

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“Jnana Sangama”, Belagavi-590018



A
Mini Project Report
On

“Dairy Product Management System”

Submitted In Partial Fulfillment for 5th Sem

BACHELOR OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

K OMKAR PRASAD (1JB20CS043)

K S LIKHITH GOWDA (1JB20CS044)

Under the Guidance of:

Dr. BINDIYA M.K.

Associate professor,
Dept. of CSE, SJBIT

Mrs. RAJATHA

Asst. professor,
Dept. of CSE, SJBIT



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SJB INSTITUTE OF TECHNOLOGY

No. 67, BGS HEALTH AND EDUCATION CITY, DR.VISHNUVARDHAN
RD KENGERI, BENGALURU-560060, KARNATAKA, INDIA.

2022 - 2023

|| Jai Sri Gurudev ||
Sri Adichunchanagiri Shikshana Trust ®
SJB INSTITUTE OF TECHNOLOGY
BGS Health & Education City, Kengeri, Bengaluru – 560 060

Department of Computer Science & Engineering



CERTIFICATE

Certified that the Mini project work entitled “**Dairy Product Management System**” carried out by Mr. **K Omkar Prasad** , Mr. **K S Likhith Gowda** bearing USN **1JB20CS043, 1JB20CS044** are bonafide students of **SJB Institute of Technology** in partial fulfillment for 5th Semester DBMS Mini Project with Laboratory in **COMPUTER SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2022-23**. 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 prescribed for the said degree.

Signature of Guide

Dr. BINDIYA M.K.
Associate professor
Dept. of CSE, SJBIT

Signature of Guide

Mrs. RAJATHA
Asst. professor
Dept. of CSE, SJBIT

Signature of HOD

Dr. KRISHNA A N
Professor & Head
Dept. of CSE, SJBIT

EXTERNAL VIVA

Name of Examiners

1.

2.

Signature with Date



ACKNOWLEDGEMENT



We would like to express our profound grateful to His Divine Soul **Jagadguru Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Mahaswamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Mahaswamiji** for providing us an opportunity to complete our academics in this esteemed institution.

We would also like to express our profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this Mini Project Work in this admired institution.

We express our gratitude to **Dr. K. V. Mahendra Prashanth**, Principal, SJB Institute of Technology, for providing us an excellent facilities and academic ambience; which have helped us in satisfactory completion of Mini Project Work.

We extend our sincere thanks to **Dr. Krishna A N**, Head of the Department, Computer Science and Engineering for providing us an invaluable support throughout the period of our Mini ProjectWork.

We wish to express our heartfelt gratitude to our guides **DR. BINDIYA M K , Associate Professor and Mrs. RAJATHA , Asst. professor** for their valuable guidance, suggestions and cheerful encouragement during the entire period of our Mini Project Work.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents, Teaching & Non teaching staffs of the department, the library staff and all our friends, who have directly or indirectly supported us during the period of our Mini Project Work.

Regards

K OMKAR PRASAD (1JB20CS043)

K S LIKHITH GOWDA (1JB20CS044)

ABSTRACT

It is always suggested by experts to open a small dairy unit to the farmers. Setting up small scale enterprise helps farmers who are having less or no previous experience. It may well provide them an additional source of income. In future, farmers can develop their product gradually to big herds. For providing assistance to the farmers many banks or financial organizations provide loan under various development schemes. Dairying is an important source of subsidiary income to small/marginal farmers and agricultural laborers. In addition to milk, the manure from animals provides a good source of organic matter for improving soil fertility and crop yields. While milk makes up a large portion of the Indian diet with growing incomes and urbanization, it is however by no means a luxury food. The Dairy Product Management System allows the admin to maintain the details of everything with great ease. The database is then connected to the main program by using inter- connection of the Xampp server and the database already created. Dairy Product Management System to maintain correct database by providing an option to update insert and delete the details.

TABLE OF CONTENTS

Acknowledgement	I
Abstract	ii
Table of contents	iii-v
List of Figures	vi-vii
List of Tables	viii

Chapter No.	Chapter Title	Page No.
1	INTRODUCTION	1-5
	1.1 Introduction	1
	1.2 Objective	1
	1.2 Scope	1
	1.4 Advantages and Disadvantages	1-2
	1.5 Database	2-3
	1.6 The use of DBMS	4
	1.7 Advantage of a DBMS	4
	1.8 Literature Survey	5
	1.9 Motivation	5
2	REQUIREMENT SPECIFICATIONS	6-11
	2.1 Specific Requirements	6
	2.2 Functional and Non Functional Req	6-9
	2.3 HTML	9
	2.4 CSS	9
	2.5 Bootstrap	10
	2.6 SQL	10
	2.7 PHP	11

3	SYSTEM DESIGN	12-17
	3.1 Entity, Attribute and Relationships	12
	3.2 Schema Diagram	13
	3.3 ER Diagrams	13-17
4	IMPLEMENTATION	18-25
	4.1 SQL	18
	4.1.1 Basic commands in SQL	18-19
	4.2 Database Structure	19
	4.3 Table Creation	20
	4.3.1 Admin Table	20
	4.3.2 Category Table	20
	4.3.3 Company Table	21
	4.3.4 Orders Table	21
	4.3.5 Products Table	22
	4.4 Modules and Their Description	22-24
	4.5 Trigger Function	24
	4.6 Stored Procedure	25
5	SYSTEM TESTING	26-18
	5.1 Unit Testing	26
	5.2 Integration Testing	26-27
	5.3 User Acceptance Testing	27
	5.4 Module Testing	28
	5.5 Test Cases	28

6	SNAPSHOTS	29-35
	6.1 Login Page	29
	6.2 Dashboard	29
	6.3 Category Management	30
	6.4 Company Management	30
	6.5 Product Management	31
	6.6 Search Management	31
	6.7 Payment Management by cash	32
	6.8 Payment Management by cash	32
	6.9 Managing Invoice	33
	6.10 Receipt of Order	33
	6.11 Report Management	34
	6.12 Date Report Management	34
	6.13 Admin Profile	35
	CONCLUSION AND FUTURE ENHANCEMENT	36
	REFERENCES	37

LIST OF FIGURES

Figure No.	Description	Page No.
Fig 1.1	Components of a DBMS	3
Fig 2.1	Non-Functional Requirements	8
Fig 3.1	Schema Diagram	13
Fig 3.2	Entity Representation in ER Diagram	15
Fig 3.3	Attribute Representation in ER Diagram	15
Fig 3.4	One to One (1:1) Relationship	15
Fig 3.5	One to Many (1:N) Relationship	15
Fig 3.6	Many to One (N:1) Relationship	16
Fig 3.7	Many to Many (N:N) Relationship	16
Fig 3.8	Participation constraints	16
Fig 3.9	ER Diagram of Dairy	17
Fig 4.1	Types of SQL Commands	19
Fig 5.1	Testing Components	27
Fig 6.1	Snapshot of Login Page	29
Fig 6.2	Snapshot of Dashboard	29
Fig 6.3	Snapshot of Category Management	30
Fig 6.4	Snapshot of Company Management	30
Fig 6.5	Snapshot of Product Management	31
Fig 6.6	Snapshot of Search Management	31
Fig 6.7	Snapshot of Cash Payment Management	32
Fig 6.8	Snapshot of Card Payment Management	32
Fig 6.9	Snapshot of Managing Invoice	33
Fig 6.10	Snapshot of Receipt of Order	33

Fig 6.11	Snapshot of Report Management	34
Fig 6.12	Snapshot of Date Report Management	34
Fig 6.13	Snapshot of Admin Profile	35

LIST OF TABLES

Table No.	Description	Page No.
4.2	Database Structure	19
4.3	Admin Table	20
4.4	Category Table	20
4.5	Company Table	21
4.6	Orders Table	21
4.7	Products Table	21
4.8	Stored Procedure	25
5.1	Test Cases	28

Chapter 1

INTRODUCTION

1.1 Introduction

Dairy or Animal Husbandry is an important source of subsidiary income to small and marginal farmers and agricultural laborers. In addition to milk, the manure from animals provides a good source of organic matter for improving the soil fertility and gives a good yield of crop. While milk makes up a large portion of the Indian diet, with growing incomes and urbanization, it is by no means a luxury food.

1.2 Objective

The main objective of the online Dairy Product Management System is to manage the detail and sale of products, digitalized payment and bills, etc. The project is totally built at from an administrative end and thus only the administration is guaranteed to access this website. The purpose of the project is to build an application program to reduce the manual work for managing the Dairy Product s online and tracks the detail about sales, payments, bills. The main aim of this project is to convert manual automation of Dairy Product management into a fast digitalized and interactive website with more enhanced features so as to improve and increase the sale for online retailers and farmers.

1.3 Scope

India is endowed with the largest livestock population in the world. It accounts for about 57.3 per cent of the world's buffalo population and 14.7 per cent of the cattle population. Thus, there is a tremendous scope/potential for increasing the milk production through profitable Dairy Product Management System.

