

ONLINE GROCERY STORE



Submitted for fulfillment of the requirement of **Mini project(3MCA7)** for

MASTER OF COMPUTER APPLICATIONS

Submitted by

MOHAN S (P18CD22S126046)

Name of the Guide

SUMATHI H R

ASST.PROF



COMMUNITY INSTITUTE OF MANAGEMENT STUDIES MCA DEPARTMENT

Affiliated to Bengaluru City University

**9th Cross, 9th Main, 2nd Block, Jayanagar,
Bangalore-560011.**

COMMUNITY INSTITUTE OF MANAGEMENT STUDIES**CERTIFICATE**

This is to certify that the report entitled “**ONLINE GROCERY STORE**” embodies the original work by **MOHAN S (P18CD22S126046)** the fulfilment of the requirements for Mini Project (**3MCA7**) for MCA, III Semester course during the academic semester from January 2024 to June 2024 as prescribed by Bengaluru City University.

Project Guide**Head of the Department**

(Mrs. SUMATHI H R)
ASST.PROF

(Dr. MANJULA P S)

Place: Bangalore**Date:**

COMMUNITY INSTITUTE OF MANAGEMENT STUDIES



CERTIFICATE

This is to certify that the report entitled “**ONLINE GROCERY STORE**” embodies the original work by **MOHAN S(P18CD22S126046)** the fulfilment of the requirements for Mini Project (**3MCA7**) for MCA, III Semester course during the academic semester from January 2024 to June 2024 as prescribed by Bengaluru City University.

External Examiner

1. _____

2. _____

DECLARATION

I, **MOHAN S** do here by declare that the project work entitled “**ONLINE GROCERY STORE**” is a bonafide work carried out by me under the guidance of **ASST.PROF SUMATHI H R**. This project as presented in this report is our original work and has not been presented for any other University award. This project has been submitted as partial fulfillment of requirement for the Degree of Master of computer Applications III Semester of Bangalore City University.

MOHAN S(P18CD22S126046)

Acknowledgement

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report. I take pleasure in expressing gratitude towards our institution for providing me the opportunity to work on this project as a part of our course. We would like to express my gratitude and sincere thanks to our management, I would like to express my gratitude to Principal & Director. **Dr. Mahesh Kumar K R** for providing the right atmosphere at the institute.

I express my heart full thanks to **Dr. Manjula P S**, Head of the Department, for being helpful through the project. I express my deep sense of gratitude to our project guide **Mrs. Sumathi H R**, Assistant Professor for his/her unfailing encouragement, valuable guidance and suggestions to us in the course of project work. I also express my heartfelt thanks to our parents for being supportive in all our activities without whom it wouldn't have been possible for us to complete my project.

CONTENTS

SL.NO	PARTICULARS	PG.NO
1	INTRODUCTION	1-2
2	SYNOPSIS Aim Project Description System Configuration Overview of Front End Overview of Back End Architectural Design	3-8
3	DATA FLOW DIAGRAM	9-14
4	ER-DIAGRAM	15-17
5	SOURCE CODE	18-40
6	DATABASE TABLE	41-43
7	SCREEN SHOTS	44-56
8	TESTING	57-63
9	CONCLUSION	64-65
10	FUTURE ENHANCEMENT	66-67
11	BIBLIOGRAPHY	68-70

INTRODUCTION

INTRODUCTION

The Online Grocery Delivery System is a sophisticated and user-oriented web-based platform that transforms the grocery shopping experience into a seamless and convenient process. Through a meticulously organized product catalog encompassing a wide range of categories including fresh produce, pantry staples, dairy, bakery items, and beverages, users are presented with detailed product listings that include vivid images, comprehensive descriptions, nutritional insights, pricing details, and valuable customer reviews. As users navigate the platform, they can effortlessly add desired items to their shopping carts, where they can conveniently review and adjust quantities before proceeding to the secure checkout process. During checkout, users can select their preferred delivery time slots from a variety of options, while also enjoying the flexibility of choosing from multiple payment methods, such as credit/debit cards, digital wallets, and online banking.

With each order, the system dynamically manages inventory levels in real-time to ensure accurate order fulfillment, while users are kept informed about the status of their orders through automated notifications at every significant step, including order confirmation, packaging, dispatch, and estimated delivery times.

Behind the scenes, the administrative dashboard empowers platform administrators with comprehensive control over various aspects of the system. This centralized hub allows them to manage product listings, update inventory, oversee user accounts, and monitor the seamless execution of order fulfillment. Administrators can further enrich the user experience by creating promotions, discounts, and special offers.

SYNOPSIS

SYNOPSIS

Title: Online Grocery Delivery

Aim

The Online Grocery Delivery System project aims to develop a user-friendly and efficient platform that enables customers to order groceries online and have them delivered to their doorstep. This project leverages the power of e-commerce and logistics to provide a convenient and seamless shopping experience for user

Project Description

An Online Grocery Delivery System is either a brick-and-mortar supermarket or grocery store that allows online ordering, or a standalone e-commerce service that includes grocery items. There is usually a delivery charge for this service. Supermarkets that have built internet channels to better serve their clients are known as online grocers. Online grocery delivery services are available throughout Europe, Asia, and North America, mostly in urban centers. Online ordering is done through e-commerce websites or mobile apps. The COVID-19 pandemic greatly accelerated the growth of online grocers, and in the first few months of the pandemic, online grocery shopping increased by 300%. In addition, first-time online grocery shoppers accounted for 41% of online grocery shoppers. The epidemic of COVID-19 has hastened the uptake of online grocery shopping. Pre- COVID-19 food shopping activity accounted for 9% of the market, but 63 percent of consumers worldwide purchased more groceries online after the outbreak than they did before they were socially isolated. This project is a web-based Groceries shopping system for customers who can order from their homes or anywhere..

System Configuration

3.1 Hardware Requirements (MINIMUM)	
Processor	Intel i3
Hard Disk Capacity	10 GB
Memory	4GB RAM
Monitor	LG

3.2 Software Requirements (MINIMUM)	
<u>Server – Side:</u>	
Operating System	Windows 10/9/8
Front End	HTML, CSS, Java Script
Back End	PHP, MY SQL
Server	Tomcat5.0/6.X
<u>Client – Side:</u>	
Internet Browser	Internet Explorer, Google Chrome, Mozilla Firefox

Modules/Forms

- Index
- Register
- Registration Completed
- Login
- Logout
- Admin Home
- Admin Profile Update
- Admin Add Products
- Admin Update Products
- Admin View Products
- Admin Contact
- Admin Orders
- Admin view Pending Order
- Admin View Completed Order
- Admin View Users
- Customer Home
- Customer Profile Update
- Customer Shop
- Customer Search Products
- Customer Orders
- Generate Receipt
- About
- Contact
- Wishlist
- Cart
- Select Payment
- Place Your Order

Overview of Front End

HTML: - HTML serves as the foundational building block of the front-end development process, playing a crucial role in shaping the visual and structural aspects of web pages. As the primary markup language used for creating content on the internet, HTML provides the structure and framework that allows web browsers to render information in a coherent and organized manner. By employing a system of tags and elements, developers can define the various components of a web page, such as headings, paragraphs, images, links, forms, and more.

Front-end developers utilize HTML to craft the layout and arrangement of content, ensuring a clear and logical presentation that aligns with the intended user experience. HTML tags delineate the hierarchy of information, from the overall page structure down to the minutiae of individual text formatting and multimedia embedding. This hierarchical structure facilitates accessibility, search engine optimization, and user navigation, as it enables screen readers and search engines to interpret and understand the content's significance.

CSS: -Cascading Style Sheets (CSS) occupies a pivotal role in front-end web development, providing the means to breathe life and artistry into the structural skeleton forged by HTML. CSS is the language that governs the presentation and aesthetics of web pages, enabling developers to define the visual aspects that users perceive and interact with.

Through CSS, developers can control layout, colors, typography, spacing, animations, and responsiveness, thereby transforming raw HTML content into engaging and visually captivating experiences.

Overview of Back End

PHP: -PHP, known as Hypertext Preprocessor, stands as a potent and widely adopted back-end tool for web development. Functioning on the server side, PHP orchestrates the dynamic and data-centric facets of web applications. By executing scripts before the page reaches users' browsers, it manages tasks like processing user requests, database interactions, and data manipulation. PHP's capacity to generate dynamic content, tailored to user inputs and interactions, enables personalized and responsive web experiences. Notably, it seamlessly interfaces with databases, allowing for data retrieval, storage, and management. Additionally, PHP handles form submissions, validates inputs, and manages user sessions, contributing to secure and interactive applications. Its role extends to serving as the bedrock for building APIs, creating content management systems, and integrating third-party applications.

Embracing PHP frameworks further streamlines development, promoting standardized practices and accelerated coding. Despite its power, developers must exercise caution in mitigating security risks, employing techniques like input sanitization and secure database querying. In essence, PHP empowers developers to imbue web applications with dynamic functionality, robust database interactions, and secure server-side processing.

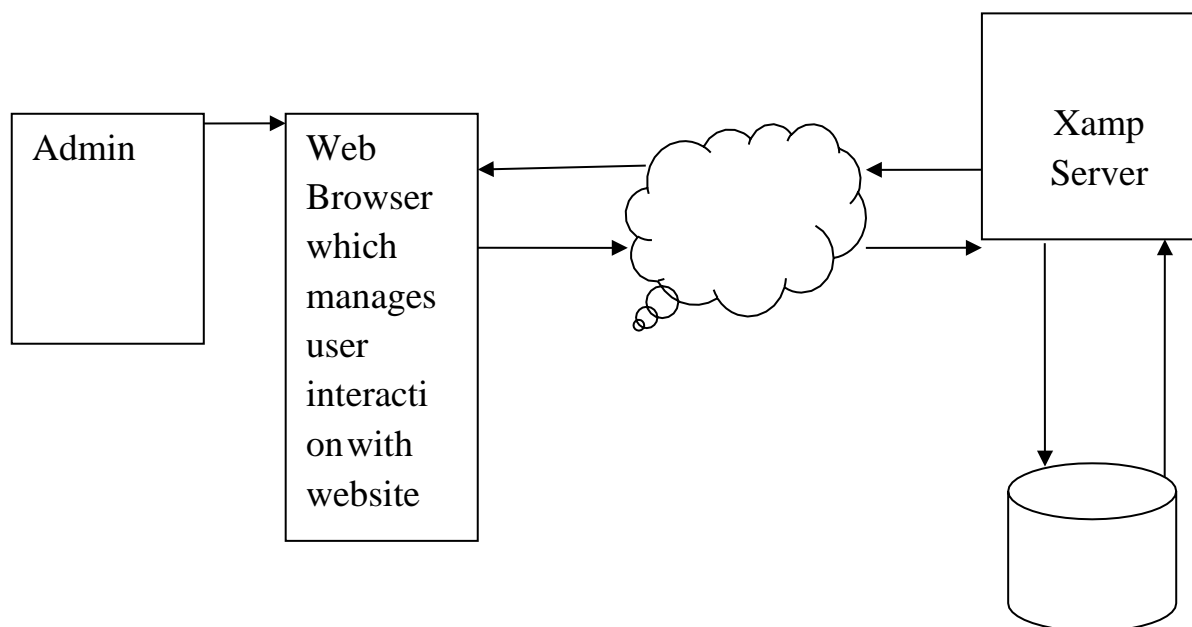
MYSQL: - MySQL stands as an integral and trusted back-end tool, providing the essential relational database management system for web development. At the core of its role lies the seamless handling of data storage and retrieval, offering a structured framework for organizing information within tables. Its data persistence capabilities enable web applications to efficiently store, manage, and retrieve data, underpinning the functionality of diverse applications.

MySQL's versatile querying capabilities, powered by SQL, empower developers to construct intricate queries that fetch, modify, insert, or delete data based on specific conditions. This interaction with the database occurs swiftly and efficiently, enhanced by indexing and optimization features that streamline data retrieval.

MySQL further contributes to an application's scalability and performance by offering mechanisms like replication and sharding, which distribute workloads and maintain consistent responsiveness even amidst increased demands. In essence, MySQL forms the bedrock of the back-end architecture, ensuring data integrity, accessibility, and efficiency for web applications.

