FINAL PROJECT REPORT

Course Name: Web Database Programming
Course Code: SE 407

Topic Name: Dairy Firm Management System

Submitted To:

Md. Ataullah Bhuiyan
Senior Lecturer
CSE Department
City University

Submitted By:

Md. Firoj Mahmud Shuvho
Id: 171442634 Batch: 44th
CSE Department
City University

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CITY UNIVERSITY, DHAKA

(index completely has broken down)

List of Figures:

Chapter 1 Introduction

1.1 Introduction

Dairy is a universal agricultural production. People milk dairy animals in almost every country across the world, and up to one billion people live on dairy farms. It is a vital part of the global food system and it plays a key role in the sustainability of rural areas in particular.

Bangladesh is a densely populated country with an acute shortage of milk. Generally dairy in the country is practiced as a part of the crop-livestock farming system with one or two cows. The average milk production in Bangladesh is less due to low productivity of dairy animals, and lack of proper care and management (BER, 2012). Due to this acute shortage in domestic milk production and high income elasticity of demand for milk there is a good potential for growth of the dairy sector. Despite the high demand of milk and milk products, dairy farming is not growing with the pace of the country's demand.

The milk production in Bangladesh as revealed from both datasets has been increasing at a faster rate during the period from 2006 to 2019 but with a strong fluctuation in milk production. Although the production of milk was highest in 2008-09 which was 2.66 million metric tons (BER, 2012). The economy of Bangladesh depends on the livestock sector and contributes 2.57 percent to the gross domestic product in Bangladesh during 2010-11. Therefore, the contribution of the livestock sector is declining to the GDP of Bangladesh. Although the sharing of livestock in the GDP has declined over the past few years, livestock production becomes very successful in increasing livestock population (BER, 2015). As a source of supplying animal protein to the human diet, milk occupies second position after meat and egg. The per capita availability of milk is 45 ml. against the minimum requirement of 250 ml (DLS, 2006/07). There is a huge gap between requirement and the availability of milk. Small scale dairy production and related backward and forward linkage activities in marketing; input supply, etc. have the potential for significant employment generation and poverty alleviation. Dairy generates more regular cash income and dairy production, processing and marketing generate more employment per unit value added compared to crops. The study will also facilitate the formulation of appropriate policies for future improvement of milk production.

It is a well-known fact that the dairy industry actively contributes to the economies of a number of communities, regions and countries. An increasing demand worldwide is noticeably emerging at present, and the industry is globalizing, thus increasing the scope and intensity of the global dairy trade. However, the question of how and on what criteria we can objectively assess the economic benefits of the dairy sector still remains.

With online management, everyone can manage the business by understanding the problems and grievances of the business easily. Such aspects help the business operations to work efficiently in order to achieve a common goal.

Dairy Farm Management System is an online platform designed to automate the record-keeping and transactions in the operation of dairy farms. Core features of this system include the cow information, vaccine, and feed monitoring as well as the record of milk collection and sales report. So the main objective of this project is to automate the complete operation of the dairy cow monitoring. Using this we can keep our future records from this system. Our project ensures that proper procedures are set up in the dairy farm that results in an improvement in milk production and cows' fertility and health.

1.2 Objectives

- To manage dairy farm daily information easily and securely
- To keep a trace of growth in the dairy business

1.3 Justification of the Study

- The new system required less time for completion of any work, decreasing the chances of error, working smoothly and very fast and saving time and manpower.
- It will provide suitability for computerized data entry, maintaining dairy information, staff information and Customer information, milk rate information. This project secure data in less amount of time.

1.4 Scope of the Study

We can implement this system in the dairy farm sector to store our daily information easily. It is very difficult to maintain a big dairy farm. Using this system everyday dairy farm management can easily be managed to produce their work. This project is easy for data storing and generates information in less amount of time. This system also helps to encourage the rapid growth of the dairy farm sector. As well as this will help to create the new platform by taking the hinds of this sector.

Chapter 2. Literature Review

2.1 Background Study

Nowadays dairy production has come a long way since the days of milking by hand. These days, machines are used to help the farmer to milk several cows at once. Most farmers do not have it so easily though, it is very tough to maintain a big dairy farm for farmers because there are many problems like the long and demanding daily routine of feeding and milking. So it is a major problem for farmers. It is difficult for data recording and generates information in less time. So, dairy farm management becomes an important concern to help the dairy farmer.

2.2 Related Works

In (John Lai etal.2018) the authors proposed, analyzed and examined how dairy farmers prioritize critical management areas in their operations and derived implications for future growth. Their analysis provided dairy operators and industry stakeholder's insights to facilitate dairy farm success and growth. J. B. kauffman et al.1986, proposed a plan for the utilization of resources to produce and market the farm output and finance the farm business, with the goal of attaining the highest possible level of satisfaction for the farm family. In this study, the success of a management strategy was assumed to be reflected in a farm's profitability as measured by annual observations on labor and management income or rate of return to equity capital (J. B. Kauffman et al, 1986).

In addition to management decision making, farm size, milk production levels, and milking systems used are also identified as factors that influenced dairy farm profitability positively (Gloy and LaDue, 2003; Gloy et al., 2002). Managing facilities or equipment on a regular basis by livestock producers may be part of the recipe for economic success, but according to the authors the fundamental aspects of management and decision making are still integrally important (Campe et al., 2015).

Depending upon the above study, we implement a Dairy Farm Management System (DFMS). So the main objective of this project is to automate the complete operation of the dairy cow monitoring, cow information, vaccine, and feed monitoring as well as the record of milk collection and sales report.

Chapter 3 Methodology

3.1 Methodology

This project has been completed by following strategy, which is given below:

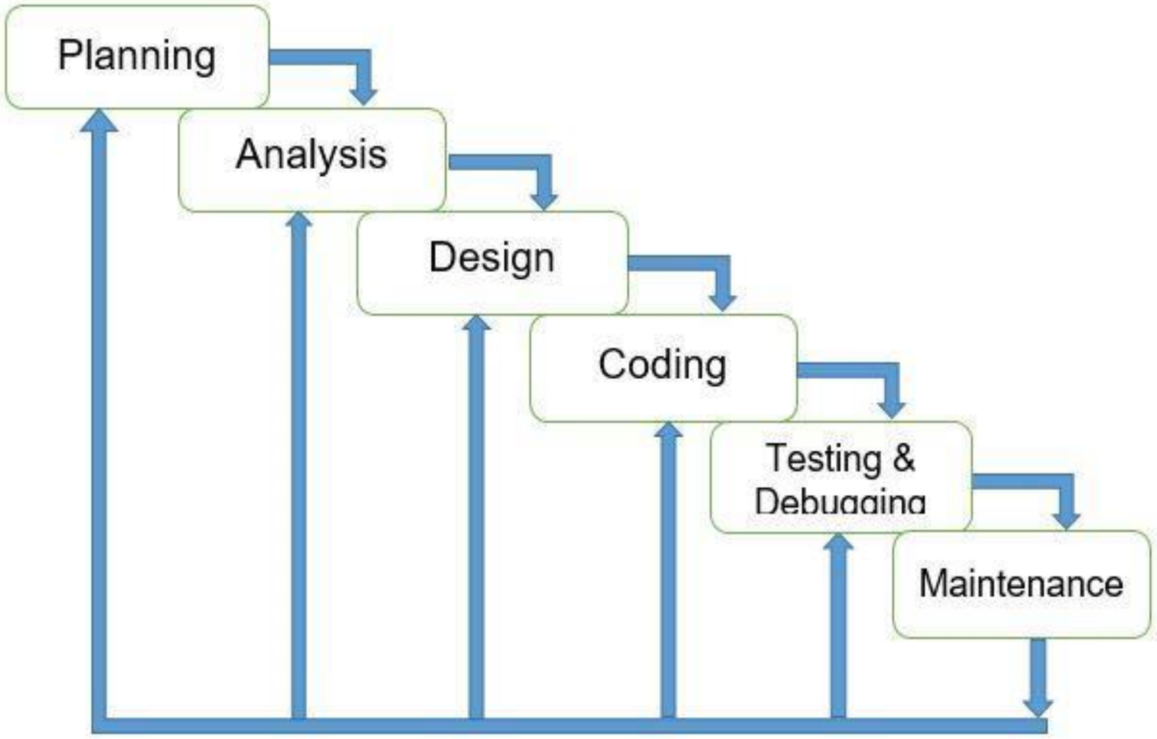


Figure: 3.1 Waterfall Model

3.2 Justification of Methodology

We have divided the whole project work in six phases using a sequential process to solve any problem to develop any system it should arrange the whole work in the segment so that accuracy can be provided. If we want to modify our system at any point according to equipment so we can do it depending on our requirement. In this workflow, each phase must be completed completely before the next steps may begin.

3.3 Description of Methodology

Full work has been divided into six parts. The work is done part by part.

- **Planning:** We seek for some problems from dairy farmers. Then we planned to solve the problem in some innovative way.
- **Analysis:** We gathered some information from dairy farms. We find out what are the problems with dairy farms. All possible requirements are captured in product requirement documents. We have read some research papers about this problem. After making a plan, we were thinking about our requirement to make our project successful.
- **Design:** First we developed a prototype version of the project. We design its structure. First, we have a login page. And the admin has a unique login and password which completely controls the working. Admin manages the complete stock management required for the entire dairy and feed. And he keeps track of all the orders, availability, quantity ordered by the dairy, and the cost of the feed. And the admin provides the vaccination date, cow information, feed monitoring report to the manager so that they can automate the complete operation of the dairy cow monitoring. The admin has all the power to add, remove, and change data wherever and whenever required. So the admin gets sales reports, medical reports, payment reports, feed orders, and bill receipts reports from this data. Admin provides username and password to each of the dairy managers after their verification. So one of the advantages of the system is the security of data. No one can use this data without verification. The admin gives permission to the dairy after verifying their required documents and others as per the conditions and provides a unique dairy number for each of the employees. Using this system dairy farm management can easily be managed to produce their work.
- **Coding:** After making a prototype of the project, we implemented the project using HTML, CSS, JavaScript, PHP, Laravel.