1.4 Advantages and Disadvantages of Dairy Product Management System

- **Advantages**

1. The initial investment in Dairy Product Management System business is low.
2. It is environment-friendly.

3. The demand for milk product is increasing rapidly.

- **Disadvantages**

1. Long hours.
2. High amount of money involved in growing and maintaining the product.
3. Many things can go wrong at a time.

1.5 DATABASE

A data is a known fact that can be recorded and that have implicit meaning. A database is a collection of related data and the way it is organized. It represents some aspect of real world, sometimes called the mini world or the universe of discourse. Changes to the mini world are reflected in the database. A database can be of any size and complexity. A database may be generated and maintained manually or it may be computerized. A computerized database may be created and maintained either by a group of application programs written specifically for the task or by a database management system.

- **DBMS**

A database management system (DBMS) is a computerized system that enables users to create and maintain a database. The DBMS is general-purpose software system that facilitates the process of defining, constructing, manipulating, and sharing databases among various users and applications maintained either by a group of application programs written specifically for the task or by a database management system. A DBMS makes it possible for end users to create, read, update and delete data in a database.

The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible. A DBMS provides concurrency, security, data integrity, consistency, controls redundancy and data independence.

Existing DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups:

- **Data definition** - Creation, modification and removal of definitions that define the organization of the data.

- **Update** - Insertion, modification, and deletion of actual data.
- **Retrieval** - Providing information the form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.
- **Administration** - Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure. Both a database and its DBMS inform to the principles of a particular database model. "Database system" refers collectively to the database model, database management system, and database.

- **Components of a DBMS**

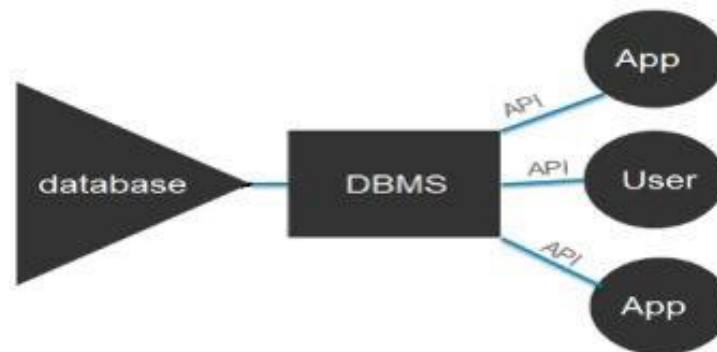


Fig 1.1 Components of a DBMS

Fig 1.1 shows the Component of a DBMS. The DBMS software is more like a wrapper around the physical database, which provides us with an easy-to-use interface to store, access and update data.

The DBMS software is capable of understanding the Database Access Language and interpret it into actual database commands to execute them on the DB.

A DBMS makes it possible for end users to create, read, update and delete data in a database.

These are the following components of 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.

1.6 Indicative areas for the use of a DBMS

- Airlines: reservations, schedules etc.
- Telecom: calls made, customer details, network usage etc.
- Universities: registration, results, grades etc.
- Sales: products, purchases, customers etc.
- Banking: all transactions etc.

1.7 Advantages of a DBMS

A Database Management System has many advantages over the traditional file system used in the earlier days, such as:

- **Data independence:** Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
- **Efficient data Access:** DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
- **Data integrity and security:** If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.
- **Data administration:** When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

1.8 Literature Survey

In the early days of computing, data management and storage was a very new concept for organizations. The traditional approach to data handling offered a lot of the convenience of the manual approach to business processes (e.g. handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

1.9 Motivation

Database Management System is a vast field in computer science and there is a lot to explore in it. This has a great potential in the real world where all that matters are data and how efficiently one stores and retrieves it. This is a great challenge posed by leading software giants like Google, Yahoo! And Oracle.

Someone who wants to learn DBMS in great depths has to understand the underlying database architecture and its variants. We wanted to explore to these depths by developing a DBMS. This project would enable us to learn the intricacies involved in developing an interpreter for the query language used as well as design the database. Additional support was given by teachers and guide to continue on the above lines.

Chapter 2

SYSTEM REQUIREMENT SPECIFICATION

A computerized way of handling information about online shopping and customers details is efficient, organized and time saving compared to manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

This software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

2.1 Specific Requirements

The specific requirements of Dairy Product Management System are stated as follows:

- **Hardware Requirements:**
 - Processor: Pentium IV or above
 - RAM - 2GB or more
 - Hard Disk - 2 GB or more
- **Software Requirements:**
 - Operating System : Windows OS, Mac OSX
 - Server :Apache
 - Database Support : My SQL7.1
 - Back-end : PHP7.1.9
- **Technologies used:**
 - Front End - HTML, CSS,BOOTSTRAP
 - Controller-PHPMYADMIN(PHP)
 - Backend –My SQL

2.2 Functional and non-functional requirements

Software system requirements are often classified as functional requirements or non-functional requirements:

- **Functional requirements:** These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.
- **Non-functional requirements:** These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements apply to the system as a whole, rather than individual system features or services.
- **Functional requirements**

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software, and the general approach taken by the organization to develop the system. When expressed as user requirements, functional requirements are those that can be understood by system users.

Functional requirements for dairy product management system include:

1. The system captures information about various dairy companies producing dairy farm products.
2. The system will add a new product that has come into the market by the company and deletes or modifies an existing product due to various reasons.
3. The system allows ordering of specific quantity of required products in case the product is getting empty.
4. The system also aims in giving a system generated invoice as a token of transaction taken place

- **Non-functional requirements**

Non-functional requirements are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and store occupancy.

Non-functional requirements are more critical than individual functional requirements. System users can usually find ways to work around a system function that

doesn't really meet their needs. However, failing to meet a non-functional requirement can mean that the whole system is unusable. For example, if an embedded control system fails to meet its performance requirements, the control functions will not operate correctly.

The implementation of these non-functional requirements may be diffused throughout the system because of two reasons:

- Non-functional requirements may affect the overall architecture of a system rather than the individual components. For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.
- A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define new system services that are required.

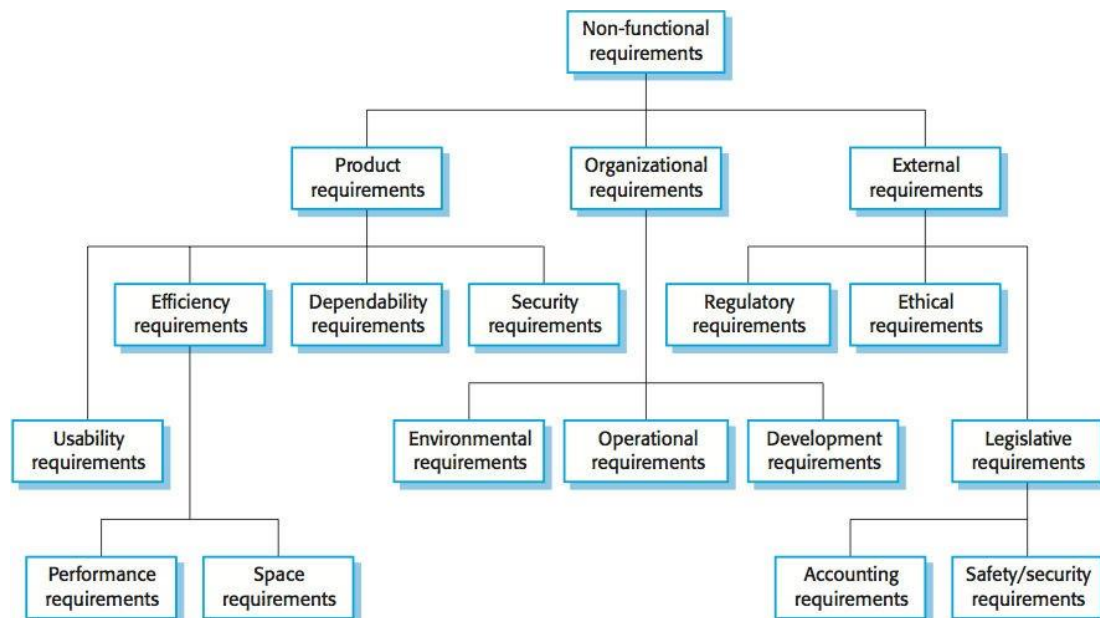


Fig 2.1: Non-Functional Requirements

Fig 2.1 shows the classification of non-functional requirements which tells non-functional requirements may come from required characteristics of the software (product requirements), the organization developing the software (organizational requirements), or from external sources:

- **Product requirements:** These requirements specify the behavior of the software. Examples include performance requirements on how fast the system must execute and how much memory it requires, reliability requirements that set out the acceptable failure rate, security requirements, and usability requirements.

- **Organizational requirements:** These requirements are broad system requirements derived from policies and procedures in the customer's and developer's organization.
- **External requirements:** This covers all requirements that are derived from factors external to the system and its development process. These may include regulatory requirements that set out what must be done for the system to be approved for use by a regulator, such as a central bank; legislative requirements that must be followed to ensure that the system operates within the law; and ethical requirements that ensure that the system will be acceptable to its users and the general public.

