Restaurant Management System

Brikal & Harsh

Guided by prof. Vasant Urvi

Project Abstract

We (Brikal & Harsh) are the students of BCA (Sem -5). We have undertaken this project. The main purpose of arranging such training is to get some practical knowledge & experience about the various aspects of management because the theoretical knowledge is not enough as there is wide gap between theoretical knowledge and practical knowledge.

This report documents the process of designing, developing, and testing a software system to be used in a restaurant; usually given the name **R**estaurant **M**anagement **S**ystem. The restaurant management system is there to help communication between all teams within a restaurant by minimising the probability of human errors. This report was created by Brikal Kunjadiya and Harsh Parmar as part of 3rd year project and was published on Sep 10, 2023.

We have put on our effort and time to create this report as best as possible. But then also being a student, there may be possibilities for errors and mistakes. We concerned authority will kindly accept this report.

Acknowledgment

We would like to thank our project guide, Prof. Vasant Urvi, for providing us guidance and Shree G.K. & C.K. Bosamia arts and commerce collage, jetpur to provide us an opportunity of minor project in third year of BCA(Sem-5).

In addition, we would like to thank our family for the support throughout our final year at collage, and for checking over our project & report.

Contents

Chapter 1

Introduction

1.1 Chapter Overview

This chapter gives basic information of project and introduction to the project by defining the problems encountered by restaurants, the main objectives that the system expects to achieve and a brief introduction to existing solutions.

1.2 Basic Information

Project title	-	RMS (Restaurant Management System)
Platform	-	Server Side: PHP
		Client Side: JAVA SCRIPT
Front End	-	HTML, CSS, JAVA SCRIPT
Back End	-	PHP, MYSQL

1.3 The Problem

The problem for many businesses is to ensure that they not only attract new customers but to ensure they maintain their existing customers. It has been argued many times that an existing customer is worth more to a business than a new customer. As the cost to attract a new customer can be up to five times the cost to retain an old customer.

Within the restaurant sector, a customer is likely to return to the restaurant in the future if they received an excellent customer service as well as appetising food. However, if they had to wait for an unreasonable amount of time or there was a mistake in the order, it is very unlikely the customer would return. Therefore, a solution to this problem would be to minimise mistakes within the order and bill, and help eradicate delays as well as encouraging team work and communication within the team.

1.4 Project Objective

The objective of this project is to build an electronic restaurant management system using all the skills and techniques from the field ensuring that no common development mistakes are reproduced. Project management is critical to all software engineering projects and keeping to a project plan will be of similar importance. One of the main objectives of any business is to maximize profit by increasing efficiency and decreasing overheads without compromising customer satisfaction. Currently, many restaurants use a paper-based system to communicate between the restaurant and kitchen which can be shown to be one of the least efficient approaches. Even though this approach is implemented in successful profitable restaurants, there are several problems which could be seen as reducing the restaurant's efficiency:

- Miscommunication caused by handwriting.
- Unmanageable order logging.

- Inefficient restaurant-kitchen communication.
- Difficult order tracking and time management.
- Difficult stock management.
- Limited statistical output.

By introducing an electronic restaurant management system these problems can be avoided or improved leading to an increase in profits.

1.5 Existing Solutions

There are many computerised restaurant management systems available but for each system there exist disadvantages or missing features. The most common type of restaurant management system contains a static order entry computer system usually in the shape of a desktop computer with a touch screen. Typically, this common approach is adequate to the restaurants requirements but still requires handwritten orders to be relayed to the order entry computer system.

A slightly different approach was implemented in restaurants in US and their franchisees all over world named **McDonald's** restaurant. The restaurant utilises a roller coaster approach to serving the food and an order entry system fully operated by the customer. As reviewed, there is no need for any waiters as the customers use touch-screen monitors to browse the menu. This invention can save on operating costs, but the initial injection of cash required is substantial as every table requires the necessary hardware. They also take orders by themselves if any customer is uncomfortable to do manually in that case customer only have to speak the menu items and everything will be inputted by there employees. This can also increase efficiency and save a lot of time and a bit manpower.

1.6 Project Proposal

The aim of this project is to create a restaurant management system that can incorporate the benefits of all the existing solutions but without any of the drawbacks as well as including many new features.

Many of the existing solutions to POS (Point-of-Sale) systems are sold with the required expensive hardware so for any business looking to work to a budget, the more enriched software solutions are just out of their range.

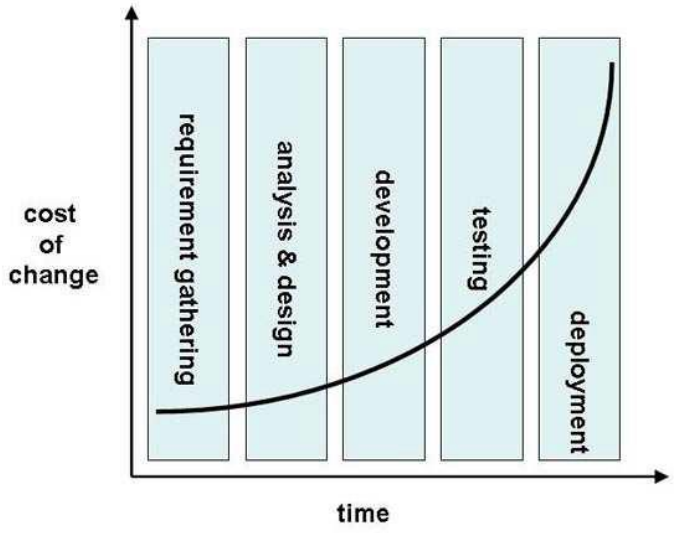
Here, below a table showing the proposed features of the system and the motivation behind the features.

Feature	Motivation
Easy Inventory management	One can easily add items, update items, remove items. so only the meals with enough ingredient stock can be sold.
Meal option and preference selection	Flexible meal options available for customers
Automated billing with GST	When user add items, the bill will automatically be generated which also includes GST, Totals everything needed.
Simple and good UI design	The software's design is very simple where user can easily add items and order food items and admin can easily manage inventory.

Chapter 2

Requirement Gathering

Short Intro: Requirement gathering is a very important step in software engineering. According to an article written by Craig Murphy, Boehm's cost model discusses how discovering errors at an early stage of software development can reduce overall costs. Figure below is an example of Boehm's cost model and uses the waterfall technique to give a visual insight into how important gathering accurate requirements can be.



2.1 Chapter Overview

This chapter clears that from where the inspiration came to build RMS. Also includes the technical requirement of the project to build and add features efficiently. We have already discussed the features that should be in the project in 1.6.

2.2 Project Inspiration

We have visited some restaurants and some restaurants are using the software to make their service more efficient by taking order from customers through software in simply add items to cart and process order. Some restaurants were using software to manage their inventory and menu items.

2.2.1 Harbhole Dosa:

A local restaurant name Harbhole Dosa they only make dosas they are very famous locally in our city. They are using this approach to make their service fast and efficient. We have visited their restaurant they take orders verbally from customers and their employee input the food items as per the order and serve the food they also manage their inventory through their software.

2.2.2 McDonald's:

It is a world-famous food chain which probably everyone knows by there burgers and other fast foods as well. They also have used the same approach as well and they are our big inspiration behind this project.

They use software two ways one is customers can order by themselves and pay in there as well. And second is they provide customers to only speak their food items and input should be given by their employees.

2.3 Technical Requirements

To make RMS system efficient and featured we need tools and technologies for hardware and software.

Hardware Requirement

- Monitor: 1
- Keyboard: 1
- Mouse: 1
- Hard drive: 512 GB

Software Requirement

- Operating system: Windows 7 and +
- Database: MYSQL
- Local Server: Wamp & Xampp
- IDE: Visual Studio Code
- Front-end Tech: Html, CSS, JAVA SCRIPT
- Back-end Tech: PHP, MYSQL

Chapter 3

Requirement Analysis

3.1 Chapter Overview

This chapter will look at the system and then discuss the required features. Also see the advantages and disadvantages of our project.

3.2 Required Features

An important part of requirements gathering is the production of a list of system features that categorises on priority. Here below is a table showing the proposed system features and allocated priorities. The priority value is in the range from 1 to 3 where 1 is high priority and 3 is low priority.

Required Feature	Priority
Minimum click touch-screen GUI design for efficient ordering.	1
Automatic bill generation with GST included	2
Interface to maintain and manage the menus and associated food items.	1
Stock control for all ingredients; reducing/increasing stock automatically.	2
Flexible food items grid design to fit any screen size.	3
User/admin login functionality.	1
Ability to define meals by images as well as text.	2

3.3 Functional Requirements

A functional requirement is a requirement that, when satisfied, will allow the user to perform some kind of function. The functional requirements of the RMS are below in table format. The priority value is in the range from 1 to 3 where 1 is high priority and 3 is low priority.

Requirements	Priority
Ability to add ingredients, meals, and menus.	1
Ability to edit ingredients, meals, and menus	1
Ability to remove ingredients, meals, and menus	1
Ability to alter the price of ingredients.	3
Ability to change transparency of an image that represents a meal	2
Ability to manage inventory overall	1
Ability to generate bill automatically	1
Ability to add a new order	1
Secured (admin login for inventory management)	1

3.4 Non-Functional Requirements

Non-functional requirements are usually some forms of constraint or restriction that must be considered when designing the solution. The non-functional requirements of the RMS are below in table format. The priority value is in the range from 1 to 3 where 1 is high priority and 3 is low priority.

Requirements	Priority
Ability to interact with the MySQL database.	1
Minimum clicks from the beginning to the end of an order.	2
Only accessible by system admin.	2
Ability to use the application using a touch screen without the need of a keyboard or mouse.	1
Ability to calculate total food items of an order	2

3.5 Project Pros and Cons

Our restaurant management system is a software that helps to manage various aspects of running a restaurant, such as ordering, billing, inventory, reporting, and more. There are many advantages and disadvantages of using a restaurant management system. Here are some which we think are top pros and cons:

3.5.1 Pros of project

- We have used Vanila CSS rather than using any front-end frameworks such as bootstrap.
- Our RMS software can save time for restaurant while taking manual orders.
- RMS can improve accuracy while billing and items total.
- To manage inventory will be simple and easy. Admin can also easily add, edit, or delete items in couple of clicks.
- The RMS is also well secured and can store a data (id – password) of the user safely

3.5.2 Cons of Project

- Admin can add food items under sub category but cannot add new main categories.
- User cannot pay through that software because pay now feature is not available.
- Our software's UI is not fully responsive for mobile devices.

