

INTRODUCTION

CHAPTER 1

1.1 ABOUT THE ORGANIZATION

K.S.Rangasamy college of Arts and Science (Autonomous) known among the reputed KSR group of educational Institutions. The foundation stone for college was laid by the K.S.Rangasamy Educational Charitable Trust in 1995. The fender and the president of the Trust is Lion Dr. K.S.Rangasamy MJF a visionary in the felt cof education. The sprawling campus of the college is found in the system atmosphere of shady palm group of about 350 acres, It is situated on the Tiruchengode –Erode state Highway 12KM from Erode and 8 KM from Tiruchengode.

From 1995-1998 the college was affiliated to University of Madras. Later staring from the academic year 1995-1999 it is affiliated to Periyar University, Salem, Tiruchengode and neighbouring places are still in developing phase and abound with rural population Having this in mind with the noble vision of opening new avenues of education to disraral mass of the college was established world and come out in flying colors in academic excellence.

The college is keen in recruit and retain faculty members of greater efficiency and rich experience. The undergraduate, post graduate and research programme are run successfully with their noble efforts and seasoned guidance. The college became an ISO 9001 -2000(E) certified institutions during 2005, accredited by NAAC from 2010 onwards and included under 12(8) of UGC Act of 1956. University grants commission. New Delhi has conferred autonomous status from the academic year 2009-2010.

As advocated by NAAC, a division of Internal Quality Assurance Cell (IQAC) has been constituted to sustain quality through work plan, specify the checks and balances to evaluate the degree of quality and to ensure that quality initiatives are taken effectively and efficiently with high standard, which takes care of the quality management of the quality.

1.2 ABSTRACT

Our objective is to introduce an agricultural Crop information system on the Internet so that it will eventually allow potential users to query and obtain the desired information.

The main pure of the project is the availability of farming information to various users through online .The user can login into their website and view the agriculture information. In the existing system, the customer need to meet the appropriate department for recognize their problem. The user does not know their new offers and crop details. They failed to get the information about agriculture system. After analyzing the limitation of the existing system, the customer need not go to agriculture department to find solution.

They can know the information in anywhere anytime and anyplace by using proposed system. They can proper information about the agriculture frequently.

The data of this system is to be stored using a central database and maintained by the admin . System is to be portable as the computer system that maintains this data may change from time to time.

1.3 STAKEHOLDERS

- Customer
- Administrator

SYSTEM STUDY

CHAPTER 2

2.1 EXISTING SYSTEM

In Existing system Customers cannot directly contact agriculture officers by searching online. Customers may not submit their grievances online. Customers cannot able get notification of any seeds information and its description properly.

2.1.1 Drawbacks of Existing System

- Verification of the users cannot be done online.
- They have to first get themselves verified by visiting before using the system.
- It requires a large database and memory.
- No proper description about seeds.

2.2 PROPOSED SYSTEM

In proposed work Customers can directly contact suppliers by searching online. Customers may submit their grievances online. Customers get notification of any new offers/scheme through their private mail. They can able to request fertilizer and seeds through sms. The database provides tools for data entry, data viewing, data analysis and report generation.

This is an on demand for more environmentally friendly farming, especially the reduced use of interaction between agri officers and customers on that region. In this projects Administrator is able to add, delete and update employee and category and view reports. Customers are able to update change password.

2.2.1 Advantages of proposed system

- Separate login areas with appropriated functionality for customers, administrators and dealers/ retailers.
- A separate page where only customers can post complaints and only assigned administrators can read and edit this page.
- Pages where dealers and retailers may post their ads and notifications.
- Customers are notified of these notifications via sms whenever new ads are published.
- Can be over for multiple villages to communicate and deal with each other.

2.2.2 Project Description

E-Farming Management System is the website application where customers can buy Seeds Fertilizers and Admin can upload products for selling. In this E-Farming Management System we also provide the servicers like giving training for the farmers to perform agriculture in a organic way. We also provide land quality inspection .The system is designed to allow customers/users by the products that are listed or book slot for the Training or Land Inspection

Our objective is to introduce an agricultural Crop information system on the Internet so that it will eventually allow potential users to query and obtain the desired information. The data of this system is to be stored using a central database and maintained by the admin. System is to be portable as the computer system that maintains this data may change from time to time.

2.2.3 Modules and Description

- Login module
- Registration module
- Contracts module
- Seed module
- Fertilizer module
- training module
- Inspection module

Module Description

- **Login module**

In the login form, customer enters value for Username and Password. The login button is used to go next for as based on the Username and Password.

- **Registration module**

The registration form is used to maintain the details about customer. This form is used to store the details like User-Id, Password, DOB, Gender, Email-id, Address, City and Phone No, Create button is used to registration details.

- **Contracts module**

The contracts form is used to display the contracts .This form is used to store the details like contracted, landsize(Ha),land location, previous Seed planted ,distance from other Seed, land ownership.

- **Seed module**

The seed details form is used to display the seed. This form is used to store the details like seed, farmed, land size, variety, hybrid seed, amount, status, location.

- **Fertilizer module**

The fertilizer details form is used to display the fertilizer. This form is used to store the details like fertilized, fertilizer, purpose, land size, amount (kg),status.

- **Training module**

The training details form is used to display the new trainings. This form is used to store the details like farmer, where, from, to.

- **Inspection module**

The inspection details form is used to display the inspection. This form is used to store the details like inspection-id, farmer-id, attend, ondate, inpection, contact number.

The following reports are generated in the system.

- Seeds sales report
- Fertilizers details
- Contracts details
- Trainings details
- Inspection details

2.3 REQUIREMENT PHASE

2.3.1 Hardware Requirements

| | |
|--------------------|----------------------------|
| Processor | : I3 PROCESSOR |
| Hard Disk Capacity | : 500GB |
| RAM | : 4 GB SD |
| Monitor | : 17 ^{inch} Color |
| Keyboard | : 101/102 Standard Keys |
| Mouse | : Optical Mouse |

2.3.2 Software Requirements

| | |
|--------------------|------------|
| Front-End | : PHP |
| Designing Language | : HTML |
| Coding language | : PHP |
| Back-End | :My-SQL |
| Operating system | : Windows8 |

2.3.3 About the Software

PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming languages While PHP originally stood for **Personal Home Page**, it now stands for **PHP : Hypertext Preprocessor**, a recursive acronym. PHP is open source No license fee is required and can be downloaded free of cost

PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page .PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data.PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP. The most common delimiters are <? Php to open and ?> to close PHP sections.

While short delimiters are used, they make script files less potable as support for them can be disabled in the PHP configuration, and they are therefore discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

FEATURES OF PHP

- PHP language has support features of other languages like c, Perl and etc. It also has some unique features of its own. Some of them are listed below in this article.
- In PHP there is no need to specify data type for variable declaration. Rather, it can be determined at the time of execution depends on the value of the variable. So that, PHP is called as loosely typed language.
- PHP provides cross platform compatibility, unlike some other server side scripting language.
- PHP has set of pre defined variables called superglobals which will be start by `_`. Some of the examples are, `$_GET`, `$_POST`, `$_COOKIE`, `$_SESSION`, `$_SERVER` and etc. So, any variable accept super global, that are start with `_` will cause error.

- PHP programming structure includes variables that the name of the variable can be change dynamically..
- Predefined error reporting constants are available to generate a warning or error notice.
- PHP supports extended regular expression that leads extensive pattern matching with remarkable speed.
- Since PHP is a single inheritance language, the parent class methods can be derived by only one directly inherited sub class. But, the implementation of traits concept, reduce the gap over this limitation and allow to reuse required method in several classes.

BACK END

MYSQL

“My S-Q-L” officially, but also called “My sequel” is the world’s most used open source relational database management system(RDBMS) that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language.

FEATURES OF MYSQL

Open-Source Nature

MYSQL is an open storage database engine. MYSQL integrates with different types of programming languages.

Licensing

The open-source system works under the GNU(General Public License) and the developers can enjoy the opportunity to use it at no cost. However, if it is sold as a proprietary product , a commercial license has to be purchased.

Performance

MYSQL is considered to be the clear leader and tends to perform better on UNIX and Linux like systems. Since MYSQL is stable in nature, most of the Internet powerhouses use it as their back-end database.

Internet Integration

The SQL server database engine includes integrated xml support. It also has the scalability availability and security features required to operate as the data storage component of the largest website.

The SQL server 2000 programming model is integrated with the window DNA architecture for developing web applications and SQL server supports features such as English Query and the Microsoft search service incorporate user friendly queries and powerful search capabilities in web applications

Scalability And Availability

The same database engine can be used across platform ranging from laptop computer running Microsoft windows through large, microprocessor servers running Microsoft windows Data center Edition

Enterprise Level Database Features

The SQL server relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database

Data Ware Housing

SQL Server includes tools for extracting and analyzing summary data for online analytical processing. SQL Server also includes tools for visually designing databases and analyzing data using English based questions.

ADVANTAGES OF MYSQL

- Its unique storage engine architecture, MYSQL performance is very high.
- Supports large number of embedded applications which makes MYSQL very flexible.
- Use of triggers, stored procedures and views which allows the developer to give a higher productivity.

Allows transactions to be rolled back, committed and crash recovery is faster.

2.3.4 Functional Requirements

- **User Registration and Authentication**

Users should be able to create accounts and authenticate themselves to access the system.

User roles and permissions should be defined to control access to different functionalities.

There must be separate login for the farmers and the admin

- **Seeds and Fertilizer Management**

Adding new seed and maintaining the seeds stock.

Maintaining the purchase of the seeds.

Adding new Fertilizer and maintaining the Fertilizer stock.

Maintaining the purchase of the seeds.

- **Planning and Scheduling**

Planning the dates for the training and inspection program

Accepting request for the training and inspection program

Maintain the slots for the training and inspection program

- **Communication and Collaboration:**

Features to facilitate communication and collaboration between farmers, suppliers, and other stakeholders.

Ability to share information, ask questions, seek advice, and engage in discussions

2.3.5 Non-functional Requirements

- **Usability**

The system should be intuitive and easy to use, with a user-friendly interface. It should require minimal training for farmers to understand and navigate through different features

- **Security**

The system should have robust security measures to protect sensitive data and ensure secure access. This includes user authentication, data encryption, secure communication protocols, and access control mechanisms.

