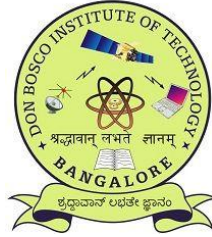




**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANASANGAMA, BELAGAVI-590014**



**MINI PROJECT REPORT**

**ON**

**“CLOTHING WEBSITE MANAGEMENT SYSTEM”**

Submitted in partial fulfillment for the requirement of 5<sup>th</sup> semester for the

**Degree of Bachelor of Engineering in  
COMPUTER SCIENCE & ENGINEERING**

For the Academic Year 2021-22

**SUBMITTED BY:**

**NEHA R RAO [1DB19CS095]**

**RUCHITHA M [1DB19CS165]**

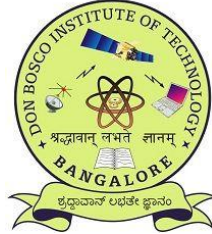
Under the guidance of:

**Mrs. SUMATHI M**

Assistant professor, Dept. of CSE, DBIT

**BENGALURU - 560074**

**DON BOSCO INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**Kumbalagodu, Bengaluru-560074**



**CERTIFICATE**

This is to certify that the Project Report entitled “CLOTHING WEBSITE MANAGEMENT SYSTEM” is a bonafide Project work carried out by NEHA R RAO (1DB19CS095) and RUCHITHA M (1DB19CS165) in partial fulfillment of 5<sup>th</sup> semester for the Degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi, during the academic year 2021-22. It is certified that all corrections/suggestions indicated for Internal Assessments have been in corporate with the degree mentioned.

Signature of Guide

\_\_\_\_\_

**Mrs. SUMATHI M**

Assistant Professor,  
Dept. of CSE,  
DBIT, Bengaluru.

Signature of HOD

\_\_\_\_\_

**Prof. B S UMASHANKAR**

Professor & Head,  
Dept. of CSE,  
DBIT, Bengaluru.

**External Viva**

**Name of the Examiners**

1. \_\_\_\_\_

2. \_\_\_\_\_

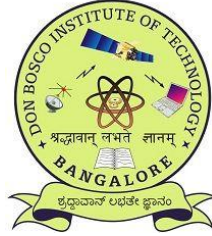
**Signature with Date**

\_\_\_\_\_

\_\_\_\_\_

# DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Bengaluru-560074



## DECLARATION

We, NEHA R RAO and RUCHITHA M, students of fifth semester B.E, Department of Computer Science and Engineering, Don Bosco Institute of Technology, Kumbalagodu, Bengaluru, declare, that the Project Work entitled “CLOTHING WEBSITE MANAGEMENT SYSTEM” has been carried out by and submitted in partial fulfillment of the requirement of 5<sup>th</sup> semester Oct 2021 - Mar 2022. The matter embodied in this report has been submitted to any university or institute for the award of any other degree or diploma.

Place: Bengaluru

Date:

NEHA R RAO [1DB19CS095]

RUCHITHA M [1DB19CS165]

## **ACKNOWLEDGEMENT**

At the various stages in making the mini project, a number of people have given me in valuable comment on the manuscript. I take this opportunity to express my deepest gratitude and appreciation to all those who helped me directly or indirectly towards the successful completion of this project.

I would like to thank our Principal Dr. HEMADRI NAIDU T, Don Bosco Institute of Technology for his support throughout this project.

I express my wholehearted gratitude to Prof. B S UMASHANKAR, who is our respectable Head of Dept. Of Computer Science. I wish to acknowledge for his valuable help and encouragement.

In this regard I owe a heartfelt gratitude to my guide, Mrs. SUMATHI M, Assistant Professor in Dept. of CSE, DBIT, Bengaluru, for her timely advice on the project and regular assistance throughout the project work. I would also like to thank the teaching and non-teaching staff members of Department of Computer Science and Engineering for their co-operation.

## **ABSTRACT**

E-commerce (electronic commerce or EC) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the internet. In emerging global economy, ecommerce has increasingly become a vital component of business strategy and a solid catalyst for economic development. The primary goal of an ecommerce website is to sell goods. E-commerce shops have become part of our daily lives. Technological advancement has made it possible for people to sit in the convenience of their homes and still shop online without going to a physical shop.

This designed application/project is mainly divided into two main categories: The Administrators and the Customers/Users. The Admin view is meant for the administrator to update the products, change prices, remove, and add products and manage customers. The Customer view will be accessible to the customers, and they will be able to handle their information such as their name, address, and contact. Also, the customer will be able to order for products from the shop. The User is only able to browse the online shop and add a product to the cart. The user is limited to use of the shop until he logs in and becomes a customer.

## TABLE OF CONTENTS

Sl. No.	Topic	Page No.
1	Introduction	1
	1.1 Overview	1
	1.2 Background and Motivation	1
	1.3 Methodology	1
2	Scope of Project	2
3	Requirements	3
4	Data Directory	4
	4.1 Detailed Schema Diagram	5
	4.2 E-R Diagram	6
5	Relational Database Design	7-8
	5.1 Database Normalization	9
6	Graphical User Interface	10
7	Source Code	11-27
	7.1 Snapshots	28-31
8	Conclusion	32
9	References	33

## CHAPTER 1

### INTRODUCTION

An eCommerce website is an information technology method in which traders, businesses/distributors/marketers can sell products/services and the customer can purchase on that website electronically by using the internet on the mobile and computer.

#### 1.1 Overview

This report discusses the result of the work done in development of "Ecommerce database" on "HTML, CSS, JAVASCRIPT and BOOTSTRAP" as Frontend Platform and "MySQL and PHP" as Backend Platform.

At the development of an application –

HTML language is used in creating web pages. CSS is used to define styles for web pages, including the design, layout, and variations in the display for different devices and screen sizes. JAVASCRIPT is a high-level language which could be used independently or inculcated into the webpage. BOOTSTRAP is a collaboration of CSS. MySQL is a free source database system, and it enables the cost-effective delivery of reliable and a high-performance and scalable Web-based and embedded database applications. PHP is a server-side scripting language that is used to develop Static websites or Dynamic websites or Web application.

#### 1.2 Background and Motivation

E-commerce (electronic commerce or EC) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the internet. This website was created to provide great convenience to the shoppers and the retailers. It does not have any time and place limitations, it works around –the –clock so consumers can do purchase at anytime and anywhere.

#### 1.3 Methodology

To implement the above goals, we need to specify the Application and various components of the Architecture, specify the bindings between the tasks and the resources either manually or by the design tools, specify the port interconnections between the resources.

## **CHAPTER 2**

### **SCOPE OF PROJECT**

The scope of the project is clear to give a simple and attractive application to simplify the work as well as to reduce the efforts while doing it offline or we can say by doing it with old methods. In this application we are able to save database of all products present on the site. Using this website, we can buy any product very easily.



## CHAPTER 3

### REQUIRMENTS

#### 3.1 Software requirements

The software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functional of an application.

##### 3.1.1 Front End

User interface: HTML, CSS, JAVASCRIPT and BOOTSTRAP

Operating system: Windows 10

Web browser: Chrome, Internet Explorer

##### 3.1.2 Back End

Database: MySQL

Application server: XAMPP SERVER(v3.3.2.2) for Apache server (localhost)

PHPStorm(v2021.3.2) for server - side scripting

#### 3.2 Hardware Requirement:

The software should run on any sort of desktop or laptop environment, regardless of the operating system. Essential input/output devices are keyboards, mouse, and printer.

