UML AND TESTING TOOLS LAB

FARM

MANAGEMENT SYSTEM

(BATCH-3)

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
Submitted By:

1. Ms. B. Sai Sudha Sri	(A21126511069)
2. Mr. K. Hemanth Sai	(A21126511101)
3. Mr. M. Sanjay	(A21126511106)
4. Mr. D.S Sai Suryanarayana	(A21126511083)

Under The Guidance

Mrs. A. Surekha

Assistant Professor

PRAGNANAM BRAHMA
ANITS

DEPARTMENT OF INFROMATION TECHNOLOGY

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

ACKNOWLEDGMENT

An endeavour over a long period can be successful with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them. We awe our tributes to Dr. M. Rekha Sundari, Head of the Department, Information technology, ANITS, for providing us with the required facilities for the implementation of the project work.

We wish to express our sincere thanks and gratitude for lecturer in charge Mrs. A. Surekha Mam, Assistant Professor, of Information Technology, ANITS for analysing problems associated with our project work and for guiding us throughout the project. We express our warm and sincere thanks for the encouragement, untiring guidance and the confidence she had shown in us. We are immensely indebted for her valuable guidance throughout our project. We thank all the staff members of IT Department for their valuable devices and for providing resources and when required.

DEPARTMENT OF INFORMATION TECHNOLOGY

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)

(Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with ‘A’Grade,
Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

CERTIFICATE

This is to certify that the project reported entitled **“FARM MANAGEMENT SYSTEM”** submitted by B. SAI SUDHA SRI, K. HEMANTH SAI, M. SANJAY, D. SIVA SAI SURYANARAYANA in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** in Anil Neerukonda Institute of technology and sciences, Visakhapatnam is a record of work carried out under my guidance and supervision.

Lecturer In Charge	**Head of the Department**
Mrs. A. Surekha,	Prof. M. Rekha Sundari,
Department of IT	Department of IT

Table of Contents

INTRODUCTION

1.1 Problem Statement:

- The **FARM MANAGEMENT SYSTEM** project is a great project. This project is designed for successful execution of blood bank management system functionality.
- The project Farm management system is designed to make better connection among Farmers and Buyers to ensure quality food. Standardize and increase efficiency of agro culture process.
- It is a browser-based system that is designed to store, process, retrieve and analyse information concerned with the administrative and inventory management within a Farm management system.

1.2 Description:

- Admin is the main role in the system, admin can manage all the activities like managing farmer, buyers and products etc.
- Farmer is also an important role in the system. If any farmer want to sell their products, he or she has to register himself first. Once he or she register he/she can login to the system.
- To make a software fast in processing, with good user interface.
- Computerization can be helpful as a means of saving time and money.
- Provides Security to the data by using login and password method.
- To provide immediate storage and retrieval of data and information.
- Improving arrangements for farmers co-ordination
- This system is used for maintain whole information about admin, farmer, products stock and buyers.
- There are 3 ACTORS in this project.
    - Admin
    - Farmers
    - Buyers

1.3 REQUIREMENTS OF ACTORS:

Farmer:

- Farmer's Name
- Farmer's address
- Farmer's phone number
- Farmer's email
- Products
- Crop name

User:

- User's name
- User's email
- User's contact no
- User's address
- Orders

Admin

- Admin email id
- Admin name

Availability

- Currently available agro products

Records

- Farmer records
- Buyer records
- Admin details
- Products records

SOFTWARE REQUIREMENT SPECIFICATIONS:

2.1 Introduction

Purpose:

- This project is built to maintaining all the information of farmer, buyer and the product stock.
- Aim is to provide transparency in this field, As there is a notable gap in the agricultural market between farmers producing goods and consumers seeking direct access to fresh, locally sourced farm products.
- The absence of an efficient online platform impedes farmers from reaching a wider market and restricts consumers from easily accessing and purchasing farm goods.
- This lack of direct connectivity leads to underutilized agricultural production capacity, limited market reach for farmers, and consumers facing difficulties in sourcing fresh, locally produced farm products.

Scope:

- This SRS document outlines the functional and non-functional requirements of the Farm Management System, specifying the features, and user interactions.

Technologies used:

Software Requirements:

1. Front-End Development:

- Html (Hyper Text Mark-up Language)
- CSS (Cascading Style Sheets)
- Java Script
- Bootstrap

2. Back-End Development:

- PHP
- MySQL
- Python flask (Python 3.7)
- SQL Alchemy
- Operating System: Windows 10
- Google Chrome/Internet Explorer
- XAMPP (Version-3.7)
- Python main editor (user interface): PyCharm Community

Hardware Requirements:

- Computer with a 1.1 GHz or faster processor
- Minimum 2GB of RAM or more
- 2.5 GB of available hard-disk space
- 5400 RPM hard drive
- 1366 × 768 or higher-resolution display
- DVD-ROM drive

2.2 System Overview

System Description

- This project is built to maintaining all the information of farmer, buyer and the product stock.
- Aim is to provide transparency in this field, As there is a notable gap in the agricultural market between farmers producing goods and consumers seeking direct access to fresh, locally sourced farm products.
- The absence of an efficient online platform impedes farmers from reaching a wider market and restricts consumers from easily accessing and purchasing farm goods.
- This lack of direct connectivity leads to underutilized agricultural production capacity, limited market reach for farmers, and consumers facing difficulties in sourcing fresh, locally produced farm products.

Users:

- **Admin:** Admin is the main role in the system, admin can manage all the activities like farmer, buyer and product stock etc.
- **Farmer:** Farmer is also an important role in the system. If any farmer wants to sell their products, he or she has to register themselves first. Once he or she registers he/she can login to the system.
- **Buyer:** Customer can buy the fresh products directly from the farmer via online platform. He can go to the system and register himself as a customer. Once he registers, he/she can login to the system and access.

FUNCTIONAL REQUIREMENTS

3.1 Admin Functionalities:

- Check the stock of the products
- Manage farmers
- Manage buyers
- Logout
- Admin can manage all the donor and patient. He can edit the details of farmer. He can delete any farmer.

3.2 Farmer Functionalities:

- Farmer can sell their products
- Manage history of their updates
- Check the status of orders of their products
- Logout
- Once farmer makes a request to add fields or products, admin has to take action on that request based on the farmer details. Once admin accept or reject that farmer request, it will be automatically update to the farmer dashboard.
- Farmer can check the status of his request. Once his request is accepted, he or she will be called to update their dashboard.

3.3 Customer Functionalities:

- Make purchase request
- Check the status of his request
- Logout
- Once the patient makes a request for to purchase the product, he has to provide the basic details like quantity they required etc.
- Once he makes a request, it will be reflected in the admin dashboard. Now admin has to take action on that request. Admin can accept or reject that request based on the customer location or product stock available in the system.

NON – FUNCTIONAL REQUIREMENTS

4.1 Security

- The system must implement robust security measures to protect sensitive data.
- User data should be stored securely, and password hashing should be employed.
- Access control should be implemented to ensure data privacy and integrity.

4.2 Performance

- The system should be responsive and capable of handling a reasonable load.
- Database queries should be optimized for efficient retrieval of data.

4.3 Usability

- The user interface should be user-friendly, intuitive, and accessible.
- The system should support multiple browsers and devices.

4.4 Reliability

- The system should have backup and recovery mechanisms to prevent data loss.

4.5 Scalability

- The system should be scalable to accommodate a growing user base and increasing data.

UML DIAGRAMS

- UML, short for Unified Modelling Language, is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.
- The UML is a very important part of developing object-oriented software and the software development process.
- The UML uses mostly graphical notations to express the design of software projects.