- **Maintainability**

The system should be easy to maintain and update. It should have well-structured code, documentation, and modular architecture to facilitate future enhancements or bug fixes

- **Performance Efficiency**

The system should optimize resource utilization to ensure efficient performance. This includes minimizing system resource consumption (e.g., CPU, memory, network bandwidth) and optimizing database queries and data retrieval

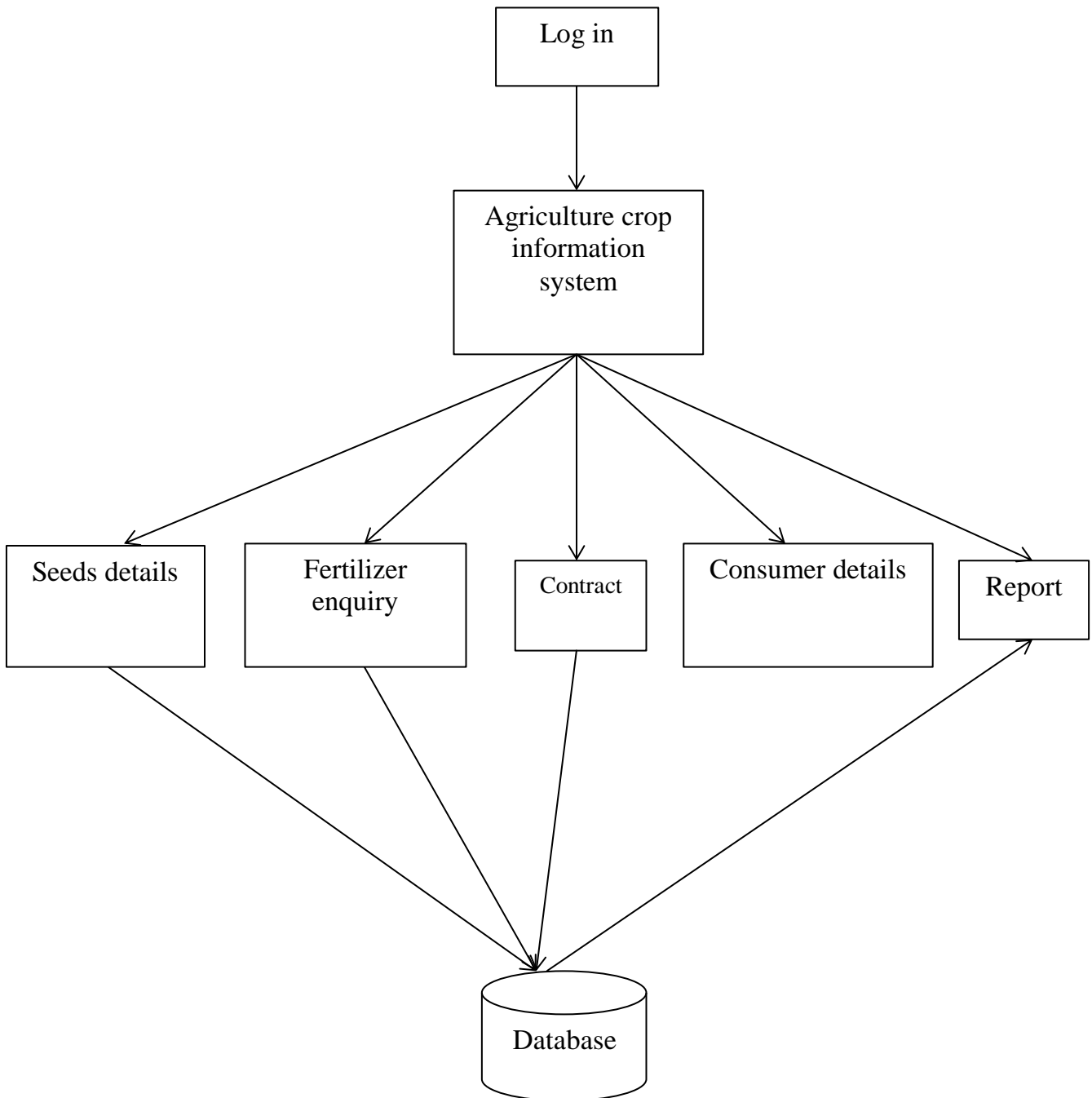
2.3.5 Data Collection

- Sellers name and details
- Customer name and details
- Name and Details of Seed and Fertilisers
- Address of the land and mobile number of the owner for inspection and training purpose