1. Processor – Intel core i5
2. RAM – 4GB
3. Storage – 256GB

## CHAPTER 4

### DATA DIRECTORY

**Table name: Users**

INSERT INTO USERS VALUES

(1,'admin@admin.com','Admin','Banglore,India',9876453221,'05-Jan-22');

INSERT INTO USERS VALUES

(2,'ruchitha@gmail.com','Ruchitha','M','Banglore,India', 9638920121,'20-Jan-22');

INSERT INTO USERS VALUES

(3,'neharao@gmail.com','Neha','R','Banglore,India',9875362718,'20-Jan-22');

INSERT INTO USERS VALUES

(4,'happysoul@gmail.com','happysoul','Banglore,India',9346782901,'24-Jan-22');

**Table name: Category**

INSERT INTO CATEGORY VALUES (1, 'Women's Wear', 'womens')

INSERT INTO CATEGORY VALUES (2, 'Men's Wear', 'mens')

INSERT INTO CATEGORY VALUES (3, 'Kids Wear', 'kids')

**Table name: Cart**

INSERT INTO CART VALUES (1, 1, 19, 2)

INSERT INTO CART VALUES (2, 3, 10, 1)

INSERT INTO CART VALUES (3, 2, 5, 3)

INSERT INTO CART VALUES (4, 1, 23, 1)

## 4.1 Detailed Schema Diagram

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

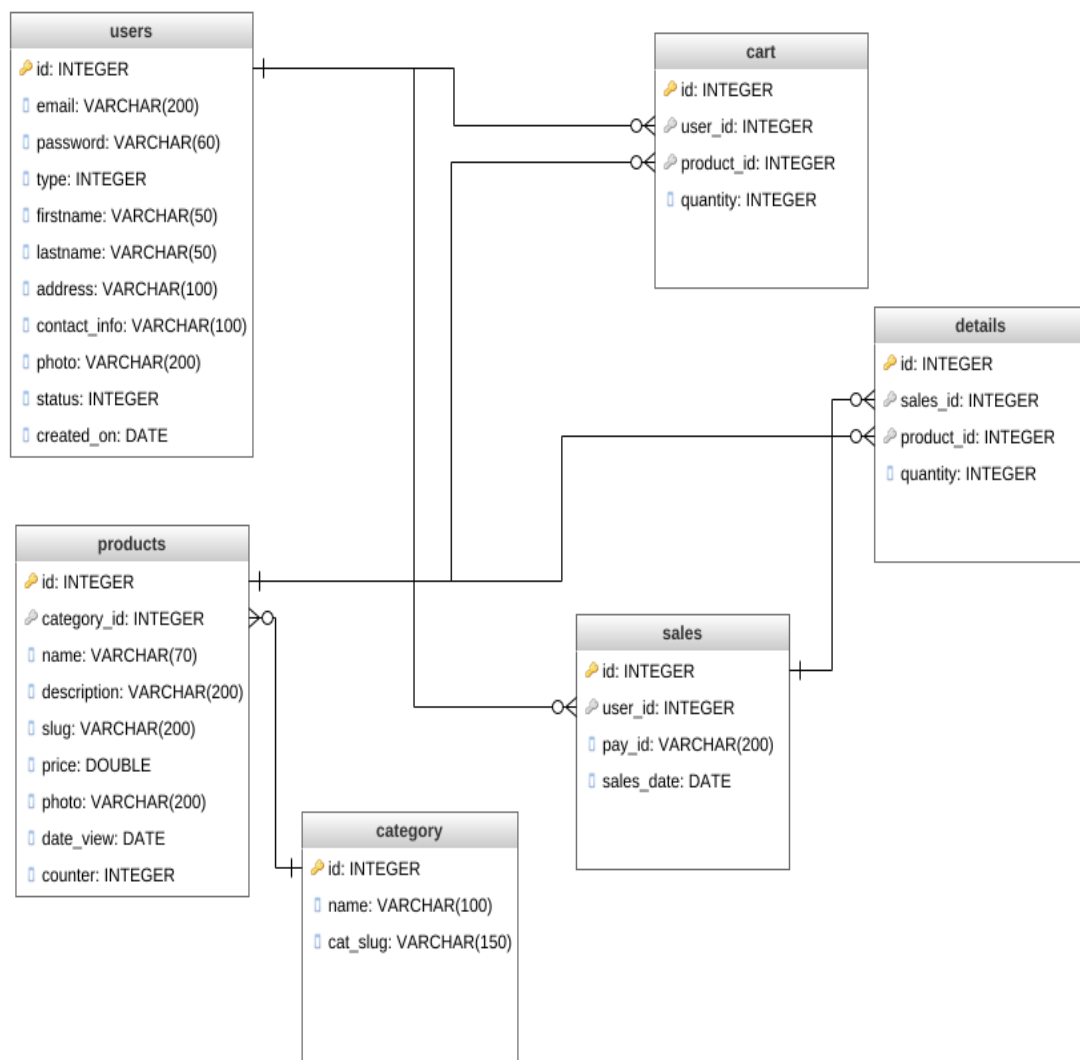


Fig. 4.1 Detailed Schema Diagram

## 4.2 E-R Diagram

The Entity-Relationship diagram or ER model describes the inter-related things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies the relationship that can exist between entities.

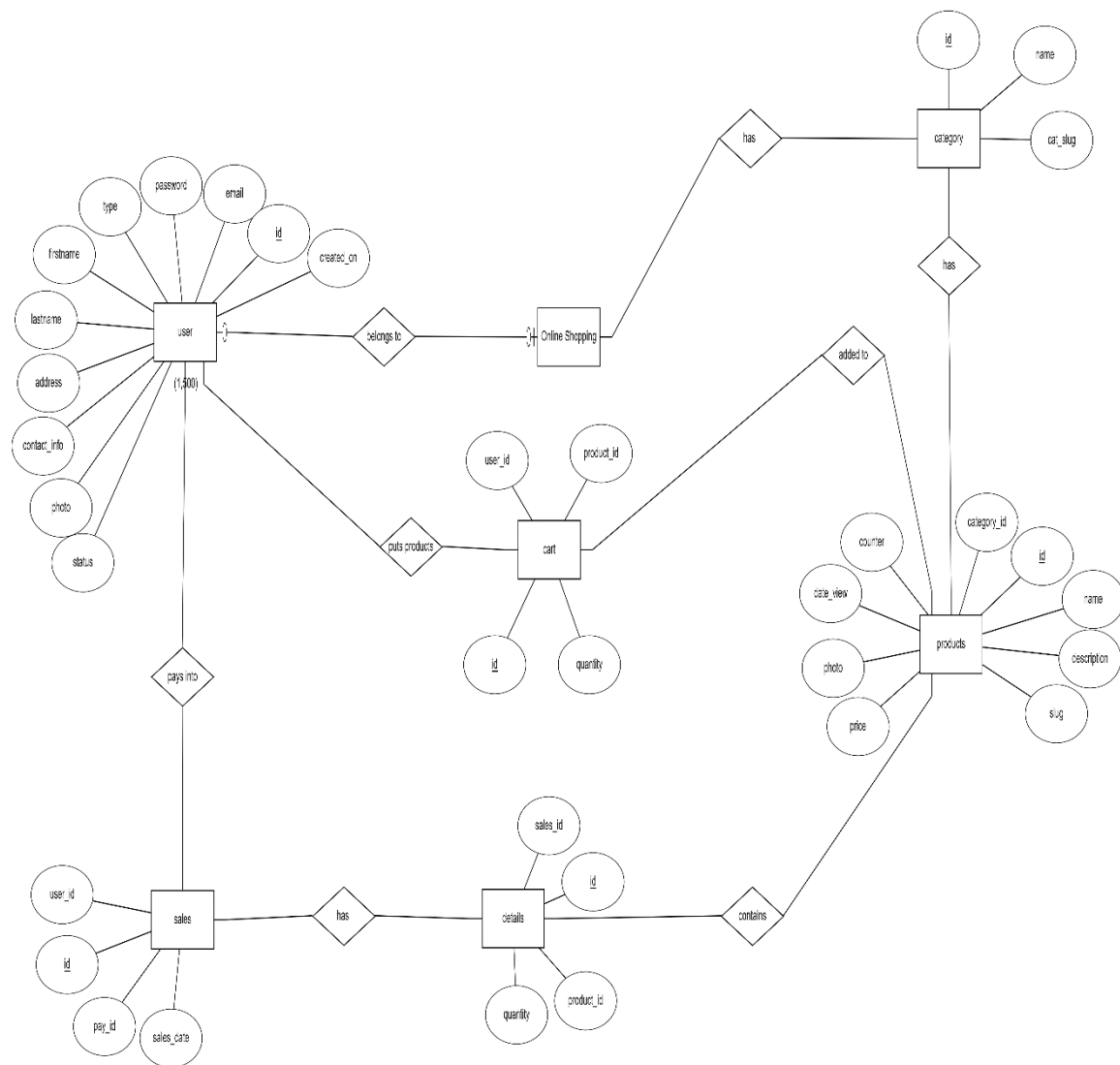


Fig. 4.2 E-R Diagram

**CHAPTER 5****RELATIONAL DATA BASE DESIGN****1. USERS:**

SL.NO	FIELD NAME	DATA TYPE	CONSTRAINTS
1.	Id	int(11)	Primary key
2.	Email	varchar(200)	Not null
3.	Password	varchar(60)	Not null
4.	Type	int(1)	Not null
5.	First name	varchar(50)	Not null
6.	Last name	varchar(50)	Not null
7.	Address	varchar(100)	Not null
8.	Contact_info	varchar(100)	Not null
9.	Photo	varchar(200)	Not null
10.	Status	int(1)	Not null
11.	Created_on	date	Not null

