**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTERS** | **TITLE** |
| 1 | **INTRODUCTION** |
| 2 | Organization Profile |
| 3 | **SYSTEM ANALYSIS**  3.1 Existing System  3.1.1 Disadvantages  3.2 Proposed System  3.2.1 Advantages  3.3. Feasibility Study  3.3.1 Economical Feasibility  3.3.2 Technical Feasibility  3.3.3 Operational Feasibility |
| 4 | **SYSTEM DESIGN**  4.1 Modules  4.2 Modules Description |
| 5 | **REQUIREMENTS SPECIFICAITON**  4.1 Hardware Requirements  4.2 Software Requirements  5.3 Programming Environment |
| 6 | **SYSTEM DESIGN**  6.1 Database Design  5.6 Table Design  5.6.1 additem  5.6.2 customerpurchase  5.6.3 customerreg  5.6.4 formerreg  5.6.5 Purchasedetails |
| 7 | 7.1 Input Design  7.2 Output Design  7.3 Data Flow Diagram  7.4 Sequence Diagram  7.5 Activity Diagram  7.6 Usecase Diagram  7.7 Class Diagram |
| 8 | **SYSTEM IMPLEMENTATION**  8.1 Testing  8.1.1 Levels of Testing  8.1.2 Types of Testing  8.1.3 Unit Testing  8.1.4 Integration Testing |
| 9 | **CONCLUSION** |
| 10 | **FUTURE ENHANCEMENT** |
| 11 | **CODINGS** |
| 12 | **SCREENSHOT** |
| 13 | **BIBLIOGRAPHY** |

INTRODUCTION

**1.INTRODUCTION**

The main objective for developing this projects was to create on online farmer supportive application that satisfies his requirements. This Project provides a lot of features to face the day to day challenges faced by farmers. Admin can add, edit, view the available details of Farmer goods. Farmer can update his status and can get his queries satisfied .Now the clients can check the availability and place their orders and Afterthat admin can check the placed orders and then send the quotation of the requested product to the client. If the client satisfy with the quotation ,they have to send the confirmation mail to the admin.If admin receive the confirmation mail from client then the product is will be delivered. Service section in admin module is to receive the feedback about the sold products. Based on the items purchase from client, fault one will be corrected.

This projects contains a lot of advance module which makes the back end system very useful. The payment can be done in both online and offline modes.

**3.1 EXISTING SYSTEM**

* In the Existing System all transactions, dealings of products, purchase of products, were done manually which is time consuming.
* Reports are prepared manually as and when needed, Maintaining of reports is very tedious work.
* To buy any product user has to collect information about it either by visiting the shop or asking people which is the better one.
* There is no computer system for handling payments. All Calculations are performed manually which may not be accurate always. Maintaining the records is really tedious work.
* The traditional culture of farming has been depreciated in recent days and the farmers are struggling nowadays with the basic ideology of farming.

**3.1.1.Some of the disadvantages are:**

* Less accuracy.
* Less security.
* Lot of paper work.
* Need huge man power.
* Slow data processing.
* Not user friendly environment.
* Difficult to keep old records.
* Time consuming

**3.2 PROPOSED SYSTEM**

* Online Farmer good shopping portal is a specific requirement of the client that integrate the buying and selling services specifically to their customers.
* Need for the new system is due to major drawbacks of existing system.
* Report can be generated at a any time within few seconds ,so that manual labour is not required, and also analysis can be performed much more frequently which help in tasking decision.
* The details regarding all users, products can also be maintained as their information is very helpful and sometimes becomes a critical requirements.
* Allow user to get registered from their places and transact for the required product.
* They will be provided with the complete guidance on cropping methodology and amount of pesticides and fertilizers required for complete utilization of area provided
* The farmer can apply for loans and funds for their financial support
* The farmers can easily connect with all other farmers available and get help at any time

**3.2.1 Some of the advantages are:**

* Less time consumption.
* More efficient.
* No need of lots of man power.
* Quick data processing
* Easy communication
* Get complete guidance
  1. **FEASIBILITY STUDY**

Feasibility study was conducted on the candidate system to see whether it was proceed with the new system.

**3.3.1 ECONOMICAL FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

**3.3.2 TECHNICAL FEASIBILITY**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology.Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

**3.3.3 OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**OVERVIEW OF THE PROJECT**

**4. SYSTEM DESIGN**

**4.1 MODULES**

1. Admin Module

2.Customer Module

3.Farmer Module

**4.2 Modules Description**

**ADMIN and FARMER MODULE:**

Admin module is used for maintaining the overall activities of the project. In this project, admin can add the available details (i.e. stock details) of the farmer goods and he/she can receive the product price details request from user. Based on the product request from the user, admin have to send the quotation for the requested products and admin can get the confirmation mail for the placed order and he/she can receive the complaints about the sold products. Based on the complaints, admin have to allot service to the fault one. And when the farmer is stuck with idea of cropping, he gets the required information based on the inputs provided and he will be able to get the financial support on examining the documents provided and the eligibility of the farmer

**CUSTOMER MODULE:**

Client module is used for placing the orders after checking the products stock. After checking the products stock, customer can request the quotation of that particular product to admin and he/she can receive the quotation form for the requested product from admin. If the customer found any fault or damage in the product, they can complaint about the fault to the service section in admin module.

**SYSTEM STUDY**

**5. SYSTEM STUDY**

**5.1. HARDWARE REQUIREMENTS**

The hardware used for the development of the project is:

PROCESSOR : Pentium III 700 MHz

RAM : 2 GB RAM

MONITOR : 15” COLOR

HARD DISK : 200 GB

FLOPPY DRIVE : 1.44 MB

CD DRIVE : LG 52X

KEYBOARD : keys enhanced

MOUSE : Logitech

**5.2. SOFTWARE REQUIREMENTS**

ENVIRONMENT : Visual Studio .NET 2010

.NET FRAMEWORK : Version 34.0

LANGUAGE : ASP.NET WITH C#

BACK END : MS-SQL SERVER 2019

**5.3 SOFTWARE DESCRIPTION**

**THE .NET FRAMEWORK**

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).

