**1. INTRODUCTION**

**1.1 OVERVIEW OF THE PROJECT**

Website for E-Agriculture for Direct Marketing of Food Crops using chatbots is a portal to help Buyers to purchase the farmer products. They provide selling agricultural products and giving solution to the farmers. They organize the dealing system team to give solution to the farmers and salting product name, description, quantity and amount. The business of marketing agriculture products to consumers. The trends of the crop act so that will be pretty important to the Buyers who access these via the help of Internet.

The main features of the information system includes information retrieval facilities for Buyers from anywhere in the form of obtaining statistical information about product name, description, quantity and farmer basic amount to know for the Buyer can bid the farmers products. This application will reduce manual work and maintain updates in database from time to time. Get item information and price details. Farmer to sell our own product in this web application. The farmer used the traditional knowledge for agricultural production and animal husbandry because agricultural education and research activity was very slow. To discuss the trends and challenges of Indian agricultural sector. In this project is overall maintaining the farmer and Buyer details. Farmers are adding the different products in this web application it used unique Buyer name and password. Buyers are view the products for bid and buy the item in separate login. Admin maintain the farmer registration, Buyer registration, and product information and bidding details.

**1.2 MODULE DESCRIPTION**

**Admin login**

Admin module logon to the login page. Admin will maintain the farmer registration & login, buyer registration & login and view the products.

**Farmer registration**

Farmer login with valid Buyername and password provided by registration. It contains fid, Buyername, email, address, contact and password.

**Farmer login**

Farmer login with valid Buyername and password. Farmer can add their new farm products. Farmer details are maintained to the fid, Buyername, email, address, contact and password.

**Add product**

Farmer can add their new farm product details such as product name, farmer name, farmer contact, farmer location, near km, product description, product type, image and price etc.

**Remove product**

Farmer can remove the product if it is any issue in the given information or something went wrong such as product name to remove the product.

**View bidding**

Farmer can view the bidding details such as product id, product name, buyer email, original price, bidding price and accept etc.

**Buyer registration**

In this module, buyer details are includes buyer bid, Buyername, email, address, contact and password.

**Buyer login**

Buyer login with valid Buyername and password. Buyer can view product, bid and bidding status details. Buyer details are maintained to the staff name, designation, email, password and mobile no etc.

**View product**

Buyer can view the farmer added product details such as pid, product name, farmer name, farmer contact, farmer location & km, description, type of product, image and price.

**Add bid**

In this module buyer can bid the product what they want with the using pid, product name, image, farmer name, farmer email, location, price, bidding price and bid.

**View bidding status**

Buyer can view the farmer accepted bidded product details such as pid, product name, farmer email, farmer contact, original price, bidding price and status.

**1.3 SYSTEM SPECIFICATION**

**1.3.1 HARDWARE SPECIFICATION**

| Processor | : | Intel core2dual |
| --- | --- | --- |
| Memory | : | 4 GB RAM or More |
| Hard disk Requirement | : | Free 500GB on installation drive |

**1.3.2 SOFTWARE SPECIFICATION**

| Operating System | : | Windows10 |
| --- | --- | --- |
| Scripting Language | : | PHP |
| Database | : | MYSQL |

**SOFTWARE FEATURES**

**About PHP**

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP widely used; free and efficient alternative to competitors such as Microsoft’s ASP.PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is similar to pearl and C.

PHP is open source that it is readily available and absolutely free. Stability, flexibility and speed are chief qualities that attract to choose PHP.PHP have multiple extensions and is extremely scalable.

**Server-side scripting**

This server-side scripting is the most traditional and main target field for PHP. Programmer needs three things to make this work. Programmer need to run the web server, with a connected PHP installation. Programmer can access the PHP program output with a web browser, viewing the PHO page through the server. All these can run on your home machine if programmers are just experimenting with PHP programming.

**Command line scripting**

Programmer can make a PHP script to run it without any server or browser. Programmers only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using croon (on\*nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.

**Features of PHP**

* PHP runs on different platforms (Windows, Linux, UNIX, etc.)
* PHP is compatible with almost all servers used today.
* PHP is free to download from the official PHP resource: www.php.net.

**About MYSQL**

MYSQL is an open-source relational database management system (RDBMS) is developed, distributed and supported by MYSQL AB. MYSQL is a popular choice of database for use in web applications MYSQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory. MYSQL is easy to use, yet extremely powerful, secure, and scalable. And because of its small size and speed, it is the ideal database solution for Web sites.

**MYSQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amount of information in a corporation network. To add, access and process data stored in a computer database we need a database management system such as MYSQL server. Since computers are very good at handling large amount of data, database management system plays a central role in computing.

**MYSQL is a relational database management system**

A relational database stores separate data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MYSQL” stands for “Structured Query Language”. SQL is the most common standardize language used to access database and is defined by the ANSI/ISO SQL standard. The SQL standard has been evolving since 1986 and several versions exist.

**MYSQL software is open source**

Open source means that it is possible for anyone to use modify the software. Anybody can download the MYSQL software uses the GPL (GNU General Public License), to define what we may and may not use do with the software.

