

FARMERS HELP

A PROJECT REPORT

submitted By

ANJANA P T

TVE20MCA-2015

to

the APJ Abdul Kalam Technological University
in partial fullfilment of the requirements for the award of the degree

of

Master of Computer Applications



Department of Computer Applications

College of Engineering

Trivandrum-695016

MARCH 2022

Declaration

I undersigned hereby declare that the project report titled "**FARMERS HELP**" submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Dr. Sabitha S, Assoc.Professor. This submission represents my ideas in my words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum

Anjana P T

Date : 17/02/2022

DEPARTMENT OF COMPUTER APPLICATIONS
COLLEGE OF ENGINEERING
TRIVANDRUM



CERTIFICATE

This is to certify that the report entitled **Farmers Help** submitted by **Anjana PT** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Internal Supervisor

Head of the Dept

Acknowledgement

First and for most I thank **GOD** almighty and to my parents for the success of this project. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to **Dr Jiji C V**, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Assoc Prof. Deepa S S**, HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

I express our sincere thanks to **Dr. Sabitha S**, Assoc. Professor, Department of Computer Applications, College of Engineering Trivandrum for her valuable guidance, support and advice that aided in the successful completion of my project.

I profusely thank other Assoc. Professors in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

Anjana P T

Abstract

The FARMERS HELP System is a web application that provides complete coordination of various activities handled by different classes of people in the farming field.

Farmers, who are registered, can upload their agricultural strategies or technologies to the system. There by providing other farmers or public can able to track those particular technologies to their farming procedures.

Farmers are able to take help from specialists for their doubts. The system helps to enhance the communication between farmers and agricultural experts. It is also a platform to buy and sell crops, seeds, equipment and pesticides online.

Through a website like this, farmers can connect directly with customers and make their transactions profitable for them without a middleman/third party. The website is a one point solution for helping farmers

Contents

1	Introduction	1
2	Literature Review	2
2.1	Existing System	2
2.2	Proposed System	2
3	Requirement Analysis	4
3.1	Purpose	4
3.1.1	Hardware Requirements	4
3.1.2	Software Requirements	4
3.2	Functional Requirements	5
3.3	Non Functional Requirements	5
3.3.1	Performance Requirements	5
3.3.2	Quality Requirements	5
4	Feasibility Study	7
4.1	Technical Feasibility	8
4.2	Economical feasibility	8
4.3	Operational Feasibility	9
4.4	Schedule Feasibility	9
5	Design And Implementation	10
5.1	Users	10
5.2	Module Design	10
5.2.1	Registration	10
5.2.2	Admin	11

5.2.3	Farmer	11
5.2.4	Exporter	11
5.2.5	Supplier	12
5.2.6	Investor	12
5.2.7	Specialist	12
5.2.8	Public	12
5.3	System Design	12
5.4	Input Design	13
5.5	Output Design	14
5.5.1	Data Flow Diagram	14
5.5.2	ER Diagram	21
5.5.3	Database Design	22
5.6	Screenshots	35
6	Coding	42
6.1	views.py	42
7	Testing and Implementation	54
7.1	Testing and various types of testing used.	54
7.1.1	Unit Testing	55
7.1.2	Integration Testing	55
7.1.3	System Testing	55
7.1.4	Test Case Description	55
8	Results and Discussion	59
8.1	Advantages and Limitations	59
8.1.1	Advantages	59
8.1.2	Limitations	59
9	Conclusion	60
10	Future Scope	61

List of Figures

5.1	Level 0 DFD	15
5.2	Level 1 DFD	16
5.3	Level 2 Farmer Management	17
5.4	Level 2 Specialist Management	17
5.5	Level 2 Public Management	18
5.6	Level 2 Supplier Management	18
5.7	Level 2 Exporters Management	19
5.8	Level 2 Investor Management	19
5.9	Level 2 Category Management	20
5.10	Level 2 Product Management	20
5.11	ER Diagram	21
5.12	Login Table	22
5.13	Farmerreg Table	23
5.14	Exporters Table	23
5.15	Specialist Table	24
5.16	Supplier Table	24
5.17	Investors Table	25
5.18	Seed Table	25
5.19	Seedcart Table	26
5.20	Seedpayment Table	26
5.21	Product Table	27
5.22	Productcart Table	27
5.23	Productpayment Table	28
5.24	Equipmentcart Table	28

5.25 Equipmentpayment Table	29
5.26 Pesticidecart Table	29
5.27 Pesticidepayment Table	30
5.28 Category Table	30
5.29 Publicreg Table	31
5.30 Suppliersproduct Table	31
5.31 Doubt Table	32
5.32 Doubtreply Table	32
5.33 Exportersorder Table	33
5.34 Farmercapitalrequest Table	33
5.35 Farmingtechniques Table	34
5.36 Feedback Table	34
5.37 Login Page	35
5.38 Farmer Registration	35
5.39 Exporter Registration	36
5.40 Supplier Registration	36
5.41 Public Registration	37
5.42 Investor Registration	37
5.43 View Profile	38
5.44 Admin-Approve Farmer	38
5.45 Farmer-Add Product	39
5.46 Specialist-View Doubts	39
5.47 Farmer-View Doubt Reply	40
5.48 Exporter-Make Payment	40
5.49 View Product Cart	41
5.50 View Equipment Booking	41

Chapter 1

Introduction

Agriculture along with its associated sectors is definitely the biggest livelihood provider in India and also the biggest contributor in the Gross Domestic Product (GDP). Over 58% of rural population is dependent on agriculture as their principal mean of occupation.

Growth in household income and consumption, expansion of food processing sector and increase in agricultural export are some of the major factors that have worked together to facilitate growth in the sector in India. Escalating private contribution in Indian agriculture, growing organic farming and use of information technology are some of the key developments in the agriculture industry.

As per estimates by the Central Statistics Office (CSO), the share of agriculture and allied sectors (including agriculture, livestock, forestry and fishery) was 15.35 per cent of the Gross Value Added (GVA) during 2015–16 at 2011–12 prices.

Being such a huge market base and a need of time, this is the best website for helping farmers in all their needs. This site helps them for making initial amount, equipment purchase, technology details, yield improvement technologies, etc. Also, the demand of food is never going to vanish, so the further scope and scalability is huge.

A special feature of this web site is that, it is not only indented for the farmers, specialist and public. It also provides separate portals for exporters, suppliers and investors. Both are specially optimized for flexibility and filtering for them respectively.

Chapter 2

Literature Review

An online portal aimed at connecting farmers and consumers has been a topic of interest for a while now. As a part of my literature review I went through various papers and presentations on this topic. The quick summary of my findings are specified in this chapter.

2.1 Existing System

Currently, the farmer goes to nearest market handover his product to a particular agent, agent ask the farmer to visit the market after a specific time to collect the cash earned out of the sold product. Agent sells the product to another agent or a dealer at the cost of that market.

There is no way for farmer to know about the deal and the exact amount at which their product was sold. There is no transparency. No facility to know the product price in different markets where farmers can sell their products for high profit.

Man power is required to maintain records. While preparing the details manually, it takes lot of time and effort. It is very slow process. No efficient facilities for farmer management and other activities of other users.

2.2 Proposed System

The website is a one point solution for helping farmers. Farmers can avail capital assistance from the outset by requesting an amount from interested investors. Then they can purchase seeds online. The equipment for farming can be purchased from the same website. Also the pesticides for cultivation can be purchased from the site. The farmers can ask doubts from agriculture

experts so that they can take better care of their crops and improve their yield.

Suppliers can add the product details like price, description, quantity to the site. They can view the bookings of their products. Once the harvest starts, farmers can sell it online. Thus the site provides the farmer a complete solution.

Through a website like this, farmers can connect directly with customers and make their transactions profitable for them without a middleman/third party. It Stores the information about all users and Registered users can view and modify their profile. Provides instant access to information.

Other important advantages of proposed system are Quick and easy registration for the user. Online data submission is secure. Real-time reports. Stores the information about all users. Stores users can view and modify their profile. With the help of proposed system the details of each user and cost details can be viewed by the admin. Adding new product categories. Faster and better management of the system. Accurate, reliable and more user-friendly. Instant access of information and provides validation and security.

Chapter 3

Requirement Analysis

3.1 Purpose

Farmers Help is a web application intended to manage different farming requirements. Requirement Specification is the part of the project which gives the details about the hardware and software requirements of our project. It also details the features of the programming language used.

3.1.1 Hardware Requirements

- Processor : Intel Core i3
- Memory : 4 GB RAM

3.1.2 Software Requirements

- Operating System : Linux/Windows
- Platform : Python
- Web Technologies : HTML,CSS, JavaScript
- Framework : DJango
- Web server: Apache
- Database : MySQL

- IDE : Visual Studio Code

3.2 Functional Requirements

The functional requirements includes all the activities or processes that should be achieved by the proposed system. Functional requirements are defined as a set of attributes that describe external system output behavior that are unambiguous, non-redundant, non-contradictory. This section usually consists of a hierarchical organization of requirements, with the business/functional requirements at the highest level and the detailed system requirements listed as their child items. Generally, the requirements are written as statements such as "System needs the ability to do x" with supporting detail and information included as necessary.

3.3 Non Functional Requirements

3.3.1 Performance Requirements

- Accuracy : Accuracy in functioning and the nature of user-friendly should be maintained by the system.
- Speed : The system must be capable of offering speed.
- Low cost: This system is very cheap to implement and is also user-friendly.
- Less Time consuming: It uses very less time comparing to the existing system .
- User Friendly: This proposed system is highly user friendly they enables to create a good environment.

3.3.2 Quality Requirements

- Scalability : The software will meet all of the functional requirements.
- Maintainability : The system should be maintainable. It should keep backups to atone for system failures, and should log its activities periodically.

- Reliability : The acceptable threshold for down-time should be large as possible. i.e. mean time between failures should be large as possible. And if the system is broken, time required to get the system backup again should be minimum.
- Availability: This system is easily available as the core equipment in building the software is easily obtained.
- High- Functionality: This system is highly functional in all environment since, They are highly adaptable.

Chapter 4

Feasibility Study

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its work ability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

1. What are the user's demonstrable needs and how does a candidate system meet them?
2. What resources are available for given candidate system?
3. What are the likely impacts of the candidate system on the organization?
4. Whether it is worth to solve the problem?

During feasibility analysis for this project, following primary areas of interest are to be considered.

Steps in feasibility analysis: Eight steps involved in the feasibility analysis are:

- Form a project team and appoint a project leader.
- Prepare system flowcharts.
- Enumerate potential proposed system.
- Define and identify characteristics of proposed system.
- Determine and evaluate performance and cost effective of each proposed system.
- Weight system performance and cost data.
- Select the best-proposed system.

- Prepare and report final project directive to management.

4.1 Technical Feasibility

A study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may include: Front-end and back-end selection. An important issue for the development of a project is the selection of suitable front-end and back-end. When I decided to develop the project I went through an extensive study to determine the most suitable platform that suits the development of the project. The aspects of my study included the following factors.

- Front-end selection: It must have a GUI that assists employees that are not from IT background, Scalability and extensibility., Flexibility, Robustness, Platform, independent, Easy to debug and maintain, Event driven programming facility.
- Back-end Selection: Multiple user support, Efficient data handling, Provide inherent features for security, Efficient data retrieval and maintenance, Easy to implant with the Front-end.

According to above stated features I selected MySQL as the back-end. The technical feasibility is frequently the most difficult task encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing system and to what extent it can support the proposed system.

