

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Jnana Sangama, Belagavi-590010



DBMS MINI PROJECT REPORT 2

ON

“HarvestHub: Empowering Sustainable Agriculture for Quality Food Supply.”

Submitted in partial fulfilment for the requirements for the fifth semester

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

For the Academic Year 2023-2024

Submitted by:

PRAVEENKUMAR
RAKSHAK R GOWDA
THOKCHOM MOBISANA

1MV21CS071
1MV21CS081
1MV21CS113

Under the guidance of:

Dr. Suma Swamy

Professor, Department of CSE

Sir M. Visvesvaraya Institute of Technology, Bengaluru



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
HUNASAMARANAHALI, BENGALURU-562157

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

Krishnadevaraya Nagar, International Airport Road,
Hunasmaranahalli, Bengaluru – 562157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

It is certified that the **DBMS Mini Project work** entitled **HarvestHub: Empowering Sustainable Agriculture for Quality Food Supply.** " is carried out by **PRAVEENKUMAR (1MV21CS071), RAKSAK R GOWDA (1MV21CS081), THOKCHOM MOBISANA (1MV21CS113)**, bonafide students of **Sir M Visvesvaraya Institute of Technology** in partial fulfillment for the 5th semester for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering of the **Visvesvaraya Technological University, Belagavi** during the academic year **2023-2024**. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

Name & Signature
of Guide

Name & Signature
of HOD

Name & Signature
of Principal

Dr . Suma Swamy
Prof & Internal Guide
Dept. of CSE, Sir MVIT
Bengaluru - 562157

Dr. Anitha T N
HOD, Dept. of CSE,
Sir MVIT
Bengaluru - 562157

Dr. Rakesh S G
Principal,
Sir MVIT
Bengaluru – 562157

External Examination:

Name of Examiner

Signature with Date

1)

2)

DECLARATION

We hereby declare that the entire project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru

Date:

Signature of Students:

PRAVEENKUMAR
(1MV21CS071)

THOKCHOM MOBISANA
(1MV21CS113)

RAKSHAK R GOWDA
(1MV21CS081)

ACKNOWLEDGMENT

It gives us immense pleasure to express our sincere gratitude to the management of **Sir M. Visvesvaraya Institute of Technology**, Bengaluru for providing the opportunity and the resources to accomplish our project work in their premises.

On the path of learning, the presence of an experienced guide is indispensable and we would like to thank our guide **Dr Suma Swamy**, Professor, Department of CSE, for her invaluable help and guidance.

Heartfelt and sincere thanks to **Dr. Anitha T N**, HoD, Department of CSE, for her suggestions, constant support and encouragement.

We would also like to convey our regards to **Dr. Rakesh S G**, Principal, Sir MVIT for providing us with the infrastructure and facilities needed to develop our project.

We would also like to thank the staff of Department of Computer Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would like to thank all our friends for their help and suggestions without which completing this project would not have been possible.

- **PRAVEENKUMAR** **1MV21CS071**
- **RAKSHAK R GOWDA** **1MV21CS081**
- **THOKCHOM MOBISANA** **1MV21CS113**

ABSTRACT

The growth in popularity of social networking leads to the problematic usage. An increased number of social network mental disorders , such as Cyber-Relationship Addiction, Information Overload , have been recently noted. Symptoms of these mental disorders are usually observed less today, resulting in delayed clinical intervention . In this paper, we argue that mining online social behavior provides an opportunity to actively identify depression at an early stage. It is challenging to detect disorders because the mental status cannot be directly observed from online social activity logs.

Our approach, new and innovative to the practice of mental depression detection, does not rely on self- revealing of those mental factors via questionnaires in Psychology. Instead, we propose a machine learning framework that exploits features extracted from social network data to almost accurately identify potential case

TABLE OF CONTENTS

Chapters	Page no.
1. Introduction	1-2
1.1 Introduction to HarvestHub	1
1.2 Scope of Project	1
1.3 Objectives Of Project	1
1.4 Project Framework	2
1.5 Data and Information	2
2. Front End and Back End	3
HTML(Front-End)	
CSS	
Bootstrap	
JavaScript	
PHP (Back-End)	
MySQL	
3. Specifications	4-5
3.1 Hardware Requirements	4
3.2 Software Requirements	4
3.3 Functional Requirements	5
3.4 Non-Functional Requirements	5
4. System Design	6-9
4.1 UML Diagram	6-7
4.2 ER Diagram	8-9
5. Conceptual Design	10-16
5.1 Create Commands	10-12

5.2 Insert Commands and Data	12-13
5.3 SQL Queries	13-16
6. Front End Design, Connectivity and Implementation	17-23
6.1 User Registration and Authentication	17-18
6.2 User Profile Management	18-19
6.3 Product Catalog and Search	19-20
6.4 Shopping Cart Management	20-20
6.5 Blog Creation and Management	21-21
6.6 Integration with phpMyAdmin and Database Connectivity	22-23
7. Source Code-Front End	24-40
7.1 User Registration and Authentication Code	24-26
7.2 User Profile Management Code	26-28
7.3 Product Catalog and Search Code	28-32
7.4 Shopping Cart Management Code	32-35
7.5 Blog Creation and Management Code	35-40
8. Results, Discussion & Screenshots	41-49
8.1 Results and Discussion	41-41
8.2 Screenshots	42-49
9. Conclusion & Scope	50-50
9.1 Conclusion	50
9.2 Scope	50
References & Weblinks	51

LIST OF FIGURES

Fig. No.	Description	Page no.
4.1	UML Diagram	6
4.2	ER Diagram	8
8.2.1	Home Page	42
8.2.2	Farmer/Buyer registration Page	42
8.2.3	Farmer/Buyer Profile Page	43
8.2.4	Product Search Page	43
8.2.5	MyCart Page	44
8.2.6	Product Purchase Page	44
8.2.7	Product View Page	45
8.2.8	Transaction details Page	45
8.2.9	Add Product Page	46
8.2.10	Blog Page	46
8.2.11	Profile page before trigger	47
8.2.10	Product purchase Trigger	47
8.2.11	Profile page after trigger	47
8.2.10	Product purchase summary report	48
8.2.10	Blog review summary report	48
8.2.11	Log Out Page	49

CHAPTER 1

INTRODUCTION

1.1 Introduction to HarvestHub

Agriculture is the farmer system where they can plan, monitor and analyze the activity of the farmers production system. It manages farmer operation with one system and organizes data in one place. It helps smart farmers become even smarter. This creates in partnership with growers and buyers. It inspire farmer to producers and buyers to consume fresh goods.

AgroCulture System will make better connection among Farmers and Buyers ensure quality food. Standardize and increase efficiency of Farm Buddy process.

1.2 Objectives of the Project

The specific objectives of the project include:

- To provide qualitative foods to the buyers.
- Implementing an automated/online AgroCulture system.
- To inspire farmer to produce quality goods and supply to the buyers.
- Eco-friendly farming system.

1.3 Scope of the Project

It is focused on studying the existing system of Agriculture in and to make sure that the peoples are getting quality fresh goods. This is also will produce:

- Less effort and less labour intensive, as the primary cost and focus primary on creating, managing, and running a secure quality food supply.
- Increasing number of buyers as individuals will find it easier and more convenient to buy goods.
- Easy management. Agriculture management system 2021-2022

Methodology/Procedure

For the development of project the designing of database was done on PHPMYADMIN, back end was coded in basic PHP and for frontend we used the same basic PHP codes. Software methodologies are concerned with the process of creating software – not so much the technical

side but the organizational aspects. Several software development approaches have been used since the origin of information technology.

1.4 Project Framework

A framework is a standardized set of concepts, practices, and criteria for dealing with a common type of problem, which can be used as a reference to help us approach and resolve new problems of a similar nature.

The aim of framework is to provide a common structure so that developers don't have to redo it from scratch and can reuse the code provided. In this way, frameworks allow's us to cut out much of the work and save a lot of time.

1.5 Data and Information

Data collection plays an important role in a projects succession and also it plays an inevitable role in the timely completion of the project. The data in the project includes contact information of the clients and their respective feedbacks/complaints which is stored in a database. To assure safety, only the admin has proper access to the information provided by the clients.

Primary Source of Data

- Primary data are the first hand data. The necessary information was collected from day to day observation, problems, instructions of supervisor. Queries and personal discussion with the staff of the organization.
- Observation of working environment

Secondary Source of Data

The Secondary sources of data were collected in order to achieve the real and fact data as far as available.

The major sources of secondary data are as follows:

- Annual reports of the concerned organization
- Related website

CHAPTER 2

TECH STACK

2.1 Front End

- HTML (HyperText Markup Language):

Description: HTML is the standard markup language for creating web pages. It provides the structure and content of web pages.

- CSS (Cascading Style Sheets):

Description: CSS is used for styling and layout purposes. It allows you to control the appearance of HTML elements, enhancing the visual presentation of your web pages.

- JavaScript:

Description: JavaScript is a dynamic programming language that runs in the browser, enabling interactive and dynamic features on web pages. It can be used to manipulate the Document Object Model(DOM) and handle user interactions.

2.2 Back End

- PHP:

Description: PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development. It enables the creation of dynamic and interactive web pages. PHP is known for its versatility, supporting various databases and is an essential component in building robust and scalable web applications.

- MySQL:

Description: MySQL is an open-source relational database management system (RDBMS) that stores and retrieves data. It is widely used in conjunction with PHP to create dynamic web applications. MySQL is known for its reliability, speed, and ease of use. It supports SQL, providing a powerful and efficient means to manage and query data in web applications.

CHAPTER 3

Specifications

3.1 Hardware Requirements :

- Server:
 - o A dedicated server or cloud instance with sufficient processing power and memory to handle web application requests efficiently.
- Storage:
 - o Adequate storage space for database storage and any file uploads.
- Network Infrastructure:
 - o A reliable internet connection to ensure seamless communication between the server and clients.
- Backup System:
 - o Regular backup system to prevent data loss in case of hardware failures or other unforeseen issues.

