A Project Report on

Blockchain Based Web Framework for Pharma Supply Chain

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Jainam Chopra(18104064)
Yash Bhosale(18104049)
Satish Gupta(18104024)

Under the Guidance of

Prof. Apeksha Mohite
Prof. Nahid Shaikh

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2021-2022

Approval Sheet

This Project Report entitled ***"Blockchain Based Web Framework for Pharma Supply Chain"*** Submitted by ***"Jainam Chopra"(18104064), "Yash Bhosale"(18104049), "Satish Gupta"(18104024)*** is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

(Prof. Nahid Shaikh)
Co-Guide

(Prof. Apeksha Mohite)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Blockchain Based Web Framework for Pharma Supply Chain"*** submitted by ***"Jainam Chopra" (18104064),"Yash Bhosale" (18104049),"Satish Gupta" (18104024)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2021-2022.

(Prof. Nahid Shaikh) Co-Guide	(Prof. Apeksha Mohite) Guide
Prof. Kiran Deshpande Head Department of Information Technology	Dr. Uttam D.Kolekar Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane

Date:

Acknowledgement

We have great pleasure in presenting the report on **Blockchain Based Web Framework for Pharma Supply Chain.** We take this opportunity to express our sincere thanks towards our guide **Prof. Apeksha Mohite** & Co-Guide **Prof. Nahid Shaikh** Department of IT, APSIT Thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards their constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Jainam Chopra
18104064

Yash Bhosale
18104049

Satish Gupta
18104024

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Jainam chopra 18104064)

(Yash Bhosale 18104049)

(Satish Gupta 18104024)

Date:

Abstract

News of the effectively completed vaccine trials have the ability to hold consolation to billions of human beings throughout the world. But as encouraging as the ones trials also can moreover be, it‘s miles incredibly vital to observe that a vaccine can not be as effective if it is not efficaciously allotted and trusted thru manner of approach of the public. As 77% of U.S. citizens are concerned about the safety and efficacy of vaccines, a whole-of-government technique is critical thru the very last mile to infuse speed, accountability, and transparency during the vaccine distribution system. There is a pressing need for product tracking and tracing on account of globalized supply chains, several suppliers, risk of counterfeit products, and customer name for in-depth visibility. Effective vaccine cold chain management system is required for precise coordination and cooperation during multiple activities to make certain the right situation of vaccines, which require temperature monitoring, sharing of records for traceability, asset transfer, claims incase of damage to the product.

This project builds an architectural shape required to merge IoT and Blockchain into the deliver chain manipulate Web platform to show the temperature of insulated vaccine or cooler box containing other perishable items. In this project, it shows that promising system of integrating IoT and Blockchain for COVID-19 vaccine delivery chain control manipulate.

Contents

List of Figures

List of Tables

List of Abbreviations

ASN:	Advanced Shipment Notice
CAGR:	Compound Annual Growth Rerport
CCL:	Cold Chain Logistics
COVID-19 :	Corona virus disease
IoT:	Internet of Things
NCCD:	National Center For Cold-Chain Development
SHA 256:	Secure Hash Algorithm 256-bit
TEE:	Trusted execution environment

Chapter 1

Introduction

1 million humans die each 12 months due to counterfeit drugs.25% vaccines attain the vacation spot in a degraded condition. 20% temperature touchy merchandise are destroyed in the course of shipping. These numbers imply that pharmaceutical delivery chain protection is regularly measured only after the harm is performed. Currently the conventional delivery chain marketplace makes use of guide intervention‘s in producing ASN, Purchase orders packaging and labelling barcodes, present day infrastructure is simply confined to monitoring the place of an asset, within the present day surroundings there is lots of accept as true with troubles among all of the events due to the fact of the opaque nature of the business, and even catching hold of the actual stakeholder accountable in the course of harm of products isn't always in any respect possible, that's one of the loopholes exploited through predominant large companies. At minimal a blockchain based ledger is wanted that shops the transactions performed withinside the procedure of asset possession switch. This trouble is solved through Satoshi Nakamoto in his paper approximately while he created bitcoins. Implementation of programs the use of blockchain ensures the high-satisfactory of virtual information that is being used.

a) Storing the records in a blockchain.
b) For correctness protocol regulations may be used.
c) And for figuring out the proprietor public key cryptography may be used.

One of the helps of the new paradigm is the blockchain era, in which the combat towards corruption and value discount are confirmed as ensures of this era, however it's far exactly in its conjunction with the one of a kind governments in which each fields have to be well tailored. Ethereum is a unfastened open-supply platform which enables builders to construct and install decentralized Applications consisting of clever contracts and different complex felony and economic programs. Ethereum is type of a programmable Bitcoin in which builders can use the underlying blockchain to create markets, shared ledgers, virtual organizations, and different infinite answers utility to a trouble that want immutable information and agreements, all with out the want for a moderator or realtor. Released in 2015, Ethereum is the brainchild of the prodigious Vitalik Buterin, who noticed the ability makes use of of Bitcoin‘s underlying blockchain era as the following steps in dashing the growth of the blockchain community. Ethereum is now presently the cryptocurrency with the second maximum coin marketplace cap and is anticipated through a few to surpass Bitcoin as each a valued funding and because the world‘s maximum famous cryptocurrency. Hence, Ethereum is nice desirable for growing a ledger that shops transactions in the course of the land possession switch procedure. The purpose is to create a ledger in conjunction with a few clever contracts with a view to triggers the diverse occasions which can be going to appear at the gadget in the course of the procedure of switch of pharma items from one celebration to another.

Firstly, Cold Chain which is our topmost priority is simply defined as low-temperature logistics system, the scope of our research and development is limited to a vaccine distribution and monitoring system. A typical cold chain starts at the farm gate level or the production center. The primary stage of cold chain is sorting, grading and pre-cooling; subsequently, the products pass through various processing, storage phases, and other infrastructure to reach the end consumer. Various components involved in the integrated cold chain are:

- Packing and cooling fresh food products (immediately after harvest or collection),
- Food processing (i.e. ripening, chilling or freezing of processed foods),
- Refrigerated transportation (cold transport and temporary warehousing under temperature controlled conditions)
- Cold storage (short or long term warehousing of chilled or frozen foods),
- Retail (refrigerated or freezer storage and displays at wholesale markets, retail markets and foodservice outlets)

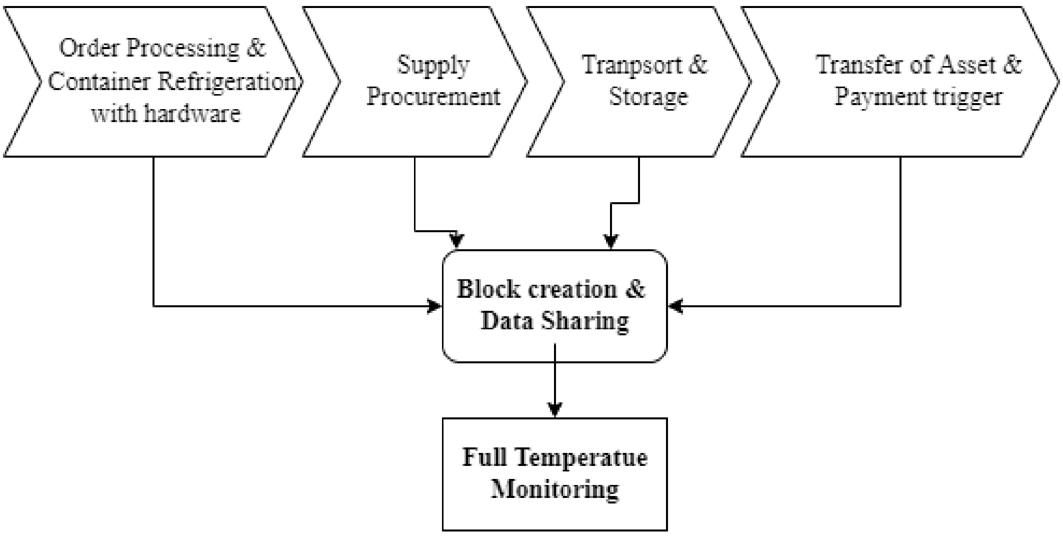


Figure 1.1: Blockchain based Coldchain Framework

A)An overview of India's Cold-chain logistics
India's cold goods market is expected to grow at a CAGR of more than 14% during the forecast period 2020-2025. The need for a cold chain is due to growth in sectors such as organized stores, processed foods, pharmaceuticals, growth transition towards horticulture, etc. The market is concentrated by just few major players namely Gati Kausar India Pvt Ltd, Snowman Logistics Pvt Ltd, ColdEx Logistics Pvt Ltd, Stellar Value Chain Solutions

Pvt Ltd, amongst others.