**MYSQL Server works in Client/ Server or embedded systems**

The MYSQL database software is a client/server system that consists of a multi-threaded SQL server that supports different backend, several different client programs and libraries, administrative tools and a wide range of Application Programming Interface(APIs). A large amount of contributed MYSQL software is available:

Modern day websites seem to be relying more and more on compel the Structured Query Language is a very popular database language, and its standardization makes it easy to store, update and access data. One of the most powerful SQL servers out there is called MYSQL and surprisingly enough, it’s free.

Some of the features of MYSQL include: Handles large databases, in the area of 50,000,000+records. No memory leaks. Tested with a commercial memory leakage detector (purify). A privilege and password system which is very flexible and secure, and which allows host-based verification. Passwords are secure since all password traffic when connecting the server is encrypted.

**Features of MYSQL**

**Client/server Architecture:** MYSQL is a client/server system. There is a database server (MYSQL) and arbitrarily many clients (application programs), which communicate with the server. The clients can run on the same computer as the server or on another computer.

**SQL Compatibility:** As before said SQL is a standardized language for querying and updating data and for the administration of a database. Through the configuration setting sol-mode we can make the MYSQL server behave for the most part compatibly with various database systems.

**Stored procedures:** Stored procedures (SPs for short) are generally used to simplify steps such as inserting or deleting a data record.

**Triggers:** Triggers are SQL commands that are automatically executed by the server in certain database operations INSERT, UPDATE, and DELETE, MYSQL has supported triggers.

**Replication:** Replication allows the contents of a database to be copied (replicated) onto a number of computers to increase protection against system and to improve the speed of database queries.

**Platform independence:** MYSQL can be executed under a number of operating systems. The most important are Apple Macintosh OS X, Linux, Microsoft Windows, and the Unix.

**Speed:** MYSQL is considered a very fast database program.

**2. SYSTEM STUDY**

**2.1 EXISTING SYSTEM**

In present system every website for farmer equipment sales process system work such as sales and view farmer equipment information is carried out manually. The present system is much time consuming. Hence for that more human resource is needed and a huge amount of time is needed for it. In present system Buyer have to walk in market to get correct price of the product and get the product. This system does not have any facilities to view item price and order of item. The main drawbacks of the existing system farmer have to go market for sell their farmer product this is more tedious task to farmer.

**2.1.1 DISADVANTAGES OF EXISITNG SYSTEM**

* Time consuming process.
* Needs of man power.
* Difficult to get item information.
* Farmer to sell agri product is very difficult.
* Every time they have to go market to sell a product.

**2.2 PROPOSED SYSTEM**

The drawbacks, which are faced during existing system, can be eradicated by using the farmer equipment sales process system. The main objective of the proposed system is to provide a Buyer-friendly interface. The system, which is proposed, now computerizes all the processes involved in farmer product. Project proposes a new technique to farmer can sell the item in this application farmer can arrange an order of the item with a certain time limit.

**2.2.1 ADVANTAGES OF PROPOSED SYSTEM**

* This application will reduce manual work and maintain updates in database from time to time.
* Less time consuming process.
* Easy way to get item information and price details.
* Farmer to sell agri product is very easiest way.
* No need to go market to sell a product.
* Easy report generation.

**3. SYSTEM DESIGN**

**3.1 FILE DESIGN**

System design is the process of planning and creating a new system to facilitate the transition from traditional methods to a modern, digital platform, particularly in the context of an e-application for farmers to sell their agricultural products through e-auctions. The purpose of the design phase is to move from the problem domain, where manual processes and slow information dissemination are prevalent, to the solution domain, where an efficient, user-friendly, and accessible platform is created. The design of this e-application is crucial as it directly impacts the effectiveness and usability of the system, ensuring that both farmers and buyers can easily engage in transactions. This phase involves translating the logical aspects of the application—such as farmer and buyer interactions, product listings, and bidding processes into the physical components of the system, including the database structure, user interfaces, and the overall flow of information. The system design, therefore, is a pivotal step in developing a robust platform that meets the needs of all users involved in the agricultural product market.

**3.2 INPUT DESIGN**

Input design is a critical phase in the development of the E-Application for Farmers to sell their food products through E-Auction. It directly impacts the efficiency and effectiveness of the system. In this application, careful attention is given to designing the input methods to ensure accurate and seamless data entry for both farmers and buyers. Farmers need to easily input product details such as name, description, quantity, and base price, while buyers must be able to view this information and place bids effortlessly. Faulty input design can lead to significant problems in the system, such as inaccurate product listings or bidding issues. Therefore, the input data is meticulously analyzed and designed to provide a user-friendly interface, ensuring that both farmers and buyers can interact with the system smoothly. The decisions made during the input design process are crucial in reducing manual work, maintaining up-to-date information in the database, and ultimately supporting the successful operation of the entire e-auction platform.

**3.3 DATABASE DESIGN**

The database design for the E-Application for Farmers involves creating tables that store data efficiently, with each table comprising rows (records) and columns (fields). This structure supports the system's core functions, such as managing farmer and buyer registrations, product listings, and bidding processes. The design minimizes redundancy, maintains a normalized format, and allows for null values where necessary. This ensures efficient data retrieval, enabling buyers to access product details and bid information easily. For visual reference, screenshots of the database schema are provided in the appendix.

