Chapter 1

INTRODUCTION

The Dairy Farm Shop Management System (DFSMS) is a web based application that can be accessed over the web. This system can be used to automate the workflow of dairy shop and their invoices.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of MySQL Server and all the user interfaces has been designed using the PHP technologies. The database connectivity is planned using the "MySQL Connection" methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

DFSMS is a web-based application which manages the products of dairy shop. It has one module i.e. admin who manages all the functions of the dairy shop.

1.1 Admin Features:

Dashboard: In this section, admin can see all detail in brief like Total listed categories, companies, products and also see the sales.

Category: In this section, admin can add new categories and edit, delete old categories.

Company: In this section, admin can add new companies and edit, delete old companies.

Product: In this section, admin can add new products and edit old products.

Search: In this section, admin can search for a product then add the product into the cart and generate invoice /receipt.

Invoices: In this section, admin can view all generated invoices/receipts.

Reports: In this section, admin can generate two reports, one is B/w date and another one is for sales.

Admin can also update his profile, change the password and recover the password.

Chapter 2

REQUIREMENT

2.1 Tool and Technology Used

2.1.1 Introduction to HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

2.1.2 Introduction to CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any

XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging WebPages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

2.1.3 Introduction to PHP:

PHP (Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. The best things in using PHP are that it is extremely simple for a new comer, but offers many advanced features for a professional programmer, Don’t be afraid reading the long list of PHP's features, You can jump in a short time and start writing simple scripts in a few hours. PHP is an acronym for "PHP: Hypertext Preprocessor", PHP is a widely-used, open source scripting language; PHP scripts are executed on the server.PHP is free to download and use.

2.1.4 PHP File

PHP files can contain text, HTML, CSS, JavaScript, and PHPcode.PHP code are executed on the server, and the result is returned to the browser as plain HTML, PHP files have extension".php"

PHP can generate dynamic page content.PHP can create, open, read, write, delete, and close files on the server.PHP can collect form data, PHP can send and receive cookies.PHP can add, delete, modify data in your database, PHP can be used to control user-access PHP can encrypt data. With

PHP you are not limited to output HTML; you can output images, PDF files, and even flash movies. You can also output any text, such as XHTML and XML.PHP runs on various platforms (Windows, Linux, UNIX, Mac OS X, etc.)PHP is compatible with almost all servers used today (Apache, IIS, etc.)PHP supports a wide range of databases.PHP is free. Download it from the official PHP resource: www.php.net.PHP is easy to learn and runs efficiently on the server side.

2.1.5 APACHE SERVER:

The Apache HTTP Server, called **Apache**, is a free and open-source cross .platform web server, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. The Apache HTTP Server is cross-platform; as of 1 June 2017 92%of Apache HTTPS Server copies run on Linux distributions. Version 2.0 improved support for non- UNIX operating systems such as Windows and os/2.16l Old versions of Apache were ported to run on OpenVMS and NetWare.

Originally based on the NCSA HTTP server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web, quickly overtaking NCSA HTTP as the dominant HTTP server, and has remained most popular since April 1996.In 2009, it became the first web server software to serve more than 100 million websites. As of July 2016, it was estimated to serve 46%of all active websites and 43% of the top million websites.

2.1.6 XAMPP:

It is a free and open source cross-platform web server solution stack package developed by ApacheFriends,2 consisting mainly of the Apache HTTP Server, Maria DB database ,and interpreters for scripts written in the PHP and programming languages.L3I4 XAMPP stands for Cross-Platform(X),Apache(A),Maria DB(M),PHP(P)and Perl(P). Xampp is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for Everything needed to set up a web server application (Apache), database (Maria DB), and scripting language (PHP)-is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac
and Windows. Since most actual web server deployments use the same components as XAMPP,

xampp makes transitioning from a local test server to a live server extremely easy as well.

Chapter 3

REQUIREMENTS SPECIFICATION

3.1 Hardware Requirements

Processor	Intel(R) Pentium(R) CPU N3700 @ 1.60 GHz
RAM	4 GB
System type	64-bit Operating System, x64 based processor
Output device	Monitor (1366*768 Resolution)
Input device	Keyboard, Mouse

3.2 Software Requirements

Language Used	PHP
Database	My SQL
User Interface Design	HTML, AJAX, JQUERY, JAVASCRIPT
Web Browser	Mozilla, Google Chrome, IE8, OPERA
Software	XAMPP Server