Chapter 4

Feasibility Study

4.1 Chapter Overview

This chapter we have studied for the feasibility of RMS system with project partner and taken the guidance from the professors and from internet. And come up with *Technical, Economic, operational*, and *Schedule feasibility*.

4.2 Technical Feasibility

It is examined to ensure that the required technologies and expertise are available to develop the system.

- **Required Skills: -**

  I. For Frontend → Knowledge and practice of Html, CSS, & JavaScript.

  II. For Backend → Knowledge of PHP & MySQL.

As the upper described technologies are well-known and we have studied in our BCA course it is feasible to confirm and good to go with it.

The integration with database should be seamless and for that MySQL is perfect to store menu items, and other data.

4.3 Economic Feasibility

The economic feasibility assessed whether the benefits of implementing the RMS project justify its costs. The following factors were considered:

4.3.1 Cost: The project involves hardware and software technology as well as skills.

- Keyboard: 300
- Mouse: 300
- Monitor: 10,000
- Hard Disk: 4,000
- RAM: 2,000
- Mother-Board: 2,000

4.3.2 Benefits: The RMS is expected to bring substantial benefits to the restaurant, including improved operational efficiency, accurate billing, better inventory management, and enhanced customer services, lead to increase revenue.

4.4 Operational Feasibility

It examines the impact of the RMS on existing restaurant operations and determines staff to embrace new system. The following factors were considered:

4.4.1 Integration with existing process: The RMS will be designed to align with the restaurant's existing processes, minimising description, and facilitating a seamless transition to the new system.

4.4.2 User Acceptance: Feedback from restaurant staff during the requirement gathering phase indicated willingness to embrace an automated system to streamline operations, indicating strong operational feasibility.

4.5 Social Feasibility

Our project is socially feasible because it does not have any harm or any insecurity on client's system, easy to use anywhere by anyone.

RMS project is safe for each business and free to access. So, it is legal and socially safe.

Chapter 5

Data Information

5.1 Chapter Overview

In this chapter there is the information related to the data. The information which is stored of client or the user of the RMS system. This phase is necessary for the project here the information about the software's data, and how it works has been shown.

5.2 Introduction

In our case of RMS, our goal was to make it as simple as we can but build it unique & by ourselves. So, we build simple but highly effective system. Therefore, we have minimal data information.

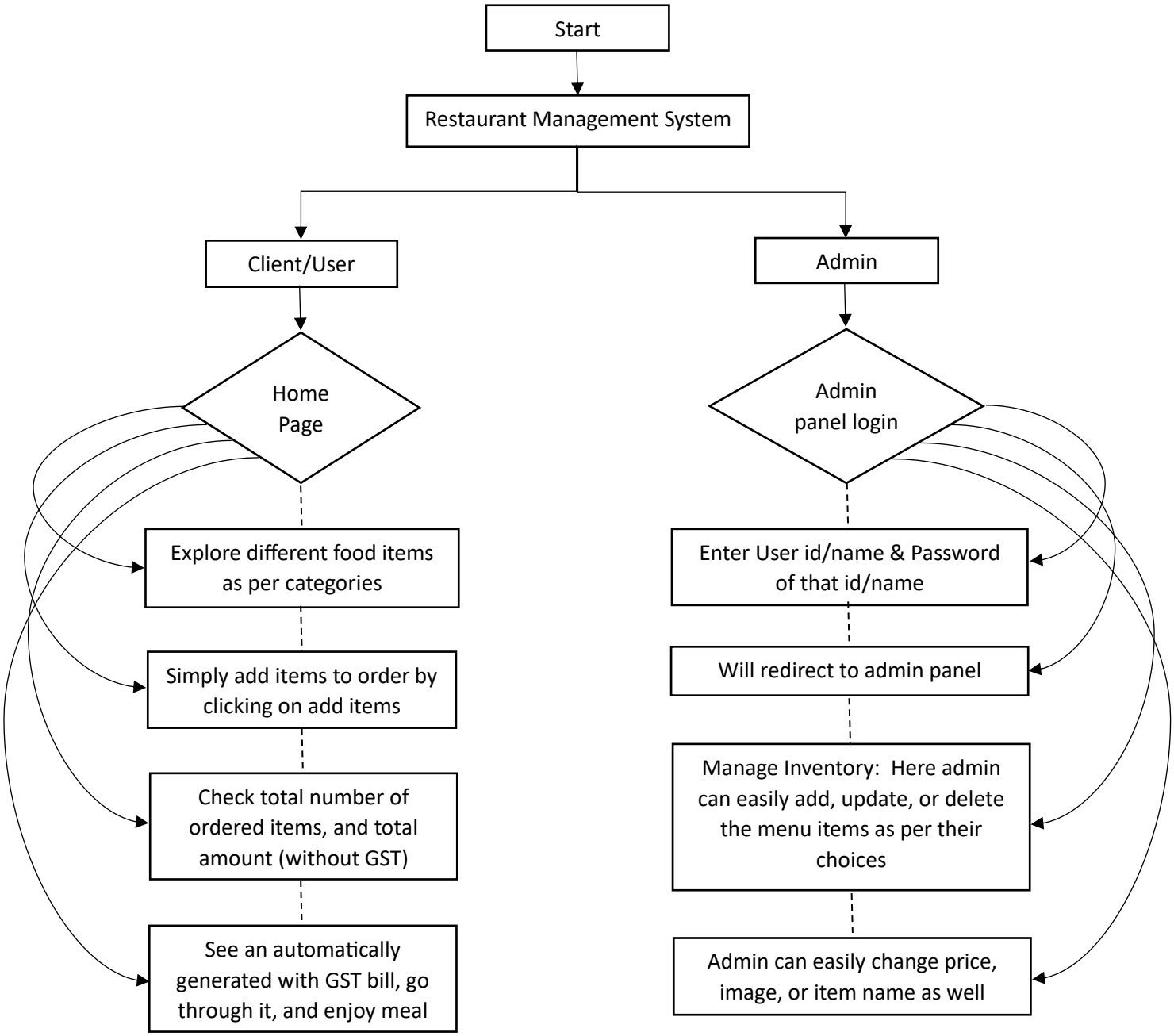
5.3 Data Overview

We have in total two different databases:

- **rms**
  - The rms database stores the data of restaurants food items. So, we have named that table as:
    - **food:** which stores the food item's image, item's price, item's name and so on.
- **user**
  - The user database stores the data of the admin or the owner of the system. So, we have named that table as:
    - **user:** which stores the data of user's user name/id and its password.

The reason behind two different databases is to have a clarity. And by chance if the data of the items or foods has been crashed or anything unwanted happened the user's data (user id & password) will be secured.

5.4 Flow Chart



Chapter 6

Data Structure

6.1 Chapter Overview

In this chapter you will see the overall structure of the data we have listed our databases and their tables where the information has been and will stored.

6.2 Databases and Tables

- Database: rms
  - Table: **food**

#	Name	Type	Collation	Null	Default	Extra
1	srno	int(3)	-	No	*None*	AUTO_INCREMENT
2	name	varchar(30)	utf8mb4_general_ci	No	None	-
3	image	varchar(100)	utf8mb4_general_ci	No	None	-
4	price	int(4)	-	No	None	-
5	ctgr	varchar(30)	utf8mb4_general_ci	No	None	-

- Database: user
  - Table: **user**

#	Name	Type	Collation	Null	Default	Extra
1	srno	int(3)	-	No	None	AUTO_INCREMENT
2	user	varchar(50)	utf8mb4_general_ci	No	None	-
3	pass	varchar(50)	utf8mb4_general_ci	No	None	-

Chapter 7

Data Flow Diagram: Admin & Client

7.1 Chapter Overview

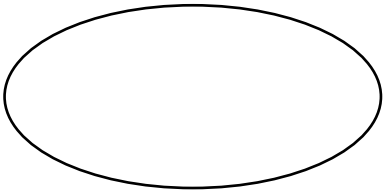
In this chapter we have reported our project's/system's flow of data through a graphical diagram. We have divided data flows into levels for better understanding and clarity. Let us see it.

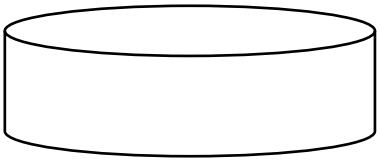
7.2 Introduction

The DFD is used to clear the flow of the data of the software or the system. Data flow diagram is divided into four main elements or shapes. And each shape has different meaning of representing. Let us look that.

