|  |
| --- |
|  |
| FARMATE |
|  |
| MiniProject Report submitted as part of ITE302 Embedded Lab course work |

**12bit0021**

**12bit0095**

**12bit0072**

November 7, 2014

Guided By: Prof.Varalakshmi M, SITE

FARMATE

MiniProject Report submitted as part of ITE302 Embedded Lab course work

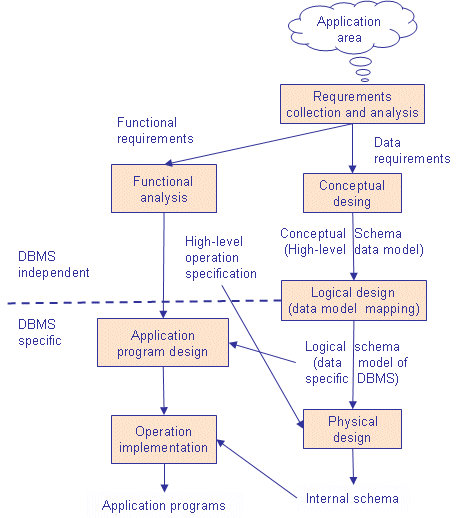
ABSTRACT

In India most of the families are depend on the Agriculture, but the profit which farmer gains is very low. While selling his product to the market with the help of the middleman, the middleman yields more profit compared to the farmer. The profit which gained by the farmer is not worth for his hard work. The only solution to avoid this problem is to drop the middleman and makes the farmer to sell his product directly to the market. In order to achieve this solution we designed a webpage to makes the farmer to sell his product directly to the market. In this webpage farmer will get into his account using his unique id. His account contains different option like selling, loan, buying accessories. Here farmer can sell his product directly to the market dealer with fixed MSP also farmer can increase or decrease bedding amount under sell option. He can take loan from the bank for his farming purpose under loan option. He can buy accessories from the accessory dealer under buy option. The farmer page is very user friendly with pictorial option. The website also contains market dealer authentication and accessory dealer authentication. In market dealer the dealer can buy products from the farmer based on the price offered by the farmer, also he can sell his product to the others which is categorized based on crops type. In accessory dealer page the dealer receives the request from the farmer as number of accessories farmer needs, even though the dealer can increase or decrease the accessory rate, offer, availability of the accessories in his page which will display in the farmer page. With the help of this website the farmer will yield more money, it will also reduce the work of the farmer.

INTRODUCTION

A database is a collection of related data. Eg. Library ( Book, Member, journal) database, Railway Reservation (Train, station, passenger, ticket) database. A Database Management System is software or a set of programs which can create and manipulate the database. Eg. Oracle, Db2, Sybase, and Informix are some commercial DBMSs. Railway Reservation System, Library Management System, Online quiz, Stock Management System are some of the DBMS applications which are developed using any commercial DBMS. For developing a DBMS application collection, analysis and design of schemas at different level is important.

DESIGN PHASES



**ANALYSIS**

**MODULES**

**ADD**

In this module the records of farmer, market dealer, accessory dealer and commodity can be added.

**DELETE**

In this module we can choose to delete the desired record by giving the value of the primary key of the entity. We can delete the record of farmer, accessory dealer, market dealer or commodity.

**QUERY**

In this module nested and complex queries are executed.

Query1. Print the details of the farmer who sells the given commodity at a minimum rate.

Query2. Given a farmer id find the total rate of his commodities and print the details of the farmer along with the sum.

Query3. Refined search: only show the list of commodities that come under the interest of the user. The search can be refined to show only a particular commodity or commodities less than a certain rate or those of high grade.

**VIEW**

This module helps us to view the records in the database. We can see the table of farmer, market dealer, accessory dealer or commodity.

ER-DIAGRAM

Buys\_from

Buys\_from

sells

buys

sells

buys

ACCESSORY

ACCESSORY\_DEALER

MARKET\_DEALER

COMMODITY

FARMER

MGT -RELATIONAL SCHEMA

|  |
| --- |
| Farmers(farmer ID,name,location,commodity\_type,user\_id,password) |

|  |
| --- |
| Market\_Dealer(dealer\_id,name,location,commodity type,user\_id,password) |

|  |
| --- |
| Accesory\_Dealer(dealer\_id,name,location,commodity\_type,user\_id,password) |

|  |
| --- |
| Commodity(name,farmer\_id,type,rate,location,quantity,quality) |

|  |
| --- |
| Accessory(name,dealer\_id,rate,type,location,commodity) |

BACK END IMPLEMENTATION

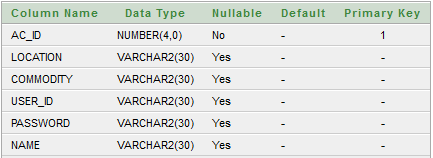


Figure : ACCESS\_DEALER TABLE WITH CONSTRAINTS

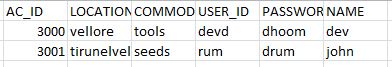


Figure 2: ACCESS\_DEALER TABLE WITH VALUES

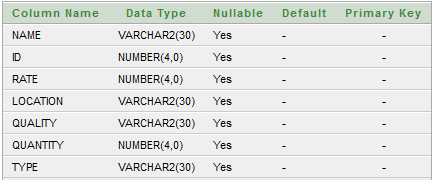


Figure 3: ACCESSORY TABLE WITH CONSTRAINTS

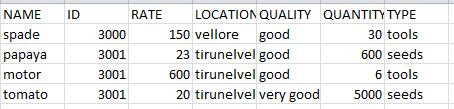


Figure 4: ACCESSORY TABLE WITH VALUES

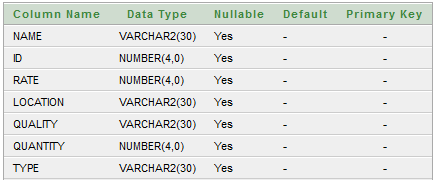


Figure 5 COMMODITY TABLE WITH CONSTRAINTS

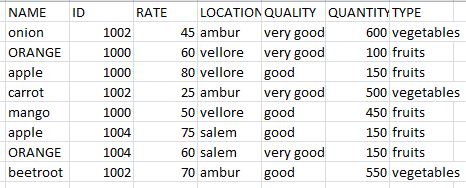


Figure 6 COMMODITY TABLE WITH VALUES

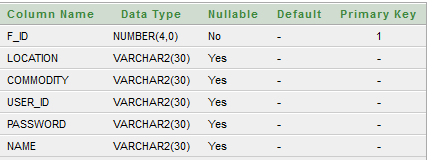


Figure 7 : FARMER TABLE WITH CONSTRAINTS



Figure8 FARMER TABLE WITH VALUES

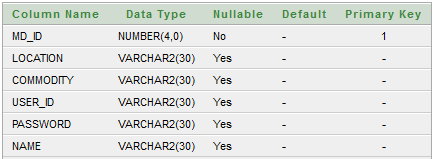
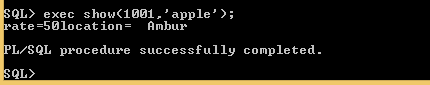