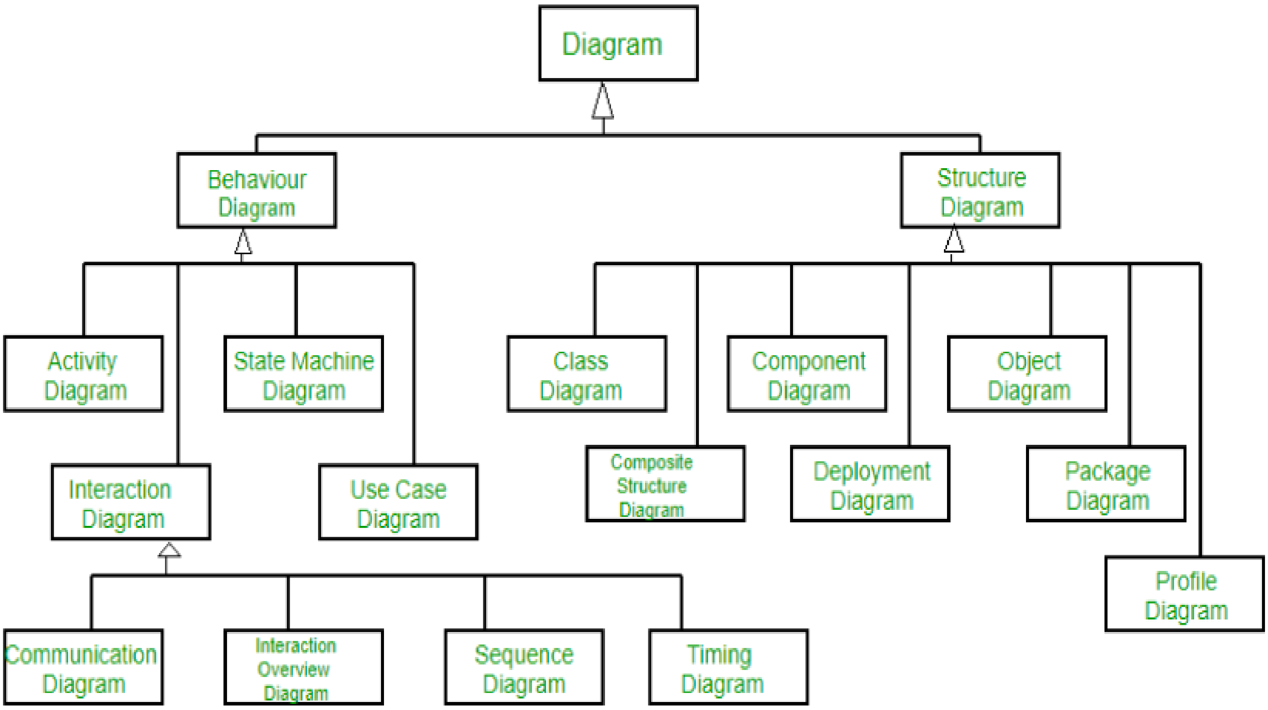


Figure 1: Classification of UML Diagrams

5.1USE CASE DIAGRAM:

- Use Case Diagrams are used to depict the functionality of a system or a part of a system.
- They are widely used to illustrate the functional requirements of the system and its interaction with external agents (actors).
- A use case is basically a diagram representing different scenarios where the system can be used. A use case diagram gives us a high-level view of what the system or a part of the system does without going into implementation details.

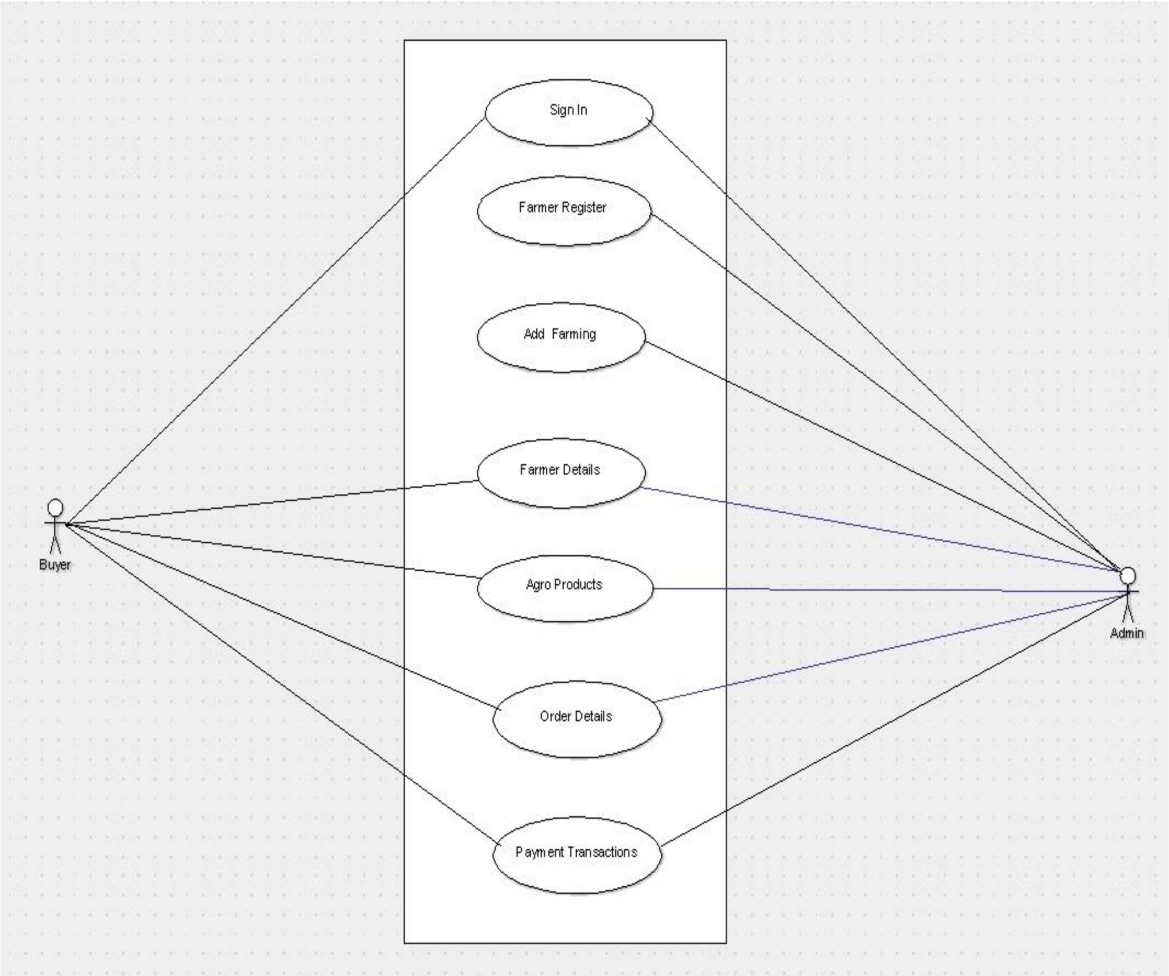


Figure 2: Use Case diagram for Farm Management system

Use Cases:

- Sign In, Farmer Register, Add Farming, Farmer Details, Agro Products, Order Details, Payment Transactions.

Actors:

- Buyer and Admin. Here the admin need to manage everything. She/he needs to manage farmer requests and their details and he can do this only by login in to the Farm Management System.

5.2 CLASS DIAGRAM:

- The most widely use UML diagram is the class diagram. It is the building block of all object-oriented software systems.
- We use class diagrams to depict the static structure of a system by showing system's classes, their methods and attributes. Class diagrams also help us identify relationship between different classes or objects.

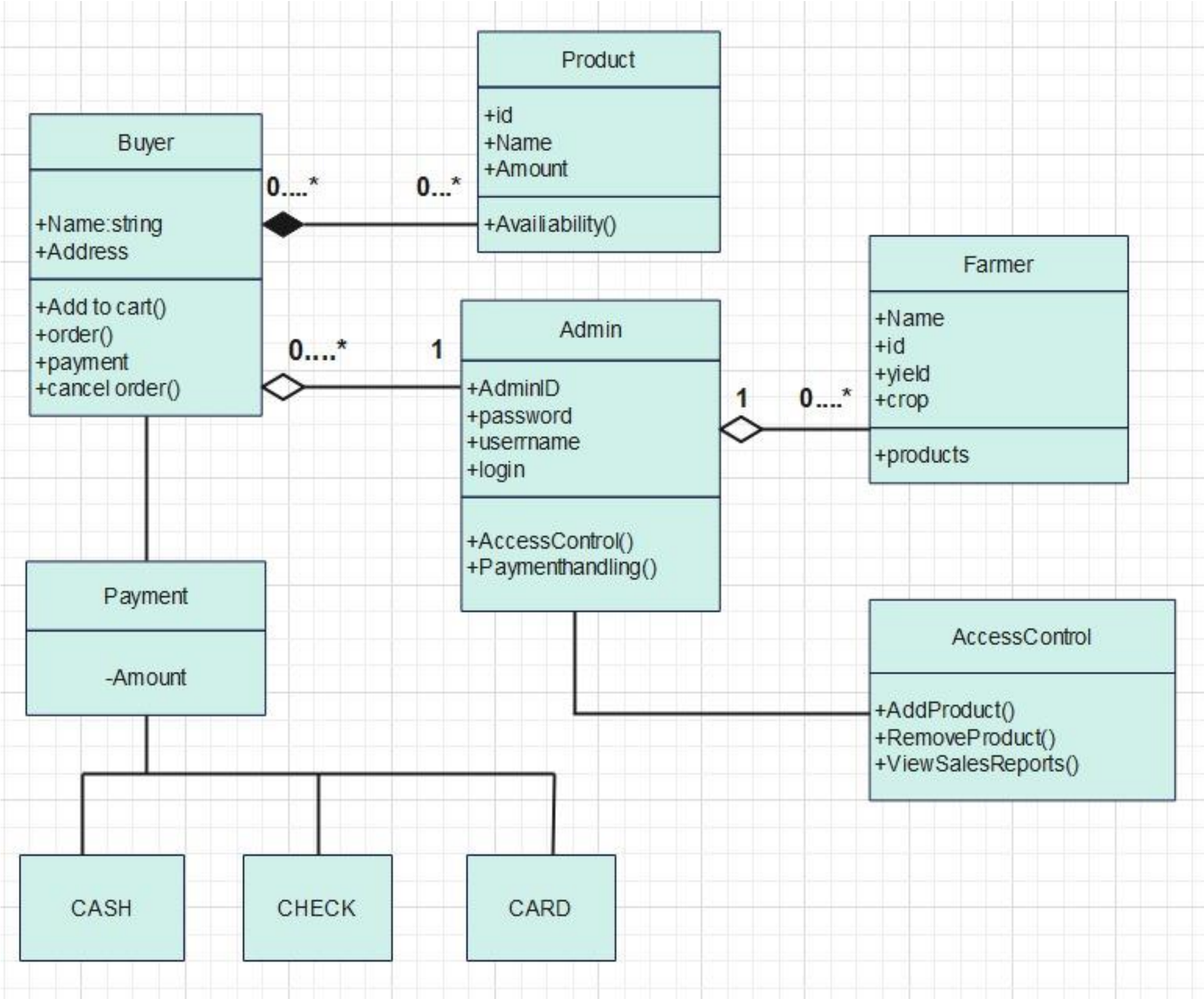


Figure 3: Class diagram for Farm Management System

Classes:

- Admin, Buyer, Product, Farmer, Payment, Access Control. And we used aggregation, composition, unidirectional association relationships between the classes.

