

Riverside City College

# **Café by Toastology**

Shopping Cart - Final Project

Aurelia Juan Sindhunirmala

CIS-17B – Section 43671

Professor Mark Lehr

June 9, 2024

## Table of Contents

<b>Cover</b>	0
I. Table of Contents	1
II. Project Description	2
III. Development Summary	2
A. Lines of Codes	2
B. Versions and Descriptions	4
IV. File and Descriptions	6
V. Project Requirements	9
VI. UML	22
VII. Screenshots of Application	24
User's Interface	23
Admin Portal	29
Database	32
VIII. Demo Video	33
User View	33
Admin View	33
IX. Preview Code	33
Index	33
User_login	35
Shop	37
Products	40

## I. Project Description

Café by Toastology is a virtual café where users may interact to purchase various delicious toasts. The backend of this program covers user and admin software. Each user and admin has a different interface and functionalities for the shopping catalog. Additionally, they are also allowed to view this website and interact with it. However, when the user wants to purchase a toast, they need to have an account. Users will be prompted from selecting the toast up to checkout, where user fill in their address, contact information, and payment type. A unique feature embedded in this program is a personality test which may help the user define their toast and help them navigate in choosing their toast. In contrast, the admin will have the backend view of the barebone of the program. Admin will have the privilege to add, update, and delete products, user accounts, and messages. This program is coded in various languages with a Mac operating system. There is HTML, CSS, and JavaScript utilized for the frontend, PHP for the backend, and MySQL for the database.

**Full code and Versions:** [Github Repo](#)

## II. Developmental Summary

The final version of Café by Toastology contains files elaborated below. While each file has a designated file type, there is a combination of all the languages used in a couple of files.

Folder	File Name	Line Count
admin	account.php	86
	back.php	43
	home.php	108
	login.php	67
	messages.php	75
	products.php	146

	register_admin.php	77
	update_account.php	83
	update_admin.php	83
	update_product.php	147
	users_account.php	78
attributes	admin_logout.php	11
	admin_msg.php	12
	connect.php	24
	footer.php	27
	header.php	63
	user_logout.php	11
css	admin.css	277
	style.css	760
js	admin_script.js	35
	script.js	106
root	about.php	79
	cart.php	102
	checkout.php	156
	contact.php	70
	index.php	68

	shop.php	95
	update_user.php	87
	user_login.php	86
	user_reg.php	85
	<b>Total</b>	3147

Café by Toastology is developed through several versions that cover updates from the previous version of the program and handling errors that occurred.

Versions	Description
1	<ul style="list-style-type: none"> <li>● Created a home page and its features, which cover a navigation bar, header, and footer. <ul style="list-style-type: none"> <li>○ Designed and implemented the homepage layout using HTML and CSS.</li> <li>○ Included a navigation bar for easy access to different sections of the program.</li> <li>○ Incorporate header and footer for available sites.</li> </ul> </li> <li>● Connect with the database through OpenOffice and MySQL. <ul style="list-style-type: none"> <li>○ Configure a MySQL database to store and manage data.</li> <li>○ Utilized OpenOffice as a tool for connecting and managing the database.</li> </ul> </li> </ul>
2	<ul style="list-style-type: none"> <li>● Create a user database that prompts users to create or login into their accounts. <ul style="list-style-type: none"> <li>○ Designed a user database schema in MySQL to store user information such as usernames, email address, and passwords.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Implemented registration and login forms using HTML, CSS, and PHP to allow users to create new accounts or log into existing ones.</li> <li>● Implement the cookies to keep track of the user's session.           <ul style="list-style-type: none"> <li>○ Utilized PHP to create and manage cookies for user sessions.</li> <li>○ Ensured that cookies store essential session information.</li> </ul> </li> </ul>
3	<ul style="list-style-type: none"> <li>● Create a message, shopping, and checkout page for users.           <ul style="list-style-type: none"> <li>○ Developed a message page where users can send and receive messages using HTML, CSS, and PHP</li> <li>○ Created a shopping page that lists products that are available for purchase, with options for users to select and add items to their cart.</li> <li>○ Implemented a checkout page that allows users to review their cart, enter payment information, and complete their purchases.</li> </ul> </li> <li>● Connect the database for messages.           <ul style="list-style-type: none"> <li>○ Designed a massage table in MySQL database to store user messages</li> <li>○ Wrote PHP scripts to handle message storage.</li> <li>○ Ensured that messages were securely stored and retrieved from the database.</li> </ul> </li> </ul>
4	<ul style="list-style-type: none"> <li>● Create the admin interface, which consists of login and its home page.           <ul style="list-style-type: none"> <li>○ Developed an admin login page and home page using HTML, CSS, and PHP.</li> </ul> </li> <li>● Connect the user, admin, and message database for the admin to view.           <ul style="list-style-type: none"> <li>○ Designed an admin dashboard to view and manage user accounts, messages, and other relevant data.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Wrote PHP scripts to fetch and display user and message data from the database.</li> </ul>
5	<ul style="list-style-type: none"> <li>● Created a product database that allows the admin to insert products, view, and edit. <ul style="list-style-type: none"> <li>○ Designed a product table in the MySQL database to store product details such as name, description, price, and images.</li> <li>○ Implemented forms and scripts for the admin to add new products, view existing products, and edit product details.</li> </ul> </li> </ul>
6	<ul style="list-style-type: none"> <li>● Connects the product database to the shop view for the user. <ul style="list-style-type: none"> <li>○ Developed a shop page that dynamically fetches and displays products from the database using PHP.</li> </ul> </li> <li>● Implement an add-to-cart button for all selected products to be in the cart. <ul style="list-style-type: none"> <li>○ Developed a shopping cart system to store selected items, manage quantities, and display the total cost.</li> </ul> </li> </ul>
7	<ul style="list-style-type: none"> <li>● Finalized the program: <ul style="list-style-type: none"> <li>○ Corrected the styling for a seamless user experience</li> <li>○ Deleting all commented-out code</li> <li>○ Making sure all related databases connect.</li> </ul> </li> </ul>

