DYNAMIC BLOCKCHAIN ADOPTION FOR GLOBAL AGRICULTURAL PRODUCTS USING CLOUD SERVICES.

PROJECT REPORT

Submitted by

ASHRITHA P	**[REGISTER NO: 211419104025]**
DESAPPRIYA G D	**[REGISTER NO: 211419104053]**
DHIVYA P	**[REGISTER NO: 211419104065]**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“Dynamic blockchain adoption for global agricultural Products using cloud services”** is the bonafide work of “**Ashritha. P (211419104025), Desappriya.G.D(211419104053), Dhivya.P(211419104065)”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. L. JABASHEELA, M.E., Ph.D., **HEAD OF THE DEPARTMENT**	**Dr.KAVITHA SUBRAMANI M.E, Ph.D.,** **SUPERVISOR** **PROFESSOR**
DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.	DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.

Certified that the above candidate(s) was examined in the End Semester Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We…**Ashritha.P(211419104025),Desappriya.G.D(211419104053), Dhivya.P(211419104065)**…. hereby declare that this project report titled "**Dynamic blockchain adoption for global agricultural Products using cloud services**", under the guidance of **Dr. Kavitha Subramani M.E,Ph.D.,** is the orginial work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ASHRITHA P

DESAPPRIYA G D

DHIVYA P

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L. JABASHEELA, M.E., Ph.D..,** for the support extended throughout the project.

We would like to thank our guide **Dr. KAVITHA SUBRAMANI M.E, Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

ASHRITHA P

DESAPPRIYA G D

DHIVYA P

ABSTRACT

In the supply chains of emerging economies, the livelihoods of smallholder farmers are subpar because of financial exclusion, fraud, exploitation, corruption, deception, child labour, and other acts typically committed by powerful actors. A problem with social sustainability arises from this situation, and immediate action is required. In our viewpoint article, we adopt a customary strategy to flash academic interest in examining and creating research needs on the most proficient method to utilize innovation to address this current and basic maintainability and production network concern. Blockchain can be used to address complexities, inefficiencies, and other societal issues. The hash technique, which is crucial for online payments and transactions, is used to execute the blockchain concept, The hashing algorithm used to secure the transaction in our perspective article is Secured hash algorithm which is simply represented as SHA-256. Regardless of the size of the input transaction, this hashing algorithm always produces an output of 256 bits or 32 bytes. That is, if we use SHA-256 to hash two different inputs, say a 1 gigabyte movie and a 5-kilobyte image, The generated hash will be 256 bits long in dual instance. It is evident that e-commerce has a significant impact on agriculture. A major cause for concern is how agricultural products are purchased. To obtain agricultural products, customers frequently must travel considerable distances, and the quality is not guaranteed. By employing a computerized strategy that will enable farmers to gain access to a variety of products that are sold as well as the total sales of each product, our concept aims to assist buyers and farmers alike in purchasing and selling agricultural goods across the nation. Through end-user communication, it creates a platform for farmers to increase profitability and serves as a unique and secure method for agro marketing. With simply a simple comprehension of site route, ranchers might sell their merchandise the country over by utilizing e-cultivating. Additionally, it enables users to instantly purchase the desired products by making an online payment, and Amazon Web Service is used to permanently store information about product availability, price, and payment history in the cloud. Users can see a wide range of products and immediately purchase the ones they want.

LIST OF FIGURES

FIGURE NO.	NAME OF THE FIGURE	PAGE NO.
4.1.1	Entity Relationship Diagram	17
4.2.1	Data flow diagram level 0	19
4.2.2	Data flow diagram level 1	19
4.3.1	Use case diagram.	21
4.3.2	Sequence diagram	22
4.3.3	Activity diagram	23
4.3.4	Class diagram	24
4.4	System architecture	25
5.1.1.1	Workflow of customer	29
5.1.1.2	Workflow of farmer	29
5.1.1.3	Workflow of administrator	30
A.2.1	Screenshot of SHA code generated for various transactions.	61
A.2.2	Screenshot of AWS console for cloud storage	61
A.2.3	Screenshot of various slides in application	62
A.2.3.1	Home page of the application	62
A.2.3.2	Products added by farmers.	62
A.2.3.3	Verification by admin	63
A.2.4	Screenshot For Testing the Working of Secured Hash Algorithm.	63
A.2.4.1	An unknown user trying to access the information of another user	63
A.2.4.2	A message denying the access from unknown user	64

LIST OF ABBREVATION

S.NO	ABBREVATION	EXPANSION
1.	DB	Data Base
2.	SMC	Secure Multiparty Computation
3.	AWS	Amazon Web Service
4.	DBC	Data Base Confidentiality

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
2.2	Comparison table based on literature survey	07
6.3.1	Test case for seller registration	34
6.3.2	Test case for customer registration	35
6.3.3	Test case for Administrator login	35
6.3.4	Test case for database connectivity to cloud	36
6.3.5	Test case for generation of hash code	36
6.3.6	Test case for the verification of blockchain concept	36

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	vi
	LIST OF ABBREVIATIONS	vii
	LIST OF TABLES	vii
1.	**INTRODUCTION**	4
	1.1 Problem Definition	4
2.	**LITERATURE SURVEY**	6
	2.1 Literature Survey	6
	2.2 Comparison table based on literature survey	7
3.	**SYSTEM ANALYSIS**	12
	3.1 Existing System	12
	3.2 Proposed System	12
	3.3 Feasibility Study	13
	3.4 Hardware Environment	14
	3.5 Software Environment	14
	3.5.1 Features of Java	14
	3.5.1.1 The Java Framework	14
	3.5.1.2 Java Servlets	15
	3.5.1.3 Java Server Pages	15

4.	**SYSTEM DESIGN**	17
	4.1 Entity Relationship Diagram	17
	4.2 Data Flow Diagram	19
	4.3 Uml Diagram	20
	4.3.1 Use Case Diagram	21
	4.3.2 Sequence Diagram	22
	4.3.3 Activity Diagram	23
	4.3.4 Class Diagram	24
	4.4 System Architecture	25
	4.4.1 Architecture Overview	25
	SYSTEM IMPLEMENTATION	28
5.	5.1 Module Description	28
	5.1.1 Module Diagram	29
	5.2 Secured hash algorithm	30
6.	**SYSTEM TESTING**	33
	6.1 Unit Testing	33
	6.2 Integration Testing	33
	6.3 Test Cases and Report	34
7.	**CONCLUSION**	38
	7.1 Results and Discussions	38
	7.2 Conclusion and Future Enhancement	38
	APPENDICES	39
	A.1 Sample Code	39
	A.1.1 Client-Side Coding	39
	A.1.2 Sever-Side Coding	55
	A.2 Sample Screens	62
	REFERENCES	65

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The supply chain and production processes have changed because of the recent spectacular growth of the sharing economy. Many platforms, like Airbnb, Uber, and eBay, have been established because of technological advancements and the rising use of mobile devices and apps. These platforms provide a range of free or paid sharing economy services, as well as bartering and exchanges of products and services. The sharing of homes, vehicles, clothes, books, toys, and digital goods are just a few examples of the various facets of our everyday lives that have been impacted by the sharing economy. In the US, roughly 44.8 million adults used to share economy services in 2016, and statista.com predicts that number would rise to 86.5 million by 2021. 1 C2C-PT, or consumer-to-consumer product trade, is a sharing model. So We can implements this technique to buy fresh agricultural products from farmers and they can fix the prize of the product and customers can get fresh products which is cost efficient without/with the involvement of the 3rd member and the sales and history of orders can also be tracked by them as well as the transaction is also more secure with the help of blockchain and AWS cloud

CHAPTER 2
LITERATURE SURVEY

\

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE SURVEY

Using a direct-to-consumer marketing approach, [1] described an E-store for Farmers Purchasing Seeds, a fully-fledged website for electronic commerce aimed at farmers. It does away with the necessity for merchants and allows for direct client contact without the tainting of services by middlemen. the quality of the crop/seed is also trusted to minimise the time and effort required. Considering security issues and utilising the cloud to permanently store data can be done from this future perspective. Using the Fruity Healthy mobile application, [2] discusses the Fruit Ordering System. Creating a fruit ordering application that enables clients to buy fruits is the project's goal to assist farmers in expanding their market. Android, iOS, Windows, and Harmony OS mobile devices may all support this app.

An intelligent system for Agricultural Bidding, Improvement of Farmer's Lifestyle by [3] mentioned that the goal of this System for online bidding is to give users more options when making investments. Users can invest in the businesses of their choice using the system, which also saves time. By removing the middleman and bringing together farmers and consumers, the application enables people to bid for farm products. The goal of this project is to give farmers a software environment that will allow them to make the most money possible. It can be improved by including a secure socket layer that would shield data from outside parties. For user attraction, more graphics can be included. [4] describes on Fresh Agricultural Product Traceability System. It involves the development of edge computing and block chain technology in the agricultural traceability system for fresh agricultural products. The benefits that can be gained by Making use of advance technology to determine the quality and freshness of the products.

[5] discusses about how blockchain concept can be implemented and to maintain freshness of the product being harvested and sold, harvested agricultural goods briefs about agricultural products on E- commerce made it more challenging of delivering green and fresh. The traditional is successful at keeping things fresh yet failed to track record. So, we tend to deliver it fresh and by using dynamic optimization, advertising effort and by using block chain adoption freshness is maintained. So that freshness of the fruits is maintained even at the time of delivery. [6] has attempted to use blockchain technology to track the freshness of products in real time, so to make it more prominent, [6] has

comprehended the information provided by stakeholders, and this study includes all parties involved in the livestock-based product supply chain. Also aids in the tracking of product information, such as who the stakeholders are and what they are marketing.

[7] has come up with an expert consultation for agriculture Secondary Markets which details about the Growing interest in dynamic access due to the relevance of wireless communications agro products. This calls for effective management practises that permit sharing between authorised main users (PU) and unauthorised secondary users (SU). In this situation, PUs must protect their right to use. product to customer. This let us know whether the farmers are trusted their identity is verified but this does not have any admin to track the product or details regarding transactions. [8] explored on an agricultural monitoring system's design and execution for smart farming this relates about the recent Novel applications that are made possible by the burgeoning Web of Things (IoT) and its affordable sensors and actors. But this system is bit tedious, and it requires lot of knowledge about the sensors how it works and very it is easy to handle.

[9] has tried Studying a Blockchain-Based Decision Support System for Agricultural Goods in Online Commerce which relates that there are certain difficult that is present in data on agricultural items sold on the internet. To overcome this blockchain technology with additional platform is constructed.so that decision can be done better, and farmer can earn good income. The benefits can be gained by using blockchain good decisions can be provided, sales of agricultural products can be promoted, and farmers can gain good income. [10] has proposed a paper on choosing a fresh agriculture e-commerce logistics model products based on analytical hierarchy approach and fuzzy thorough evaluation methods which tells of about People usually thinks of purchasing fresh agricultural products from market. Therefore, by providing some logistic selection process people can easily purchase fresh agricultural products in E-commerce using some best selection methods. By providing logistic mode selection process, fresh agricultural products can be sold.

