ST AGNES COLLEGE (AUTONOMOUS)

Mangalore 575002

A Project Report On

"AGROMART"

Submitted in partial fulfillment of the requirements for the

Award of Degree of

BACHELOR OF COMPUTER APPLICATIONS

2023-2024

Submitted by

Charishma (2167956)

Under the guidance of

Internal Guide

Ms Anitha Peris

St Agnes College (Autonomous)

Mangalore

External Guide

Mr Harish Kumar

Accolade Tech Solutions

Mangalore

ST AGNES COLLEGE

(AUTONOMOUS)

MANGALURU - 575 002

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled

"AGROMART" has been carried out by

Name:	**(Reg. No.)**
Charishma	2167956

Class IIIrd BCA **Semester** VI **for the year 2023-2024. Submitted in partial fulfillment of the requirements for the award of the degree of "Bachelor of Computer Applications".**

Valued at the End Semester Examination of April 2024

Examiners:

1.....

2.....

Date .....

.....

Project Guide

..... **.....**

Head of the Department **Principal**

DECLARATION

I hereby declare that this project entitled “Agromart” was prepared by me during the year 2023-2024, in partial fulfilment of the requirements for the award of the degree in Bachelor of Computer Applications.

I also declare that this project is the result of our own efforts and has not been submitted to any other university for the award of any degree.

Place: Mangalore

Date:

Name	**Reg No.**	**Signature**
Charishma	2167956	

ACKNOWLEDGEMENT

It was a wonderful and learning experience for me while working on the project “Agromart”. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of the industry. I enjoyed each and every bit of work I put into this project.

Therefore, I would like to take this opportunity to express my appreciation and gratitude to the people whose support kept me motivated throughout the journey of this project.

I sincerely wish to thank Mrs. Malavika Shetty from the Department of Computer Applications for her guidance, expertise and efforts. Thank you for your valuable advice, insights and feedback that helped me in improving my skills.

I express my deep gratitude to Mr Harish Kumar from Accolade Tech Solutions, for his guidance, proficiency and commitment. Thank you for the in-depth explanations, encouragement and for your assistance throughout this project.

I would like to thank Ms Anitha Peris, Project Guide, Department of Computer Applications for her support. Thank you for your valuable advice, insights and feedback that helped me in improving my skills.

I truly appreciate my teachers from the Department of Computer Applications for their patience and the time spent helping me in many occasions. I would like to thank them for sharing their insightful knowledge about various topics and nuances.

I am grateful to all the people involved in the development of this project for their support and motivation and for making the project successful. Without each one of you, it might not have been possible. A massive thanks to you all!

INDEX

SYNOPSIS

SYNOPSIS

1.TITLE OF THE PROJECT

Agro Mart

2. INTRODUCTION

In our innovative online agriculture ecommerce platform, we are integrating cutting-edge machine learning technology to enhance the farming experience. Leveraging a deep learning-based Convolutional Neural Network (CNN) algorithm, our system enables farmers to upload photos of leaves for precise identification of the corresponding plant species. This feature empowers farmers with real-time information about their crops, aiding in efficient agricultural management. Additionally, our platform extends beyond mere rental services, incorporating modules for farmers and sellers/companies. Farmers can seamlessly browse, select, and rent agricultural machines To further support farmers, we offer on-the-spot machine replacement, ensuring uninterrupted agricultural activities. This holistic approach leverages technology to streamline processes and promote sustainable farming practices in the digital era.

3. OBJECTIVES AND SCOPE OF THE PROJECT.

The objectives of our project encompass the development and implementation of a comprehensive online agriculture ecommerce platform, centered around the efficient rental of agricultural machines. Our primary goal is to create a user-friendly interface that facilitates seamless interaction between farmers, sellers/companies, and NGOs. The platform aims to simplify the process of renting agricultural equipment, empowering farmers with the tools they need for optimal crop management. One key objective involves the integration of a deep learning-based Convolutional Neural Network (CNN) algorithm to accurately identify plant species through leaf photos uploaded by farmers. Beyond this technological innovation, our project also seeks to establish a robust subsidy system, managed and verified by NGOs, to provide financial support in cases of losses. The scope of the project extends to encompass not only the rental process but also the broader ecosystem of agricultural management, ensuring a holistic approach to digital solutions in the agriculture sector. By focusing on user convenience, technological advancements, and comprehensive support mechanisms, our platform aspires to contribute to the sustainable growth of the agriculture industry in the digital age.

4. PROJECT CATEGORY:

Web Application/Machine Learning and RDBMS

5. TOOLS / PLATFORs

- **Visual Studio Code**
- **PHP**
- **XAMPP**
- **MySQL**
- **Jupiter notebook**
- **PyCharm**

6. SOFTWARE REQUIREMENTS

- Front end : Visual Studio Code
- Back end : My SQL 5.0, PyCharm, Jupiter Notebook
- Languages : Python ,HTML, CSS, PHP, Java Script
- Frame work : Django
- Server: IIS 7.0

7. HARDWARE REQUIREMENTS

- RAM : 8GB or above
- Hard disk : 10 GB or above
- Processor : 2.4 GHZ or above

8. MODULE DESCRIPTION

1. **User Registration Module:** This module will allow farmers, sellers, and NGOs to register on the website by providing their basic information like name, email, phone number, etc.
2. **Login Module**: This module will allow registered users to log in to the website by providing their login credentials. The module should also include password recovery and account deletion functionality.
3. **Farmer Dashboard Module**: This module will allow farmers to view available rental machines, place orders, view order history, and apply for subsidies if they are in loss.
4. **Seller Dashboard Module:** This module will allow sellers to add their machines to the website, view orders, and manage their inventory.

5. **NGO Dashboard Module:** This module will allow NGOs to view subsidy applications, verify documents, and approve subsidies and generating reports.
6. **Additional Machine/Equipment's Selling Module:** This module will allow farmers/users to sell their additional machines for rent on the website.
7. **Notifications Module:** This module will allow users to receive notifications about their orders, subsidies, and other important updates.
8. **Admin Panel Module:** This module will allow the website administrator to manage users, orders, subsidies, and other website settings.
9. **EKYC module:** This module includes features for verifying the identity of users, such as uploading ID proof and using biometric authentication.
10. **Order Tracking:** This module includes features for tracking the orders placed by the users
11. **Online Payment Gateway:** A secure virtual bridge that facilitates electronic transactions between farmers and users. It encrypts sensitive information, ensuring a safe and seamless transfer of funds for online purchase.
12. **Job Post:** Advertisement that outlines the details of job opening including job description and qualification and application instruction.
13. **User profile management:** This module allows users to edit their profile information, such as their name, contact information, and payment details.
14. **Image Recognition:** In this module, we've integrated an image recognition feature using a CNN model. When users upload leaf images, the system employs the CNN algorithm to predict and identify the plant species. This enhances the platform's ability to give accurate information about the submitted leaves.

9. FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT.

- Incorporate Internet of Things (IoT) devices and sensors into agricultural machinery to gather real-time data on crop conditions, soil health, and machine performance. This data can be leveraged for more informed decision-making by farmers.
- Establish a robust feedback mechanism to gather input from users, including farmers, sellers, and NGOs. This feedback can be instrumental in identifying areas for improvement and tailoring the platform to better meet the needs of its users.

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

SOFTWARE REQUIREMENTS SPECIFICATION

2.1. INTRODUCTION

The software development process commences with the meticulous gathering of requirements and thorough analysis. Once the requirements are clearly and accurately defined, they are documented in a Software Requirements Specification (SRS). This comprehensive document serves as a guide for developers, providing them with a detailed outline of the functionalities and features the software should possess. Simultaneously, the testing team relies on the SRS to gain a clear understanding of the expected system behaviour. The document meticulously delineates all the essential requirements that the developed system is intended to fulfil.

2.2 OVERALL DESCRIPTION

2.2.1 PRODUCT PERSPECTIVE

Product perspective is the relationship of product of other product defining if the product is independent or is a part of large product. The architecture which is used in the project is 3 tier architecture with presentation layer, application layer, and product layer. In our innovative online agriculture ecommerce platform, we are integrating cutting-edge machine learning technology to enhance the farming experience. Leveraging a deep learning-based Convolutional Neural Network (CNN) algorithm, our system enables farmers to upload photos of leaves for precise identification of the corresponding plant species. This feature empowers farmers with real-time information about their crops, aiding in efficient agricultural management. Additionally, our platform extends beyond mere rental services, incorporating modules for farmers, sellers/companies, and NGOs. Farmers can seamlessly browse, select, and rent agricultural machines, while NGOs play a crucial role in verifying subsidy applications for any incurred losses. To further support farmers, we offer on-the-spot machine replacement, ensuring uninterrupted agricultural activities. This holistic approach leverages technology to streamline processes and promote sustainable farming practices in the digital era.

2.2.2 PRODUCT FUNCTION

- User Registration Module
- Login Module
- Farmer Dashboard Module

- Seller Dashboard Module
- NGO Dashboard Module
- Machine Replacement Module
- Additional Machine/Equipment's Selling Module
- Notifications Module
- Admin Panel Module
- EKYC module
- Order Tracking module
- Online Payment Gateway
- Job Post
- User profile management
- Image Recognition

2.2.3 USER CHARACTERISTICS

The expected users of Farmer Store are mostly involved in the product development projects. These users are well-familiar with Windows Operating System and have basic understanding of the Software Development Life Cycle. Typical user set include developers, software architects, quality assurance personnel, graphics designers, requirements specialists, and a variety of managerial personnel.

Admin:

- The admin has the authority to add and manage farmers, as well as add and manage product categories.
- Additionally, the admin can view orders placed by customers.

Users:

- Customers can access the system by logging in, which enables them to search for products of interest and add them to their cart.
- They can then complete purchases using an online payment gateway.
- Additionally, customers have the ability to review their purchased products and view their past order history and payment details.

- Moreover, they can also access job posts posted by farmers.

Farmers:

- Farmers have the capability to add products grown by them to the system.
- They can also view order details and payment information pertaining to their sales. Additionally, farmers have the option to post job listings within the system.
- Furthermore, they can apply for subsidy applications and track the status of their applications.

NGO:

- NGOs can log in to the system and have the ability to view subsidy request applications submitted by farmers.
- They can manage these requests by either approving or rejecting the applications based on their assessment and criteria.

2.2.4 GENERAL CONSTRAINTS

- Requires all the mandatory fields to be filled with the proper information.

2.2.5 ASSUMPTIONS

- User should be familiar with Basic computer knowledge.
- The system is completely dependent on internet connection.
- The Information provided by the student is assumed to be genuine.

2.3. SPECIAL REQUIREMENTS

2.3.1 SOFTWARE REQUIREMENTS

- Front end: Visual Studio Code
- Back end: My SQL 5.0, PyCharm, Jupiter Notebook
- Languages: Python, HTML, CSS, PHP, Java Script
- Frame work: Django
- Server: IIS 7.0

2.3.2 HARDWARE REQUIREMENTS

- RAM: 8GB or above
- Hard disk: 10 GB or above
- Processor: 2.4 GHZ or above

2.4. FUNCTIONAL REQUIREMENT

1. **User Registration Module:** This module will allow farmers, sellers, and NGOs to register on the website by providing their basic information like name, email, phone number, etc.

   - **Input:** username, password, confirm password, user type, phone, email.
   - **Output:** A new user is created.
   - **Processing:** If user already exist appropriate message is displayed.

2. **Login Module**: This module will allow registered users to log in to the website by providing their login credentials. The module should also include password recovery and account deletion functionality.

   - **Input:** user name, password.
   - **Output:** User logged in.
   - **Processing:** If user name and password is invalid, error message is displayed

3. **Farmer Dashboard Module**: This module will allow farmers to view available rental machines, place orders, view order history, and apply for subsidies if they are in loss.

4. **Seller Dashboard Module:** This module will allow sellers to add their machines to the website, view orders, and manage their inventory.

5. **NGO Dashboard Module:** This module will allow NGOs to view subsidy applications, verify documents, and approve subsidies and generating reports.

6. **Machine Replacement Module:** This module will allow farmers to request a replacement for a faulty machine.

7. **Additional Machine Selling Module:** This module will allow farmers to sell their additional machines for rent on the website.

8. **Notifications Module:** This module will allow users to receive notifications about their orders, subsidies, and other important updates.

9. **Admin Panel Module:** This module will allow the website administrator to manage users, orders, subsidies, and other website settings.

10. **EKYC module:** This module includes features for verifying the identity of users, such as uploading ID proof and using biometric authentication.

11. **Tracking module:** This module includes features for tracking the location and status of machines, such as using GPS and RFID technology.

12. **User profile management:** This module allows users to edit their profile information, such as their name, contact information, and payment details.

13. **Safety module:** This module includes features for ensuring the safety of users, such as monitoring user behavior, implementing security measures, and providing safety tips.

14. **Image Recognition:** In this module, we've integrated an image recognition feature using a CNN model. When users upload leaf images, the system employs the CNN algorithm to predict and identify the plant species. This enhances the platform's ability to give accurate information about the submitted leaves.

2.5. DESIGN CONSTRAINTS

- Requires specifying the information for all the mandatory fields
- The application shall have a relational database.
- The application shall be implemented using Python, JavaScript & PHP
- The application shall display error messages to the user when an error is detected

2.6. SYSTEM ATTRIBUTES

- **Availability:** Agromart shall be available in internet 24x7 and capable of supporting a multiple login.
- **Security:** Agromart shall be managed by the administrator via predetermined roles
- **Maintainability:** During maintenance stage, the SRS can be referred for the validation
- **Portability: since** it is a system, it is portable.
- **Timeliness:** The system carries out all the operations with consumptions of very less time

2.7. OTHER REQUIREMENTS

None

SYSTEM DESIGN

System Analysis and Design

3.1 Introduction

System Analysis:

System design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

In System design focus is on deciding which modules are needed for the system, the specifications of these modules should be interconnected is called system design.

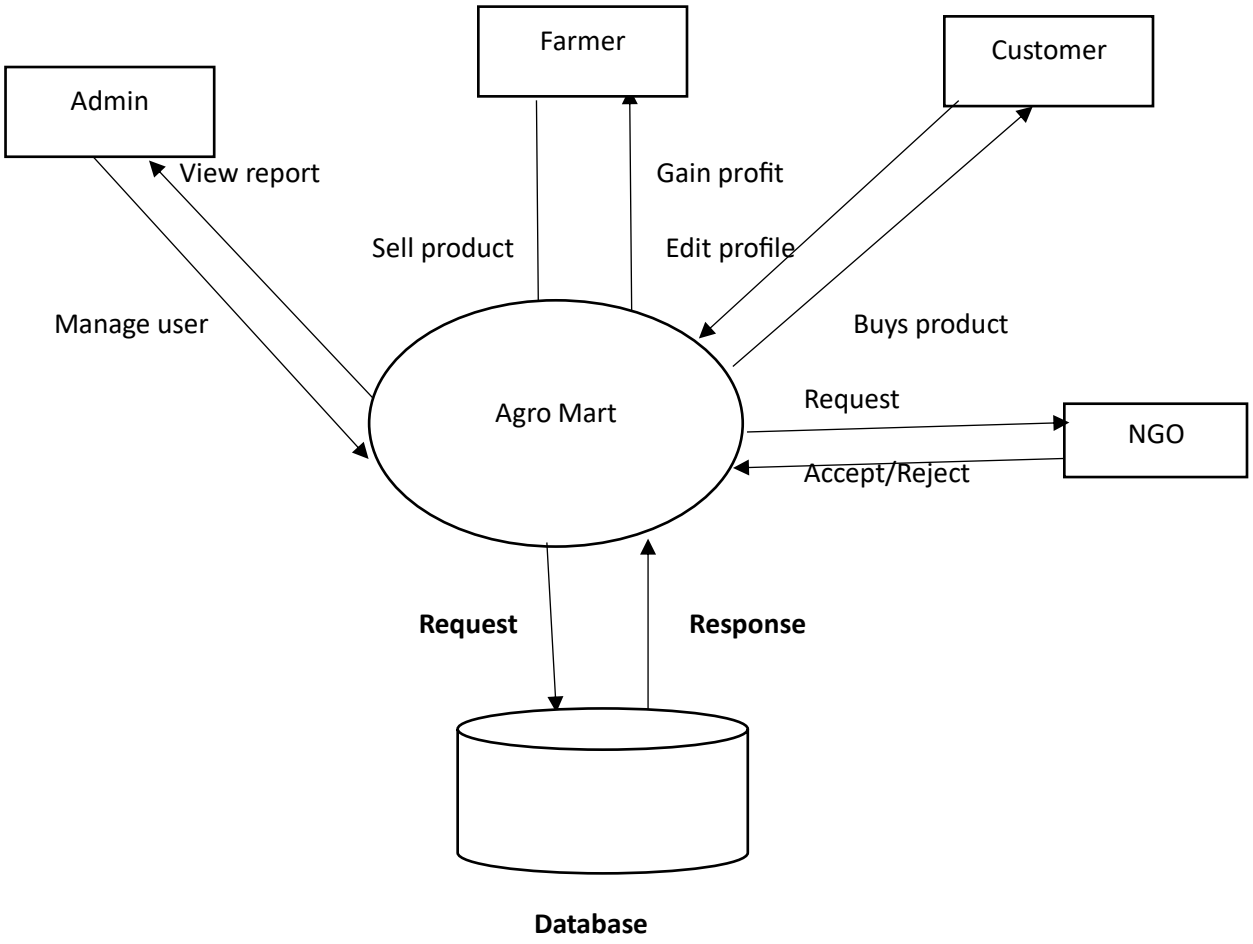
System Design:

System design is also called top-level design. Here we consider a system to be set of components with clearly defined behavior that interact with each other in a fixed manner to produce some behavior. In a system design, the design consists of module definitions, with each module supporting a functional abstraction.

3.2 Context Flow Diagram:

It is common practice to draw the context-level data flow diagram first, which shows the interaction between the system and external agents which acts as data source and data sinks. On the context diagram the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire systems the single process, and gives no clues as to its internal organization.

3.2 Context Flow Diagram:



Context Flow Diagram

Context Level DFD

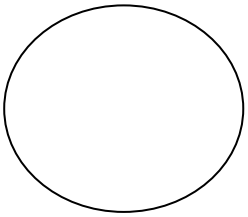
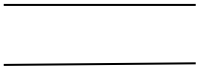
In my project, the general users can buy products like fresh fruits and vegetables directly from the farmer through online bookings. The farmer can add products and post jobs that require laborers which will be visible to the general user through his/her user interface. The farmers also has the plant detection (Machine Learning) feature which allows them to predict the names of the plants just by scanning the leaves. The administrator can post articles and manage the users registered to the website. The NGO provides subsidy to the farmers in need if they register themselves to the NGO. The verification of the farmers will be done through EKYC process.

3.3 Data Flow Diagram:

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing. It is common practice for a designer to draw a context- level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

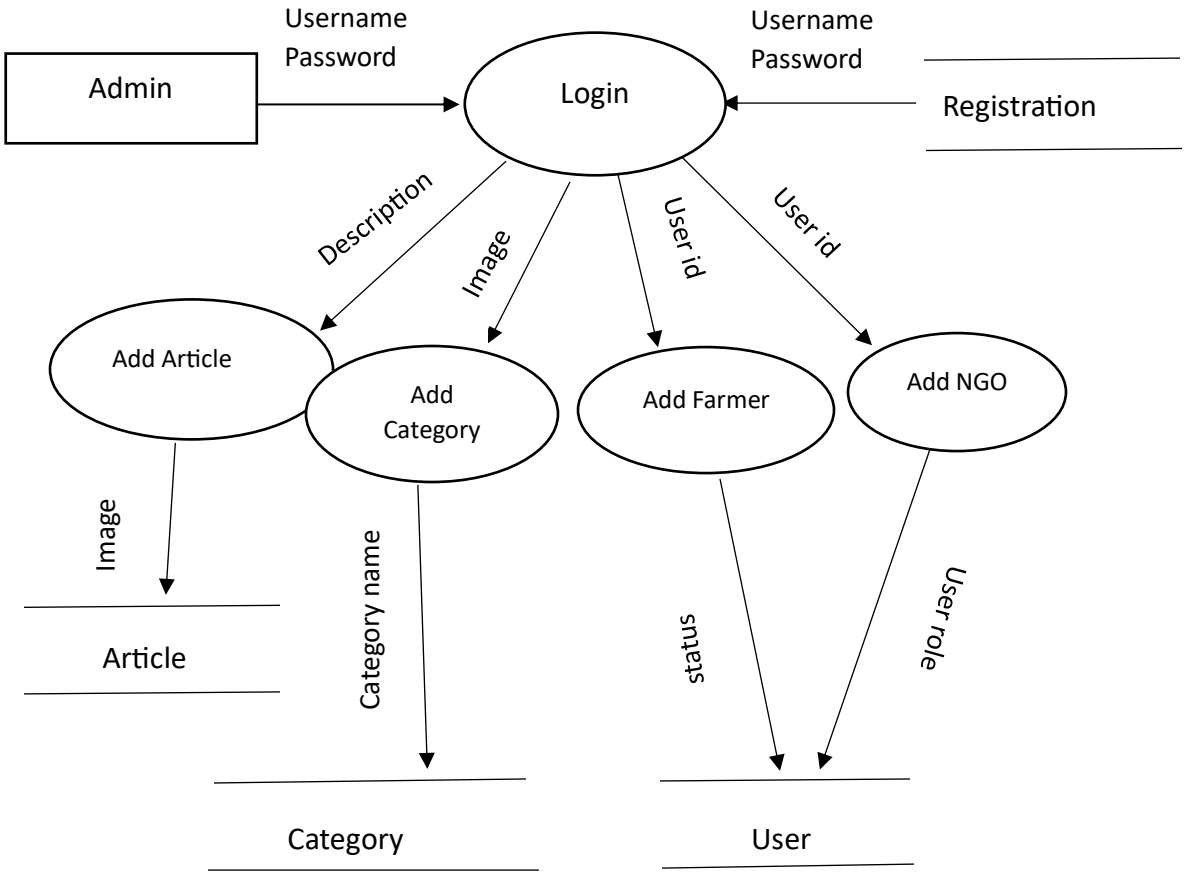
A DFD represents flow of data through a system. Data flow diagrams are commonly used during problem analysis. It views a system as a function that transforms the input into
desired output. A DFD shows movement of data through the different transformations or processes in the system.

Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram. The appropriate register saved in database and maintained by appropriate authorities.
In the normal convention, logical DFD can be completed using some notations.

DIAGRAM	DESCRIPTION
	Represents Source or destination of data
	Represents a process that transforms Incoming data into Outgoing flows
	Represents data flow
	Represents data stores

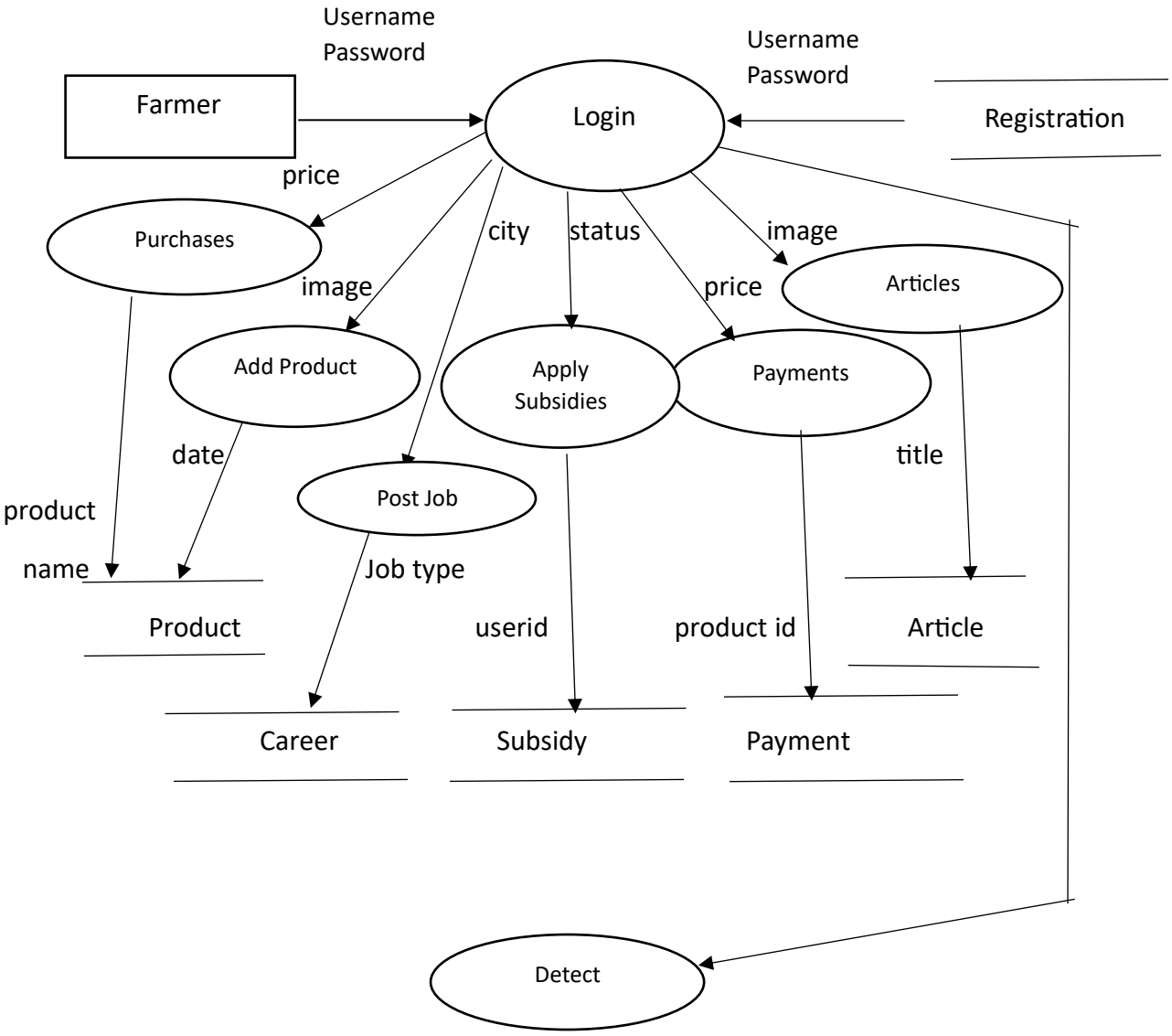
3.4.1 Level 1 Data Flow Diagram

administrator has to enter the valid username and password in order to login to the system. These username and passwords are compared with the one which is stored in the backend, whether he is administrator or not. If the admin is logged in successfully he/she can navigate to the modules that are only accessible by the administrator. The modules that are accessible to administrator are adding articles, adding categories, adding farmers and NGO along with viewing all these modules and managing the users. Finally, all the details are stored in the database if and only if all the entries are valid and appropriate as per the requirement.



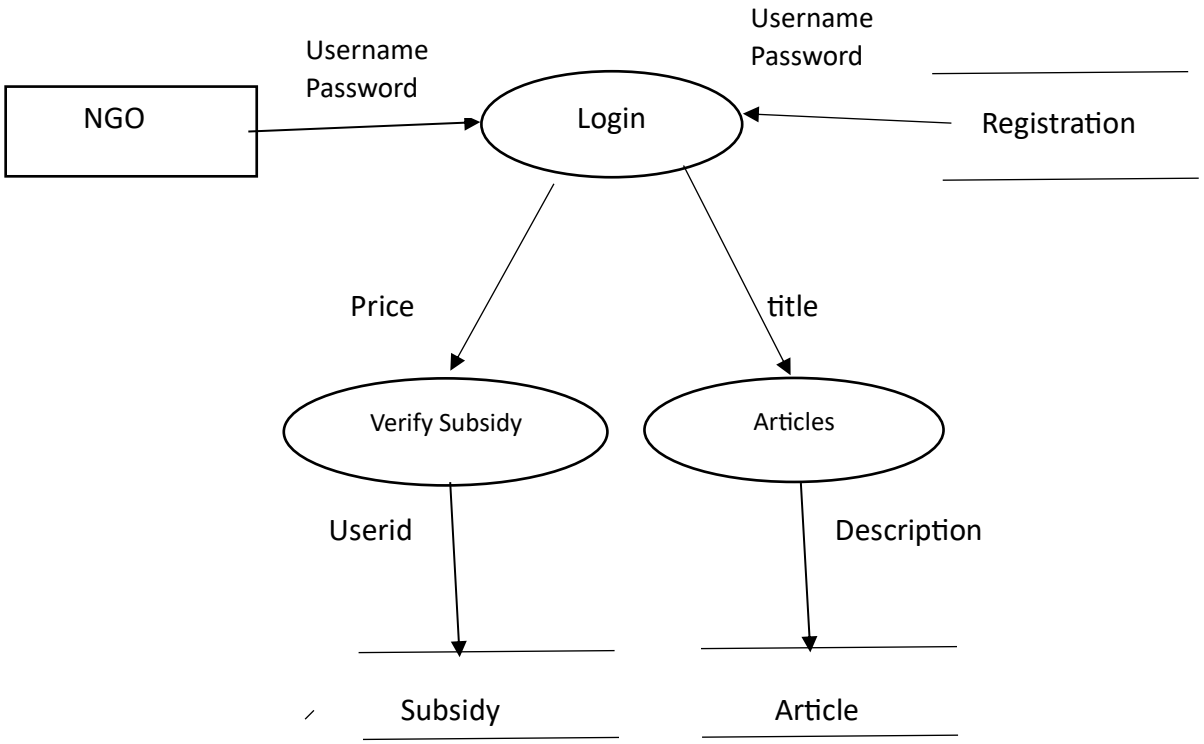
3.4.2 Level 2 Data Flow Diagram

The farmer has to enter the valid username and password in order to login to the system. These username and passwords are compared with the one which is stored in the backend, whether he is user or not. If the user is logged in successfully, he/she can navigate to the modules that are only accessible by the farmer. The modules that are accessible to the farmers are adding products, posting jobs, viewing the orders placed by the customers and their payment details, detecting the plants by scanning their leaves, applying for subsidies to the NGO and viewing articles posted by the administrator. Finally, all the details are stored in the database.



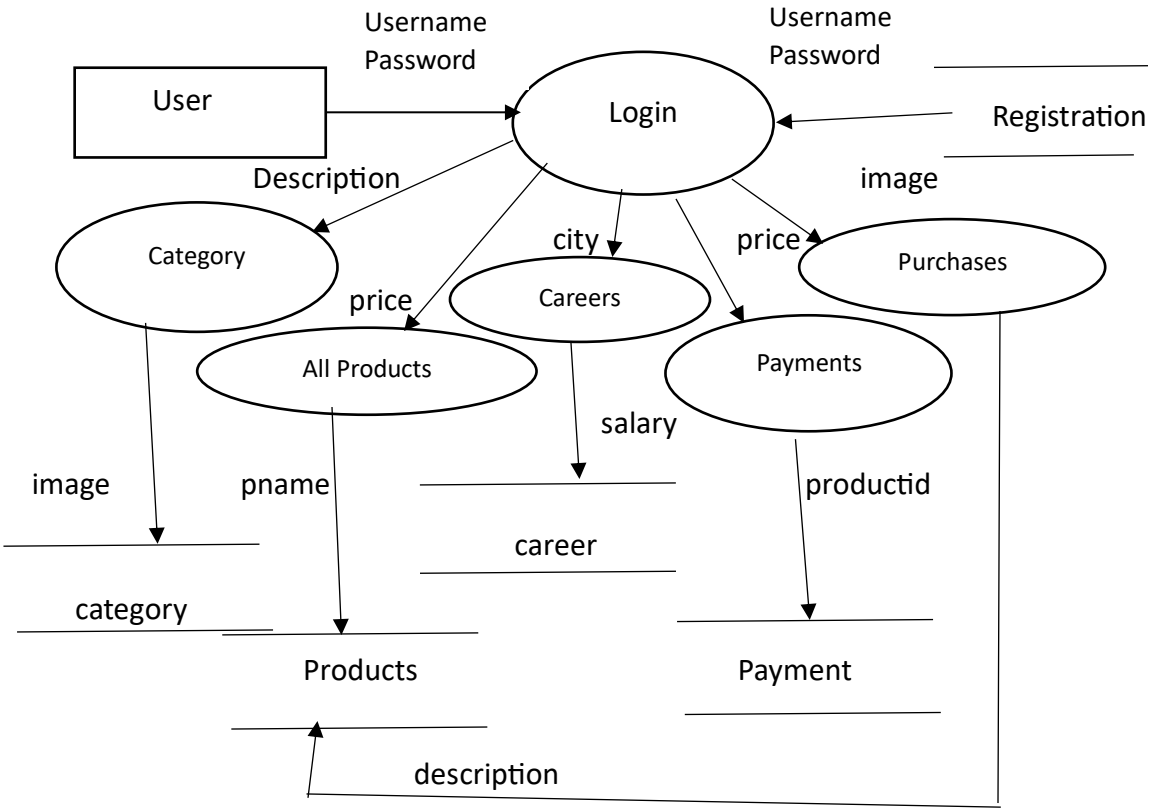
3.4.3 Level 3 Data Flow Diagram

The NGO staff has to enter the valid username and password in order to login to the system. These username and passwords are compared with the one which is stored in the backend, whether he is user or not. If the user is logged in successfully he/she can navigate to the modules that are only accessible by the NGO user. The modules that are accessible to the NGO staffs are subsidy verification and viewing the articles posted by the administrator. Finally, all the details are stored in the database.



3.4.4 Level 4 Data Flow Diagram

The user has to enter the valid username and password in order to login to the system. These username and passwords are compared with the one which is stored in the backend, whether he is user or not. If the user is logged in successfully, he/she can navigate to the modules that are only accessible by the user. The modules that are accessible to the users are viewing and placing an order for the products, viewing job opportunities details, viewing the purchase and payment details. Finally, all the details are stored in the database.


Username
Password
Username
Password
User
Login
Registration
Description
image
Category
city
price
Purchases
price
Careers
All Products
Payments
salary
image
pname
productid
career
category
Products
Payment
description

DATABASE DESIGN

4.1 Introduction:

Database Design

Database description describes all the databases used in the software to store all the records. The database in turn is further described in detail giving all the fields used with their data type, constraints available to them and description. Constraints include primary key, foreign key, etc., which allow the entities to be uniquely identified

In this database description we describe all databases which are used to store all the records of the Train Seat Scheduling System.

4.2 Purpose:

- Database description describes the entire database used in the software to store all the records.
- Database design is the process of producing a detailed data model of a database. Database design is a collection of related data.
- This document describes standards to use when designing and developing the database.

4.3 Scope:

A good database is one that is simple to understand and well planned. The database doesn't have redundant tables. One can use ER Diagram (Entity Relationship Diagrams) in order to make a good database. This database design is used to understand the software hotel KOT and billing

- Organize the system into modules
- Organize sub-modules for each module
- Allocate tasks to processors
- Choose an approach to manage data store
- Handle access to global resources
- Choose implementation logic.

4.4 Normalized Tables:

Table name: User Table

Column Name	Data Type	Length	Constraints
Id	Integer	100	Primary Key
Name	Varchar	100	Not null
Contact	Varchar	12	Not null
Email	Varchar	100	Not null
Password	Varchar	25	Not null

Date	Datetime	-----	Not null
Status	Tinyint	01	Not null
Userrole	Varchar	10	Not null

Table name: Article Table

Column Name	Data Type	Length	Constraints
Id	Integer	10	Primary
Title	Varchar	100	Not null
Description	Text	-----	Not null
Date	Datetime	-----	Not null
Image	Varchar	100	Not null
Status	Tinyint	01	Not null

Table name: Career Table

Column Name	Data Type	Length	Constraints
Id	Integer	10	Primary
Title	Varchar	100	Not null
Description	Text	-----	Not null
JobType	Varchar	50	Not null
Date	Datetime	-----	Not null
Status	Tinyint	01	Not null
City	Varchar	25	Not null
Salary	Varchar	25	Null
FarmerId	Integer	10	Not null

Table name: Category Table

Column Name	Data Type	Length	Constraints
Id	Integer	10	Primary
Category_Name	Varchar	50	Not null
Description	Text	-----	Not null
Status	Tinyint	1	Not null

Date	Datetime	-----	Not null
Image	Varchar	50	Not null
Color	Varchar	10	Not null

Table name: Payment Table

Column Name	Data Type	Length	Constraints
Id	Integer	10	Primary
UserId	Integer	10	Not null
ProductId	Integer	10	Not null
Payment_Number	Varchar	25	Not null
Status	Tinyint	01	Not null
Date	Datetime	-----	Not null
Price	Varchar	15	Null

Table name: Product Table

Column Name	Data Type	Length	Constraints
Id	Integer	10	Primary
Pname	Varchar	50	Not null
Description	Text	-----	Not null
Image	Varchar	100	Not null
Price	Varchar	10	Not null
PriceType	Varchar	50	Not null
Date	Datetime	-----	Not null
Status	Tinyint	01	Not null
UserId	Integer	10	Not null
CategoryId	Integer	10	Not null

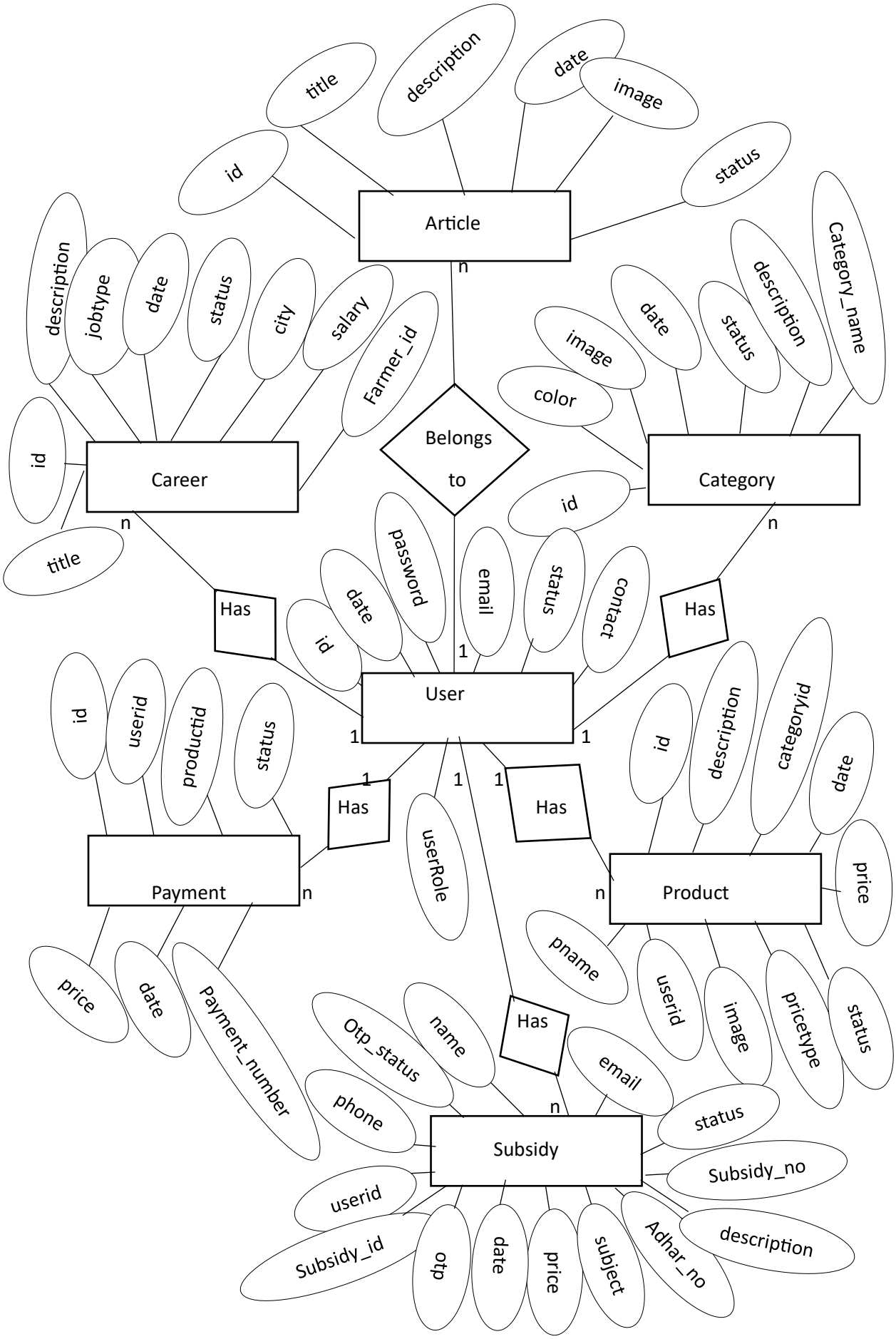
Table name: Subsidy Table

Column Name	Data Type	Length	Constraints
Subsidy_Id	Integer	11	Primary
Subsidy_No	Varchar	50	Not null

User_Id	Integer	11	Not null
Name	Varchar	50	Not null
Price	Varchar	50	Not null
Subject	Varchar	100	Not null
Adhar_No	Varchar	100	Not null
Phone	Varchar	25	Not null
Email	Varchar	50	Not null
Description	Varchar	500	Not null
Status	Varchar	50	Not null
Date	Date	-----	Not null
Otp_Status	Varchar	50	Not null
Otp	Varchar	50	Not null

4.1.1 ER Diagram:

ERD stands for entity relationship diagram. People also call these types of diagrams ER diagrams and Entity Relationship Models. An ERD visualizes the relationships between entities like people, things, or concepts in a database. An ERD will also often visualize the attributes of these entities. By defining the entities, their attributes, and showing the relationships between them, an ER diagram can illustrate the logical structure of databases. This is useful for engineers hoping to either document a database as it exists or sketch out a design of a new database.


Article
id
title
description
date
image
status
Career
id
title
description
jobtype
date
status
city
salary
Farmer_id
Belongs to
Category
id
image
color
date
status
description
Category_name
Has
Has
User
id
password
email
status
Contact
date
Payment
id
userid
productid
status
Has
Has
Has
Product
id
description
categoryid
date
price
pname
userid
image
pricetype
status
Has
Subsidy
Otp_status
name
email
phone
status
Subsidy_no
userid
Subsidy_id
otp
date
price
subject
Adhar_no
description
Payment_number
price
date
userRole
1
n

DETAILED DESIGN

Detailed Design

5.1 Introduction:

Detail design starts after the system design phase is completed. The goal of this phase is to develop the internal logic of each of the models identified during the system design.

Before deciding on the logic of the module, formal specification of the module maybe developed. The specification should be such that they are complete, unambiguous and precise and they do not suggest any particular implementation. Two modules units are frequently chosen for formal specification. Functional modules and data abstraction modules.

Every system requires not only data, but also the structure of that data. A database management system (DBMS) collects and structures related files so that many users can retrieve, manipulate and store data. Here we use MS-SQL as the DBMS. Database is a collection of inter related data stored with minimum redundancy to serve many users quickly and efficiently. Database designs are designed to manage large bodies of information and also for easy and flexible retrieval of data.