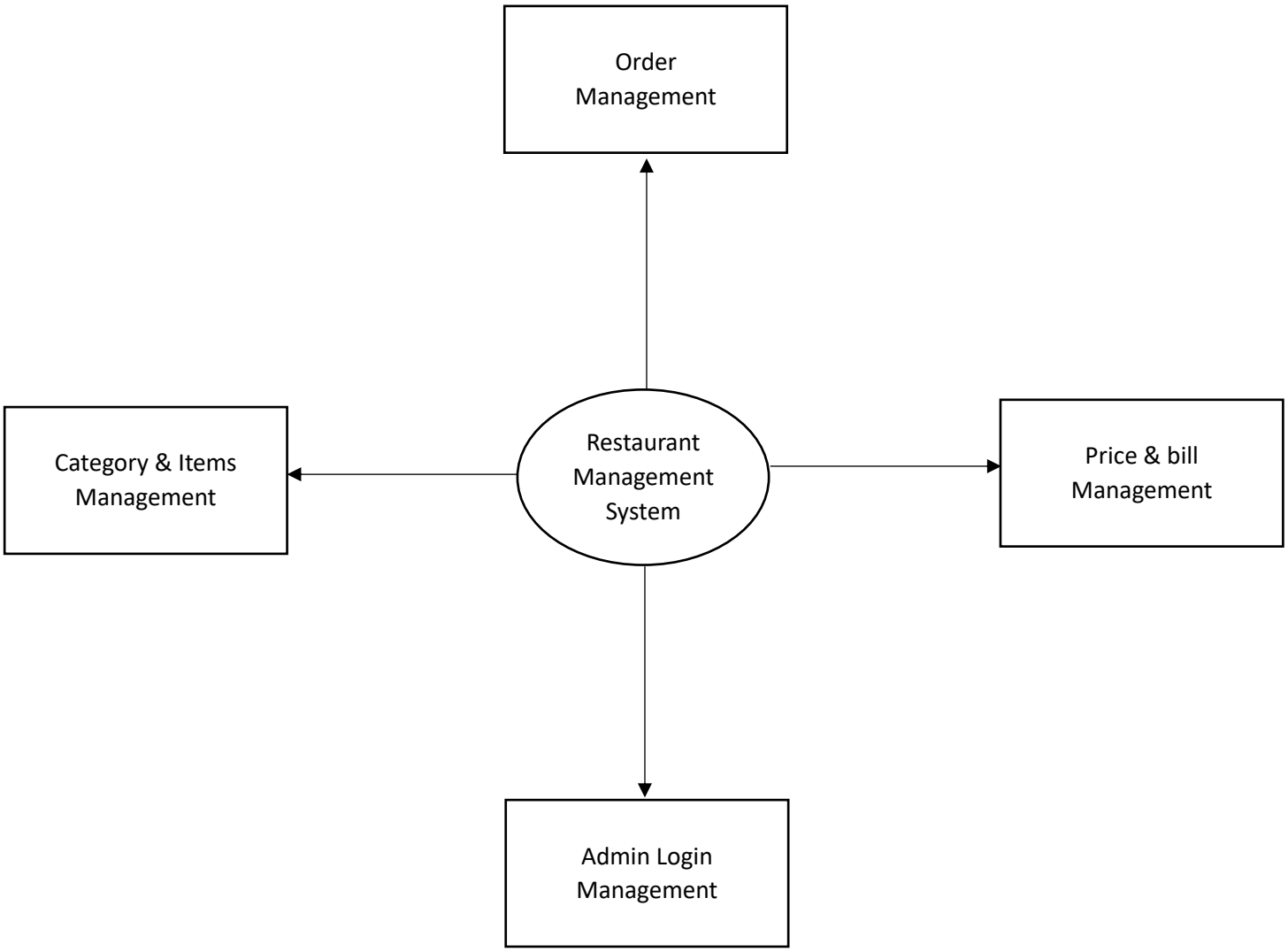
I. - Defines Flow of Data

II.  - Defines Input & Output of Data

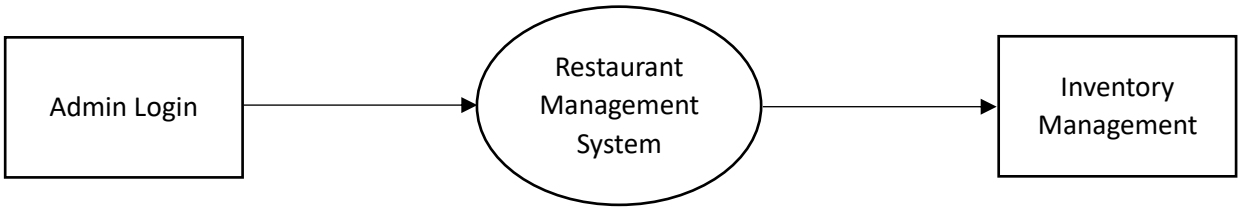
III.  - Defines Processing of Data

IV.  - Defines Database of Data Storage

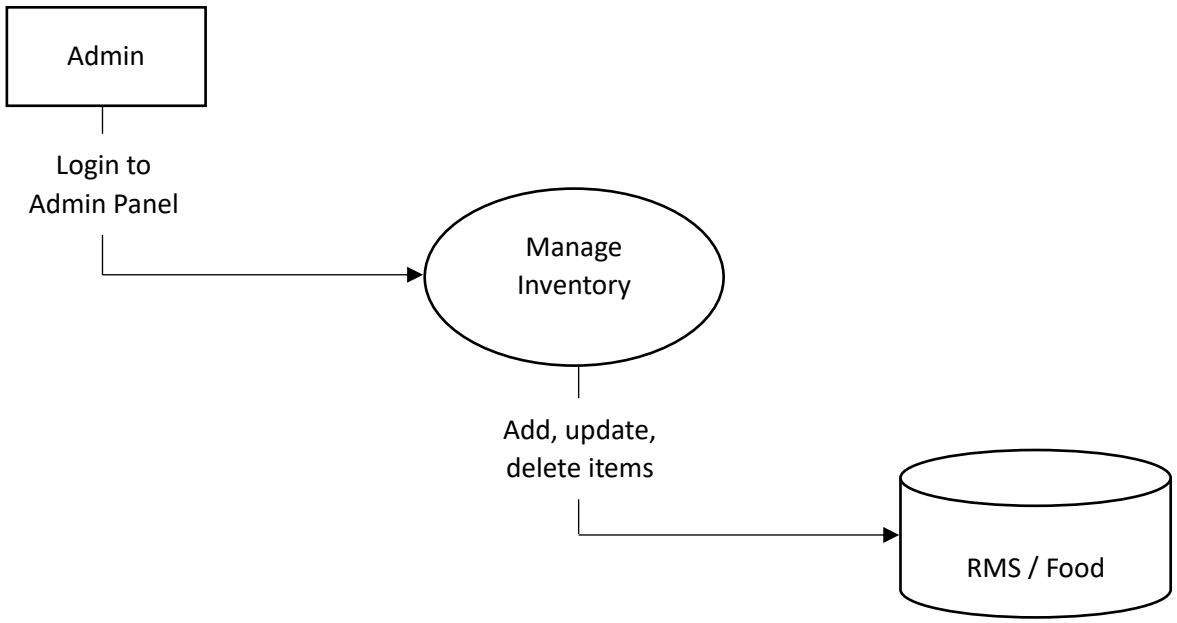
❖ RMS - Context (0 level) Data Flow Diagram



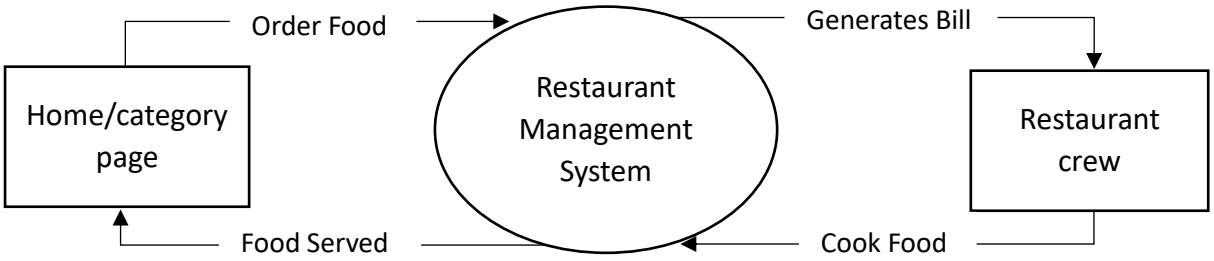
Admin – Context (0 level) Data Flow Diagram



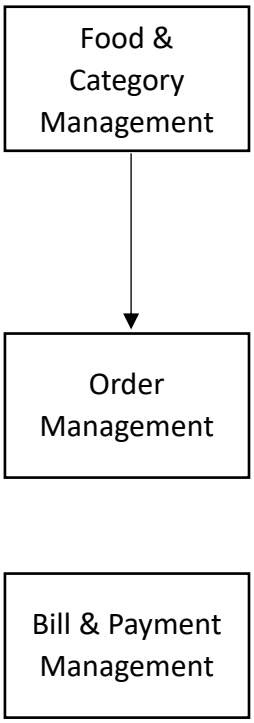
❖ Admin – 1st level Data Flow Diagram

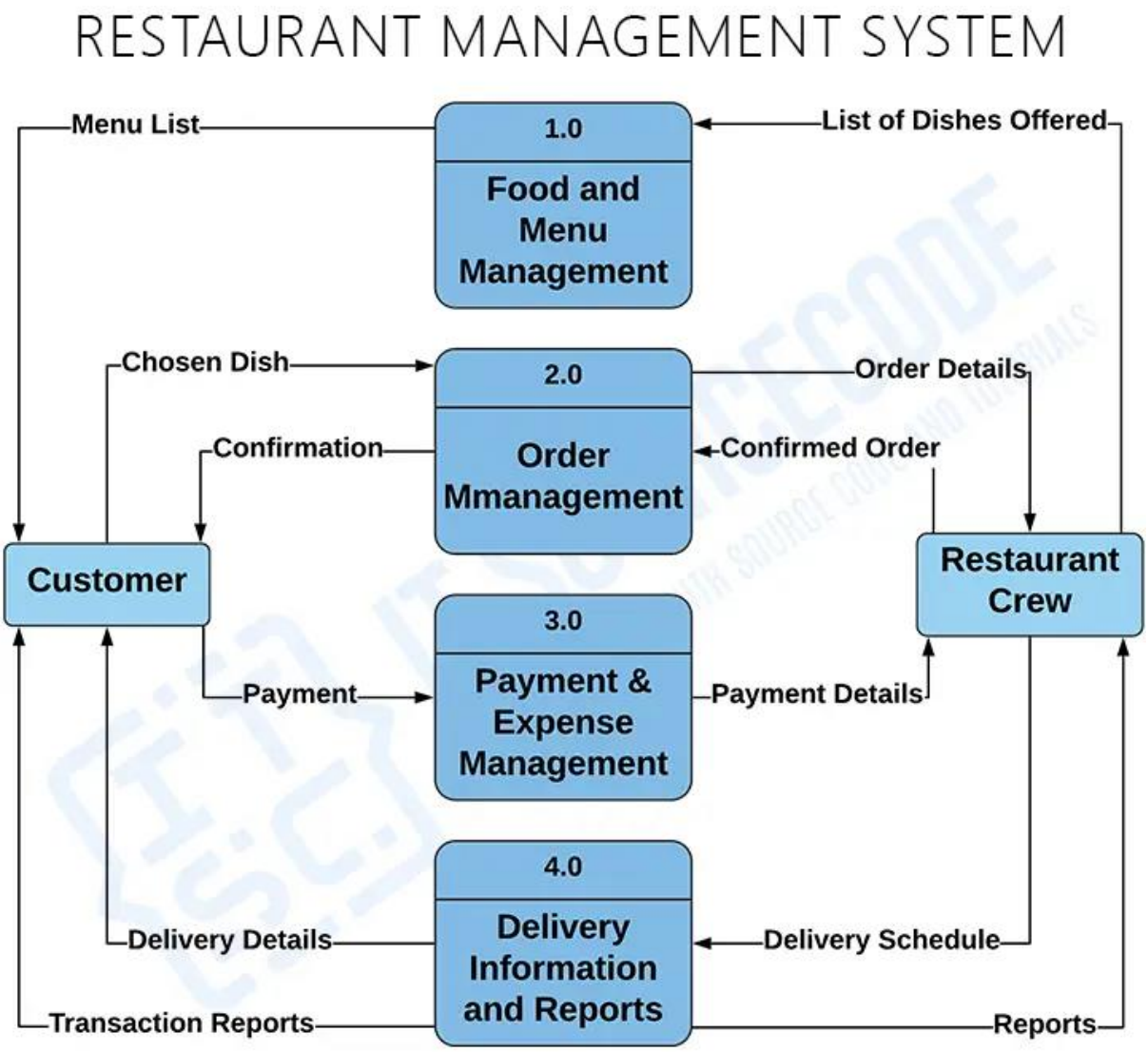


❖ Client/User – Context (0 level) Data Flow Diagram



❖ Client/ User – 1st level Data Flow Diagram




RESTAURANT MANAGEMENT SYSTEM
Menu List
1.0
Food and Menu Management
List of Dishes Offered
Chosen Dish
2.0
Order Mmanagement
Order Details
Confirmation
Confirmed Order
Customer
Restaurant Crew
3.0
Payment & Expense Management
Payment
Payment Details
4.0
Delivery Information and Reports
Delivery Details
Delivery Schedule
Transaction Reports
Reports
DATA FLOW DIAGRAM LEVEL 1