2.2 COMPARISON TABLE BASED ON LITERATURE SURVEY

YEAR	AUTHOR	TITLE	METHODOLOGY	MERITS AND DEMERITS	FUTURE SCOPE
2022	Angel Infanta Ramesh, Ayush Raghuwanshi, Eshan Goel, Dakshayani G	An E-store for Farmers Buying Seeds	A fully-fledged Electronic Commerce website for the farmers based on a Direct-to-Consumer marketing strategy. It eliminates the	Merits : Reduces the time and efforts spent but also entrusts the	Future work can be done by taking security concerns into account and leveraging the

			need for retailers and have direct interaction with the customers without services being warped by the intermediaries.	crop/seed quality. Demerits : Insufficient security and privacy for customers	cloud to permanently store data.
2021	Suraya Abu Bakar, Liew Pei Ling	Fruit Ordering System through Fruity Healthy Mobile Application	The objective of this project is to help farmers increase their sales market by developing a fruit ordering application that allows customers to purchase fruits.	Merits : It is compatible with Android, iOS, Harmony OS, and Windows phones	Customers' database can be recorded so that sellers can choose who to target and how to enhance sales in the future.
2019	Nalinipriya G, Sangeetha R, Saniya K, Sri Dhanusiya Navarath S.	Agro Bidding - A Smart Dynamic System for Enhancement of Farmer's Lifestyle	The objective of the online auction system is that the user can have better choice for their investment. Also it is time saving and through this system user can invest in their own selected firm. The application allows consumers to bid for the farm produce, thus eradicating middleman and benefiting both farmers and consumers.	Merit : It is a popular method for buying and selling products, It aims to provide a software environment for farmers to gain maximum profit.	Adding an secure socket layer would prevent data's from third parties. More graphics can be added to attract users.
2021	Shuhui Pan, Min Zuo, Wenjing Yan, Qingchuan Zhang, Wei Wei	Agricultural super docking Traceability System of Fresh Agricultural Products	This paper introduces edge computing and block chain technology to design the agricultural super docking traceability system for fresh agricultural products	Merits: Make use of advance technology to determine the quality and freshness of the products.	Blockchain technology can be used to implement the safe storage tamper-resistant of data.
2022	Yuting Li, Chunqiao Tan	Dynamic block chain adoption for freshness keeping in the fresh agricultural	agricultural products on E- commerce increased the difficulty of maintaining the greenness and freshness in delivery. The traditional is effective in keeping freshness but	Merits: freshness of the fruits is maintained even at the time of delivery. But the freshers is	To maintain the freshness new block technology is Used so that freshness is maintained.

		product supply chain.	failed to track record. So we tend to deliver it fresh and by using dynamic optimization, advertising effort and by using block chain adoption freshness is maintained	not guaranteed by consumer	
2022	Anil kumar Anal, Tauji w.Tsusaka , Khwanchol Kamapan	Adoption of block chain Technology for Enhanced traceability of livestock-based products.	This paper reviews studies on blockchain technology applications to the agri-food supply chain system and food industry and discusses potential adaptation of blockchain technology for livestock-based products. this paper encompasses stakeholders along the supply chain of livestock-based products to understand the stakeholder’s information.	This helps in tracking of the product and its information regarding who the stakeholders and what they are selling.	Along with tracking it can also enable the receiver to know it's supplier.
2019	Juan Vanerio, Federico Larroca	Online Expert-Based Prediction for agriculture secondary markets.	The growing importance of wireless communications drives an increasing interest in dynamic access to agro productes. This requires efficient management policies that allow sharing between licensed primary users (PU) and unlicensed secondary users (SU). On such scenario, PUs shall preserve their usage right product to customer.	This let us know whether the farmers are trusted.their identity is verified but this doesn't have any admin to track the product or details regarding transactions	Tracking and security in transaction can be implemented
2020	Jan bauer Nils Aschenbruck	Design and implementati on of an agricultural monitoring system for	Nowadays, the emerging Internet of Things (IoT) along with their low-cost sensors and actors enable novel applications and new opportunities for a more precise, site-	This system is bit tedious and it requires lot of knowledge about the sensors how it	Implementing an web application to detect the crops and in selling these to the customer

		smart farming	specific, and sustainable agriculture in the context of Smart Farming, we present a holistic agricultural monitoring system, its design,The system primarily focuses on the leaf area real-world challenges and experiences gained many development	works. it's easy to handle	directly from farm.
2020	Chao Xie Xiaoyong Xiao	Research on Decision Support System of E-commerce Agricultural Products Based on Blockchain	There are certain difficult that is present in information of E-commerce agricultural products. To overcome this blockchain technology with additional platform is constructed.so that decision can be done better and farmer can earn good income.	By using blockchain good decisions can be provided, sales of agricultural products can be promoted, and farmers can gain good income.	Complexity and time lag can be prevented.

CHAPTER 3
SYSTEM ANALYSIS

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The evolution of technology has given a major impact in E-retailing of agricultural products through E-commerce sites, There are wide range of applications which deals with the buying and selling of agricultural products online, but in all these applications privacy and security of the customer has not been the major concern, the blooming technology such as blockchain and cloud computing was not properly implemented for transactions and data storage**,** so it may cause some serious issues and it may lead to collapse the trading service or to leakage users data.

Technique:

Market Cleaning Mechanism.

Disadvantage:

It is a strategy to analyze the aggregated data.

3.2 PROPOSED SYSTEM

The smallholder farmer is one of the supply chain's forgotten actors, especially when it comes to business, technological, and economic solutions. Sustainable supply chains should not ignore these members, particularly those in developing countries. These unseen actors stand to gain the most from increased accessibility, visibility, empowerment, and sustainability, all of which can be achieved through thoughtful technology and research. To achieve sustainability, blockchain and other technologies can address inefficiencies, complexities, and conditions in the supply chain. Two practical examples provide some preliminary insights into the use of technology, particularly blockchain, to address the sustainable supply chain that demonstrate the complexities, benefits, opportunities, and risks. Apart from these technologies amazon web service provide a vital cloud services for information storage.

Technique:

Hash Function

Advantage:

It synchronizes the data or transaction when generate.

3.3 FEASIBILITY STUDY

Feasibility studies aim to uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented objectively and rationally by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, feasibility studies precede technical development and project implementation.

They are 3 types of Feasibility.

- Economical feasibility
- Technical feasibility
- Operational feasibility

ECONOMICAL FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. to estimate whether the new system will perform adequately or not.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

OPERATIONAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

3.4 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

PROCESSOR	:	DUAL CORE 2 DUOS
• RAM	:	2 GB DD RAM
• HARD DISK	:	250 GB

3.5 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

FRONT END	:	J2EE (JSP)
BACK END	:	JAVA (SERVLET), MYSQL 5.5, AWS
OPERATING SYSTEM	:	WINDOWS 7
IDE	:	ECLIPSE

3.5.1 FEATURES OF JAVA

3.5.1.1 THE JAVA FRAMEWORK

Java is a programming language originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century and is widely used from application software to web applications The java framework is a new platform independent that simplifies application development internet. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

3.5.1.2 JAVA SERVLETS

Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server. Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server so these are safe and portable. Unlike applets they do not require support for java in the web browser. Unlike CGI, servlets don't use multiple processes to handle separate request. Servets can be handled by separate threads within the same process. Servlets are also portable and platform independent. the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using File Transfer Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

3.5.1.3 JAVA SERVER PAGES

Java Server Pages or JSP for short is Sun's solution for developing dynamic web sites. JSP provide excellent server side scripting support for creating database driven web applications. JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple and its maintenance also becomes very easy. In today's environment most web sites servers dynamic pages based on user request. Database is very convenient way to store the data of users and other things. JDBC provide excellent database connectivity in heterogeneous database environment. Using JSP and JDBC its very cc easy to develop database driven web application.

Java Server Pages (JSP) technology is the Java platform technology for delivering dynamic content to web clients in a portable, secure and well-defined way. The Java Server Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML, and XML templates, and Java code, which is secure, fast, and independent of server platforms.

CHAPTER 4
SYSTEM DESIGN

CHAPTER 4
SYSTEM DESIGN

4.1ENTITY-RELATIONSHIP DIAGRAM

An architecture diagram is a graphical representation of a set of concepts that are part of an architecture, including their principles, elements, and components. It is also defined asa visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.

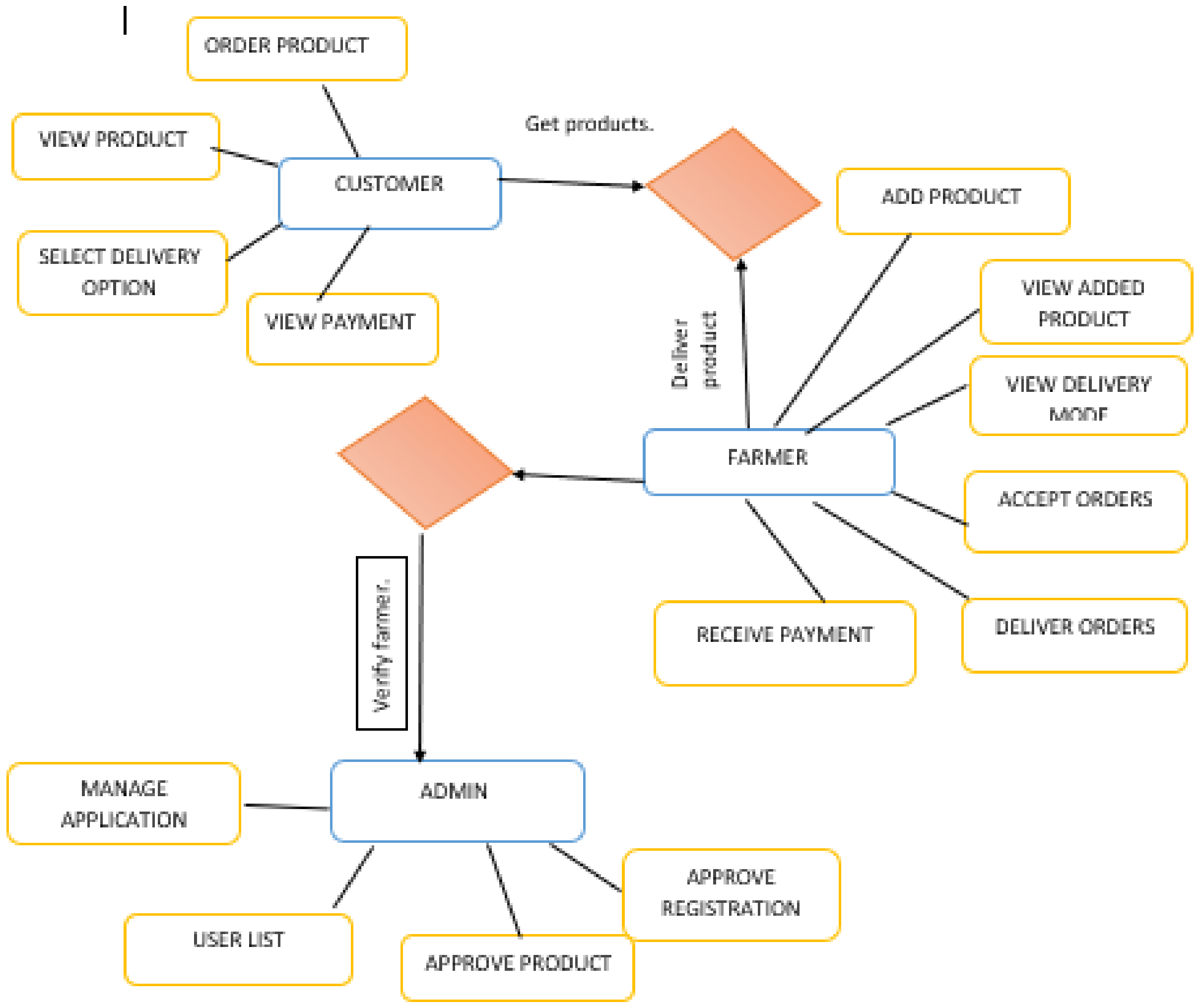


Fig. 4.1.1 Entity Relationship Diagram.

In customer entity User can view all the crops after farmer add their products into the application and if user need the crop’s he/she add to his cart and make the payment.

In farmer entity, Farmers can add the project and check the product details is ok or not. The farmer fixes the delivery method is deliverable or not deliverable. Every payment will be view by the farmer like payment history and complete account details.

Admin entity is used to activate every user registration. Admins have every access if user register their details, it will be passed to the admin and admin accept the registration. Admins have the access to approve the farmer added product. Maintain all the user list product list and all the details about the user.

4.2 DATAFLOW DIAGRAM

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM). Superficially, DFDs can resemble flow charts or Unified Modeling Language (UML), but they are not meant to represent details of software logic. DFDs make it easy to depict the business requirements of applications by representing the sequence of process steps and flow of information using a graphical representation or visual representation rather than a textual description. When used through an entire development process, they first document the results of business analysis. Then, they refine the representation to show how information moves through, and is changed by, application flows. Both automated and manual processes are represented.