B) Gaps and challenges:
As per a recent study by NCCD, there is a shortfall of almost 70,000 pack houses in the country. The gap of 70,000 pack houses indicates an average need for one between for every 10 villages. This evaluation indicates that the cold-chain backend is underdeveloped and large investment needs to flow into rural India, at grass roots level. The optimal requirement of cold stores in the form of bulk stores and distribution hubs is 35.1 million tons in capacity, as against the existing cold stores capacity of 32.86 million tons from about 7129 units. As per a recent baseline census undertaken14, 1219 cold stores of estimated 5 million tons in size, were found permanently closed/not available. The total number of functional cold stores is 5367 amounting to a total storage capacity of 26.85 million tons.

C) Challenges in Indian Cold Chain infrastructure.

- **High lifecycle cost:** High real estate prices with an increase of more than 280% in last decade adds up significantly to the total cost, as for building a cold storage with 1 million cubic feet of space which requires an acre of land. Interstate barriers, intercity and state taxes, and bad roads etc. are other factors contributing to the rise in project and operational costs. In the country like the USA, the similar scale of installation cost less than half of the investment required in India.
- **Quality cold warehouse infrastructure:** Nearly 75% of the cold storage infrastructure created in the past is suitable only to store single commodity, rendering them of no use for utilizing for the multi temperature and multi commodity storage.
- **Standards and protocols in construction and operation of facility:** Technical standards followed in India are mostly unsuitable for Indian conditions, which results in poor performance of a standard refrigerated system.
- **Uneven distribution of cold storage:** Storage facility throughout the supply chain is another major challenge. Majority of cold storage facilities are located at the point of production, creating a lack of efficient supply chain to the downstream operations/ markets.

Chapter 2

Literature Review

While Implementing the architecture of IoT and Blockchain for maintaining a transparent supply chain management, the study led us deep into the research with the problems related with the synchronization of all the above mentioned technologies.
Many strategies for tackling these challenges have been reported in the literature. Referring to [1], it introduces a new concept BigchainDB to fill the gap in the decentralized systems at scale. The paper concludes with an outline of a use case and therefore the challenges to adopt blockchain technology within the future food provide chain traceability systems as mentioned. Current supply chain monitoring is limited just to certain logistic activities, real time condition monitoring of medicines, vaccines still a problem. The proper use of IOT and Blockchain is made to sense tamperproof data using TEE(Trusted execution environment), and tamper proof software using cloud and blockchain which give impregnable security in terms of data integrity which use SHA 256 algorithm the most secure algorithm in terms of cryptography and hashing the input data it cannot be breached. So, this combination of tamperproof hardware and tamperproof software helps building tamperproof and transparent system [2].

One of the most critical issues, monitoring assets from the manufacturing level monitoring components exactly from the time they are produced till the time they are delivered to end customer. Many a times the parts/components are stored in such conditions which either damage the component or destroy it resulting in degrading the performance of the final product. Hence finding the root liable authority for any damage caused is a major problem [3]. Blockchain issues token recipes for every component involved in making a product, which starts monitoring the condition right from the production line where the parts/components are produced and kept to be stored. As the components changes hands or gets added with new components or product the ownership also gets changed and a new block is created using smart contracts, keeping the initial ownership records intact.

Manual intervention increases the chances of data tamper while the products moves from one party to another. This paper[4] focuses on the issues such as product recalls, triggering defaulter payments, claiming insurances, and looking into the clearing of custom houses in case of international export/import, as a small business or a consumer is not aware about the import and export laws, this system proposes developing a software using Ethereum Web Framework which directly uses the IF... THEN logic for triggering Smart Contracts.

Previous study by[5] planned a Blockchain answer for secure IoT knowledge acquisition used for weather observance. In their research, an architecture consisting of ESP8266 silicon chip and temperature and pressure sensing element of GY-BMP280 has been developed. the information collected from the system displayed on the net interface of Grafana to totally read over the weather. knowledge collected then sent to Raspberry Pi entryway for additional processed and can structure for the transactions within the chain.

One of the most critical issues, currently not fully managed in CCL, is the safe transportation of temperature sensitive medicines, vaccines, and biological samples. The diffusion of emerging technologies enabling the Internet of Things such as embedded systems, mobile Apps and Cloud services, it is possible to obtain continuous monitoring of sensitive substances and environments. Previous work by [11] proposed a system called SensIC for monitoring the refrigerated storage of drugs, vaccines, medical devices and any biological samples (blood and its derivatives, saliva, urine, cells of various types and classification). The proposed system offers immediate alarm tools in case of malfunction of the refrigeration systems (fridge and/or freezers) using implementation of smart device called SensBox and a cloud infrastructure dedicated to device management.

Previous study by [12] proposed a Blockchain solution for secure IoT data acquisition used for weather monitoring. In their study, a system consisting of ESP8266 microchip and temperature and pressure sensor of GY-BMP280 has been developed. The data collected from the system displayed on the web interface of Grafana to thoroughly view over the weather. Data collected then sent to Raspberry Pi gateway for further processed and will make up for the transactions inside the chain.

Vaccine management and monitoring, cold chain management, and immunization safety are the key areas of logistics support. Logistics ensures that vaccines are available at the right time, in the correct amount, and in the correct condition. The cold chain refers to the storage and transportation of vaccines at recommended temperatures from the manufacturing location to where they will be used. An effective cold chain ensures that vaccines will remain effective and usable when they are administered [13]. It has been established that any breach in temperature control can degrade a vaccine, making it lose its full potency. An unbroken cold chain with temperatures below the required storage temperature has to be maintained, or it will risk defrosting and losing its efficacy within a day of temperature breach above the required storage temperature. Hence, a good temperature monitoring and control is required throughout the transport process to ensure vaccine quality and safety.

IoT is an encouraging model that integrates several communication and technical solutions. The IoT is defined as a field in which each physical object is to be connected at any time and at any place with the help of internet and to be able in identifying these devices to other devices [14].

Chapter 3

Project Design

3.1 Proposed System:

- Device Layer
  The objective of the device layer is to collect the temperature data within cooler box (insulated container). The temperature sensor for this project prototype will utilising DHT-11 sensors and this sensor has been chosen mainly due to the consideration of characteristics as in Table 3.1.

Parameter	Specification
Measurement Range	20-90%RH 0-50 degrees celsius
Humidity Acurracy	+/- 5%RH
Temperature Accuracy	+/-2 degrees Celsius
Response Time	10sec

Table 3.1: DHT-11 Sensor specification

  The sensors connected to the BBC Micro bit- a pocket-sized computer that shows how software and hardware work together; board for data processing as the signal generated is in electrical pulse to be translated by BBC Micro bit based on the driver program, obtained from the manufacturer, combined into the 'temperature.py' program written inside the board. The 'temperature.py' program is a set C codes and command to measure the temperature inside the cooler box every 5 minutes; to translate the electrical signal to temperature; and to push the data to the Firebase Platform in the Middleware.

- Communication Layer
  The wrapped data from the device layer is then transported through the communication layer via 4G using TCP/IP. For security purpose, a pre-set API key and Channel ID was assigned to the system. This information is embedded into the 'coldbox.py' program together with authentication details to sign on process to the Firebase Platform.

- Application Layer
  The application is defined as the layer to process the data. In this layer, the 'business logic' where the temperature breaches are calculated. For the purpose, the project

utilised the embedded features of Matlab. The code will check for the temperature breach and send in the email to the admin of the CCL process. At this layer also where the data will be to create the blockchain for data integrity. Duly noted that the IoT system is resource constraint. Hence, the project must choose the most suitable platform for the project.

Previous research by investigated a model with IOTA/Tangle network which received data from wireless multi-sensor device, Bosch XDK110. Motivated from study by , for this project, IOTA was selected as the Blockchain platform. This is due to the fact it is much easier to setup and scalable with the increase of the size of the network if compare to the other platform. Furthermore, the IOTA offers a hashing option between SHA3 and SHA256. With a lighter option, it will reduce the needs for higher processing at the IoT system.

- Setup Consideration
  As an overview, IoT is an integrated system comprised of devices to collect the data, connectivity, storage, data processing and visualization applications. However, the fidelity of the information generated and the autonomous process from the innovative applications created concerns of security and privacy that may hampered further progress of IoT. While Blockchain is a technology that warrant the data integrity by using the immutable ledger in a distributed network. The ledger is connected with has value of each previous block, and the ledger, including all of the information about the transaction is stored in each node of the distributed network. The distributed ledger is maintained by making consensus using define consensus algorithm to achieve same data blocks across all nodes in the network.