SYSTEM DESIGN

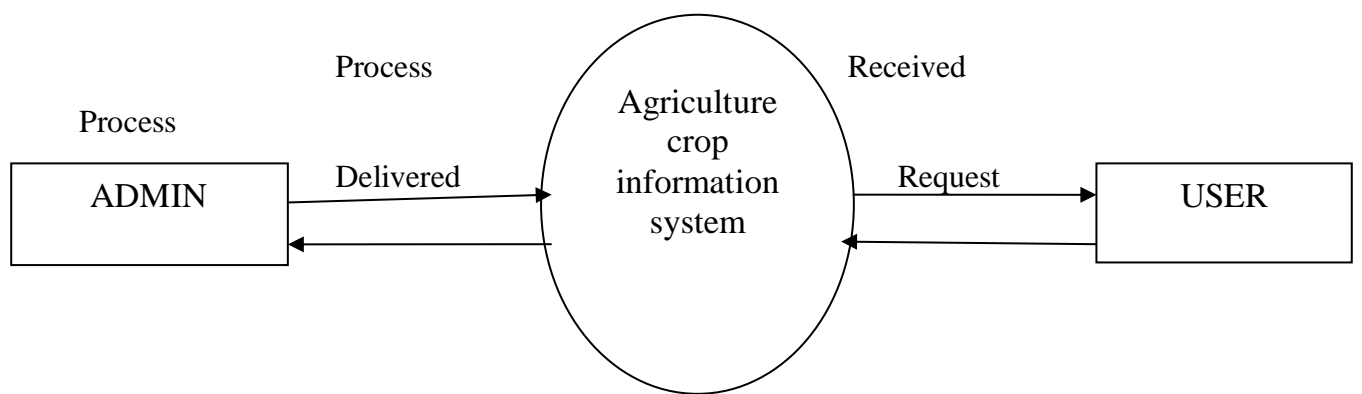
CHAPTER 3

3.1 System Flow Diagram

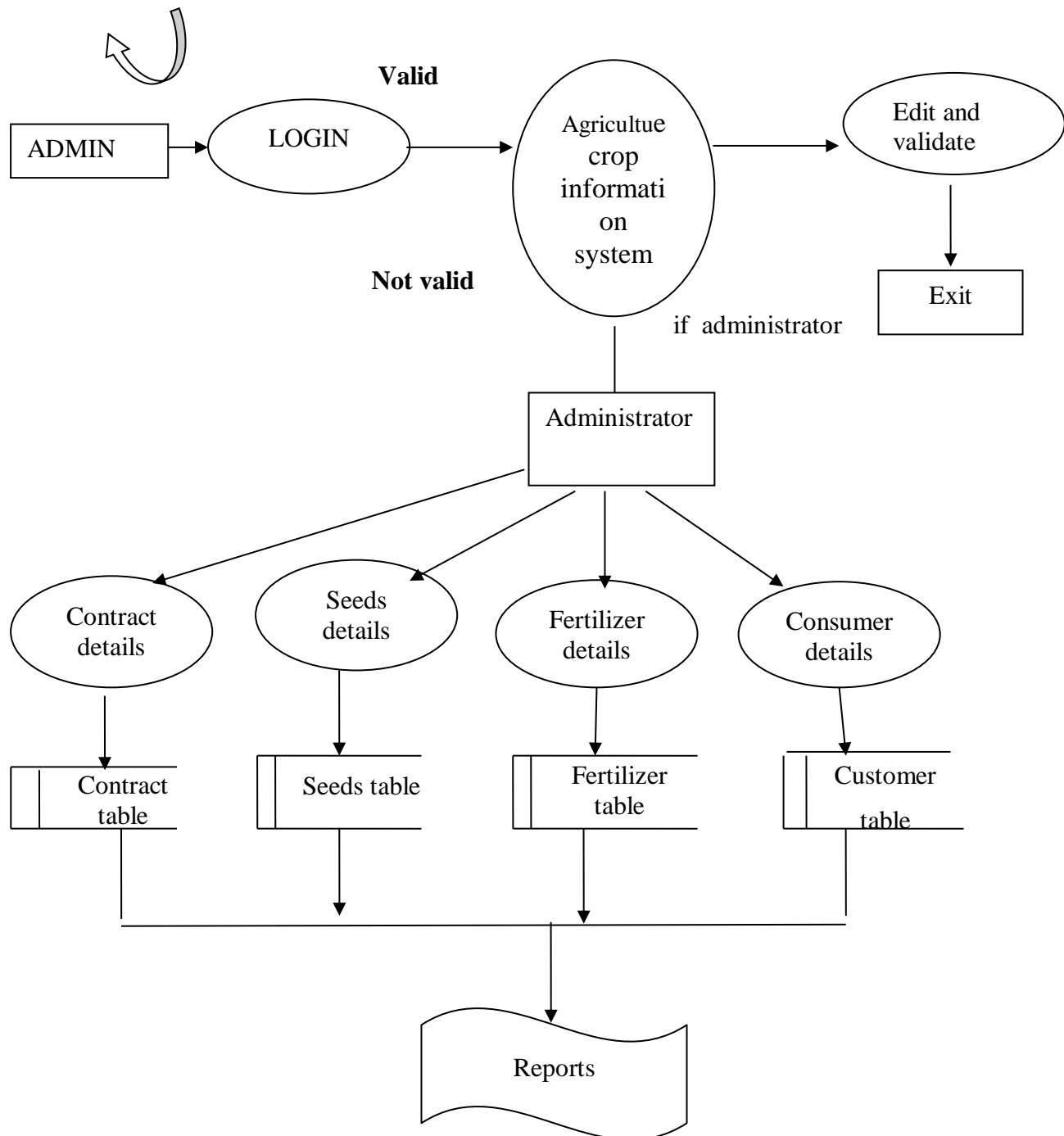


3.2 Data Flow Diagram

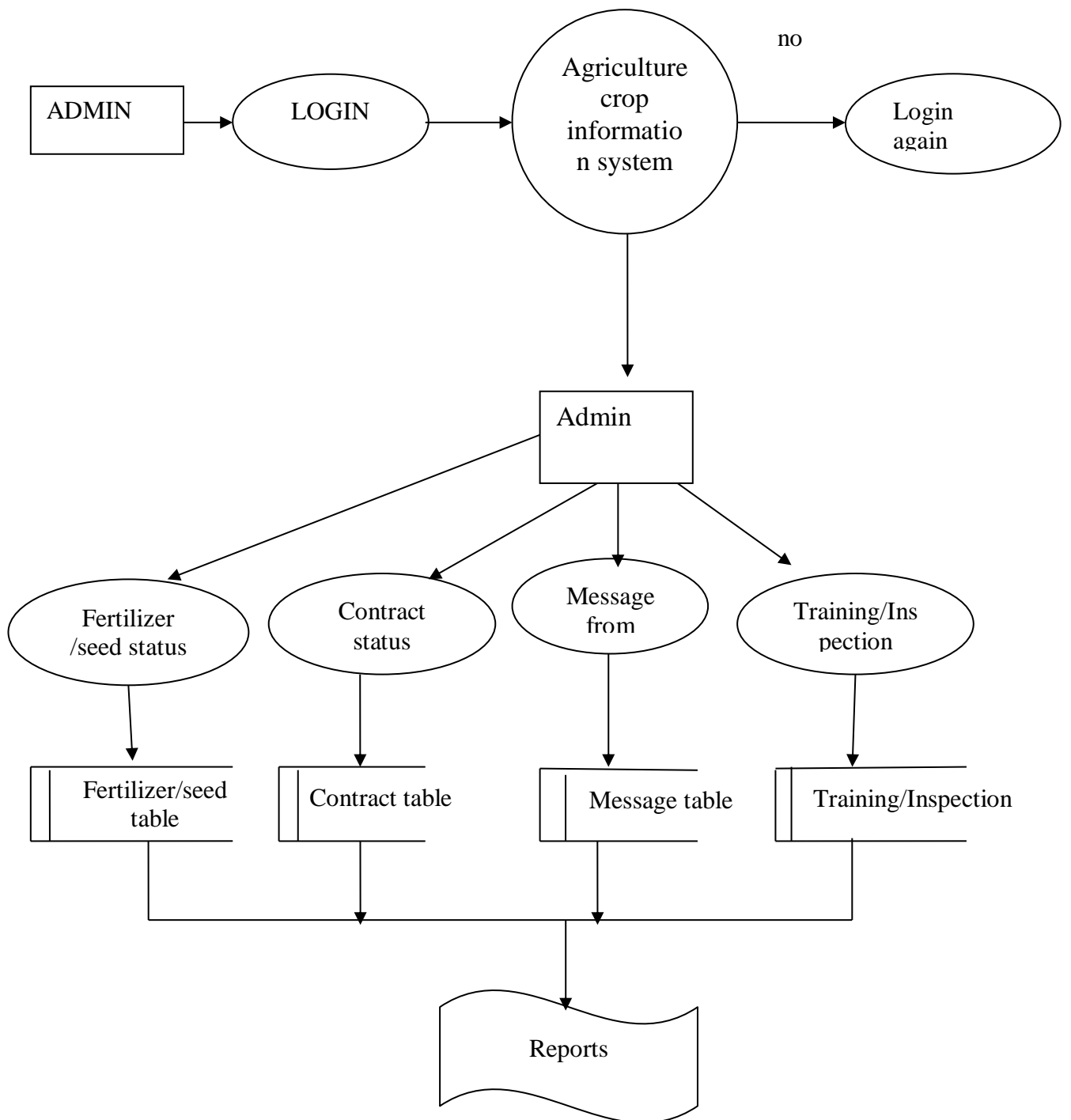
LEVEL 0



LEVEL 1



LEVEL 2



3.3 Database Design

Database design deals with the table structure and organization. The purpose of the database is to present easy information for the user. The general theme behind is to handle the information as an integrated one. The database system is basically computerized record maintain the information available demand.

Database is a collection of interrelated data stored in a format that the user can access the information easily and quickly while accessing the database. Redundancy should be avoided to serve many users who use the database efficiently and also to increase the system performance to attain all these objectives the data should be stored only once.

3.4 Table Design

TABLE : Admin

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|-------------|
| 1 | id | Int | 10 | primary key | id |
| 2 | username | Varchar | 25 | Not Null | User name |
| 3 | password | Varchar | 25 | Not Null | password |

TABLE : book fertilizer

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|--------------|
| 1 | id | Int | 10 | primary key | id |
| 2 | fertilizer | Varchar | 25 | Not Null | fertilizer |
| 3 | pdate | Date | - | Not Null | Packing date |
| 4 | kg | Int | 25 | Not Null | Kilo gram |
| 5 | pkg | Int | 25 | Not Null | package |
| 6 | total | Int | 25 | Not Null | total |
| 7 | username | Varchar | 25 | Not Null | User name |
| 8 | status | Varchar | 25 | Not Null | status |

TABLE : bookseed

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|--------------|
| 1 | id | int | 10 | primary key | id |
| 2 | seed | varchar | 25 | Not Null | seed |
| 3 | pdate | date | - | Not Null | Packing date |
| 4 | kg | int | 25 | Not Null | Kilo gram |
| 5 | pkg | int | 25 | Not Null | package |
| 6 | total | int | 25 | Not Null | total |
| 7 | username | varchar | 25 | Not Null | User name |
| 8 | status | varchar | 25 | Not Null | status |

TABLE : fertilizer

PRIMARY KEY: id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|---------------|
| 1 | id | int | 10 | primary key | id |
| 2 | sname | varchar | 25 | Not Null | Supplier name |
| 3 | pdate | date | - | Not Null | Packing date |
| 4 | fertilizer | varchar | 25 | Not Null | fertilizer |
| 5 | kg | int | 25 | Not Null | Kilo gram |
| 6 | pkg | int | 25 | Not Null | package |
| 7 | total | int | 25 | Not Null | total |

TABLE : inspection

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|-----------------|
| 1 | id | int | 10 | primary key | id |
| 2 | idate | date | - | Not Null | Inspection date |
| 3 | venue | varchar | 25 | Not Null | venue |
| 4 | ifor | varchar | 25 | Not Null | Inspection for |

TABLE : offer

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|-------------|-----------|------|-------------|------------------|
| 1 | id | int | 10 | primary key | id |
| 2 | oname | varchar | 25 | Not Null | Offer name |
| 3 | ofrom | date | - | Not Null | Offer from |
| 4 | oto | date | - | Not Null | Offer to |
| 5 | opercentage | varchar | 25 | Not Null | Offer percentage |

TABLE : seeds

PRIMARY KEY: id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|--------------|
| 1 | id | int | 10 | primary key | id |
| 2 | sname | varchar | 25 | Not Null | Seed name |
| 3 | pdate | date | - | Not Null | Packing date |
| 4 | seed | varchar | 25 | Not Null | seed |
| 5 | kg | int | 25 | Not Null | Kilo Gram |
| 6 | pkg | int | 25 | Not Null | package |
| 7 | total | int | 25 | Not Null | total |

TABLE : supplier

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|---------------|
| 1 | id | int | 10 | primary key | id |
| 2 | sname | varchar | 25 | Not Null | Supplier name |
| 3 | mobile | bigint | 10 | Not Null | mobile |
| 4 | email | varchar | 25 | Not Null | email |
| 5 | address | varchar | 25 | Not Null | address |

TABLE : training

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|---------------|
| 1 | id | int | 10 | primary key | id |
| 2 | tdate | varchar | 25 | Not Null | Training date |
| 3 | venue | bigint | 10 | Not Null | venue |
| 4 | tfor | varchar | 25 | Not Null | Training for |

TABLE : user

PRIMARY KEY : id

| SNO | FIELD NAME | DATA TYPE | SIZE | CONSTRAINT | DESCRIPTION |
|-----|------------|-----------|------|-------------|-------------|
| 1 | id | int | 10 | primary key | id |
| 2 | name | varchar | 25 | Not Null | name |
| 3 | mobile | bigint | 10 | Not Null | mobile |
| 4 | email | varchar | 25 | Not Null | email |
| 5 | address | varchar | 100 | Not Null | address |
| 6 | username | varchar | 25 | Not Null | username |
| 7 | password | varchar | 25 | Not Null | password |

3.5 Input Design

Input design is a part of overall system design, which requires careful attention. Input of data as designed as user friendly and easier. Input design is a process of converting the user-oriented description of the input to the computer based information system into programmer-oriented specification. The objectives of the input design are to create an input layout that is easy to follow and prevent operator errors.

- **Login details**

In the login form, customer enters value for Username and Password. The login button is used to go next for as based on the Username and Password.

- **Registration details**

The registration form is used to maintain the details about customer. This form is used to store the details like UserId, Password, DOB, to registration details.

- **Contracts details**

The contracts form is used to display the contracts .This form is used to store the details like contracted, land size (Ha),land location, previous crop planted ,distance from other crop, land ownership.

- **Seed details**

The seed details form is used to display the seed. This form is used to store the details like seedid, customerid, land size, variety, hybrid seed, amount, status, location.

- **Fertilizer details**

The fertilizer details form is used to display the fertilizer. This form is used to store the details like fertilizerid, fertilizer, purpose, land size, amount (kg), and status.

- **Trainings details**

The training details form is used to display the new trainings. This form is used to store the details like customer, where, from, to.

- **Inspection details**

The inspection details form is used to display the inspection. This form is used to store the details like inspectionid, customerid, atten, ondate, inpection, contact number.

3.6 Output Design

Computer output is the most important and direct source of information to the user. Outputs from computers are required primarily to communicate the results of processing to the customers. They also used to provide permanent copy of these results for late reference. Thus it is designed in user friendly way to avoid user burden.

The output design refers to the results and information that are generated by the system for many end users. To produce the outputs which are displayed in a screen. Efficient and intelligent output design improves the system relationships with the user and help in decision making. The output of the system is in the form of reports.

The following reports are generated in the system.

- Seeds details
- Fertilizers details
- Contracts details
- Trainings details
- Inspection details

- **Seed details**

A seed details is used to generate the details about the seed which are all having entered in Online farming contract system

- **.Fertilizer details**

A fertilizer details is used to generate the details about the fertilizer which are all having entered in Online farming contract system.

- **Contracts details**

A contracts detail is used to generate the details about the contracts which are all having entered in Online farming contract system.