5.3STATE DIAGRAM:

- A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioural diagram and it represents the behaviour using finite state transitions.
- State diagrams are also referred to as State machines and State-chart Diagrams.
- These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behaviour of a class in response to time and changing external stimuli.

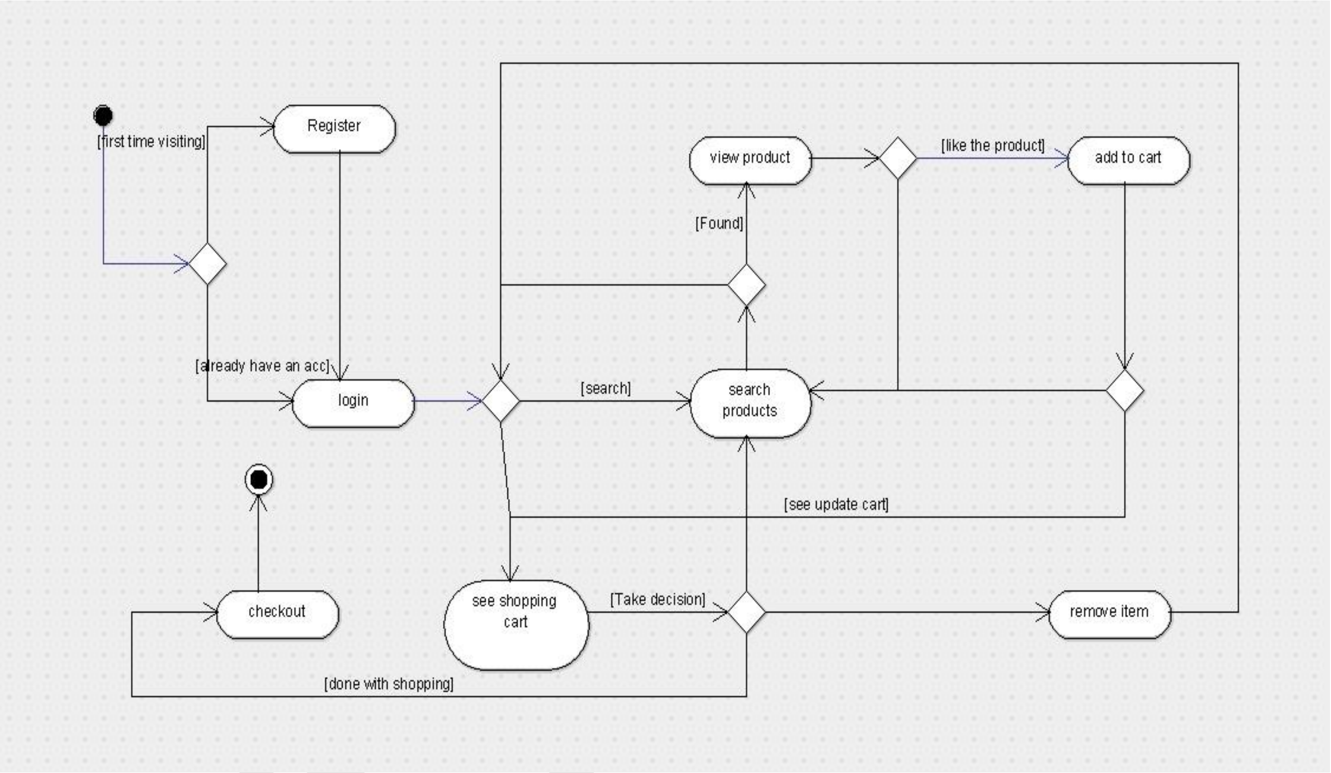


Figure 4: State diagram for Farm Management System.

Login Components:

- As shown in the above State diagram we have created for admin and customer/buyer to login into the system and access the internal functionality as shown in the above State diagram.

5.4 ACTIVITY DIAGRAM:

- We use Activity Diagrams to illustrate the flow of control in a system. We can also use an activity diagram to refer to the steps involved in the execution of a use case.
- We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens.
- We describe or depict what causes a particular event using an activity diagram.

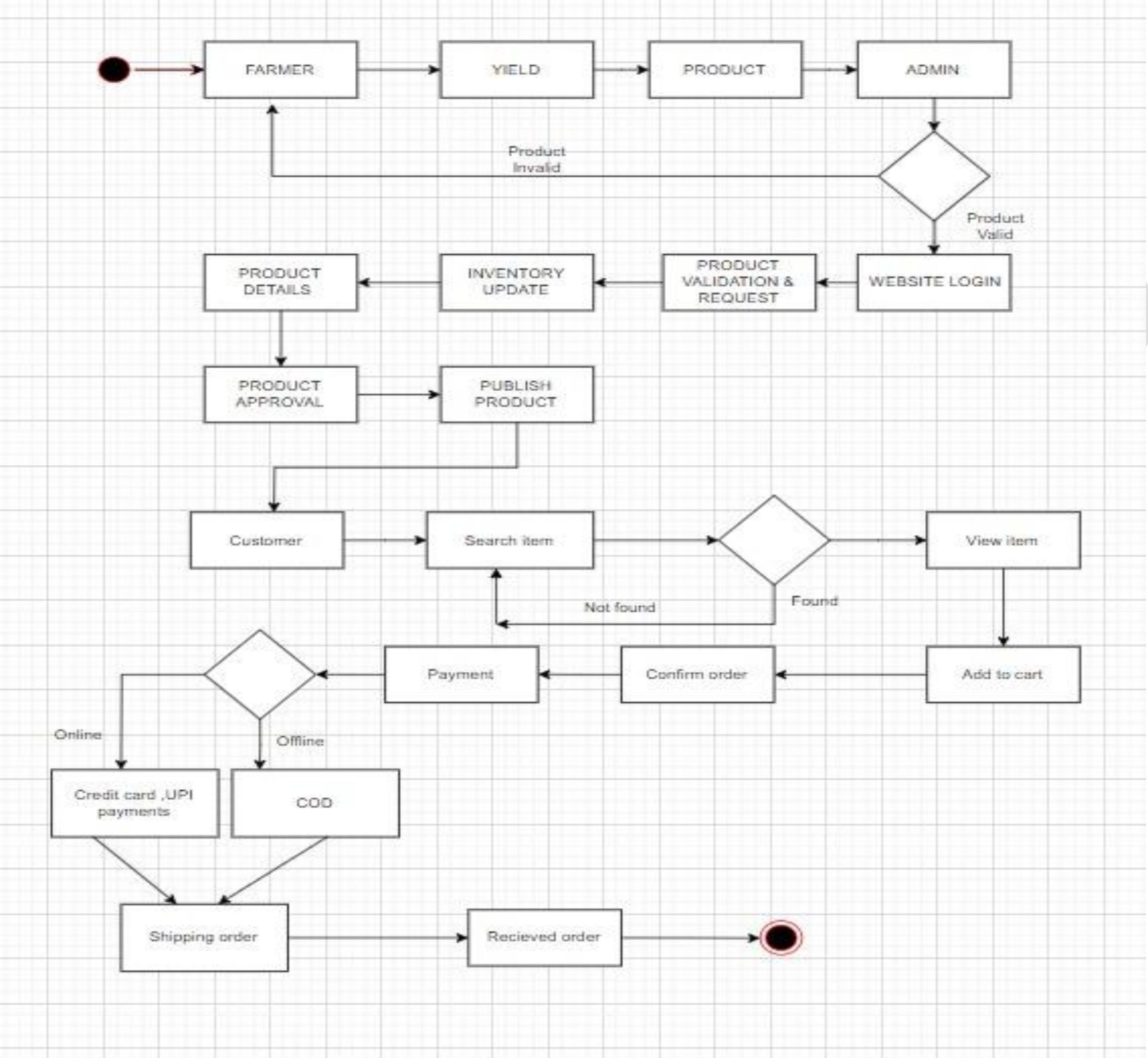


Figure 5: Activity diagram for Farm Management System.

Activities:

- Admin will login into the system and he will manage the farmer details, manage the customer, manage the farmer requests and manage the patient requests and then he will get logout from the system.
- And the buyer will login into the system and then he will request for to purchase the product and check their accepted/rejected requests and get logout from the system.