Chapter 8

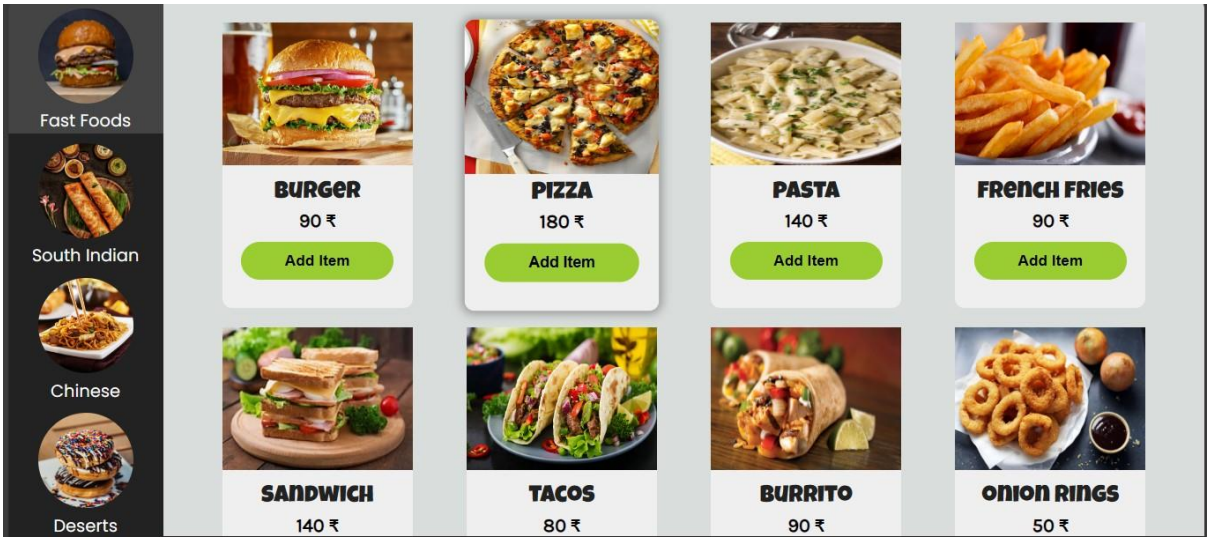
Designing: Admin & Client

8.1 Chapter Overview

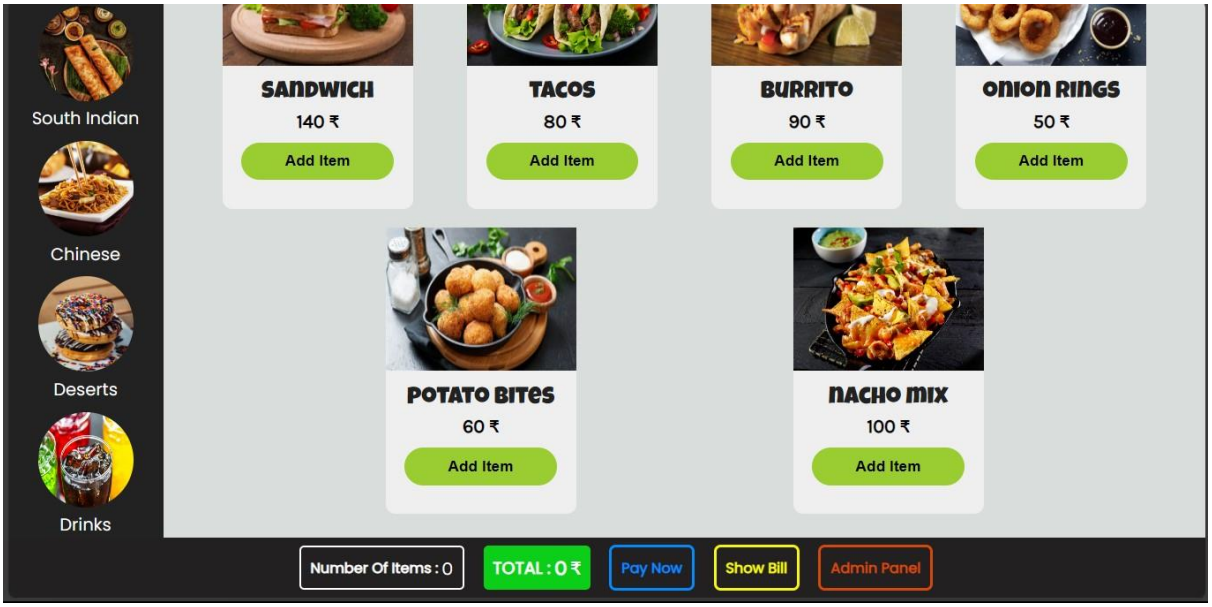
In this chapter we have displayed our projects overall layout and design which we have build using vanilla CSS and used other front-end web technologies. In short, we have displayed our home page (category page), Login page (admin panel login), Inventory managing page.

8.2 Clint (user) side page layout

- Category / Home page



-



Chapter 9
Coding

9.1 Chapter Overview

Here we have inserted our full code in a well-structured manner for clarity which includes files such as:

1. index.php 2. index.js 3. index.css 4. admin.php 5. admin.js
6. admin.css 7. adminPanel.php 8. adminPanel.js 9. adminPanel.css 10. getitems.php

9.2 Coding Client/User side

1. index.php

```
<?php
    $server = "localhost";
    $username = "root";
    $password = "";
    $database = 'rms';
    $con = mysqli_connect($server, $username, $password, $database);
    if (!$con) {
        die("connection to this database failed due to ".mysqli_connect_error());
    }
    //  echo "connected";
    if (isset($_GET['ctgr'])) {
        $category = $_GET['ctgr'];
        $sql = "SELECT * FROM `food` WHERE `ctgr` = '" . $category . "'";
        $result = mysqli_query($con, $sql);
        $num = mysqli_num_rows($result);
    } else {
        $category = 'fastfood'; // Set 'fastfood' as the default category
        $sql = "SELECT * FROM `food` WHERE `ctgr` = '$category'";
        $result = mysqli_query($con, $sql);
        $num = mysqli_num_rows($result);
```

```
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./index.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Inter&family=Karla&family=Luckiest+Guy&family=Ope
n+Sans&family=Poppins&family=Ubuntu&display=swap" rel="stylesheet">
    <title>RMS</title>
</head>
<body>
   <div class="container">
        <div class="main">
            <!-- left side -->
            <div class="leftPanel">
                <ul class="ctgr">
                    <li id="fastfood" class="active">
                        <img
src="https://th.bing.com/th/id/OIP.MYrVEIwwKi2UiC55nj5KJgHaGJ?pid=ImgDet&rs=1" alt="burger"
height="100px" width="100px">
                        <div class="title">
                            Fast Foods
                        </div>
                    </li>
                    <li id="southindian">
                        <img src="https://th.bing.com/th/id/OIP.9MfNFYWqS-
xzcl3XoPTMIwHaHa?pid=ImgDet&rs=1" alt="dosa">
                        <div class="title">
                            South Indian
                        </div>
                    </li>
```

```
                <li id="chinese">
                    <img
src="https://th.bing.com/th/id/R.2f106f02742dfca0c8e9f206b4b4a97a?rik=KsOhYeh1Z8Z27A&riu=http
%3a%2f%2fmagazine.foodpanda.pk%2fwp-
content%2fuploads%2fsites%2f13%2f2016%2f06%2fChinese-Featured-
Image.jpg&ehk=vIhaprTmcXRQTE0oWa1Dcm9gx6wELUQTBKyDbP0hAKU%3d&risl=&pid=ImgR
aw&r=0" alt="chinese">
                    <div class="title">
                      Chinese
                    </div>
                </li>
                <li id="deserts">
                    <img src="https://images.unsplash.com/photo-1551024601-
bec78aea704b?ixid=MXwxMjA3fDB8MHxzZWFyY2h8Mnx8ZGVzc2VydHxlbnwwfHwwfA%3D%3D
&ixlib=rb-1.2.1&w=1000&q=80" alt="Deserts">
                    <div class="title">
                      Deserts
                    </div>
                </li>
                <li id="drinks">
                    <img src="https://news.harvard.edu/wp-content/uploads/2019/03/beverage-
3548084_1920-1.jpg?w=1200&h=800&crop=1" alt="coldDrink">
                    <div class="title">
                      Drinks
                    </div>
                </li>
            </ul>
        </div>
        <!-- right side -->
        <div class="rightPanel">
            <div class="items">
                <?php
                if ($num > 0) {
                    while($row = mysqli_fetch_assoc($result)) {
                        echo '
                            <div class="card">
                                <img src="' . $row['image'] . '" alt="ff" class="cardImg">
```

```
                        <div class="name">
                            ' . $row['name'] . '
                        </div>
                        <div class="price">
                            <span class="prc">' . $row['price'] . '</span> ₹
                        </div>
                        <div class="quantity">
                            Quntity : <input type="number" name="qty" id="qty" class="qty" value="1"
placeholder="Quntity">
                        </div>
                        <button class="addItem">Add Item</button>
                    </div>
                ';
            }
        }
        ?>
    </div>
  </div>
</div>
<!-- bottom Panel -->
<div class="bottomPanel">
  <div class="numItem">
    <h3>Number Of Items : </h3> <div class="itemCount" id="itemCount">0</div>
  </div>
  <div class="tot">
    <h3>TOTAL : </h3>  <div class="total">0</div> <span class="rupee">₹</span>
  </div>
  <div class="pay">
    <button class="paynow" onclick="paynow()">Pay Now</button>
  </div>
  <div class="show">
    <button class="showbill" onclick="showBill()">Show Bill</button>
  </div>
  <div class="bottomItems">
```

```
            <button class="admin" id="admin">Admin Panel</button>
          </div>
        </div>
    </div>
    <div class="bill">
        <h2>Bill</h2>
        <div class="line"></div>
        <div class="billTitle">
            <div>srno</div><div>items</div><div>Quntity</div><div>price</div>
        </div>
        <div class="line"></div>
        <div class="billItems" id="billitems">
        </div>
        <div class="line"></div>
        <div class="billTotal">
        </div>
        <div class="gst">
         <div class="tax">
            SGST @9% :  <span id="cgst">0</span>
         </div>
         <div class="tax2">
            CGST @9% : <span id="sgst">0</span>
         </div>
         <div class="tax3">
            Round Off : <span id="roundOff"> 0</span>
         </div>
         <div class="line"></div>
         <div class="tax4">
            GRAND TOTAL : <span id="gTotal"> 0</span><span class="rupee"> ₹</span>
         </div>
        </div>
    </div>
    <div id="admin-container">
    </div>
```

```
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
   <script src="./index.js"></script>
</body>
</html>
```

