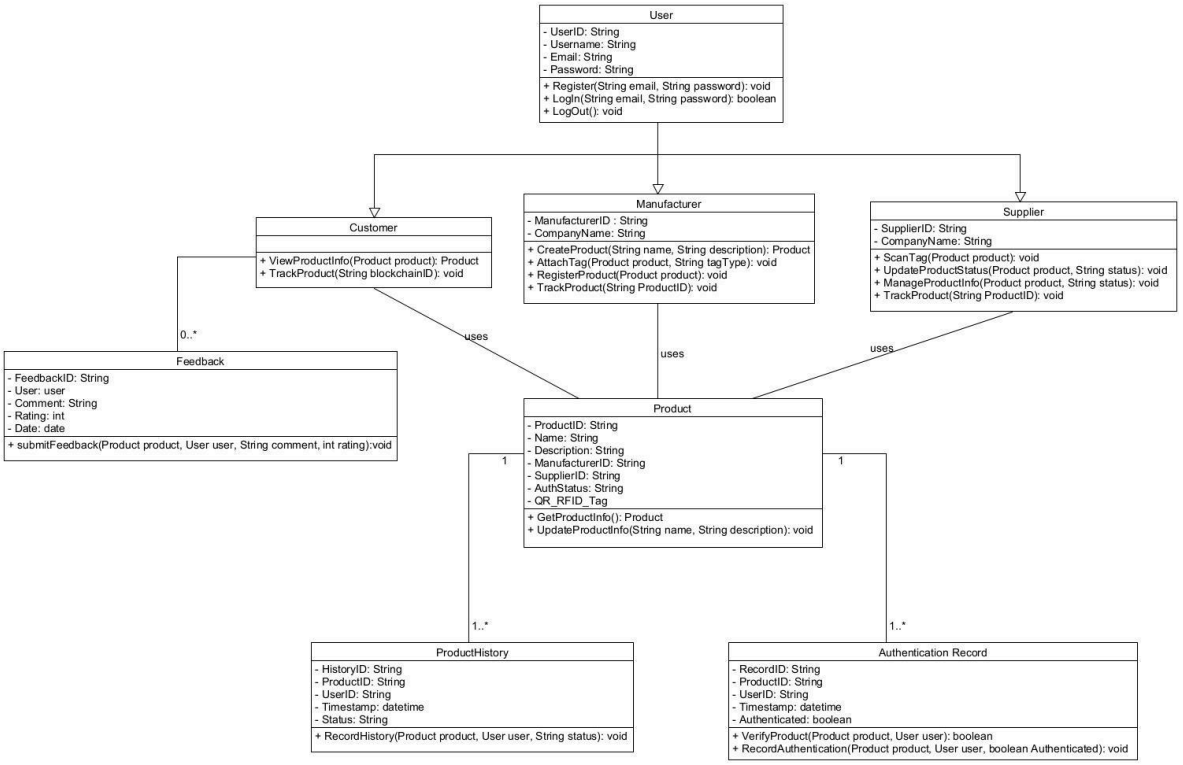
The frontend and backend are seamlessly integrated to provide real-time interaction with the blockchain, enabling users to register, track, and verify products through the web and mobile interfaces. This integration ensures that:

- **Product Registration and Tracking:** When a manufacturer registers a product on the web or mobile platform, the frontend sends the necessary product data to the backend, which then triggers the smart contract on Ethereum to store the product details on the blockchain. Users can then track the product's movement across the supply chain through the frontend, with real-time updates from the blockchain.
- **Verification of Authenticity:** Consumers can scan QR codes via the mobile, which prompts the frontend to request product data from the backend. The backend, in turn, retrieves the data from the blockchain and sends it back to the frontend, where users can view the product's entire history and verify its authenticity.
- **Ownership Transfers:** As products are transferred between supply chain entities, the frontend sends a request to the backend to execute the corresponding smart contract, recording the ownership change on the blockchain. This ensures that the entire lifecycle of a product is transparently documented and accessible to all authorised users.

UML Class Diagram

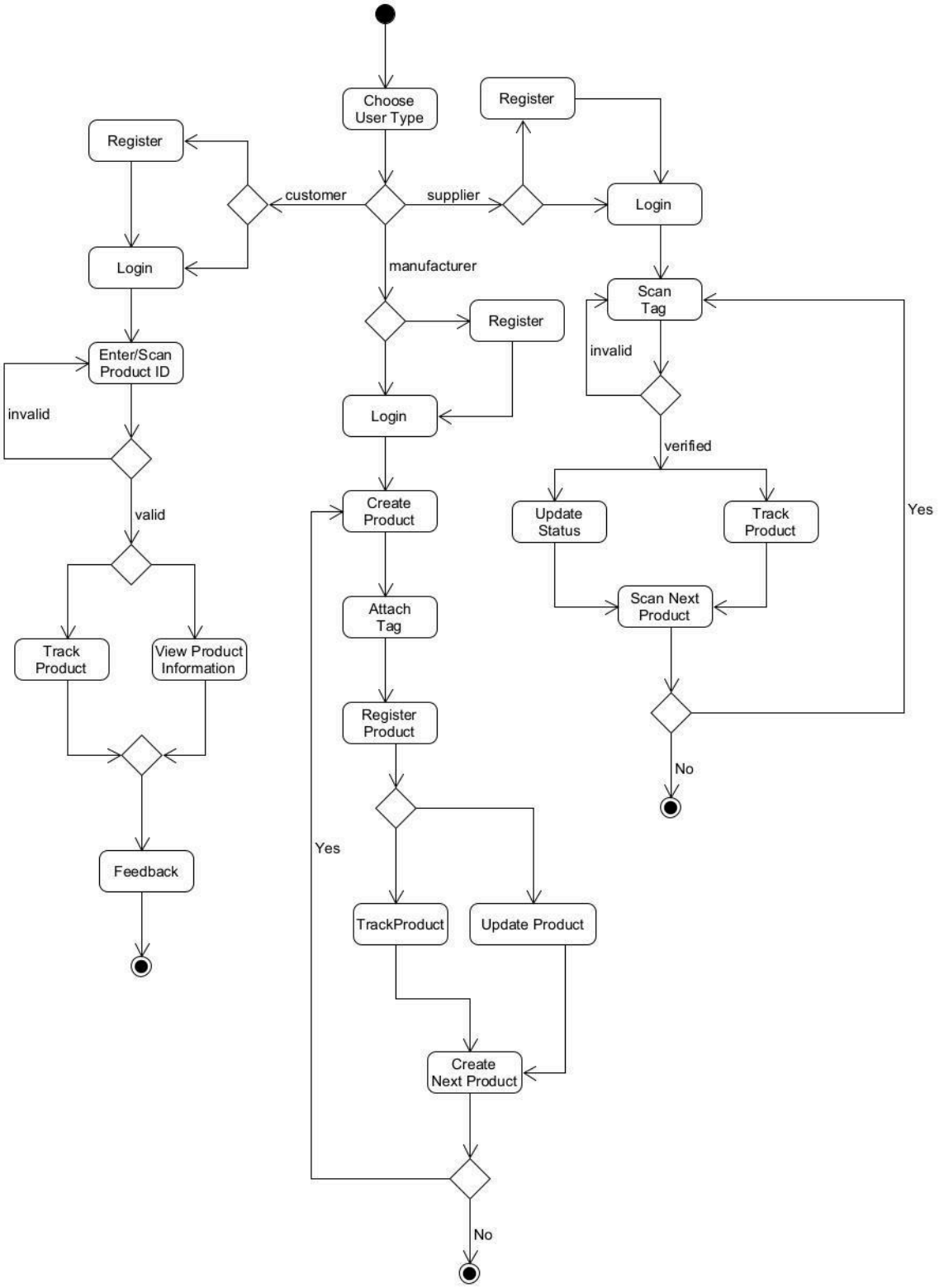
The system design is centred around several key classes that represent different entities:

- **User Class**: Represents the various types of system users (manufacturers, suppliers, and consumers), each with unique attributes such as **UserID**, **Username**, and **Email**.
- **Product Class**: Stores information about the products registered in the system, with attributes like **ProductID**, **Name**, **Description**, and **AuthenticationStatus**.
- **ProductHistory Class**: Keeps a log of every action performed on a product, including changes in ownership, allowing full traceability of a product's lifecycle.
- **AuthenticationRecord:** Represents the authentication record of a product with attributes such as product ID, record ID, user ID, time, Authentication status (True or False).
- **Feedback:** Stores any feedback submitted by the customer with its ID, user information, comment and so on.



UML Class Diagram of the System

State Machine Diagram



State Machine Diagram of the System

The above figure represents the state machine diagram of the system for all of the users involved (Customers, Manufacturers and Suppliers) and their respective functionalities. While the customers can only scan and view the product information to verify its authenticity, manufacturers will be able to register a completely new product, attach the tag information, update it as required and generate reports additionally. The suppliers will not be able to register a completely new product but be able to scan, update, track and generate reports respectively.

Data Structure and Algorithms

Data structures and algorithms play a critical role in the Etherweave system by ensuring efficient product authentication, verification, and data management. Custom data structures such as Merkle Trees and blockchain-based linked lists are utilised to maintain data integrity and facilitate fast verification processes. The Merkle Tree structure is implemented to guarantee data consistency and allow rapid authentication checks by only verifying the path from the product's data to the root hash. The blockchain-based linked list stores product transactions immutably, ensuring a transparent record of product ownership and movement. Additionally, specific algorithms, such as those used for generating unique QR and verifying product authenticity, contribute to the robustness and security of the system. Together, these data structures and algorithms enable Etherweave to provide a scalable, secure, and tamper-proof solution to combat counterfeiting.

Custom Data Structures

Merkle Tree for Product Authentication

Purpose: In order to guarantee the integrity of data and quick verification of product authenticity.
Structure: Binary tree where each leaf contains hash of product data (product ID, QR/RFID tag, status) and non leaf nodes carry its node hash. The final hash is represented by the root node and contains the entire tree structure.
Algorithm: New product or updated product data hash is generated and placed in a leaf node in the system. Verification is only required on the path from a product's hash to the root, making verification efficient without the need for the whole dataset.
Efficiency: This structure facilitates fast and tampering/consistency checks for large sets of product data, owing to the presence of the verifier, so that tampering or inconsistencies in the data can be detected.

Blockchain-based Linked List

Purpose: To store product transactions in an immutable, ordered way.
Structure: The linked list-like chain on the blockchain is formed by storing each transaction (product creation, update, verification) as a block that links to the previous block.
Algorithm: A block contains the hash of the previous block, the current transaction data, and a timestamp. A new block is added when a new transaction is created (e.g. product registration), its hash is linked to the previous block.

Efficiency: This means that the data is tamper proof, because changing any single block would require recalculating the whole chain.

Algorithms

QR/RFID Tag Generation Algorithm

Purpose: To generate unique and secure QR codes or RFID tags for products.
Algorithm:
Combine the product ID, manufacturer ID, and timestamp to generate a unique identifier.
Encrypt the identifier using a hash function (e.g., SHA-256) to generate a unique QR code or RFID tag.
Store the generated tag on the blockchain for verification.

Product Verification Algorithm

Purpose: To verify the authenticity of a product using blockchain data.
Algorithm:
The user scans the QR code or RFID tag.
The system retrieves the product's hash from the blockchain and compares it to the scanned data.
If the hashes match, the product is verified; otherwise, an alert is triggered for potential counterfeit.
Efficiency: This algorithm ensures fast and secure product verification using blockchain's immutable ledger.

Database Design

The database is designed to store user credentials, product information, and transaction history, complementing the immutable blockchain records. Below is a schema that outlines the tables, fields, and relationships used in the system.