### III. File and Descriptions

Each file created in this program serves its own purpose. This section will cover the functionality of each and every file.

#### Admin Folder

- Consists of ten files that contain the PHP script that handles the administrative functionalities of the admin portal, such as managing user accounts, products, and messages.

<b>File Name</b>	<b>Description</b>
account.php	Displays the admin's account details and allows updates to their information.
back.php	A back button that utilized javascript to go back to the previous page through their history
home.php	The main dashboard for the admin, displays all the options the admin can do
login.php	Allowing administrators to authenticate and gain access to the admin interface
messages.php	Displays messages sent by users to the admin, allowing the admin to view and manage communications
products.php	Manages the products listed on the website, allowing admin to add, update, and delete products
register_admin.php	Admin registration page, enabling the creation of new admin accounts
update_admin.php	Allows the admin to update their account information
update_product.php	Allows the admin to update product details, including name, price, details, and image
users_account.php	Displays and manages user accounts, allowing the admin to view user details and perform termination accounts

### Attributes Folder

- Consists of essential scripts and components that are utilized across different parts of the website.

<b>File Name</b>	<b>Description</b>
admin_logout.php	Logs out the admin and ends the main session
admin_msg.php	Displays the admin-specific messages or alerts

connect.php	Establishes a connection to the MySQL database
footer.php	Includes the footer section used across various pages
header.php	Includes the header section used across various pages
user_logout.php	Logs out the user and ends the user session

### JS Folder

- This folder contains JavaScript field used for adding interactivity and dynamic behavior to the website

File Name	Description
admin_script.js	JavaScript specific to the admin interface, enhancing its functionality
script.js	General JavaScript files used for interactivity and dynamic features

### Root Files

File Name	Description
about.php	The page provides information about the website utilization
cart.php	Display's user's shopping cart, showing selected products and allows for quantity adjustments and removal of items
checkout.php	Once the user confirms their cart, they may checkout their purchases by entering payment information and shipping address
contact.php	Provides a form for users to send messages or inquiries

	to the administrator
index.php	The homepage for the website, serving the entry point for users and displaying a welcome page
shop.php	Listing products available for purchase and allowing users to add items to their cart
update_user.php	Allows the user to update their information, such as username, email, and password
user_login.php	Prompts user to authenticate and access their accounts
user_reg.php	Allowing new user to create an account

## IV. Project Requirements

### 1. MVC - How you delineated your objects

- Model: Handles data-related logic. This can be found through the database interactions via PHP.
- Model Function:

```
class Model {
    constructor() {
        this.navbarActive = false;
        this.profileActive = false;
    }

    toggleNavbar() {
        this.navbarActive = !this.navbarActive;
        this.profileActive = false;
    }

    toggleProfile() {
        this.profileActive = !this.profileActive;
        this.navbarActive = false;
    }

    reset() {
```

```

        this.navbarActive = false;
        this.profileActive = false;
    }
}

```

- One of the implementations is the utilization of PHP scripts for database operations, which includes ‘connect.php’, SQL queries within the PHP scripts.
- Example:

```

<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "cafeDatabase";

try {
    $conn = new
    PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
    $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>

```

- View: Represents the UI of the application which includes any HTML, CSS, and JavaScript files.