- **Testing and Debugging:** Testing means verifying correct behavior. Testing can be done at all stages of module development: requirements analysis, interface design, algorithm design, implementation, and integration with other modules. In the following attention will be direct al implementation testing. Implementation testing is not restricted to execution testing and debugging is a cyclic activity involving execution testing and code correction. The testing that is done during debugging has a different aim than final module testing. Final module testing aims to demonstrate correctness, whereas testing during debugging is primarily aimed at locating errors.
- **Maintenance:** Maintenance is the process of modifying a product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software applications after delivery to correct faults and to improve performance. In this term we try to correct faults, improve the design, implement enhancements, interface with other systems, migrate legacy software. And accommodate programs so that different hardware, software, system features, and telecommunications facilities can be used.

Chapter- 4, Analysis, Design & Development

4.1 Requirement analysis

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Requirements analysis also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. This project is made up of hardware and software. There are flowing hardware and software requirements.

4.2 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. Hardware requirement for the project are given below:

- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 2 GB or above

4.3 Software Requirements

- ❖ Recommended Operating Systems
    - Windows: 7 or newer

- MAC: OS X v10.7 or higher
- Linux: Ubuntu

- ❖ Composer

Composer is an application-level package manager for the PHP programming language that provides a standard format for managing dependencies of PHP software and required libraries. It was developed by Nils Adermann and Jordi Boggiano. Composer runs from the command line and installs dependencies (e.g. libraries) for an application. It also allows users to install PHP applications that are available on "Packagist" which is its main repository containing available packages. It also provides auto load capabilities for libraries that specify auto load information to ease usage of third-party code.

- ❖ XAMPP:

XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl, and it allows you to build WordPress site offline, on a local web server on your computer. This simple and lightweight solution works on Windows, Linux, and Mac – hence the “cross-platform” part.

- ❖ HTML

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. HTML is a markup language that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.

- ❖ CSS

Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML. CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

- ❖ JavaScript

JavaScript is a scripting language that enables to create dynamically updating content, control multimedia, animate images, and pretty much everything else. So, JavaScript is a scripting or programming language that allows implementing complex features on web pages.

- ❖ PHP

PHP is a popular general-purpose scripting language that is especially suited to web development. PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

- ❖ Laravel

Laravel is probably one of the most popular frameworks on the market right now. Laravel meets a diverse range of programming needs and projects, from beginners to advance and is well suited for all types and kinds of projects.

- ❖ Server

Server: Apache server are describing below which server we are use:

Apache: Apache is an open source web server creation, deployment and management software. It’s initial release date in 1995.

- ❖ Database

Database: MySQL database are discussing below which database we are use:
MySQL is a source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). MySQL can perform virtually all platforms, including Linux, UNIX and windows. Although it can perform in a wide range of applications, MySQL is most often associated with web application and online publishing.
MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

4.4 Design Description

4.4.1 ER diagram of the Dairy Farm Management System

The description of ER diagram of the Dairy Farm Management System is given below:

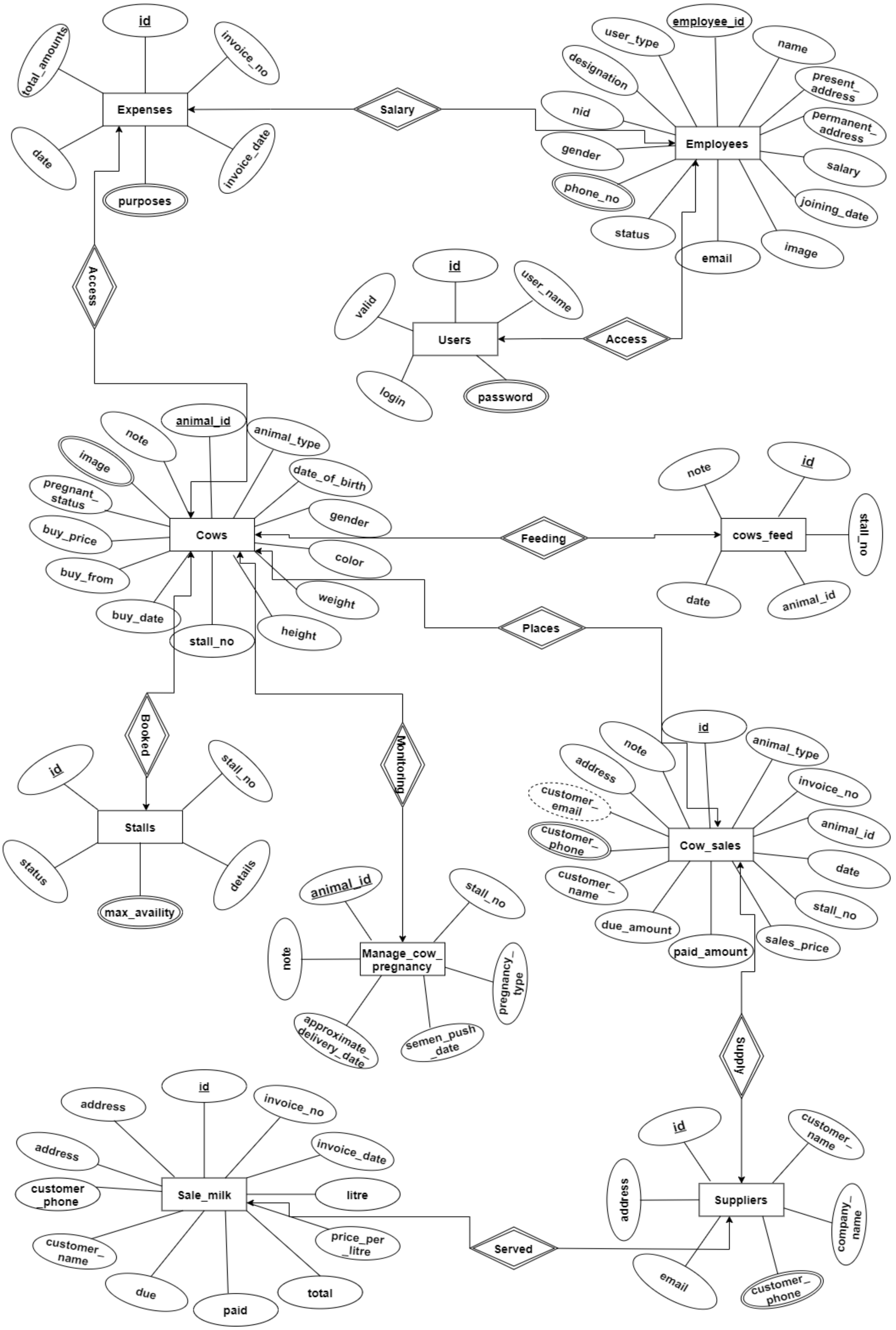


Figure-4.1 ER-Model of dairy farm management system

Here is the description of every entity in this ER-Model.

1. **Employees:** This entity have the relationship which denoted by the diamond shape work for salary to employee salary, employee type to user type. Here have some attributes which already shown in ER diagram and we'll do required work by selecting desire attributes.

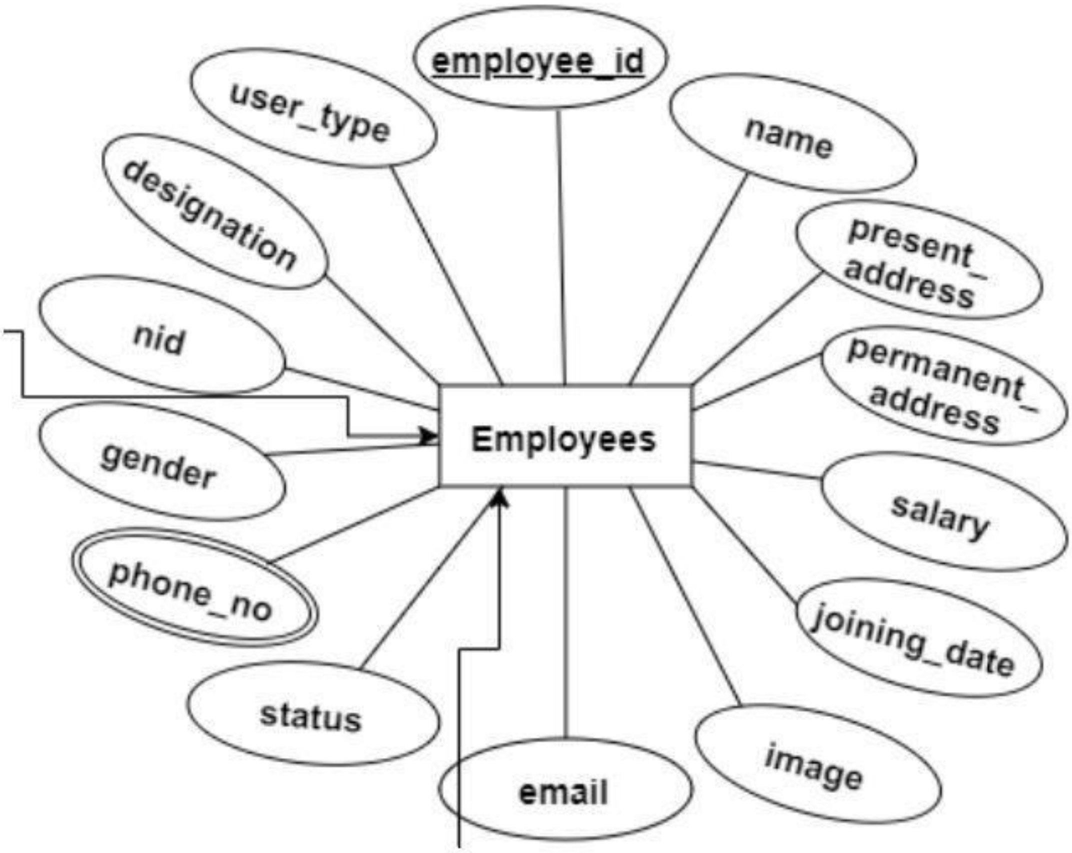


Figure- 4.2 Employees entity

2. **Cows:** This entity have relationship which denoted by the diamond shape name feeding to cows feed, relationship to stall, monitoring to manage cow pregnancy, places to cow sales, types to animal types, color to colors. We'll select required attribute from entities and will work accordingly.

