VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA” BELAGAVI-590018



2018-2019

A

DBMS Mini Project Report

On

FARMBOOK

Submitted in partial fulfillment of the requirements in 5th Semester of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

By

Chandan P

[1JB16CS034]

UNDER THE GUIDANCE OF

**Dr.Srikantaiah K C**

**Professor**

**Department of CSE**

**SJBIT, Bengaluru**

**Charulata M**

**Assistant Professor**

**Department of CSE**

**SJBIT, Bengaluru**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### S. J. B INSTITUTE OF TECHNOLOGY

B G S HEALTH AND EDUCATION CITY

Kengeri, Bengaluru-560060.

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

### S. J. B INSTITUTE OF TECHNOLOGY

### BGS Health & Education City, Kengeri, Bengaluru-560060.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

Certified that the DBMS mini project work entitled “farmbook” carried out by Chandan P [1JB16CS034] is a bonafide student of S J B Institute of Technology in partial fulfillment for the award of “BACHELOR OF ENGINEERING” in COMPUTER SCIENCE AND ENGINEERING as prescribed by VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI during the academic year 2018-2019. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work.

Signature of Guide

**Dr.Srikantaiah K C**

Professor

Dept. of CSE, SJBIT

Signature of HOD

**Dr. Krishna A N**

Professor and Head

Dept. of CSE, SJBIT

Signature of Guide

**Charulata M**

Assistant Professor

Dept. of CSE, SJBIT

EXTERNAL VIVA

Name of the Examiners Signature with Date

1.

2.





ACKNOWLEDGEMENT

I would like to express our profound gratefulness to His Divine Soul Jagadguru Padmabhushan **Sri Sri Sri Dr.Balagangadharanatha Maha Swamiji** and His Holiness Jagadguru **Sri Sri Sri Dr.Nirmalanandanatha Maha Swamiji** for providing us an opportunity to complete our academics in this esteemed Institution.

I would like to express my profound thanks to Revered **Sri Sri Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this project in this admired institution.

I express my gratitude to **Dr.Puttaraju**, Principal, SJB Institute of Technology, for providing me excellent facilities and academic ambience, which have helped me in the satisfactory completion of the mini project work.

I extend my sincere thanks to **Dr.Krishna A N**, Professor and Head of Department of Computer Science and Engineering, SJB Institute of Technology, for providing me invaluable support throughout the period of my mini project work.

I express my deepest gratitude and sincere thanks to mini project guides **Dr.Srikantaiah K C**, Professor, and **Charulata M** ,Assistant Professor, Department of Computer Science and Engineering, SJB Institute of Technology, Bengaluru, for their valuable guidance and support.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, the teaching & non-teaching staff of the department, the library staff and all my friends, who have directly or indirectly supported me during the period of my mini project work.

Regards,

Chandan P

(1JB16CS034)

## 

# **Abstract**

Today, there’s a lot of difference between the price at which a consumer buys a food product and the amount given to the farmers by the middle men. These middle men have been consuming most part of the profit, making farmers lives more miserable day by day.

This project called ‘farmbook’ aims at forming a direct between the consumers and the farmers thus eliminating the unnecessary middle men. Now, farmers can have greater profits. This project also aims at giving farmers a lot of details and information regarding the crop they are growing or the one they want to in the future.

So this website could reduce or even stop the larger number of farmer suicides and also lift the farmers in the below poverty line column.

* **Acknowledgement………………………………………………..i**
* **Abstract…………………………………………………………...ii**

**Table of Contents**

**List of figures………………………………………………….0**

Chapter 1 **Introduction………………………………………..1**

1.1 Data, Database and DBMS…………………………………1

1.2 Components of a DBMS……………………………………1

1.3 Relational Data model………………………………………2

1.4 ER to Relation model mapping algorithm……………….….2

1.5 Normalization……………………………………………….4

1.6 About Farmbook…………………………………………….4

1.7 Motivation…………………………………………………...5

1.8 Applications…………………………………………………6

Chapter 2 **Literature Survey…………………………………..7**

Chapter 3 **Design………………………………………………..9**

3.1 Entities, attributes and their domains……………………….9

3.2 Relationship types…………………………………………..10

3.3 ER diagram………………………………………………….11

3.4 Relational schema…………………………………………...11

3.5 Normalization……………………………………………….12

Chapter 4 **System Requirements……………………………..13**

4.1 HTML5……………………………………………………..13

4.2 JSP………………………………………………………….13

4.3 SQL…………………………………………………………14

4.4 Required software…………………………………………..15

4.5 Constraints…………………………………………………15

Chapter 5 **Implementation……………………………………16**

5.1 SQL………………………………………………………..16

5.2 HTML……………………………………………………..20

* 1. Connectivity……………………………………………….25

5.4 Triggers and stored procedures……………………………25

Chapter 6 **Snapshots…………………………………………..27**

6.1 Index Page…………………………………………………27

6.2 Farmers Page………………………………………………28

6.3 Admin Page………………………………………………..29

**Conclusion……………………………………………………31**

**References……………………………………………………32**

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig no.** | **Descriptions** | **Page no.** |
| 1.1 | Components of a DBMS | 1 |
| 3.1 | ER Diagram | 11 |
| 3.2 | Schema Diagram | 11 |
| 5.1 | farmer table | 16 |
| 5.2 | crop table | 17 |
| 5.3 | area table | 18 |
| 5.4 | ratio table | 19 |
| 5.5 | orders table | 19 |
| 5.6 | deletedusers table | 20 |
| 6.1 | Index page | 26 |
| 6.2 | About section | 27 |
| 6.3 | Farmers page | 27 |
| 6.4 | Admin page | 28 |

Chapter 1

**Introduction**

**1.1 Data, Database and DBMS**

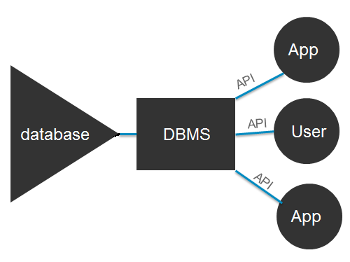
Data is a known fact that can be recorded. It can also be defined as an information that has been translated into a form that is efficient for movement or processing. Eg: name, USN, address, crop name, etc. A datum is a unit of data. Meaningful data combines to form information. Hence, information is interpreted data i.e. data provided with semantics

Database is a collection of related data without an implicit meaning. It can also be defined as a structured set of data held in a computer, especially one that is accessible in various ways. Eg. student database, farmer database, employee database, etc.

Database management system (DBMS) is a collection of programs that enable users to create and maintain the database. It is a general purpose software which contains defining, constructing, manipulating, sharing, protecting the database. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, MySQL, Microsoft ACCESS, or EXCEL to store data in the form of database.

Database systems are meant to handle large collection of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

**1.2 Components of a DBMS**



**Fig.1.1 Components of a DBMS**

* Users: Users may be of any kind, such as database administrators, system developers or database users.
* Database application: Database application may be Departmental, Personal, Organizational and /or Internal.
* DBMS: Software that allows users to create and manipulate database access.
* Database: Collection of logical data as a single unit.