3.2 Software Requirements :

- Operating System:
 - o Linux-based OS (e.g., Ubuntu, CentOS) for the server environment.
- Web Server:
 - o Nginx or Apache for serving the web application.
- Database Management System:
 - o MySQL for storing and retrieving data efficiently.
- Programming Language:
 - o PHP for backend development
- Frontend Technologies:

- o HTML, CSS, and JavaScript for building the user interface.
- Version Control:
 - o Git for version control and collaborative development.
- Integrated Development Environment (IDE):
 - o Visual Studio Code

3.3 Functional Requirements :

- XAMPP (PhpMyAdmin):
 - o Manage MySQL server and databases.
 - o Browse, drop, create, copy, rename, and alter databases, tables, views, columns, and indexes.
 - o Execute, edit, and bookmark SQL statements.
 - o Import/export data to various formats (CSV, XML, PDF, etc.).
 - o Administer multiple servers, manage users, and privileges.
 - o Support InnoDB tables and foreign keys.
 - o Create, edit, export, and drop stored procedures, functions, events, and triggers.
 - o Synchronize databases on the same and remote servers.

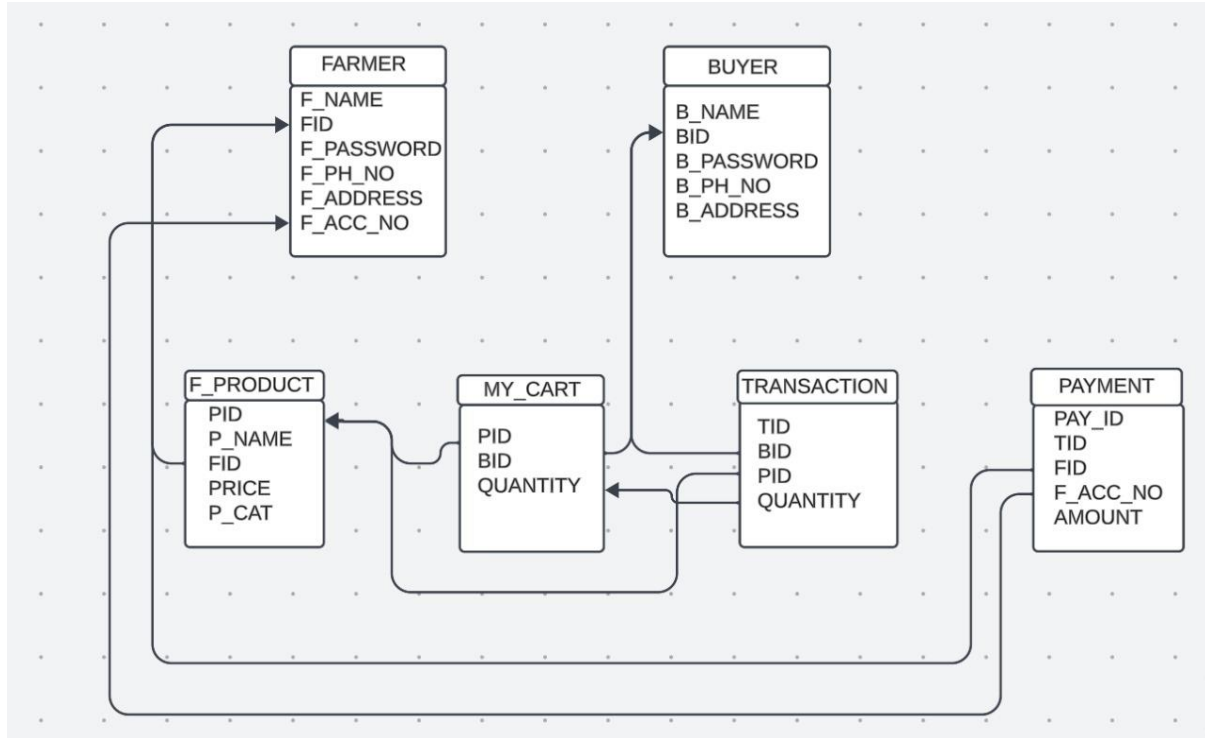
3.4 Non-Functional Requirements :

- PHP:
 - o Requires PHP 5.2.0 or newer with session support, SPL extension, and JSON support.
 - o Needs PHP zip extension for ZIP file support.
 - o Recommends installing mbstring and ctype extensions for multibyte string support.
 - o Requires GD2 support for displaying inline thumbnails of JPEGs.
 - o Suggests mcrypt extension for the "cookie" authenticated method, especially for 64-bit machine.

CHAPTER 4

System Design

4.1 UML Diagram



4.1 UML Diagram of HarvestHUB

PLATFORM USED FOR UML

Lucidchart is a cloud-based diagramming tool for creating UML diagrams and other visual representations. It offers real-time collaboration, an intuitive interface, and integration with popular productivity tools like Google Workspace and Microsoft Office. Users can easily design and share flowcharts, organizational charts, and more online.

DESCRIPTION

1. Entities:

- **FARMER** : Represents individuals who want to sell the agriculture products.
- **BUYER** : Represents individual who buy products from farmers.
- **TRANSACTION** : Amount paid by the buyer.
- **PAYMENT** : Amount paid to the farmer.
- **F_PRODUCTS** : Represents products uploaded by users.

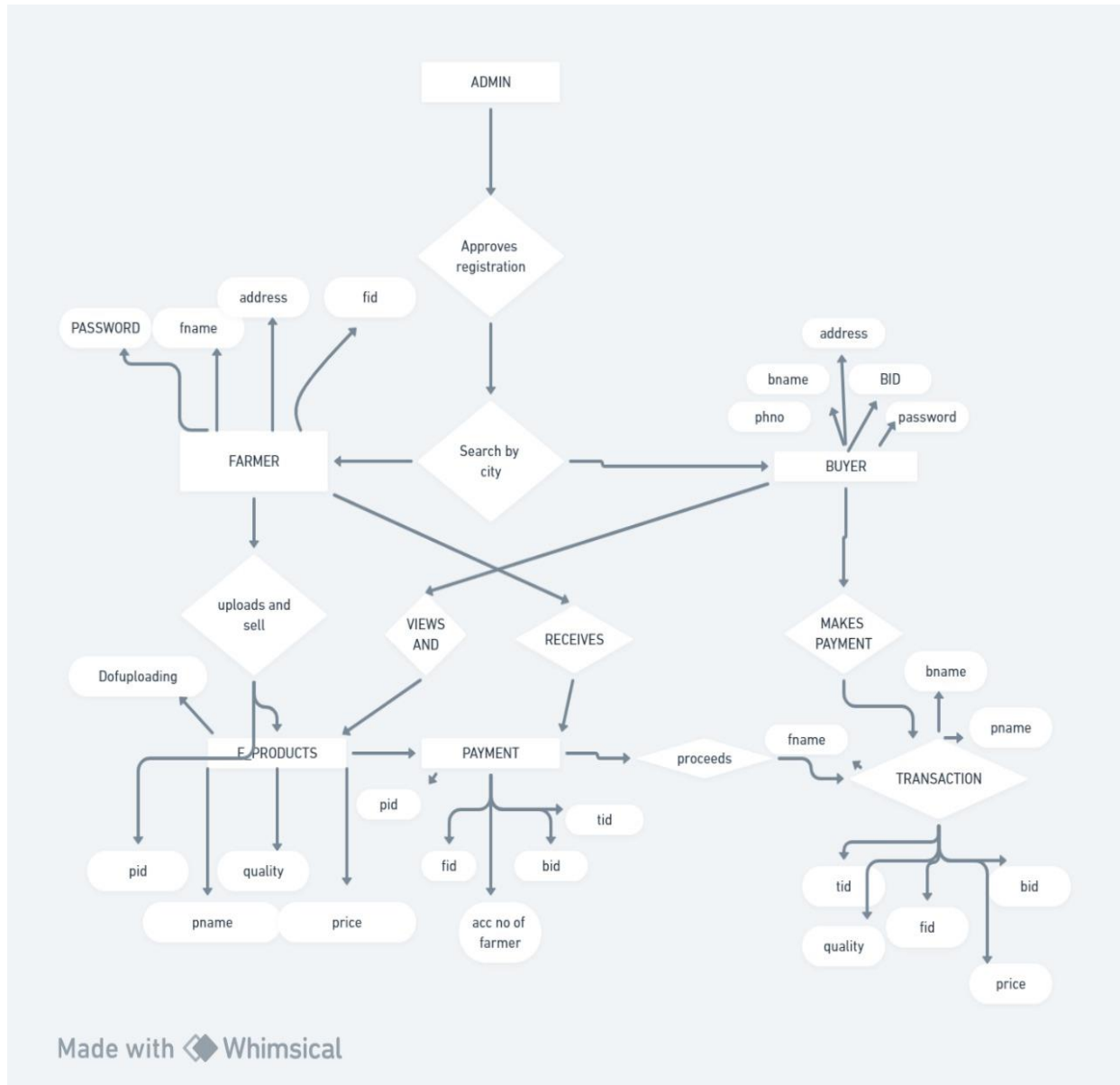
2. Relationships:

- A Farmer can sell multiple products (one-to-many relationship).
- Each Buyer can view and buy multiple products. (many-to-many relationship).
- Each Buyer have to make a transaction for each product purchased (one-to-many relationship).
- Each Transaction lead to one Payment (one-to-one relationship)
- Farmer receives payment whenever his products are sold. (one-to-many relationship)

3. Attributes:

- FARMER : FID, F_USERNAME, PASSWORD, F_NAME, F_PH_NO,etc.
- BUYER : BID, B_NAME, PASSWORD, B_USERNAME, B_PH_NO etc.
- TRANSACTION : TID, BID, PID, NAME, PRICE, FID, QUANTITY etc.
- PAYMENT : PID, TID,FID, ACC_OF_FARMER etc.
- F_PRODUCTS : PID, PNAME, PRICE, D_OF_UPLOAD etc.

4.2 ER Diagram



4.2 ER Diagram of HarvestHUB

PLATFORM USED FOR UML

Whimsical is a collaborative visual workspace platform that allows users to create diagrams, flowcharts, wireframes, and mind maps. It emphasizes simplicity and collaboration, enabling real-time editing and sharing of visual content. Whimsical is often used for project planning,

brainstorming, and design collaboration, providing a flexible and intuitive interface for teams and individuals to express ideas visually.