- Example:

```

class View {
    constructor() {
        this.navbar = document.querySelector('.header .flex
.navbar');
        this.profile = document.querySelector('.header .flex
.profile');
    }

    updateNavbar(active) {
        if (active) {
            this.navbar.classList.add('active');
        } else {
            this.navbar.classList.remove('active');
        }
    }
}

```

```

        }

    }

    updateProfile(active) {
        if (active) {
            this.profile.classList.add('active');
        } else {
            this.profile.classList.remove('active');
        }
    }

    reset() {
        this.navbar.classList.remove('active');
        this.profile.classList.remove('active');
    }
}

```

```

<?php
    include ('attributes/connect.php');
    session_start();

    if(isset($_SESSION['user_id'])){
        $user_id = $_SESSION['user_id'];
    }else{
        $user_id = 'guest';
    };

    if(isset($_POST['send'])){
        $name = $_POST['name'];
        $email = $_POST['email'];
        $number = $_POST['number'];
        $msg = $_POST['msg'];

        $select_message = $conn->prepare("SELECT * FROM
`messages` WHERE name = ? AND email = ? AND number = ? AND
message = ?");
        $select_message->execute([$name, $email, $number,
$msg]);

        if($select_message->rowCount() > 0){
            $message[] = 'already sent message!';
        }else{

```

```

        $insert_message = $conn->prepare("INSERT INTO
`messages`(`user_id`, `name`, `email`, `number`, `message`)
VALUES (?, ?, ?, ?, ?)");
        $insert_message->execute([$user_id, $name,
$email, $number, $msg]);

        $message[] = 'sent message successfully!';
    }
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Cafe by Toastology</title>

    <!-- Icon -->
    <link rel="icon" href="../pic/icon.png">

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

    <!-- Font Awesome -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6
.5.2/css/all.min.css"
integrity="sha512-SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6T
a86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9QevSCWr3W6A=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
    <link rel="stylesheet"
href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css
" >

    <!-- Font -->
    <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=Poppins:ital
        ,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900
        ;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&disp
        lay=swap" rel="stylesheet">
    </head>
    <body>
        <?php include ('attributes/header.php') ?>

        <section class="contact">
            <form action="" method="post">
                <h3> Leave Us a Message</h3>
                <input type="text" name="name"
placeholder="enter your name" required maxlength="20"
class="box">
                <input type="email" name="email"
placeholder="enter your email" required maxlength="50"
class="box">
                <input type="number" name="number" min="0"
max="9999999999" placeholder="enter your phone number"
required onkeypress="if(this.value.length == 10) return
false;" class="box">
                <textarea name="msg" class="box"
placeholder="enter your message" cols="30"
rows="10"></textarea>
                <input type="submit" value="send message"
name="send" class="btn">
            </form>
        </section>

        <?php include ('attributes/footer.php') ?>
        <script src="js/script.js"></script>
    </body>
</html>

```

- Controller: Handles the logic and user input
  - Includes the PHP scripts that process submissions and manage sessions
  - Example:

```

class Controller {
    constructor(model, view) {
        this.model = model;
    }
}

```

```

this.view = view;

    document.querySelector('#menu-btn').onclick = () =>
this.toggleNavbar();
    document.querySelector('#user-btn').onclick = () =>
this.toggleProfile();
    window.onscroll = () => this.reset();
}

toggleNavbar() {
    this.model.toggleNavbar();
    this.view.updateNavbar(this.model.navbarActive);
    this.view.updateProfile(this.model.profileActive);
}

toggleProfile() {
    this.model.toggleProfile();
    this.view.updateNavbar(this.model.navbarActive);
    this.view.updateProfile(this.model.profileActive);
}

reset() {
    this.model.reset();
    this.view.reset();
}
}

```

```

<?php
include ('../attributes/connect.php');
include ('../attributes/admin_msg.php');
session_start();

if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $pass = sha1($_POST['pass']);

    $select_admin = $conn->prepare("SELECT * FROM
`admins` WHERE name = ? AND password = ?");
    $select_admin->execute([$name, $pass]);
    $row = $select_admin->fetch(PDO::FETCH_ASSOC);

    if($select_admin->rowCount() > 0){

```

```

        $_SESSION['admin_id'] = $row['id'];
        echo "successful";
        header('location:home.php');
    }else{
        $message[] = 'incorrect username or password!';
        echo 'incorrect username or password';
    }
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Cafe by Toastology - Admin</title>

    <!-- Icon -->
    <link rel="icon" href="../pic/icon.png">

    <!-- CSS -->
    <link rel="stylesheet" href="../css/admin.css">

    <!-- Font Awesome -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6
.5.2/css/all.min.css"
integrity="sha512-SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6T
a86w+fsb2TkcmfRyVX3pBnMFcV7oQPJk19QevSCWr3W6A=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
    <link rel="stylesheet"
href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css
" >

    <!-- Font -->
    <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=Poppins:ital