LEVEL 0:

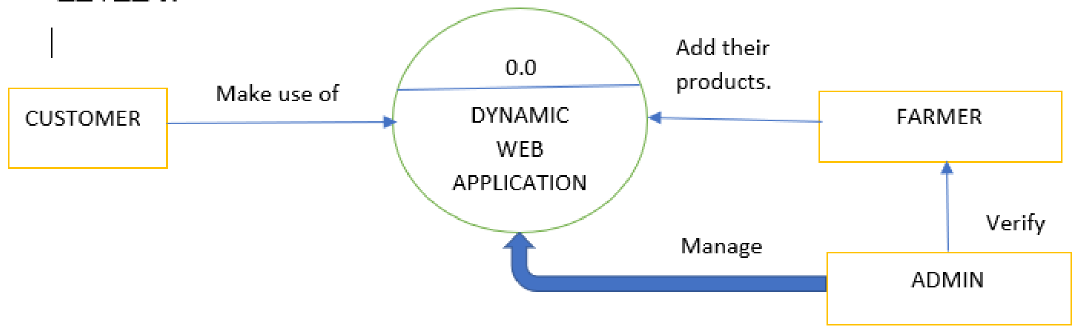


Fig. 4.2.1 Data flow diagram level 0.

LEVEL 1:

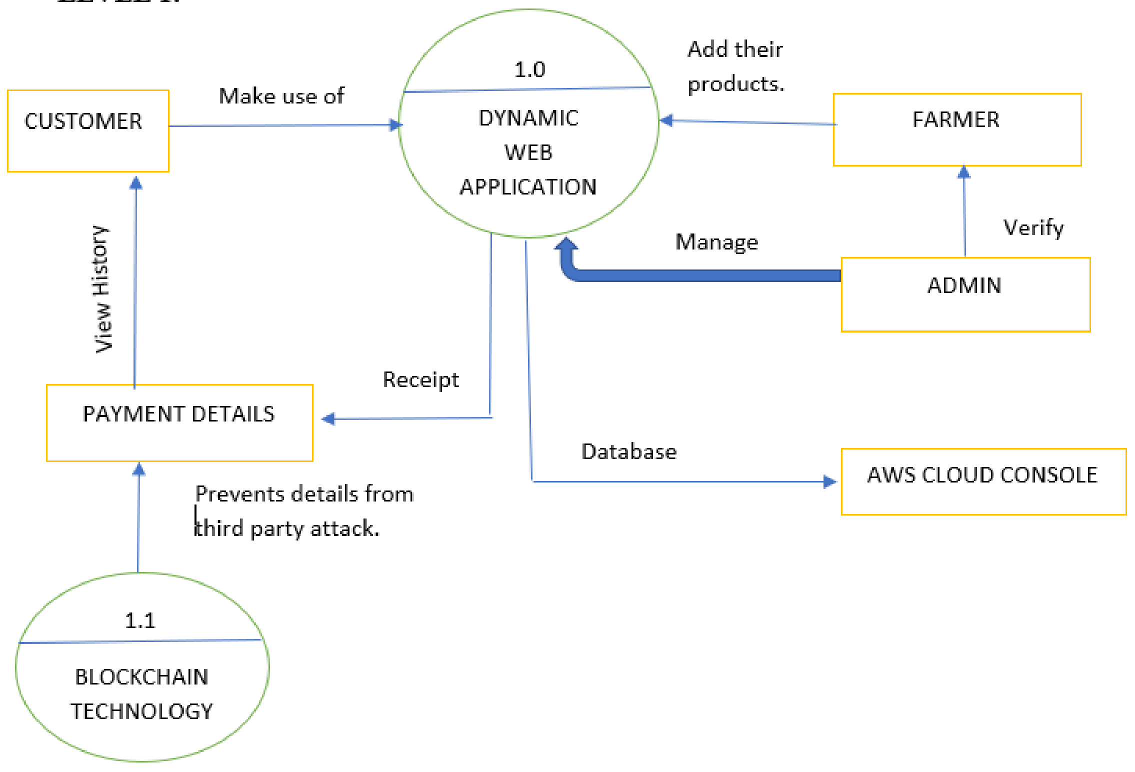


Fig. 4.2.2 Data flow diagram level 1

4.3 UML DIAGRAMS

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.OMG is continuously making efforts to create a truly industry standard stands for Unified Modeling Language is different from the other common programming languages such asC++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints' can be described as a general-purpose visual modeling language to visualize, specify, construct,and document software system. Although UML is generally used to model software systems, itis not limited within this boundary. It is also used to model non-software systems as well. Forexample, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has adirect relation with object-oriented analysis and design. After some standardization, UML hasbecome an OMG standard.

4.3.1 USE CASE DIAGRAM

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.

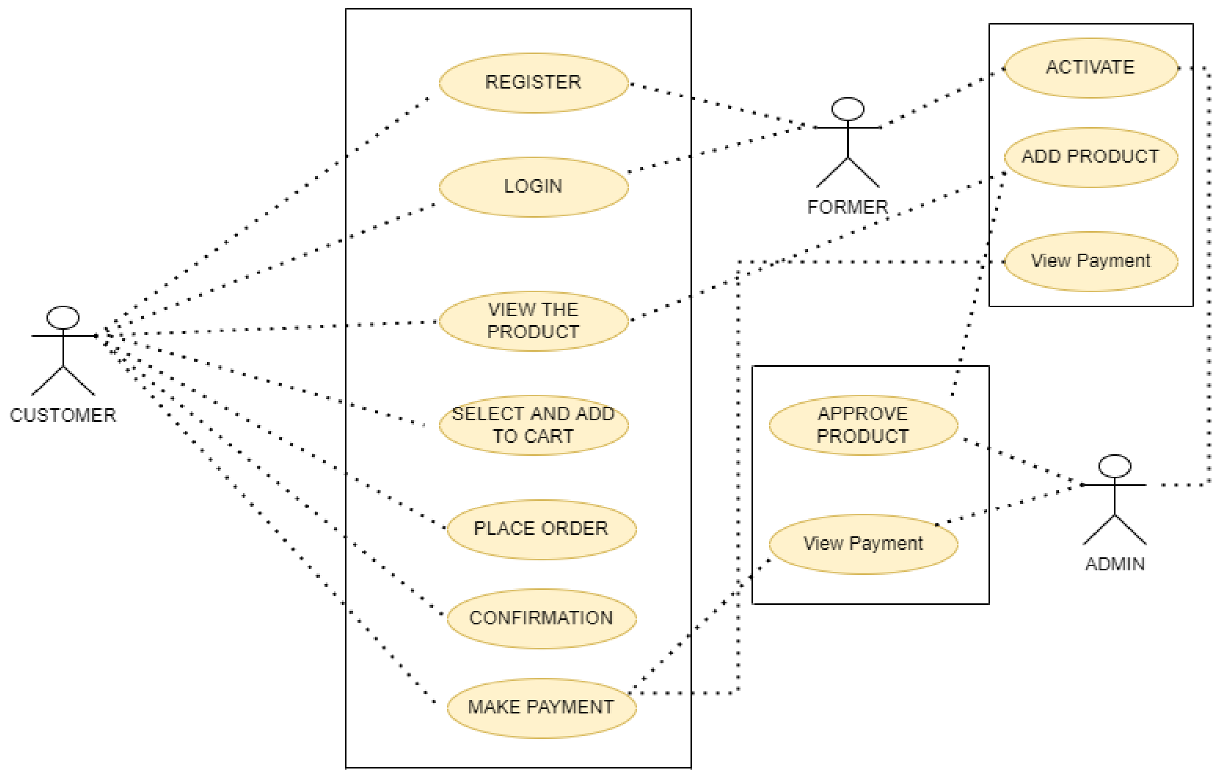


Fig 4.3.1 Use Case Diagram

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

4.3.2 SEQUENCE DIAGRAM

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

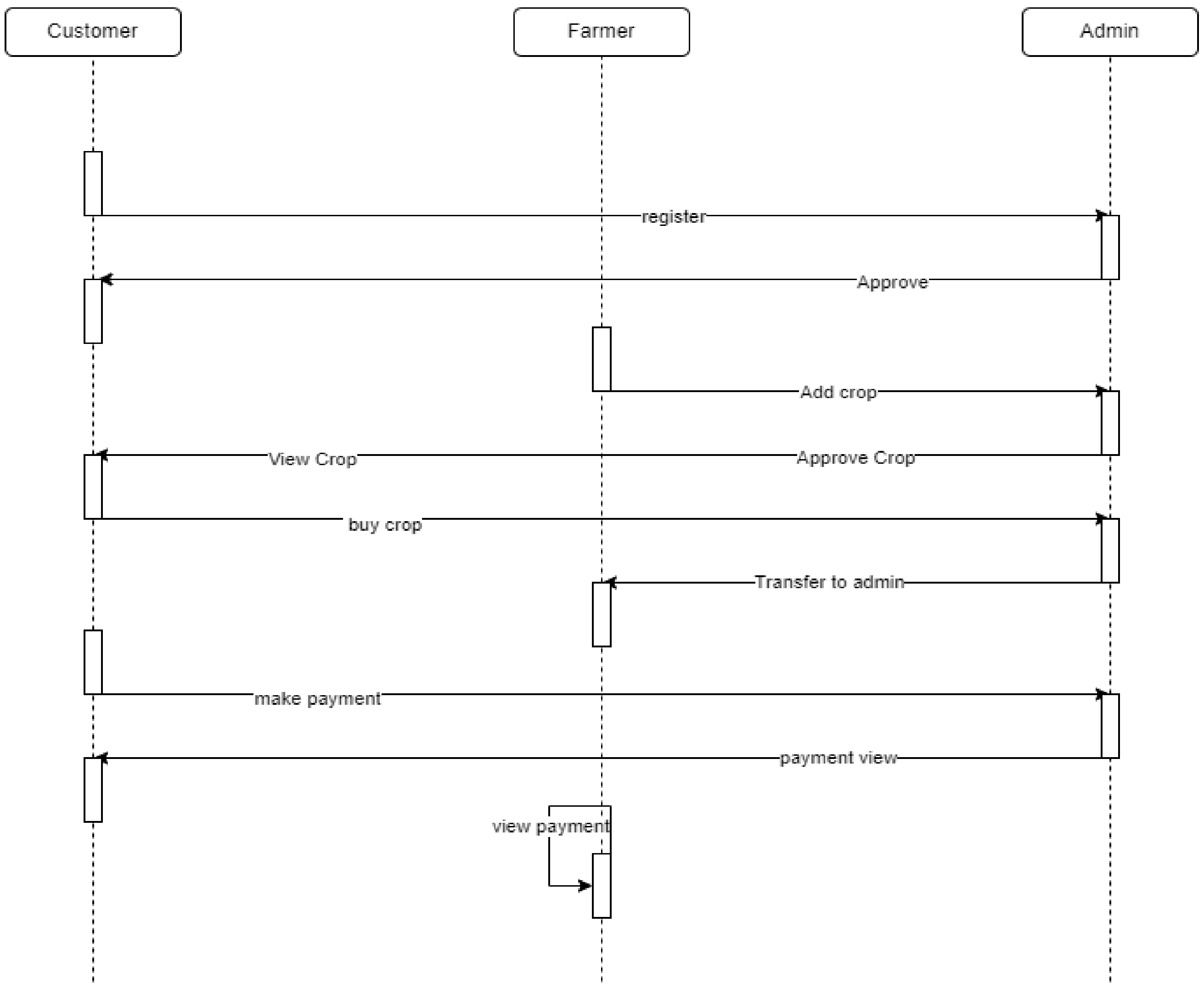


Fig 4.3.2 Sequence Diagram

4.3.3 ACTIVITY DIAGRAM

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

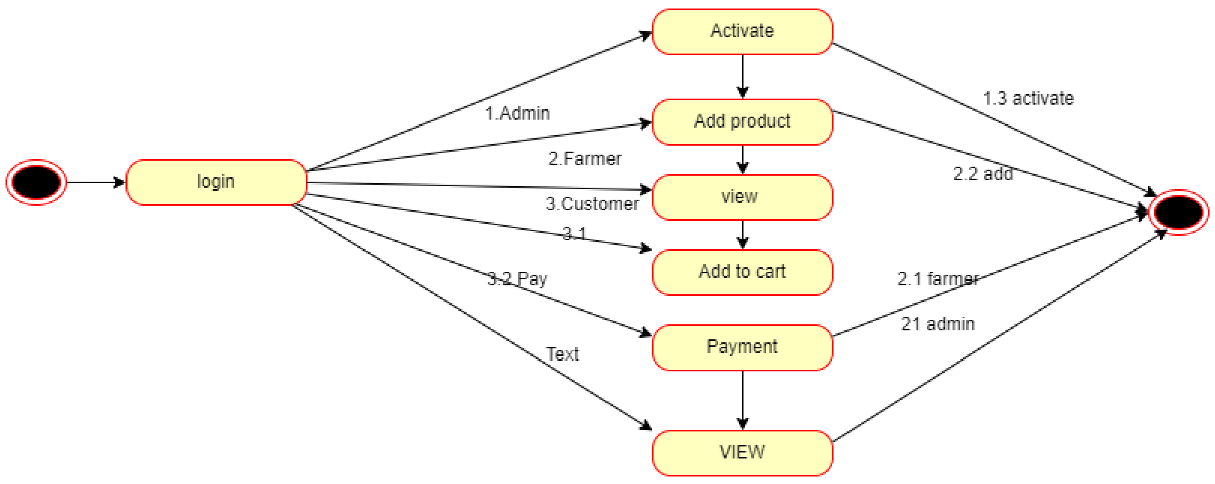


Fig 4.3.3 Activity Diagram

4.3.4 CLASS DIAGRAM

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and it may inherit from other classes. A class diagram is used to visualize, describe, document various aspects of the system, and also construct executable software code.