2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which program run. The most important features are

* Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
* Memory management, notably including garbage collection.
* Checking and enforcing security restrictions on the running code.
* Loading and executing programs, with version control and other such features.
* The following features of the .NET framework are also worth description:

**Managed Code**

The code that targets .NET, and which contains certain extra

Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

**Managed Data**

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

**Common Type System**

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

**Common Language Specification**

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

**THE CLASS LIBRARY**

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

Features of MS SQL Server

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects,

They are,

1. TABLE

2. QUERY

3. FORM

4. REPORT

5. MACRO

**TABLE:**

A database is a collection of data about a specific topic.

**VIEWS OF TABLE:**

We can work with a table in two types,

1. Design View

2. Datasheet View

**Design View**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

**Datasheet View**

To add, edit or analyses the data itself we work in tables datasheet view mode.

**QUERY:**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited). Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

**SYSTEM DESIGN**

**6. SYSTEM DESIGN**

**6.1 DATABASE DESIGN**

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to the object classes and named relationships.

However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system.

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

Determine the relationships between the different data elements.Superimpose a logical structure upon the data on the basis of the relationships.

**The design process consists of the following steps**

1. Determine the purpose of your database- This helps prepare you for the remaining steps.

2. Find and organize the information required- Gather all of the types of information you might want to record in the database, such as product name and order number.

3. Divide the information into tables- Divide your information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.

4. Turn information items into columns- Decide what information you want to store in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employee table might include fields such as Last Name and Hire Date.

**6.2 TABLE DESIGN**

**Farmer login table**

|  |  |
| --- | --- |
| **COLUMN NAME** | **DATA TYPE** |
| Guidance | varchar(50) |
| Business | varchar(50) |
| Communication | varchar(50) |
| Financial support | varchar(50) |

**Guidance Login**

|  |  |
| --- | --- |
| COLUMN NAME | DATA TYPE |
| Home | int(autoincrement) |
| 1. Length | varchar(50) |
| 1. Breadth | varchar(50) |
| 1. Width | varchar(50) |
| Crop type | varchar(50) |
| Soil type | varchar(50) |

**Business Table**

|  |  |
| --- | --- |
| **COLUMN NAME** | **DATA TYPE** |
| Buying | int(autoincrement) |
| 1. Pestiicides | varchar(50) |
| 1. Fertilizers | varchar(500) |
| 1. Seed | varchar(50) |
| Selling | varchar(50) |
| 1. Vegetables | varchar(50) |
| 1. Fruits | varchar(50) |
| 1. Crops | varchar(50) |
| Status | varchar(50) |
| photo | varchar(50) |

**Farmerreg**

|  |  |
| --- | --- |
| **COLUMN NAME** | **DATA TYPE** |
| id | int(autoincrement) |
| Name | INT |
| Fathername | varchar(50) |
| Age | varchar(50) |
| Gender | varchar(50) |
| Address | varchar(50) |
| Contactno | varchar(50) |
| Username | varchar(50) |
| Password | varchar(50) |
| Emailid | varchar(50) |
| Status | varchar(50) |
| photo | varchar(50) |

**Purchasedetails3**

|  |  |
| --- | --- |
| COLUMN NAME | DATA TYPE |
| id | int(autoincrement) |
| Productname | varchar(50) |
| Category | varchar(50) |
| Quantity | varchar(50) |
| Cost | varchar(50) |
| Totalcost | varchar(50) |
| Paymenttype | varchar(50) |
| Purchasedate | varchar(50) |
| Status | varchar(50) |

**7.1 INPUT DESIGN:**

Input design is the process of converting user-oriented input into computer-based format.The goal of designing input data is to make data entry as easy as possible and free from errors. Errors in the input are handled in the input design. The input data is the lifeblood of a system and have to be analyzed and designed with utmost case and consideration. The decisions made during the input design are:

* + - To provide cost effective method of input
    - To achieve the highest possible level of accuracy
    - To ensure that input is understand by the user

In this project various resources like textboxes, dropdown list, link buttons etc are used for accepting user inputs.The main type of inputs to this project are files which can be a text file or image file. The other inputs to this project include file names, destination user name, integer values for splitting files into packets. The file name is nothing but the file that is to be splitted into packets and transferred to the specified destination. Destination user name is the user who is going to reassemble the splitted packets.

Taking into account, all the above needs of the input designing, the new process designing could have the following features. In all cases processing should be automatic and manual work should be kept minimum. Similarly, the user input should also be kept minimum or least extend possible. By giving the above input calculations should be automatic. The auto calculation is the motto of this new process, while providing the auto calculation the results with minimum time operation, should be accurate and free from errors.

**7.2 OUTPUT DESIGN:**

Output design generally refers to the results and information’s that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. The basic requirements of output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose.

The important task of any system lies in the capability of producing high quality output and reports. Computer output is the most important and direct source of information to the user. A good output design contains all required information and well formatted for the better visualization and avoid the complexity in displaying the data this system was developed by keeping in mind and proper output is displayed for the users accurately.

In this project the output is the reassembled file that was received by the user at the destination and the messages found in the client’s inbox and the report that was generated which shows the number of packets transferred along with acknowledgement. These outputs are obtained in a new page on the browser.

**7.3 DATA FLOW DIAGRAM (DFD)**

A data flow diagram (DFD) is a graphical representation of the “flow “of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing .The DFD is designed to show how a system is designed into smaller portions and to highlight the flow of data between those parts.

- A data source or destination

- Represents flow of data

- Process that transforms data stream

- Represents data storage

**DATA FLOW DIAGRAM:**

A DFD is a modeling tool that allows us to show a system as a network of processes (mini systems) connected to each other by paths of data.

**DATA FLOW:**

Each arrow in a DFD Represents a path (or pipeline of conveyer belt) of data, called the Data Flow.