Figure 9 MARKET\_DEALER TABLE WITH CONSTRAINTS

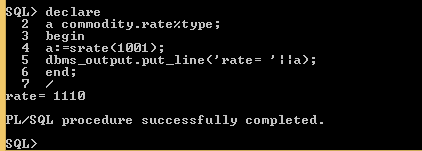


Figure10 MARKET\_DEALER TABLE WITH VALUES

## Figure 11.Cursor implementation



### Figure 12.Implementation of procedure show



## Figure 13.Implementation of function srate

## 

## Figure 14. Implementation of trigger

DESIGN

APPLICATION PROGRAM DESIGN:

SOFTWARE REQUIREMENTS

• Web Presentation : HTML, CSS, JavaScript

• Database Connectivity : PHP

• Backend Database : Oracle Db

• Operating System : Windows XP/2003, later

• Web Server : Apache

• Browser : Mozilla/chrome

HARDWARE REQUIREMENTS

• Pentium processor : 1.1 GHZ

• RAM Capacity : 256 MB (min)

• Hard Disk : 20GB

• CD-ROM Drive : 32 HZ

• Keyboard : Standard keyboard.

• Mouse : Optical

• Monitor : 15’’ Color Monitor

**<HTML>**

HTML is a markup language for describing web documents (web pages). HTML stands for Hyper Text Markup Language. A markup language is a set of markup tags.HTML documents are described by HTML tags. Each HTML tag describes different document content.

HTML Tags

<MARQUEE>

The HTML <marquee> tag is used for scrolling piece of text or image displayed either horizontally across or vertically down your web site page depending on the settings.

<META>

The <meta> tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

<IFRAME>

The HTML <iframe> tag is used to create an inline frame.

<JAVASCRIPT>:

JavaScript is the programming language of the Web. All modern HTML pages are using JavaScript. It is the most popular programming language in the world. It is the language for HTML, for the Web, for computers, servers, laptops, tablets, smart phones, and more.

The <script> and </script> tells where the JavaScript starts and ends. The lines between <script> and </script> contain the JavaScript code.

validateUsername ( )

The function below checks if the user entered anything at all in the username field. If it’s not blank, we check the length of the string and permit only usernames that are between 5 and 15 characters. Next, we use the JavaScript regular expression /\W/ to forbid illegal characters from appearing in usernames. We want to allow only letters, numbers and underscopes.

validatePassword ( )

The function below checks the password field for blankness and allow only letters and numbers - no underscopes this time. So we should use a new regular expression to forbid underscopes. This one /[\W\_]/ allow only letters and numbers. Next, we want to permit only passwords that contain letters and at least one numeral.

<PHP>

PHP is a server scripting language, and is a powerful tool for making dynamic and interactive Web pages quickly. It is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is an acronym for "PHP Hypertext Preprocessor" .It is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.). It is compatible with almost all servers used today (Apache, IIS, etc.). It supports a wide range of databases.

PARAMETERS:

mysql\_connect()

Open a connection to a MySQL Server. This method takes three parameters like server name, username and password.

server

It can also include a port number. e.g. "hostname:port" or a path to a local socket e.g. ":/path/to/socket" for the localhost.

If the PHP directive mysql.default\_host is undefined (default), then the default value is 'localhost:3306'. In SQL safe mode, this parameter is ignored and value 'localhost:3306' is always used.

username

The username. Default value is defined by mysql.default\_user. In SQL safe mode, this parameter is ignored and the name of the user that owns the server process is used.

password

The password. Default value is defined by mysql.default\_password. In SQL safe mode, this parameter is ignored and empty password is used.

Mysql\_select\_db()

This method is used to select the database. It takes two parameters like database name and the connection variable. Ex: ‘mgdb’, $conn

Mysql\_query()

This method is used to execute queries. It also takes two parameters like query and connection variable. Ex : ‘insert into caseregister values (1024, abdul)’, $conn

Mysql\_close()

This method is used to close the database connection. It takes the connection variable as a parameter.

FRONT END IMPLEMENTATION

**MAIN HOME PAGE**

This is the official page of the FARMATE admin that allows the administrator access to the management system.