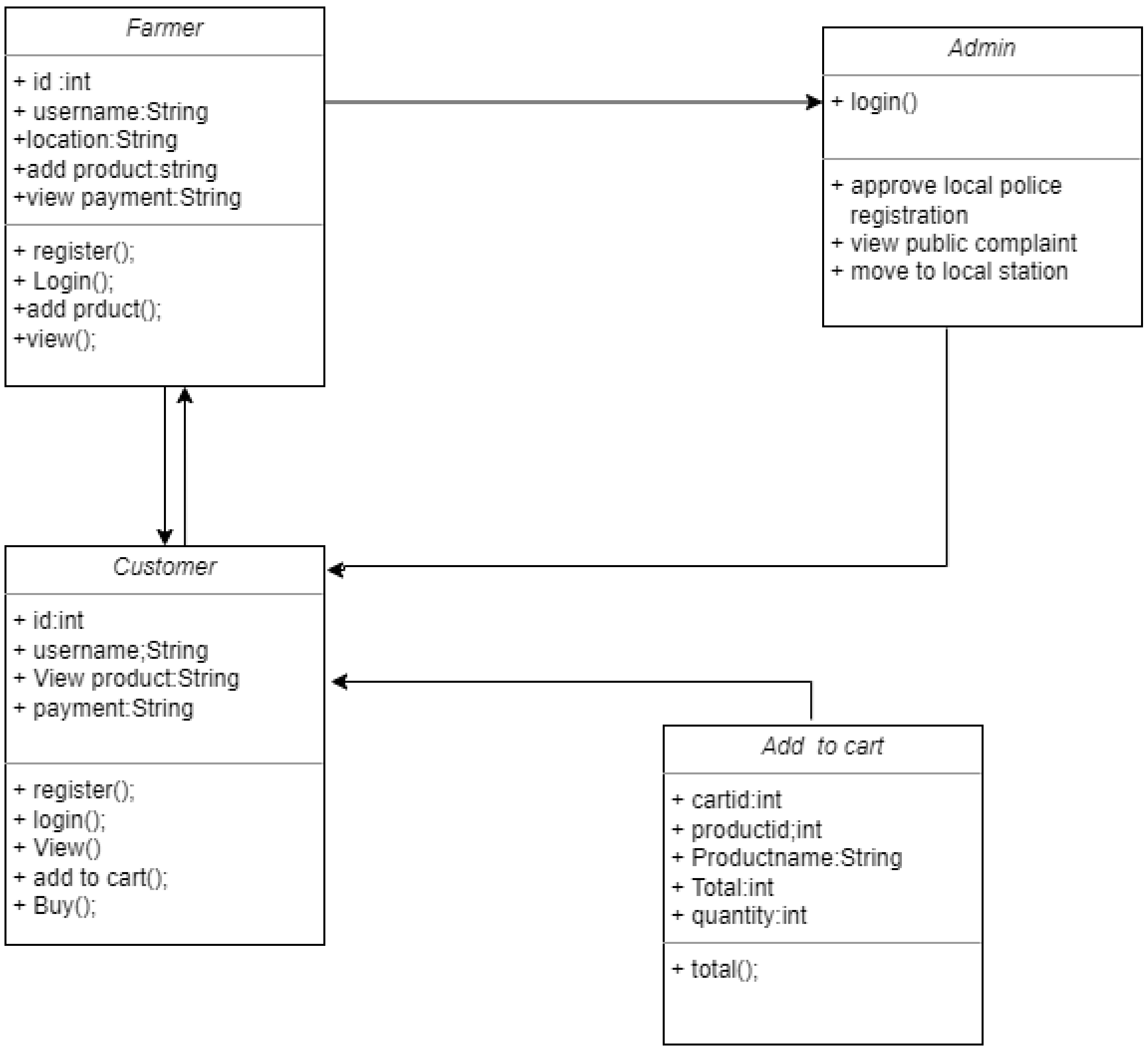


Fig 4.3.4 Class Diagram

4.4 SYSTEM ARCHITECTURE

4.4.1 ARCHITECTURE OVERVIEW

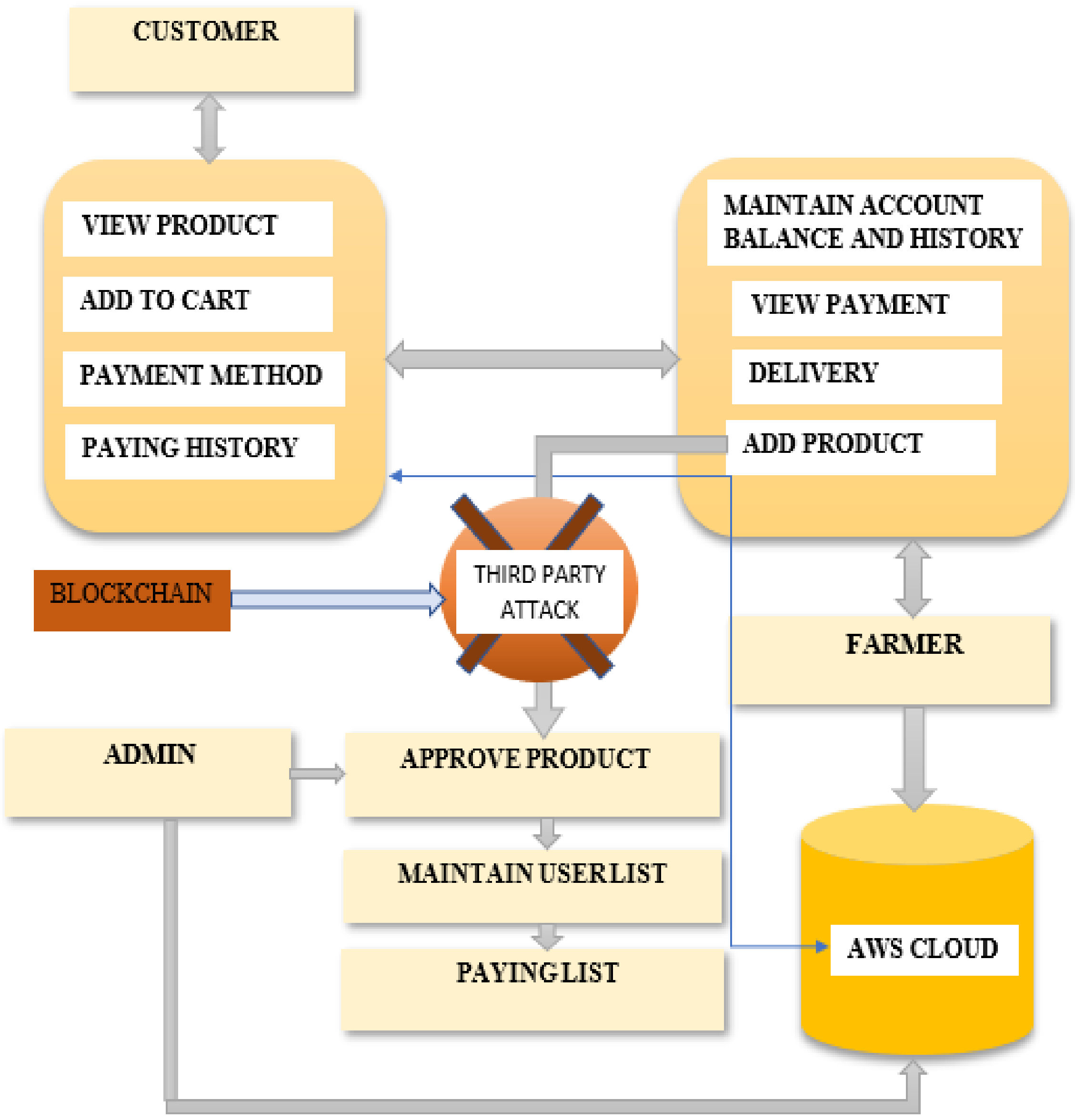


Fig. 5.1 System architecture.

The most crucial duty for AWS is maintaining the integrity, honesty, and accessibility of user data. The automatic monitoring system in AWS makes it more secure so that unauthorized access and third-party assaults may be stopped. The objective of AWS is to keep their own originality by offering trust and security for the individuals who use it. Several monitoring tools are available. Alarms, for instance, can be configured to alert the user of any outside threat. In this scenario, the participants are consumers, farmers, and administrators. Customers are the application's end users, and they may use it to get fresh produce that is available in their neighborhood, which eliminates the need for customers to go far in order to find fresh produce. Farmers, on the other hand, can upload their products and profit from the customers. Finally, the administrator's role is to verify the farmers and their products.

CHAPTER 5

SYSTEM IMPLEMENTATION

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 MODULE DESCRIPTION

LIST OF MODULES

- User
- Farmer
- Administrator

USER

In user module, user can view all the products added by the farmer's and can proceed to buy the products if he/she is in need of it, the products that they choose to buy can be delivered either to their place and it can be purchased directly from the farmer by the customer.

FARMER

In farmer module, farmers can add the product and check the product details. The farmer fixes the delivery mode as deliverable or not deliverable. Every payment will be view by the farmer like payment history and complete account details.

ADMIN

In admin module, admin should activate every user registration. if user register their details, it will be notified to the admin and admin can either accept the registration or deny it. Admins have the access to approve the farmer added product. Maintain all the user list, product list and all the details about the application.

5.1.1 MODULE DIAGRAM

USER

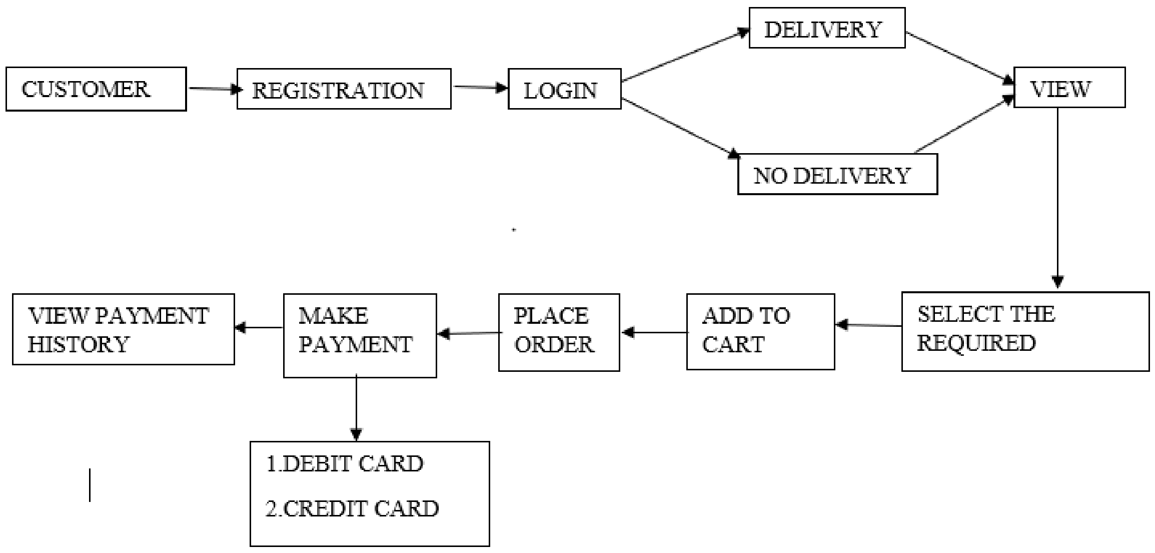


Fig.5.1.1.1 Workflow of customer.