Chapter 4

DESIGN AND IMPLEMENTATION

4.1 ER Diagram

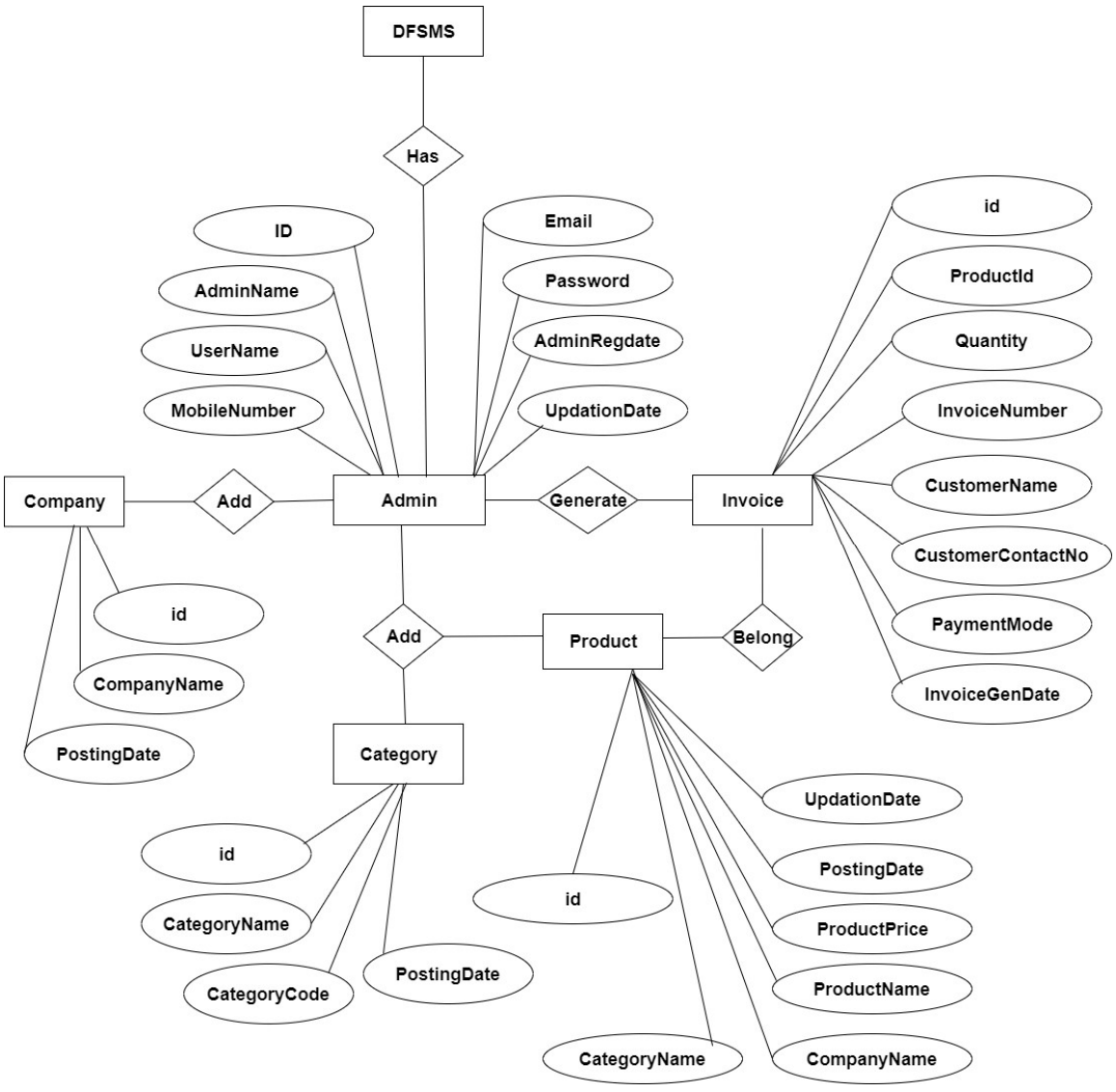


Fig. 4.1: ER diagram of Dairy Farm Shop Management System

4.2 UML Diagram

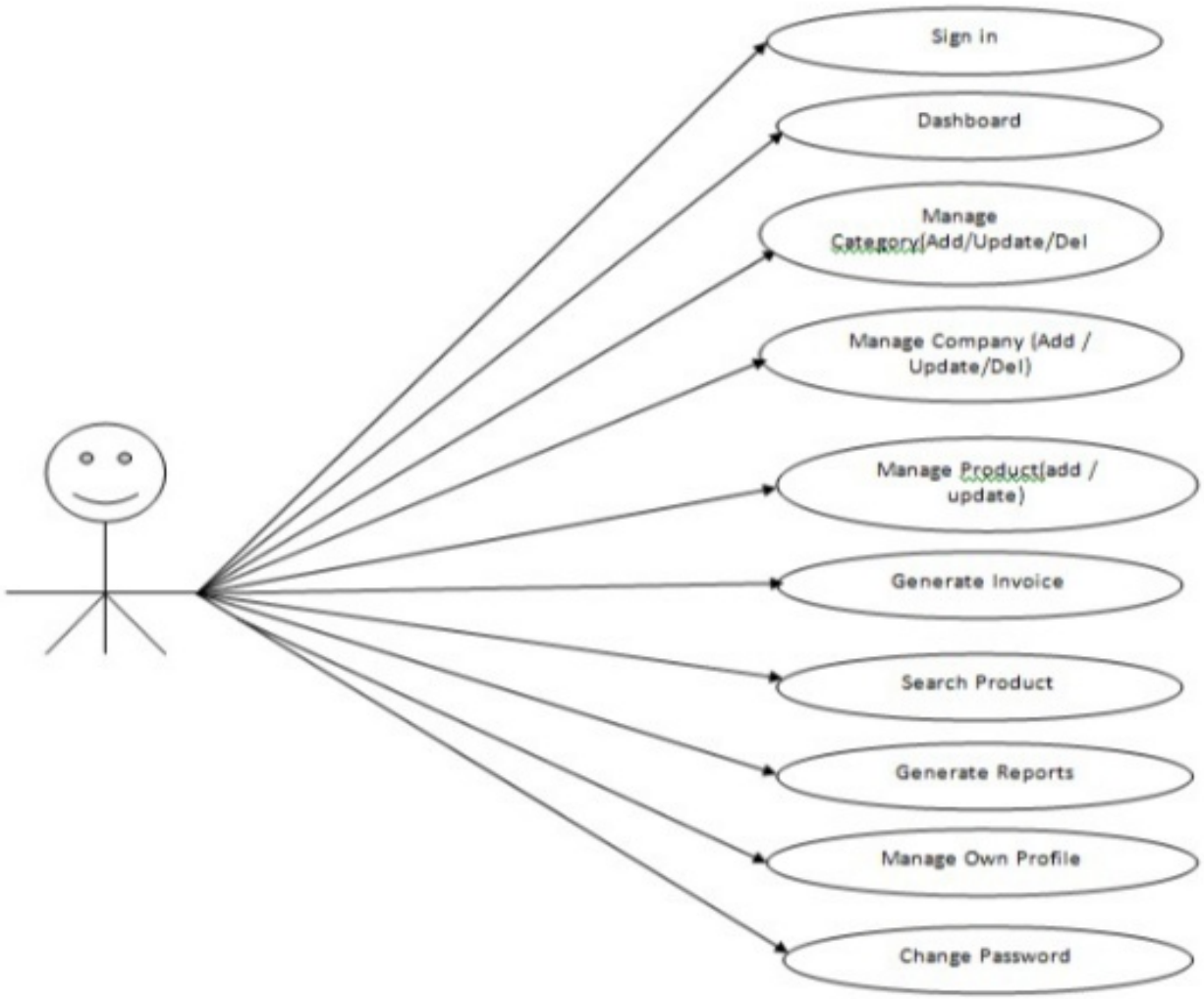


Fig 4.2 UML Diagram for dairy farm shop management system

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Chapter 5

TESTING

INTRODUCTION

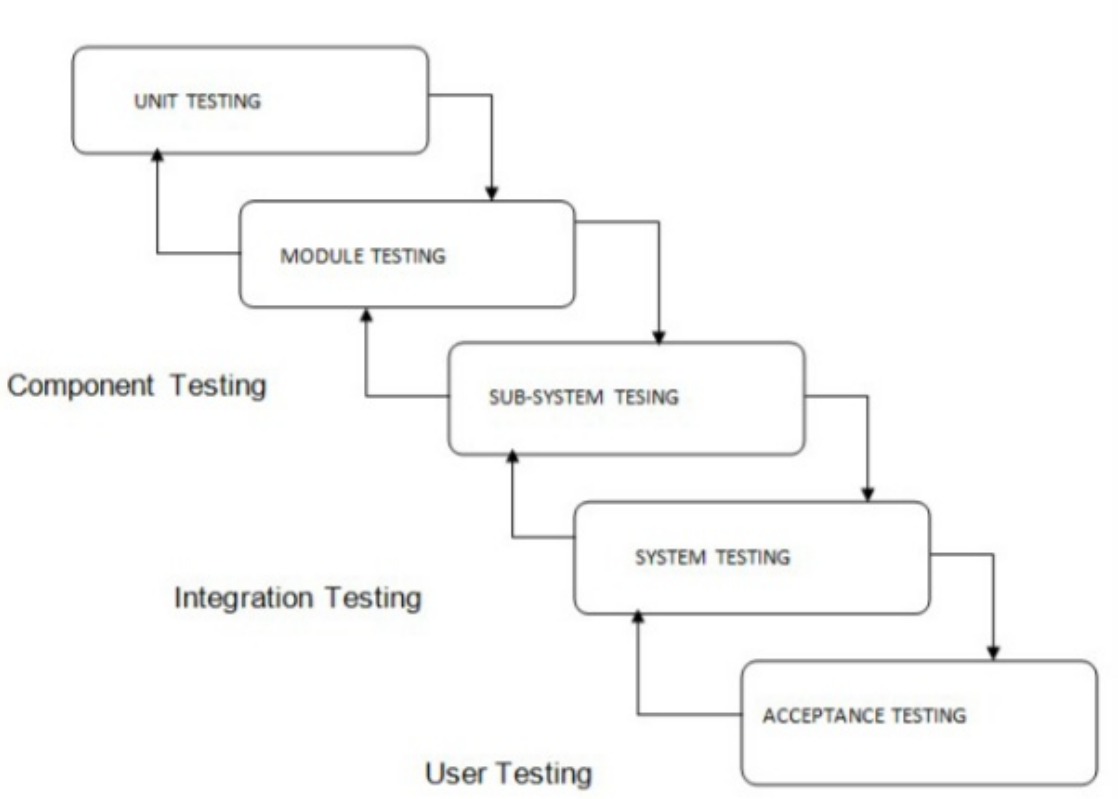
Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.



5.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

➢ WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

5.2 BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V (G) =E-N+2 or

V (G) =P+1 or

V (G) =Number of Regions

Where V (G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

5.3 CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

5.4 DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

5.5 LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.

- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Chapter 6

DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

Dairy Farm Shop Management System (DFSMS) contains 5 MySQL tables:

Tbladmin: This table stores admin login details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(5)			No	*None*		AUTO_INCREMENT
2	AdminName	varchar(45)	latin1_swedish_ci		Yes	*NULL*		
3	UserName	char(45)	latin1_swedish_ci		Yes	*NULL*		
4	MobileNumber	bigint(11)			Yes	*NULL*		
5	Email	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
6	Password	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
7	AdminRegdate	timestamp			Yes	current_timestamp()		
8	UpdationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

Table 6.1: Admin table

Tblcategory: This table stores category details

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

Table 6.2: Category table

Tblcompany: This table stores company details

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

Table 6.3: Company table

Tblorders: This table stores invoice details of dairy products.

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

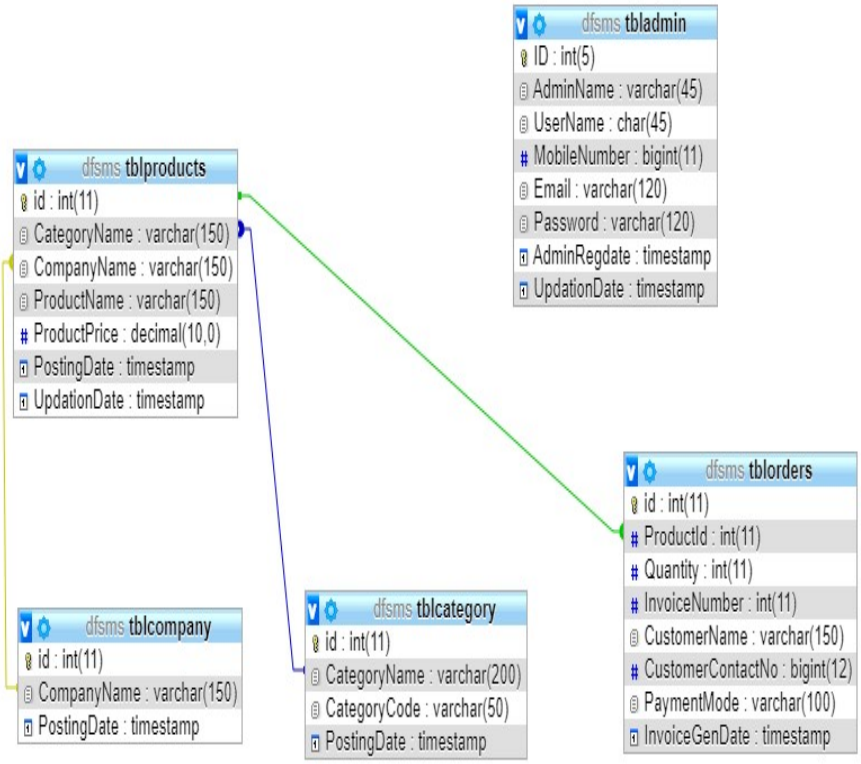
Table 6.4: Orders table

Tblproducts: This table store dairy product details.

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

Table 6.5: Products table

MySQL Tables Relationship



Chapter 7

SNAPSHOTS

A screenshot is a digital image of what should be visible on a monitor, TV, or other visual output device. A common screenshot is created by the operating system or software running on the device. A screenshot capture may also be created by taking a photo of the screen.