**2.PRODUCTS:**

SL NO.	FIELD NAME	DATA TYPE	CONSTRAINTS
1.	Id	int(11)	Primary key
2.	Category id	int(11)	Not null
3.	Name	varchar(70)	Not null
4.	Description	varchar(200)	Not null
5.	Slug	varchar(200)	Not null
6.	Price	double	Not null
7.	Photo	varchar(200)	Not null
8.	Data_view	date	Not null
9.	Counter	int(11)	Not null

**3.CART:**

SL NO.	FIELD NAME	DATA TYPE	CONSTRAINTS
1.	Id	int(11)	Primary key
2.	User id	int(11)	Not null
3.	Product id	int(11)	Not null
4.	Quantity	int(11)	Not null

**4.CATEGORY:**

SL NO.	FIELD NAME	DATA TYPE	CONSTRAINTS
1.	Id	int(11)	Primary key
2.	Name	varchar(100)	Not null
3.	Cat_slug	varchar(150)	Not null

## 5.DETAILS:

SL NO.	FIELD NAME	DATA TYPE	CONSTRAINTS
1.	Id	int(11)	Primary key
2.	Sales id	int(11)	Not null
3.	Product id	int(11)	Not null
4.	quantity	int(11)	Not null

## 6.SALES:

SL NO.	FIELD NAME	DATA TYPE	CONSTRAINTS
1.	Id	int(11)	Primary key
2.	User id	int(11)	Not null
3.	Pay id	varchar(50)	Not null
4.	Sales date	Date	Not null

## 5.1 Database Normalization:

First Normal Form: -

The relation is in 1NF if it has no repeating groups. All tables have no repeating groups, so they are in 1NF.

Tables: -

1] Users: -

i d	email	pass word	type	firstnam e	lastnam e	addres s	contac t_info	photo	statu s	created_ on
--------	-------	--------------	------	---------------	--------------	-------------	------------------	-------	------------	----------------

2] Products: -

id	category_id	name	description	slug	price	photo	date_view	counter
----	-------------	------	-------------	------	-------	-------	-----------	---------

3] Cart: -

id	user_id	product_id	quantity
----	---------	------------	----------

4] Category: -

id	name	cat_slug
----	------	----------

5] Details: -

id	sales_id	product_id	quantity
----	----------	------------	----------

6] Sales: -

id	user_id	pay_id	sales_date
----	---------	--------	------------

## CHAPTER 6

### GRAPHICAL USER INTERFACE

The application is very user friendly and uses a GUI interface implemented in PHP and HTML to Communicate with the user. Various features are self – explanatory. Forms are easy to fill in and components can be added, removed, and updated very easily through a single dialog box. The application includes tool-tip hints to give a brief description of the particular input field.

List boxes are used to display all the components at once so that user can see all the components of a particular type at once. One can just select the component and modify and remove the component. (based on the access control of the person)

Features:

Clean separation of various components to facilitate easy modification and revision. All the data is maintained in a separate file to facilitate easy modification. All the data required for different operations is kept in a separate file. Quick and easy saving and loading of database file.



## CHAPTER 7

### SOURCE CODE

#### 1] Admin view- Add products: -

```
<?php
    include 'includes/session.php';
    include 'includes/slugify.php';
    if(isset($_POST['add'])){
        $name = $_POST['name'];
        $slug = slugify($name);
        $category = $_POST['category'];
        $price = $_POST['price'];
        $size = $_POST['size'];
        $description = $_POST['description'];
        $filename = $_FILES['photo']['name'];

        $conn = $pdo->open();

        $stmt = $conn->prepare("SELECT *, COUNT(*) AS numrows FROM
products WHERE slug=:slug");
        $stmt->execute(['slug'=>$slug]);
        $row = $stmt->fetch();

        if($row['numrows'] > 0){
            $_SESSION['error'] = 'Product already exist';
        }
        else{
            if(!empty($filename)){
                $ext = pathinfo($filename, PATHINFO_EXTENSION);
                $new_filename = $slug.'.'.$ext;
                move_uploaded_file($_FILES['photo']['tmp_name'],
'../images/'.$new_filename);
            }
            else{
                $new_filename = '';
            }
            try{
                $stmt = $conn->prepare("INSERT INTO products (category_id,
name, description, slug, price, photo) VALUES (:category, :name,
:description, :slug, :price, :photo)");
                $stmt->execute(['category'=>$category, 'name'=>$name,
'description'=>$description, 'slug'=>$slug, 'price'=>$price,
'photo'=>$new_filename]);
                $_SESSION['success'] = 'Product added successfully';
            }
            catch(PDOException $e){
                $_SESSION['error'] = $e->getMessage();
            }
        }
        $pdo->close();
    }
    else{
        $_SESSION['error'] = 'Fill up product form first';
    }
}
```

```
header('location: products.php');
?>
```

## 2] Admin view – Delete products: -

```
<?php
include 'includes/session.php';

if(isset($_POST['delete'])){
    $id = $_POST['id'];

    $conn = $pdo->open();

    try{
        $stmt = $conn->prepare("DELETE FROM products WHERE id=:id");
        $stmt->execute(['id'=>$id]);

        $_SESSION['success'] = 'Product deleted successfully';
    }
    catch(PDOException $e){
        $_SESSION['error'] = $e->getMessage();
    }
    $pdo->close();
}
else{
    $_SESSION['error'] = 'Select product to delete first';
}
header('location: products.php');

?>
```

## 3] Admin view- Edit/Update Products: -

```
<?php
include 'includes/session.php';
include 'includes/slugify.php';

if(isset($_POST['edit'])){
    $id = $_POST['id'];
    $name = $_POST['name'];
    $slug = slugify($name);
    $category = $_POST['category'];
    $price = $_POST['price'];
    $description = $_POST['description'];

    $conn = $pdo->open();

    try{
        $stmt = $conn->prepare("UPDATE products SET name=:name,
slug=:slug, category_id=:category, price=:price,
description=:description WHERE id=:id");
        $stmt->execute(['name'=>$name, 'slug'=>$slug,
'category'=>$category, 'price'=>$price, 'description'=>$description,
'id'=>$id]);
        $_SESSION['success'] = 'Product updated successfully';
    }
    catch(PDOException $e){
        $_SESSION['error'] = $e->getMessage();
    }
}
```

```
$pdo->close();
}
else{
    $_SESSION['error'] = 'Fill up edit product form first';
}
header('location: products.php');
?>
```

#### 4] Admin view – Add category: -

```
<?php
include 'includes/session.php';
if(isset($_POST['add'])){
    $name = $_POST['name'];
    $conn = $pdo->open();
    $stmt = $conn->prepare("SELECT *, COUNT(*) AS numrows FROM
category WHERE name=:name");
    $stmt->execute(['name'=>$name]);
    $row = $stmt->fetch();
    if($row['numrows'] > 0){
        $_SESSION['error'] = 'Category already exist';
    }
    else{
        try{
            $stmt = $conn->prepare("INSERT INTO category (name) VALUES
(:name)");
            $stmt->execute(['name'=>$name]);
            $_SESSION['success'] = 'Category added successfully';
        }
        catch(PDOException $e){
            $_SESSION['error'] = $e->getMessage();
        }
    }
    $pdo->close();
}
else{
    $_SESSION['error'] = 'Fill up category form first';
}
header('location: category.php');
?>
```

#### 5] Admin view – Delete Category: -

```
<?php
include 'includes/session.php';
if(isset($_POST['delete'])){
    $id = $_POST['id'];

    $conn = $pdo->open();
    try{
        $stmt = $conn->prepare("DELETE FROM category WHERE id=:id");
        $stmt->execute(['id'=>$id]);
        $_SESSION['success'] = 'Category deleted successfully';
    }
    catch(PDOException $e){
        $_SESSION['error'] = $e->getMessage();
    }
}
```

```
$pdo->close();
}
else{
    $_SESSION['error'] = 'Select category to delete first';
}

header('location: category.php');

?>
```

## 6] Admin view- Edit/Update Category: -

```
<?php
include 'includes/session.php';
if(isset($_POST['edit'])){
    $id = $_POST['id'];
    $name = $_POST['name'];
    try{
        $stmt = $conn->prepare("UPDATE category SET name=:name WHERE
id=:id");
        $stmt->execute(['name'=>$name, 'id'=>$id]);
        $_SESSION['success'] = 'Category updated successfully';
    }
    catch(PDOException $e){
        $_SESSION['error'] = $e->getMessage();
    }

    $pdo->close();
}
else{
    $_SESSION['error'] = 'Fill up edit category form first';
}
header('location: category.php');

?>
```