FARMER

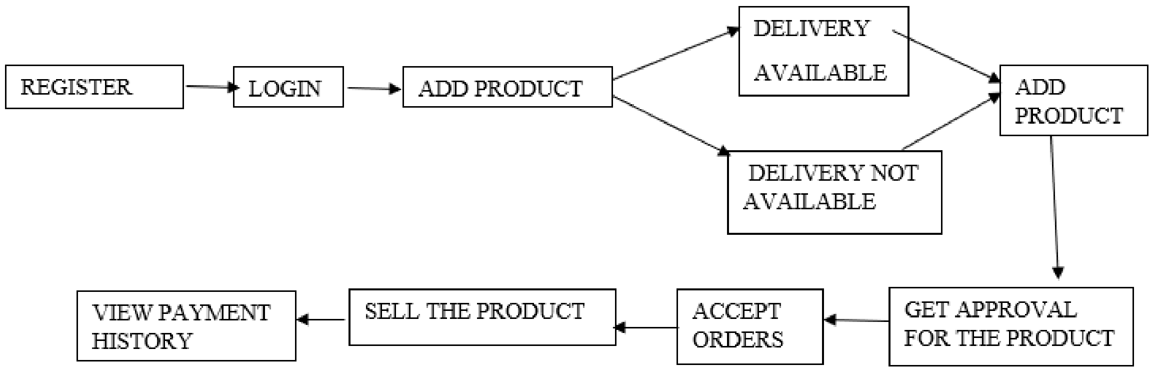


Fig. 5.1.1.2 Workflow of farmer.

ADMIN

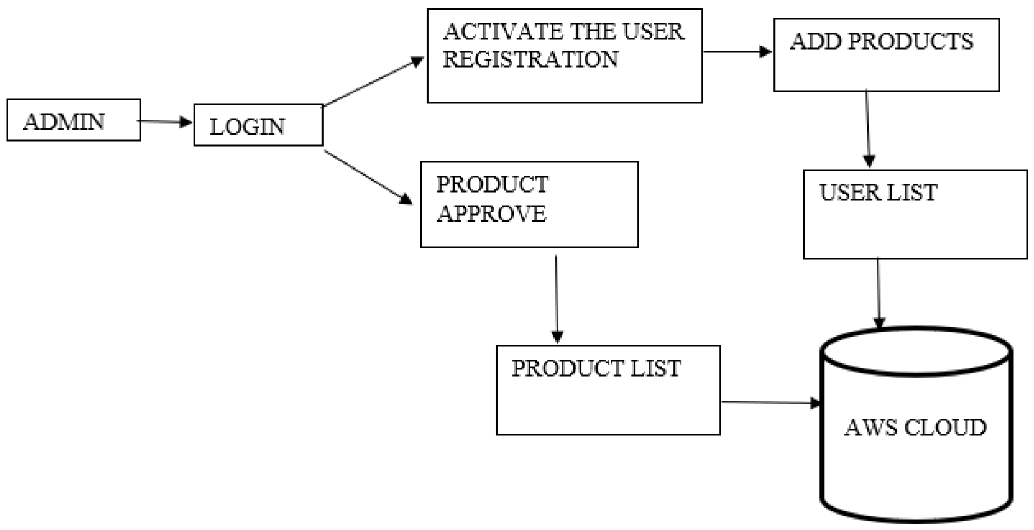


Fig. 5.1.1.3 Workflow of administrator.

5.2 SECURED HASH ALGORITHM

The hashing algorithm used to secure the transaction in our perspective article is SHA-256. Regardless of the size of the input transaction, this hashing algorithm always produces an output of 256 bits or 32 bytes. That is, if we use SHA-256 to hash two different inputs, say a 1 gigabyte movie and a 5-kilobyte image, The generated hash will be 256 bits long in dual instance. The only distinction is the hash pattern will be located between the two. The equations of SHA-256 code for various transactions is as follows.

$$T_1 = h + ch(e,f,g) + (\varepsilon^{256}, e) + w_{t} + e_{t}$$

$$T_2 = (\varepsilon^{256}, a) + maj(a,b,c)$$

Were ,

T_1, T_2 = variable

h , a = content buffer of h and a

ch = conditional function

ε = rotation function

Maj = majority function

The hash technique, which is crucial for online payments and transactions, is used to execute the blockchain concept. A hash code is generated when data is processed using a hashing algorithm and is determined in blocks known as hash blocks. These codes can be generated in increments of 1 bit, 128 bit, or 256 bits; however, in our work, 256-bit codes are generated for each transaction to increase the payment's security against third-party attack. Understanding the concept of Hashing is critical for understanding how blockchain works. Hashing is the act of taking any length input and producing a fixed length output item. Consider the use of blockchain in online transactions. Transactions of varying lengths are processed using a specific hashing algorithm, and all produce a fixed length output. The duration of the incoming transaction has no bearing on this output.

CHAPTER 6

SYSTEM TESTING

CHAPTER 6

SYSTEM TESTING

6.1 UNIT TESTING

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Unit testing is software verification and validation method in which the individual units of source code are tested fit for use. A unit is the smallest testable part of an application. In this testing, each class is tested to be working satisfactorily.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

6.2 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when put together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready. All the errors found in the system are corrected for the next phase.

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

6.3 TEST CASES AND REPORT

TEST REPORT:01

PRODUCT: Seller Registering to the application

USECASE: Seller Registration

Test case id	**Testcase/ action to be performed**	**Expected Result**	**Actual Result**	**Pass/Fail**
1	The seller/Farmer registers himself	Registered Successfully	Registered Successfully	Pass

	as shown in A.2.3.1			
2	Adding Products as shown in A.2.3.2	Adds the products successfully	Adds the product Successfully	Pass

Table 6.3.1 Test case for seller registration

TEST REPORT:02

PRODUCT: Customer registering themselves to the application

USECASE: Customer Registration

Testcase id	**Testcase/Action to be performed**	**Expected Result**	**Actual Result**	**Pass/Fail**
1	Customers give in their location and email id and registers themselves as shown in A.2.3.1	Registered Successfully	Registered Successfully	Pass
2	Adds the product to cart and buys them as shown in A.2.3.1	Successfully done	Successfully done	Pass

Table 6.3.2 Test case for customer registration

TEST REPORT:03

PRODUCT: Admin logins himself and approves the seller and their product

USECASE: Admin login

Testcase id	**Testcase/Action to be performed**	**Expected Result**	**Actual Result**	**Pass/Fail**
1	Admin logins and verifies the seller through the given id as shown in A.2.3.3	Verified successfully	Verified successfully	Pass
2	Admin approves the product as shown in A.2.3.3	Approved Successfully	Approved Successfully	Pass

Table 6.3.3 Test case for Administrator login

TEST REPORT:04

PRODUCT: Aws cloud console for cloud storage

USECASE: Aws cloud storage

Testcase id	Testcase/Action to be performed	Expected Result	Actual Result	Pass/Fail
1	The database is connected to the cloud and can view the database as shown in A.2.2	Successfully done	Successfully done	Pass

Table 6.3.4 Test case for database connectivity to cloud

TEST REPORT:05

PRODUCT: Hash code Generated in sql and data connectivity

USECASE: Hash code generation

Testcase id	Testcase/Action to be performed	Expected Result	Actual Result	Pass/Fail
1	Hash code to be generated for protecting data as shown in A.2.1	Successfully generated	Successfully generated	Pass

Table 6.3.5 Test case for generation of hash code

TEST REPORT:06

PRODUCT: Preventing Access of unknown user with the help of blockchain

USECASE: Access Denial of unknown user

Testcase id	Testcase/Action to be performed	Expected Result	Actual Result	Pass/Fail
1	if an unknown user tries to access their login access is denied as shown in A.2.4.2	Access Denied Successfully	Access denied Successfully	Pass

Table 6.3.6 Test case for the verification of blockchain concept

CHAPTER 7

CONCLUSION

CHAPTER 7
CONCLUSION

7.1 RESULTS & DISCUSSION

People and the country's economy grow as technology advances, but one community that has not been affected is farmers. Farmers' economic development is still lagging due to intermediaries who buy products at lower prices and sell them to end users at a higher profit, this imbalance is simply due to farmers who do not have knowledge about using technology wisely, but as the days changed and internet reached every corner of the world, people began to use all the applications wisely, in this perspective article, we have come up with a solution that benefits everyone. Here farmer can directly sell their product to the consumer without any intermediate and with the price that they wish to fix for their products. If farmers are exporting products to long distance, it may cost more and the quality(freshness) of the product can be spoiled, so we are connecting the consumers with the nearby farmers. The consumers can choose the product only from the nearby farmers, and they can choose whether it is taken away or the farmers will deliver it. This service has been enhanced with additional features using AWS (for permanent data storage) and blockchain(for secured transactions).

7.2 CONCLUSION AND FUTURE ENCHANCEMENT

In this the transaction between the consumers are made easier. This helps us to understand the issues of customers easily. The art of doing business is made comfortable for every farmer. farmers are benefited by selling online and it is made easier for them to understand. It acts like a channel for businesses to reach out to customers and a way for a community to work together to solve problems for customers. The methods that are used to improve this application in the future are Setting up a genuine database system, Increasing thenumber of messages exchanged and the size of those messages while also increasing the efficiency of protocols and Application of two or more algorithms.

APPENDICES

APPENDIX 1 SAMPLE CODE

APPENDIX 1.1 CLIENT-SIDE CODING

Index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

</head>

<style>

* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: "Abel", sans-serif;

  font-size: 10px;

  scroll-behavior: smooth;

}

.wrapper {
  width: auto;

  height: 100vh;

background-image: linear-gradient( rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),
url("https://images.unsplash.com/photo-1523741543316-beb7fc7023d8?ixlib=rb-
1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=c
rop&w=387&q=80");

  background-position: center;

  background-size: cover;

  background-repeat: no-repeat;

  backdrop-filter: opacity(80%);

}

.Container {

  width: 100%;

  height: 100%;

  display: flex;

  justify-content: center;

  align-items: center;

}

.nav {

  position: fixed;

  top: 0;

  left: 0;

  width: 100%;

  height: 80px;

  border-bottom: 1px solid rgba(255, 255, 255, 0.521);
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 0 50px;
}
.logo {
  font-family: "Abel", sans-serif;
  font-size: 2.5rem;
  font-weight: 600;
  letter-spacing: 0.7rem;
  color: white;
  margin: 4%;
}
.menu {
  display: inline-block;
  line-height: 80px;
}
.menu ul {
  list-style: none;
  /* display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center; */
```

```
}
.menu ul li {
  display: inline-block;
}
.menu ul li a {
  text-decoration: none;
  font-family: "Raleway", sans-serif;
  font-size: 1.2rem;
  font-weight: 600;
  letter-spacing: 0.1rem;
  color: white;
  border: 1px solid transparent;
  border-radius: 4px;
  padding: 10px 15px;
  margin: 0 5px;
  transition: 0.5s ease;
}
.menu ul li a:hover {
  border-color: white;
}
.menu ul li:nth-child(5) a {
  color: #fff200;
  border: 1px solid #fff200;
```