4.2 Economical feasibility

Economic justification is generally the Bottom Line consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this, the cost and the benefits associated with the proposed system are weighed and if it suits the basic purpose of the project, the project is moved to the analysis and design phase. The financial and the economic questions during the preliminary investigation are verified to estimate the following:

- The cost to conduct a full system investigation.

- The cost of hardware and software for the class of application being considered.
- The proposed system will give the minute information, as a result the performance is improved
- This feasibility checks whether the system can be developed with the available funds.

4.3 Operational Feasibility

It is mainly related to human organizations and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organization structures are disturbed?
- What new skills will be required?
- Do the existing staff members have these skills?
- If not, can they be trained in due course of time?

The system is operationally feasible as it very easy for the End users to operate it. It only needs basic information about Windows platform.

4.4 Schedule Feasibility

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development .

Chapter 5

Design And Implementation

5.1 Users

Seven types of users are there.

- Admin
- Farmer
- Exporter
- Supplier
- Specialist
- Investor
- Public

5.2 Module Design

5.2.1 Registration

In order to access the services of the site, each user must register with the system by providing their personal details and other required information. Once the user is successfully registered, he or she can login to the site for availing of its services

5.2.2 Admin

The administrator is the authority who can validate those Farmers, Exporters, Suppliers, Public, Investor and Specialist who register on the site. The administrator interface consists of a username and password using which the admin can login to his portal.

- Approve Users: Admin is responsible for approving the users. Only after the admin approval user can access the site.
- Add category: The admin will add the categories of products like seeds, equipment and pesticides

5.2.3 Farmer

The farmer is one of the main users of the system.

- Add Crops
- Purchase Products
- Add and view Farming Techniques
- Chat With Specialist
- Request Capital

5.2.4 Exporter

Exporters are those who buy goods from the farmer. They can order agricultural products as per their requirement.

- View products
- Add order
- Add Feedback

5.2.5 Supplier

The suppliers are those who sell seeds, equipment, pesticides etc to the farmer.

- Add Product
- View Booking
- Add Feedback

5.2.6 Investor

- Invest Money: People who want to invest in the agricultural sector can register so that they can provide farmers with capital assistance

5.2.7 Specialist

They are agricultural experts. They Can

- View Queries
- Respond to Queries

5.2.8 Public

In addition to exporters who can buy goods in bulk, the public can also buy crops from farmers.

- View and Purchase Products
- View Farming Techniques
- Add Feedback

5.3 System Design

System Design's main aim is to identify the modules that should be in the system, the specifications of these modules and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats and the major modules in

the system and their specifications are decided. The most creative and challenging phase of the system development process is design phase, it is a solution, a “how to” approach to the creation of the proposed system Design, the first step in the development of an engineered product is initiated only after a clear exposition of expected product functions becomes available. Based on the user requirements and the detailed analysis of a new system, the new system must be designed. This is the phase of system designing. Normally the design proceeds in two stages: preliminary or general design and structure or detailed design.

- Preliminary or general design: In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and benefits to be derived are estimated. If the project is still considered to be feasible, we move to detailed design stage.
- Structure or detailed design: In the detailed design stage, computer oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structured design is a blue print of a computer system solution to a given problem having the same components and inter- relationship among the same components as the original problem. Input, output and processing specifications are drawn up in detail. In the design stage, the programming language and the platform in which the new system will run are also decided. There are several tools and techniques used for designing.

5.4 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?

- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

5.5 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysing and designing computer output, they should identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

5.5.1 Data Flow Diagram

DFD is one of the graphical representation techniques used in a project to show the flow of the data through a project. DFD helps us to obtain an idea about the input, output, and process involved. The things absent in a DFD are control flow, decision rules, and loops. It can be described as a representation of functions, processes that capture, manipulate, store, and distribute data between a system and the surrounding and between the components of the system. The visual representation helps for good communication.

It shows the journey of the data and how will it be stored in the last. It does not provide details about the process timings or if the process shall have a parallel or sequential operation. It

is very different from a traditional flow chart or a UML that shows the control flow or the data flow.

The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process box, which plays vital role in studying the current system. In level 0 the basic data flow of the application is showcased. It does not show the flow of data much deeper. It will be evaluated in the higher levels of Data Flow Diagram.

The process in the context level diagram is exploded into other process at the first level DFD. The Data Flow Diagram of 'FARMERS HELP' is shown below.