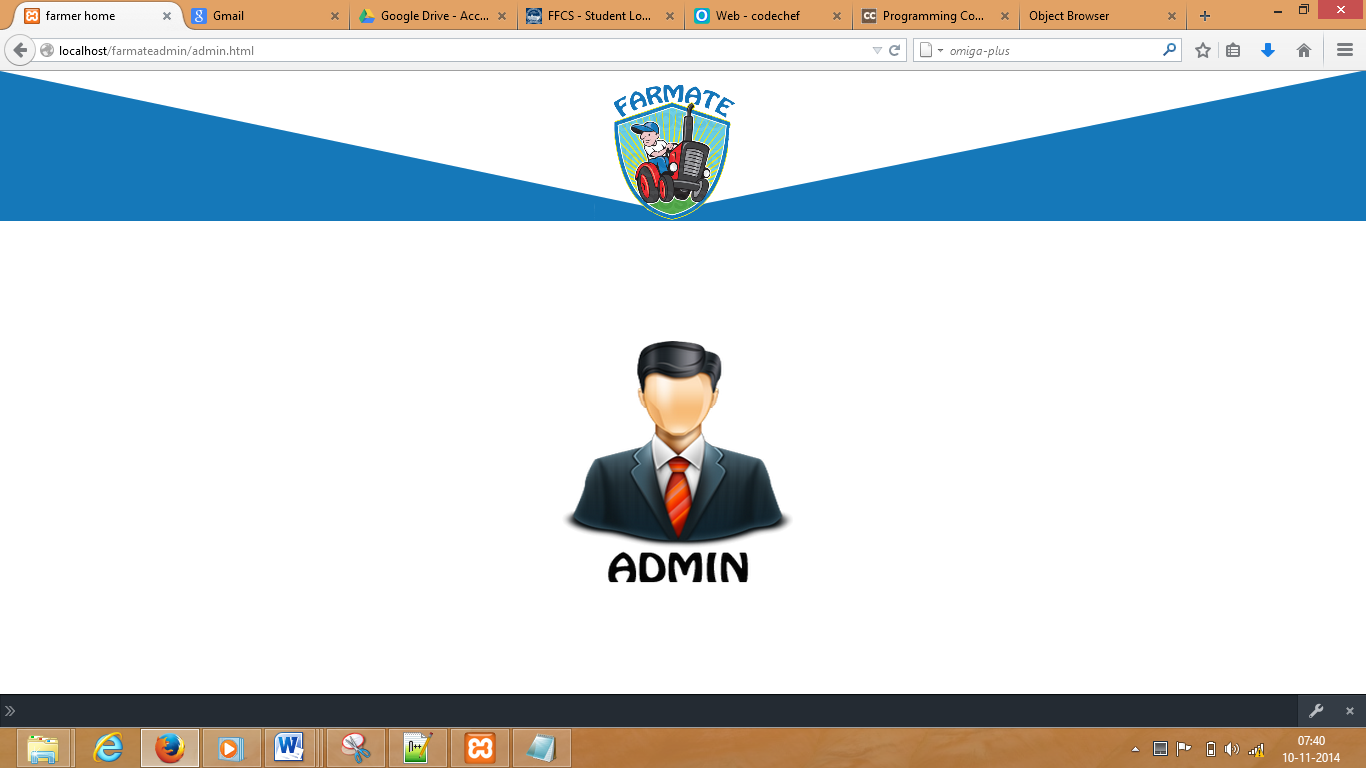


Figure 11 : MAIN HOME PAGE

**OPTIONS HOME PAGE**

This module allows the admin access to the records maintained in the database. He can add or remove recods, view them and implement complex queries.

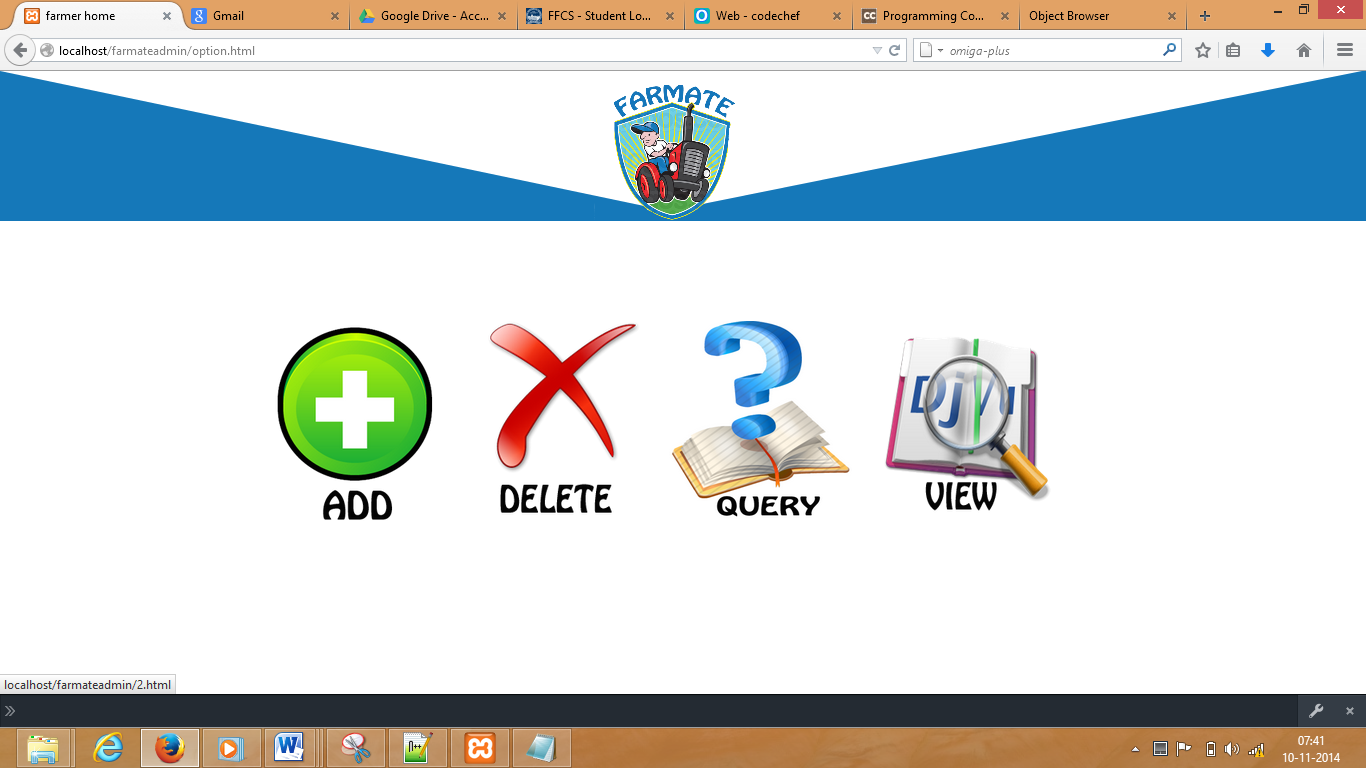


Figure 12 : OPTIONS HOME PAGE

**CHOOSE ENTITY PAGE**

This module allows the admin to choose the entity where he wants to implement the desired option that he selects from the options home page. The available entities are farmer, market dealer, accessory dealer, commodity, accessory.



Figure 13 : CHOOSE ENTITY PAGE

**ADD ENTITY FORM**

This module allows the admin to add an entity record. Admin needs to fill the form detail depending on type of entity. There are different forms for different entities.