Architectural Design

An architectural model (in software) is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem. It is the concept that focuses on the components or elements of a structure or system and unifies them into a coherent and functional whole, according to a particular approach in achieving the objective(s) under the given constraints or limitations. An architecture design is a graphical representation of a set of concepts that are part of an architecture, including their principles, elements and components



DATA FLOW DIAGRAM

DATA FLOW DIAGRAM

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and are transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

PHYSICAL VS LOGICAL DFD

A logical DFD captures the data flows that are necessary for a system to operate. It describes the processes that are undertaken, the data required and produced by each process, and the stores needed to hold the data. On the other hand, a physical DFD shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intends it to be in the future (Required Physical DFD).

Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet.

DATA FLOW SYMBOLS AND THEIR MEANINGS:





An entity: A source of data or a destination for data.

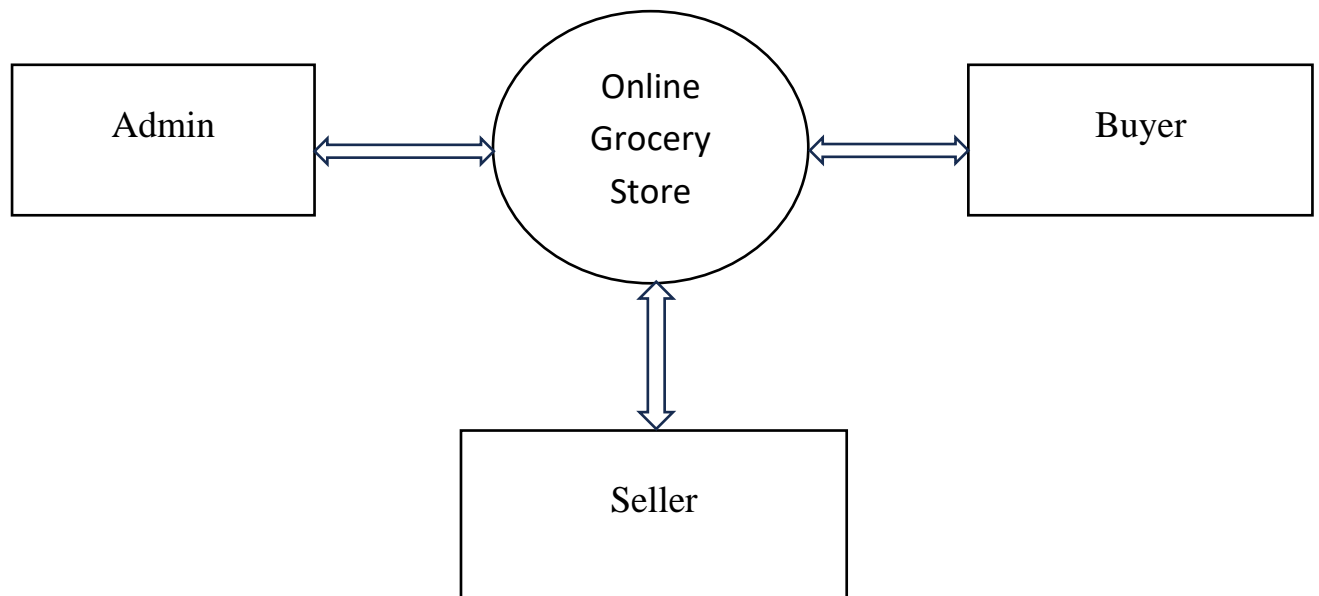
Source/Sink: Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.

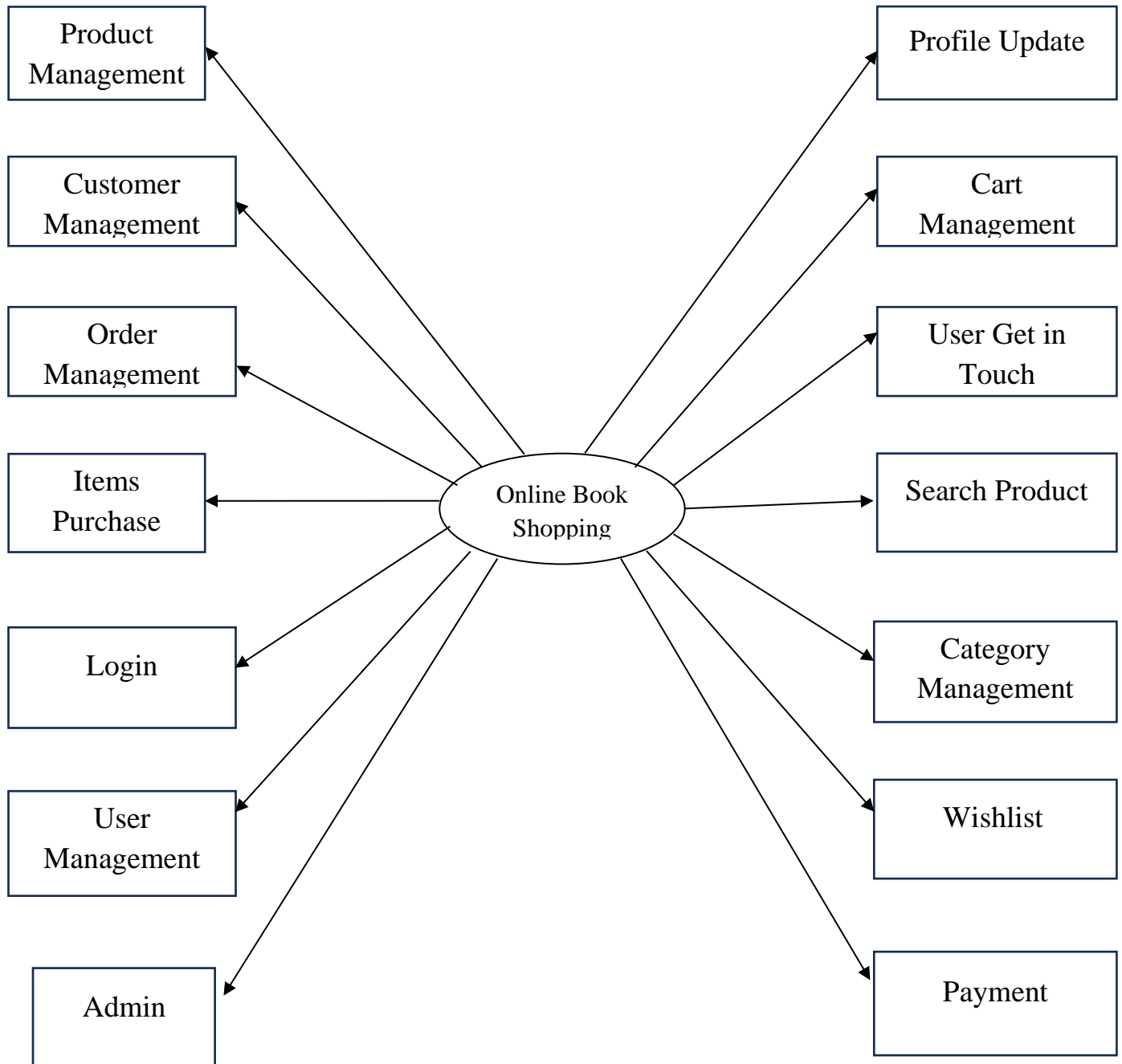
Process: Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.

Data Store: Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper

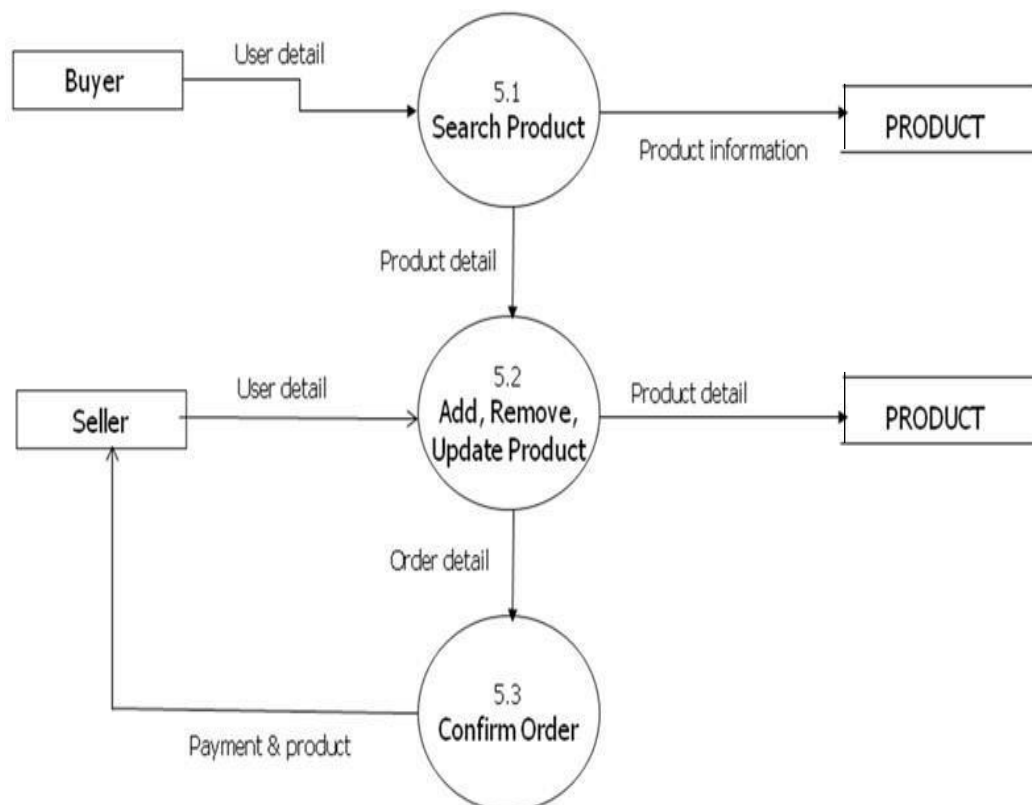
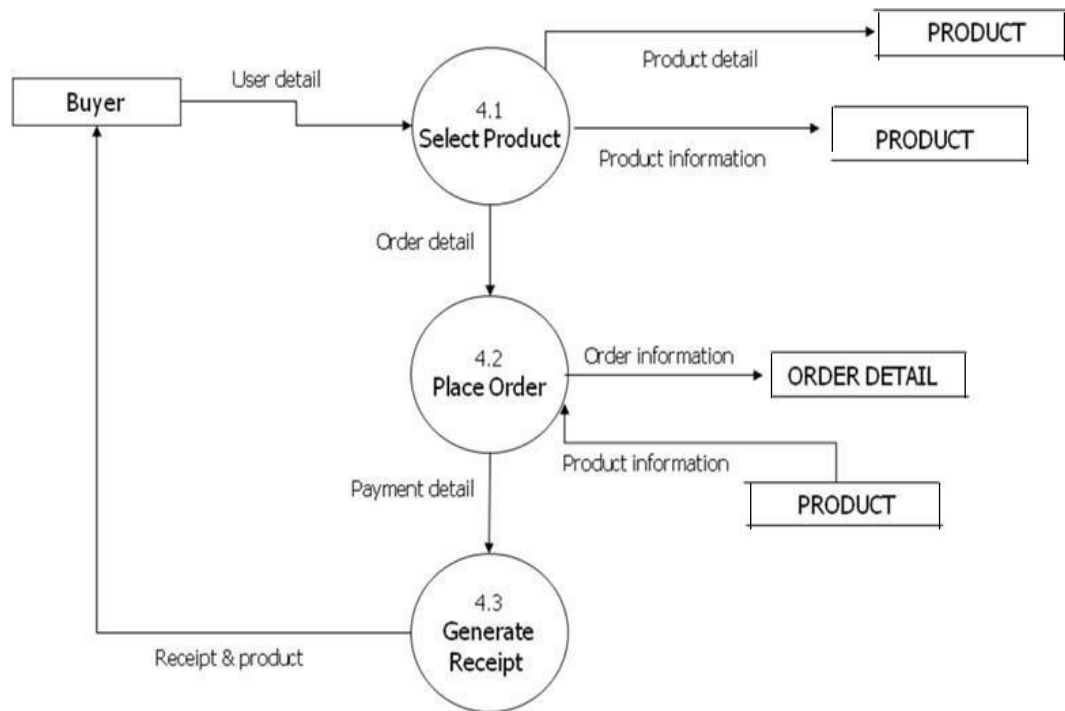
Data Flow: Represented by a unidirectional arrow. Data Flows show how data is moved through the System. Data Flows are labelled with a description of the data that is being passed through i

	dataflow	Arrows showing direction of flow
	process	circles
	file	horizontal pair of lines
	data-source, sink	rectangular box