Login Page

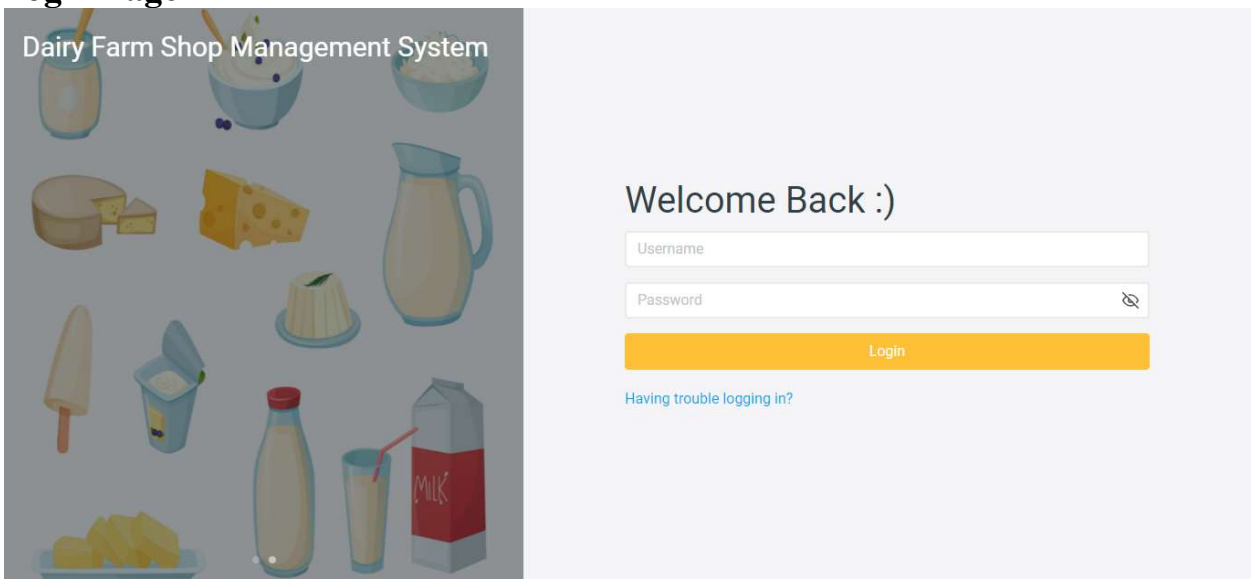


Fig. 7.1: Login page

Dashboard

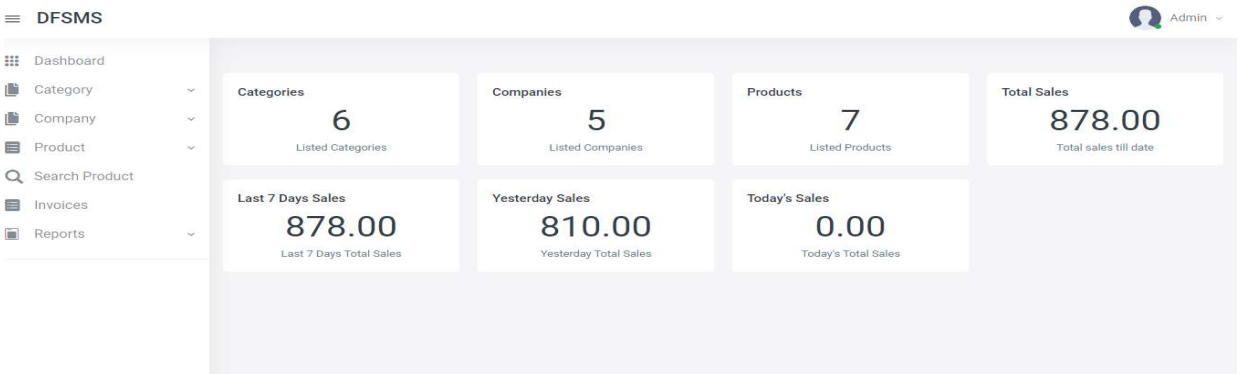


Fig. 7.2: Dashboard page

Admin Profile

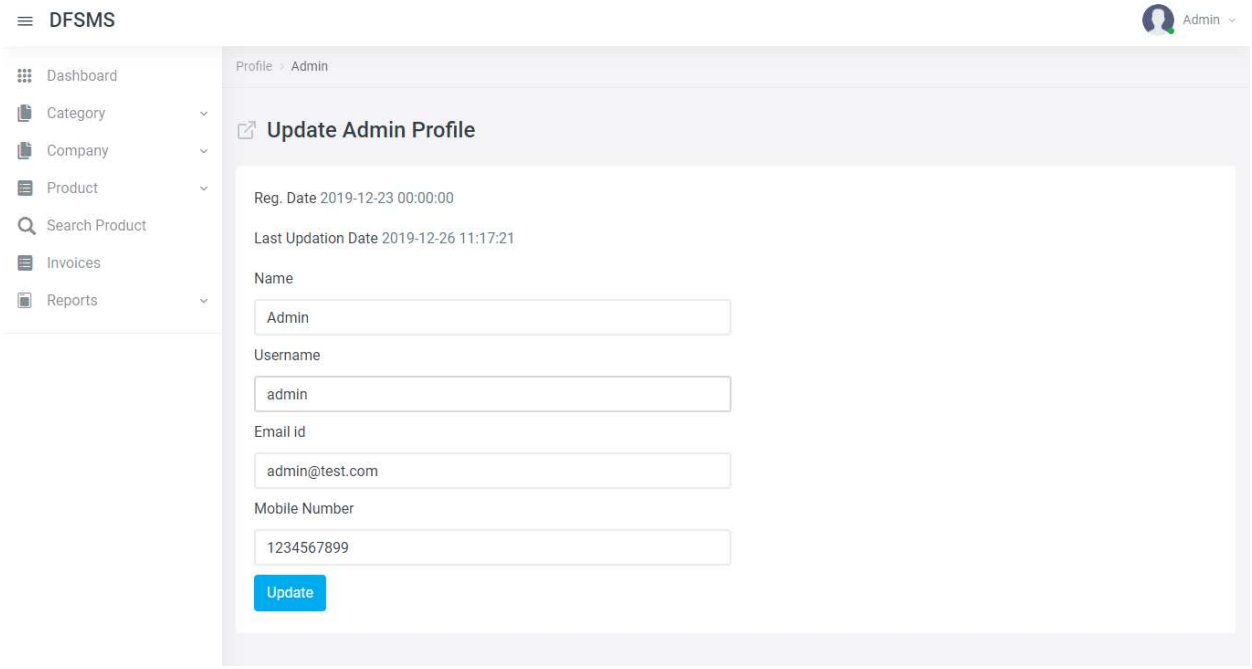


Fig. 7.3: Admin profile page

Change Password

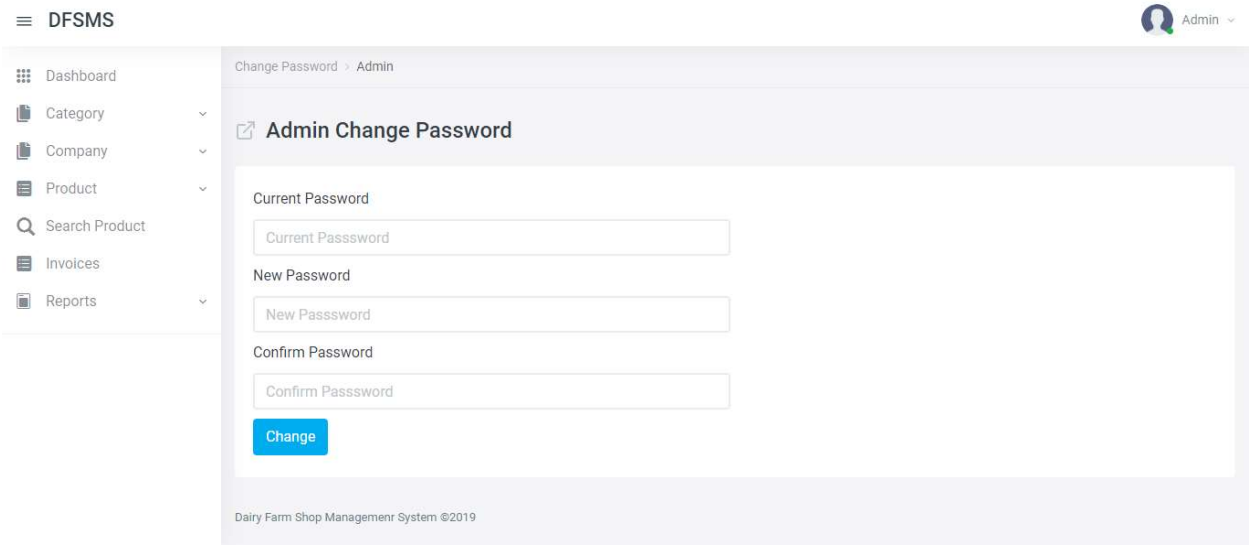


Fig. 7.4: Change password page

Add Category

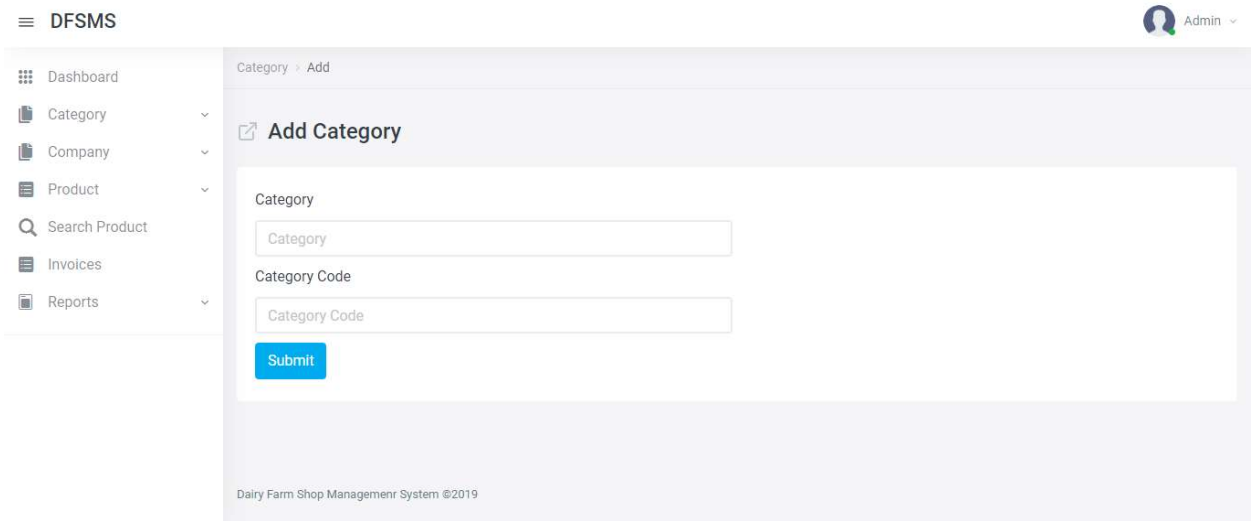


Fig.7.5: Add category page

Manage Category