CONTEXT DIAGRAM

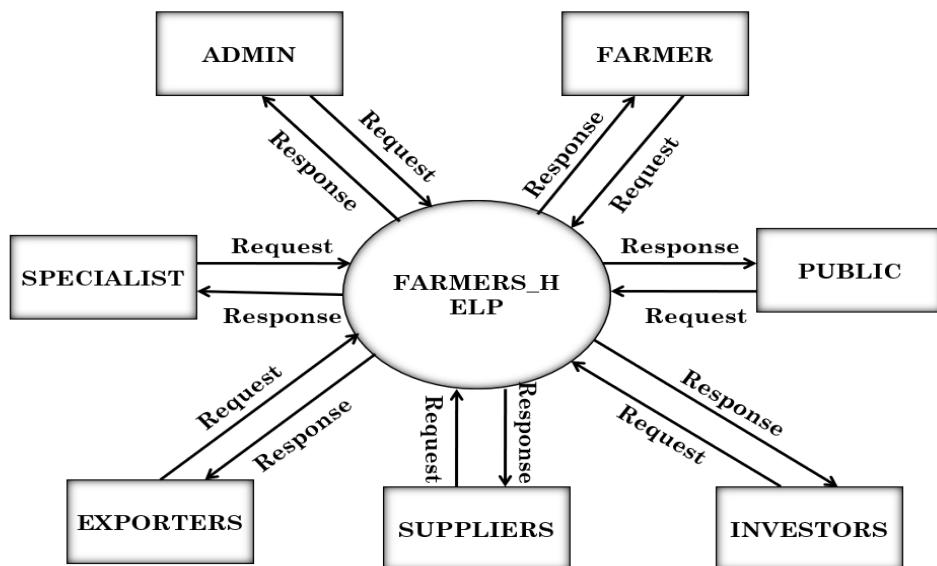


Figure 5.1: Level 0 DFD

LEVEL 1

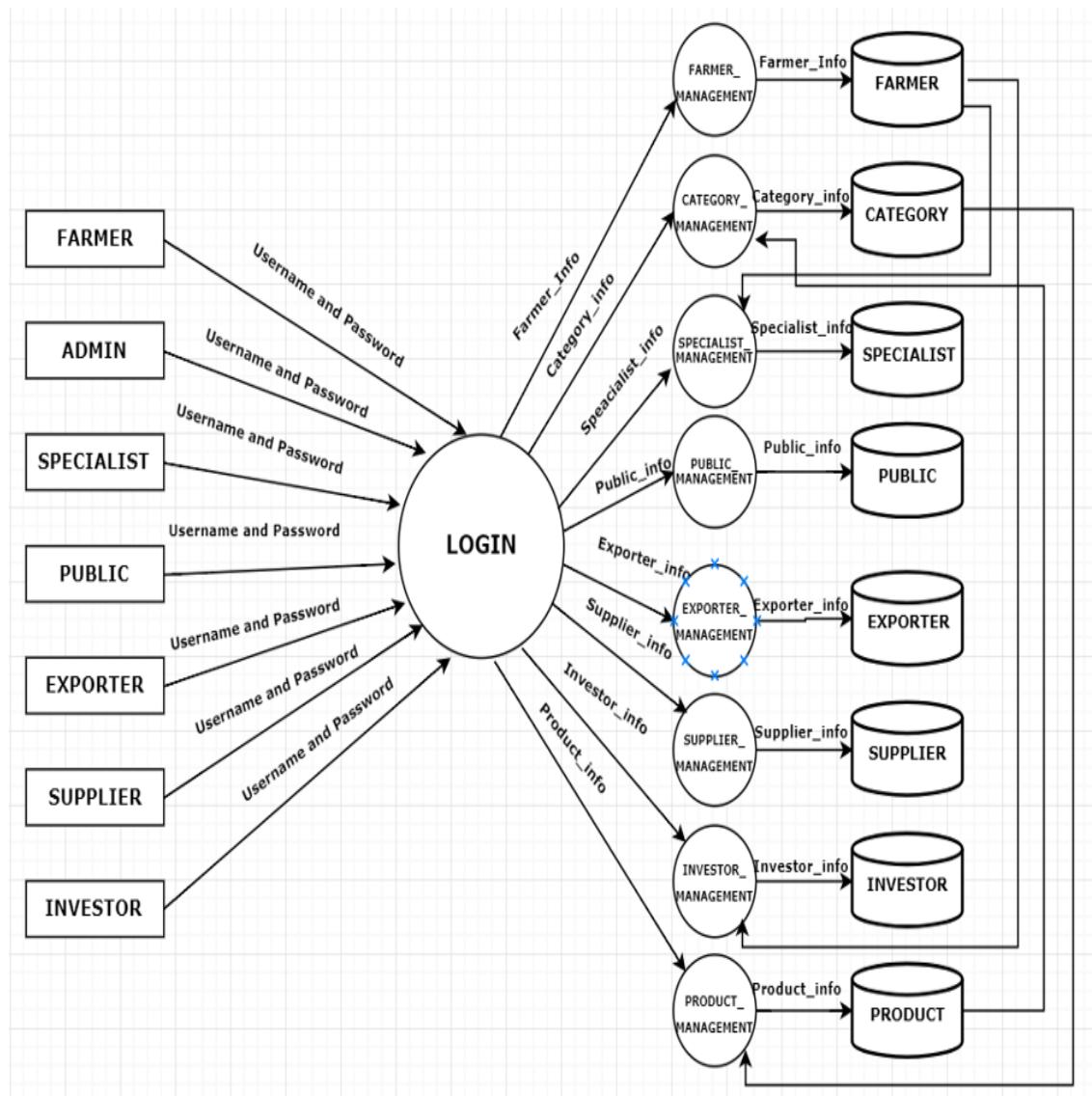


Figure 5.2: Level 1 DFD

LEVEL 2

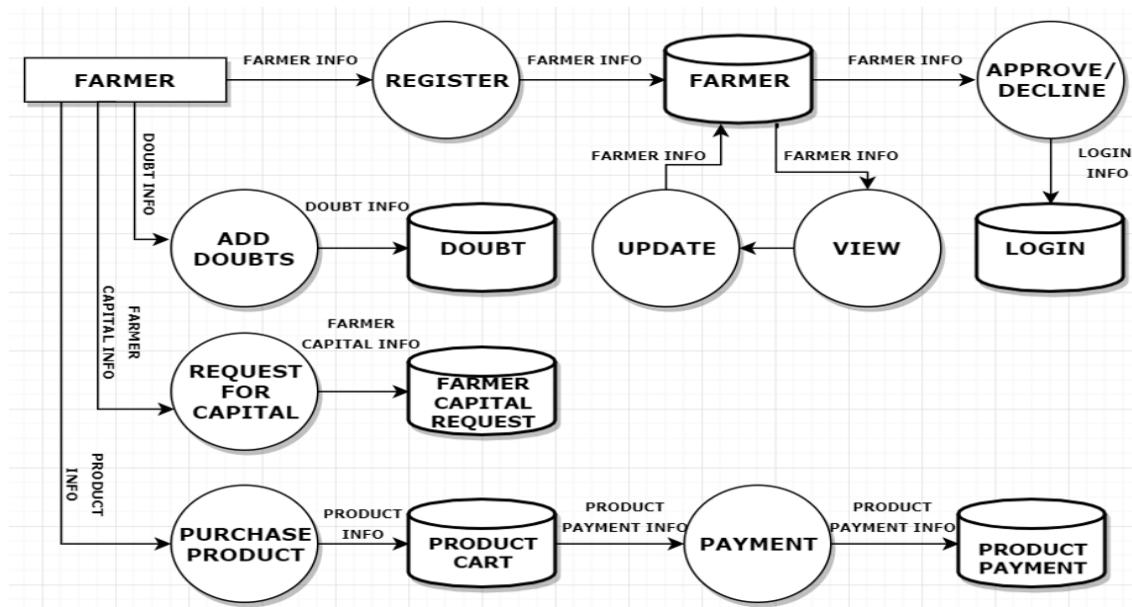


Figure 5.3: Level 2 Farmer Management

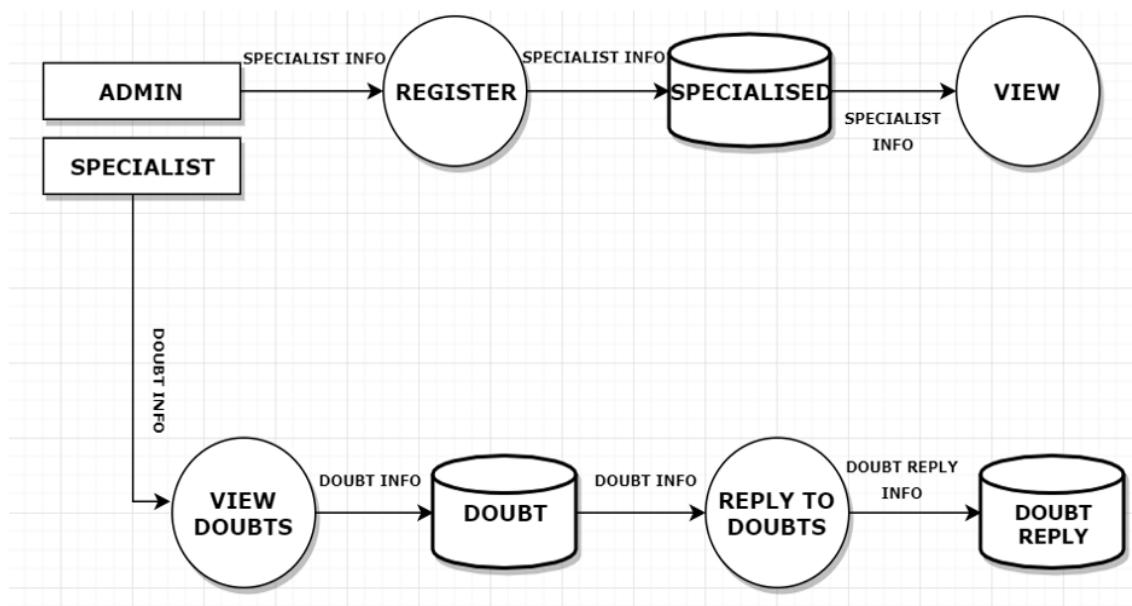


Figure 5.4: Level 2 Specialist Management

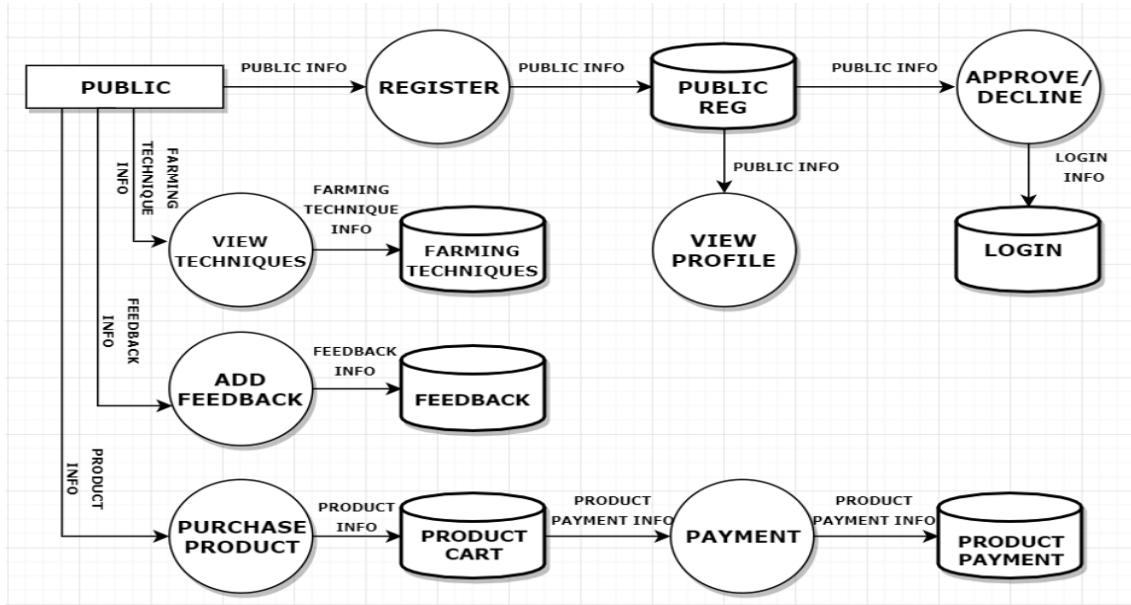


Figure 5.5: Level 2 Public Management

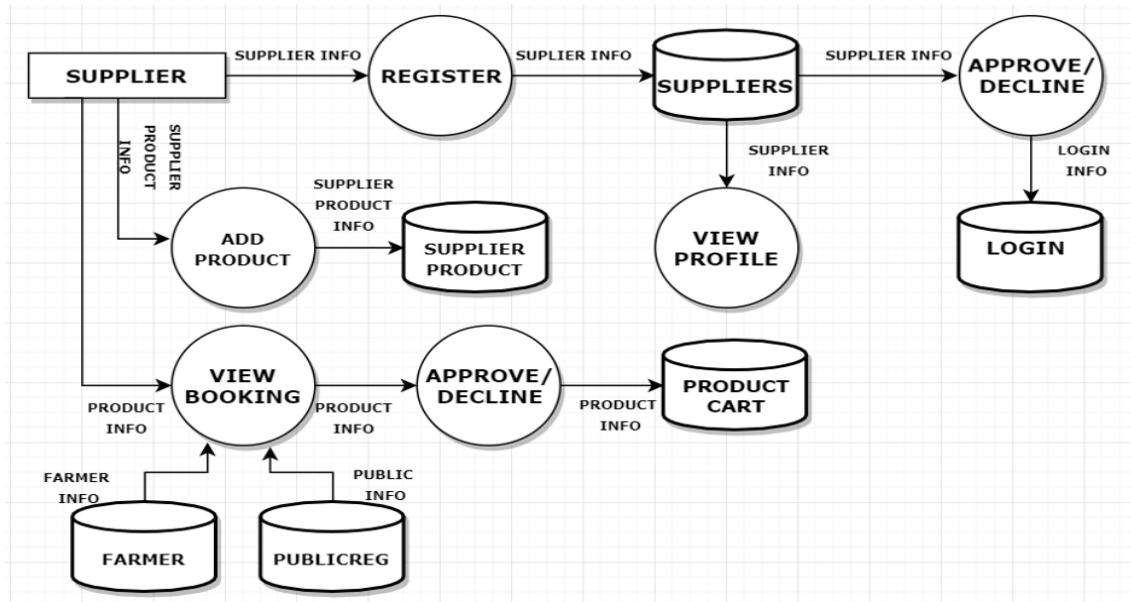


Figure 5.6: Level 2 Supplier Management

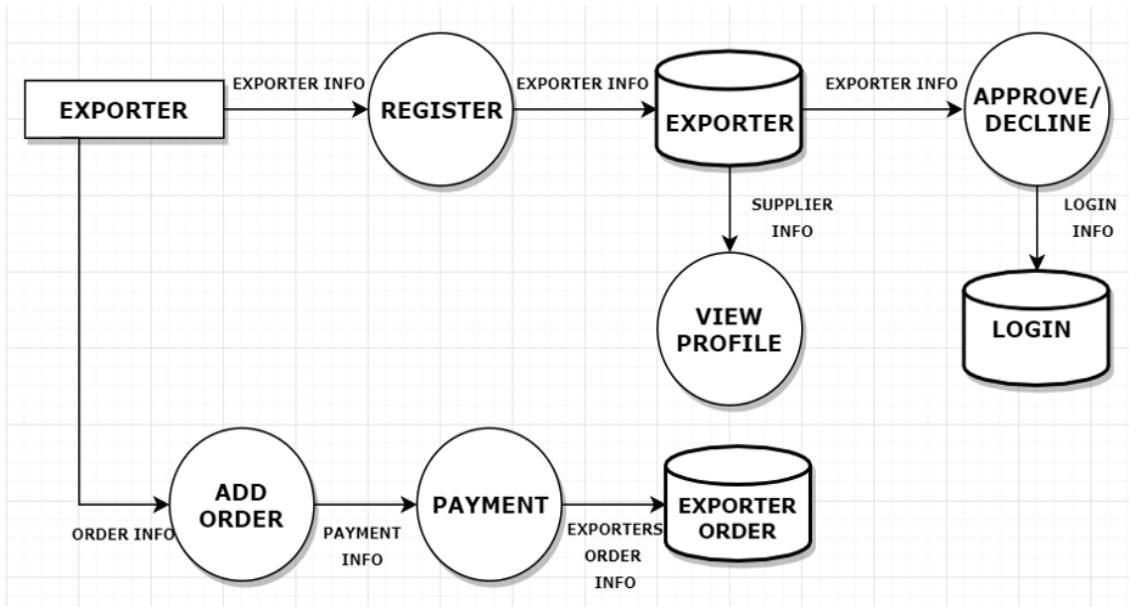


Figure 5.7: Level 2 Exporters Management

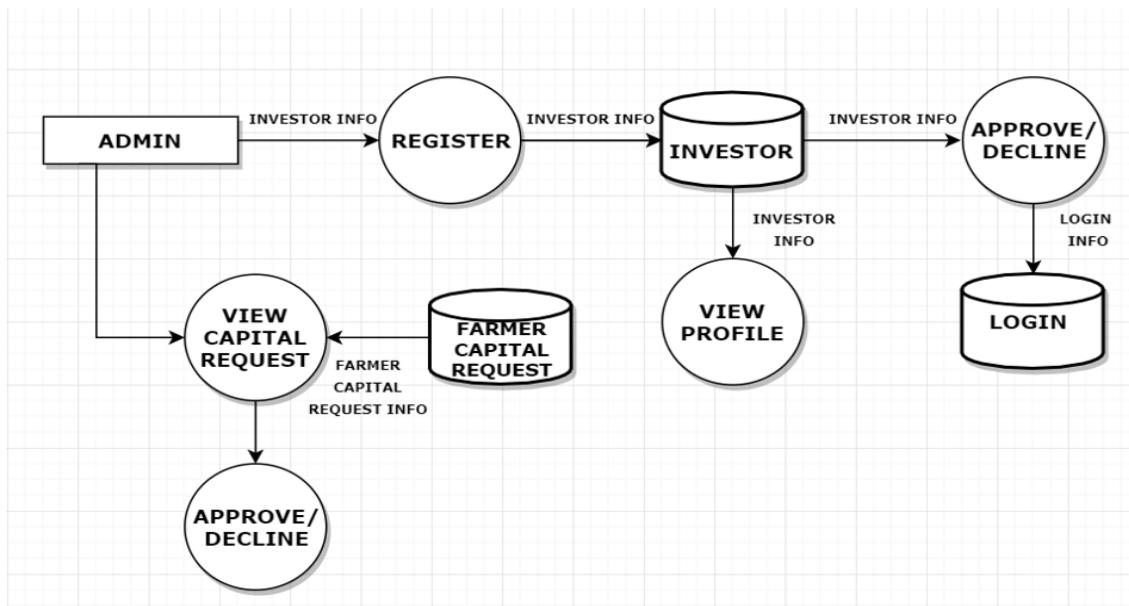


Figure 5.8: Level 2 Investor Management

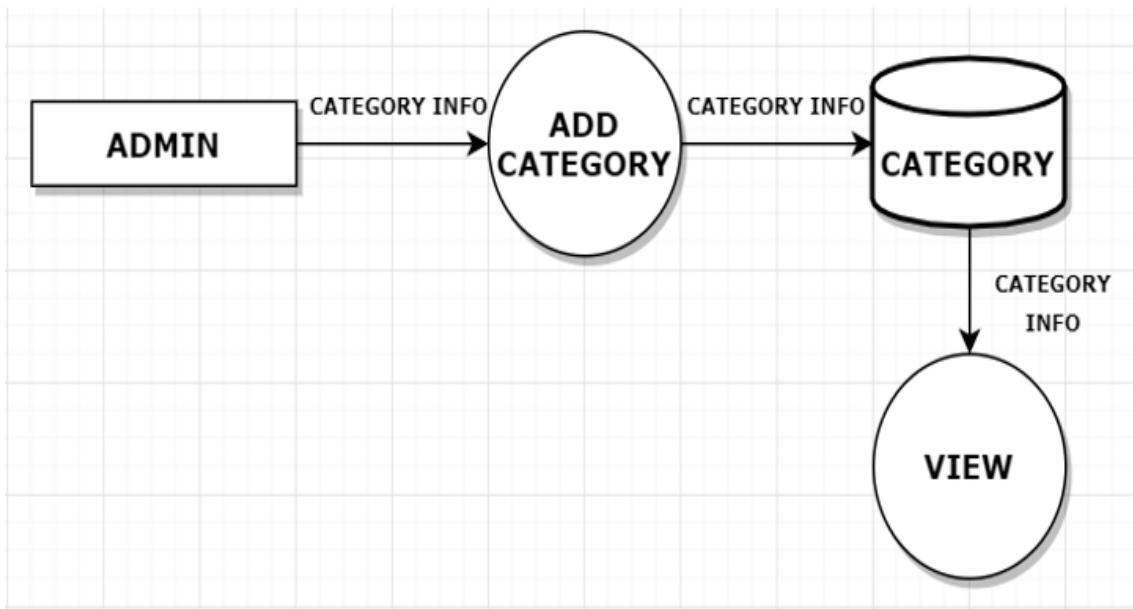


Figure 5.9: Level 2 Category Management

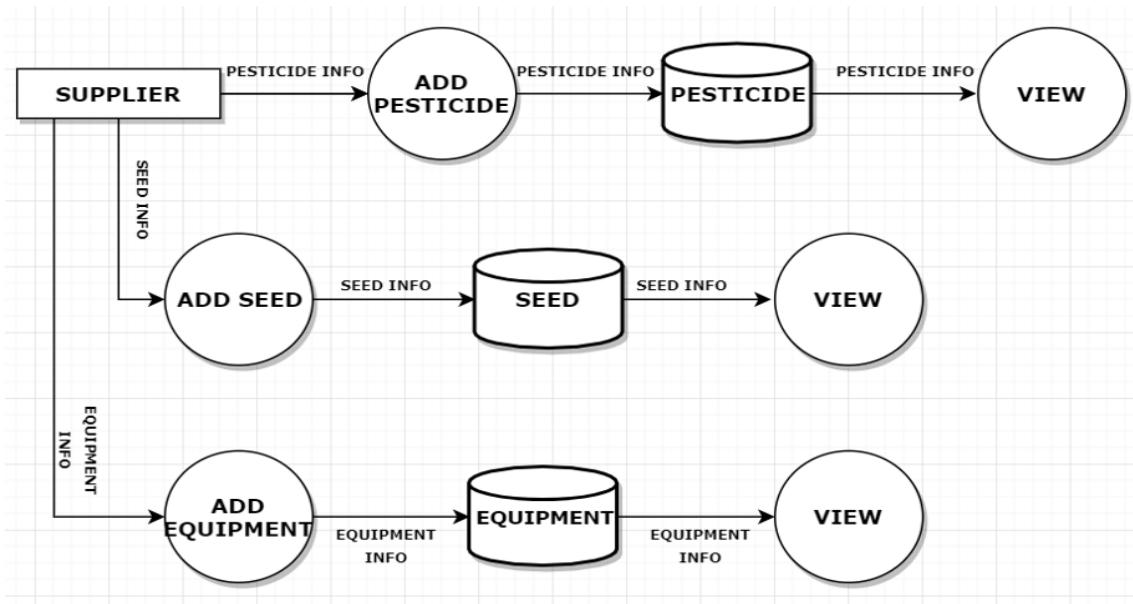


Figure 5.10: Level 2 Product Management

5.5.2 ER Diagram

ER Diagram (Entity relationship diagram) is a visual representation of data that describes data is related to each other. ER Diagrams are used to represent relationship between entity sets.

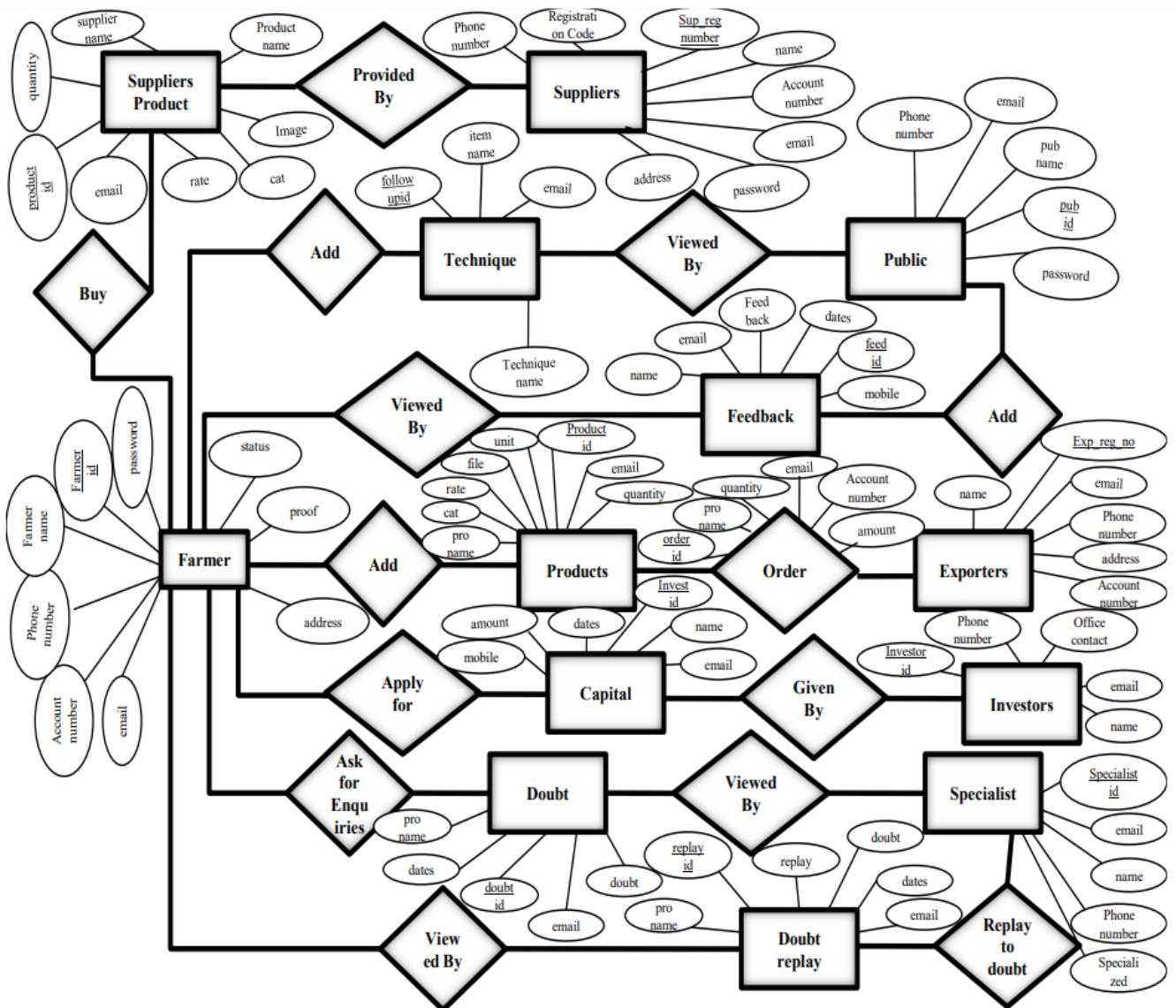


Figure 5.11: ER Diagram

5.5.3 Database Design

A database system is an overall collection of different database software components and database containing the parts viz. Database application programs, front-end components, Database Management Systems, and Databases.

- **Table Name : Login**
To store login details

Field Name	Field Size	Description	Constraints
Email	Varchar(50)	Email id (username)	Primary Key
password	Varchar(50)	Password	
usertype	Varchar(50)	User type	
status	Varchar(50)	Status	

Figure 5.12: Login Table

- Table Name : farmerreg**