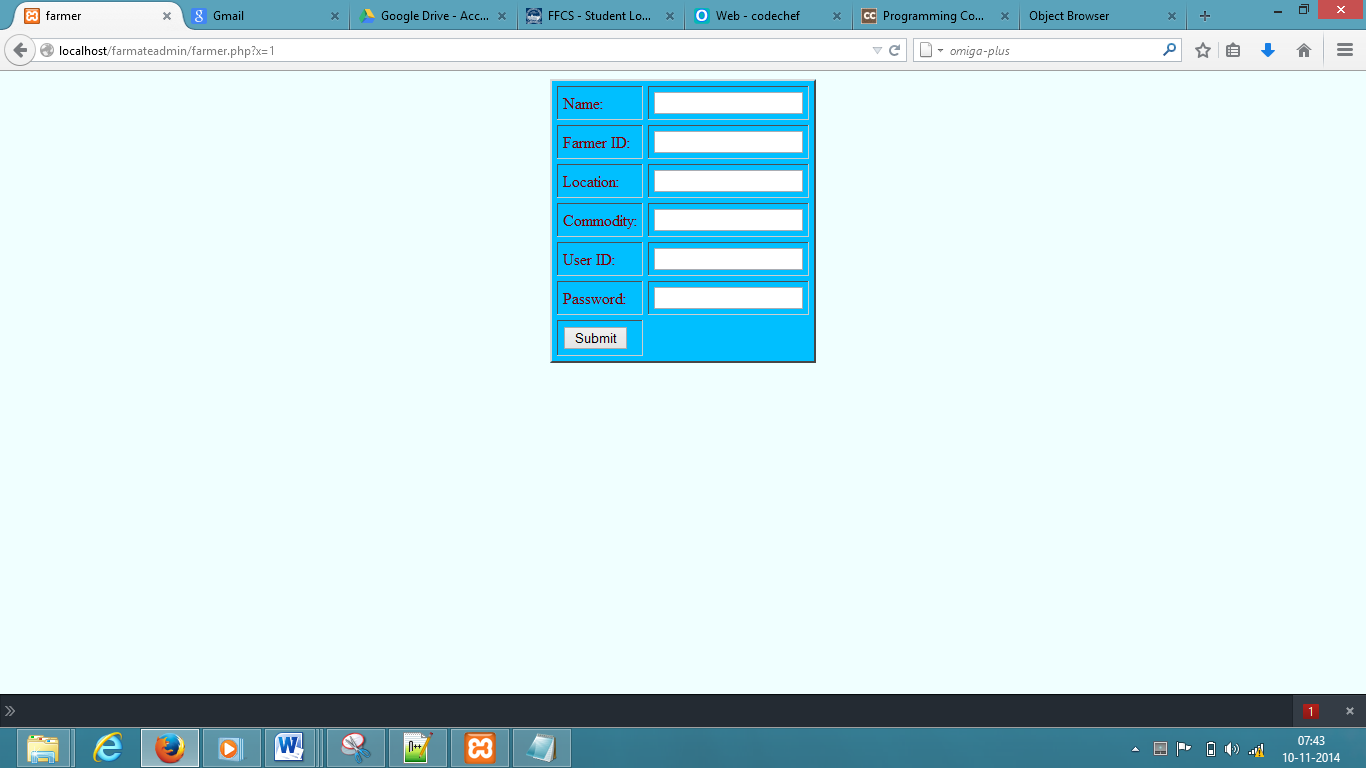


Figure 14 : ADD FARMER FORM

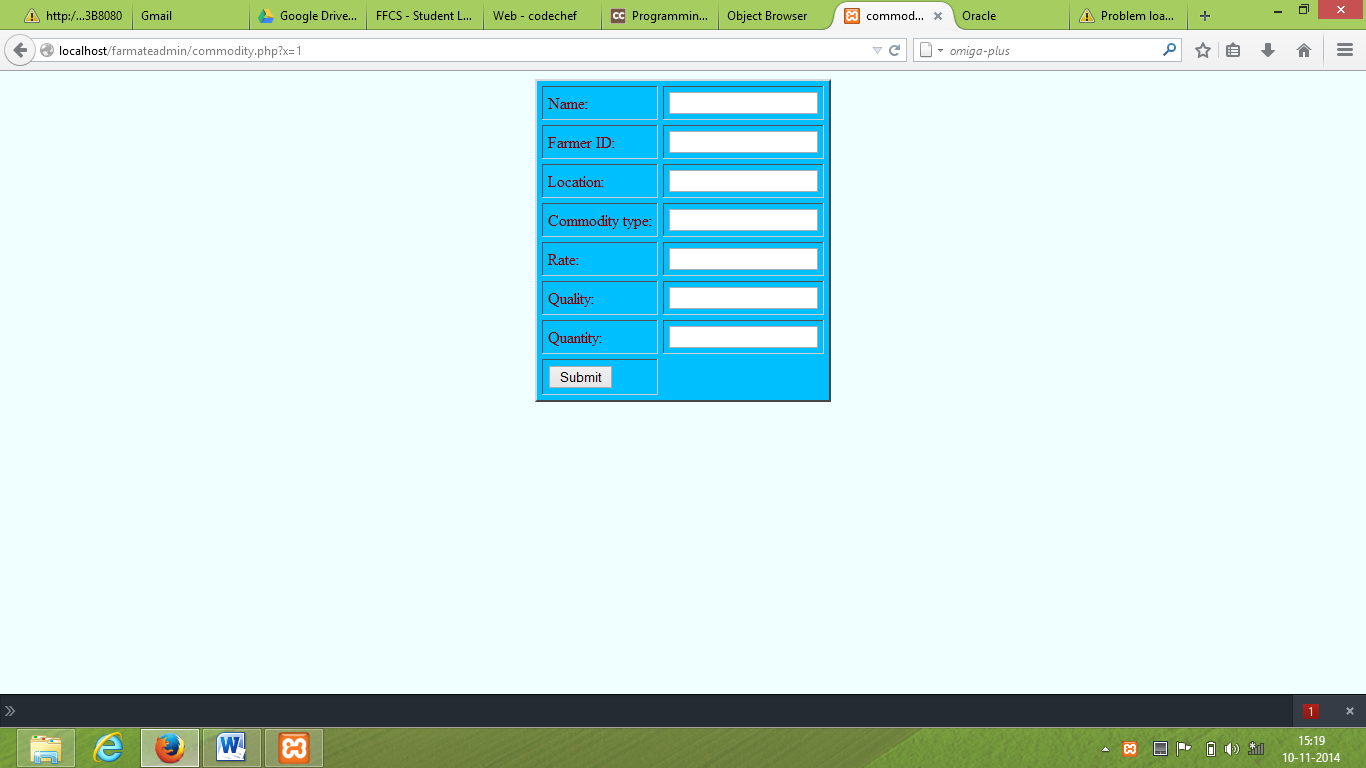


Figure 15 : ADD COMMODITY FORM

**DELETE ENITY PAGE:**

This module enables the admin to delete records of different entities. For deleting farmer, or market dealer, or accessory dealer records he needs to give their unique ID. For deleting records of commodity or accessory he needs to type their name and the ID of their respective farmer or accessory dealer.