**1.3 Relational Data Model**

The relational model (RM) for [database](https://en.wikipedia.org/wiki/Database) management is an approach to managing [data](https://en.wikipedia.org/wiki/Data) using a structure and language consistent with [first-order predicate logic](https://en.wikipedia.org/wiki/First-order_logic), first described in 1969 by English computer scientist [Edgar F. Codd](https://en.wikipedia.org/wiki/Edgar_F._Codd), where all data is represented in terms of [tuples](https://en.wikipedia.org/wiki/Tuple), grouped into [relations](https://en.wikipedia.org/wiki/Relation_(database)). A database organized in terms of the relational model is a [relational database](https://en.wikipedia.org/wiki/Relational_database).

The purpose of the relational model is to provide a [declarative](https://en.wikipedia.org/wiki/Declarative_programming) method for specifying data and queries: users directly state what information the database contains and what information they want from it, and let the database management system software take care of describing data structures for storing the data and retrieval procedures for answering queries.

Most relational databases use the [SQL](https://en.wikipedia.org/wiki/SQL) data definition and query language; these systems implement what can be regarded as an engineering approximation to the relational model. A table in an SQL database schema corresponds to a predicate variable; the contents of a table to a relation; key constraints, other constraints, and SQL queries correspond to predicates. However, SQL databases [deviate from the relational model in many details](https://en.wikipedia.org/wiki/Relational_model#SQL_and_the_relational_model), and Codd fiercely argued against deviations that compromise the original principles.

**1.4 ER to Relational Model Mapping algorithm**

This algorithm consists of 7 steps:

1. Mapping strong entity to relations:

For each strong entity type ‘E’ create a table or relation R that includes all the simple attributes of E. Since the strong entities will have their own primary keys, those attributes will become the keys for respective table.

1. Mapping weak entity to relations:

For each weak entity ‘W’ in the ER diagram, with owner entity type ‘E’, create a new table or relation and include all simple attributes of W as attributes of R. In addition, include primary key attribute of the relation that corresponds to owner entity as foreign key of R. The primary key of R is the primary key of owner entity and partial key of weak entity.

1. Mapping binary 1:1 relationship types:

For each binary 1:1 relationship type ‘R’ in the ER diagram, identify the relations S and T that corresponds to entity type participates in R. There are three approaches:

* Foreign key approach: Choose of the relations S (total participation side) and include primary key of T as foreign key of S. Include all the simple descriptive attributes of 1:1 relationship types as the attributes of S.
* Merge relationship approach: In this approach, merge the two entity types and relationship types into a single relation. This maybe appropriate when both participations are total.
* Cross reference / relationship approach: In this approach, setup a new relation R for the purpose of cross-functioning the primary keys of S and T.

1. Mapping binary 1:N relationship types:

For each 1:N relationship type R, identify the relation S that represents the participating entity type at the N side of relationship type. Include primary key of S as he foreign key of T and also include all descriptive attributes.

1. Mapping binary M:N relationship types

For each M:N relationship R, create a new relation S to represent R. Include primary keys of the relations that represent the participating entity types as foreign keys of S and their combination will form the primary key for S and also include all the simple descriptive attributes of R.

1. Mapping multivalued attributes:

For each multivalued attribute ’A’, create a new relation and include an attribute corresponding to A and primary key ‘k’ of the participating entity or relationship. The primary key of new relation is combination of A and k.

1. Mapping of n-ery relationship type:

For n-ery relationship type R, n>2, create a new relation to represent R. Include primary keys of all participating entity types as foreign keys and also include all descriptive attributes of R, The primary key of newly created relation is the combination of all the foreign keys.

**1.5 Normalization**

It is a design for producing a set of good tables with all constraints incorporated as per the business constraints. Normalization process takes a relation schema through a series of tests to verify whether it satisfies a certain normal form. Normalization of relations is based on the functional dependencies and primary keys to achieve desirable properties of minimizing redundancy and minimizing update anomaly. Types of normal form are:

1. First Normal Form (1NF)

A relation schema R is in 1NF if every attribute of R takes only a single value. When a table contains multivalued attribute, we say that its not in first normal form. We identify multivalued attributes and remove using following techniques:

1. Use multiple tuples one per value
2. Use multiple columns one per value
3. Use a separate table
4. Second Normal Form (2NF)

A relation schema R is in 2NF if it satisfies 1NF and also if all non-prime attributes A in R should be fully functionally dependent on primary key in R. The test for 2NFinvolves testing for functional dependencies whose LHS attribute are part of primary key. If primary key contains a single attribute, the test need not be applied. If a relation schema is not in 2NF, it can be second normalized into a number of 2NF relations in which non-prime attributes are associated only with the part of the primary key on which they are fully functionally dependent.

1. Third Normal Form (3NF)

A relation schema R is in 3NF if it is in 2NF and no non-prime attributes of R is transitively dependent on primary key of R. A functional dependency X->Y is a transitive dependency if there’s a set of attributes Z that is not a subset of any key of R and both X->Z and Z->Y holds.

**1.6 About Farmbook**

Farmbook is a platform for the farmers to sell the crops they grow by making an order online. It also gives farmers necessary details about their crops, the area in which it is grown, the soil types, the right amount of nutrition which needs to be provided to the crops and many more. It also gives details regarding the current market price for a particular crop as well as the price Farmbook is going to offer.

The main aim is to eradicate the middle men as much as possible and also digitalizing the market process for farmers. It also necessary for the farmers to know about the soil in which they grow a crop. They also need to know about the crop that they are growing and also about various nutritional requirements.

There is also an admin login option wherein the admin only has access to it and has extra privileges like viewing the tables in the database and also deleting users directly. The admin can also update the prices of the crops.

With this simple UI and an efficient database, it is possible to change the way farmers’ market is seen in the near future. Farmers will now have greater profits and the consumers will have to pay less to the same product, since all the middlemen have been eliminated and Farmbook is the only direct link between the farmers and the consumers. Welcome to Farmbook!

**1.7 Motivation**

The main source of motivation for this project comes mainly from the farmers and partly from the customers. The farmers work really hard and spend all the energy and resources they have got to grow crops. But because of the large number of middlemen, the farmers are paid a low amount of price for their growth but the crops are sold to the consumers at high amount of price.

In order to eliminate this problem, Farmbook is created. In order to eliminate all the middlemen Farmbook is the best solution. Now, farmers will be paid with better prices and also the consumers will be paying less price for the same crop and same amount.

This also makes selling process for the farmers digitalized. They will now have spend less than five minutes to sell their crops. They can also grow better crops in the same region because Farmbook provides necessary information about the place, the soil as well as the crop.

A large number of farmers are committing suicide because of unpaid loans and lower prices for their crops. Also a large number of farmers are entering the table of below poverty line citizens. The poor are becoming even more poor. So in order to stop this, Farmbook was created.

**1.8 Applications**

The main application of this mini project will be on the farmers. The easy to use user interface will make the farmers comfortable while using the website. The strong and efficient database and backend will also make the website smooth running. The main application of this mini project comes into picture before and after growing the crops.

Before growing the crops, the farmers need to know about the place, crop and soil type. They can also get the nutrient requirement for their crops and hence choose the appropriate fertilizer.

