Design and Development of a Crop Quality Monitoring and Classification System using IoT and Blockchain

M Sangeetha¹*, Goriparthi Thejaswini², A Shoba³, S Santoshi Gaikwad⁴, R T Amretasre⁵, and S Nivedita⁶

¹Department of Electronics and Communication SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India
²Blue Binaries Engineering and Solutions Pvt. LTD, Chennai, India
³Deloitte US-India, Hyderabad, India
⁴Schneider Electric India Pvt. LTD, Chennai, Tamil Nadu, India
⁵Cognizant Technology Solutions, Chennai, Tamil Nadu, India
⁶WIPRO Chennai, Tamil Nadu, India

Email: *sangeetm@srmist.edu.in

Abstract. In this paper, an Internet of Things (IoT) based Crop Monitoring and Classification system is proposed for automated sensing, storing, and monitoring real-time parameters that play an important role in determining a crop's quality and yield. Sensors are placed in-situ in the field and the warehouse to monitor the crop. The long-Range wide area network (LoRa) module is used for communication between the sensing unit placed at the field, warehouse, and data processing unit. The yield is classified based on qualitative analysis posed by the imperative sensor data. Further, to enable an equitable gateway of resource sharing between the distributor and the farmer, a Blockchain-based transaction is taught to enhance trust and security. This proposed method aims to eliminate intermediaries in the trade, thereby helping farmers get the price for their product details stored in an immutable database, which also displays the farmer's quality of the crop reaped.

Keywords: Ethereum Blockchain, Internet of Things (IoT), Internet of Underground Things (IOUT), Long-range Communications.

1. Introduction

Agriculture is the lifeblood of India, with nearly two-thirds of the Indian population dependent on it. It is a source of livelihood and the main source of food, fodder, and fuel. Agriculture in India is currently facing numerous hurdles in financial instability due to different forms of risks involved in the production, marketing, and pricing of commodities. The main reasons for such problems are unfavorable climatic conditions, unpredictable weather, lack of adequate rainfall, poor irrigational facilities, and frequent crop failures. Farmers cannot sell their goods at a decent price in the regular market due to intermediaries who get benefited from the farmers by getting their commodities at a lesser price. This paper aims to directly connect the farmers and the distributors, eliminating the middlemen using Blockchain technology.
Blockchain networks consist of farmers, distributors, and consumers who can communicate and share information by registering themselves. This data sharing is done with the utmost safety, speed, and transparency. Once the data is entered, it becomes visible to all the registered members in the Blockchain network, thereby giving them the freedom to approve or reject information. Upon validation of data, it gets recorded in Blocks, which gets organized into a chronological chain that no one can alter. This technology enables the depiction of the entire agricultural cycle in a blockchain that provides a transparent and trusted source of information to the farmers. The farmers have ready access to data relating to the quality of seeds, soil moisture, environment, and climatic conditions, payments, sale price, and demand—everything from one platform. Blockchain thus acts as a direct link between farmers and distributors, eliminating intermediaries in the process. The small farmers are immensely benefited as they can connect and reach the market without middlemen’s help.

This ensures that farmers get the right price for their produce, thereby increasing their income, giving more transparency in the supply chain process.

2. Literature Survey

The Global tomato production is 130 million tons, of which 88 million tons are offered at a fresh market, and the remaining 42 million tons are sent to processing units. The dismal fact is that 42% account for postharvest losses. The pre-harvest problems are mainly due to inadequate fertilization, insufficient irrigation facility, poor pruning activities, etc... In contrast, the postharvest problems are caused by wrongful physical handling, calcium chloride application, and insufficient and improper storage facilities. To overcome these problems, both pre and postharvest factors need to be properly and efficiently managed [1]. In a world dominated by digital technology, IoT plays a significant role in our day-to-day life. It has created an ecosystem by linking many systems, thereby giving a smart performance to every task. Thus, IoT has become inevitable, and agriculture is no exception to it. In this paper, the authors have attempted to devise Blockchain-based traceability in Agrifood Supply Chain Management. The shortcomings of data integrity, tampering, single-point failures in earlier IoT-based traceability, and providence system can be overcome by introducing the Blockchain.