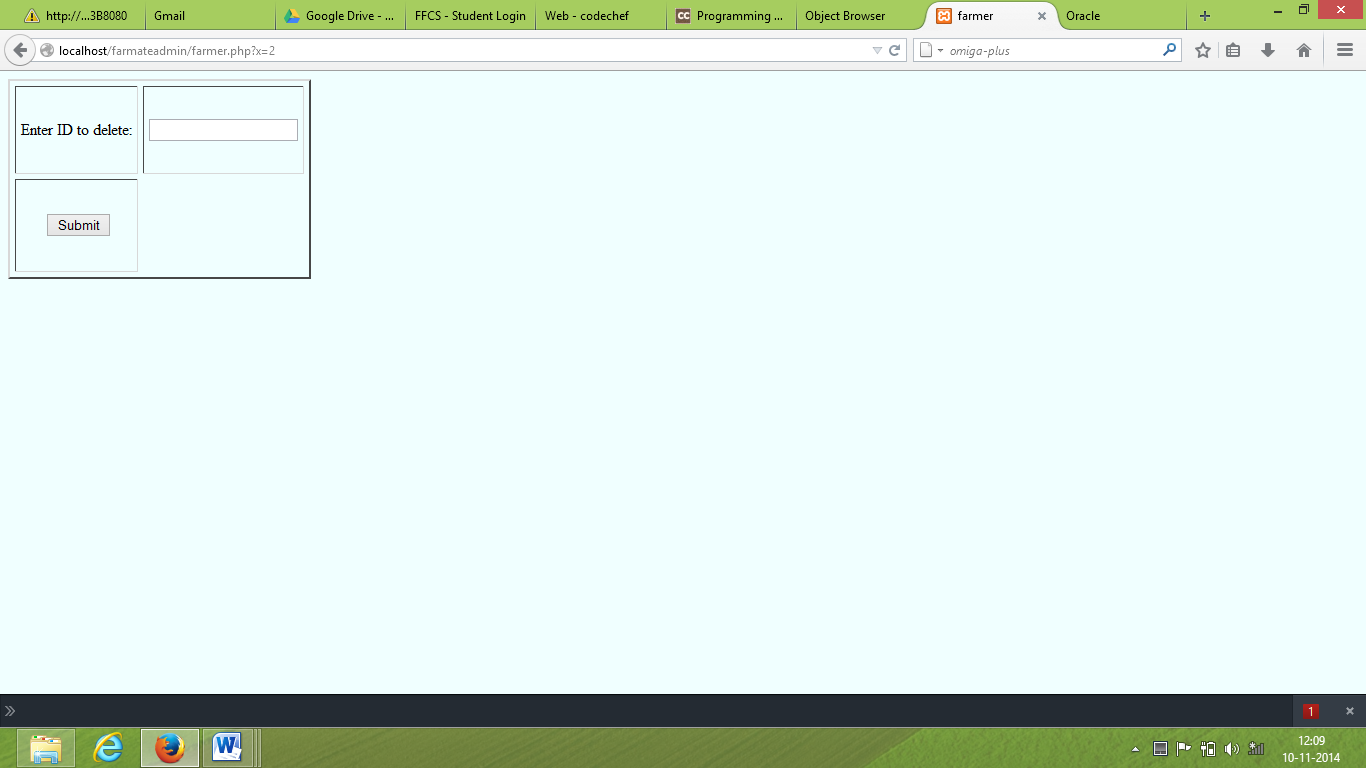


Figure16: DELETE FARMERS

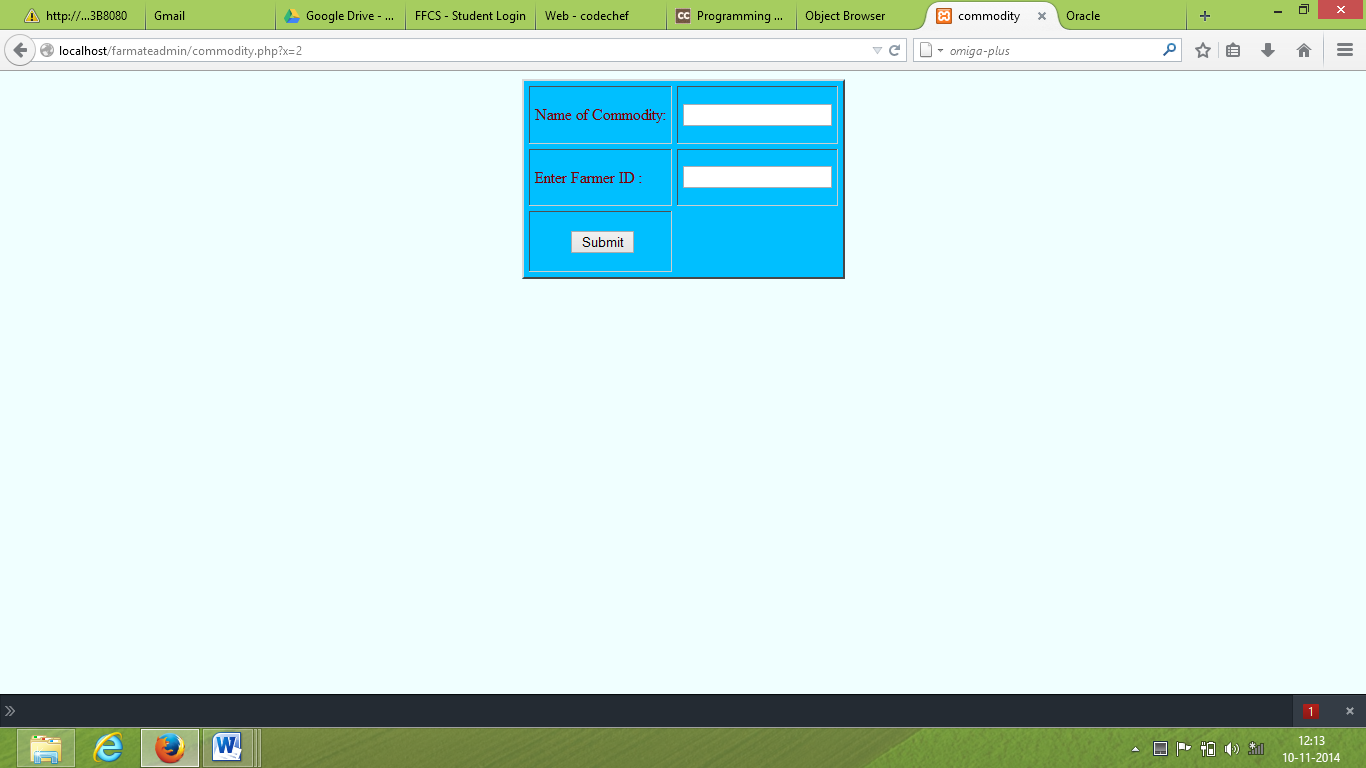


FIGURE 17: DELETE COMMODITY

**VIEW ENTITY PAGE**

This module allows the admin to view the existing records of different enities in the database. The admin can seperatly view the records made in the tables of farmers, market dealers, accessory dealers, accessories or commodities through this page.