Database Schema

User

Field Name	Data Type	Description	Constraint
userID	String	Unique identifier for the user	Primary Key, Not Null
userName	String	User's username for login	Unique, Not Null
email	String	User's email address	Unique, Not Null
password	String	Hashed password for authentication	Not Null
role	Enum	User role (Manufacturer, Supplier, Customer)	Enum: Manufacturer, Supplier, Customer

Product

Field Name	Data Type	Description	Constraint
productID	String	Unique identifier for the product	Primary Key, Not Null
name	String	Name of the product	Not Null
description	String	Brief description of the product	-
manufacturerID	String	Manufacturer who created the product (FK to Users)	Foreign Key to Users(userID)
QR_RFID_tag	Enum	QR code or RFID tag attached to the product	Unique, Not Null

Product History

Field Name	Data Type	Description	Constraint
historyID	String	Unique identifier for each product history entry	Primary Key, Not Null
productID	String	Product that the history entity is associated with	Foreign Key to Products
userID	String	User who performed the action	Foreign Key to Users
status	Enum	Status of the product (Registered, Verified, InTransit, Delivered)	Enum
timestamp	Timestamp	Date and time of the history entry	Auto-generated

Authentication Record

Field Name	Data Type	Description	Constraint
recordID	String	Unique identifier for each authentication record	Primary Key, Not Null
productID	String	Product being authenticated	Foreign Key to Products
userID	String	User performing the authentication	Foreign Key to Users
authenticated	Boolean	Boolean flag indicating whether the product was authenticated successfully	Not Null

Blockchain Architecture

Etherweave ensures that all interactions related to product creation, ownership transfers, and verification are securely recorded, preventing unauthorised alterations and ensuring traceability across the product's lifecycle. The key components of Etherweave's blockchain architecture include:

1. **Product Creation:**

Manufacturers initiate the first step in the product life cycle by registering products on the blockchain. To create a product, the manufacturer provides essential details such as:

- **Product Name** (e.g., brand and model),
- **Batch Number** (to track production runs),
- **Production Date** (timestamp for when the product was manufactured),
- **Other Metadata** (such as origin, materials, or any unique identifiers).

Once these details are submitted, a **smart contract** on the Ethereum blockchain is executed, which stores the product information as a new **block** on the blockchain. This block is cryptographically linked to previous blocks, ensuring its place in the overall blockchain ledger. Each block contains:

- **Transaction Details:** Information about the product creation event.
- **Product Data:** Metadata provided by the manufacturer.
- **Cryptographic Hashes:** A unique identifier for the block, which ensures that it cannot be tampered with after creation. Any attempt to alter the product details would invalidate the cryptographic link between blocks, making such attempts easily detectable.

This process guarantees **immutability**, meaning that once a product is created and recorded on the blockchain, no one can alter the data without leaving a trace. This provides assurance to all stakeholders that the information regarding the product's creation is authentic and cannot be manipulated.

2. **Ownership Transfer:**

As products move through the **supply chain**, ownership transfers from manufacturers to distributors, retailers, and ultimately to consumers. Each time a product changes hands, the **ownership transfer** is recorded on the blockchain via smart contracts, creating a permanent and auditable record of the transaction. This process includes:

- **Initiating a Transfer:** When a product changes hands, the current owner (e.g., manufacturer or distributor) initiates a transfer on Etherweave's platform by providing the necessary details of the transaction, such as the new owner's information.
- **Logging on the Blockchain:** The smart contract associated with the product executes the ownership transfer and stores it as a new block on the blockchain. The blockchain maintains a **chain of custody**, linking each ownership transfer to the

previous one through **cryptographic hashes**. This ensures that the product's journey through the supply chain is transparent and traceable.

- **Verification of Transfer:** All stakeholders in the supply chain can verify the legitimacy of the ownership transfer by querying the blockchain and reviewing the product's transaction history. Each transfer is recorded in a verifiable manner, ensuring that the product's history cannot be falsified or altered.

By ensuring that each ownership transfer is securely recorded on the blockchain, Etherweave creates a transparent and tamper-proof record of the product's journey from the point of manufacture to the final consumer. This not only increases trust within the supply chain but also deters counterfeit goods, as unauthorised parties cannot falsify ownership records.

3. **Verification:**

One of the critical features of Etherweave is the ability for **consumers** to easily verify the authenticity of products. This is achieved through the use of **QR codes** which are attached to each product by the manufacturer during the registration process. The verification process involves:

- **Scanning the Code or Tag:** Consumers can use a smartphone or other NFC-enabled device to scan the product's QR code. This action triggers a query to the blockchain, requesting the product's information and transaction history.
- **Blockchain Query:** Etherweave's backend interacts with the Ethereum blockchain to retrieve the product's data, including:
- **Product Registration Information:** Details about the product's creation, such as when and where it was manufactured, and its batch number.
- **Ownership History:** A complete chain of ownership transfers, from the manufacturer to the current owner, including timestamps and transaction details.
- **Verification Response:** The consumer receives the product's full transaction history in real-time, allowing them to confirm that the product is authentic and has passed through legitimate channels. If the product is not registered on the blockchain or if there are any discrepancies in its transaction history, the consumer can immediately identify potential fraud or counterfeiting.

This **verification mechanism** gives consumers confidence that they are purchasing genuine products. Additionally, it provides manufacturers with a powerful tool to combat counterfeiting and protect their brand reputation.

Key Features of Etherweave's Blockchain Architecture:

1. **Immutability:**

   One of the core advantages of using blockchain in Etherweave is its immutability. Once a transaction is recorded on the blockchain, it cannot be altered or deleted. This guarantees that the data regarding product creation, ownership transfer, and verification remains trustworthy and tamper-proof.

2. **Cryptographic Hashing:**

   Each block in the Etherweave blockchain is linked to the previous block through **cryptographic hashes**. This ensures that every transaction, whether it's a product registration or ownership transfer, is securely connected to the blockchain's history. Any attempt to alter the data would invalidate the hash, making it impossible to modify information without detection.

3. **Transparency and Traceability:**

   Etherweave ensures full **transparency** of the product lifecycle by making all transactions publicly accessible on the blockchain. All stakeholders, from manufacturers to consumers, can easily trace a product's journey through the supply chain and verify the legitimacy of ownership and authenticity at every step.

4. **Decentralisation:**

   Etherweave is built on the **Ethereum blockchain**, a decentralised network, meaning there is no central authority controlling the data. This decentralisation prevents any single entity from altering or corrupting the product information, adding an extra layer of security to the system.