Nevertheless, the three biggest challenges for implementing Blockchain in IoT system are:

1. Scalability, whenever the transaction grows may lead to centralisation for the administration of registries. Consequently, one of the feature of Blockchain of distributed network will be eliminated.

2. Computing power and time, IoT devices is known to have very small computing power which insufficient to run a complex encryption algorithm.

3. Storage, IoT devices may not have storage capacity to store the copy of all the transaction all nodes keep a copy of all transactions that ever occurred in the blockchain since its creation. The size will increase as time goes by and IoT devices might not be able to store it.

3.2 UML Diagram

3.2.1 Use case diagram:

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. In this use case diagram, we have defined what use cases will be available for the primary actor like the user. And the use cases need a secondary actor like FirebaseAuth, Database and in Figure 3.1. Some use cases have included use cases or extended use cases e.g. while login, web application needs to verify the email and password details before directing the user into the application. So, the login use case must have verify but it may not be necessary to have a login error use case every time. Some use cases need to be performed on the secondary actor side like for verification app sends the request to firebase for authentication. The user can log in, Register, Create an order, Look for status of product, View the current location of the shipment and The Location data can be accessed from the Analysis use case.

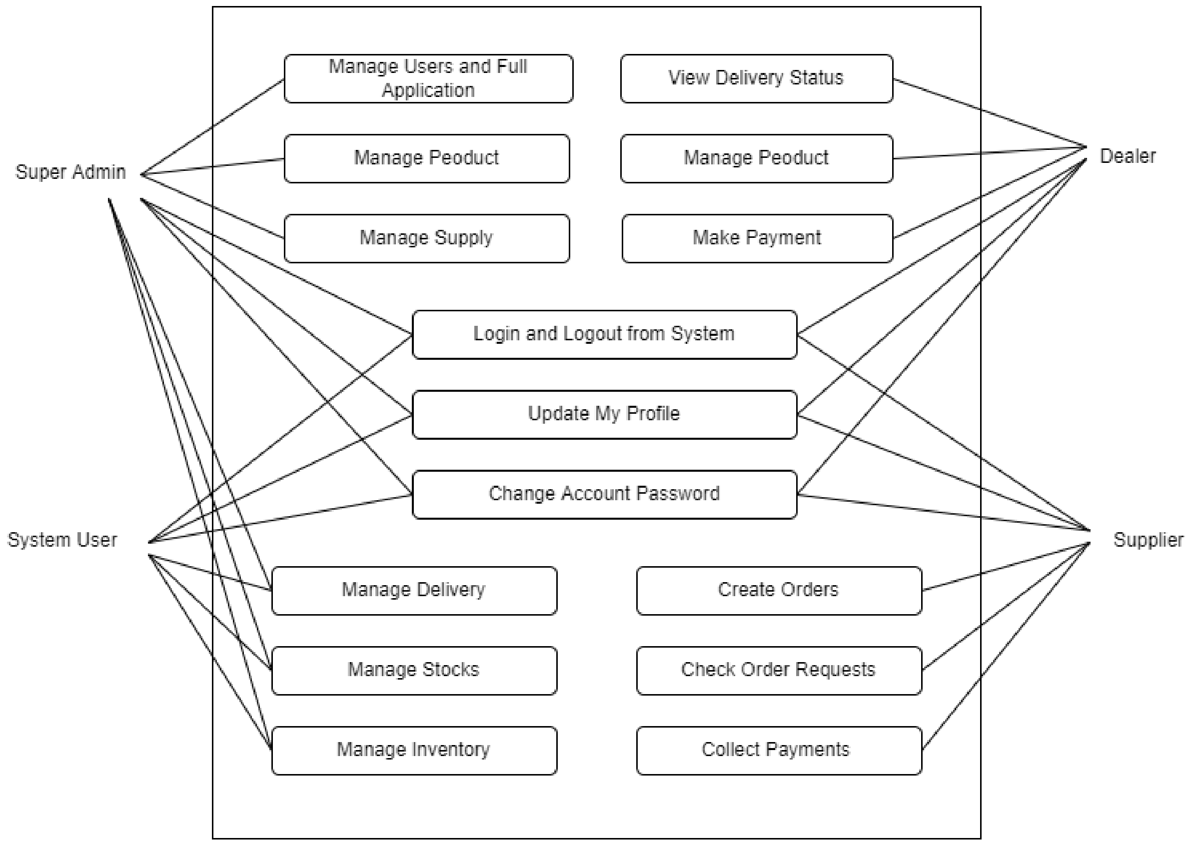


Figure 3.1: Usecase Diagram of Product Supply Management System

3.2.2 Class Diagram:

Class diagram is a static diagram. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. In the class diagram, the permission class is the main class which is the first thing to execute before implementing different methods. The attributes needed for the login class are email and password. And the operations performed are checking the fields, validating email and authenticating the user to log in. The snapshot of the user details is stored. So, The user doesn't need to log in every time using the application. The handleAuth operation checks the state of user details and redirects to the home page in Figure 3.2. The user then can perform different actions from the classes on the home page

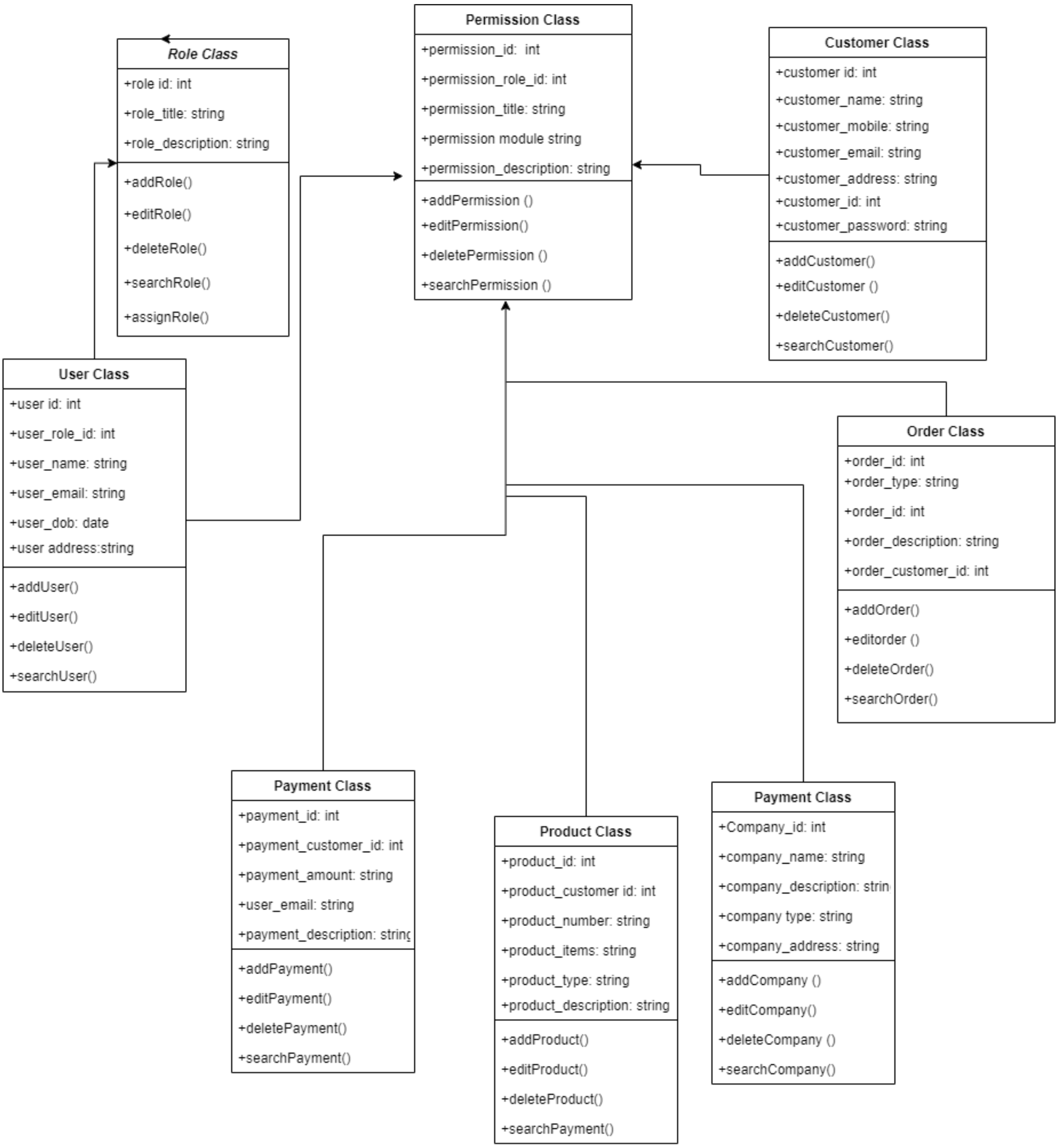


Figure 3.2: Class Diagram for Supply Chain Web application

Chapter 4

Project Implementation

Building the project was divided into three parts:

1. Supply chain application building.
2. Devloping a blockchain infrastructure environment.
3. Building an embedded IoT device for monitoring purpose.