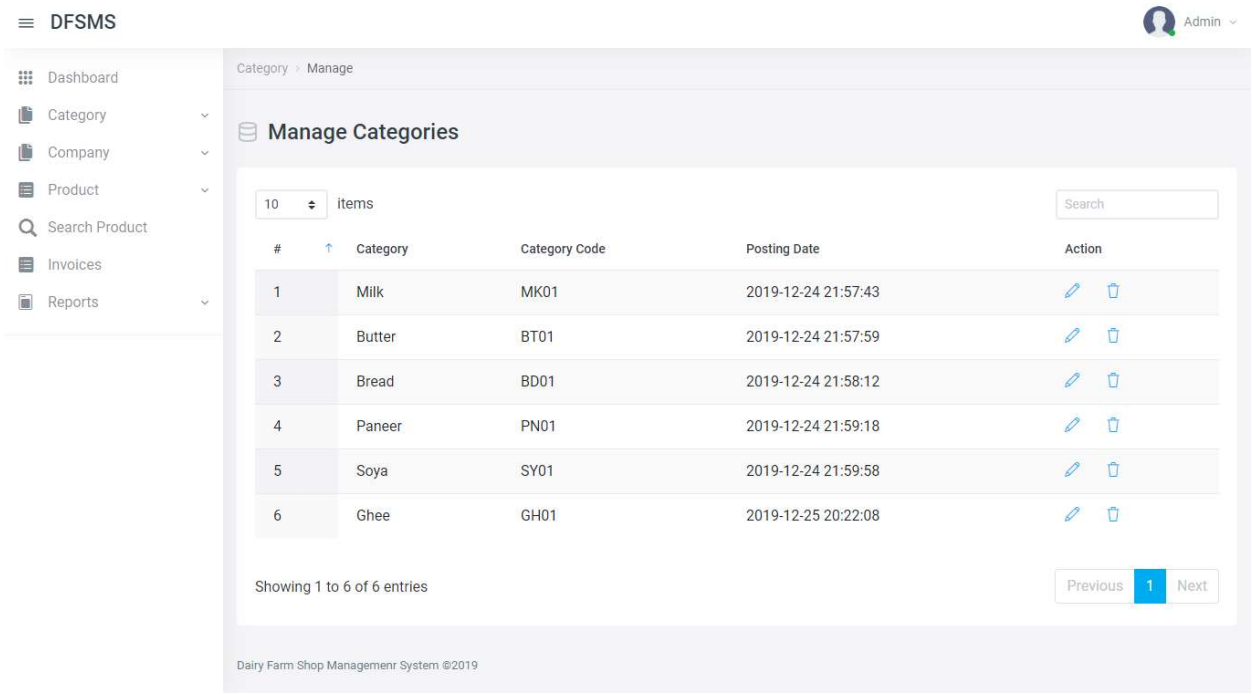


Fig.7.6: Manage category page

Update Category

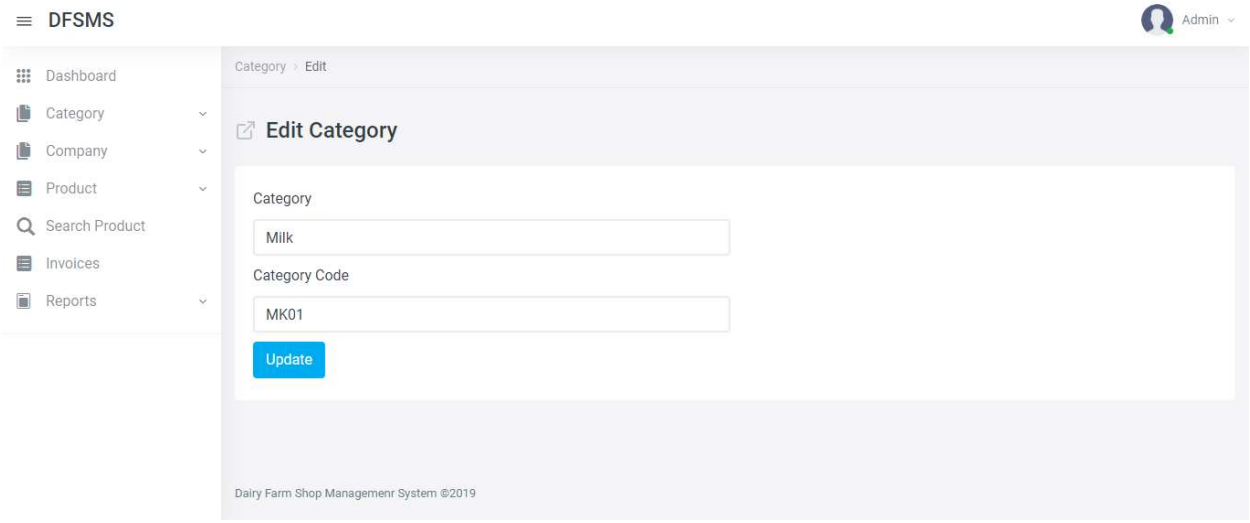


Fig. 7.7: Update category page

Add Company

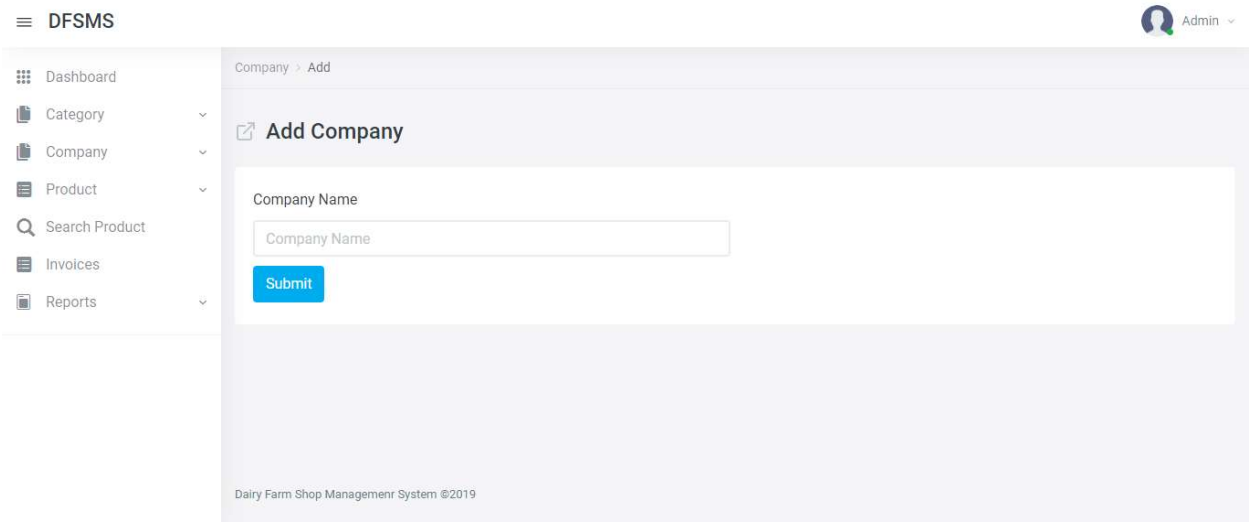


Fig. 7.8: Add company page

Manage Company

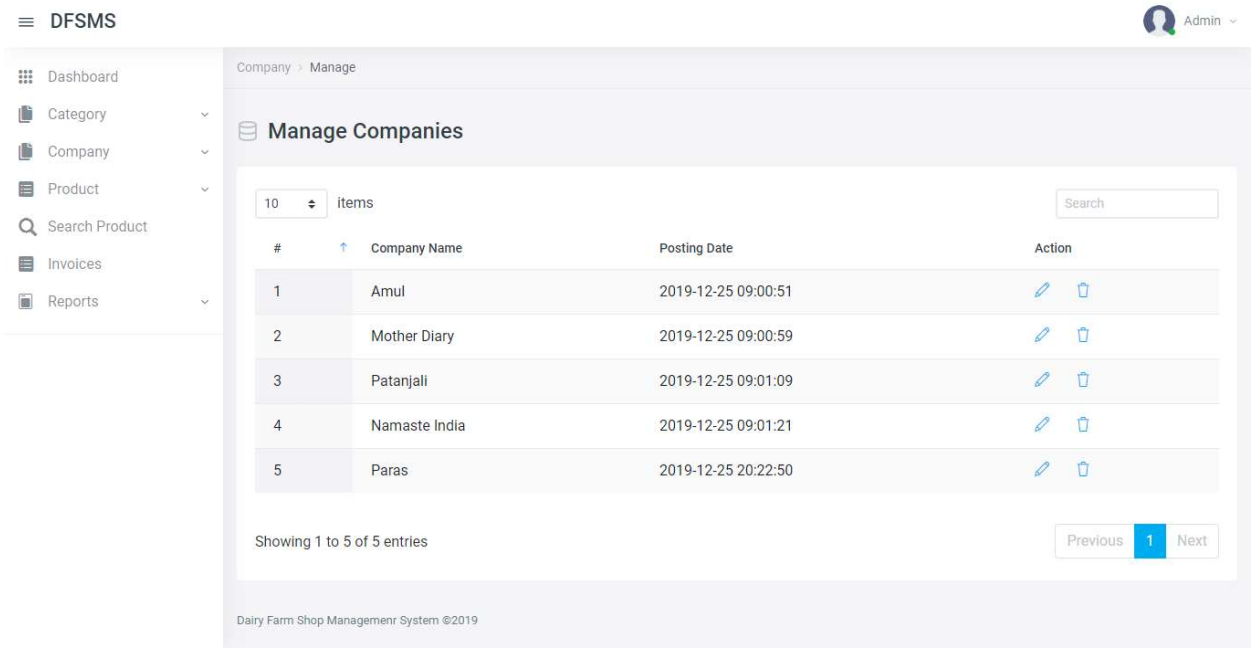


Fig.7.9: Manage company page

Update Company

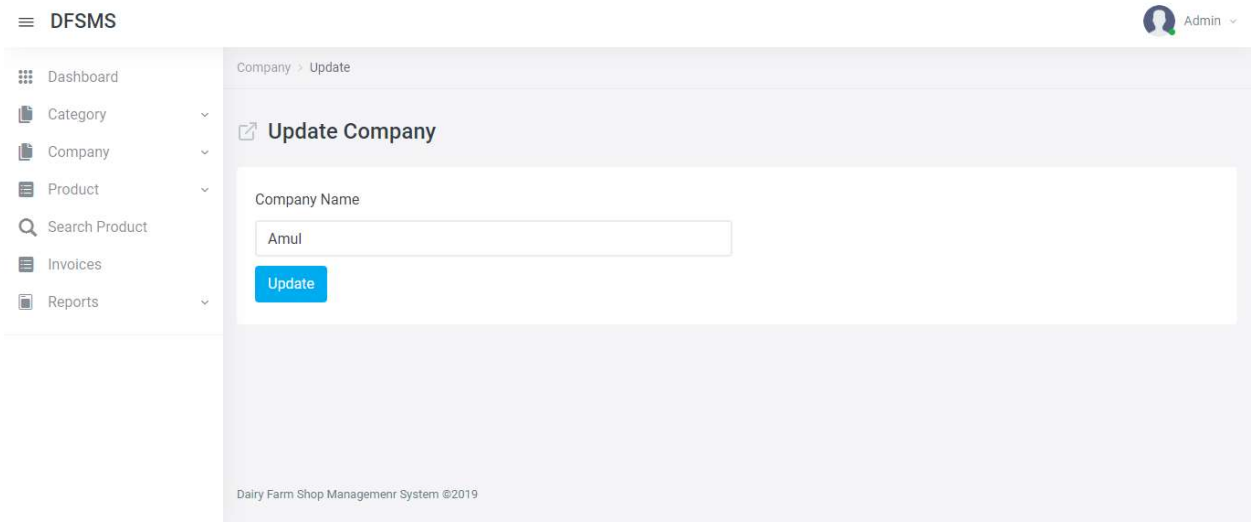


Fig.7.10: Update company page

Add Product