5.2. STRUCTURE OF THE SOFTWARE PACKAGE:

Admin:

- The admin has the authority to add and manage farmers, as well as add and manage product categories.
- Additionally, the admin can view orders placed by customers.

Users:

- Customers can access the system by logging in, which enables them to search for products of interest and add them to their cart.
- They can then complete purchases using an online payment gateway.
- Additionally, customers have the ability to review their purchased products and view their past order history and payment details.
- Moreover, they can also access job posts posted by farmers.

Farmers:

- Farmers have the capability to add products grown by them to the system.
- They can also view order details and payment information pertaining to their sales. Additionally, farmers have the option to post job listings within the system.

- Furthermore, they can apply for subsidy applications and track the status of their applications.

NGO:

- NGOs can log in to the system and have the ability to view subsidy request applications submitted by farmers.
- They can manage these requests by either approving or rejecting the applications based on their assessment and criteria.

5.3 MODULAR DECOMPOSITION OF THE COMPONENTS:

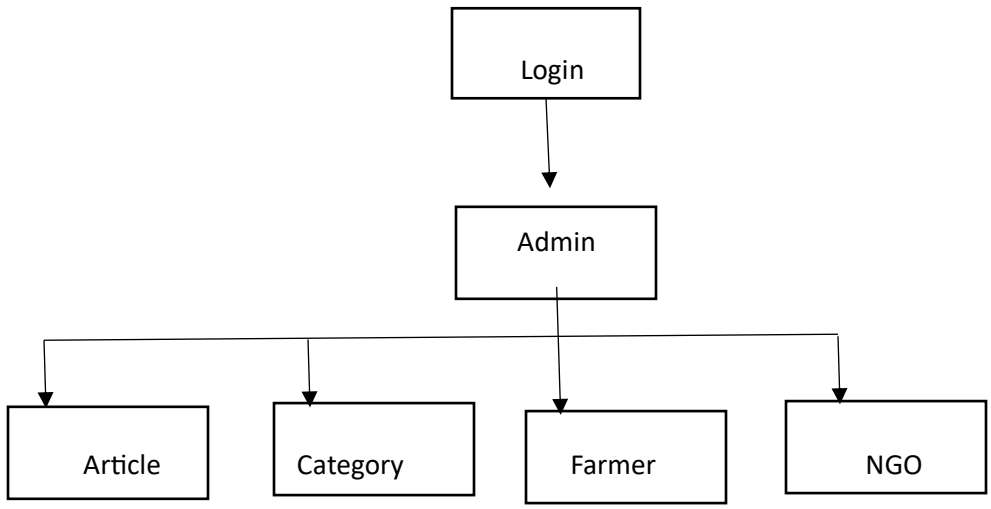
5.3.1 Admin Module

5.3.1.1 Identification of the modules:

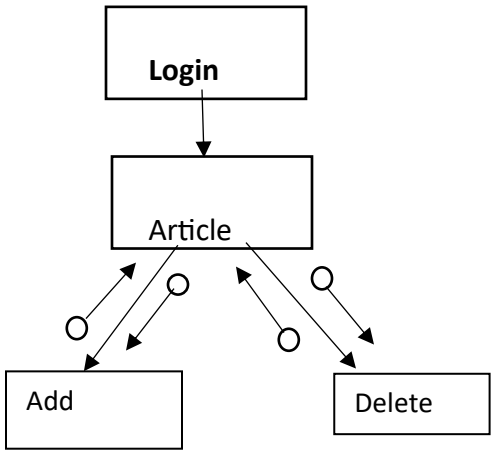
The modules identified in this component are:

- Login
- Home
- Article
- Category
- Farmer
- Manage User
- Setting
- Logout

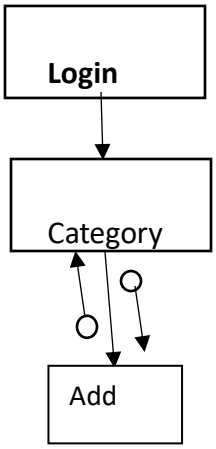
5.3.1.2: Structured Chart for the Admin_Module



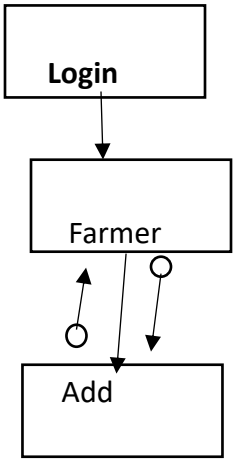
Article:



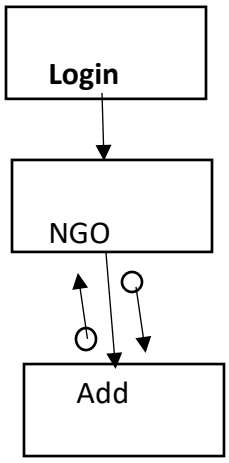
Category:



Farmer:



NGO:



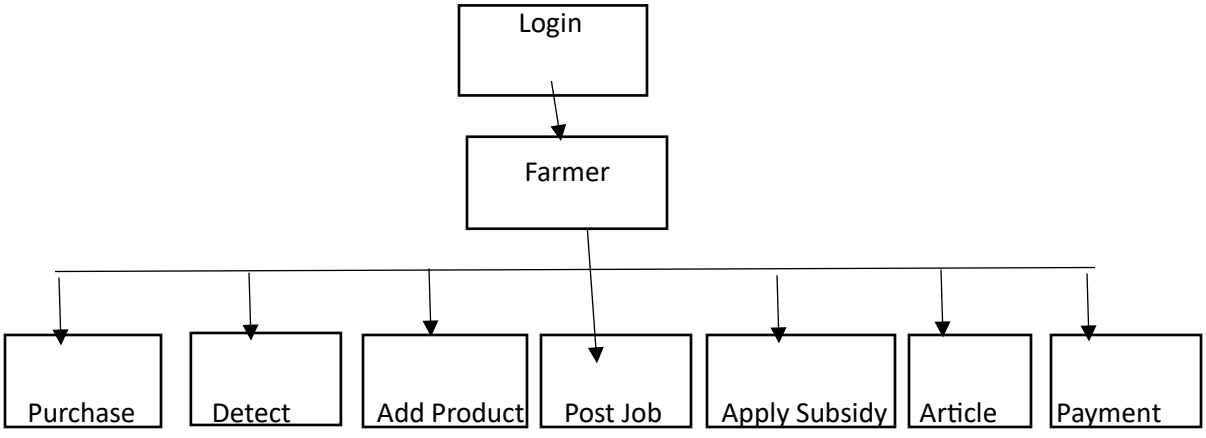
5.3.2: Farmer Module

5.3.2.1 Identification of the modules:

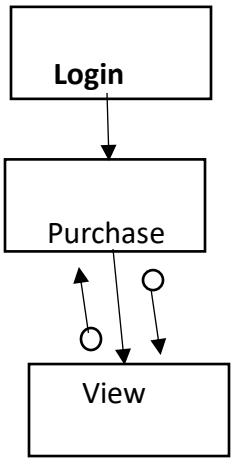
5.3.2.2 The modules identified in this component are:

- Login
- Purchase
- Detect
- Add Product
- Post Job
- Apply subsidy
- Article
- Payment
- Logout

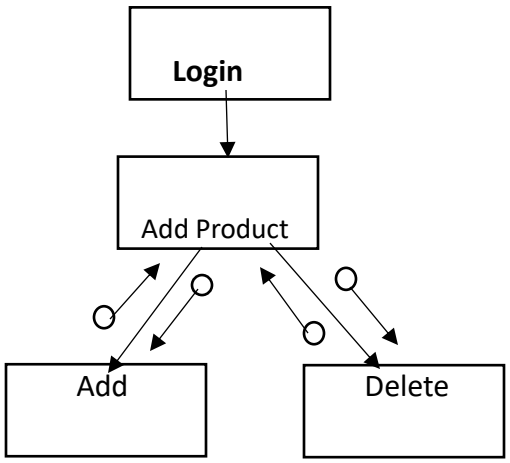
5.3.2.2: : Structured Chart for the Farmer Module



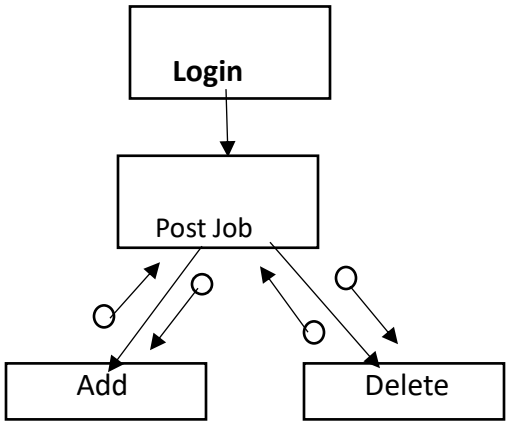
Purchase:



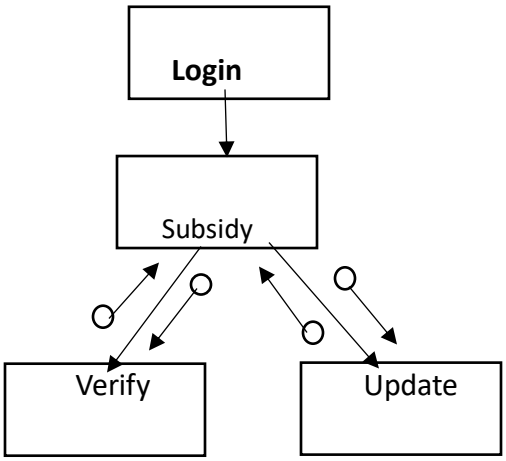
Add Product:



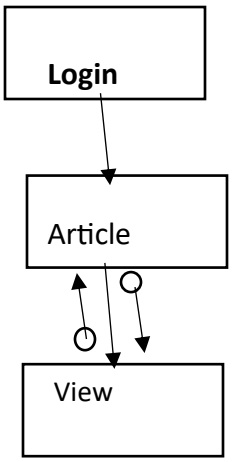
Post Job:



Apply Subsidy:



Article:



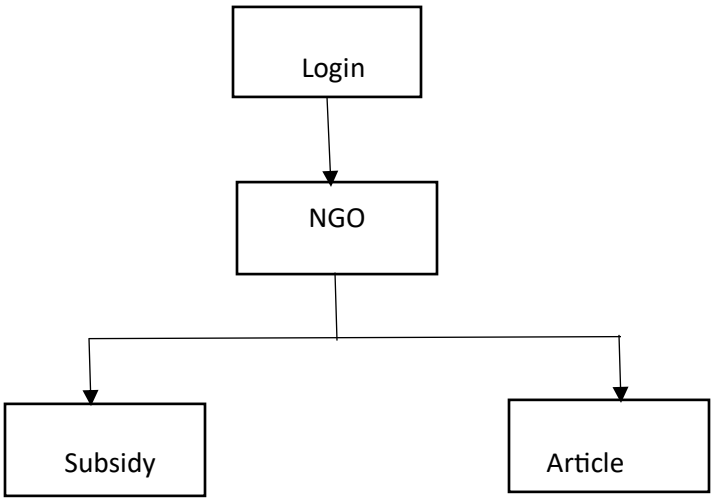
5.3.3: NGO Module

5.3.3.1 Identification of the modules:

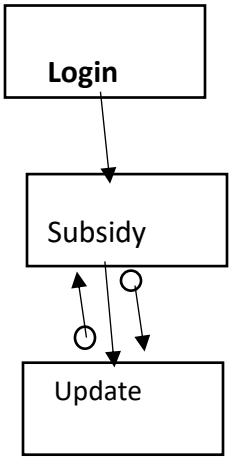
The modules identified in this component are:

- Login
- Subsidy
- Article
- Logout

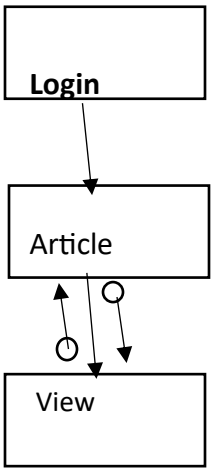
5.3.2.2: : Structured Chart for the NGOs Module



Subsidy:



Article:



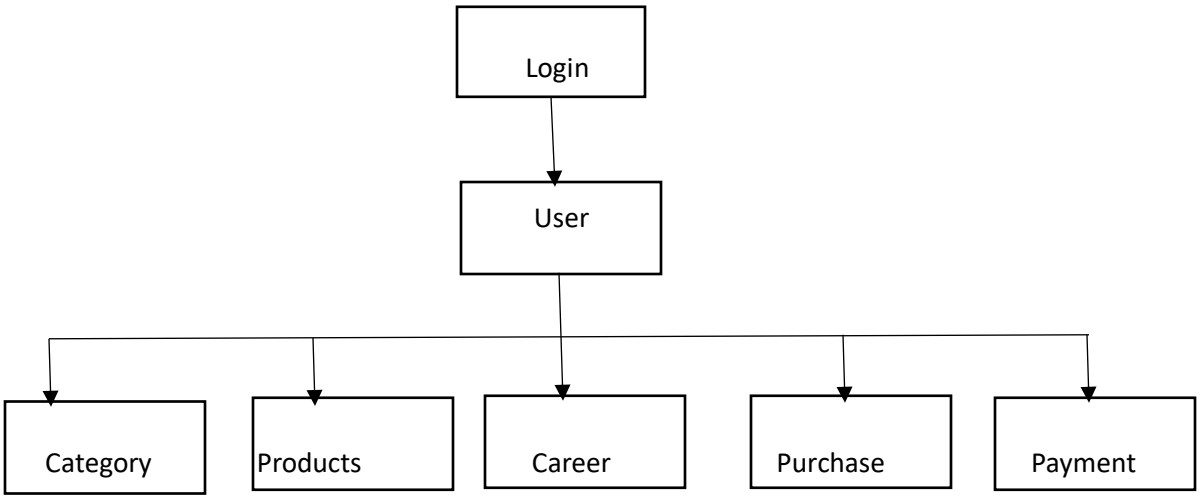
5.3.4: User Module

5.3.4.1 Identification of the modules:

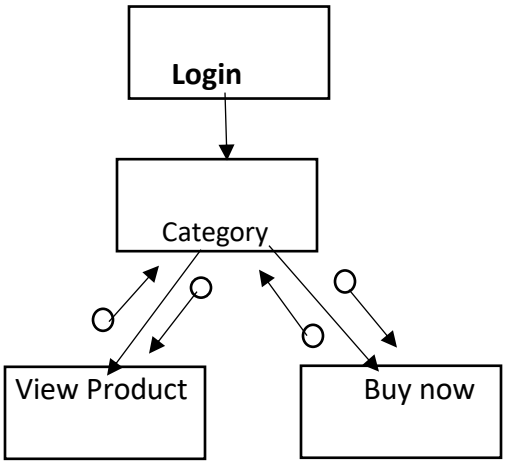
The modules identified in this component are:

- Login
- Home
- Category
- Products
- Purchase
- Payments
- Logout

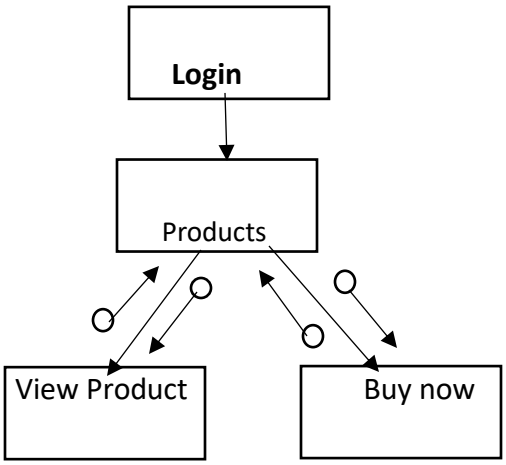
5.3.4.2 Structured Chart for the NGOs Module



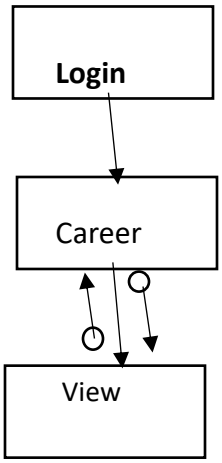
Category:



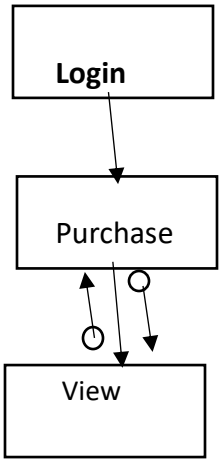
Products:



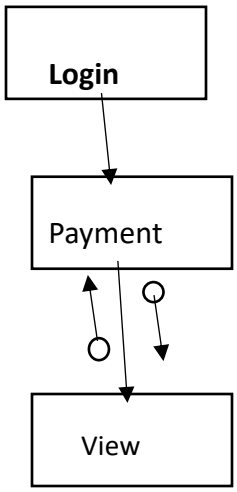
Career:



Purchase:



Payment:



PROGRAM CODE LISTING

CODING

6.1 Introduction

The goal of coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and that performs the computation specified by the design. The coding phase affects both testing and maintenance profoundly. The goal during this phase is not to simplify the job of the programmer. Rather, the goal should be simplify the job of the tester and the maintainer.

There are many different criteria for judging a program, including readability, size of the program, execution time and required memory. Having readability and understanding as a clear objective of the coding activity can itself help in producing software that is more maintainability.

6.2 Programming Style

It is impossible to provide an exhaustive list of what to do and what not to do produce simple readable code. Being able to do this will amount to providing an algorithm for writing good code. Next we will list some general rules that usually apply.

6.2.1 Names

Selecting module and variable names often not considered important by novice programmers. Only when one starts reading programs written by other where the variable names are cryptic and not representative does one realize the importance of selecting proper names. Most variables in a program reflect some entity in the problem domain, and the modules reflect some process. Variable names should be closely related to the entity they represent, and module name should reflect their activity. It is bad practice to choose cryptic names just to avoid typing or totally unrelated names. It is also bad practice to use the same name for multiple purposes.

6.2.2 Control Constructs

It is desirable that as much as possible single-entry, single exit constructs be used. It is also desirable to use the few standard control constructs rather than using a wide variety of constructs, just because they are available in the language.

6.2.3 Information Hiding

Information hiding should be supported where possible. Only the access function for the data structures should be made visible while hiding the data structure behind these functions.

6.2.4 User-Defined Types

Modern languages allow users to define types like the Enumerated type. When such facilities are available, they should be exploited where applicable. For example, when working with dates, the type can be defined for the day of the week.

6.2.5 Nesting

The different control constructs, the if-then-else, can be nested. If the nesting becomes too deep, the programs become harder to understand. In case of deeply nested if-then-else, it is often difficult to determine then if statement to which a particular else clause is associated. Wherever possible, deep nesting should be avoided, even if it means a little inefficiency.

6.2.6 Module Size

We discussed this issue during system design. A programmer should carefully examine any routine with very few statements (say fewer than 5) or with too many statements (say more than 50). Large modules often will not be functionally cohesive and too small modules might incur unnecessary overhead. There are can be no hard-and-fast rule about module sizes the guiding principle should cohesion and coupling.

6.2.7 Module Interface

A module with a complex interface should be carefully examined. Such modules might not be functionally cohesive and might be implementing multiple functions. As a rule of thumb, any module whose interface has more than five parameters should be carefully examined and broken into multiple modules with simpler interface if possible.

6.2.8 Program Layout

How the program is organized and presented can have great effect on the readability of it. Proper indentation, blank spaces, and parentheses should be used to enhance the readability of programs. Automated tools are available to “pretty print” a program, but it is good practice to have a clear layout of programs.

6.2.9 Side Effects

When a module is invoked, it sometimes has side effect of modifying the program state beyond the modifications of parameters listed in the module interface definition, for example, modifying global variables. Such side effects should be avoided where possible, and if a module has side effects, they should be properly documented.

6.2.10 Robustness

A program is robust if it does something planned even for exceptional conditions. A program might encounter exceptional conditions in such forms as incorrect input, the incorrect value of some variable, and overflow. In general, a program should check for validity inputs, where possible, and should check for possible overflow of the data structures. If such situations do arise, the program should not just “crash” it should produce some meaningful message and exit gratefully.