2.3 HTML

HTML or Hyper Text Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like), within the web page content. HTML tags most commonly come in pairs like (<p>,</p>) although some tags represent empty elements and so are unpaired like (
), for example Error! Filename not specified. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, further tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create Interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

2.4 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and Interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from

document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility. CSS can also allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader). It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

2.5 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other Interface components. Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light and dark colored tables, page headings, more prominent pull quotes, and text with a highlight.

2.6 SQL

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.

2.7 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Pre-processor. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line Interface capability and can be used in standalone graphical applications. PHP is free software released under the PHP License. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

Chapter 3

SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

System Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored. The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

3.1 Entity, Attributes and Relationships

Entity is a physical object in the real world with an independent existence. Group of distinct or similar objects is called Entity set. Each entity has attributes. Entity type is an object in the real world that can be distinguished from other objects. Each entity has attributes - the particular properties that describe it.

- **Entity types are:**

Admin (Adminid, AdminName, UserName, MobileNumber, Email, Password, AdRegdate)

Category (Categoryid, CategoryName, CategoryCode)

Company (Companyid, CompanyName)

Orders (Orderid, Productid, Quantity, InvoiceNumber, CustomerName, CustomerContactno, Paymentmode)

Products (Productid, CatName, CompName, ProductName, ProductPrice, PostingDate)

- **Relationship**

Relationship is an association among two or more entities. Relationship set collects a set of similar relationships into a relationship set. Instance of a relationship set is a set of relationship at a particular point in time.

3.2 Schema Diagram

A schema contains schema objects, which could be tables, columns, data types, stored procedures, relationships, primary keys, foreign keys. A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

ADMIN

<u>AdminId</u>	AdminName	UserName	MobileNumber	Email	Password	AdRegdate
----------------	-----------	----------	--------------	-------	----------	-----------

CATEGORY

<u>CategoryId</u>	CategoryName	CategoryCode
-------------------	--------------	--------------

COMPANY

<u>CompanyId</u>	Company Name
------------------	--------------

ORDERS

<u>OrderId</u>	ProductId	Quantity	InvoiceNumber	CustomerName	CustomerContactNo	Paymentmode
----------------	-----------	----------	---------------	--------------	-------------------	-------------

PRODUCTS

<u>ProductId</u>	CategoryName	CompName	ProductName	ProductPrice	PostingDate
------------------	--------------	----------	-------------	--------------	-------------

Fig 3.1: Schema Diagram of Dairy Products Management System

Fig. 3.1 is the database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them.

3.3 ER diagram

An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. An Entity-Relationship model is usually the result of systematic analysis to define and describe what is

important to process in area of business. An ER model does not define the business processes; it only presents a business data schema in graphical form. An ER model can also be expressed in a verbal form. Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called “primary keys”. Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagram rather than entity-relationship models.

Three main components of ERD are the entities, the relationship between those entities, and the cardinality. Cardinality defines the relation between the entities.

The four main cardinal relationships are:

- **One-to-One (1:1):** For example each customer in a database is associated with one mailing address.
- **One-to-many (1:N):** For example, a single customer might place an order for multiple products. The customer is associated with multiple entries, but all those entries have a single connection back to the same customer.
- **Many-to-One (N:1):** For example, many employees will have only one manager above them but one manager can have many employees below him.
- **Many-to-Many (M:N):** For example, at a company where all call centre agents work with multiple customers. Each agent is associated with multiple, customers, and multiple customers might also be associated with multiple agents.

Participation Constraints: There are two types

- **Total participation/Full participation:** When all entities are participating in a relationship type.
- **Partial participation:** When only few entities are participating in a relationship type.

Components of entity relationship diagram:

The three main components of an entity relationship diagram are

- Entities
- Attributes
- Relationships

The various symbols used to represent these components are:

1. Entities:



Fig 3.2: Entity Representation in ER Diagram

2. Attributes:

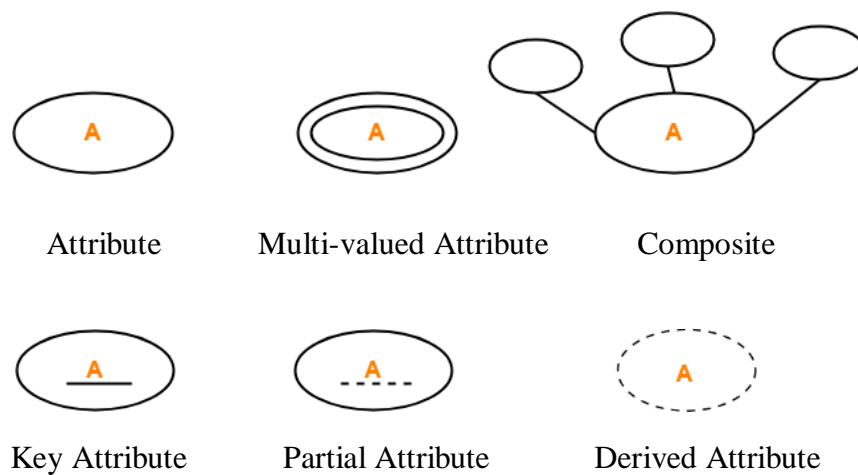


Fig 3.3: Attribute Representation in ER Diagram

3. Relationships:

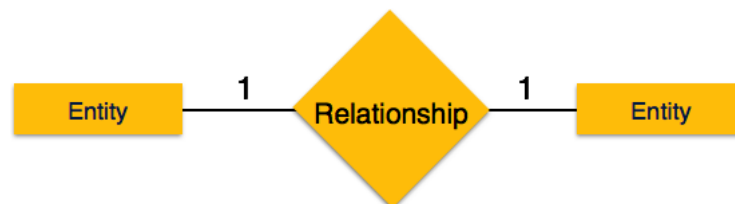


Fig 3.4: One to One (1:1) Relationship

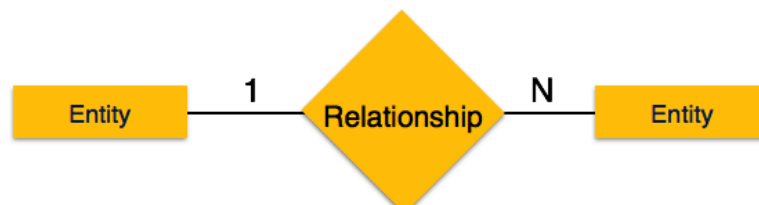


Fig 3.5: One to Many (1:N) Relationship

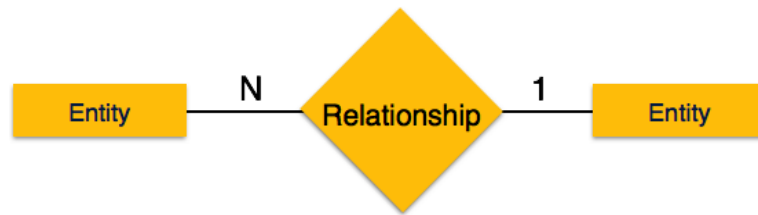


Fig 3.6: Many to One (N:1) Relationship

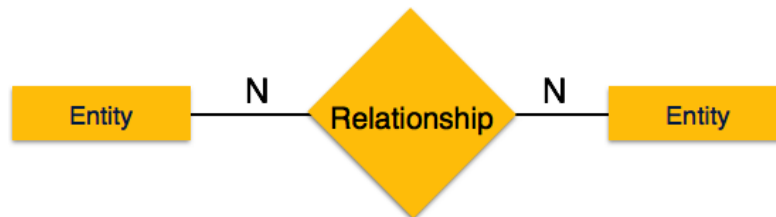


Fig 3.7: Many to Many (N:N) Relationship

4. Participation constraints:

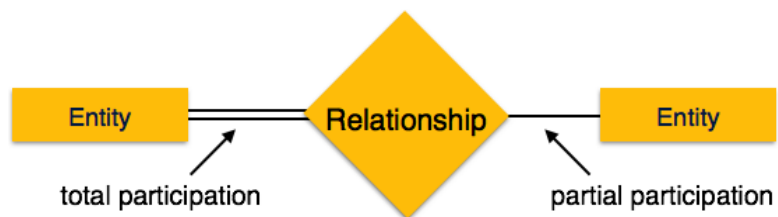


Fig 3.8: Participation constraints

- ER Diagram for Dairy Products Management System

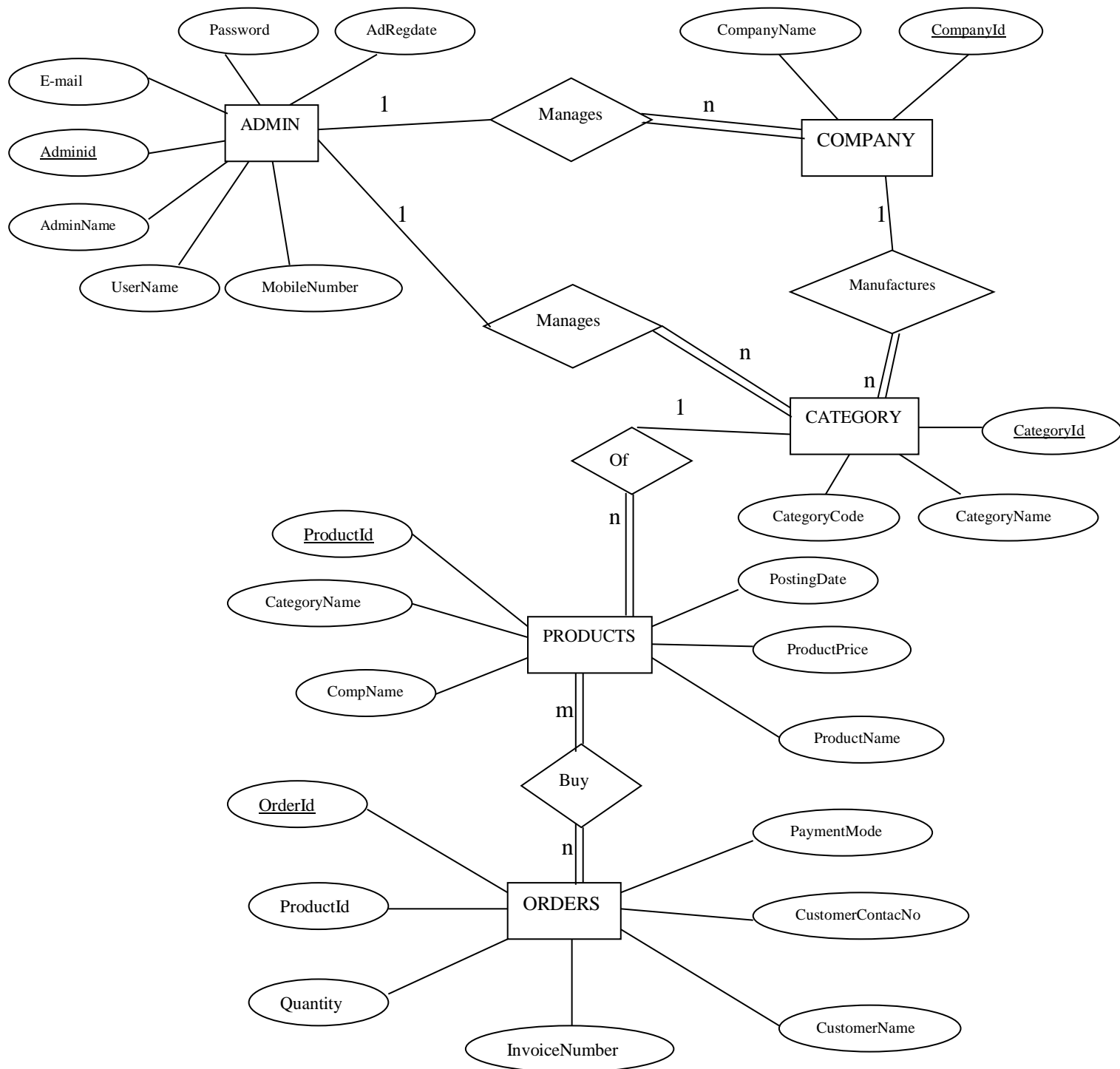


Fig 3.9: ER Diagram of Dairy Products Management System

Chapter 4

SYSTEM IMPLEMENTATION

The implementation phase is where we install the DBMS on the required hardware, optimize the database to run best on that hardware and software platform, and create the database and load the data. The initial data could be either new data captured directly or existing data imported from another DBMS. We also establish database security in this phase and give the various users that we've identified access applicable to their requirements.

4.1 SQL

MYSQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is used for the purpose of data definition and data manipulation. Program runs as a server providing multi-user access to a number of databases. MySQL is a multithreaded, multi-user SQL database management system (DBMS). It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

4.1.1 Basic commands in SQL

1. Create command: Used to create tables in SQL. Syntax is:

CREATE TABLE <TABLE NAME> (Field 1 datatype ,Field 2 datatype,.....,Field n datatype);

2. Insert command: Used to add values into the table. Syntax is:

INSERT INTO <TABLE NAME> ('value1', 'value2',..., 'valuesn');

3. Describe command: Used to know the structure of table. Syntax is:

DESC <TABLENAME>;

4. Select command: Used to retrieve data from tables. Syntax is:

SELECT * FROM <TABLE NAME>;

5. Commit command: Used to save the tables. Syntax is:

COMMIT;

6. Delete command: Used to delete records from the database. Syntax is:

DELETE FROM <TABLE NAME>;

7. Update command: Used to change values of the database. Syntax is:

UPDATE <TABLE NAME> SET <Newvalue>;

8. Drop command: Used to drop the entire table. Syntax is:

DROP TABLE <TABLENAME>;

9. Alter command: Used to alter the values in the table .Syntax is:

ALTER TABLE <TABLE NAME>;

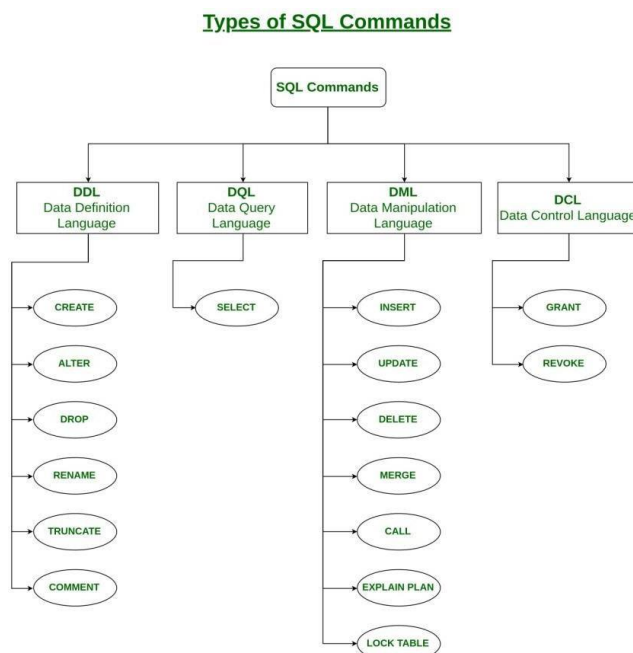


Fig 4.1: Types of SQL DML Commands

4.2 Database Structure:

Table 4.2 Database Structure

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> tbladmin		1	InnoDB	latin1_swedish_ci	16.0 K18	-
<input type="checkbox"/> tblcategory		7	InnoDB	latin1_swedish_ci	16.0 K18	-
<input type="checkbox"/> tblcompany		5	InnoDB	latin1_swedish_ci	16.0 K18	-
<input type="checkbox"/> tblorders		3	InnoDB	latin1_swedish_ci	16.0 K18	-
<input type="checkbox"/> tblproducts		8	InnoDB	latin1_swedish_ci	16.0 K18	-
5 tables	Sum	24	InnoDB	utf8mb4_general_ci	80.0 K18	0 B

A database is an organized collection of data. Instead of having all the data in a list with a random order, a database provides a structure to organize the data. One of the most common data structures is a database table. A database table consists of rows and columns.

4.3 Table Creation:

4.3.1 Admin Table

```
CREATE TABLE 'admin'(  
  'id' int(11) NOT NULL,  
  'UserName' varchar(100) NOT NULL,  
  'Password' varchar(100) NOT NULL,  
  'UpdateDate' timestamp);
```

Table 4.3: Admin Table

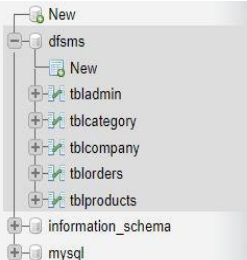
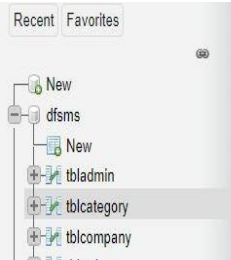


Table	Action	Rows	Type	Collation	Size	Overhead
tbladmin	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB	
tblcategory	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16.0 KiB	
tblcompany	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 KiB	
tblorders	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	
tblproducts	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16.0 KiB	
5 tables	Sum	22	InnoDB	utf8mb4_general_ci	80.0 KiB	0

4.3.2 Category Table

```
CREATE TABLE 'tblcategory'(  
  'id' int(11) NOT NULL,  
  'CategoryName' varchar(200) NOT NULL,  
  'CategoryCode' varchar(50) NOT NULL,  
  'PostingDate' timestamp);
```

Table 4.4: Category Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	CategoryName	varchar(200)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	CategoryCode	varchar(50)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	PostingDate	timestamp			Yes	current_timestamp()			Change Drop More

4.3.3 Company Table

```
CREATE TABLE 'tblcompany'(  
  'id' int(11) NOT NULL,  
  'CompanyName' varchar(150) NOT NULL,  
  'PostingDate' timestamp);
```