To store farmer registration details

Field Name	Field Size	Description	Constraints
farmerid	Int	Farmer id	Primary Key
FarmersName	Varchar(50)	Farmer name	
HouseName	Varchar(50)	House Name	
State	Varchar(50)	State	
District	Varchar(20)	District	
panchayath	Varchar(20)	Panchayath	
Postoffice	Varchar(50)	Postoffice	
PostalZipCode	Varchar(30)	PostalZipCode	
phonenumber	Bigint(10)	Phone number	
proof	Varchar(100)	Image	
status	Varchar(50)	Status(Approved or not)	
AccountNumber	Bigint(20)	Account Number	
NameoftheBank	Varchar(30)	Name of the Bank	
IFSC	Varchar(20)	IFSC	

Figure 5.13: Farmerreg Table

-
- Table Name : Exporters**
To store exporters' registration details

Field Name	Field Size	Description	Constraints
Exp_Regnumber	Int	Exporters Id	Primary key
ExpotersName	Varchar(50)	Expoters Name	
RegistrationCode	Varchar(50)	Registration Code	
Country	Varchar(50)	Country	
State	Varchar(50)	State	
District	Varchar(50)	District	
PhoneNumber	Bigint(10)	Phone number	
Dates	Varchar(50)	Dates	
File	Varchar(50)	Image	
AccountNumber	Varchar(50)	Account Number	
BankName	Varchar(50)	Bank Name	
IFSC	Varchar(50)	IFSC	
status	Varchar(100)	Status(Approved or not)	

Figure 5.14: Exporters Table

- **Table Name : Specialist**
To store specialist registration details

Field Name	Field Size	Description	Constraints
Specializedid	Int(100)	Specialist id	Primary Key
Name	Varchar(50)	Specialist name	
MobileNumber	Bigint(10)	Mobile Number	
OfficeName	Varchar(50)	Office Name	
OfficeContact	Bigint(10)	Office contact number	
Designation	Varchar(50)	Designation	
Specialized	Varchar(100)	Specialization	
Message	Varchar(500)	Message	