The Agri-Block IoT – is a fully decentralized Blockchain-based traceability solution for agriculture, food supply management. It explains the classical use case between domains, from-farm-to-fork. It uses two different Blockchain, namely Ethereum and Hyperledger Sawtooth, and finally evaluates by comparing both in terms of latency, CPU, and network usage. Thus, the Agri-Block IoT, IoT, and Blockchain technology integration provides transparent, fault-tolerant, immutable, and auditable records for the Agri-Food traceability system [2]. Ethereum uses a smart contract on the Ethereum Virtual Machine for different applications to use decentralization, making it more viable for different applications [3]. The request for Blockchain implementation is the hosting location. The Edge technology is popular but lacks in computation resources. Studies suggest using cloud and fog to solve the problem of hosting location [4].

A design made by recapturing Blockchain as accelerated–moderator is achieved so that the user can own and control data. This enables instruction carrying, storing, querying, and sharing of data [5]. An attempt to devise a Blockchain that resolves made the consensus problem in [6]. Malware variants give a wake-up call to secure the Internet of Things [7]. Blockchain-based systems are leveraged to strengthen IoT security as it provides decentralized access and immutability by which malicious actions can be detected and prevented [8]. Blockchain benefits agriculture in ample ways and thereby increases overall productivity and makes transactions hassle-free. The IoT devices with embedded systems have become an inevitable part of the modern lifestyle. The devices currently in use, viz., Zig-Bee, Wi-Fi, and Bluetooth, require high power. Hence they are not suitable for battery-operated devices. The need for long-distance communication with limited battery usage and high security has made the Long-Range Wide Area network (LoRa) a worthy alternative.
In this paper, an IoT-based Crop Monitoring and Classification system are proposed for automated sensing, storing, and monitoring real-time parameters that play an important role in determining a crop's quality and yield. The plant is chosen for the experimentation; Solanum *Lycopersicum* is commonly known as the “Tomato” because it produces fruit within 50-60 days and more susceptible to the parameters chosen. The yield is classified into three categories, namely A, B, and C, based on their quality. Category A denotes high-quality produce, category B medium, and category C low quality. The pre-harvest and postharvest parameters such as Ammonia \([15]\) & \([17]\) content, Soil Moisture, Temperature, Humidity, and CO\(_2\) play a significant role in determining the quality of tomato, which details are as follow in table 1.

| Location   | Parameters | Optimum Conditions |
|------------|------------|--------------------|
| On-field   | Ammonia    | 3 ppm – 5 ppm      |
|            | Soil Moisture | 70 % - 80 %       |
| Warehouse  | Temperature | 12.5 °C – 20 °C \([1]\) |
|            | Humidity    | 90 % - 95 % \([1]\) |
|            | CO\(_2\)    | 1 % - 5 % \([1]\)   |

### 3. System Architecture

Figure 1 shows the system architecture of the proposed Crop Quality Monitoring and Classification System. It consists of IoT sensors placed on-field and the IoT sensors placed in a warehouse. Arduino microcontrollers and LoRa transmitters are placed at each location, process the collected data, and transfer to the LoRa receiver connected to Arduino, which is serially connected to a Raspberry Pi act a gateway node for the proposed set up depicted in Figure 1. The collected real-time data is continuously stored in a database, in this case, the Azure database. This collected data is displayed to the Blockchain participants using a user interface developed using the .NET framework. The transaction held between the farmer and distributor is carried out through Ganache and Metamask, Ethereum based applications.

![Figure 1. Proposed Crop Quality Monitoring and Classification System Architecture](image)

#### 3.1 System Description

**3.1.1. Sensors and their Calibration:** The sensors are placed in on-field and warehouse measures. On-field sensors measure the soil moisture and the ammonia content present in the soil. In contrast, the warehouse
sensors measure temperature and humidity and the carbon dioxide level inside the warehouse.

