



**ATMIYA UNIVERSITY**

**For the award of the degree of B.Sc. Information Technology Semester – VI (2021-2024)**

**Project Report on Dessert Delight Submitted by Devangi Shingala**

**Submitted to**

**Department of Computer Science & Information Technology Atmiya University, Rajkot.**

DESSERT DELIGHT

“Delight At First Bite”

A PROJECT SUBMITTED TO

**Atmiya University**

## Department of Computer Science & Information Technology

### RAJKOT



*Submitted in partial fulfillment of the requirements for the degree of*

## “B.Sc. Information Technology”

**Sem-6 (Year 2021-2024)**

|  |  |
| --- | --- |
| **Submitted By: -** | **Guided By: -** |
| **Devangi Shingala J.** | **Mr. Malay Solanki** |
| **Khushi Menpara** | **Mr. Kshitij Vachhani** |



**Project ID:** BIT6F1Y003  **Date:** 06-03-2024

### Certificate

This is to certify that,

**Ms. Devangi Shingala** of

B.Sc. Information Technology

Semester VI

Has satisfactorily completed the project on

**Dessert Delight**

For, Department of Computer Science & Information Technology,

ATMIYA University, Rajkot.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature Signature**

**Mr. Kshitij Vachhani Dr. Divyesh Gohel** (Project Guide) (Program Coordinator)

****

**PROJECT PROFILE**

|  |  |
| --- | --- |
| **Project Title** | **“Dessert Delight”** |
| **Organization** | **Atmiya University-Rajkot** |
| **Front-End Tools** | **Bootstrap5, Google Chrome, Visual Studio Code** |
| **Back-End Tools** | **XAMPP (for Database- MySQL, Apache)** |
| **Language** | **HTML, CSS, JavaScript, PHP, Bootstrap5** |
| **Platform Used** | **Windows Platform, Firefox Browser** |
| **Developed By** | 1. **Shingala Devangi J.** 2. **Menpara Khushi A.** |
| **Project Guide** | 1. **Mr. Malay Solanki.** 2. **Mr. Kshitij Vachhani.** |

**STUDENT PROFILE**

|  |  |
| --- | --- |
| **Name** | **Shingala Devangi Jayeshbhai.** |
| **Program** | **B.Sc. I.T.** |
| **Enrolment No** | **210802106** |
| **Subject Code** | **21BITCR601** |
| **Class** | **F1** |
| **Batch** | **Y** |
| **Roll No** | **45** |
| **Project Title** | **Dessert Delight** |
| **Address** | **Kalawad Road, Rajkot.** |
| **Mobile** | **9428316741** |
| **Email Id** | [**devangi1010@gmail.com**](mailto:devangi1010@gmail.com) |

The project of "Dessert Delight" has been a unique experience for me. I am very thankful to the Department of Computer Science at Atmiya University, which gave me the opportunity to take on this project.

I express my gratitude to our Program Coordinator, Dr. Divyesh Gohel, and my project guides, Mr. Kshitij Vachhani and Mr. Malay Solanki, for their invaluable support and guidance during thecourse of the project. Your useful advice and suggestions were really helpful to me during theproject completion. In this aspect, I am eternally grateful to you.

I am also very thankful to "Atmiya University" for including the project work as part of the syllabus, without which I would not have gained the experience of developing software like this.

Thanking You.

I, hereby declare that the project work entitled **“Dessert Delight”** is the original work done by me, and I further declare that it is never submitted anywhere else in part or in full.

Signature Shingala Devangi J.

[210802106]

**ABSTRACT**

Our project Dessert Delight revolves around creating an online shopping website, a digital marketplace that allows customers to browse, select, and purchase products from the comfort of their homes. This website aims to provide a user-friendly and convenient shopping experience while benefiting both customers and sellers.

To do this, we will:

* **User-Friendly Interface:**
  + We will design a simple and intuitive website layout, making it easy for customers to navigate and find the products they desire.
* **Product Catalog:**
  + The website will feature a diverse range of products, categorized for easy exploration.
* **Feedback System:**
  + We will add a way for users to provide feedback and reviews. This is like sharing your thoughts about a product with others, which helps everyone make better choices.
* **Admin Side:**
  + Sellers will have access to a dedicated Admin Side where they can manage their product, manage Category, and handle customer feedback and review.

Our project aims to simplify the online shopping experience for consumers while providing a platform for sellers to expand their businesses. By prioritizing user-friendliness, we intend to create a valuable online shopping destination that benefits both customers and sellers alike.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Chapter Title** | **Page no** |
| **CHAPTER 1: INTRODUCTION** | | **1** |
| 1.1 | Problem Statement | 2 |
| 1.2 | Project Scope | 3 |
| 1.3 | Purpose | 4 |
| **CHAPTER 2: REQUIREMENTS AND ANALYSIS** | | **5** |
| 2.1 | System Analysis | 6 |
| 2.2 | Software and Hardware Requirements | 8 |
| **CHAPTER 3: PROJECT PLANNING AND**  **SCHEDULING** | | **9** |
| **CHAPTER 4: SYSTEM DESIGN** | | **13** |
| 4.1 | Logical Design | 14 |
|  | 4.1.1 Use Case Diagram | 14 |
|  | 4.1.2 Data flow diagram (DFD) | 16 |
|  | 4.1.3 ER Diagram | 21 |
|  | 4.1.4 Module Design | 23 |
|  | 4.1.5 Data Dictionary | 25 |
| **CHAPTER 5: SCREENLAYOUT AND TESTING** | | **28** |
| 5.1 | Screen Layout | 29 |
|  | 5.1.1 Admin Side | 29 |
|  | 5.1.2 User Side | 42 |
| 5.2 | Testing Approach | 59 |
| **CHAPTER 6: SYSTEM SECURITY AND MEASURES** | | **64** |
| **CHAPTER 7: FUTURE SCOPE AND ENHANCEMENT** | | **66** |
| **CHAPTER 8: CONCLUSION AND LIMITATIONS** | | **68** |
| **CHAPTER 9: BIBLOGRAPHY** | | **71** |

Chapter 1: Introduction

# CHAPTER 1: INTRODUCTION

### Problem Statement:

* **Inventory Management:**
  + The shop struggles to maintain accurate records of its inventory, leading to overstocking or understocking of popular and seasonal candies.
  + Inventory tracking is currently done manually, making it time-consuming and error-prone.
  + Expiry dates of products are often overlooked, resulting in the sale of expired items.
* **Sales Tracking:**
  + The shop lacks a comprehensive Sales tracking system, making it difficultto analyze sales trends, popular products, and customer preferences.
  + Sales data is recorded manually, which can lead to data entry errors and inconsistencies in financial records.
* **Customer Satisfaction:**
  + Inefficient inventory management sometimes results in unavailability of preferred candies, leading to customer dissatisfaction.
  + Lack of a loyalty program or customer database limits the shop's ability to engage with and reward loyal customers.
* **Reporting and Analysis:**
  + The shop lacks tools to generate reports and analyze data, hindering decision-making processes and future planning.
  + The absence of historical sales data makes it challenging to forecast demand and optimize inventory.
* **Project Objectives:**
  + The primary objectives of the Dessert Delight Inventory and Sales Dessert Delight Management System project are as follows:
  + Develop a user-friendly software solution that allows efficient tracking of inventory, including stock levels, product details.
  + Improve customer satisfaction by ensuring product availability and implementing a customer database or loyalty program.

By addressing these objectives, the Dessert Delight Inventory and Sales Management System aims to streamline operations, enhance customer satisfaction, and improve the overall profitability of our Dessert Delight.

### Project Scope:

* **Product Catalog:**
  + Create an organized and visually appealing catalog of ice cream products, including various flavors, sizes, and toppings.
  + Include high-quality images, detailed descriptions, and pricing information for each product.
* **User Registration and Authentication:**
  + Allow customers to create accounts with their personal information for a seamless shopping experience.
  + Implement secure user authentication and password recovery mechanisms.
* **Order Management:**
  + Develop an order management system to track and process customer orders.
  + Allow customers to view their order history and check the status of their current orders.
* **Category Management:**
  + In Category Management there is an option to add new category so that user can get more options to choose.
* **Customer Reviews and Ratings:**
  + Allow customers to leave reviews and ratings for ice cream products.
  + Display product ratings and reviews on product pages to assist other customers in making informed decisions.
* **User Account Management:**
  + Provide customers with the ability to their profiles, addresses, and contact information.
  + Implement a "forgot password" feature for account recovery.
* **Mobile Responsiveness:**
  + Ensure that the website is responsive and accessible on various devices, including smartphones and tablets.
* **Security and Privacy:**
  + Implement strong security measures to protect customer data and payment information.
  + Comply with data privacy regulations and guidelines.

### Purpose:

* **Expand Business Reach:**
  + Enable the candy shop to reach a wider audience by establishing an online presence, attracting customers who prefer the convenience of ordering ice cream products from the comfort of their homes.
* **Enhance Customer Convenience:**
  + Offer customers a user-friendly platform to browse, select, and purchase ice cream products at their convenience, 24/7.
* **Improve Customer Experience:**
  + Provide an engaging and interactive shopping experience with detailed product information, reviews, and easy-to-use features.
* **Efficient Order Management:**
  + Streamline order processing and fulfillment, making it easier for the shop to manage customer orders and inventory.
* **Data-Driven Decisions:**
  + Collect and analyze customer data and sales information to make informed decisions about product offerings, marketing strategies, and customer engagement.
* **Scalability:**
  + Create a platform that can scale to accommodate future growth and expansion of the business.
* **Compliance and Security:**
  + Ensure that the website complies with data privacy regulations and provides a secure environment for online transactions, building trust with customers.
* **Customer Feedback:**
  + Gather valuable feedback from customers through reviews and ratings, helping the shop enhance its offerings and services.

Chapter 2: Requirements and Analysis

# CHAPTER 2: REQUIREMENTS AND ANALYSIS

### System Analysis:

System analysis is a crucial phase in the development of any information system or software project. It involves a comprehensive study and evaluation of the existing system or the problem domain to understand its current state, identify its shortcomings, and define the requirements for a new system. Here is an overview of the key aspects and activities involved in system analysis:

* **Understanding the Problem Domain:**
  + Identify the organization's objectives and goals that the system aims to support or improve.
  + Interview stakeholders, including end-users, managers, and subject matter experts, to gain a deep understanding of the problem domain.
  + Gather and document information about existing processes, workflows, data, and technologies.
* **Problem Identification:**
  + Identify and document the shortcomings and limitations of the existing system, including inefficiencies, bottlenecks, and areas for improvement.
  + Define the specific problems or challenges that the new system should address.
* **Requirements Gathering:**
  + Elicit and document detailed functional and non-functional requirements for the new system.
  + Prioritize requirements based on their importance and impact on the organization's goals.
  + Involve stakeholders in requirement validation and verification to ensure accuracy and completeness.
* **Feasibility Study:**
  + Evaluate the technical, operational, economic, and schedule feasibility of the proposed system.
  + Assess the potential risks and challenges associated with system development and implementation.
  + Determine whether the benefits of the new system outweigh the costs and risks.
* **Data Analysis:**
  + Analyze the data requirements of the system, including data sources, data formats, data storage, and data flow.
  + Create data models (e.g., entity-relationship diagrams) to represent the data structure and relationships.
* **Process Modeling:**
  + Create process models (e.g., flowcharts or process diagrams) to represent current and future workflows and business processes.
  + Identify opportunities for process improvement and optimization.
* **System Design Constraints:**
  + Define any constraints or limitations that may affect the system's design, such as hardware and software platforms, security requirements, and regulatory compliance.
* **Use Case Development:**
  + Develop use cases to describe how different actors (users and systems) interact with the proposed system.
  + Use cases to clarify system behavior and user interactions.
* **Prototyping and Mockups:**
  + Create prototypes or mockups of the user interface to visualize and refine the system's user experience.
  + Gather user feedback to improve the design.
* **Documentation:**
  + Create detailed system analysis documentation, including requirement specifications, data models, process models, and use cases.
  + Ensure that documentation is clear, well-organized, and accessible to all project stakeholders.
* **Risk Assessment:**
  + Identify potential risks and develop risk mitigation strategies.
  + Anticipate and plan for contingencies to address unforeseen challenges during system development.
* **Review And Validation:**
  + Conduct reviews and validation sessions with stakeholders to ensure that the system analysis accurately reflects their needs and expectations.
* **Final System Proposal:**
  + Summarize the results of the system analysis in a final proposal thatoutlines the scope, requirements, feasibility, and recommended approach for system development.

### Software and Hardware Requirements:

* **Software Requirements:**

|  |  |  |
| --- | --- | --- |
| **Software Component** | **Version/Specification** | **Purpose** |
| Operating System | Windows Server 2016 | Server-side environment |
| Web Server | IIS 10 | Host and serve web applications |
| Database Server | XAMPP | Store and manage application data |
| Application Framework | Visual Studio Code | Framework for building web applications |
| Frontend Framework | PHP | User interface development |
| Code Editor | Visual Studio Code | Development and code editing |
| Version Control System | Git | Source code versioning and collaboration |

* **Hardware Requirements:**

|  |  |  |
| --- | --- | --- |
| **Hardware Component** | **Minimum Requirement** | **Recommended Requirement** |
| Processor | Dual-core, 2.0 GHz | Quad-core, 2.5 GHz |
| RAM | 4 GB | 8 GB |
| Storage | 50 GB HDD | 256 GB SSD |
| Network | 1 Gbps Ethernet | 1 Gbps Ethernet |
| Display | 1024x768 resolution | 1920x1080 resolution |

Chapter 3: Project Planning and Scheduling

**WorkInformation**

# CHAPTER 3: PROJECT PLANNING AND SCHEDULING

### Planning & and Scheduling:

|  |  |  |
| --- | --- | --- |
| **Date** | **Provide Time for Task** | **Work** |
| 12/12/23 | 3or 4days | Decide Project name And Definition |
| 15/12/23 | 2or 3days | Explore Existing Websites |
| 17/12/23 | 1or 2days | Listing Features |
| 19/12/23 | 4or 5days | Draw Diagrams |
| 21/1/24 | 3or 4days | Design Logo and Tagline |
| 23/1/24 | 5or 6days | Page Designing on paper |
| 28/1/24 | 4or 5days | Page Designing |
| 18/2/24 | 15or16 days | Coding |
| 10/3/24 | 9or 10days | Documentation |

**Time Line Chart:**



**ProvidingTimeforDessert Delight**

Documentation

10

Coding

16

PageDesigning

5

PageDesigningon paper

6

DesignLogoandTagline

4

DrawDiagram

5

ListingFeatures

2

ExploreExistingWebsites

3

DecideprojectnameandDefinition

4

0

5

10

15

20

**Days**

ProvidingTimeforDessertDelight

**System Development Life Cycle:**

The **System Development Life Cycle (SDLC)** provides a well-structured frameworkthat gives an idea, of how to build a system. It consists of steps as follows- Plan, Analyze, Design, Develop, Implement and Maintain. In this article, we will see all the stages of system development. We will delve into the significance of each stage, emphasizing the critical role played by System Design in the overall process.

The System Development Life Cycle encompasses a series of interconnected stages that ensure a systematic approach to system development. The stages include Planning, Analysis, Design, Development, Implementation, and Maintenance. Each stage contributes to the successful completion of the system, with System Design serving as a crucial component.

### Stage 1: Planning

* The Planning stage lays the groundwork for the entire SDLC. It involves identifying the system’s goals, defining project scope, setting objectives, establishing timelines, and determining available resources. Planning ensures that the development process aligns with organizational needs and sets a clear direction for subsequent stages.

### Stage 2: Analyzing

* During the Analysis stage, the focus is on gathering and understanding the requirements of the system. This includes conducting interviews, studying existing processes, and identifying stakeholders’ needs. The gathered informationserves as a basis for designing a system that meets users’ expectations and addresses organizational challenges.

### Stage 3: Designing

* System Design is a critical stage in the SDLC, where the requirements gathered during the Analysis phase are translated into a detailed technical plan. It involves designing the system’s architecture, database structure, and user interface, and defining system components. The Design stage lays the foundation for the subsequent development and implementation phases.

### Stage 4: Development

* The Development stage involves the actual coding and programming of thesystem. Based on the design specifications, developers write code, create database structures, and implement necessary functionalities. Rigorous testingand quality assurance are performed to ensure the system’s accuracy, performance, and adherence to the design requirements.

### Stage 5: Implementation

* This stage involves deploying the developed system into the production environment. This includes activities such as system installation, data migration, training end-users, and configuring necessary infrastructure. Implementation requires careful planning and coordination to minimize disruptions and ensure a smooth transition from the old system to the new one.

### Stage 6: Maintenance

* Maintenance is an ongoing stage that involves monitoring, managing, and enhancing the system’s performance and functionality. It includes activities such as bug fixes, updates, security patches, and addressing user feedback. Regular maintenance ensures the system remains reliable, secure, and adaptable to changing business needs.

Software DLC is all about building software in a phase approach systematically whereas System DLC is about implementing hardware and software in a phased manner systematically.