5.5 SEQUENCE DIAGRAM:

- The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios.
- It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.
- In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

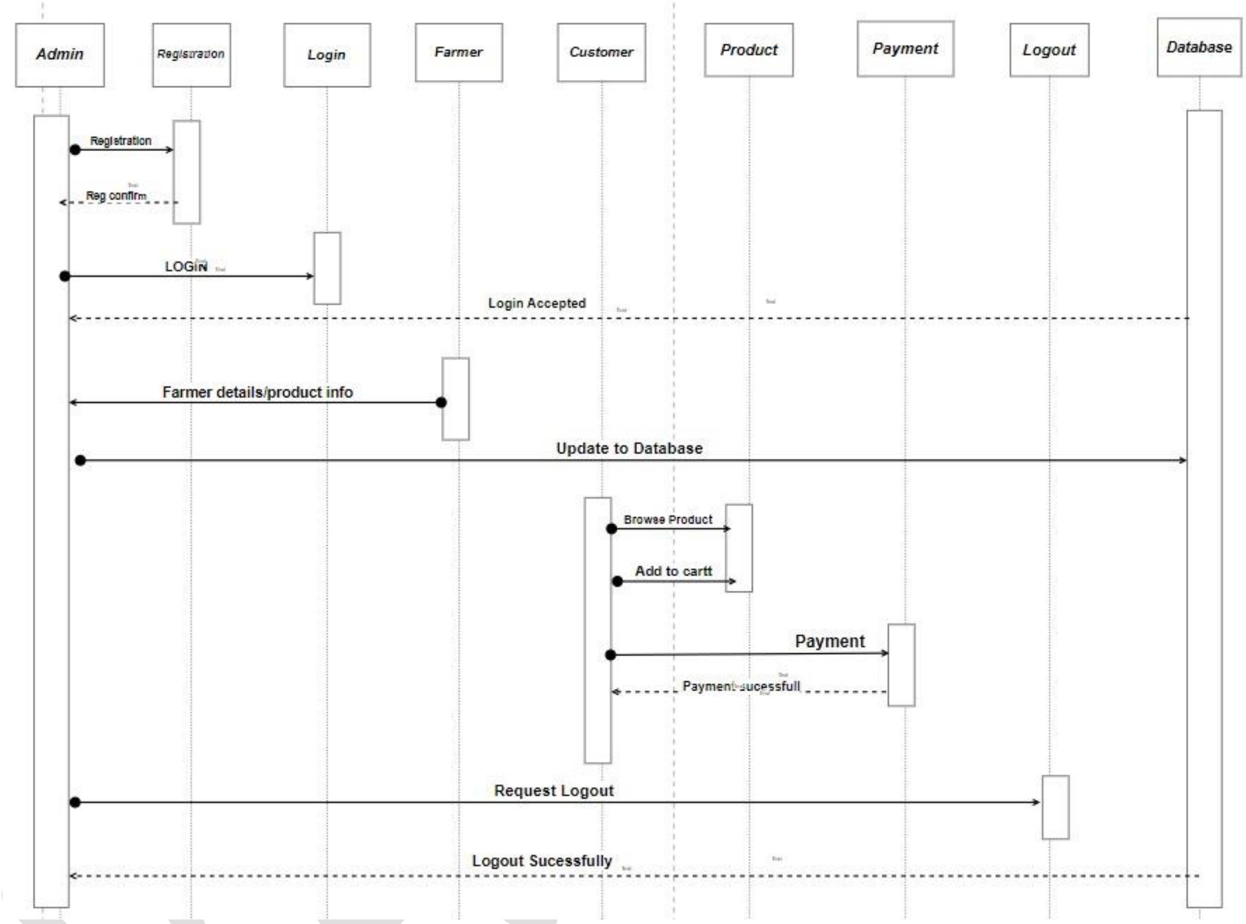


Figure 6: Sequence diagram for Farm Management System.

Actors:

- Admin
- Registration
- Login
- Farmer
- Customer
- Product
- Payment
- Logout
- Database

Messages:

- Registration.
- Registration Confirmation
- Login
- Login accepted
- Farmers details or product information
- Update to database
- Browse product
- Add to cart
- Payment
- Payment successful
- Request to logout
- Logout successful

5.6COLLABORATION DIAGRAM:

- The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently.
- Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.
- An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

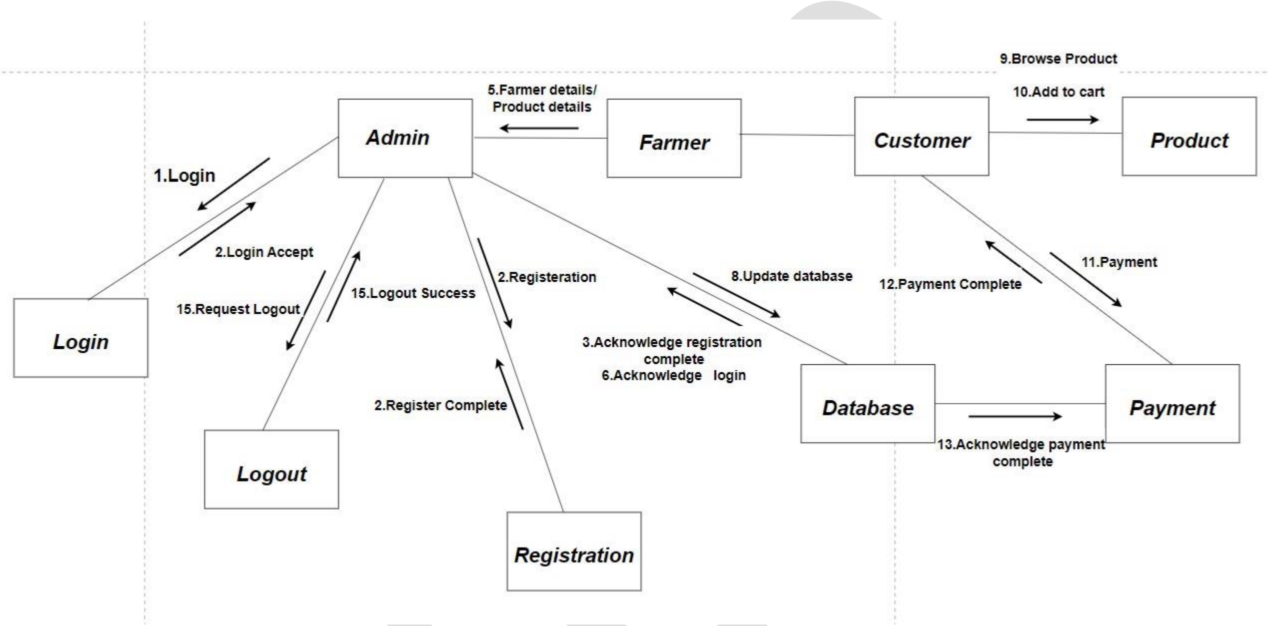


Figure 7: Collaboration diagram for Farm Management System.

Actors:

- Admin
- Farmer
- Customer

Messages:

- Login
- Login accept
- Registration
- Acknowledge registration/login
- Farmer details/product details
- Update database
- Browse product
- Add to cart
- Payment
- Payment successful
- Reguest logout
- Logout success

5.7 COMPONENT DIAGRAM:

- A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable.
- It models the physical view of a system such as executable, files, libraries, etc. that resides within the node.
- It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable.
- The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behaviour is explained by the provided and required interfaces.

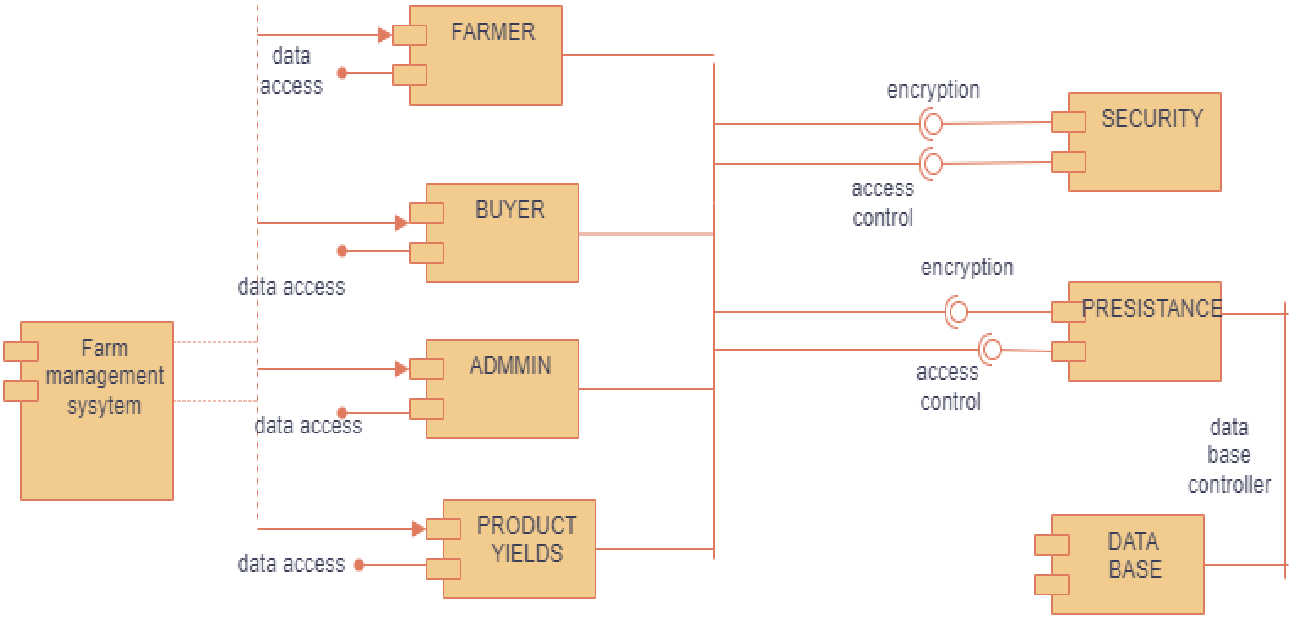


Figure 8: Component diagram for Farm Management System.