4.1 Supply chain Web app implementation:

The web framework to provide the users with an easy to use frontend to interact with the system. The below fig is a snapshot of the main.js file which shows the function calling from web page to fetch the values of various functions.

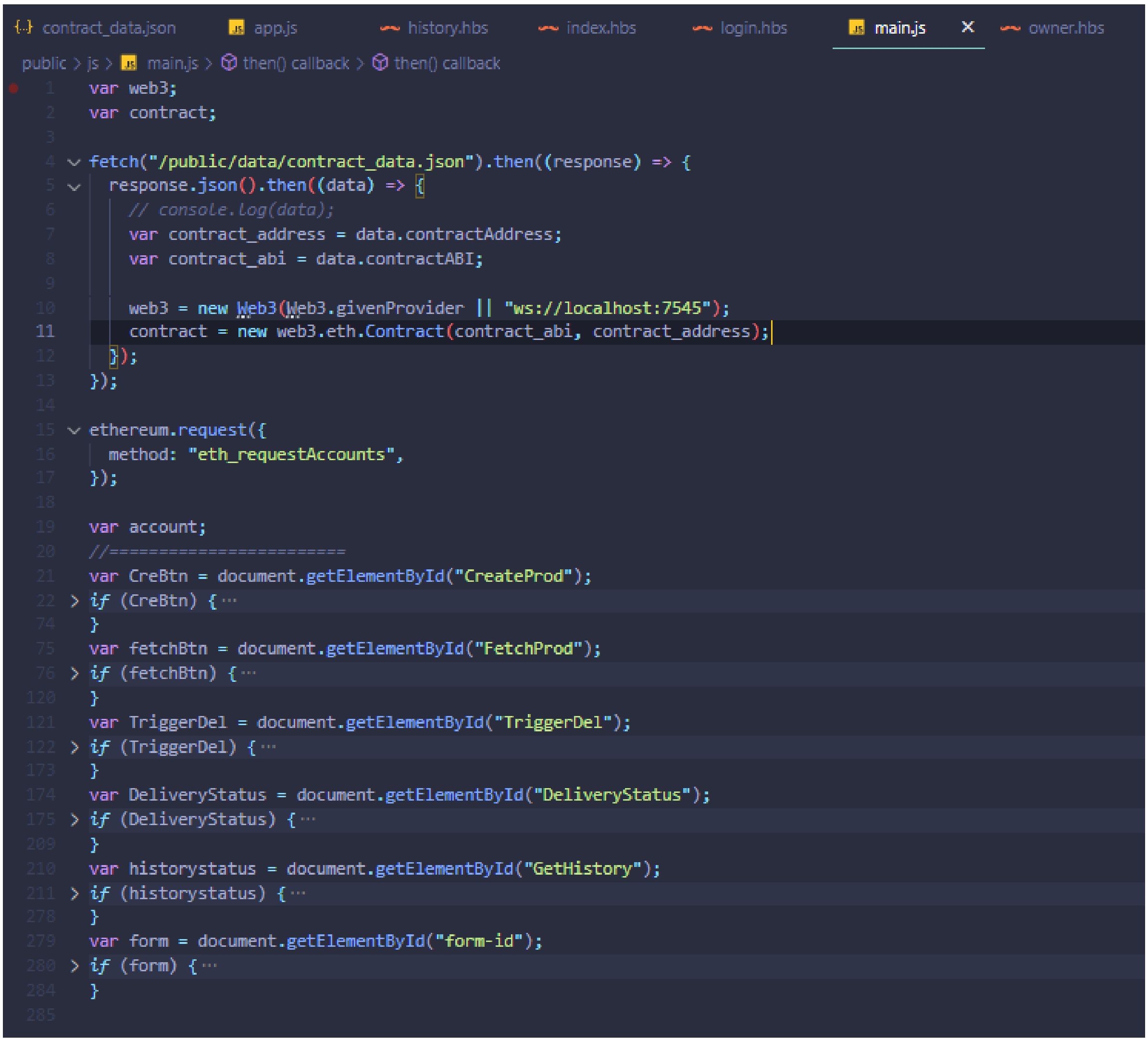
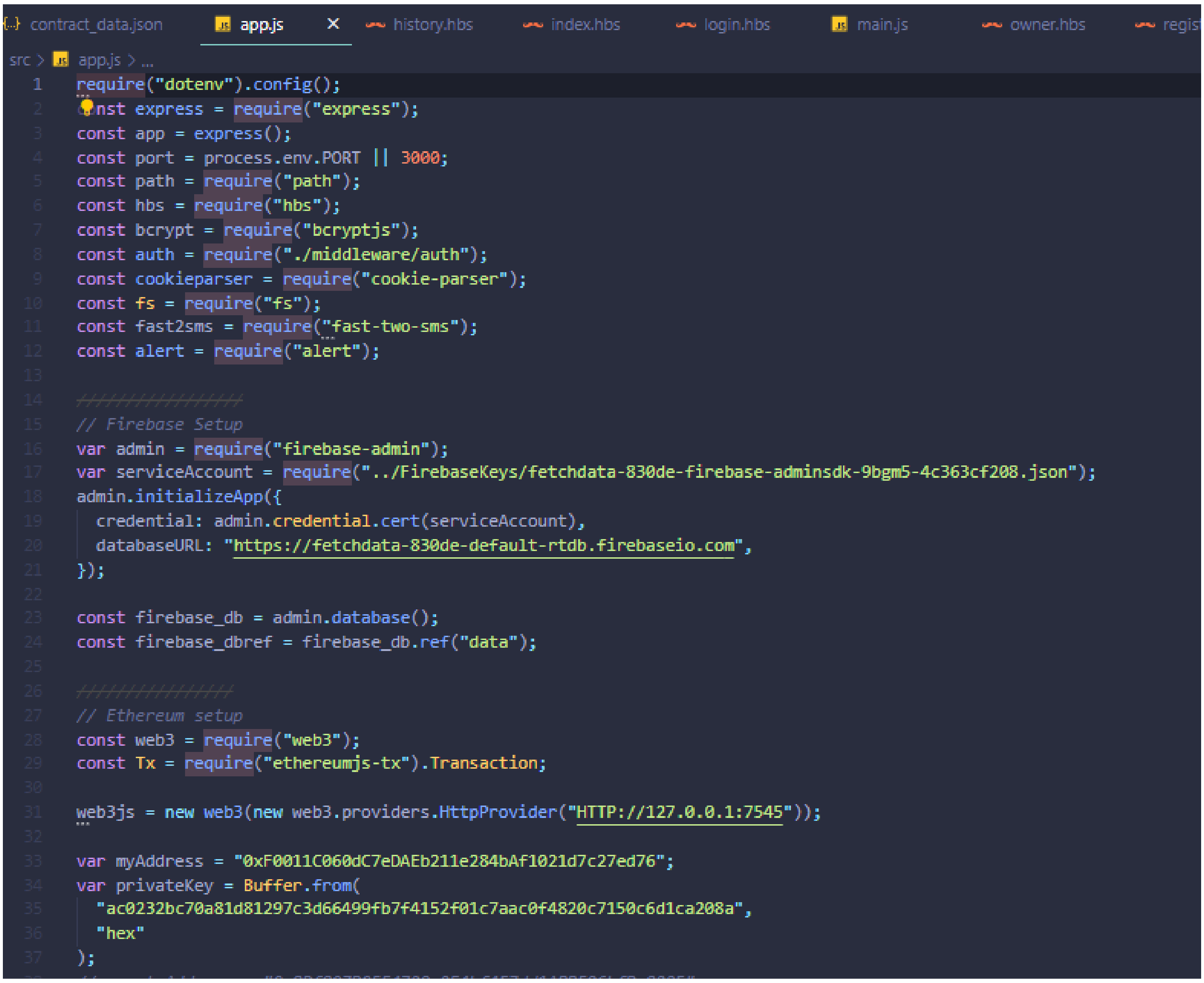


Figure 4.1: main.js file showing the function calling from webpage

Figure 4.2 is a snapshot of the app.js file which shows the Libraries imported and the Database and web3 connections done for the login and user registration purpose. Database used is Firebase database by google as it provides high scalability.