Table 4.5: Company Table



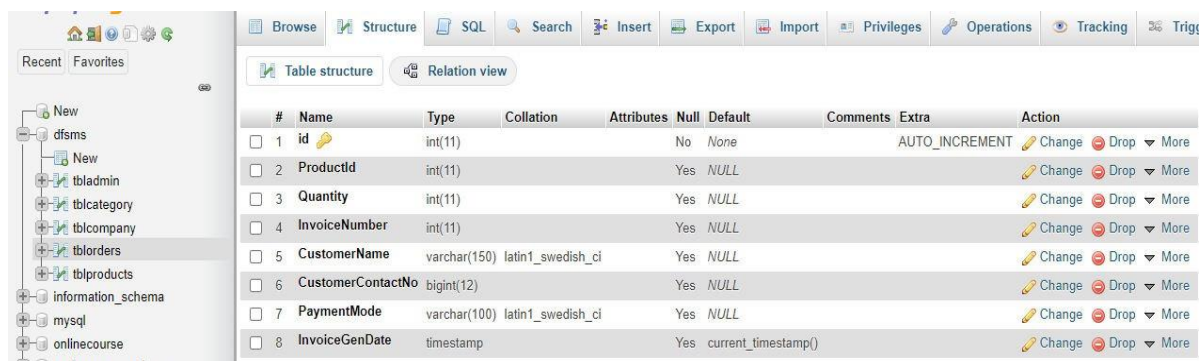
The screenshot shows the MySQL Table structure for 'tblcompany'. The table has three columns: 'id' (int(11), primary key, AUTO_INCREMENT), 'CompanyName' (varchar(150), latin1_swedish_ci), and 'PostingDate' (timestamp). The interface includes a sidebar with a tree view showing the database structure, and a top menu with options like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	CompanyName	varchar(150)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	PostingDate	timestamp			Yes	current_timestamp()			Change Drop More

4.3.4 Orders Tables

```
CREATE TABLE 'tblorders'(  
  'id' int(11) NOT NULL,  
  'ProductId' int(11) NOT NULL,  
  'Quantity' int(11) NOT NULL,  
  'InvoiceNumber' int(11) NOT NULL,  
  'CustomerContactNo' bigint(12) NOT NULL,  
  'PaymentMode' varchar(100) NOT NULL,  
  'InvoiceGenDate' timestamp);
```

Table 4.6: Orders Table



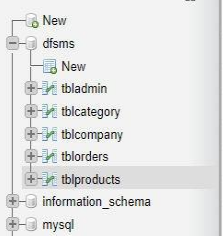
The screenshot shows the MySQL Table structure for 'tblorders'. The table has eight columns: 'id' (int(11), primary key, AUTO_INCREMENT), 'ProductId' (int(11)), 'Quantity' (int(11)), 'InvoiceNumber' (int(11)), 'CustomerName' (varchar(150), latin1_swedish_ci), 'CustomerContactNo' (bigint(12)), 'PaymentMode' (varchar(100), latin1_swedish_ci), and 'InvoiceGenDate' (timestamp). The interface includes a sidebar with a tree view showing the database structure, and a top menu with options like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	ProductId	int(11)			Yes	NULL			Change Drop More
3	Quantity	int(11)			Yes	NULL			Change Drop More
4	InvoiceNumber	int(11)			Yes	NULL			Change Drop More
5	CustomerName	varchar(150)	latin1_swedish_ci		Yes	NULL			Change Drop More
6	CustomerContactNo	bigint(12)			Yes	NULL			Change Drop More
7	PaymentMode	varchar(100)	latin1_swedish_ci		Yes	NULL			Change Drop More
8	InvoiceGenDate	timestamp			Yes	current_timestamp()			Change Drop More

4.3.5 Products Table

```
CREATE TABLE 'tblproducts'(  
    'id' int(11) NOT NULL,  
    'CategoryName' varchar(150) NOT NULL,  
    'CompanyName' varchar(150) NOT NULL,  
    'ProductName' varchar(150) NOT NULL,  
    'PostingDate' varchar(100) NOT NULL,  
    'UpdationDate' timestamp);
```

Table 4.7: Products Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	CategoryName	varchar(150)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	CompanyName	varchar(150)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	ProductName	varchar(150)	latin1_swedish_ci		Yes	NULL			Change Drop More
5	ProductPrice	decimal(10,0)			Yes	current_timestamp()			Change Drop More
6	PostingDate	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More
7	UpdationDate	timestamp			Yes	NULL		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

4.4 Modules and Their Description

- **Connectivity to Database**

```
<?php  
  
$con=mysqli_connect("localhost", "root", "", "dfsms");  
  
if(mysqli_connect_errno()){  
  
echo "Connection Fail".mysqli_connect_error();  
  
}  
  
?>
```

- **Add Product**

```
<?php  
  
session_start();
```



```
include('includes/config.php');

if (strlen($_SESSION['aid']==0)) {
header('location:logout.php');
} else{
if(isset($_POST['submit']))
{
$catname=$_POST['category'];
$company=$_POST['company'];
$pname=$_POST['productname'];
$pprice=$_POST['productprice'];

$query=mysqli_query($con,"insert into
tblproducts(CategoryName,CompanyName,ProductName,ProductPrice)
values('$catname','$company','$pname','$pprice')");
if($query){
echo "<script>alert('Product added successfully.');
```

- Invoice

```
<?php
session_start();
```

```
error_reporting(0);

include('includes/config.php');

if (strlen($_SESSION['aid']==0)) {
header('location:logout.php');
} else{

if(isset($_GET['del'])){

$cmpid=substr(base64_decode($_GET['del']),0,-5);

$query=mysqli_query($con,"delete from tblcategory where id='$cmpid'");

echo "<script>alert('Category record deleted.');
```

4.5 Trigger Function

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs.

In this project two triggers are implemented.

1. This procedure is implemented on the INSERT operation. Whenever a new category of the product is to be inserted into the database, this trigger is invoked automatically when product name is entered and the name of the product is updated in the database in uppercase letters even if the product is entered in lowercase letters.

--TRIGGER for CategoryName

CREATE DEFINER='root'@'localhost'

CREATE TRIGGER 'CategoryName'

BEFORE INSERT

ON 'tblcategory'

FOR EACHROW

```
SET NEW.CategoryName=UPPER(NEW.CategoryName);
```

2. Whenever we want to insert tuple to table 'product', then before inserting this table, trigger named 'price' will be executed. This trigger will check the ProductPrice attribute. If it is less than 1 then an error message will be printed stating "ERROR: PRODUCT PRICE MUST BE ATLEAST Re 1 !"

```
--CREATE TRIGGER `price` BEFORE INSERT ON `tblproducts`  
  
FOR EACH ROW BEGIN  
  
IF NEW.ProductPrice < 1 THEN  
  
SIGNAL SQLSTATE '45000'  
  
SET MESSAGE_TEXT = 'ERROR: PRODUCT PRICE MUST BE ATLEAST Re 1 !';  
  
END IF;  
  
END
```

4.6 Stored Procedure

This procedure is implemented to display the various orders in admin page which is:

```
DELIMITER $$
```

```
CREATE DEFINER = 'root'@'localhost'
```

```
PROCEDURE 'dispOrders'
```

```
BEGIN
```

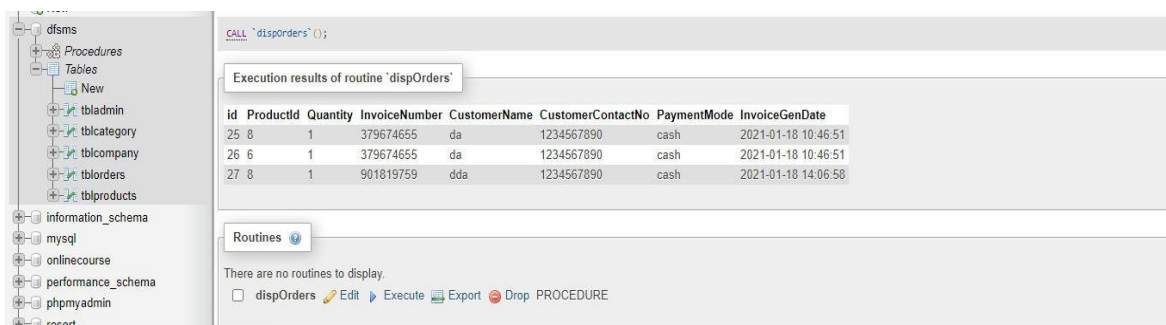
```
NOSQL
```

```
SELECT * FROM tblorders
```

```
END$$
```

```
DELIMITER ;
```

Table 4.8: Stored Procedure



id	ProductId	Quantity	InvoiceNumber	CustomerName	CustomerContactNo	PaymentMode	InvoiceGenDate
25	8	1	379674655	da	1234567890	cash	2021-01-18 10:46:51
26	6	1	379674655	da	1234567890	cash	2021-01-18 10:46:51
27	8	1	901819759	dda	1234567890	cash	2021-01-18 14:06:58

Chapter 5

SYSTEM TESTING

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.

System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

5.1 Unit testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

5.2 Integration testing

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing

Integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

- Component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components.
- System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

5.3 User acceptance testing

Users or client – This could be either someone who is buying a product (in the case of commercial software) or someone who has had a software custom built through a software service provider or the end user if the software is made available to them ahead of the time and when their feedback is sought out.

The team can be comprised of beta testers or the customer should select UAT members internally from every group of the organization so that each and every user role can be tested accordingly.