## 7] Admin view- Add Users: -

```
<?php
include 'includes/session.php';
if(isset($_POST['add'])){
    $firstname = $_POST['firstname'];
    $lastname = $_POST['lastname'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $address = $_POST['address'];
    $contact = $_POST['contact'];
    $conn = $pdo->open();
    $stmt = $conn->prepare("SELECT *, COUNT(*) AS numrows FROM users
WHERE email=:email");
    $stmt->execute(['email'=>$email]);
    $row = $stmt->fetch();
    if($row['numrows'] > 0){
        $_SESSION['error'] = 'Email already taken';
    }
    else{
        $password = password_hash($password, PASSWORD_DEFAULT);
```

```
$filename = $_FILES['photo']['name'];
$now = date('Y-m-d');
if(!empty($filename)){
    move_uploaded_file($_FILES['photo']['tmp_name'],
'../images/'.$filename);
}
try{
    $stmt = $conn->prepare("INSERT INTO users (email, password,
firstname, lastname, address, contact_info, photo, status, created_on)
VALUES (:email, :password, :firstname, :lastname, :address, :contact,
:photo, :status, :created_on)");
    $stmt->execute(['email'=>$email, 'password'=>$password,
'firstname'=>$firstname, 'lastname'=>$lastname, 'address'=>$address,
'contact'=>$contact, 'photo'=>$filename, 'status'=>1,
'created_on'=>$now]);
    $_SESSION['success'] = 'User added successfully';

}
catch(PDOException $e){
    $_SESSION['error'] = $e->getMessage();
}
}
$pdo->close();
}
else{
    $_SESSION['error'] = 'Fill up user form first';
}
header('location: users.php');
?>
```

### 8] Admin view – Delete Users: -

```
<?php
include 'includes/session.php';

if(isset($_POST['delete'])){
    $id = $_POST['id'];

    $conn = $pdo->open();

    try{
        $stmt = $conn->prepare("DELETE FROM users WHERE id=:id");
        $stmt->execute(['id'=>$id]);

        $_SESSION['success'] = 'User deleted successfully';
    }
    catch(PDOException $e){
        $_SESSION['error'] = $e->getMessage();
    }

    $pdo->close();
}
else{
    $_SESSION['error'] = 'Select user to delete first';
}

header('location: users.php');
?>
```

**9] Admin view – Edit/Update Users: -**

```
<?php
    include 'includes/session.php';

    if(isset($_POST['edit'])){
        $id = $_POST['id'];
        $firstname = $_POST['firstname'];
        $lastname = $_POST['lastname'];
        $email = $_POST['email'];
        $password = $_POST['password'];
        $address = $_POST['address'];
        $contact = $_POST['contact'];

        $conn = $pdo->open();
        $stmt = $conn->prepare("SELECT * FROM users WHERE id=:id");
        $stmt->execute(['id'=>$id]);
        $row = $stmt->fetch();

        if($password == $row['password']){
            $password = $row['password'];
        }
        else{
            $password = password_hash($password, PASSWORD_DEFAULT);
        }

        try{
            $stmt = $conn->prepare("UPDATE users SET email=:email,
password=:password, firstname=:firstname, lastname=:lastname,
address=:address, contact_info=:contact WHERE id=:id");
            $stmt->execute(['email'=>$email, 'password'=>$password,
'firstname'=>$firstname, 'lastname'=>$lastname, 'address'=>$address,
'contact'=>$contact, 'id'=>$id]);
            $_SESSION['success'] = 'User updated successfully';
        }
        catch(PDOException $e){
            $_SESSION['error'] = $e->getMessage();
        }

        $pdo->close();
    }
    else{
        $_SESSION['error'] = 'Fill up edit user form first';
    }

    header('location: users.php');

?>
```

**10] Customer/User view – Homepage: -**

```
<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue layout-top-nav">
<div class="wrapper">
```

```

<?php include 'includes/navbar.php'; ?>

<div class="content-wrapper">
  <div class="container">

    <!-- Main content -->
    <section class="content">
      <div class="row">
        <div class="col-sm-9">
          <?php
            if(isset($_SESSION['error'])){
              echo "
                <div class='alert alert-danger'>
                  ".$_SESSION['error'].
                </div>
              ";
              unset($_SESSION['error']);
            }
          ?>
          <div id="carousel-example-generic" class="carousel
slide" data-ride="carousel">
            <ol class="carousel-indicators">
              <li data-target="#carousel-example-generic"
data-slide-to="0" class="active"></li>
              <li data-target="#carousel-example-generic"
data-slide-to="1" class=""></li>
              <li data-target="#carousel-example-generic"
data-slide-to="2" class=""></li>
            </ol>
            <div class="carousel-inner">
              <div class="item active">
                
              </div>
              <div class="item">
                
              </div>
              <div class="item">
                
              </div>
            </div>
            <a class="left carousel-control" href="#carousel-
example-generic" data-slide="prev">
              <span class="fa fa-angle-left"></span>
            </a>
            <a class="right carousel-control" href="#carousel-
example-generic" data-slide="next">
              <span class="fa fa-angle-right"></span>
            </a>
          </div>
        </div>
        <div class="col-sm-3">
          <?php include 'includes/sidebar.php'; ?>
        </div>
      </div>
    </div>
  </div>

```

```
</section>

</div>
</div>

<?php include 'includes/footer.php'; ?>
</div>

<?php include 'includes/scripts.php'; ?>
</body>
</html>
```

## 11] Customer/User view - Products: -

```
<?php include 'includes/session.php'; ?>
<?php
    $conn = $pdo->open();

    $slug = $_GET['product'];

    try{

        $stmt = $conn->prepare("SELECT *, products.name AS prodname,
category.name AS catname, products.id AS prodid FROM products LEFT JOIN
category ON category.id=products.category_id WHERE slug = :slug");
        $stmt->execute(['slug' => $slug]);
        $product = $stmt->fetch();

    }
    catch(PDOException $e){
        echo "There is some problem in connection: " . $e->getMessage();
    }

    //page view
    $now = date('Y-m-d');
    if($product['date_view'] == $now){
        $stmt = $conn->prepare("UPDATE products SET counter=counter+1
WHERE id=:id");
        $stmt->execute(['id'=>$product['prodid']]);
    }
    else{
        $stmt = $conn->prepare("UPDATE products SET counter=1,
date_view=:now WHERE id=:id");
        $stmt->execute(['id'=>$product['prodid'], 'now'=>$now]);
    }

?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue layout-top-nav">
<script>
(function(d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s); js.id = id;
    js.src =
'https://connect.facebook.net/en_US/sdk.js#xfbml=1&version=v2.12';
    fjs.parentNode.insertBefore(js, fjs);
```



```

}(document, 'script', 'facebook-jssdk'));
</script>
<div class="wrapper">

    <?php include 'includes/navbar.php'; ?>

    <div class="content-wrapper">
        <div class="container">

            <!-- Main content -->
            <section class="content">
                <div class="row">
                    <div class="col-sm-9">
                        <div class="callout" id="callout" style="display:none">
                            <button type="button" class="close"><span aria-
hidden="true">&times;</span></button>
                            <span class="message"></span>
                        </div>
                        <div class="row">
                            <div class="col-sm-6">
                                ">
                                    <br><br>
                                    <form class="form-inline" id="productForm">
                                        <div class="form-group">
                                            <div class="input-group col-sm-5">

                                                <span class="input-group-btn">
                                                    <button type="button" id="minus"
class="btn btn-default btn-flat btn-lg"><i class="fa fa-
minus"></i></button>

                                                </span>
                                                <input type="text" name="quantity"
id="quantity" class="form-control input-lg" value="1">
                                                <span class="input-group-btn">
                                                    <button type="button" id="add"
class="btn btn-default btn-flat btn-lg"><i class="fa fa-plus"></i>
                                                    </button>
                                                </span>
                                                <input type="hidden" value="<?php echo
$product['prodid']; ?>" name="id">
                                                </div>
                                                <button type="submit" class="btn btn-
primary btn-lg btn-flat"><i class="fa fa-shopping-cart"></i> Add to
Cart</button>

                                            </div>
                                        </form>
                                    </div>
                                    <div class="col-sm-6">
                                        <h1 class="page-header"><?php echo
$product['prodname']; ?></h1>
                                        <h3><b>&#8377; <?php echo
number_format($product['price'], 2); ?></b></h3>
                                        <p><b>Category:</b> <a
href="category.php?category=<?php echo $product['cat_slug']; ?>"><?php
echo $product['catname']; ?></a></p>