**TABLE DESIGN**

**Table name :** Admin login

**Primary key:** aid

**Description :** This table is used to store the admin login details.

| **Field name** | **Data type** | **Width** | **Description** |
| --- | --- | --- | --- |
| aid | integer | 10 | Admin identification |
| uname | varchar | 15 | Buyer name |
| pwd | varchar | 08 | Password |

**Table name :** Farmer Registration

**Primary key:** fid

**Description :** This table is used to store the farmer registration details.

| **Field name** | **Data type** | **Width** | **Description** |
| --- | --- | --- | --- |
| fid | integer | 10 | Farmer identification |
| name | varchar | 15 | Farmer name |
| email | integer | 10 | Farmer email id |
| addr | varchar | 30 | Address of the farmer |
| contact | varchar | 15 | Contact of the farmer |
| pwd | integer | 10 | Farmer Password |

**Table name :** Farmer Logina

**Foreign key:** fid

**Description :** This table is used to store the farmer login details.

| **Field name** | **Data type** | **Width** | **Description** |
| --- | --- | --- | --- |
| fid | integer | 10 | Farmer identification |
| name | varchar | 15 | Farmer name |
| email | integer | 10 | Farmer email id |

**Table name :** Buyer Registration

**Primary key:** bid

**Description :** This table is used to store the Buyer registration details.

| **Field name** | **Data type** | **Width** | **Description** |
| --- | --- | --- | --- |
| bid | integer | 10 | Buyer identification |
| bname | varchar | 15 | Buyer name |
| email | integer | 10 | Buyer email id |
| addr | varchar | 20 | Address of the Buyer |
| contact | varchar | 15 | Buyer contact |
| pwd | integer | 10 | Buyer Password |

**Table name :** Buyer Login

**Foreign key:** bid

**Description :** This table is used to store the Buyer login details.

| **Field name** | **Data type** | **Width** | **Description** |
| --- | --- | --- | --- |
| bid | integer | 10 | Buyer identification |
| name | varchar | 15 | Buyer name |
| email | integer | 10 | Buyer email id |

**Table name :** Product

**Primary key:** pid

**Description :** This table is used to store the product details.

| **Field name** | **Data type** | **Width** | **Description** |
| --- | --- | --- | --- |
| pid | integer | 10 | Product identification |
| selleremail | varchar | 15 | Farmer email id |
| prodname | varchar | 15 | Product name |
| descp | varchar | 15 | Product description |
| type | varchar | 20 | Product type |
| fname | varchar | 15 | Farmer name |
| fno | integer | 10 | Farmer contact |
| floc | varchar | 15 | Farmer location |
| km | interger | 10 | Farmer location in kilometer |
| photo | file | - | Product image |
| price | integer | 10 | Product original price |

**Table name :** Biddapp

**Primary key:**oid

**Foreign key :** fid, pid ,bid

**Description :** This table is used to store the bidding details.

| **Field name** | **Data type** | **Width** | **Description** |
| --- | --- | --- | --- |
| oid | integer | 10 | Order identification |
| bid | integer | 10 | Buyer identification |
| fid | integer | 10 | Farmer identification |
| prodname | varchar | 10 | Product name |
| oprice | integer | 10 | Product original price |
| bprice | integer | 10 | Product bidding price |
| sts | varchar | 10 | Bidding Status |

**3.4 CODE DESIGN**

Code is an order collection of symbols designed to provide unique identification of an entry or attribute. Sometimes used in the place of name of the item they can be specified all object’s physical or performances characteristics or operational instructions. They can also show inter relationship and may sometime be used to achieve secrecy or confidentiality.

Code designs submit the Buyer id, farmer id, product id and order id generate the unique form design.

**3.5 OUTPUT DESIGN**

The output design must be in such a way the Buyer must able to understand the given details. So each detail given in the output should have some meaning in displaying the data. The output design is displayed in the form of data view. Output Design generally refers to the results and information’s that are generated by the system for many end-Buyers, output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. The objective of a system finds its shape in terms of the output. The analysis of the objective of a system leads to determination of outputs. External outputs are those whose destination will be outside the organization and which require special attention as they project the image of the organization. Internal outputs are those whose destination is within the organization. It is to be carefully designed as they are the Buyer’s main interface with the system.

**4. SYSTEM TESTING AND IMPLEMENTATION**

**4.1 SYSTEM TESTING**

Testing is carried out after the development of the proposed system. The principle activity of system development is preparing the source code. In this system the source code is developed for each module separately. The source code is prepared for master files and they are compiled and corrected. Then the source code for the transaction files are prepared, compiled and corrected. Then the modules are combined and corrected as a whole module.

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segments has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present. Test configuration includes test plan and test cases and test tools.

**TESTING OBJECTIVES**

Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

* [Finding defects](http://istqbexamcertification.com/what-is-defect-or-bugs-or-faults-in-software-testing/) which may get created by the programmer while developing the software.
* Gaining confidence in and providing information about the level of [quality](http://istqbexamcertification.com/what-is-software-quality/).
* To prevent defects.
* To make sure that the end result meets the business and Buyer requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
* To gain the confidence of the customers by providing them a quality product

**Testing methodologies**

Testing methodologies are the strategies and approaches used to test a particular product to ensure it is fit for purpose. Testing methodologies usually involve testing that the product works in accordance with its specification, has no undesirable side effects when used in ways outside of its design parameters and worst case will fail-safely (e.g. a nuclear reactor will shut down on failure).