- **Training details**

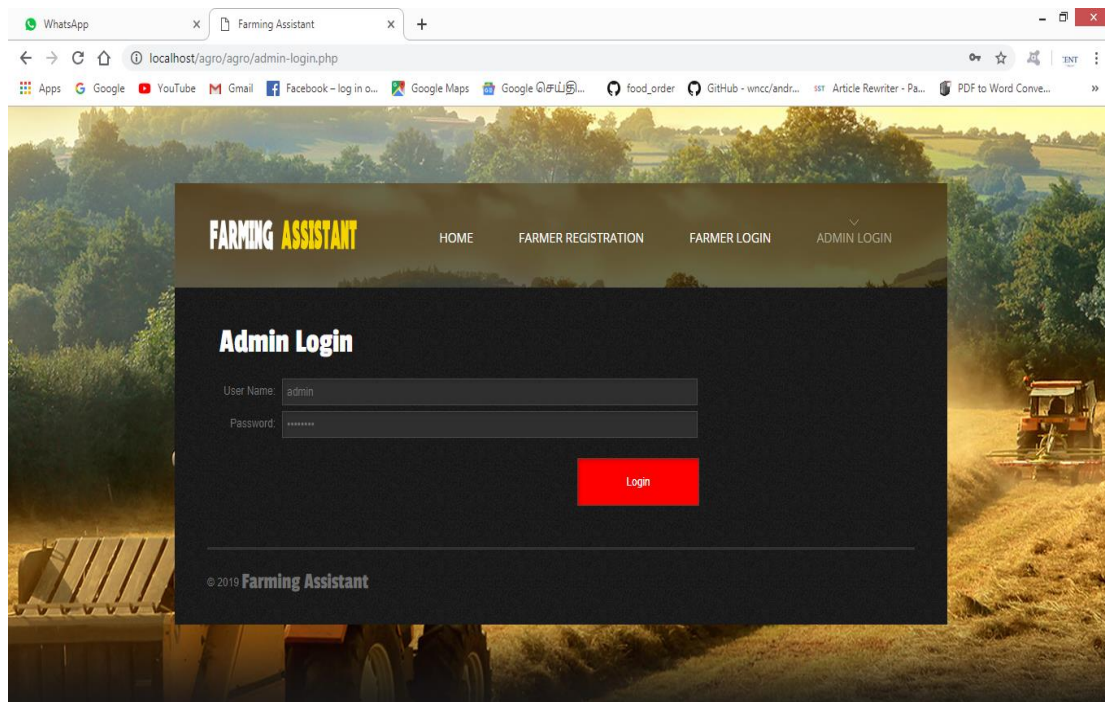
A training detail is used to generate the details about the tranings which are all having entered in Online farming contract system.

- **Inspection details**

An inspection details is used to generate the details about the inspection which are all having entered in Online farming contract system.

3.7 Forms Designs

ADMIN SIDE FORMS:



The image shows a web browser window displaying the 'Farming Assistant' Admin Login form. The browser's address bar shows the URL 'localhost/agro/agro/admin-login.php'. The page has a dark background with a semi-transparent login form overlay. The form includes a navigation bar with links: HOME, FARMER REGISTRATION, FARMER LOGIN, and ADMIN LOGIN. The main heading is 'Admin Login'. Below it, there are input fields for 'User Name' (containing 'admin') and 'Password' (masked with dots). A red 'Login' button is positioned to the right of the password field. At the bottom left of the form, it says '© 2019 Farming Assistant'. The background of the page features a blurred image of a tractor in a field.

Fig: 3.7.1 Admins Login Form

FARMING ASSISTANT

HOME ADD PURCHASE BOOKING DETAILS REPORTS LOGOUT

Add Supplier

Supplier Name: ramu and co

Mobile: 9898453321

E-Mail: ramu@gmail.com

Address: 334 periyar nagar erode-3

Add

© 2019 Farming Assistant

Fig.3.7.2 Supplier Adding Form

FARMING ASSISTANT

HOME ADD PURCHASE BOOKING DETAILS REPORTS LOGOUT

Add Training Program

Training Date: 01-Apr-2019

Training for: organic feed assignment

Venue: erode-parimalam mahal

Add

© 2019 Farming Assistant

Fig3.7. Used For Add Training Slots

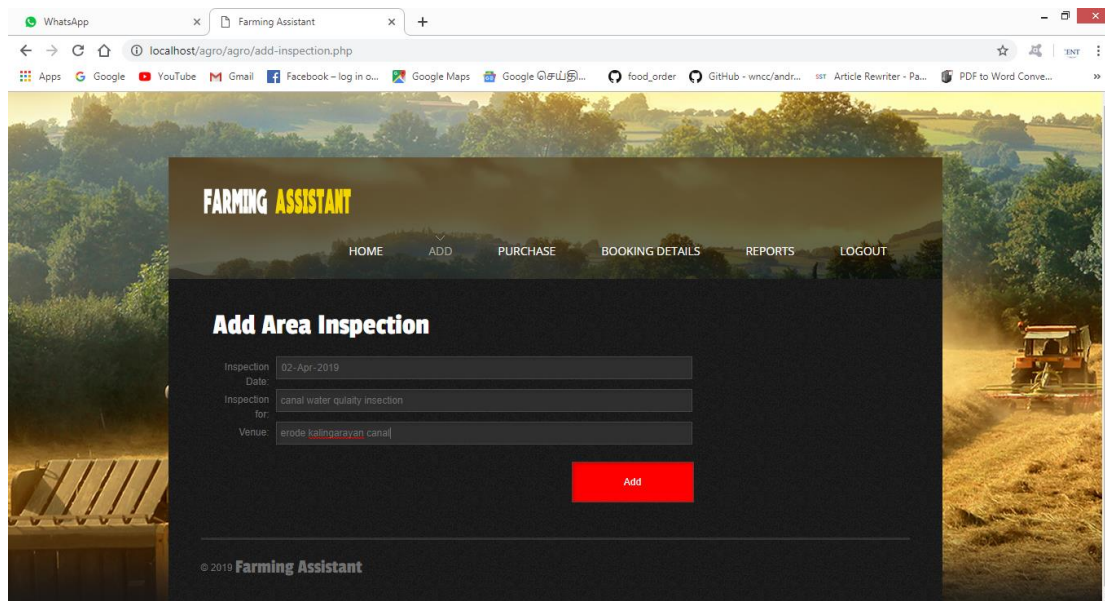


Fig:3.7.4 Used To Add The Slots For Area Inspection

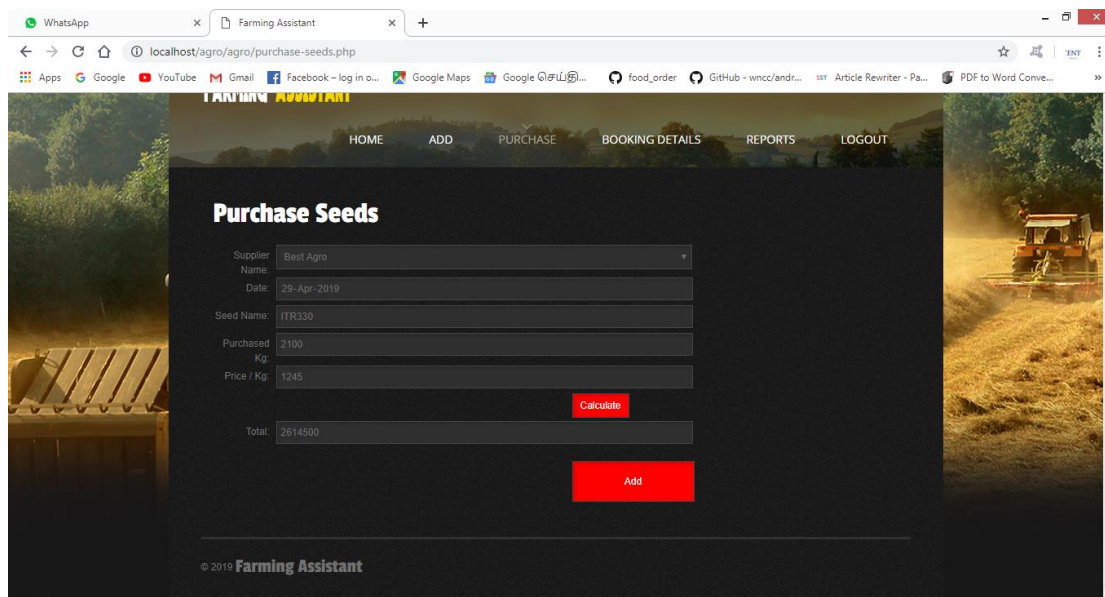


Fig:3.7.5 Used For Purchaes Seeds

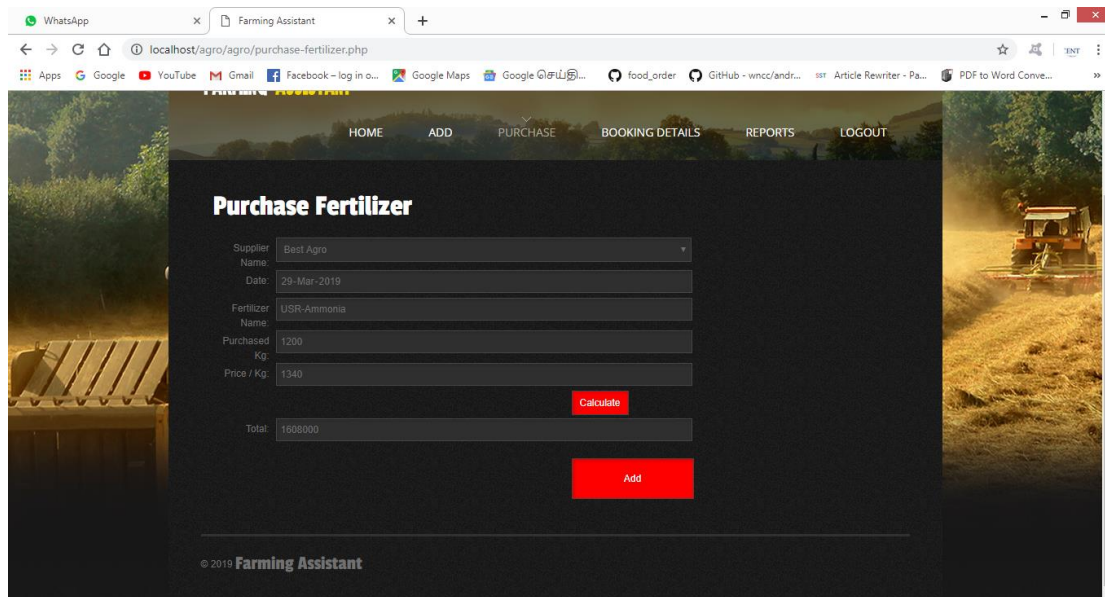


Fig:3.7 Used For Purchaes Fertilizer

USER SIDE FORMS:

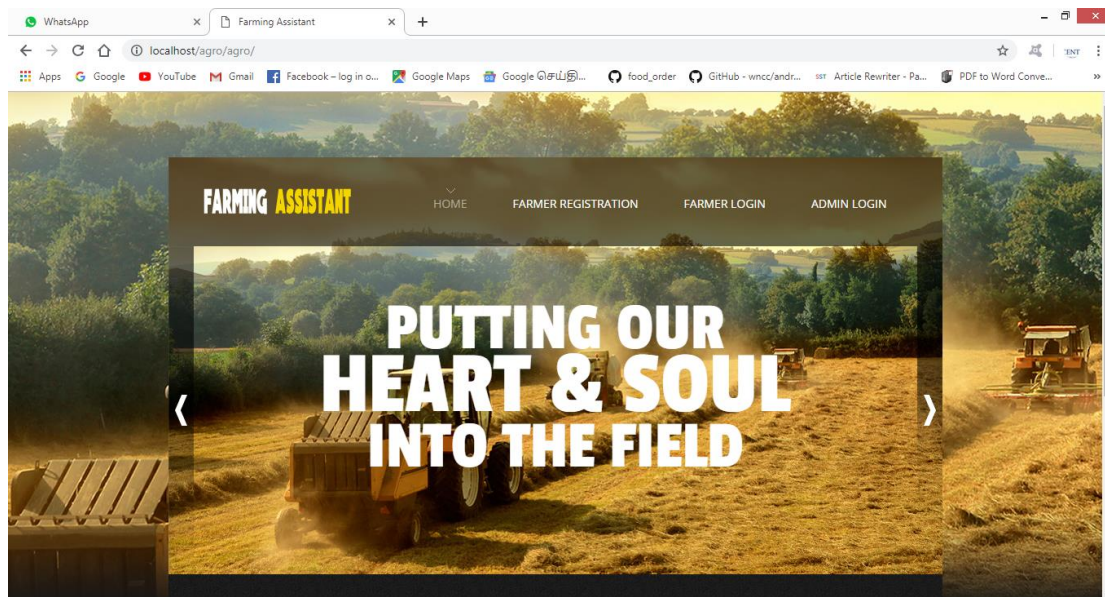


Fig:3.7.7 Home Page Of The Farming Assistant

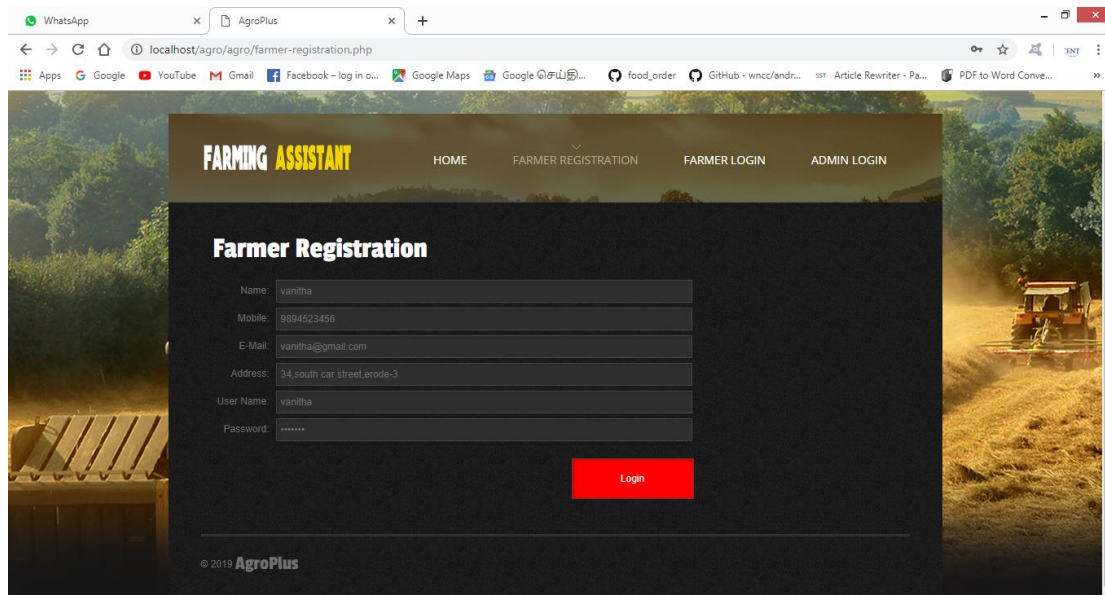


Fig:3.7.8 In This For We Register The Details Of The Farmer

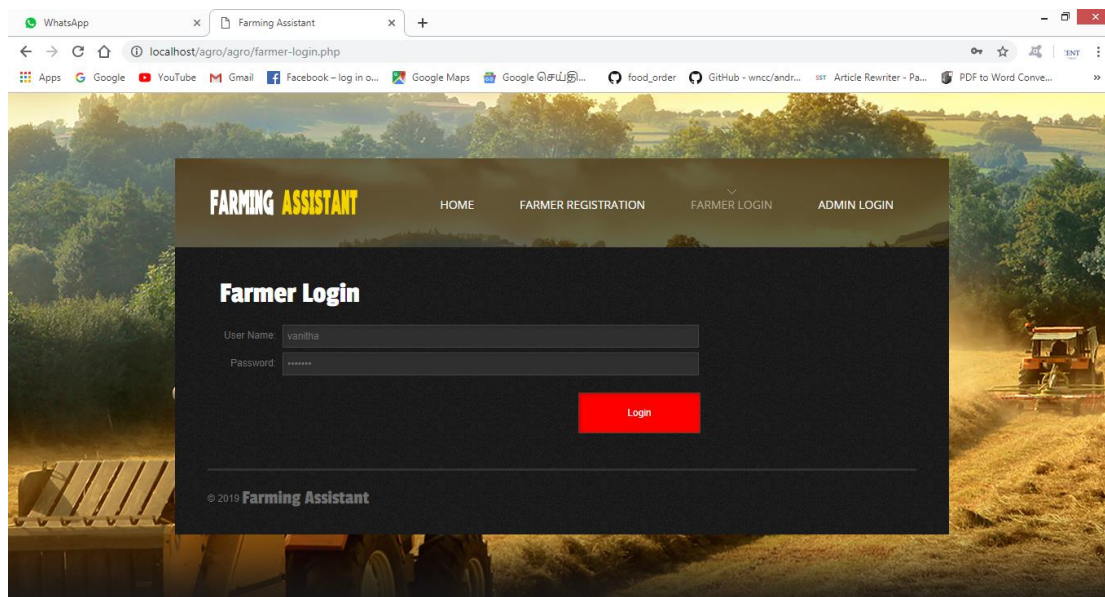


Fig:3.7.9 Used For The Farmer Login

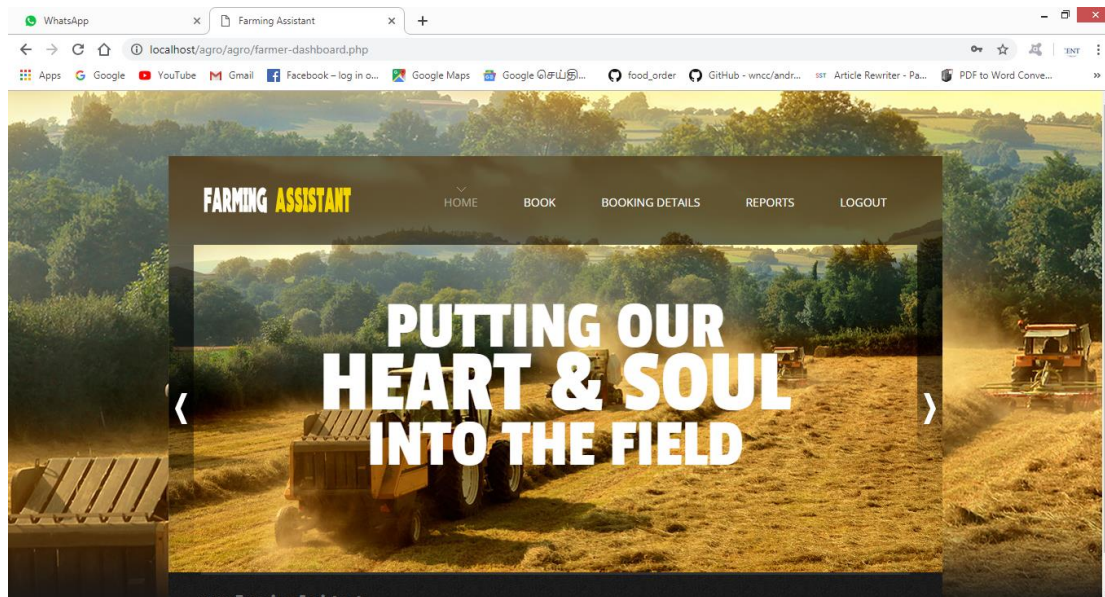


Fig:3.7.10 Home Screen After Login As A User(Farmer)

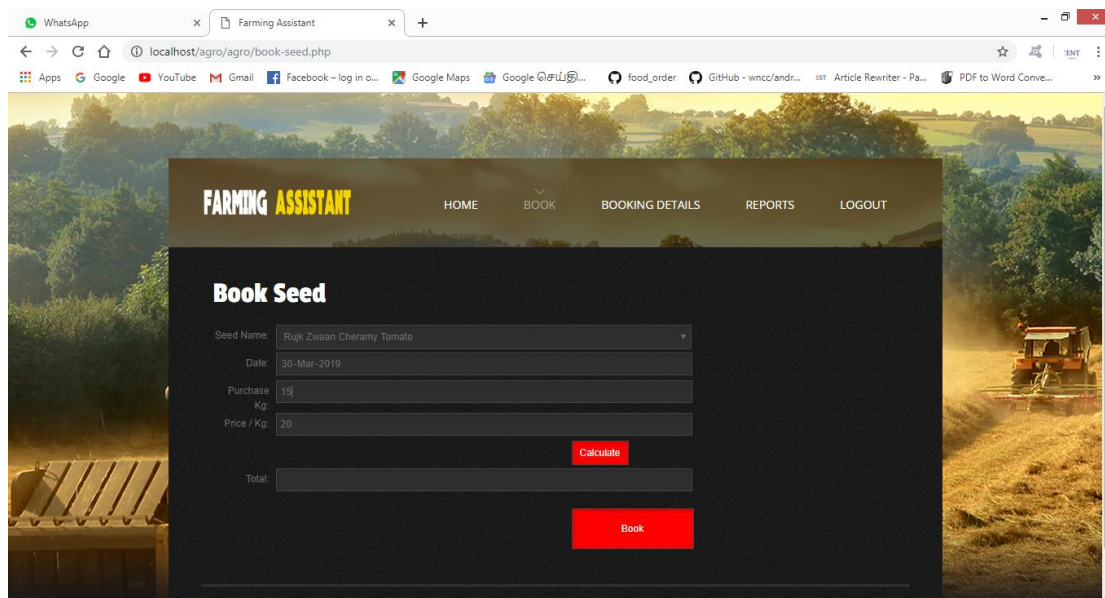


Fig:3.7.11 Used To Book Seeds

SYSTEM TESTING AND IMPLEMENTATION

CHAPTER 4

4.1 Testing Techniques

- **Unit testing**
- **Integration testing**
- **Validation testing**

Unit Testing

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces before modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Then the form level testing is made. For example, storage of data to the table is in the correct manner. In this system, each form is considered as a separate unit and tested for errors. Every user input is unit tested for a valid accepted range.