Chapter 4: System

Design

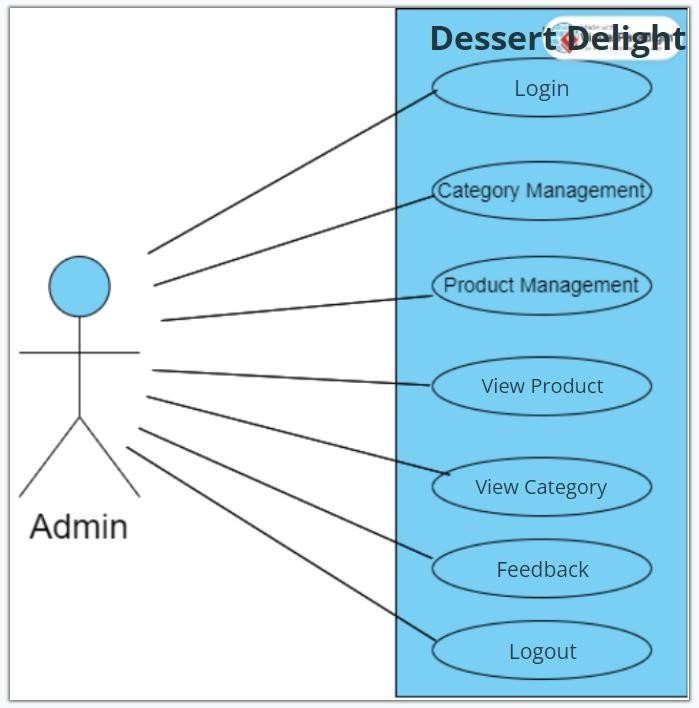
# CHAPTER 4: SYSTEM DESIGN

### Logical Design:

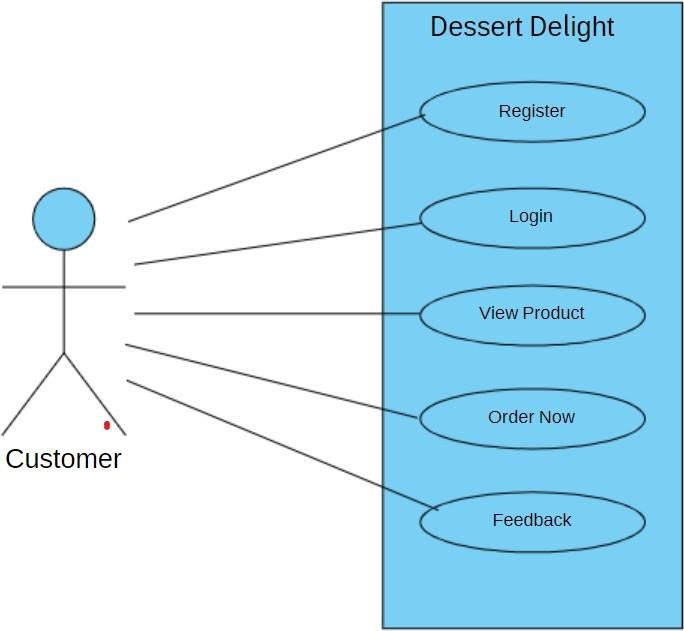
* + 1. **Use Case Diagram:**
       - In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system.
       - Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe whatthe system does and how the actors use it, but not how the system operates internally.
       - Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process.
       - Use-case diagrams are helpful in the following situations:
         * Before starting a project, you can create use-case diagrams to model a business so that all participants in the project share an understanding of the workers, customers, and activities of the business.
         * While gathering requirements, you can create use-case diagrams to capture the system requirements and to present to others what the system should do.
         * During the analysis and design phases, you can use the use cases and actors from your use-case diagrams to identify the classes that the system requires.
         * During the testing phase, you can use use-case diagrams to identify tests for the system.

### Symbols:

* **Admin Side:**



### User Side:



* + 1. **Data Flow Diagram (DFD):**

A data flow diagram shows the way information flows through a process orsystem. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

Data flow diagrams visually represent systems and processes that would be hard to describe in just words. You can use these diagrams to map out an existing system and make it better or to plan out a new system for implementation.Visualizing each element makes it easy to identify inefficiencies and produce the best possible system.

### Data flow diagram levels:

Data flow diagrams are also categorized by level. Starting with the most basic, level 0, DFDs get increasingly complex as the level increases. As you build your own data flow diagram, you will need to decide which level your diagram will be.

### Level 0 DFD:

Also known as context diagram, are [the most basic data flow diagrams](https://www.lucidchart.com/pages/templates/data-flow-diagram-level-0). They provide a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities. For instance, the example shown below illustrates the hotel reservation process with the flow of information between admin and guests.

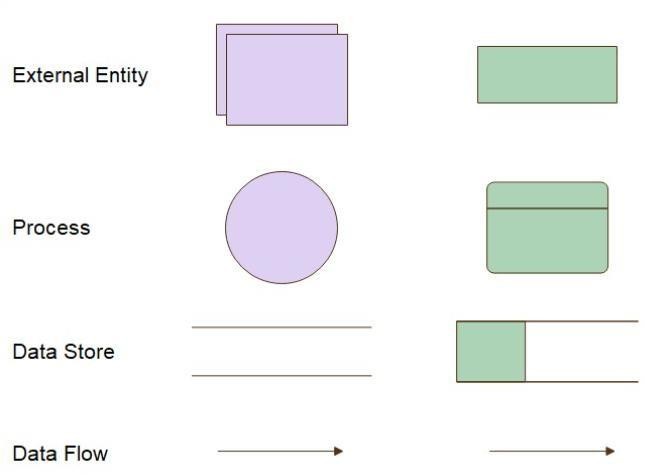
### Level 1 DFD:

**Level 1 DFDs** are still a general overview, but they go into more detail than a context diagram. In level 1 DFD, the single process node from the context diagram is broken down into sub-processes. As these processes are added, the diagram will need additional data flows and data stores to link them together. In the hotel reservation example, this can include adding the room selection and inquiry processes to the reservation system, as well as data stores.

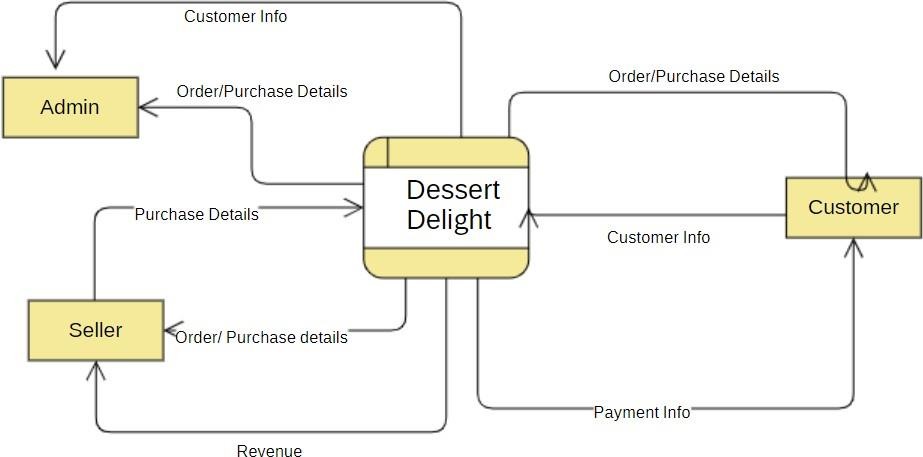
### Level 2 DFD:

Level 2 DFDs simply break processes down into more detailed sub- processes.

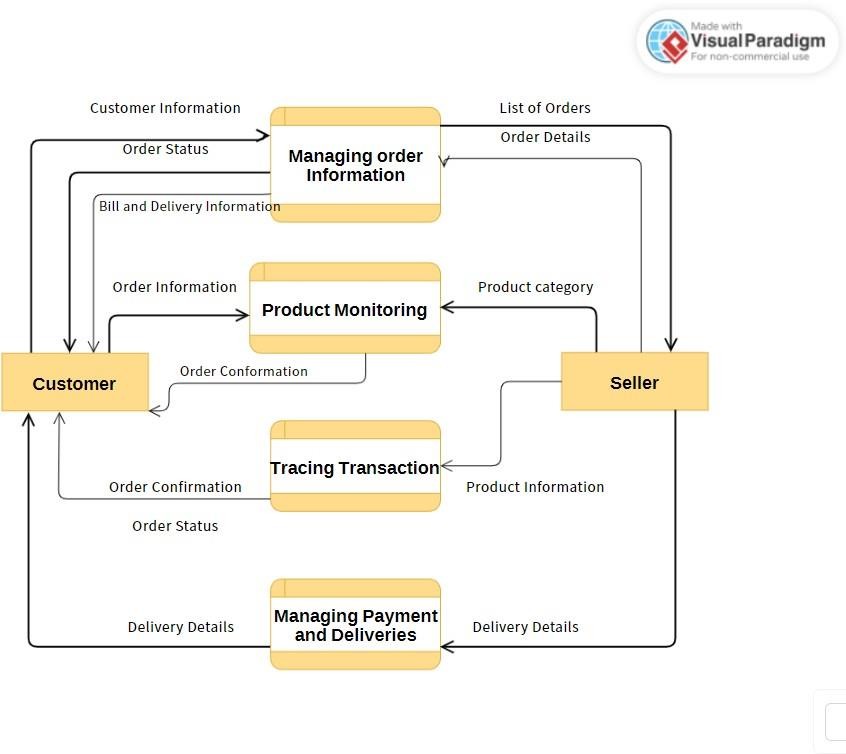
### Symbols:



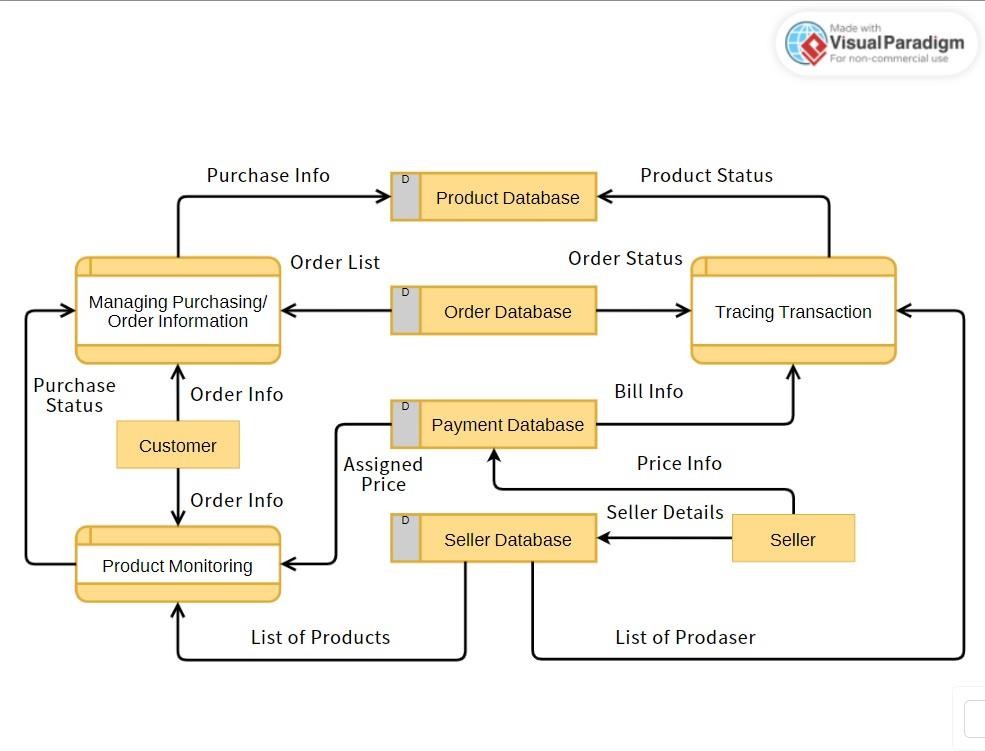
* **0th level DFD Admin Side:**



### 1st level DFD Admin Side:



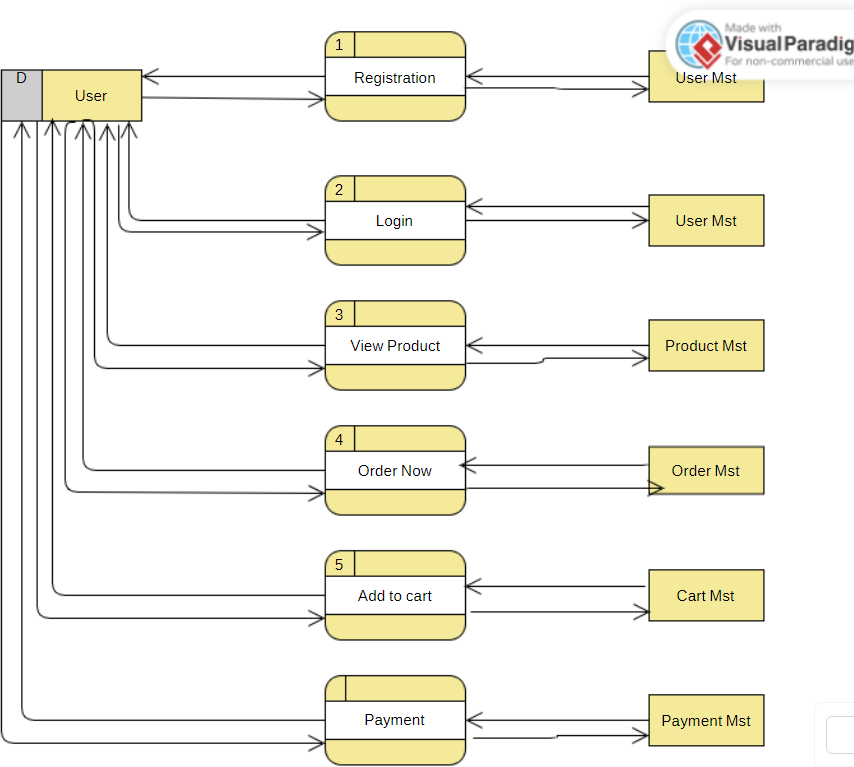
* **2nd level DFD Admin Side:**



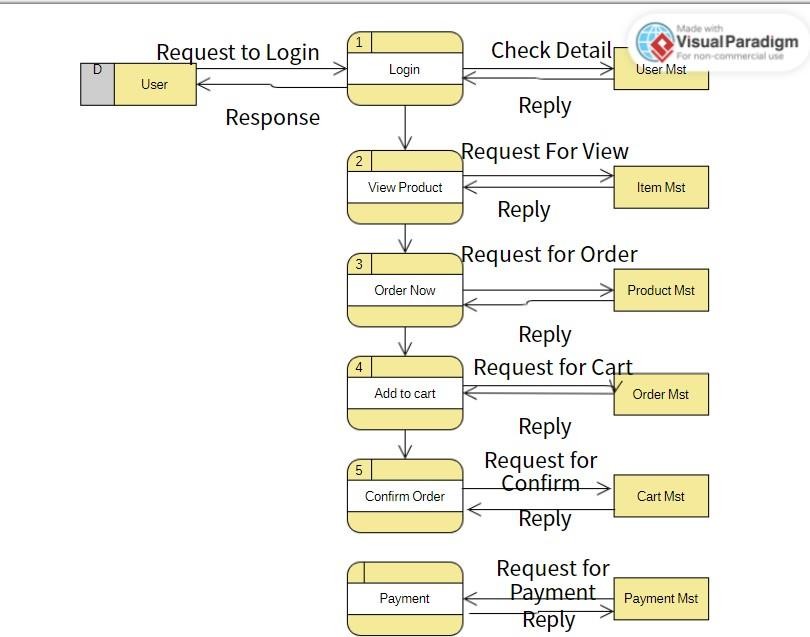
### 0th level DFD User Side:



* **1st level DFD User Side:**



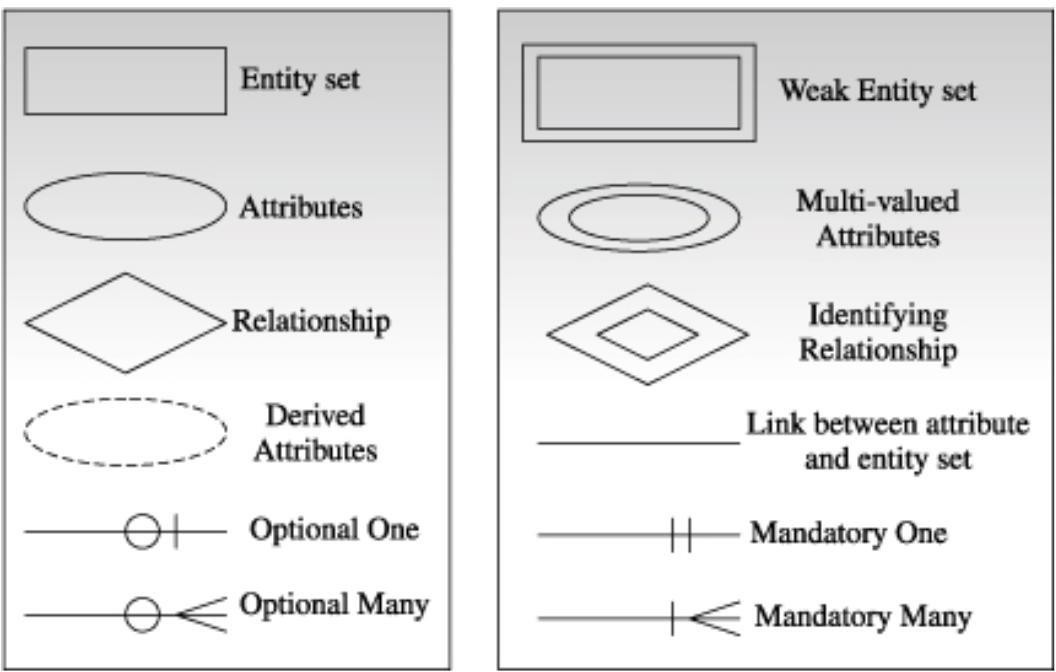
### 2nd level DFD User Side:



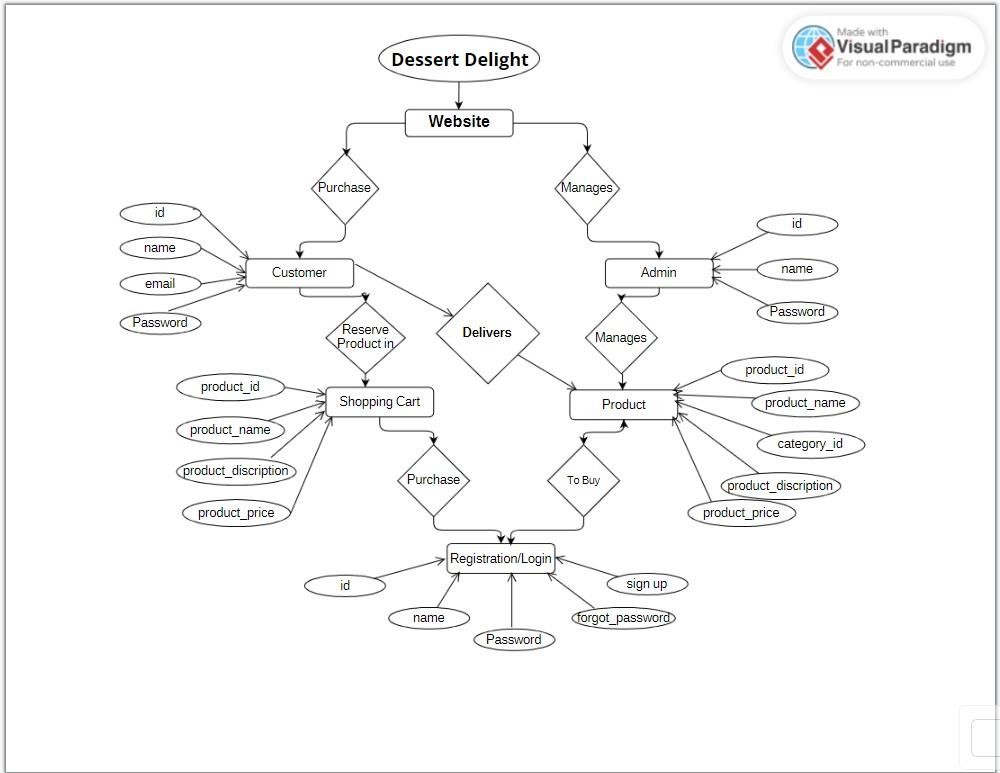
* + 1. **E-R Diagram:**

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

* + **Symbols:**



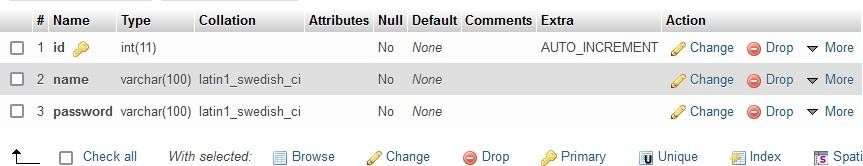
### E-R Diagram:



* + 1. **Modules Design:**
* **Admin side before Login:**
  + Login:
  + Forgot Password:
* **Admin Side after Login:**
  + Home:
  + Product Management:
    - Add Product:
    - Update Product:
    - Delete Product:
  + Category Management:
    - Add Category:
    - Update Category:
    - Delete Category:
  + View Product:
    - Add New Product:
  + View Category:
    - Add new Category:
  + Feedback:
  + FAQ:
    - User Reply:
  + Events:
    - Add:
    - Update:
    - Delete:
  + Offers:
    - Add:
    - Update:
    - Delete:
  + Logout:
* **User Side after Login:**
  + Home:
  + About:
  + Sign Up:
    - Registration:
  + Login:
    - Forgot Password:
* **User Side after Login:**
  + Home:
  + About:
  + Product:
    - All Category:
  + Services:
  + Gallery:
  + Contact:
  + Offers:
  + Events:
  + FAQ:
  + Spin wheel:
  + Feedback:

### Data Dictionary:

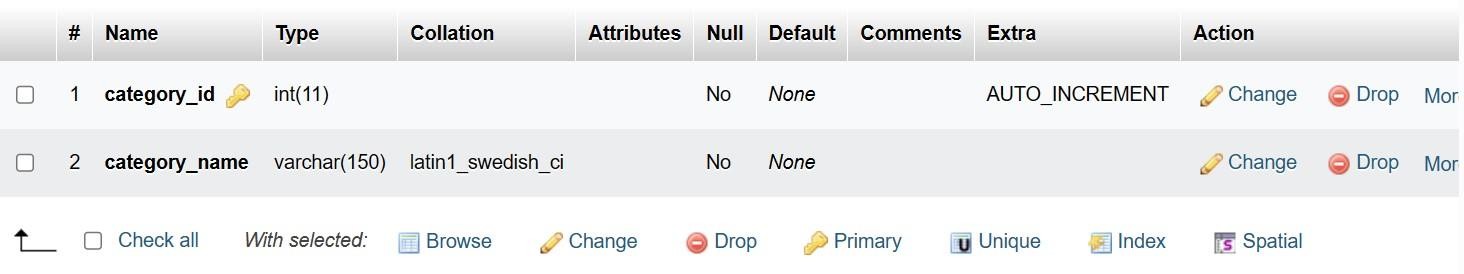
* **Admin Login:**



* **Cart:**



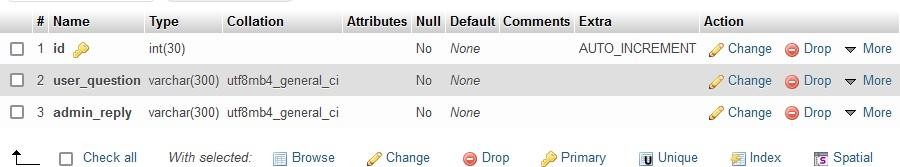
* **Category:**



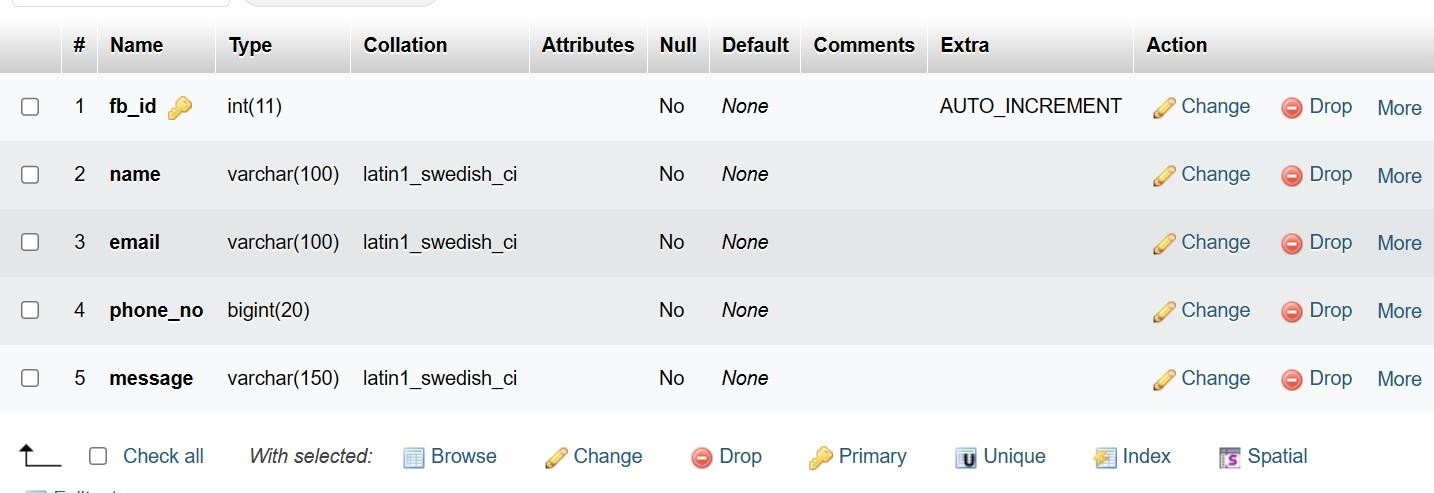
* **Event:**



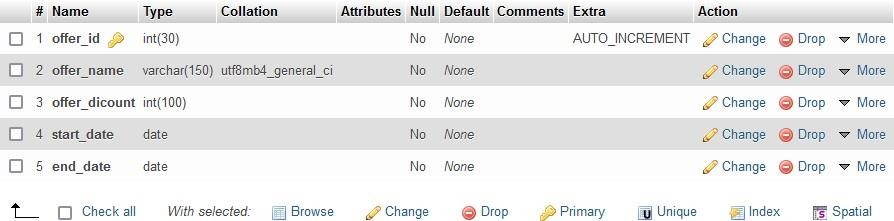
* **FAQ’S:**



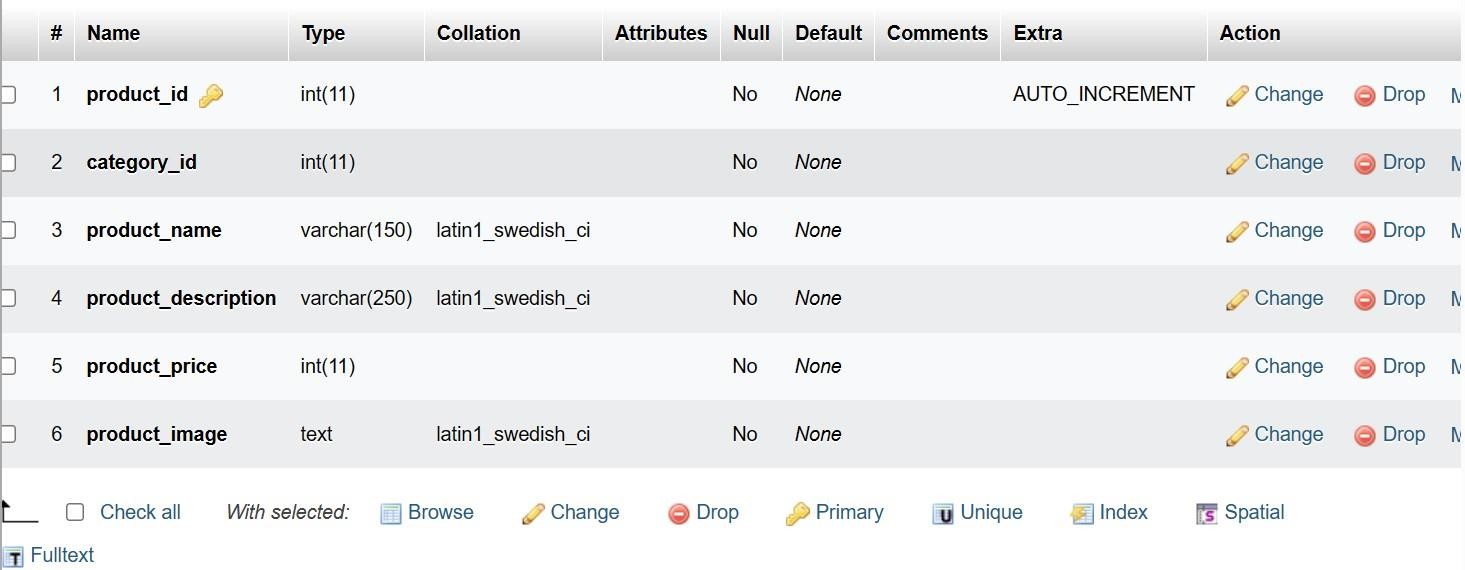
* **Feedback:**



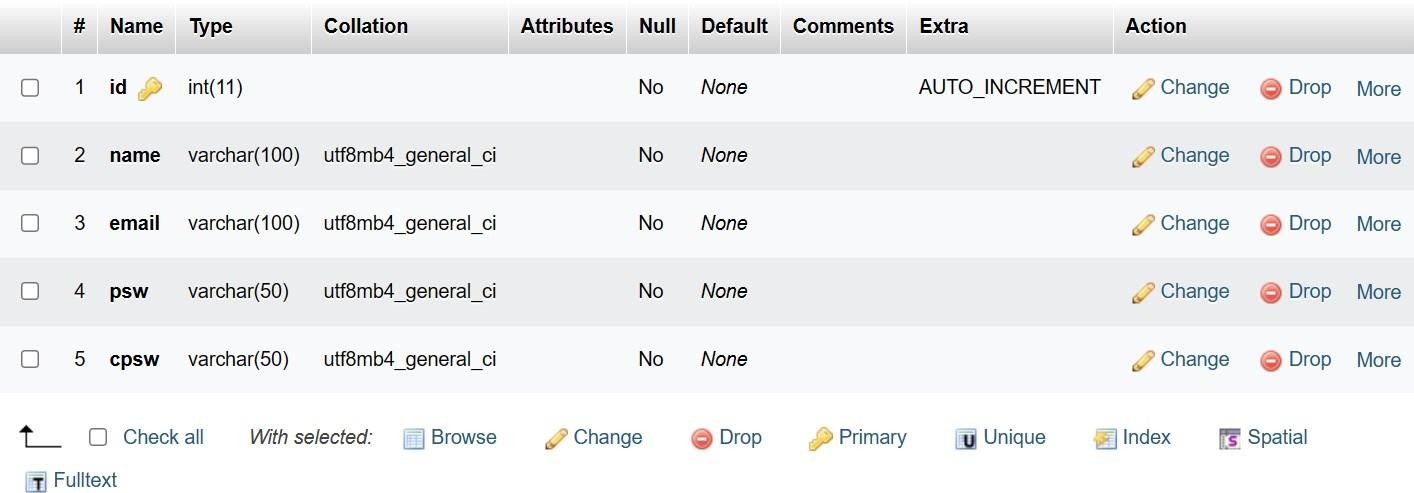
* **Offers:**



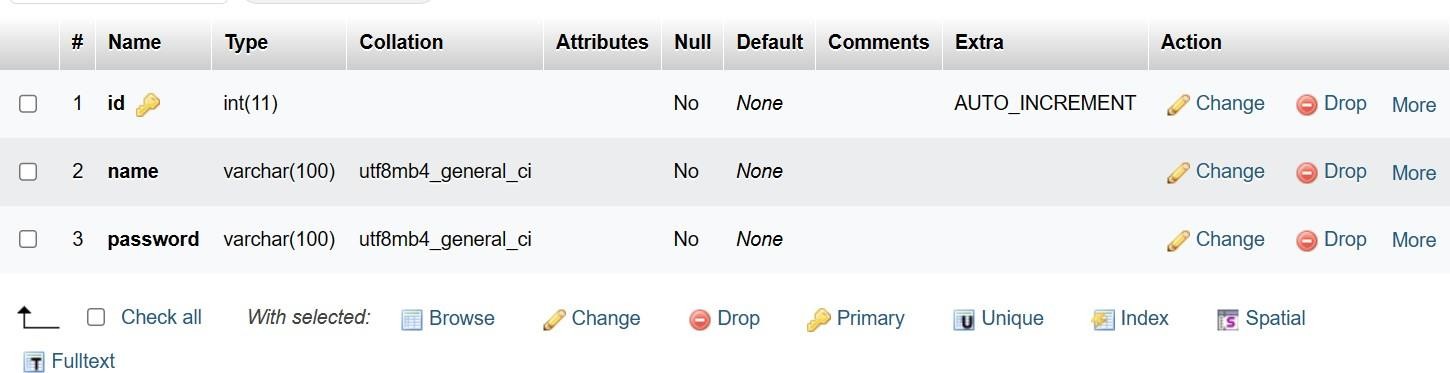
* **Product:**



* **Sign Up:**



* **User Login:**



Chapter 5: Screen- Layout and Testing

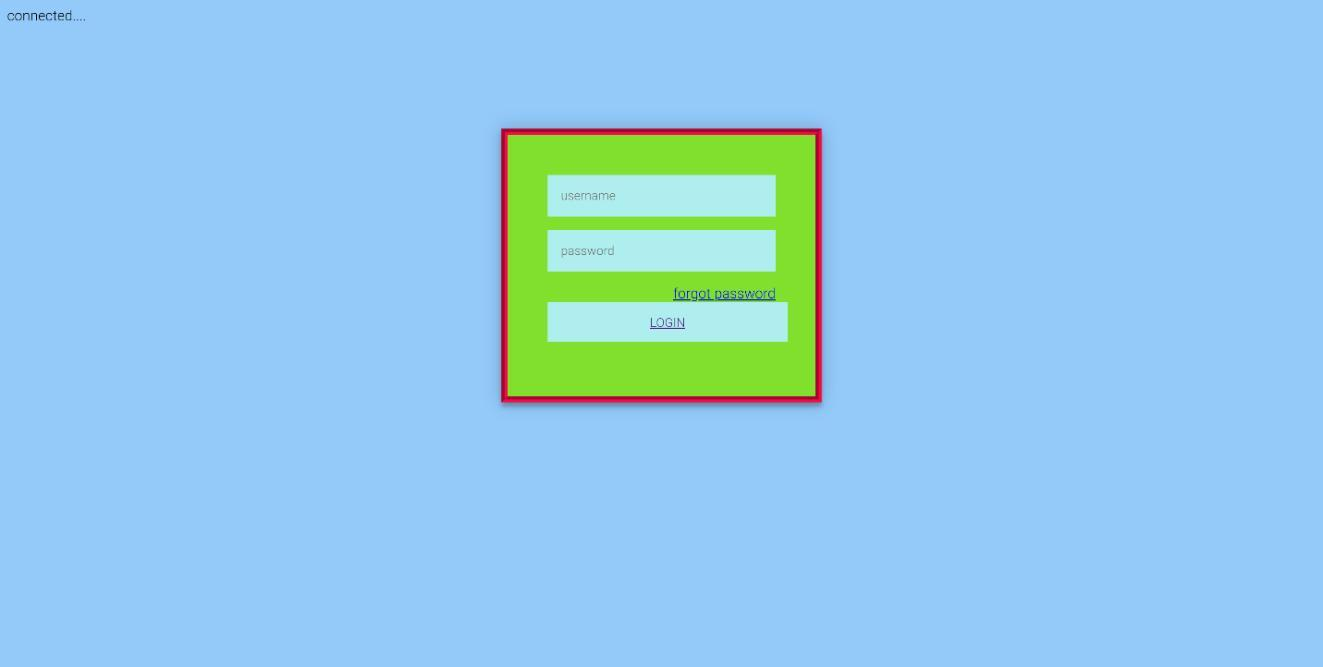
# CHAPTER 5: SCREENLAYOUT AND TESTING

### Screen Layout:

* + 1. **Admin Side:**

### Before Login:

* **Login:**



### Code File:

**<?php session\_start();**

**include "../conn.php"; if(isset($\_REQUEST["ad\_login"]))**

**{**

**$getuser=mysqli\_query($con,"select \* from adminlogin where name='".$\_REQUEST["unm"]."' AND password='".$\_REQUEST["psw"]."'");**