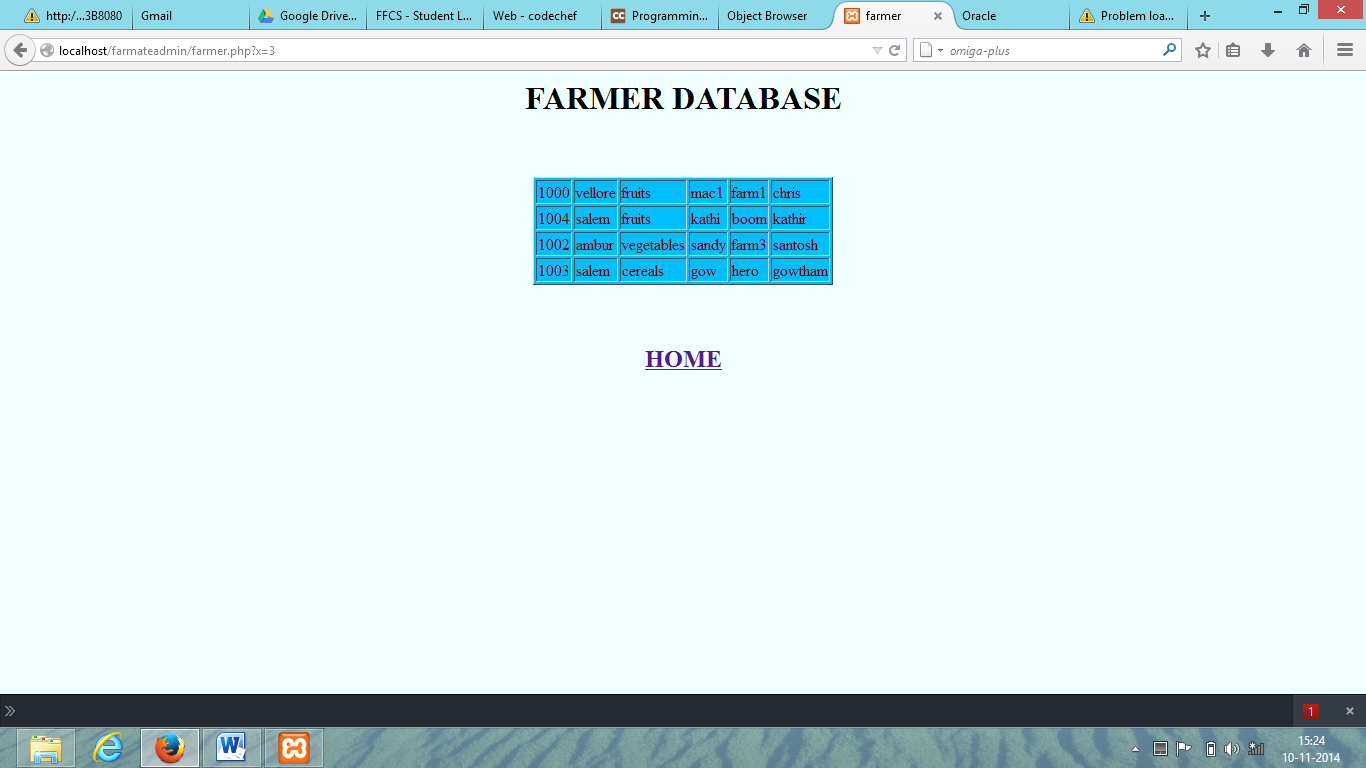


Figure 18: VIEW FARMER DATABASE

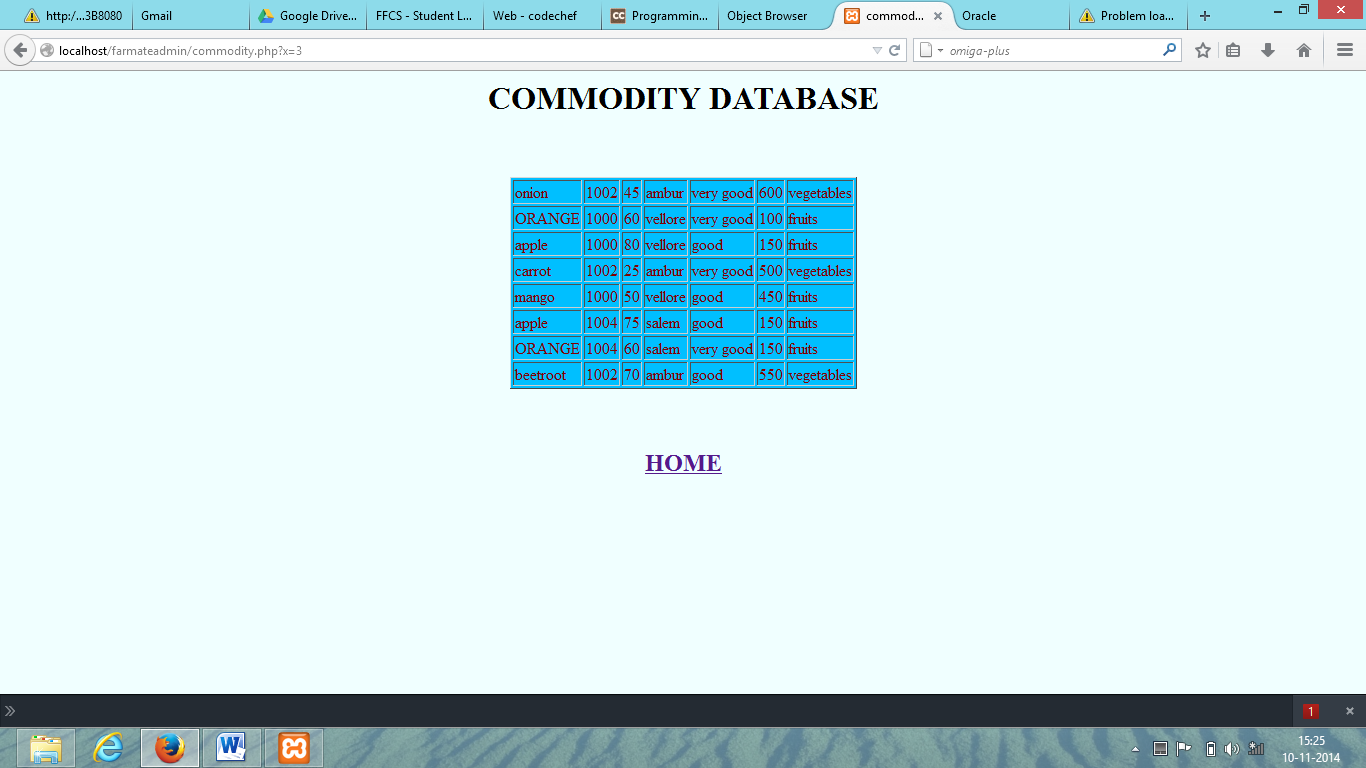


Figure 19: VIEW COMMODITY DATABASE

**IMPORTANT QUERRY PAGE**

This module allows the admin to perform some refined search i.e. complex queries. The functionalities he can perform include-

* Finding the details of a given commodity sold at the minimum rate
* Finding the total of the rates of different commodities sold by every farmer
* Finding details of a particular commodity at a particular location sold below a particular location