3.1.1.1. **FC-28- Soil Moisture Sensor:** Soil moisture is a pre-harvest factor affecting postharvest tomatoes' postharvest quality. Because water makes up about 90-95% of the tomato, which implies that low water accumulation in the tomato will result in low soil moisture content and reduce the tomato size, on the other hand, the inflated amount of moisture in the soil will leech the essential micro and macronutrients from the soil, leading to yield loss. FC-28 has an operating voltage of 3.3- 5 V and a range of 0 - 100%. The sensor consists of two probes through which the current flows when placed in the soil. Wet soil conducts more current making the resistance low and hence moisture high. Dry soil conducts less current, making the resistance high and hence moisture low. This sensor is calibrated by checking the deviation between dry and wet soil weight. The deviation is then added to the code for correction and hence calibrated.

3.1.1.2. **Ammonia Sensor - MQ137:** Plants inhale nitrogen in the form of ammonium [13] (NH₄) and nitrate (NO₃), either from the mineralization of organic components soil from the fertilizers. But due to non-optimum soil conditions and the environment, NH₄ is converted into NH₃ at the surface of the soil due to the chemical reaction of the fertilizers with water molecules present in the soil and the surroundings. This ultimately results in the loss of soil nutrients. MQ137 has an operating voltage of 5V and a range of 10-300ppm, whose resistance various when exposed to a particular gas. It has an in-built heater that increases the temperature that reacts with the gas. The sensor should be pre-heated before use and is sensitive to ammonia (NH₃) gas and carbon monoxide (CO) gas. [16] To calibrate the ammonia sensor, 10 ml of 2% ammonia solution is taken, and the sensor and the solution are sealed in a container. The deviated value is recorded and later added to code for correction and calibration.

3.1.1.3. **Temperature and Humidity Sensor DHT-11:** High temperature can accelerate respiration and metabolic activities, which sequentially triggers ethylene production. This results in the ripening of the tomato at a faster pace. Meanwhile, chilling tomatoes' injuries are the consequence of storing them at a very low temperature, typically less than 10°C. Thus, maintaining an optimum temperature of 12.5°C - 20°C in the storage facility will maintain the quality of the harvested tomato. Thus, tomatoes stored at this temperature will have a longer shelf-life.

The temperature and humidity sensor (DHT-11) has an operating voltage of 3.3-5V with a temperature range from 0-50°C and a humidity range of 20-90%. It consists of two components- the humidity sensing component and a thermistor. The humidity sensing component has two electrodes with a moisture-holding substrate between them which changes the electrodes' resistance. The thermistor changes resistance with a temperature change. For calibration, a standard humidity environment is made in a sealed container by placing Sodium Chloride in it with the hygrometer and DHT-11 set-up. Leave it for 6 hours. Both the hygrometer and DHT-11 readings are noted. Later, NaCl is replaced with Silicon Moisture Absorber, and the procedure is repeated. The deviations are then added to the code.

3.1.1.4. **Carbon dioxide Sensor MG811:** Elevated amount of CO₂ will enhance the metabolic activity, thus reducing its shelf-life and degrading the quality. MG811 has an operating voltage of 6V with a range of 0-10000ppm. When the internal heating element is activated, this gas sensor responds to CO₂ gas by generating a small voltage in proportion to the amount of CO₂ gas present in the air exposed to the internal element. For calibration, the CO₂ sensor is pre-heated for 2-3 hours. Once the system has reached the "steady-state,"
the sensor's output voltage at 400ppm (atmospheric condition) is noted. Then the output voltage at 40,000 ppm is determined by the exhale concentration method. Substitute the respective voltage in the code.

3.1.2. Lora Module: Long Range (LoRa) [10]-[11] is a new wireless IoT connectivity operated at low battery power to transfer a small amount of data at short intervals over a long-range. This technology uses a star topology, and it has a central hub connected to all the other nodes; the complexity of the network is low; hence the power consumption is low. The modulation technique used by LoRa is called Chirp Spread Spectrum to achieve a significantly high communication range while maintaining low power characteristics, where chirp is a signal whose frequency increases or decreases over time which is multiplied with the wanted data signal. It adopts the ALOHA method in which the frame is sent only when there is any data to send; otherwise, no transmission occurs. Only if the frame is received successfully is another frame sent. Else the same frame is retransmitted. This technology offers good Quality of Service (QoS) and is immune to multipath, fading, and interference.