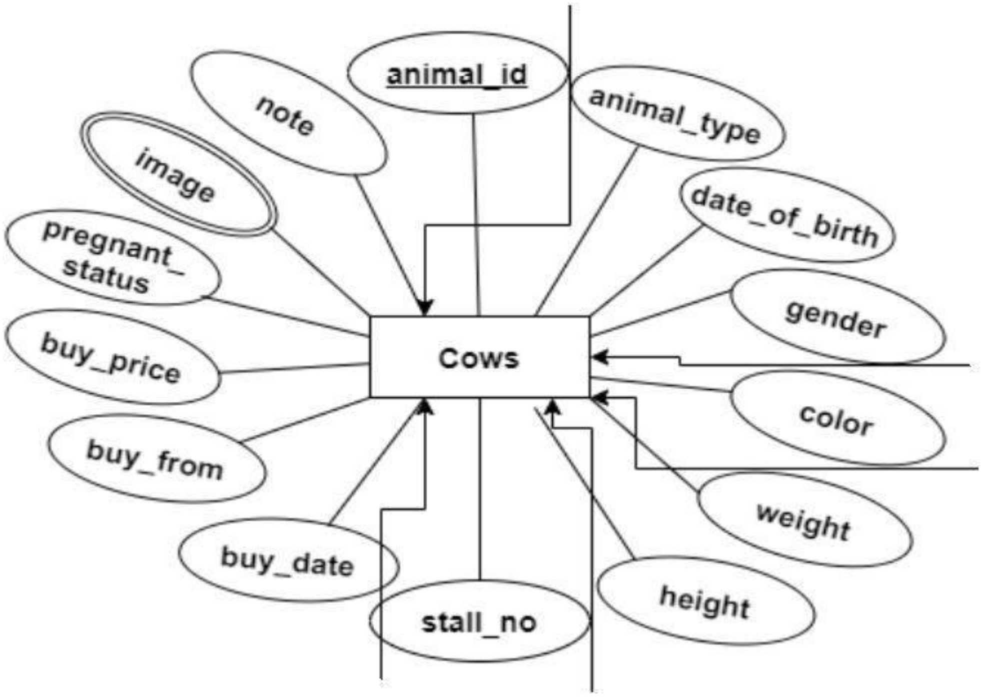


Figure-4.3 Cows entity

3. **Cows feed:** These entities have relationship which denoted by the diamond shape name feeding to cows, has to food items and staff number. From this entity we'll get required information as we want.

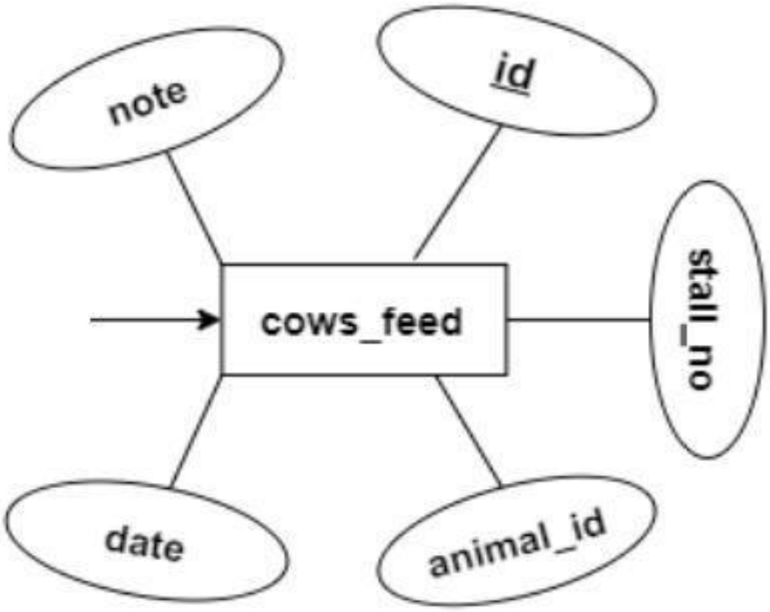


Figure-4.4 Cows Feed entity

4. **Manage cow pregnancy:** This entity has relationship which denoted by the diamond shape name monitoring to cow. From this relationship we'll see required information accordingly.

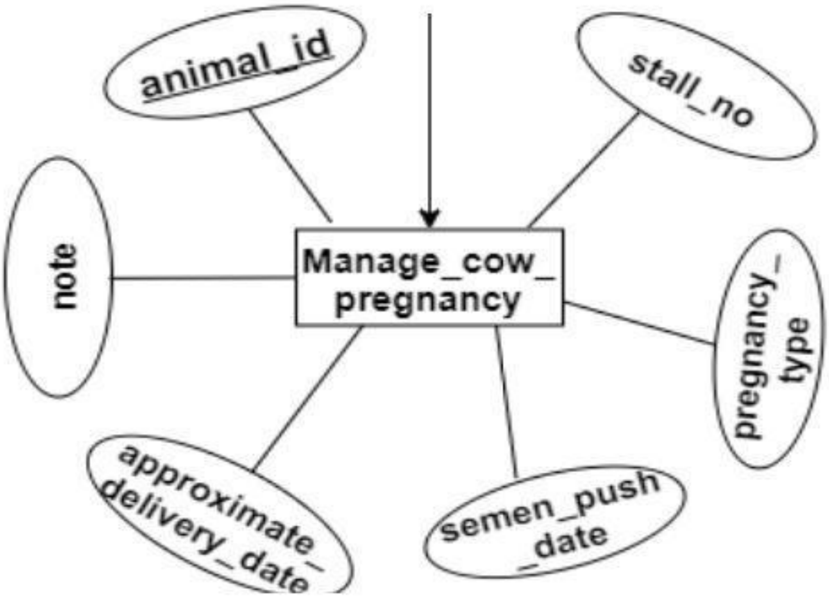


Figure-4.5 Manage Cow Pregnancy entity

5. **Cow sales:** This entity have relationship which denoted by the diamond shape name served to suppliers, places to cows. We'll select required attributes from these entities.

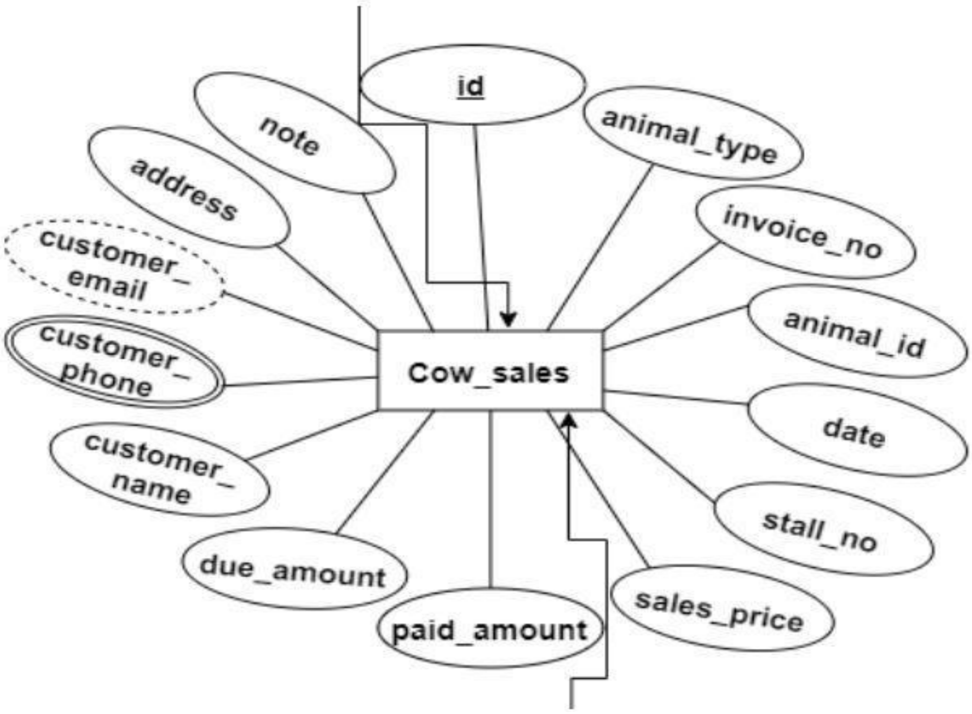


Figure-4.6 Cow sales entity

6. **Sale milk:** This entity has relationship which denoted by the diamond shape name served to sale milk. From this relationship we'll get multiple information accordingly.