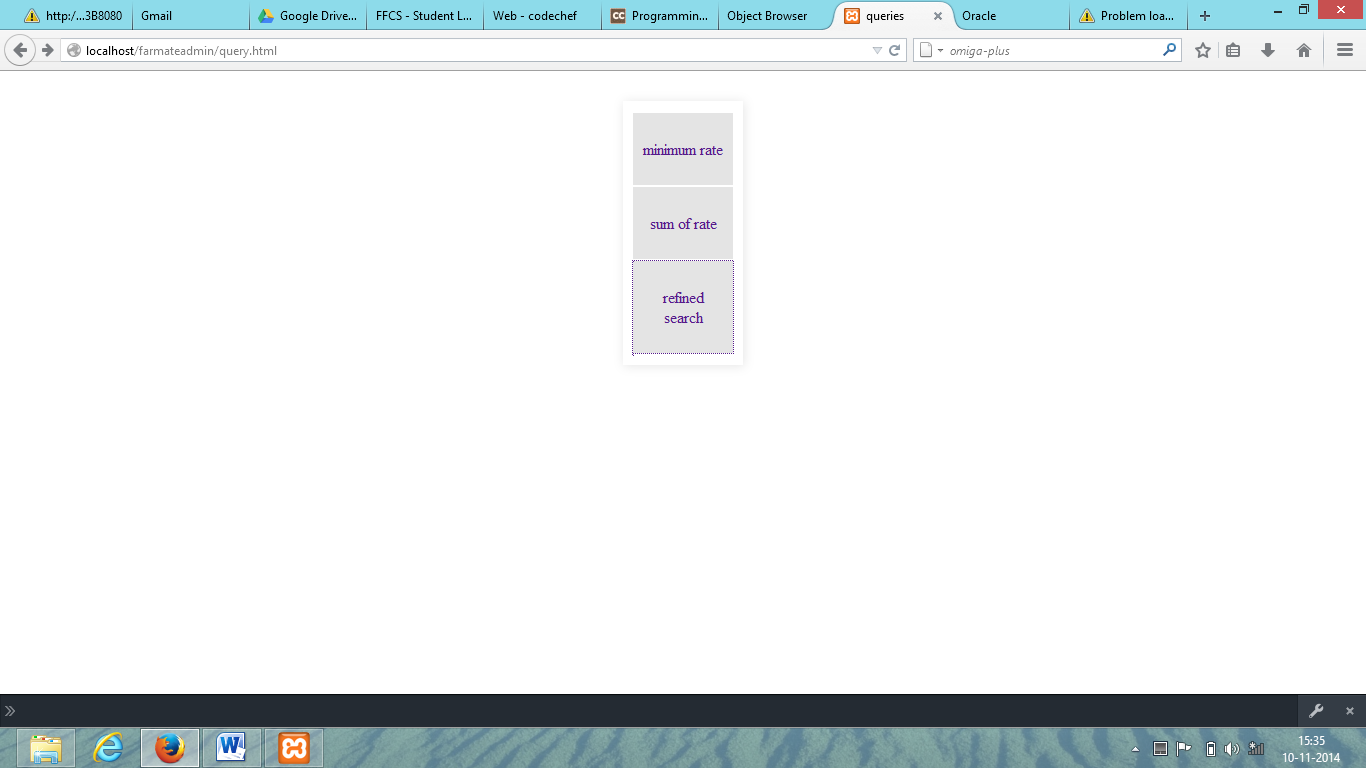


Figure 20:MAIN QUERRY PAGE

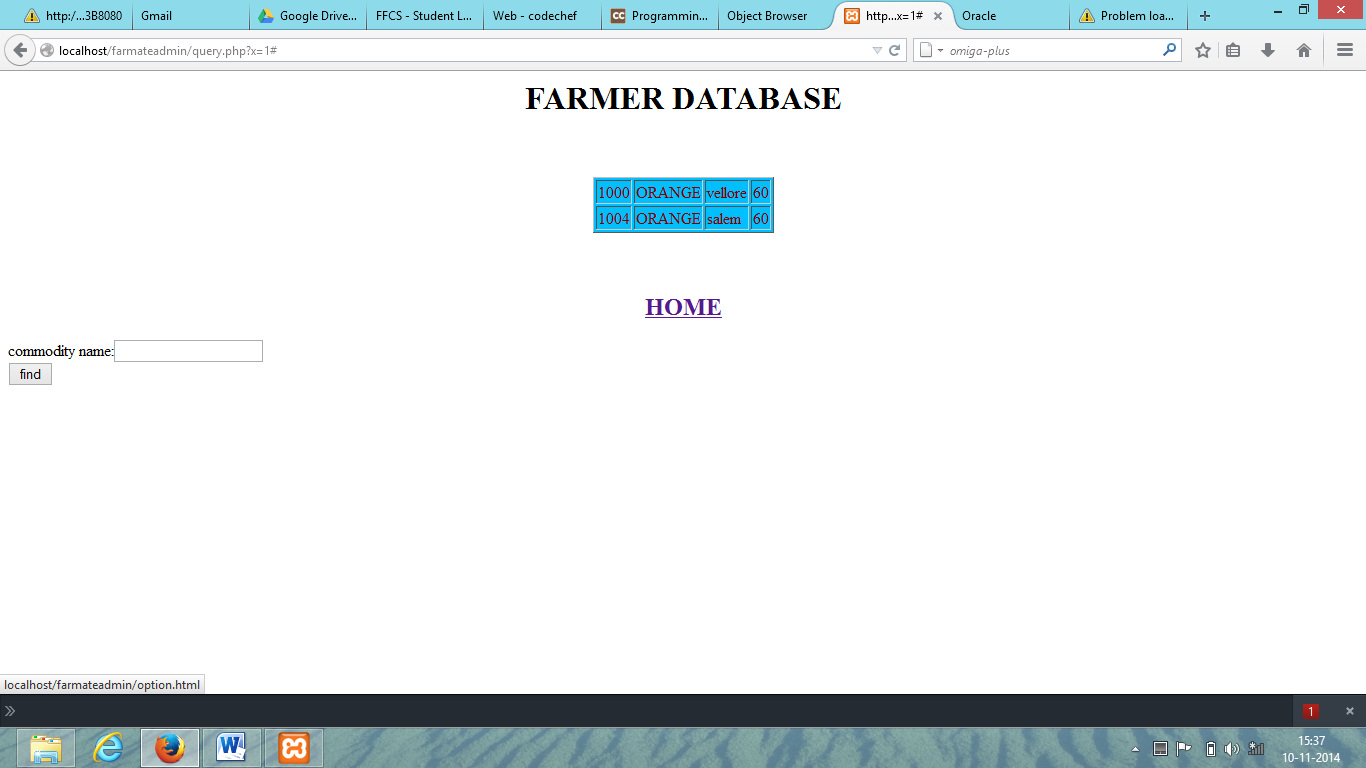


Figure 20: QUERRY-1 PAGE

CONCLUSION:

For the better living condition of the farmers it is very important for the middlemen to be honest. But as this honesty didn’t exist , the existence of middlemen merely added to the misery of farmers rather than being a helpinghand. So the middlemen eventually became a red light crisis. And it became very vital to remove this crisis for the betterment of farmers. And hence , FARMATE successfully serves the purpose. It acts as a handy , trustworthy and reliable link between the farmers and the market dealers or accessory dealers. With its pictorial and user friendly interface, it doesn’t demand much education , making it readily usable to the farmers. Thus the middlemen crisis is eliminated which was the ultimate motto of the project.