**PROCESS:**

A process represents an automated or manual activity that transforms incoming data flows to outgoing data flows.

**DATA-STORE:**

A Data-Store holds the data (for reference), which the system cannot remember.

External events and system responses ate From/To “Sources” or “Destination” of the system.

**Components of DFD**

The basic **components of DFD** are following

Source/Destination

Process

Data Source

**DFD** are based on the strategy of **structured techniques**, Namely:

1. Graphics
2. Top Down Approach
3. Separation of logical from the physical
4. Partitioning

DFD is do not show

* How things begin & end
* Loops
* Decisions

Guidance

**DATAFLOW DIAGRAM**

Communication

Admin

Financial Support

Customer

Farmer

item

product details

add,update and delete

Generate bill and print

purchase details

purchase

display

view purchase details

report generation

view sales details

search

display

view product details

7.5 Activity Diagram

**Validate**

**LogOut**

**Item add**

**Add to card**

**Get any help services**

**Reports**

**Approvel from admin**

**Login From account**

**farmer Registration**

**7.6 USEASE DIAGRAM**

User

additems

User

Farmer

user

admin

User

Farmer

7.6Class Diagram

Farmer

+Registration

+Login ()

+Request ()

item

+itemid ()

+itemame ()

user

+userid

+status

+Decryption

orders

+itemid

+userid

feedback

+feedbackid

+userid

**SYSTEM IMPLEMENTATION**

**8.SYSTEM IMPLEMENTATION**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

* The active user must be aware of the benefits of using the system
* Their confidence in the software built up
* Proper guidance is impaired to the user so that he is comfortable in using the application

**SYSTEM IMPLEMENTATION**

Implementation is the most crucial stage in achieving a successful system and giving the user’s confidence that the new system is workable and effective. Implementation of a modified application to replace an existing one. This type of conversation is relatively easy to handle, provide there are no major changes in the system.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

The system implementation consists of several activities. They Are:

* Build and Test networks.
* Build and Test Databases.
* Install and Test new software packages.
* Write and Test new programs.

**DELIVERY PHASE:**

This is the final part of the implementation phase to deliver the new system into operation. To achieve this you must complete the following:

* Conduct a system test to work sure that the new system works.
* Prepare a conservation plan to smooth the transition to the new system.
* Provide training and documentation for individuals of the new system.
* Install databases used by the new system.
* Convert from the old system to the new system and evaluate the project and final system**.**

**SYSTEM TESTING**

**8. TESTING**

**8.1 TESTING**

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

**8.1.1 Levels of Testing**

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below…

Acceptance Testing

System Testing

Integration Testing

Unit Testing

**Client Needs**

Requirements

Design

Code

**8 .1.2 Types of Testing**

Software testing is an important element of software quality assurance and represents the ultimate review of specification, design and coding. It increasing visibility of software as a system element and the costs associates with a software failure are motivating forces for all well planned through testing .The system is tested with giving wrong information. Cascade deletion and, the software developer checks updating. Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

**8.1.3 UNIT TESTING**

The first step in testing is Unit testing. Individual testing are tested to ensure that they operate correctly. Each component is tested independently, without other system components. The module interface is tested to ensure that information properly flow into and out of the program.These are tested that the module operates at boundary established to limit or restrict processing. Unit testing is normally considered as an adjunct to the coding step. After the coding has been developed, received and verified for correct syntax, unit testing begins. Here each module is tested to provide its correctness, validity and determine any missing operations and to verify whether the objectives have been met, errors are noted down and corrected immediately.

Unit testing is the important and major part of the project. So errors can be rectified easily in each module and program clarity can be increased. In this project, the entire system is divided into several modules and is developed individually. Hence, unit testing is conducted to individual modules.

**8.14 INTEGRATION TESTING**

The second step in the testing process is the Integration testing. Integration testing is the systematic technique for constructing the program structure while conducting tests to uncover errors associated with integrating. After the unit test, each module is gradually integrated to form one final system.

All the modules when unit tested will work properly but after integrating the data can cause error one module can have an inadvertent, adverse effect on another; sub functions when combined may not produce the desired major function; global data structures can cause problems, etc.

Hence, the objective of integration testing is to take unit tested modules and build a final program structure. In this project, modules are combined to find the overall performance of the system.

**CONCLUSION & FUTURE ENHANCEMENT**

**9. CONCLUSION**

* In conclusion, the software can be used as an inventory system to provide a frame work that enables the mangers to make reasonable transactions made within a limited time frame.
* Each transaction made on the system go hand in hand with the data being updated in the database in our case it is Microsoft Sql Server 2007 which is the back end.

**10 FUTURE ENHANCEMENT**

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one. The present system gives more security assurance to the database. There is no chance of missing a record. And it is almost nil to errors. It consumes minimum time to work on it. Any type of queries can be resolved using the software.

It is very easy to maintain customer and staff records with the present online system. Comparative analysis is aimed easily using this software. This makes the feedback to be effective to the users

**12. SAMPLE CODING**

**Login.php**

<?php

session\_start();

$user = dataFilter($\_POST['uname']);

$pass = $\_POST['pass'];

$category = dataFilter($\_POST['category']);

require '../db.php';

if($category == 1)

{

$sql = "SELECT \* FROM farmer WHERE fusername='$user'";

$result = mysqli\_query($conn, $sql);

$num\_rows = mysqli\_num\_rows($result);

if($num\_rows == 0)

{

$\_SESSION['message'] = "Invalid User Credentialss!";

header("location: error.php");

}

else

{

$User = $result->fetch\_assoc();

if (password\_verify($\_POST['pass'], $User['fpassword']))

{

$\_SESSION['id'] = $User['fid'];

$\_SESSION['Hash'] = $User['fhash'];

$\_SESSION['Password'] = $User['fpassword'];

$\_SESSION['Email'] = $User['femail'];

$\_SESSION['Name'] = $User['fname'];

$\_SESSION['Username'] = $User['fusername'];