```

```
,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900
;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
</head>
<body>
    <div class = "admin-login">
        <section class="admin-form">
            <form action="" method="post">
                <h3>Login now</h3>
                <p>Default username = <span>admin</span> &
                password = <span>1212</span></p>
                <input type="text" name="name" required
                placeholder="enter your username" maxlength="20"
                class="box" oninput="this.value = this.value.replace(/\s/g,
                '')">
                <input type="password" name="pass" required
                placeholder="enter your password" maxlength="20"
                class="box" oninput="this.value = this.value.replace(/\s/g,
                '')">
                <input type="submit" value="login now"
                class="btn" name="submit">
            </form>

            <div class = "switch">
                <a href ="../index.php" class = "btn"> Back
                to User View </a>
            </div>
        </section>
    </div>

    <?php include ('../attributes/footer.php')?>
    <script src="../js/admin_script.js"></script>
</body>
</html>
```

## 2. Objects - Javascript/PHP - Serialization

- Serialization is utilized to convert PHP data into a format that is easily stored. In this example serialization is stored with arrays of name, role, and links.

```
$creatorInfo = array(
```

```

        'name' => 'Aurelia Sindhunirmala',
        'role' => 'Second Year Computer Science Student',
        'links' => array(
            'linkedin' =>
                'https://www.linkedin.com/in/aurelia-sindhunirmala-b14280216/',
                'github' => 'https://github.com/AureliaSindhu',
                'email' => 'aurelia.sindhu@gmail.com'
            )
        );
    };

$serializedCreatorInfo = json_encode($creatorInfo);

```

```

<script>
    var creatorInfo = <?php echo $serializedCreatorInfo; ?>;
    console.log(creatorInfo);
</script>

```

### 3. Passing info with Javascript Objects to PHP Objects with Cookies

- Cookies are implemented to read the user's data
- Example:

```

if($select_user->rowCount() > 0){
    $_SESSION['user_id'] = $row['id'];

    setcookie("email", $row['email'], time() + (86400 * 30),
"/"); // expires in 1 hour

    header('Location:index.php');
    exit();
} else{
    $message[] = 'incorrect username or password!';
}

```

### 4. Reading/Writing Files/Local Storage

- Local Storage Example:

```

function saveDataToLocalStorage(key, value) {
    localStorage.setItem(key, JSON.stringify(value));
}

function getDataFromLocalStorage(key) {

```

```

        const data = localStorage.getItem(key);
        return data ? JSON.parse(data) : null;
    }

document.addEventListener('DOMContentLoaded', () => {
    const adminId = admin;
    saveDataToLocalStorage('adminId', adminId);

    const storedAdminId = getDataFromLocalStorage('adminId');
    if (storedAdminId) {
        console.log('Admin ID:', storedAdminId);
    } else {
        console.log('Admin ID not found in local storage.');
    }
});

```

## 5. JSON - Javascript

- The script is utilized to implement an interactive navigation menu.

```

● let navbar = document.querySelector('.header .flex .navbar');

● let profile = document.querySelector('.header .flex .profile');

●

● const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

● if (!emailPattern.test(email)) {

●     alert("Invalid email address!");

● }

● document.addEventListener('DOMContentLoaded', () => {

●     const model = new Model();

●     const view = new View();

●     const controller = new Controller(model, view);

● });

●

● document.querySelector('#menu-btn').onclick = () =>{

●     navbar.classList.toggle('active');

●     profile.classList.remove('active');

● }

```

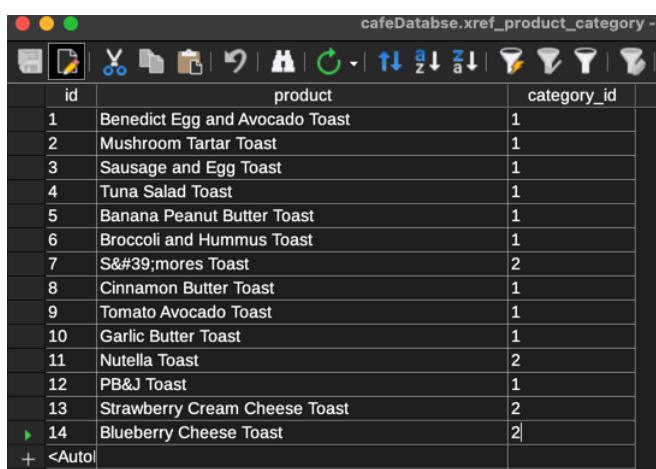
```

•
•
•   document.querySelector('#user-btn').onclick = () =>{
•
•     profile.classList.toggle('active');
•
•     navbar.classList.remove('active');
•
•   }
•
•
•   window.onscroll = () =>{
•
•     navbar.classList.remove('active');
•
•     profile.classList.remove('active');
•
•   }
•