6.3 CODING

Configuration File: config.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "agromart";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
{
   die("Connection failed: " . mysqli_connect_error());
}
?>
```

Registration:register.php

```
<?php
session_start();
require_once 'include/config.php';
date_default_timezone_set('Asia/Kolkata');
?>
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style.css">
<link rel="stylesheet"href=https://cdnjs.cloudflare.com/ajax/libs/font -awesome /4.7.0/css/font-awesome.min.css>
    <title>Agro Mart - Register</title>
    <style>
* {
    box-sizing: border-box;
}
body {
    background: #ffe369;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    font-family: 'Montserrat', sans-serif;
    height: 100vh;
    margin: -20px 0 50px;
}
h1 {
    font-weight: bold;
    margin: 0;
}
p {
    font-size: 14px;
```

```
    font-weight: 100;
    line-height: 20px;
    letter-spacing: 0.5px;
    margin: 20px 0 30px;
}
span {
    font-size: 12px;
}
a {
    color: #333;
    font-size: 14px;
    text-decoration: none;
    margin: 15px 0;
}
button {
    border-radius: 20px;
    border: 1px solid #ffc107;
    background-color: #ffc107;
    color: #FFFFFF;
    font-size: 12px;
    font-weight: bold;
    padding: 12px 45px;
    letter-spacing: 1px;
    text-transform: uppercase;
    transition: transform 80ms ease-in;
}
form {
    background-color: #FFFFFF;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0 50px;
```

```
    height: 100%;
    text-align: center;
}
input {
    background-color: #eee;
    border: none;
    padding: 12px 15px;
    margin: 8px 0;
    width: 100%;
}
.container {
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 14px 28px rgba(0,0,0,0.25), 0 10px 10px rgba(0,0,0,0.22);
    position: relative;
    overflow: hidden;
    width: 768px;
    max-width: 100%;
    min-height: 480px;
}
.form-container {
    position: absolute;
    top: 0;
    height: 100%;
}
.log-in-container {
    left: 0;
    width: 50%;
    z-index: 2;
}
.overlay-container {
    position: absolute;
    top: 0;
```

```
    left: 50%;
    width: 50%;
    height: 100%;
}
.overlay {
    background: #81c408;
    background: -webkit-linear-gradient(to right, ##81c408, ##81c408);
    background: linear-gradient(to right, ##81c408, ##81c408);
    background-repeat: no-repeat;
    background-size: cover;
    background-position: 0 0;
    color: #FFFFFF;
    position: relative;
    left: -100%;
    height: 100%;
    width: 200%;
}
.overlay-panel {
    position: absolute;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0 40px;
    text-align: center;
    top: 0;
    height: 100%;
    width: 50%;
}
.overlay-right {
    right: 0;
}
.social-container {
```

```
    margin: 50px 0;
}
.social-container a {
    border: 1px solid #DDDDDD;
    border-radius: 50%;
    display: inline-flex;
    justify-content: center;
    align-items: center;
    margin: 0 5px;
    height: 40px;
    width: 40px;
}
    </style>
</head>
<body>
<?php
if(isset($_POST['register'])){
    $name = $_POST['name'];
    $contact = $_POST['phone'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $userrole = "Farmer";
    $status = 1;
    $date = date('Y-m-d H:i:s');
    $insertQuery = "INSERT INTO user (name, email, date, status, contact, userRole, password) VALUES ('$name', '$email', '$date', '$status', '$contact', '$userrole', '$password')";
    if (mysqli_query($conn, $insertQuery)) {
        echo "<script>alert('User added successfully');location.href='index.php'</script>";
    } else {
 echo "<script>alert('Unable to process your request!');location.href='index.php'</script>";
    }
}
?>
```

```
    <div class="container" id="container">
        <div class="form-container log-in-container">
            <form method="post">
                <h1>Register</h1>
              <input type="name" name="name" placeholder="Name" />
                <input type="phone" name="phone" placeholder="Phone Number" />
                <input type="email" name="email" placeholder="Email" />
                <input type="password" name="password" placeholder="Password" />
                <a href="index.php">Already a User?..Click here to login</a>
                <button type="submit" name="register">Register</button>
            </form>
        </div>
        <div class="overlay-container">
            <div class="overlay">
                <div class="overlay-panel overlay-right">
                    <h1>Agro Mart</h1>
                     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

Admin Module: article.php

```
<?php
session_start();
require_once 'include/config.php';
if (empty($_SESSION['isLogin'])) {
    echo "<script>alert('Kindly login to proceed');location.href='index.php'</script>";
}
require_once 'include/header.php';
?>
<?php
if (isset($_POST['add_submit'])) {
    $title = addslashes(trim($_POST['title']));
    $description = addslashes(trim($_POST['description']));
    $status = 1;
    $date = date('Y-m-d H:i:s');
 $imagePath = time() . "." . pathinfo($_FILES['image']['name'], PATHINFO_EXTENSION);
    if (move_uploaded_file($_FILES['image']['tmp_name'], "images/" . $imagePath)) {
            $insertQuery = "INSERT INTO article (title, description, date, image, status) VALUES ('$title', '$description', '$date', '$imagePath', '$status')";
    if (mysqli_query($conn, $insertQuery)) {
        echo "<script>alert('Article added successfully');location.href='article.php'</script>";
    } else {
   echo "<script>alert('Unable to process your request!');location.href='article.php'</script>";
    }
        } else {
        echo "<script>alert('Unable to upload image on server.');</script>";
    }
}
if (isset($_POST['delete_submit'])) {
    $date = date('Y-m-d H:i:s');
 if (mysqli_query($conn, "UPDATE article SET status = 0 WHERE id = '$_POST[delete_id]'")) {
        echo "<script>alert('Article deleted successfully');location.href='article.php'</script>";
    } else {
```

```
echo "<script>alert('Unable to process your request!');location.href='article.php'</script>";
    }
}
?>
<!-- Single Page Header start -->
<div class="container-fluid page-header pymm-5">
    <h1 class="text-center text-white display-6">Article</h1>
    <ol class="breadcrumb justify-content-center mb-0">
        <li class="breadcrumb-item"><a href="#">Home</a></li>
        <li class="breadcrumb-item"><a href="#">Pages</a></li>
        <li class="breadcrumb-item active text-white">Article</li>
    </ol>
</div>
<!-- Single Page Header End -->
<!-- Article Start-->
<div class="container-fluid fruite py-5">
    <div class="container py-5">
        <h1 class="mb-4">Add Articles</h1>
        <div class="row g-4">
            <div class="col-lg-12">
                <div class="row g-4">
                    <div class="col-9"></div>
                    <div class="col-xl-3">
                        <div class="input-group w-100 mx-auto d-flex">
                            <a class="btn border border-secondary rounded-pill px-3 text-primary" data-bs-toggle='modal'
data-bs-target='#add'><i class="bi bi-plus-square-fill me-2 text-primary"></i> Add Article</a>
                        </div>
                    </div>
                </div>
                <div class="row g-4">
                    <table class="table table-striped">
                        <thead>
                            <tr>
```

```
                    <th scope="col">Id</th>
                    <th scope="col">Title</th>
                    <th scope="col">Description</th>
                    <th scope="col">Date</th>
                    <th scope="col">status</th>
                    <th scope="col">Action</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    $sql = "SELECT * FROM article where status = 1";
                    $res = mysqli_query($conn,$sql);
                    if(mysqli_num_rows($res) > 0) {
                        $i = 1;
                        while($row = mysqli_fetch_array($res)) {
                            echo "<tr>";
                            echo "<td>$i</td>";
                            echo "<td>$row[title]</td>";
                            echo "<td>$row[description]</td>";
                            echo "<td>".date_format(date_create($row['date']), 'Y-m-d')."</td>";
                            echo "<td>$row[status]</td>";
                            echo "<td>";
                            echo "<form method='post'>";
          echo "<input autocomplete='off'  type='hidden' name='delete_id' value='$row[id]'/>
          <button type='submit' name='delete_submit' onClick='return confirm(" . "'Are you sure you want to delete?'" . ")' class='btn btn-danger shadow btn-xs sharp'><i class='fa fa-trash'></i></button>";
          echo "</form>";
          echo "</td>";
          echo "</tr>";
           $i++;
      }
 }
 ?>
```

```
        </tbody>
        </table>
        </div>
          <div class="modal fade" id="add" tabindex="-1" aria-labelledby="addLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-center modal-xl">
  <form method="POST" enctype="multipart/form-data">
  <div class="modal-content" style="width: 500px;">
<div class="modal-header">
 <h1 class="modal-title fs-5" id="addLabel">Add Article</h1>
 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"> </button>
  </div>
  <div class="modal-body">
  <div class="row">
  <div class="col-xl-12 mb-3">
  <label class="form-label">Title<span class="text-danger">*</span></label>
 <input autocomplete='off' type="text" class="form-control" maxlength="100" required name="title">
  </div>
  <div class="col-xl-12 mb-3">
 <label class="form-label">Description<span class="text-danger">*</span></label>
  <textarea class="form-control" rows="4" required name="description"></textarea>
  </div>
  <div class="col-xl-12 mb-3">
  <label class="form-label">Date<span class="text-danger">*</span></label>
  <input autocomplete='off' type="date" class="form-control" required name="date">
    </div>
  <div class="col-xl-12 mb-3">
 <label class="form-label">Upload Image<span class="text-danger">*</span></label>
  <input type="file" class="form-control" name="image" accept="image/*" required>
  </div>
  </div>
  </div>
<div class="modal-footer">
<button type="button" class="btn btn-danger light" data-bs-dismiss="modal">Close </button>
```

```
 <button type="submit" name="add_submit" class="btn btn-primary">Submit</button>
 </div>
</div>
 </form>
 </div>
 </div>
</div>
 </div>
 </div>
</div>
<?php
require_once 'include/footer.php';
?>
```

User Module: purchase-view-user.php

```
<?php
session_start();
require_once 'include/config.php';
if (empty($_SESSION['isLogin'])) {
    echo "<script>alert('Kindly login to proceed');location.href='index.php'</script>";
}
require_once 'include/header.php';
?>
<?php
$email = $_SESSION['email'];
$sql = "SELECT * FROM user WHERE email = '$email'";
$res = mysqli_query($conn, $sql);
if ($res) {
    if (mysqli_num_rows($res) > 0) {
        $row = mysqli_fetch_assoc($res);
        $userid = $row['id'];
    }
}
?>
<div class="container-fluid page-header py-5">
    <h1 class="text-center text-white display-6">Purchases</h1>
    <ol class="breadcrumb justify-content-center mb-0">
        <li class="breadcrumb-item"><a href="#">Home</a></li>
        <li class="breadcrumb-item"><a href="#">Pages</a></li>
        <li class="breadcrumb-item active text-white">Purchases</li>
    </ol>
</div>
<!-- Single Page Header End -->
        <!-- Fruits Shop Start-->
        <div class="container-fluid fruite py-5">
            <div class="container py-5">
                <div class="tab-class text-center">
                    <div class="row g-4">
```

```
                    <div class="col-lg-4 text-start">
                        <h1>Our Organic Products</h1>
                    </div>
                </div>
                <div class="tab-content">
                    <div id="tab-1" class="tab-pane fade show p-0 active">
                        <div class="row g-4">
                            <div class="col-lg-12">
                                <div class="row g-4">
 <?php
 $sql = "SELECT p.pname, p.image, py.price, py.payment_number FROM payment py, product p WHERE p.id
= py.productid AND py.status = 1 AND py.userid = '$userid'";
 $res = mysqli_query($conn, $sql);
if (mysqli_num_rows($res) > 0) {
   while ($row = mysqli_fetch_array($res)) {
?>
  <div class="col-md-6 col-lg-4 col-xl-3">
  <div class="rounded position-relative fruite-item">
 <div class="fruite-img">
 <img src="images/<?php echo $row['image']; ?>" class="img-fluid w-100 rounded-top" alt="">
   </div>
 <div class="p-4 border border-secondary border-top-0 rounded-bottom">
<h4><?php echo $row['pname']; ?></h4>
 <p>Payment No: <?php echo $row['payment_number']; ?></p>
   <div class="">
 <p class="text-dark fs-5 fw-bold mb-0">Price: $<?php echo $row['price']; ?></p>
  </div>
 </div>
 </div>
 </div>
 <?php
  }
 }
?>
```

```
 </div>
 </div>
</div>
 </div>
 </div>
</div>
</div>
</div>
<!-- Fruits Shop End-->
<?php
require_once 'include/footer.php';
?>
```

Farmer Module: subsidy.php

```
<?php
session_start();
require_once 'include/config.php';
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;

if (empty($_SESSION['isLogin'])) {
    echo "<script>alert('Kindly login to proceed');location.href='index.php'</script>";
}
require_once 'include/header.php';
$user_id = $_SESSION['user_id'];
?>
<?php
if (isset($_POST['verify'])) {
    $name = addslashes(trim($_POST['name']));
    $email = addslashes(trim($_POST['email']));
    $phone = addslashes(trim($_POST['phone']));
    $subject = addslashes(trim($_POST['subject']));
    $amount = addslashes(trim($_POST['amount']));
    $adharno = addslashes(trim($_POST['adharno']));
    $description = addslashes(trim($_POST['description']));
    $subsidy_no = "SUB" . mt_rand(10000, 99999);
    $otpstatus = "Pending";
    $otp = (string) mt_rand(100000, 999999);
    $status = "Processing";
    $date = date('Y-m-d H:i:s');
      $insertQuery = "INSERT INTO subsidy (subsidy_no, name, price, subject, adhar_no, phone, email, description, status, date, otp_status, otp, user_id) VALUES ('$subsidy_no', '$name', '$amount', '$subject', '$adharno',  '$phone', '$email', '$description', '$status', '$date', '$otpstatus', '$otp', '$user_id')";
    if (mysqli_query($conn, $insertQuery)) {
        require 'vendor-email/autoload.php';
        $title = "Agromart";
        $mail = new PHPMailer();
```

```
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
        $mail->Port = 465;
        $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
        $mail->SMTPAuth = true;
        $mail->IsHTML(true);
        $mail->Username = 'haitulunadu@gmail.com';
        $mail->Password = 'oynqcivlwgswtyrn';
        $mail->setFrom('haitulunadu@gmail.com', $title);
        $mail->addReplyTo('haitulunadu@gmail.com', $title);
        $mail->addAddress($email, "User");
        $mail->Subject = 'OTP verification - ' . $title;
       $mail->Body = 'Dear User, <br> Please verify your mail with OTP<br> Your OTP is <strong>' . $otp . '</strong><br><br> Thank you<br>Team ' . $title;
        if ($mail->send()) {
            echo "<script>alert('Please verify from OTP sent to your mail $email');location.href='verifyotp.php?subno=$subsidy_no'</script>";
        } else {
            // Handle email sending failure
 echo "<script>alert('Unable to send OTP email!');location.href='subsidy.php'</script>";
        }
    } else {
        // Handle SQL query execution failure
 echo "<script>alert('Unable to process your request!');location.href='subsidy.php'</script>";
    }
}
?>
<style>
   .modal-body {
        display: flex;
        justify-content: center;
        align-items: center;
    }
    .otp-input-container {
```

```
        margin: 0 5px;
    }
    .otp-input {
        width: 40px;
        height: 40px;
        font-size: 20px;
        text-align: center;
    }
    .otp-submit-button,
    .resend-otp-button {
      margin-top: 10px;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }
    .otp-submit-button {
        background-color: #007bff;
        color: #fff;
    }
    .resend-otp-button {
        background-color: #ccc;
        color: #000;
    }
</style>
<!-- Single Page Header start -->
<div class="container-fluid page-header py-5">
    <h1 class="text-center text-white display-6">Subsidy</h1>
    <ol class="breadcrumb justify-content-center mb-0">
        <li class="breadcrumb-item"><a href="#">Home</a></li>
        <li class="breadcrumb-item"><a href="#">Pages</a></li>
        <li class="breadcrumb-item active text-white">Subsidy</li>
    </ol>
</div>
```

```
<!-- Single Page Header End -->
<!-- Article Start-->
<div class="container-fluid fruite py-5">
    <div class="container py-5">
        <h3 class="mb-4">Apply Subsidy</h3>
        <form method="POST" class="row">
            <div class="col-xl-4 mb-3">
                <label class="form-label">Name<span class="text-danger">*</span></label>
  <input autocomplete='off' type="text" class="form-control" maxlength="100" required name="name">
            </div>
            <div class="col-xl-4 mb-3">
                <label class="form-label">Phone no<span class="text-danger">*</span></label>
      <input autocomplete='off' type="text" class="form-control" maxlength="100" required name="phone">
            </div>
            <div class="col-xl-4 mb-3">
        <label class="form-label">Email<span class="text-danger">*</span></label>
    <input autocomplete='off' type="email" class="form-control" maxlength="100" required name="email">
            </div>
            <div class="col-xl-4 mb-3">
       <label class="form-label">Adhar no<span class="text-danger">*</span></label>
  <input autocomplete='off' type="text" class="form-control" maxlength="100" required name="adharno">
            </div>
            <div class="col-xl-4 mb-3">
     <label class="form-label">Subject<span class="text-danger">*</span></label>
  <input autocomplete='off' type="text" class="form-control" maxlength="100" required name="subject">
            </div>
            <div class="col-xl-4 mb-3">
      <label class="form-label">Amount<span class="text-danger">*</span></label>
   <input autocomplete='off' type="text" class="form-control" maxlength="100" required name="amount">
            </div>
            <div class="col-xl-8 mb-3">
 <label class="form-label">Description<span class="text-danger">*</span></label>
      <textarea class="form-control" required name="description"></textarea>
            </div>
```

```
            <div>
        <button type="submit" class="btn border border-secondary rounded-pill px-3 text-primary" name="verify"
><i class="bi bi-plus-square-fill me-2 text-primary"></i> Verify</button>
            </div>
        </form>
        <h3 class="mb-2 mt-5">Applied subsidy</h3>
        <div class="row g-4">
            <div class="col-lg-12">
                <div class="row g-4">
                    <div class="col-9"></div>
                </div>
                <div class="row g-4">
                    <table class="table table-striped">
                        <thead>
                            <tr>
                            <th scope="col">Id</th>
                            <th scope="col">Subsidy no</th>
                            <th scope="col">Subject</th>
                            <th scope="col">Description</th>
                            <th scope="col">Amount</th>
                            <th scope="col">Date</th>
                            <th scope="col">status</th>
                            </tr>
                        </thead>
                        <tbody>
        <?php
     $sql = "SELECT * FROM subsidy where otp_status = 'Verified' and user_id = '$user_id'";
    $res = mysqli_query($conn,$sql);
if(mysqli_num_rows($res) > 0) {
                            $i = 1;
                            while($row = mysqli_fetch_array($res)) {
                                echo "<tr>";
                                echo "<td>$i</td>";
                                echo "<td>$row[subsidy_no]</td>";
```

```
                        echo "<td>$row[subject]</td>";
                        echo "<td width=400px;>$row[description]</td>";
                        echo "<td>$row[price]</td>";
                        echo "<td>".date_format(date_create($row['date']), 'Y-m-d')."</td>";
                        echo "<td>$row[status]</td>";
                        echo "</tr>";
                        $i++;
                    }
                }
                ?>
            </tbody>
        </table>
    </div>
</div>
</div>
</div>
<?php
require_once 'include/footer.php';
?>
```

NGO Module: view-subsidy.php

```
<?php
session_start();
require_once 'include/config.php';
if (empty($_SESSION['isLogin'])) {
    echo "<script>alert('Kindly login to proceed');location.href='index.php'</script>";
}
require_once 'include/header.php';
?>
<?php
if (isset($_POST['update_status'])) {
    $status = addslashes(trim($_POST['status']));
    $subsidy_id = addslashes(trim($_POST['subsidy_id']));
    $update = "UPDATE subsidy SET status = '$status' where subsidy_id = '$subsidy_id'";
        if (mysqli_query($conn, $update)) {
  echo "<script>alert('Status updated successfully');location.href='view-subsidy.php'</script>";
        } else {
 echo "<script>alert('Unable to process your request!');location.href='view-subsidy.php' </script>";
        }
}
?>

<!-- Single Page Header start -->
<div class="container-fluid page-header py-5">
    <h1 class="text-center text-white display-6">Subsidy</h1>
    <ol class="breadcrumb justify-content-center mb-0">
        <li class="breadcrumb-item"><a href="#">Home</a></li>
        <li class="breadcrumb-item"><a href="#">Pages</a></li>
        <li class="breadcrumb-item active text-white">Subsidy</li>
    </ol>
</div>
<!-- Single Page Header End -->
<!-- Article Start-->
<div class="container-fluid fruite py-5">
```

```
<div class="container py-5">
    <h1 class="mb-4">Verify subsidy</h1>
    <div class="row g-4">
        <div class="col-lg-12">
            <div class="row g-4">
                <div class="col-9"></div>
            </div>
            <div class="row g-4">
                <table class="table table-striped">
                    <thead>
                        <tr>
                        <th scope="col">Id</th>
                        <th scope="col">Name</th>
                        <th scope="col">Phone</th>
                        <th scope="col">Subject</th>
                        <th scope="col">Description</th>
                        <th scope="col">amount</th>
                        <th scope="col">Date</th>
                        <th scope="col">status</th>
                        <th scope="col">Action</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php
                        $sql = "SELECT * FROM subsidy where otp_status = 'Verified'";
                        $res = mysqli_query($conn,$sql);
                        if(mysqli_num_rows($res) > 0) {
                            $i = 1;
                            while($row = mysqli_fetch_array($res)) {
                                echo "<tr>";
                                echo "<td>$i</td>";
                                echo "<td>$row[name]</td>";
                                echo "<td>$row[phone]</td>";
                                echo "<td>$row[subject]</td>";
```

```
                    echo "<td>$row[description]</td>";
                    echo "<td>$row[price]</td>";
                    echo "<td>".date_format(date_create($row['date']), 'Y-m-d')."</td>";
                    echo "<td>$row[status]</td>";
                    echo "<td>
            <button type='button' class='btn btn-primary' data-bs-toggle='modal' data-bs-target='#update$row[subsidy_id]'>Update </button>
                    </td>";
                    echo "</tr>";
                    $i++;
                    ?>
          <div class="modal fade" id="update<?php echo $row['subsidy_id']; ?>" tabindex="-1" aria-labelledby="addLabel" aria-hidden="true">
  <div class="modal-dialog">
    <form method="POST" enctype="multipart/form-data">
 <div class="modal-content">
  <div class="modal-header">
  <h1 class="modal-title fs-5" id="addLabel">Update</h1>
 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"> </button>
    </div>
  <div class="modal-body">
  <div class="row">
    <div class="col-xl-12 mb-3">
   <label class="form-label">Status<span class="text-danger">*</span></label>
    <select class="form-select" name="status">
<option value="Processing" <?php if ($row['status'] == 'Processing') echo 'selected'; ?>>Processing</option>
 <option value="Approved" <?php if ($row['status'] == 'Approved') echo 'selected'; ?>>Approved</option>
 <option value="Rejected" <?php if ($row['status'] == 'Rejected') echo 'selected'; ?>>Rejected</option>
 </select>
  </div>
</div>
<input type="hidden" name="subsidy_id" value="<?php echo $row['subsidy_id']; ?>">
 </div>
 <div class="modal-footer">
```

```
 <button type="button" class="btn btn-danger light" data-bs-dismiss="modal">Close</button>
  <button type="submit" name="update_status" class="btn btn-primary">Submit</button>
   </div>
   </div>
   </form>
 </div>
</div>
 <?php
  }
  }
   ?>
 </tbody>
      </table>
 </div>
</div>
       </div>
    </div>
</div>
<?php
require_once 'include/footer.php';
?>
```