3.1.3. Gateway Node: Raspberry Pi is a small single-board computer. Connecting peripherals like Keyboard, mouse, and display to the Raspberry Pi can be used as a mini personal computer. Raspberry Pi is slower than a laptop or desktop but is still a computer that can provide all the expected features or abilities at low power consumption. Raspbian OS is the official Operating System (OS) that is efficiently optimized for Raspberry Pi. The standard 802.11x Wi-Fi is often the most obvious choice for an IoT application. An IoT device's requirements concerning its connectivity technology are the cost of deployment, ease of implementation, reliability, range, throughput, scalability, and security. Wi-Fi poses to be advantageous due to its low cost of infrastructure, good range, high throughput, and ease of implementation.

3.2 Software Architecture

Software architecture for the proposed system is shown in Figure 2. It consists of Web Applications and the Blockchain network design.

![Figure 2. Software Architecture](image)

Web application utilizes a typical client-server model. The client requests data using a mobile or a web browser or even as a desktop application. The server process the user’s request and send a response, which bypasses the Presentation layer, Business/ logic layer, data storage layer. For this prototype, an ASP.NET core framework is utilized to develop the web application. Entity Framework 5 is used on top of the ASP.NET core to enable data binding between the database server and the web application to server requests from the client.

A bottom-to-top approach of content elucidation is taken up for the software layers to explain a cohesive piece of explanation from an embedded systems perspective.
3.2.1. Data Layer: After the data is being processed with a specific data formatting structure in the gateway node, it is sent to the data layer for storage. Since a Blockchain network is implemented, the data layer would also serve as the Blockchain world state. For this application, an Azure SQL database on an Azure SQL server is utilized to store the data. Azure SQL is a managed cloud database provided by Microsoft as a part of Azure cloud services. It is a Platform as a Service (PaaS) with high scalability to set up availability zones for high-end usage. Since the prototype is initially not aimed at higher environments for commercial user handling, many advanced options such as availability zones or sets are not used.

A single SQL server that acts as an Always ON listener responds to all the client data requests. A single database is configured. Multiple tables are populated to accommodate the varying and dynamic data content from the sensor's integrated embedded systems, with a specific schema set for each table.

The database schema and the data layer architecture can be enhanced and scaled using SQL Clustering or even individual windows servers acting as a SQL server after customizing the configurations concerning emerging requirements. This is one reason for the prototype to be built cognitively utilizing a Microsoft service-based Technology stack. However, the prototype developed doesn't compromise on security. The server and the database only allow few users with their specific login details as an administrator into the system. On increasing demand and consumption, the security can be enhanced by using Azure Active Directory, which utilizes OAuth, a standard open-source access delegation.

3.2.2. Business or Logic Layer: As specified, the web application uses an ASP.NET core-based architecture. The framework is written in the C# programming language. Moreover, the application is built over an MVC-type infrastructure that stands for Model-View-Controller. It quintessentially classifies application development into three main components, as stated before. Each component is built to handle a specific part of Development. The flow of control occurs because the user interacts with the view and requests something appropriate. It can either be to update or to fetch data from the database. The controller then takes up that request and passes it on to the Model is the database and schema structure implemented from the Entity framework. The latter provides a "edmx" file, a Microsoft Visual Studio Entity Data Model file. This file is what provides data binding between the code and the database. The ModelModel takes up the request and runs the job accordingly, either update or provide appropriate data to the view, which is then visible to the end-user. The Controllers are nothing but APIs (Application Programming Interface) that connects to the ModelModel. It carries the responsibility of the client request on its header as HTTP Status Codes.