**Unit testing**

Unit testing is essential for verifying the code produced during the development of the E-Application for farmers to sell their food products through E-Auction. The goal is to ensure the internal logic of the modules, such as farmer and buyer registration, product listing, bidding, and information retrieval, works as intended. Using the detailed design description as a guide, critical paths are tested to identify and correct errors within each module. These tests were conducted during the programming phase to ensure the system's functionality aligns with the requirements of facilitating seamless transactions between farmers and buyers.

**Integration testing**

Integration testing is a systematic technique used to construct and verify the program structure while simultaneously uncovering errors related to the interfaces between modules. In the context of the E-Application for farmers to sell their food products through e-auction, integration testing ensures that all the modules—such as farmer registration, buyer registration, product information management, and bidding details—work together seamlessly. The objective is to take the unit-tested modules and assemble them according to the design, ensuring they interact correctly to form a cohesive system. During this process, the entire application, including features like information retrieval for buyers, product bidding, and database updates, is tested as a whole. Any errors that arise in the interfaces between modules, such as issues in product information retrieval or bidding functionality, are identified and resolved to ensure a smooth and functional user experience.

**Validation testing**

In the validation testing for the E-Application for farmers to sell their food products through e-auction, we need to ensure that the input fields are functioning correctly according to the specified conditions. For instance, in the product name field, only alphabetical characters and certain special symbols should be allowed, and numeric values should be restricted. During testing, if numeric values are entered into fields designated for text, such as product names, an error should be triggered. Each module of the application is tested with incorrect inputs to ensure that validation rules are enforced. For example, the product name field should only accept alphabetic characters, while fields like quantity and price should strictly accept numeric values. This validation ensures that the data entered by farmers and buyers is accurate and conforms to the expected format, thereby maintaining the integrity of the information in the system.

**Functional testing**

The functional testing methodology for the e-application helping farmers sell products through e-auctions involves several key stages. It begins with **Unit Testing** of individual modules like product listing and bidding. **Integration Testing** follows, ensuring these modules interact correctly, such as buyers viewing and bidding on products while the admin manages data. **System Testing** then verifies the entire system's functionality, including real-time updates and accurate information retrieval. Finally, **Acceptance Testing** ensures the system meets user needs, providing a smooth experience. Throughout, the system is validated by comparing actual outputs with expected results to ensure efficiency and accuracy.

**4.2 SYSTEM IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the Buyers that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of Buyers. The implementation process begins with preparing a plan for the implementation of the system.

According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed. Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the Buyers confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system. As the part of system testing we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the Buyer requirements are satisfied. The ultimate aim is quality assurance.

**4.3 SYSTEM MAINTENANCE**

According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed. Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the Buyers confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system. As the part of system testing we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the Buyer requirements are satisfied. The ultimate aim is quality assurance.

**5. CONCLUSION**

The application works according to the restrictions provided in their respective browsers. The application satisfies the Admin. The speed of the transactions become more enough now. The website creation is the web designing project created for displaying the details about the web portal using the coding languages like Html & Css for designing. The interface are so designed and channeled the admin can never make any mistake while using the application, till the time either they save or cancel the current operation all other operations are blocked. This project has been successfully developed and interpreted and system was developed according to the admin requirements. The system produces accurate results and it also reduces a lot of overheads, which the manual system faced. The information requirements may still increase.

**FUTURE ENHANCEMENT**

There is a wide scope for future development of the software. The world of computer fields is not static it is always subject to change. The technology which is famous today will become outdated very next day. To keep abstract of technical improvements, the system may be refinement. So it is not concluded. Yet it will improve with further enhancements.

It is essential to change the software when new software arrives with more advanced feathers. So it is much necessary for further development. Further enhancements can be done in an efficient manner with disruption to the system.

**BIBLIOGRAPHY**

**BOOK REFERENCES**

1. Jesus Castagnetto, Sascha Schumann, “Professional Php Programming”, Addison wisely Publication, Fifth Edition.
2. Jay Greenspan, Brad Bulgar, “Mysql/Php Database Applications”, Tata McGraw-Hill Publishing Company, Third Edition.
3. William Stallings, “Cryptography And Network Security”, Tata McGraw-Hill Publishing Company, Third Edition.
4. Bruce Schneier, “Applied Cryptography”, Pearson Education, Second Edition.
5. Rogers Pressman, “Software Engineering and Applications”, Galgotie Publication, Sixth Edition.

**REFERENCES WEBSITE**

* 1. [www.onlinetutorial.com](http://www.onlinetutorial.com)
  2. www.cryptography.com
  3. www.tenders.com
  4. www.computerhope.com/starthtm.htm‎
  5. [www.webdesign.about.com/od/webdesignbasics/u/webdesignbasics.htm](http://www.webdesign.about.com/od/webdesignbasics/u/webdesignbasics.htm)
  6. www.w3schools.com/php/php\_mysql\_intro.asp