5. **Smart Contracts:**

   Etherweave utilises **smart contracts** to automate and enforce the rules governing product registration, ownership transfer, and verification. These contracts ensure that all actions are executed based on predefined conditions and recorded on the blockchain in a secure and transparent manner.

Project Summary

Initial Requirements:

The Etherweave system was conceived to address the challenges of counterfeit products by providing a secure, transparent, and scalable solution for tracking products throughout the supply chain. The core requirements included:

- **User-Friendly Interface:** The system needed to be accessible to various users, including manufacturers, suppliers, and consumers, through both web and mobile interfaces.
- **Product Authenticity:** A blockchain system that can check the authenticity of a product.
- **Real-Time Access:** The system required real-time access to product information and transaction history to provide an up-to-date, transparent record of product movements.

Development Progress:

1. **Smart Contract Development:**
    - The core functionalities, including **product creation**, **registration**, and **ownership transfer**, were implemented using **Solidity**. These smart contracts automate the processes on the Ethereum blockchain, ensuring that every product's lifecycle is securely recorded and tracked.
2. **Testing:**
    - Etherweave was tested using **Ganache and Hardhat**, a local blockchain simulator that mirrors the Ethereum environment. Testing focused on simulating real-world scenarios like product tracking and ownership transfer, and results showed successful validation and accurate tracking throughout the supply chain.
3. **Web Interface:**
    - A fully functional **web interface** was developed using modern technologies like **React**, **Node.js**, and **JavaScript**. This interface allows users to register products, scan **QR codes**, and track product authenticity in real time, ensuring that both manufacturers and consumers can easily interact with the system.
4. **Mobile Application:**
    - The **mobile application** is currently under development, with a focus on making product verification even easier through **NFC technology**. This will allow consumers to quickly scan products using their smartphones, facilitating instant authenticity checks.

Iteration Management:

Iteration Management played a crucial role in the Etherweave project, ensuring that the system evolved based on feedback and testing. Each development cycle followed the principles of **agile methodology**, allowing the team to continuously refine and adapt the system to meet project goals.

1. **Iteration Overview:** The development process was broken into iterative cycles, with each iteration focused on refining specific components:
    - **Iteration 1:** Defined the overall scope and gathered requirements. Basic blockchain architecture and smart contract designs were established, and **RBAC** was introduced to manage user roles.
    - **Iteration 2:** Focused on building core functionalities, including product registration and verification. During this phase, QR codes were integrated into the system, and user feedback was gathered to refine the product registration process.
    - **Iteration 3:** The primary goal was to develop the front-end interface and integrate it with the smart contracts on the blockchain. Performance enhancements were also made to handle increased transaction volumes and improve system responsiveness.
    - **Iteration 4:** This phase focused on **scalability** and **security**. **Multi-factor authentication (MFA)** was introduced to enhance security, and testing was conducted using Ganache to simulate Ethereum blockchain interactions under high loads.

2. **Iteration Planning:**
    - **Goal Setting:** Each iteration began with clear objectives aligned with the project's requirements. For example, the goal for the third iteration was to integrate the front-end with the smart contracts.
    - **Task Breakdown:** The team used project management tools like **JIRA** and **Trello** to divide the iteration into smaller, manageable tasks. Tasks were assigned based on team members' expertise and tracked throughout the iteration.
    - **User Stories:** Each iteration was guided by **user stories** representing the needs of manufacturers, suppliers, and consumers. User stories were prioritised to ensure high-impact features were delivered early.

3. **Iteration Execution and Testing:**
    - **Development:** Smart contracts were coded and tested to ensure compliance with system requirements. As each iteration progressed, both front-end and back-end components were continuously improved and refined.
    - **Testing:** After development, testing was conducted using tools like **Ganache and hardhat** to simulate the blockchain environment. Usability testing was also performed to refine the user interface.

    - **Feedback Loop:** Feedback from testing was incorporated into subsequent iterations, ensuring that user experiences and system performance improved over time.
4. **Review and Retrospective:**
    - **Review Meetings:** At the end of each iteration, meetings were held to assess progress. The team reviewed whether the iteration's goals were achieved and identified areas for improvement.
    - **Lessons Learned:** A retrospective followed each iteration to discuss what worked well and what challenges were encountered. This reflection helped the team refine their approach in future iterations.

Iteration Outcomes:

1. **Adaptation:** The iterative development approach allowed the team to adapt quickly to new requirements. For example, user feedback during the third iteration highlighted the need for **NFC integration** for quicker product verification.
2. **Continuous Improvement:** Each iteration resulted in a more refined and functional system. Etherweave's development benefited from ongoing feedback, ensuring that the system was scalable, secure, and user-friendly.

Meeting Original Requirements:

Authentication and Verification:

- **Requirement:** Etherweave was required to allow manufacturers, suppliers, and consumers to verify product authenticity through blockchain technology.
- **Outcome:** The system enables manufacturers to scan **QR codes** of products, which can be scanned by consumers to verify authenticity. These codes provide access to a product's entire transaction history stored on the blockchain.
- **Changes:** The functionality did not change significantly from the initial requirements, as it was well-defined and effectively implemented from the beginning.

Product Registration and Tracking:

- **Requirement:** The system had to support manufacturers and suppliers in registering new products and tracking them across the supply chain.
- **Outcome:** Etherweave provides an intuitive interface for manufacturers and suppliers to register, update, and track products. Blockchain integration ensures that each stage of the product's lifecycle can be verified.
- **Changes:** While a comprehensive set of fields was initially proposed for product registration, user feedback suggested streamlining the process by removing less critical fields to improve the workflow. This change simplified the user experience while maintaining essential functionality.

Security and Data Integrity

- **Requirement:** This uses blockchain's cryptographic mechanisms to ensure high levels of security and integrity on product information.
- **Outcome:** Ethereum's blockchain technology guarantees the immutability of data, and the system is based on it. Data is only manipulated by people who are authentically authorised, and transactions are recorded securely.
- **Changes:** Sensitive user data stored outside the blockchain (e.g. personal details) were also added to additional layers of encryption. This was done to comply with the modern standards when it comes to data protection, for example GDPR.