After growing the crops, the farmers can sell their grown crops easily and for a better price using Farmbook. Once the farmer places the order, a phone call from Farmbook will be made and later the crops will be collected and the farmers will be paid on the spot.

Thus through this application, the farmers in the below poverty line can be lifted to a better place. The number of farmers committing suicide can be reduced or even stopped if a good cooperation comes into picture. And also, the main motive of Farmbook will be accomplished.

Chapter 2

**Literature Survey**

This explains about the similar projects like Farmbook. There are two projects similar to Farmbook:

* Farmer Portal (https://farmer.gov.in/ )

This is a government of India website that gives farmers many useful options like crop management, risk management, exports and imports, census on livestock, map views, weather details, diseases and symptoms and many others options. There are many advantages and disadvantages of using this website like :

1. Advantages :

* It provides better details about the weather.
* It provides information on a broader basis using the census.
* It also has the feature of extreme weather alert and map view.
* It has crop management and post-harvest options which gives farmers better management over their crops as well the harvest.
* It has details about major crops.
* It also hosts programs and schemes related to farmers and the information about these activities is displayed in the ‘Programmes & Schemes’section.
* It has insurance options.
* Easy to use interface.

1. Disadvantages :

* The main disadvantage is that it doesn’t cover farmers growing a variety of crops as the details of only the major crops are given.
* It has information regarding only 6 crops.
* It has only the information about many topics and no topic in detail.
* It doesn’t have an option for farmers to sell their crops directly to the market. So this won’t change anything about farmers’ lives or the price that their crops are paid for by the middlemen.
* FarmersWeb (<https://www.farmersweb.com/> )

FarmersWeb is a farm management software for family and farm cooperatives. FarmersWeb helps farmers to inform their buyers about product availability. The advantages and disadvantages of this website are:

1. Advantages:

* It has an easy option for farmers to inform their buyers.
* It manages orders efficiently and keeps online records
* It manages multiple forms of payments.
* Multiple custom pickup points and delivery zones can be set.
* Simple and easy to use, minimalist user interface.

1. Disadvantages:

* This website doesn’t eliminate the middlemen and hence farmers’ crops will still be paid with a low price.
* Only the first one month is free. Later, the website charges the farmers with a specific amount of money based on the features that the farmer wants.
* Since this website coordinates farmers to the dealers, this website is itself another dealer/middleman.

Chapter 3

**Design**

**3.1 Entities, Attributes and their Domains**

The following are the entities ,attributes and their respective domains:

1. FARMER

Farmer is an entity which holds the details of a farmer when he registers into the website. The various attributes and the domains are:

|  |  |
| --- | --- |
| **Attribute** | **Domain** |
| name | varchar(30) |
| adrnum | bigint(12) |
| gender | varchar(6) |
| dob | date |
| place | varchar(15) |
| address | varchar(30) |
| uname | varchar(20) |
| password | varchar(20) |
| phone | bigint(10) |

1. CROP

The crop entity holds the details of various commonly grown crops. The attributes and domains of this entity are:

|  |  |
| --- | --- |
| **Attribute** | **Domain** |
| name | varchar(20) |
| type | varchar(20) |
| Scientific\_name | varchar(25) |
| market\_price | decimal(9,2) |
| our\_price | decimal(9,2) |

1. ORDERS

This entity has the information regarding the orders placed by the farmers. The various attributes and their domains are:

|  |  |
| --- | --- |
| **Attribute** | **Domain** |
| ordno | int |
| cropname | varchar(20) |
| quantity | mediumint |
| amount | bigint |

1. RATIO

The ratio entity consists of many information about nutrients and the ratios in which they are to be applied. The attributes and their domains are:

|  |  |
| --- | --- |
| **Attribute** | **Domain** |
| id | tinyint |
| azone | varchar(40) |
| soil | varchar(45) |
| ratios | It’s a multivalued attribute. |

**3.2 Relationship types**

|  |  |  |
| --- | --- | --- |
| **Relationship** | **Cardinality** | **Participation** |
| GROWS  -FARMER, CROP | M:N | Total participation on ‘FARMER’ side. Partial participation on ‘CROP’ side. |
| MAKES ORDER  -FARMER, ORDERS | N:1 | Partial participation on both the sides. |
| NEEDS NUTRIENT RATIOS  -CROP, RATIO | 1:1 | Total participation on both the sides. |

These are the various relationships, their cardinality and the participation. The relationships, cardinality ratios and participation are important factors while drawing an ER diagram.

Based on section 3.1 and 3.2, we’ll now place an ER diagram in section 3.3 .

**3.3 ER Diagram**

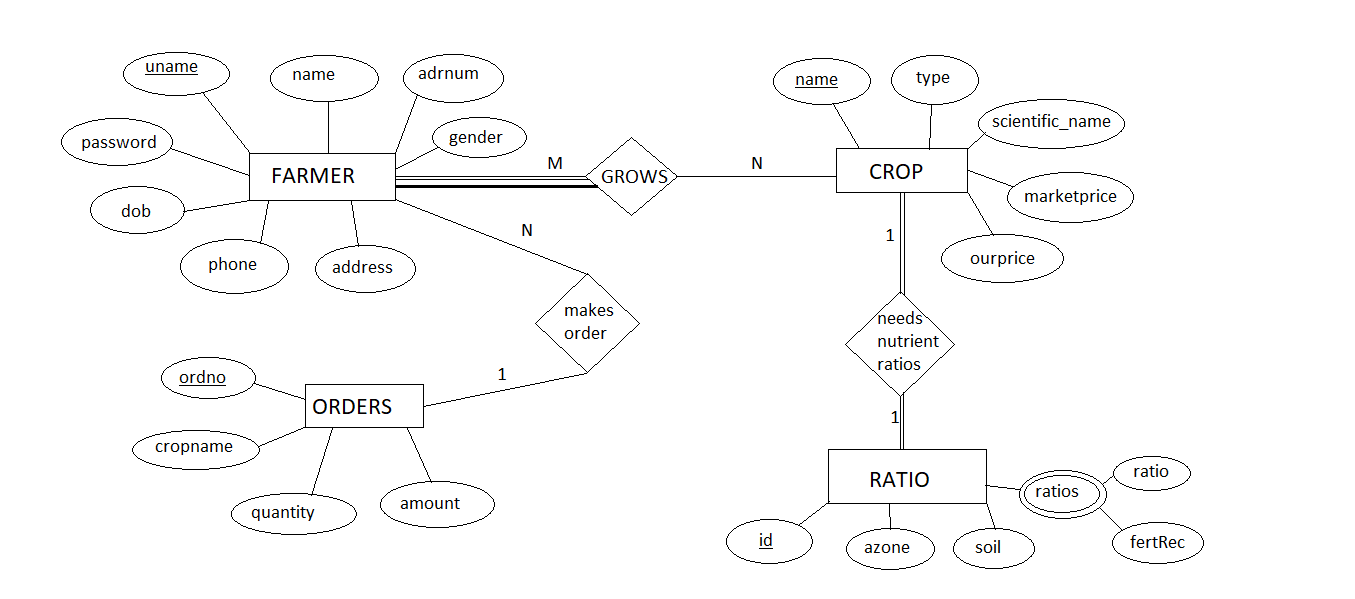


Fig.3.1 ER diagram

Now that we have the ER diagram ready, a relational schema can be easily obtained.