Zero Level Data Flow Diagram

First Level Data Flow Diagram

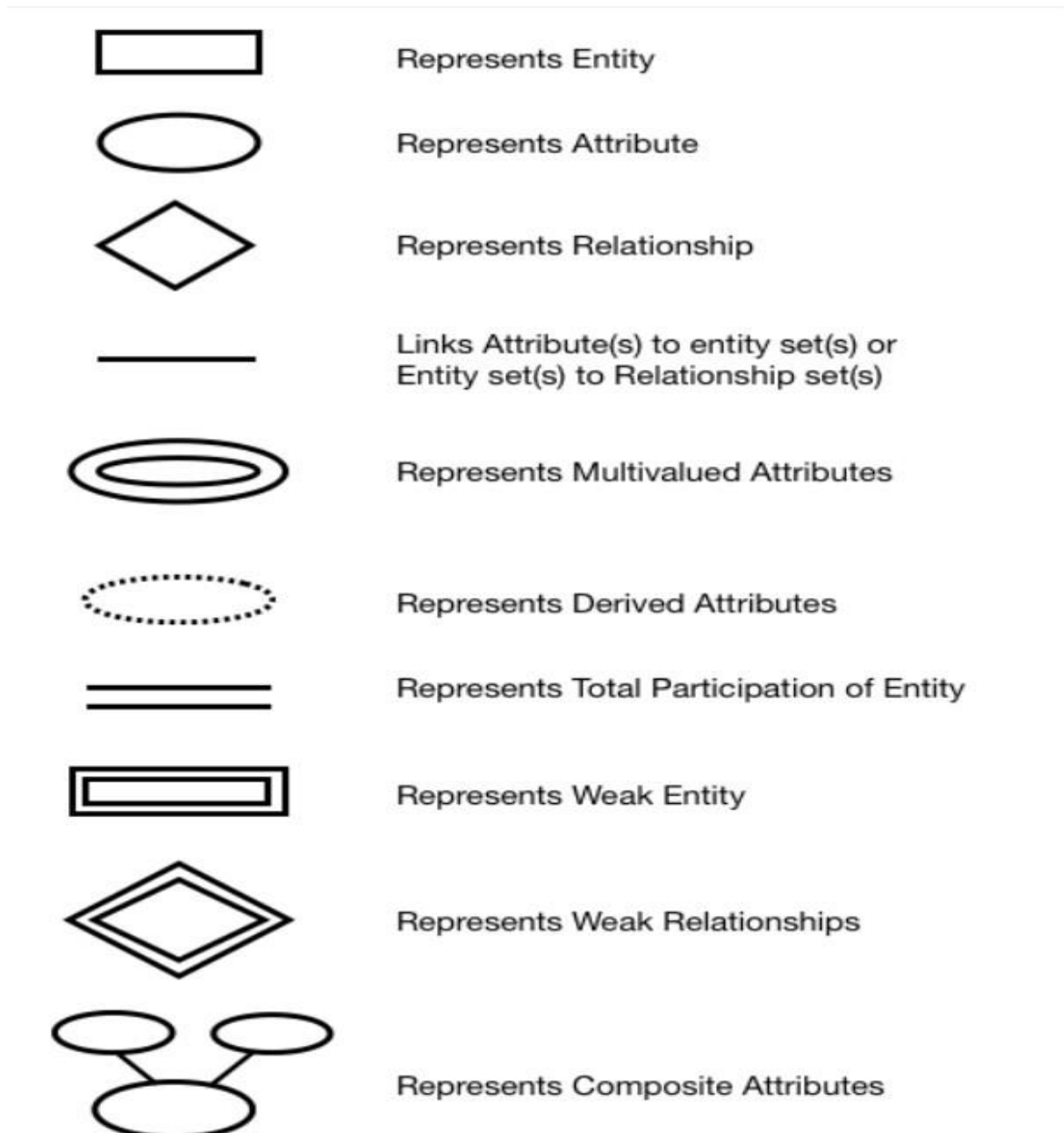
Second Level Data Flow Diagram

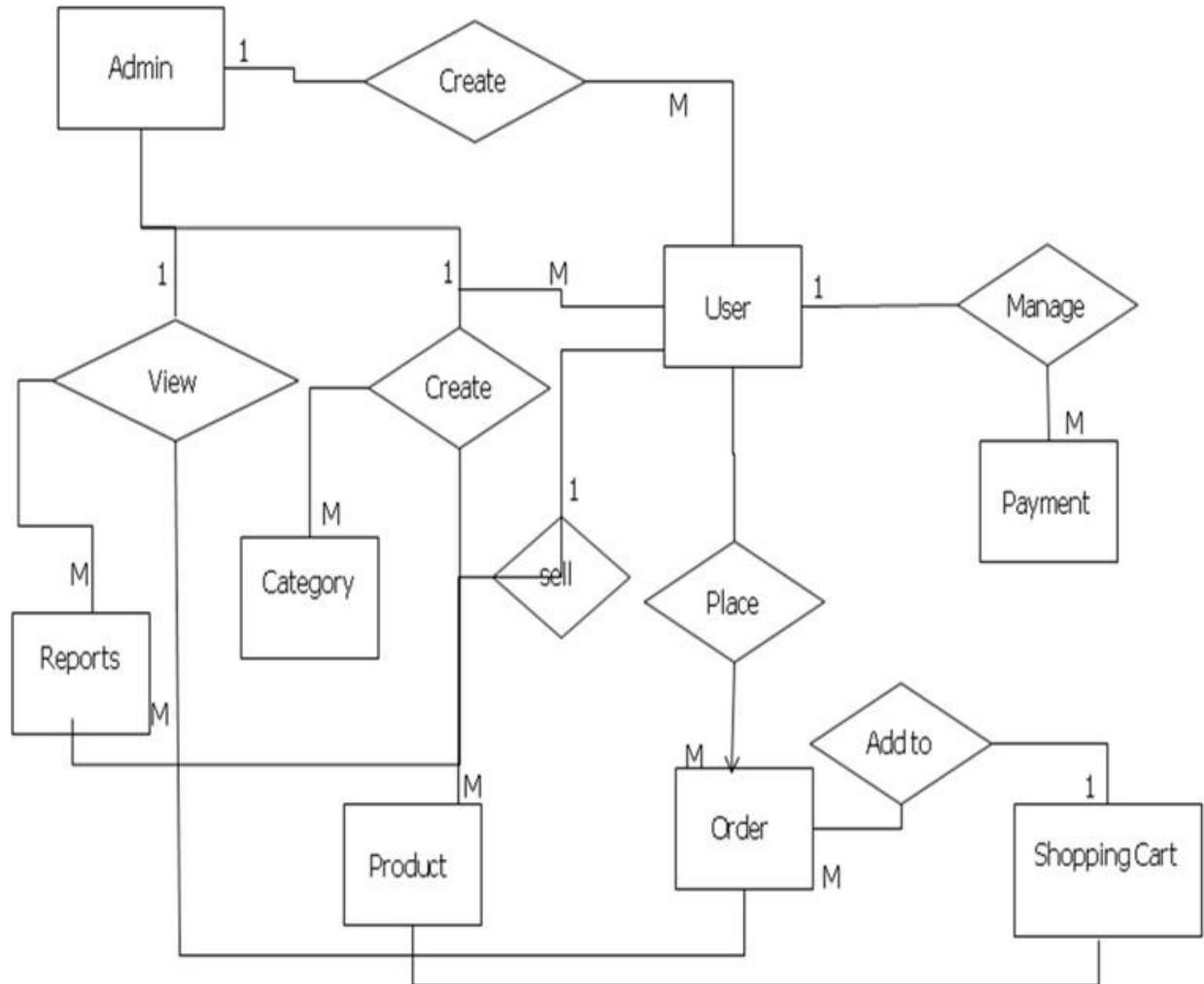


ENTITY RELATIONSHIP MODEL

ENTITY-RELATIONSHIP MODEL

An Entity-Relationship (ER) model is a visual representation that depicts the logical structure of a database. It consists of entities, which represent objects or concepts, and their relationships, which describe how these entities are connected to each other.



ER Diagram

SOURCE CODE

SOURCE CODE

Register Page

```
?php
Include'config.php';
if(isset($_POST['submit']))
{
    $name = $_POST['name'];
    $name = filter_var($name, FILTER_SANITIZE_STRING);
    $email = $_POST['email'];

    $email = filter_var($email, FILTER_SANITIZE_STRING);

    $pass = md5($_POST['pass']);

    $pass = filter_var($pass, FILTER_SANITIZE_STRING);

    $cpass = md5($_POST['cpass']);

    $cpass = filter_var($cpass, FILTER_SANITIZE_STRING);

    $image = $_FILES['image']['name'];

    $image = filter_var($image, FILTER_SANITIZE_STRING);

    $image_size = $_FILES['image']['size'];

    $image_tmp_name = $_FILES['image']['tmp_name'];
    $image_folder = 'uploaded_img/'.$image;

    $select = $conn->prepare ("SELECT * FROM `users` WHERE email = ?");

    $select->execute([$email]);

    if($select->rowCount() > 0)
    {
        $message[] = 'user email already exist!';

    } else
    {
```

```

    If ($pass != $cpass)
    {
        $message[] = 'confirm password not matched!';
    }
    Else
    {
        $insert = $conn->prepare("INSERT INTO `users`(name, email, password, image)
        VALUES(?,?,?,?)");

        $insert->execute([$name, $email, $pass, $image]);
        if($insert)
        {
            if($image_size > 2000000)
            {
                $message[] = 'image size is too large!';
            }else
            {
                move_uploaded_file($image_tmp_name, $image_folder);
                $message[]='registeredsuccessfully!';
                header('location:login.php');
            }
        }
    }
}
?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>register</title>

```

```

<!-- font awesome cdn link -->
<linkrel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font_awesome/6.1.1/css in.css">

<!-- custom css file link -->
<link rel="stylesheet" href="css/components.css">
</head>
<body>
<?php if(isset($messag){
    foreach($message as $message){echo '
        <div class="message">

            <span>'.$message.'</span>

            <i class="fas fa-times" onclick="this.parentElement.remove();"></i>

        </div>';
    }
}

?>

<section class="form-container">
    <form action="" enctype="multipart/form-data" method="POST">

        <h3>register now</h3>

        <input type="text" name="name" class="box" placeholder="enter your name"
required>

        <input type="email" name="email" class="box" placeholder="enter youremail"
required>

        <input type="password" name="pass" class="box" placeholder="enter your
password" required>

        <input type="password" name="cpass" class="box" placeholder="confirm your
password" required>

        <input type="file" name="image" class="box" required accept="image/jpg,
image/jpeg, image/png">

        <input type="submit" value="register now" class="btn" name="submit">

        <p>already have an account? <a href="login.php">login now</a></p>

```

```
</form>
```

```
</section>
```

```
</body>
```

```
</html>
```

Login Page

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>login</title>
```

```
<!-- font awesome cdn link -->
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
```

```
<!-- custom css file link -->
```

```
<link rel="stylesheet" href="css/components.css">
```

```
</head>
```

```
<body>
```

```
<?php if(isset($message)){
    foreach($message as $message){ echo '
        <div class="message">
```

```

            <span>'.$message.'</span>
```

```

            <i class="fas fa-times" onclick="this.parentElement.remove();"></i>
```

```

    </div>';
  }
}
?>

<section class="form-container">

  <form action="" method="POST">

    <h3>login now</h3>

    <input type="email" name="email" class="box" placeholder="enter your email"
required>

    <input type="password" name="pass" class="box" placeholder="enter your
password" required>

    <input type="submit" value="login now" class="btn" name="submit">

    <p>don't have an account? <a href="register.php">register now</a></p>

  </form>

</section>

</body></html>

```

Home Page

```

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>home page</title>

```

```
<!-- font awesome cdn link -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

<!-- custom css file link -->

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<?php include 'header.php'; ?>