**$res=mysqli\_fetch\_row($getuser);**

**$nores=mysqli\_num\_rows($getuser); if($nores>0)**

**{**

**$\_SESSION["ad\_session"]=$res[1];**

**echo "<script>window.location='index1.php';</script>";**

**}**

**else**

**echo "<script>window.location='login.php';</script>";**

**}**

**else**

**echo "<script>window.location='login.php';</script>";**

**?>**

### After Login:

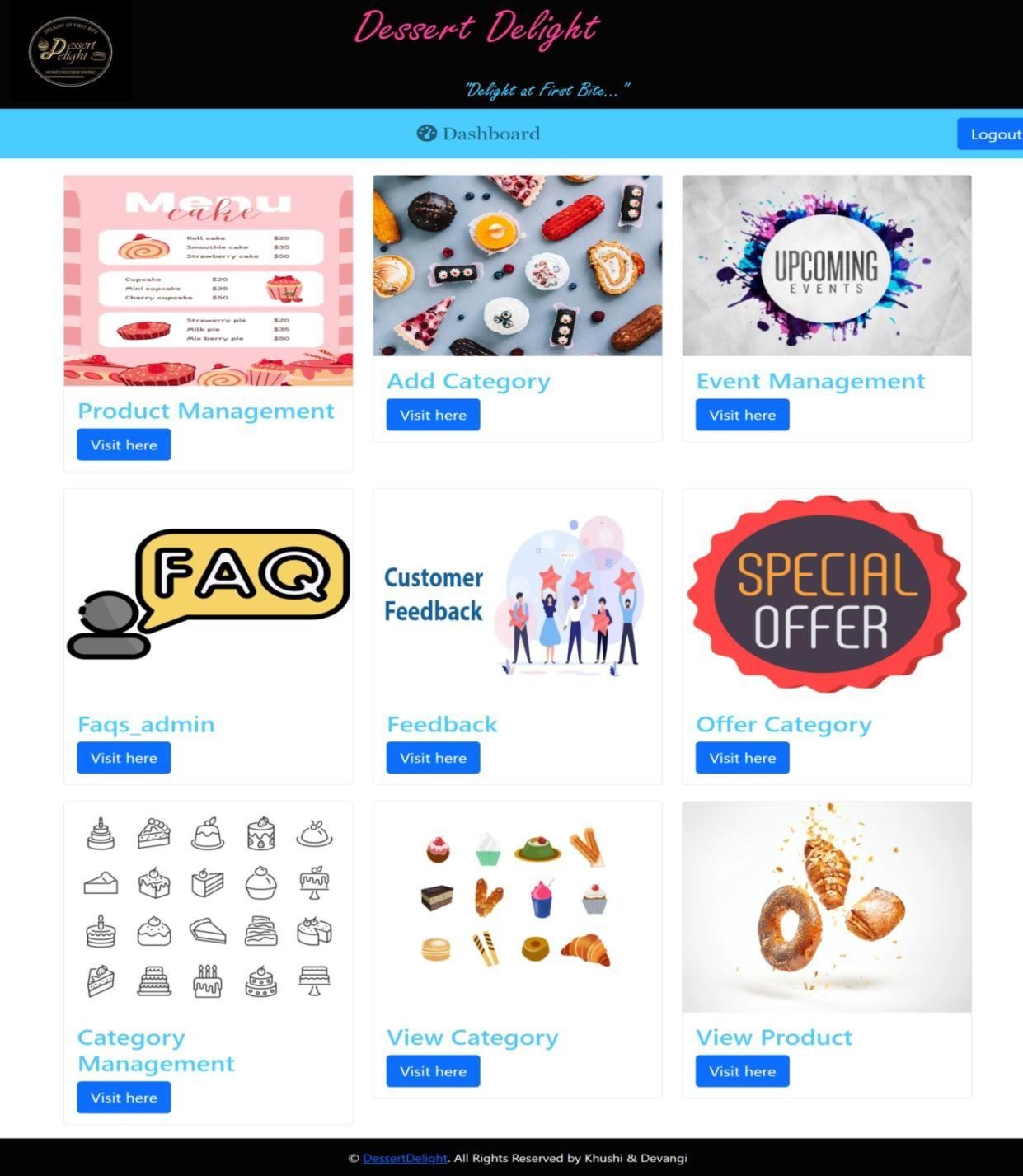
* **Header:**



### Footer:



* **Home:**



### Code File:

**<imgsrc="../images/addcate.jpg" class="card-img-top" alt="Card Image">**

**<div class="card-body">**

**<h2 class="card-title" style="color: #4accff;">Add 1Category</h2>**

**<a href="Categoryadd.php" class="btnbtn-primary btn-lg">Visit here</a>**

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

**<div class="col-md-4">**

**<div class="card">**

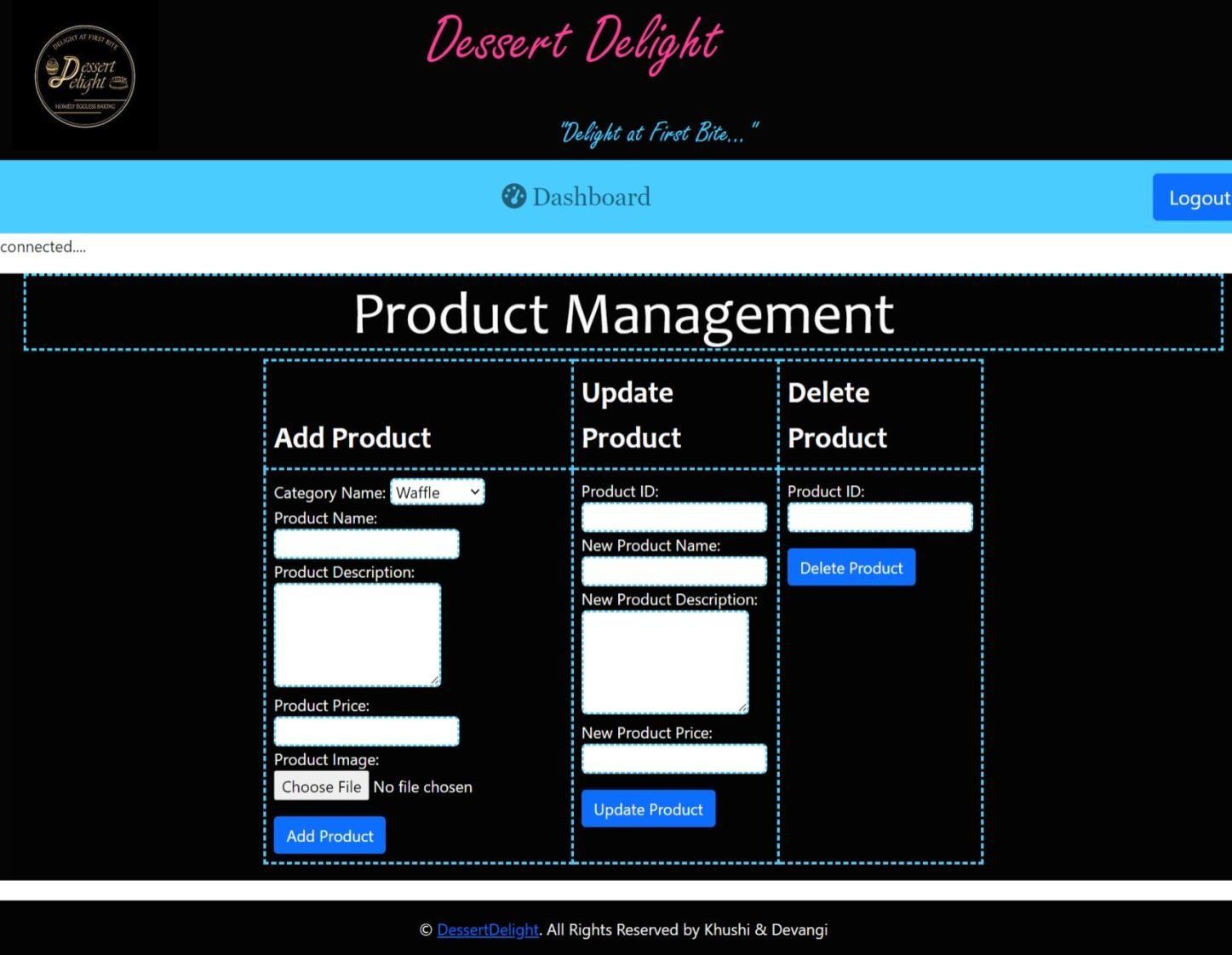
**<imgsrc="../images/event.jpg" class="card-img-top" alt="Card Image">**

**<div class="card-body">**

**<h2 class="card-title"style="color: #4accff;">Event Management</h2>**

**<a href="event\_management.php" class="btnbtn-primary btn-lg">Visit here</a>**

### Product Management.php:



* **Code File:**

### For Insert data:

**$query = "INSERT INTO product (category\_id,product\_name, product\_description, product\_price, product\_image ) VALUES ('$category\_id','$product\_name', '$product\_description', $product\_price, '$path')";**

**$result = mysqli\_query($con, $query);**

### For Update data:

**mysqli\_query($con, "UPDATE product SET product\_name = '$new\_product\_name', product\_description = '$new\_product\_description', product\_price = '$new\_product\_price' WHERE product\_id = '$product\_id'");**

### For Delete data:

**mysqli\_query($con, "DELETE FROM product WHERE product\_id = '$product\_id'");**

### Add Category.php:



* **Code File:**

**<?php**

**if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {**

**if (isset($\_POST['submit'])) {**

**$category\_name = $\_POST['category\_name'];**

**$category\_description = $\_POST['category\_description'];**

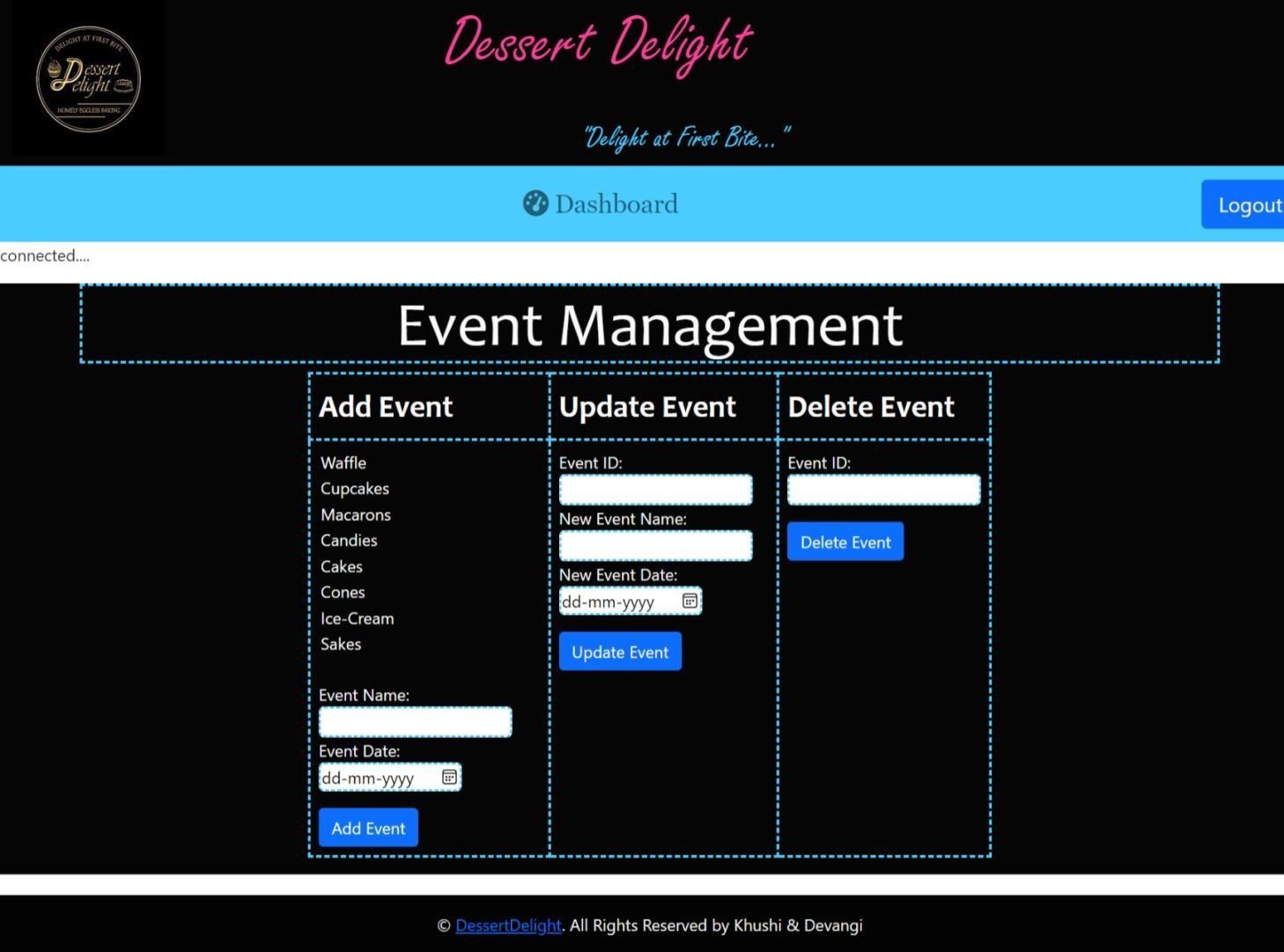
**echo "<h4 class='text-success'>Category added successfully!</h4>";**

**}**

**}**

**?>**

### Event Management.php:



* **Code File:**

### For Insert Data:

**$query = "INSERT INTO event (event\_name, event\_date) VALUES ('$event\_name', '$event\_date')";**

**$result = mysqli\_query($con, $query);**

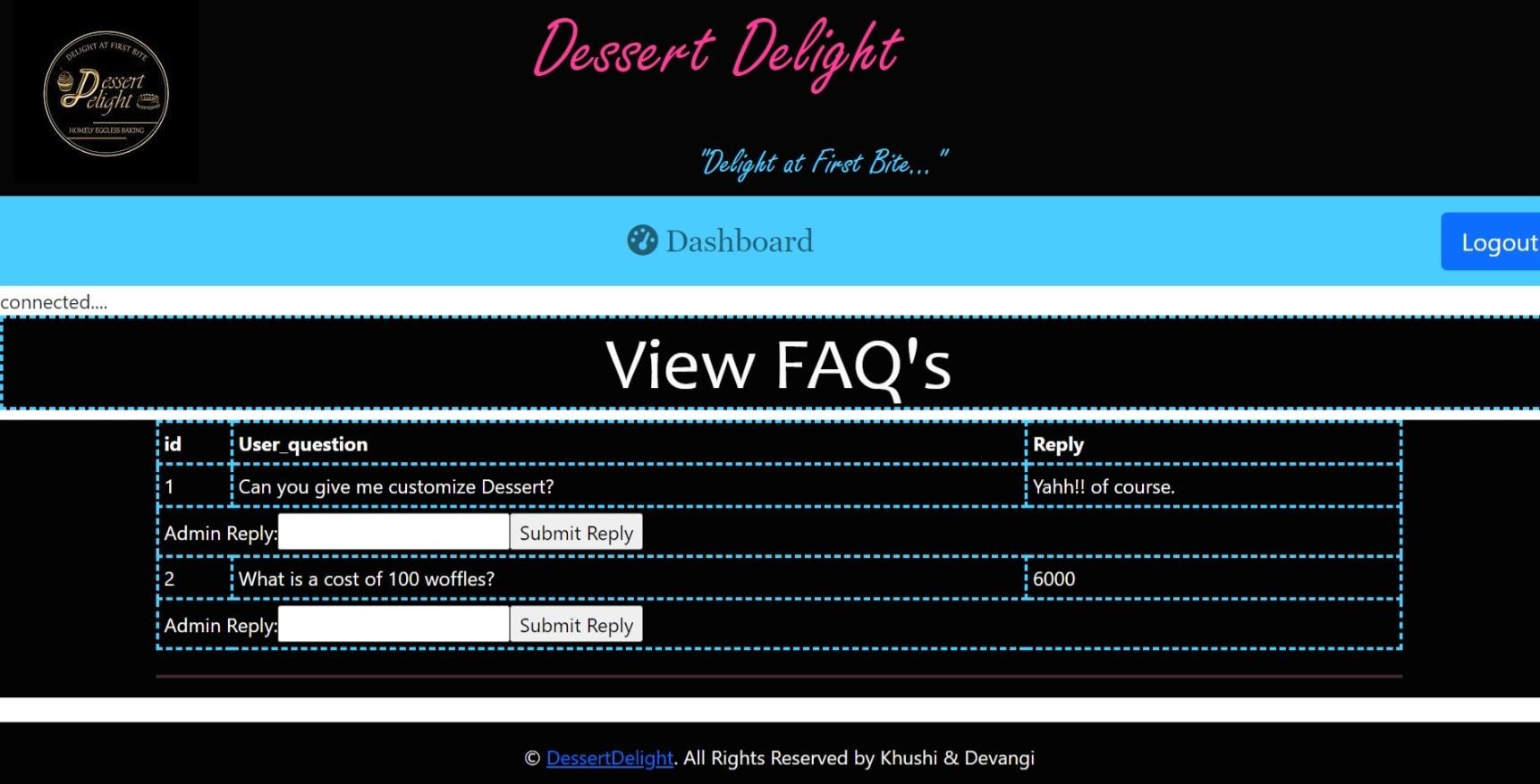
### For Update Data:

**mysqli\_query($con, "UPDATE event SET event\_name = '$new\_event\_name', event\_date = '$new\_event\_date' WHERE event\_id = '$event\_id'");**