2. index.js

```
//admin button
var adminButton = document.getElementById('admin');
adminButton.addEventListener('click', () => {
   history.pushState(null,'','/RMS/admin.php');
   handleRoute();
})
function handleRoute() {
   var path = window.location.pathname;
   var admin = document.getElementById('admin-container')
   if (path === '/RMS/admin.php') {
      window.location.reload();
   } else {
       admin.textContent = '404 Page not found';
   }
}
//add items
 // click effect on left panel
 const list = document.querySelectorAll('li');
 var id = null;
 function retrive() {
    $.ajax({
          url: 'getitems.php',
          method: 'GET',
          data: { ctgr: `${id}` },
          success: function (response) {
             // Process the JSON response and update the right panel
             // with the retrieved items.
```

```
            if (response.length > 0) {
                var itemsHTML = '';
                response.forEach(function(item) {
                    itemsHTML += `
                        <div class="card">
                            <img src="${item.image}" alt="ff" class="cardImg">
                            <div class="name">
                                ${item.name}
                            </div>
                            <div class="price">
                                <span class="prc">${item.price}</span> ₹
                            </div>
                            <div class="quantity">
                                Quntity : <input type="number" name="qty" id="qty" class="qty" value="1"
placeholder="Quntity">
                            </div>
                            <button class="addItem">Add Item</button>
                        </div>
                    `;
                });
                document.querySelector('.items').innerHTML = itemsHTML;
            }
        },
            error: function (xhr, status, error) {
            console.error(error);
            }
        });
}
function clk() {
    list.forEach((item) => item.classList.remove('active'));
        this.classList.add('active');
        id = this.id;
        if (id === 'fastfood') {
            retrive(id);
```

```
        } else if (id === 'southindian') {
            retrive(id);
        } else if (id === 'chinese') {
            retrive(id);
        } else if (id === 'deserts') {
            retrive(id);
        }else if (id === 'drinks') {
            retrive(id);
        }
}
list.forEach((item) =>
item.addEventListener('click', clk)
)
//add Itmes
var cardItems = [];
var count = 0;
var itemCount = document.getElementById('itemCount');
var amoount = document.querySelector('.total');
var bill = document.querySelector('.bill');
var billItems = document.getElementById('billitems');
var billTotal = document.querySelector('.billTotal');
function getTotalPrice() {
    var totalPrice = 0;
    for (let i = 0; i < cardItems.length; i++) {
        var items = cardItems[i];
        var price = parseFloat(items.price * items.quantity);
        totalPrice += price;
    }
    return totalPrice;
}
function getTotalQuantity() {
    var totalQuantity = 0;
    for (let i = 0; i < cardItems.length; i++) {
        var items = cardItems[i];
```

```
        var quantity = parseFloat(items.quantity);
        totalQuantity +=  quantity;
    }
    return totalQuantity;
}
document.querySelector('.items').addEventListener('click', function(event) {
    if (event.target.classList.contains('addItem')) {
        handleAddItemClick(event.target);
    }
})
var qty = document.querySelector('.qty');
function handleAddItemClick(button) {
    var card = button.parentNode;
    var quantity = card.querySelector('.quantity');
    quantity.style.display = "flex";
    if (button.innerText === 'Confirm') {
        confirm(card);
        quantity.style.display = "none";
        button.innerText = "Add Item";
    } else {
        button.innerText = "Confirm";
    }
}
// var addItem = document.querySelectorAll('.addItem');
// addItem.forEach((button) => {
//     button.addEventListener('click', handleAddItemClick);
// })
var cgst = document.getElementById('cgst');
var sgst = document.getElementById('sgst');
var rOff = document.getElementById('roundOff');
var gTotal = document.getElementById('gTotal');
var grandTotal = 0;
var value = 0;
```

```
function confirm(card) {
    while (card && !card.classList.contains('card')) {
        card = card.parentNode;
    }
    if (card) {
    var cardImg = card.querySelector('.cardImg');
    var name = card.querySelector('.name').innerText;
    var price = card.querySelector('.prc').innerText;
    var qty = card.querySelector('.qty').value;
    if (qty < 0 || qty == 0) {
        alert('Please Inter a Valid Quantity');
        qty = 0;
    } else {
        var item = {
            name: name,
            price: price,
            quantity: qty,
        };
    }
    // alert('Clicked "Add Item" button:\nName: ' + name + '\nPrice: ' + price);
    cardItems.push(item);
    console.log(cardItems);
    count++;
    var totalPrice = getTotalPrice();
    // tax
    var sgstTax = calSgst(totalPrice);
    var cgstTax = calCgst(totalPrice);
    itemCount.textContent = getTotalQuantity();
    amoount.innerText = totalPrice;
    sgst.innerText = sgstTax;
    cgst.innerText = cgstTax;
        subTotal = totalPrice + sgstTax + cgstTax;
        roundOff =  Math.round(subTotal)
        val = roundOff - subTotal;
```

```
        value = val;
        grandTotal = subTotal;
    rOff.innerText = value.toFixed(2);
    gTotal.innerText = parseInt(grandTotal)
    // console.log(val.toFixed(2));
    // console.log(parseInt(grandTotal));
    }
}
function paynow(){
    window.location.reload();
}
function calSgst(total) {
    return (total * 9)/100;
}
function calCgst(total) {
    return (total * 9)/100;
}
function showBill() {
    if (bill.style.visibility === "hidden") {
        bill.style.visibility =  "visible";
    } else {
        bill.style.visibility =  "hidden";
    }
    billItems.innerHTML = '';
    for (let i = 0; i< cardItems.length; i++) {
        var item = cardItems[i];
        var itemHtml = `
            <div>${i+1}</div> <div>${item.name}</div> <div>${item.quantity}</div> <div>${item.price *
item.quantity} <span class="rupee">₹</span></div>
        `;
        billItems.innerHTML += itemHtml
    }
    billTotal.innerHTML = ` Total : ${getTotalPrice()} <span class="rupee">₹</span>`;    }
```