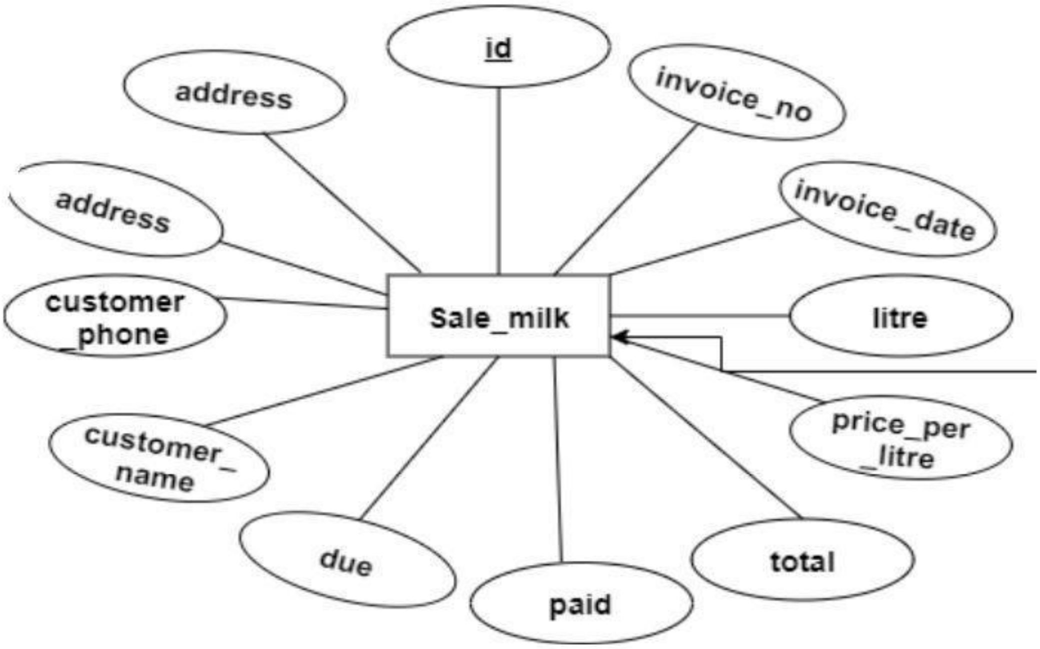


Figure-4.7 Sale milk entity

7. **Suppliers:** This entity has relationship which denoted by the diamond shape name served to suppliers. We'll see required information from these entities and their attributes.

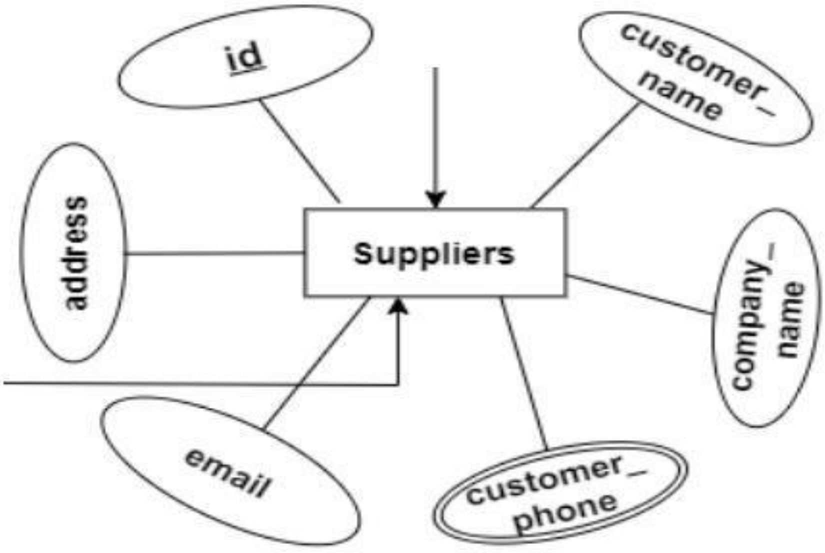


Figure-4.8 Suppliers entity

8. **Expenses:** This entity also have only one relationship which denoted by the types to expense purposes. Here we'll see our all required information from the attributes.

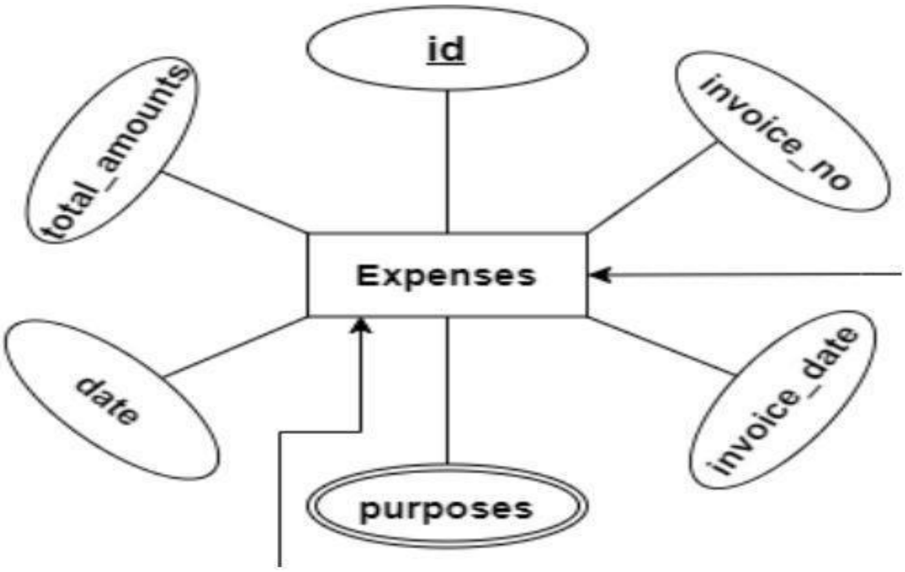


Figure-4.9 Expenses entity

9. **Users:** This entity have only one relationship which denoted by the diamond shape name access to user type. From this entity we've get access to user type.

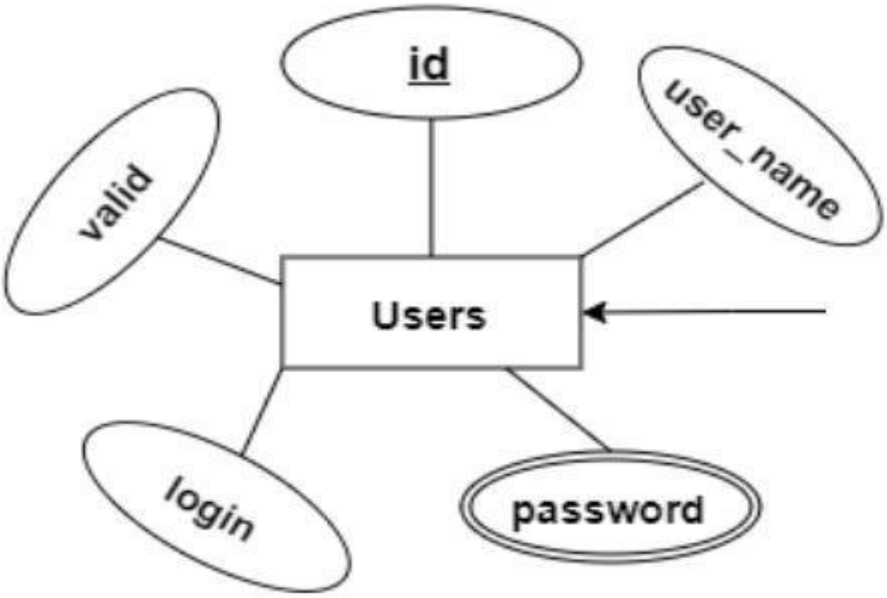


Figure-4.10 Users entity

10.**Stalls:** This entity have relationship which denoted by the diamond shape name has many to stall ledger and relationship to cows. We'll do our required work by selecting desire attributes from these entities.

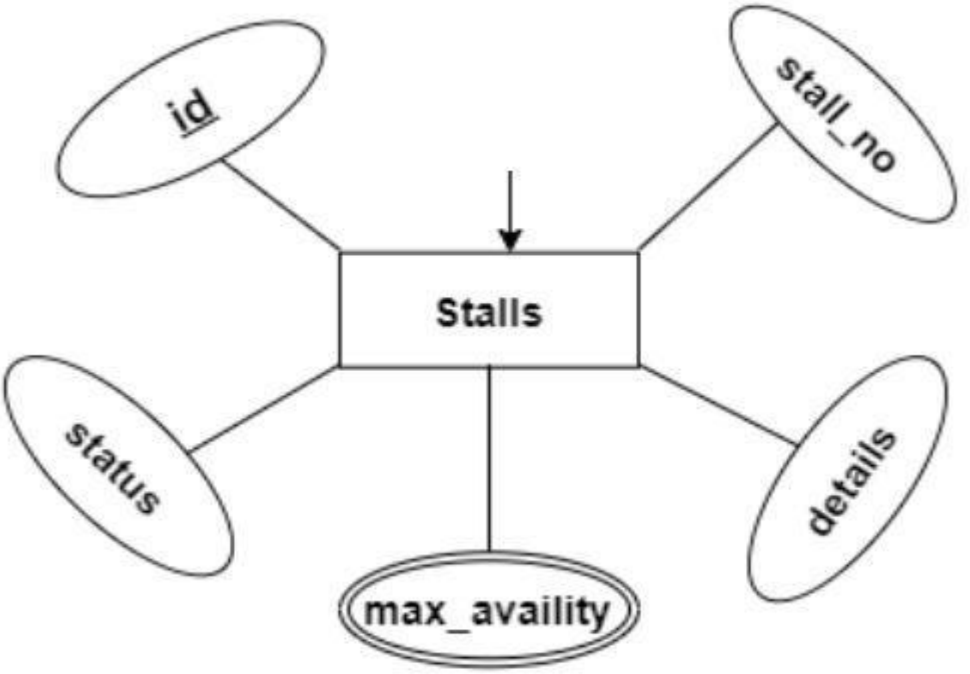


Figure-4.11 Stalls entity

4.4.2 Use case diagram of the Dairy Farm Management System project

Here is the use case diagram of the project:

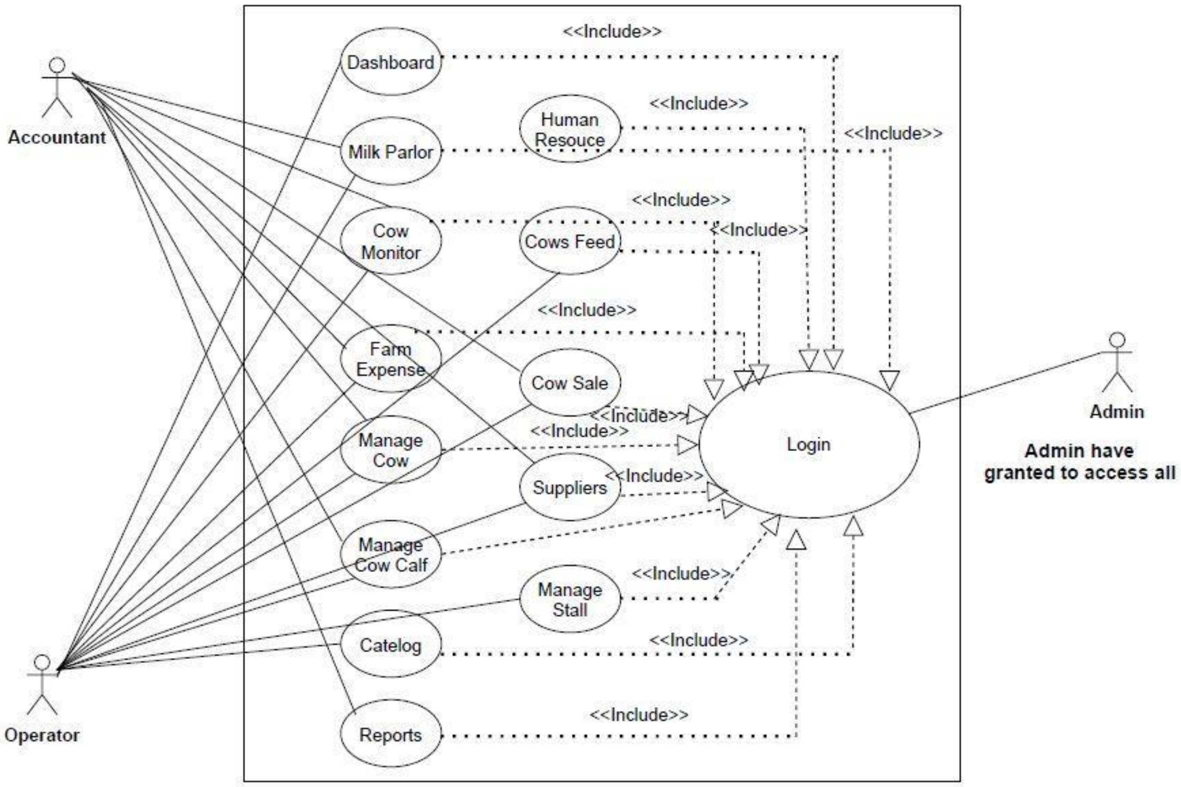


Fig: Dairy farm management system usecase diagram

Figure-4.12 Use case diagram of the Dairy farm management system

1. **Accountant**: In this project accountant is a professional who performs accounting functions such as account analysis, auditing, or financial statement analysis. Accountant can manage milk parlor, cow monitor, farm expense, manage cow, manage cow calf, reports, cow sale and suppliers.
2. **Operator**: A computer operator is a role in IT which oversees the running of systems, ensuring that the machines, and projects are running properly. Operator manages the complete stock management required for the entire dairy and feed. And he keeps track of all the orders, availabilities, quantity ordered by the dairy, and the cost of the feed. And the operator provides the vaccination date, cow information, feed monitoring report to the admin so that they can automate the complete operation of the dairy cow monitoring.

Operator can manage milk parlor, cow monitor, farm expense, manage cow, manage cow calf, catalogue, cows feed, cow sales, suppliers and manage stalls.

3. **Admin**: The definition of an administrator is a person who is in a position of authority or who manages people, practices and policies. An Administrator provides system support to either an individual or team and is vital for the smooth-running of a project. In this project, Admin will verify the reports entered by the operator and accountant. Admin provides username and password to each of the operator and accountant after their verification. No one can use this data without verification. The admin has all the power to add, remove, and change data wherever and whenever required. So the admin gets sales reports, medical reports, payment reports, feed orders, and bill receipts reports from this data

Chapter 5: Project Descriptions

The dairy farm management system comprises two core parts which are dairy farm user Interface and dairy farm database. The dairy farm user Interface is the front-end of the software and dairy farm database is the back-end of the software. The two parts are interrelated with a suitable programming language that is especially appropriate for web development management systems.
Therefore, in this chapter we discuss the complete system in three fold.

5.1 Dairy Farm User Interface:

The frontend of a website is the part that users visually see first in their browser or application. Front end development is mostly focused on the client side of the software how it will become user-friendly.

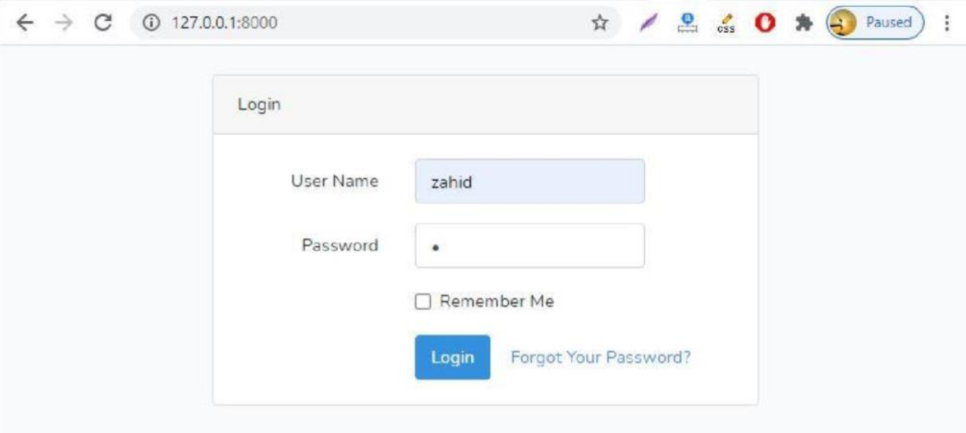
HTML: HTML Stands for Hyper Text Markup Language which is used to design the frontend portion of web pages. It is a combination of Hypertext and Markup language. Hypertext is used to link between the web pages. Markup language is used to define text documentation within tags which can define the structure of the web pages.

CSS: CSS stands for Cascading Style sheet which allows to apply styles to the web pages. CSS gives developers a flexible, precise way to create attractive, interactive web addresses.
JavaScript: JavaScript is a scripting language that is used to create the web pages dynamic elements on static web pages.

We also use frameworks and libraries to design dairy farm management software. They are given below:

- Bootstrap
- JQuery
- Laravel
- Ajax

Every software interface is more important because it helps the user to interact and perform specific tasks in the software. Visiting the web address user will see the login interface shown in fig. 5.1



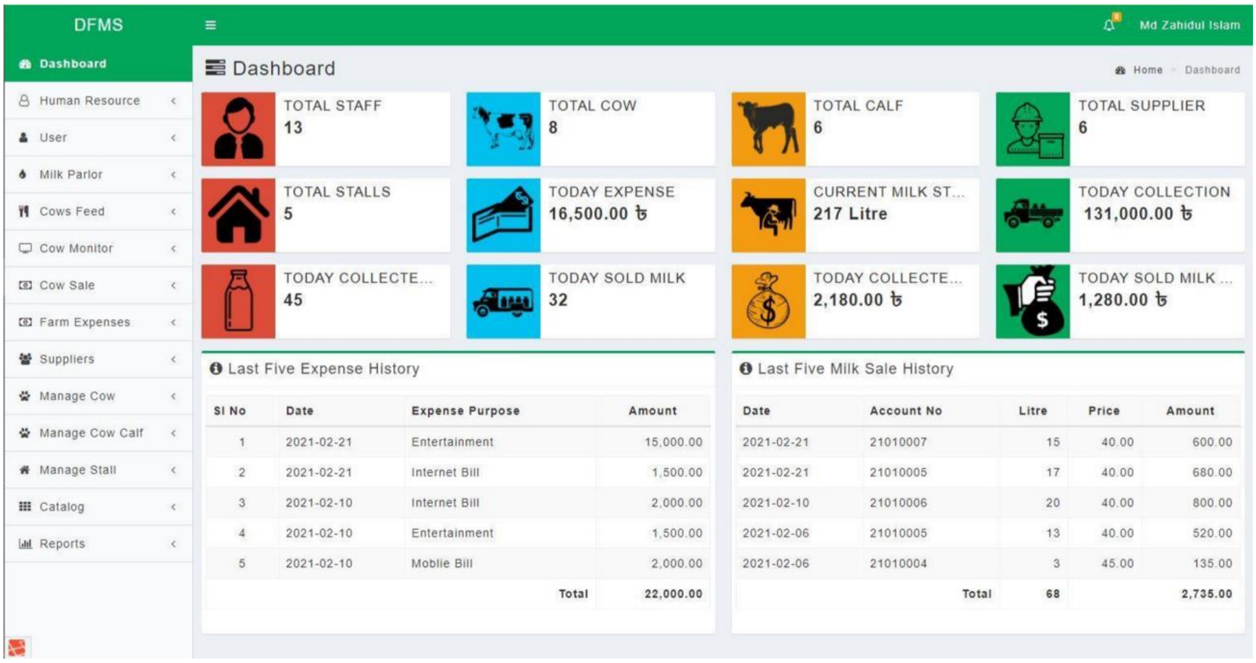
User Name: zahid

Password: •

□ Remember Me

Figure 5.1 Login page of the dairy farm management system

By entering the valid username and password, the user can see the dashboard. If someone enters an invalid username or password, he or she can’t access this software. After access this software user can see some features of our software where users can select some features which can perform the desired operation accordingly. The home page features are demonstrated in fig 5.2



Dashboard

TOTAL STAFF 13

TOTAL COW 8

TOTAL CALF 6

TOTAL SUPPLIER 6

TOTAL STALLS 5

TODAY EXPENSE 16,500.00 ৳

CURRENT MILK ST... 217 Litre

TODAY COLLECTION 131,000.00 ৳

TODAY COLLECTE... 45

TODAY SOLD MILK 32

TODAY COLLECTE... 2,180.00 ৳

TODAY SOLD MILK ... 1,280.00 ৳

Last Five Expense History

Sl No	Date	Expense Purpose	Amount
1	2021-02-21	Entertainment	15,000.00
2	2021-02-21	Internet Bill	1,500.00
3	2021-02-10	Internet Bill	2,000.00
4	2021-02-10	Entertainment	1,500.00
5	2021-02-10	Moblie Bill	2,000.00
		Total	22,000.00

Last Five Milk Sale History

Date	Account No	Litre	Price	Amount
2021-02-21	21010007	15	40.00	600.00
2021-02-21	21010005	17	40.00	680.00
2021-02-10	21010006	20	40.00	800.00
2021-02-06	21010005	13	40.00	520.00
2021-02-06	21010004	3	45.00	135.00
	Total	68		2,735.00

Figure 5.2: Dashboard of the dairy farm management system.