### For Delete Data:

**mysqli\_query($con, "DELETE FROM event WHERE event\_id = '$event\_id'");**

### FAQ’S Admin.php:



* + **Code File:**

**if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {**

**if (isset($\_POST['submit\_add\_event'])) {**

**$admin\_reply = $\_POST['admin\_reply'];**

**$faq\_id = $\_POST['faq\_id']; // Add this line to get the FAQ id**

**$query = "UPDATE faqs SET admin\_reply = '$admin\_reply' WHERE id =**

**$faq\_id";**

**$result = mysqli\_query($con, $query); if ($result) {**

**echo "<script>window.location='faqs\_admin.php'</script>"; }}?>**

**<?php**

**$query = "SELECT \* FROM faqs"; // Adjust your query**

**$result = mysqli\_query($con, $query);**

**if ($result) { while ($row = mysqli\_fetch\_array($result)) {**

**echo "<td style='border: 3px dashed rgb(74, 204, 255);'>" . $row[0] . "</td>"; action='{$\_SERVER['PHP\_SELF']}'>";**

**echo "<input type='hidden' name='faq\_id' value='" . $row['id'] .**

**"'>";**

**echo "<label for='admin\_reply'>Admin Reply:</label>";**

**}} else {**

**echo "Error fetching faqs: " .mysqli\_error($con); }**

### Feedback.php:



* + **Code File:**

**<?php**

**. "</td>";**

**. "</td>";**

**. "</td>";**

**. "</td>";**

**. "</td>";**

**$query = "SELECT \* FROM feedback"; // Adjust your query**

**$result = mysqli\_query($con, $query); if ($result) {**

**while ($row = mysqli\_fetch\_array($result)) {**

**echo "<tr style='border: 3px dashed rgb(74, 204, 255);'>";**

**echo "<td style='border: 3px dashed rgb(74, 204, 255);'>" . $row[0] echo "<td style='border: 3px dashed rgb(74, 204, 255);'>" . $row[1] echo "<td style='border: 3px dashed rgb(74, 204, 255);'>" . $row[2] echo "<td style='border: 3px dashed rgb(74, 204, 255);'>" . $row[3] echo "<td style='border: 3px dashed rgb(74, 204, 255);'>" . $row[4]**

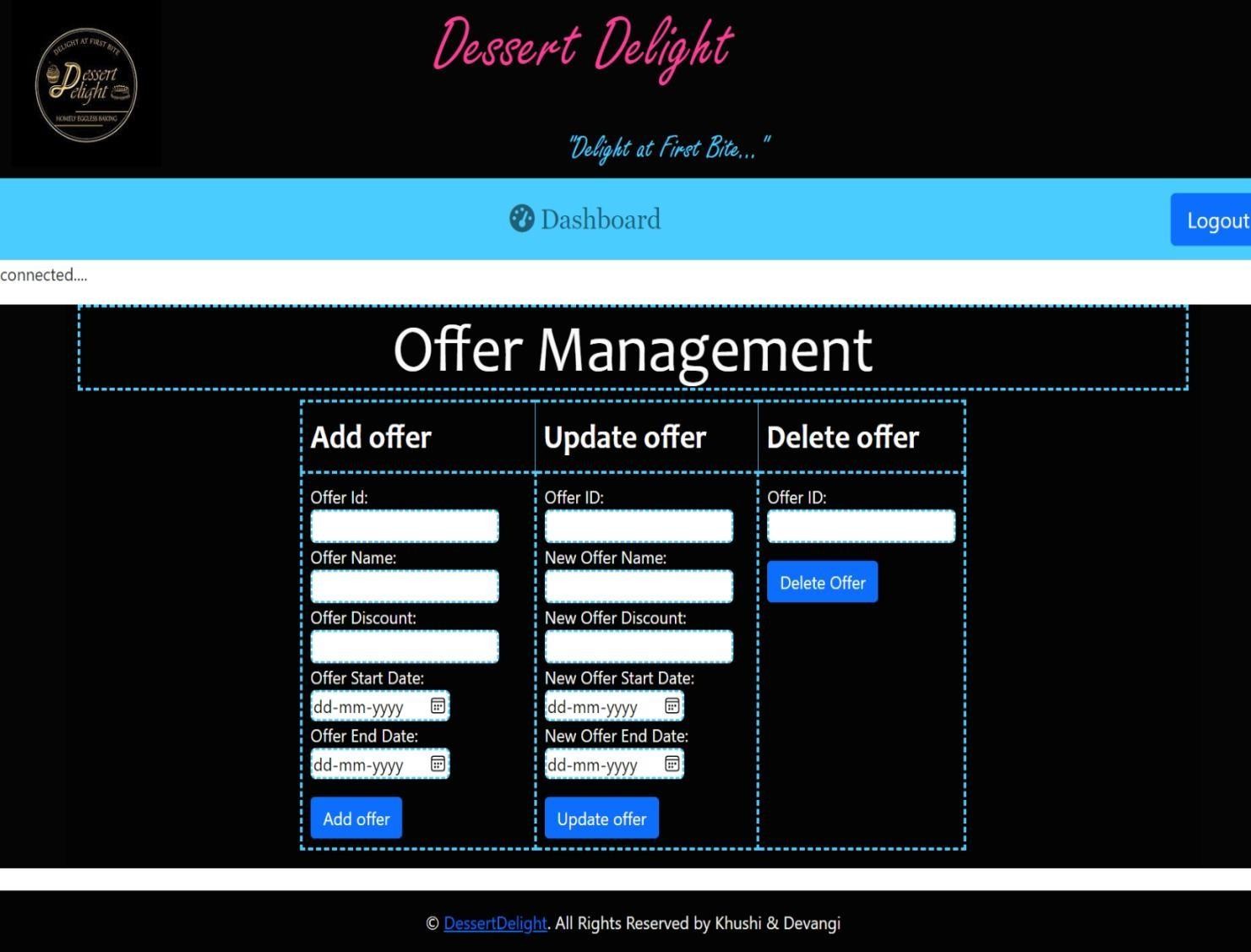
**echo "</tr>"; }**

**} else {**

**echo "Error fetching feedback: " .mysqli\_error($con);**

**} ?>**

### Offer Category.php:



* + **Code File:**

### For Insert Query:

**$query = "INSERT INTO offers (offer\_id, offer\_name, offer\_dicount, start\_date, end\_date) VALUES ('$offer\_id', '$offer\_name', '$offer\_dicount', '$start\_date', '$end\_date')";**

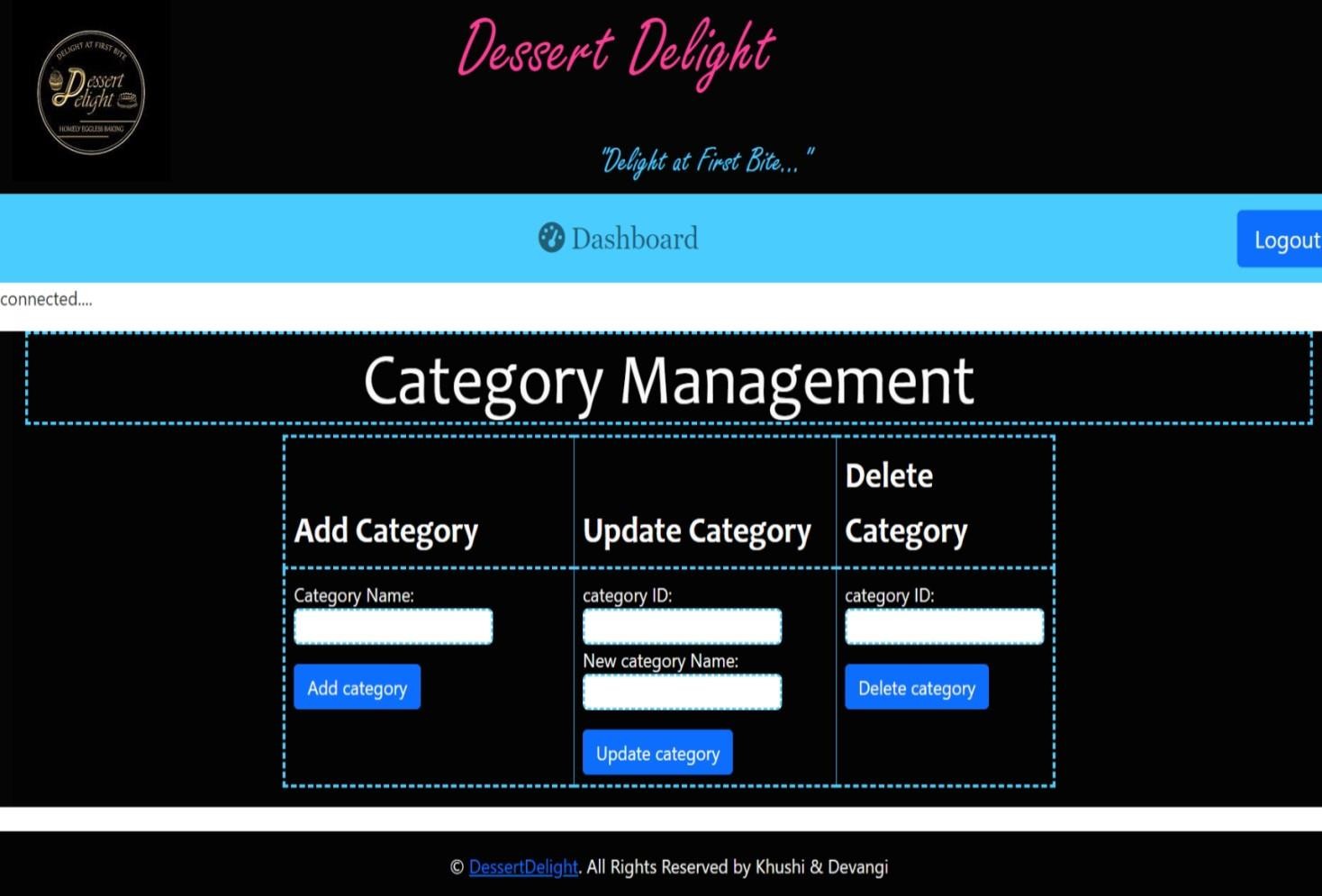
### For Update Query:

**mysqli\_query($con, "UPDATE offers SET offer\_name = '$new\_offer\_name', offer\_dicount = '$new\_offer\_dicount', start\_date = '$new\_start\_date', end\_date = '$new\_end\_date' WHERE offer\_id = '$new\_offer\_id'");**

### For Delete Query:

**mysqli\_query($con, "DELETE FROM offers WHERE offer\_id = '$offer\_id'");**

### Category Management.php:



* + **Code File:**

### For Insert Query:

**$query = "INSERT INTO category (category\_name) VALUES ('$category\_name')";**

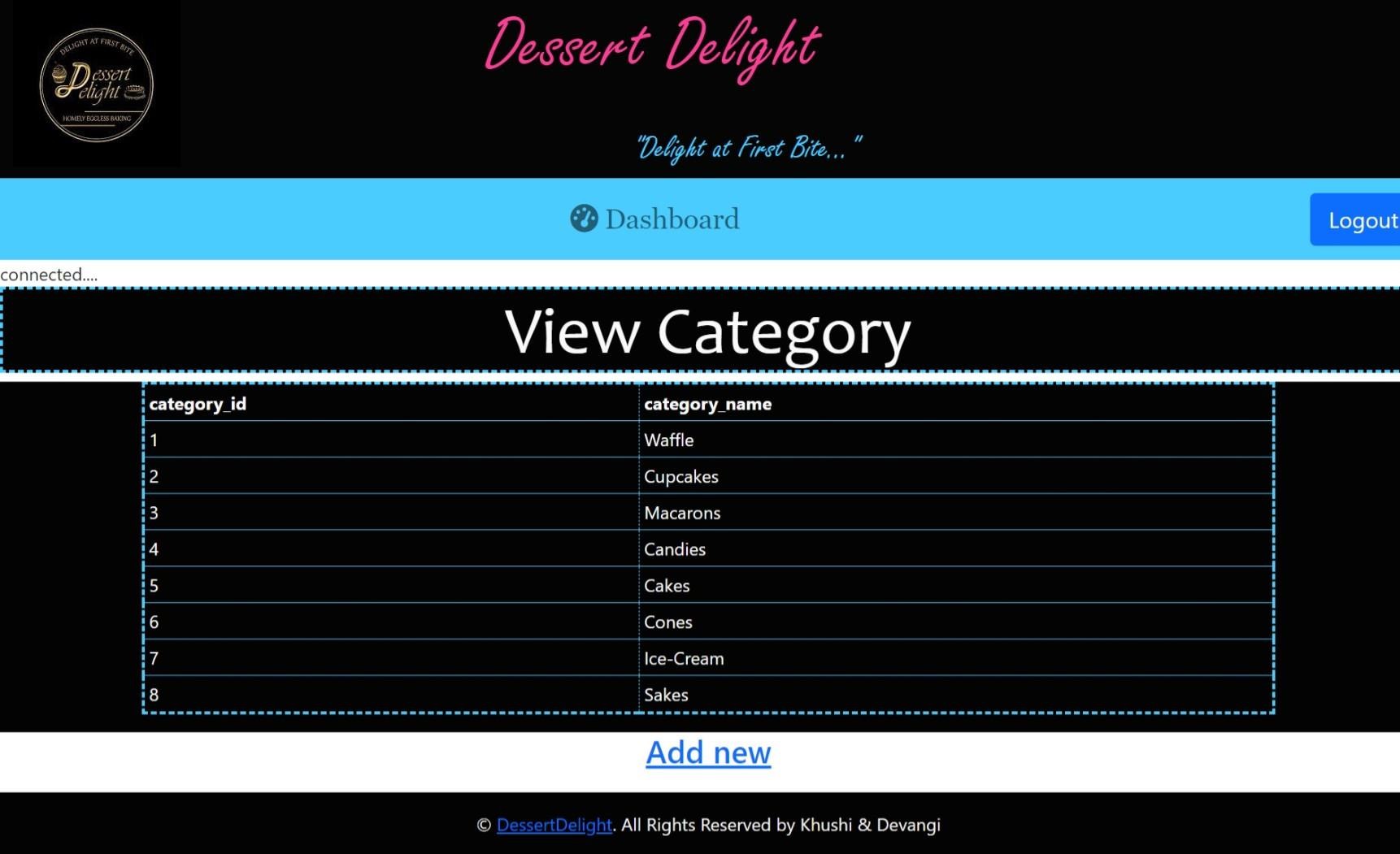
### For Update Query:

**mysqli\_query($con, "UPDATE category SET category\_name = '$new\_category\_name' WHERE category\_id = '$category\_id'")**

### For Delete Query:

**mysqli\_query($con, "DELETE FROM category WHERE category\_id = '$category\_id'");**

### View Category.php:



* + **Code File:**

**<?php**

**echo "<tr>";**

**$no = mysqli\_num\_fields($result); for ($i = 0; $i< $no; $i++) {**

**$fieldinfo = mysqli\_fetch\_field\_direct($result, $i);**

**echo "<th style='border: 1px dashed rgb(74, 204, 255);'>" . $fieldinfo-**

**>name . "</th>";**

**}**

**echo "<tr>";**

**while ($row = mysqli\_fetch\_array($result)) { echo "<tr>";**

**} ?>**

### View Product.php:



* + **Code File:**

**<?php**

**echo "<tr>";**

**$no = mysqli\_num\_fields($result); for ($i = 0; $i< $no; $i++) {**

**$fieldinfo = mysqli\_fetch\_field\_direct($result, $i);**

**echo "<th style='border: 1px dashed rgb(74, 204, 255);'>" . $fieldinfo-**

**>name . "</th>";**

**}**

**echo "<tr>";**

**"</td>";**

**while ($row = mysqli\_fetch\_array($result)) { echo "<tr>";**

**echo "<td style='border: 1px dashed rgb(74, 204, 255);'>" . $row[0] .**

**echo "<td style='border: 1px dashed rgb(74, 204, 255);'><imgsrc='../" .**

**$row[5] . "' height=50px width=50px></td>"; echo "</tr>";**

**}**

**?>**

### LogOut.php:



* + **Code File:**

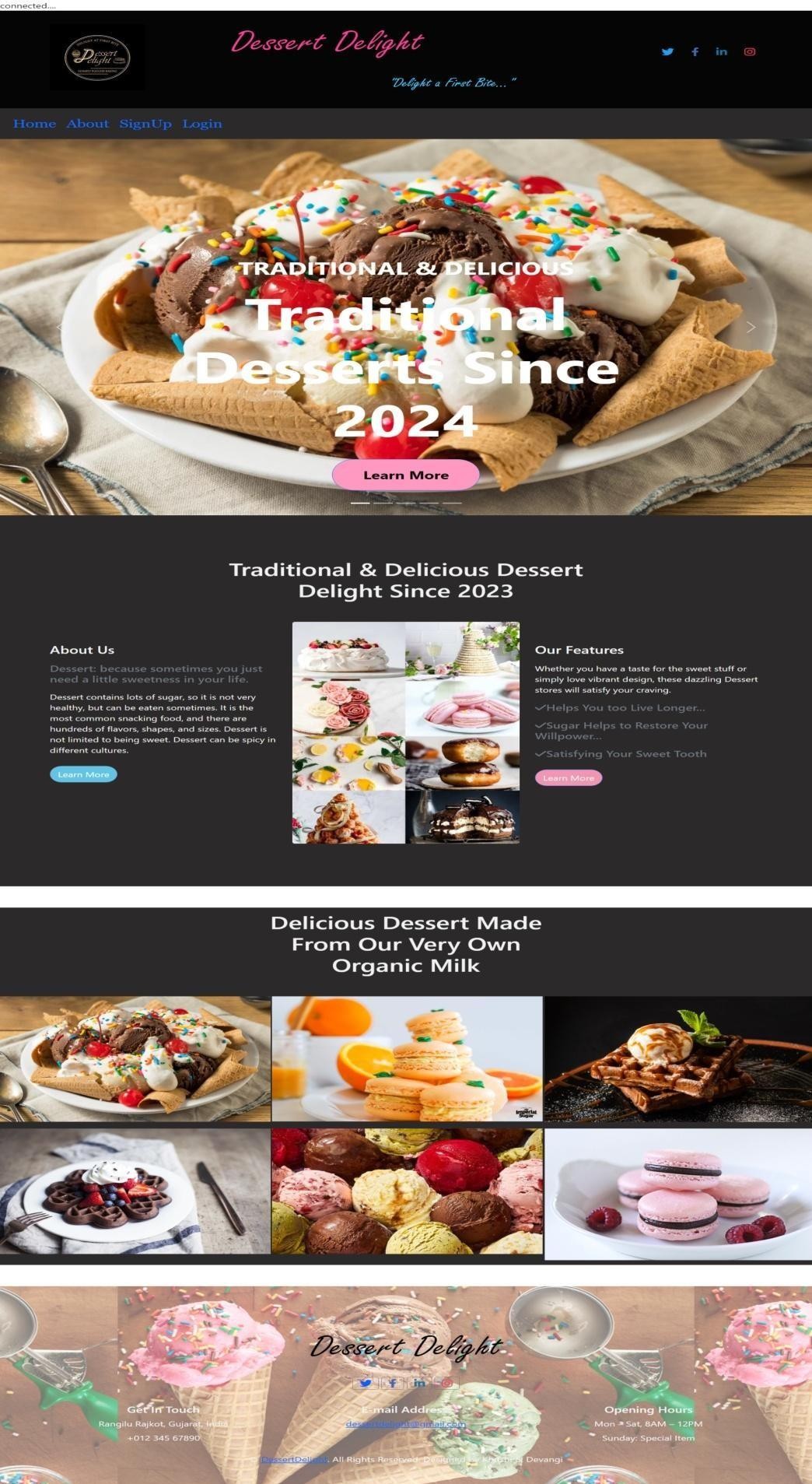
**<?php session\_start(); session\_unset(); session\_destroy();**