**3.4 Relational Schema Diagram**

FARMER

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | adrnum | gender | dob | place | Address | uname | password | phone |

CROP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | type | scientific\_name | marketprice | ourprice |

ORDERS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ordno | cropname | quantity | amount | uuname |

AREA

|  |  |  |
| --- | --- | --- |
| id | azone | soil |

RATIOS

|  |  |  |  |
| --- | --- | --- | --- |
| areaID | crop | ratio | fertRec |

Fig 3.2 Schema diagram

This is the relational schema obtained after applying the ER to relational model mapping algorithm.

**3.5 Normalization**

This section involves applying the first normal form, second normal form and third normal in order to normalize the tables.

* Applying First Normal Form (1NF)

After applying the first normal form the multivalued attributes are removed. In the Ratio table, there was a multivalued attribute called which had two sub-attribute ‘ratio’ and ‘fertRec’. It is done by forming separate tables. The first table will have all the attributes of ‘Ratios’l table. The newly formed table will have the sub-attributes of the multivalued attribute along with the primary key of the ‘Ratios’ table. Thus the tables formed are:

AREA

|  |  |  |
| --- | --- | --- |
| Id | azone | soil |

RATIOS

|  |  |  |  |
| --- | --- | --- | --- |
| areaID | crop | ratio | fertRec |

* Applying Second Normal Form (2NF)

In the second normal form, all the non-prime attributes should fully functionally depend on the primary key in the relation schema. Since all our tables except the Ratios table have only one primary key, the test need not be applied. The ‘ratio’ and ‘fertRec’ attributes are already fully functionally dependent on the primary keys of the ‘RATIOS’ table (formed in the above step). So this passes the 2NF.

* Applying Third Normal Form (3NF)

Since the non-prime attributes in all the schemas are not transitively dependent on the primary key of the relational schema, our relational schema passes 3NF too.

Chapter 4

**System Requirements**

### 4.1 HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard.

It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new video, audio and canvas elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as <main>, <section>, <article>, <header>, <footer>, <aside>, <nav> and <figure>, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite> and <menu> have been changed, redefined or standardized. The APIs and Document Object Model (DOM) are now fundamental parts of the HTML5 specification and HTML5 also better defines the processing for any invalid document.

### 4.2 Java Server Side Pages (JSP)

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP, but it uses the Java programming language. To deploy and run Java Serves Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime, therefore JSP is a Servlet; each JSP servlet is cached and re-used until the original JSP is modified. JSP can be used independently or as the view component of a server-side model-view-controller design, normally with JavaBean as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other Java program, they must be executed with a Java Virtual Machine (JVM) that interacts with the server’s host operating system to provider an abstract, platform-neutral environment.

A Java Server Pages compiler is a program that parses JSPs and transforms them into executable Java Servlets. A program of this type is usually embedded into the application server and run automatically the first time a JSP is accessed, but pages may also be precompiled to test for errors.

Some JSP containers support configuring how often the container checks JSP file timestamps to see whether the page has changed. Typically, this timestamp would be set to a short interval (perhaps seconds) during software development, and a longer interval (perhaps minutes, or even never) for a deployed Web application.

The JSP has many advantages like extension to servlet, easy to maintain, fast development (no need to recompile and redeploy), less code than servlet. The lifecycle of a JSP page follows: translation of JSP page, compilation of JSP page, class-loading, instantiation, initialization, request processing, destroy.

### 4.3 SQL (Structured Query Language)

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like ISAM or VSAM, SQL offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks".[[10]](https://en.wikipedia.org/wiki/SQL#cite_note-codd-relational-model-10) Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

**4.4 Required Software**

1. For HTML and JSP: Eclipse
2. Server: Apache 10.0
3. Operating System: Windows 10
4. Database Support: MySQL 8.

**4.5 Constraints**

The user or the farmer has to login first in order to use the various features available.

If the user is new, then the user will have to fill the registration form first and then login to use the features. All the details in the registration form are compulsory to fill.

Once the farmer places order for his crops to get picked up for sale, the farmer will be given an order number. The farmer will have to enter the same order number in order to cancel the order.

Chapter 5

**Implementation**

**5.1 SQL**

This section includes the create table commands and the insert commands for all the tables used in Farmbook. They are:

1. ‘farmer’ table

mysql> create table farmer (

name varchar(30) not null,

adrnum bigint not null,

gender varchar(6) not null,

dob date not null,

place varchar(15) not null,

address varchar(30) not null,

uname varchar(20) primary key,

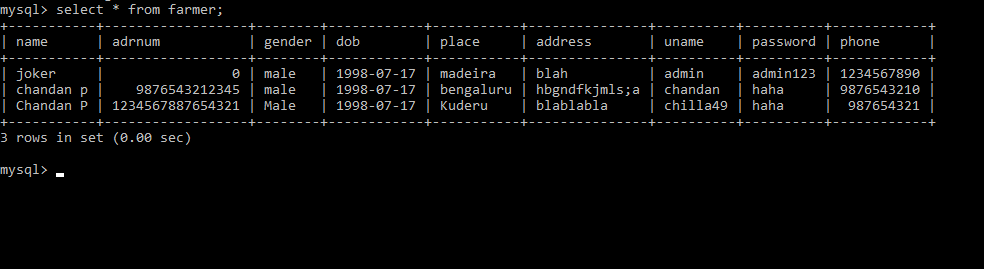
password varchar(20) not null,

phone bigint not null

);

mysql> insert into farmer values(‘chandan p’,9876543212345,’male’,’1998-01-17’,’kuderu’,’No.14,street 11, Vijayanagar,Bangalore’,’chilla49’,’haha’,9876543210);

The following figure (Fig.5.1) shows the contents of farmer table after insertion.



**Fig.5.1 farmer table**

1. ‘crop’ table

mysql> create table crop (

name varchar(20) primary key,

type varchar(20),

scientific\_name varchar(25),

marketprice dec(9,2),

ourprice dec(9,2)

);

mysql> insert into crop values(‘cotton’,’cash crop’,’gossypium’,65.50,78.30);

The following figure (Fig.5.2) shows the contents of crop table after insertion.