3.2.3. Presentation Layer: The web application presents the user with an HTML, CSS, and Bootstrap-based view model. The presentation layer interacts with the controllers and the data model to provide data to the user accordingly. Bootstrap is a free, open-source front-end development framework that enables the web application's responsiveness. A responsive web application can automatically change its view configuration concerning the user device's display size.

3.3. Blockchain: There are many problems faced due to centralizing all the data being collected. Firstly, since the data is being stored in a central server, any third-party attack will completely lose data. Secondly, since the data is public, it can easily be hacked and changed. These problems can be solved when the data is decentralized, and all the sole owners of the data have a copy of the data instead of depending on one server. This leads to the basic idea behind Blockchain. It is generally defined as the decentralized ledger of data not controlled or owned by a central owner. The Blockchain participants do not share any personal data and instead use a hashed id introduced this technology for a public transaction ledger for the crypt currency Bitcoin has solved double spending without a trusted authority or a centralized server. It is also used in supply chains, intellectual property, cloud storage, and many more applications are being discovered.

3.3.1. Ethereum versus Hyperledger
3.3.1.1. Ledger Network: The first difference is that Ethereum Blockchain is a public Blockchain network and hence used by developers for learning the technology. The Ethereum network provides an option of using it as a private network as well. Whereas Hyperledger is a private Blockchain network mainly used by organizations for business purposes.

3.3.1.2. Consensus Mechanism: The next main difference is the consensus mechanism used by the network. Ethereum uses a Proof-of-work algorithm so that all the nodes reach an agreement. However, Hyperledger gives the user an option to choose between No-op (no consensus) or an agreement protocol called Practical Byzantine Fault Tolerance (PBFT). The parties involved in the transaction can agree on a key such that both can gain the outcome. Hence, Hyperledger controls the consensus and restricts access to transactions, thereby increasing scalability and privacy.

3.3.1.3. Development: Another difference is the programming languages used in developing Blockchain networks. Ethereum uses Solidity language for writing smart contracts. On the other hand, smart contracts are known as chain code in Hyperledger networks written using Golang, a language that Google creates.

3.3.1.4. Cryptocurrency: The final difference is the Cryptocurrency used by the Blockchain networks. Ethereum uses ethers for transactions, but in the case of Hyperledger, there isn't a need for cryptocurrencies for the transaction, and therefore, no mining activity is required.

3.3.2. Working of Ethereum: Ethereum is generally defined as an open-source, decentralized Block chain having additional smart contract functionality. Ether is the Crypto currency used in the Block chain network to enable the transaction between two parties: the network participants. As mentioned before, smart contracts in Ethereum are coded using the Solidity programming language. Smart contracts are a code that can enable the exchange of ethers or anything of value between the participating entities. When specific conditions are met, the smart contracts are executed automatically without any third party triggering them. Unlike other Block chain networks that give only limited use cases, Ethereum gives the developer to program the smart contract as to the developers to use.

3.3.3. Transactions in Ethereum network: As the name Block chain [9] says, each transaction is added as a block when the network's participating bodies agree on the transaction. The chain's first block is usually called the genesis block, which carries an empty state and indicates no transactions have been made. When the first transaction is carried out, the chain elongates, and a new block is added to the chain along with the genesis block. Transaction refers to the world state change that happens when any of the participants trigger the smart contract.

A block contains a series of transactions. Each block also has the previous block's hash value and a hash value of its own, thereby forming the chain. If any foreign participant wants to change the data, it will also change the same block's hash value, intended to break the chain. Similarly, to validate a transaction, it must go through a validation process which in Ethereum terms is known as the Mining process. In Ethereum, the Proof-of-Work (PoW) consensus mechanism is being used for validation. Each miner in the network will solve a problem to add the block in the chain, and by doing so, they also receive ethers in return. Any mathematical problem can be given, such as a Hash function or an Integer factorization problem. Solving the PoW problem will form the hash that is required for that block. More complex problems will require more hash power, and hence the problem must be kept simple. Since the problem complexity might affect the time taken for the block generation, thereby the workflow hangs. The hash produced by the consensus is the proof that a miner node validates the transaction, and only after the validation process, the block is being added to the network. Since Ethereum is a public Blockchain, miners worldwide will try to validate
the block, but the fastest among them will receive the reward for validating the block and adding it to the chain. A new block is added to the growing chain and is being verified by the nodes that hold a miner's title. As aforementioned, only after the trusted participants of the Blockchain verify the transaction, the data is added to the chain, and hence the data is secured.