$\_SESSION['Mobile'] = $User['fmobile'];

$\_SESSION['Addr'] = $User['faddress'];

$\_SESSION['Active'] = $User['factive'];

$\_SESSION['picStatus'] = $User['picStatus'];

$\_SESSION['picExt'] = $User['picExt'];

$\_SESSION['logged\_in'] = true;

$\_SESSION['Category'] = 1;

$\_SESSION['Rating'] = 0;

if($\_SESSION['picStatus'] == 0)

{

$\_SESSION['picId'] = 0;

$\_SESSION['picName'] = "profile0.png";

}

else

{

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

$\_SESSION['picName'] = "profile".$\_SESSION['picId'].".".$\_SESSION['picExt'];

}

//echo $\_SESSION['Email']." ".$\_SESSION['Name'];

header("location: profile.php");

}

else

{

//echo mysqli\_error($conn);

$\_SESSION['message'] = "Invalid User Credentials!";

header("location: error.php");

}

}

}

else

{

$sql = "SELECT \* FROM buyer WHERE busername='$user'";

$result = mysqli\_query($conn, $sql);

$num\_rows = mysqli\_num\_rows($result);

if($num\_rows == 0)

{

$\_SESSION['message'] = "Invalid User Credentialss!";

header("location: error.php");

}

else

{

$User = $result->fetch\_assoc();

if (password\_verify($\_POST['pass'], $User['bpassword']))

{

$\_SESSION['id'] = $User['bid'];

$\_SESSION['Hash'] = $User['bhash'];

$\_SESSION['Password'] = $User['bpassword'];

$\_SESSION['Email'] = $User['bemail'];

$\_SESSION['Name'] = $User['bname'];

$\_SESSION['Username'] = $User['busername'];

$\_SESSION['Mobile'] = $User['bmobile'];

$\_SESSION['Addr'] = $User['baddress'];

$\_SESSION['Active'] = $User['bactive'];

$\_SESSION['logged\_in'] = true;

$\_SESSION['Category'] = 0;

//echo $\_SESSION['Email']." ".$\_SESSION['Name'];

header("location: profile.php");

}

else

{

//echo mysqli\_error($conn);

$\_SESSION['message'] = "Invalid User Credentials!";

header("location: error.php");

}

}

}

function dataFilter($data)

{

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

**Signup.php**

<?php

session\_start();

if ($\_SERVER["REQUEST\_METHOD"] == "POST")

{

$name = dataFilter($\_POST['name']);

$mobile = dataFilter($\_POST['mobile']);

$user = dataFilter($\_POST['uname']);

$email = dataFilter($\_POST['email']);

$pass = dataFilter(password\_hash($\_POST['pass'], PASSWORD\_BCRYPT));

$hash = dataFilter( md5( rand(0,1000) ) );

$category = dataFilter($\_POST['category']);

$addr = dataFilter($\_POST['addr']);

$\_SESSION['Email'] = $email;

$\_SESSION['Name'] = $name;

$\_SESSION['Password'] = $pass;

$\_SESSION['Username'] = $user;

$\_SESSION['Mobile'] = $mobile;

$\_SESSION['Category'] = $category;

$\_SESSION['Hash'] = $hash;

$\_SESSION['Addr'] = $addr;

$\_SESSION['Rating'] = 0;

}

require '../db.php';

$length = strlen($mobile);

if($length != 10)

{

$\_SESSION['message'] = "Invalid Mobile Number !!!";

header("location: error.php");

die();

}

if($category == 1)

{

$sql = "SELECT \* FROM farmer WHERE femail='$email'";

$result = mysqli\_query($conn, "SELECT \* FROM farmer WHERE femail='$email'") or die($mysqli->error());

if ($result->num\_rows > 0 )

{

$\_SESSION['message'] = "User with this email already exists!";

//echo $\_SESSION['message'];

header("location: error.php");

}

else

{

$sql = "INSERT INTO farmer (fname, fusername, fpassword, fhash, fmobile, femail, faddress)

VALUES ('$name','$user','$pass','$hash','$mobile','$email','$addr')";

if (mysqli\_query($conn, $sql))

{

$\_SESSION['Active'] = 0;

$\_SESSION['logged\_in'] = true;

$\_SESSION['picStatus'] = 0;

$\_SESSION['picExt'] = png;

$sql = "SELECT \* FROM farmer WHERE fusername='$user'";

$result = mysqli\_query($conn, $sql);

$User = $result->fetch\_assoc();

$\_SESSION['id'] = $User['fid'];

if($\_SESSION['picStatus'] == 0)

{

$\_SESSION['picId'] = 0;

$\_SESSION['picName'] = "profile0.png";

}

else

{

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

$\_SESSION['picName'] = "profile".$\_SESSION['picId'].".".$\_SESSION['picExt'];

}

$\_SESSION['message'] =

"Confirmation link has been sent to $email, please verify

your account by clicking on the link in the message!";

$to = $email;

$subject = "Account Verification ( ArtCircle.com )";

$message\_body = "

Hello '.$user.',

Thank you for signing up!

Please click this link to activate your account:

http://localhost/AgroCulture/Login/verify.php?email=".$email."&hash=".$hash;

//$check = mail( $to, $subject, $message\_body );

header("location: profile.php");

}

else

{

//echo "Error: " . $sql . "<br>" . mysqli\_error($conn);

$\_SESSION['message'] = "Registration failed!";

header("location: error.php");

}

}

}

else

{

$sql = "SELECT \* FROM buyer WHERE bemail='$email'";

$result = mysqli\_query($conn, "SELECT \* FROM buyer WHERE bemail='$email'") or die($mysqli->error());

if ($result->num\_rows > 0 )

{

$\_SESSION['message'] = "User with this email already exists!";

//echo $\_SESSION['message'];

header("location: error.php");

}

else

{

$sql = "INSERT INTO buyer (bname, busername, bpassword, bhash, bmobile, bemail, baddress)

VALUES ('$name','$user','$pass','$hash','$mobile','$email','$addr')";

if (mysqli\_query($conn, $sql))

{

$\_SESSION['Active'] = 0;

$\_SESSION['logged\_in'] = true;

$sql = "SELECT \* FROM buyer WHERE busername='$user'";

$result = mysqli\_query($conn, $sql);

$User = $result->fetch\_assoc();

$\_SESSION['id'] = $User['bid'];

$\_SESSION['message'] =

"Confirmation link has been sent to $email, please verify

your account by clicking on the link in the message!";

$to = $email;

$subject = "Account Verification ( ArtCircle.com )";

$message\_body = "

Hello '.$user.',

Thank you for signing up!