<div class="home-bg">

    <section class="home">
        <div class="content">

            <span>SRI RAGHAVENDRA STORES</span>

            <h3>Reach For A Healthier You With Organic Foods</h3>

            <p>Whoever said money can't buy happiness didn't know where to shop</p>

            <a href="about.php" class="btn">about us</a>

        </div>

    </section>

</div>

<section class="home-category">

    <h1 class="title">shop by category</h1>

    <div class="box-container">

        <div class="box">

            
```


Fertilizers

Fertilisers are additional substances supplied to the crops to increase their productivity.

[Fertilizers](category.php?category=fruits)

Eatable Items

Eatable Items means edible items and drinks which are not furnished and they are ready for immediate consumption

[Eatable Items](category.php?category=meat)

Snacks

a small portion of food that is smaller than a regular meal, consumed between meals.

[Snacks](category.php?category=vegetables)

Other Grocery

A grocer is a shopkeeper who sells foods such as flour, sugar, and tinned foods.

[Other Grocery](category.php?category=fish)

```

    </div>
</div>

</section>

<section class="products">

    <h1 class="title">latest products</h1>

    <div class="box-container">

        <?php

$select_products = $conn->prepare("SELECT * FROM `products` LIMIT 6");

    $select_products->execute();
    if($select_products->rowCount() > 0){
while($fetch_products = $select_products->fetch(PDO::FETCH_ASSOC)){

    ?>

    <form action="" class="box" method="POST">

<div class="price">Rs<span><?= $fetch_products['price']; ?></span>/-</div>

    <a href="view_page.php?pid=<?= $fetch_products['id']; ?>" class="fas fa-
eye"></a>

    <div class="name"><?= $fetch_products['name']; ?></div>

    <input type="hidden" name="pid" value="<?= $fetch_products['id']; ?>">

    <input type="hidden" name="p_name" value="<?= $fetch_products['name'];
?>">

```

```
<input type="hidden" name="p_price" value="<?= $fetch_products['price'];
?>">

<input type="hidden" name="p_image" value="<?= $fetch_products['image'];
?>">

<input type="number" min="1" value="1" name="p_qty" class="qty">

<input type="submit" value="add to wishlist" class="option-btn"
name="add_to_wishlist">

<input type="submit" value="add to cart" class="btn" name="add_to_cart">

</form>

<?php
}

}else{

    echo '<p class="empty">no products added yet!</p>';

}

?>

</div>

</section>

<?php include 'footer.php'; ?>

<script src="js/script.js"></script>
</body>
</html>
```

Admin Page

```

<!DOCTYPE html>
<html lang="en">
<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>admin page</title>

    <!-- font awesome cdn link -->

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/6.1.1/css/
almin.css">

    <!-- custom css file link -->
    <link rel="stylesheet" href="css/admin_style.css">
</head>
<body>

<?php include 'admin_header.php'; ?>

<section class="dashboard">

    <h1 class="title">dashboard</h1>

    <div class="box-container">

        <div class="box">

            <?php

                $total_pendings = 0;

                $select_pendings = $conn->prepare("SELECT * FROM `orders` WHERE
                payment_status = ?");

                $select_pendings->execute(['pending']);

                while($fetch_pendings = $select_pendings->fetch(PDO::FETCH_ASSOC))

```

```

{

    $total_pendings += $fetch_pendings['total_price'];

};

?>

<h3>Rs<?= $total_pendings; ?>/-</h3>

<p>total pendings</p>

<a href="admin_orders.php" class="btn">see orders</a>

</div>

<div class="box">

<?php

    $total_completed = 0;
$select_completed=$conn->prepare("SELECT*FROM orders 'WHERE payment_ status =
?");

    $select_completed->execute(['completed']);

    while($fetch_completed = $select_completed->fetch(PDO::FETCH_ASSOC))
    {

        $total_completed += $fetch_completed['total_price'];

    };

?>

<h3>Rs<?= $total_completed; ?>/-</h3>

<p>completed orders</p>

<a href="admin_orders.php" class="btn">see orders</a>

</div>

<div class="box">

```

```
<?php
```

```
$select_orders = $conn->prepare("SELECT * FROM `orders`");
```

```
$select_orders->execute();
```

```
$number_of_orders = $select_orders->rowCount();
```

```
?>
```

```
<h3><?= $number_of_orders; ?></h3>
```

```
<p>orders placed</p>
```

```
<a href="admin_orders.php" class="btn">see orders</a>
```

```
</div>
```

```
<div class="box">
```

```
<?php
```

```
$select_products = $conn->prepare("SELECT * FROM `products`");
```

```
$select_products->execute();
```

```
$number_of_products = $select_products->rowCount();
```

```
?>
```

```
<h3><?= $number_of_products; ?></h3>
```

```
<p>products added</p>
```

```
<a href="admin_products.php" class="btn">see products</a>
```

```
</div>
```

```
<div class="box">
```

```
<?php
```

```
$select_users = $conn->prepare("SELECT * FROM `users` WHERE user_type = ?");
```

```

$select_users->execute(['user']);

$number_of_users = $select_users->rowCount();

?>
<h3><?= $number_of_users; ?></h3>

<p>total users</p>

<a href="admin_users.php" class="btn">see accounts</a>

</div>

<div class="box">

<?php

$select_admins = $conn->prepare("SELECT * FROM `users` WHERE user_type = ?");
    $select_admins->execute(['admin']);

    $number_of_admins = $select_admins->rowCount();

?>

<h3><?= $number_of_admins; ?></h3>

<p>total admins</p>

<a href="admin_users.php" class="btn">see accounts</a>

</div>

<div class="box">

<?php

$select_accounts = $conn->prepare("SELECT * FROM `users`");

$select_accounts->execute();
$number_of_accounts = $select_accounts->rowCount();

?>

```

```

<h3><?= $number_of_accounts; ?></h3>

<p>total accounts</p>

<a href="admin_users.php" class="btn">see accounts</a>

</div>

<div class="box">

<?php

    $select_messages = $conn->prepare("SELECT * FROM `message`");

    $select_messages->execute();

    $number_of_messages = $select_messages->rowCount();

?>

<h3><?= $number_of_messages; ?></h3>

<p>total messages</p>

<a href="admin_contacts.php" class="btn">see messages</a>

</div>

</div>

</section>

<script src="js/script.js"></script>
</body>

</html>

```

Update Products

```

<?php

@include'config.php';session_start();

```



```
$admin_id=$_SESSION['admin_id'];
if(!isset($admin_id))
{
    header('location:login.php');

};
if(isset($_POST['update_product']))
{
    $pid = $_POST['pid'];

    $name = $_POST['name'];

    $name = filter_var($name, FILTER_SANITIZE_STRING);

    $price = $_POST['price'];

    $price = filter_var($price, FILTER_SANITIZE_STRING);

    $category = $_POST['category'];

    $category = filter_var($category, FILTER_SANITIZE_STRING);
    $details = $_POST['details'];

    $details = filter_var($details, FILTER_SANITIZE_STRING);

    $image = $_FILES['image']['name'];

    $image = filter_var($image, FILTER_SANITIZE_STRING);

    $image_size = $_FILES['image']['size'];

    $image_tmp_name = $_FILES['image']['tmp_name'];

    $image_folder = 'uploaded_img/'.$image;

    $old_image = $_POST['old_image'];

    $update_product = $conn->prepare("UPDATE `products` SET name = ?,category
= ?, details = ?, price = ? WHERE id = ?");

    $update_product->execute([$name, $category, $details, $price, $pid]);

    $message[] = 'product updated successfully!';if(!empty($image)){
        if($image_size > 2000000){
```

```

$message[] = 'image size is too large!';

}
Else

{
$update_image = $conn->prepare("UPDATE `products` SET image = ?
WHERE id = ?");

$update_image->execute([$image,$pid]);
if($update_image)
{
    move_uploaded_file($image_tmp_name,$image_folder);
    unlink('uploaded_img/'.$old_image);
    $message[] = 'image updated successfully!';

}

}

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>update products</title>

    <!-- font awesome cdn link -->

```

```

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css">

<!-- custom css file link -->
<link rel="stylesheet" href="css/admin_style.css">

</head>

<body>

<?php include 'admin_header.php'; ?>

<section class="update-product">

    <h1 class="title">update product</h1>

    <?php

        $update_id = $_GET['update'];
        $select_products = $conn->prepare("SELECT * FROM `products` WHERE id= ?");

        $select_products->execute([$update_id]);

        if($select_products->rowCount() > 0)
        {
            while($fetch_products = $select_products->fetch(PDO::FETCH_ASSOC)){

        ?>

<form action="" method="post" enctype="multipart/form-data">

    <input type="hidden" name="old_image" value="<?=$fetch_products['image']; ?>">

    <input type="hidden" name="pid" value="<?=$fetch_products['id']; ?>">

    
    <input type="text" name="name" placeholder="enter product name" required
    class="box" value="<?=$fetch_products['name']; ?>">

    <input type="number" name="price" min="0" placeholder="enter product price"
    required class="box" value="<?=$fetch_products['price']; ?>">

    <select name="category" class="box" required>

```

```
<option selected><?= $fetch_products['category']; ?></option>
```

```
<option value="Fertilizers">Fertilizers</option>
```

```
<option value="Eatable Items">Eatable Items</option>
```

```
<option value="Snacks">Snacks</option>
```

```
<option value="Other Grocery">Other Grocery</option>
```

```
</select>
```

```
<textarea name="details" required placeholder="enter product details" class="box"
cols="30" rows="10"><?= $fetch_products['details']; ?></textarea>
```

```
<input type="file" name="image" class="box" accept="image/jpg, image/jpeg,
image/png">
```

```
<div class="flex-btn">
```

```
<input type="submit" class="btn" value="update product" name="update_product">
```

```
<a href="admin_products.php" class="option-btn">go back</a>
```

```
</div>
```

```
</form>
```

```
<?php
}
```

```
}
```

```
Else
```

```
{
```

```
echo '<p class="empty">no products found!</p>';
```

```
}
```

```
?>
```

```
</section>
```

```
<script src="js/script.js"></script>
```

```
</body>
```

```
</html>
```

Cart Page

```

<?php
@include'config.php';session_start();
$user_id=$_SESSION['user_i']
if(!isset($user_id))
{
    header('location:login.php');

};
if(isset($_GET['delete'])) {

    $delete_id = $_GET['delete'];

    $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE id = ?");

    $delete_cart_item->execute([$delete_id]);header('location:cart.php');
}

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

    $delete_cart_item = $conn->prepare("DELETE FROM `cart` WHERE user_id =
?");

    $delete_cart_item->execute([$user_id]);header('location:cart.php');
}

if(isset($_POST['update_qty'])) {

    $cart_id = $_POST['cart_id'];

    $p_qty = $_POST['p_qty'];

    $p_qty = filter_var($p_qty, FILTER_SANITIZE_STRING);

    $update_qty = $conn->prepare("UPDATE `cart` SET quantity = ? WHERE id =?");

    $update_qty->execute([$p_qty, $cart_id]);
    $message[] = 'cart quantity updated';
}

```

```

?>
<!DOCTYPE html>
<html lang="en">
<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>shopping cart</title>

    <!-- font awesome cdn link -->

<linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/6.1.1/css/all.min.css">

    <!-- custom css file link -->

    <link rel="stylesheet" href="css/style.css">

</head>

<body>

<?php include 'header.php'; ?>

<section class="shopping-cart">

    <h1 class="title">products added</h1>
    <div class="box-container">

        <?php

            $grand_total = 0;

$select_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id =?");

            $select_cart->execute([$user_id]);
            if($select_cart->rowCount() > 0)
            {
                while($fetch_cart = $select_cart->fetch(PDO::FETCH_ASSOC))

```

```

    {
?>
<form action="" method="POST" class="box">

    <a href="cart.php?delete=<?= $fetch_cart['id']; ?>" class="fas fa-times"
onclick="return confirm('delete this from cart?');"></a>

    <a href="view_page.php?pid=<?=$fetch_cart['pid'];?>" class="fas fa-eye"></a>

    <div class="name"><?= $fetch_cart['name']; ?></div>

    <div class="price">Rs<?= $fetch_cart['price']; ?>/-</div>

    <input type="hidden" name="cart_id" value="<?= $fetch_cart['id']; ?>">

<div class="flex-btn">
<input type="number" min="1" value="<?=$fetch_cart['quantity'];?>" class="qty" name="p
_qty">