DESCRIPTION

- A Farmer can sell multiple products (one-to-many relationship).
- Each Buyer can view and buy multiple products. (many-to-many relationship).
- Each Buyer have to make a transaction for each product purchased (one-to-many relationship).
- Each Transaction lead to one Payment (one-to-one relationship)
- Farmer receives payment whenever his products are sold. (one-to-many relationship)

CHAPTER 5

CONCEPTUAL DESIGN

5.1 Create Commands

```
CREATE TABLE FARMER (  
    FID INT,  
    F_NAME VARCHAR(255),  
    F_PASSWORD VARCHAR(255) NOT NULL,  
    F_PH_NO VARCHAR(20) NOT NULL,  
    F_ADDRESS VARCHAR(255),  
    F_ACC_NO INT,  
    CONSTRAINT PK_FID PRIMARY KEY (FID),  
    CONSTRAINT PK_AC PRIMARY KEY (F_ACC_NO)  
);
```

```
CREATE TABLE BUYER (  
    BID INT,  
    B_NAME VARCHAR(255),  
    B_PASSWORD VARCHAR(255) NOT NULL,  
    B_PH_NO VARCHAR(20),  
    B_ADDRESS VARCHAR(255) ,  
    CONSTRAINT PK_BID PRIMARY KEY (BID)  
);
```

```
CREATE TABLE F_PRODUCT (  
    PID INT,  
    P_NAME VARCHAR(255),  
    FID INT,  
    PRICE DECIMAL(10, 2) NOT NULL,  
    P_CAT VARCHAR(50) NOT NULL,  
    CONSTRAINT PK_PID PRIMARY KEY (PID),
```

CONSTRAINT FOREIGN KEY (FID) REFERENCES FARMER(FID) ON DELETE CASCADE

);

CREATE TABLE MY_CART (

BID INT,

FID INT,

PRODUCT_ID INT,

QUANTITY INT,

TOTAL_PRICE DECIMAL(10, 2),

CONSTRAINT FK_BUYER FOREIGN KEY (BID) REFERENCES BUYER (BID)
ON DELETE CASCADE,

CONSTRAINT FK_FARMER FOREIGN KEY (FID) REFERENCES FARMER
(FID) ON DELETE CASCADE,

CONSTRAINT FK_PRODUCT FOREIGN KEY (PID) REFERENCES
F_PRODUCT (PID) ON DELETE CASCADE

);

CREATE TABLE PITRANSACTION (

TID INT,

BID INT,

FID INT,

PID INT,

TRANSACTION_DATE DATETIME,

AMOUNT DECIMAL(10, 2),

STATUS VARCHAR(20),

CONSTRAINT PK_TID PRIMARY KEY (TID),

CONSTRAINT FK_BUYER FOREIGN KEY (BUYER_ID) REFERENCES
BUYER(BID) ON DELETE CASCADE,

CONSTRAINT FK_FARMER FOREIGN KEY (FID) REFERENCES FARMER
(FID) ON DELETE CASCADE,

CONSTRAINT FK_PRODUCT FOREIGN KEY (PID) REFERENCES
F_PRODUCT (PID) ON DELETE CASCADE

);

```
CREATE TABLE PAYMENT (  
    PAY_ID INT ,  
    TID INT,  
    PAYMENT_DATE DATETIME,  
    AMOUNT DECIMAL(10, 2),  
    PAYMENT_METHOD VARCHAR(50),  
    STATUS VARCHAR(20),  
    CONSTRAINT FK_TRANSACTION FOREIGN KEY (TID) REFERENCES  
    TRANSACTION (TRANSACTION_ID) ON DELETE CASCADE  
);
```

5.2 Insert Commands and Data

```
INSERT INTO `blogdata`
```

```
(`blogId`, `blogUser`, `blogTitle`, `blogContent`, `blogTime`, `likes`)
```

```
VALUES (19, 'ThePhenom', 'First Blog', '<p>Its Awesome website</p>\r\n', '2018-02-25 13:09:41', 1);
```

```
INSERT INTO `blogfeedback`
```

```
(`blogId`, `comment`, `commentUser`, `commentPic`, `commentTime`)
```

```
VALUES (19, 'Mast yarr', 'ThePhenom', 'profile0.png', '2018-02-25 13:09:54');
```

```
INSERT INTO `farmer`
```

```
(`fid`, `fname`, `fusername`, `fpassword`, `fhash`, `femail`, `fmobile`, `faddress`, `factive`,  
`frating`, `picExt`, `picStatus`)
```

```
VALUES (3, 'Kaivalya Hemant Mendki', 'ThePhenom',  
'$2y$10$22ezmzHRa9c5ycHmVm5RpOnlT4LwFaDZar1XhmLRJQKGrcVRhPgti', '61b4a6b  
e663682e8cb037d9719ad8cd', 'kmendki98@gmail.com', '8600611198', 'abcde', 0, 0, 'png', 0);
```

```
INSERT INTO `fproduct`
(`fid`, `pid`, `product`, `pcat`, `pinfo`, `price`, `pimage`, `picStatus`)
VALUES(3, 27, 'Mango', 'Fruit', '<p>Mango raseela</p>\r\n', 500, 'Mango3.jpeg', 1),
(3, 28, 'Ladyfinger', 'Vegetable', '<p>Its veggie</p>\r\n', 1000, 'Ladyfinger3.jpg', 1),
(3, 29, 'Bajra', 'Grains', '<p>bajre di rti</p>\r\n', 400, 'Bajra3.jpg', 1),
(3, 30, 'Banana', 'Fruit', '<p>Jalgaon banana</p>\r\n', 400, 'Banana3.jpg', 1);
```

```
INSERT INTO `likedata` (`blogId`, `blogUserId`) VALUES
(19, 3);
```

```
INSERT INTO `mycart` (`bid`, `pid`) VALUES
(3, 27),
(3, 30);
```

```
INSERT INTO `transaction`
(`tid`, `bid`, `pid`, `name`, `city`, `mobile`, `email`, `pincode`, `addr`)
VALUES (1, 3, 28, 'sa,j,cns', 'sajc', 'sajch', 'kmendki98@gmail.com', 'sacu', 'ckaskjc');
```

5.3 SQL Queries

1. Retrieve the names of farmers along with the products they have listed for sale:

```
SELECT f.fname, fp.product
FROM farmer f
INNER JOIN fproduct fp ON f.fid = fp.fid;
```

Output :

fname	product
Kaivalya Hemant Mendki	Mango
Kaivalya Hemant Mendki	Ladyfinger
Kaivalya Hemant Mendki	Bajra
Kaivalya Hemant Mendki	Banana

2. Retrieve the blog titles and the number of likes each blog post has:

```
SELECT bd.blogTitle,
       ( SELECT COUNT(*)
         FROM likedata ld
         WHERE ld.blogId = bd.blogId) AS LikesCount
FROM blogdata bd;
```

Output :

blogTitle	LikesCount
First Blog	2

3. Update the price of all products listed by a specific farmer:

```
UPDATE fproduct fp
INNER JOIN farmer f ON fp.fid = f.fid
SET fp.price = fp.price * 1.1
WHERE f.fname = 'Kaivalya Hemant Mendki';
```

Output : 4 rows affected.

4. Retrieve the total number of products listed in each category:

```
SELECT pcat, COUNT(*) AS TotalProducts
FROM fproduct
GROUP BY pcat;
```

Output :

pcat	TotalProducts
Fruit	3
Grains	1
Vegetable	1

5. Retrieve the name of the buyer, product purchased, and transaction details:

```
SELECT b.bname, fp.product, t.city, t.mobile, t.email, t.addr
FROM buyer b
INNER JOIN transaction t ON b.bid = t.bid
INNER JOIN fproduct fp ON t.pid = fp.pid;
```

Output : empty

6. Retrieve blog posts with less than 10 likes:

```
SELECT *
FROM blogdata
WHERE blogId IN ( SELECT blogId
                  FROM likedata
                  GROUP BY blogId
                  HAVING COUNT(*) < 10);
```

Output :

blogId	blogUser	blogTitle	blogContent	blogTime	likes
19	ThePhenom	First Blog	<p>Its Awesome website<img alt="wink" src="https://..."	2018-02-25 18:39:41	2

7. TRIGGER :

```
CREATE TRIGGER update_farmer_rating AFTER INSERT ON fproduct
FOR EACH ROW
BEGIN
```

```
    DECLARE farmer_id INT;
```

```
    SELECT fid INTO farmer_id FROM fproduct WHERE pid = NEW.pid;
```

```
    UPDATE farmer SET frating = frating + 1 WHERE fid = farmer_id;
```

```
END;
```

Output:

AFTER INSERT ON fproduct specifies that the trigger should activate after a new row is inserted into the fproduct table.

FOR EACH ROW indicates that the trigger will execute once for each row affected by the triggering event.

DECLARE farmer_id INT; declares a variable to store the farmer ID.

The SELECT statement retrieves the fid (farmer ID) associated with the newly inserted product.

The UPDATE statement updates the frating (farmer rating) in the farmer table by incrementing it by 1 for the corresponding farmer ID.

8. PROCEDURE:

DELIMITER //

CREATE PROCEDURE CalculateAveragePrices()

BEGIN

DECLARE avgFruitPrice DECIMAL(10,2);

DECLARE avgVegetablePrice DECIMAL(10,2);

DECLARE avgGrainPrice DECIMAL(10,2);

SELECT AVG(price) INTO avgFruitPrice FROM fproduct WHERE pcat = 'Fruit';

SELECT AVG(price) INTO avgVegetablePrice FROM fproduct WHERE pcat =
'Vegetable';

SELECT AVG(price) INTO avgGrainPrice FROM fproduct WHERE pcat = 'Grains';

SELECT CONCAT('Fruits Avg. Price=', avgFruitPrice) AS Fruit_Avg_Price,

CONCAT('Vegetables Avg. Price=', avgVegetablePrice) AS Vegetable_Avg_Price,

CONCAT('Grains Avg. Price=', avgGrainPrice) AS Grain_Avg_Price;

END //

DELIMITER ;

Output :

The stored procedure you created, CalculateAveragePrices, is designed to calculate and display the average prices of fruits, vegetables, and grains in your product table. However, stored procedures in MySQL don't directly produce output like regular SQL queries. They perform actions within the database.