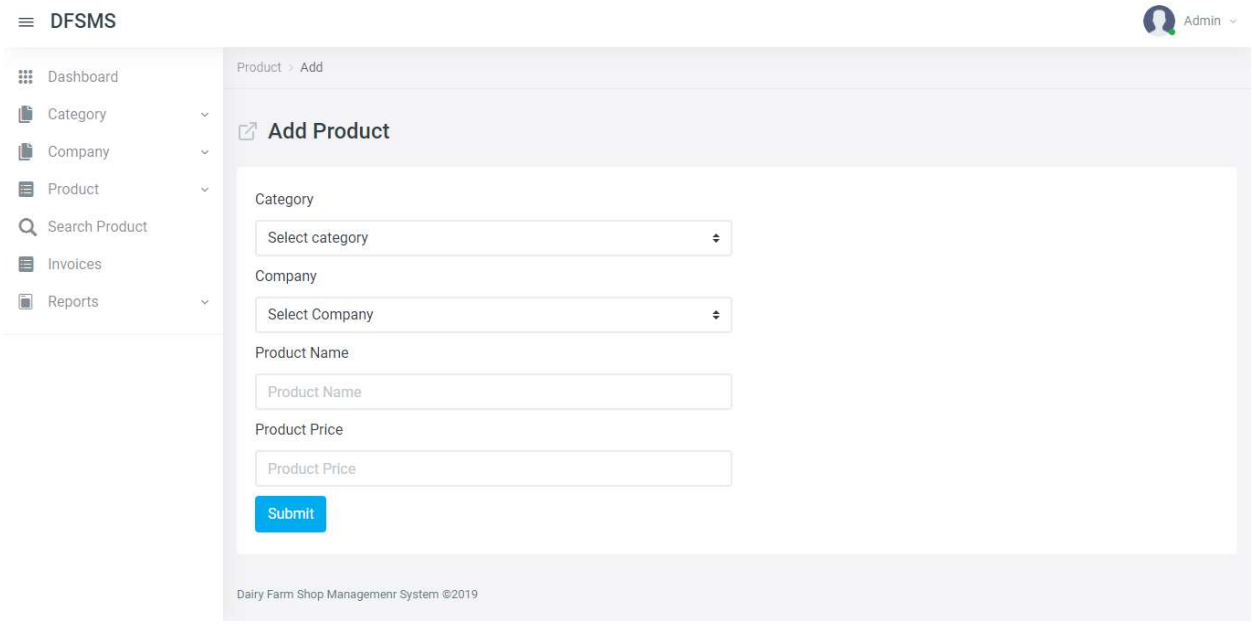


Fig.7.11: Add product page

Manage Product

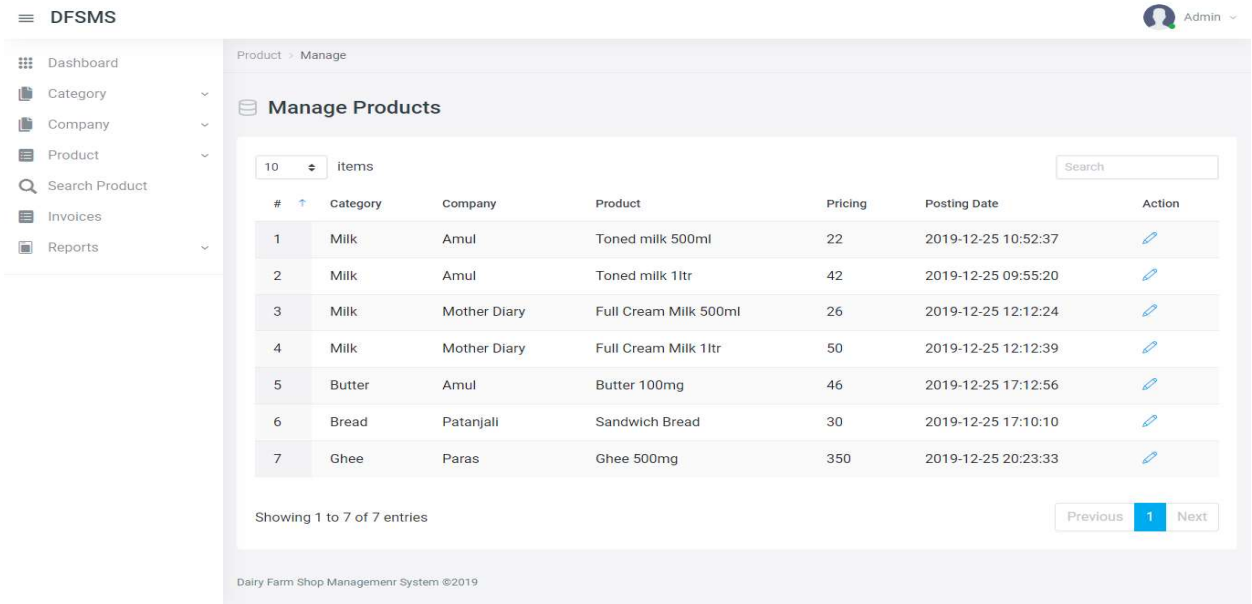


Fig.7.12: Manage product page

Update Product

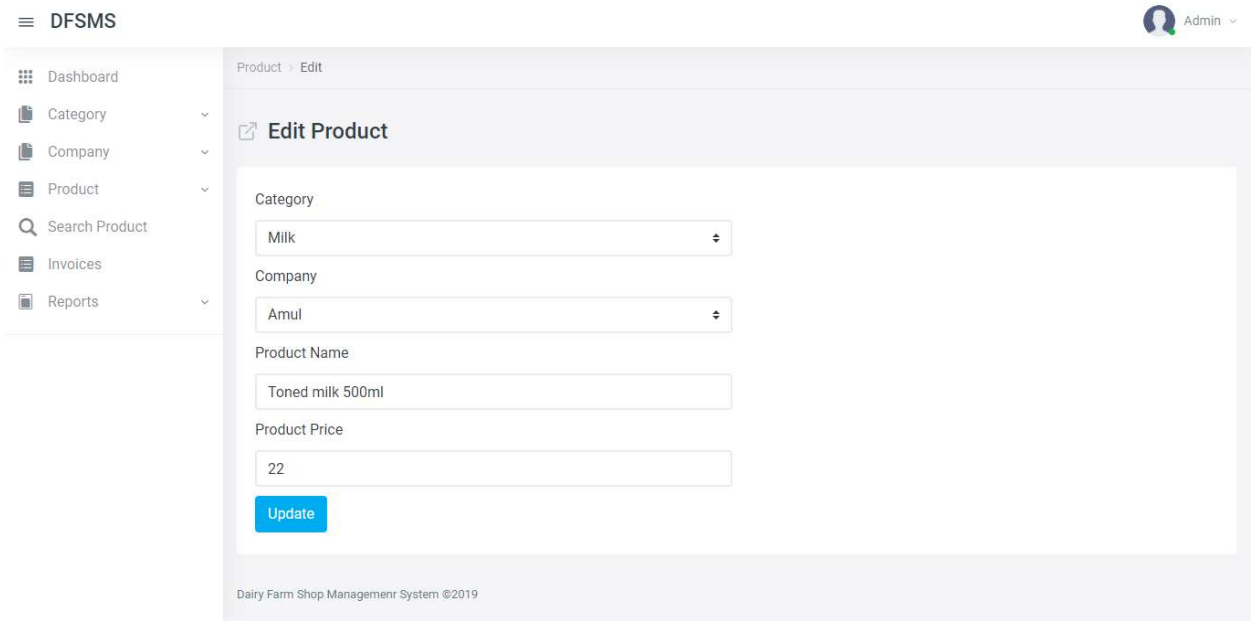


Fig.7.13: Update product page

Search Product

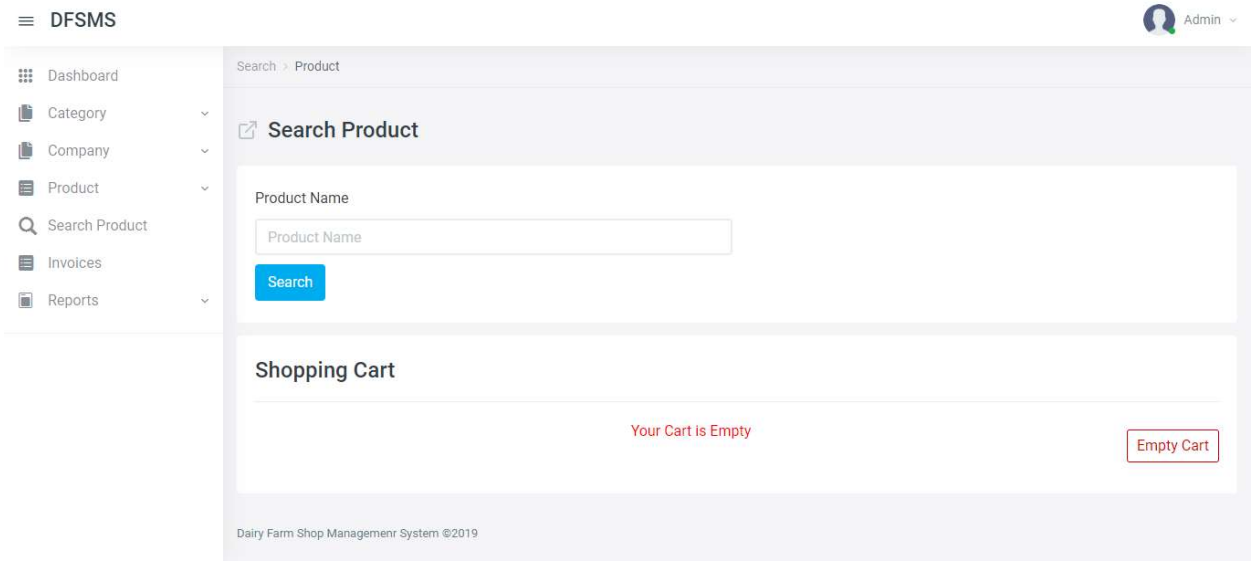


Fig.7.14: Search product page

Search Product with product

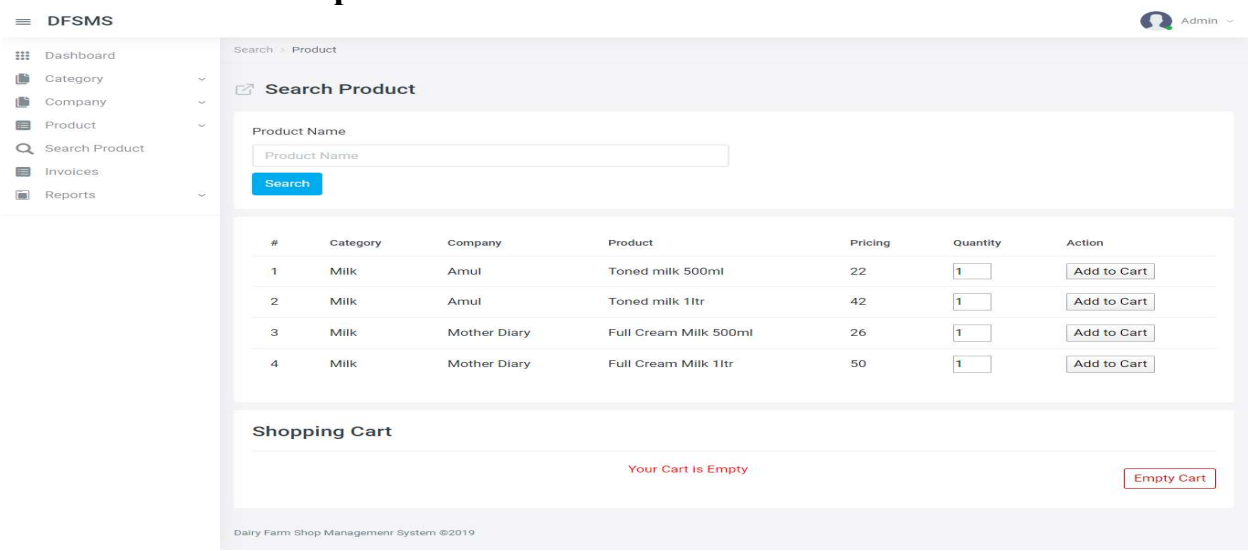


Fig.7.15: Search product with product page

Product Add to Cart