    <input type="submit" value="update" name="update_qty" class="option-btn">

</div>

<div class="sub-total">sub total:
    <span>Rs<?= $sub_total = ($fetch_cart['price'] * $fetch_cart['quantity']); ?>/-</span>
</div>

</form>

<?php

    $grand_total += $sub_total;

    }

    }
else
    {

        echo '<p class="empty">your cart is empty</p>';

    }

?>

```

</div>

<div class="cart-total">

<p>grand total : Rs<?=\$grand_total; ?>/-</p>

continue shopping

<a href="cart.php?delete_all" class="delete-btn<?=(\$grand_total>1)?":'disabled'; ?>">delete all

<a href="type.php" class="btn <?=(\$grand_total > 1)?":'disabled'; ?>">proceed to checkout

</div>

</section>

<?php include 'footer.php'; ?>

<script src="js/script.js"></script>

</body>

</html>

DATABASE TABLE

Database Table

Cart Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(100)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	user_id	int(100)			No	None			Change Drop More
<input type="checkbox"/> 3	pid	int(100)			No	None			Change Drop More
<input type="checkbox"/> 4	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	price	int(100)			No	None			Change Drop More
<input type="checkbox"/> 6	quantity	int(100)			No	None			Change Drop More
<input type="checkbox"/> 7	image	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More














Message Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(100)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	user_id	int(100)			No	None			Change Drop More
<input type="checkbox"/> 3	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	number	varchar(12)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 6	message	varchar(500)	utf8mb4_general_ci		No	None			Change Drop More














Order table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(100)			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	int(100)			No	None			Change Drop More
3	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	number	varchar(12)	utf8mb4_general_ci		No	None			Change Drop More
5	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	method	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
7	address	varchar(500)	utf8mb4_general_ci		No	None			Change Drop More
8	total_products	varchar(1000)	utf8mb4_general_ci		No	None			Change Drop More
9	total_price	int(100)			No	None			Change Drop More
10	placed_on	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
11	payment_status	varchar(20)	utf8mb4_general_ci		No	pending			Change Drop More














Product Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 	int(100)			No	None		AUTO_INCREMENT	 Change  Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More
3	category	varchar(20)	utf8mb4_general_ci		No	None			 Change  Drop More
4	details	varchar(500)	utf8mb4_general_ci		No	None			 Change  Drop More
5	price	int(100)			No	None			 Change  Drop More
6	image	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More

Users Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 	int(100)			No	None		AUTO_INCREMENT	 Change  Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More
3	email	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More
4	password	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More
5	user_type	varchar(20)	utf8mb4_general_ci		No	user			 Change  Drop More
6	image	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More

Wishlist Table

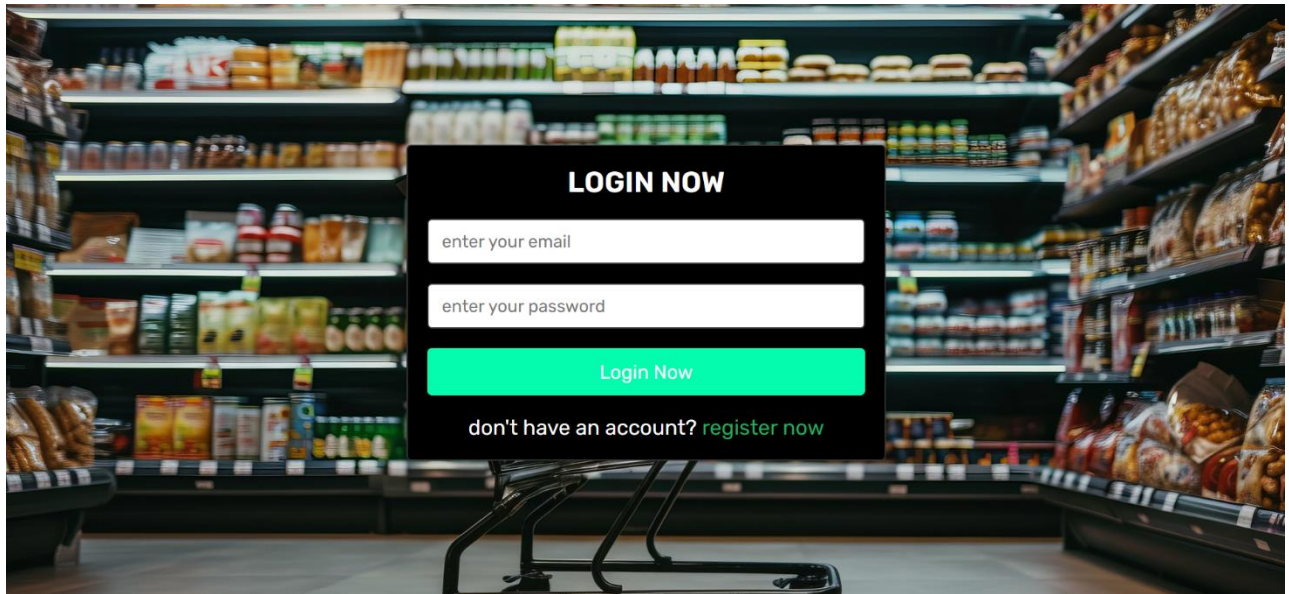
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 	int(100)			No	None		AUTO_INCREMENT	 Change  Drop More
2	user_id	int(100)			No	None			 Change  Drop More
3	pid	int(100)			No	None			 Change  Drop More
4	name	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More
5	price	int(100)			No	None			 Change  Drop More
6	image	varchar(100)	utf8mb4_general_ci		No	None			 Change  Drop More

SCREEN SHOTS

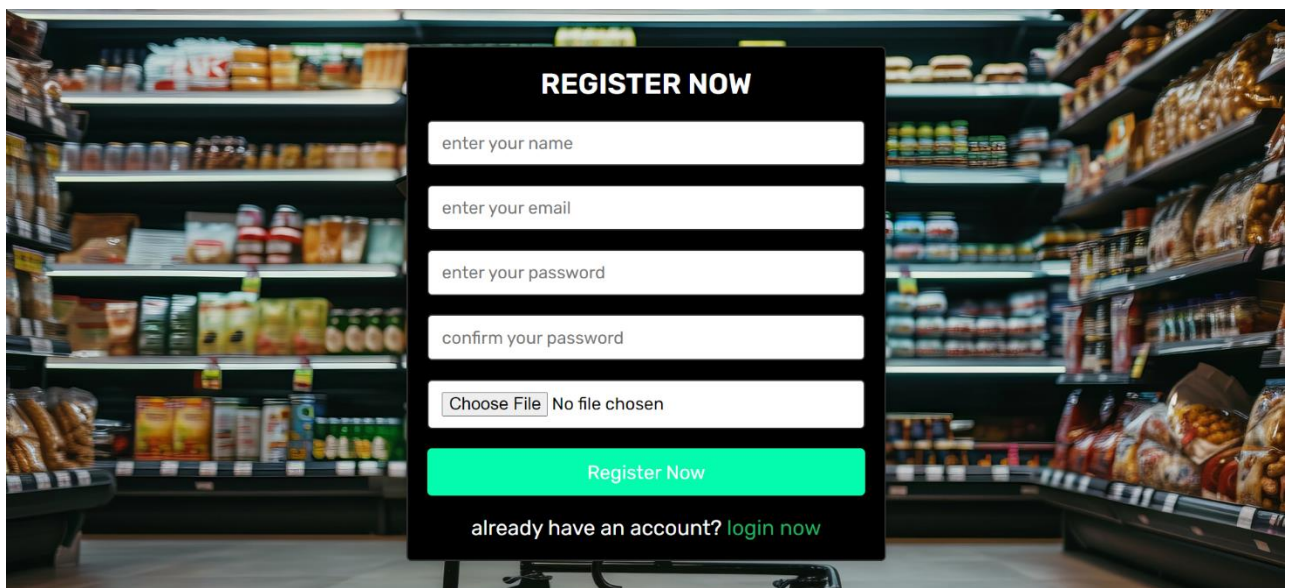
SCREEN SHOTS

Input Design

Login







Register




Update Profile

SRI RAGHAVENDRA STORES.

[home](#) [shop](#) [orders](#) [about](#) [contact](#)

   (0)  (0)

UPDATE PROFILE



username :

old password :

email :

new password :

update pic :

No file chosen

confirm password :





Update Profile

Go Back

Contact Us

SRI RAGHAVENDRA STORES.

[home](#) [shop](#) [orders](#) [about](#) [contact](#)

   (0)  (0)

GET IN TOUCH

Send Message

Add Products

The screenshot shows the 'AdminPanel' interface with a navigation bar containing 'home', 'products', 'orders', 'users', and 'queries'. The main heading is 'ADD NEW PRODUCT'. The form includes fields for 'enter product name', 'enter product price', 'select category' (a dropdown menu), and 'enter product details' (a text area). There is a 'Choose File' button next to the category dropdown, which shows 'No file chosen'. A green 'Add Product' button is at the bottom of the form. Below the form, the text 'PRODUCTS ADDED' is displayed.