- Declaration of Variables: It initializes variables to hold the computed average prices.
- Calculation of Averages: Using SQL queries, it computes the average prices for each product category individually.
- Presentation of Results: The procedure organizes the calculated averages into a result.
- Invocation in PHP Code: Within your PHP script, you invoke this stored procedure to execute its logic.
- Retrieval and Display: Following execution, you retrieve the computed averages to exhibit them on site, providing users with valuable insights into product price trends.

CHAPTER 6

Frontend Design, Connectivity and Implementation

6.1 User Registration and Authentication

This module serves as the gateway for users, facilitating their access to the platform while ensuring a secure and seamless experience. Here's how it works:

6.1.1 User Registration

- **Registration Form:** Users are presented with a user-friendly registration form, accessible directly from the platform's homepage.
- **Data Collection:** The registration form collects essential user details such as name, username, contact information, email address, password, address, and category (whether they are farmers or buyers).
- **Validation:** Before submission, the form performs client-side validation to ensure all required fields are filled correctly, providing instant feedback to users.
- **Server-Side Processing:** Upon submission, the entered data is sent securely to the server for further processing.
- **Database Interaction:** PHP scripts handle the server-side processing, interacting with the database to store user information securely.
- **Registration Confirmation:** Users receive confirmation of successful registration, either through a confirmation message on the platform or via email.

6.1.2 User Authentication

- **Login Interface:** Authenticated access to the platform is granted through a straightforward login interface, accessible from the platform's homepage.
- **Credential Submission:** Upon entering their credentials (username, password, and category), users submit the login form for authentication.
- **Server-Side Authentication:** PHP scripts validate the submitted credentials against those stored securely in the database, ensuring accuracy and security.
- **Session Management:** Successful authentication initiates a session, granting users access to personalized features and content throughout their session.
- **Secure Access:** The platform employs industry-standard encryption techniques to safeguard user credentials during transmission and storage, prioritizing user privacy and data security.

6.1.3 Flexibility for Farmers and Buyers

- **Catering to Diverse Users:** The registration module accommodates both farmers and buyers, offering tailored experiences based on their specific needs and roles within the platform.
- **Category Selection:** During registration, users specify their category (farmer or buyer), enabling the platform to customize features and content accordingly.
- **Distinct User Experiences:** Farmers and buyers enjoy unique functionalities and access levels, optimized to streamline their interactions with the platform and enhance their overall user experience.

User Registration and Authentication serve as the foundation of the platform, ensuring that users can securely access and interact with its features. By providing a seamless registration process and robust authentication mechanisms, the platform establishes trust and confidence among its user base, fostering long-term engagement and satisfaction.

6.2 User Profile Management

This module empowers users to seamlessly interact with their profiles, facilitating personalized experiences on the platform. Users can effortlessly view and update their profile information, ensuring that their online presence accurately reflects their identity and preferences.

- **Profile Viewing:** Users can easily access their profile details, including their name, username, ratings, email, mobile number, and address. A profile image adds a visual touch, enhancing the user's connection to their online identity.
- **Profile Editing:** The module provides a convenient interface for users to modify their profile information. Form fields pre-filled with existing data streamline the editing process, enabling users to make quick updates to their personal details.
- **Image Upload:** Users have the flexibility to upload or remove profile images directly from the profile editing page. This feature allows users to personalize their profiles with images that resonate with their identity.
- **Data Validation:** The module incorporates robust validation mechanisms to ensure that user-provided information adheres to specified criteria. Mandatory fields are clearly marked, guiding users to complete essential profile updates accurately.
- **Responsive Design:** The user interface is designed with responsiveness in mind, ensuring optimal viewing and interaction across devices of various screen sizes. This

approach guarantees a consistent and user-friendly experience for all users, regardless of their device preferences.

- **Enhanced User Experience:** By offering a seamless profile management experience, the module enhances user satisfaction and engagement with the platform. Users can effortlessly maintain up-to-date profile information, fostering a sense of ownership and trust.
- **Personalization:** Empowering users to customize their profiles enables personalized interactions and content recommendations tailored to their preferences. This personalization fosters deeper connections between users and the platform, driving user loyalty and retention.
- **Streamlined Communication:** Accurate and comprehensive user profiles facilitate effective communication between users and other platform stakeholders. Whether connecting with fellow users, vendors, or support representatives, users can trust that their profile information accurately represents their identity and preferences.

6.3 Product Catalog and Search

In the Product Catalog and Search module of our project, users can search for products based on various criteria, thereby enhancing their ability to discover and purchase items of interest. This functionality enables users to:

- **Search for Products:** Users can search for products by entering specific keywords or selecting predefined categories such as fruits, vegetables, or grains. The search functionality allows for a more refined browsing experience, making it easier for users to find the products they are looking for.
- **Filter Products:** Users can filter products based on different attributes such as product category, price range, or availability. This allows users to narrow down their search results and find products that meet their specific requirements.
- **View Product Details:** Upon finding a product of interest, users can view detailed information about the product, including its name, description, price, and images. This helps users make informed decisions before making a purchase.
- **Navigate Product Catalog:** Users can browse through the product catalog to explore the available products. The catalog provides an organized listing of all products available

in the system, making it easy for users to browse through different categories and discover new items.

- **Add to Cart:** Users have the option to add products to their shopping cart directly from the product catalog. This allows users to quickly add items of interest to their cart and proceed to checkout for purchase.

Overall, the Product Catalog and Search module enhances the user experience by providing robust search and browsing functionalities, enabling users to easily find and purchase products of interest.

6.4 Shopping Cart Management

This module facilitates users to interact with their shopping carts, enabling them to perform various actions such as adding products, managing cart contents, and proceeding to checkout for purchase. The key functionalities include:

- **Adding Products to Cart:** Users can add products to their carts by clicking on the "Add to Cart" button or similar action triggers. This action initiates the process of storing the selected product in the user's cart.
- **Viewing Cart Contents:** Users can view the contents of their shopping cart, including details such as product names, prices, quantities, and subtotal amounts. This allows users to review the items they have added to their cart before proceeding with the checkout process.
- **Managing Cart Contents:** Users can manage the contents of their shopping cart by adjusting the quantity of items, removing items, or clearing the entire cart. These options provide users with flexibility and control over their shopping experience.
- **Proceeding to Checkout:** Once users have finalized their product selections, they can proceed to the checkout process to complete their purchase. This involves providing necessary billing and shipping information, selecting payment methods, and confirming the order.
- **Handling Cart Updates:** The system should dynamically update the cart contents in real-time as users make changes, ensuring that users have accurate and up-to-date information about their shopping carts.

6.5 Blog Creation and Management

This module empowers users to create and manage their own blogs, allowing them to share insights, experiences, and updates with a wider audience. Users can compose blog posts, upload multimedia content, and interact with readers through comments and feedback. The module provides a platform for users to express themselves creatively, build a community around shared interests, and establish their online presence as thought leaders or influencers.

- **Blog Viewing and Interaction:** The blog page enables users to explore a collection of blog posts authored by various contributors. Each blog post is displayed with its title, content, author's name, and timestamp of publication. Users can interact with the blog posts in several ways:
- **Like:** Users can express appreciation for a blog post by clicking the 'Like' button associated with each post. The number of likes for each post is dynamically updated and displayed alongside the button.
- **Commenting:** Users can engage in discussions by leaving comments on blog posts. A text area is provided below each post for users to input their comments. Upon submission, the comments are displayed below the post, showing the commenter's name, profile picture, comment text, and timestamp.
- **Interaction Buttons:** Each blog post has interaction buttons that allow users to perform actions such as liking the post and viewing or submitting comments. These buttons are designed to enhance user engagement and facilitate interaction with the content.
- **Blog Creation:** The blog page also provides a button to allow users to create new blog posts. Clicking this button redirects users to a blog creation form, where they can input the title and content of their new blog post.
- **Management Features:** Contributors have access to management features to maintain their blog posts. These features include editing and deleting existing posts, ensuring users have control over the content they publish.

Overall, the blog page serves as a dynamic platform for users to discover, interact with, and contribute to a community of bloggers, fostering a collaborative environment for sharing ideas, experiences, and insights."

6.6 Integration with phpMyAdmin and Database Connectivity

This section provides insights into how frontend applications are integrated with phpMyAdmin for efficient database management and administration. Additionally, it discusses the process of establishing database connectivity using PHP scripts, encompassing CRUD operations and error handling.

6.6.1 Integration with phpMyAdmin

phpMyAdmin serves as a robust and user-friendly tool for managing MySQL databases. In the HarvetHub project, phpMyAdmin is seamlessly integrated to facilitate database creation, schema design, data manipulation, and administration tasks. Key aspects of this integration include:

- **Database Creation:** Through phpMyAdmin, databases are created to store essential project data, such as user information, product details, and blog content.
- **Schema Design:** Tables, columns, and relationships are defined using phpMyAdmin's intuitive interface, ensuring a well-structured database schema that aligns with project requirements.
- **Data Manipulation:** phpMyAdmin enables effortless insertion, updating, deletion, and querying of data within the database tables. This functionality empowers administrators to manage project data efficiently.
- **User Management:** User accounts and privileges can be managed through phpMyAdmin, ensuring secure access control and authentication mechanisms.

6.6.2 Database Connectivity with PHP

PHP scripts are utilized to establish seamless connectivity with the MySQL database, enabling frontend applications to interact with backend data effectively. The process involves the following steps:

- **Connection Establishment:** PHP scripts establish a connection to the MySQL database server using appropriate credentials, such as hostname, username, password, and database name.
- **CRUD Operations:** Once the connection is established, PHP scripts execute CRUD (Create, Read, Update, Delete) operations to manipulate data stored in the database.

This includes inserting new records, retrieving existing data, updating records, and deleting entries as needed.

- **Error Handling:** Robust error handling mechanisms are implemented within PHP scripts to ensure graceful handling of database-related errors. This includes detecting connection failures, query errors, and data validation issues, and providing informative error messages to users.
- **Data Sanitization and Validation:** Before executing SQL queries, PHP scripts sanitize and validate user input to prevent SQL injection attacks and ensure data integrity. This involves using prepared statements, parameterized queries, and input validation techniques.

By integrating frontend applications with phpMyAdmin for efficient database management and leveraging PHP scripts for establishing database connectivity and executing CRUD operations, the HarvetHub project maintains a robust and scalable backend infrastructure.