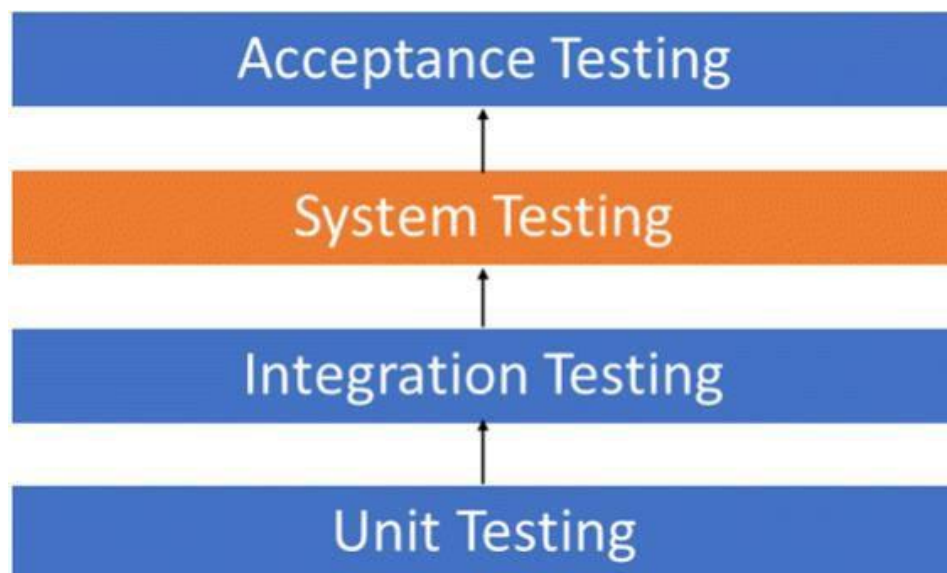


Fig 5.1: Testing Components

5.4 Module Testing

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program.

Module testing is largely a white box oriented. The objective of doing Module, testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module level testing allows to implement parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

5.5 Test Cases

Table 5.1: Test Cases

Sl no.	Test Input	Expected Results	Observed Results	Remarks
1	Insert a Record	New tuple should be inserted	Query OK 1 row affected or inserted	PASS
2	Insert a Record	Display the record	Required record displayed	PASS
3	Delete a Record	Delete the record	Query OK 1 row affected or deleted	PASS
4	Create Trigger	Trigger Created	Query OK Trigger created	PASS
5	Create Stored Procedure	Stored Procedures Created	Query OK Stored Procedure created	PASS

Table 5.1 is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

Chapter 6

RESULTS

This chapter consists of working screenshots of the project.

6.1 Login page:

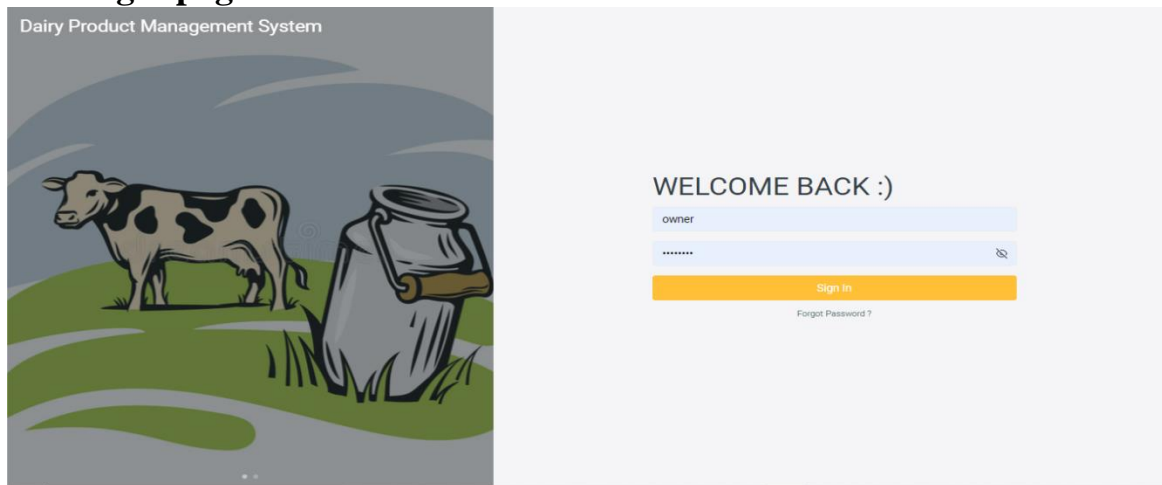


Fig 6.1: Snapshot of Login Page

Fig 6.1 shows the login page of the dairy products management system. This page asks the admin to enter the necessary credentials like Username and Password and then login to the system. If the wrong password is given then the system immediately raises an error and ask to re-enter the correct password.

6.2 Dashboard:

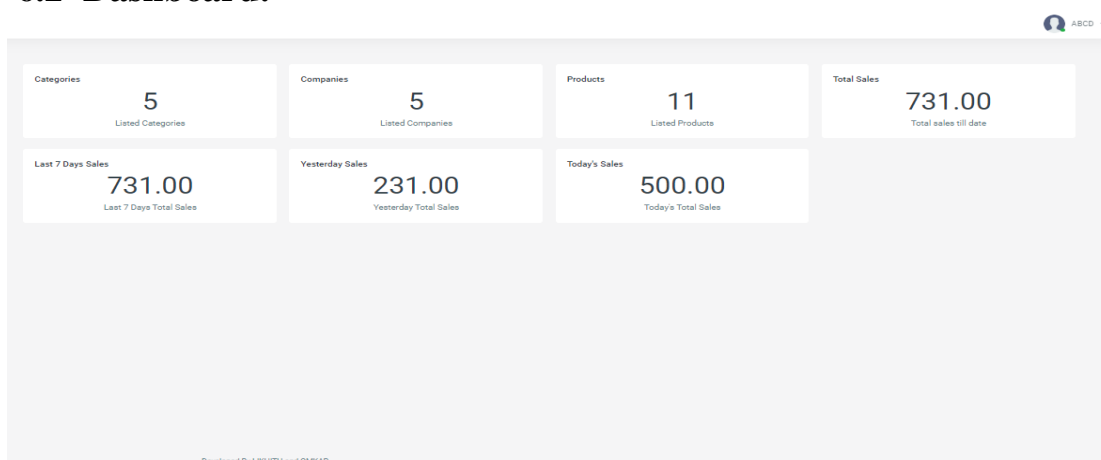
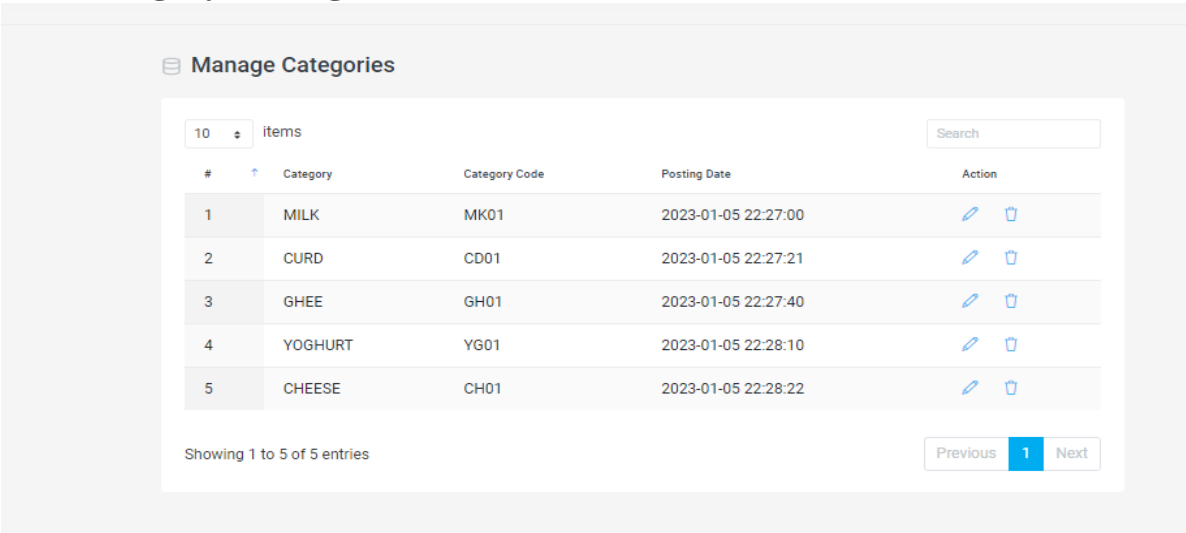


Fig 6.2: Snapshot of Dashboard

Fig 6.2 shows the Dashboard which provides an at-a-glance view of the key performance indicators relevant to a particular business process. It is considered as a form of data visualization. Here it consists the reports of Categories, Companies, Products, Total Sales, Last 7 Days Sales, Yesterday Sales, Today's Sales.