```
}
.menu ul li:nth-child(5) a:hover {
  color: black;
  background-color: #fff200;
}
.header {
  text-align: center;
}
.header h1 {
  font-family: "Raleway", sans-serif;
  font-size: 4rem;
  font-weight: 600;
  letter-spacing: 0.2rem;
  color: white;
  padding: 45% 20px 8px;
}
.header p {
  font-family: "Raleway", sans-serif;
  font-size: 1.5rem;
  font-weight: 600;
  letter-spacing: 0.2rem;
  color: white;
  padding: 10px 15px;
```

```
}
button {
  font-size: 1.5rem;
  font-weight: 600;
  letter-spacing: 0.15rem;
  color: black;
  background-color: #fff200;
  padding: 20px 30px;
  margin: 50px 5px 0;
  border: none;
  cursor: pointer;
}
  </style>
<body>
<div class="wrapper">
  <div class="Container">
        <div class="nav">
            <div class="logo">
</div>
            <div class="menu">
                <ul class="navMenu">
                    <li><a href="#">Home</a></li>
                    <li><a href="customerlog.jsp">Customer</a></li>
                    <li><a href="sellerlog.jsp">Seller</a></li>

                    <li><a href="adminlog.jsp">Admin</a></li> </ul>

            </div>

        </div>

        <div class="header">

            <h1>Welcome</h1>

</div>

    </div>

</div>

</body>

</html>
```

Customerlogin.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

<style>

.myDiv {

  border: 5px outset #77c732;

  background-color: lightblue;
  border-radius: 10px;
  width:600px;
  height:400px;
  margin: auto;
  padding-top:30px;
  box-shadow: 25px 20px 20px #888888;

}
.myDiv2 {
  font-size:25px;
  font-style: italic;
font-weight: bold;
color: #ffff00;
}
.myDiv3 {
  font-size:20px;
  font-style: italic;
font-weight: bold;
color: white;
}
a:link{
  background-color: #f44336;
  color: white;
  padding: 10px 20px;

  text-align: center;

  text-decoration: none;

  display: inline-block;

  border-radius:8px;

}

body {

  background-image: url("images/log1.png");

  background-repeat: no-repeat;

  background-size: cover;

}

</style>

</head>

<body>

<br><br><br><br><br>

<center>

<div class="myDiv2">

Consumer-1 Login

</div>

</center>

<br><br>

<center>

  <form action="Cons1logservlet" method="post">
  <input type="text" name="email" placeholder="email" style="width:280px;height:40px;border-
radius: 10px;text-align:center;"><br><br>

  <input type="password" name="password" placeholder="Password"
style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>

  <input type="submit" value="Submit" style="width:100px;height:40px;border-radius:
10px;"><br><br>

<div class="myDiv3">

New User <a href="consumer1reg.jsp">Signup</a>

</div>

</form></center>

</body>

</html>
```

sellerreg.jsp;

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

 <link rel="stylesheet" href="css1/bootstrap.min.css">

    <link rel="stylesheet" href="css1/font-awesome.min1.css">

  <style>

  body {
```

```
  background-image:url("images/27.jpg");

  background-size:cover;

}

  .card {

  background-image:url("images/2.gif");

  background-size:cover;

}

  </style>

</head>

<body>

  <div class="container">

     <form method="post" action="sellerregservlet" enctype="multipart/form-data">

     <div class="row">

       <div class="card col-sm-6" style="padding-bottom:10px;background-color:
#333;color:#fff;padding:10px;margin-left:30%;width: 38em;margin-bottom: 25px">

<div class="form-group">

          <label for="full_name" style="color:#fff;"> Name:</label>

          <input type="text" class="form-control" id="full_name" placeholder="Full Name"
name="name" required>

        </div>

        <div class="form-group">

          <label for="email"  style="color:#fff;">Email :</label>

          <input type="email" class="form-control" id="email" placeholder=" Email" name="email"
required>

</div>
```

```
        <div class="form-group">

          <label for="text"  style="color:#fff;">Mobile :</label>

          <input type="text" class="form-control" id="email" placeholder=" contact No "
name="mobile"  required>

          </div>

<div class="form-group">

          <label for="text"  style="color:#fff;">Location :</label>

          <input type="text" class="form-control" id="email" placeholder="Your Location"
name="location"  required>

          </div>

        <div class="form-group">

          <label for="text"  style="color:#fff;">Address :</label>

          <input type="text" class="form-control" id="email" placeholder=" Enter Address "
name="address"  required>

          </div>

<div class="form-group">

          <label for="text"  style="color:#fff;">Password :</label>

          <input type="text" class="form-control" id="password1" placeholder="Password "
name="psw"  required>

          </div>

<div class="form-group">

          <label for="text"  style="color:#fff;">Re-Enter password:</label>

          <input type="password" class="form-control" id="password2" placeholder="Confirm
Password" name="cpsw"  required>

          </div>

   <div class="form-group">
```

```
          <label for="card_photo"  style="color:#fff;">ID Proof :</label>

          <input type="file" class="form-control" placeholder="Enter password" name="ID Proof"
accept="image/*" onchange="preview_image(event)" required>

        </div>

        <div class="form-group">

          <label  style="color:#fff;">Picture Here!!!</label><br>

          <img src="" id="output_image"/ height="200px">

        </div>

        <center><button id="submit" name="donator_register" class="btn btn-primary btn-block"
style="width:50%;" onclick="return Validate()">Submit</button></center><br>

      </div>

       </div>

      </form>

</div>

    <hr>

</body>

<script type='text/javascript'>

  function preview_image(event)

  {

    var reader = new FileReader();

    reader.onload = function()

    {

      var output = document.getElementById('output_image');

      output.src = reader.result;
    }

    reader.readAsDataURL(event.target.files[0]);

  }

  </script>

  <script>

window.onload = function () {

document.getElementById("password1").onchange = validatePassword;

document.getElementById("password2").onchange = validatePassword;

                                        }

function validatePassword() {

                                                  var pass2 = 
document.getElementById("password2").value;

                                                  var pass1 = 
document.getElementById("password1").value;

                                                  if (pass1 != pass2)

          document.getElementById("password2").setCustomValidity("Password Doesn't 
Match");else

          document.getElementById("password2").setCustomValidity('');

                                                  //empty string means no validation error

                                        }

                              </script>

</html>
```

sellerlog.jsp;

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

<link rel="stylesheet" href="css/mlogin.css">

</head>

<style>

#text{text-allign:left;}

a{

text-decoration:none;

color:red;

</style>

<body>

<div class="content">

    <form action="sellerlogservlet" method="post">

      <center>

        <div class="column">

          <div class="row">

            <H1>Seller Login</H1>

            <img id="profile" src="images/7.png">


          </div>
```

```
                    <br>

                      <div class="form-group">

      </div>

     <br>

                    <div class="row">

                         <label for=""> </label>

                         <input type="text" required = "true" name="mail" id="" placeholder="Enter
Email" style="width:250px;height:30px;"  >

                        </div>

                        <br>

                    <div class="row">

                        <label for=""></label>

                        <input type="text" required="true" name="password" id="myInput"
placeholder="Enter Your Location"  autocomplete="off" style="width:250px;height:30px;" >

        </div><br>

   <div class="row">

                        <button type="submit">Login</button><br><br>

                    </div>

                    <a href="sellerreg.jsp">New User SignUp</a>

                </div>

               </center>

 </form>

         </div>

</body>
```

```
<script>
function myFunction() {
  var x = document.getElementById("myInput");
  if (x.type === "password") {
    x.type = "text";
  } else {
    x.type = "password";
  }
}
</script>
</html>
```

APPENDIX 1.2 SERVER-SIDE CODING

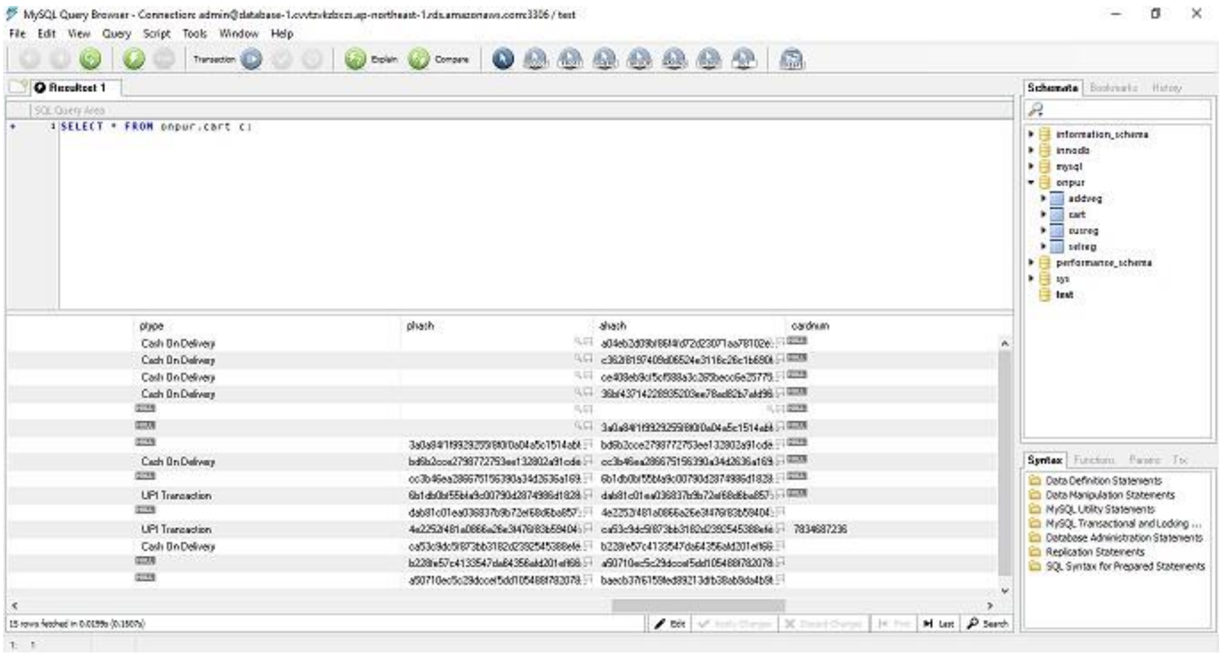
Consumerservlet;

```
package servlet;
import imple.Imple;
import inter.Inter;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import bean.Consumer1bean;

/**

 * Servlet implementation class Cons1logservlet

 */

@WebServlet("/Cons1logservlet")

public class Cons1logservlet extends HttpServlet {

	private static final long serialVersionUID = 1L;

    /**

     * @see HttpServlet#HttpServlet()

     */

    public Cons1logservlet() {

        super();

        // TODO Auto-generated constructor stub

    }



	/**

	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

	 */

	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

		// TODO Auto-generated method stub

	}

/**
```

```
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

         */

        protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

                // TODO Auto-generated method stub

                String email=request.getParameter("email");

                HttpSession session=request.getSession();

                session.setAttribute("mail",email);

                String password=request.getParameter("password");

                System.out.println("Password:"+password);

                Consumer1bean al=new Consumer1bean();

                al.setEmail(email);

                al.setPassword(password);

                Inter u=new Imple();

                boolean r=u.cal(al);

                if(r==true)

                {

                        response.sendRedirect("consumer1main.jsp");

                }

                else

                {

                        response.sendRedirect("Error");

                }}}

Boolfileservlet;
```

```
package servlet;

import imple.Imple;

import inter.Inter;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import bean.Bookfilebean

/**

 * Servlet implementation class Bookfileservlet

 */

@WebServlet("/Bookfileservlet")

public class Bookfileservlet extends HttpServlet {

        private static final long serialVersionUID = 1L;

     /**

    * @see HttpServlet#HttpServlet()

    */

   public Bookfileservlet() {

       super();

       // TODO Auto-generated constructor stub

   }
/**

	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

	 */} }

	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

		// TODO Auto-generated method stub

	}

    /**

	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

	 */

	protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

		// TODO Auto-generated method stub

		String booktype=request.getParameter("booktype");

		   System.out.println("booktype: "+booktype);

		   String bookname=request.getParameter("bookname");

		   System.out.println("bookname: "+bookname);

		   String file=request.getParameter("file");

		   System.out.println("file: "+file);

		 String bookquality=request.getParameter("bookquality");

		   System.out.println("bookquality: "+bookquality);

		   String price=request.getParameter("price");

		   System.out.println("price: "+price);

		    String email=request.getParameter("email");
    System.out.println("email: "+email);

    Bookfilebean bk=new Bookfilebean();

    bk.setBooktype(booktype);

    bk.setBookname(bookname);

    bk.setFile(file);

    bk.setBookquality(bookquality);

    bk.setPrice(price);

    bk.setEmail(email);

Inter r=new Imple();

    int m=r.bkf(bk);

   if(m==1)

   {

            response.sendRedirect("consumer1main.jsp");

   }

   else

   {

            response.sendRedirect("Error");

    }
```