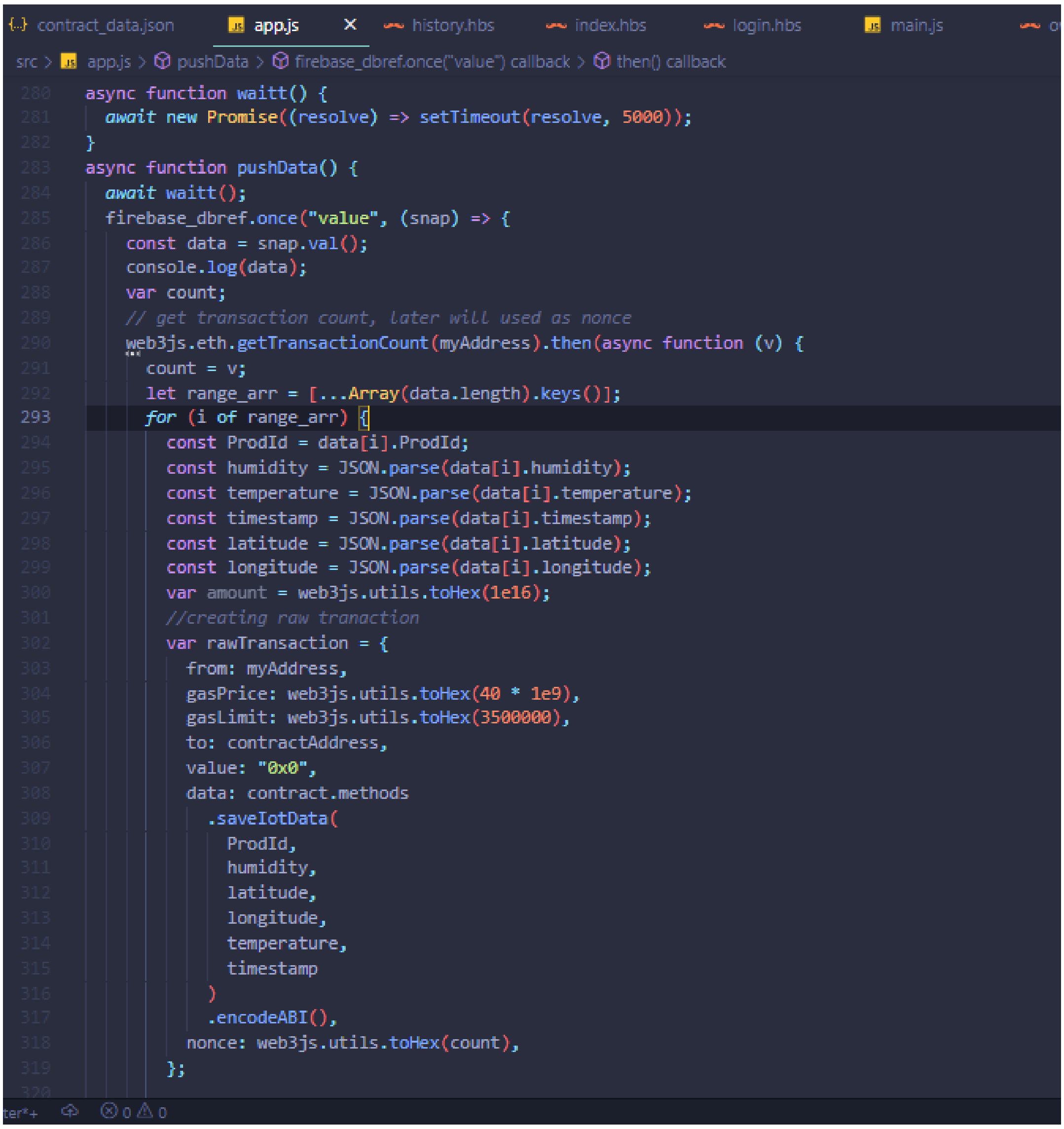


```
require("dotenv").config();
const express = require("express");
const app = express();
const port = process.env.PORT || 3000;
const path = require("path");
const hbs = require("hbs");
const bcrypt = require("bcryptjs");
const auth = require("./middleware/auth");
const cookieparser = require("cookie-parser");
const fs = require("fs");
const fast2sms = require("fast-two-sms");
const alert = require("alert");

////////////////
// Firebase Setup
var admin = require("firebase-admin");
var serviceAccount = require("../FirebaseKeys/fetchdata-830de-firebase-adminsdk-9bgm5-4c363cf208.json");
admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://fetchdata-830de-default-rtdb.firebaseio.com",
});

const firebase_db = admin.database();
const firebase_dbref = firebase_db.ref("data");

///////////////
// Ethereum setup
const web3 = require("web3");
const Tx = require("ethereumjs-tx").Transaction;

web3js = new web3(new web3.providers.HttpProvider("HTTP://127.0.0.1:7545"));

var myAddress = "0xF0011C060dC7eDAEb211e284bAf1021d7c27ed76";
var privateKey = Buffer.from(
  "ac0232bc70a81d81297c3d66499fb7f4152f01c7aac0f4820c7150c6d1ca208a",
  "hex"
);
```

Figure 4.2: The app.js file showing the libraries imported, Database, Web3 connetions

4.2 Developing blockchain infrastructure:

One of the major concerns while developing the project was how to transfer the sensor data automatically from the hardware devices to the blockchain for data transparency and sharing over to other parties.Figure 4.3 shows the automation process through which data is fetched from firebase into the blockchain.



```
async function waitt() {
  await new Promise((resolve) => setTimeout(resolve, 5000));
}
async function pushData() {
  await waitt();
  firebase_dbref.once("value", (snap) => {
    const data = snap.val();
    console.log(data);
    var count;
    // get transaction count, later will used as nonce
    web3js.eth.getTransactionCount(myAddress).then(async function (v) {
      count = v;
      let range_arr = [...Array(data.length).keys()];
      for (i of range_arr) {
        const ProdId = data[i].ProdId;
        const humidity = JSON.parse(data[i].humidity);
        const temperature = JSON.parse(data[i].temperature);
        const timestamp = JSON.parse(data[i].timestamp);
        const latitude = JSON.parse(data[i].latitude);
        const longitude = JSON.parse(data[i].longitude);
        var amount = web3js.utils.toHex(1e16);
        //creating raw tranction
        var rawTransaction = {
          from: myAddress,
          gasPrice: web3js.utils.toHex(40 * 1e9),
          gasLimit: web3js.utils.toHex(3500000),
          to: contractAddress,
          value: "0x0",
          data: contract.methods
            .saveIotData(
              ProdId,
              humidity,
              latitude,
              longitude,
              temperature,
              timestamp
            )
            .encodeABI(),
          nonce: web3js.utils.toHex(count),
        };
```

Figure 4.3: Pipeline for data migration from firebase to blockchain

Figure 4.4 shows the execution of smart contracts. As we know in building a blockchain infrastructure smart contracts are an integral part of the system, because these are the policies which are set and help all the nodes to know their permissions or rights and communicate with each other.

```
pragma solidity >=0.5.0 <0.9.0;
// SPDX-License-Identifier: MIT
contract test{
     struct Prod {
        uint256 ProdId;
        string ProdName;
        string ProdQty;
        string CreatedDate;
        string PickedDate;
        string DeliveredDate;
        bool IsPicked;
        bool Isdelivered;
        string CurrentCustody;
        string Buyer;
        string Owner;
        string IotID;
    }
    struct IOTData {
        uint[] humidity;
        uint[] temperature;
        string[] latitude;
        string[] longitude;
        uint256[] timestamp;
    }

    mapping(uint256 => Prod) public Items;
    mapping(uint256 => uint[]) public IotData_humidity;
    mapping(uint256 => uint[]) public IotData_temperature;
    mapping(uint256 => string[]) public IotData_latitude;
    mapping(uint256 => string[]) public IotData_longitude;
    mapping(uint256 => uint256[]) public IotData_timestamp;
```

Figure 4.4: Boiler plate for smart contracts

Figure 4.5 Creating product and Triggering delivery smart contract. Figure 4.5 describes the defined smart contract functions which are:

- CreateProd- which helps a registered manufacturer in creating a product on the system
- TriggerDelivery- when the container is procured with the desired items then this trigger can be used and the shipment can be sent for delivery

```
function CreatedProd(uint id,string memory _ProdName,string memory _quantity, string memory _CreatedDate,string memory _Owner,string memory _Buye
 {
     Items[id].ProdId = id;
     Items[id].ProdName = _ProdName;
     Items[id].ProdQty = _quantity;
     Items[id].CreatedDate = _CreatedDate;
     Items[id].IsPicked = false;
     Items[id].Isdelivered = false;
     Items[id].Owner = _Owner;
     Items[id].Buyer = _Buyer;
     Items[id].CurrentCustody = _Owner;
     return id;
     // return block.timestamp;
 }
 function GetProd(uint id) public view returns(Prod memory){
     return Items[id];
 }
 function TriggerDelivery(uint id,string memory _Date,string memory _VehicleNo,string memory _IotId) public{
     require(Items[id].IsPicked == false, "Already Triggered Delivery");
     Items[id].PickedDate =  _Date;
     Items[id].IotID = _IotId;
     Items[id].CurrentCustody = _VehicleNo;
     Items[id].IsPicked = true;
 }
  function DeliveredDate(uint id,string memory _Date) public{
     require(Items[id].IsPicked == true,"Your Product has Not picked Yet");
     require(Items[id].Isdelivered == false, "Already Delivered");
     Items[id].Owner = Items[id].Buyer;
      Items[id].CurrentCustody = Items[id].Buyer;
     Items[id].DeliveredDate = _Date;
     Items[id].Isdelivered = true;
 }
```

Figure 4.5: The Create product and Trigger delivery Smart contract

Figure 4.6 depicts the smart contract function for saving the IoT data, and fetching that IoT data everytime the block makes a call to build a new block on the blockchain & then distributing that data to all the nodes present at that time.

```
    ///////

    function saveIotData(uint id, uint[] memory humidity, string[] memory latitude, string[] memory longitude, uint[] memory temperature, uint256[] memory
        uint n = humidity.length;

        for(uint i=0; i<n; i++) {
            IotData_humidity[id].push(humidity[i]);
            IotData_temperature[id].push(temperature[i]);
            IotData_latitude[id].push(latitude[i]);
            IotData_longitude[id].push(longitude[i]);
            IotData_timestamp[id].push(timestamp[i]);
        }
    }

    ///

    function getIOTData(uint id) public view returns(IOTData memory){
        IOTData memory data;
        data.humidity = IotData_humidity[id];
        data.temperature = IotData_temperature[id];
        data.latitude = IotData_latitude[id];
        data.longitude = IotData_longitude[id];
        data.timestamp = IotData_timestamp[id];

        return data;
    }
}
```

Figure 4.6: The push and pull request smart contracts for IoT data

Chapter 5

Testing

5.1 Functional Testing

5.1.1 Unit Testing:

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests before the software or feature is passed over to the test team. Unit testing can be conducted manually. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. The Unit testing is best suited for our application development phase. In that phase, we started to code in units create different modules. And test each module separately, like the login page, register page, home page, history page, live status page etc. All these pages are tested and debugged before going further integrating. And check whether we are getting the desired output from each module as for the objectives.

5.1.2 Integration Testing:

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests. As we have discussed unit testing the next step is integration testing. All the units which we have tested and debugged are now ready to integrate into a whole single module. The integration part is crucial as we need to know which unit must interact without error, calling them in a different class accessing the instance of that class all these can be cleared with help of the class diagram which was represented in Project Design. So accordingly, modules are integrated and checked whether they behave as for the objectives.

5.2 Non Functional Testing

5.2.1 Compatibility Testing :

Compatibility testing is used to gauge how an application or piece of software will work in different environments. It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The goal is to ensure that your software's functionality is consistently supported across any environment you expect your endusers to be using. The framework we are using to develop our application is Java script handlebars and Firebase Database. It is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase. We make sure that our web-application is compatible with both IOS and Android operating systems. The features we developed are perfectly run in multiple operating systems without an error. For this reason, compatibility testing is best suited for our project.

Chapter 6

Result

6.1 Home page

We have implemented a minimalistic UI that if a user firstly interacts with an application, it would be more attractive and user friendly. When the user interacts with the application first time can register and log into the application. If the user is already registered but forgot the password there is a forgot password option available simply by clicking it user can reset the password. After login, the user will be able to access the home page.

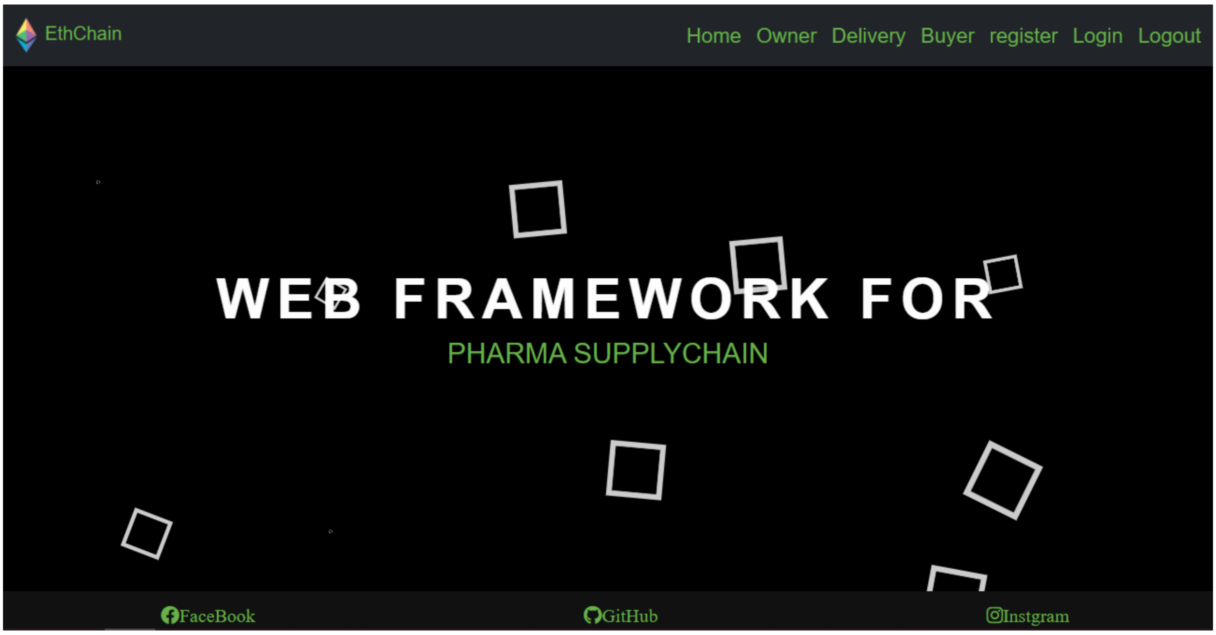


Figure 6.1: Home Page

6.2 Seller's page for creating product:

Figure 6.2 shows the Seller's Page.Here entering a proper product ID according to the inventory helps in proper management of products. Also mentoning the mobile number of the receiver is mandatory to generate a secrete code for status tracking.