**?>**

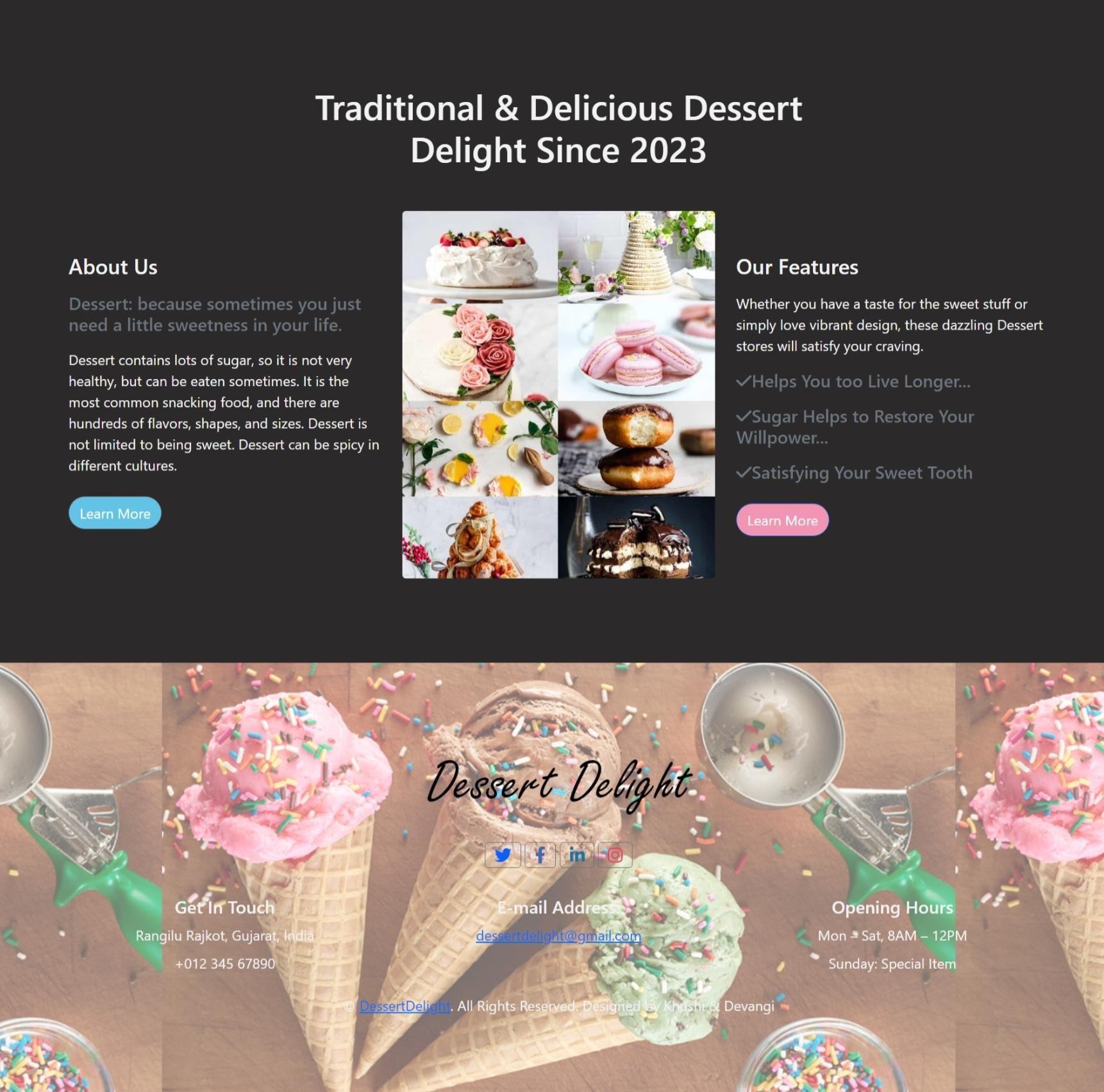
* + 1. **User Side:**

### Before Login:

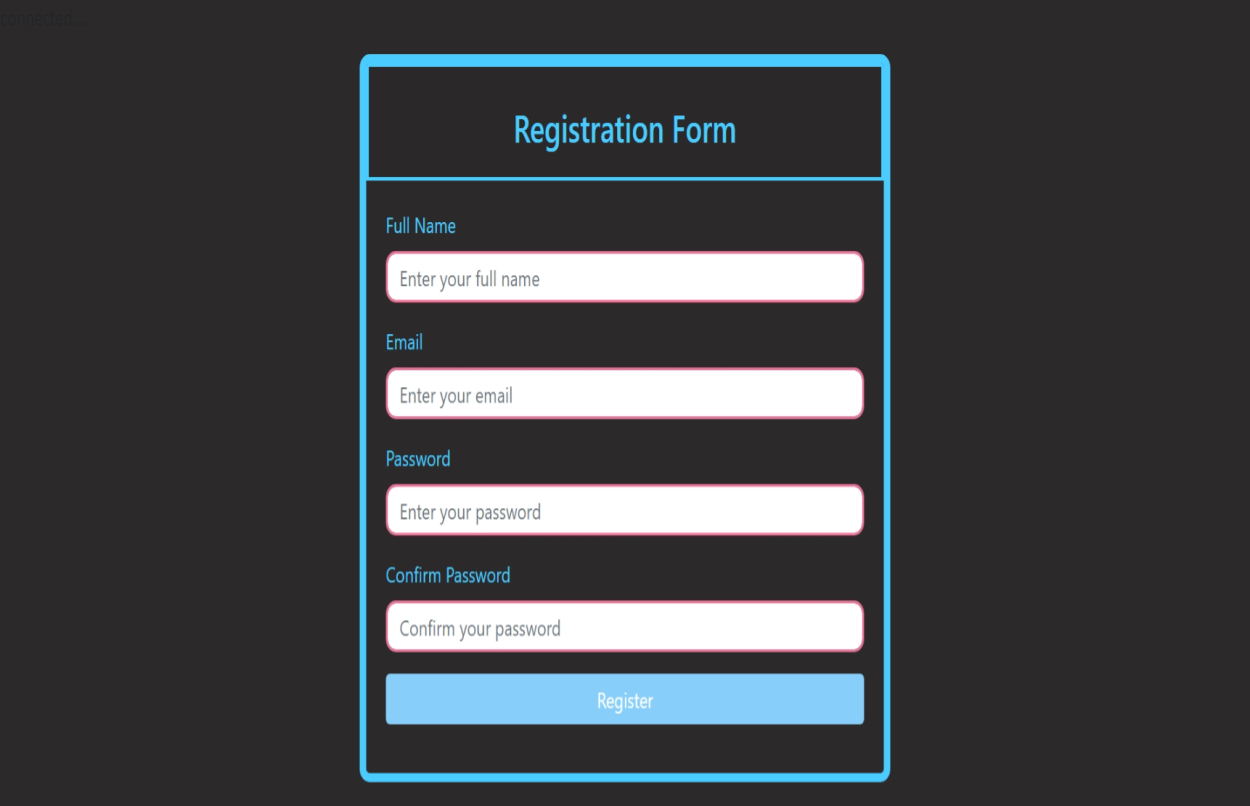
* + **Index.php:**



### About.php:



* + **SignUp.php:**



### Code File:

**<?php include('conn.php'); error\_reporting(E\_ALL); ini\_set('display\_errors', 1);**

**if (isset($\_POST["submit"])) {**

**$name = $\_POST["name"];**

**$email = $\_POST["email"];**

**$psw = $\_POST["psw"];**

**$cpsw = $\_POST["cpsw"];**

**$signup\_query = "INSERT INTO signup (name, email, psw, cpsw) VALUES ('$name', '$email', '$psw', '$cpsw')";**

**if (mysqli\_query($con, $signup\_query)) {**

**echo "Data inserted into signup table successfully.<br>";**

**} else {**

**echo "Error inserting data into signup table: " .mysqli\_error($con) . "<br>";**

**}**

**$userlogin\_query = "INSERT INTO userlogin (name, password) VALUES ('$name', '$psw')";**

**if (mysqli\_query($con, $userlogin\_query)) {**

**echo "Data inserted into userlogin table successfully.<br>";**

**} else {**

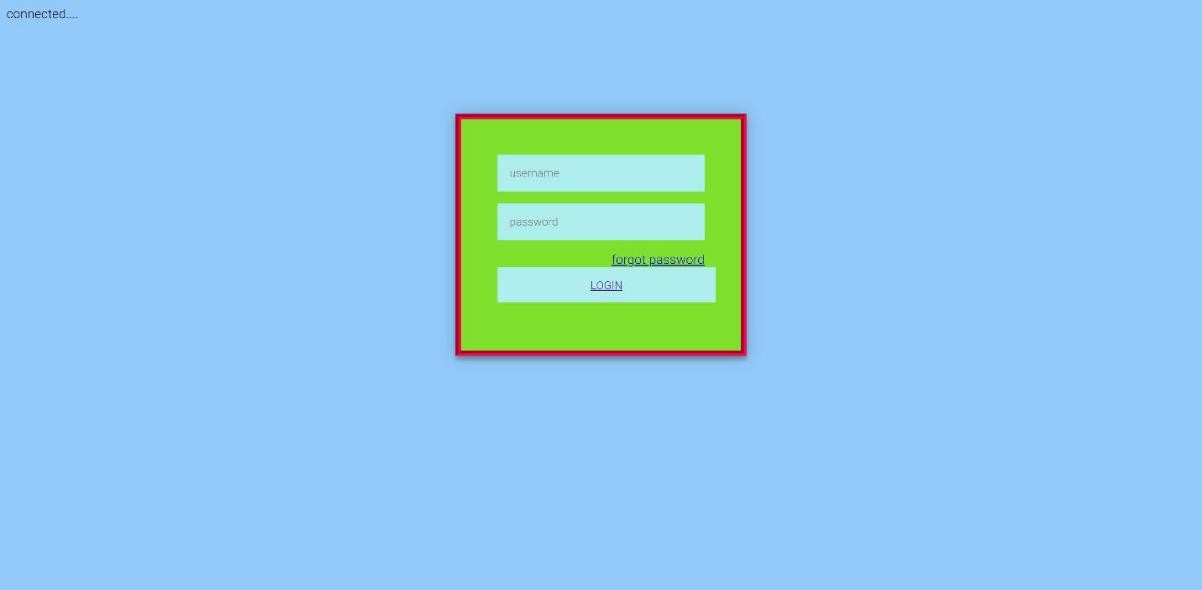
**echo "Error inserting data into userlogin table: " .mysqli\_error($con) . "<br>";**

**}**

**}**

**?>**

### Login:



* + **Code File:**

**<?php include "conn.php"; ?>**

**<?php if(isset($\_REQUEST["submit"])) {**

**$name = $\_REQUEST["name"];**

**$psw = $\_REQUEST["password"];**

**mysqli\_query($con, "INSERT INTO userlogin (name, password) VALUES ('$name', '$psw')");**

**echo "<script>window.location='afterlogin.php'</script>";**

**}**

**?>**

**<?php session\_start();**

**include "../conn.php"; if(isset($\_POST["ad\_login"])) {**

**$name = $\_POST["name"];**

**$password = $\_POST["password"];**

**// You should perform validation and security checks here**

**$getuser = mysqli\_query($con, "SELECT \* FROM userlogin WHERE name='$name' AND password='$password'");**

**$res = mysqli\_fetch\_row($getuser);**

**$nores = mysqli\_num\_rows($getuser); if ($nores> 0) {**

**$\_SESSION["ad\_session"] = $res[1];**

**echo "<script>window.location='afterlogin.php';</script>";**

**} else {**

**echo "Login failed.";**

**}**

**} else {**

**echo "<script>window.location='index.php';</script>";**

**}**

**?>**

### After Login:

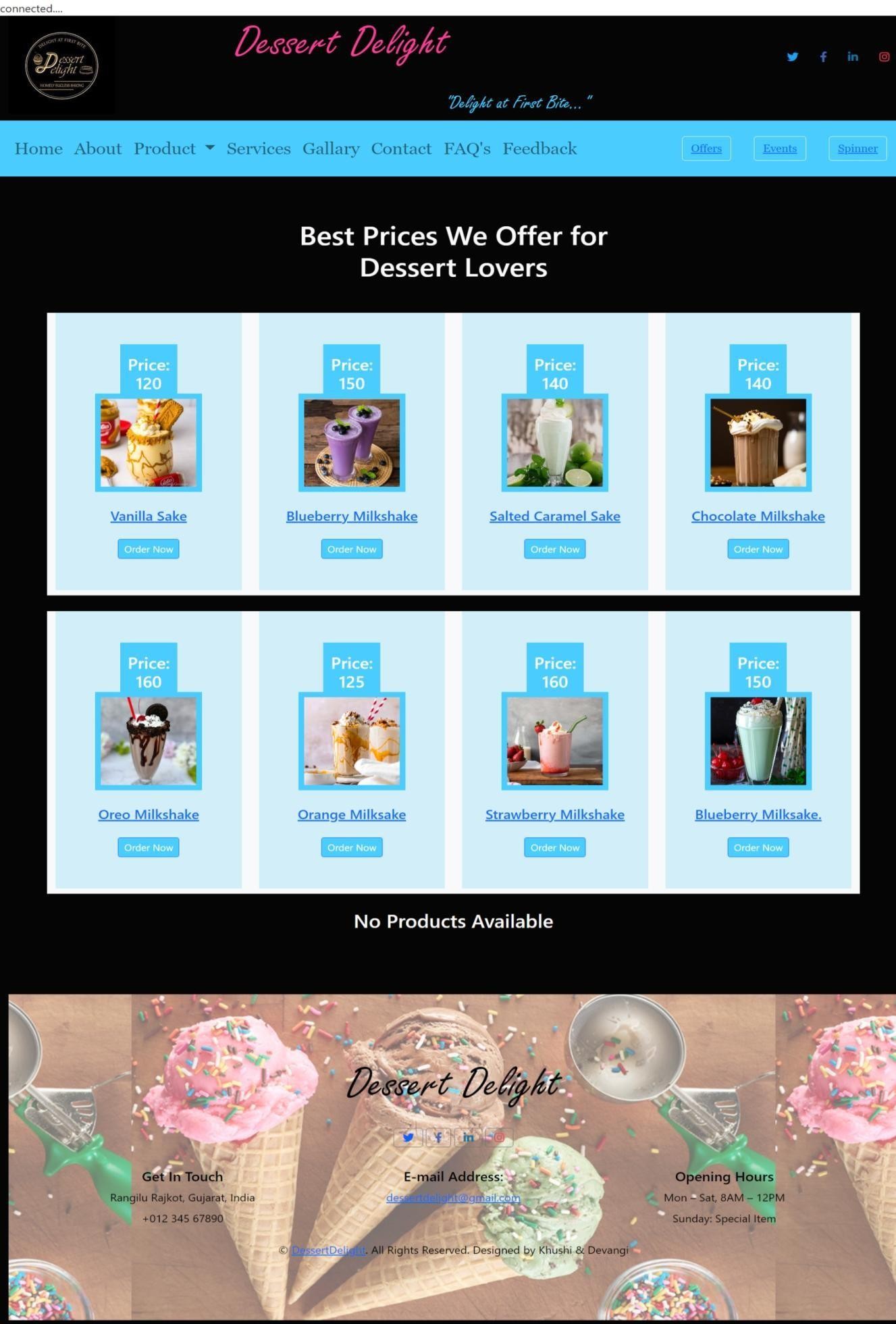
* + **Header:**



### Footer:



* + **Product:**



### Code File:

**<?php**

**include("header.php");**

**$pro=mysqli\_query($con,"select \* from product where category\_id=".$\_REQUEST["cid"]);**

**?>**

**<div class="container-fluid py-3" style="background-color: #050505;color: #fcfafa;">**

**<div class="container py-5">**

**<?php**

**if(mysqli\_num\_rows($pro)>0)**

**{**

**?>**

**<div class="row justify-content-center">**

**<div class="col-lg-5">**

**<h1 class="section-title position-relative text-center mb-5">Best Prices We Offer for Dessert Lovers</h1>**

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

**<div class="row ">**

**<?php**

**while ($rpro = mysqli\_fetch\_array($pro)) {**

**?>**

**<div class="col-lg-3 col-md-6 mb-4 pb-2 bg-light" >**

**<div class="product-item d-flex flex-column hover-overlay align-items-center text- center squer py-5 px-3" style="background-color: hsla(195, 83%, 58%, 0.2);" >**

**<div class=" squer-circle mt-n5 py-3" style="height:75px; width: 80px; background-color: rgb(74, 204, 255);">**

**<h4 class="font-weight-bold text-white mb-0">Price: <?php echo $rpro[4]; ?></h4>**

**</div>**

**<div class="position-relative squer-circle mt-n3 mb-4 p-2" style="width: 150px; height: 150px; background-color: rgb(74, 204, 255);">**

**<a href="ProductDetail.php?id=<?php echo $rpro[0]; ?>">**

**<img class="squer-circle w-100 h-100" src="<?php echo $rpro[5]; ?>" alt="Card image cap" style="object-fit: cover;">**

**</a></div>**

**<!-- Other product details ... -->**

**<h5 class="font-weight-bold mb-4">**

**<a href="ProductDetail.php?id=<?php echo $rpro[0]; ?>"><?php echo $rpro[2];**

**?></a>**

**</h5>**

**<a href="cart.php?id=<?php echo $rpro[0]; ?>" class="btnbtn-smbtn-primary" style="background-color: rgb(74, 204, 255);">Order Now</a></div></div>**