3. index.css

```
*{
    box-sizing: border-box;
    margin: 0;
    padding: 0;
    /* overflow: hidden; */
}

body{
    height: 100vh;
    width: 100%;
}
.main{
    display: flex;
}
/* left side */
.leftPanel{
    /* background-color: #111021; */
    background-color: #222;
    height: 90vh;
    width: 25%;
    z-index: 1;
    overflow: auto;
    /* overflow-y: hidden; */
    /* box-shadow: rgba(0, 0, 0, 0.2) -5px 0px 0px inset; */
}
::-webkit-scrollbar{
    width: 0px;
    background-color: transparent;
}
.ctgr img{
    margin-top: 10px;
    border-radius: 50%;
    height: 100px;
    width: 100px;
```

```
    z-index: 2;
}
.ctgr{
    display: flex;
    flex-direction: column;
    align-items: center;
    height: 90vh;
}
.title{
    color: #fff;
    align-items: center;
    margin-top: 5px;
    font-size: 18px;
    font-family: 'Poppins', sans-serif;
    font-weight: 500;
}
li{
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    width: 100%;
}
.active{
    /* background-color: #33363d; */
    background-color: #444;
    height: 142px;
    width: 100%;
    transition: 0.5s;
}



/* right side */
.rightPanel{
```

```
    background-color: #d9dddc;
    height: 100vh;
    width: 100%;
}
.items{
    padding: 15px;
    display: flex;
    justify-content: space-evenly;
    flex-wrap: wrap;
    overflow: auto;
    max-height: 90vh;
}
.card{
    height: 300px;
    width: 200px;
    background-color: #efefef;
    display: flex;
    flex-direction: column;
    align-items: center;
    border-radius: 10px;
    margin: 10px;
}
.card:hover{
    transform: scale(1.02);
    box-shadow: 0px 0px 10px 3px rgba(0, 0, 0, 0.3);
    transition: 0.3s;
    overflow: hidden;
}
.card:hover > .cardImg{
    transform: scale(1.12);
    transition: 1s;
}
.cardImg{
    height: 150px;
```

```
    width: 200px;
}
.card > .name{
    margin-top: 10px;
    margin-bottom: 5px;
    font-size: 25px;
    font-family: 'Luckiest Guy', sans-serif;
    font-weight: 500;
    color: #222;
}
.card > .price{
    margin-bottom: 10px;
    font-family: 'karla', sans-serif;
    font-weight: 600;
    font-size: 20px;
}
.card > button{
    width: 80%;
    background-color: yellowgreen;
    border: none;
    outline: none;
    padding: 10px;
    font-weight: 700;
    border-radius: 20px;
    font-size: 16px;
}
/* bottomPanel */
.bottomPanel{
    position: absolute;
    left: 0;
    bottom: 0;
    z-index: 2;
    height: 10vh;
    width: 100%;
```

```
    background-color: #222021;
    display: flex;
    align-items: center;
    justify-content: center;
}
.bottomItems{
    margin-left: 20px;
}
.numItem{
    display: flex;
    margin-left: 20px;
    border: 2px solid #fff;
    padding: 10px;
    border-radius: 5px;
    color: #fff;
}
.numItem > h3{
    font-family: 'poppins', sans-serif;
    font-size: 15px;
    font-weight: 700;
    cursor: pointer;
}
.numItem > .itemCount {
    font-size: 20px;
    font-family: 'karla', sans-serif;
}
.tot{
    display: flex;
    margin-left: 20px;
    background-color:  rgb(12, 207, 25);
    padding: 10px;
    border-radius: 5px;
    color: #fff;
    font-family: 'poppins', sans-serif;
```

```
    font-size: 15px;
    font-weight: 700;
    cursor: pointer;
}
.tot > .total {
    font-size: 22px;
    font-family: 'karla', sans-serif;
}
.tot > .rupee{
    font-family: 'karla';
    font-size: 21px;
}
.pay{
    margin-left: 20px;
}
.paynow{
    padding: 10px;
    border-radius: 7px;
    background-color: #222021;
    border: 3px solid rgb(6, 139, 255);
    color: rgb(6, 139, 255);
    font-family: 'poppins', sans-serif;
    font-size: 15px;
    font-weight: 700;
    cursor: pointer;
}
.show{
    margin-left: 20px;
}
.showbill{
    padding: 10px;
    border-radius: 7px;
    background-color: #222021;
    border: 3px solid rgb(248, 248, 13);
```

```
    color: rgb(248, 248, 13);
    font-family: 'poppins', sans-serif;
    font-size: 15px;
    font-weight: 700;
    cursor: pointer;
}
.quantity{
    display: flex;
    justify-content: center;
    font-family:Arial, Helvetica, sans-serif;
    display: none;
    margin-top: 5px;
    margin-bottom: 5px;
}
.qty{
    width: 20%;
    border: none;
    outline: none;
    border-radius: 3px;
    background-color: #fff;
    text-align: center;
}
.admin{
    padding: 10px;
    border-radius: 7px;
    background-color: #222021;
    border: 3px solid rgb(213, 74, 10);
    color: rgb(213, 74, 10);
    font-family: 'poppins', sans-serif;
    font-size: 15px;
    font-weight: 700;
    cursor: pointer;
}
.bill{
```

```
    position: absolute;
    right: 0;
    bottom: 10%;
    background-color: #cbcbcb;
    border-radius: 10px;
    height: 500px;
    width: 350px;
    z-index: 2;
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 10px;
    font-family: 'poppins', sans-serif;
    visibility: hidden;
    overflow: scroll;
}
.billTitle{
    width: 100%;
    display: flex;
    justify-content: space-around;
}
.billItems{
    width: 100%;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
    justify-content: center;
    justify-items: center;
    gap: 10px;
    overflow: auto;
}
.billTotal{
    margin-top: 5px;
    transform: translateX(100px);
}
```

```
.line{
    margin-block: 7px;
    height: 2px;
    background-color: #000;
    width: 100%;
    z-index: 10;
}
.rupee{
    font-family: sans-serif;
}
.gst{
    display: flex;
    flex-direction: column;
    transform: translateX(85px);
}
.tax3{
    transform: translateX(8px);
}
/* mediaquery */
@media screen and (min-width: 901px) {
    .leftPanel {
      width: 20%;
    }
}
@media screen and (min-width: 1001px) {
    .leftPanel{
        width: 15%;
    }
}
@media screen and (max-width: 600px) {
    .leftPanel {
      width: 50%;
    }
    .ctgr img {
```

```
  margin-top: 5px;
  border-radius: 50%;
  height: 80px;
  width: 80px;
}
.title {
  margin-top: 3px;
  font-size: 14px;
}
.rightPanel {
  padding-top: 10px;
  background-color: #d9dddc;
  height: 90vh;
  width: 50vh;
}
.items {
  padding: 5px;
}
.card {
  height: 200px;
  width: 140px;
  margin: 5px;
}
.cardImg {
  height: 100px;
  width: 140px;
}
.card > .name {
  margin-top: 5px;
  font-size: 20px;
}
.card > .price {
  margin-bottom: 5px;
  font-size: 16px;
```

```
}
.card > button {
  width: 70%;
  padding: 8px;
  font-size: 14px;
}
.bottomPanel {
  height: 10vh;
}
.numItem {
  margin-left: 10px;
}
.numItem > h3 {
  font-size: 12px;
}
.numItem > .itemCount {
  font-size: 16px;
}
.tot {
  margin-left: 10px;
  font-size: 12px;
}
.tot > .total {
  font-size: 18px;
}
.pay {
  margin-left: 10px;
}
.paynow {
  padding: 8px;
  font-size: 12px;
}
.show {
  margin-left: 10px;
```

```
    }
    .showbill {
      padding: 8px;
      font-size: 12px;
    }
    .bill {
      height: calc(100vh - 200px);
      width: 80%;
    }
    .billItems {
      grid-template-columns: 1fr 1fr;
    }
    .rupee {
      font-size: 12px;
    }
    .gst {
      transform: none;
    }
    .tax3 {
      transform: none;
    }
}
```

9.3 Coding Admin side

4. admin.php

```
<?php
    $server = "localhost";
    $username = "root";
    $password = "";
    $database = 'user';
    $con = mysqli_connect($server, $username, $password, $database);
    if (!$con) {
        die("connection to this database failed due to ".mysqli_connect_error());
    }
```

```
    // echo "Succesfully connected to db";
if (isset($_POST['user']) && isset($_POST['pass'])) {
    $user = $_POST['user'];
    $pass = $_POST['pass'];
    $sql = "SELECT `user`, `pass` FROM `user` WHERE `srno` = '1'";
    $result = mysqli_query($con, $sql);
    $num = mysqli_num_rows($result);
    if ($num > 0) {
        while($row = mysqli_fetch_assoc($result)) {
            // echo " username is ". $row['user']. " and password is " . $row['pass'] . "<br>";
            if ($user === $row['user'] && $pass === $row['pass']) {
                // echo "login successfull";
                $login = "loged in";
            } else if ($user === $row['user'] && $pass !== $row['pass'] ) {
                // echo "password is incorrect";
                $login = "incorrect pass";
            } else if ($user !== $row['user']){
                // echo "please check your username or password";
                $login = "incorrect user";
            }
        }
    }
} else {
    $login = "user not found";
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./admin.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```
    <link
href="https://fonts.googleapis.com/css2?family=Inter&family=Karla&family=Luckiest+Guy&family=Ope
n+Sans&family=Poppins&family=Ubuntu&display=swap" rel="stylesheet">
    <script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>
    <script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>
    <title>Document</title>
</head>
<body>
    <div class="container">
        <div class="login">
        <div class="home">
            <div onclick="home()"><ion-icon name="home-outline"></ion-icon></div>
        </div>
            <div class="title">
                Hello user
            </div>
            <div class="line"></div>
                <form action="admin.php" id="'login" class="inp" method="post">
                    <input type="text" id="user" name="user" placeholder="Enter Your Name" required=true>
                    <?php
                        if ($login === "incorrect user") {
                            echo "<p class='error'>please check your username or password</p>";
                        }
                    ?>
                    <input type="password" name="pass" id="pass" placeholder="Enter Your password"
required=true>
                    <?php
                        if ($login === "incorrect pass") {
                            echo "<p class='error'>your password is incorrect</p>";
                        }
                    ?>
                    <button class="sub" id="sub">Submit</button>
                </form>
        </div>
    </div>
```

```
    <script>
      var log = "<?php echo $login ?>";
    </script>
    <script src="./admin.js"></script>
</body>
</html>
```

5. admin.js

```
var login = document.getElementById('login');
        var but = document.getElementById('home');
        var sub = document.getElementById('sub');
        console.log(log);
        if (log === "loged in") {
            window.location.href = '/RMS/adminPanel.php';
        }
        function home(){
            window.location.href = '/RMS/';
        }
```

6. admin.css

```
body{
    background-color: #d9dddc;
    overflow: hidden;
}
.container{
    height: 100vh;
    width: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
.login{
    display: flex;
```

```
    flex-direction: column;
    height: 350px;
    width: 360px;
    background-color: #e7e9ec;
    align-items: center;
    border-radius: 20px;
    position: relative;
}
.title{
    margin-top: 30px;
    font-family: 'poppins' , sans-serif;
    font-size: 40px;
}
.line{
    display: inline;
    width: 100%;
    height: 2px;
    background-color: #FFF;
    z-index: 2;
    margin-top: 30px;
}
input{
    width: 100%;
    padding: 12px;
    margin-bottom: 30px;
    border-radius: 10px;
    outline: none;
    border: none;
}
.inp{
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-top: 40px;
```

```
    position: relative;
}
.sub{
    border-radius: 5px;
    width: 60%;
    padding: 5px;
    font-size: 15px;
    font-weight: 700;
    font-family: sans-serif;
    border: 2px solid #000;
    background-color: #e7e9ec;
}
.sub:hover{
    background-color: #555;
    color: #fff;
    transition: 0.3s;
}
.error{
    display: flex;
    justify-content: center;
    align-items: center;
    color: red;
    font-size: 14px;
    position: absolute;
    bottom: 13%;
    width: 300px;
    font-family: 'Poppins', sans-serif;
}
.home{
    position: absolute;
    left: 2%;
    top: 1%;
    font-size: 30px;
    cursor: pointer;
```

```
}
```