Please click this link to activate your account:

http://localhost/AgroCulture/Login/verify.php?email=".$email."&hash=".$hash;

//$check = mail( $to, $subject, $message\_body );

header("location: profile.php");

}

else

{

//echo "Error: " . $sql . "<br>" . mysqli\_error($conn);

$\_SESSION['message'] = "Registration not successfull!";

header("location: error.php");

}

}

}

function dataFilter($data)

{

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

**Profile.php**

<?php

session\_start();

if ( $\_SESSION['logged\_in'] != 1 )

{

$\_SESSION['message'] = "You must log in before viewing your profile page!";

header("location: error.php");

}

else

{

$email = $\_SESSION['Email'];

$name = $\_SESSION['Name'];

$user = $\_SESSION['Username'];

$mobile = $\_SESSION['Mobile'];

$active = $\_SESSION['Active'];

}

?>

<!DOCTYPE html>

<html >

<head>

<title>AgroCulture</title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

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

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script src="../bootstrap\js\bootstrap.min.js"></script>

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

<meta name="description" content="" />

<meta name="keywords" content="" />

<!--[if lte IE 8]><script src="css/ie/html5shiv.js"></script><![endif]-->

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

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

<script src="../js/skel-layers.min.js"></script>

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

<link rel="stylesheet" href="../css/skel.css" />

<link rel="stylesheet" href="../css/style.css" />

<link rel="stylesheet" href="../css/style-xlarge.css" />

</head>

<body>

<?php

require 'menu.php';

?>

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

<div class="container">

<header class="major">

<h2>Welcome</h2>

</header>

<p>

<?php

if ( isset($\_SESSION['message']) )

{

echo $\_SESSION['message'];

unset( $\_SESSION['message'] );

}

?>

</p>

<?php

if ( !$active )

{

echo

"<div>

Account is not verified! Please confirm your email by clicking

on the email link!

</div>";

}

?>

<h2><?php echo $name; ?></h2>

<p><?= $email ?></p>

<?php if($\_SESSION['Category'] == 1): ?>

<div class="row uniform">

<div class="6u 12u$(xsmall)">

<a href=../profileView.php class="button special">My Profile</a>

</div>

<div class="6u 12u$(xsmall)">

<a href="logout.php" class="button special">LOG OUT</a>

</div>

</div>

<?php else: ?>

<div class="row uniform">

<div class="6u 12u$(xsmall)">

<a href=../market.php class="button special">Digital Market</a>

</div>

<div class="6u 12u$(xsmall)">

<a href="logout.php" class="button special">LOG OUT</a>

</div>

</div>

<?php endif; ?>

</body>

</html>

**Menu.php**

<?php

if(isset($\_SESSION['logged\_in']) AND $\_SESSION['logged\_in'] == 1)

{

$loginProfile = "My Profile: ". $\_SESSION['Username'];

$logo = "glyphicon glyphicon-user";

if($\_SESSION['Category']!= 1)

{

$link = "profile.php";

}

else {

$link = "../profileView.php";

}

}

else

{

$loginProfile = "Login";

$link = "../index.php";

$logo = "glyphicon glyphicon-log-in";

}

?>

<!DOCTYPE html>

<header id="header">

<h1><a href="index.php">AgroCulture</a></h1>

<nav id="nav">

<ul>

<li><a href="../index.php"><span class="glyphicon glyphicon-home"></span> Home</a></li>

<li><a href="../myCart.php"><span class="glyphicon glyphicon-shopping-cart"> MyCart</a></li>

<li><a href="<?= $link; ?>"><span class="<?php echo $logo; ?>"></span><?php echo" ". $loginProfile; ?></a></li>

<li><a href="../market.php"><span class="glyphicon glyphicon-grain"> Digital-Market</a></li>

<li><a href="../blogView.php"><span class="glyphicon glyphicon-comment"> BLOG</a></li>

</ul>

</nav>

</header>

</body>

</html>

**DATABASE**

CREATE TABLE `blogdata` (

`blogId` int(10) NOT NULL,

`blogUser` varchar(256) NOT NULL,

`blogTitle` varchar(256) NOT NULL,

`blogContent` longtext NOT NULL,

`blogTime` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`likes` int(10) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `blogdata`

--

INSERT INTO `blogdata` (`blogId`, `blogUser`, `blogTitle`, `blogContent`, `blogTime`, `likes`) VALUES

(19, 'ThePhenom', 'First Blog', '<p>Its Awesome website<img alt="wink" src="https://cdn.ckeditor.com/4.8.0/full/plugins/smiley/images/wink\_smile.png" style="height:23px; width:23px" title="wink" /></p>\r\n', '2018-02-25 13:09:41', 1);

-- --------------------------------------------------------

--

-- Table structure for table `blogfeedback`

--

CREATE TABLE `blogfeedback` (

`blogId` int(10) NOT NULL,

`comment` varchar(256) NOT NULL,

`commentUser` varchar(256) NOT NULL,

`commentPic` varchar(256) NOT NULL DEFAULT 'profile0.png',

`commentTime` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `blogfeedback`

--

INSERT INTO `blogfeedback` (`blogId`, `comment`, `commentUser`, `commentPic`, `commentTime`) VALUES