Users can see more features by clicking the dropdown menu. Each of the features can be explored to see more operations. Each of the features of the Dashboard is discussed below:

- **Human resource:** Using the menu system administrators can add staff, staff list, and employee salary features. Besides, he can grant permission to other users to do specific tasks.
- **User:** From this menu, we'll get individual user access like executive, officer, admin, accountant and operator. These users can use other features and perform their work accordingly.
- **Milk parlor:** This menu helps to store information about when and how many liters of milk are milking from the cows. Besides, the amount of sold milk is also stored.
- **Cows feed:** This menu stores date of feeding, stall no, cow number, note for the future so that we can know what type of feed the cow gets and the action menu to edit and delete any data from the database.
- **Cow monitor:** Cow monitor stores the routine monitor information as well as animal pregnancy report. From this menu we can get animal pregnancy status and we'll take necessary steps gradually.
- **Cow sale:** From cow's sale menu we can get the sale list and sale due collection information accordingly. From the sales due collection, we'll get invoice number and details of due.
- **Farm expenses:** Farm expenses stores expense list, expense purpose details of dairy farm. Expense purpose means farm maintenance cost or stationary buy cost.
- **Supplier:** Supplier menu stores the supplier information such as supplier image, supplier name, company name, phone number, email address. From this menu we can see the supplier information easily.
- **Manage cow/ Manage calf:** Manage cow/ manage calf menu stores the cow details like cow id, image, gender, animal type buy date, and buying price. We'll see the cow's information from these menus.
- **Manage stall:** This menu stores the all store list of the dairy farm, stall number details, action to edit and delete.
- **Reports:** It shows the output result of input value. We'll see the office expense report, employee salary report, milk collection report, milk sale report, cow sale report from this section.

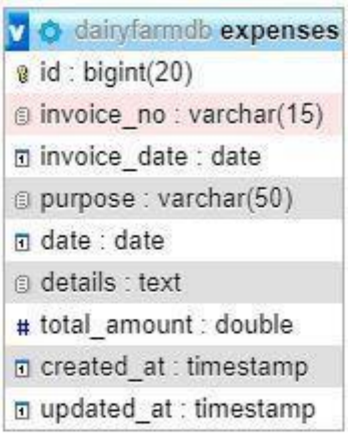
5.2 Database of Dairy Farm Management System:

The backend development refers to the server side, which consists of the server that provides data on request, the application which channels it and database that organizes the information. **Backend** developers build how to database works. The backend usually consists of three parts:

- ❖ Server
- ❖ Application
- ❖ Database

Using PHP language, we develop our backend of the software that are discuss below:

- ➢ **Expenses:** This entity has a relationship with all expense purposes. Examples: employee_salary and manage_cow_pregnancy (semen_cost, other_cost) etc. Also expenses have seven attributes. They are invoice_no, invoice_date, purpose, date, details, total_amount.



dairyfarmdb expenses

id : bigint(20)

invoice_no : varchar(15)

invoice_date : date

purpose : varchar(50)

date : date

details : text

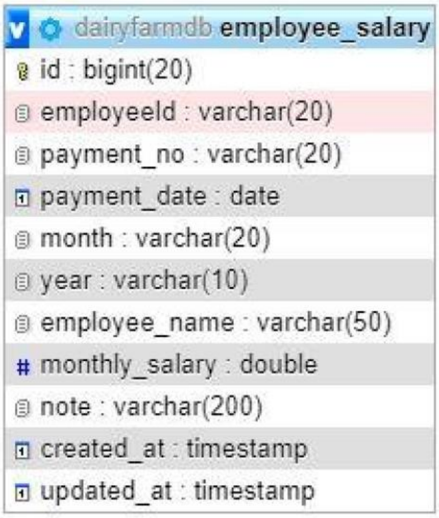
total_amount : double

created_at : timestamp

updated_at : timestamp

Figure: 5.3 expenses table structure

- **Employees_salary:** This entity has a relationship with the employee_salary entity. In employees entity attribute id is primary key and in employee_salary, epmployee_id is foreign key. This entity has eight attributes,which is employeeId, payment_no, payment _date, month, year, employee_name, monthly_salary, and note.

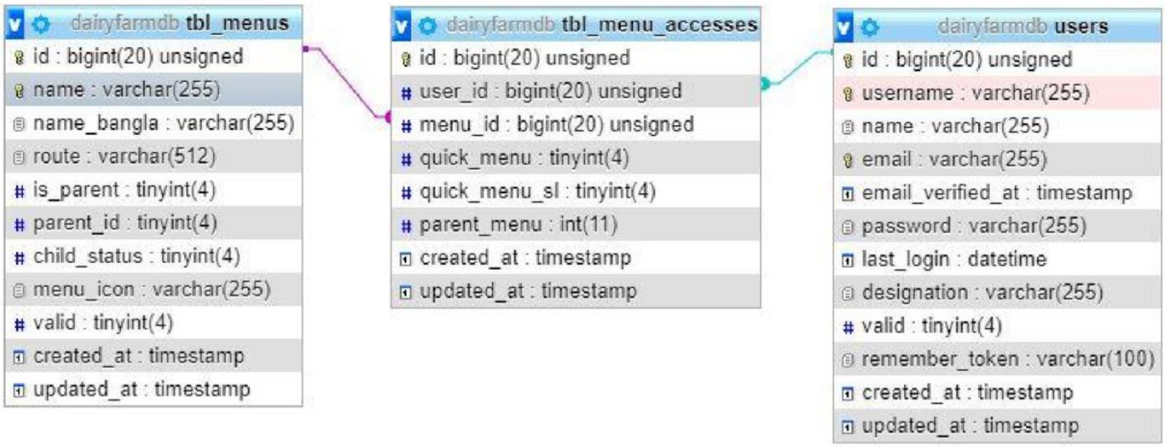


dairyfarmdb employee_salary

id : bigint(20)
employeeId : varchar(20)
payment_no : varchar(20)
payment_date : date
month : varchar(20)
year : varchar(10)
employee_name : varchar(50)
monthly_salary : double
note : varchar(200)
created_at : timestamp
updated_at : timestamp

Figure: 5.4 employee salary table structure

- **User:** This entity has seven attributes. Which is username, name, email, email_verified_at, password, last_login, designation.



dairyfarmdb tbl_menus

id : bigint(20) unsigned
name : varchar(255)
name_bangla : varchar(255)
route : varchar(512)
is_parent : tinyint(4)
parent_id : tinyint(4)
child_status : tinyint(4)
menu_icon : varchar(255)
valid : tinyint(4)
created_at : timestamp
updated_at : timestamp

dairyfarmdb tbl_menu_accesses

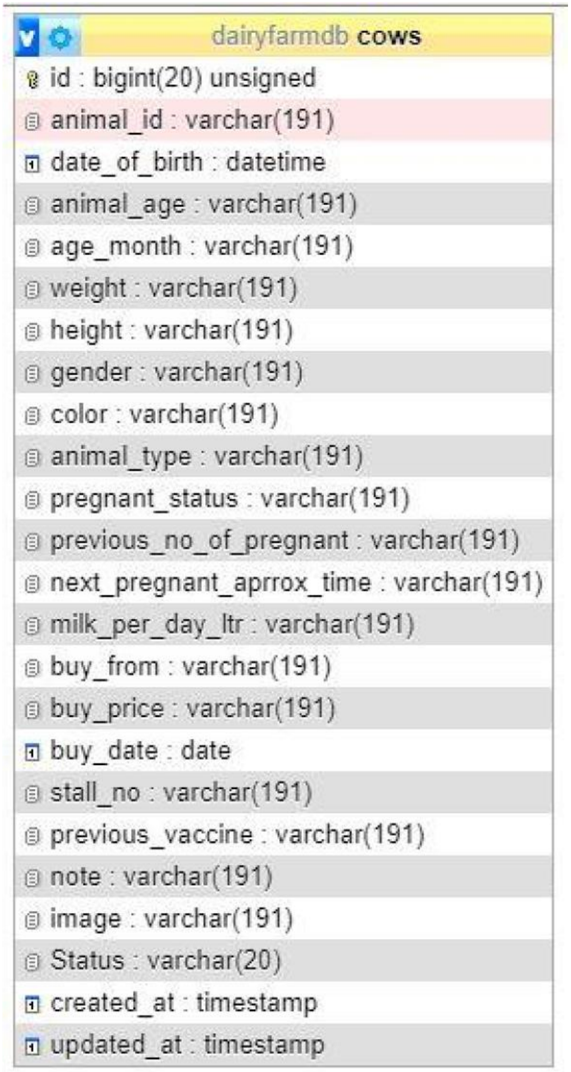
id : bigint(20) unsigned
user_id : bigint(20) unsigned
menu_id : bigint(20) unsigned
quick_menu : tinyint(4)
quick_menu_sl : tinyint(4)
parent_menu : int(11)
created_at : timestamp
updated_at : timestamp

dairyfarmdb users

id : bigint(20) unsigned
username : varchar(255)
name : varchar(255)
email : varchar(255)
email_verified_at : timestamp
password : varchar(255)
last_login : datetime
designation : varchar(255)
valid : tinyint(4)
remember_token : varchar(100)
created_at : timestamp
updated_at : timestamp

Figure 5.5: Users table with access module

- **Cows:** This entity has a relationship cow_feed, stalls, cow_sales, milk_parlors and manage_cow_pregnancy. This entity has some attributes, which is animal_id, date_of_birth, animal_age, age_month, weight, height, gender, color, animal_type, pregnant_status, previous_no_of_pregnant, next_pregnant_approx_time, milk_per_day_ltr, buy_from, buy price, buy_date, stall_no, previous_vaccine, note, image, status.