```

```

        <p><b>Description:</b></p>
        <p><?php echo $product['description']; ?></p>
    </div>
</div>
<br>
    <div class="fb-comments" data-
href="http://localhost/ecommerce/product.php?product=<?php echo $slug;
?>" data-numposts="10" width="100%"></div>
    </div>
    <div class="col-sm-3">
        <?php include 'includes/sidebar.php'; ?>
    </div>
</div>
</section>

    </div>
</div>
<?php $pdo->close(); ?>
<?php include 'includes/footer.php'; ?>
</div>

<?php include 'includes/scripts.php'; ?>
<script>
$(function(){
    $('#add').click(function(e){
        e.preventDefault();
        var quantity = $('#quantity').val();
        quantity++;
        $('#quantity').val(quantity);
    });
    $('#minus').click(function(e){
        e.preventDefault();
        var quantity = $('#quantity').val();
        if(quantity > 1){
            quantity--;
        }
        $('#quantity').val(quantity);
    });
});
</script>
</body>
</html>

```

## 12] Customer/User view – Login: -

```

<?php include 'includes/session.php'; ?>
<?php
    if(isset($_SESSION['user'])) {
        header('location: cart_view.php');
    }
?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition login-page">
<div class="login-box">
    <?php
        if(isset($_SESSION['error'])) {

```

```

        echo "
            <div class='callout callout-danger text-center'>
                <p>".$_SESSION['error'].</p>
            </div>
        ";
        unset($_SESSION['error']);
    }
    if(isset($_SESSION['success'])) {
        echo "
            <div class='callout callout-success text-center'>
                <p>".$_SESSION['success'].</p>
            </div>
        ";
        unset($_SESSION['success']);
    }
    ?>
    <div class="login-box-body">
        <p class="login-box-msg">Sign in to start your session</p>

        <form action="verify.php" method="POST">
            <div class="form-group has-feedback">
                <input type="email" class="form-control" name="email"
placeholder="Email" required>
                <span class="glyphicon glyphicon-envelope form-control-
feedback"></span>
            </div>
            <div class="form-group has-feedback">
                <input type="password" class="form-control" name="password"
placeholder="Password" required>
                <span class="glyphicon glyphicon-lock form-control-
feedback"></span>
            </div>
            <div class="row">
                <div class="col-xs-4">
                    <button type="submit" class="btn btn-primary btn-block
btn-flat" name="login"><i class="fa fa-sign-in"></i> Sign In</button>
                </div>
            </div>
        </form>
        <br>
        <a href="password_forgot.php">I forgot my password</a><br>
        <a href="signup.php" class="text-center">Register a new
membership</a><br>
        <a href="index.php"><i class="fa fa-home"></i> Home</a>
    </div>
</div>

<?php include 'includes/scripts.php' ?>
</body>
</html>

```

### 13] Customer view – Add to cart: -

```

<?php
    include 'includes/session.php';

    $conn = $pdo->open();

```

```
$output = array('error'=>false);

$id = $_POST['id'];
$quantity = $_POST['quantity'];

if(isset($_SESSION['user'])) {
    $stmt = $conn->prepare("SELECT *, COUNT(*) AS numrows FROM cart
WHERE user_id=:user_id AND product_id=:product_id");
    $stmt->execute(['user_id'=>$user['id'], 'product_id'=>$id]);
    $row = $stmt->fetch();
    if($row['numrows'] < 1){
        try{
            $stmt = $conn->prepare("INSERT INTO cart (user_id,
product_id, quantity) VALUES (:user_id, :product_id, :quantity)");
            $stmt->execute(['user_id'=>$user['id'], 'product_id'=>$id,
'quantity'=>$quantity]);
            $output['message'] = 'Item added to cart';

        }
        catch(PDOException $e){
            $output['error'] = true;
            $output['message'] = $e->getMessage();
        }
    }
    else{
        $output['error'] = true;
        $output['message'] = 'Product already in cart';
    }
}
else{
    if(!isset($_SESSION['cart'])) {
        $_SESSION['cart'] = array();
    }

    $exist = array();

    foreach($_SESSION['cart'] as $row){
        array_push($exist, $row['productid']);
    }

    if(in_array($id, $exist)){
        $output['error'] = true;
        $output['message'] = 'Product already in cart';
    }
    else{
        $data['productid'] = $id;
        $data['quantity'] = $quantity;

        if(array_push($_SESSION['cart'], $data)){
            $output['message'] = 'Item added to cart';
        }
        else{
            $output['error'] = true;
            $output['message'] = 'Cannot add item to cart';
        }
    }
}
```

```
}  
  
$pdo->close();  
echo json_encode($output);  
  
?>
```

#### 14] Customer view – Delete from cart: -

```
<?php  
include 'includes/session.php';  
  
$conn = $pdo->open();  
  
$output = array('error'=>false);  
$id = $_POST['id'];  
  
if(isset($_SESSION['user'])) {  
    try {  
        $stmt = $conn->prepare("DELETE FROM cart WHERE id=:id");  
        $stmt->execute(['id'=>$id]);  
        $output['message'] = 'Deleted';  
    }  
    catch(PDOException $e) {  
        $output['message'] = $e->getMessage();  
    }  
}  
else {  
    foreach($_SESSION['cart'] as $key => $row) {  
        if($row['productid'] == $id) {  
            unset($_SESSION['cart'][$key]);  
            $output['message'] = 'Deleted';  
        }  
    }  
}  
  
$pdo->close();  
echo json_encode($output);  
  
?>
```

#### 15] Customer view – Edit/Update cart: -

```
<?php  
include 'includes/session.php';  
  
$conn = $pdo->open();  
  
$output = array('error'=>false);  
  
$id = $_POST['id'];  
$qty = $_POST['qty'];  
  
if(isset($_SESSION['user'])) {  
    try {  
        $stmt = $conn->prepare("UPDATE cart SET quantity=:quantity
```

```

WHERE id=:id");
    $stmt->execute(['quantity'=>$qty, 'id'=>$id]);
    $output['message'] = 'Updated';
}
catch(PDOException $e){
    $output['message'] = $e->getMessage();
}
}
else{
    foreach($_SESSION['cart'] as $key => $row){
        if($row['productid'] == $id){
            $_SESSION['cart'][$key]['quantity'] = $qty;
            $output['message'] = 'Updated';
        }
    }
}
}

$pdo->close();
echo json_encode($output);

?>

```

## 16] Customer view – View cart: -

```

<?php include 'includes/session.php'; ?>
<?php include 'includes/header.php'; ?>
<body class="hold-transition skin-blue layout-top-nav">
<div class="wrapper">

    <?php include 'includes/navbar.php'; ?>

    <div class="content-wrapper">
        <div class="container">

            <!-- Main content -->
            <section class="content">
                <div class="row">
                    <div class="col-sm-9">
                        <h1 class="page-header">YOUR CART</h1>
                        <div class="box box-solid">
                            <div class="box-body">
                                <table class="table table-bordered">
                                    <thead>
                                        <th></th>
                                        <th>Photo</th>
                                        <th>Name</th>
                                        <th>Price</th>
                                        <th width="20%">Quantity</th>
                                        <th>Subtotal</th>
                                    </thead>
                                    <tbody id="tbody">
                                        </tbody>
                                </table>
                            </div>
                        </div>
                    </div>
                    <?php
                        if(isset($_SESSION['user'])){
                            echo "
                                <div id='paypal-button'></div>