**APPENDIX**

**A. DATA FLOW DIAGRAM**

**Level 0:**

Provide

Admin

Farmers

Id

Buyer

Request

Response

**Level 1:**

Manage

View Farmer details

Admin

uname

pro\_db

seller\_db

View Product details

Password

bidder\_db

View Buyer details

Process

**Level 2:**

Farmers

Buyername

Password

View Buyer bidding

pro\_db

Buyer

Add product

View product

Bid product

bid\_db

**B. SYSTEM FLOW DIAGRAM**

Admin

Login

Invalid

Valid

Manage

Buyer Details

Product

Details

Farmer Details

Bidding

Details

Storage

Product

Database

Buyer Database

Farmer

Database

Biddingapp Database

Report

Buyer Report

Farmer Report

Product Report

Bidding Report

**C. ENTITY RELATIONSHIP DIAGRAM**

Farmer

Manage

Admin

Product details

Add/ View

Buyer details

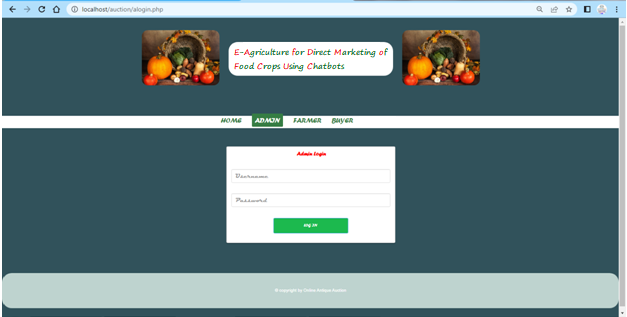
View/Bid

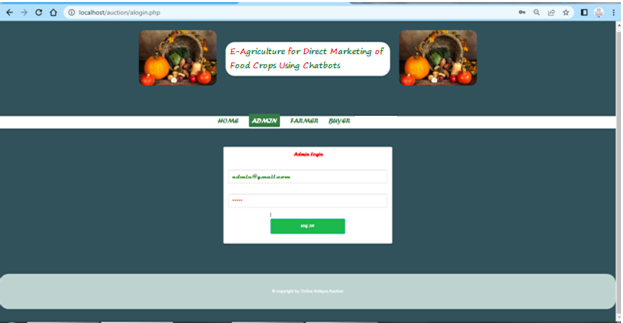
**D. SAMPLE SCREENS**

**Home page**

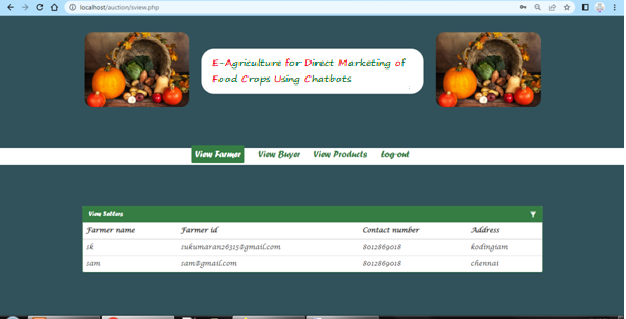
****

**Admin Login**

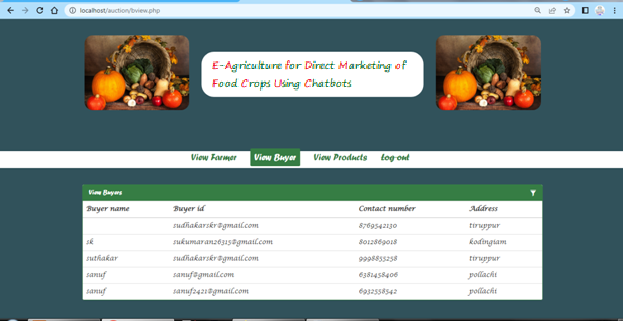




**Admin View Farmer**

****

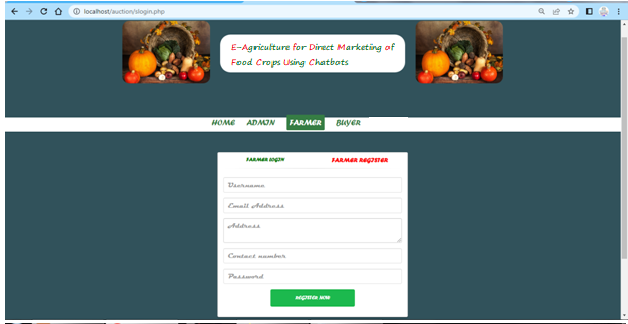
**Admin View Buyer**

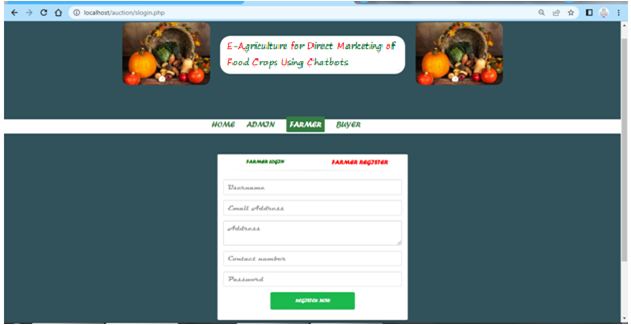
****

**Admin View Product**

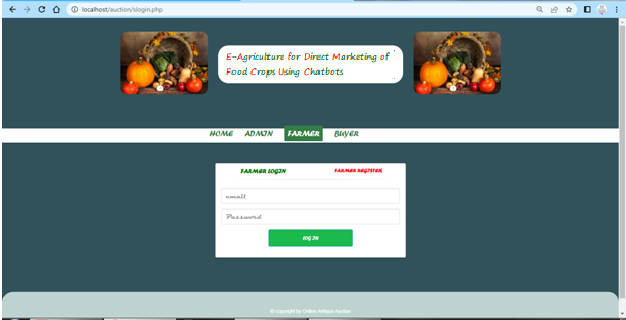
****

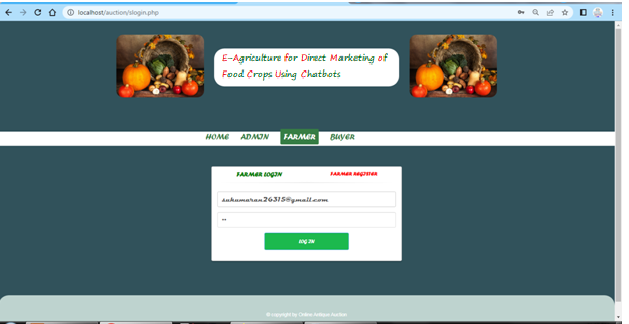
**Farmer Registration**

****

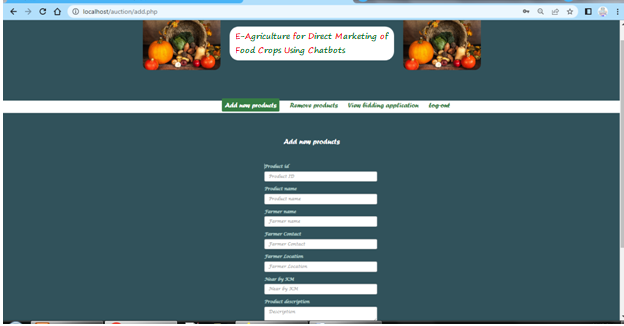
****

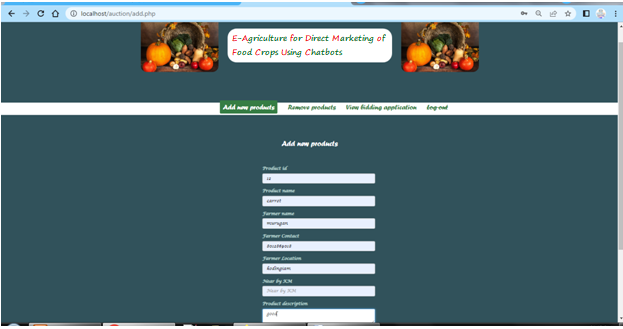
**Farmer Login**

****

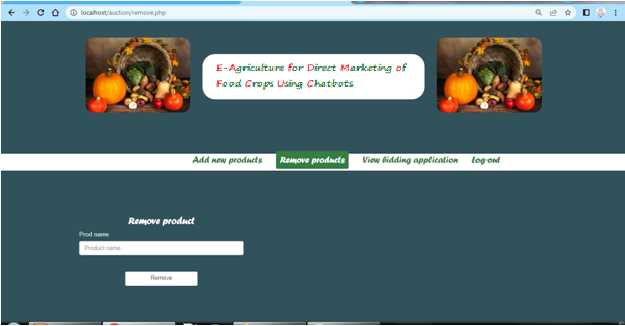
****

**Farmer Add Product**

****

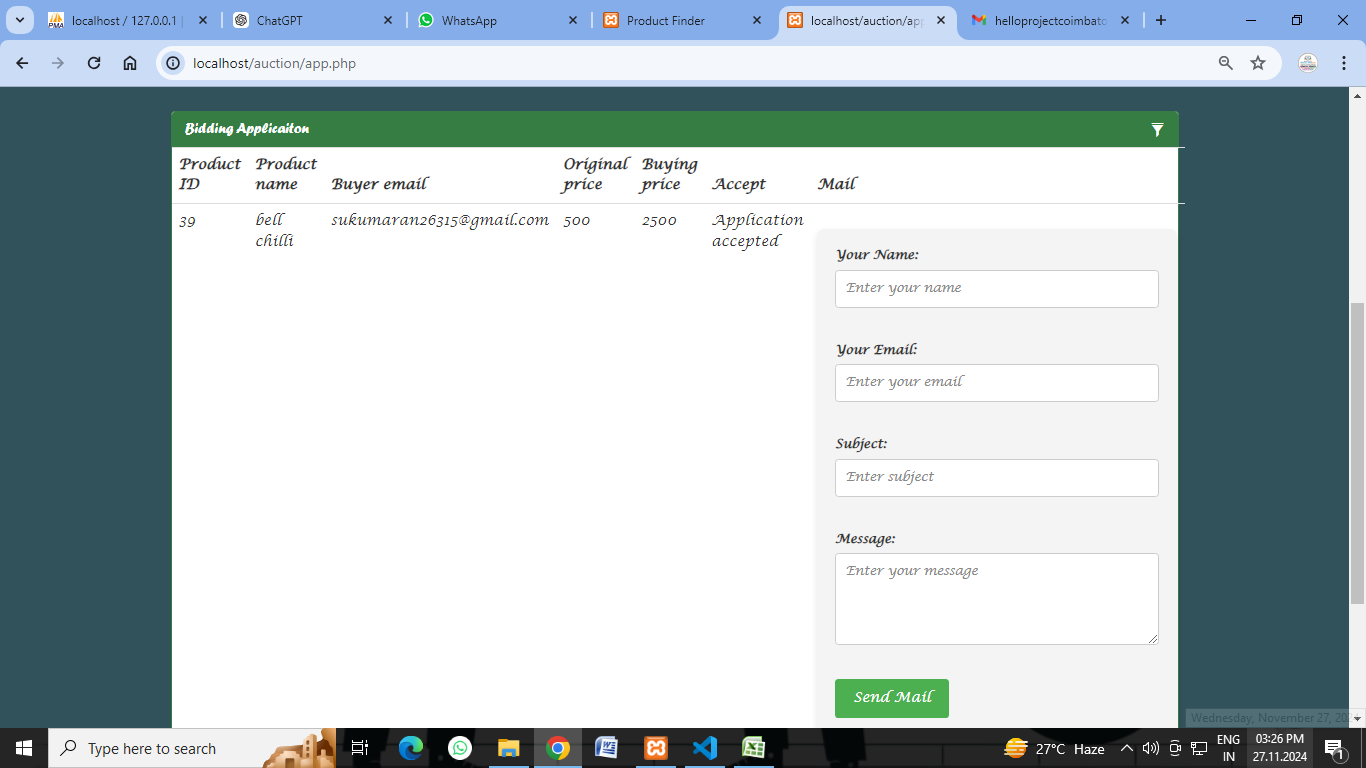