Reporting and Analytics

- **Requirement:** Administrate users should be provided with the ability to generate reports for product authenticity, user activity and system performance.
- **Outcome:** The reporting feature was implemented successfully, including real time analytics of product verification, user engagement and system activity, for administrators.
- **Changes:** The reporting interface was made more interactive and customizable, giving user feedback by allowing the users to filter the data based on specific criteria (e.g. product categories, time frames). This change ensures that the system is in a desirable solution space for multiple users.

User Interface and Experience

- **Requirement:** Build a friendly interface for mobile and web platforms, for all user types (beginner, advanced).
- **Outcome:** The project created a responsive web interface and mobile platform, meant for easy use. Users can move around product verification, tracking and management sections with ease.
- **Changes:** Early testing informed streamlining of the navigation menu and optimising page load times. User feedback pushed for speed and simplicity, especially on mobile devices, and this was driven.

Requirements Traceability Matrix

Requirement	Met (Y/N)	Notes/Changes
Product Authentication and Verification	Y	Fully implemented; no major changes.
Product Registration and Tracking	Y	Simplified data fields for efficiency.
Role-Based Security and Access	Y	Added extra encryption layers for compliance with data protection laws.
Real-Time Product Status Updates	Y	Implemented on both web and mobile platforms.
Reporting and Analytics	Y	Made more customizable based on user feedback.
Responsive User Interface	Y	Optimised for mobile devices; improved navigation and load times.

.

Testing and Validation

Testing Strategy

- **Unit Testing**: Conducted for individual smart contracts using **Hardhat** to validate each function. This involved testing the correct execution of product registrations, ownership transfers, and user roles.
- **Integration Testing**: Verified interactions between the web/mobile front end and blockchain backend. Used **Postman** to test API endpoints and ensure proper communication between services.
- **Database Testing**: Verified the consistency and integrity of data stored in **PostgreSQL**, ensuring that relational data synchronised correctly with blockchain transactions.
- **User Acceptance Testing (UAT)**: Conducted with potential users to gather feedback on usability and performance. Surveys and direct interviews were conducted to ensure the application was easy to understand and meet user needs.

Test Cases

- **Product Registration**: Tested with various product details to ensure successful blockchain recording, including edge cases with long batch numbers and special characters.
- **Ownership Transfer**: Verified that all ownership changes were correctly logged and immutable, with additional testing to ensure that unauthorised users could not alter ownership.
- **Database Consistency**: Verified that data recorded on the blockchain matched data stored in **PostgreSQL**, ensuring there were no discrepancies.
- **Security Testing**: Verified that data transfer between components is encrypted and that only authorised roles could invoke specific smart contract functions.

Results

Testing confirmed that the Etherweave system could handle large transaction volumes with minimal latency. Users appreciated the straightforward verification process, and system reliability was validated through stress testing that simulated peak loads.The synchronisation between **PostgreSQL** and the blockchain was also confirmed to be consistent across all tested scenarios.