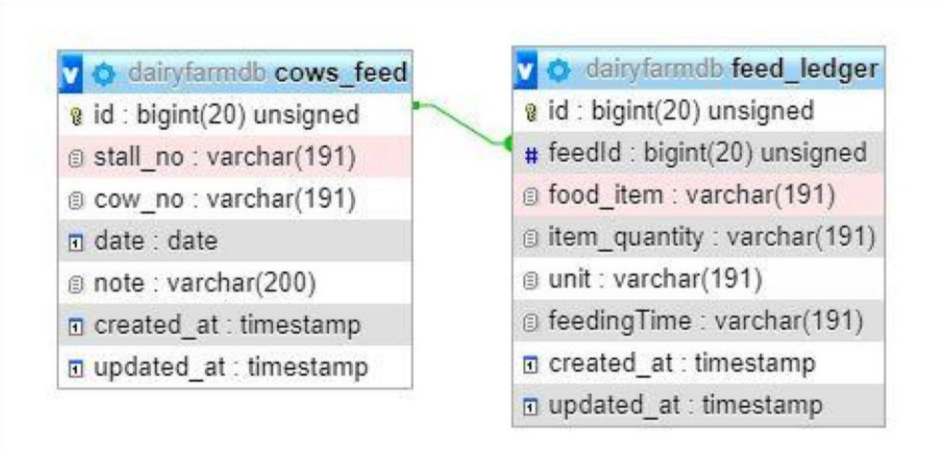


dairyfarmdb cows

id : bigint(20) unsigned
animal_id : varchar(191)
date_of_birth : datetime
animal_age : varchar(191)
age_month : varchar(191)
weight : varchar(191)
height : varchar(191)
gender : varchar(191)
color : varchar(191)
animal_type : varchar(191)
pregnant_status : varchar(191)
previous_no_of_pregnant : varchar(191)
next_pregnant_aprrox_time : varchar(191)
milk_per_day_ltr : varchar(191)
buy_from : varchar(191)
buy_price : varchar(191)
buy_date : date
stall_no : varchar(191)
previous_vaccine : varchar(191)
note : varchar(191)
image : varchar(191)
Status : varchar(20)
created_at : timestamp
updated_at : timestamp

Figure: 5.6 cows tables structure

- **Cows_feed:** This entity has a relationship with cows, cow_calf and feed ledger. This entity has 3 attributes, which is stall_no, cow_no, date, note, created_at, updated_at.



dairyfarmdb cows_feed

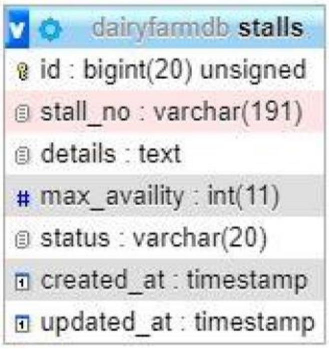
id : bigint(20) unsigned
stall_no : varchar(191)
cow_no : varchar(191)
date : date
note : varchar(200)
created_at : timestamp
updated_at : timestamp

dairyfarmdb feed_ledger

id : bigint(20) unsigned
feedId : bigint(20) unsigned
food_item : varchar(191)
item_quantity : varchar(191)
unit : varchar(191)
feedingTime : varchar(191)
created_at : timestamp
updated_at : timestamp

Figure: 5.7 cows_feed table structure

- **Stalls:** These entities have relationships with stall_ledger, cows and cow_calf. This entity has four attributes, which is stall_no, details, max_availlity, and status.

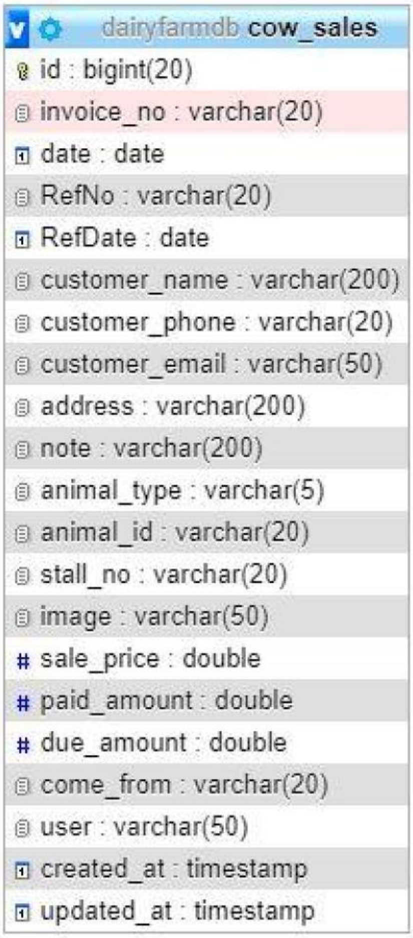


dairyfarmdb stalls

id : bigint(20) unsigned
stall_no : varchar(191)
details : text
max_availity : int(11)
status : varchar(20)
created_at : timestamp
updated_at : timestamp

Figure: 5.8 stall table structure

- **Cow_sales:** This entity has relationships with cows, cow_calf, stall_ledger and suppliers. This entity has some attributes, which is invoice_no, date, RefNo, RefDate, customer_name, customer_phone, customer_email, address, note, animal_type, animal_id, stall_no, image, sale_price, paid_amont, due_amount, come_from, user



dairyfarmdb cow_sales

id : bigint(20)

invoice_no : varchar(20)

date : date

RefNo : varchar(20)

RefDate : date

customer_name : varchar(200)

customer_phone : varchar(20)

customer_email : varchar(50)

address : varchar(200)

note : varchar(200)

animal_type : varchar(5)

animal_id : varchar(20)

stall_no : varchar(20)

image : varchar(50)

sale_price : double

paid_amount : double

due_amount : double

come_from : varchar(20)

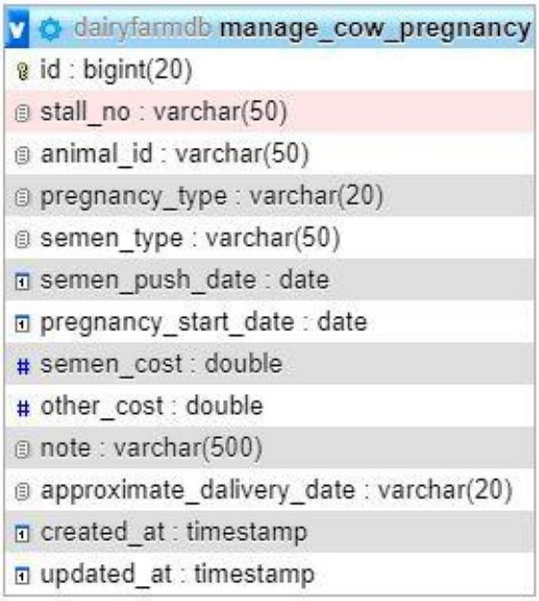
user : varchar(50)

created_at : timestamp

updated_at : timestamp

Figure 5.9: cow sale table structure

- **Manage cow pregnancy:** This entity has a relationship with cows for cow pregnancy reports. This entity has ten attributes, which is stall_no, animal_id, pregnancy_type, semen_type, semen_push_date, pregnancy_start_date, semen_cost, other_cost, note, approximate_dalivery_date.

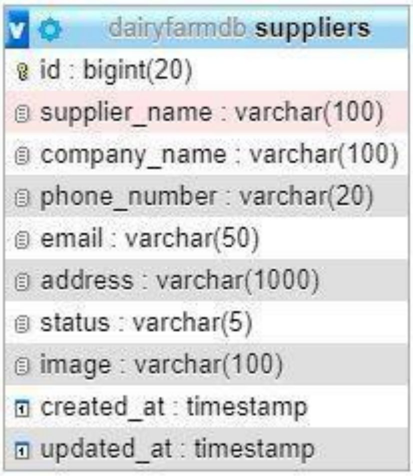


dairyfarmdb manage_cow_pregnancy

id : bigint(20)
stall_no : varchar(50)
animal_id : varchar(50)
pregnancy_type : varchar(20)
semen_type : varchar(50)
semen_push_date : date
pregnancy_start_date : date
semen_cost : double
other_cost : double
note : varchar(500)
approximate_dalivery_date : varchar(20)
created_at : timestamp
updated_at : timestamp

Figure 5.10: mangae_cow_pregnancy

- **Supplier:** This entity has a relationship with sale_milk and cow_sales. Suppliers served cows and milk to all around. This entity has seven attributes, which is supplier_name, company_name, phone_number, email, address, status, image