**Fig.5.2 crop table**

1. ‘area’ table

mysql> create table area (

id tinyint primary key,

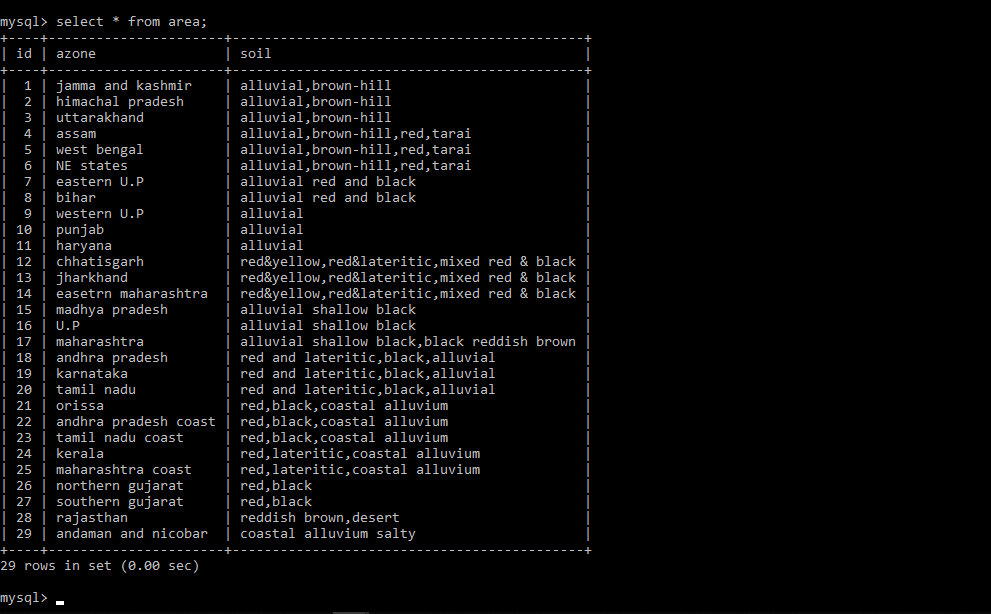
azone varchar(40),

soil varchar(45)

);

mysql> insert into area values(1,’jammu and kashmir’,’alluvial,brown-hill’);

The following figure (Fig.5.3) shows the contents of area table after insertion.



**Fig.5.3 area table**

1. ‘ratios’ table

mysql> create table ratios (

areaID tinyint references area(id),

crop varchar(20) references crop(name),

ratio varchar(15),

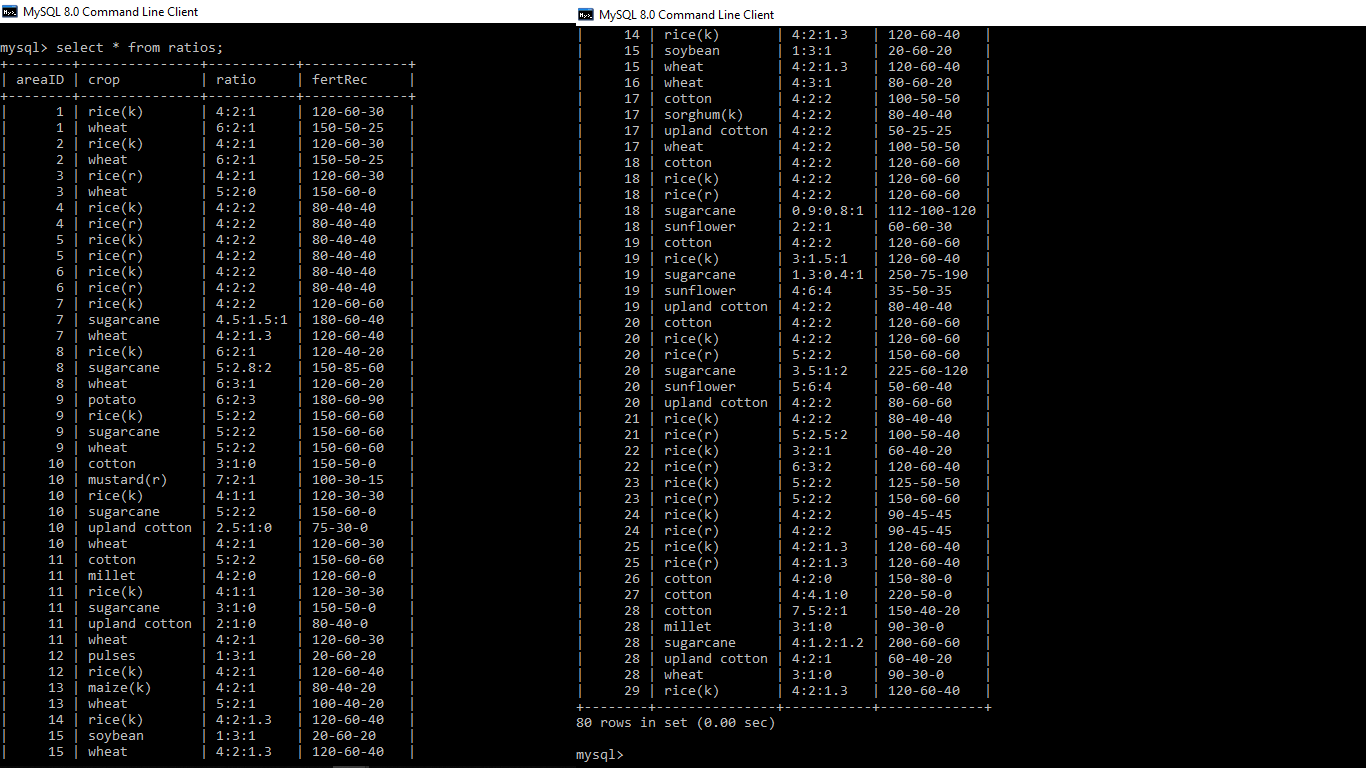
fertRec varchar(15),

primary key(areaID,crop)

);

mysql> insert into ratios values(1,’rice(k)’,’4:2:1’,’120-60-30’);

The following figure (Fig.5.4) shows the contents of ratios table after insertion.



**Fig.5.4 ratios table**

1. ‘orders’ table

mysql> create table orders (

ordno int primary key,

cropname varchar(20) references crop(name),

quantity mediumint,

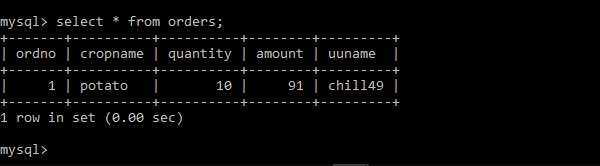
amount bigint,

uuname varchar(30) refrences farmer(name)

);

mysql> insert into orders values (1,’potato’,10,91,’chilla49’);

The following figure (Fig.5.5) shows the contents of orders table after insertion.



**Fig.5.5 orders table**

1. ‘deletedusers’ table

mysql> create table deletdusers (

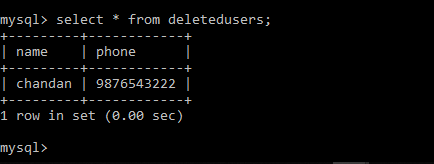
name varchar(30) references farmer(name),

phone bigint references farmer(phone)

);

mysql> insert into deletedusers values(‘chandan’,9876543222);

The following figure (Fig.5.6) shows the contents of area table after insertion.