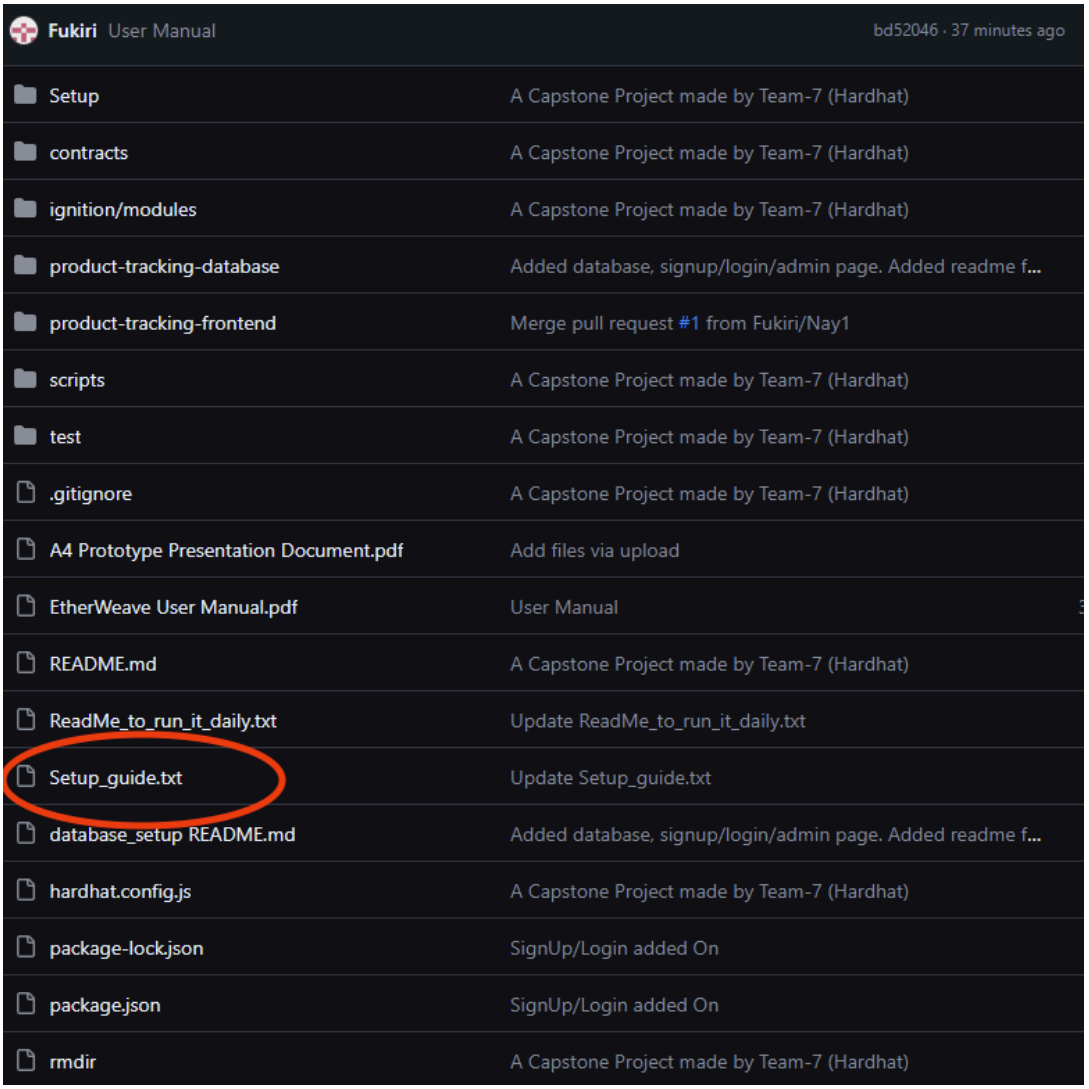
Current Project State

At the moment, the Blockchain based Traceable Anti counterfeiting System is completely functional, having surpassed or matched the initial project planning phase requirements. Based on extensive user feedback and real world application, the system has been through extensive testing and refinement. The core functionality is identical to the original vision, but usability, security and efficiency were slightly modified for better usability. The changes were based on feedback from early testers and stakeholders to make sure the system is robust and user friendly.

The project is well poised to be deployed for wider industries subject to counterfeit products with these accomplishments. And then the next steps from there are further real world testing with those stakeholders and expanding the system to be able to adapt to the market's changing needs.

System Installation Process

For complete system setup, please refer to '**Setup_guide.txt**' on Github
ReadMe File: Follow the detailed instructions in the ReadMe file to set up the developmentenvironment and deploy the system to different environments (local, testnet,and mainnet).


Fukiri User Manual
bd52046 · 37 minutes ago
Setup
A Capstone Project made by Team-7 (Hardhat)
contracts
A Capstone Project made by Team-7 (Hardhat)
ignition/modules
A Capstone Project made by Team-7 (Hardhat)
product-tracking-database
Added database, signup/login/admin page. Added readme f...
product-tracking-frontend
Merge pull request #1 from Fukiri/Nay1
scripts
A Capstone Project made by Team-7 (Hardhat)
test
A Capstone Project made by Team-7 (Hardhat)
.gitignore
A Capstone Project made by Team-7 (Hardhat)
A4 Prototype Presentation Document.pdf
Add files via upload
EtherWeave User Manual.pdf
User Manual
README.md
A Capstone Project made by Team-7 (Hardhat)
ReadMe_to_run_it_daily.txt
Update ReadMe_to_run_it_daily.txt
Setup_guide.txt
Update Setup_guide.txt
database_setup README.md
Added database, signup/login/admin page. Added readme f...
hardhat.config.js
A Capstone Project made by Team-7 (Hardhat)
package-lock.json
SignUp/Login added On
package.json
SignUp/Login added On
rmdir
A Capstone Project made by Team-7 (Hardhat)

Project Closeout

Lessons Learned:

- ➔ **Challenges:**
  - ◆ **Blockchain Integration with Traditional Systems:**One of the most significant challenges was integrating blockchain with existing supply chain management systems. Traditional supply chains often rely on centralised databases, whereas Etherweave's blockchain architecture is decentralised. This required substantial adjustments to ensure smooth integration and compatibility.
  - ◆ **Lack of Blockchain Knowledge:** The team initially faced a challenge due to a **lack of in-depth knowledge** about blockchain technology. Overcoming this required extensive learning through **textbooks**, **videos**, **YouTube tutorials**, and external help from **Yudi Zhang**, who provided valuable insights into blockchain development and its integration.
  - ◆ **Time Constraints:** The team had to reorganise meeting times due to various commitments personally because of being international students and part time jobs and various assignments.
  - ◆ **Knowledge Barrier:** The team had to refer to various sources inorder to intercept concepts introduced by other teammates.
  - ◆ **Lack of Team Members:** The team had to put in a lot of effort compared to other teams due to having less number of members.
  - ◆ **Scaling and User Adoption:**Ensuring that the system could scale and be adopted by users unfamiliar with blockchain technology was another challenge.
  - ◆ **Balancing Security and User Experience:** Finding the right balance between security and user experience was critical. The team had to design a system that was both secure and user-friendly, requiring several rounds of testing and redesign.

- ➔ **Technical Solutions:**
  - ◆ **Smart Contracts for Data Integrity:** The use of **smart contracts** proved to be an effective solution for ensuring product data integrity and automating processes such as product registration and ownership transfer.
  - ◆ **Stakeholder Education:** Overcoming the blockchain knowledge gap among stakeholders required a focused effort on education. With the knowledge gained from various sources, the team developed a better understanding of how blockchain systems work, which was then passed on.

Teamwork Evaluation

Team collaboration played a pivotal role in meeting project milestones. The adoption of Agile practices, including frequent stand-ups and sprint reviews, helped ensure effective communication and progress tracking. Each team member contributed by taking ownership of specific modules, and overall team synergy led to the successful completion of the project.

Post-Project Review

The Etherweave project successfully met its primary objectives, showcasing the system's ability to enhance product tracking and combat counterfeiting through blockchain technology. By leveraging the security and transparency of blockchain, Etherweave created a robust, immutable system for tracking products across their entire lifecycle, from manufacturing to consumer purchase.

System Success

Meeting Objectives:

The Etherweave system was designed to create a transparent, tamper-proof ledger that records all product-related transactions, such as creation, movement, and ownership transfers. The blockchain ensures that once a product's data is entered into the system, it cannot be altered or deleted. This feature ensures product authenticity and improves trust among manufacturers, suppliers, and consumers.

Etherweave's integration of QR codes enables quick and secure product verification. Consumers and supply chain stakeholders can scan a QR code to access product data instantly, ensuring that counterfeit products are immediately identifiable and prevented from entering the supply chain.

Performance Testing:

Performance testing demonstrated that Etherweave is capable of handling large transaction volumes without degradation in performance. This scalability makes the system suitable for industries that manage vast amounts of product movement and require real-time updates on product status.

Stress tests showed that the system can maintain fast response times even as the number of transactions increases, further proving its potential to handle high-traffic environments such as global supply chains or industries dealing with large-scale product distribution.

Verification Efficiency:

The combination of blockchain technology and QR code scanning allows for seamless, instant verification. Consumers and other stakeholders can quickly verify the authenticity of a product, gaining access to a product's entire transaction history with just a scan.