Machine Learning Code:

```
import os
import matplotlib.pyplot as plt
from PIL import Image
import math
from flask import Flask, render_template, request,redirect,url_for
app = Flask(__name__)
dir_example = "/content/drive/MyDrive/plant"
classes = os.listdir(dir_example)
print(classes)
train = "/content/drive/MyDrive/plant"
# from tensorflow.keras.models import h5py
from keras.models import load_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D, Dropout
from tensorflow.keras.losses import sparse_categorical_crossentropy
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf
# Create a generator
train_datagen = ImageDataGenerator(
  rescale=1./255
)
train_datagen = train_datagen.flow_from_directory(
        train,
        batch_size=32,
        target_size=(300, 300),
        class_mode='sparse')
#Printing the training set
labels = (train_datagen.class_indices)
print(labels,'\n')
for image_batch, label_batch in train_datagen:
     break
```

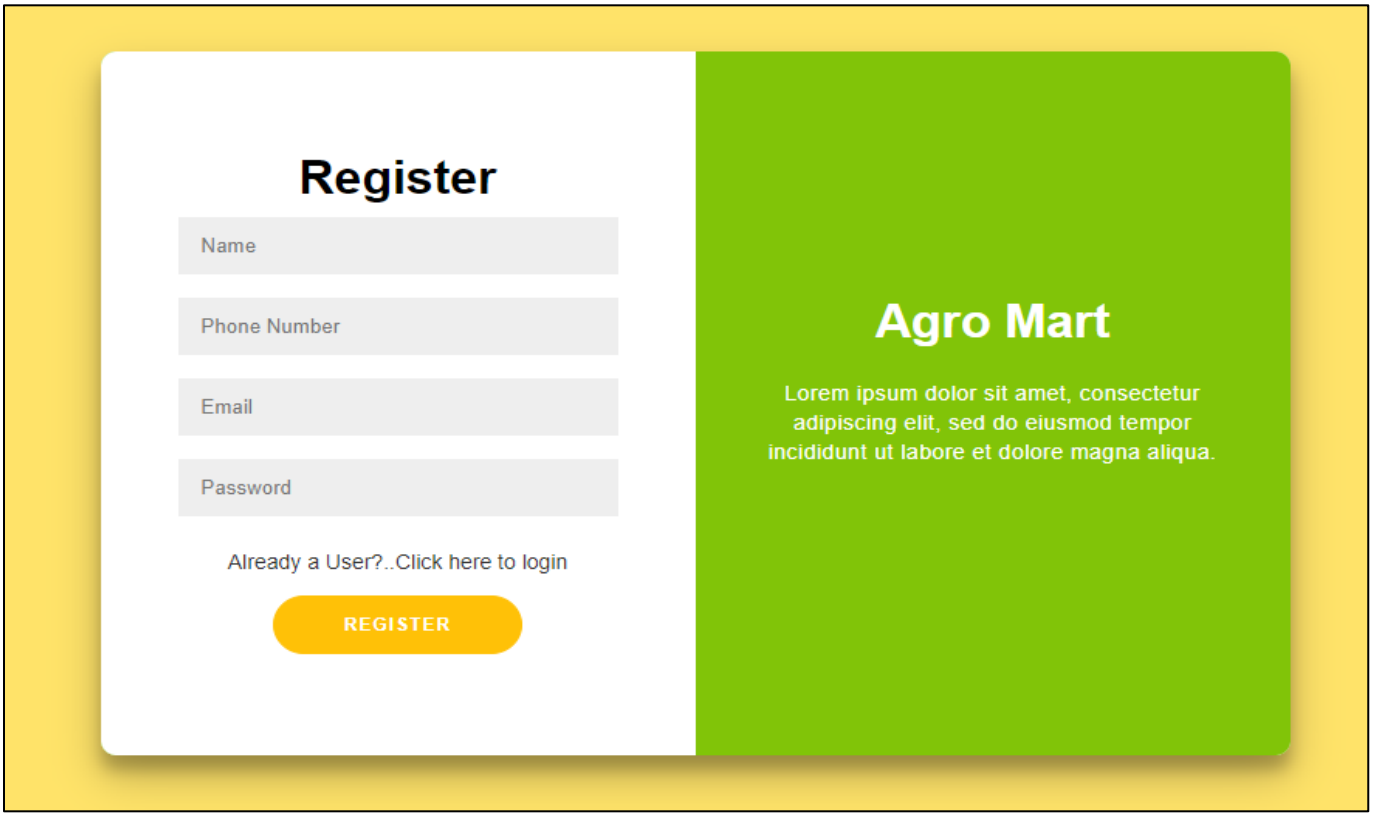
```
image_batch.shape, label_batch.shape
# print(label_batch)
model=Sequential()
#Convolution blocks
model.add(Conv2D(32, kernel_size = (3,3), padding='same',input_shape=(300,300,3),activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Conv2D(64, kernel_size = (3,3), padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Conv2D(32, kernel_size = (3,3), padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
#Classification layers
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(32,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(10,activation='softmax'))
model.summary()
model.compile(optimizer = 'Adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
model.fit(train_datagen,
        epochs=25,
        steps_per_epoch=382//32)
model.save("platt25.h5")
model = load_model('platt25.h5')
print(train_datagen.class_indices)
Labels = '\n'.join(sorted(train_datagen.class_indices.keys()))
with open('Labels.txt', 'w') as file:
  file.write(Labels)
class_names = list(labels.keys())
from tensorflow.keras.preprocessing import image
import numpy as np
test_img = '/content/drive/MyDrive/plant/Apple/AppleHealthy(104).JPG'
img = image.load_img(test_img, target_size = (300,300))
```

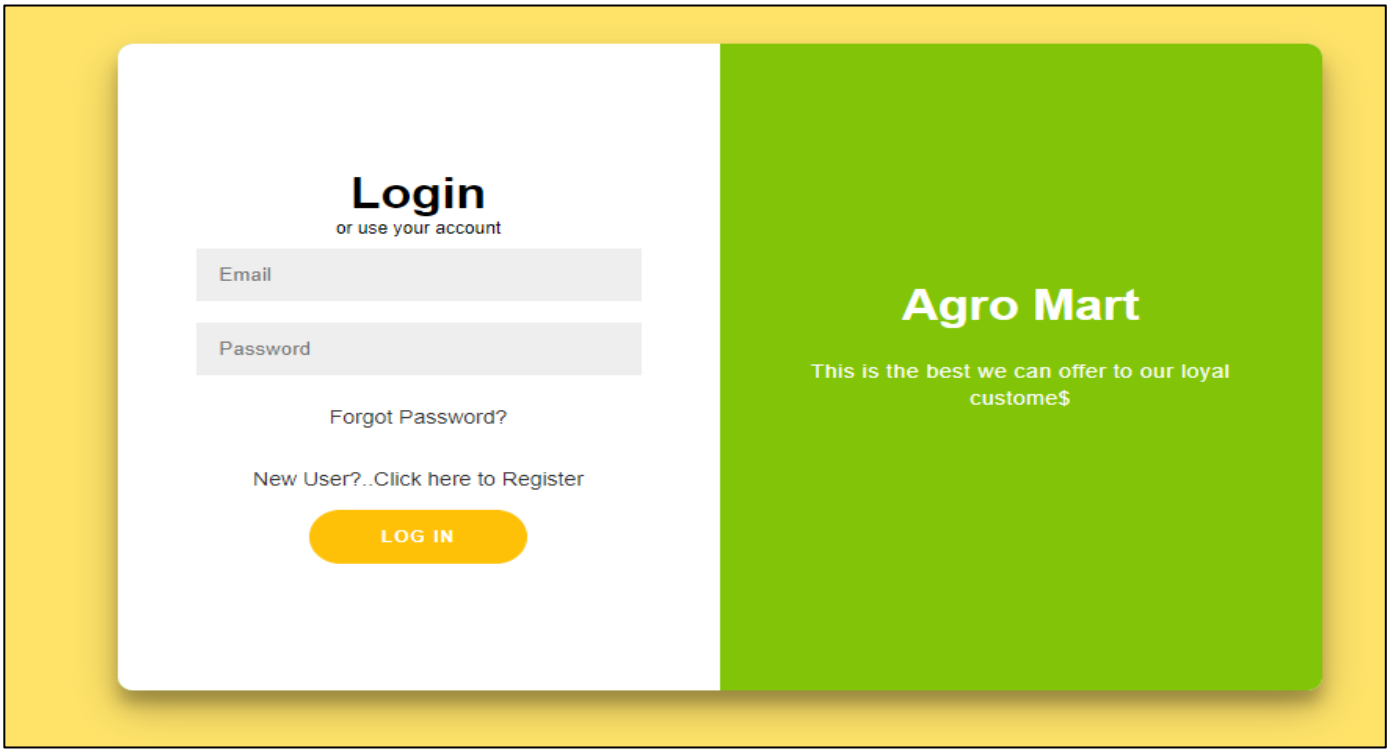
```
img = image.img_to_array(img, dtype=np.uint8)
img = np.array(img)/255.0
prediction = model.predict(img[np.newaxis, ...])
print("Probability: ",np.max(prediction[0], axis=-1))
predicted_class = class_names[np.argmax(prediction[0], axis=-1)]
print("Classified: ",predicted_class,'\n')
plt.axis('off')
plt.imshow(img.squeeze())
plt.title("Loaded Image")
```

USER INTERFACE

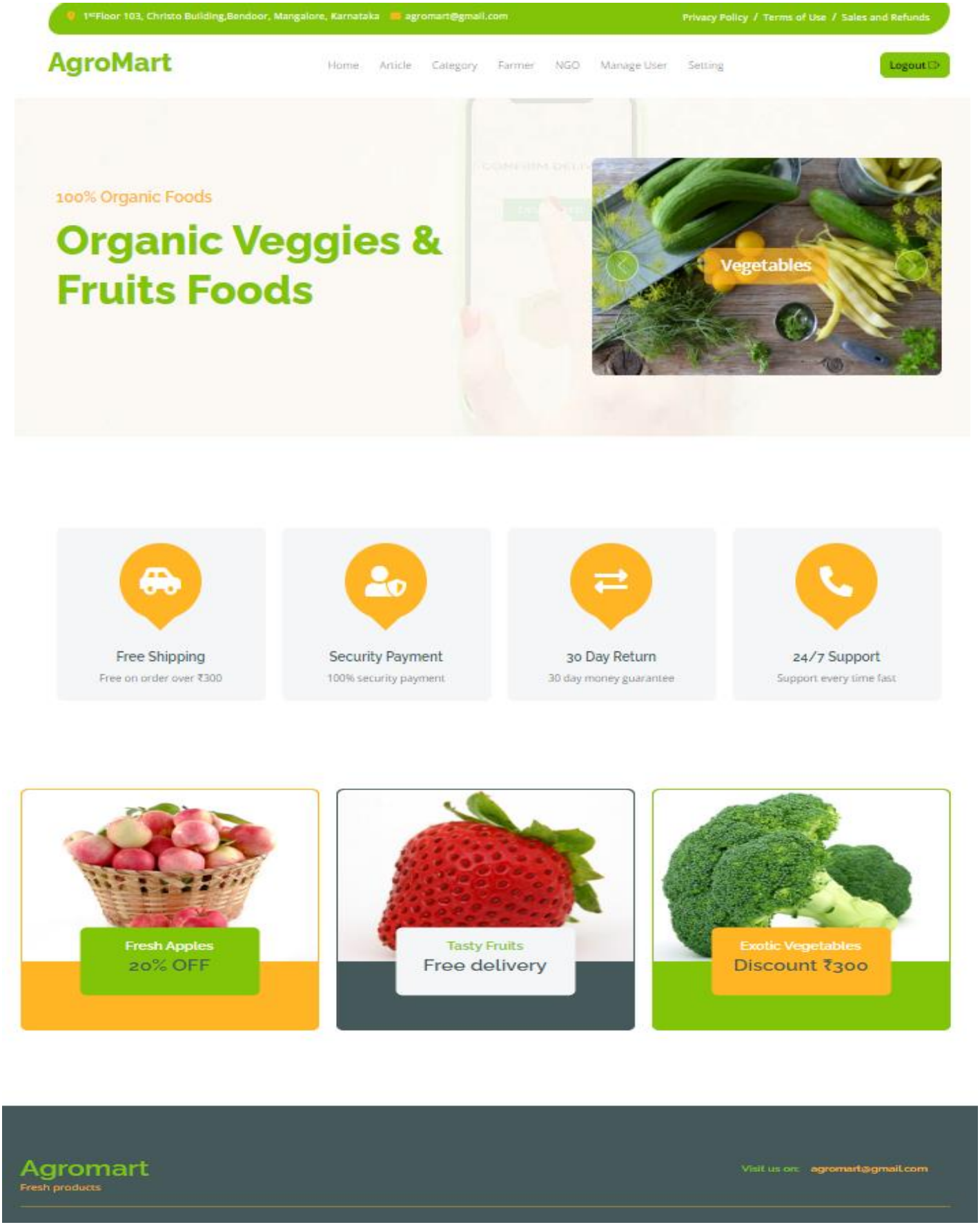
USER INTERFACE

Registration/Login Page:

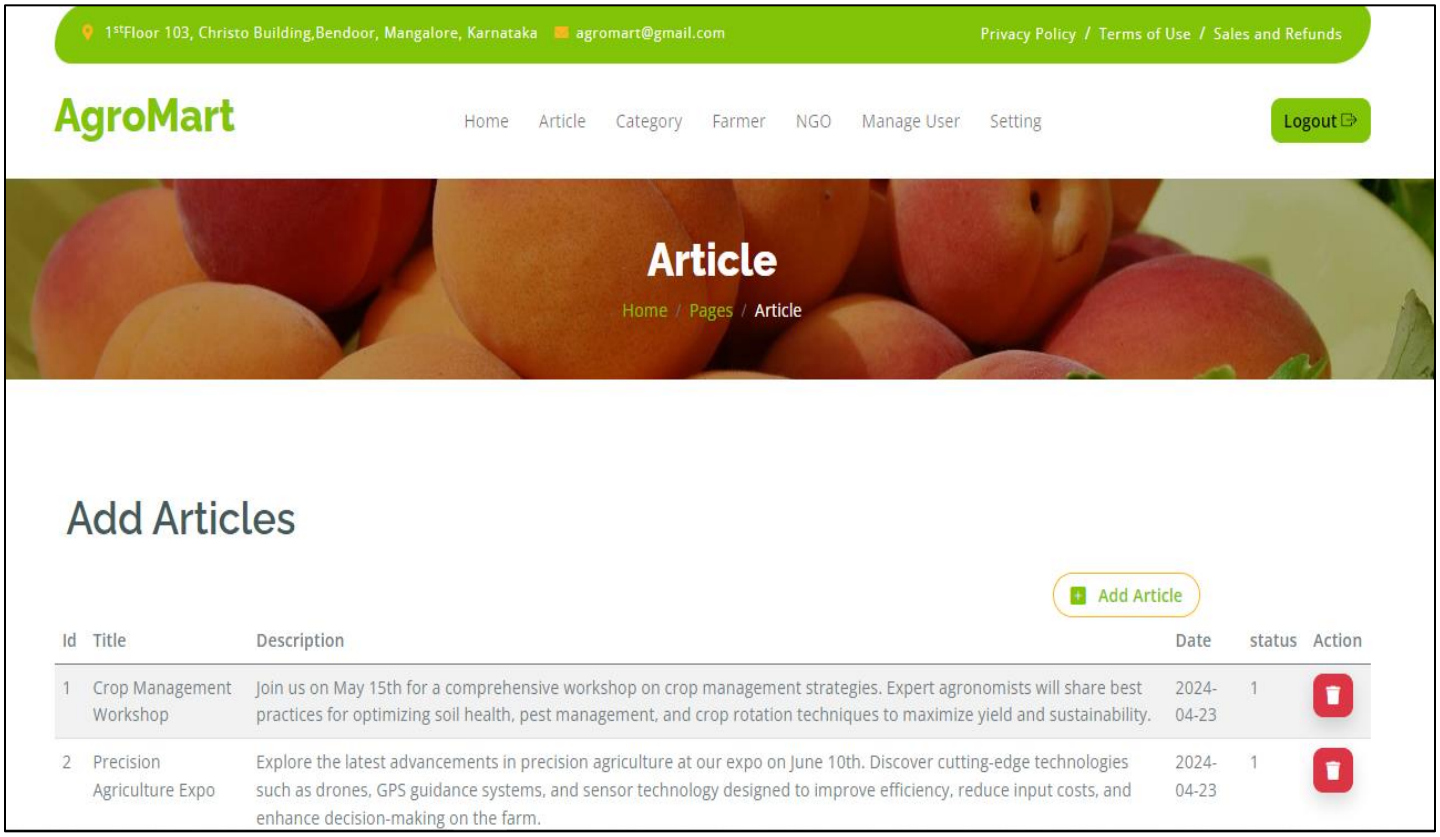




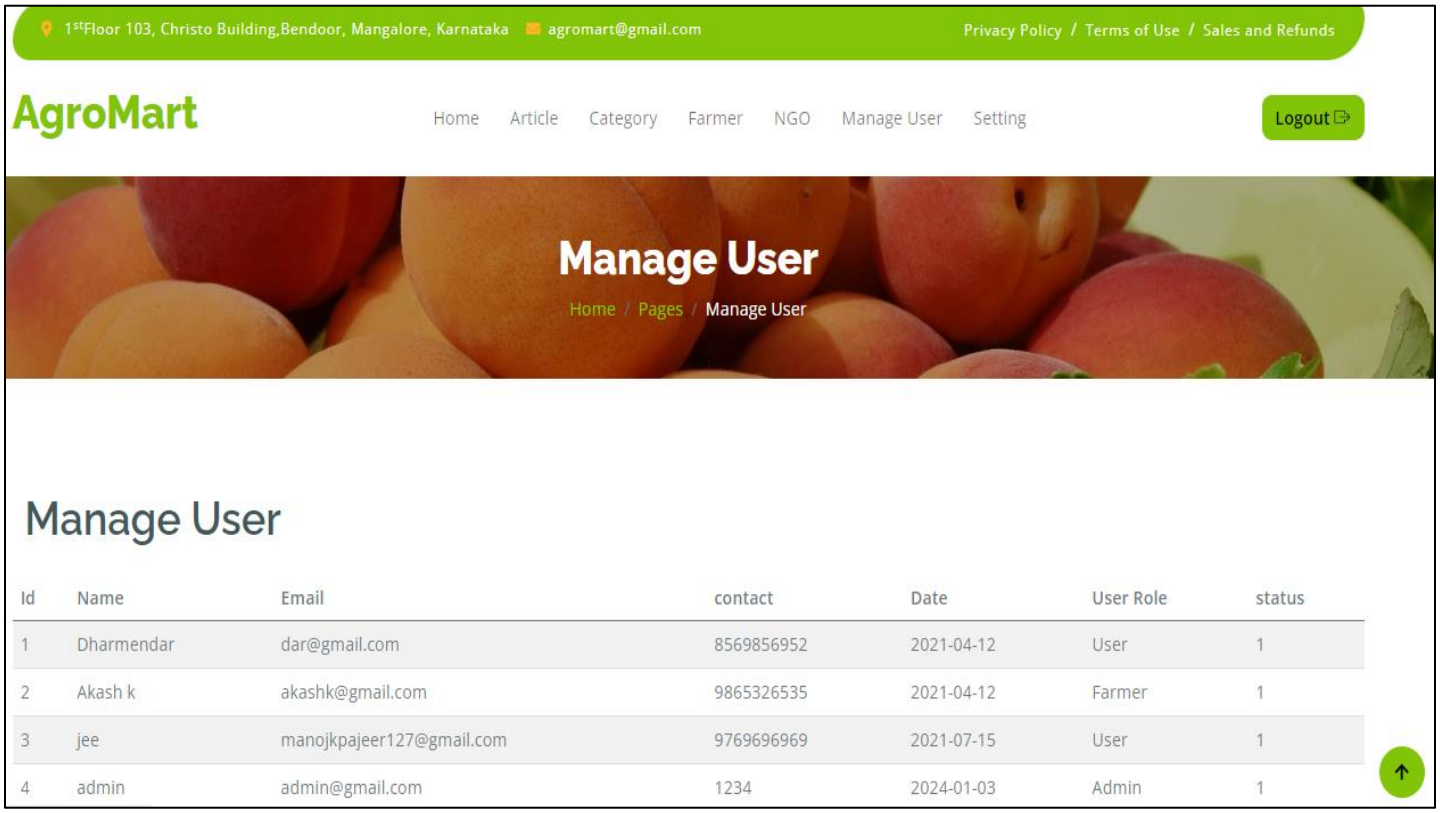
Admin: Home Page



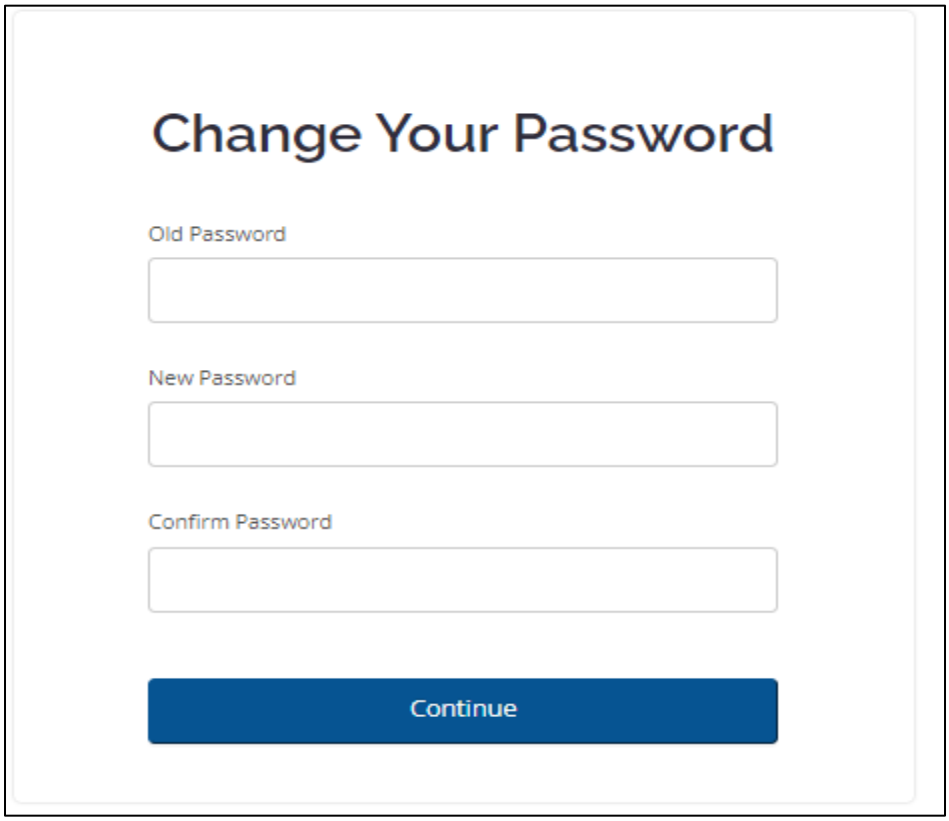
Add Article Page:



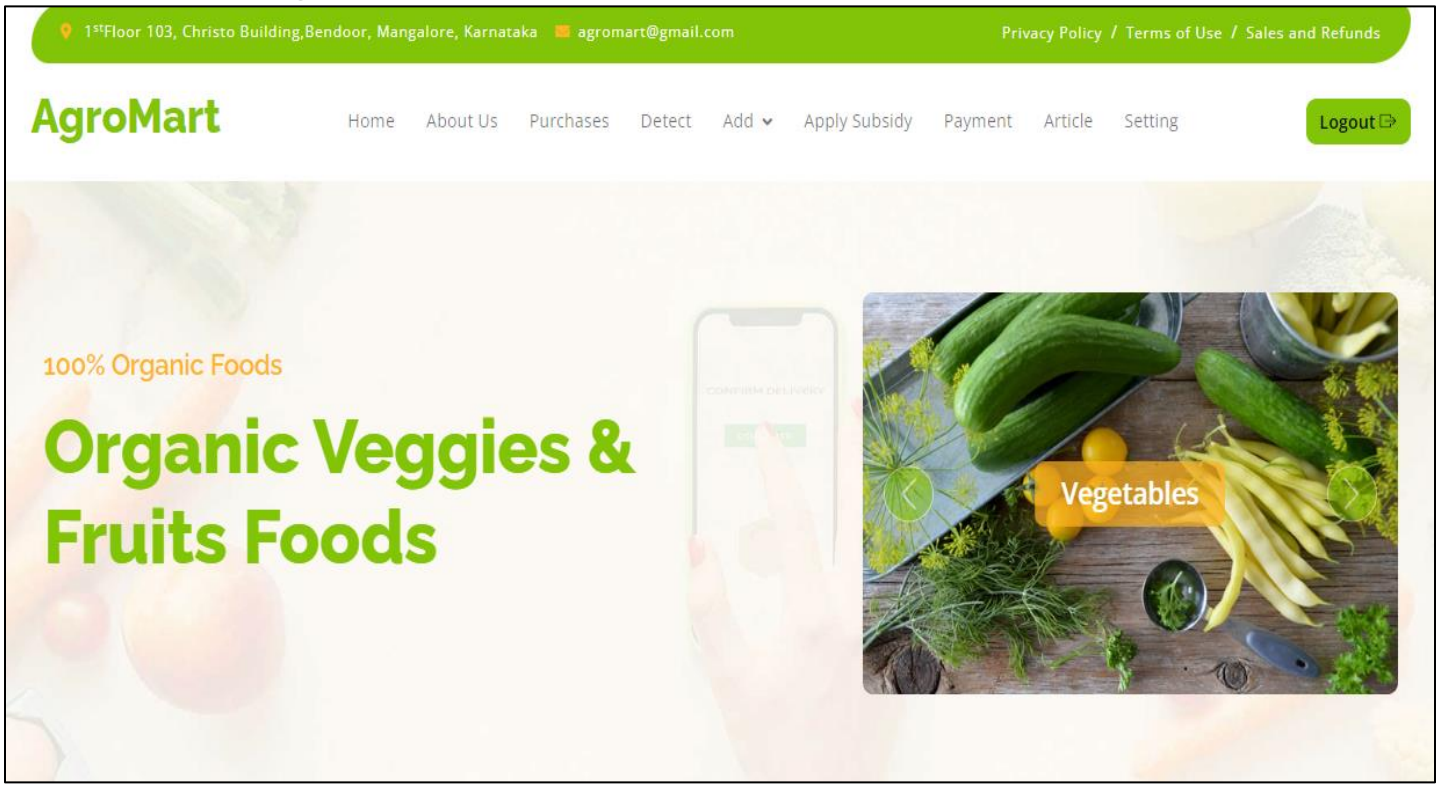
Manage User Page:



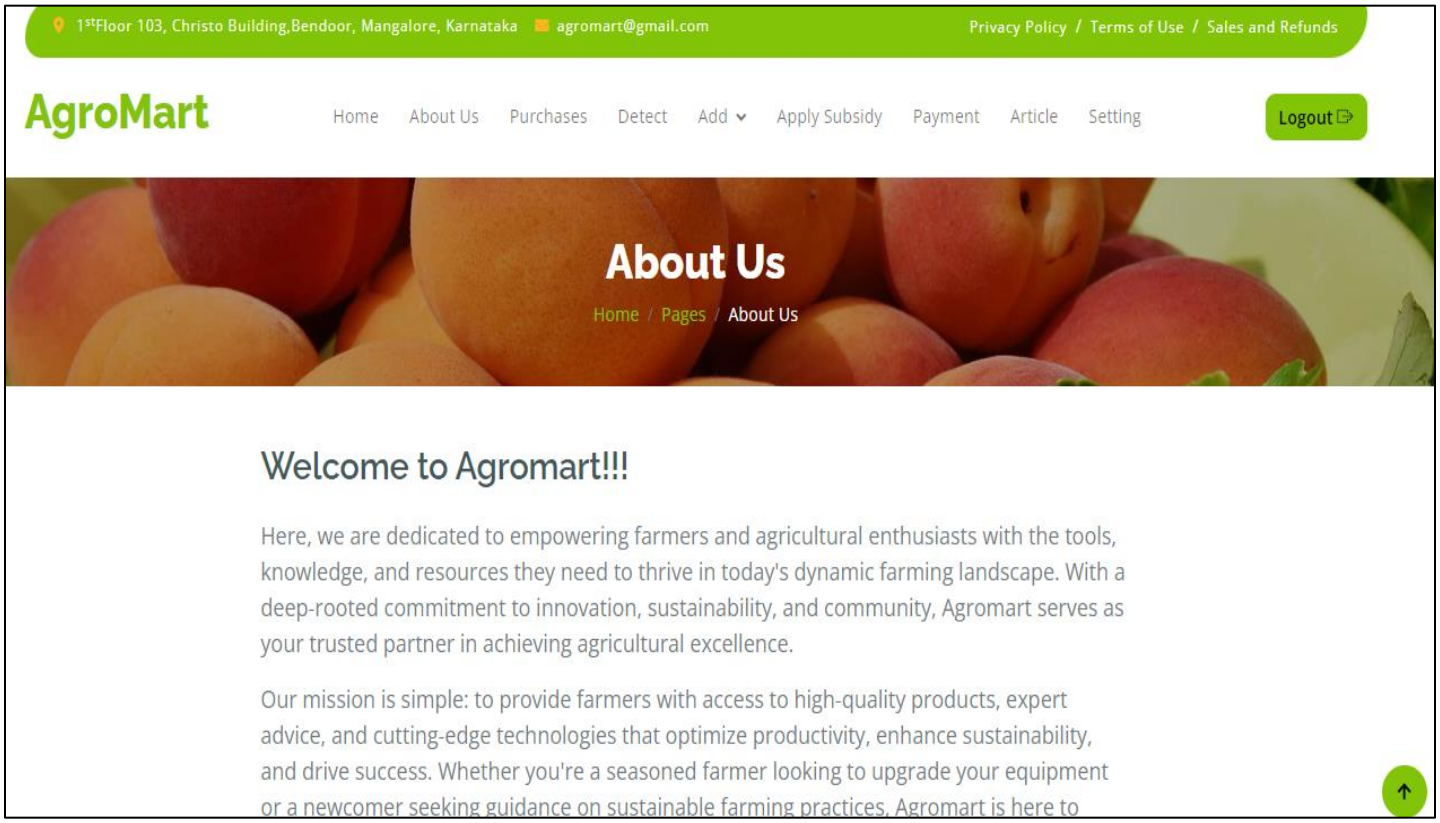
Setting:



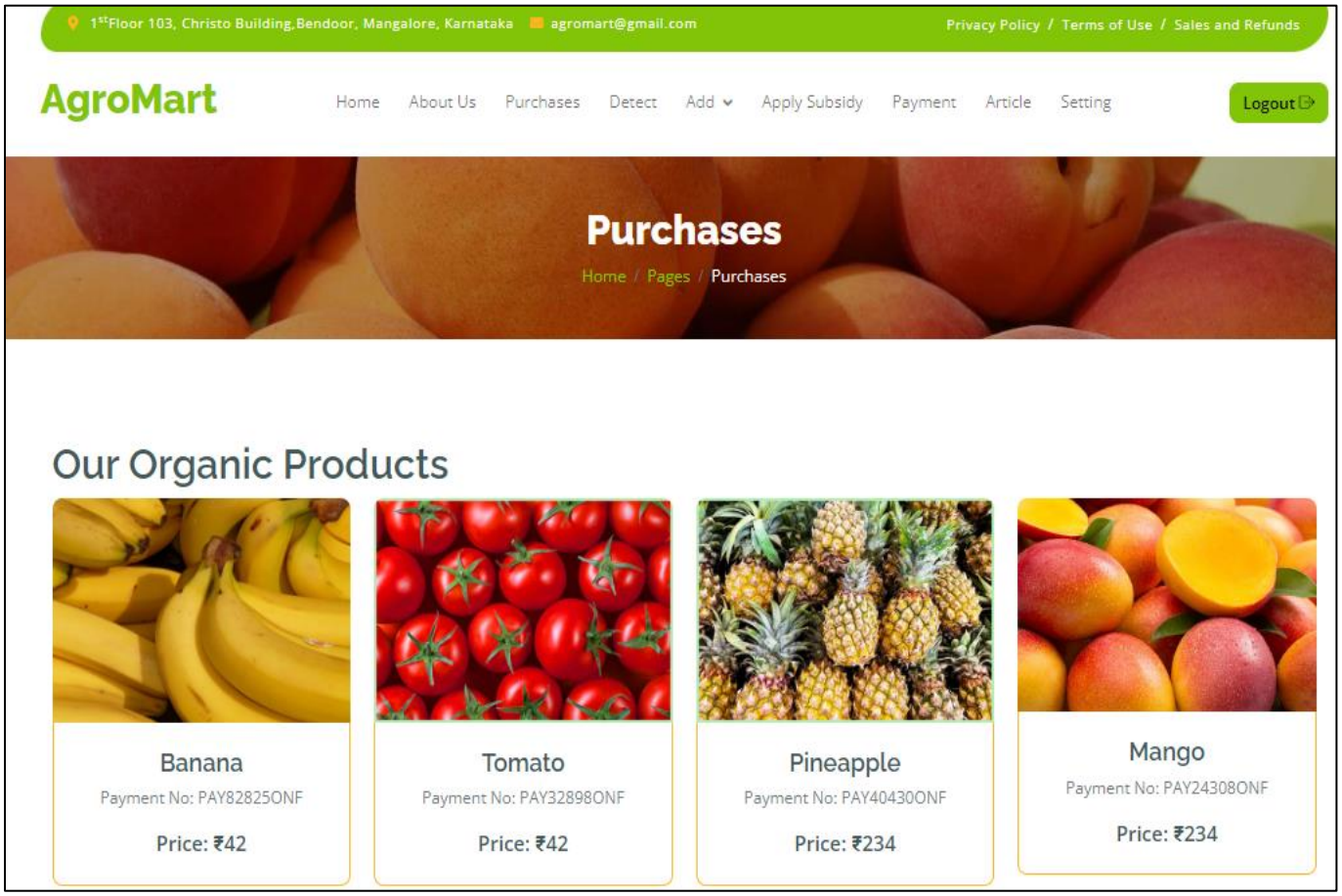
Farmer: Home Page



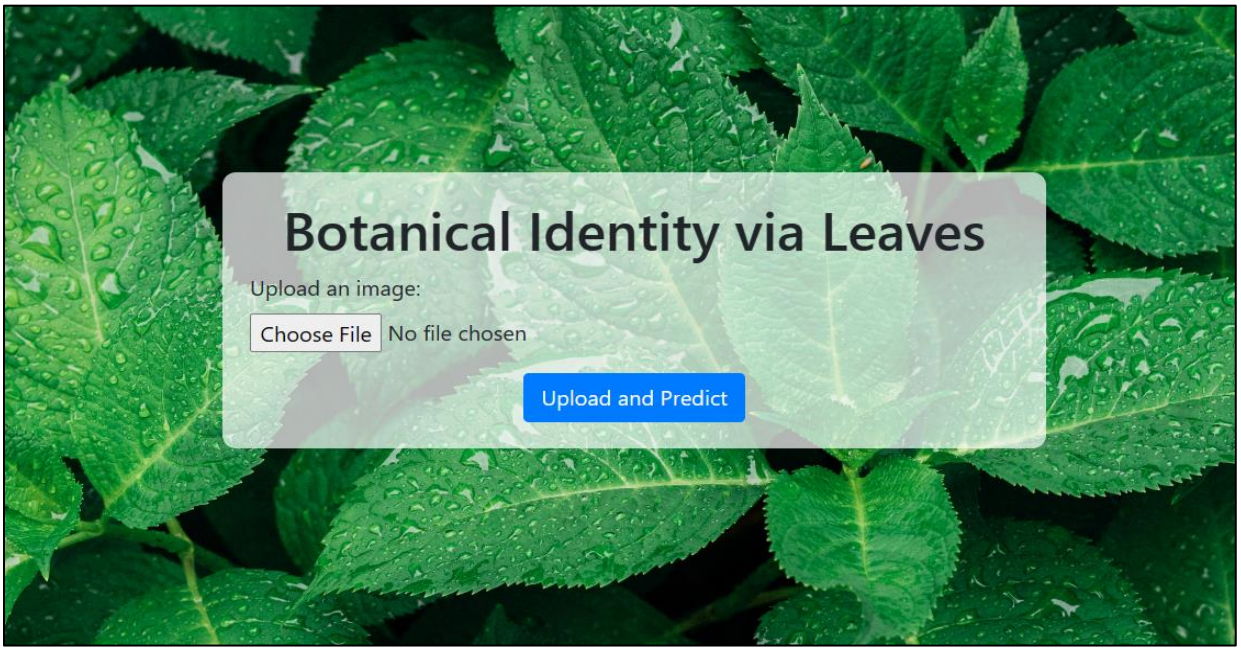
About Us Page:

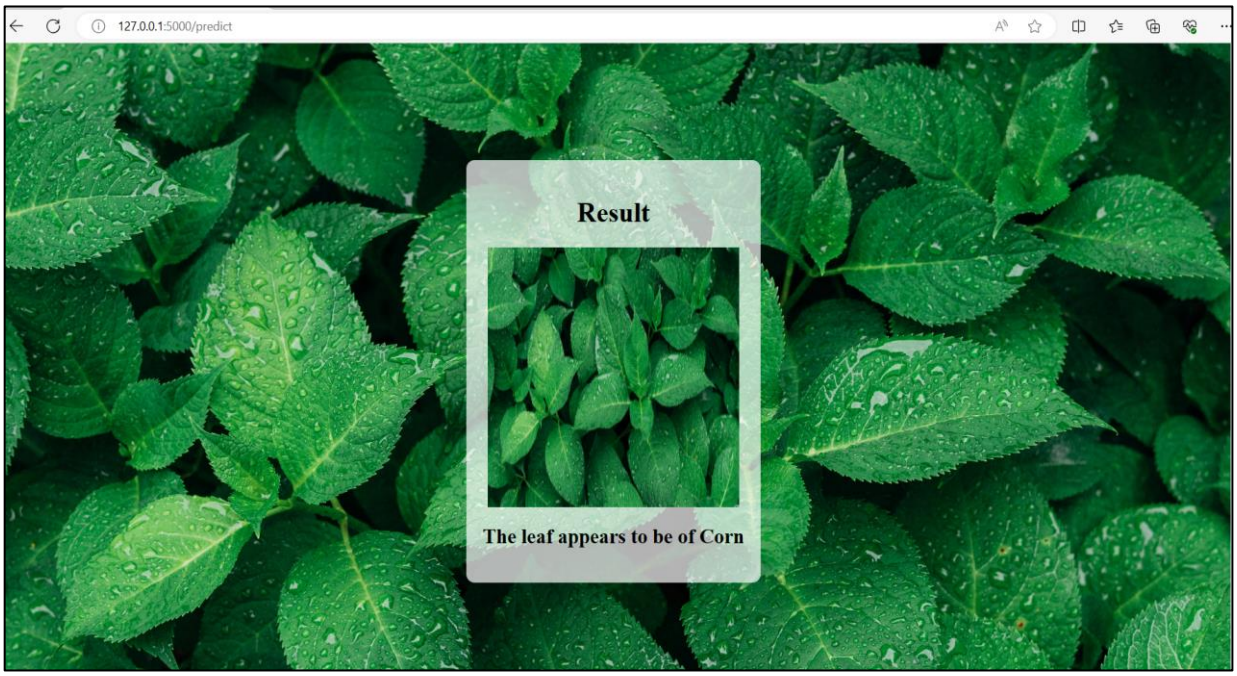


Purchases Page:

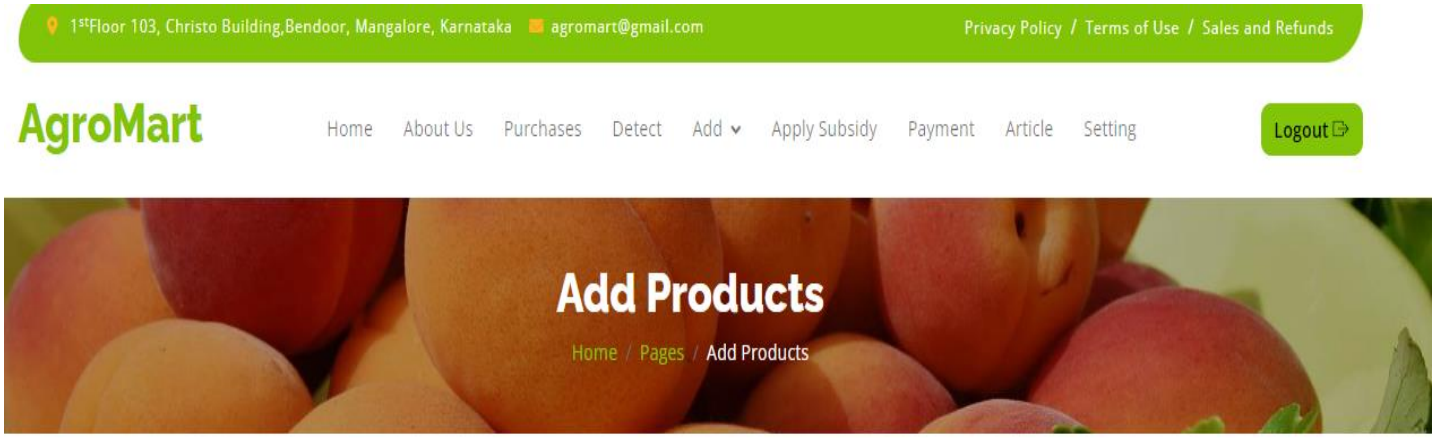


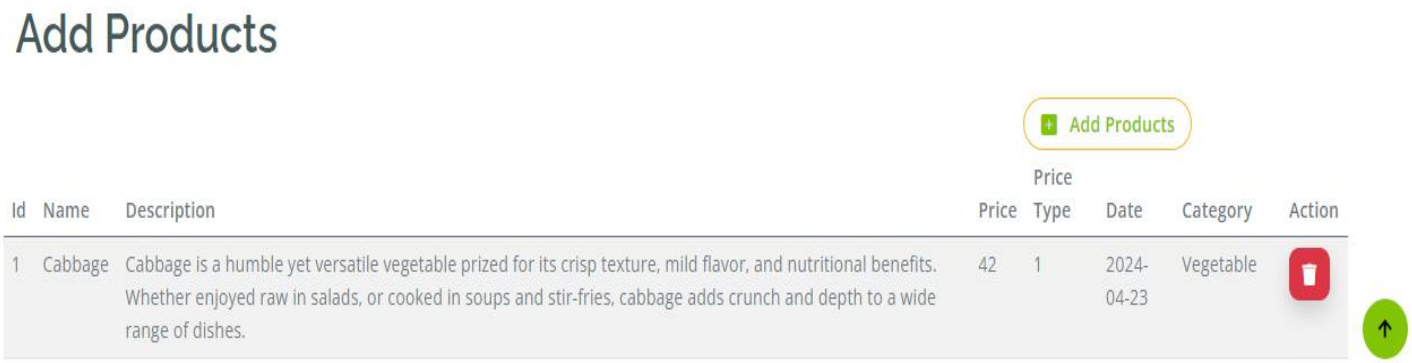
Detect Page:

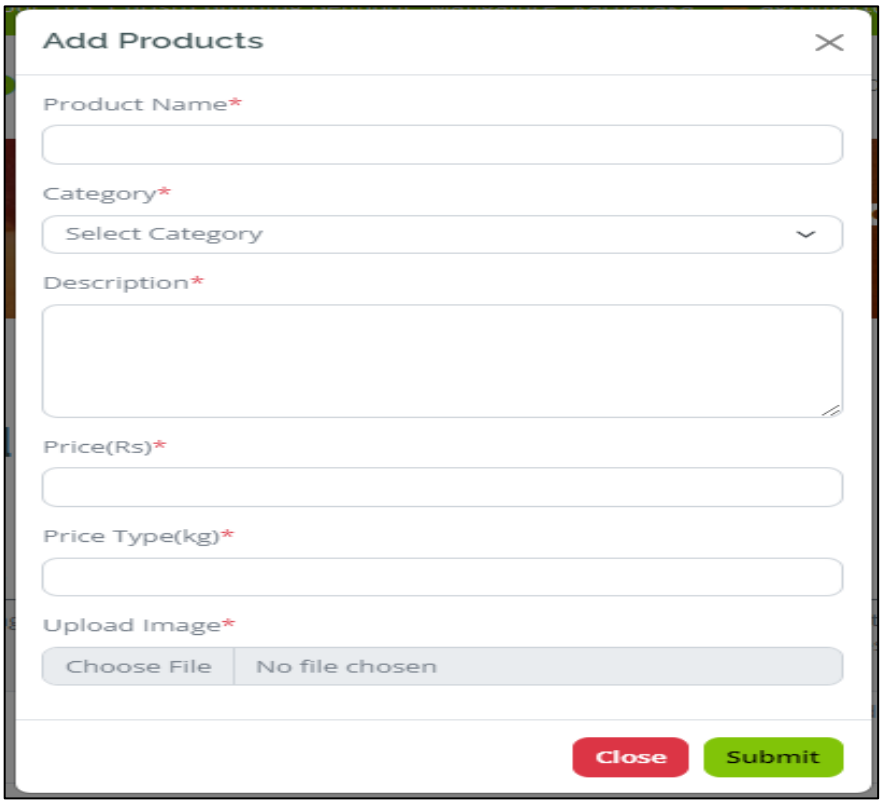

Botanical Identity via Leaves
Upload an image:
Choose File
No file chosen
Upload and Predict


127.0.0.1:5000/predict
Result
The leaf appears to be of Corn

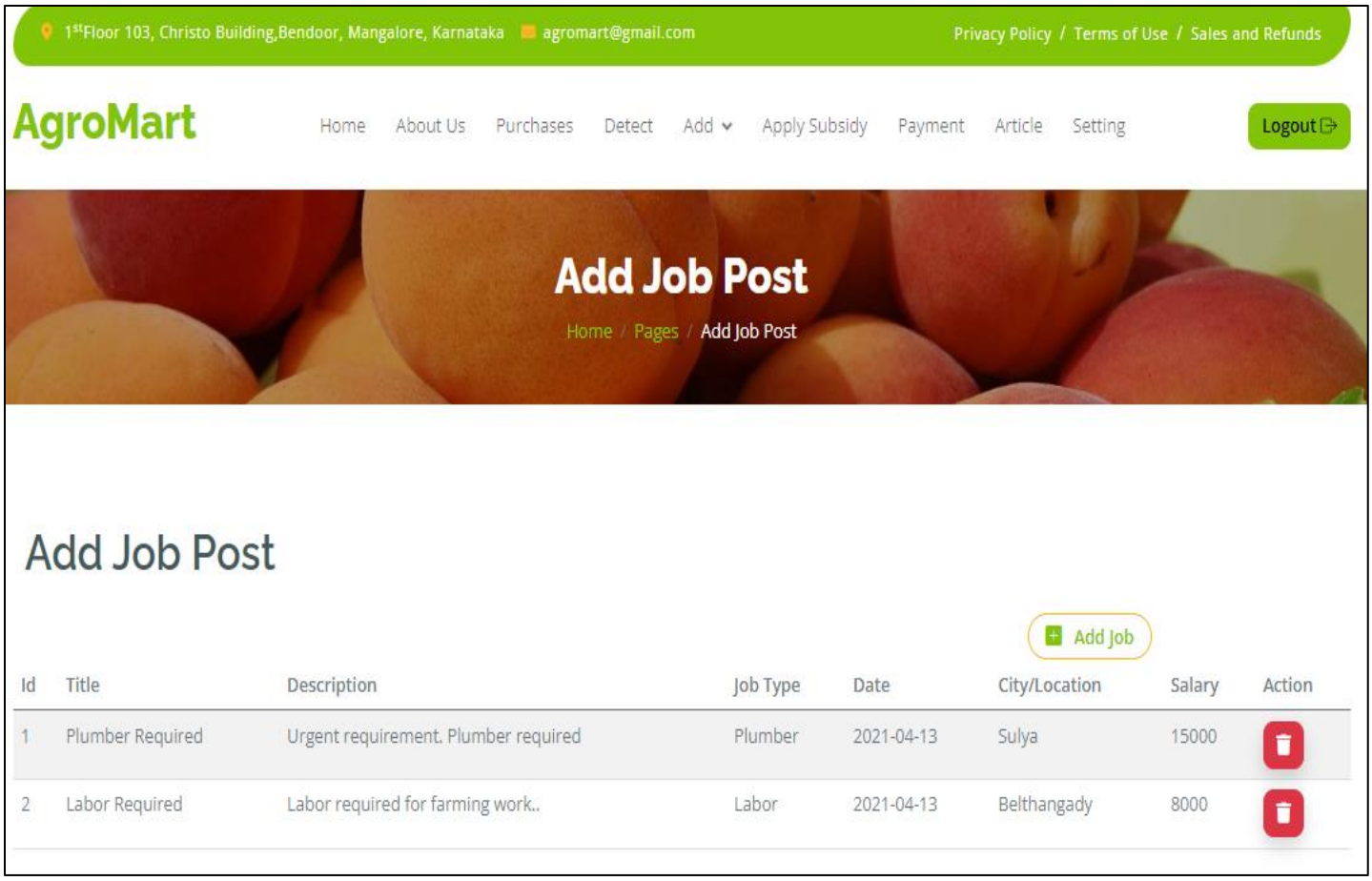
Add Products:

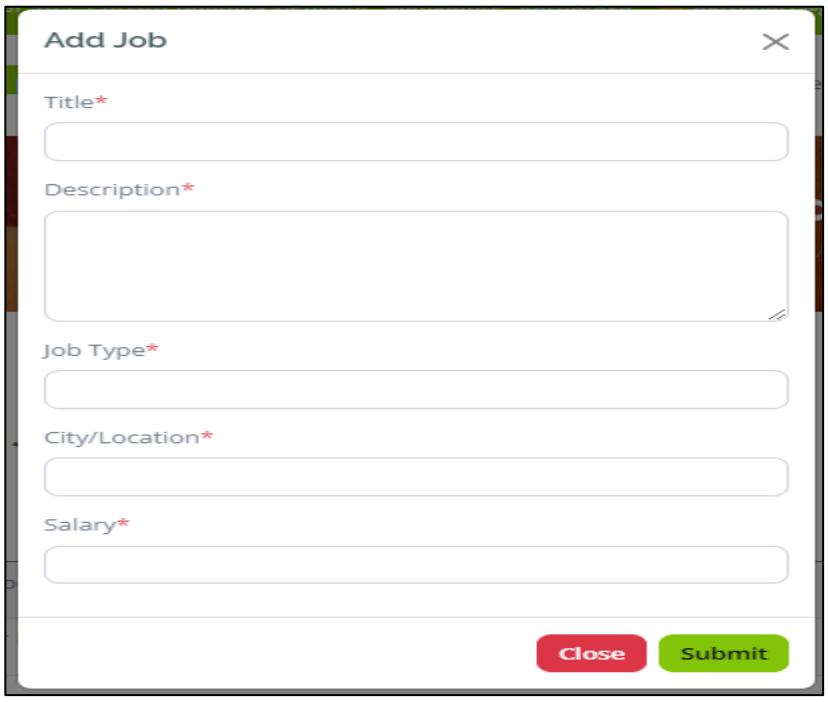




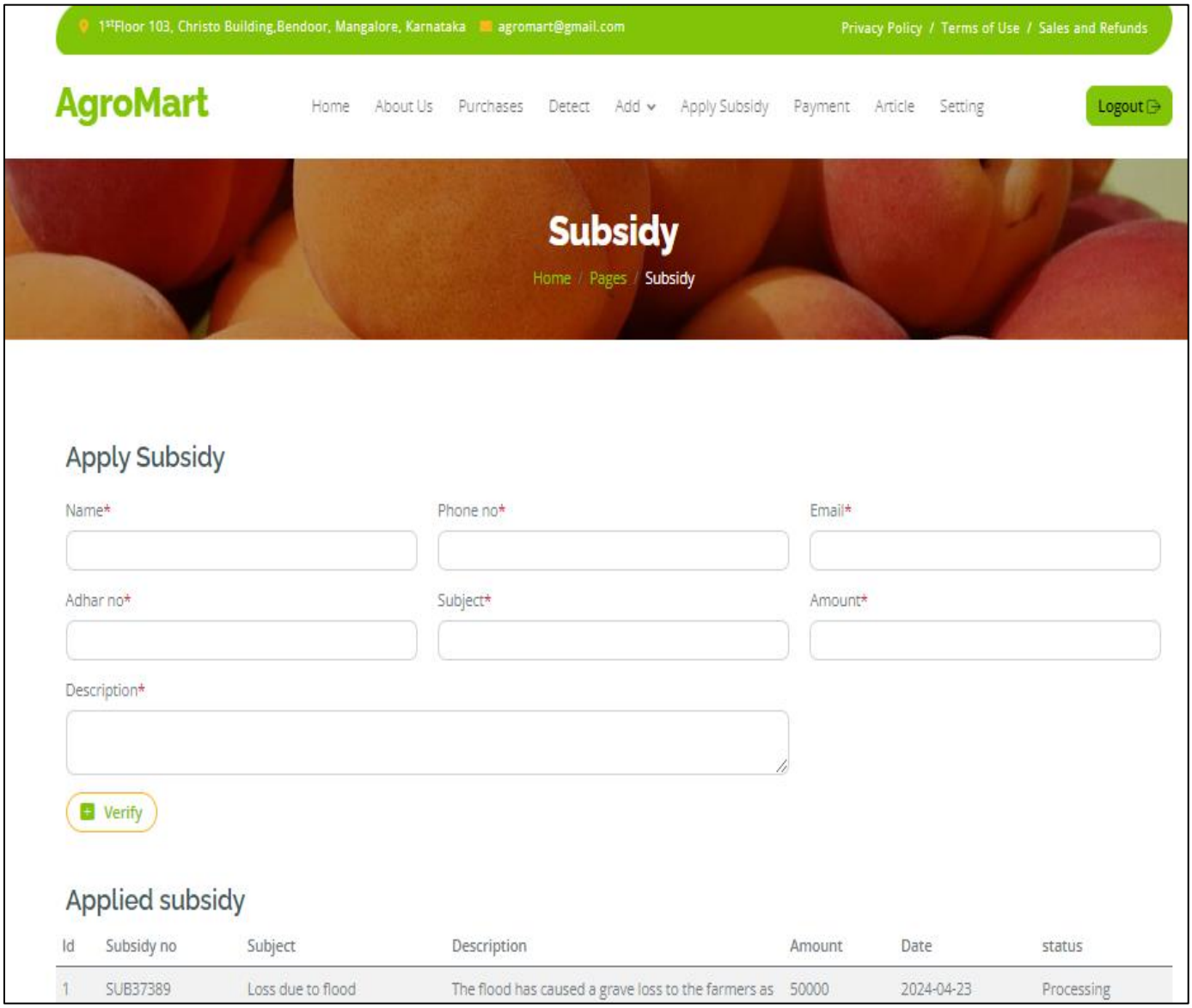


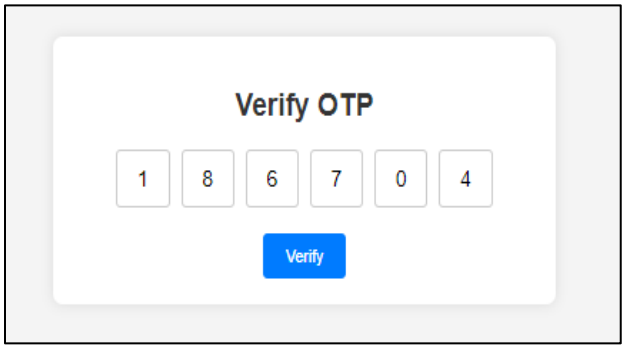
Post Jobs:

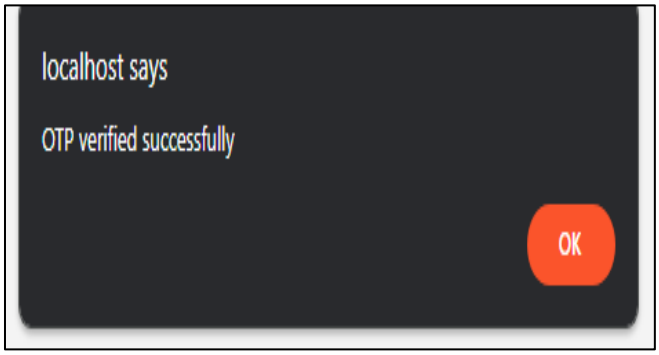

1st Floor 103, Christo Building,Bendoor, Mangalore, Karnataka
agromart@gmail.com
Privacy Policy / Terms of Use / Sales and Refunds
AgroMart
Home
About Us
Purchases
Detect
Add
Apply Subsidy
Payment
Article
Setting
Logout
Add Job Post
Home / Pages / Add Job Post
Add Job Post
Add Job
Id
Title
Description
Job Type
Date
City/Location
Salary
Action
1
Plumber Required
Urgent requirement. Plumber required
Plumber
2021-04-13
Sulya
15000
2
Labor Required
Labor required for farming work..
Labor
2021-04-13
Belthangady
8000


Add Job
Title*
Description*
Job Type*
City/Location*
Salary*
Close
Submit

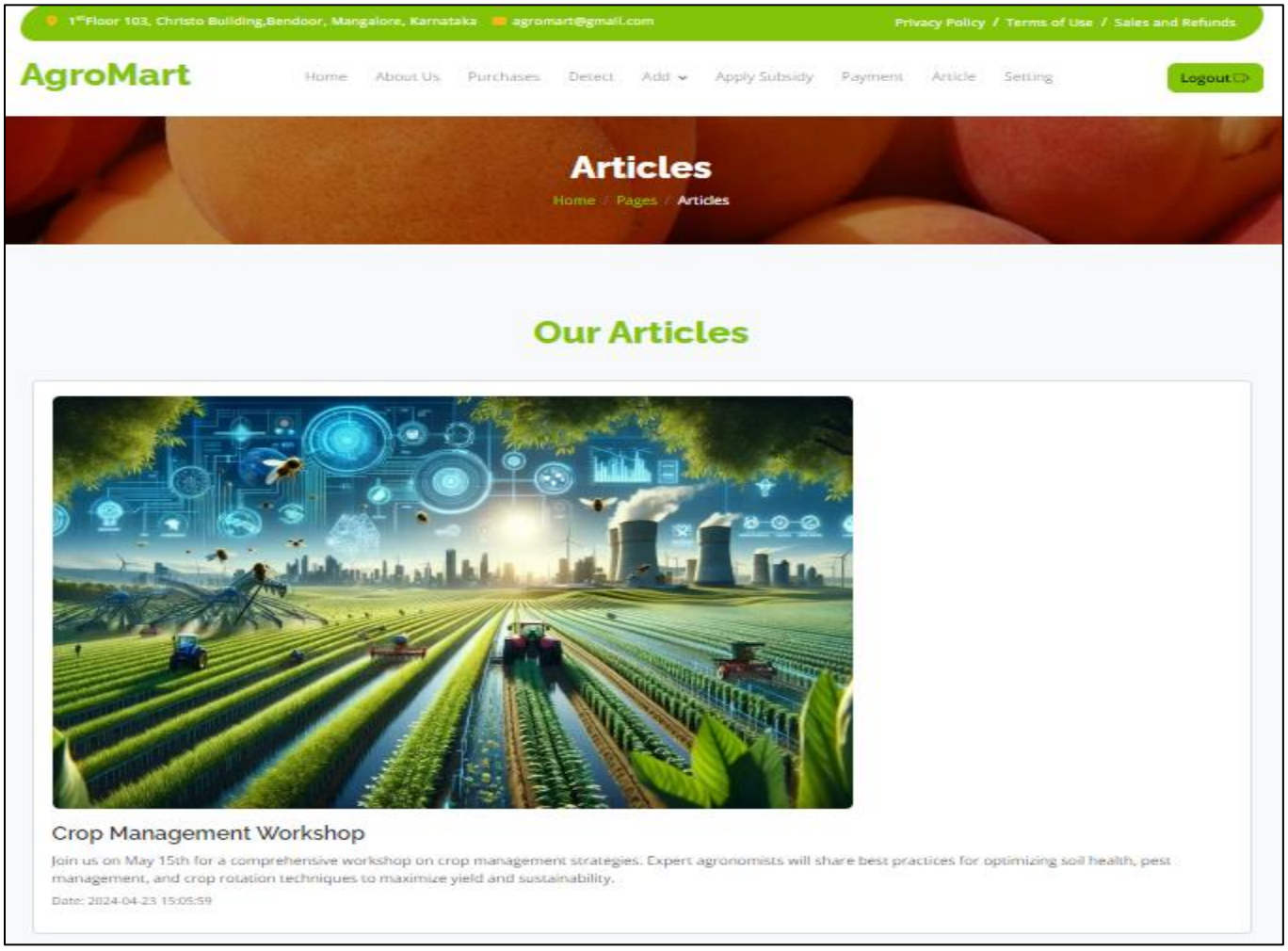
Apply Subsidy Page:



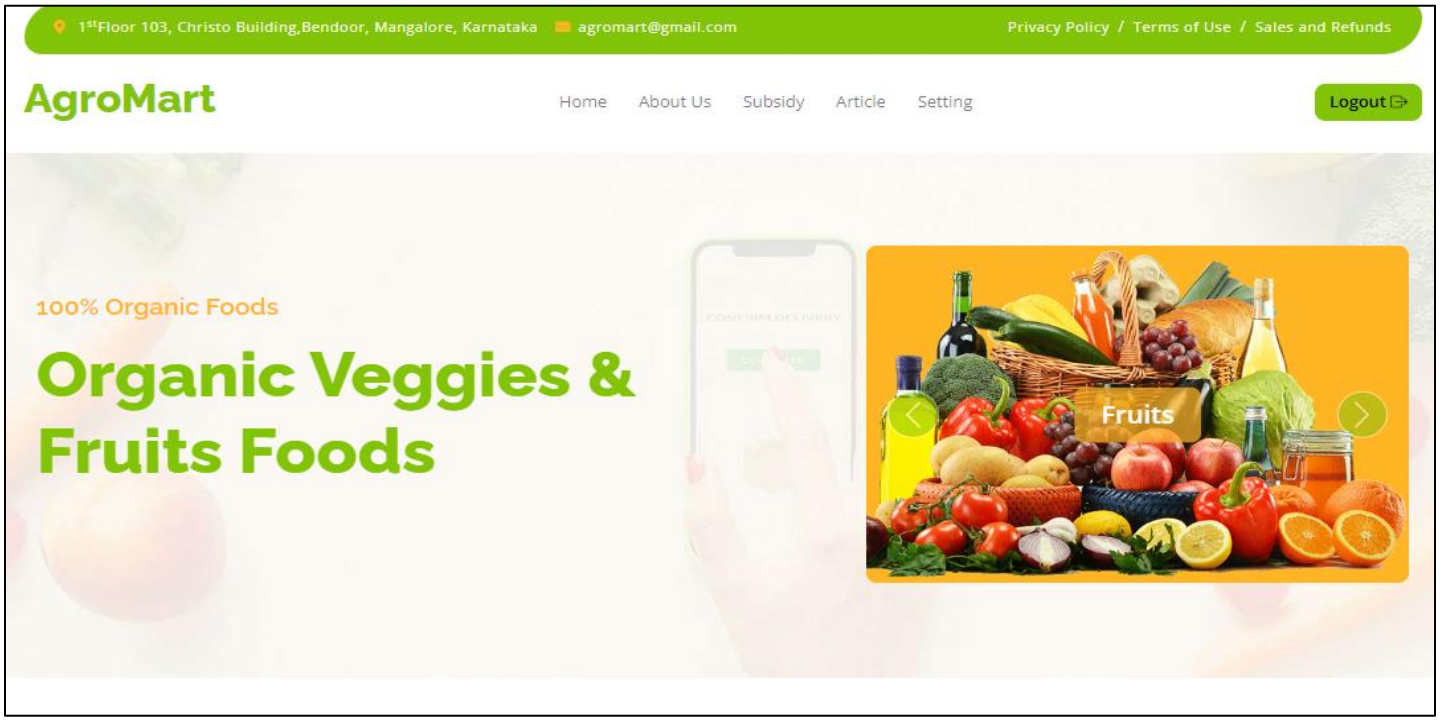




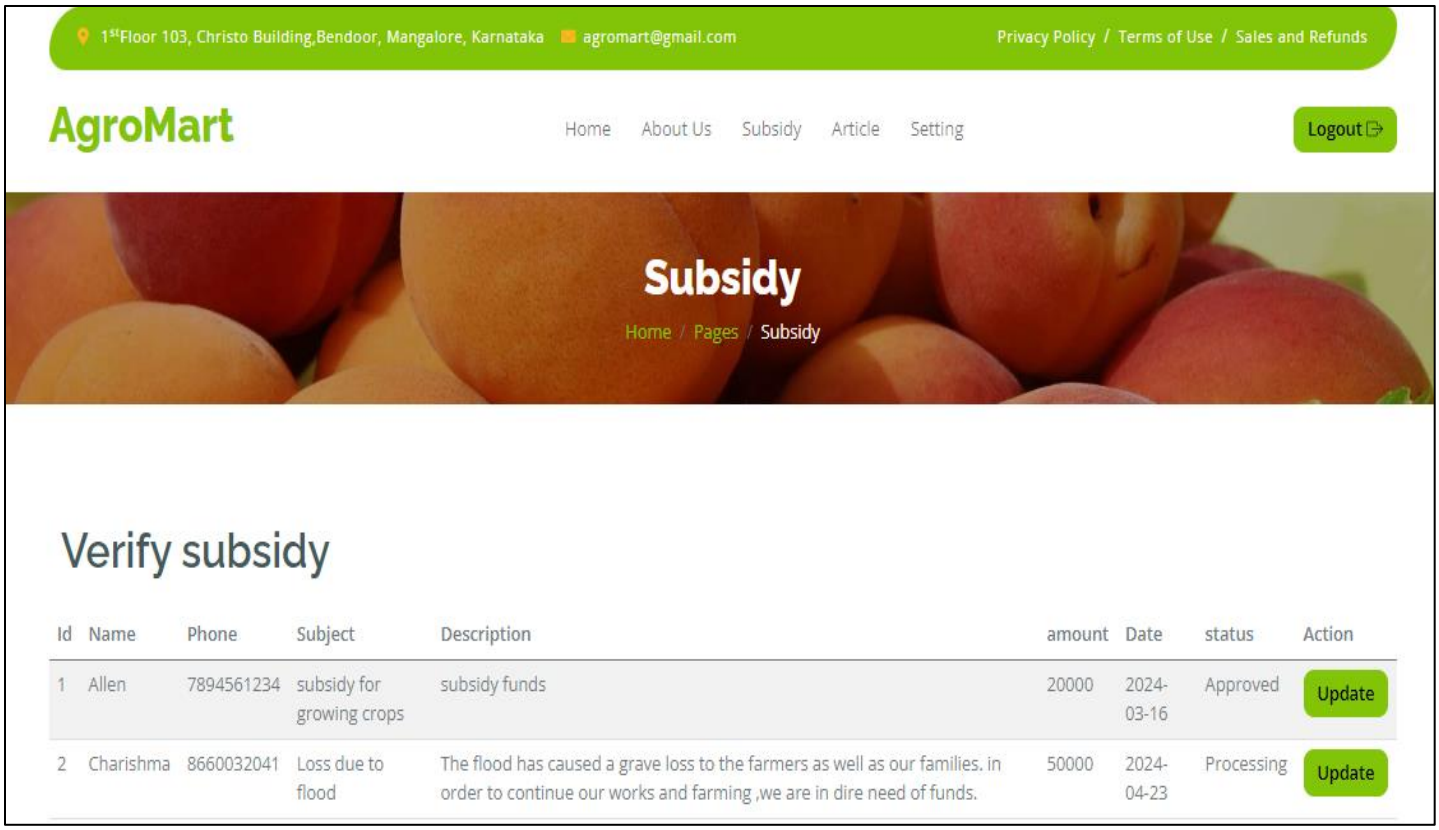
Articles:



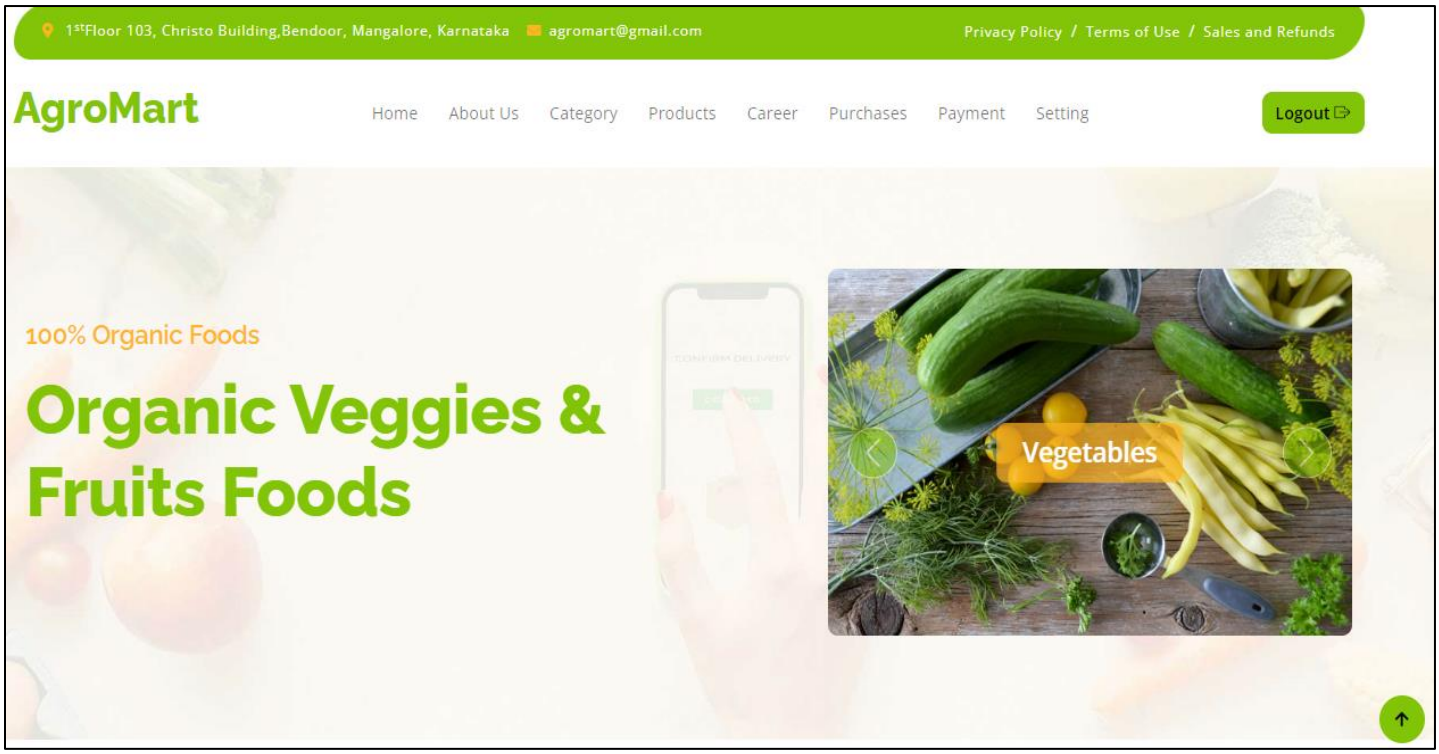
NGO Home Page:



Subsidy Verification Page:

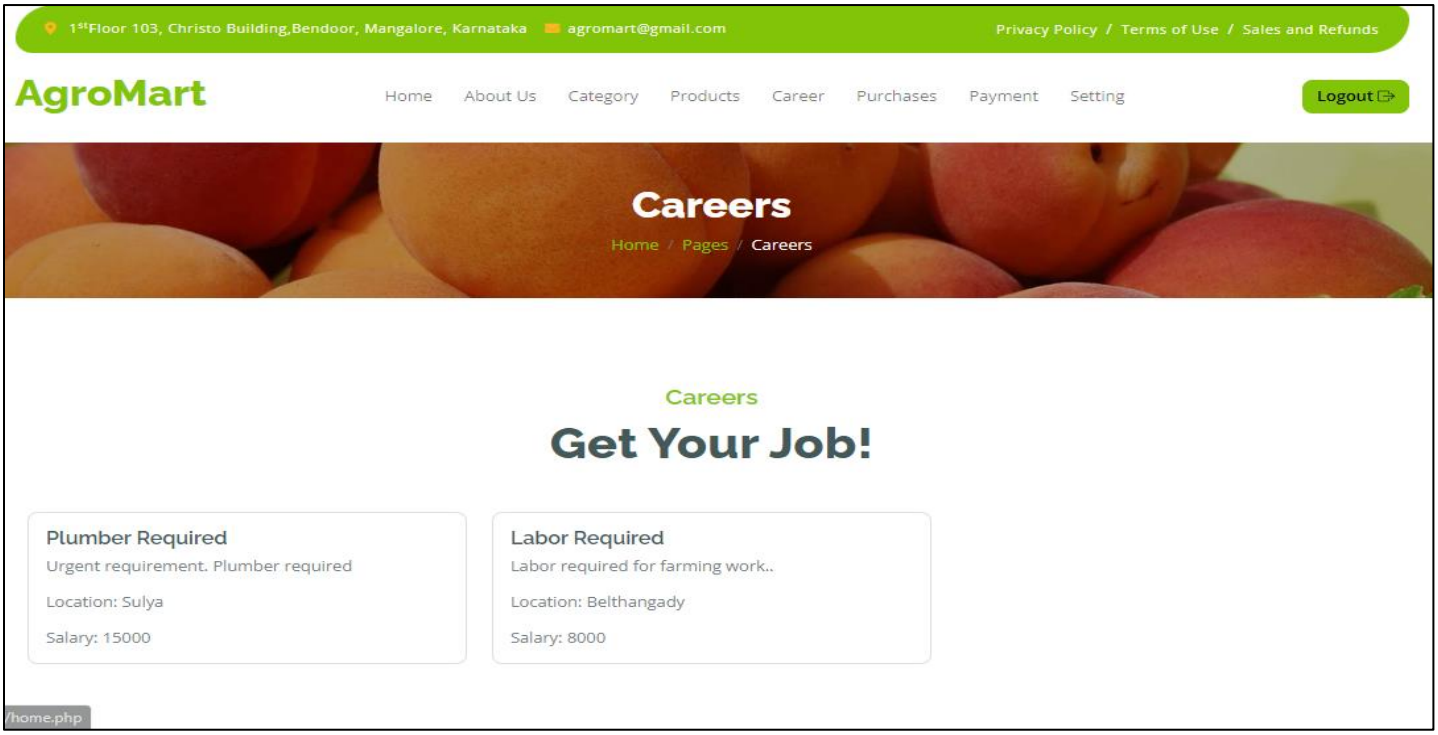


User Home Page:

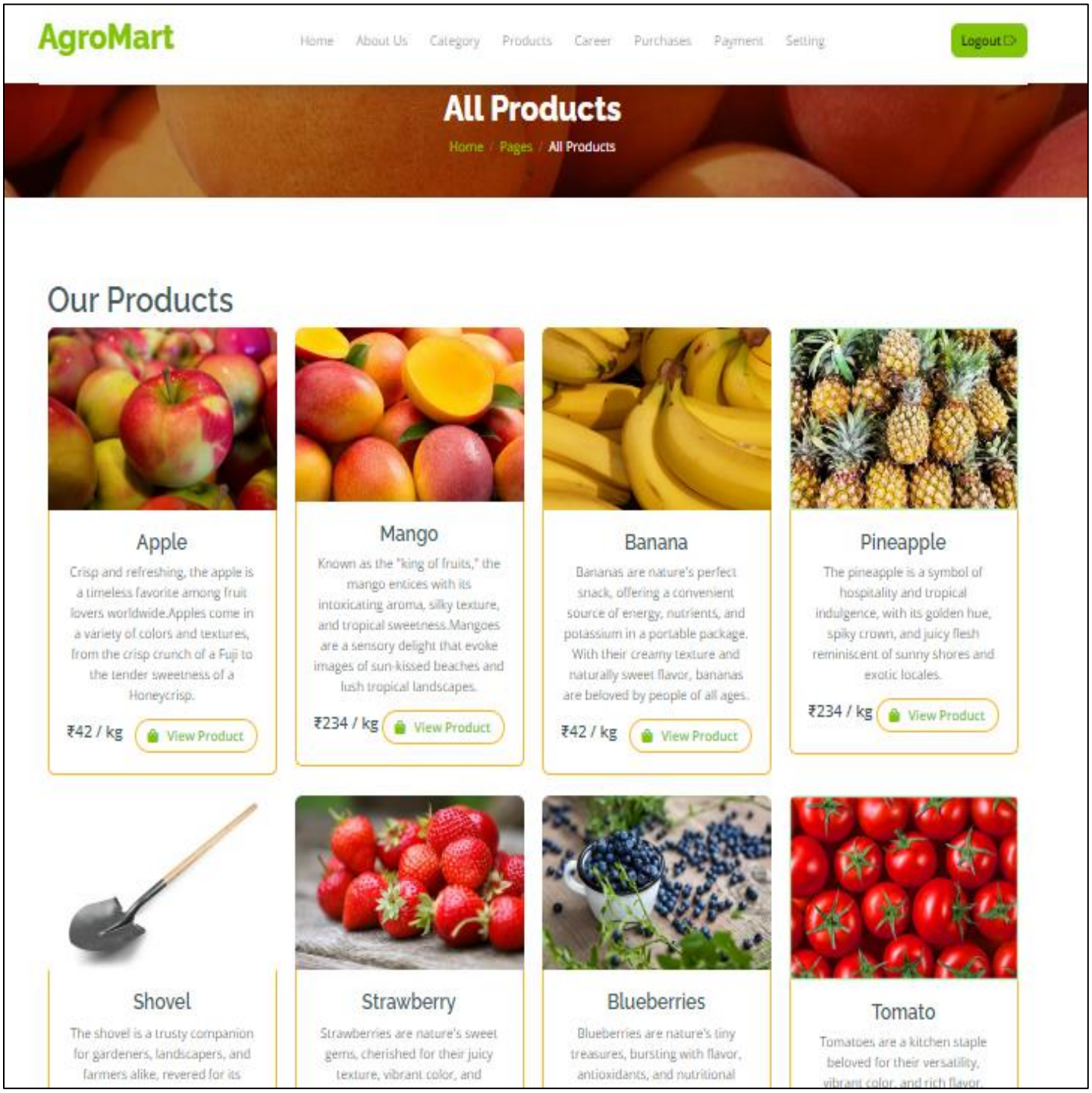


Category – Fruits Page:

Careers Page:



All Products:



TESTING

TESTING

8.1 Introduction:

Software testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. This includes the process of executing the program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification.

The testing phase consists of evaluating the software that has been developed in order to conform that it produces the output required in a safe and efficient manner. In this phase inherent errors that occur, have to be handled and the user should be informed so that he/she can follow the guidelines and instructions and get around the error and obtain the output.

During testing, the program to be tested is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as expected. Due to its approach, dynamic testing can only ascertain the presence of errors in the program the exact nature of the errors is not usually decided by testing.

Testing forms the first step in determining the errors in a program. Clearly the success of testing in revealing errors in programs depends critically on the test cases. Because code is the only product that can be executed and whose actual behavior can be observed, testing is the phase where the errors remaining from all the previous phases must be detected.

The program to be tested is executed with a set of test cases and the output of the program for the test cases are evaluated to determine if the programming is performing as expected. Testing forms the first step in determining errors in a program. The success of testing in revelling errors in programs depends critically on the test cases.

8.2 Objectives Of Testing

The objectives of testing are

- Testing is a process of executing a program with the intent of finding errors.
- A successful test case is one that discovers an as of yet and discovered error. System testing is a stage of implementation which is aimed at ensuring that the system works accurately and efficiently as per the user need, before the live operation commences as stated before, testing is vital to the success of a system system testing makes a logical assumption that if all parts of system are correct the goal will successfully be achieved. A series of tests are performed before the system is ready for the user acceptance test.

There Are Two Types Of Software Testing:

- **Black Box Testing**: Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.
- **White box testing**: This testing is based on knowledge of the internal logic of an application's code. Also known as glass box testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths and conditions.

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be pragmatic as 'for condition x your derived result is y'; where as other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.
White box testing is applicable at the unit, integration and system levels of the software testing.

8.3 Testing Methodology

The different types of testing are as follows:

8.3.1 Unit Testing:

Unit testing focuses efforts on the smallest unit of software design this is known as module testing or white box testing, the modules are tested separately. The test is carried out during the programming stage itself. In this step, each module is found to be working satisfactorily as regards to the expected output from the module.

8.3.2 Integration Testing:

In integration testing the different units of the system are integrated together to form the complete system. This type of testing checks the system as a whole to ensure that it is doing what it's supposed to do the testing of an integrated system can be carried out top-down, bottom-up or Big-Bang. In this type of testing some parts are tested with white box testing and some with black box testing techniques. This type of testing plays a very important role in increasing systems productivity. We have checked the system by using integration testing techniques.

8.3.3 System Testing:

Apart from testing the system to validate the functionality of software against the requirements. It is also necessary to test the non functional aspect of the system, some examples of non functional tools include tests to check the performance data security usability volume load and stress that we have used in a project to test the various module. System testing consists of the following steps:

- Program(s) testing
- String testing
- System Testing
- System Documentation
- User Acceptance test

8.3.4 Field Testing:

The special type of testing may be very important in some projects. Here the system is tested in actual operational surroundings the interfaces with other systems and the real world are checked. This type of testing is very rarely used so far our project is concerned we haven't tested a project using field testing.

8.3.5 Acceptance Testing:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and retest system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user in

this project. It is the job of the developer of the planner to check the system that whether the system fulfills the goals or not.

8.4 Testing Criteria:

Testing for valid user name:

Test case	Input	Test description	Output
1	User name starts with number	User name cannot start with number	Must Enter Characters
2	User name is left blank	User name cannot be left blank	Must Enter username

Testing for valid password:

Test case	Input	Test description	Output
1	Password is left blank	Password cannot be blank	Must Enter password
2	Invalid password entered	Valid password must be entered	Password mismatch

Testing for valid Email address:

Test case	Input	Test description	Output
1	Email address is not in Correct format	Email address Should have Correct format	Invalid Expression
2	Email address with space	Email address cannot have space	Invalid Expression
3	Email is left blank	Email cannot be blank	Must Enter Email ID

Testing for data insertion:

Test case	Input	Test description	Output
1	Mandatory fields left empty	Mandatory fields cannot be left empty	Must enter data

2	Duplicate entry	Duplicate entry not allowed	Appropriate error message
3	Input without above faults	Valid input	Record inserted Successfully

Testing for deletion:

Test case	Input	Test description	Output
1	Deletion attempted when no entries selected	Entries, if not selected, cannot be deleted	Select any field
2	Valid deletion without above faults	Valid deletion	Record deleted Successfully

Testing for phone number:

Test case	Input	Test description	Output
1	Phone number entered with alphabets	Phone number Cannot have alphabets	Enter only numbers
2	Phone number entered is more than 10 digits	Phone number with more than 10 digits cannot be entered	Invalid phone number

Testing for valid duration:

Test case	Input	Test description	Output
1	Invalid duration	“From” date must be smaller than “To” date	“Valid from” date must be smaller than “Valid To” date

Testing for change password:

Test case	Input	Test description	Output
1	Any field left blank	All fields are compulsory	Must enter Password
2	Invalid password	Valid password must be entered	Enter correct password
3	Retyped password does not match	Retyped password must match	Password mismatch
4	Valid input	Valid input	Password changed successfully

Other cases:

Test case	Input	Test description	Output
1	Click on logout	Application Will close	Application closes and redirects to login page
2	Click on clear	Entering details	Record is not submitted to the database and Text boxes cleared
3	Click on save	Entering details	Record is submitted to the database
4	Click on save	Not entering the details	Appropriate error message is displayed
5	Click on date control	Selecting date	Date is displayed
6	Click on Drop down list	Selecting data	Data is displayed

CONCLUSION

CONCLUSION

In conclusion, Agromart stands as a beacon of innovation in the agricultural sector, bridging the gap between farmers and consumers through its dynamic e-commerce platform. By empowering farmers to showcase and sell their products directly to customers online, Agromart revolutionizes traditional agricultural marketing channels. This direct connection not only eliminates intermediaries, reducing costs and increasing profits for farmers, but also provides consumers with access to fresher, higher-quality produce.

Moreover, Agromart's platform fosters transparency and trust within the agricultural supply chain. Customers can trace the origin of their purchases back to the source, gaining insight into the farming practices and ethical standards employed by the producers. This transparency not only enhances consumer confidence but also incentivizes farmers to uphold sustainable and responsible farming methods.

In essence, Agromart's commitment to facilitating direct sales between farmers and consumers embodies the future of agriculture – one that is sustainable, transparent, and mutually beneficial. By leveraging technology to streamline the agricultural supply chain, Agromart not only enhances the livelihoods of farmers but also enriches the lives of consumers by providing access to wholesome, farm-fresh products.

BIBLIOGRAPHY

BIBLIOGRAPHY

- https://themewagon.com/ - Used for the template of the project.
- https://colorlib.com/ - Used for the template of the registration form.
- https://chat.openai.com/ - Used to generate content for the webpages.
- https://www.pinterest.com/ - Used to download images for the website.