4. Implementation

Figure 3. Methodology

The sensors placed in On-field and Warehouse collect real-time data, and it is recorded in the world state. These readings are digitalized using the Arduino UNO board placed separately at each location. A LoRa module is connected to each of the Arduino boards to enable long-range wireless transmission of data. A receiver LoRa is used to collect all the transmitted data and send it to the gateway, a Raspberry Pi module. The data received is sent serially to the Raspberry Pi, and data formatting is done to differentiate between the respective sensor node's data. This data is now sent to the world-state through Wi-Fi. For this Model QL database based on the Azure server is used. Various tables are created to provide a more flexible and efficient approach to the prototype. Figure 3 shows the implementation of the system.

4.1 Experimental Set-up:
For On-field measurements, the Tomato plant was made to grown using seeds of PKM-1 variety in 6 garden bags, as shown in Figure 4 (a). Each bag was 26 centimeters in height and 21 centimeters in diameter. Out of these 6 bags, 3 bags contained 10 kg of soil mixed with 10 grams of fertilizer, and the other three bags contained 10 kg of soil with 20 grams of fertilizers. The fertilizer used had an NPK ratio of 5-10-10. After 15 days, the grown saplings were about 10 cm in height, as shown in figure 4 (b). A total of 12 tomato plants were grown with each bag containing 2 tomato plants (figure 4(c)). The plants were watered twice every day. Figure 4 (c) shows the tomato plant after 77 days from the planting day. Also, we could see the fruit in it [14].
Figure 5(a) shows the on-field hardware set-up, which contained the MQ137 ammonia sensor, soil moisture sensor, and LoRa transmitter module connected to Arduino UNO. This hardware set-up was placed in the garden bag where the plant was grown watered the plant half an hour before measuring the on-field parameters. Ammonia content is measured in parts per million (ppm) and soil moisture in percentage.

Figure 5(b) shows the warehouse set-up, where a cardboard box of 30cm x 30cm x 15cm dimensions was taken in which stored the harvested tomatoes. DHT-11 sensor for humidity and temperature measurement and CO₂ sensor and the LoRa transmitter module connected with the Arduino controller was placed inside the cardboard box.

A 10000mAh power bank powered the on-field and the Warehouse hardware set-up. The on-field and warehouse readings were taken twice a day, one at around 10 a.m. and the other at 6 p.m. for over 3 months. A LoRa receiver module received the readings from both the LoRa transmitter with Arduino UNO, which was placed at about 2 meters from both the transmitters and viewed using the serial monitor of Arduino IDE. An excel sheet was maintained with everyday readings, as shown in Table 2.
Table 2. Real-time data

| S. No | Timing | ON-Field | Warehouse |
|-------|--------|----------|-----------|
|       |        | Soil Moisture | Ammonia | Temperature | Humidity | CO2 |
| 1     | Morning | 47% | 6ppm | 28.5 | 70% | 404ppm |
|       | Evening | 100% | 0ppm | 30 | 63% | 404ppm |
| 2     | Morning | 24% | 19ppm | 30.4 | 60% | 399ppm |
|       | Evening | 21% | 0ppm | 27 | 57% | 460ppm |
| 3     | Morning | 59% | 7ppm | 29 | 60% | 399ppm |
|       | Evening | 30% | 0ppm | 28.5 | 59% | 404ppm |

Measured Parameter values were processed in Raspberry Pi that had serial communication with the receiver Arduino. The obtained parameters were compared with the optimum conditions and classified as Category A, Category B, or Category C, as shown in Table 3.