```

## 6. Databases SQL - Identify Entities, Xref, and Enum Tables

- Represents main objects like users, admins, products, and carts.
- Entities: Represent key objects in the application.
  - Users: Represented by the users table.
  - Products, Orders, and Messages: Would be represented by their respective tables.
- Cross-reference (Xref) Tables: Used for many-to-many relationships.
  - Examples:
    - The product id is xref to product id and category.



	id	product	category_id
	1	Benedict Egg and Avocado Toast	1
	2	Mushroom Tartar Toast	1
	3	Sausage and Egg Toast	1
	4	Tuna Salad Toast	1
	5	Banana Peanut Butter Toast	1
	6	Broccoli and Hummus Toast	1
	7	S'mores Toast	2
	8	Cinnamon Butter Toast	1
	9	Tomato Avocado Toast	1
	10	Garlic Butter Toast	1
	11	Nutella Toast	2
	12	PB&J Toast	1
	13	Strawberry Cream Cheese Toast	2
	14	Blueberry Cheese Toast	2
	+	<Auto	

- Enum Tables: Used for predefined sets of values.

- Example: Payment methods (payment\_methods table): Could list possible payment methods (e.g., 'cash on delivery', 'credit card')

```
<div class="inputBox">
    <span>Payment :</span>
    <select name="method" class="box"
required>
        <option value="cash on delivery">Cash On
Delivery</option>
        <option value="credit card">Credit
Card</option>
        <option value="paytm">Debit
Card</option>
        <option value="paypal">Gift
Card</option>
    </select>
</div>
```

## 7. Form Validation - Regular Expressions

- Creating a regular expression for an email pattern

```
const emailPattern = /^[^@\s]+@[^\s]+\.\[^@\s]+$/;
if (!emailPattern.test(email)) {
    alert("Invalid email address!");
}
```

```
$email = $_POST['email'];
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $errors[] = "Invalid email format";
}
```

## 8. User-Admin-Login

- User Login:
- user\_login.php: Handles user authentication.

```
if(isset($_POST['submit'])){
    $email = $_POST['email'];
    $pass = sha1($_POST['pass']);

    $select_user = $conn->prepare("SELECT * FROM `users` "
WHERE email = ? AND password = ?");
    $select_user->execute([$email, $pass]);
```

```

$row = $select_user->fetch(PDO::FETCH_ASSOC);

if($select_user->rowCount() > 0){
    $_SESSION['user_id'] = $row['id'];

    setcookie("email", $row['email'], time() + (86400 * 30), "/");
    // expires in 1 hour

    header('Location:index.php');
    exit();
} else{
    $message[] = 'incorrect username or password!';
}
}

```

- Admin Login:

- admin/login.php: Handles admin authentication.

```

if(isset($_POST['submit'])){
    $name = $_POST['name'];
    $pass = sha1($_POST['pass']);

    $select_admin = $conn->prepare("SELECT * FROM
`admins` WHERE name = ? AND password = ?");

    $select_admin->execute([$name, $pass]);
    $row = $select_admin->fetch(PDO::FETCH_ASSOC);

    if($select_admin->rowCount() > 0){
        $_SESSION['admin_id'] = $row['id'];
        echo "successful";
        header('location:home.php');
    } else{
        $message[] = 'incorrect username or password!';
        echo 'incorrect username or password';
    }
}

```

## 9. Cookies - Sessions - Securing Pages

- Cookies are used to store the user's email after a successful login
  - Example: utilized to create a cookie named 'email' that stores the user's email address and it will expire in 30 days

```

if($select_user->rowCount() > 0){

```

```
$_SESSION['user_id'] = $row['id'];

setcookie("email", $row['email'], time() +
(86400 * 30), "/"); // expires in 1 hour

header('Location:index.php');
exit();

} else{
    $message[] = 'incorrect username or password!';
}
```

- Sessions are used to store data on the server side to persist user-specific information across different pages during the visit

### ○ Example

```
if(isset($_SESSION['user_id'])){
    $user_id = $_SESSION['user_id'];
} else{
    $user_id = '';
};
```