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

-- --------------------------------------------------------

--

-- Table structure for table `buyer`

--

CREATE TABLE `buyer` (

`bid` int(100) NOT NULL,

`bname` varchar(100) NOT NULL,

`busername` varchar(100) NOT NULL,

`bpassword` varchar(100) NOT NULL,

`bhash` varchar(100) NOT NULL,

`bemail` varchar(100) NOT NULL,

`bmobile` varchar(100) NOT NULL,

`baddress` text NOT NULL,

`bactive` int(100) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `farmer`

--

CREATE TABLE `farmer` (

`fid` int(255) NOT NULL,

`fname` varchar(255) NOT NULL,

`fusername` varchar(255) NOT NULL,

`fpassword` varchar(255) NOT NULL,

`fhash` varchar(255) NOT NULL,

`femail` varchar(255) NOT NULL,

`fmobile` varchar(255) NOT NULL,

`faddress` text NOT NULL,

`factive` int(255) NOT NULL DEFAULT '0',

`frating` int(11) NOT NULL DEFAULT '0',

`picExt` varchar(255) NOT NULL DEFAULT 'png',

`picStatus` int(10) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `farmer`

--

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

(3, 'Kaivalya Hemant Mendki', 'ThePhenom', '$2y$10$22ezmzHRa9c5ycHmVm5RpOnlT4LwFaDZar1XhmLRJQKGrcVRhPgti', '61b4a64be663682e8cb037d9719ad8cd', 'kmendki98@gmail.com', '8600611198', 'abcde', 0, 0, 'png', 0);

-- --------------------------------------------------------

--

-- Table structure for table `fproduct`

--

CREATE TABLE `fproduct` (

`fid` int(255) NOT NULL,

`pid` int(255) NOT NULL,

`product` varchar(255) NOT NULL,

`pcat` varchar(255) NOT NULL,

`pinfo` varchar(255) NOT NULL,

`price` float NOT NULL,

`pimage` varchar(255) NOT NULL DEFAULT 'blank.png',

`picStatus` int(10) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `fproduct`

--

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);

-- --------------------------------------------------------

--

-- Table structure for table `likedata`

--

CREATE TABLE `likedata` (

`blogId` int(10) NOT NULL,

`blogUserId` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `likedata`

--

INSERT INTO `likedata` (`blogId`, `blogUserId`) VALUES

(19, 3);

-- --------------------------------------------------------

--

-- Table structure for table `mycart`

--

CREATE TABLE `mycart` (

`bid` int(10) NOT NULL,

`pid` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `mycart`

--

INSERT INTO `mycart` (`bid`, `pid`) VALUES

(3, 27),

(3, 30);

-- --------------------------------------------------------

--

-- Table structure for table `review`

--

CREATE TABLE `review` (

`pid` int(10) NOT NULL,

`name` varchar(255) NOT NULL,

`rating` int(10) NOT NULL,

`comment` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `transaction`

--

CREATE TABLE `transaction` (

`tid` int(10) NOT NULL,

`bid` int(10) NOT NULL,

`pid` int(10) NOT NULL,

`name` varchar(255) NOT NULL,

`city` varchar(255) NOT NULL,

`mobile` varchar(255) NOT NULL,

`email` varchar(255) NOT NULL,

`pincode` varchar(255) NOT NULL,

`addr` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `transaction`

--

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');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `blogdata`

--

ALTER TABLE `blogdata`

ADD PRIMARY KEY (`blogId`);

--

-- Indexes for table `buyer`

--

ALTER TABLE `buyer`

ADD PRIMARY KEY (`bid`),

ADD UNIQUE KEY `bid` (`bid`);

--

-- Indexes for table `farmer`

--

ALTER TABLE `farmer`

ADD PRIMARY KEY (`fid`),

ADD UNIQUE KEY `fid` (`fid`);

--

-- Indexes for table `fproduct`

--

ALTER TABLE `fproduct`

ADD PRIMARY KEY (`pid`);

--

-- Indexes for table `likedata`

--

ALTER TABLE `likedata`

ADD KEY `blogId` (`blogId`),

ADD KEY `blogUserId` (`blogUserId`);

--

-- Indexes for table `transaction`

--

ALTER TABLE `transaction`

ADD PRIMARY KEY (`tid`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `blogdata`

--

ALTER TABLE `blogdata`

MODIFY `blogId` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=20;

--

-- AUTO\_INCREMENT for table `buyer`

--

ALTER TABLE `buyer`

MODIFY `bid` int(100) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `farmer`

--

ALTER TABLE `farmer`

MODIFY `fid` int(255) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- AUTO\_INCREMENT for table `fproduct`

--

ALTER TABLE `fproduct`

MODIFY `pid` int(255) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=31;

--

-- AUTO\_INCREMENT for table `transaction`

--

ALTER TABLE `transaction`

MODIFY `tid` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `buyer`

--

ALTER TABLE `buyer`

ADD CONSTRAINT `buyer\_ibfk\_1` FOREIGN KEY (`bid`) REFERENCES `farmer` (`fid`);