**<?php**

**}?>**

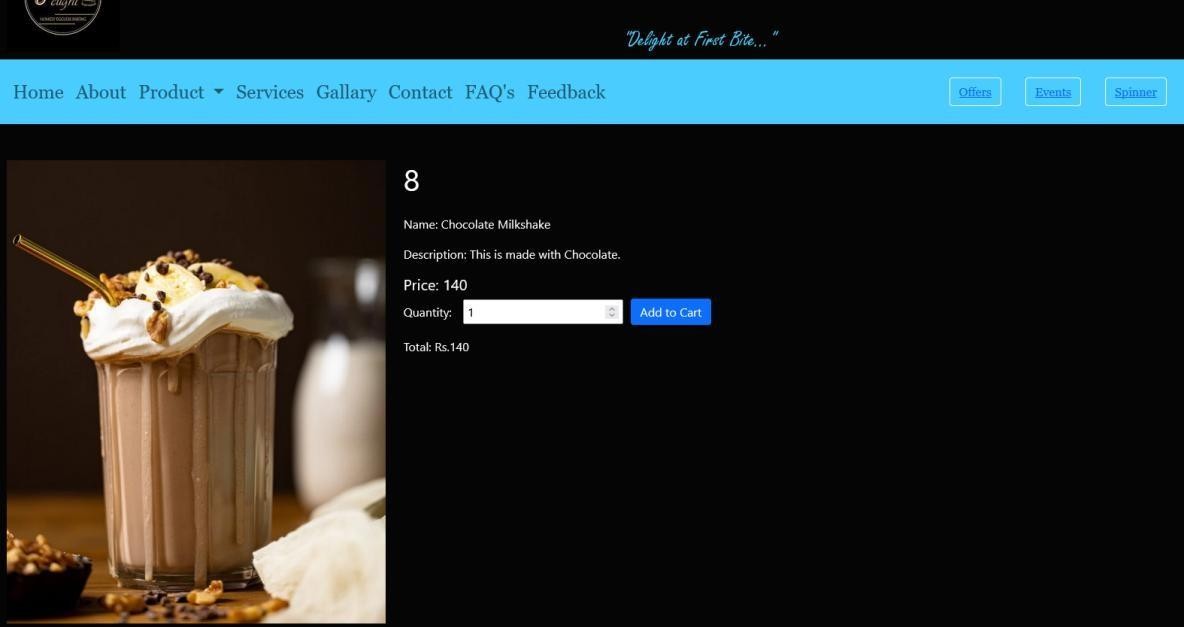
**<h3 class="title-w3ls text-center text-bl mb-5">No Products Available</h3>**

**<?php**

**}**

**?></div></div>**

### Cart.php:



* + **Code File:**

**<?php**

**if(isset($\_REQUEST["id"])) {**

**$productId = $\_REQUEST["id"];**

**$pro = mysqli\_query($con, "select \* from product where product\_id=$productId");**

**if($pro) {**

**$r = mysqli\_fetch\_row($pro);**

**if($r) { echo '<div class="col-lg-4 mb-8 mb-lg-0 product-image-container">'; echo '<imgsrc="' . $r[5] . '" alt="Product Image" class="img-fluid"**

**style="max-width: 100%; height: auto;">';**

**echo '<h1 class="mb-4">' . $r[1] . '</h1>';**

**echo '<p>Name: ' . $r[2] . '</p>';**

**echo '<p>Description: ' . $r[3] . '</p>'; echo '<h5>Price: ' . $r[4] . '</h5>';**

**echo '<form action="payment.php" method="GET" style=" display: flex; align-items: center; ">';**

**echo '<input type="hidden" name="pro\_id" value="' . $productId . '">'; echo '<input type="hidden" name="pro\_name" value="' . $r[2] . '">'; echo '<input type="hidden" name="price" value="' . $r[4] . '">';**

**echo '<label for="quantity">Quantity:</label>';**

**echo '<button type="submit" class="btnbtn-primary" style="margin- left:10px;line-height: normal;">Add to Cart</button>';**

**echo '</form>';**

**echo '<p>Total: Rs.<span id="total">' . ($r[4] \* 1) . '</span></p>'; echo '<script>**

**document.getElementById("quantity").addEventListener("input", function() { var total = price \* quantity;**

**document.getElementById("total").textContent = total.toFixed(2); line-height: 1;**

**});**

**</script>';**

### Payment.php:



* + **Code File:**

**<?php include("header.php");**

**$absolute\_path = 'conn.php'; // Adjust this path include($absolute\_path);**

**?>**

**<?php**

**if (isset($\_GET['pro\_name']) &&isset($\_GET['quantity']) &&isset($\_GET['price']))**

**{**

**$productName = $\_GET['pro\_name'];**

**$quantity = $\_GET['quantity'];**

**$price = $\_GET['price'];**

**$total = $price \* $quantity;**

**$insertQuery = "INSERT INTO cart (pro\_name, price, quantity, total) VALUES ('$productName', '$price', '$quantity', '$total')";**

**if (mysqli\_query($con, $insertQuery))**

**{**

**$insertSuccess = true;**

**}**

**else**

**{**

**}**

**$insertSuccess = false;**

**echo "Error adding product to cart: " .mysqli\_error($con);**

**}**

**else**

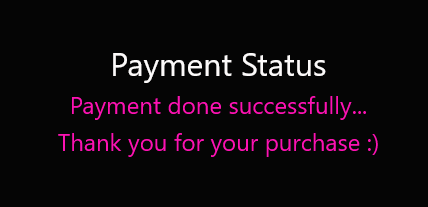
**{**

**echo "Invalid request."; exit;**

**}**

**?>**

### Process\_payment.php:



* + **Code File:**

**<?php include("header.php");**

**?>**

**<div class="container-fluid py-5" style="background-color: #050505;color: #fcfafa;">**

**<div class="row">**

**<div class="col-md-6 offset-md-3" style="text-align:center;">**

**<h2>Payment Status</h2>**

**<h4 style="color: #fc12b2; ">Payment done successfully...</h4>**

**<h4 class="text-center" style="color: #fc12b2;"> Thank you for your purchase :)**

**</h4>**

**</div>**

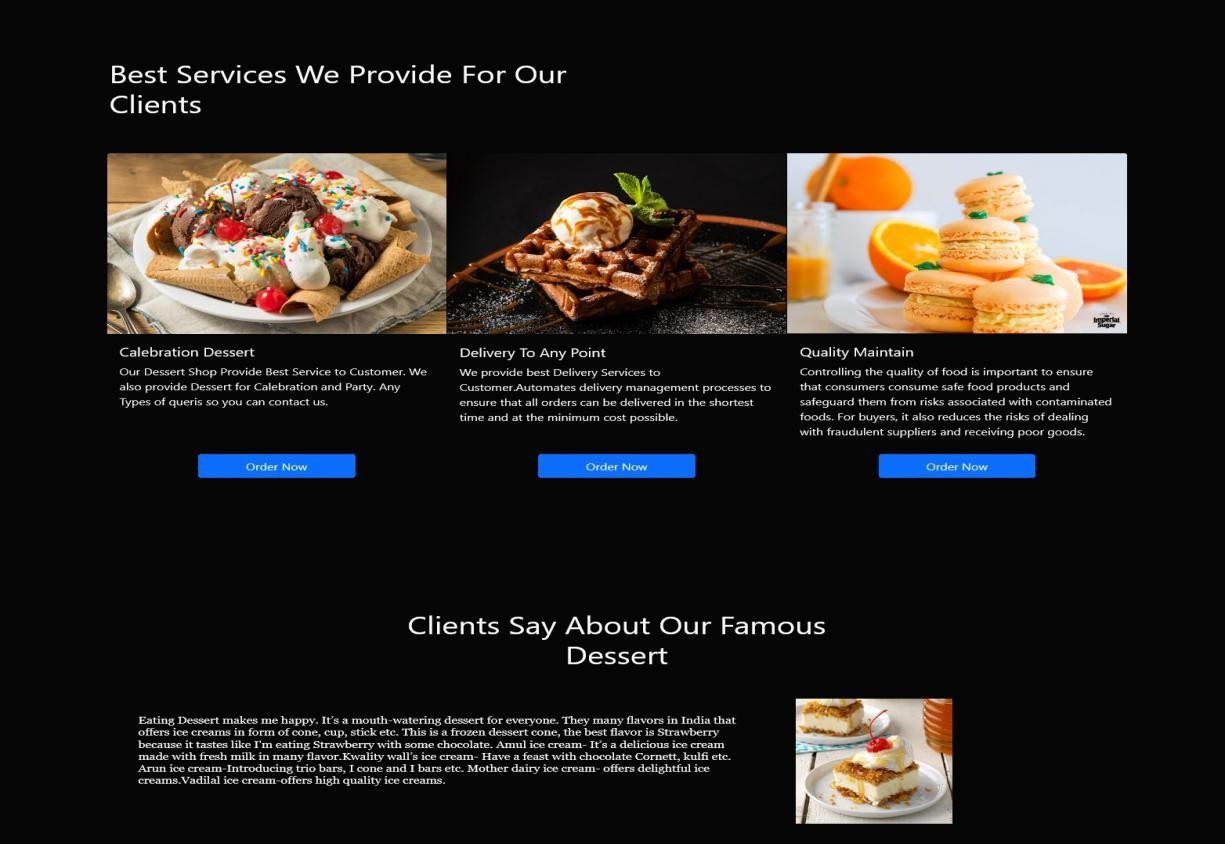
**</div>**

**</div>**

**<?php include("footer.php");**

**?>**

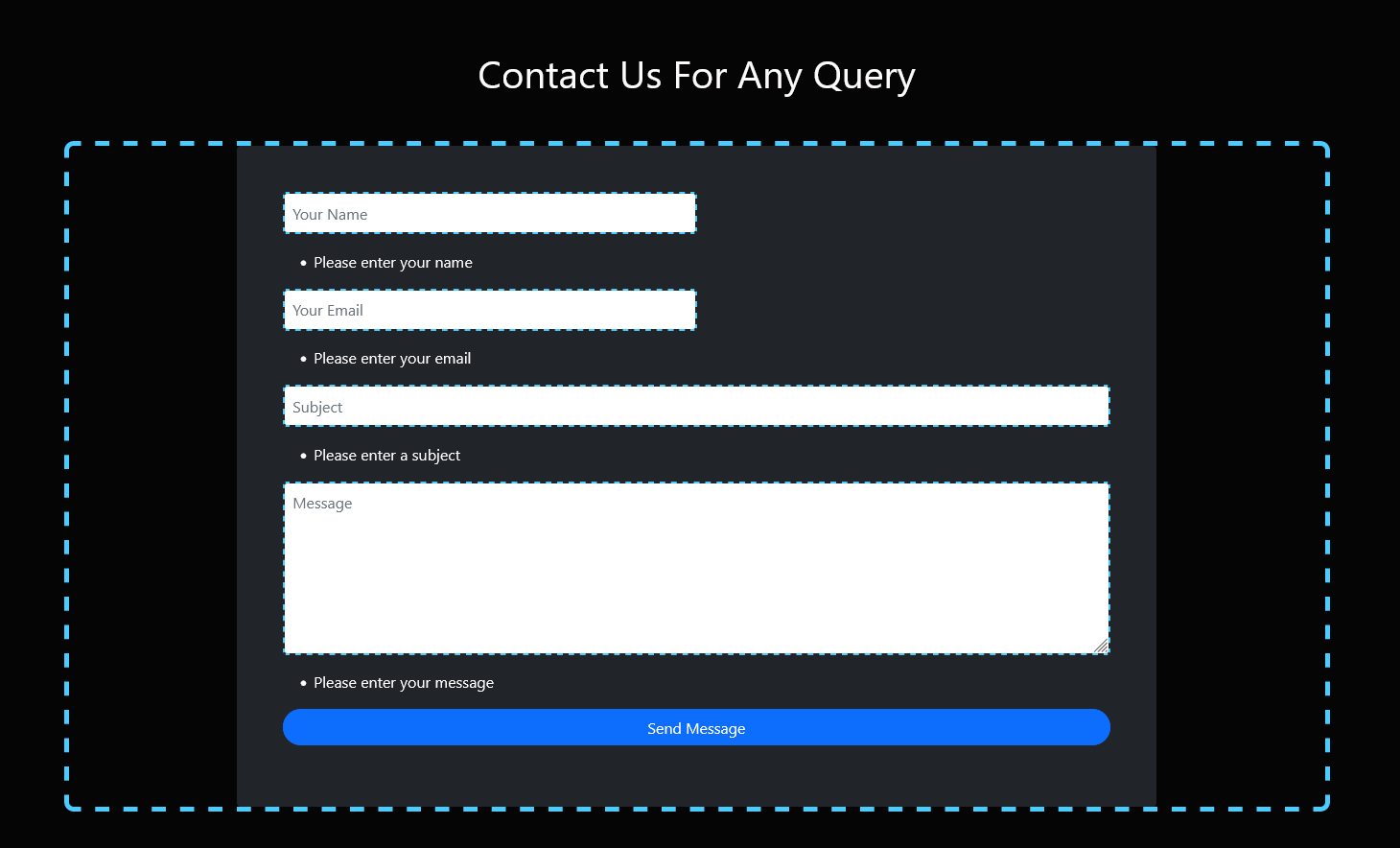
### Services.php:



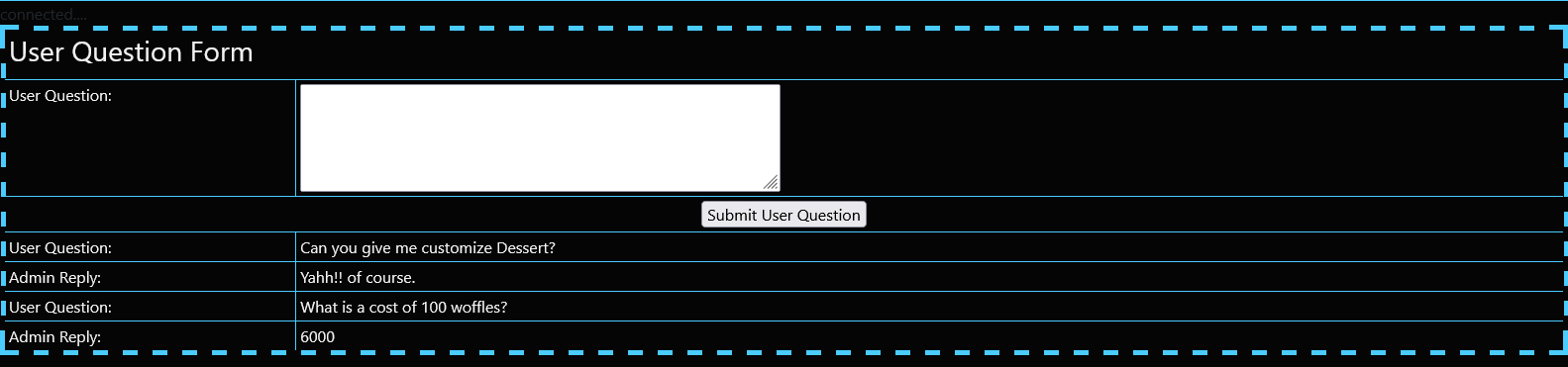
* + **Gallery.php:**



### Contact.php:



* + **FAQ’s.php:**



### Code File:

**<?php include("header.php"); include("conn.php");**

**if ($\_SERVER["REQUEST\_METHOD"] == "POST"**

**&&isset($\_POST['submit\_user\_question'])) {**

**$user\_question = $\_POST["user\_question"];**

**$query = "INSERT INTO faqs (user\_question) VALUES ('$user\_question')";**

**$result = mysqli\_query($con, $query); if ($result) {**

**echo "<script>window.location='FAQs.php'</script>";**

**} else {**

**echo "Error adding user question: " .mysqli\_error($con);**

**}**

**}**

**if ($\_SERVER["REQUEST\_METHOD"] == "POST"**

**&&isset($\_POST['submit\_admin\_reply'])) {**

**$faq\_id = $\_POST["faq\_id"];**

**$admin\_reply = $\_POST["admin\_reply"];**

**$update\_query = "UPDATE faqs SET admin\_reply = '$admin\_reply' WHERE id**

**= $faq\_id";**

**$update\_result = mysqli\_query($con, $update\_query); if (!$update\_result) {**

**echo "Error updating admin reply: " .mysqli\_error($con);**

**}}**

**$query = "SELECT \* FROM faqs";**

**$result\_res = mysqli\_query($con, $query);**

**?>**

**<?php**

**if ($result\_res) {**

**while ($row = mysqli\_fetch\_array($result\_res)) { echo "<tr>";**

**echo "<td>User Question:</td>";**

**echo "<td>{$row['user\_question']}</td>"; echo "</tr>";**

**if (empty($row['admin\_reply'])) { echo "<tr>";**

**echo "<td><label for='admin\_reply'>Admin Reply:</label></td>"; echo "<td>";**

**echo "<form method='post' action='{$\_SERVER['PHP\_SELF']}'>"; echo "<input type='hidden' name='faq\_id' value='" . $row['id'] . "'>"; echo "<textarea name='admin\_reply' rows='4' cols='50'**