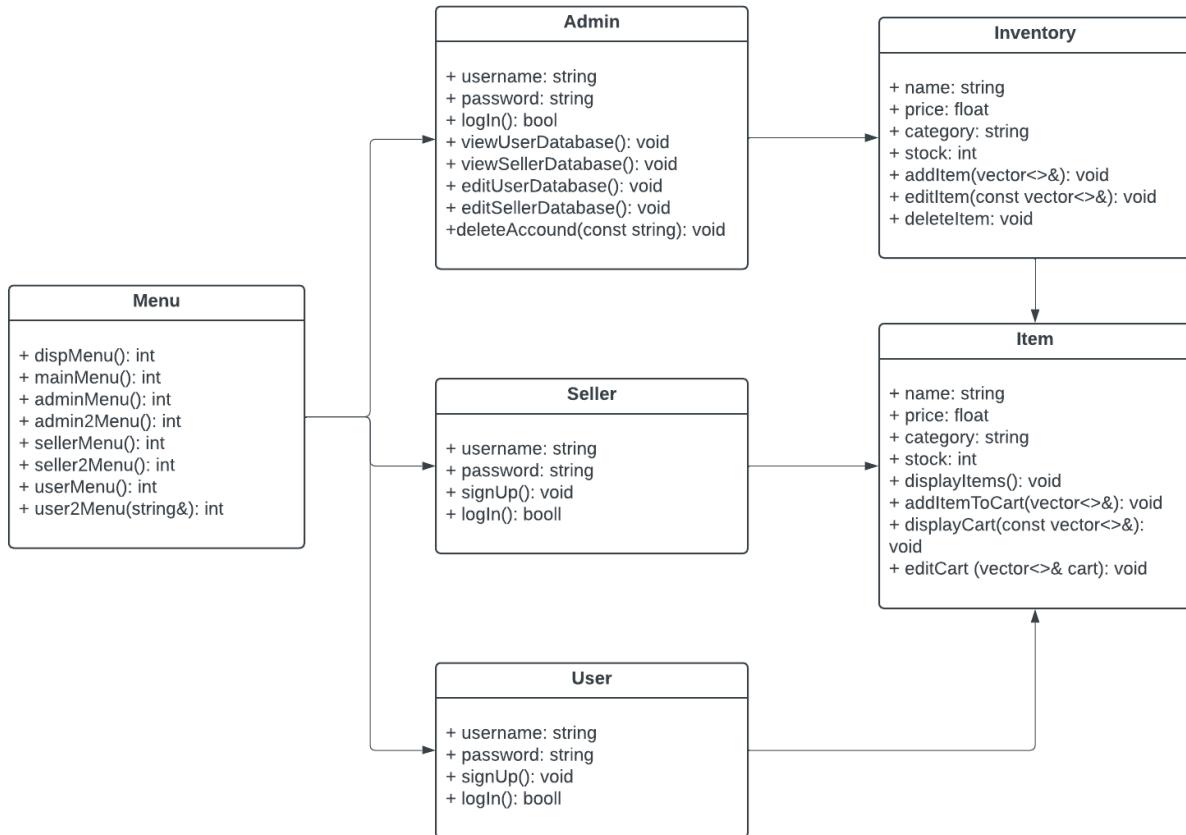
- Securing Pages is implemented by checking the session status and redirecting to the login page if the user is not authenticated

### ○ Example

```
if (!isset($_SESSION['user_id'])) {  
    header('location:user_login.php');  
    exit();  
}
```

V. UML

This is the UML of Cafe by Toastology the shopping catalog program:

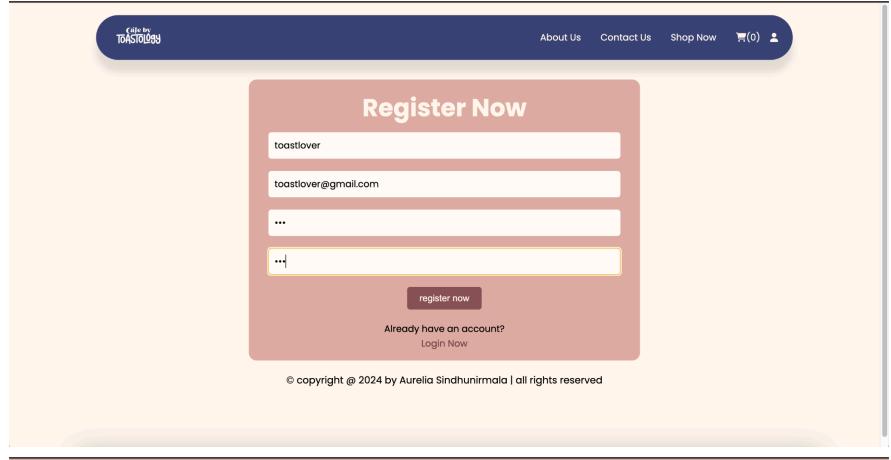
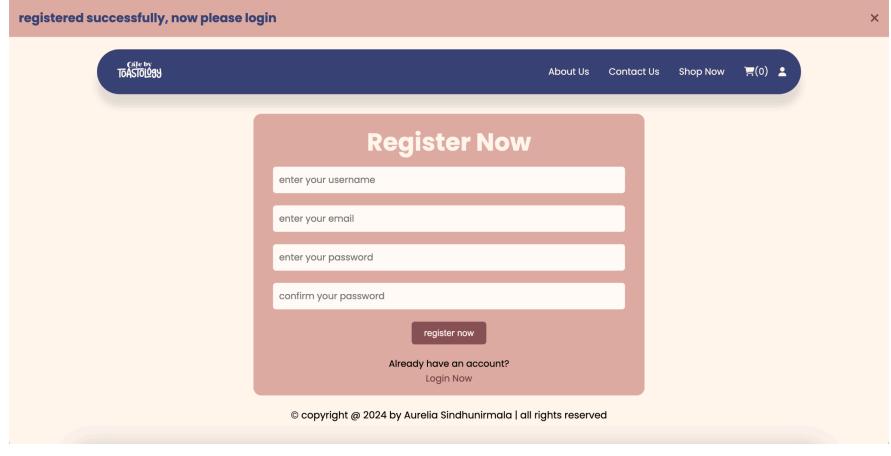
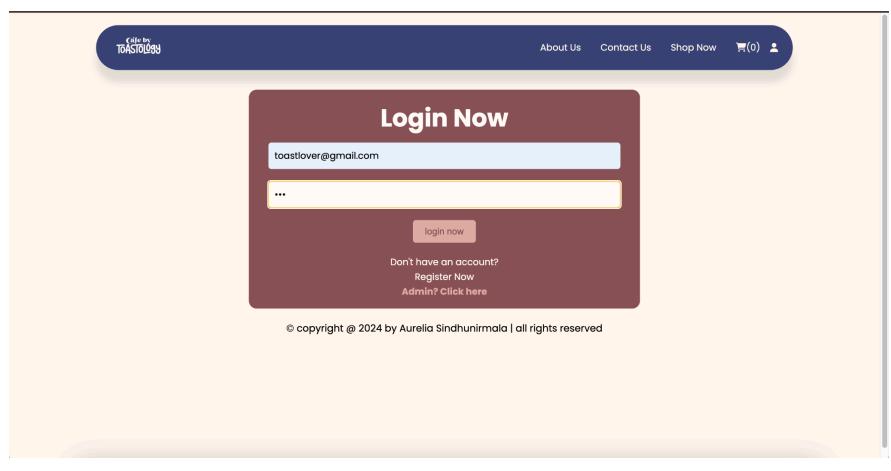