Figure 5.15: Specialist Table

- **Table Name : Supplier**
To store supplier registration details

Field Name	Field Size	Description	Constraints
Sup_regnumber	Int(50)	Supplier registration number	Primary Key
suppliersName	Varchar(50)	Supplier name	
RegistrationCode	Varchar(50)	RegistrationCode	
Category	Varchar(100)	Category	
District	Varchar(50)	District	
PhoneNumber	Bigint(10)	Phone Number	
File	Varchar(50)	Image	
AccountNumber	Varchar(20)	AccountNumber	
BankName	Varchar(50)	BankName	
IFSC	Varchar(20)	IFSC	
Status	Varchar(50)	Status(Approved or not)	

Figure 5.16: Supplier Table

- **Table Name : Investors**
To store investors' registration details

Field Name	Field Size	Description	Constraints
Investorid	Int(100)	Investors id	Primary Key
Name	Varchar(50)	Investors name	
MobileNumber	Bigint(10)	Mobile Number	
OfficeName	Varchar(50)	Office Name	
OfficeContact	Bigint(10)	Office contact number	
Designation	Varchar(50)	Designation	
Message	Varchar(500)	Message	

Figure 5.17: Investors Table

- **Table Name : Seed**
To store seed details

Field Name	Field Size	Description	Constraints
Seedid	Int(100)	Seed id	Primary key
Name	Varchar(50)	Seed name	
File1	Varchar(50)	Image	
scientificname	Varchar(50)	Scientific Name	
verity	Varchar(50)	Seed amount	
Rate	Int(10)	Rate	
Quantity	Int(10)	Quantity	
Description	Varchar(50)	Seed description	
Unit	Varchar(50)	Seed unit	
Email	Varchar(50)	Email	Foreign Key

Figure 5.18: Seed Table

- **Table Name : Seedcart**
To store seedcart details

Field Name	Field Size	Description	Constraints
cartno	Int(100)	Cart Number	Primary key
seedid	Varchar(50)	Seed id	Foreign Key
dates	date	Date	
email	Varchar(50)	Email	Foreign Key

Figure 5.19: Seedcart Table

- **Table Name : Seedpayment**
To store seedpayment details

Field Name	Field Size	Description	Constraints
paymentid	Int(100)	Payment id	Primary Key
Cardnumber	Varchar(50)	Card Number	
cartno	Int(100)	Cart Number	Foreign Key
cvv	Varchar(50)	CVV	
Accountnumber	Varchar(50)	Account Number	
Ifsc	Varchar(50)	IFSC	
Amount	Bigint(50)	Amount	
Dates	Date	Date	
User	Varchar(50)	User	

Figure 5.20: Seedpayment Table

- **Table Name : Product**
To store product details

Field Name	Field Size	Description	Constraints
Productid	Int(100)	Product id	Primary key
productname	Varchar(50)	Product name	
categoryid	Varchar(50)	Category id	Foreign Key
file	Varchar(50)	Image	
Rate	Bigint (10)	Rate	
Quantity	Bigint(10)	Quantity	
Unit	Varchar(50)	Product unit	
Email	Varchar(50)	Email	Foreign Key

Figure 5.21: Product Table

- **Table Name : productcart**
To store productcart details

Field Name	Field Size	Description	Constraints
cartno	Int(100)	Cart Number	Primary key
productid	Varchar(50)	Product id	Foreign Key
dates	date	Date	Not null
email	Varchar(50)	Email	Foreign Key

Figure 5.22: Productcart Table

- Table Name : Productpayment**

To store product payment details

Field Name	Field Size	Description	Constraints
paymentid	Int(100)	Payment id	Primary Key
Cardnumber	Varchar(50)	Card Number	
cartno	Int(100)	Cart Number	Foreign Key
cvv	Varchar(50)	CVV	
Accountnumber	Varchar(50)	Account Number	
Ifsc	Varchar(50)	IFSC	
Amount	Bigint(50)	Amount	
Dates	Date	Date	
User	Varchar(50)	User	

Figure 5.23: Productpayment Table

- Table Name : Equipmentcart**

To store equipment cart details

Field Name	Field Size	Description	Constraints
cartno	Int(100)	Cart Number	Primary key
productid	Varchar(50)	Product id	Foreign Key
dates	date	Date	
email	Varchar(50)	Email	Foreign Key
status	Varchar(50)	Status (Approved or not)	

Figure 5.24: Equipmentcart Table

- **Table Name : Equipmentpayment**
To store equipment payment details

Field Name	Field Size	Description	Constraints
paymentid	Int(100)	Payment id	Primary Key
Cardnumber	Varchar(50)	Card Number	
cartno	Int(100)	Cart Number	Foreign Key
cvv	Varchar(50)	CVV	
Accountnumber	Varchar(50)	Account Number	
Ifsc	Varchar(50)	IFSC	
Amount	Bigint(50)	Amount	
Dates	Date	Date	
User	Varchar(50)	User	

Figure 5.25: Equipmentpayment Table

- **Table Name : Pesticidecart**
To store pesticide cart details

Field Name	Field Size	Description	Constraints
cartno	Int(100)	Cart Number	Primary key
productid	Varchar(50)	Product id	Foreign Key
dates	date	Date	
email	Varchar(50)	Email	Foreign Key
status	Varchar(50)	Status (Approved or not)	

Figure 5.26: Pesticidecart Table

- **Table Name : Pesticidepayment**

To store pesticide payment details

Field Name	Field Size	Description	Constraints
paymentid	Int(100)	Payment id	Primary Key
Cardnumber	Varchar(50)	Card Number	
cartno	Int(100)	Cart Number	Foreign Key
cvv	Varchar(50)	CVV	
Accountnumber	Varchar(50)	Account Number	
Ifsc	Varchar(50)	IFSC	
Amount	Bigint(50)	Amount	
Dates	Date	Date	
User	Varchar(50)	User	

Figure 5.27: Pesticidepayment Table

- **Table Name : category**

To store category details

Field Name	Field Size	Description	Constraints
categoryid	Int(100)	Category id	Primary Key
categoryname	Varchar(50)	Category Name	

Figure 5.28: Category Table

- **Table Name : publicreg**

To store public details

Field Name	Field Size	Description	Constraints
pubid	Int(100)	Public id	Primary Key
publicname	Varchar(50)	Public Name	
phonenumber	Bigint(10)	Phone Number	
password	Varchar(50)	Password	

Figure 5.29: Publicreg Table

- **Table Name : suppliersproduct**

To store suppliers product details

Field Name	Field Size	Description	Constraints
productid	Int(100)	Product id	Primary key
Sup_regnumber	Varchar(50)	Supplier id	Foreign Key
categoryid	Varchar(50)	Category	Foreign Key
productname	Varchar(50)	Product Name	
file	Varchar(50)	Image	
Rate	Int(10)	Rate	
Quantity	Int(10)	Quantity	
Unit	Varchar(50)	product unit	
Email	Varchar(50)	Email	Foreign Key

Figure 5.30: Suppliersproduct Table

- **Table Name : doubt**
To store doubt details

Field Name	Field Size	Description	Constraints
doubtid	Int(100)	Doubt id	Primary key
productname	Varchar(50)	Product name	
dates	date	Date	
email	Varchar(50)	Email	Foreign Key
doubt	Varchar(100)	Doubt	

Figure 5.31: Doubt Table

- **Table Name : doubtreply**
To store doubt replay details

Field Name	Field Size	Description	Constraints
replayid	Int(100)	Repaly id	Primary key
dates	date	Replay Date	
doubtid	Varchar(100)	Doubt	Foreign Key
replay	Varchar(100)	Replay to the doubt	

Figure 5.32: Doubtreply Table

- Table Name : exportersorder**
To store exporters order details

Field Name	Field Size	Description	Constraints
orderid	Int(100)	Order id	Primary Key
productname	Varchar(50)	Product name	
quantity	BigInt(100)	Quantity	
email	Varchar(50)	Email	Foreign Key
Date	Date	Date	
Status	Varchar(50)	Status	
cardnumber	Varchar(50)	Card number	
cvv	Varchar(50)	CVV	
accountnumber	Varchar(50)	Account number	
ifsc	Varchar (50)	IFSC	
amount	Bigint(100)	Amount	

Figure 5.33: Exportersorder Table

- Table Name : farmercapitalrequest**
To store farmer capital request details

Field Name	Field Size	Description	Constraints
investid	Int(100)	Investment id	Primary key
name	Varchar(50)	Name	
dates	date	Date	
email	Varchar(50)	Email	Foreign Key
amount	Bigint(100)	Amount Needed	
mobile	Bigint(100)	Mobile Number	

Figure 5.34: Farmercapitalrequest Table

- **Table Name : farmingtechniques**
To store farming techniques details

Field Name	Field Size	Description	Constraints
followupid	Int(100)	Followup id	Primary key
itemname	Varchar (100)	Item name	
techniquename	Varchar (100)	Technique name	
followups	Varchar (100)	Details of technique	
email	Varchar (100)	Email	Foreign Key

Figure 5.35: Farmingtechniques Table

- **Table Name : feedback**
To store feedback details

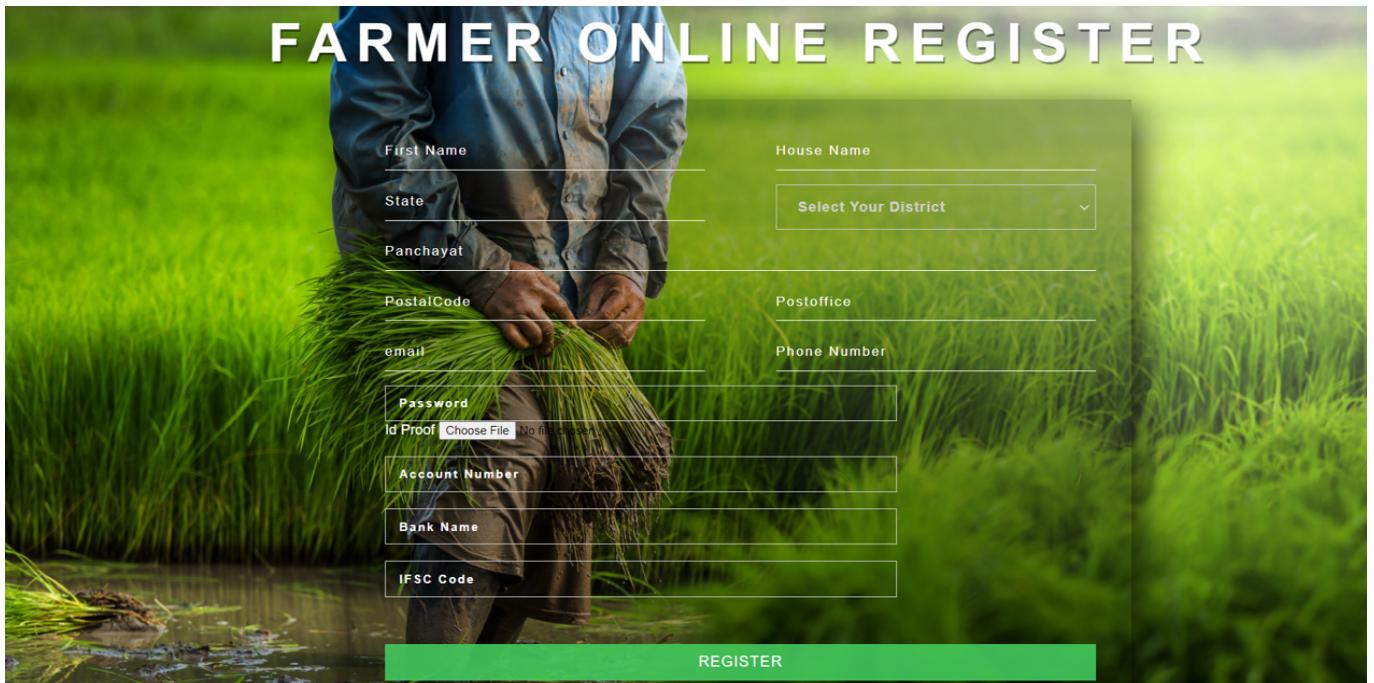
Field Name	Field Size	Description	Constraints
feedid	Int(100)	Feedback id	Primary key
name	Varchar(50)	Name	
dates	date	Date	
email	Varchar(50)	Email	Foreign Key
feedback	Varchar(500)	Feedback	
mobile	Bigint(100)	Mobile Number	