Integration Testing

Integration testing sometimes called integration and Testing, abbreviated "I&T" is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrates system ready for system testing.

The purpose of integration testing is to verify functional performance, and reliability requirements placed on major design items. These "design items", ie.assemblages(or groups of units), are exercised through their interfaces using Black box testing, success and error cases being simulated via appropriate parameter and data inputs.

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus the system testing is a confirmation that all is correct and an opportunity to show the user that the system works

Validation testing

Validation can be defined in many ways, but a simple definition is that it can be reasonably expected by the clients, which is defined in the software requirement specification, a document that describes all user-visible attributes of the software.

4.2 Test Cases with Expected result and Actual result

TEST CASE 1:

INSERTING DATABASE OF ADMIN

| S.N O | Test condition | Valid data/invalid data | Expected Result | Actual result | Test Results |
|----------|--|-------------------------------|--|---------------------------------|--------------|
| 1 | User name and password | Admin/admin 123 | Accepted the data | Accepted the data | Success |
| 2 | Correct Username and wrong password | Admin/123 | Error message will be displayed. | Wrong user name and password | Success |
| 3 | Wrong Username and correct password | 1234/adminn 123 | Error message will be displayed. | Wrong user name and password | Success |

TEST CASE 2:**INSERTING DATABASE FOR USER REGISTRATION**

| S.NO | Test data | Valid data/Invalid data | Excepted Result | Actual result | Test Results |
|------|--------------|-------------------------|----------------------------------|-------------------------------------|----------------|
| 1 | Name | deepan | Accepted the data | Accepted the data | Success |
| 2 | Phone Number | 9042005121 | Accepted the data | Accepted the data | Success |
| | | 90400A5121 | Error message will be displayed. | Phone Number only contains digits. | Not Successful |
| 3 | E-mail id | deepanraja21@gmail.com | Accepted the data | Accepted the data | Success |
| | | deepanraja21gmail.com | Error message will be displayed. | Please include "@" in email address | Not Successful |

TEST CASE 3:**INSERTING DATABASE OF UESR**

| S.NO | Test condition | Valid data/invalid data | Excepted Result | Actual result | Test Results |
|------|-------------------------------------|-------------------------|----------------------------------|------------------------------|--------------|
| 1 | User name and password | deepan/123 | Accepted the data | Accepted the data | Success |
| 2 | Correct Username and wrong password | deepan/dhi123 | Error message will be displayed. | Wrong user name and password | Success |
| 3 | Wrong Username and correct password | deepan/123 | Error message will be displayed. | Wrong user name and password | Success |

4.3 Test Reports

| S. | | | | | |
|-----------|------------------------------|--|--|---|---|
| T. N O | MODULES | TOTAL FUNCTI ON TO BE TESTED | NUMBER OF FUNCTIO N TESTED WITH SUCCESS | NUMBER OF FUNCTION NOT TESTED | NUMBER OF FUNCTION WITH ERROR |
| 1 | DATABASE OF ADMIN | 3 | 1 | | 2 |
| 2 | CUSTOMER REGISTRATI ON | 3 | 1 | | 2 |
| 3 | DATABASE OF USERNAME | 5 | 3 | 3 | 2 |

4.4 System Implementations

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 be replaced an existing one. This type of conversation is relatively easy to handle provide there are no major changes in the system. Every developed system must be implement fulfil the mode of development. There are many software implementation methods. In this system, direct change over from existing system to computer system is carried out. After designing of the system is over the user was consulted with the demonstration. This is done to fed if any logical error occurs in the system since the complete has been developed according to user requirements the demonstration is necessary. Various combination of test data were used to test the system accuracy and reliability. This implementation plans solving planning, investigation of the current system and its constraints on implementation design the methods to achieve the changes over and evaluation to change over method.

CONCLUSION

CHAPTER 5

CONCLUSION

The project, “**E- FARMING MANAGEMENT SYSTEM**” has been successfully tested and has been found to replace the existing system effectively. It is also possible to eliminate the human errors to creep into this kind of work bulk quantity of data has been processed. This project has been designed to suit all the extract needs.

This project is easy to implemented and operate .All the futures ,which are given in this system, will be successfully implemented and hence each and every module was functionality tested and found correct. The project aims to maintain feedback from user frequently. The administrator had all privileges to access this website.

The proposed system has been designed keep in mind the user requirements as the primary concern. More over the system has been developed in such a way the future modification can be done with case. The system was tested with all possible samples of data and the system provides to be fully efficient and the user maintenance and manipulation were carried out satisfactory

SCOPE FOR FUTURE ENHANCEMENT

CHAPTER 6

SCOPE FOR FUTURE ENHANCEMENT

This computerized method is a well-suited application for the real time business activities. It possess many robust features, still it can be expanded for additional features.

The requirements may change when technology and time changes. When the technology and time changes the system has to be enhanced, this system itself provides us the facility to perform some additional features. Every application has its own merits and demerits. The project has covered almost all the requirements .Further requirements and changing the existing modules or adding new modules can append improvements.

BIBLIOGRAPHY

CHAPTER 7

7.1 REFERENCES BOOKS

- Domain-Driven Design in by Carlos Buenosvinos , Christian Soronellas , **Keyvan Akbary** Publisher: Picket Published: June 2017
- **PHP Objects, Patterns, and Practice**, Fifth Edition by [Matt Zandstra](#) Publisher: [Apress](#) Published: December 2016
- Doyle, Matt. Beginning PHP 5.3 (Wrox Programmer to Programmer), 2009.
- Lerdorf, Rasmus. PHP Pocket Reference, O'Reilly, 2000.
- Nixon, Robin. Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites, O'Reilly, 2012.
- Nixon, Robin. Robin Nixon's PHP Crash Course: Learn PHP in 14 easy lectures, O'Reilly, 2012.
- Ullman, Larry. PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide (3rd Edition), 2012.
- Ullman, Larry. PHP for the Web: Visual QuickStart Guide, 2011.
- Welling, Luke and Thomson, Laura. PHP and MySQL Web Development, Third Edition, Sams, 2008.

7.2 WEBSITE REFERENCES

- <https://www.w3schools.com/PHP/default.asp>
- <https://www.codecademy.com/en/tracks/PHP>
- <https://www.codeschool.com/learn/PHP>
- <https://www.sololearn.com/Course/PHP/>
- <https://www.guru99.com/PHP-tutorials.PHP>

APPENDICES

CHAPTER 8

8.1 Source Code

```
<?php

// This file is www.developphp.com curriculum material

// Written by Adam Khoury January 01, 2011

// http://www.youtube.com/view_play_list?p=442E340A42191003

session_start(); // Start session first thing in script

// Script Error Reporting

error_reporting(E_ALL);

ini_set('display_errors', '1');

// Connect to the MySQL database

include "storescripts/connect_to_mysql.php";

?>


<?php

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////

//      Section 1 (if user attempts to add something to the cart from the product page)
```

```

////////////////////////////////////
////////////////////////////////////
if (isset($_POST['pid'])) {
    $pid = $_POST['pid'];

    $wasFound = false;

    $i = 0;

    // If the cart session variable is not set or cart array is empty
    if (!isset($_SESSION["cart_array"]) || count($_SESSION["cart_array"]) < 1) {

        // RUN IF THE CART IS EMPTY OR NOT SET

        $_SESSION["cart_array"] = array(0 => array("item_id" => $pid, "quantity" => 1));

    } else {

        // RUN IF THE CART HAS AT LEAST ONE ITEM IN IT

        foreach ($_SESSION["cart_array"] as $each_item) {

            $i++;

            while (list($key, $value) = each($each_item)) {

                if ($key == "item_id" && $value == $pid) {

                    // That item is in cart already so let's adjust its quantity using array_splice()

                    array_splice($_SESSION["cart_array"], $i-1, 1, array(array("item_id" => $pid,
"quantity" => $each_item['quantity'] + 1)));

                    $wasFound = true;

                } // close if condition

            } // close while loop

        } // close foreach loop

        if ($wasFound == false) {

            array_push($_SESSION["cart_array"], array("item_id" => $pid, "quantity" => 1));

```

```

if (isset($_POST['pid'])) {

$pid = $_POST['pid'];

$wasFound = false;

$i = 0;

// If the cart session variable is not set or cart array is empty

if (!isset($_SESSION["cart_array"]) || count($_SESSION["cart_array"]) < 1) {

// RUN IF THE CART IS EMPTY OR NOT SET

$_SESSION["cart_array"] = array(0 => array("item_id" => $pid, "quantity" => 1));

} else {

// RUN IF THE CART HAS AT LEAST ONE ITEM IN IT

foreach ($_SESSION["cart_array"] as $each_item) {

$i++;

while (list($key, $value) = each($each_item)) {

if ($key == "item_id" && $value == $pid) {

// That item is in cart already so let's adjust its quantity using array_splice()

array_splice($_SESSION["cart_array"], $i-1, 1, array(array("item_id" => $pid,

"quantity" => $each_item['quantity'] + 1)));

$wasFound = true;

} // close if condition

} // close while loop

} // close foreach loop

if ($wasFound == false) {

array_push($_SESSION["cart_array"], array("item_id" => $pid, "quantity" => 1));