6.3 Category Management:



Manage Categories

10 items

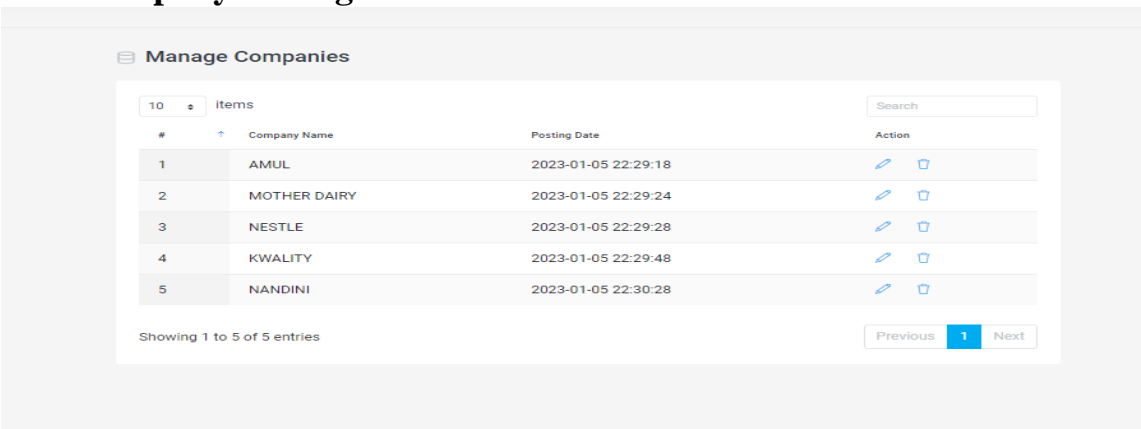
#	Category	Category Code	Posting Date	Action
1	MILK	MK01	2023-01-05 22:27:00	Edit Delete
2	CURD	CD01	2023-01-05 22:27:21	Edit Delete
3	GHEE	GH01	2023-01-05 22:27:40	Edit Delete
4	YOGHURT	YG01	2023-01-05 22:28:10	Edit Delete
5	CHEESE	CH01	2023-01-05 22:28:22	Edit Delete

Showing 1 to 5 of 5 entries [Previous](#) **1** [Next](#)

Fig 6.3: Snapshot of Category Management

Fig 6.3 shows the Category Management page that helps the admin to insert a new product or delete or modify an existing category name. Some of the well known categories are Milk, Butter, Bread, etc...

6.4 Company Management:



Manage Companies

10 items

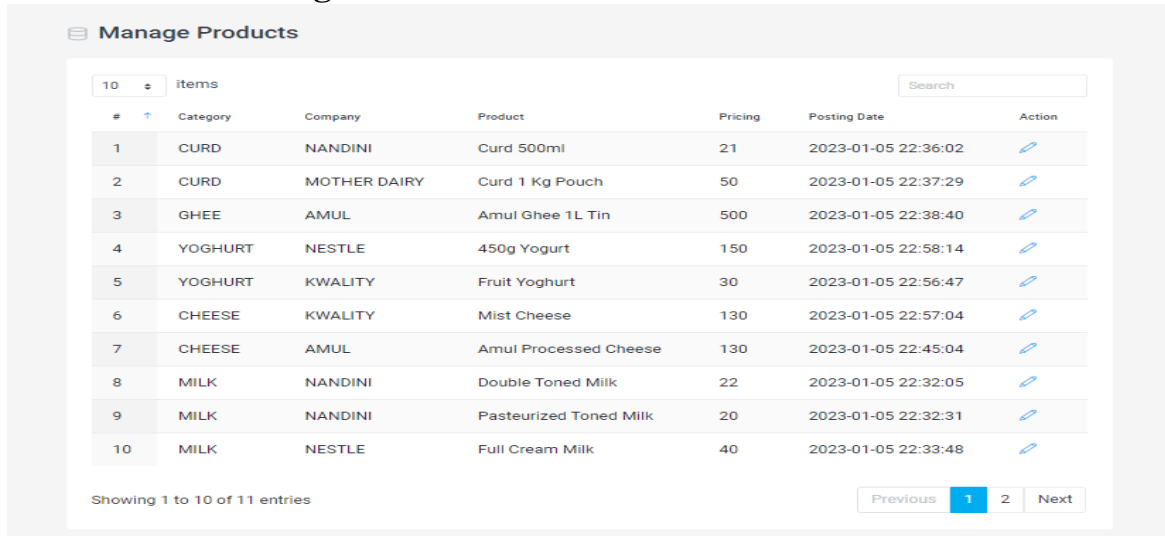
#	Company Name	Posting Date	Action
1	AMUL	2023-01-05 22:29:18	Edit Delete
2	MOTHER DAIRY	2023-01-05 22:29:24	Edit Delete
3	NESTLE	2023-01-05 22:29:28	Edit Delete
4	KWALITY	2023-01-05 22:29:48	Edit Delete
5	NANDINI	2023-01-05 22:30:28	Edit Delete

Showing 1 to 5 of 5 entries [Previous](#) **1** [Next](#)

Fig 6.4: Snapshot of Company Management

Fig 6.4 shows the Company Management page that enables the admin to insert a new Dairy Company into a system or delete/modify an existing company. Some of the well known Dairy Companies are Amul , Kwality, Nandini, etc...

6.5 Product Management:



Manage Products

10 items

#	Category	Company	Product	Pricing	Posting Date	Action
1	CURD	NANDINI	Curd 500ml	21	2023-01-05 22:36:02	Edit
2	CURD	MOTHER DAIRY	Curd 1 Kg Pouch	50	2023-01-05 22:37:29	Edit
3	GHEE	AMUL	Amul Ghee 1L Tin	500	2023-01-05 22:38:40	Edit
4	YOGHURT	NESTLE	450g Yogurt	150	2023-01-05 22:58:14	Edit
5	YOGHURT	KWALITY	Fruit Yoghurt	30	2023-01-05 22:56:47	Edit
6	CHEESE	KWALITY	Mist Cheese	130	2023-01-05 22:57:04	Edit
7	CHEESE	AMUL	Amul Processed Cheese	130	2023-01-05 22:45:04	Edit
8	MILK	NANDINI	Double Toned Milk	22	2023-01-05 22:32:05	Edit
9	MILK	NANDINI	Pasteurized Toned Milk	20	2023-01-05 22:32:31	Edit
10	MILK	NESTLE	Full Cream Milk	40	2023-01-05 22:33:48	Edit

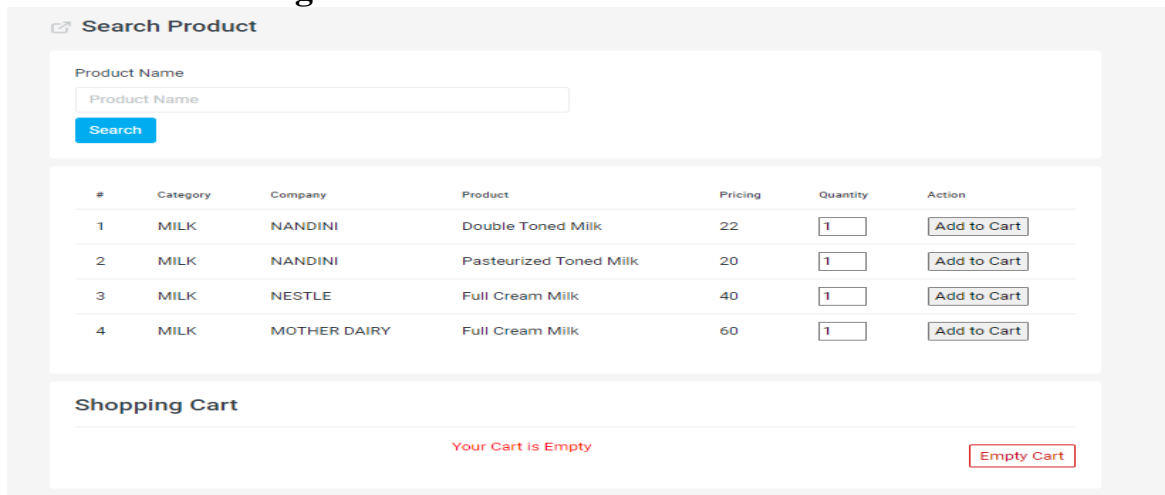
Showing 1 to 10 of 11 entries

Previous **1** 2 Next

Fig 6.5: Snapshot of Product Management

Fig 6.5 shows the Product Management page that enables the admin to edit the category field, company field, and product field incase wrong details are entered. This page also allows to set a reasonable price on every product.

6.6 Search Management:



Search Product

Product Name

#	Category	Company	Product	Pricing	Quantity	Action
1	MILK	NANDINI	Double Toned Milk	22	<input type="text" value="1"/>	<input type="button" value="Add to Cart"/>
2	MILK	NANDINI	Pasteurized Toned Milk	20	<input type="text" value="1"/>	<input type="button" value="Add to Cart"/>
3	MILK	NESTLE	Full Cream Milk	40	<input type="text" value="1"/>	<input type="button" value="Add to Cart"/>
4	MILK	MOTHER DAIRY	Full Cream Milk	60	<input type="text" value="1"/>	<input type="button" value="Add to Cart"/>

Shopping Cart

Your Cart is Empty

Fig 6.6: Snapshot of Search Management

Fig 6.6 shows the Search Management page which enable to search if a particular product exist or not. In addition this page also allows to specify the quantity of desired product required and order the same.