FUTURE ENHANCEMENT:

Although the interface is very simple and user friendly, its not that catchy. So UI is a main aspect for future enhancement. Other plans include, integrating transport facilities, secure payment procedures and warehouse facilities to preserve food crops. We are also planning to develop an app to serve the same purpose. We also aim at developing a device with FARMATE as the stand alone app to avoid misuse and also identify each farmer uniquely. This unique identification could also serve other purposes such as voting using internet. We also plan to add features that directly hand over the subsidies and reservations the government has set aside for farmers, thus making the farmers aware of the privileges meant for them;

.

APPENDIX

SQL COMMANDS:

## MARKET DEALER

CREATE TABLE "MARKET\_DEALER"

( "MD\_ID" NUMBER(4,0),

"LOCATION" VARCHAR2(30),

"COMMODITY" VARCHAR2(30),

"USER\_ID" VARCHAR2(30),

"PASSWORD" VARCHAR2(30),

"NAME" VARCHAR2(30),

PRIMARY KEY ("MD\_ID") ENABLE

)

/

### INSERT QUERY

INSERT INTO market\_dealer VALUES ($a,'$b','$c','$d','$e','$f')

## FARMER

CREATE TABLE "FARMER"

( "F\_ID" NUMBER(4,0),

"LOCATION" VARCHAR2(30),

"COMMODITY" VARCHAR2(30),

"USER\_ID" VARCHAR2(30),

"PASSWORD" VARCHAR2(30),

"NAME" VARCHAR2(30),

PRIMARY KEY ("F\_ID") ENABLE

)

/

### INSERT QUERY

INSERT INTO farmer VALUES ($a,'$b','$c','$d','$e','$f')

## ACCESSORY DEALER

CREATE TABLE "ACCESSORY\_DEALER"

( "AC\_ID" NUMBER(4,0),

"LOCATION" VARCHAR2(30),

"COMMODITY" VARCHAR2(30),

"USER\_ID" VARCHAR2(30),

"PASSWORD" VARCHAR2(30),

"NAME" VARCHAR2(30),

PRIMARY KEY ("AC\_ID") ENABLE

)

/

### INSERT QUERY

INSERT INTO accessory\_dealer VALUES ($a,'$b','$c','$d','$e','$f')

## ACCESSORY

CREATE TABLE "ACCESSORY"

( "NAME" VARCHAR2(30),

"ID" NUMBER(4,0),

"RATE" NUMBER(4,0),

"LOCATION" VARCHAR2(30),

"QUALITY" VARCHAR2(30),

"QUANTITY" NUMBER(4,0),

"TYPE" VARCHAR2(30),

FOREIGN KEY ("ID")

REFERENCES "ACCESSORY\_DEALER" ("AC\_ID") ENABLE

)

/

### INSERT QUERY

INSERT INTO accessory VALUES ('$f',$a,$d,'$b','$e','$g','$c')

|  |  |  |
| --- | --- | --- |
|  |  |  |

## COMMODITY

CREATE TABLE "COMMODITY"

( "NAME" VARCHAR2(30),

"ID" NUMBER(4,0),

"RATE" NUMBER(4,0),

"LOCATION" VARCHAR2(30),

"QUALITY" VARCHAR2(30),

"QUANTITY" NUMBER(4,0),

"TYPE" VARCHAR2(30),

FOREIGN KEY ("ID")

REFERENCES "FARMER" ("F\_ID") ENABLE

)

/

### INSERT QUERY

INSERT INTO commodity VALUES ('$f',$a,$d,'$b','$e','$g','$c')

QUERIES USED IN PROJECT:

* select farmer.f\_id,commodity.name,commodity.rate from farmer join commodity on farmer.f\_id=commodity.id where commodity.name='$f' and rate=(select min(rate) from commodity where name='$f')
* select name,rate,quality,id from commodity where name='$f' and location='$g' and $h<100
* select farmer.name,farmer.f\_id,farmer.location,(select sum(rate) as sum from commodity where commodity.id=farmer.f\_id) from farmer

CONNECTIVITY CODING IN <PHP>

//establish connection

$conn = oci\_connect("system","macman\_25");

if (!$conn)

{

die('Could not connect: ' . oci\_error());

}

PL/SQL Code

// CURSORS

declare

cursor c1 is select id,rate,type from commodity;

id commodity.id%type;

a commodity.rate%type;

sts commodity.type%type

num number;

begin

num := &num;

open c1;

loop

fetch c1 into id, a, sts;

exit when c1%notfound;

if (num = id) then

dbms\_output.put\_line(id);

dbms\_output.put\_line(a);

dbms\_output.put\_line(sts);

end if;

end loop;

if (c1%notfound) then

dbms\_output.put\_line('Emp ID not available');

end if;

close c1;

end;

/

// TRIGGERS......

CREATE OR REPLACE TRIGGER "DEL"

before delete on farmer for each row

begin

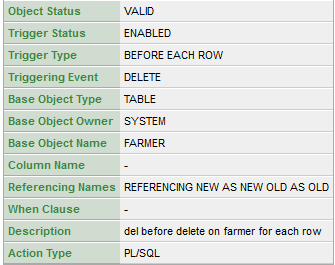
delete from commodity where commodity.id=:old.f\_id;

end;

/

ALTER TRIGGER "DEL" ENABLE

/



EVALUATION SHEET

COURSE CODE:ITE302 COURSE TITLE: DATABASE SYSTEMS

|  |  |  |
| --- | --- | --- |
| *Scheme of Evaluation-Internals 50 Marks* | | |
| ***Component*** | ***MAX Marks*** | ***Marks obtained*** |
| *Review-I [15]*  *(after CAT I)* | *ER Diagram, Relational Schema-10*  *Complete Back End -5*  *[Marks based on Identification of entity ,*  *relationships, keys, creating tables with proper datatypes and constraints]* |  |
| *Review-II [15]*  *(after CAT II)* | *FRONT END(in Java/PHP)-15*  *[Marks based on number of Query forms and*  *Entry forms, Validation for entry forms and*  *handling backend(ODBC/JDBC) errors]* |  |
| *Review- III [20]*  *(before last working day)* | *PL/SQL-10*  *[Marks based on appropriate and efficient usage of functions and triggers]*  *Document-10*  *[Filing together Abstract , Module Description, ERdiagram, Relational Schema, Back end details,Front end details , form snapshots, conclusion and future enhancement]* |  |
| *Total (50)* | |  |
| *FACULTY SIGNATURE*  *DATE* | |  |