--

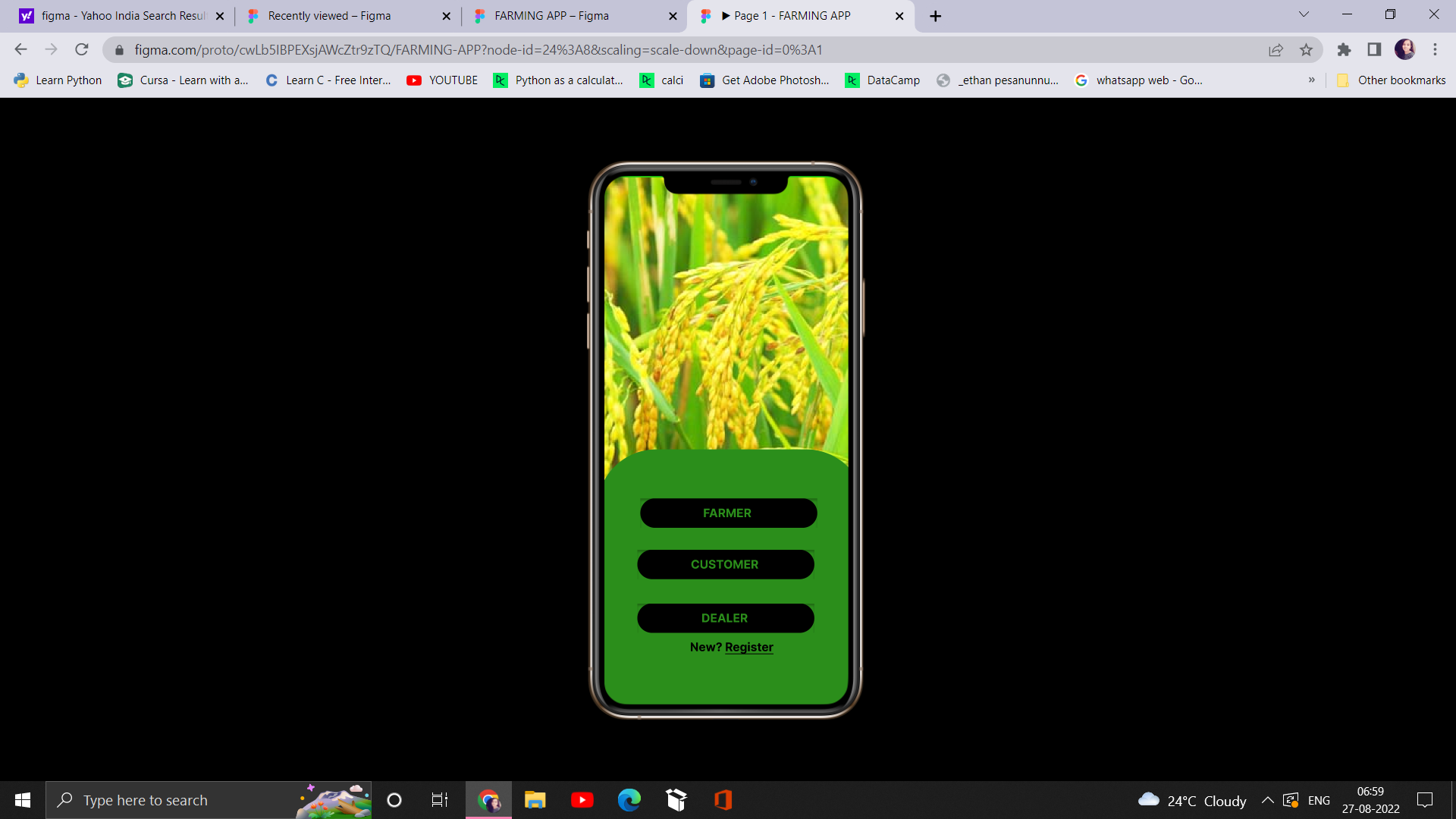
-- Constraints for table `likedata`

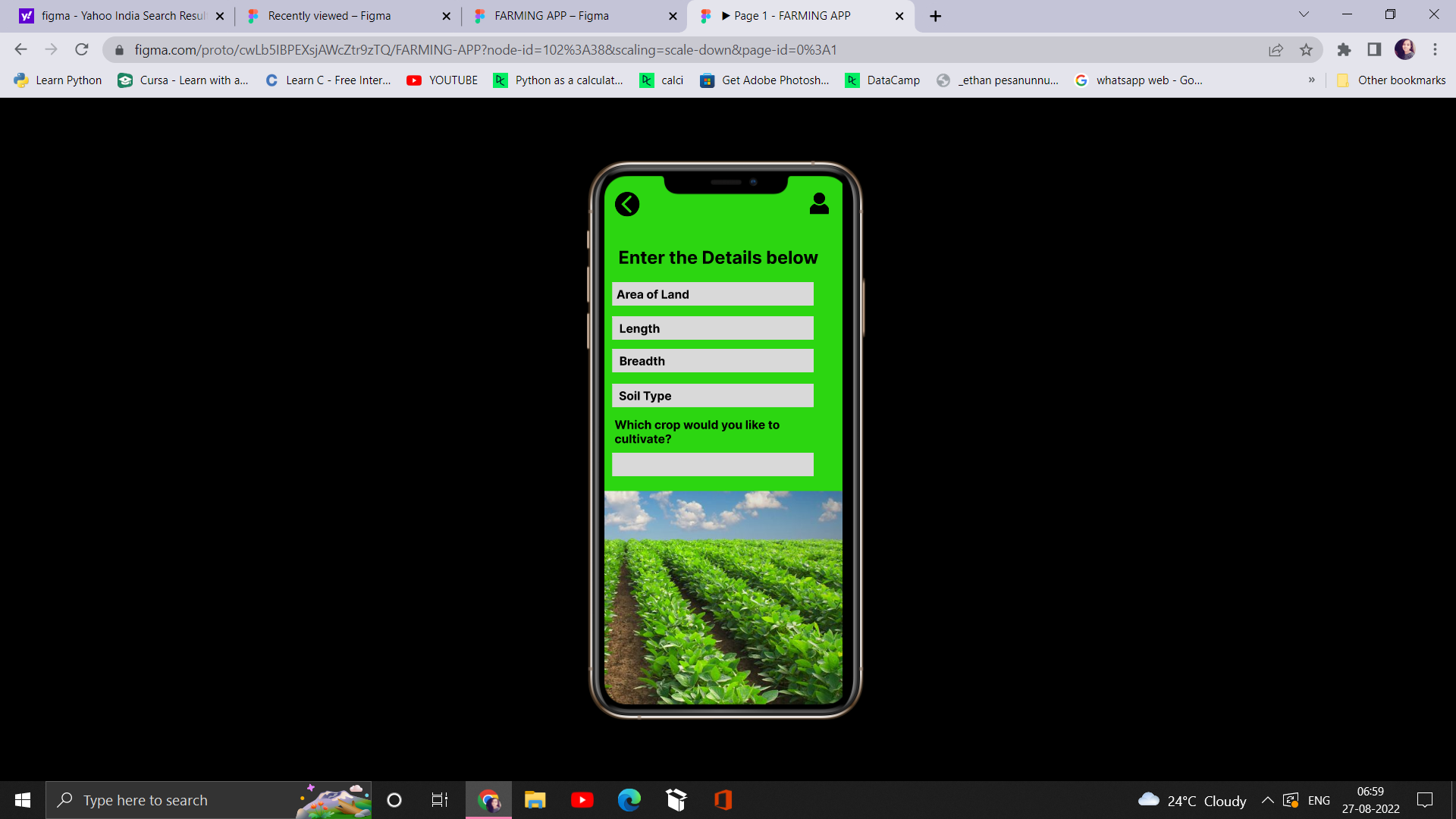
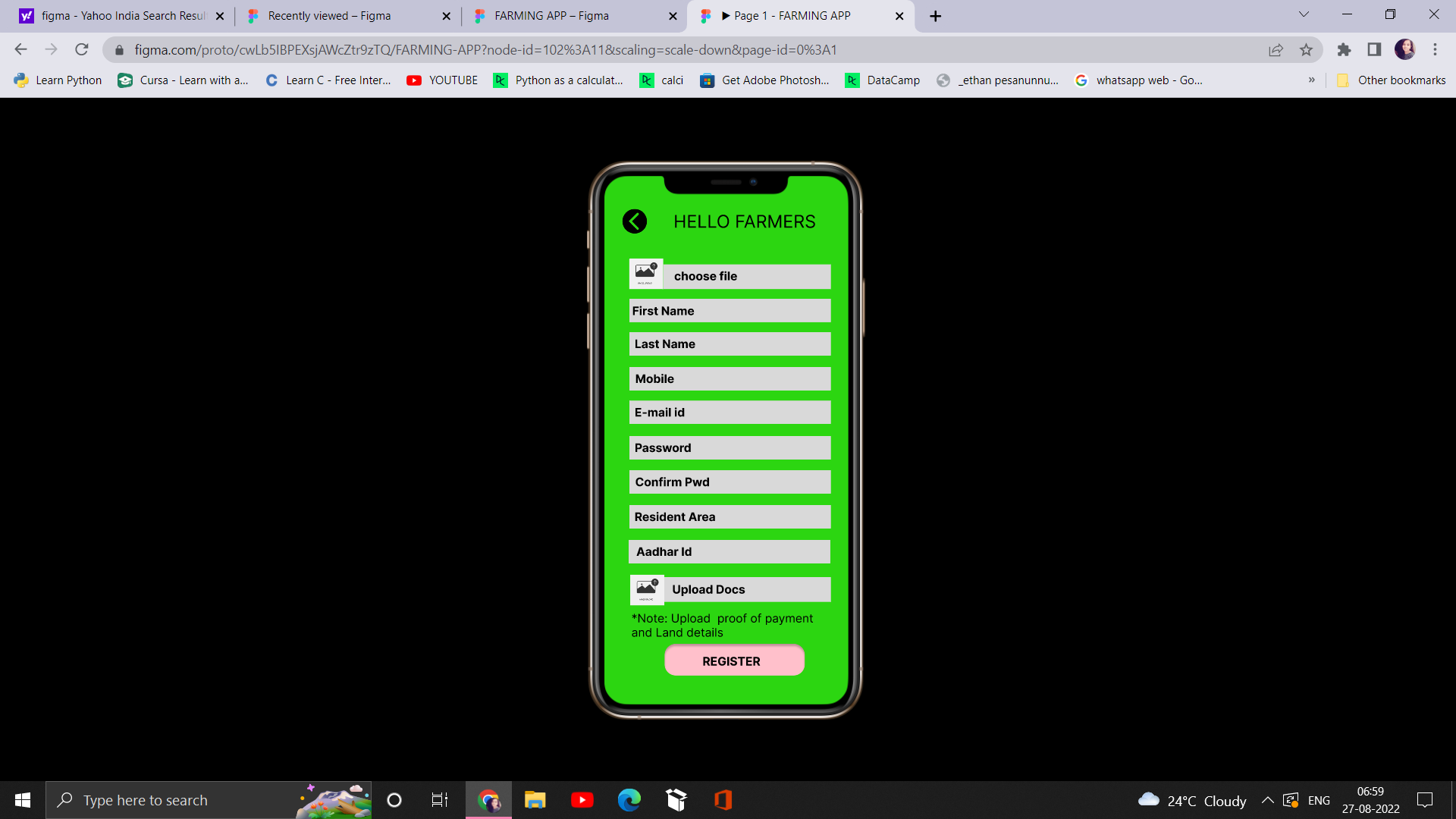
--

ALTER TABLE `likedata`

ADD CONSTRAINT `likedata\_ibfk\_1` FOREIGN KEY (`blogId`) REFERENCES `blogdata` (`blogId`);

**12. SCREENSHOTS**

****



**BIBLIOGRAPHY**

**13. BIBLIOGRAPHY**

**Books Referred**

* Android Programming:The Big Nerd Ranch Guide by Bill Philips& Brain Hardy
* Android Design Patterns:Interaction Design Solutions for Developers by Greg Nudelman
* Professional Android Application development by Reto Meier.
* ASP.NET Professional by Wrox

**Websites Referred**

* <https://developer.android.com>
* <https://www.tutorialspoint.com>
* <http://www.vogella.com>
* http:// [www.c-sharpcorner.com](http://www.c-sharpcorner.com)
* http:// [www.asp.net](http://www.asp.net)
* <https://developer.android.com/guide>  
  <https://developer.android.com/reference/java/lang/ref/Reference>
* <https://www.quora.com/What-are-the-Android-developer-reference-sites>
* <https://www.tutorialspoint.com/android/>
* https://www.journaldev.com/android