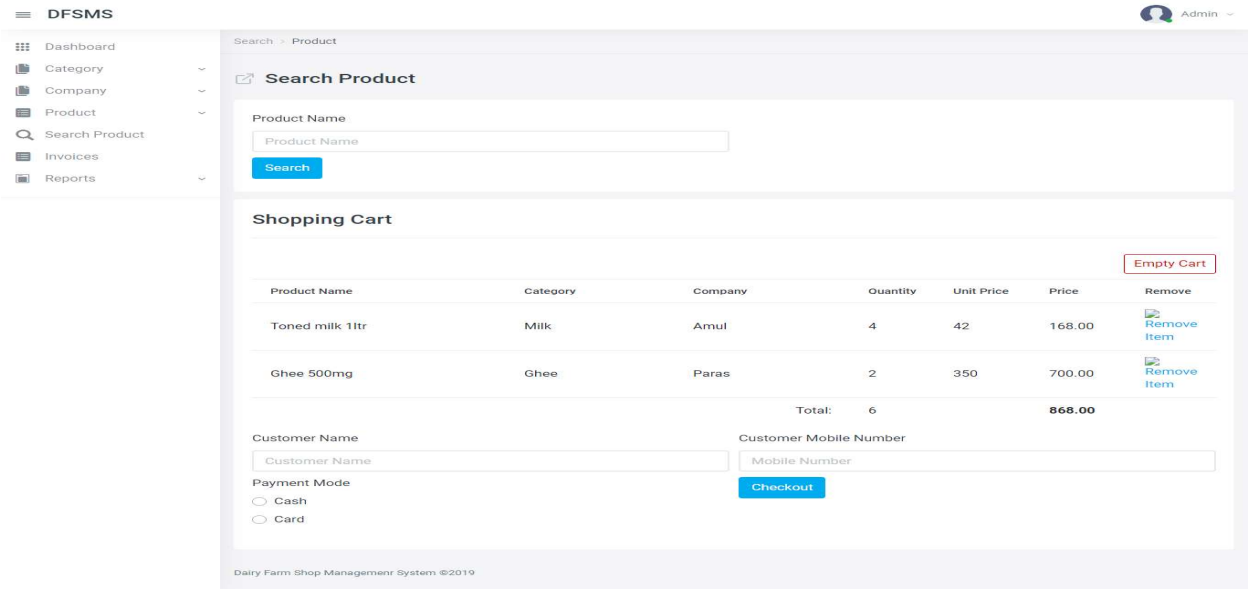


Fig.7.16: Product add to cart page

View Invoice

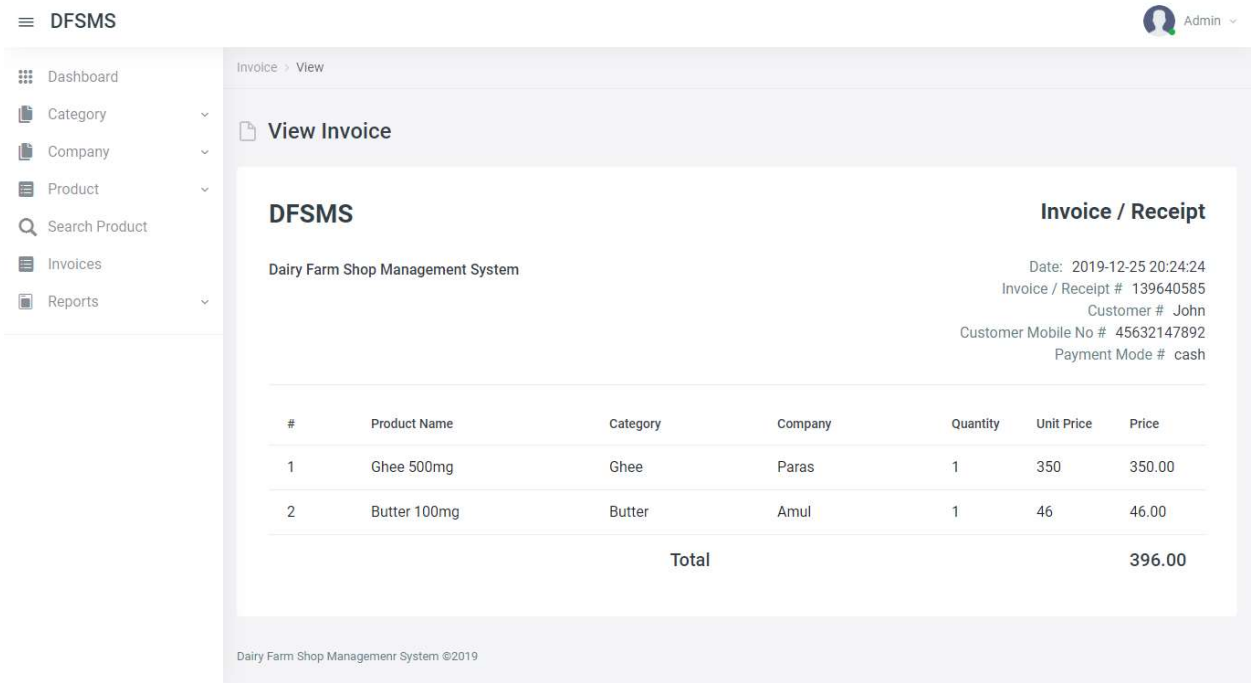


Fig.7.17: View invoice page

Between Date Report

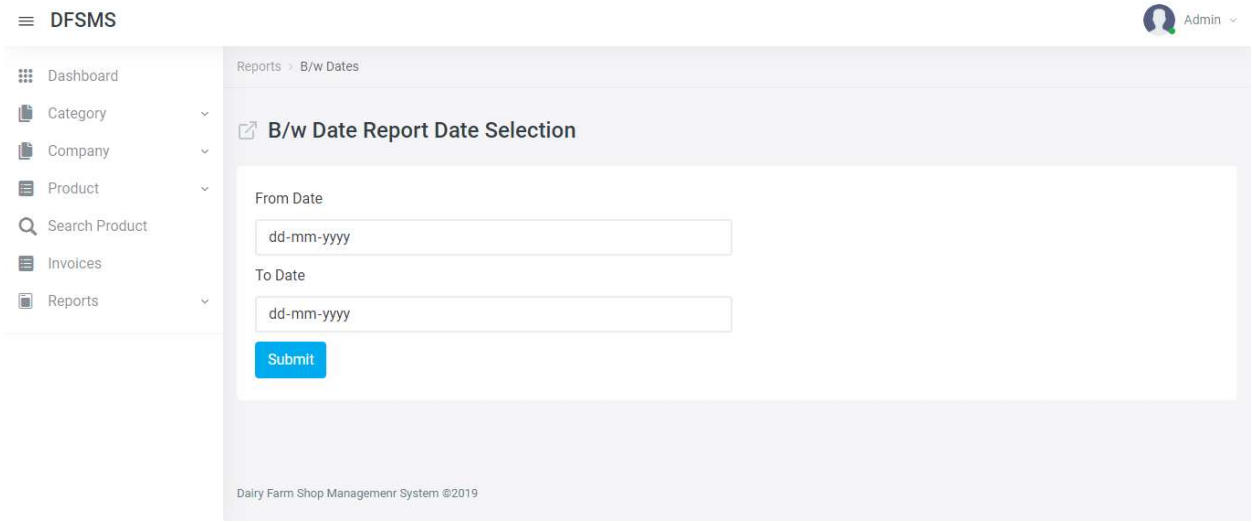


Fig.7.18: Report between page

Detail of Between Date Report

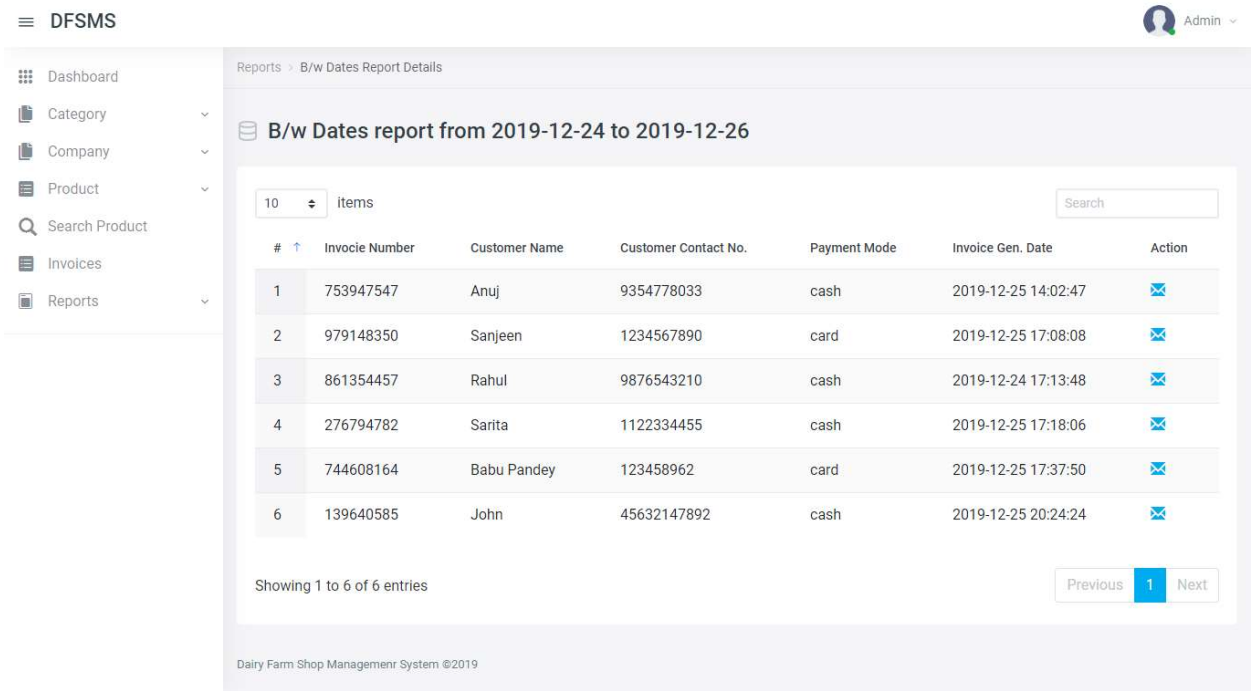


Fig.7.19: Detail of between date report page

Sales Report

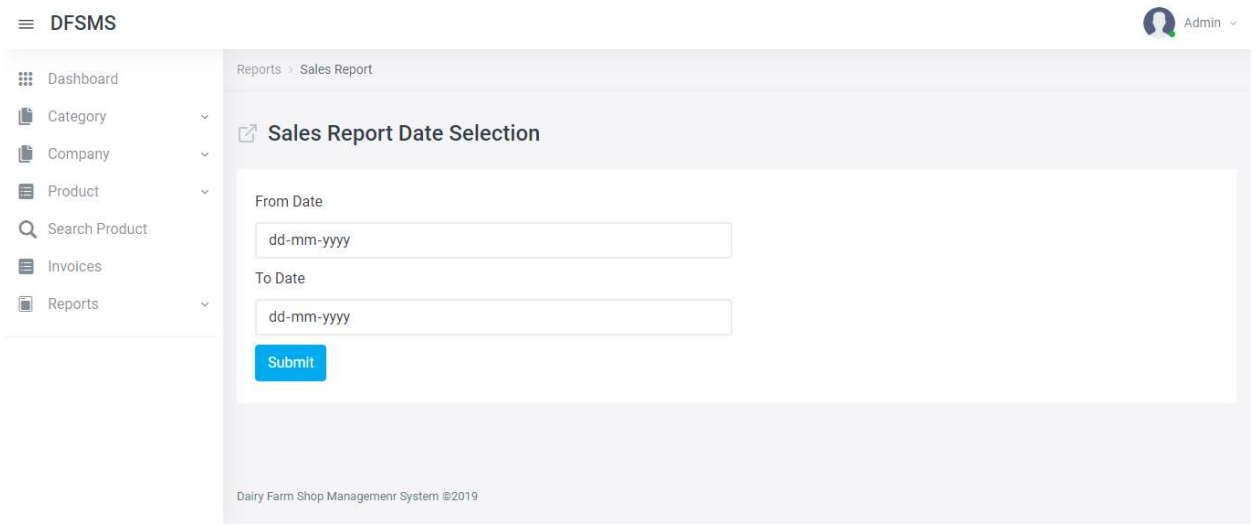