```

```
        ";
    }
    else{
        echo "
        <h4>You need to <a href='login.php'>Login</a>
to checkout.</h4>
        ";
    }
    ?>
</div>
<div class="col-sm-3">
    <?php include 'includes/sidebar.php'; ?>
</div>
</div>
</section>

</div>
</div>
<?php $pdo->close(); ?>
<?php include 'includes/footer.php'; ?>
</div>

<?php include 'includes/scripts.php'; ?>
<script>
var total = 0;
$(function(){
    $(document).on('click', '.cart_delete', function(e){
        e.preventDefault();
        var id = $(this).data('id');
        $.ajax({
            type: 'POST',
            url: 'cart_delete.php',
            data: {id:id},
            dataType: 'json',
            success: function(response){
                if(!response.error){
                    getDetails();
                    getCart();
                    getTotal();
                }
            }
        });
    });
});

$(document).on('click', '.minus', function(e){
    e.preventDefault();
    var id = $(this).data('id');
    var qty = $('#qty_'+id).val();
    if(qty>1){
        qty--;
    }
    $('#qty_'+id).val(qty);
    $.ajax({
        type: 'POST',
        url: 'cart_update.php',
        data: {
            id: id,
            qty: qty,
```

```
    },
    dataType: 'json',
    success: function(response) {
        if(!response.error) {
            getDetails();
            getCart();
            getTotal();
        }
    }
});
});

$(document).on('click', '.add', function(e) {
    e.preventDefault();
    var id = $(this).data('id');
    var qty = $('#qty_'+id).val();
    qty++;
    $('#qty_'+id).val(qty);
    $.ajax({
        type: 'POST',
        url: 'cart_update.php',
        data: {
            id: id,
            qty: qty,
        },
        dataType: 'json',
        success: function(response) {
            if(!response.error) {
                getDetails();
                getCart();
                getTotal();
            }
        }
    });
});

getDetails();
getTotal();

});

function getDetails() {
    $.ajax({
        type: 'POST',
        url: 'cart_details.php',
        dataType: 'json',
        success: function(response) {
            $('#tbody').html(response);
            getCart();
        }
    });
}

function getTotal() {
    $.ajax({
        type: 'POST',
        url: 'cart_total.php',
        dataType: 'json',
    })
}
```



```
        success: function(response) {
            total = response;
        }
    });
}
</script>
<!-- Paypal Express -->
<script>
paypal.Button.render({
    env: 'sandbox', // change for production if app is live,

    client: {
        sandbox: 'ASb1ZbVxG5ZFzCWLdYLi_d1-
k5rmSjvBZhxp2etCxBKXaJHxPba13JJD_D3dTNriRbAv3Kp_72cgDvaZ',
        //production: 'AaBHKJFEej4V6yaArjzSx9cuf-
UYesQYKqynQVCdBlKuZKawDDzFyuQdidPOBSGEhWaNQnnvfzuFB9SM'
    },

    commit: true, // Show a 'Pay Now' button

    style: {
        color: 'gold',
        size: 'small'
    },

    payment: function(data, actions) {
        return actions.payment.create({
            payment: {
                transactions: [
                    {
                        //total purchase
                        amount: {
                            total: total,
                            currency: 'USD'
                        }
                    }
                ]
            }
        });
    },

    onAuthorize: function(data, actions) {
        return actions.payment.execute().then(function(payment) {
            window.location = 'sales.php?pay='+payment.id;
        });
    },

}, '#paypal-button');
</script>
</body>
</html>
```

## 7.1 Snapshots of the application

The Admin can view the dashboard which consists of details such as the profile, total sales, number of products, number of users and today's sales. The Admin can navigate through various other options of adding users, products etc is shown in fig. 7.1.

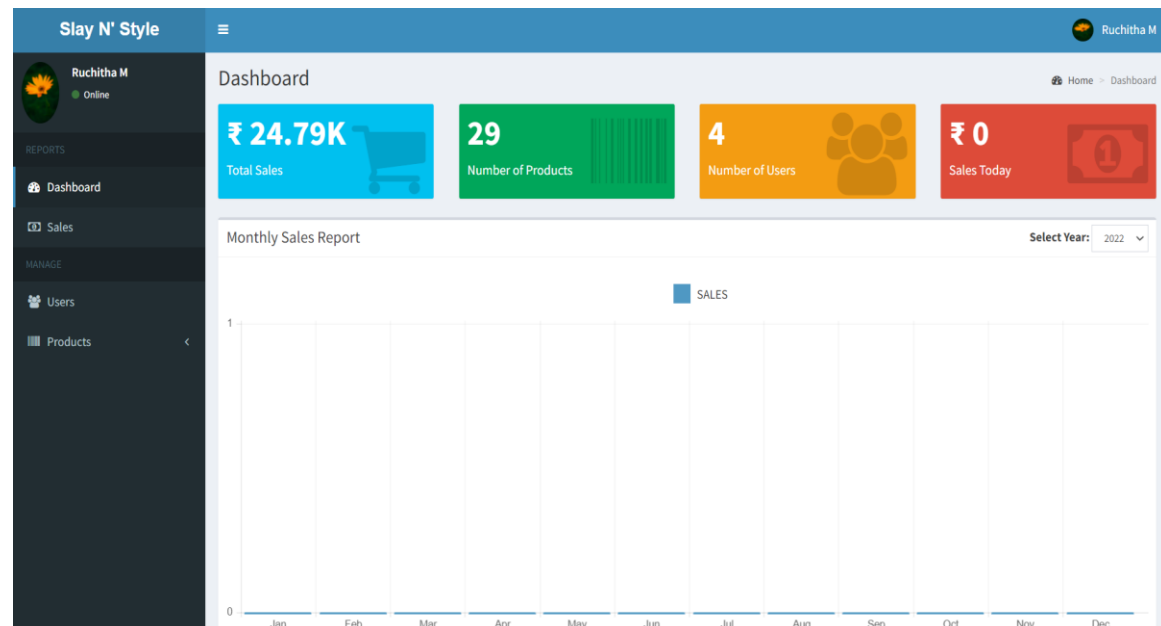


Fig. 7.1 Admin View – Dashboard/Homepage

The Admin can add or delete or update the details of products in this section is shown in fig 7.2.

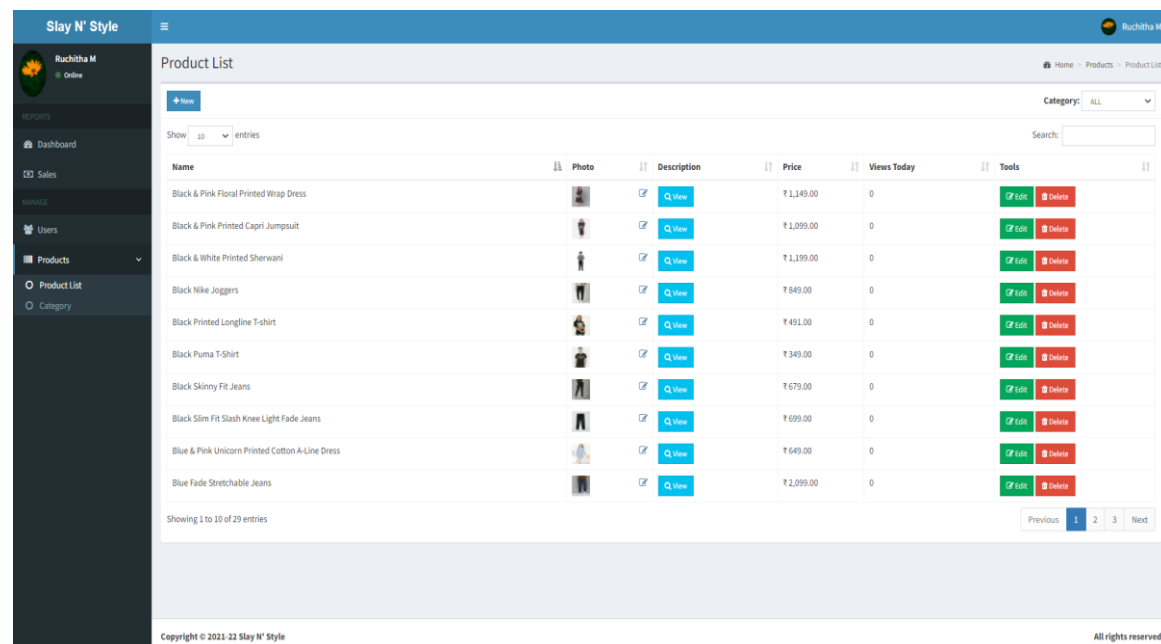


Fig. 7.2 Admin View – Add/Delete/Update Products page

The Admin can perform addition or deletion or updation of user details in this section is shown in fig 7.3.