**required></textarea>";**

**echo "<input type='submit' name='submit\_admin\_reply' value='Submit Admin Reply'>";**

**echo "</form>"; echo "</td>"; echo "</tr>";**

**} else {**

**echo "<tr>";**

**echo "<td>Admin Reply:</td>";**

**echo "<td>{$row['admin\_reply']}</td>"; echo "</tr>";**

**}**

**}**

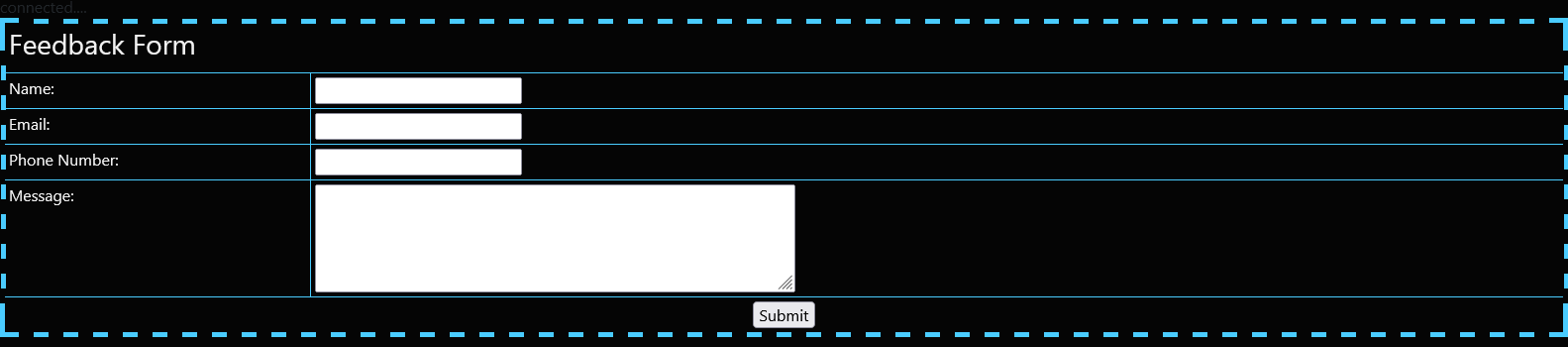
**} else {**

**echo "Error fetching FAQs: " .mysqli\_error($con);**

**}**

**?>**

### Feedback.php:



* + **Code file:**

**<!--header start-->**

**<?php include("header.php"); include("conn.php");**

**?>**

**<!--header end-->**

**<?php**

**// Check if the form has been submitted**

**if ($\_SERVER["REQUEST\_METHOD"] == "POST") {**

**// Process form submission**

**$name = $\_POST["Name"];**

**$email = $\_POST["Email"];**

**$phoneNumber = $\_POST["PhoneNumber"];**

**$message = $\_POST["message"];**

**// Insert data into the database**

**$query = "INSERT INTO feedback (Name, Email, phone\_no, Message) VALUES ('$name', '$email', '$phoneNumber', '$message')";**

**$result = mysqli\_query($con, $query); if ($result) {**

**echo "<script>windows.location='feedbackview.php'</script>";**

**} else {**

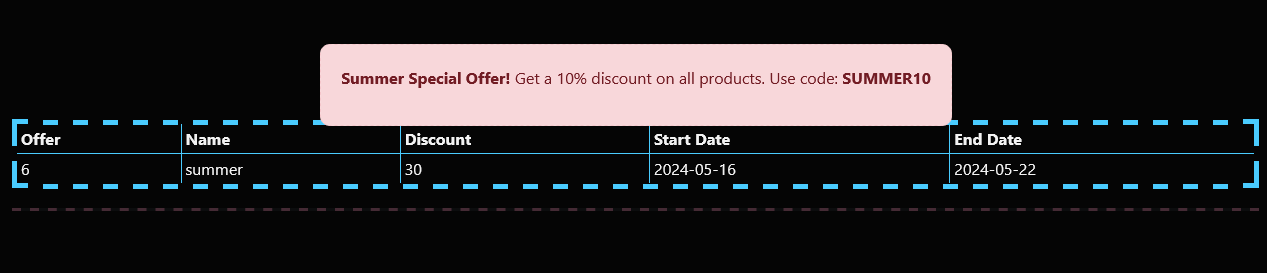
**echo "Error adding feedback: " .mysqli\_error($con);**

**}**

**}**

**?>**

### Offers.php:



* + **Code File:**

**<?php**

**$query = "SELECT \* FROM offers"; // Adjust your query**

**$result = mysqli\_query($con, $query);**

**if ($result) {**

**while ($row = mysqli\_fetch\_array($result))**

**{**

**echo "<tr>";**

**echo "<td>" . $row[0] . "</td>";**

**echo "<td>" . $row[1] . "</td>";**

**echo "<td>" . $row[2] . "</td>";**

**echo "<td>" . $row[3] . "</td>";**

**echo "<td>" . $row[4] . "</td>"; echo "</tr>";**

**else {**

**}**

**}**

**echo "Error fetching offers: " .mysqli\_error($con);**

**}**

**?>**

### Events.php:



* + **Code File:**

**<?php**

**$query = "SELECT \* FROM event"; // Adjust your query**

**$result = mysqli\_query($con, $query); if ($result)**

**{**

**while ($row = mysqli\_fetch\_array($result))**

**{**

**echo "<tr>";**

**echo "<td>" . $row[0] . "</td>";**

**echo "<td>" . $row[1] . "</td>";**

**echo "<td>" . $row[2] . "</td>"; echo "</tr>";**

**else {**

**}**

**}**

**echo "Error fetching feedback: " .mysqli\_error($con);**

**}**

**?>**

### Spinner.php:



* + **Code File:**

**<script>**

**function spinWheel() {**

**var spinner = document.querySelector('.spinner'); var items = document.querySelectorAll('.item');**

**var firecracker = document.getElementById('firecracker');**

**var randomNumber = Math.floor(Math.random() \* items.length); var degrees = 360 \* 3 + (randomNumber \* (360 / items.length));**

**spinner.style.transform = 'rotate(' + degrees + 'deg)'; setTimeout(function() { items.forEach(function(item) {**

**item.style.display = 'none';});**

**var winner = items[randomNumber].innerHTML; items[randomNumber].style.display = 'block';**

**firecracker.style.display = 'block'; setTimeout(function() { firecracker.style.display = 'none'; alert('You claimed free : ' + winner);**

**}, 1000); // Adjust the duration as needed**

**}, 3000); // Adjust the duration as needed**

**}**

**</script>**

### Testing Approach:

* **Top – Down Approach:**
  + The top-down approach to software development is a systematic method that starts with creating a high-level design of the entire system and progressively breaking it down into smaller, more manageable components. This approach offers a structured framework for handling complex software projects. Initially, a high-level system design is established, defining the system's primary components, interactions, and data and control flow. The system is then decomposed into subsystemsand further into smaller units, allowing developers to work on specific components simultaneously.
  + Testing plays a critical role in this approach, with thorough testing and validation conducted at various stages, from unit testing to system testing. The iterative nature of the top-down approach allows for continuous refinement based on feedback and evolving requirements. Ultimately, the software is deployed once it meets the desired quality and functionality standards.
  + While the top-down approach offers benefits like providing an early system overview and enabling parallel development, it also has challenges, including the need for accurate initial high-level design and potential delays if changes are required later in the development process. Striking a balance between planning and flexibility is key to its success.

### Bottom-Up Approach:

* + Bottom-Up Approach is one in which the smaller problems are solved, and then these solved problems are integrated to find the solution to a bigger problem. Therefore, it uses composition approach.
  + It requires a significant amount of communication among different modules. It is generally used with objectoriented programming paradigm such as [C++,](https://www.tutorialspoint.com/cplusplus/index.htm) JAVA, and Paython. Data encapsulation and data hiding is also implemented in this approach. The bottom-up approach is generally used in testing modules.

### Black Box Testing:

* + Black box testing is a testing technique where the tester examines the software's functionality without any knowledge of its internal code, structure, or implementation details.
  + Testers treat the software as a "black box," focusing solely on inputs, expected outputs, and the system's behavior.
  + The primary goal is to validate that the software meets its specified requirements and functions correctly from the user's perspective.
  + Test cases for black box testing are typically derived from the software's functional specifications, user stories, or use cases.
  + Black box testing includes various methods such as functional testing, integration testing, system testing, and acceptance testing.

### White Box Testing:

* + White box testing, also known as glass box or structural testing, is the opposite of black box testing. Testers have access to the internal code and design of the software being tested.
  + Testers design test cases based on an understanding of the software's code, data structures, algorithms, and architecture.
  + The primary goal is to ensure that the code is well-structured, adheres to coding standards, and that all code paths and branches are tested thoroughly.
  + White box testing is useful for uncovering issues such as code vulnerabilities, logical errors, and performance bottlenecks.
  + Techniques used in white box testing include code coverage analysis, control flow analysis, and data flow analysis.

### Grey Box Testing:

* + Grey box testing is a hybrid approach that combines elements of both black box and white box testing.
  + Testers have partial knowledge of the internal code and use this knowledge to design test cases that focus on specific code paths or areasof interest.
  + Grey box testing is often used when a balance between the thoroughnessof white box testing and the user-centric focus of black box testing is required.
  + It can be particularly valuable for uncovering security vulnerabilities, integration issues, and complex interactions within a software system.
  + Testers may use techniques like API testing, database testing, or code reviews to gain insight into the internal workings of the software
* **General Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Test Case** | **Expected Result** | **Passed/Failed** |
| 1 | Launch the spin wheel feature and spin it. | Users should be able to spin the wheel and receive a reward or  message based on the outcome. | Passed |
| 2 | Add an event from the admin -side website. | The event details should  be successfully added to the user's application. | Passed |
| 3 | Login into the application by giving valid Username and Password. Then  logout from the | The Login page should be displayed and user should not be allowed  to view the previous | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
|  | application. Then click on  the Back button of the browser. | page. |  |
| 4 | Browse through the available offers and select  one to view details. | Users should be able to view the details of the  selected offer. | Passed |
| 5 | Participate in a promotional event advertised on the website. | Users should be able to participate in the event as per the provided instructions. | Failed |
| 6 | Check whether the  password field is stored in the database. | The password should be stored in the database. | Passed |
| 7 | Check the title of the internet explorer window. | The name should be meaningful and should reflect the action done in window. | Passed |
| 8 | Share an offer or event on social media platforms directly from the website. | The shared link should contain accurate  information about the offer or event. | Failed |
| 9 | Receive a notification for upcoming events or new  offers upon logging into the website. | Users should receive notifications about  relevant events or offers upon logging in. | Failed |
| 10 | Check the size of buttons in all windows. | All the buttons should be of same size. | Passed |
| 11 | Check the size of pop up Screen. | All the popup screens should be of appropriate  size according to the action it is doing. | Passed |
| 12 | Attempt to register for an event that has reached its maximum capacity. | Users should receive a notification indicating  that the event is full and registration is closed. | Failed |
| 13 | Redeem a promotional code or discount during checkout of an event or offer. | Users should be able to successfully apply the code and see the  discounted price, if applicable. | Failed |

|  |  |  |  |
| --- | --- | --- | --- |
| 14 | Check the placement of all the controls. | All controls should be properly placed. | Passed |
| 15 | Check the colour in all window. | The colour should be uniform. | Passed |
| 16 | Check the font and size of font in different pages. | The font and size of font should be same. | Passed |
| 17 | Explore the spin wheel, event, and offer functionalities across different devices and browsers. | The functionalities should work consistently and seamlessly across  various devices and browsers. | Passed |
| 18 | Check for any accessibility issues with the spin wheel, events, and offers features. | The features should be accessible to users with disabilities, complying  with accessibility standards. | Passed |
| 19 | Verify that users can easily find information about upcoming events and current offers on the  website's homepage. | The homepage should prominently display information about events and offers. | Passed |
| 20 | Search Functionality | Users can search for desserts by name,  category, or ingredients. | Failed |
| 21 | Filter Functionality | Users can filter desserts by dietary preferences (e.g., vegan, gluten- free), price range, and  popularity. | Failed |
| 22 | Featured Desserts | Featured desserts are prominently displayed on the homepage or a dedicated section,  enticing users to explore further. | Passed |

|  |  |  |  |
| --- | --- | --- | --- |
| 23 | Special Offers and Promotions | Users are informed about special offers, discounts, and promotions on selected  desserts through banners or notifications. | Failed |
| 24 | Dessert Details Page | Each dessert has a dedicated details page providing information  such as description, price. | Passed |
| 25 | Dessert Ratings and Reviews | Users can rate desserts and leave reviews, providing valuable feedback to other  customers and the Dessert Delight team. | Passed |
| 26 | Feedback Submission | Users can submit  feedback through a form on the website. | Passed |
| 27 | Feedback Management | Website administrators can access and manage  submitted feedback. | Passed |
| 28 | FAQ Accessibility | The FAQs section is easily accessible from the website's main navigation menu or  footer. | Passed |
| 29 | User Interaction | Users can rate the helpfulness of individual FAQs and submit new questions or  suggestions. | Passed |
| 30 | Responsive Design | The FAQs section is responsive and displays properly on different  devices and screen sizes. | Passed |

Chapter 6: Security and Measures

# CHAPTER 6: SYSTEM SECURITY AND MEASURES:

* Securing an e-commerce website like "Dessert Delight" involves implementing various measures to protect sensitive data, prevent unauthorized access, and ensure the integrity and confidentiality of transactions. Additionally, measuring the effectiveness of these security measures is crucial for ongoing improvement and maintenance of the website. Below are some key aspects of security and measurement for an e-commerce website implemented using PHP:

1. **Data Encryption:**

* Use encryption protocols (e.g., SSL/TLS) to secure data transmission between the website and users. This ensures that sensitive information, such as payment details, is encrypted and cannot be intercepted by malicious actors.

1. **Regular Software Updates:**

* Keep all software components, including the operating system, web server, e- commerce platform, and plug-ins, up to date with the latest security patches and updates.

1. **User authentication:**

* Enforce strong password policies and encourage multi-factor authentication (MFA) for user accounts to prevent unauthorized access.

1. **Data Backup & Recovery:**

* Implement automated data backup solutions to ensure data recovery in case of system failures, data corruption, or cyber attacks.

1. **Session Management:**
   * Use secure session management techniques to prevent session hijacking and fixation.
   * Implement session timeouts and regenerate session IDs after successful authentication
2. **Security Headers:**
   * Set appropriate security headers in HTTP responses to mitigate attacks like Cross-Site Scripting (XSS), Clicking, and MIME sniffing.

Chapter 7: Future Scope and Enhancement

# CHAPTER 7: FUTURE SCOPE AND ENHANCEMENT

* **Interactive Features:**
  + Incorporate interactive elements such as sliders, carousels, and interactive galleries to showcase desserts and engage users.
* **Seasonal Offers and Specials:**
  + Create seasonal offers, discounts, and special offers to incentivize users to try new desserts and increase sales during peak seasons.
* **Improved Navigation:**
  + Streamline navigation by implementing clear menus and categories for different types of desserts. Users should be able to easily find what they are looking for.
* **Online Ordering System:**
  + Develop an integrated online ordering system that allows users to place orders directly through the website. Include features such as customizable orders, delivery options, and secure payment gateways.
* **Integration with Social Media:**
  + Allow users to share their favorite desserts on social media platforms. Also, integrate social media feeds to showcase user-generated content and engage with the community.
* **Performance Optimization:**
  + Optimize website performance by minimizing loading times, optimizing images, and reducing unnecessary scripts to provide a seamless user experience.
* **Customization Options:**
  + Enable customers to customize dessert boxes with their preferred dessert’s, flavors, and packaging, creating a personalized and unique gift option.
* **Video Content:**
  + Incorporate video content such as recipe tutorials, behind-the-scenes footage, and chef interviews to make the website more engaging and dynamic.
* **Advanced Search and Filtering:**
  + Improve the search functionality with advanced filters, such as dietary preferences (e.g., vegan, gluten-free) and dessert types (e.g., gummies, chocolates).
* **AI Power Recommendations:**
  + Use artificial intelligence to analyze customer behavior and recommend products, increasing cross-selling and up selling opportunities.
* Foster customer loyalty ,and open up new revenue streams for the dessert delight's e-commerce business.

Chapter 8: Conclusion and Limitations

# CHAPTER 8: CONCLUSION AND LIMITATIONS

In conclusion, the establishment and successful operation of the Dessert Delight represent a significant achievement in the pursuit of delivering delightful confectionery experiences to our valued customers. The project's objectives were met with dedication, creativity, and a customer-centric approach, resulting in the creation of a thriving Dessert Delight business.

Throughout the development and operation of the Dessert Delight, several key takeaways and accomplishments have emerged:

* **Online Presence:**
  + The Dessert Delight’s e-commerce website has provided a convenient and accessible platform for customers to explore, select, and purchase a wide range of Dessert products from the comfort of their homes.
* **Customer Satisfaction:**
  + The emphasis on user-friendly design, personalized recommendations, and interactive features has contributed to heightened customer satisfaction and loyalty.
* **User Friendly:**
  + User can order from our website wherever they are and can enjoy our delicious rang of Desserts at their comfort zone.
* **Continuous Improvement:**
  + A commitment to continuous improvement, responsive customer support, and proactive problem-solving have been central to the Dessert Delight’s success.

**Limitations:**

However, it is important to acknowledge the limitations encountered during the journey of establishing and operating the Dessert Delight:

* **Technical Challenges:**
  + Technical issues, such as software compatibility and initial setup complexities, required additional time and effort to overcome.
* **Marketing Efforts:**
  + The successful adoption of the website by customers relied on effective marketing and promotion efforts, which required ongoing strategies and investment.
* **External Factors:**
  + External factors, such as market competition and changing consumer preferences, influenced the shop's growth trajectory and required adaptability.
* **Pay On Delivery:**
  + Currently we are only providing pay on delivery option to the customers but in future we will also provide other payment modes also.

In Conclusion, the candy shop has established itself as a beloved destination for dessert enthusiasts, offering a wide array of sweet delights and an engaging online shopping experience. While challenges and limitations were encountered, the project's dedication to customer satisfaction and continuous improvement positions the Dessert Delight for continued success in the dynamic confectionery industry. With a clear understanding of its strengths, limitations, and the ever-evolving needs of its customer base, the Dessert Delight is well-prepared to navigate future opportunities and challenges, and to continue delighting customers with sweet indulgences.

Chapter 9: Bibliography

# CHAPTER 9: BIBLOGRAPHY

* I have used this kind of references for making this document and project:

1. For making the website user friendly: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
2. For making website easy understandable: <https://www.w3schools.com/>
3. For finding design:

<https://htmlcodex.com/ice-cream-shop-website-template/>

1. For Diagram Creation: <https://online.visual-paradigm.com/>
2. For Images: <https://www.pinterest.com/>
3. For MySql Query: <https://www.geeksforgeeks.org/>