```

```

    }

    }

    header("location: cart.php");

    exit();

}

?>

<?php

////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////

//      Section 2 (if user chooses to empty their shopping cart)

////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////

if (isset($_GET['cmd']) && $_GET['cmd'] == "emptycart") {

    unset($_SESSION["cart_array"]);

}

?>

<?php

////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////

//      Section 3 (if user chooses to adjust item quantity)

////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////

if (isset($_POST['item_to_adjust']) && $_POST['item_to_adjust'] != "") {

```



```
// execute some code

$item_to_adjust = $_POST['item_to_adjust'];

$quantity = $_POST['quantity'];

$quantity = preg_replace('#[^0-9]#i', '', $quantity); // filter everything but numbers

if ($quantity >= 100) { $quantity = 99; }

if ($quantity < 1) { $quantity = 1; }

if ($quantity == "") { $quantity = 1; }

$i = 0;

foreach ($_SESSION["cart_array"] as $each_item) {

    $i++;

    while (list($key, $value) = each($each_item)) {

        if ($key == "item_id" && $value == $item_to_adjust) {

            // That item is in cart already so let's adjust its quantity using array_splice()

            array_splice($_SESSION["cart_array"], $i-1, 1, array(array("item_id" =>

                $item_to_adjust, "quantity" => $quantity)));

        } // close if condition

    } // close while loop

} // close foreach loop

}
```

?>

<?php

//

//

// Section 4 (if user wants to remove an item from cart)

```
// Access the array and run code to remove that array index

$key_to_remove = $_POST['index_to_remove'];

if (count($_SESSION["cart_array"]) <= 1) {

unset($_SESSION["cart_array"]);

} else {

unset($_SESSION["cart_array"][$key_to_remove]);

sort($_SESSION["cart_array"]);

}

}

?>

<?php

// Section 5 (render the cart for the user to view on the page)

$cartOutput = "";

$cartTotal = "";

$pp_checkout_btn = "Start PayPal Checkout Button";

$product_id_array = "";

if (!isset($_SESSION["cart_array"]) || count($_SESSION["cart_array"]) < 1) {

$cartOutput = "<h2 align='center'>Your shopping cart is empty</h2>";

} else {
```

```

$pp_checkout_btn .= '<form action="https://www.paypal.com/cgi-bin/webscr"
method="post">

<input type="hidden" name="cmd" value="_cart">

<input type="hidden" name="upload" value="1">

<input type="hidden" name="business" value="you@youremail.com">';

// Start the For Each loop

$i = 0;

foreach ($_SESSION["cart_array"] as $each_item) {

$item_id = $each_item['item_id'];

$sql = mysql_query("SELECT * FROM products WHERE id='$item_id' LIMIT 1");

while ($row = mysql_fetch_array($sql)) {

$product_name = $row["product_name"];

$price = $row["price"];

$details = $row["details"];

}

$pricetotal = $price * $each_item['quantity'];

$cartTotal = $pricetotal + $cartTotal;

setlocale(LC_MONETARY, "en_US");

$pricetotal = money_format("%10.2n", $pricetotal);

// Dynamic Checkout Btn Assembly

$x = $i + 1;

$pp_checkout_btn .= '<input type="hidden" name="item_name_' . $x . '" value="' .

$product_name . '">

<input type="hidden" name="amount_' . $x . '" value="' . $price . '">

```

```

<input type="hidden" name="quantity_' . $x . '" value="" . $each_item['quantity'] . '">
';

// Create the product array variable
$product_id_array .= "$item_id-".$each_item['quantity'].",";

// Dynamic table row assembly
$cartOutput .= "<tr>";

$cartOutput .= '<td><a href="product.php?id=' . $item_id . '">' . $product_name .
'</a><br /><imgsrc="inventory_images/' . $item_id . '.jpg" alt="" . $product_name. "'
width="40" height="52" border="1" /></td>';

$cartOutput .= '<td>' . $details . '</td>';

$cartOutput .= '<td>$' . $price . '</td>';

$cartOutput .= '<td><form action="cart.php" method="post">
<input name="quantity" type="text" value="" . $each_item['quantity'] . '" size="1"
maxlength="2" />
<input name="adjustBtn' . $item_id . '" type="submit" value="change" />
<input name="item_to_adjust" type="hidden" value="" . $item_id . '" />
</form></td>';

// $cartOutput .= '<td>' . $each_item['quantity'] . '</td>';

$cartOutput .= '<td>' . $pricetotal . '</td>';

$cartOutput      .=      '<td><form      action="cart.php"      method="post"><input
name="deleteBtn'      .      $item_id      .      '"      type="submit"      value="X"      /><input
name="index_to_remove" type="hidden" value="" . $i . '" /></form></td>';

$cartOutput .= '</tr>';

$i++;
}

```

```

setlocale(LC_MONETARY, "en_US");

$cartTotal = money_format("% 10.2n", $cartTotal);

$cartTotal = "<div style='font-size:18px; margin-top:12px;' align='right'>Cart Total :

".$cartTotal." USD</div>";

// Finish the Paypal Checkout Btn

$pp_checkout_btn .= '<input type="hidden" name="custom" value=""

$product_id_array . ">

<input type="hidden" name="notify_url"

value="https://www.yoursite.com/storescripts/my_ipn.php">

<input type="hidden" name="return"

value="https://www.yoursite.com/checkout_complete.php">

<input type="hidden" name="rm" value="2">

<input type="hidden" name="cbt" value="Return to The Store">

<input type="hidden" name="cancel_return"

value="https://www.yoursite.com/paypal_cancel.php">

<input type="hidden" name="lc" value="US">

<input type="hidden" name="currency_code" value="USD">

<input type="image" src="http://www.paypal.com/en_US/i/btn/x-click-but01.gif"

name="submit" alt="Make payments with PayPal - its fast, free and secure!">

</form>';

}

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head>

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

<title>Your Cart</title>

<link rel="stylesheet" href="style/style.css" type="text/css" media="screen" />

</head>

<body>

<div align="center" id="mainWrapper">

<?phpinclude_once("template_header.php");?>

<div id="pageContent">

<div style="margin:24px; text-align:left;">

<br />

<table width="100%" border="1" cellspacing="0" cellpadding="6">

<tr>

<td width="18%" bgcolor="#C5DFFA"><strong>Product</strong></td>

<td width="45%" bgcolor="#C5DFFA"><strong>Product Description</strong></td>

<td width="10%" bgcolor="#C5DFFA"><strong>Unit Price</strong></td>

<td width="9%" bgcolor="#C5DFFA"><strong>Quantity</strong></td>

<td width="9%" bgcolor="#C5DFFA"><strong>Total</strong></td>

<td width="9%" bgcolor="#C5DFFA"><strong>Remove</strong></td>

</tr>

<?php echo $cartOutput; ?>

<!--<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

```

```

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr> -->

</table>

<?php echo $cartTotal; ?>

<br />

<br />

<?php //echo $pp_checkout_btn; ?>

<br />

<br />

</div>

<?phpinclude_once("template_footer.php");?>

</div>

</body>

</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

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

<title>Untitled Document</title>

</head>

<body>

```

Yes this is supposed to be empty like this... happy coding.

</body>

</html>

Index

<?php

// This file is www.developphp.com curriculum material

// Written by Adam Khoury January 01, 2011

// http://www.youtube.com/view_play_list?p=442E340A42191003

// Script Error Reporting

error_reporting(E_ALL);

ini_set('display_errors', '1');

?>

<?php

// Run a select query to get my latest 6 items

// Connect to the MySQL database

include "storescripts/connect_to_mysql.php";

\$dynamicList = "";

\$sql = mysql_query("SELECT * FROM products ORDER BY date_added DESC
LIMIT 6");

\$productCount = mysql_num_rows(\$sql); // count the output amount

if (\$productCount > 0) {

while(\$row = mysql_fetch_array(\$sql)){

\$id = \$row["id"];

\$product_name = \$row["product_name"];

\$price = \$row["price"];

\$date_added = strftime("%b %d, %Y", strtotime(\$row["date_added"]));


```

$dynamicList .= '<table width="100%" border="0" cellspacing="0" cellpadding="6">

<tr>

<td width="17%" valign="top"><a href="product.php?id=' . $id . "'></a></td>

<td width="83%" valign="top">' . $product_name . '<br />
$' . $price . '<br />

<a href="product.php?id=' . $id . "'>View Product Details</a></td>

</tr>

</table>';

}

} else {

}

mysql_close();

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

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

<title>Store Home Page</title>

<link rel="stylesheet" href="style/style.css" type="text/css" media="screen" />

</head>

<body>

<div align="center" id="mainWrapper">

```

```

<?phpinclude_once("template_header.php");?>

<div id="pageContent">

<table width="100%" border="0" cellspacing="0" cellpadding="10">

<tr>

<td width="32%" valign="top"><h3>What the Hell?</h3>

<p>This website is very temporarily being used as an online live showcase area for an

E - Commerce tutorial script set Adam is creating which can be seen on his channel

here:<br />

target="_blank">http://www.youtube.com/flashbuilding</a></p>

<p>It is not an actual store and it will change directly after the tutorial series. <br />

<br />

This tutorial series is for educational purposes only. Use the scripts at your own

risk.</p></td>

<td width="35%" valign="top"><h3>Latest Designer Fashions</h3>

<p><?php echo $dynamicList; ?><br />

<p><br />

</p></td>

<td width="33%" valign="top"><h3>Handy Tips</h3>

</tr>

</table>

</div>

<?phpinclude_once("template_footer.php");?>