This efficient verification process not only combats counterfeiting but also promotes greater transparency and trust in the supply chain, benefiting both businesses and consumers.

Project Acceptance

Etherweave will be used for wider testing in pilot programs within high-risk industries, including luxury goods, pharmaceuticals, and electronics. These industries are particularly vulnerable to counterfeit products, making Etherweave a valuable solution for improving supply chain transparency and product authentication.

- **Pilot Programs:** Etherweave will undergo extended testing in real-world environments, where manufacturers and consumers will provide feedback on its usability, performance, and impact on counterfeit prevention. These pilot programs are designed to assess how the system performs under actual industry conditions, with the feedback being used to refine the system further.
- **Feedback Integration:** Feedback from manufacturers, distributors, and consumers will help guide further development, focusing on improving user experience, optimising performance, and expanding functionality based on industry needs. This iterative feedback process ensures that Etherweave continues to evolve to meet the demands of its users.

Transition Plan

The transition plan for Etherweave focuses on scaling the system for broader industry adoption, introducing new features, and refining its performance based on feedback from pilot programs. This plan outlines the steps necessary to ensure Etherweave's continued success as it enters new industries and grows its user base.

Additional Features:

1. **Automated Inventory Management:**

Etherweave's next phase will include the development of **automated inventory management** features, allowing businesses to track product levels in real-time. By integrating blockchain technology with automated systems, businesses can have a clear view of their stock levels, reducing the risk of oversupply or undersupply.

The automated inventory system will allow companies to monitor their products throughout the supply chain, ensuring that inventory is consistently updated with accurate information regarding product status, location, and ownership.

Businesses will be able to set up automated alerts when inventory levels are low or when products reach certain stages in the supply chain, optimising their logistics and inventory control processes.

2. **Predictive Analytics:**

Etherweave will introduce **predictive analytics** capabilities, allowing businesses to forecast product demand and optimise their supply chains accordingly. Using historical data stored on the blockchain, the system can predict trends, identify potential supply chain disruptions, and help businesses plan for future demand.

These analytics will help businesses reduce waste, minimise overproduction, and improve overall supply chain efficiency. Predictive analytics will also assist in detecting patterns that could indicate counterfeiting risks, allowing companies to take preventive action.

3. **Enhanced Security Protocols:**

In response to user feedback, Etherweave will enhance its security features, including the introduction of **multi-factor authentication (MFA)** and **role-based access control (RBAC)** improvements. These enhancements will ensure that only authorised personnel can access sensitive data or initiate critical actions, providing an additional layer of protection against unauthorised access or tampering.

Data encryption will also be further optimised, ensuring that all communications and transactions between the blockchain and users are securely encrypted from end to end, reducing the risk of data breaches.

4. Scalability Testing:

As Etherweave gains wider adoption, **scalability testing** will be a critical focus to ensure that the system can handle larger user bases and transaction volumes. The following steps will be undertaken during scalability testing:

- **Stress Testing:**

Etherweave will be stress-tested to evaluate how the system performs under high transaction loads and increased user traffic. These tests will simulate real-world scenarios, such as handling thousands of product verifications per hour or accommodating large-scale product distribution during peak times.

The goal is to ensure that the system can maintain optimal performance, even as the number of users and transactions grows exponentially.

- **Infrastructure Optimization:**

Based on the results of stress testing, Etherweave's backend infrastructure will be refined to support larger databases and faster query times. Optimising the integration between the blockchain and PostgreSQL will ensure that the system can efficiently manage both static blockchain data and dynamic auxiliary data, such as user sessions or logs.

Cloud infrastructure, such as AWS, will be leveraged to ensure the system can scale horizontally and vertically, adding more computing power or storage capacity as needed to meet growing demands.

- **Load Balancing:**

To handle increasing traffic and ensure the system remains responsive, load balancing techniques will be implemented. This will distribute traffic across multiple servers, ensuring that no single server becomes overloaded. It will also improve redundancy, ensuring that the system remains available even if one server fails.

User Feedback Integration:

User feedback from pilot programs will play a pivotal role in guiding Etherweave's future development. By continuously gathering and analysing feedback from manufacturers, suppliers, and consumers, the Etherweave team will prioritise enhancements and adjustments that improve user experience and meet industry-specific needs.

1. **User Experience Refinement:**
    - Feedback from users will help refine the Etherweave interface, ensuring that both the web and mobile applications are intuitive and easy to navigate. Any pain points identified during the pilot programs will be addressed, ensuring that the system is accessible to all users, regardless of their technical expertise.
    - User onboarding processes may be streamlined based on feedback, making it easier for new users to register products, verify authenticity, and track product movements.
2. **Security Enhancements:**
    - Users will provide insights into how Etherweave's security features are functioning in real-world scenarios. Based on their feedback, additional security protocols may be introduced, such as more customizable permission settings or real-time alerts for suspicious activities.

Industry Expansion:

Etherweave's initial focus has been on industries such as luxury goods, pharmaceuticals, and electronics—sectors that are highly susceptible to counterfeit products. As part of its transition, Etherweave plans to expand into new industries where product authenticity is equally critical.

1. **Agriculture:**
    - Counterfeiting in the agriculture sector, particularly with fertilisers, seeds, and pesticides, poses serious risks to farmers and food security. Etherweave will expand into agriculture by allowing farmers and suppliers to track the authenticity of agricultural inputs, ensuring that only certified products are used.
2. **Automotive Parts:**
    - The counterfeit automotive parts market is a significant concern, as fake parts can lead to safety hazards. Etherweave will help track and authenticate automotive parts throughout the supply chain, from manufacturers to dealerships, ensuring that consumers receive genuine, high-quality parts.
3. **Medical Equipment:**
    - The medical equipment industry is another sector prone to counterfeiting, with fake equipment posing risks to patient safety. By integrating Etherweave, medical equipment manufacturers and suppliers can ensure that their products are genuine, properly certified, and traceable throughout the distribution process.