CHAPTER 7

Frontend Source Code

7.1 User Registration and Authentication Code

php

Copy code

```
<?php
```

```
// Start session and include necessary files
```

```
// Check if form is submitted
```

```
if($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    // Retrieve form data
```

```
    $username = $_POST['uname'];
```

```
    $password = $_POST['pass'];
```

```
    $category = $_POST['category'];
```

```
    // Call function/method to authenticate user
```

```
    if (authenticateUser($username, $password, $category)) {
```

```
        // User authentication successful
```

```
        // Set session variables and redirect
```

```
    } else {
```

```
        // Authentication failed
```

```
        // Redirect to error page or display message
```

```
    }
```

```
}
```

```
?>
```

2. Registration (Login/signUp.php):

php

Copy code

```
<?php

// Start session and include necessary files

// Check if form is submitted

if($_SERVER["REQUEST_METHOD"] == "POST") {

    // Retrieve form data and sanitize

    // Call function/method to register user

    if (registerUser($name, $username, $mobile, $email, $password, $address, $category)) {

        // Registration successful

        // Redirect to login page or display success message

    } else {

        // Registration failed

        // Redirect to error page or display message

    }

}

?>
```

3. Modal Display JavaScript:

javascript

Copy code

```
<script>

// Get the login modal

var loginModal = document.getElementById('id01');
```

```
// Get the registration modal

var registrationModal = document.getElementById('id02');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {

    if (event.target == loginModal || event.target == registrationModal) {

        loginModal.style.display = "none";

        registrationModal.style.display = "none";

    }

}

</script>
```

7.2 User Profile Management Code

// profile view code

```
<header>

    <span          class="image"          left"></span>

    <br>

    <h2><?php echo $_SESSION['Name'];?></h2>

    <h4><?php echo $_SESSION['Username'];?></h4>

</header>

<div class="row">

    <div class="col-sm-3"></div>

    <div class="col-sm-3">

        <b><font size="+1" color="black">RATINGS : </font></b>
```

```

        <font size="+1"><?php echo $_SESSION['Rating'];?></font>

    </div>

    <div class="col-sm-3">

        <b><font size="+1" color="black">Email ID : </font></b>

        <font size="+1"><?php echo $_SESSION['Email'];?></font>

    </div>

</div>

<br />

<div class="row">

    <div class="col-sm-3"></div>

    <div class="col-sm-3">

        <b><font size="+1" color="black">Mobile No : </font></b>

        <font size="+1"><?php echo $_SESSION['Mobile'];?></font>

    </div>

    <div class="col-sm-3">

        <b><font size="+1" color="black">ADDRESS : </font></b>

        <font size="+1"><?php echo $_SESSION['Addr'];?></font>

    </div>

</div>

// profile edit code

<header>

    <span          class="image          left"></span>

    <br>

```

```
<h2><?php echo $_SESSION['Name'];?></h2>

<h4><?php echo $_SESSION['Username'];?></h4>

</header>

<form method="post" action="Profile/updateProfile.php">

    <div class="row uniform">

        <div class="8u 12u$(xsmall)">

            <input type="text" name="name" id="name" value="<?php echo
$_SESSION['Name'];?>" placeholder="Full Name" required />

        </div>

        <!-- More form fields for other profile data -->

    </div>

    <div class="12u$">

        <center>

            <input type="submit" class="button special" value="Update Profile" />

        </center>

    </div>

</form>
```

7.3 Product Catalog and Search code

```
<?php

    session_start();

    require 'db.php';

?>

<!DOCTYPE html>

<html lang="en">
```

```
<head>

    <!-- Meta tags and CSS/JS links -->

</head>

<body>

<?php require 'menu.php'; ?>

<section id="main" class="wrapper style1 align-center">

    <div class="container">

        <h2>Welcome to the digital market</h2>

        <?php if(isset($_GET['n']) AND $_GET['n'] == 1): ?>

            <!-- Filter selection form -->

            <h3>Select Filter</h3>

            <form method="GET" action="productMenu.php?">

                <input type="text" value="1" name="n" style="display: none;"/>

                <center>

                    <div class="row">

                        <!-- Filter selection dropdown -->

                        <div class="col-sm-2">

                            <div class="select-wrapper" style="width: auto">

                                <select name="type" id="type" required style="background-
                                color:white;color: black;">

                                    <option value="all" style="color: black;">List All</option>

                                    <option value="fruit" style="color: black;">Fruit</option>

                                    <option value="vegetable" style="color: black;">Vegetable</option>

                                    <option value="grain" style="color: black;">Grains</option>

                                </select>
```

```
</div>

</div>

<!-- Submit button -->

<div class="col-sm-2">

    <input class="button special" type="submit" value="Go!" />

</div>

</div>

</center>

</form>

<?php endif; ?>

<!-- Displaying products based on filter -->

<section id="two" class="wrapper style2 align-center">

    <div class="container">

        <?php

            // Retrieving Products Query:

            if(!isset($_GET['type']) OR $_GET['type'] == "all")

            {

                $sql = "SELECT * FROM fproduct WHERE 1";

            }

            if(isset($_GET['type']) AND $_GET['type'] == "fruit")

            {

                $sql = "SELECT * FROM fproduct WHERE pcat

                    = 'Fruit'";

            }

        }

    }

}
```

```
if(isset($_GET['type'])AND$_GET['type']=="vegetable")
{
    $sql = "SELECT * FROM fproduct WHERE pcat
    = 'Vegetable'";
}

if(isset($_GET['type']) AND $_GET['type'] == "grain")
{
    $sql = "SELECT * FROM fproduct WHERE pcat
    = 'Grains'";
}

$result = mysqli_query($conn, $sql);

?>

<div class="row">

<?php
while($row = $result->fetch_array()):

    $picDestination = "images/productimages/".$row['pimage'];

?>

<div class="col-md-4">

<section>

    <strong><h2    class="title"    style="color:black;"><?php    echo
    $row['product'];?></h2></strong>

    <a href="review.php?pid=<?php echo $row['pid']; ?>"></a>

    <div style="align: left">
```



```
<blockquote><?php echo "Type: ".$row['pcat']; ?><br><?php echo "Price:
".$row['price']." /-"; ?><br></blockquote>

</div>

</section>

</div>

<?php endwhile; ?>

</div>

</div>

</section>

</div>

</section>

</body>

</html>
```

7.4 Shopping Cart Management Code

```
<?php

    session_start();

    require 'db.php';

    // Essential Functionality: Check if user is logged in

    if(!isset($_SESSION['logged_in']) OR $_SESSION['logged_in'] == 0)

    {

        $_SESSION['message'] = "You need to first login to access this page !!!";

        header("Location: Login/error.php");

    }
```

```
// Get buyer ID from session

$bid = $_SESSION['id'];

// Essential Functionality: Add product to cart if flag is set

if(isset($_GET['flag']))

{

    $pid = $_GET['pid'];

    // Insert product into cart

    $sql = "INSERT INTO mycart (bid,pid)

        VALUES ('$bid', '$pid)";

    $result = mysqli_query($conn, $sql);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <!-- Meta tags and CSS/JS links -->

</head>

<body>

<?php require 'menu.php'; ?>

<section id="main" class="wrapper style1 align-center">

    <div class="container">

        <h2>My Cart</h2>

        <div class="row">

            <?php
```

```

$sql = "SELECT * FROM mycart WHERE bid = '$bid'";

$result = mysqli_query($conn, $sql);

while($row = $result->fetch_array()):

    $pid = $row['pid'];

    $sql = "SELECT * FROM fproduct WHERE pid = '$pid'";

    $result1 = mysqli_query($conn, $sql);

    $row1 = $result1->fetch_array();

    $picDestination = "images/productImages/".$row1['pimage'];

?>

<div class="col-md-4">

    <section>

        <strong><h2          class="title"          style="color:black;"><?php          echo
        $row1['product'];?></h2></strong>

        <a href="review.php?pid=<?php echo $row1['pid']; ?>"></a>

        <div style="align: left">

            <blockquote><?php echo "Type: ".$row1['pcat']; ?><br><?php echo "Price:
            ".$row1['price']." /-"; ?><br></blockquote>

            <!-- Remove item from cart functionality -->

            <form action="removeItem.php" method="post">

                <input type="hidden" name="pid" value="<?php echo $pid; ?>">

                <input type="submit" name="remove" value="Remove from Cart">

            </form>

        </div>

    </section>

```

```
</div>

<?php endwhile; ?>

</div>

</div>

</section>

</body>

</html>
```