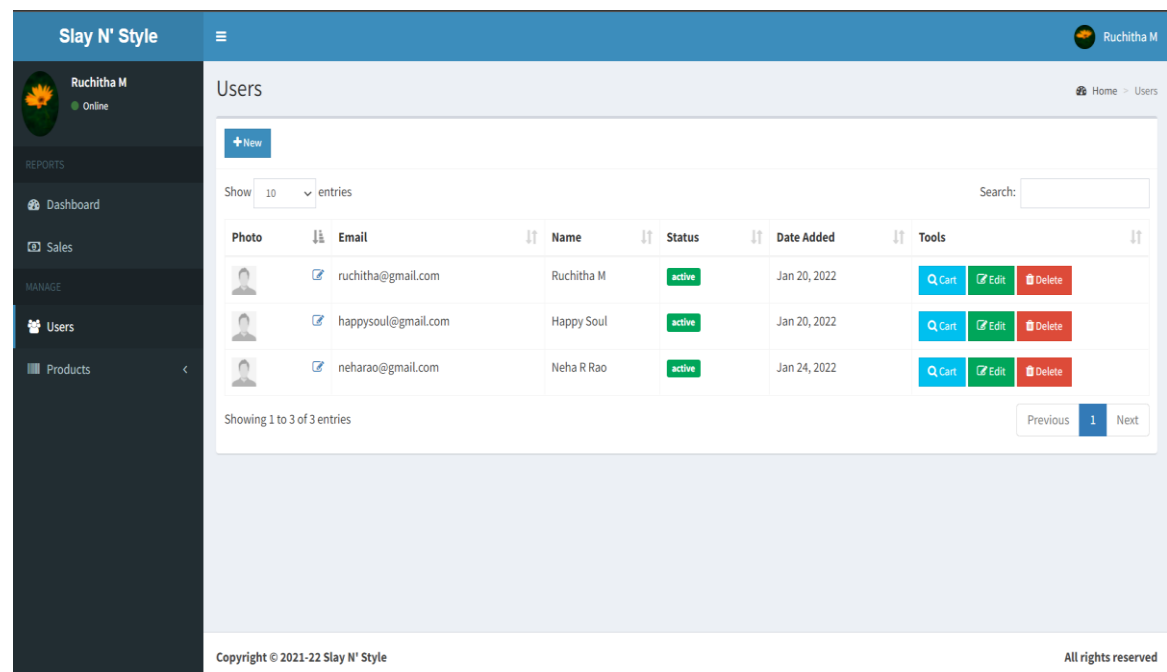


Fig. 7.3 Admin view – Add/Delete/Update Users page

The Admin can perform addition or deletion or updation of category details in this section is shown in fig 7.4.

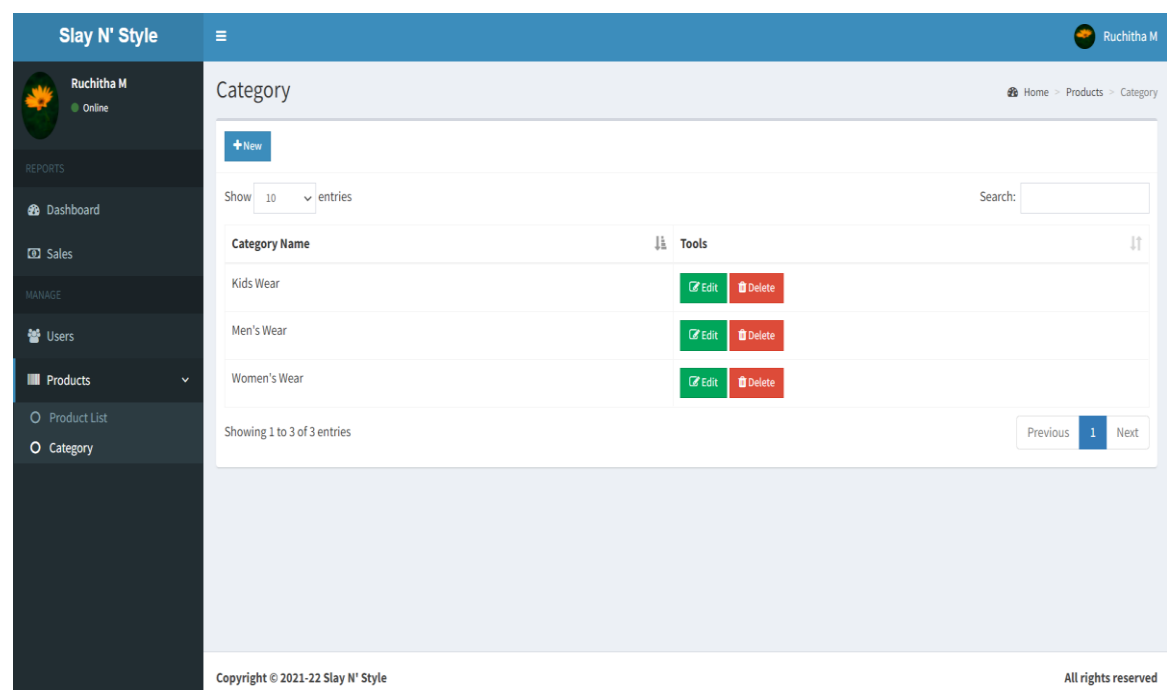


Fig. 7.4 Admin view – Add/Delete/Update Category page

The Customer/ User views this page as the homepage as soon as he/she opens the website. It contains the details of the company, what the company intends to sell. It also has the options to login from this homepage is shown in fig.7.5.

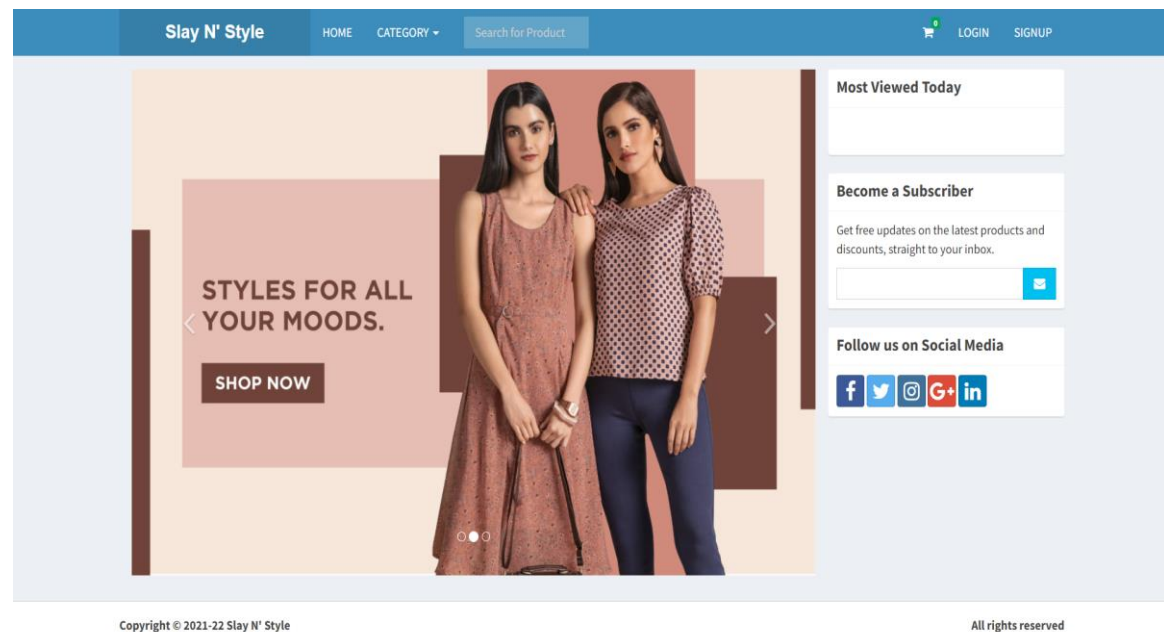


Fig. 7.5 Customer/User View – Homepage

The Customer / User can find the listing of products, navigate through them, and then access the products. He/she can choose to add the product to cart or just view them is shown in 7.6

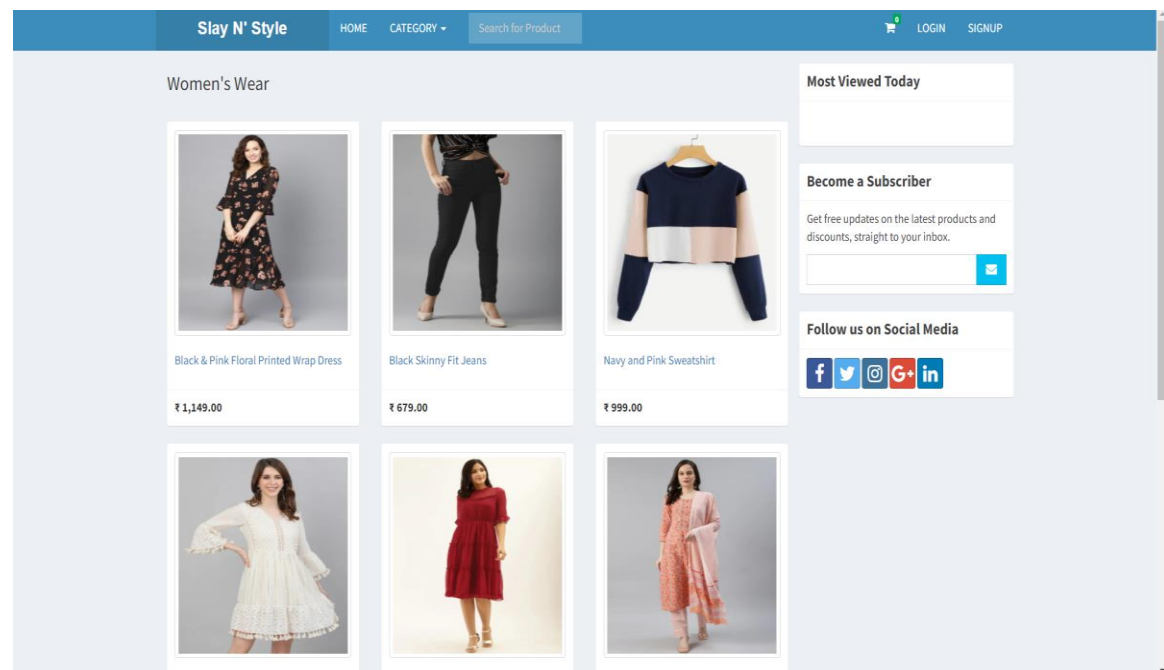


Fig. 7.6 Customer / User view – Products page

This is where the User can login into his/her account and continue to shop the products and can even purchase them is shown in 7.7