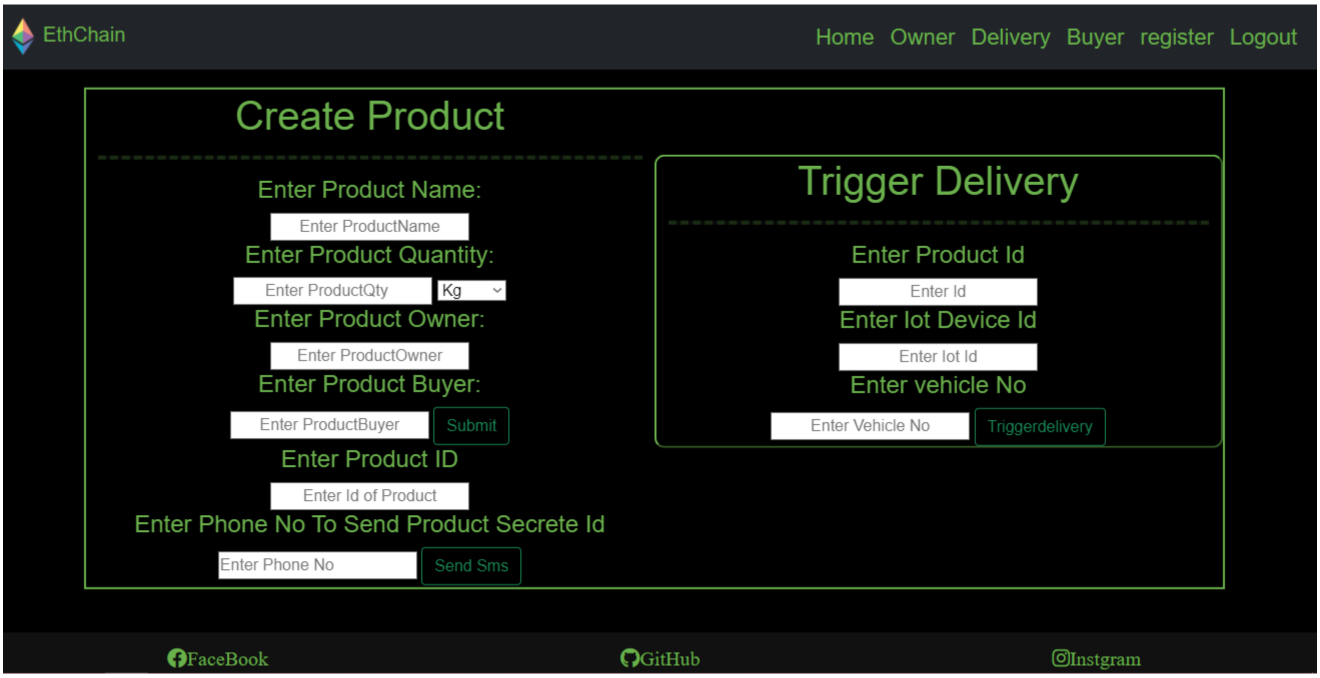


Figure 6.2: Seller's Page

6.3 Checking the product details section

Figure 6.3 Buyer Page. Here as the buyer enters the product id generated by the seller, the buyer can get all the details of the shipment right from the delivery date to product created date.

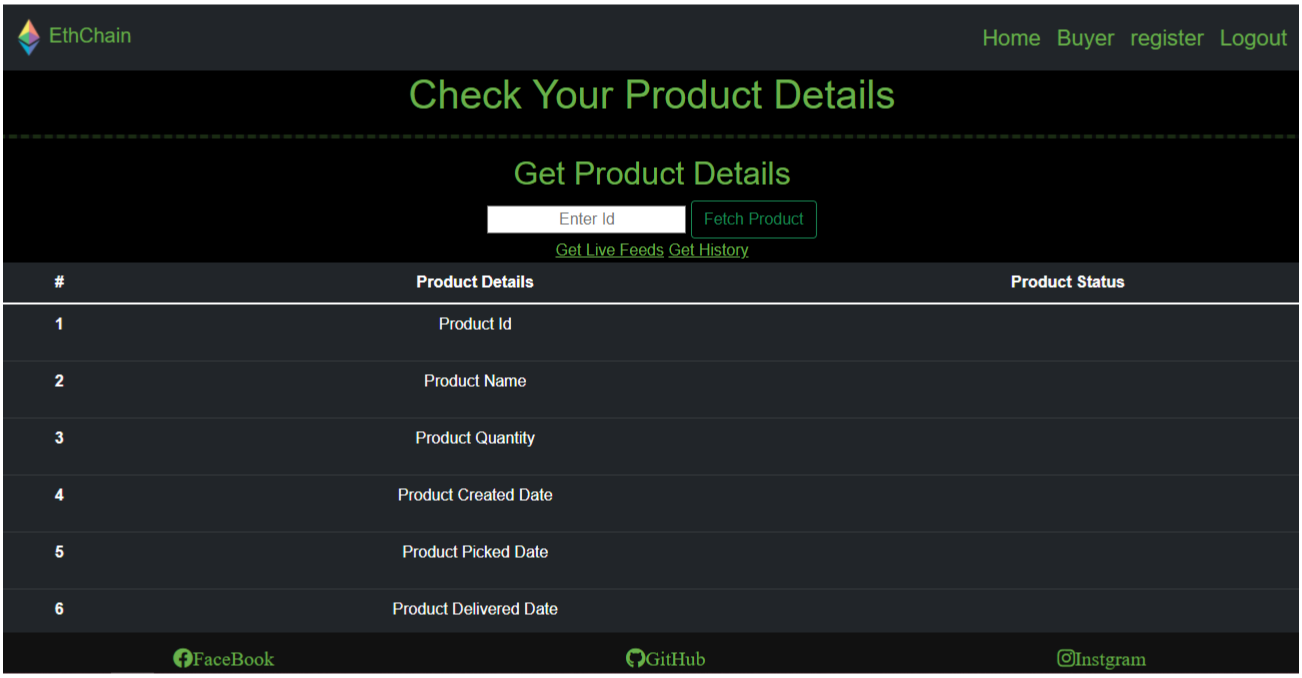


Figure 6.3: Buyer Page

6.4 Start Delivery page:

As the procurement of the required container is done the delivery can be triggered by the seller's end and the container can be handed over to the transporter.

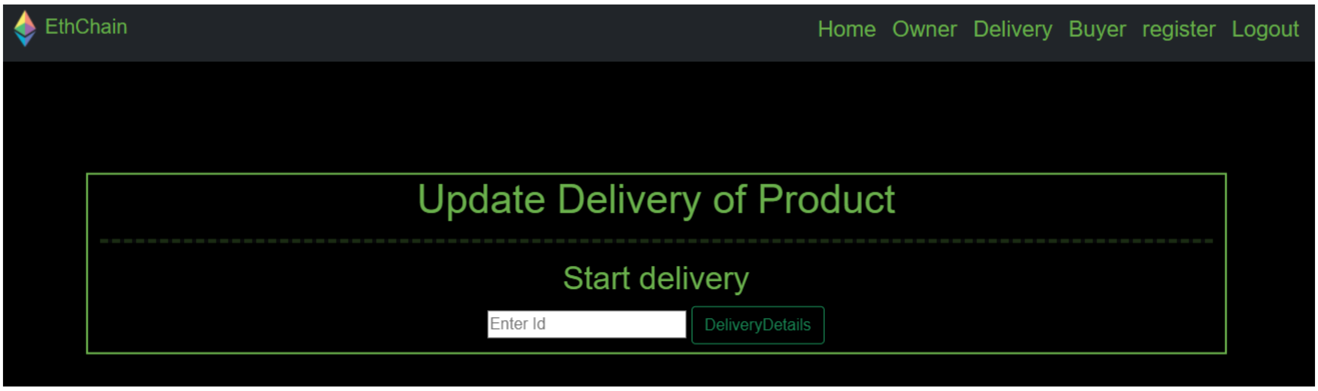


Figure 6.4: Delivery Page

6.5 Live Product Tracking page for Product status:

The most interesting feature and the heart of this system is live monitoring and tracking of a product, and adding the same data on to the blockchain.

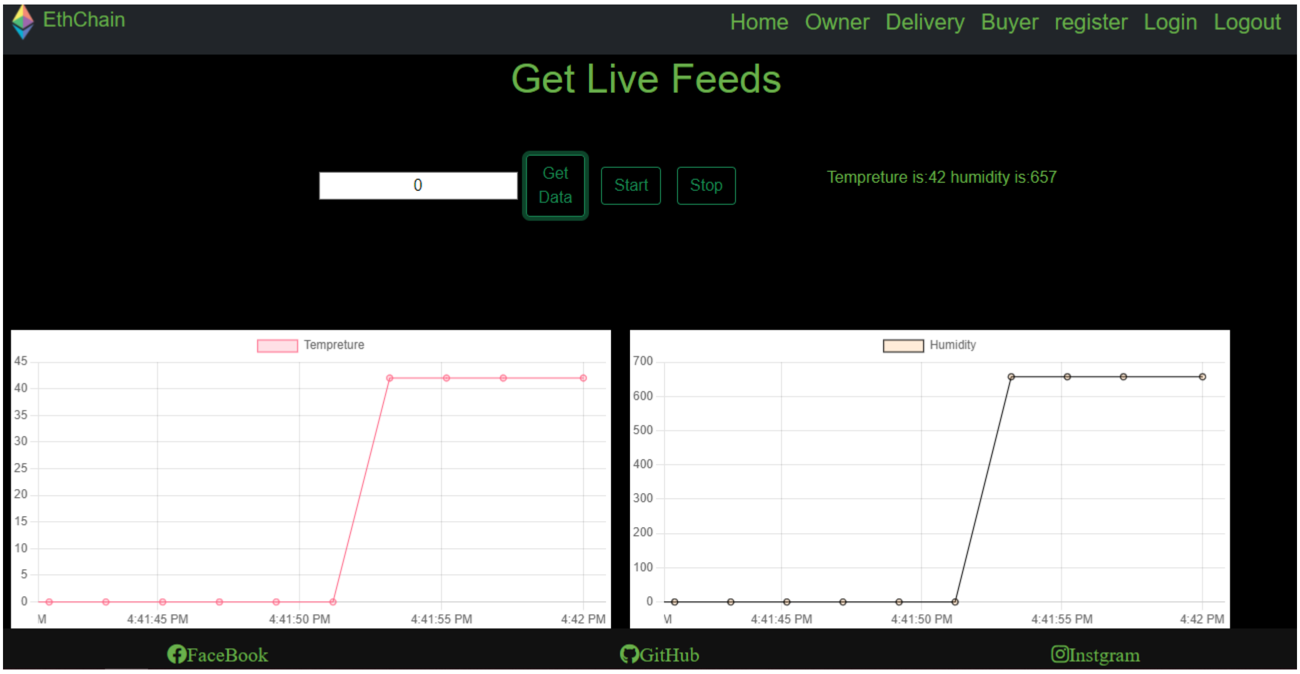


Figure 6.5: Live Tracking

6.6 History page:

Figure 6.6 in this page is added to display the history of the sensor data containing last five values which is fetched directly from the backend of blockchain.