7. adminPanel.php

```
<?php
    $server = "localhost";
    $username = "root";
    $password = "";
    $database = 'rms';
    $con = mysqli_connect($server, $username, $password, $database);
     if (!$con) {
        die("connection to this database failed due to ".mysqli_connect_error());
    }
   //  echo "connected";
   if (isset($_POST['name']) && isset($_POST['price']) && isset($_POST['ctgr'])) {
       $name = $_POST['name'];
       // $image = $_POST['image'];
       $price = $_POST['price'];
       $category = $_POST['ctgr'];
       if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
           $file = $_FILES['image'];
           $fileName = $file['name'];
           $fileTmpName = $file['tmp_name'];
           // Specify the directory to store images
           $uploadDir = './images/';
           // Generate a unique filename
           $uniqueFileName = uniqid() . '_' . $fileName;
           // Move the uploaded file to the specified directory
           if (move_uploaded_file($fileTmpName, $uploadDir . $uniqueFileName)) {
               // File upload successful, store the file name in the database
               $image = $uploadDir . $uniqueFileName;
               $sql = "INSERT INTO `food` (`name`, `image`, `price`, `ctgr`) VALUES ('$name',
'$image', '$price', '$category')";
               $result = mysqli_query($con, $sql);
               // if ($result) {
```

```
            //     echo "Item added successfully.";
            // } else {
            //     echo "Error: " . mysqli_error($con);
            // }
        } else {
            // echo "Error uploading file.";
        }
    } else {
        // echo "No file uploaded.";
    }
}
if (isset($_POST['oldname']) && isset($_POST['oldprice']) && isset($_POST['newname']) &&
isset($_POST['ctgr']) && isset($_POST['newprice'])) {
    $oldname = $_POST['oldname'];
    $oldprice = $_POST['oldprice'];
    $category = $_POST['ctgr'];
    $newname = $_POST['newname'] === "" ? $oldname : $_POST['newname'];
    $newprice = $_POST['newprice'] === "" ? $oldprice : $_POST['newprice'];

    if (isset($_FILES['newimage']) && $_FILES['newimage']['error'] === UPLOAD_ERR_OK) {
        $file = $_FILES['newimage'];
        $fileName = $file['name'];
        $fileTmpName = $file['tmp_name'];
        $uploadDir = './images/';
        $uniqueFileName = uniqid() . '_' . $fileName;

        if (move_uploaded_file($fileTmpName, $uploadDir . $uniqueFileName)) {
            $image = $uploadDir . $uniqueFileName;
            $sql = "UPDATE `food` SET `name` = '$newname', `image` = '$image', `price` =
'$newprice' WHERE `name` = '$oldname' AND `price` = '$oldprice'";
            $result = mysqli_query($con, $sql);
        //     if ($result) {
        //     echo "Item updated successfully.";
        //     } else {
        //         echo "Error: " . mysqli_error($con);
```

```
            //     }
            // } else {
            //     echo "Error uploading file.";
            }
        } else {
            $sql = "UPDATE `food` SET `name` = '$newname', `price` = '$newprice' WHERE `name` = '$oldname' AND `price` = '$oldprice' AND `ctgr` = '$category'";
            $result = mysqli_query($con, $sql);
            // if ($result) {
            //     echo "Item updated successfully.";
            // } else {
            //     echo "Error: " . mysqli_error($con);
            // }
        }
    }
    if (isset($_POST['deletename']) && isset($_POST['deleteprice']) && isset($_POST['ctgr'])) {
        $deletename = $_POST['deletename'];
        $deleteprice = $_POST['deleteprice'];
        $category = $_POST['ctgr'];
        $sql = "DELETE FROM `food` WHERE `name` = '$deletename' AND `price` = '$deleteprice' AND `ctgr` = '$category' ";
        $result = mysqli_query($con, $sql);
        // if ($result) {
        //     echo "delted Item successfully";
        // } else {
        //     echo "error";
        // }
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./adminPanel.css">
```

```
    <script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>
    <script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>
    <title>adminpage</title>
</head>
<body>
    <div class="container">
        <ul class="left">
            <div class="home-container">
                <div class="home" onclick="home()">
                    <ion-icon class="icon" name="home-outline"></ion-icon><span>HOME</span>
                </div>
            </div>
            <li class="active">
                <img
src="https://th.bing.com/th?id=OUG.37E2343F89DB28F95F8376BBDC7F13AA&w=236&h=320&c=7
&rs=1&qlt=90&bgcl=ecececc&o=6&pid=PersonalBing" alt="inventory">
                <div class="title">
                    Inventory
                </div>
            </li>
            <!-- <li>
                <img
src="https://th.bing.com/th?id=OUG.3534FF4E5D38047E9600A22D971C4936&w=236&h=320&c=7&
rs=1&qlt=90&bgcl=ecececc&o=6&pid=PersonalBing" alt="table" class="inventory">
                <div class="title">
                    Reservation
                </div>
            </li> -->
        </ul>
        <div class="right">
            <div class="layout">
                <div class="card">
                    <form action="adminPanel.php" method="post" enctype="multipart/form-data">
                        <img src="https://media.istockphoto.com/id/1457889029/photo/group-of-food-with-high-
content-of-dietary-fiber-arranged-side-by-side.jpg?s=612x612&w=is&k=20&c=n4-
M3CyEMJdmZEsXN92sIQAxQPDJeGPX2tkBk1s_RtE=" alt="food">
```

```
                <div class="insert">
                    <div class="opt">
                        Select Catagory :
                        <select name="ctgr" id="ctgr">
                            <option value="fastfood">Fast Food</option>
                            <option value="southindian">South Indian</option>
                            <option value="chinese">Chinese</option>
                            <option value="deserts">Deserts</option>
                            <option value="drinks">Drinks</option>
                        </select>
                    </div>
                    <input type="text" name="name" id="name" placeholder="Name of the Food"
required=true>
                    <input type="file" name="image" id="image" required=true class="pic">
                    <input type="number" name="price" id="price" placeholder="Price of the Food"
required=true>
                    <button>Add Item</button>
                </div>
            </form>
        </div>

        <div class="card2">
            <form action="adminPanel.php" method="post" enctype="multipart/form-data">
                <h1>UPDATE ITEM</h1>
                <div class="line"></div>
                <div class="insert">
                    <div class="opt">
                        Select Catagory :
                        <select name="ctgr" id="ctgr">
                            <option value="fastfood">Fast Food</option>
                            <option value="southindian">South Indian</option>
                            <option value="chinese">Chinese</option>
                            <option value="deserts">Deserts</option>
                            <option value="drinks">Drinks</option>
                        </select>
```

```
                </div>
                <div class="upd">
                    <input type="text" name="oldname" id="oldname" placeholder="Enter old name"
required=true>
                    <input type="number" name="oldprice" id="oldprice" placeholder="Enter old price"
required=true>
                </div>
                <div class="line"></div>
                <input type="text" name="newname" id="newname" placeholder="Enter New Name">
                <input type="file" name="newimage" id="newimage" class="pic" >
                <input type="number" name="newprice" id="newprice" placeholder="Enter New
Price">
                <button class="updBtn">Update Item</button>
            </div>

        </form>
    </div>

    <div class="card3">
        <form action="adminPanel.php" method="post" enctype="multipart/form-data">
            <h1>Delete Item</h1>
            <div class="line"></div>
            <div class="insert">
                <div class="opt">
                    Select Catagory :
                    <select name="ctgr" id="ctgr">
                        <option value="fastfood">Fast Food</option>
                        <option value="southindian">South Indian</option>
                        <option value="chinese">Chinese</option>
                        <option value="deserts">Deserts</option>
                        <option value="drinks">Drinks</option>
                    </select>
                </div>
                <input type="text" name="deletename" id="deletename" placeholder="Name of the
Food" required=true>
```

```
                    <input type="number" name="deleteprice" id="deleteprice" placeholder="Price of the
Food" required=true>
                    <button>Delete Item</button>
                </div>
            </form>
        </div>
    </div>
</div>
<script src="./adminPanel.js"></script>
</body>
</html>
```

8. adminPanle.js

```
const list = document.querySelectorAll('li');
function clk() {
    list.forEach((item) =>
        item.classList.remove('active'));
        this.classList.add('active');
}
list.forEach((item) =>
    item.addEventListener('click', clk)
)
function home() {
    window.location.href = '/RMS/';
}
```

7. adminPanel.css

```
*{
    box-sizing: border-box;
    margin: 0;
    padding: 0;
    overflow: hidden;
}
```

```
.body{
    height: 100vh;
    width: 100%;
}
.container{
    display: flex;
}
.left{
    width: 20%;
    height: 100vh;
    background-color: #222;
}
li{
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 30px 10px;
}
li > img{
    height: 140px;
    width: 140px;
    border-radius: 10px;
    margin-bottom: 10px;
}
li > .title{
    font-size: 16px;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: 600;
    color: #fff;
}
.active{
    background-color: #444;
    width: 100%;
}
```

```
.right{
    width: 80%;
    height: 100vh;
    background: url('https://wallpapercave.com/wp/wp4289147.jpg');
    background-size: cover;
}
.layout{
    padding: 30px;
    display: flex;
    flex-wrap: wrap;
    overflow: auto;
    height: 100vh;
}
.card{
    height: 400px;
    width: 300px;
    margin: 20px;
    background-color: #e9e9e9;
    border-radius: 10px;
    box-shadow: 0px 0px 10px 5px rgba(0, 0, 0, 0.7);
    /* display: none; */
}
.card2{
    height: 400px;
    width: 300px;
    margin: 20px;
    background-color: #e9e9e9;
    border-radius: 10px;
    box-shadow: 0px 0px 10px 5px rgba(0, 0, 0, 0.7);
    /* display: none; */
}
.card3{
    height: 280px;
    width: 300px;
```

```
    margin: 20px;
    background-color: #e9e9e9;
    border-radius: 10px;
    box-shadow: 0px 0px 10px 5px rgba(0, 0, 0, 0.7);
    /* display: none; */
}
form{
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 7px;
}
form > img{
    height: 200px;
    width: 280px;
    border-radius: 10px;
    margin-bottom: 10px;
}
.insert{
    display: flex;
    flex-direction: column;
    align-items: center;
}
input{
    margin-bottom: 10px;
    padding: 5px;
    border: none;
    outline: none;
    border-radius: 5px;
}
.pic{
    transform: translateX(40px);
}
.opt{
```

```
    font-family: sans-serif;
    margin-bottom: 7px;
    font-size: 13px;
}
.insert > button{
    width: 50%;
    background-color: #111021;
    color: #fff;
    padding: 4px;
    font-size: 13px;
    font-weight: 500;
    border: none;
    outline: none;
    border-radius: 5px;
    font-family:Verdana, Geneva, Tahoma, sans-serif;
    cursor: pointer;
}
select{
   margin-block: 5px;
}
.line{
    height: 1px;
    width: 100%;
    background-color: #000;
    margin: 10px;
}
h1{
    padding: 10px;
    font-family:Arial, Helvetica, sans-serif;
}
.upd{
    display: flex;
    flex-direction: column;
    align-items: center;
```

```
    /* margin: 20px; */
}
.upd > input{
    margin: 10px;
}
.card2 > form > .insert > input{
    margin: 5px;
}
.card2 > form > .insert > button{
    margin: 10px;
}
.card3 > form > .insert > input{
    margin-block: 10px;
}
.card3 > form > .insert > button{
    margin: 10px;
}
.home-container{
    background-color: #171717;
    color: #fff;
    height: 60px;
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 25px;
    font-family: sans-serif;
    font-weight: 600;
    cursor: pointer;
}
.icon{
    transform: translateY(2px);
    margin-right: 10px;
}
```