**Fig.5.6 deletedusers table**

**5.2 HTML**

**5.2.1 Index page**

The index page has two form- one for registration and one form login. The following code shows how to create a form:

<form action="reg.jsp" method="post">

<p>name:</p>

<input type="text" name="name" placeholder="type your name" class="w3-input w3-round-xxlarge" required>

<p>aadhar number:</p>

<input type="number" name="adrnum" placeholder="your aadhar number" class="w3-input w3-round-xxlarge" required>

<p>gender</p>

<input type="text" name="gender" placeholder="male/female/other" class="w3-input w3-round-xxlarge" required>

<p>dob:</p>

<input type="date" name="dob" placeholder="yyyy-mm-dd format please" class="w3-input w3-round-xxlarge" required>

<p>create unique username</p>

<input type="text" name="uname" placeholder="test your creativity skills" class="w3-input w3-round-xxlarge" required>

<p>Password</p>

<input type="password" name="password" placeholder="secrettt!" class="w3-input w3-round-xxlarge" required><br><br>

<input type="submit" name="loginButton" value="Create!" class="w3-button w3-round-xxlarge w3-block w3-black">

</form>

Similarly, the login form is also created.

**5.2.2 Registration page(jsp)**

The registration page is invoked by the registration form. The following code gets the parameters from the form:

<%@page import="java.sql.\*,javax.sql.\*" %>

<%

String name=request.getParameter("name");

String adrnum=request.getParameter("adrnum");

String gender=request.getParameter("gender");

String dob=request.getParameter("dob");

String uname=request.getParameter("uname");

String password=request.getParameter("password");

The following code creates a statement and resultset. Later the values are placed into the farmer table.

Statement st=connection.createStatement();

ResultSet rs;

**int** i=st.executeUpdate("insert into farmer values('"+name+"','"+adrnum+"','"+gender+"','"+dob+"',''"+uname+"','"+password+"' ')");

**5.2.3 Login page**

The login page is also a jsp code which is invoked from the login form in the index page. The following code matches the entered credentials with the values in the database and if they are correct, then it redirects to the farmers page for normal users and into the admin page for an admin.

if(rs.next())

{

if(rs.getString(8).equals("admin123"))

{

out.println("Welcome "+uuname+". . . Loading . . .");

//waits for 3seconds and then loads info.html

response.setHeader("Refresh","2;url=admin.jsp");

//response.sendRedirect("info.html");

}

else if(rs.getString(8).equals(upassword))

{

out.println("Welcome "+uuname+". . . Loading . . .");

//waits for 3seconds and then loads info.html

response.setHeader("Refresh","3;url=farmers.jsp");

//response.sendRedirect("info.html");

}

else

{

out.println("invalid password!Try again. . . Going back to HOME");

response.setHeader("Refresh","3;url=index.html");

}

}

else

{

out.println("invalid username!Try again. . . Going back to HOME");

response.setHeader("Refresh","3;url=index.html");

}

**5.2.4 Farmers page**

This page contains many forms like the ‘compare prices’ form, ‘know your area info’ form, amount calculator form, form to place order and a form to cancel order. The following code snippet is for the first form. The rest of forms too take in inputs in a similar way but with the values changed. A frame is used to display the results after the submit button is clicked.

<iframe name="frame2" height=100 width=100%></iframe><br>

<form action="prices.jsp" method="post" target="frame2">

Crop name<select name=cropname >

<% while(rs2.next()){ %>

<option value="<%= rs2.getString(1) %>" ><%= rs2.getString(1) %></option>

<% } %></select><br><input type="submit" value="see prices" ></form>

**5.2.5 JSP page to calculate prices**

The following code is used to fetch the marketprice and the ourprice attribute values from the database and display them.

ResultSet rs=st.executeQuery("select marketprice,ourprice from crop where name='"+cropname+"'");

rs.next();

float marketprice= rs.getFloat(1);

float ourprice= rs.getFloat(2);

%>

Market price:Rs.<%=marketprice %>

ourprice price:Rs.<%=ourprice %>

**5.2.6 JSP page which displays the crop information**

The following code snippet displays the crop informations like crop name, ratio of nutrients and fertilizer recommendation.

ResultSet rs=st.executeQuery("select id,soil from area where azone='"+azone+"'");

rs.next();

String id=rs.getString(1);

String soil=rs.getString(2);

%>

<h5 style="color:white;">Soil type : <%=soil %></h5>

<%ResultSet rs1=st.executeQuery("select crop,ratio,fertRec from ratios where areaId='"+id+"'");

**while**(rs1.next())

{

%><tr>

<th><label><%= rs1.getString(1)%></label></th>

<th><label><%= rs1.getString(2)%></label></th>

<th><label><%= rs1.getString(3)%></label></th>

</tr>

<% }

**5.2.7 Amount calculator**

This code snippet helps to calculate the amount by multiplying the current price with the quantity.

ResultSet rs=st.executeQuery("select ourprice from crop where name='"+cropname+"'");

rs.next();

**float** price= rs.getFloat(1);

**float** amount=price\*q;

Similarly the jsp code for placing order and deleting order works. It uses the update and delete sql commands instead of the select command.

**5.2.8 Styles page(CSS)**

The CSS file has the styles for various sections. The following code snippet is the css styles for create account section.

#createAccount

{

position: relative;

top: 50px;

left: 170px;

background: rgba(63, 57, 57, 0.205);

color: rgb(160, 154, 154);

width: 300px;

height: 100%;

padding: 30px 30px;}

**5.2.9 Displaying tables in the Admin page**

The following code snippet is for displaying the farmer page in the admin page. Similarly all the other tables are also displayed. Here ‘selectFarmer’ is the stored procedure.

ResultSet rs=st.executeQuery("call selectFarmer;");

**while**(rs.next())

{

%><tr><th><label><%= rs.getString(1)%></label></th>

<th><label><%= rs.getString(2)%></label></th>

<th><label><%= rs.getString(3)%></label></th>

<th><%= rs.getString(4)%></th>

<th><%= rs.getString(5)%></th>

<th><%= rs.getString(6)%></th>}

#### **5.3 Connectivity**

This is a code snippet to show how JSP is used to connect to the local MySQL database using the localhost server.

Class.forName("com.mysql.cj.jdbc.Driver");

Connection connection= DriverManager.getConnection(“

jdbc:mysql://localhost:3306/db1?useSSL=false","root","Chandan@1998");

**5.4 Triggers and Stored Procedures**

The following command is used to create a trigger called ‘del’. This trigger is invoked whenever a tuple is deleted from the ‘farmer’ table. The name and phone number of the deleted user is stored into another table called ‘deletedusers’.

mysql> delimiter $$

mysql> create trigger del

after delete on farmer

for each row

begin

insert into deletedusers values(old.name,old.phone);

end$$

After this command the trigger will be created.

The following command is used to create stored procedure called ‘selectfarmer’. This stored procedure can be invoked using the call command.

mysql> delimiter //

mysql> create procedure selectFarmer ()

begin

select \* from farmer;

end//

Query OK,0 rows affected (0.17 sec)

When the ‘call selectFarmer’ command is used, the ‘farmer’ table is displayed.