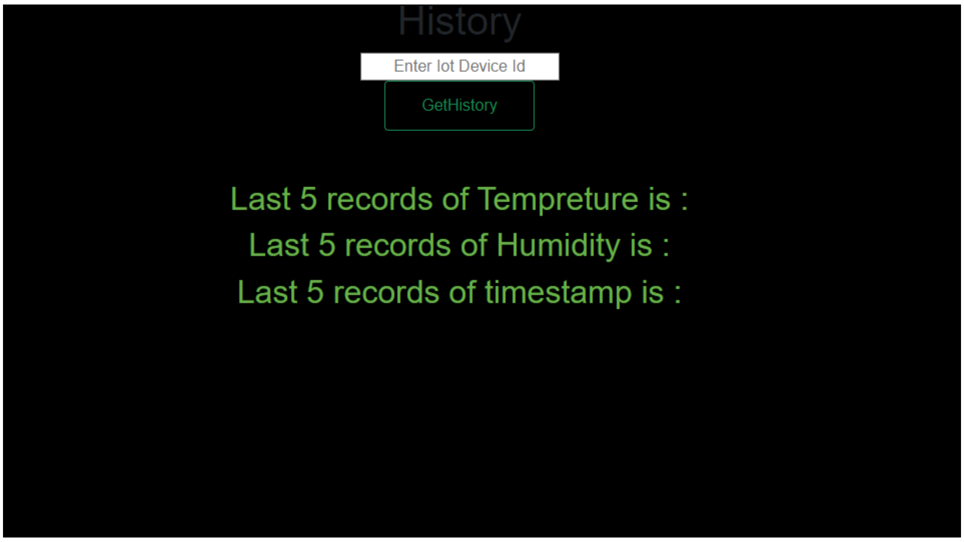


Figure 6.6: History Page

Chapter 7

Conclusions and Future Scope

Blockchain technology gets a hype in the Information Technology world because of its security, immutability, transparency, and decentralization as best defined as "Blockchain is shared, immutable ledgers for recording the history of transactions. It fosters a new generation of transactional applications that establish trust, accountability, and transparency-from contracts to deeds to payments". But there is a lot of misconceptions about blockchain technology; such as some people think smart contract carry the same legal value as a normal contract but it's not true, the smart contract is a piece of code which resides on blockchain and execute upon meeting a certain condition. Similarly, most of the people think of cryptocurrencies when they hear the word blockchain, but blockchain is not cryptocurrency it is the underlying technology used for the cryptocurrencies and it can be used for any other non-financial applications as well.
In chapter 3 of project design we have come across three major issues while integrating Iot technology with Blockchain they are:

1. Scalability, whenever the transaction grows may lead to centralisation for the administration of registries. Consequently, one of the feature of Blockchain of distributed network will be eliminated.

2. Computing power and time, IoT devices is known to have very small computing power which insufficient to run a complex encryption algorithm.

3. Storage, IoT devices may not have storage capacity to store the copy of all the transaction all nodes keep a copy of all transactions that ever occurred in the blockchain since its creation. The size will increase as time goes by and IoT devices might not be able to store it.

From the above work we come have concluded that if we try to fill the holes in the existing system, even then the new system developed may have its own issues and challenges. For our system we have used IOTA based Ethereum blockchain which also includes the gas fees for initiating any transaction so this makes our solution a lot more costlier, to tackle with ths problem we have certain things to be done as future work they are: Adopting a hyperledger based architecture which lets us build a custom blockchain such as a bitcoin framework , this will help in eliminating the cost of gas as you don't need an eternal wallet. Also to make this system user friendly we will have to build a proper Platform as a Service (PaaS) business model so that businesses can subscribe for our Blockhain plus IoT platform and use all the functionalities right from the supply chain and invoice management app till the hardware devices needed to be added to every container.

Bibliography

[1] Guojun Ji and Rong Guo (2009). "Research on the security of cold-chain logistics," 2009 6th International Conference on Service Systems and Service Management, pp. 757-761, doi: 10.1109/ICSSSM.2009.5174982.

[2] Y. Xv, X. Zhang, X. Qiu and X. Liang (2020). "Analysis of cold chain development based on ISM model under the situation of (COVID-19)," 2020 16th Dahe Fortune China Forum and Chinese High-educational Management Annual Academic Conference (DFHMC), pp. 254-257, doi: 10.1109/DFHMC52214.2020.00055.

[3] D. Zhang and T. Han (2020). "Analysis of risk control factors of medical cold chain logistics based on ISM model," 2020 Chinese Control And Decision Conference (CCDC), pp. 4222-4227, doi: 10.1109/CCDC49329.2020.9164042.

[4] Shih, Chih-Wen and Chih-Hsuan Wang (2015). "Integrating wireless sensor network with statistical quality contorl to develop a cold chain system in food industries," Computer Statndards & Interfaces: 62-78.

[5] Monteleone, Sergio and Mario Sampaio (2017). "A novel deployment of smart Cold Chain system using 2G-RFID-Sys temperature monitoring in medicine Cold Chain based on Internet of Things," 2017 IEEE International Conference on Service Operations and Logistics, Informatics, Basel, Switzerland: MDPI. 205-210.

[6] Ruan, Junhu and Yan Shi (2016). "Monitoring and assessing fruit freshness in IoT-based e-commerce delivery using scenario analysis and interval number appro aches," Information Science: 557-570.

[7] Chandra, Abel Avitesh and Seong Ro Lee (2014). "A Method of WSN and Sensor Cloud System to Monitor Cold Chain Logistics as part of the IoT Technology," International Journal of Multimedia and Ubiquitous Engineering: 145-152.

[8] L. Ding, J. Wang and L. Li (2015). "Privacy-Preserving Temperature Query Protocol in Cold-Chain Logistics," 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 113-116, doi: 10.1109/IHMSC.2015.98.

[9] N. N. Ahamed, T. K. Thivakaran and P. Karthikeyan (2021). "Perishable Food Products Contains Safe in Cold Supply Chain Management Using Blockchain Technology," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 167-172, doi: 10.1109/ICACCS51430.2021.9442057.

[10] T. Bengiovanni et al. (2020). "Risk Management and Healthcare: IoT Technologies and Smart Monitoring System for a Good Cold Chain Management," 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech), pp. 1-6, doi: 10.23919/SpliTech49282.2020.9243821.

[11] V. Voicu, D. Petreus and R. Etz (2020). "IoT Blockchain for Smart Sensor," 2020 43rd International Spring Seminar on Electronics Technology (ISSE), pp. 1-5, doi: 10.1109/ISSE49702.2020.9120915.

[12] Usama Salama, Lina Yao and Hye-young Paik (2018). "An Internet of Things Based Multi-Level Privacy-Preserving Access Control for Smart Living," Informatics (Basel), vol. 5, no. 23, doi:10.3390/informatics5020023.

[13] S. Sawardekar and R. Pawar, "Data Security Approach in IoT Environment," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-7, doi: 10.1109/ICCCNT45670.2019.8944831

[14] Makhdoom, I., Abolhasan, M. and Ni, W. (2018). "Blockchain for IoT: The Challenges and a Way Forward," 15th International Joint Conference on e- Business and Telecommunications, pp. 594–605.

[15] W. F. Silvano, D. De Michele, D. Trauth and R. Marcelino (2020). "IoT sensors integrated with the distributed protocol IOTA/Tangle: Bosch XDK110 use case," 2020 X Brazilian Symposium on Computing Systems Engineering (SBESC), pp. 1-8, doi: 10.1109/SBESC51047.2020.9277865.