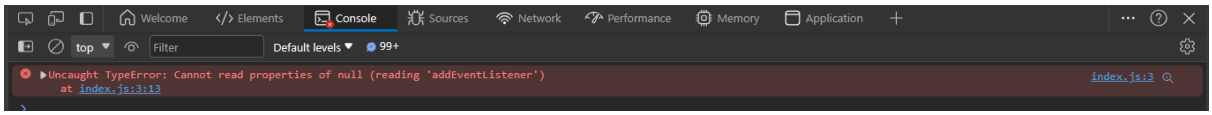
10. getitems.php

```
<?php
    $server = "localhost";
    $username = "root";
    $password = "";
    $database = 'rms';
    $con = mysqli_connect($server, $username, $password, $database);

    if (!$con) {
        die("connection to this database failed due to ".mysqli_connect_error());
    }

    if (isset($_GET['ctgr'])) {
        $category = $_GET['ctgr'];
        // Use proper escaping to prevent SQL injection
        $safeCategory = mysqli_real_escape_string($con, $category);
        $sql = "SELECT * FROM `food` WHERE `ctgr` = '$safeCategory'";
        $result = mysqli_query($con, $sql);
        $num = mysqli_num_rows($result);
        $items = array();
        if ($num > 0) {
            while($row = mysqli_fetch_assoc($result)) {
                $items[] = $row;
            }
        }
        // Return the items as JSON response
        header('Content-Type: application/json');
        echo json_encode($items);
    }
?>
```

Chapter 10

Error & solution

1. DOM manipulation error



When we try to access our list of categories at that we used to listen that event by using class-name but, we used id instead of class-name so we got this error.

Error code:

```
function handleRoute() {
    var path = window.location.pathname;
    var admin = document.querySelector('admin-container');
```

Solution Code:

```
function handleRoute() {
    var path = window.location.pathname;
    var admin = document.getElementById('admin-container');
```

2. PHP update error

Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near ''name` = 'golo' AND `price` = '20' AND `ctgr` = 'deserts'' at line 1 in D:\xampp\htdocs\RMS\adminPanel.php:84 Stack trace: #0 D:\xampp\htdocs\RMS\adminPanel.php(84): mysqli_query(Object(mysqli), 'UPDATE `food` S...') #1 {main} thrown in **D:\xampp\htdocs\RMS\adminPanel.php** on line **84**

Error code:

```
else {
            $sql = "UPDATE `food` SET `name` = '$newname' WHERE `price` = '$newprice'
`name` = '$oldname' AND `price` = '$oldprice' AND `ctgr` = '$category'";
            $result = mysqli_query($con, $sql);
```

Solution Code:

```
else {
            $sql = "UPDATE `food` SET `name` = '$newname', `price` = '$newprice' WHERE
`name` = '$oldname' AND `price` = '$oldprice' AND `ctgr` = '$category'";
            $result = mysqli_query($con, $sql);
```

3. Query passing error

Fatal error: Uncaught ArgumentCountError: mysqli_query() expects at least 2 arguments, 1 given in D:\xampp\htdocs\RMS\admin.php:20 Stack trace: #0 D:\xampp\htdocs\RMS\admin.php(20): mysqli_query('SELECT `user`, ...') #1 {main} thrown in **D:\xampp\htdocs\RMS\admin.php** on line **20**

Error code:

```
if (isset($_POST['user']) && isset($_POST['pass'])) {

    $user = $_POST['user'];
    $pass = $_POST['pass'];

    $sql = "SELECT `user`, `pass` FROM `user` WHERE `srno` = '1'";
    $result = mysqli_query($sql);
    if ($result) {
        echo "successful";
    } else {
        echo mysqli_error();
    }
    $num = mysqli_num_rows($result);
```

Solution code:

```
if (isset($_POST['user']) && isset($_POST['pass'])) {

    $user = $_POST['user'];
    $pass = $_POST['pass'];

    $sql = "SELECT `user`, `pass` FROM `user` WHERE `srno` = '1'";
    $result = mysqli_query($con, $sql);
    $num = mysqli_num_rows($result);
```

Chapter 11

Testing

11.1 Chapter Overview

This chapter describes the different testing techniques used to test the system including the results of some of the testing techniques applied. The tests did not just test the code directly but also tested some of the non-functional aspects of the system.

11.2 Unit Testing

In our restaurant management system (RMS), unit testing plays a crucial role in verifying the correctness of various functions and components. This section outlines our approach to unit testing in the RMS project.

The primary **purpose** of unit testing is to validate that each function or module of the RMS performs as expected. It helps us catch and rectify errors in the early stages of development, ensuring that the system's components function correctly before they are integrated into the complete system.

❖ Test Cases

We have created a suite of unit test cases that cover critical components and functionalities of the RMS. These test cases are designed to verify the correctness of individual methods and functions within the system cases reported below:

Menu Item Addition Test:

- **Purpose:** To verify that the system correctly adds new menu items.
- **Test Scenario:** We create a test menu item, add it to the system, and check if it appears in the menu.
- **Result:** The newly added menu item was displayed in the category page.

Order Processing Test:

- **Purpose:** To ensure that the order processing function works as expected.
- **Test Scenario:** We create a sample order, were we added items and check if the total added items were shown or not.
- **Result:** The added items should show on bill.

Admin login Test:

- **Purpose:** To ensure that the form is validate or not.
- **Test Scenario:** We have simply logged in were we input wrong username and password intentionally one by one.
- **Result:** Our system said that please check username and password.

11.3 Integration Testing

The primary goal of integration testing is to ensure that these elements work together seamlessly, as they would in a real-world scenario.

The **purpose** of integration testing for our RMS is to, Detect and address issues related to data flow and communication between different parts of the system. Verify that components integrated into the system cooperate correctly, producing the expected outcomes. Validate that external services and dependencies are properly connected and functioning as required.

❖ Test Scenarios

To execute integration testing, we have developed specific test scenarios for each integration point. These scenarios cover typical interactions between components and evaluate the system's ability to handle various data flows and inputs.

Menu Management:

Verified that menu items are correctly integrated into the ordering and billing processes, ensured that menu updates are reflected throughout the system.

Order Billing Process:

Tested the accuracy, speed, and mathematical calculations to ensure that is there any system error while processing order's bill.

Inventory Management:

Testing how changes in inventory (e.g., updating item) are integrated with the menu and category pages.

11.4 Regression Testing

Regression testing is an essential aspect of ensuring the stability and reliability of the restaurant management system (RMS). It involves systematically retesting previously validated features and functionalities to ensure that recent code changes or updates have not introduced new issues or adversely affected existing functionality.

The primary **purpose** of regression testing is to safeguard the quality of the RMS by preventing the unintended introduction of defects or regressions as the software evolves.

❖ Regression Test Suite

We have established a comprehensive regression test suite that covers critical aspects of the RMS, including key features, user interactions, and common use cases. This suite consists of test cases designed to reevaluate the following areas:

Menu Management:

- Ensured that adding, updating, and deleting menu items still functions correctly.
- Verified that the menu display remains accurate and responsive.

Order Processing:

- Confirmed that bill creation, displayed items total continue to work seamlessly.
- Checked that bill calculation updates are accurate.

Admin login:

- Validate that the login and authentication system remain secure.

❖ Regression Testing Process

Our regression testing process involves the following steps:

4. Selection of Test Cases:

We carefully select a subset of test cases from our regression test suite that are relevant to recent code changes or updates.

5. Test Execution:

The selected test cases are executed systematically to evaluate the areas impacted by the changes.

6. Comparison of Results:

We compare the results of the regression tests with baseline results obtained from previous testing phases.

Chapter 12

Implementation

12.1 Chapter Overview

The implementation chapter typically focuses on describing how we implemented or developed the project's components, features, or functionalities. It provides step by step insights into the technical aspects of our project, explaining the tools, technologies, coding practices, and methodologies used during the development process.

12.2 Technologies Used

The implementation of the Restaurant Management System (RMS) was carried out using the following technologies:

- Programming Language: JS, PHP
- Database Management System: MySQL
- Front-End: HTML, CSS, JavaScript
- Development Tools: VS Code

12.3 Architecture Overview

The RMS follows a three-tier architecture consisting of the following layers:

1. Presentation Layer – The front-end of the RMS is built using HTML, CSS, and JavaScript to create an interactive and user-friendly interface.
2. Application Layer – Mainly build using PHP.
3. Data Layer – MySQL is utilized as the relational database management system for storing and managing data.

12.4 Database Design

Entity-Relationship Diagram (ERD)

The database for the RMS consists of the following main entities:

- RMS: Stores information about each menu item, including its name, card image, price, and category.
- User: Securely stores admin data (e.g., Password).

Chapter 13

Conclusion

13.1 Chapter Overview

The conclusion of the RMS project documentation serves as the final chapter in the journey of creating a robust and user-friendly restaurant management solution. It encapsulates the key achievements, lessons learned, and the anticipated impact of the RMS on the restaurant management landscape.

13.2 Key Achievements

- **Robust Architecture:** The RMS is built on a solid three-tier architecture with technologies like PHP, JS, MySQL, ensuring a strong foundation for scalability and flexibility.
- **User-Centric Design:** The user interface prioritizes user-friendliness and accessibility, offering an intuitive and on touch experience for both restaurant staff and customers.
- **Comprehensive Functionality:** The RMS encompasses essential features such as menu management, automatic bill generation, and secure admin authentication, making it a valuable asset for restaurant operations.
- **Rigorous Testing:** A thorough testing strategy, including unit, integration, and regression testing, has ensured the reliability and stability of the system.

13.3 Lessons Learned

- **Agile Development:** Embracing Agile methodologies allowed for adaptability, responsiveness to professors' feedback, and project alignment with user needs.
- **Security Prioritization:** Recognizing the critical importance of data security, robust authentication mechanisms, and encryption practices were integrated into the RMS.
- **Continuous Improvement:** The project highlighted the importance of ongoing maintenance, enhancements, and staying attuned to evolving needs.

13.4 Impact and Future Directions

The RMS project is poised to transform restaurant management by enhancing customer satisfaction, optimizing menu offerings, and streamlining staff workflows. The conclusion anticipates user adoption, ongoing support, and collaboration within the restaurant management community for future development.