Chapter 6

**Snapshots**

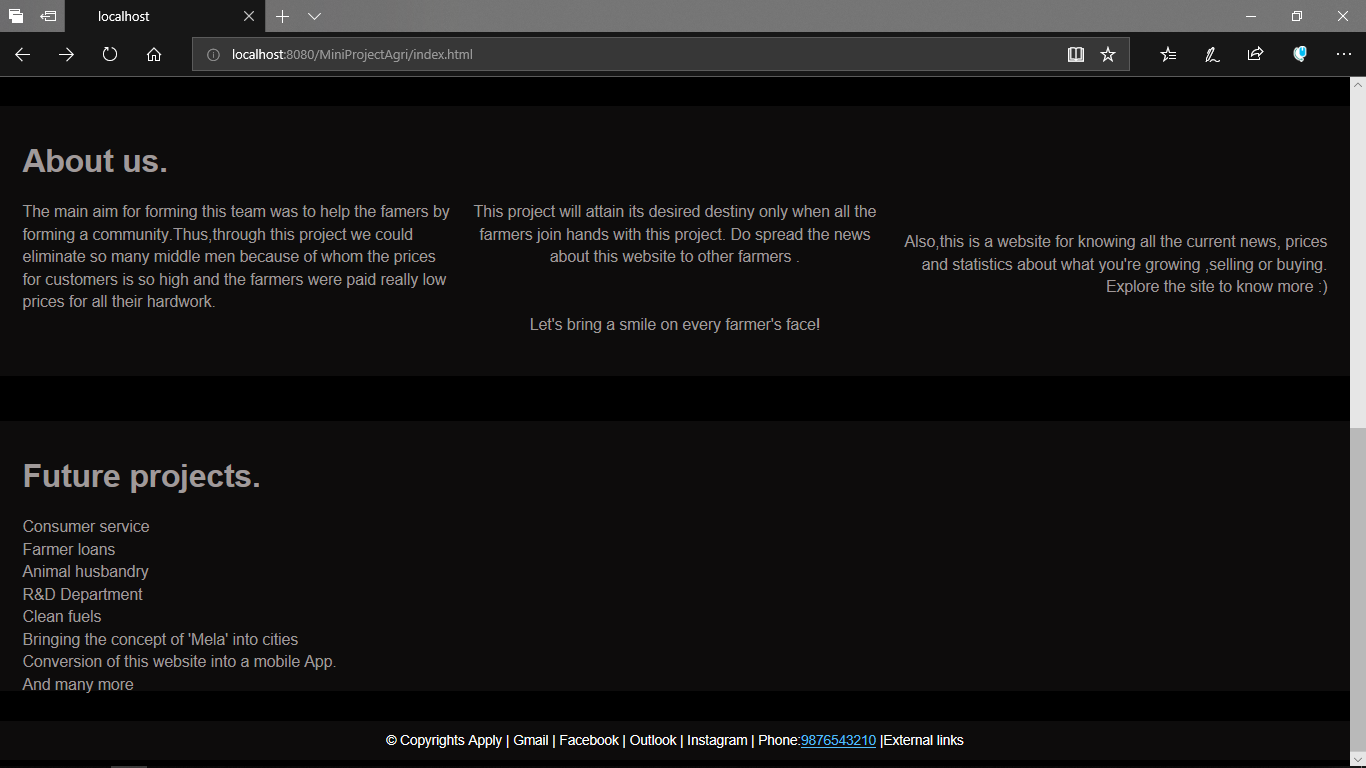
**6.1 Index page**



**Fig.6.1 Index page**

The index page (fig 6.1) is the first page of the website. It has two form- one for registration and another for login. The name of the website ‘farmbook’ is displayed at the top left corner. When it is clicked, the index page is displayed. The register form consists of various text fields to be filled and all the fields are strictly required. The user will have to choose a unique username and password while registering. This username and password is used for logging in. When the submit button, named as ‘create!’, is clicked, the data is stored into the database. The user can now login using the login form, in order to view and use the various other features in ‘farmer’ page, the details about which is in section 6.2. The admin will also have to use the same login for to login into the admin page. More details regarding the admin page is in section 6.3.

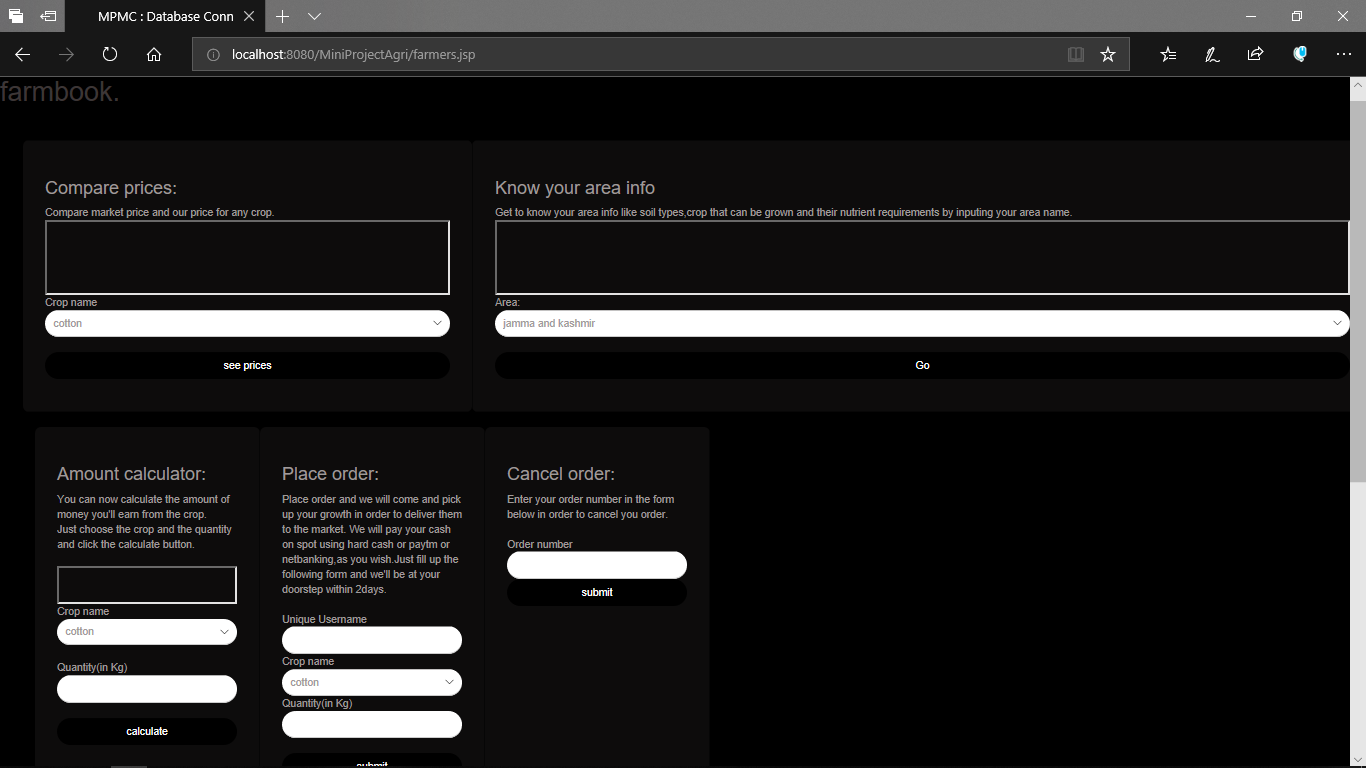
When the ‘about’ is clicked, the about section (fig 6.2) of the index page is displayed. The about section contains information about ‘farmbook’ as well as a list of its future projects. It also contains a footer at the end of page which contains information regarding how to contact and also copyrights.