Component:

- Farm Management System

Nodes:

- Farmer
- Buyer
- Admin
- Product yields
- Security
- Perinstance
- Database

5.8DEPLOYMENT DIAGRAM:

- Deployment Diagrams are used to represent system hardware and its software. It tells us what hardware components exist and what software components run on them.
- We illustrate system architecture as distribution of software artifacts over distributed targets. An artifact is the information that is generated by system software.
- They are primarily used when software is being used, distributed or deployed over multiple machines with different configurations.

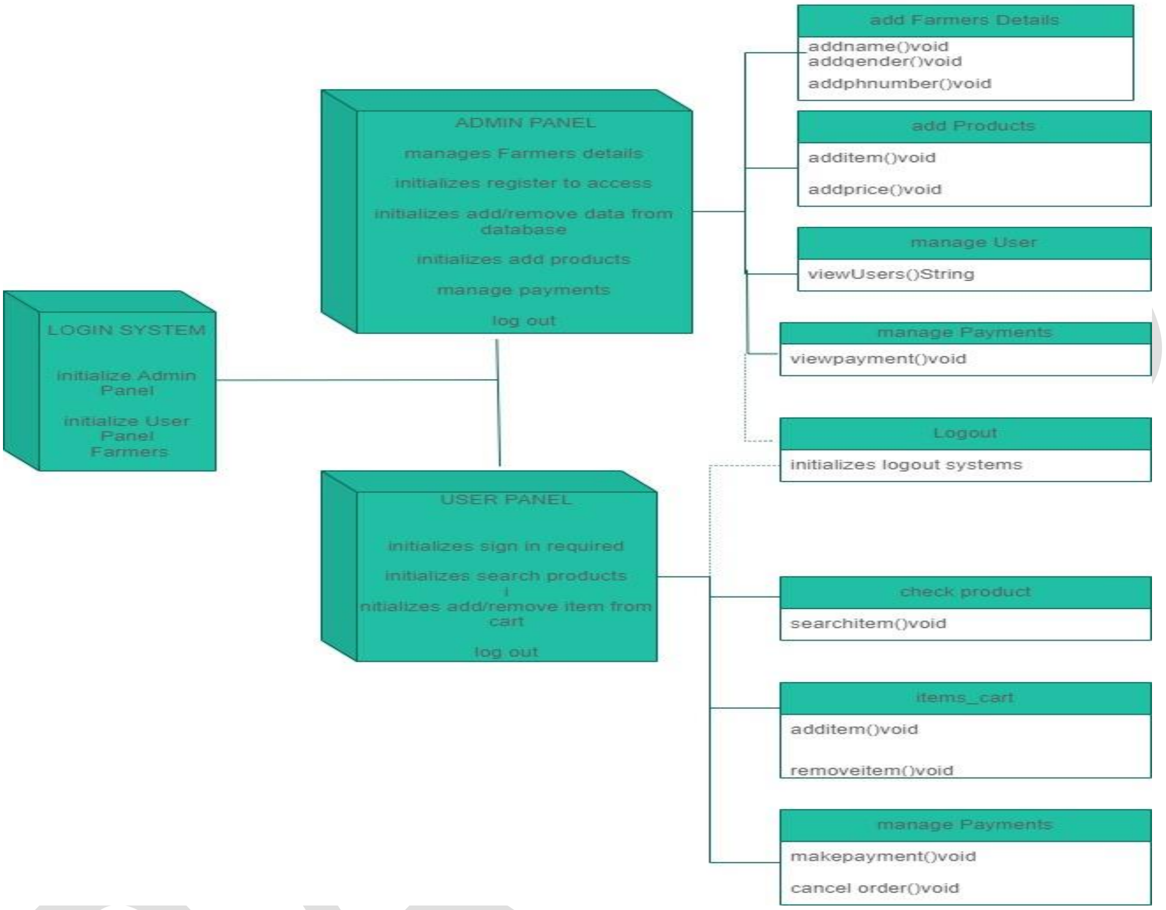


Figure 9: Deployment diagram for Farm Management System.

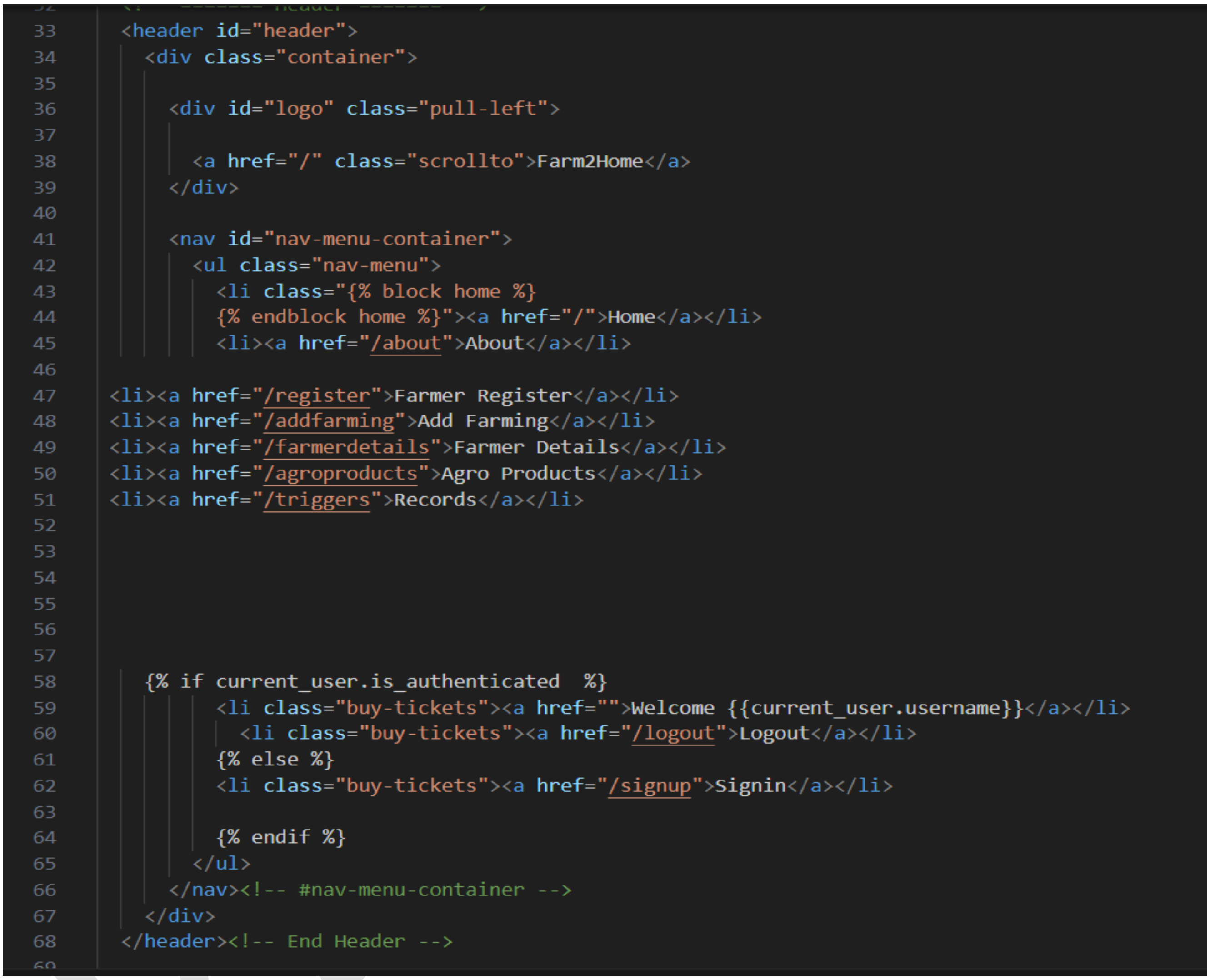
- In the above diagram, the admin will login into the system. After successful login admin can manage product, farmer registrations etc
- User can check the product after successful login into the system and can purchase the product through online or offline mode.