7.5 Blog Creation and Management code

```
<?php

    session_start();

    require 'db.php';

    // Essential Functionality: Check if user is logged in

    if(!isset($_SESSION['logged_in']) OR $_SESSION['logged_in'] == 0)

    {

        $_SESSION['message'] = "You need to first login to access this page !!!";

        header("Location: Login/error.php");

    }

    if($_SERVER["REQUEST_METHOD"]=="POST"AND

    isset($_SESSION['logged_in']) AND $_SESSION['logged_in'] == 1)

    {

        if(isset($_POST['comment']) AND $_POST['comment'] != "")

        {

            $sql = "SELECT * FROM blogdata ORDER BY blogId DESC";
```

```
$result = mysqli_query($conn, $sql);

while($row = $result->fetch_array())
{
    $check = "submit".$row['blogId'];

    if(isset($_POST[$check]))
    {
        $blogId = $row['blogId'];

        break;
    }
}

$comment = dataFilter($_POST['comment']);

if(isset($_SESSION['logged_in']) AND $_SESSION['logged_in'] == 1)
{
    $commentUser = $_SESSION['Username'];

    $pic = $_SESSION['picName'];
}
else {
    $commentUser = "Anonymous";

    $pic = "profile0.png";
}

if(isset($blogId))
{
    // Inserting Comments:

    $sql = "INSERT INTO blogfeedback (blogId, comment,
    commentUser, commentPic)
```

```
VALUES ('$blogId' , '$comment', '$commentUser', '$pic');"

$result = mysqli_query($conn, $sql);

    }

}

else

{

    // Fetching Blog Data:

    $sql = "SELECT * FROM blogdata ORDER BY blogId DESC";

    $result = mysqli_query($conn, $sql);

    while($row = $result->fetch_array())

    {

        $check = "like".$row['blogId'];

        if(isset($_POST[$check]))

        {

            $blogId = $row['blogId'];

            break;

        }

    }

    $likeCheck = "isLiked".$blogId;

    if(!isset($_SESSION[$likeCheck]) OR $_SESSION[$likeCheck] == 0)

    {

        $id = $_SESSION['id'];

        $sql = "SELECT * FROM likedata WHERE blogId = '$blogId'

        AND blogUserId = '$id'";
```

```
$result = mysqli_query($conn, $sql);

$num_rows = mysqli_num_rows($result);

if($num_rows == 0)
{
    // Inserting Likes:

    $sql = "INSERT INTO likedata (blogId, blogUserId)

            VALUES('$blogId', '$id)";

    $result = mysqli_query($conn, $sql);

    // Updating Like Count:

    $sql = "UPDATE blogdata SET likes = likes + 1 WHERE
    blogId = '$blogId'";

    $result = mysqli_query($conn, $sql);

    $_SESSION[$likeCheck] = 1;
}

}

}

}

// Function to filter data

function dataFilter($data) {

    $data = trim($data);

    $data = stripslashes($data);

    $data = htmlspecialchars($data);

    return $data;

}
```

```
// Fetching blog data from database

$sql = "SELECT * FROM blogdata ORDER BY blogId DESC";

$result = mysqli_query($conn, $sql);

?>

<!DOCTYPE HTML>

<html>

    <head>

        <!-- Head section content -->

    </head>

    <body class="subpage">

        <?php require 'menu.php'; ?>

        <section id="main" class="wrapper">

            <div class="inner">

                <div class="container" style="width: 70%">

                    <?php while ($row = $result->fetch_array()) : ?>

                        <div class="box">

                            <!-- Displaying blog content -->

                            <h2><?= $row['blogTitle']; ?></h2>

                            <blockquote>

                                <?= $row['blogContent']; ?>

                                <p>--- <?= $row['blogUser']; ?></p>

                                <p><?= $row['blogTime']; ?></p>

                            </blockquote>

                        </div>

                    </?php ?>

                </div>

            </div>

        </section>

    </body>

</html>
```



```
<!-- Form for comment submission -->

<form method="post" action="blogView.php">

    <!-- Comment input and submission button -->

</form>

<!-- Displaying comments -->

<?php

    $id = $row['blogId'];

    $sql = "SELECT * FROM blogfeedback WHERE blogId = '$id'";

    $result1 = mysqli_query($conn, $sql);

    if ($result1) :

        while ($row1 = $result1->fetch_array()) :

            ?>

            <div class="con darker">

                <!-- Displaying comment content -->

            </div>

            <?php endwhile; ?>

            <?php endif; ?>

        </div>

    <?php endwhile; ?>

</div>

</div></section>

<!-- Scripts --></body>
```

CHAPTER 8

Results, Discussion and Screenshots

8.1 Results and Discussion

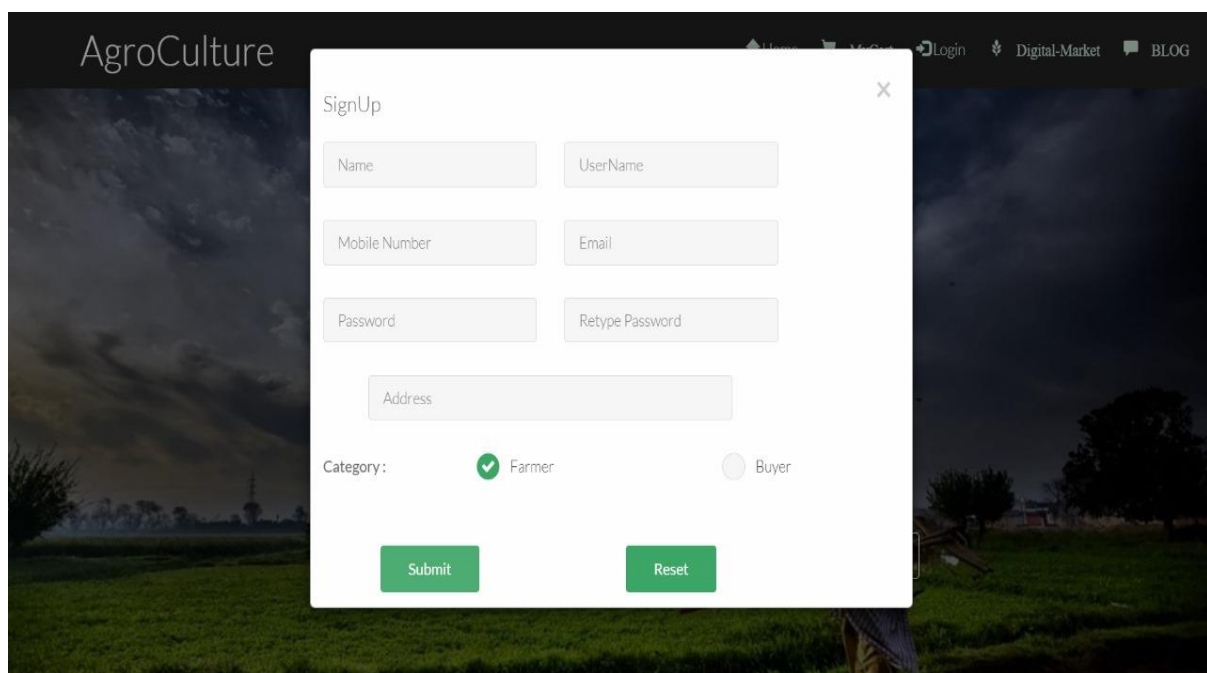
The implementation of the HarvestHub system has achieved significant milestones in bridging the gap between farmers and buyers, facilitating direct transactions and promoting quality food production. Below are the results and discussions based on the core objectives and future enhancements of the project:

- **Facilitation of Farmer-to-Buyer Transactions:** The system successfully enables farmers to register and list their products for sale, providing a platform for direct interaction with buyers. Buyers can browse through the listed products, place orders, and make purchases directly from the farmers, fostering transparency and trust in the agricultural supply chain.
- **Promotion of Quality Food Production:** By empowering farmers to showcase and sell their products directly to consumers, the HarvestHub system incentivizes the production of quality goods. Buyers have access to fresh, locally sourced produce, thereby promoting sustainable farming practices and supporting local agricultural communities.
- **Enhanced Features for Future Development:** Future iterations of the web platform can introduce advanced features such as pre-ordering and scheduling of product deliveries, enhancing user convenience and satisfaction. Implementing interactive features that facilitate direct communication between farmers and buyers can further strengthen the connection and trust between the two parties.
- **User Feedback and Iterative Improvement:** Continuous user feedback and iterative improvements are essential for enhancing the functionality and usability of the HarvestHub system. By soliciting feedback from farmers and buyers, the system can evolve to meet the evolving needs and preferences of its users, ensuring its relevance and effectiveness in the long term.
- **Potential for Expansion and Scalability:** The success of the HarvestHub system lays the foundation for potential expansion into new markets and regions, catering to a broader audience of farmers and buyers. Scalability considerations, such as optimizing system performance and accommodating increased user traffic, are crucial for supporting future growth and sustainability.