Add to Cart

The screenshot shows the 'SRI RAGHAVENDRA STORES' website with a navigation bar containing 'home', 'shop', 'orders', 'about', and 'contact'. The main heading is 'PRODUCTS ADDED'. A message box states 'Your Cart Is Empty'. Below this, a box shows the 'grand total : Rs.0/-' and three buttons: 'Continue Shopping' (orange), 'Delete All' (red), and 'Proceed To Checkout' (green).

QUICK LINKS

- > [home](#)
- > [shop](#)
- > [about](#)

EXTRA LINKS

- > [cart](#)
- > [wishlist](#)
- > [login](#)





CONTACT INFO

- +91 7019869357
- +91 9632820209
- raghavendrastores@gmail.com

Place Order

SRI RAGHAVENDRA STORES

[home](#) [shop](#) [orders](#) [about](#) [contact](#)

   (0)  (2)

PLACE YOUR ORDER

your name :

enter your name

your number :

xxx-xxx-xxxx

your email :

example@gmail.com

payment method: Cash/Upi On delivery

Cash/Upi On Delivery

address line 01 :

e.g. flat number

landmark :

e.g. landmark

city :

e.g. Bengaluru

state :

e.g. Karnataka

country :

e.g. India

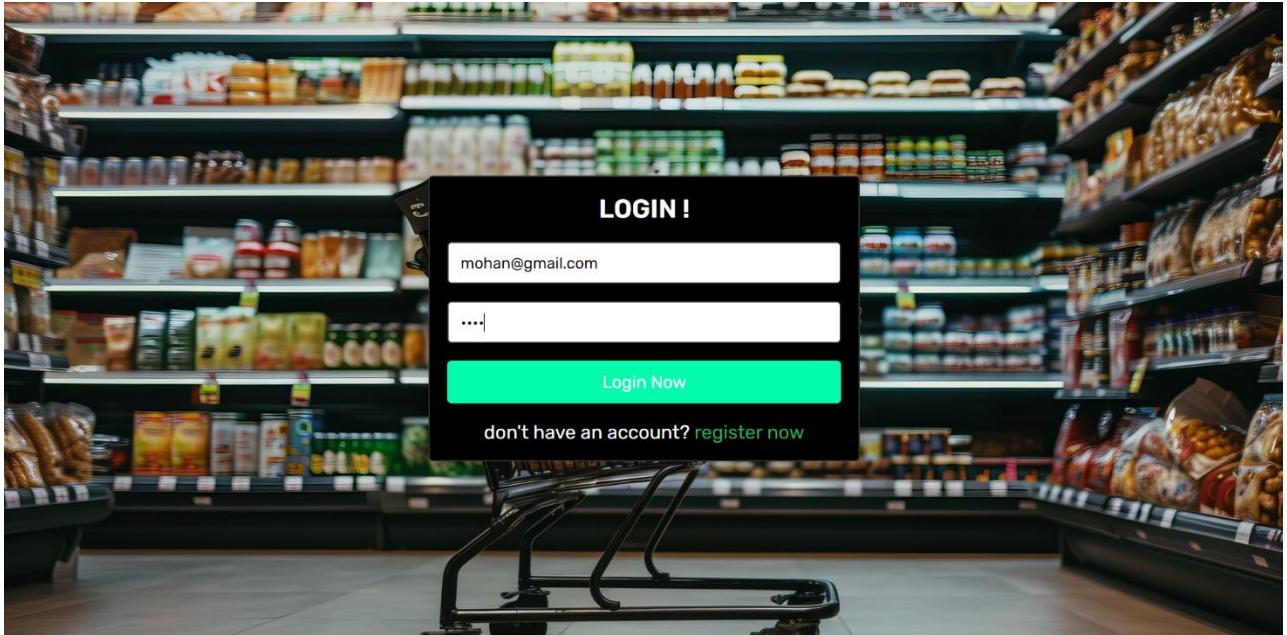
pin code :

e.g. 123456

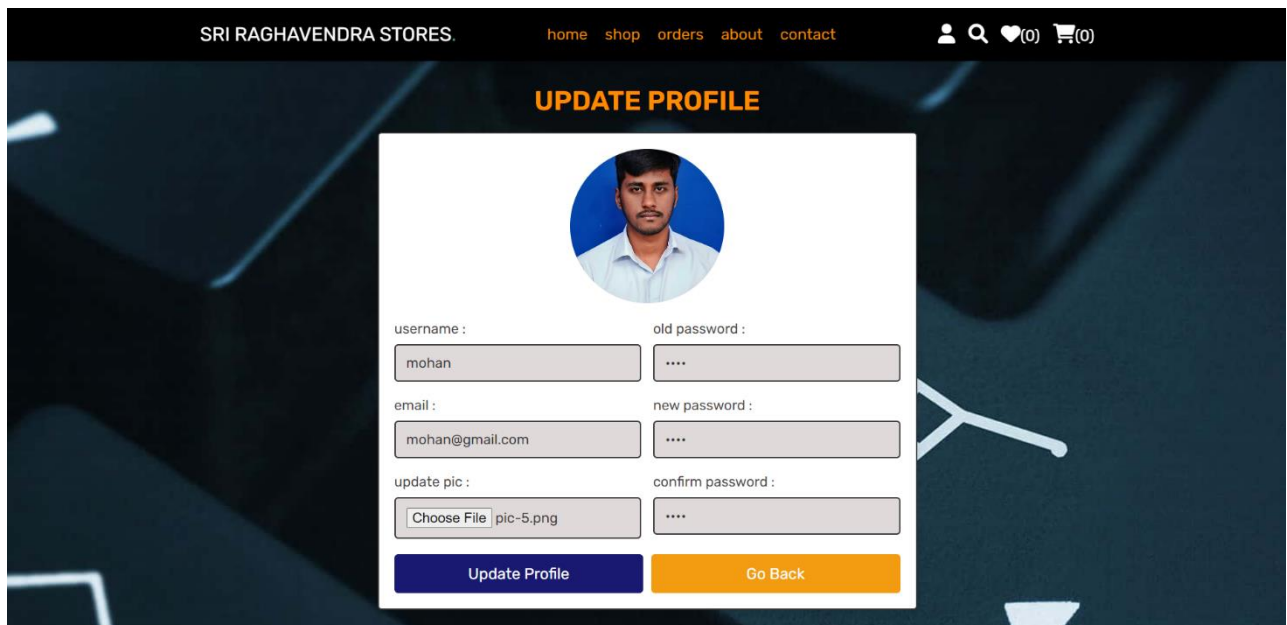
Place Order

Output Design

Login Successful



Profile Updated Successfully



Message Sent Successfully

SRI RAGHAVENDRA STORES. home shop orders about contact

GET IN TOUCH

Mohan S

mohan@gmail.com

7019869357

Products in your shop is good quality ones.
Requesting to add more items to purchase.

Send Message

Products Added Successfully

AdminPanel home products orders users queries

ADD NEW PRODUCT

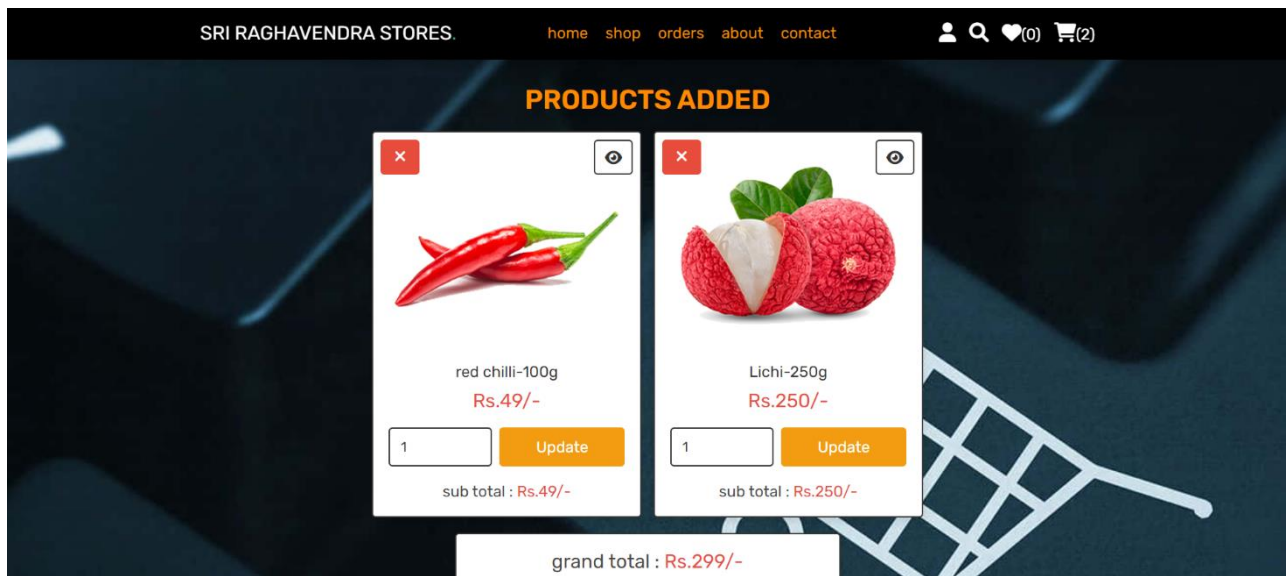
Red chilly-100g 49

vegetables Choose File red chilly.png

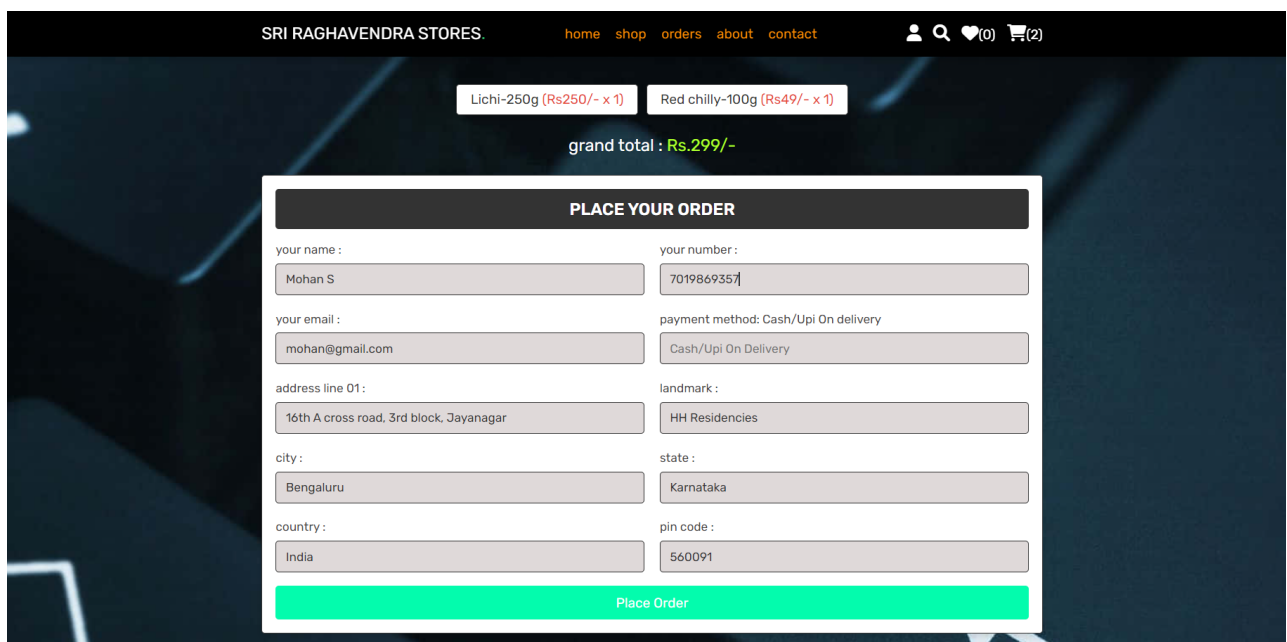
Kashmiri red chilly 49rs/100g

Add Product

Products Added To Cart Successfully



Order Placed Successfully




Index Page

SRI RAGHAVENDRA STORES


REACH FOR A HEALTHIER YOU WITH ORGANIC FOODS

Whoever said money can't buy happiness didn't know where to shop.

Login Now




SHOP BY CATEGORY




VEGETABLES

it is usually herbaceous plant grown for an edible part that is




FRUITS

the sweet and fleshy product of a tree or other plant that



SNACKS

a small portion of food that is smaller than a regular meal.



OTHER GROCERY

a grocer is a shopkeeper who sells items such as grocery

Home Page

SRI RAGHAVENDRA STORES.

home shop orders about contact


👤 🔍 ❤️(0) 🛒(0)

SRI RAGHAVENDRA STORES





REACH FOR A HEALTHIER YOU WITH ORGANIC FOODS

Whoever said money can't buy happiness didn't know where to shop.

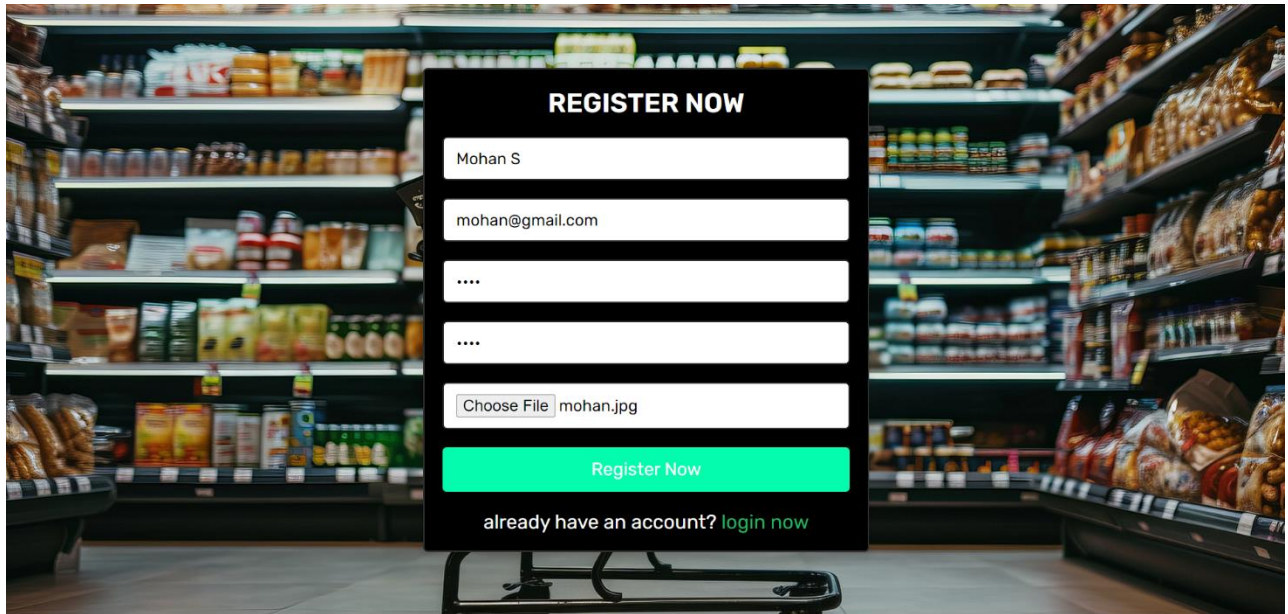
About Us



SHOP BY CATEGORY

Register Page



A screenshot of a 'REGISTER NOW' form overlay on a background image of a grocery store aisle. The form is black with white text and input fields. It includes fields for a name, email, two masked password fields, and a file upload field. A green 'Register Now' button and a link to 'login now' are at the bottom.

REGISTER NOW

Mohan S

mohan@gmail.com

....

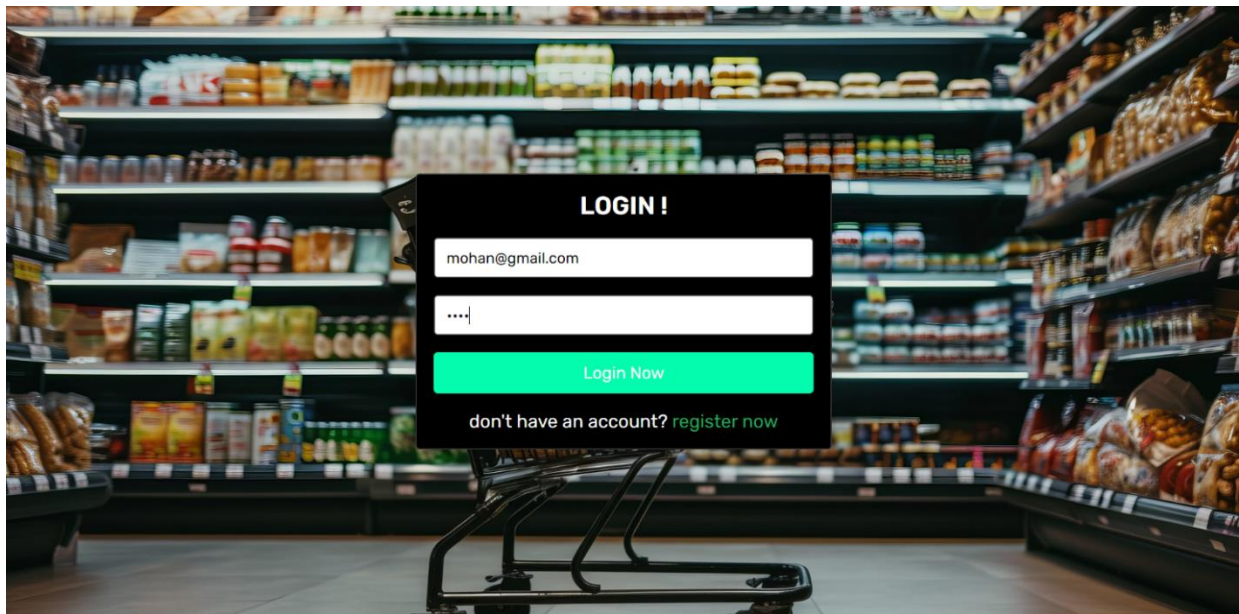
....

Choose File | mohan.jpg

Register Now

already have an account? [login now](#)

Login page



A screenshot of a 'LOGIN !' form overlay on a background image of a grocery store aisle. The form is black with white text and input fields. It includes fields for an email and a masked password. A green 'Login Now' button and a link to 'register now' are at the bottom.

LOGIN !

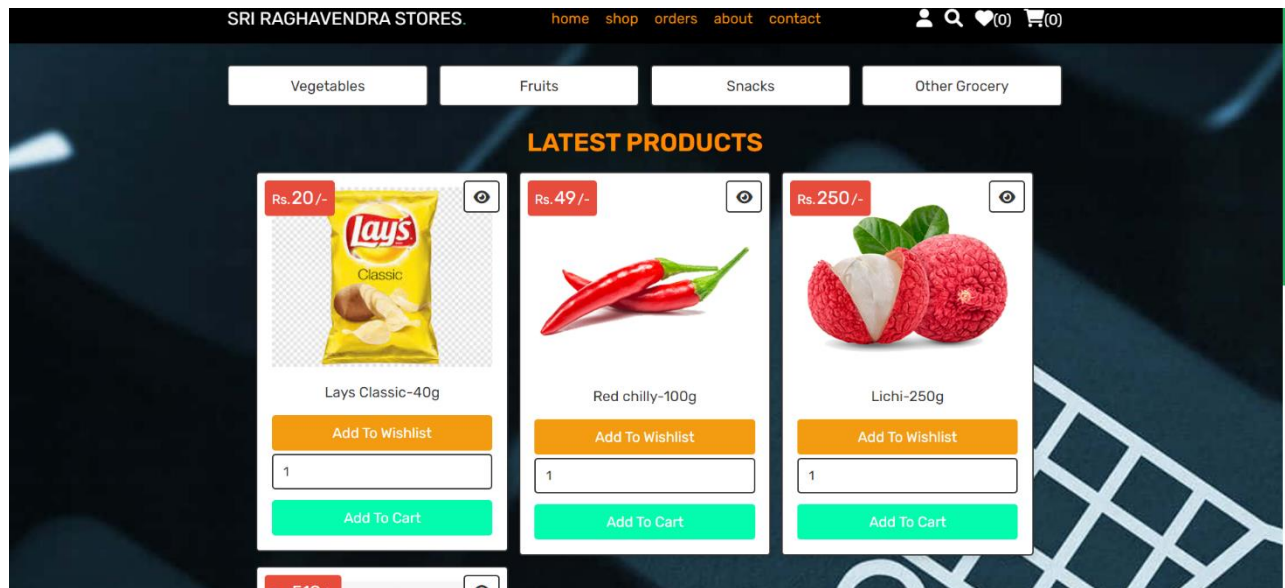
mohan@gmail.com

....

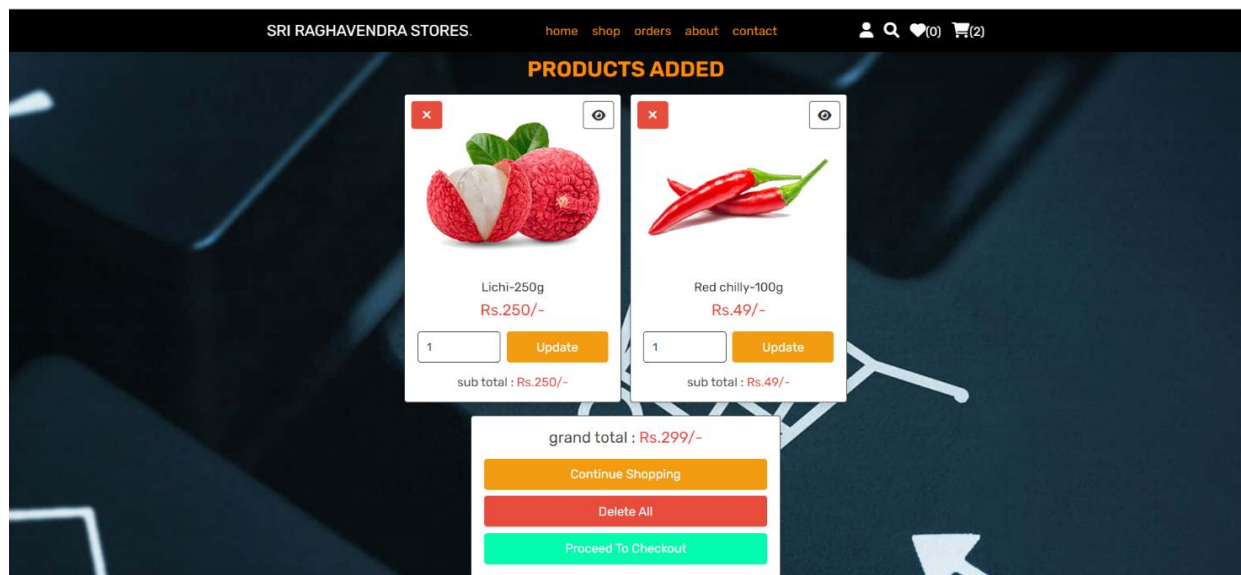
Login Now

don't have an account? [register now](#)

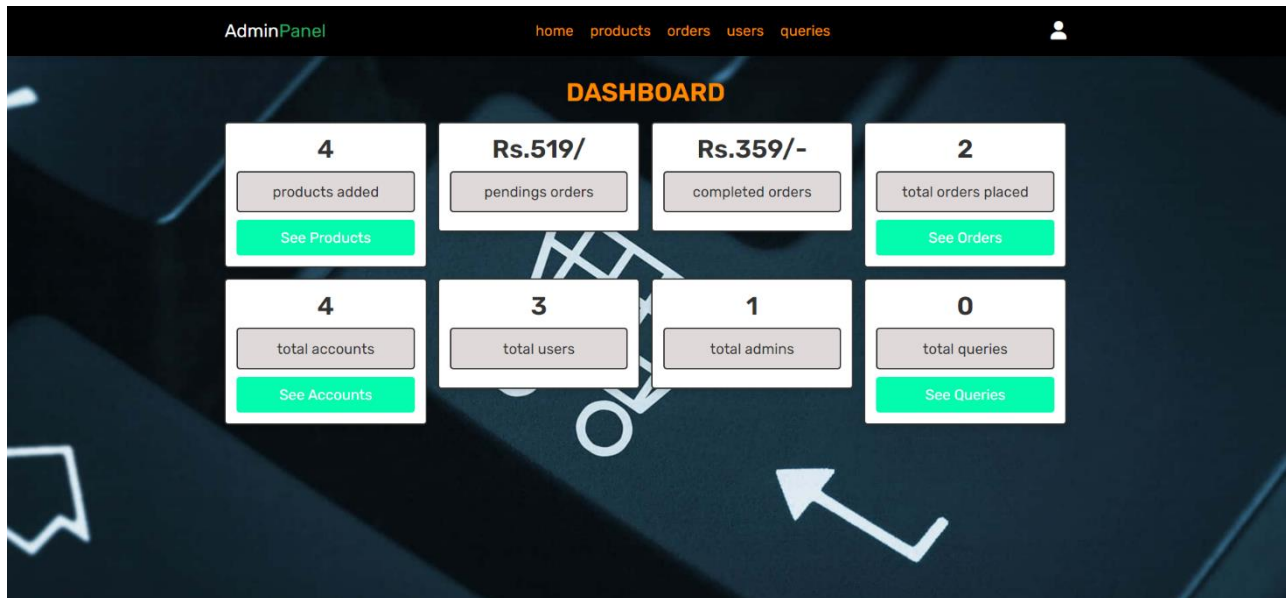
Shop Page



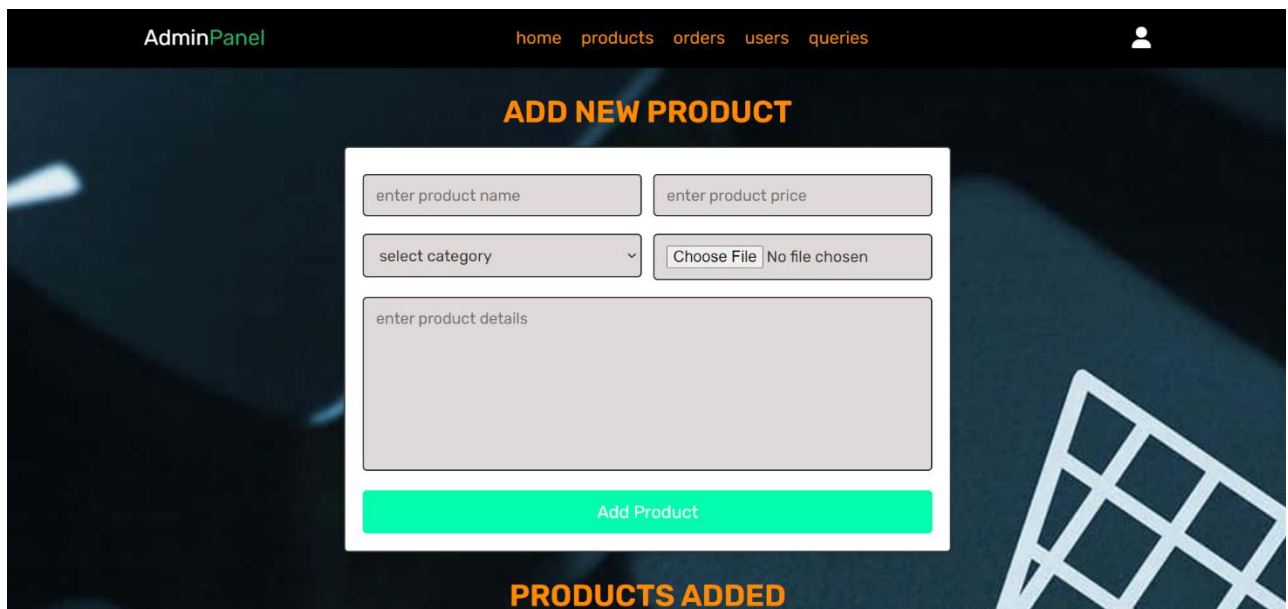
Cart page



Admin Page



Add Product Page



The screenshot shows the 'ADD NEW PRODUCT' form in the Admin Dashboard. The form is centered on a dark blue background with a grid pattern. It includes input fields for product name, price, category, and details, along with a file upload button. A green 'Add Product' button is at the bottom of the form. Below the form, the text 'PRODUCTS ADDED' is visible.

ADD NEW PRODUCT

enter product name enter product price

select category Choose File No file chosen

enter product details

Add Product

PRODUCTS ADDED

TESTING

TESTING

System Testing

The code is tested at various levels in software testing. Unit, system and user acceptance testing's are often performed. This is a grey area as many different opinions exist as to what the stages of testing are and how much if any iteration occurs. Iteration is not generally part of the waterfall model, but usually some occur at this stage.

Software testing is the execution of program to find its faults. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested and on the functional externals, that is conducting test to uncover errors and ensure that defined inputs will produce actual results agreed with required results. The following test strategies were adopted to test the system.