Conclusion

The Etherweave project successfully addresses the pervasive issue of product counterfeiting through the use of blockchain technology. By leveraging the transparency and immutability of the Ethereum blockchain, the system provides a reliable solution for verifying product authenticity throughout the supply chain. The integration of smart contracts, PostgreSQL for relational data has resulted in a scalable and secure architecture capable of managing large transaction volumes.

The project demonstrated effective use of blockchain to maintain data integrity, enhanced user accessibility with web and mobile interfaces, and ensured efficient data management through a hybrid approach utilising both on-chain and off-chain data storage. Although challenges remain regarding scalability and user education, the groundwork laid by Etherweave provides a promising basis for further development and real-world application.

Future iterations will focus on expanding system capabilities, optimising scalability, and fostering user adoption through educational initiatives. Overall, Etherweave represents a significant step towards mitigating the impact of counterfeit products in global markets, ensuring greater transparency, security, and trust across supply chains.

Team Contribution Acknowledgment Document

Project: Etherweave – Blockchain-based Traceable Anti-counterfeiting System

Course: CSIT321

Group: Group 7

Date: October 11, 2024

We, the undersigned members of Group 7, hereby acknowledge that all team members have contributed equally to the development, execution, and completion of the Etherweave project. Each individual has actively participated in various aspects of the project, including:

- System design
- Smart contract development
- Frontend and backend development
- Testing and debugging
- Documentation and report writing

The division of work was fair, and every member contributed according to their strengths and expertise. Furthermore, we confirm that we are fully satisfied with the contributions made by each team member and believe that the success of the project was due to our collective teamwork and collaboration.

By signing this document, we confirm that all members contributed equally and are in agreement with the contributions made by their peers.

Signatures:

Team Member Name	Student ID	Signature	Date
Hein Khant Zaw	7692870	_____HKZ_____	10/10/2024
Nay Oo Lwin	7691713	_____NOL_____	10/10/2024
Joel Joseph	7447218	_____JJKJ_____	10/10/2024
Alwin Basil Shah	7451775	_____ABS_____	10/10/2024

Project Acceptance

Project Supervisor: Dr Yudi Zhang

Client: Dr Yudi Zhang

Client Representative: Dr Yudi Zhang

Student Project Team: **Group 16**

Project Sign-off conditions:

- Dr Yudi Zhang is satisfied with the results of the project and agrees that the requirements of the project proposal have been met.
- Dr Yudi Zhang has received copies of all relevant documents and files.
- Dr Yudi Zhang has received all deliverables in an appropriate working condition.
    - Any external libraries, frameworks and/or plugins have been used within the scope of their respective licences.
- The student project team (listed above) retains exclusive joint ownership of the source code for the system developed, restricting external distribution and the reselling of the code.
- Dr John Le has permission to modify, customise, update and expand the source code for future usage within the organisation.
- Dr John Le has received full ownership of all design assets used with the project.

By signing this document, I acknowledge that I have received all stated deliverables and agree to the above terms.

Client Representative: _____ Date: _____

By signing this document, I acknowledge that I have delivered all stated deliverables and agree to the above terms.

Student team:	Alwin Basil Shah	Date: 10/10/2024
	Joel Joseph	Date: 10/10/2024
	Nay Oo Lwin	Date: 10/10/2024
	Hein Khant Zaw	Date: 10/10/2024

References

Nakamoto, S. 2008, *Bitcoin: A Peer-to-Peer Electronic Cash System*, viewed 10 October 2024, https://bitcoin.org/bitcoin.pdf.

Wood, G. 2014, *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, Ethereum Project, Yellow Paper, viewed 10 October 2024, https://ethereum.github.io/yellowpaper/paper.pdf.

Buterin, V. 2013, *Ethereum Whitepaper*, Ethereum Foundation, viewed 10 October 2024, https://ethereum.org/en/whitepaper/.

Antonopoulos, A.M. 2017, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, O'Reilly Media, Sebastopol, CA.

Swan, M. 2015, *Blockchain: Blueprint for a New Economy*, O'Reilly Media, Sebastopol, CA.

Wüst, K. & Gervais, A. 2018, 'Do you need a blockchain?'. *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, IEEE, Zug, Switzerland, pp. 45-54.

Hughes, A., Park, A., Kietzmann, J. & Archer-Brown, C. 2019, 'Beyond bitcoin: What blockchain and distributed ledger technologies mean for firms', *Business Horizons*, vol. 62, no. 3, pp. 273-281.

Mougayar, W. 2016, *The Business Blockchain: Promise, Practice, and the Application of the Next Internet Technology*, Wiley, Hoboken, NJ.

Christidis, K. & Devetsikiotis, M. 2016, 'Blockchains and smart contracts for the internet of things', *IEEE Access*, vol. 4, pp. 2292-2303.

Zheng, Z., Xie, S., Dai, H.N., Chen, X. & Wang, H. 2017, 'An overview of blockchain technology: Architecture, consensus, and future trends', *2017 IEEE International Congress on Big Data (BigData Congress)*, IEEE, Honolulu, HI, pp. 557-564.

Kshetri, N. 2018, '1 Blockchain's roles in meeting key supply chain management objectives', *International Journal of Information Management*, vol. 39, pp. 80-89.

Tapscott, D. & Tapscott, A. 2016, *Blockchain revolution: How the technology behind bitcoin is changing money, business, and the world*, Penguin, New York, NY.

Fairfield, J.A. 2014, ‘Smart contracts, bitcoin bots, and consumer protection’, *Washington and Lee Law Review Online*, vol. 71, no. 2, pp. 35-50.

Swan, M. 2015, ‘Blockchain thinking: The brain as a decentralised autonomous corporation’, *IEEE Technology and Society Magazine*, vol. 34, no. 4, pp. 41-52.

Xu, X., Weber, I. & Staples, M. 2019, *Architecture for Blockchain Applications*, Springer, Cham.

Zhang, Y., Kasahara, S., Shen, Y., Jiang, X. & Wan, J. 2019, ‘Smart contract-based access control for the internet of things’, *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594-1605.

Morkunas, V.J., Paschen, J. & Boon, E. 2019, ‘How blockchain technologies impact business models’, *Business Horizons*, vol. 62, no. 3, pp. 295-306.