Nodes:

- Login system
- Admin panel
- User panel

WEBSITE

6.1 SAMPLE CODE:

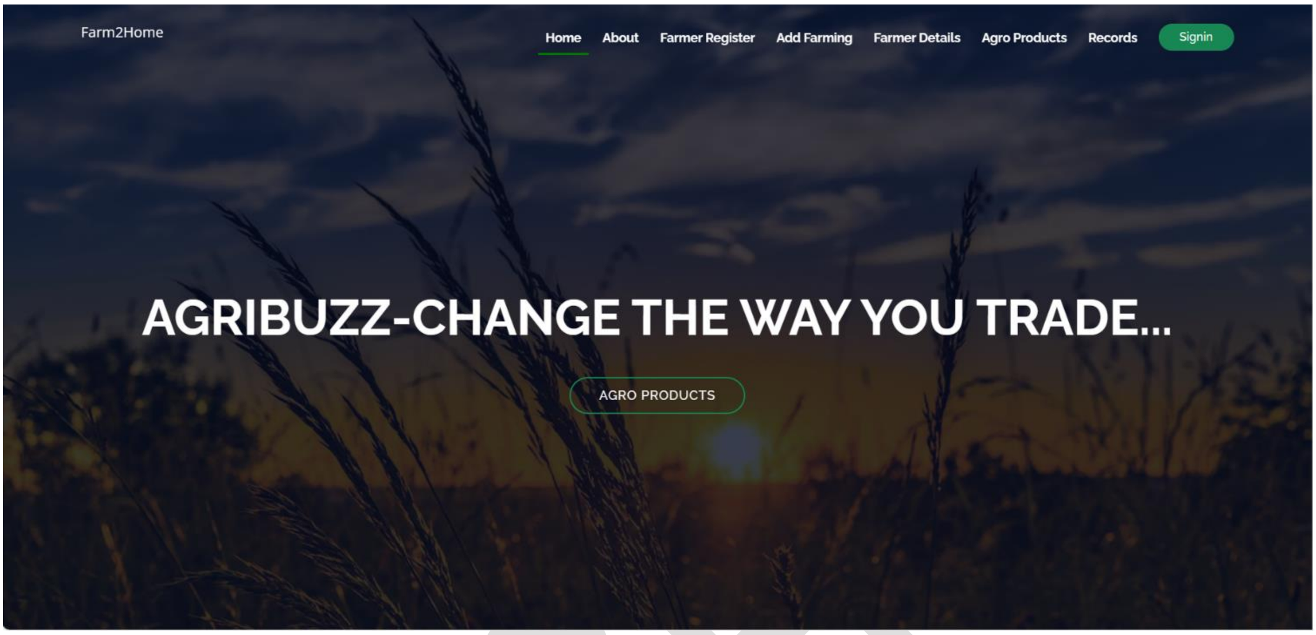


```
 <header id="header">
  <div class="container">

   <div id="logo" class="pull-left">

    <a href="/" class="scrollto">Farm2Home</a>
   </div>

   <nav id="nav-menu-container">
    <ul class="nav-menu">
     <li class="{% block home %}
     {% endblock home %}"><a href="/">Home</a></li>
     <li><a href="/about">About</a></li>

<li><a href="/register">Farmer Register</a></li>
<li><a href="/addfarming">Add Farming</a></li>
<li><a href="/farmerdetails">Farmer Details</a></li>
<li><a href="/agroproducts">Agro Products</a></li>
<li><a href="/triggers">Records</a></li>






  {% if current_user.is_authenticated  %}
     <li class="buy-tickets"><a href="">Welcome {{current_user.username}}</a></li>
      <li class="buy-tickets"><a href="/logout">Logout</a></li>
     {% else %}
     <li class="buy-tickets"><a href="/signup">Signin</a></li>

     {% endif %}
    </ul>
   </nav><!-- #nav-menu-container -->
  </div>
 </header><!-- End Header -->
```

The above is the sample code of “Home”

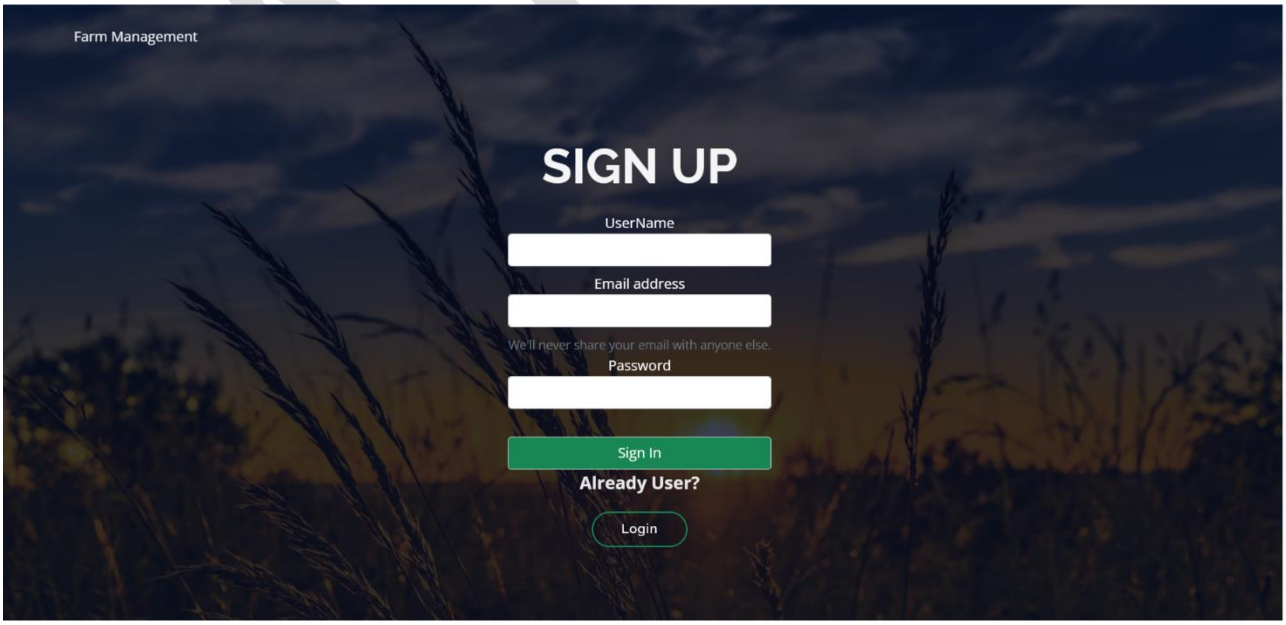
The homepage serves as the initial point of interaction for users accessing the application or website. It aims to provide a welcoming and informative landing page with various sections and features to engage users.

6.2 Home Page:



AGRIBUZZ-CHANGE THE WAY YOU TRADE...

6.3 Sign Up:



SIGN UP

UserName

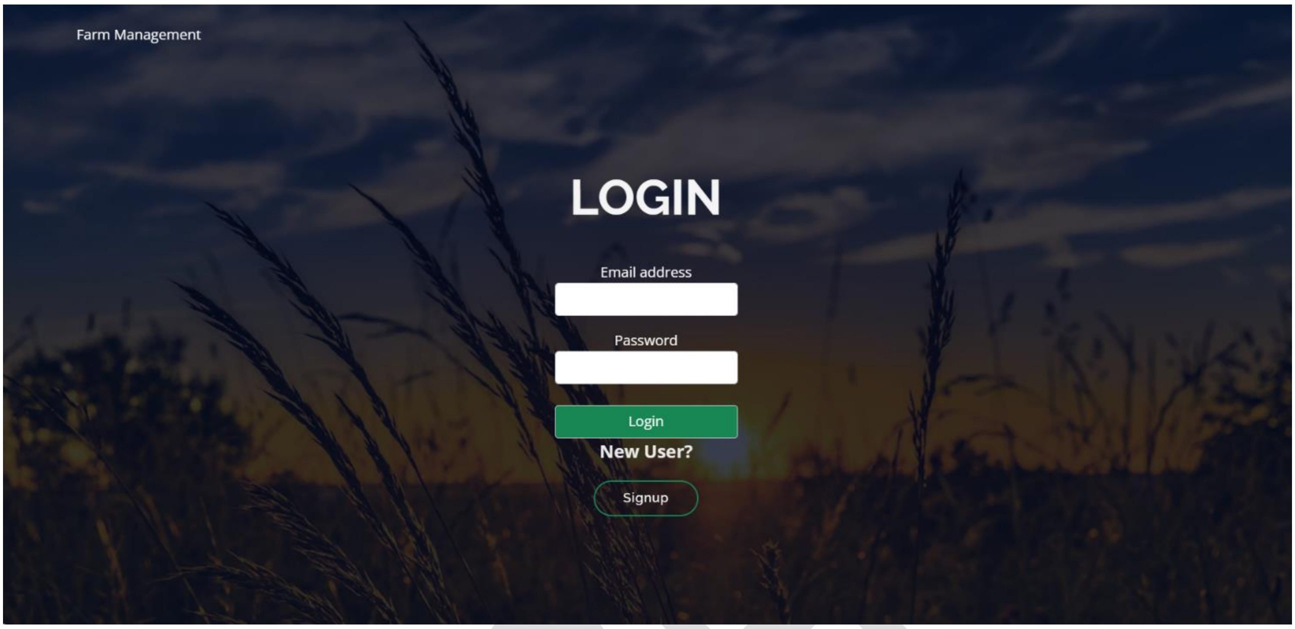
Email address

We'll never share your email with anyone else.

Password

Already User?