```

</div>

</body>

</html>

8.2 Screen Layout

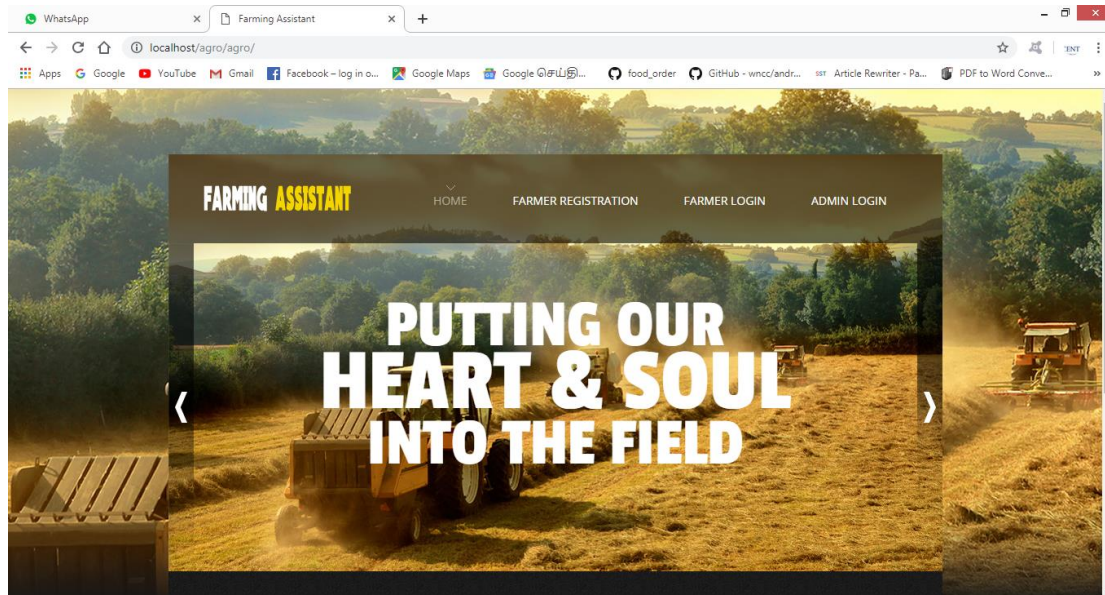


Fig:8.2.1 Home Page

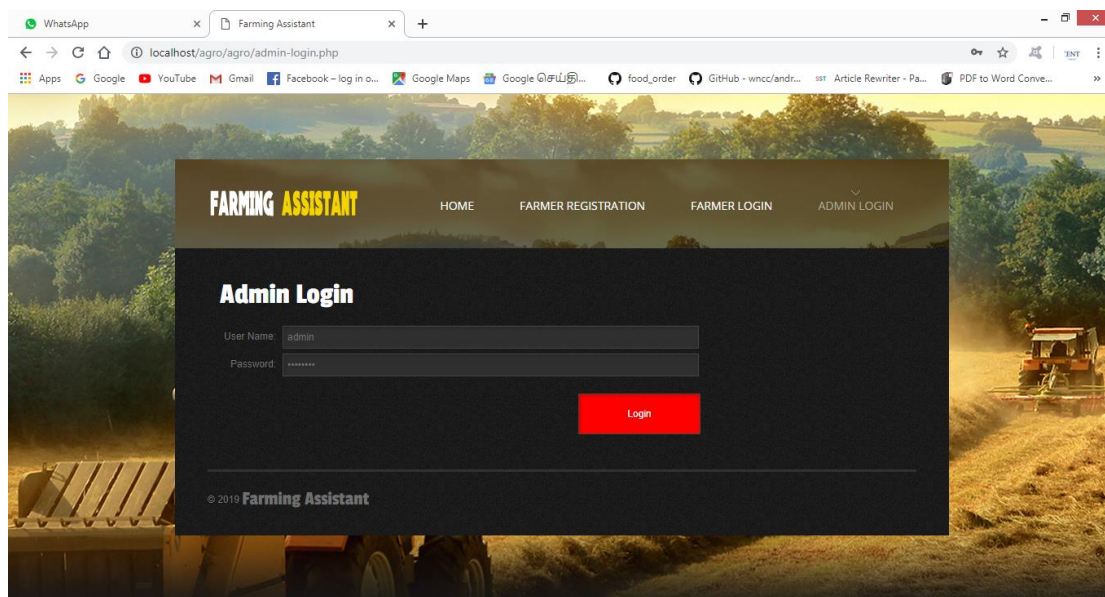


Fig:8.2.2 Admin Login

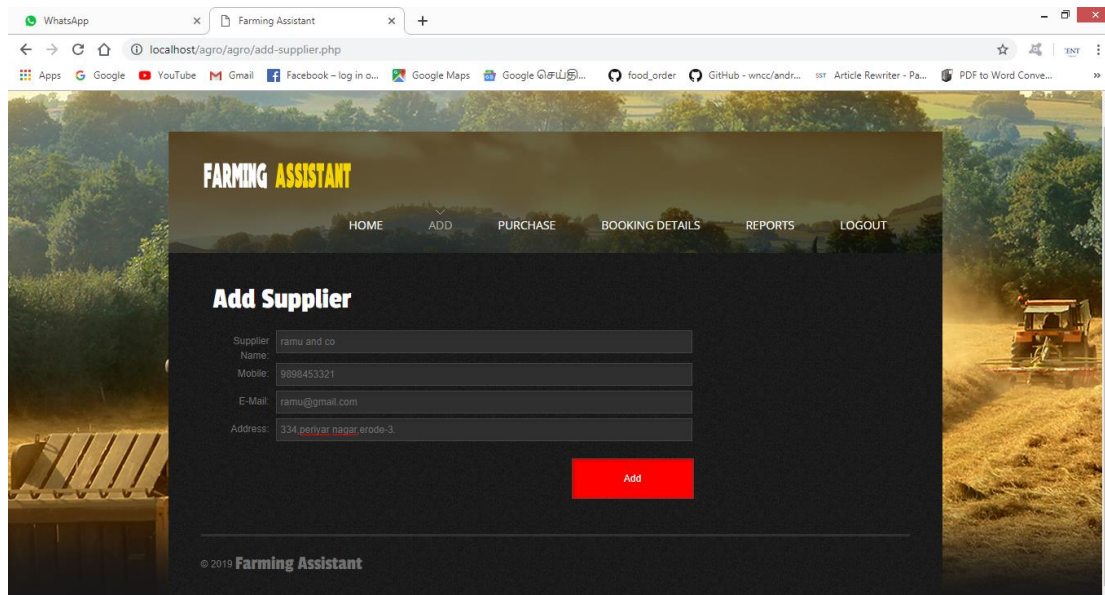


Fig:8.2.3 Admin Home Page

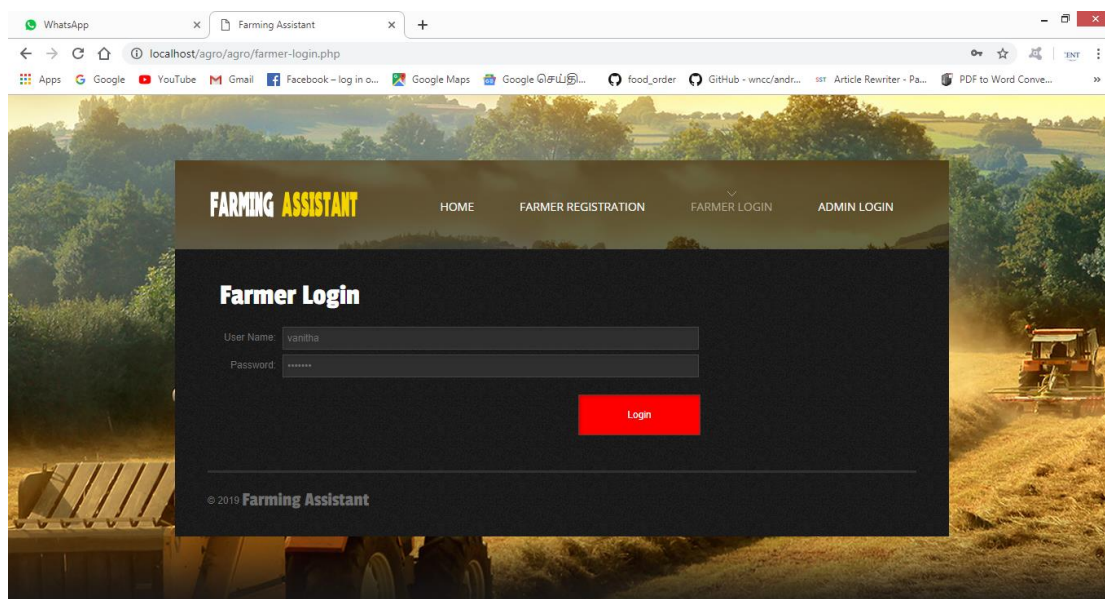


Fig:8.2.4 User Login

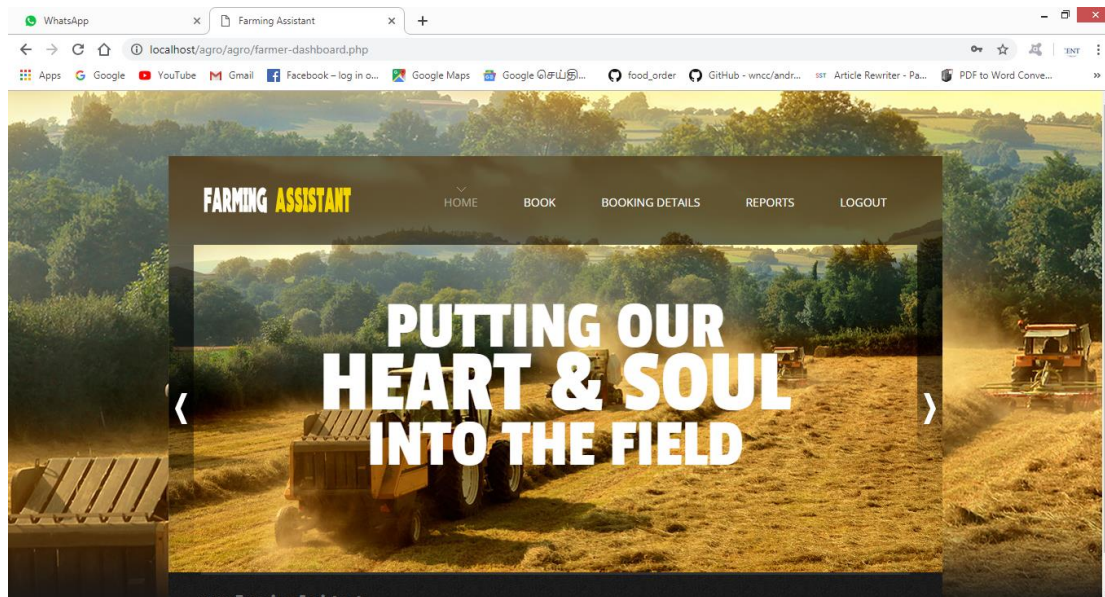


Fig:8.2.5 User Home Page