****

**Farmer Remove Product**

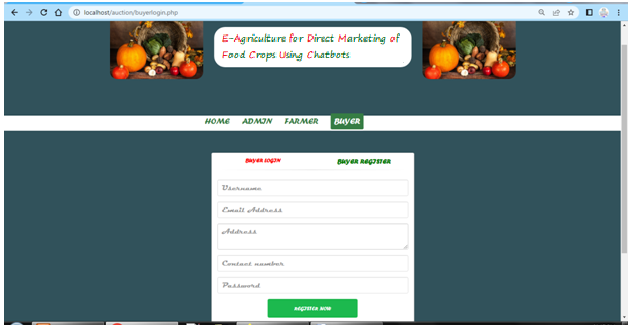
****

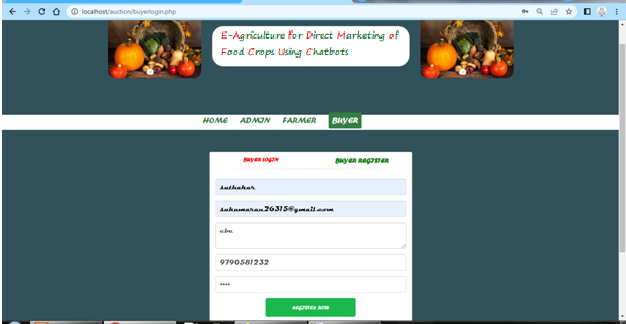
****

**Farmer View Bidding**

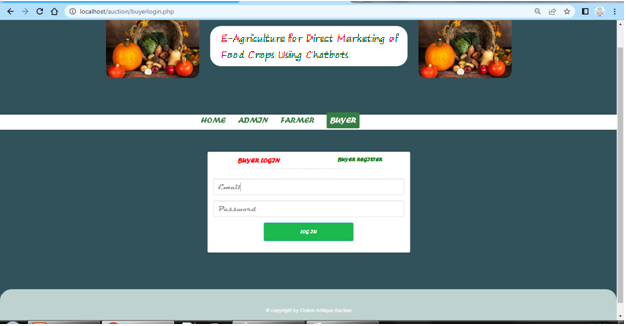
****

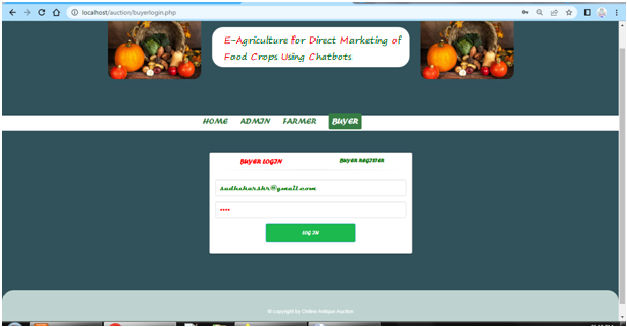
**Buyer Register**

****

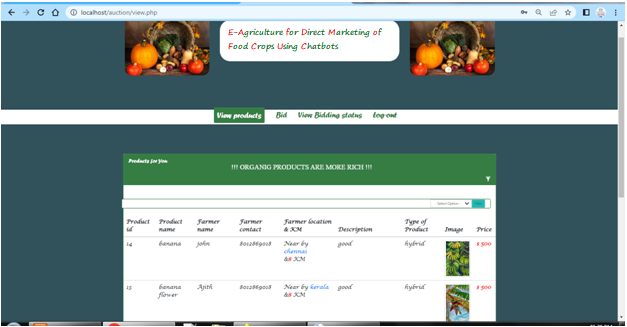
****

**Buyer Login**

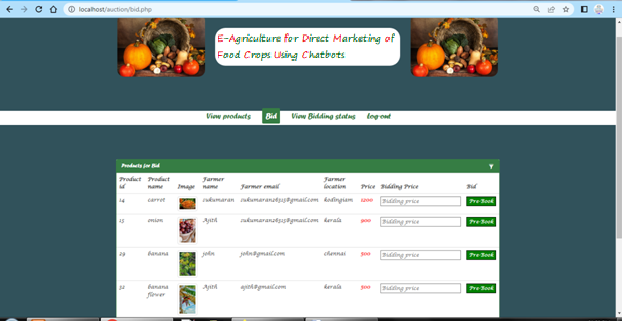
****

****

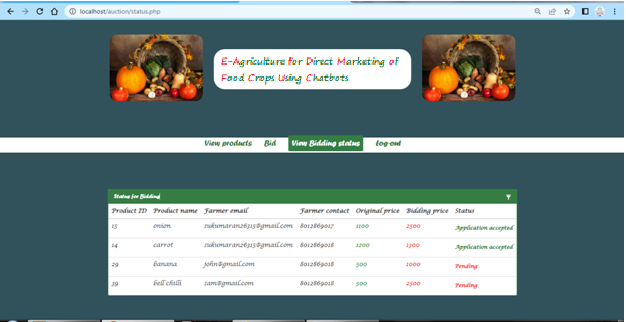
**Buyer View Product**

****

**Buyer View Bid**

****

**Buyer View Bidding Status**

****

**E. SAMPLE CODING**

<html >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title></title>

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

<link rel="stylesheet" type="text/css" href="css/bootstrap.css" />

<script src="js/jquery.js" ></script>

<script src="js/jquery.min.js"></script>

<?php

session\_start();

$id=$\_SESSION['sellerid'];

$\_SESSION['sellerid']=$id;

include('config.php');

?>

<style>

.banner

{

margin:20px;

}

.row{

margin-top:40px;

padding: 0 10px;

}

.clickable{

cursor: pointer;

}

.panel-primary .panel-heading {

font-size: 15px;

background-color:#367d44;

border-color:#367d44;

}

.panel-primary {

border-color:#367d44;

}

.panel-heading div {

font-size: 15px;

}

.panel-heading div span{

margin-left:5px;

}

.panel-body{

display: none;

}

.menu

{

margin-top:100px;

}

.menu a

{

font-size:22px;

color:#367d44;

margin-right:20px;

text-decoration:none;

}

.menu .active

{

background-color:#367d44;

padding:10px;

border-radius:3px;