6.7 Payment Management through Cash:

Product Name

Product Name

Search

Shopping Cart

Empty Cart

Product Name	Category	Company	Quantity	Unit Price	Price	Remove
Double Toned Milk	MILK	NANDINI	1	22	22.00	Remove Item
Total:			1		22.00	

Customer Name

Customer Name

Customer Mobile Number

Mobile Number

Payment Mode

☒ Cash

☐ Card

Checkout

Developed By LIKHITH and OMKAR

Fig 6.7: Snapshot of Cash Payment Management

Fig 6.7 shows the Cash Mode payment. Once the required quantity of desired product is selected the page asks for payment through either cash or card. The cash option is selected.

6.8 Payment Management through Card option:

Product Name

Product Name

Search

Shopping Cart

Empty Cart

Product Name	Category	Company	Quantity	Unit Price	Price	Remove
Double Toned Milk	MILK	NANDINI	1	22	22.00	Remove Item
Total:			1		22.00	

Customer Name

Customer Name

Customer Mobile Number

Mobile Number

Payment Mode

☐ Cash

☒ Card

Checkout

Developed By LIKHITH and OMKAR

Fig 6.8: Snapshot of Card Payment Management

Fig 6.8 shows the Card Mode payment. Once the required quantity of desired product is selected the page asks for payment through either cash or card. The card option is selected.

6.9 Managing Invoice:

#	Invoice Number	Customer Name	Customer Contact No.	Payment Mode	Invoice Gen. Date	Action
1	532613379	omkar	9898989898	card	2023-01-05 12:58:54	✉
2	718947279	abd	1111111111	cash	2023-01-05 13:00:39	✉
3	169191257	omkar	9898989898	cash	2023-01-05 22:54:07	✉
4	314610362	xyz	9898989898	cash	2023-01-06 06:58:02	✉

Showing 1 to 4 of 4 entries

Previous 1 Next

Fig 6.9: Snapshot of Managing Invoice

Fig 6.9 shows the Managing Invoice page which allows the admin to manage the various Invoice generated upon every order. This page also gives a short hand information about Invoice number, customer name, contact number, payment mode, and Invoice generated date.

6.10 Receipt of Order:

#	Product Name	Category	Company	Quantity	Unit Price	Price
1	Curd 500ml	CURD	NANDINI	1	21	21.00
Total						21.00

Fig 6.10: Snapshot of Receipt of Order

Fig 6.10 shows the Receipt of Order. It includes the cost of the products purchased or services rendered to the buyer. Invoices can also serve as legal records, if they contain the names of the seller and client, description and price of goods or services, and the terms of payment.

6.11 Report Management

B/w Date Report Date Selection

From Date
01-01-2023

To Date
06-01-2023

Submit

Fig 6.11: Snapshot of Report Management

Fig 6.11 shows the Report Management page which allows the admin to select two different dates in order to get a view of the orders placed during that interval.

6.12 Date Report Management

B/w Dates report from 2023-01-01 to 2023-01-06

10 items

Search

#	Invoice Number	Customer Name	Customer Contact No.	Payment Mode	Invoice Gen. Date	Action
1	532613379	omkar	9898989898	card	2023-01-05 12:58:54	✉
2	718947279	abd	1111111111	cash	2023-01-05 13:00:39	✉
3	169191257	omkar	9898989898	cash	2023-01-05 22:54:07	✉
4	314610362	xyz	9898989898	cash	2023-01-06 06:58:02	✉

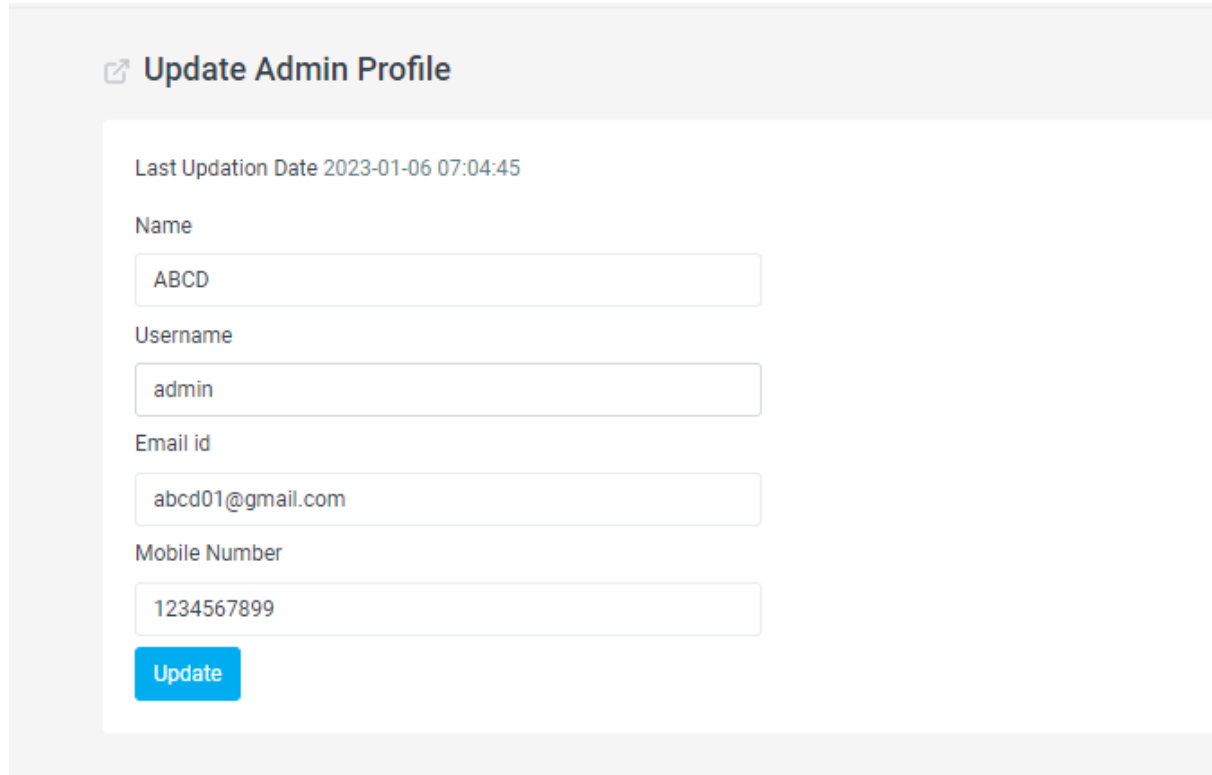
Showing 1 to 4 of 4 entries

Previous 1 Next

Fig 6.12: Snapshot of Date Report Management

Fig 6.12 shows the Date Report Management which gives a list of orders placed between any two dates that the admin desires to get a statistical analysis.

6.13 Admin Profile



The screenshot displays a web interface for updating an admin profile. At the top, there is a header bar with a blue icon and the text "Update Admin Profile". Below this, the "Last Updation Date" is shown as "2023-01-06 07:04:45". The form contains five input fields: "Name" with the value "ABCD", "Username" with "admin", "Email id" with "abcd01@gmail.com", and "Mobile Number" with "1234567899". A blue "Update" button is positioned at the bottom left of the form area.

Fig 6.13: Snapshot of Admin Profile

Fig 6.13 shows the Admin Profile page that allows the admin to update the required information.

CONCLUSION AND FUTURE ENHANCEMENT

With the theoretical inclination of our syllabus, it becomes very essential to take the utmost advantage of this opportunity of gaining practical experience that comes along. The construction of this mini project "DAIRY PRODUCTS MANAGEMENT SYSTEM" was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts, thus making us more competent.

This Database was mainly created to maintain and keep track of dairy products sale. The database is therefore more efficient mechanism to store and organize data the spreadsheet as it allows a centralized facility that can easily be modified. By developing this automated system, it has reduced the time and work load per person required to maintain transaction of sales on daily basis. This system has also reduced the need for paper required to document the sales and quantity of Dairy Product.

FUTURE ENHANCEMENT:

- Using the available data a chart or a graph is to be plotted to derive the sales taken place on a regular basis in order to know the increase or decrease in the sales.
- Allow retailers to use the website so has to infer the increment in sales in dairy product shop.

REFERENCES

The following books were referred during the analysis and execution phase of the project

- [1] Ramez Elmasri, Shamikant B. Navathe; Fundamentals of Database Management System; 5 th Edition; Pearson;2017.
- [2] Robin Nixon, “Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5”, 4 thEdition, O“Reilly Publications, 2015
- [3] Dr. Rajiv Chopra; Database Management Systems; 5th Edition; SChand;2017.
- [4] W3Schools (PHP reference) —<https://www.w3schools.com/php/default.asp>
- [5] W3Schools (HTML reference) —<https://www.w3schools.com/css/default.asp>
- [6] PHP Official Documentation —<http://php.net/docs.php>
- [7] The MySQL Documentation —<https://dev.mysql.com/doc/refman/8.0/en/>