Figure 5.36: Feedback Table

5.6 Screenshots

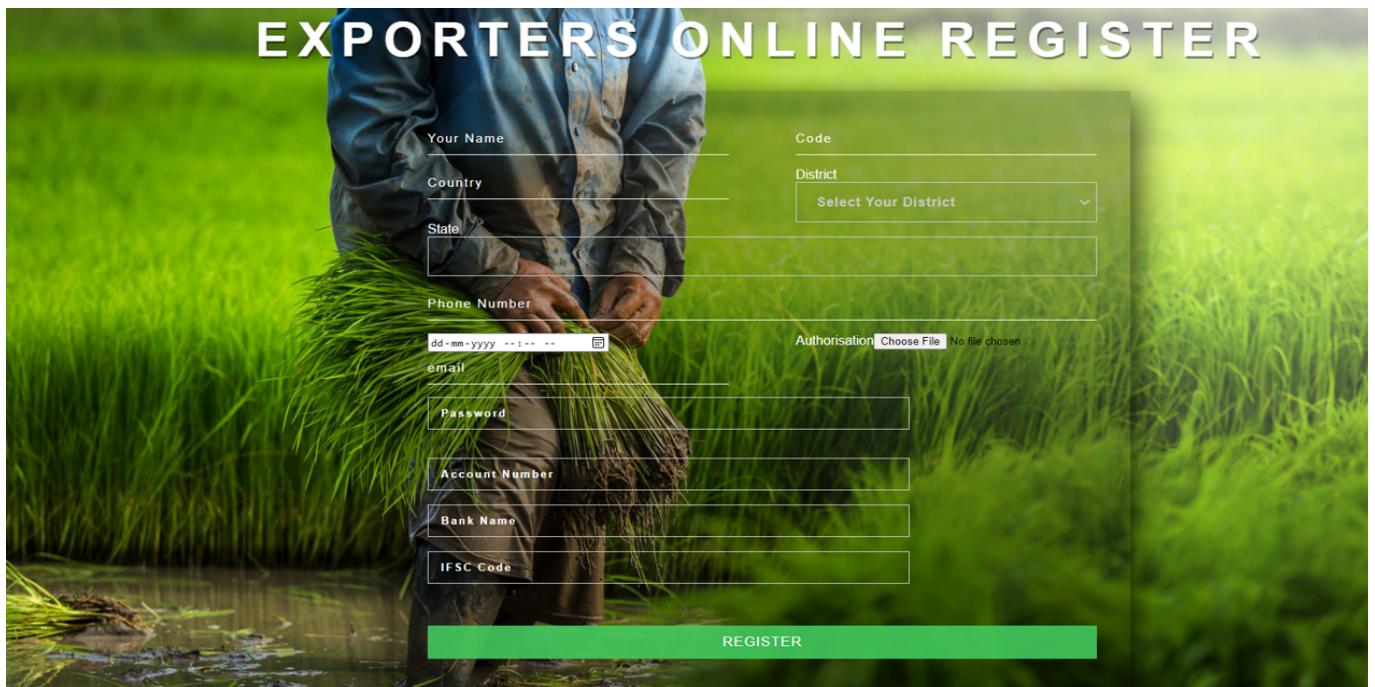


Figure 5.37: Login Page



The registration form is titled "FARMER ONLINE REGISTER" at the top center. It features a background photograph of a farmer in a field. The form consists of several input fields arranged in two columns. The left column includes fields for "First Name", "State", "Panchayat", "PostalCode", "email", "Password", "Id Proof" (with a "Choose File" button), "Account Number", "Bank Name", and "IFSC Code". The right column includes fields for "House Name", "Select Your District" (a dropdown menu), "Postoffice", and "Phone Number". A green "REGISTER" button is located at the bottom of the form.

Figure 5.38: Farmer Registration



EXPORTERS ONLINE REGISTER

Your Name _____

Country _____

State _____

Phone Number _____

dd-mm-yyyy - - : - -

Authorisation No file chosen

email _____

Password _____

Account Number _____

Bank Name _____

IFSC Code _____

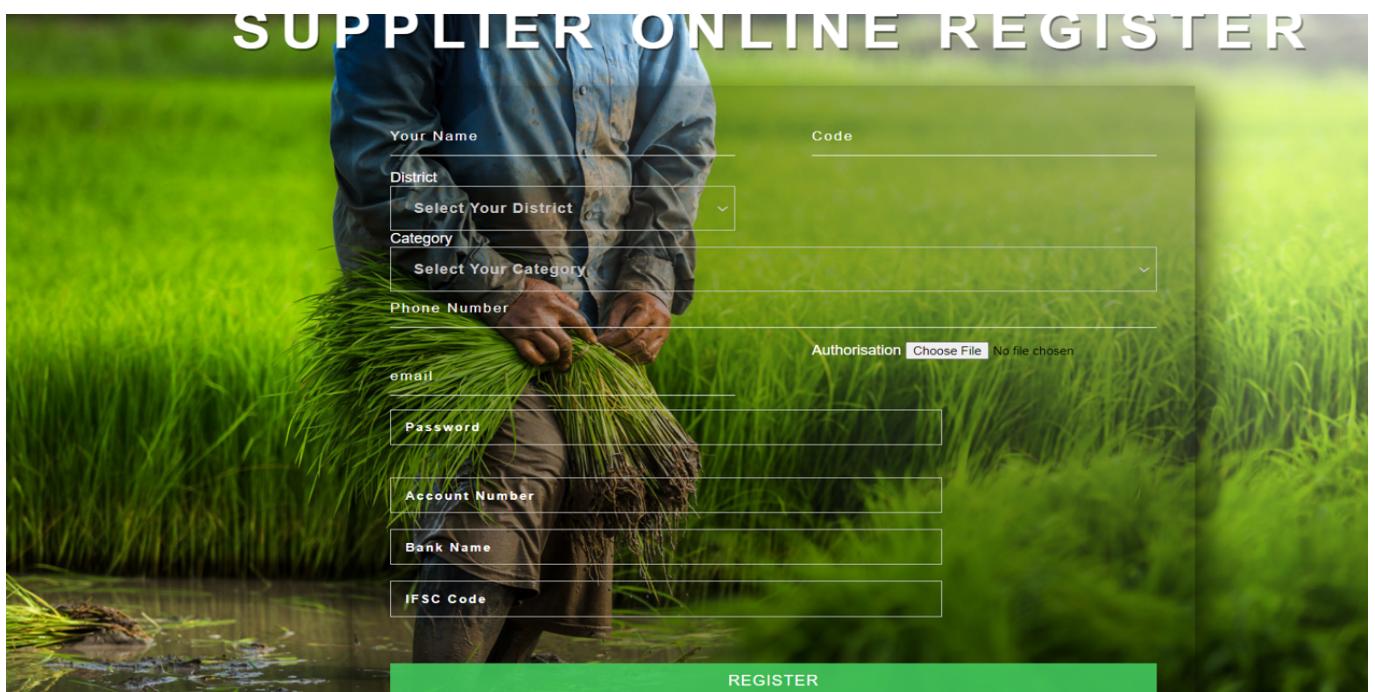
Code _____

District

REGISTER

This figure shows the 'EXPORTERS ONLINE REGISTER' form. It features a background image of a person working in a lush green rice field. The form includes fields for personal information like name, country, state, phone number, date of birth, email, password, account number, bank name, and IFSC code. It also includes fields for a code and district, with dropdown menus for selecting them. A large green 'REGISTER' button at the bottom is partially obscured by the background image.

Figure 5.39: Exporter Registration



SUPPLIER ONLINE REGISTER

Your Name _____

District

Category

Phone Number _____

Authorisation No file chosen

email _____

Password _____

Account Number _____

Bank Name _____

IFSC Code _____

Code _____

REGISTER

This figure shows the 'SUPPLIER ONLINE REGISTER' form. It has a similar structure to the exporter registration form, featuring a background image of a person in a rice field. It includes fields for name, district, category, phone number, email, password, account number, bank name, and IFSC code. It also includes fields for a code and district, with dropdown menus for selection. A large green 'REGISTER' button is at the bottom.

Figure 5.40: Supplier Registration



Figure 5.41: Public Registration

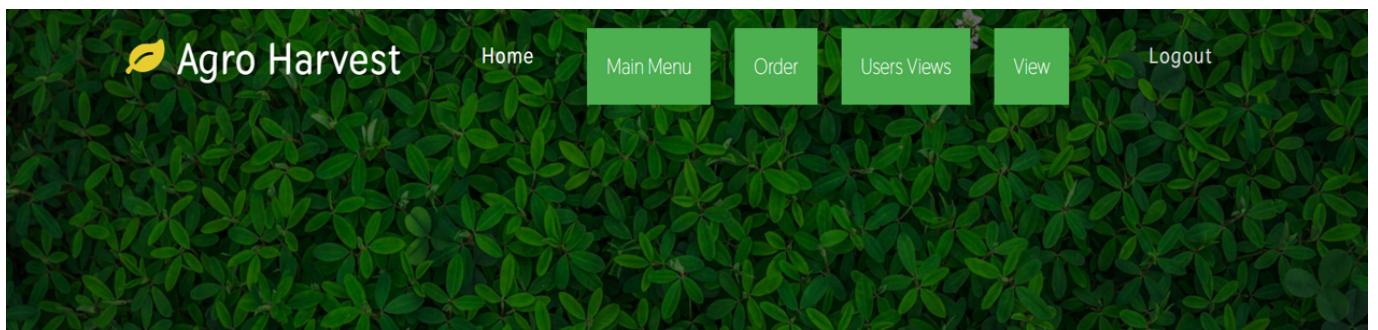
A screenshot of an investor registration form. The form includes the following fields: "investor Name", "Email", "Mobile Number", "Office Name", "Office Contact", "Designation", "amount upto", and a large text area for "About Me". A "SUBMIT" button is located at the bottom right of the form.

Figure 5.42: Investor Registration

Name	Febi
House Name	Vazhattu
State	Kerala
District	Thiruvananthapuram
Panchayath	Trivandrum
Post Office	Trivandrum
PostalZipCode	676619
Email	febi@gmail.com
Phone Number	7558810669
Photo Proof	
Password	*****
Account Number	12348978999
Name of the bank	SBI
IFSC	SBINOOO8969

UPDATE

Figure 5.43: View Profile



Approval

FarmersName	HouseName	State	District	panchayath	Postoffice	PostalZipCode	email	phonenumber	proof	AccountNumber	N
Fida	Kollumpurath	Kerala	Kollam	Kollam	Kollam	685619	fida@gmail.com	7909685678		12323628901	S



Figure 5.44: Admin-Approve Farmer



A screenshot of a web form titled "Farmer-Add Product". The form consists of several input fields and a submit button. The fields are as follows:

- A dropdown menu labeled "vegetable" containing the option "vegetable".
- A text input field labeled "Your Productname".
- A file input field labeled "Choose File" with the placeholder "No file chosen".
- A text input field labeled "unit".
- A text input field labeled "rate".
- A text input field labeled "quantity".
- A large green "SUBMIT" button at the bottom.