text-decoration:none;

color:#fff;

}

</style>

<script>

(function(){

'use strict';

var $ = jQuery;

$.fn.extend({

filterTable: function(){

return this.each(function(){

$(this).on('keyup', function(e){

$('.filterTable\_no\_results').remove();

var $this = $(this),

search = $this.val().toLowerCase(),

target = $this.attr('data-filters'),

$target = $(target),

$rows = $target.find('tbody tr');

if(search == '') {

$rows.show();

} else {

$rows.each(function(){

var $this = $(this);

$this.text().toLowerCase().indexOf(search) === -1 ? $this.hide() : $this.show();

})

if($target.find('tbody tr:visible').size() === 0) {

var col\_count = $target.find('tr').first().find('td').size();

var no\_results = $('<tr class="filterTable\_no\_results"><td colspan="'+col\_count+'">No results found</td></tr>')

$target.find('tbody').append(no\_results);

}

}

});

});

}

});

$('[data-action="filter"]').filterTable();

})(jQuery);

$(function(){

// attach table filter plugin to inputs

$('[data-action="filter"]').filterTable();

$('.container').on('click', '.panel-heading span.filter', function(e){

var $this = $(this),

$panel = $this.parents('.panel');

$panel.find('.panel-body').slideToggle();

if($this.css('display') != 'none') {

$panel.find('.panel-body input').focus();

}

});

$('[data-toggle="tooltip"]').tooltip();

})

</script>

</head>

<body style="background-color: #31525B"> <br><br>

<div class="container">

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