APPENDIX 2 SAMPLE SCREEN

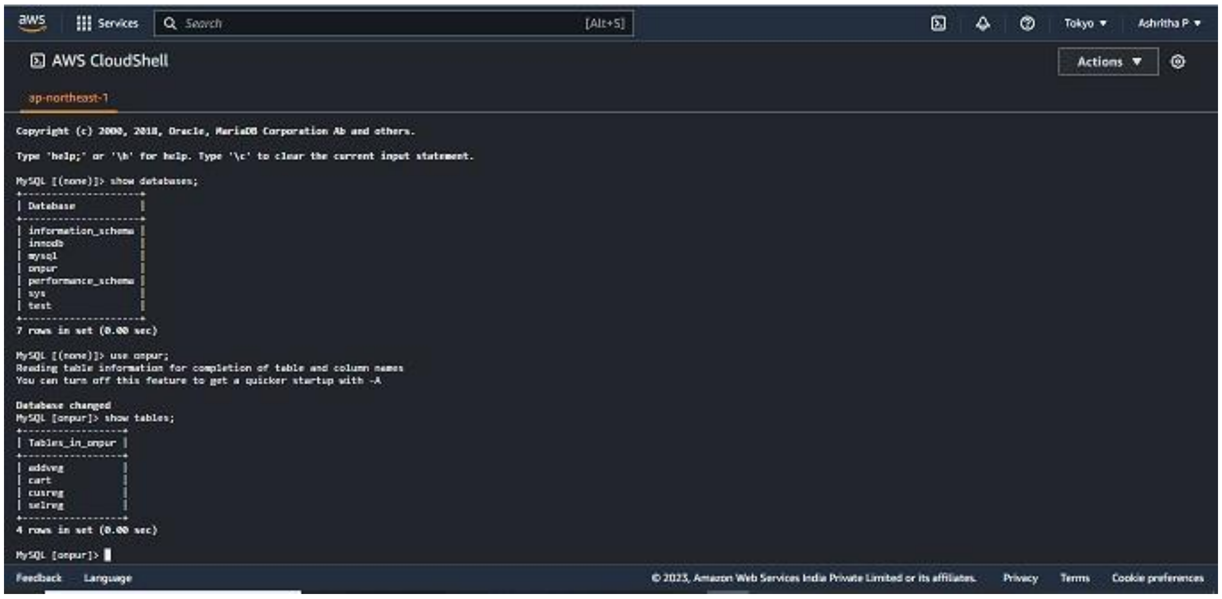

MySQL Query Browser - Connection: admin@database-1.cvvtzvkzbzcz.ap-northeast-1.rds.amazonaws.com:3306 / test
File Edit View Query Script Tools Window Help
Transaction
Explain
Compare
Resultset 1
SQL Query Area
Schemata Bookmarks History
information_schema
innodb
mysql
onpur
addveg
cart
cusreg
selreg
performance_schema
sys
test
Syntax Functions Params Trx
Data Definition Statements
Data Manipulation Statements
MySQL Utility Statements
MySQL Transactional and Locking ...
Database Administration Statements
Replication Statements
SQL Syntax for Prepared Statements
Edit
Apply Changes
Discard Changes
First
Last
Search

```
SELECT * FROM onpur.cart c;
```

ptype	phash	ahash	cardnum
Cash On Delivery		a04eb2d09bf86f4fd72d23071aa78102e	NULL
Cash On Delivery		c362f8197409d06524e3116c26c1b680	NULL
Cash On Delivery		ce409eb9cf5cf988a3c269becc6e25775	NULL
Cash On Delivery		36bf43714228935203ee78ad82b7afd98	NULL
NULL			NULL
NULL		3a0a84f1f99292559f80f0a04a5c1514ab	NULL
NULL	3a0a84f1f99292559f80f0a04a5c1514ab	bd6b2cce2798772753ee132802a91cde	NULL
Cash On Delivery	bd6b2cce2798772753ee132802a91cde	cc3b46ea286675156390a34d2636a169	NULL
NULL	cc3b46ea286675156390a34d2636a169	6b1db0bf55b4a9c00790d2874986d1828	NULL
UPI Transaction	6b1db0bf55b4a9c00790d2874986d1828	dab81c01ea036837b9b72ef68d6ba857	NULL
NULL	dab81c01ea036837b9b72ef68d6ba857	4e2252f481a0866a26e3f476f83b59404	
UPI Transaction	4e2252f481a0866a26e3f476f83b59404	ca53c9dc5f873bb3182d2392545388efe	7834687236
Cash On Delivery	ca53c9dc5f873bb3182d2392545388efe	b228fe57c4133547da64356ald201ef66	
NULL	b228fe57c4133547da64356ald201ef66	a50710ec5c29dccef5dd105488f782078	
NULL	a50710ec5c29dccef5dd105488f782078	baecb37f6159fed89213dfb38ab9da4b9	

15 rows fetched in 0.0199s (0.1507s)

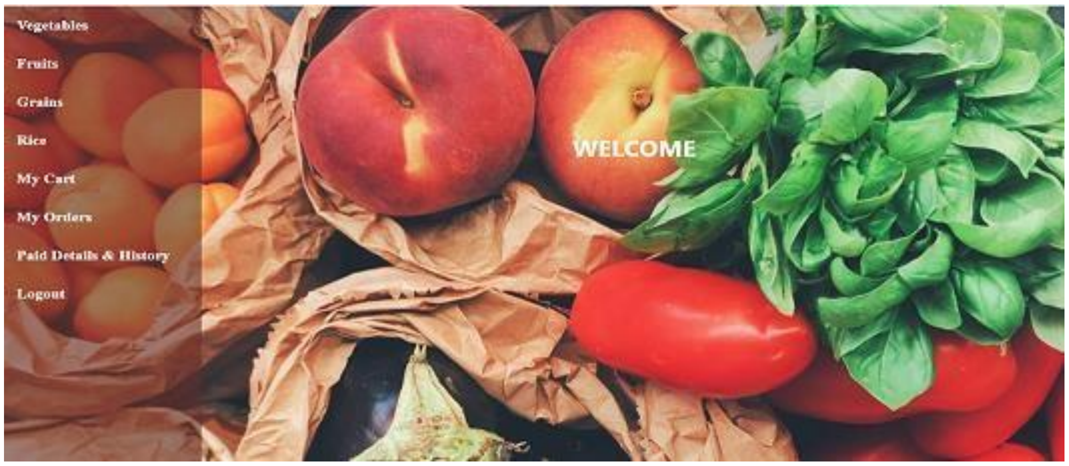
Scr A.2.1 Screenshot of Sha Code Generated For Various Transactions**.**



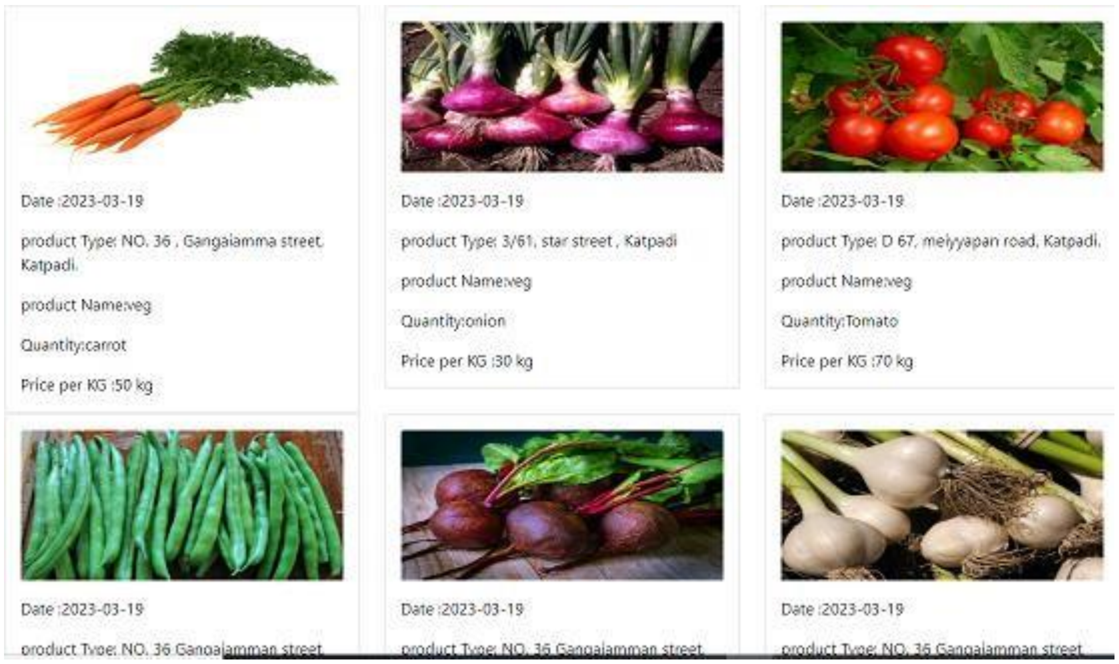
```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| onpur              |
| performance_schema |
| sys                |
| test               |
+--------------------+
7 rows in set (0.00 sec)

MySQL [(none)]> use onpur;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [onpur]> show tables;
+-----------------+
| Tables_in_onpur |
+-----------------+
| addveg          |
| cart            |
| cusreg          |
| selreg          |
+-----------------+
4 rows in set (0.00 sec)

MySQL [onpur]> 
```

Scr A.2.2 screenshot of aws console for cloud storage.

Scr. **A.2.3 SCREENSHOT OF VARIOUS SLIDES IN APPLICATION**.



Scr A.2.3.1 Home page of the application


Date :2023-03-19

product Type: NO. 36 , Gangaiamma street, Katpadi.

product Name:veg

Quantity:carrot

Price per KG :50 kg

Date :2023-03-19

product Type: 3/61, star street , Katpadi

product Name:veg

Quantity:onion

Price per KG :30 kg

Date :2023-03-19

product Type: D 67, meiyyapan road, Katpadi.