6.4 login:

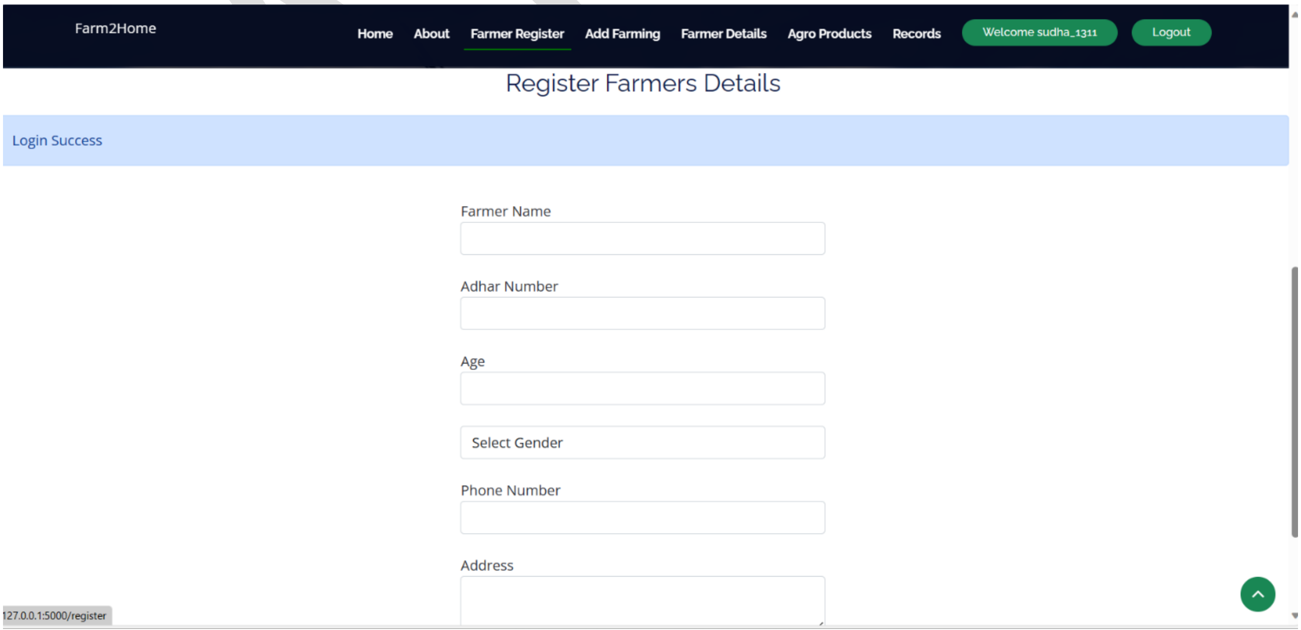


LOGIN

Email address

Password

6.5 Farmer Register:



Register Farmers Details

Login Success

Farmer Name

Adhar Number

Age

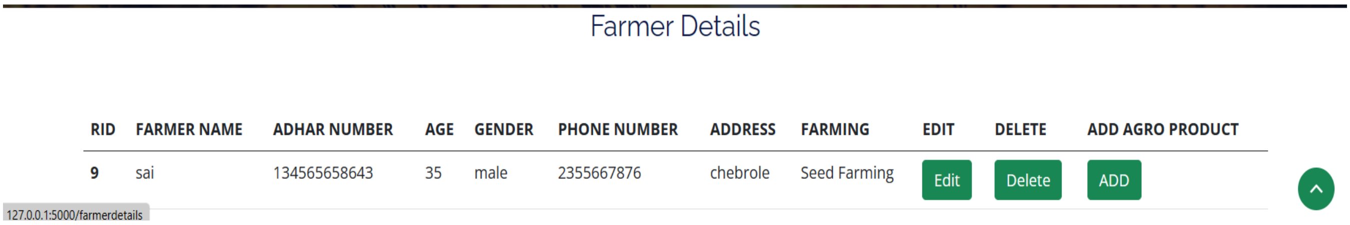
Select Gender

Phone Number

Address

127.0.0.1:5000/register

6.6 Farmer Details:

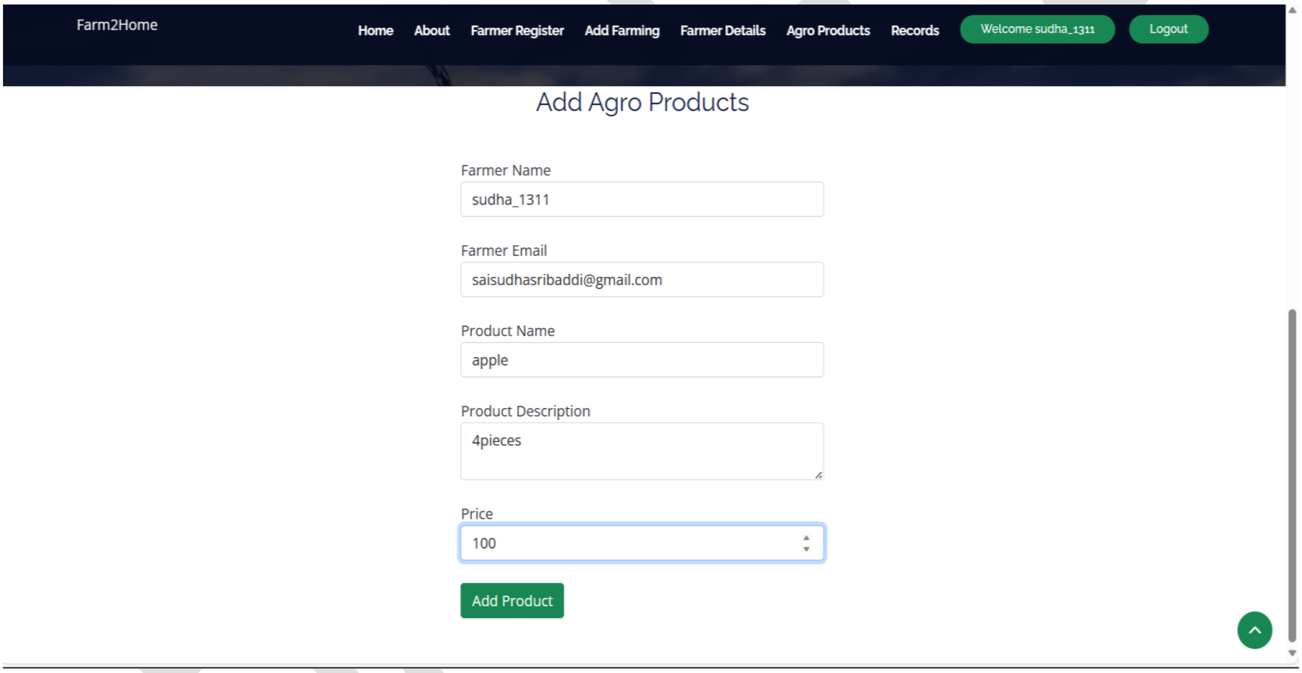


Farmer Details

RID	FARMER NAME	ADHAR NUMBER	AGE	GENDER	PHONE NUMBER	ADDRESS	FARMING	EDIT	DELETE	ADD AGRO PRODUCT
9	sai	134565658643	35	male	2355667876	chebrole	Seed Farming	Edit	Delete	ADD

127.0.0.1:5000/farmerdetails

6.7 Add Products:



Add Agro Products

Farmer Name: sudha_1311

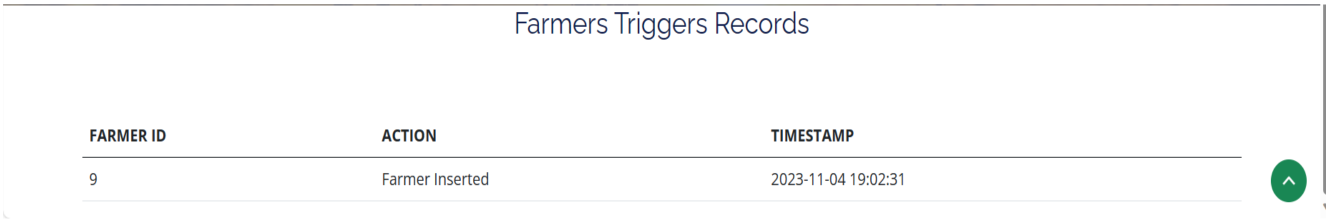
Farmer Email: saisudhasribaddi@gmail.com