Figure 5.45: Farmer-Add Product



Figure 5.46: Specialist-View Doubts



Doubt Reply

Doubt	Item name	Replay	Dates
Is it harmful to apply to cardamom every month?	Cardamom	It will be okay to some extend but high usage can be harmful	2022-03-04

Figure 5.47: Farmer-View Doubt Reply

Make Payment

Your cardnumber

cvv

Accountnumber

ifsc

amount

dd-mm-yyyy

SUBMIT

Figure 5.48: Exporter-Make Payment



Figure 5.49: View Product Cart

A screenshot of the "Agro Harvest" website. The header features the logo and navigation links: "Home", "Addproduct", "Viewpesticidebooking", "Viewequipmentbooking", and "Logout". The main content area is titled "Equipment Booking" and displays a table with one row of data. The table columns are "BookingNumber", "itemCode", "Date", "email", and "Action". The data shows a booking with number 1, item code 1, date 2022-03-04, email febi@gmail.com, and action "Approve". At the bottom of the page, there is a footer with the "Agro Harvest" logo, address, phone number, email, social media links, and copyright information.

Figure 5.50: View Equipment Booking

Chapter 6

Coding

6.1 views.py

```
from django.shortcuts import render, HttpResponseRedirect
import pymysql
from django.core.files.storage import FileSystemStorage
from datetime import datetime
from datetime import date
# Create your views here.
db=pymysql.connect("localhost","root","","mycolorfarm")
c=db.cursor()

def index(request):
    if request.POST:
        Username=request.POST.get("Username")
        request.session["username"]=Username
        Password=request.POST.get("Password")
        print("select count(*),usertype,status from login where username=' " +
              Username + "' and password=' " + Password + "'")
        res=c.execute("select count(*),usertype,status from login where username=' " +
                     Username + "' and password=' " + Password + "'")

```

```

rs=c.fetchone()
db.commit()
print(rs)
if rs[0]>0 and rs[1]=="admin" and rs[2]=="Approved":
    return HttpResponseRedirect('/admin_home/')
if rs[0]>0 and rs[1]=="Public" and rs[2]=="Approved":
    return HttpResponseRedirect('/public_home/')

if rs[0]>0 and rs[1]=="farmer" and rs[2]=="Approved":
    return HttpResponseRedirect('/farmer_home/')
if rs[0]>0 and rs[1]=="Exporters" and rs[2]=="Approved":
    return HttpResponseRedirect('/exporters_home/')
if rs[0]>0 and rs[1]=="Specialist" and rs[2]=="Approved":
    return HttpResponseRedirect('/specialist_home/')
if rs[0]>0 and rs[1]=="suppliers" and rs[2]=="Approved":
    return HttpResponseRedirect('/suppliers_home/')
else:
    msg="invalid username"
return render(request,'common/common_login.html')

def admin_home(request):
    return render(request,'admin/admin_home.html')

def admin_about(request):
    return render(request,'admin/admin_about.html')

def farmer_home(request):
    return render(request,'farmer/farmer_home.html')

def exportersaddorderrequest(request):

    from datetime import datetime
    print("*"*50)
    if request.POST:
        Productname=request.POST.get("Productname")

```

```

Quantity=request.POST.get("Quantity")
email=request.session["username"]
dates=datetime.today().strftime("%Y-%m-%d")
print("insert into exportersorder(Productname,Quantity,email,dates,status)
      values ('"+Productname+"','"+Quantity+"','"+email+"','"+"
      +dates+"','pending')")
c.execute("insert into
          exportersorder(Productname,Quantity,email,dates,status) values
          ('"+Productname+"','"+Quantity+"','"+email+"','"+"
          +dates+"','pending')")
db.commit()

return render(request,'exporters/exportersaddorderrequest.html')

def exporters_home(request):
    return render(request,'exporters/exportershome.html')

def specialist_home(request):
    return render(request,'specialist/specialisthome.html')
def public_home(request):
    return render(request,'public/public_home.html')
def suppliers_home(request):
    return render(request,'suppliers/suppliershome.html')

def common_register(request):
    return render(request,'common/common_register.html')

def farmerreg(request):
    msg=""
    res=0
    qry=""
    rs=""
    if request.POST and request.FILES.get("uploads"):

```

```

FarmersName=request.POST.get("FarmersName")
HouseName=request.POST.get("HouseName")
State=request.POST.get("State")
District=request.POST.get("District")
panchayath=request.POST.get("panchayath")
Postoffice=request.POST.get("Postoffice")
PostalZipCode=request.POST.get("PostalZipCode")
email=request.POST.get("email")
PhoneNumber=request.POST.get("PhoneNumber")
password=request.POST.get("password")
status="Pending"
AccountNumber=request.POST.get("AccountNumber")
NameoftheBank=request.POST.get("NameoftheBank")
IFSC=request.POST.get("IFSC")
c.execute("select count(*) as cnt from farmerreg where email='"
+ str(email) + "' or phonenumber='"
+ str(PhoneNumber) + "'")
rs=c.fetchone()
db.commit()

if rs[0]>0 :
    msg="Already registread"
else:

    myfile=request.FILES.get("uploads")
    fs=FileSystemStorage()
    filename=fs.save(myfile.name,myfile)
    uploadurl=fs.url(filename)
    c.execute("insert into farmerreg(FarmersName,HouseName,State,District,
    panchayath,Postoffice,PostalZipCode,email,
    phonenumber,proof,password,status,AccountNumber,
    NameoftheBank,IFSC) values('"
    + str(FarmersName) +
    "','" + str(HouseName) + "','" + str(State) +
    "','" + str(District) + "','" + str(panchayath) + "','" +
    str(Postoffice) + "','" + str(PostalZipCode)

```

```

+ "','" + str(email) + "','" + str(PhoneNumber)
+ "','" + str(uploadurl) + "','" + str(password) + "','" + str(status) +
"','" + str(AccountNumber) + "','" + str(NameoftheBank) + "','" +
str(IFSC) + "')")

db.commit()

usertype="farmer"
status="pending"

c.execute("insert into login(username,password,usertype,status) values('"
+ str(email) + "','" + str(password) + "','" + str(usertype) + "','" +
str(status) + "')")

db.commit()

msg="saved"

return render(request,'common/farmerreg.html')

def Exportersreg(request):
    msg=''

    if request.POST and request.FILES.get("uploads"):

        ExportersName=request.POST.get("ExportersName")
        RegistratoinCode=request.POST.get("RegistratoinCode")
        Country=request.POST.get("Country")
        State=request.POST.get("State")
        District=request.POST.get("District")
        PhoneNumber=request.POST.get("PhoneNumber")
        dates=request.POST.get("dates")
        email=request.POST.get("email")
        Password=request.POST.get("password")
        AccountNumber=request.POST.get("AccountNumber")
        BankName=request.POST.get("BankName")
        IFSC=request.POST.get("IFSC")

        print("select count(*) as cnt from Exporters where RegistratoinCode=' " +
              str(RegistratoinCode) + "' or PhoneNumber=' " + str(PhoneNumber) + "' or
              email=' " + str(email) + "'")

        res=c.execute("select count(*) as cnt from Exporters

```

```

where RegistrationCode=' + str(RegistrationCode) + "' or PhoneNumber=' + 
str(PhoneNumber) + "' or email=' + str(email) + '')

rs=c.fetchone()

db.commit()

print(rs[0])

if rs[0]>0:

    msg="already registered"

else:

    myfile=request.FILES.get("uploads")

    fs=FileSystemStorage()

    filename=fs.save(myfile.name,myfile)

    uploadurl=fs.url(filename)

    status="pending"

    print("insert into Exporters(ExpotrsName,RegistrationCode,Country,
State,District,PhoneNumber,dates,file,email,
Password,AccountNumber,BankName,IFSC,status) values('" +
str(ExpotrsName) + "','" + str(RegistrationCode) + "','" +
str(Country) + "','" + str(State) + "','" + str(District) + "','" +
str(PhoneNumber) + "','" + str(dates) + "','" + str(uploadurl) + "','" +
+ str(email) + "','" + str>Password) + "','" + str(AccountNumber) +
"','" + str(BankName) + "','" + str(IFSC) + "','" + str(status) + "')")

c.execute("insert into

    Exporters(ExpotrsName,RegistrationCode,Country,State,
District,PhoneNumber,dates,file,email,Password,
AccountNumber,BankName,IFSC,status) values('" + str(ExpotrsName) +
"','" + str(RegistrationCode) + "','" + str(Country) + "','" +
str(State) + "','" + str(District) + "','" + str(PhoneNumber) + "','" +
+ str(dates) + "','" + str(uploadurl) + "','" + str(email) + "','" +
str>Password) + "','" + str(AccountNumber) + "','" + str(BankName) +
"','" + str(IFSC) + "','" + str(status) + "')")

usertype="Exporters"

c.execute("insert into login(username,password,usertype,status)
values('"+ email + "','" + Password + "','" + 'Exporters' + ',' + 'pending' + ')')

db.commit()

```

```

msg="saved"

return render(request,'common/Exporters.html')

def publicreg(request):
    rs=''

    if request.POST:
        yourname=request.POST.get("yourname")
        phonenumber=request.POST.get("phonenumber")
        Username=request.POST.get("Username")
        Password=request.POST.get("Password")
        res=c.execute("select count(*) as cnt from publicreg where phonenumber='"
                     + phonenumber + "' or Username='"
                     + Username + "')")
        rs=c.fetchone()
        db.commit()

        if rs[0]>0:
            msg="Already registered"
        else:
            c.execute("insert into publicreg(yourname,phonenumber,Username,Password)
                      values('"
                      + str(yourname) + "','"
                      + str(phonenumber) + "','"
                      + str(Username) + "','""
                      + str>Password) + "')")

            usertype="Public"
            status="Approved"
            c.execute("insert into login(username,password,usertype,status) values('"
                      + str(Username) + "','""
                      + str>Password) + "','""
                      + str(usertype) + "','""
                      + str(status) + "')")
            db.commit()

    return render(request,'common/common_register.html')

def adminpublicapproval(request):
    c.execute("select * from publicreg")
    res=c.fetchall()

    return render(request,'admin/admin_publicapproval.html',

```

```

        {"res":res})

def adminfarmerapproval(request):
    status="Pending"
    c.execute("select * from farmerreg where status='Pending'")
    res=c.fetchall()
    if request.GET.get("id"):
        id=request.GET.get("id")
        c.execute("select email from farmerreg where farmerid='"+ id +"' ")
        email=c.fetchone()
        c.execute("update farmerreg set status='Approved' where farmerid ='" + id +
                  "' ")
        c.execute("update login set status='Approved' where username='" + email[0] +
                  "' ")
    db.commit()

    return render(request,'admin/adminfarmerapproval.html',
        {"res":res})

def adminExportersapproval(request):
    status="pending"
    c.execute("select * from exporters where status='pending'")
    res=c.fetchall()
    db.commit()
    if request.GET.get("id"):
        id=request.GET.get("id")
        c.execute("select * from exporters where Exp_regnnumber='"
                  + id + "' ")
        res1=c.fetchone()
        db.commit()
        email=res1[9]
        c.execute("update exporters set status='Approved' where Exp_regnnumber='"
                  + id + "' ")
        c.execute("update login set status='Approved' where username='" + email + "' ")