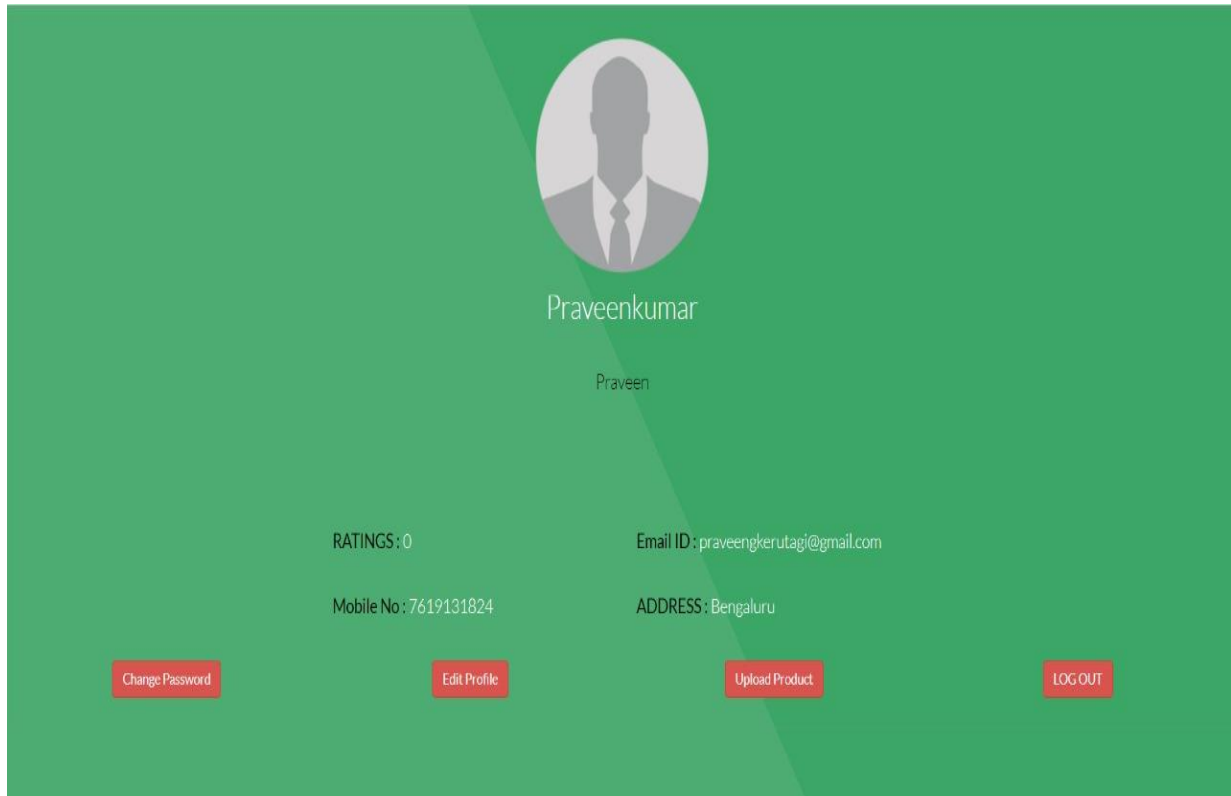
8.2 Screenshots



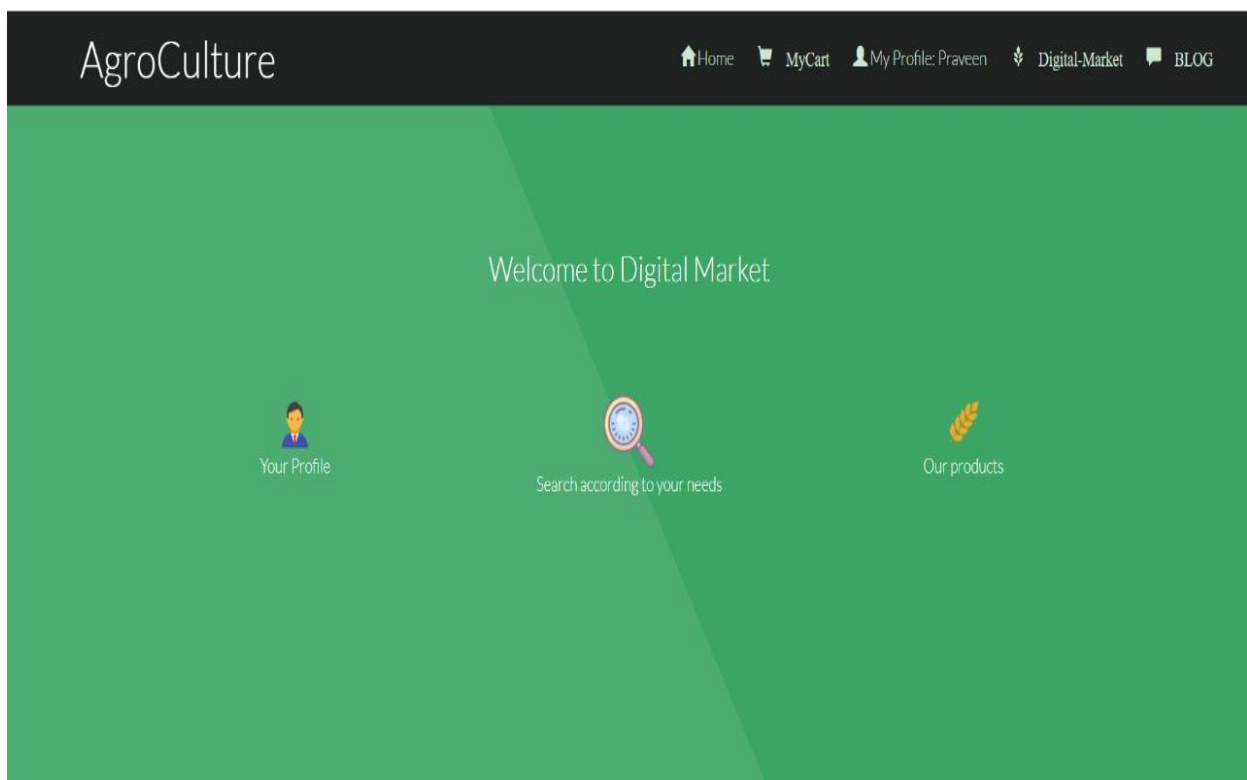
8.2.1 Home Page



8.2.2 Farmer/Buyer registration Page



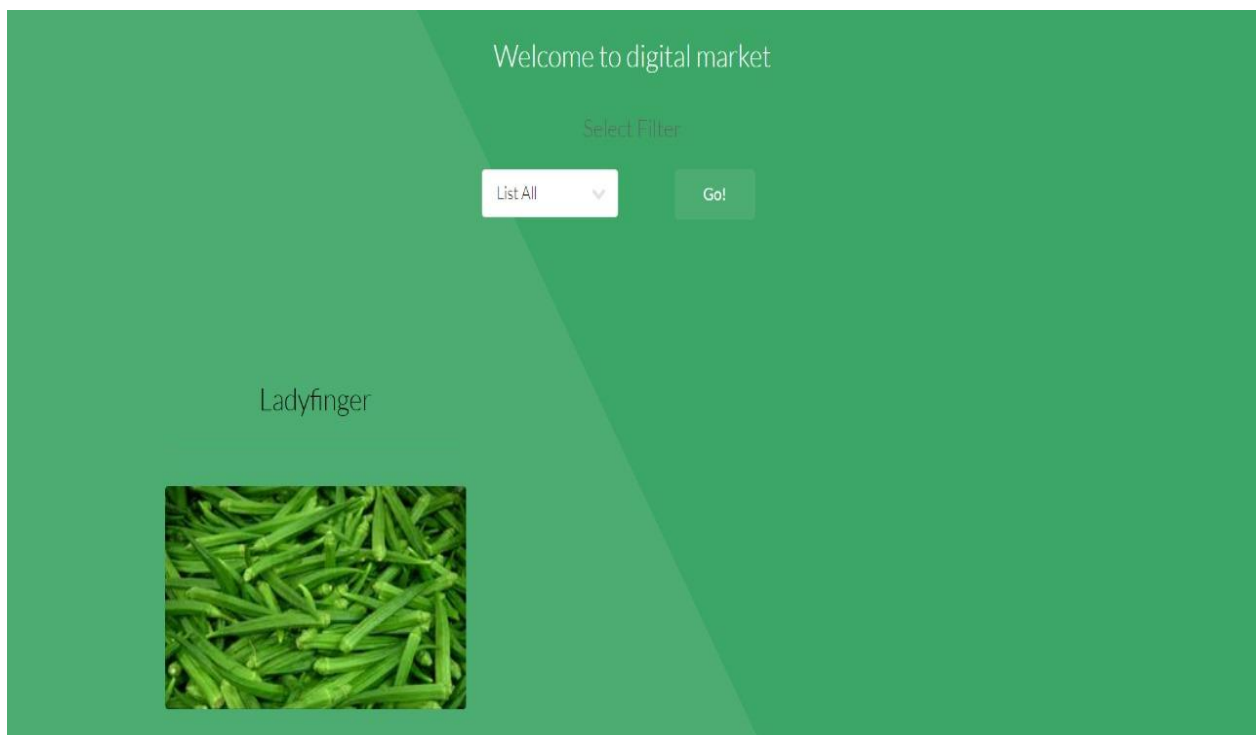
8.2.3 Farmer/Buyer Profile Page



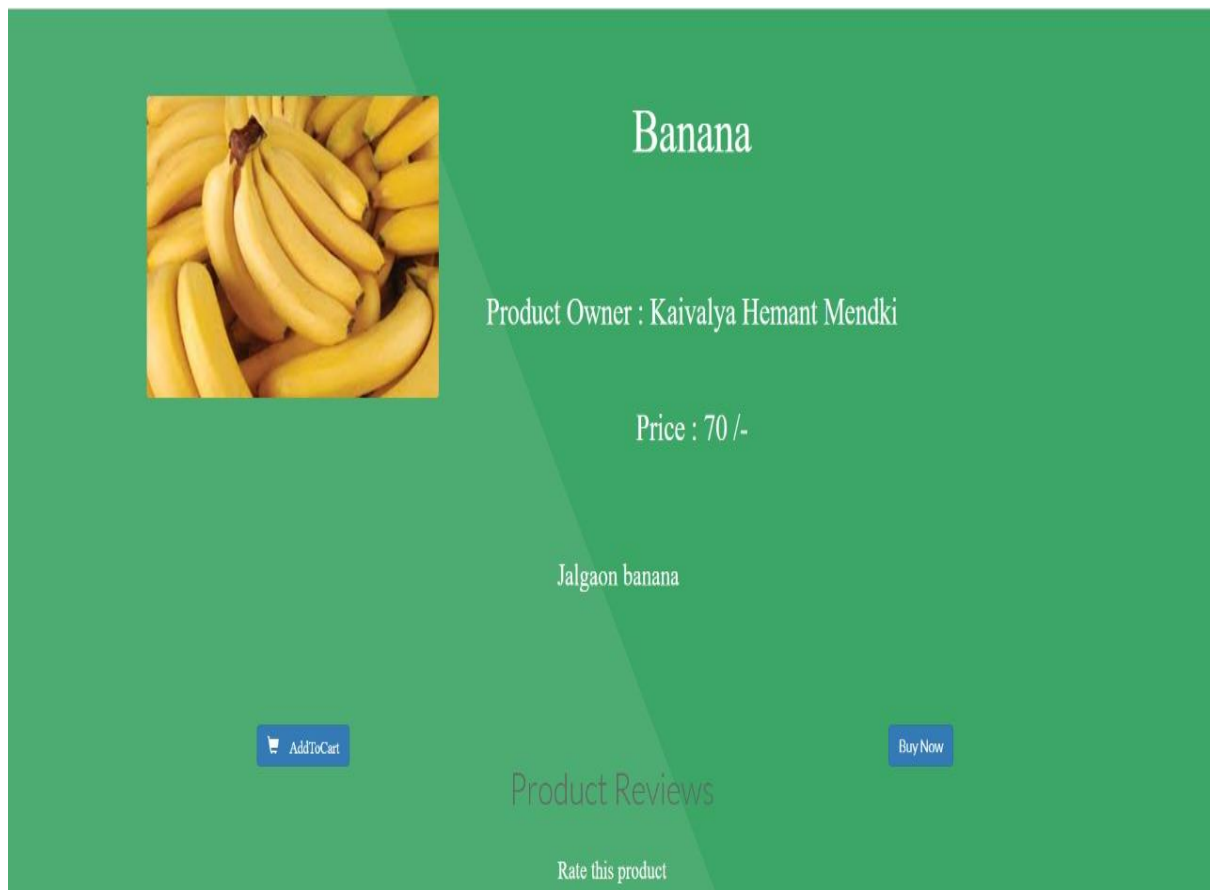
8.2.4 Product Search Page



8.2.5 MyCart Page



8.2.6 Product Purchase Page



8.2.7 Product View Page

Transaction Details

<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="button" value="Confirm Order"/>	

8.2.8 Transaction details Page

Enter the Product Information here...!!

Choose File No file chosen

- Category -

Product Name

Source

Styles • Format • Font • Size • A A ?

Price

Submit

8.2.9 Add Product Page

Write a Blog

Makar Sankranti

"Wishing all our HarvestHub users a Makar Sankranti filled with warmth, prosperity, and the sweetness of til gul. May this auspicious festival bring you abundant harvests of joy, success, and togetherness with loved ones. Happy Makar Sankranti from the HarvestHub family!"