**Fig.6.2 About section**

**6.2 Farmers page**

The farmers page (Fig.6.3) consists of various features like comparing prices, knowing the area information, amount calculator, placing order and cancelling order.



**Fig.6.3 farmers page**

The ‘compare prices’ section takes in the crop name as value and displays the market price and the price which the Farmbook offers inside the frame(box) provided. This is very important because the farmer will now know the difference in the prices the dealers or middlemen will be offering, and the prices Farmbook will be offering.

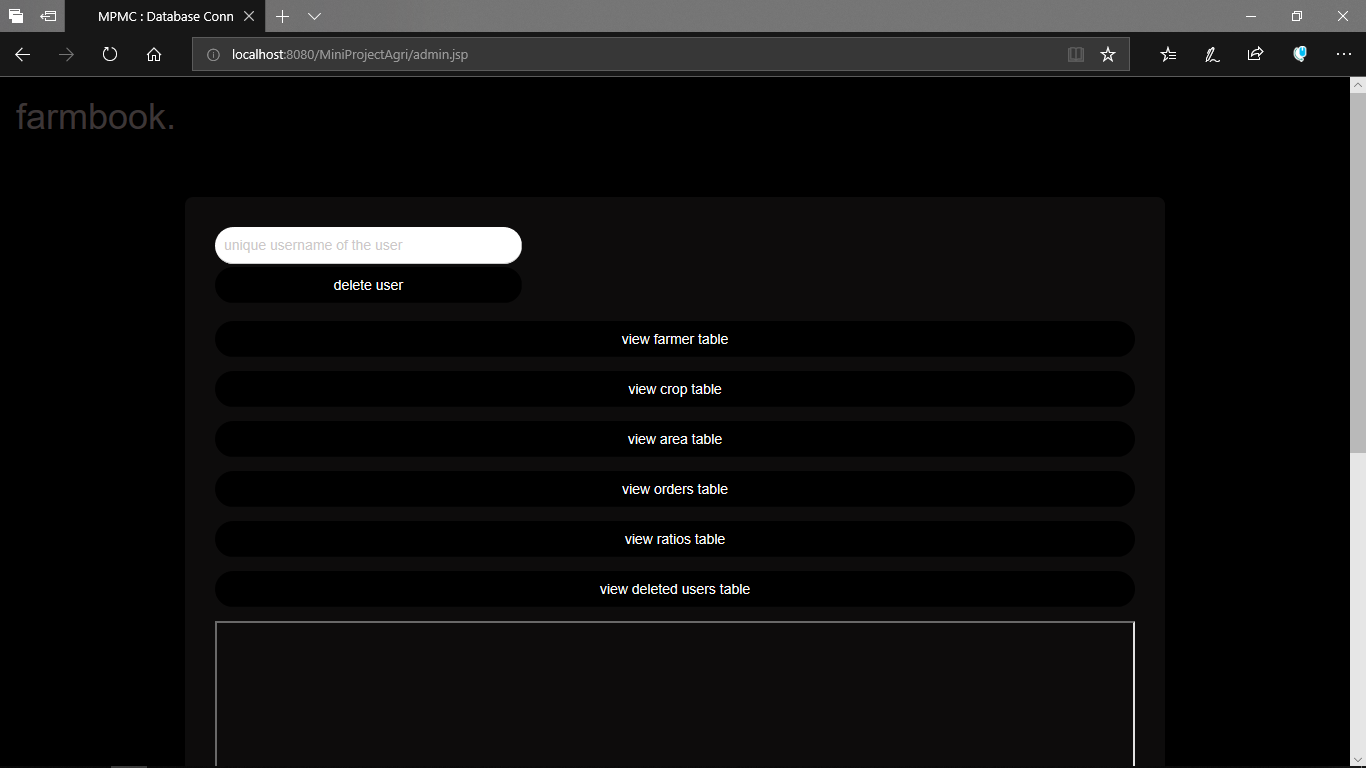
The ‘Know your area info’ takes in area (states) as input and displays the soil types found there, crops that can be grown and also the nutrient requirements for those crops. This information is displayed in the box in a tabular form. It is really important for a farmer to know about his/her area. So now, the farmer will know which crop to grow and the nutrient requirements for that crop. So now, with the nutrient requirements in his/her mind, the farmer can choose the right kind of fertilizer for the crops.

The ‘Amount calculator’ section helps to calculate the amount the Farmbook will offer the farmer for a given crop and a given quantity. The user will have to choose the crop name and the also type the quantity of crops he/she is planning to sell. The result will be displayed int the box just like a calculator.

The ‘Place order’ section helps notify the Farmbook that the farmer is ready with the crops and that the crops can be picked. The user will have to input the username, crop name and the quantity of crops he/she is going to sell. The farmer will then be contacted and vehicles will be sent for transportation after confirmation. An order number is generated and the farmer can use the same number in order to cancel the order in the ‘Cancel order’ section.

**6.3 Admin page**

The admin page (Fig.6.4) consists of various features. Here is a snapshot of the same.



**Fig.6.4 Admin page**

The admin page consists of the ‘delete user’ option along with buttons to view the various tables in the database like farmer table, crop table, area, ratios, orders and deleted users table. The table will be displayed in the box which present below the last button.

The admin has privileges to view the various tables in the database only a click away. The admin can also delete users by just giving their username. Using triggers, the deleted users’ name and phone numbers will be stored in the ‘deletedusers’ table, in case they need to be contacted in the future. These admin features are not available to any other user. The admin will have to use a combination of a specific username and a specific password using the same login section (in the index page) in order to login to the admin page.

**Conclusion**

With a large number of farmers entering below poverty line and the consumer needs ever increasing, there’s a need to cut down the middlemen or dealers for financial as well as moral reasons.

The construction of this mini project “Farmbook” was one of these opportunities. Farmbook is being launched because a need for a destination that is beneficial for both farmers as well as consumers is required. Farmers can now easily know details about the area, crop, nutrient requirements and many more. A farmer can also easily sell their crops to Farmbook at a better rate compared to the price offered by dealers or middlemen, thus earning a better profit and returns from their crops.

The Farmbook is developed using eclipse and MySQL fully meets the objectives of the system for which it has been developed. The system is operated at a high level of efficiency and all the users understand its advantage. The system solves the problem. It was intended to solve as requirement specification.

**References**

• W3Schools (HTML and CSS) – https://www.w3schools.com

• JSP- https://www.javatpoint.com

• The MySQL Documentation – https://dev.mysql.com/doc

• Stack Overflow for doubts – https://stackoverflow.com

• For crop related research – <https://www.testbook.com/blog>

and https//:www.naasindia.org/

• A textbook called ‘Fundamentals of Database system’ by Ramez Elmasri and Shamkant B Navathe.

• A textbook called ‘Database Management Systems’ by Ramakrishna and Gehrke.