Fig.7.20: Sales report page

Detail of Sales Report

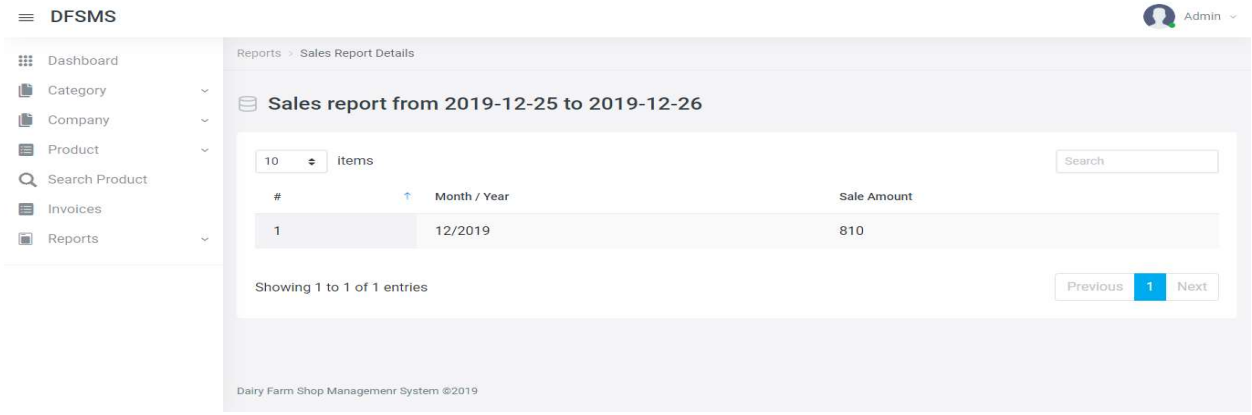


Fig.7.21: Detail of sales report page

Chapter 8

CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in PHP and MySQL web based application. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

BENEFITS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can say that the project is user friendly which is one of the primary concerns of any good project.

- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.
- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency,

LIMITATIONS:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

BIBLIOGRAPHY

For PHP

- https://www.w3schools.com/php/default.asp
- https://www.sitepoint.com/php/
- https://www.php.net/

For MySQL

- https://www.mysql.com/
- http://www.mysqltutorial.org

For XAMPP

https://www.apachefriends.org/download.html

APPENDIX

```
<?php
Session start();
//error_reporting (0);
include ('includes/config.php');
if (strlen($_SESSION['aid']==0)) {
  header('location:logout.php');
  } else{
// Add Category Code
If (isset ($_POST['submit']))
{
//Getting Post Values
$catname=$_POST['category'];
$catcode=$_POST['categorycode'];
$query=mysqli_query($con,"insert into tblcategory(CategoryName,CategoryCode)
values('$catname','$catcode')");
if($query){
echo "<script>alert('Category added successfully.');</script>";
echo "<script>window.location.href='add-category.php'</script>";
} else{
echo "<script>alert('Something went wrong. Please try again.');</script>";
echo "<script>window.location.href='add-category.php'</script>";
}
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
```

```
scale=1.0, user-scalable=no" />
   <title>Add Product</title>
   <link href="vendors/jquery-toggles/css/toggles.css" rel="stylesheet" type="text/css">
   <link href="vendors/jquery-toggles/css/themes/toggles-light.css" rel="stylesheet"
type="text/css">
   <link href="dist/css/style.css" rel="stylesheet" type="text/css">
</head>
<body>
   <!-- HK Wrapper -->
         <div class="hk-wrapper hk-vertical-nav">

<!-- Top Navbar -->
<?php include_once('includes/navbar.php');
include_once('includes/sidebar.php');
?>



      <div id="hk_nav_backdrop" class="hk-nav-backdrop"></div>
      <!-- /Vertical Nav -->



      <!-- Main Content -->
      <div class="hk-pg-wrapper">
         <!-- Breadcrumb -->
         <nav class="hk-breadcrumb" aria-label="breadcrumb">
            <ol class="breadcrumb breadcrumb-light bg-transparent">
<li class="breadcrumb-item"><a href="#">Product</a></li>
<li class="breadcrumb-item active" aria-current="page">Add</li>
            </ol>
```

```
        </nav>
        <!-- /Breadcrumb -->

        <!-- Container -->
        <div class="container">
            <!-- Title -->
            <div class="hk-pg-header">
                <h4 class="hk-pg-title"><span class="pg-title-icon"><span class="feather-icon"><i
data-feather="external-link"></i></span></span>Add Product</h4>
            </div>
            <!-- /Title -->

            <!-- Row -->
            <div class="row">
                <div class="col-xl-12">
<section class="hk-sec-wrapper">

<div class="row">
<div class="col-sm">
<form class="needs-validation" method="post" novalidate>

<div class="form-row">
<div class="col-md-6 mb-10">
<label for="validationCustom03">Category</label>
<input type="text" class="form-control" id="validationCustom03" placeholder="Category"
name="category" required>
<div class="invalid-feedback">Please provide a valid category name.</div>
</div>
</div>

<div class="form-row">
```

```
<div class="col-md-6 mb-10">
<label for="validationCustom03">Category Code</label>
<input type="text" class="form-control" id="validationCustom03" placeholder="Category Code"
name="categorycode" required>
<div class="invalid-feedback">Please provide a valid category code.</div>
</div>
</div>

<button class="btn btn-primary" type="submit" name="submit">Submit</button>
</form>
</div>
</div>
</section>

</div>
</div>
</div>



        <!-- Footer -->
<?php include_once('includes/footer.php');?>
        <!-- /Footer -->

      </div>
      <!-- /Main Content -->

   </div>

   <script src="vendors/jquery/dist/jquery.min.js"></script>
   <script src="vendors/popper.js/dist/umd/popper.min.js"></script>
   <script src="vendors/bootstrap/dist/js/bootstrap.min.js"></script>
```

```
    <script src="vendors/jasny-bootstrap/dist/js/jasny-bootstrap.min.js"></script>
    <script src="dist/js/jquery.slimscroll.js"></script>
    <script src="dist/js/dropdown-bootstrap-extended.js"></script>
    <script src="dist/js/feather.min.js"></script>
    <script src="vendors/jquery-toggles/toggles.min.js"></script>
    <script src="dist/js/toggle-data.js"></script>
    <script src="dist/js/init.js"></script>
    <script src="dist/js/validation-data.js"></script>

</body>
</html>
<?php } ?>

<?php
session_start();
//error_reporting(0);
include('includes/config.php');
if (strlen($_SESSION['aid']==0)) {
  header('location:logout.php');
  } else{
// Add Category Code
if(isset($_POST['update']))
{
$cid=substr(base64_decode($_GET['catid']),0,-5);
//Getting Post Values
$catname=$_POST['category'];
$catcode=$_POST['categorycode'];
$query=mysqli_query($con,"update tblcategory set
CategoryName='$catname',CategoryCode='$catcode' where id='$cid'");
echo "<script>alert('Category updated successfully.');</script>";
echo "<script>window.location.href='manage-categories.php'</script>";
```

```
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
scale=1.0, user-scalable=no" />
   <title>Edit Category</title>
   <link href="vendors/jquery-toggles/css/toggles.css" rel="stylesheet" type="text/css">
   <link href="vendors/jquery-toggles/css/themes/toggles-light.css" rel="stylesheet"
type="text/css">
   <link href="dist/css/style.css" rel="stylesheet" type="text/css">
</head>

<body>



        <!-- HK Wrapper -->
        <div class="hk-wrapper hk-vertical-nav">

<!-- Top Navbar -->
<?php include_once('includes/navbar.php');
include_once('includes/sidebar.php');
?>
   <div id="hk_nav_backdrop" class="hk-nav-backdrop"></div>
      <!-- /Vertical Nav -->
      <!-- Main Content -->
      <div class="hk-pg-wrapper">
         <!-- Breadcrumb -->
```

```
        <nav class="hk-breadcrumb" aria-label="breadcrumb">
            <ol class="breadcrumb breadcrumb-light bg-transparent">
<li class="breadcrumb-item"><a href="#">Category</a></li>
<li class="breadcrumb-item active" aria-current="page">Edit</li>
            </ol>
        </nav>
        <!-- /Breadcrumb -->

        <!-- Container -->
        <div class="container">
            <!-- Title -->
            <div class="hk-pg-header">
                <h4 class="hk-pg-title"><span class="pg-title-icon"><span class="feather-icon"><i
data-feather="external-link"></i></span></span>Edit Category</h4>
            </div>
            <!-- /Title -->

            <!-- Row -->
            <div class="row">
                <div class="col-xl-12">
<section class="hk-sec-wrapper">

<div class="row">
<div class="col-sm">
<form class="needs-validation" method="post" novalidate>
<?php
$cid=substr(base64_decode($_GET['catid']),0,-5);
$ret=mysqli_query($con,"select * from tblcategory where ID='$cid'");
$cnt=1;
while ($row=mysqli_fetch_array($ret)) {
?>
```

```
<div class="form-row">
<div class="col-md-6 mb-10">
<label for="validationCustom03">Category</label>
<input type="text" class="form-control" id="validationCustom03" value="<?php echo
$row['CategoryName'];?>" name="category" required>
<div class="invalid-feedback">Please provide a valid category name.</div>
</div>
</div>

<div class="form-row">
<div class="col-md-6 mb-10">
<label for="validationCustom03">Category Code</label>
<input type="text" class="form-control" id="validationCustom03" value="<?php echo
$row['CategoryCode'];?>" name="categorycode" required>
<div class="invalid-feedback">Please provide a valid category code.</div>
</div>
</div>
<?php } ?>
<button class="btn btn-primary" type="submit" name="update">Update</button>
</form>
</div>
</div>
</section>
</div>
</div>
</div>
 <!-- Footer -->
<?php include_once('includes/footer.php');?>
        <!-- /Footer -->
</div>
      <!-- /Main Content -->
```

```
    </div>
    <script src="vendors/jquery/dist/jquery.min.js"></script>
    <script src="vendors/popper.js/dist/umd/popper.min.js"></script>
    <script src="vendors/bootstrap/dist/js/bootstrap.min.js"></script>
    <script src="vendors/jasny-bootstrap/dist/js/jasny-bootstrap.min.js"></script>
    <script src="dist/js/jquery.slimscroll.js"></script>
    <script src="dist/js/dropdown-bootstrap-extended.js"></script>
    <script src="dist/js/feather.min.js"></script>
    <script src="vendors/jquery-toggles/toggles.min.js"></script>
    <script src="dist/js/toggle-data.js"></script>
    <script src="dist/js/init.js"></script>
    <script src="dist/js/validation-data.js"></script>

</body>
</html>
<?php } ?>
```