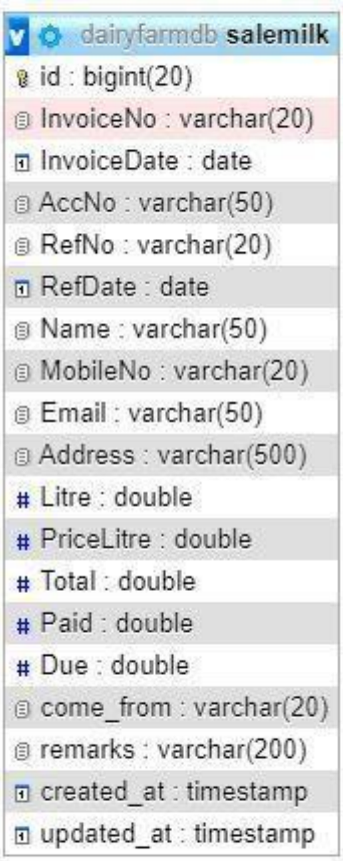


dairyfarmdb suppliers

id : bigint(20)
supplier_name : varchar(100)
company_name : varchar(100)
phone_number : varchar(20)
email : varchar(50)
address : varchar(1000)
status : varchar(5)
image : varchar(100)
created_at : timestamp
updated_at : timestamp

Figure 5.11: suppliers table structure

- **Sale_milk:** This entity has a relationship with milk_ledger and suppliers. This entity has 15th attributes, which is InvoiceNo, invoiceDate, AccNo, RefNo, RefDate, Name, MobileNo, Email, Address, Liter, PriceLitre, Total, Paid, Two, come_from.



dairyfarmdb salemilk
id : bigint(20)
InvoiceNo : varchar(20)
InvoiceDate : date
AccNo : varchar(50)
RefNo : varchar(20)
RefDate : date
Name : varchar(50)
MobileNo : varchar(20)
Email : varchar(50)
Address : varchar(500)
Litre : double
PriceLitre : double
Total : double
Paid : double
Due : double
come_from : varchar(20)
remarks : varchar(200)
created_at : timestamp
updated_at : timestamp

Figure 5.12: salemilk table structure

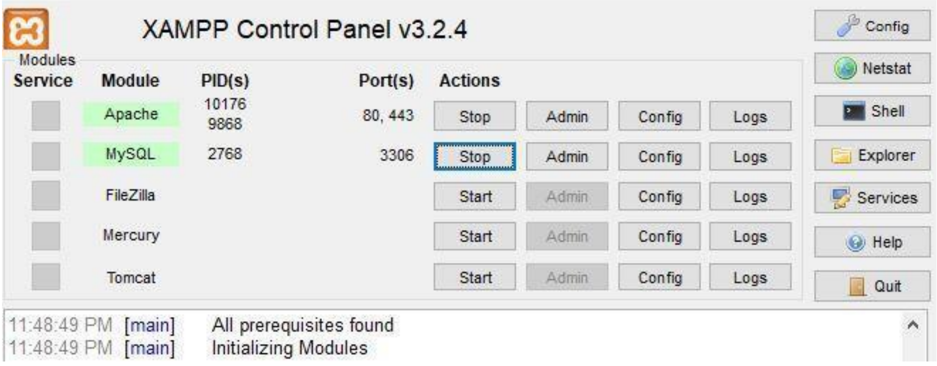
5.3 Interaction of User Interface and Database:

In this section we are using PHP Language' Laravel framework for user interface and Database is my MySQL.
Here describe how to connect the user interface with the database.

A database is a separate application that stores a collection of data. Each database has more distinct table for creating, accessing, managing & searching the data it holds. We use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys. Laravel is a framework of PHP, here we are using Laravel version 6.0 and MVC (Model-View-Controller) is a software design Pattern. Laravel by default provides the support of MySQL. MySQL is well known as open source RDBMS.

Connection process with database are given below step by step:

Step 1. First we have to start Apache and MySQL server from the XAMPP Control Panel for create a database.

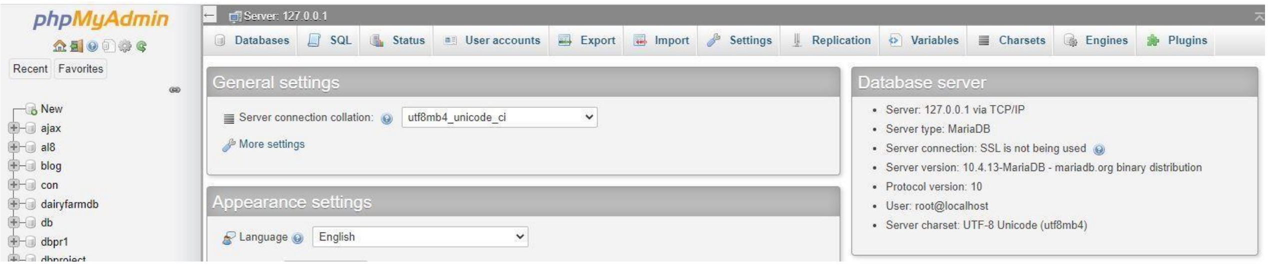


Service	Module	PID(s)	Port(s)	Actions			
□	Apache	10176 9868	80, 443	Stop	Admin	Config	Logs
□	MySQL	2768	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
11:48:49 PM  [main]   All prerequisites found
11:48:49 PM  [main]   Initializing Modules
```

Fig: 5.13 XAMPP Control Panel

Step 2. Open any browser Like Chrome, Mozilla Firefox and type in URL link of this browser localhost/phpmyadmin. Then we can get a window like fig.5.14



Server connection collation: utf8mb4_unicode_ci

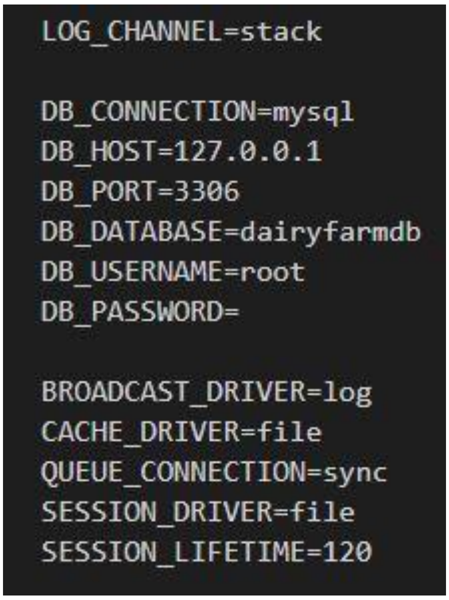
Language: English

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.13-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Fig 5.14 MySQL Database

Step 3. Now, click on the Databases tab and there, write the database named as dairyfarmdb and click on create. Then we get a database, that's name is dairyfarmdb.

Step 4. After installation a laravel project we get an .env file in this project. We have to add hostname, port number, database name, username, password like this.



```
LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dairyfarmdb
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
```

Fig 5.15 Database Connection

Step 5. Create a migration table schema for creating table using **php artisan make:migration create_tablename_table** command. After create table schema we have to write required field for store data from the user interface. And then migrate it using **php artisan migrate** command. Then we get the table in database.

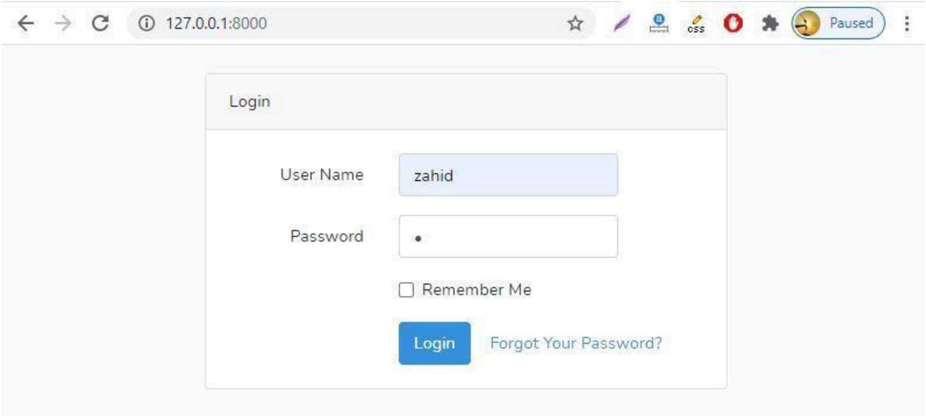
Step 6. Now we are creating a view file in resource/views directory with the name filename.blade.php. And write code for view.

Step 7. Write the route in the **web.php** file in **routes** directory.

Step 8. Now, run the following Laravel artisan command for start the server:

```
php artisan serve
```

Step 9. After run this command we get a URL and using this URL browse it in the browser. We will get a window in the browser like below:



127.0.0.1:8000

Login

User Name: zahid

Password: •

□ Remember Me

Fig 5.16 Login form (First view)

MySQL is the backbone of any web project. Whether it’s a CMS or an online task tracking platform, you need a MySQL database(s) to keep track of app and user information and server data to the application processes. Other side Users send a request from the view page and route take it and send it to the controller. The controller analyzes the request using the model if have to take data then go to the database and take data. Then shows in the view page with the data according to the user's needs.

Chapter 6: Conclusions

6.1 Conclusions

We developed the dairy farm management system that will help a dairy company to manage dairy farm-related operational information. This system is quite efficient and easy to use. Therefore, it saves time and brings growth in dairy farm yields due to improved record-keeping facilities. The system meets the needs and expectations of dairy with timely and accurate farm reports. This system will also facilitate the formulation of appropriate policies for future improvement of milk products.

6.2 Limitations

- The export system is not developed. We can’t export any kind of report in pdf, excel, xml file from this software.
- The transactions are executed in off-line mode. However online payments should be included in the payment section. But we cannot develop any online payment gateway.
- To keep the cows healthy, we regularly vaccinate the cows against various diseases so that the cows on our farm stay healthy, wealthy and strong. But we are unable to add this vaccine section in our software.

6.3 Future Scope

In future we have a plan to develop online transactions, vaccine modules, and report export to pdf, excel, xml etc. according to the stronger control and security system.