Product Name: apple

Product Description: 4pieces

Price: 100

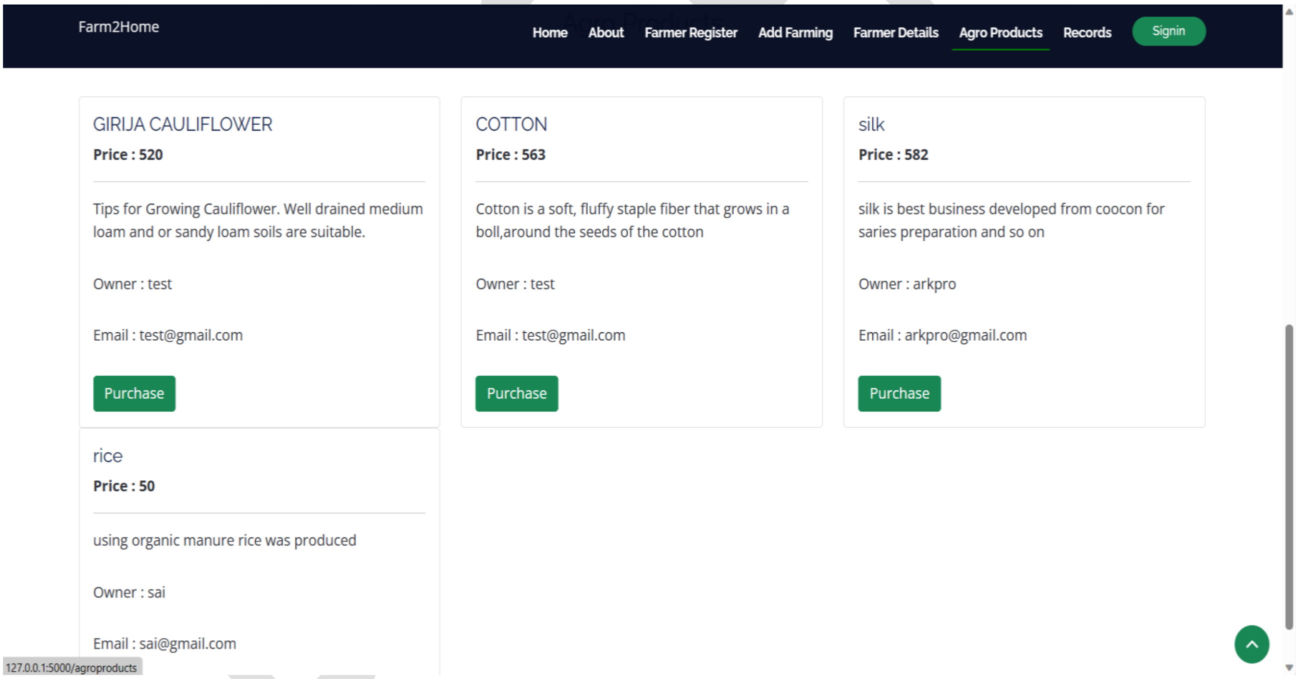
6.8 Records:



Farmers Triggers Records

FARMER ID	ACTION	TIMESTAMP
9	Farmer Inserted	2023-11-04 19:02:31

6.9 Agro Products:



GIRIJA CAULIFLOWER

Price : 520

Tips for Growing Cauliflower. Well drained medium loam and or sandy loam soils are suitable.

Owner : test

Email : test@gmail.com

COTTON

Price : 563

Cotton is a soft, fluffy staple fiber that grows in a boll,around the seeds of the cotton

Owner : test

Email : test@gmail.com

silk

Price : 582

silk is best business developed from coocon for saries preparation and so on

Owner : arkpro

Email : arkpro@gmail.com

rice

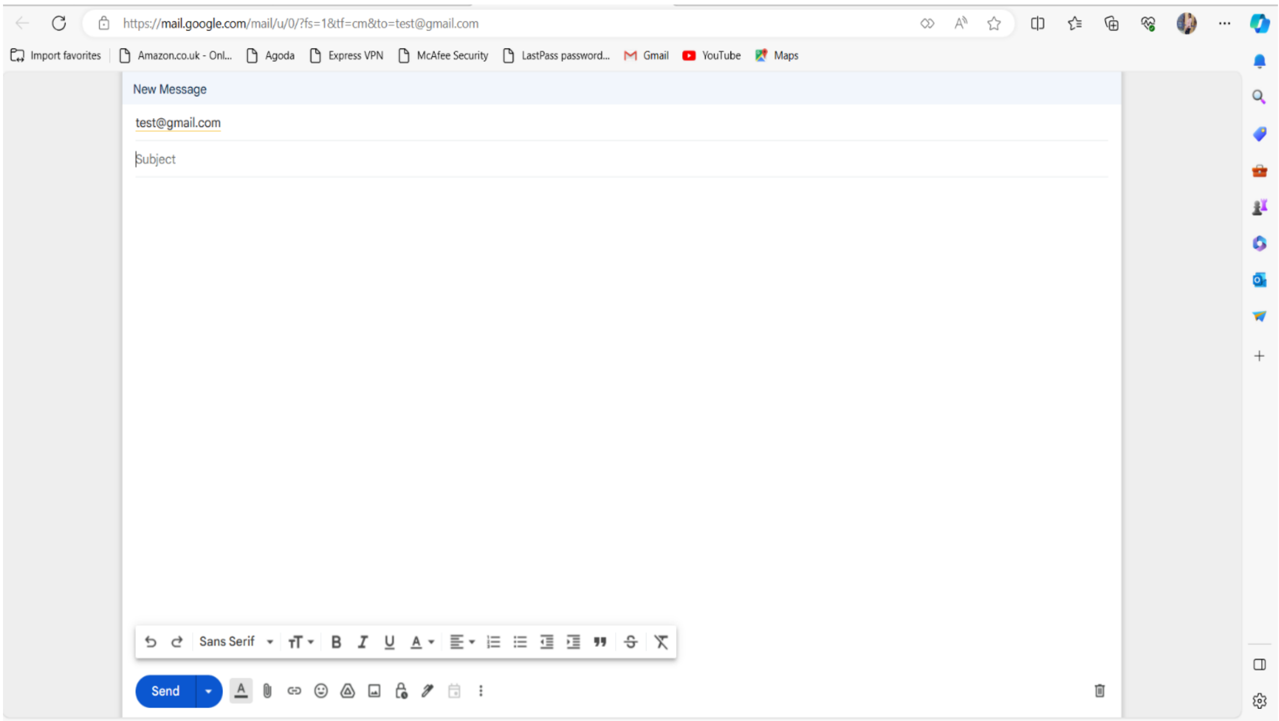
Price : 50

using organic manure rice was produced

Owner : sai

Email : sai@gmail.com

6.10 Purchase:



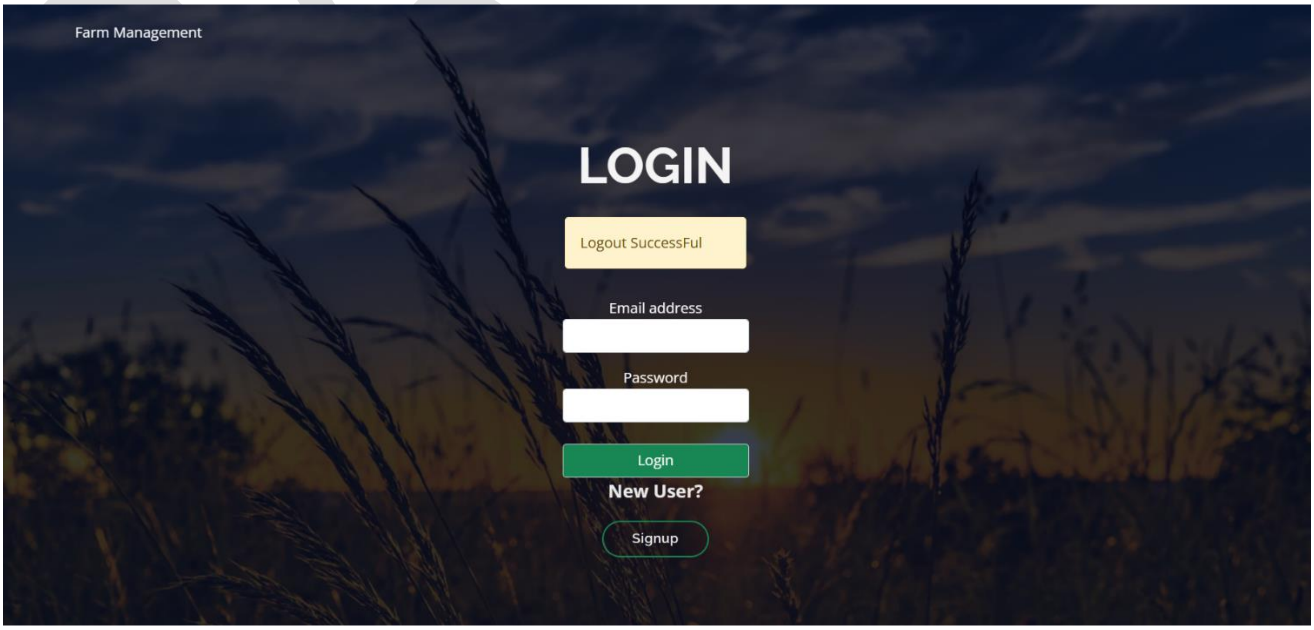
https://mail.google.com/mail/u/0/?fs=1&tf=cm&to=test@gmail.com

New Message

test@gmail.com

Subject

6.11 Logout:



Farm Management

LOGIN

Logout SuccessFul

Email address

Password

CONCLUSION

FARM MANAGEMENT SYSTEM successfully implemented based on online selling which helps us in administrating the agroproducts user for managing the tasks performed in farmers. The project successfully used various functionalities of Xampp and python flask and also create the fully functional database management system for online portals.

Using MySQL as the database is highly beneficial as it is free to download, popular and can be easily customized. The data stored in the MySQL database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQL.

With the theoretical inclination of our syllabus it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project "Farm Management System" was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project

References:

- https://www.youtube.com
- https://www.google.com