Testing Objectives

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an undiscovered error Testing Principles
- All tests should be traceable to end user requirements.
- Tests should be planned long before testing begins.
- Testing should begin on a small scale and progress towards testing in large.
- Exhaustive testing is not possible.
- To be most effective testing should be conducted by a independent third party.

Test Approaches

i. Black Box Testing

This method focuses on the functional requirements of the software. This testing enables to derive set input conditions that will fully exercise all functional requirements of the program.

Black Box Testing attempts to find errors in the following category.

- Incorrect or missing functions.
- Interface errors.
- Performance errors.
- Initialization and Termination errors.

ii. White Box Testing

This is performed early in the testing process, while Black Box testing is applied during the last stage of testing. In this test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been used to generate the test case in the following test cases:

- Guarantee that all independent paths have been executed.
 - Execute all logical decisions from their True and False side.
 - Execute all loops at their boundaries and within their operational bounds.
- Testing strategies

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

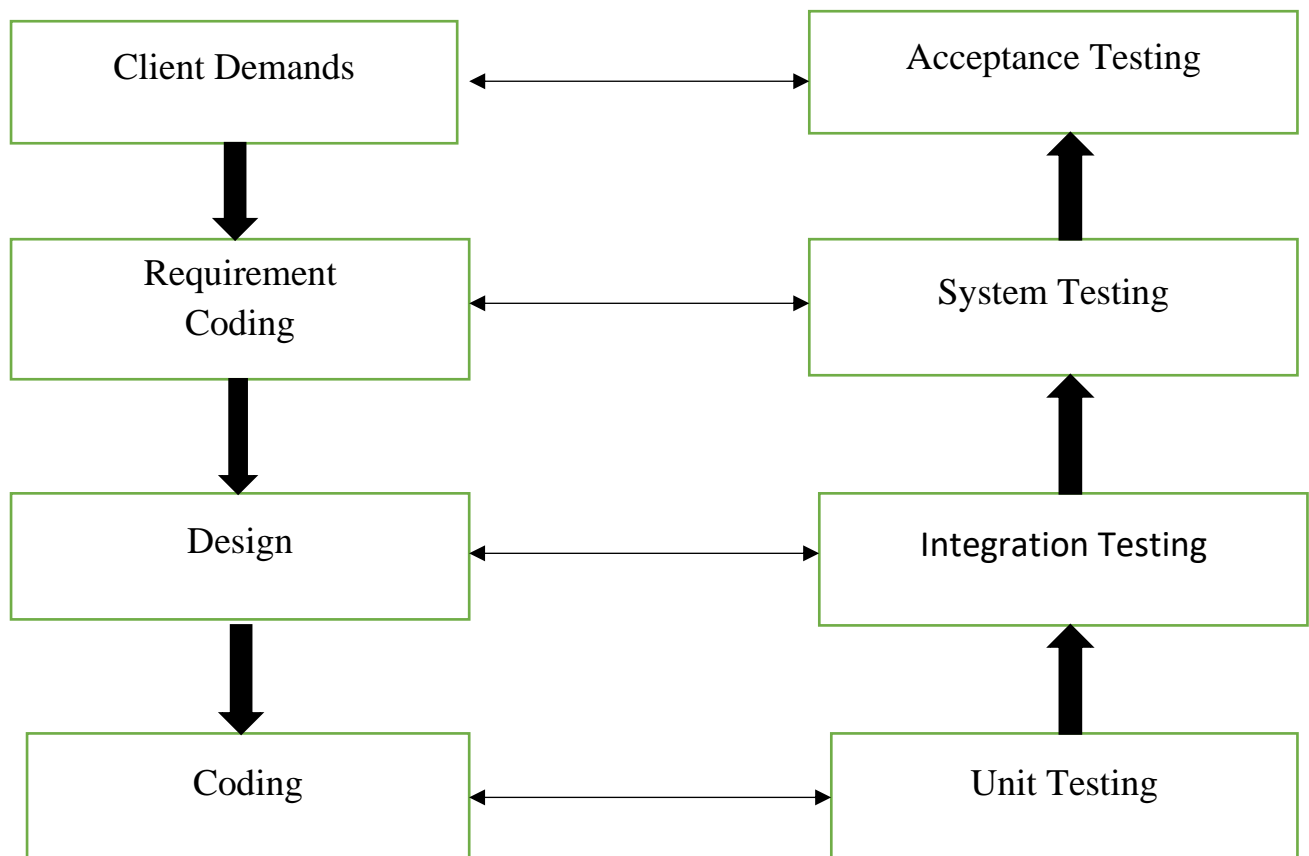
There are two general strategies for testing software. They are as follows:

Code Testing: This examines the logic of the program. To follow this test, cases are developed such that every path of program is tested.

Specification Testing: Specification Testing examines the specification, starting what the program should do and how it should perform under various conditions. Then test cases are developed for each condition and combinations of conditions and to be submitted for processing

Levels of Testing

In order to uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are



The stages of Testing Process are Unit Testing: Individual components are tested to ensure that they operate correctly. Each component tested independently without other system components.Ex. Check for Username and Password with the table, after the next module is loaded.

Integration Testing: Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. This testing is done using the bottom-up approach to integrate the software components of the software system in to functioning whole.

System Testing: System testing is actually a series of different tests whose primary purpose is fully to exercise the computer-based system. The system tests that where applied are recovery testing and performance testing. Finally, a review or audit is conducted which is a final evaluation that occurs only after operating the system long enough for user to have gained a familiarity with it. System testing was done by the inspection team to verify that all the functionality identified is the software requirement specification has been implemented. Defects that crept in the system has been found defect free and is working well. System testing is concerned with interfaces, design logic, control flow recovery, procedures throughput, capacity and timing characteristics of the entire system. For blank field, alphabets, number and special character validation.

Acceptance Testing: User acceptance of the system is the key factor for the success of any system. This is done by user. The system is given to the user and they test it with live data. Acceptance testing involves the planning and execution of functional test. Performance tests, stress tests in order to demonstrate that the implemented system satisfies its requirements. Two sets of acceptance test can be run, those developed by the customer. The system has been tested for its performance at unit level by the individuals through performance testing that is designed to test the run time performance of the software. The performance of the fully integrated system is tested and was found good.

Test Case:

SL NO	Input	Expected O/P	Actual O/P	Result
1	Valid Username and Password	It should display respective page according to user type.	Respective Home is displayed .	Passed
2	Invalid Username and Password	It should give appropriate error message saying “Enter proper User-Id and Password”	Error message is displayed	Passed
3	Add/Update/Delete Product details	Add/Update/delete action is taken.	Add/Update/delete done successfully	Passed
4	User enters valid Username and password.	Respective Home is displayed	Respective Home is displayed .	Passed
5	Change Password	Respective Home is displayed	Respective Home is displayed	Passed
6	Invalid card number	It should give Appropriate error message	Error Message displayed	Passed