--- Praveen

2024-02-20 00:12:30

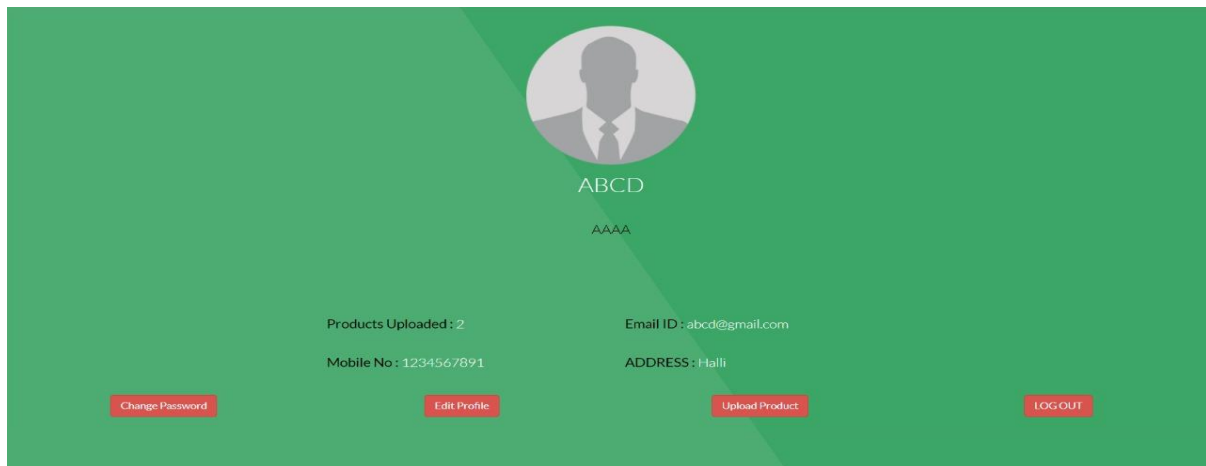
Like 0

Comments: 0

Write a Comment!

Submit

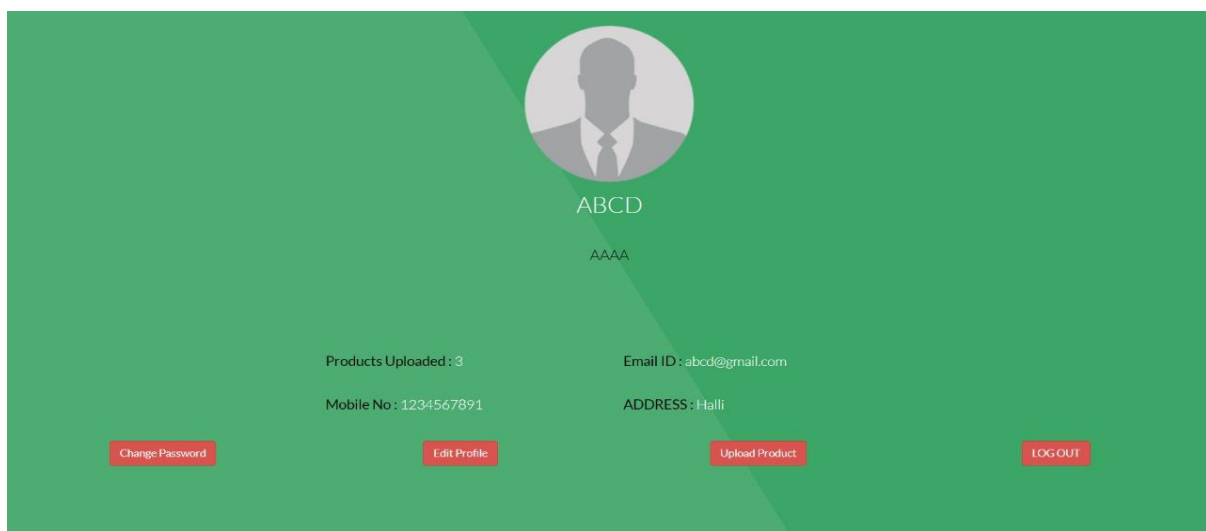
8.2.10 Blog Page



8.2.10 Profile Page before Trigger

A screenshot of a product upload form with a green background. At the top, it says 'Enter the Product Information here...!!'. Below this is a 'Choose File' button next to the filename 'Bajra3.jpg'. There is a dropdown menu labeled 'Grains' and a text input field containing 'Bajra'. Below these is a rich text editor with a toolbar and a large text area. At the bottom left is a text input field containing '140', and at the bottom right is a green 'Submit' button.

8.2.11 Product upload Trigger



8.2.12 Profile Page after Trigger



HarvetHub

Product Summary Report

Generated by: Praveenkumar

Report generated on February 22, 2024

This report provides a summary of product transactions categorized by product category. It aims to provide insights into the sales performance of various product categories.

Fruit

Product Name	Price	Purchases
Apple	100	1
Banana	70	1
Mango	50	0

Grains

Product Name	Price	Purchases
Bajra	150	0
rice	100	1
wheat	250	0

Vegetable

Product Name	Price	Purchases
Brinjal	40	1
Ladyfinger	35	3
potato	50	0

This report adheres to Confidential - For internal use only.

8.2.13 Product purchase summary report



HarvetHub

Blog Post Report

Generated by: Praveenkumar

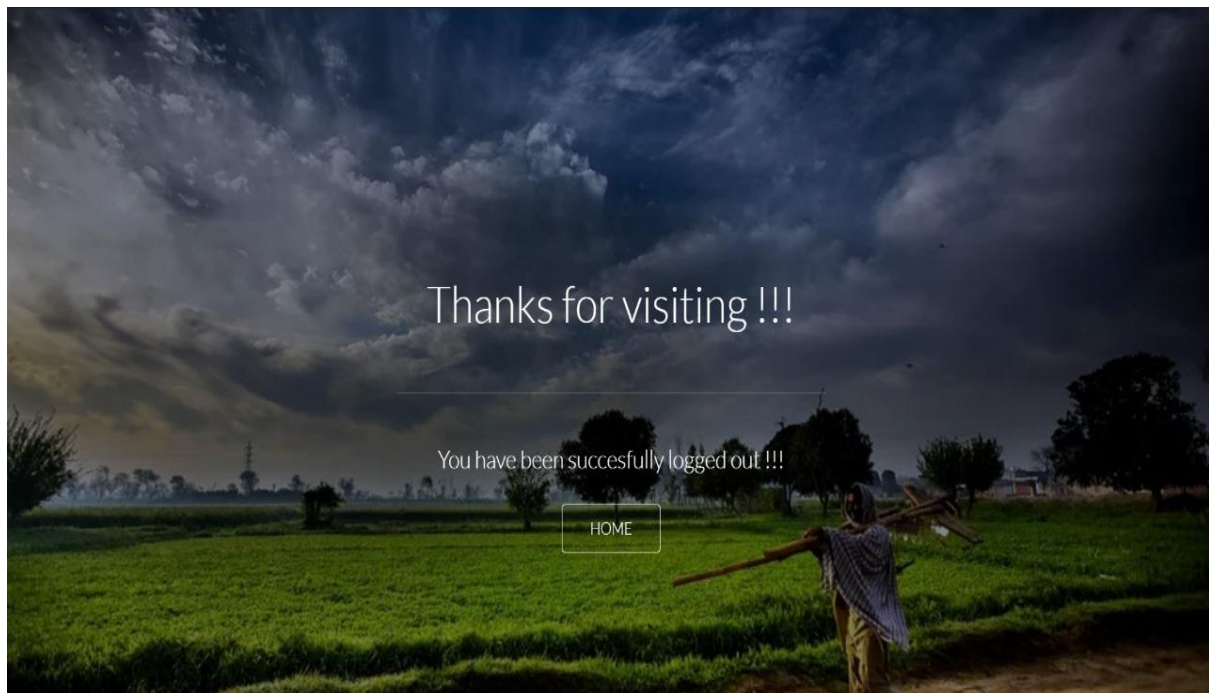
Report generated on February 22, 2024

This report analyzes blog post performance based on likes, offering insights to optimize content strategy and enhance user engagement. It provides stakeholders with actionable intelligence to drive organizational growth and maximize the impact of our blogging platform.

Title	Author	Post Date	Likes
Welcome to HarvetHub	Praveenkumar	2018-02-25 18:39:41	3
Makar Sankranti	Praveen	2024-02-20 00:12:30	2
Does Organic Farming have future?	AAAA	2024-02-21 22:53:21	1

This report adheres to Confidential - For internal use only.

8.2.14 Blog review report



8.2.15 Log out Page

CHAPTER 9

Conclusion and Scope

9.1 Conclusion

The HarvestHub system represents a significant step forward in revolutionizing the agricultural industry by bridging the gap between farmers and buyers and promoting quality food production. Through its innovative features and user-friendly interface, the system has successfully facilitated direct transactions, empowered farmers, and provided consumers with access to fresh, locally sourced produce. The project has achieved its primary objectives of providing qualitative foods to buyers, implementing an automated/online agriculture system, inspiring farmers to produce quality goods, and promoting eco-friendly farming practices. The system's success underscores the importance of leveraging technology to address pressing challenges in the agricultural sector and create a more transparent and sustainable food supply chain.

9.2 Scope

While the current implementation of the HarvestHub system has delivered significant benefits, there is ample room for further enhancement and expansion. Future iterations of the system can explore additional features such as advanced booking of products, real-time communication between farmers and buyers, and integration with agricultural analytics tools for data-driven decision-making. Moreover, the scope extends to scalability considerations, including optimizing system performance, accommodating increased user traffic, and expanding into new markets and regions. By continuously evolving and adapting to the evolving needs of its users, the HarvestHub system can continue to drive positive change in the agricultural industry and contribute to a more sustainable future.

References and Weblinks

- www.w3schools.com
- www.tutorialspoint.com
- <https://www.w3schools.com/php/default.asp>
- <https://www.sitepoint.com/php/>
- <https://www.php.net/>
- <https://www.javatpoint.com/php.in/>
- <https://www.mysql.com/>
- <http://www.mysqltutorial.org>
- <http://www.javatpoint.com/mysql.org>
- <https://www.apachefriends.org/download.html>