Table 3. Classification of Crop based on the measured parametric values

| Category | Ammonia | Soil Moisture | CO2 | Temperature | Humidity |
|----------|---------|---------------|-----|-------------|----------|
| A        | 3.5ppm  | 70-80%        | 3%  | 12.5-20°C   | 90-95%   |
| B        | 3ppm    | 80-90%        | 2%  | 20-25°C     | 95-100%  |
| C        | 3ppm    | 60-70%        | 1%  | 10-12°C     | 80-90%   |

These values, along with the categorization was updated in Microsoft Azure Database at regular intervals.

4.2 Hardware Deployment

Hardware deployment essentially consists of two main nodes to sense the appropriate parameters in On-field and warehouse to determine the crop's quality. The first node is placed in-situ on-field to measure the soil moisture and the ammonia content present in the soil. The second node is placed in the warehouse, where the tomatoes were kept after a successful harvest. In the warehouse, the temperature, humidity, and
CO₂ emission levels were measured. Lora transmitter is used to collect the on-field and the warehouse data. Figure 6 shows the measured parametric values of on-field and the warehouse.

These data are digitalized using the Arduino UNO board placed separately at each location. The receiver Arduino is connected to Raspberry Pi through serial communication will act as a gateway node. The collected data were sent to the Microsoft Azure Database. Figure 7 shows the database containing the measured data.

![Microsoft Azure Database](image)

**Figure 7.** Microsoft Azure Database with the measured data’s

Data are stored in the databases using the library "product," which helps make SQL queries. FreeTDS is a set of libraries for UNIX and Linux that allows your programs to talk to Microsoft SQL Server. These libraries are configured to establish a generic connection to our server by providing the server's server location, username, and password. Once the configurations are made, a connection test can be made using SQL, and if configurations are done connection aptly will be established.

### 4.3 Software Deployment

To publish the web application on the World Wide Web for the end-users to access it over the internet, the web application must be deployed over a specialized web server. Azure app service is a part of Azure's PaaS-based cloud services is used. The ASP.NET code is deployed to the Azure app service through a web deployment set-up that uses File Transfer Protocol (FTP). The system that deploys the code onto the Azure app service needs to get configuration checks such as IIS and required libraries on the web platform installer. Once the code is successfully deployed, the web application will be given a Fully Qualified Domain Name (FQDN), with the application’s name and the azure’s domain name. Figure 8 shows the end-user interface.
4.4 Blockchain Deployment

Ganache is used for setting up a personal Ethereum Blockchain to deploy contracts, develop applications, and run tests [9]-[12]. Figure 9 shows the personalized Ethereum Blockchain created for the proposed work.

Ganache runs local instances of Ethereum. Metamask is a browser-based Ethereum wallet that allows web applications with the Ethereum Blockchain without running a full Ethereum node. The Ganache Ethereum network and Metamask wallet are connected using the RPC HTTP URL provided in Ganache. The accounts can be imported to the wallet by providing the account's private key, available in Ganache. Each account is assigned with 100 ethers each, and transactions are performed using the SEND and DEPOSIT buttons, as shown in Figure 10.

Any transactions you make with this account in Metamask will be reflected in Ganache and vice-versa.
5. Conclusion
Agriculture is the backbone of the Indian economy. Nearly two-thirds of the Indian population is dependent on agriculture for a living. The average Indian farmer does not get the right price for his produce due to intermediaries' presence. The proposed IoT-based Crop Quality Measurement and Monitoring system use Blockchain technology to directly connect the farmers with the distributor to be rightly rewarded. This system can be developed as a mobile app so to obtain more benefits for the farmers. This work's future scope can be an autonomous web server on the cloud accessible for all the customers, with low latency, higher throughput, increased security, and built-in disaster recovery.

Acknowledgement
We express our gratitude to SRM the Innovation and Incubation Center (SIIC) of SRM Institute of Science and Technology, to fund this work. We would like to thank Dr. E. Madhu Sudhanan, Dept. of Agricultural Entomology, Tamilnadu Agricultural University, Coimbatore, for his guidance regarding crop choice for pursuing this work for providing pertinent information about the chosen plant.