product Name:veg

Quantity:Tomato

Price per KG :70 kg

Date :2023-03-19

product Type: NO. 36 Gangaiamman street

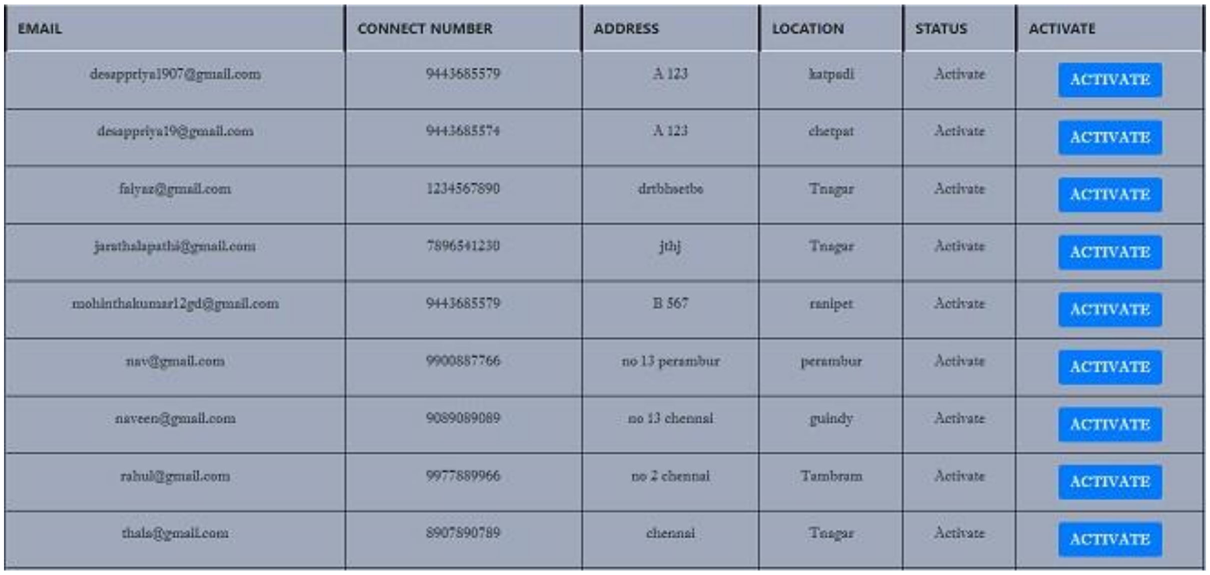
Date :2023-03-19

product Type: NO. 36 Gangaiamman street

Date :2023-03-19

product Type: NO. 36 Gangaiamman street

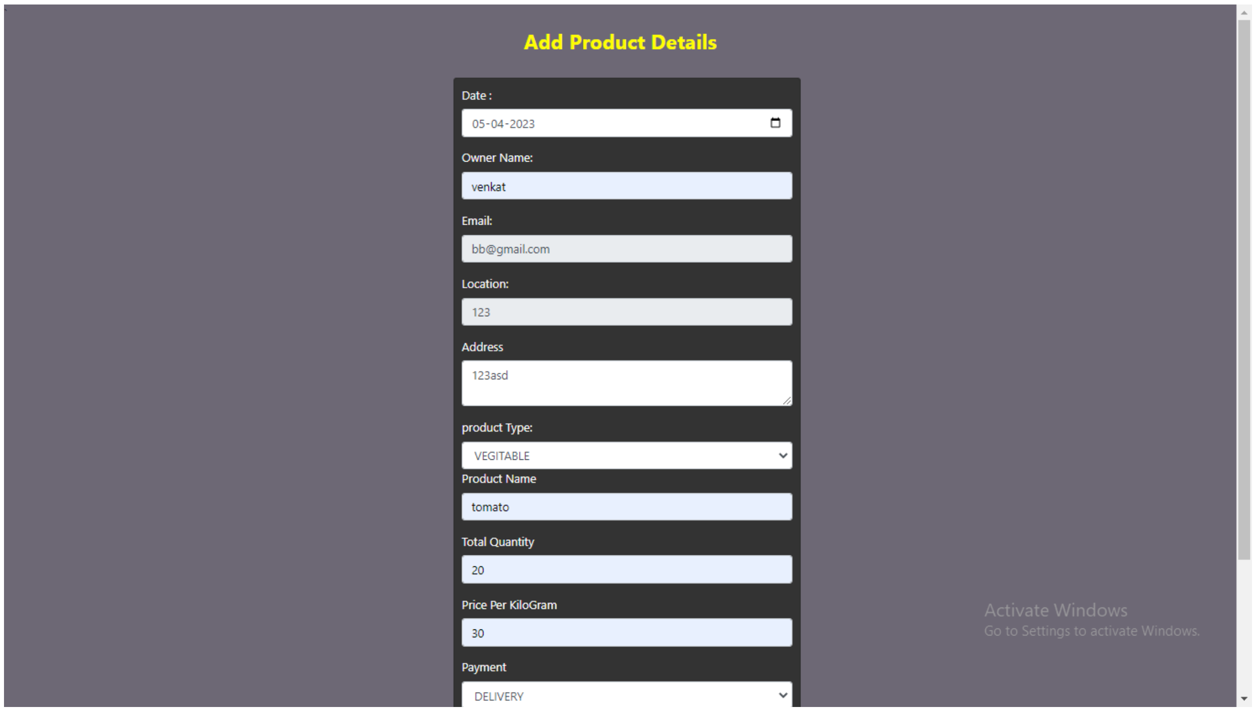
Scr. A.2.3.2 Products added by farmers.



EMAIL	CONNECT NUMBER	ADDRESS	LOCATION	STATUS	ACTIVATE
desappriya1907@gmail.com	9443685579	A 123	katpadi	Activate	ACTIVATE
desappriya19@gmail.com	9443685574	A 123	chetpat	Activate	ACTIVATE
faiyaz@gmail.com	1234567890	drtbhsetbe	Tnagar	Activate	ACTIVATE
jarathalapathi@gmail.com	7896541230	jthj	Tnagar	Activate	ACTIVATE
mohinthakumar12gd@gmail.com	9443685579	B 567	ranipet	Activate	ACTIVATE
nav@gmail.com	9900887766	no 13 perambur	perambur	Activate	ACTIVATE
naveen@gmail.com	9089089089	no 13 chennai	guindy	Activate	ACTIVATE
rahul@gmail.com	9977889966	no 2 chennai	Tambram	Activate	ACTIVATE
thala@gmail.com	8907890789	chennai	Tnagar	Activate	ACTIVATE

Scr. A.2.3.3 Verification by Admin

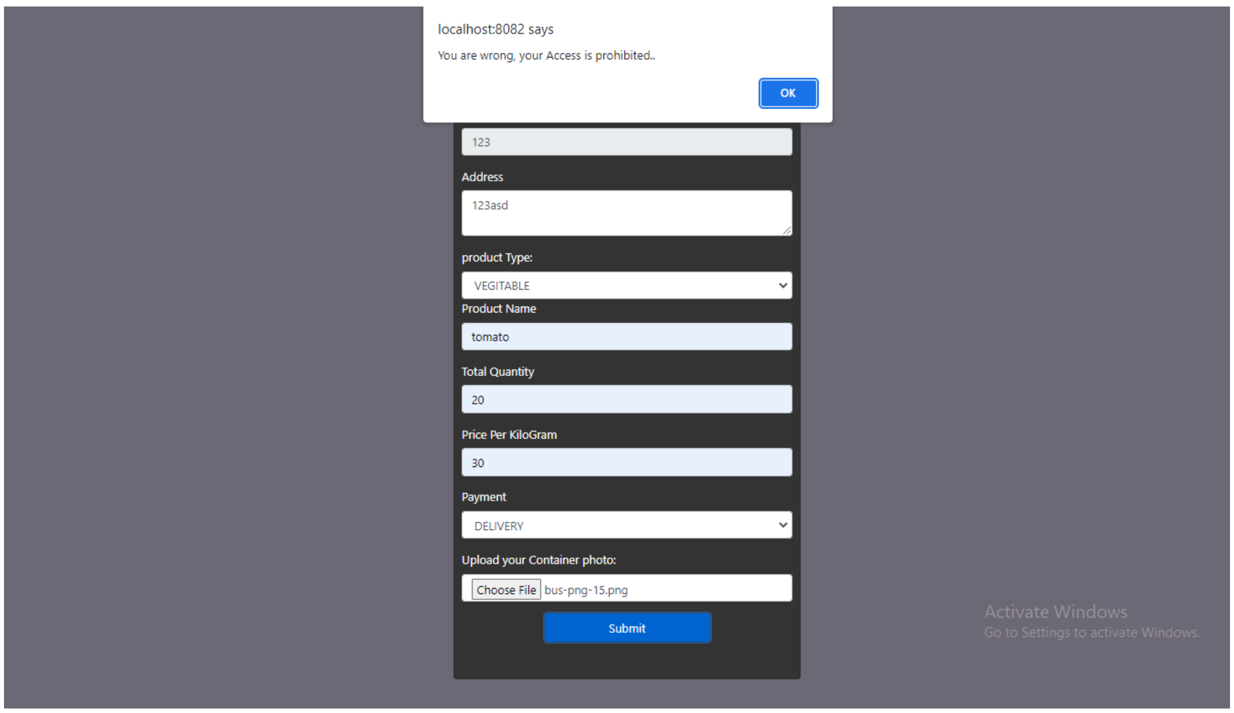
Scr. A.2.4 SCREENSHOT FOR TESTING THE WORKING OF SECURED HASH ALGORITHM



Add Product Details

Date :	05-04-2023
Owner Name:	venkat
Email:	bb@gmail.com
Location:	123
Address	123asd
product Type:	VEGITABLE
Product Name	tomato
Total Quantity	20
Price Per KiloGram	30
Payment	DELIVERY

Scr. A.2.4.1 An unknown user trying to access the information of other user



localhost:8082 says

You are wrong, your Access is prohibited..

123

Address

123asd

product Type:

VEGITABLE

Product Name

tomato

Total Quantity

20

Price Per KiloGram

30

Payment

DELIVERY

Upload your Container photo:

Choose File bus-png-15.png

Scr. A.2.4.2 A message denying the access from unknown user

REFERENCES

[1] A. Aflaki, B. Feldman, and R. Swinney, “Becoming strategic: Endogenous consumer time preferences and multiperiod pricing,” *Oper. Res.*, vol. 68, no. 4, pp. 1116–1131, 2020.

[2] S. Asian and X. Nie, “Coordination in supply chains with uncertain demand and disruption risks: Existence, analysis, and insight,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 9, pp. 1139–1154, Sep. 2014.

[3] Y. Aviv and A. Pazgal, “Optimal pricing of seasonal products in the presence of forward-looking consumers,” *Manuf. Service Oper. Manage.*, vol. 10, no. 3, pp. 339–359, 2008.

[4] I. Bellos, M. Ferguson, and L. B. Toktay, “The car sharing economy: Interaction of business model choice and product line design,” *Manuf. Service Oper. Manage.*, vol. 19, no. 2, pp. 185–201, 2017.

[5] S. Benjaafar and M. Hu, “Operations management in the age of the sharing economy: What is old and what is new?,” *Manuf. Service Oper. Manage.*, vol. 22, no. 1, pp. 93–101, 2019.

[6] S. Benjaafar, G. Kong, X. Li, and C. Courcoubetis, “Peer-to-peer product sharing: Implications for ownership, usage, and social welfare in the sharing economy,” *Manage. Sci.*, vol. 65, no. 2, pp. 477–493, 2018.

[7] N. Boysen, D. Briskorn, and S. Schwerdfeger, “Matching supply and demand in a sharing economy: Classification, computational complexity, and application,” *Eur. J. Oper. Res.*, vol. 278, no. 2, pp. 578–595, 2019.

[8] G. P. Cachon and R. Swinney, “Purchasing, pricing, and quick response in the presence of strategic consumers,” *Manage. Sci.*, vol. 55, no. 3, pp. 497–511, 2009.

[9] K. Cao, X. Xu,Y. Bian, andY. Sun, “Optimal trade-in strategy of businessto- consumer platform with dual-format retailing model,” *Omega*, vol. 82, pp. 181–192, 2019.

[10] C. H. Chiu, H. L. Chan, and T. M. Choi, “Risk minimizing price-rebatereturn
contracts in supply chains with ordering and pricing decisions: A multimethodological analysis,” *IEEE Trans. Eng. Manage.*, vol. 67, no. 2, pp. 466–482, 2020.

[11] C. T.M., J. Zhang, andY. J. Cai, “Consumer-to-consumer digital-productexchange in the sharing economy system with risk considerations: Will
digital-product-developers suffer?,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 12, pp. 5049–5057, Dec. 2020.

[12] T. M. Choi and Y. He, “Peer-to-peer collaborative consumption for fashion
products in the sharing economy: Platform operations,” *Transp. Res. Part E: Logistics Transp. Rev.*, vol. 126, pp. 49–65, 2019.

[13] R. H. Coase, “Durability and monopoly,” *J. Law Econ.*, vol. 15, no. 1, pp. 143–149, 1972.

[14] M. Gupta, P. Esmaeilzadeh, I. Uz, and V. M. Tennant, “The effects of national cultural values on individuals’ intention to participate in peer-topeer sharing economy,” *J. Bus. Res.*, vol. 97, pp. 20–29, 2019.

[15] I. Henssssdel and A. Lizzer, “Adverse selection in durable goods markets,” *Amer. Econ. Rev.*, vol. 89, no. 5, pp. 1097–1115, 1999.

[16] I. Hendel and A. Lizzer, “Interfering with secondary markets,” *RAND J. Econ.*, vpl. vol. 30, no. 1, pp. 1–21, 1999b.