<img src="images/logo.jpg" class="img-responsive" style="border-radius : 20px" />

</div>

<div class="col-md-6"><br>

<h1 style="font-size: 50px ; font-family: Forte ; background-color: white ; text-align: center ; border-radius: 30px ; color: green"> <b style="color: red">O</b>NLINE <b style="color: red">A</b>NTIQUE <b style="color: red">A</b>UCTION</h1>

</div>

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

<img src="images/logo.jpg" class="img-responsive" style="border-radius : 20px" />

</div>

</div>

<div class="menu" style="background-color: white ; font-family: Forte">

<div class="container" style="padding-left: 290px">

<ul class="list-inline">

<li><a href="add.php">Add new products</a></li>

<li ><a href="remove.php">Remove products</a></li>

<li><a href="a pp.php" class="active">View bidding application</a></li>

<li><a href="logout.php"><del>Log out</del></a></li>

</ul>

</div>

</div>

<br><br><br>

<div class="container">

<div class="row">

<div class="col-md-12">

<div class="panel panel-primary">

<div class="panel-heading">

<h3 class="panel-title" style="font-family: Forte"> Bidding Applicaiton</h3>

<div class="pull-right" style="margin-top:-15px;">

<span class="clickable filter" data-toggle="tooltip" title="Toggle table filter" data-container="body">

<i class="glyphicon glyphicon-filter"></i>

</span>

</div>

</div>

<div class="panel-body">

<input type="text" class="form-control" id="dev-table-filter" data-action="filter" data-filters="#dev-table" placeholder="Filter" />

</div>