References
[1] KA Isaac, A Harrison, KK Ernest, and O Hayford. Preharvest and postharvest factors affecting the quality and shelf life of harvested tomatoes. International Journal of Agronomy, 20(15-16), 2015.
[2] Miguel Pincheira Caro, Muhammad Salek Ali, Massimo Vecchio, and Raffaele Giaffreda. Blockchain-based traceability in agri-food supply chain management: A practical implementation. In 2018 IoT Vertical and Topical Summit on Agriculture- Tuscany (IOT Tuscany), pages 1–4. IEEE, 2018.
[3] Seyoung Huh, Sangrae Cho, and Soohyung Kim. Managing IoT devices using a blockchain platform. In 2017 19th international conference on advanced communication technology (ICACT), pages 464–467. IEEE, 2017.
[4] Mayra Samaniego and Ralph Deters. Blockchain as a service for IoT. In 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications
(GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCCom) and IEEE Smart Data (SmartData), pages 433–436. IEEE, 2016.

[5] Guy Zyskind, Oz Nathan, et al. Decentralizing Privacy: Using Blockchain to protect personal data. In 2015 IEEE Security and Privacy Workshops pages 180–184. IEEE, 2015.

[6] Runchao Han, Vincent Gramoli, and Xiwei Xu. Evaluating blockchains for IoT. In 2018 9Th IFIP international conference on new technologies, mobility and security (NTMS), pages 1–5. IEEE, 2018.

[7] Pradeep Kannadiga and Mohammad Zulkernine. Didma: A distributed intrusion detection system using mobile agents. In Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network, pages 238–245. IEEE, 2005.

[8] N Kshetri. Can Blockchain strengthen the internet of things? It prof. 19 (4), 68–72 (2017), 2017.

[9] Oscar Bermeo-Almeida, Mario Cardenas-Rodriguez, Teresa Samaniego-Cobo, Enrique Ferruzola-Gómez, Roberto Cabezas-Cabezas, and William Bazán-Vera. Blockchain in agriculture: A systematic literature review. In International Conference on Technologies and Innovation, pages 44–56. Springer, 2018.

[10] Shilpa Devalal and A Karthikeyan. Lora technology-an overview. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pages 284–290. IEEE, 2018.

[11] Alireza Zourmand, Andrew Lai Kun Hing, Chan Wai Hung, and Mohammad Abdul Rehman. Internet of things (IoT) using Lora technology. In 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), pages 324–330. IEEE, 2019.

[12] Borja Bordel, Diego Martin, Ramón Alcarria, and Tomás Robles. A blockchain-based water control system for the automatic management of irrigation communities. In 2019 IEEE International Conference on Consumer Electronics (ICCE), pages 1–2. IEEE, 2019.

[13] Li Yan, Zhidan Zhang, Yuan Chen, Qiang Gao, Wenxi Lu, and Ahmed Mohamed Abdelrahman. Effect of water and temperature on ammonia volatilization of maize straw returning. Toxicological & Environmental Chemistry, 98(5-6):638–647, 2016.

[14] MD Cahn, EV Herrero, RL Snyder, and BR Hanson. Water management strategies for improving fruit quality of drip-irrigated processing tomatoes. Acta horticulture, 2001.

[15] Biswanath Dari, Christopher W Rogers, and Olga S Walsh. Understanding factors controlling ammonia volatilization from fertilizer nitrogen applications.

[16] Li Yan, Zhidan Zhang, Yuan Chen, Qiang Gao, Wenxi Lu, and Ahmed Mohamed Abdelrahman. Effect of water and temperature on ammonia volatilization of maize straw returning. Toxicological & Environmental Chemistry, 98(5-6):638–647, 2016.

[17] Baobao Pan, Shu Kee Lam, Arvin Mosier, Yiqi Luo, and Deli Chen. Ammonia volatilization from synthetic fertilizers and its mitigation strategies: a global synthesis. Agriculture, Ecosystems & Environment, 232:283–289, 2016.