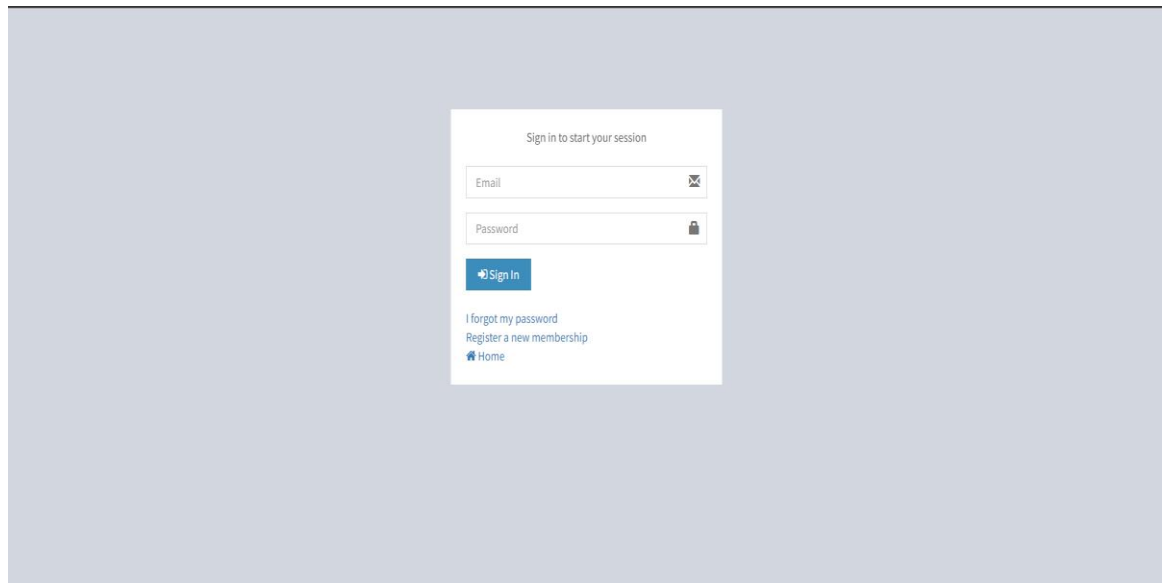
A screenshot of a login page for a clothing website. The page has a light blue background. In the center, there is a white box containing the login form. At the top of the box, it says "Sign in to start your session". Below this, there are two input fields: "Email" and "Password". The "Email" field has an eye icon on the right, and the "Password" field has a lock icon on the right. Below the input fields is a blue button with a white arrow and the text "Sign In". Under the button, there are three links: "I forgot my password", "Register a new membership", and "Home" with a house icon.

Fig 7.7 Customer / User view – Login page

The Customer can view the products that he/she has added to cart and can proceed to check from this page.

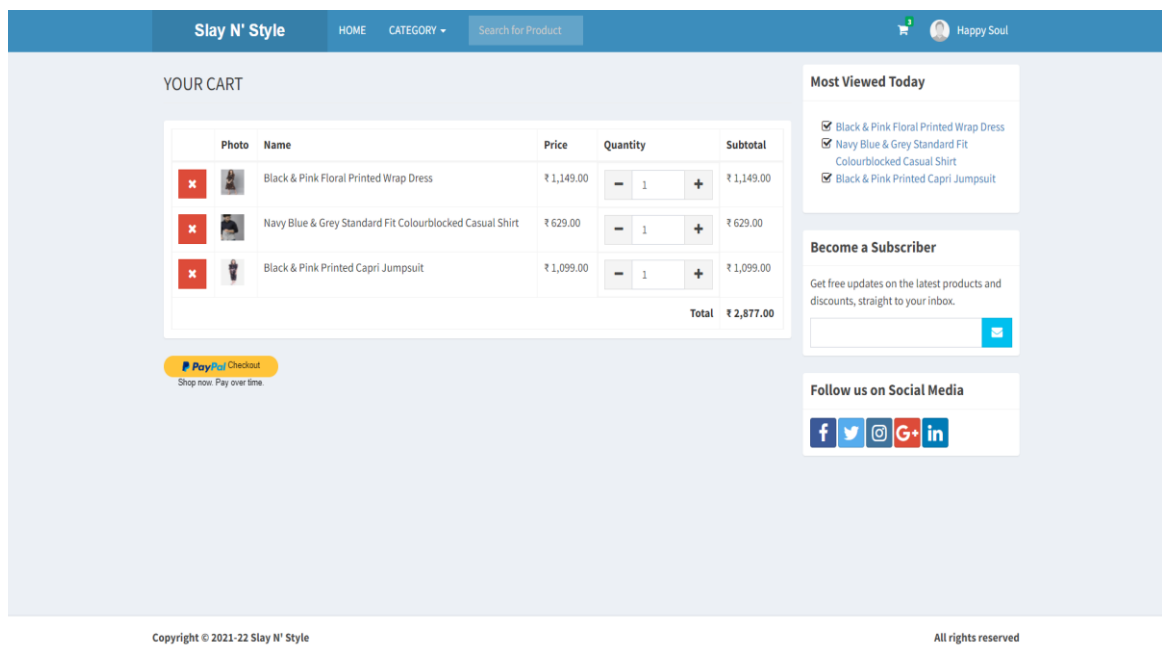
A screenshot of a customer's shopping cart page. The page has a blue header with the text "Slay N' Style" and navigation links "HOME" and "CATEGORY". There is a search bar with the text "Search for Product". On the right side of the header, there is a shopping cart icon with a number "3" and a user profile icon with the name "Happy Soul". The main content area is titled "YOUR CART". It contains a table with the following columns: "Photo", "Name", "Price", "Quantity", and "Subtotal". The table lists three items: "Black & Pink Floral Printed Wrap Dress" with a price of ₹ 1,149.00 and a subtotal of ₹ 1,149.00; "Navy Blue & Grey Standard Fit Colourblocked Casual Shirt" with a price of ₹ 629.00 and a subtotal of ₹ 629.00; and "Black & Pink Printed Capri Jumpsuit" with a price of ₹ 1,099.00 and a subtotal of ₹ 1,099.00. The total price is ₹ 2,877.00. Below the table, there is a "PayPal Checkout" button with the text "Shop now. Pay over time." On the right side of the page, there are three sections: "Most Viewed Today" with a list of products, "Become a Subscriber" with a text input field and a "Subscribe" button, and "Follow us on Social Media" with icons for Facebook, Twitter, Instagram, Google+, and LinkedIn. At the bottom of the page, there is a copyright notice "Copyright © 2021-22 Slay N' Style" and the text "All rights reserved".

Fig. 7.8 Customer view – Cart view page

## **CONCLUSION**

This website was created to provide great convenience to the shoppers and the retailers. It does not have any time and place limitations, it works around –the –clock so consumers can do purchase at anytime and anywhere. It provides great delivery services to the customers so they can plan the day around. It aids you to adapt to the new normal by making sure to have cashless transactions, effectively minimizing physical touch as we continue to provide services to the customers.

It is a perfect space for all clothing retailers.it reduces the inventory cost and eliminates the requirement of physical location. We have implemented various functionalities of HTML, CSS, JAVASCRIPT and BOOTSTRAP, MySQL, and PHP in this project. Thus, we have successfully developed an online shopping database management which helps us in centralizing the data used for managing the tasks performed in an online shopping.

## **REFERENCES**

1. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, 2017, Pearson.
2. Ramakrishnan, and Gehrke, Database management Systems, 3rd Edition, 2014, McGraw Hill
3. <https://youtu.be/KLWA2vCERSQ>
4. <https://erdplus.com/>
5. <https://resources.fabric.inc/blog/answers/ecommerce-database-design-example>