```

```

        db.commit()

    return render(request,'admin/adminExportersapproval.html',
{"res":res})

def admin_addSpecialist(request):

    msg=""
    res=0
    res1=0

    if request.POST:
        Name=request.POST.get("Name")
        Email=request.POST.get("Email")
        MobileNumber=request.POST.get("MobileNumber")
        OfficeName=request.POST.get("OfficeName")
        OfficeContact=request.POST.get("OfficeContact")
        Designation=request.POST.get("Designation")
        Specialized=request.POST.get("Specialized")
        Message=request.POST.get("Message")

        res=c.execute("select count(*) as cnt from specialized where Email='"+
                     str(Email) + "' or MobileNumber='"+
                     str(MobileNumber) + "' or
                     OfficeContact='"+
                     str(OfficeContact) + "'")
        res=c.fetchone()
        res1=c.execute("select count(*) as cnt1 from login where username='"+
                     str(Email) + "'")
        res1=c.fetchone()

        db.commit()

        if res[0]>0 or res1[0]>0:
            msg="already registered"
        else:
            status="Approved"
            usertype="Specialist"
            c.execute("insert into specialized(Name,Email,MobileNumber,OfficeName,
            OfficeContact,Designation,Specialized,Message) values('"+
            str(Name) + "','"+
            str(Email) + "','"+
            str(MobileNumber) + "','"+
            str(OfficeName) + "','"+
            str(OfficeContact) + "','"+
            str(Designation) + "','"+
            str(Specialized) + "','"+
            str(Message) + "')")
    
```

```

        str(Specialized) +'" ,'" + str(Message) +'" )")  

c.execute("insert into login(username,password,usertype,status) values('"  

        + str>Email) + " ,'" + str(MobileNumber) +'" ,'" + str(usertype) +  

        " ,'" + str(status) +'" )")  

db.commit()  

msg="saved"  

return render(request,'admin/admin_addSpecialist.html')  

def admin_farmer_payment(request):  

    res=""  

    rs=0  

    res=c.execute("select * from farmerreg")  

    rs=c.fetchall()  

    return render(request,'admin/admin_farmer_payment.html',  

{'rs':rs})  

def paymentaction(request):  

    res1=""  

    from datetime import datetime  

    if request.GET.get("id"):  

        id=request.GET.get("id")  

        c.execute("select * from farmerreg where farmerid='"  

        + id + '"')  

        res1=c.fetchall()  

        res1=res1[0]  

        if request.POST:  

            farmerid=request.POST.get("farmerid")  

            dates=datetime.today().strftime('%Y-%m-%d')  

            amount=request.POST.get("Amount")  

            c.execute("insert into farmerpayment(farmerid,amount,dts) values('" +  

                farmerid + " ,'" + dates + " ,'" + amount + " ')")  

            db.commit()  

            return HttpResponseRedirect('/admin_farmer_payment/')

```

```

return render(request,'admin/paymentaction.html',
{"res1":res1})

def admincategoryadd(request):
    if request.POST:
        name=request.POST.get("CatName")
        c.execute("insert into category(Categoryname) values('"
        + name + "')")
        db.commit()
    return render(request,'admin/category.html')

def farmeraddproduct(request):
    c.execute("select * from category")
    rs=c.fetchall()
    if request.POST and request.FILES.get("file1"):
        cat=request.POST.get("cat")
        Name=request.POST.get("Name")
        myfile=request.FILES.get("file1")
        fs=FileSystemStorage()
        filename=fs.save(myfile.name,myfile)
        uploadurl=fs.url(filename)
        unit=request.POST.get("unit")
        rate=request.POST.get("rate")
        quantity=request.POST.get("quantity")
        email=request.session["username"]

        c.execute("insert into product(cat,productname,file,unit,rate,quantity,email)
values('" + cat + "','" + Name + "','" + uploadurl
+ "','" + unit + "','" + rate + "','" + quantity
+ "','" + email + "')")
        db.commit()

```

```

return render(request,'farmer/farmeraddproduct.html',
{'data':rs})

def farmeraddseed(request):
    if request.POST and request.FILES.get("file1"):

        Name=request.POST.get("Name")
        scientificname=request.POST.get("scientificname")
        verity=request.POST.get("verity")
        rate=request.POST.get("rate")
        quantity=request.POST.get("quantity")
        Description=request.POST.get("Description")
        unit=request.POST.get("unit")
        myfile=request.FILES.get("file1")
        fs=FileSystemStorage()
        filename=fs.save(myfile.name,myfile)
        uploadurl=fs.url(filename)
        email=request.session["username"]

        c.execute("insert into seed(Name,file1,scientificname,verity,rate,quantity
,Description,unit,email) values('" + Name + "','" +uploadurl + "','" +
scientificname + "','" + verity + "','" + rate + "','" + quantity + "','" +
+ Description + "','" + unit + "','" + email + "')")
        db.commit()

    return render(request,'farmer/farmeraddseed.html')

```

Chapter 7

Testing and Implementation

7.1 Testing and various types of testing used.

Once a software is developed, the major activity is to test whether the actual results match with the experimental results. This process is called testing. It's used to make sure that the developed system is defect free. The main aim of testing is to find the errors and missing operations by executing the program. It also ensure that all of the objectives of the project are met by the developer. The objective of testing is not only to evaluate the bugs in the created software but also finding the ways to improve the efficiency, usability and accuracy of it. It aims to measure the functionality, specification and performance of a software program. Tests are performed on the created software and their results are compared with the expected documentation. When there are too much errors occurred, debugging is performed. And the result after debugging is tested again to make sure that the software is error free. The major testing processes applied to this project are unit testing, integration testing and system testing. In unit testing, our aim is to test all individual units of the software. It makes sure that all of the units of the software works as it intended. In integration testing, the combined individual units are tested to check whether it met the intended function or not. It helps us to find out the faults that may arise when the units are combined. In system testing the entire software is tested to make sure that it satisfies all of the requirements. The tables shown below describes the testing process occurred during the development of this project "Farmers Help". This defines the various steps took to create the project error free.

7.1.1 Unit Testing

It involves the basic testing of a piece of code, the size of which is often undefined in practice. During the unit testing it is tested to know whether that particular unit in the proper manner as expecting, if not appropriate modifications are applied to get proper outputs

7.1.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated IT) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

7.1.3 System Testing

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input.

7.1.4 Test Case Description

Table 7.1: Test Cases and Results

S-No	Procedures	Expected result	Actual result	Pass or Fail
1	Admin-Login	The administrator logs into the admin portal using his/her login credentials	Same as expected	Pass
2	Farmer-Register and Login	Farmers register with the system which enables them to access their portal.	Same as expected	Pass

Continued on next page

Table 7.1 – Continued from previous page

S-No	Procedures	Expected result	Actual result	Pass or Fail
3	Admin-Approve Farmer	Registered Farmers are approved by the admin	Same as expected	Pass
4	Exporter-Register and Login	Exporters register with the system which enables them to access their portal.	Same as expected	Pass
5	Admin-Approve Exporter	Registered exporters are approved by the admin	Same as expected	Pass
6	Supplier-Register and Login	Suppliers register with the system which enables them to access their portal.	Same as expected	Pass
7	Admin-Approve Supplier	Registered suppliers are approved by the admin	Same as expected	Pass
8	Public-Register and Login	Public register with the system which enables them to access their portal.	Same as expected	Pass
9	Admin-Register Investor	Investors are registered by the admin	Same as expected	Pass

Continued on next page

Table 7.1 – Continued from previous page

S-No	Procedures	Expected result	Actual result	Pass or Fail
10	Admin-Register Specialist	Specialists are registered by the admin	Same as expected	Pass
11	Specialist-Login	The Specialist logs into the admin portal using hisher login credentials	Same as expected	Pass
12	Farmers-edit Profile	farmers can edit their profile and the changes made are reflected in their portal	Same as expected	Pass
13	Suppliers-add products	add new products along with the details so that farmers can make order	Same as expected	Pass
14	Farmers-add products	add new products along with the details so that exporters can make order	Same as expected	Pass
15	Farmers-request for capital	farmers who have requested for capital assistance are either approved or rejected	Same as expected	Pass

Continued on next page

Table 7.1 – *Continued from previous page*

S-No	Procedures	Expected result	Actual result	Pass or Fail
16	Farmers-add techniques	add farming techniques to make it available to the public and other farmers	Same as expected	Pass
17	Farmers-add queries	Farmers add questions about farming so that specialists can answer them	Same as expected	Pass
18	Specialist Reply	Specialists answer farmers' questions so that farmers can improve their farming	Same as expected	Pass
19	Exporters-Order	Exporters make order	Same as expected	Pass

Chapter 8

Results and Discussion

The main objective of the project is to bridge the gap between the farmers and consumers. It provides complete coordination of various activities handled by different classes of people in the farming field. Despite offering a wide range of advantages over the existing system, ‘FARMERS HELP’ has its own set of limitations. But the limitations are outnumbered by the advantages that the project offers and they can be overcome in the future.

8.1 Advantages and Limitations

8.1.1 Advantages

- Through a website like this, farmers can connect directly with customers and make their transactions profitable for them without a middleman/third party.
- It Stores the information about all users and users can view and modify their profile.
- Provides instant access to information.
- Easy to use features

8.1.2 Limitations

- Lack of disabled-friendly features
- Lack of Automatic notifications to the registered users to their mobile number or email
- Lack of payment integration

Chapter 9

Conclusion

Color Your Farm is website that provides benefits in the society. It improves the social value of our country. This application provides almost everything for providing support and assistance to the farmers. The farmers can take expert assistance in any of the cultivation stages like plot preparation, planting, managing, detection of diseases and infections by making enquiries to them. This system is developed in such a way that the modules developing in the future can be linked easily to the system, without affecting the existing system, since it provides a hierarchical structure.

Top down programming approach has been adopted while developing the project; each task is divided into separate modules. Hence the modification and enhancement can be easily made without affecting any other part of the program. This system has been developed to satisfy the user needs. The entire system is user friendly. The performance of the system is provided efficiently. The system was tested with all possible test data and was found to have an effective planning of the functions or processes with a high degree of accuracy and user friendliness.

The project entitled “FARMERS HELP “is used to automate all the activities in a farming field. The primary aim of the project was to develop website, which makes the working of the system in an easiest way. The developed website meets most of the requirements and every effort has been done to make the tasks work simpler and easier. The website developed is implemented and tested successfully

Chapter 10

Future Scope

The future scope of this project is that the work can be extended to multiple dimensions. One of the main features of the web technology is its extensibility. The Farmers Help is a complete system as per the user requirement. But it is obvious that the user requirements will change in future as the environment changes.

The word of computers is not always static. It is always subject to change. The software too has to mold to suit the requirements of the future. The system is open to reasonable changes and these changes can be brought about very easily. But drastic changes like changes in the processing flow, changes in system functioning etc may not be able to accommodate in the existing system. It can be strongly said that the system has a fair scope for future development that doesn't affect the process flow severely. For the future we can implement facilities like:

- Make it usable to all uses.
- Build an android app for the same.
- Add messaging and email features to notify.
- Payment Integration

Bibliography

- [1] Sindhu M R, Aditya Pabshettiwar, Ketan . K. Ghumatkar, Pravin . H . Budhehalkar, Paresh . V . Jaju, “E-FARMING”, G. H. Raisoni College of Engineering and Management, Pune, India., International Journal of Computer Science and Information Technologies (IJCSIT).
- [2] Dr M . Rajarajeswari , “AGRICULTURE MANAGEMENT SYSTEM USING – PYTHON”, December 2019, Journal of The Gujarat Research Society.
- [3] Megha Nayak, Pinky Wankhede, Neha Khapekar, Komal Dhote., “E-COMMERCE SITE FOR AGRICULTURE PRODUCTS”.,April 2019, Dept. of Computer Science Engineering, Nagpur Institute of Technology college, Maharashtra, Nagpur., International Research Journal of Engineering and Technology (IRJET)