7	Valid Email id	Respective Page is displayed	Respective Page is displayed	Passed

Implementation

The Online Grocery Delivery System is a digital platform designed to facilitate the seamless ordering and delivery of groceries to customers' doorsteps. The system comprises a user-friendly website and/or mobile application that allows customers to browse through a wide range of grocery products, select items based on their preferences, and add them to their virtual shopping carts. Customers can create accounts, manage their profiles, and save delivery addresses for convenience.

Upon finalizing their shopping, customers proceed to the checkout process where they can review their order, apply any applicable discounts or promotional codes, and choose a preferred delivery time slot. The system also offers various secure payment options, including credit/debit cards, digital wallets, and online banking. Behind the scenes, the system integrates with a comprehensive inventory management system that ensures real-time tracking of product availability. This helps prevent customers from ordering items that are out of stock. Additionally, the system may employ recommendation algorithms to suggest related or frequently purchased items, enhancing the shopping experience.

Once an order is confirmed, the system notifies the nearest fulfillment center or partnering grocery store. Staff at the fulfillment center then pick up the order and prepare it for delivery. Delivery personnel, either employed by the platform or contracted, receive the order details and the customer's address. They use navigation tools to optimize their routes and ensure timely deliveries.

Customers can track the status of their orders, including real-time updates on when the delivery is in route and when it's expected to arrive. They may also have the option to communicate with the delivery personnel directly through the app for any special instructions or changes to the delivery.

CONCLUSION

CONCLUSION

Conclusion

In conclusion, the online grocery delivery system represents a transformative solution that has redefined the traditional grocery shopping experience. By seamlessly integrating cutting-edge technologies and user-centric features, this system has revolutionized the way consumers access essential goods. The comprehensive product catalog, intuitive user interface, and personalized shopping experience have enhanced convenience, enabling users to effortlessly browse, select, and purchase a wide array of groceries from the comfort of their homes. The real-time inventory management, secure payment processing, and accurate order tracking mechanisms ensure reliability and transparency throughout the entire process. Additionally, the system's commitment to data security, customer support, and sustainable practices underscores its dedication to both user satisfaction and responsible business practices. By delivering products to consumers' homes, the homebound aged, and handicapped can participate in the grocery shopping experience. Even though there has been a great decline in the number of pure-play online stores, there appears to be a solid market for grocery shopping online.

FUTURE ENHANCEMENT

Future Enhancement

The following things can be done in the future.

- The users could subscribe for price alerts which would enable them to receive messages when price for products fall below a particular level.
- Users can have multiple shipping and billing information saved. During checkout, they can use the drag and drop feature to select shipping and billing information.
- The current system can be extended to add new category for products.
- Provide real-time inventory information to customers, so they know if a particular item is in stock or not. This can help reduce frustration and increase transparency.

BIBLIOGRAPHY

BIBLIOGRAPHY

Books

- ☐ “Software Engineering”, by Ian Somerville, Sixth Edition, Pearson Education Ltd 2007.
- ☐ “Web Programming”, by ‘Chris Bates’ Wiley Dream tech India, 2nd Edition.
- ☐ Database Management Systems, by Navathe.
- ☐ Programming THE WORLD WIDE WEB – Robert W Sabesta

References

- ☐ <http://www.php.net.com> for **Php**.
- ☐ <http://w3schools.com> for **Form Designing**
- ☐ <http://en.wikipedia.org/wiki/PHP> for **Php**
- ☐ <http://www.mysql.com/click.php?e=35050> for **MySQL**

- font awesome cdn link: <https://cdnjs.com/libraries/font-awesome>
- clip path generator: <https://bennettfeely.com/clippy/>
- google fonts: <https://fonts.google.com/specimen/Pop...>
- Editor : - **visual studio code** with Laetus: **Dark Vibrant Theme**
- Browser : - **google chrome**
- UI Tool : - Figma
- Images / Video / SVG : - 01 - <https://www.freepik.com/>
- 02 - <https://storyset.com/>
- 03 - <https://undraw.co/>
- 04 - <https://pixabay.com/>
- 05 - <https://unsplash.com/>
- 06 - <https://pixabay.com/>
- 07 - <https://www.flaticon.com/>