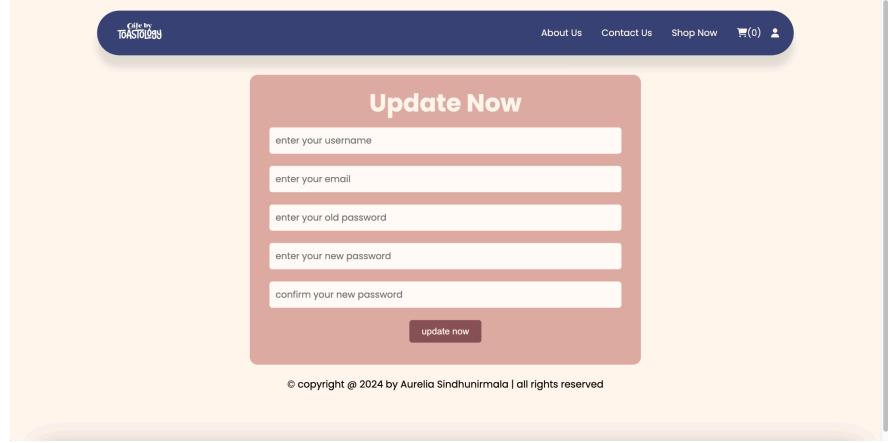
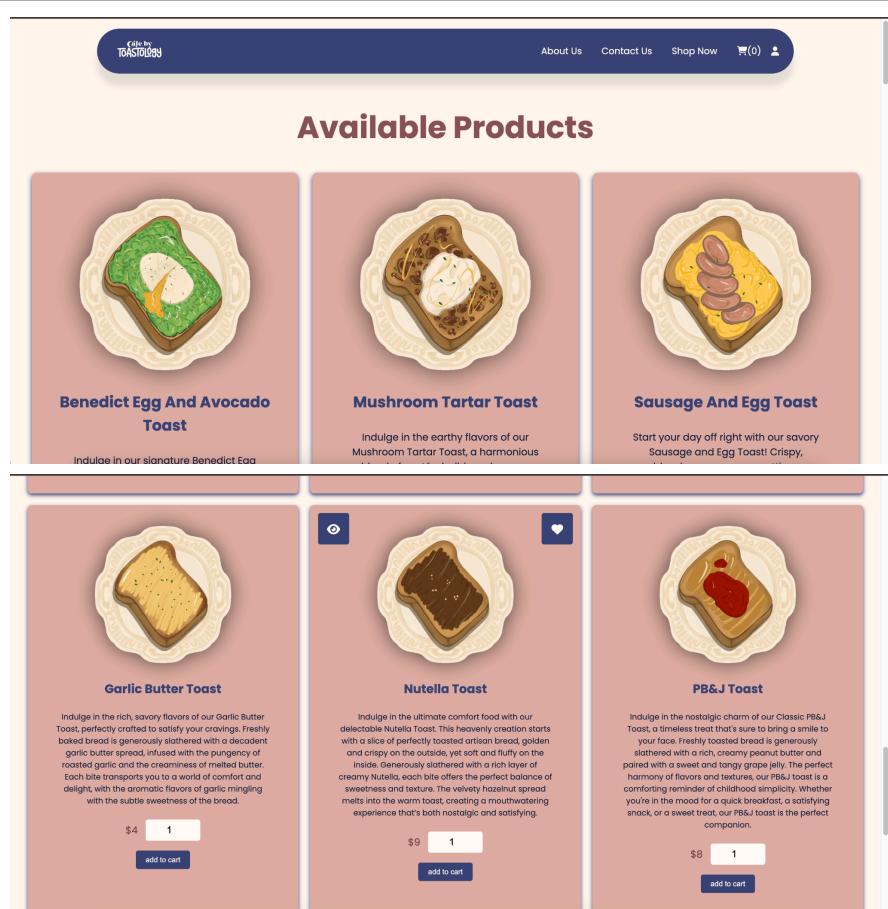
## VI. Screenshots of Application

### User's Interface



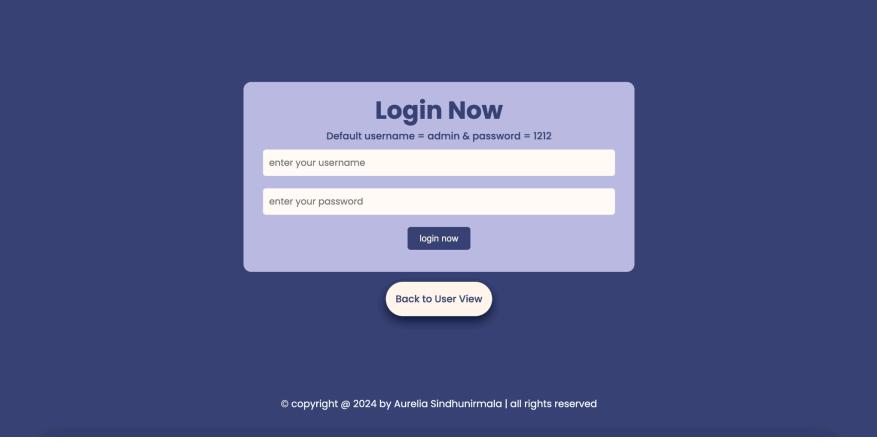
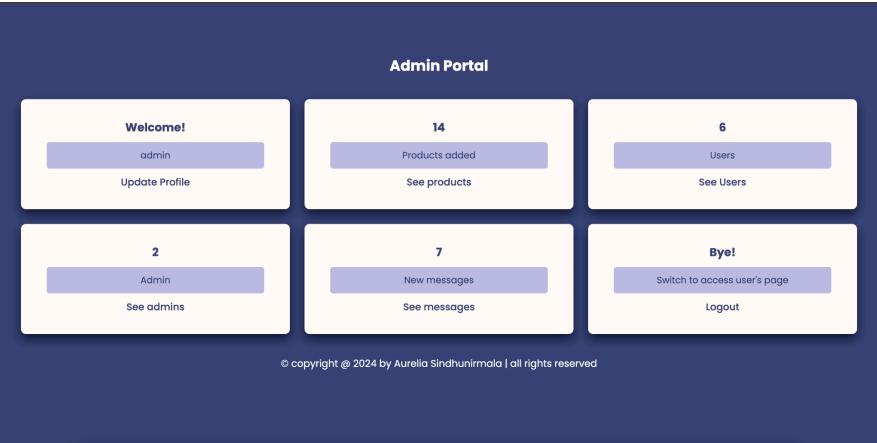
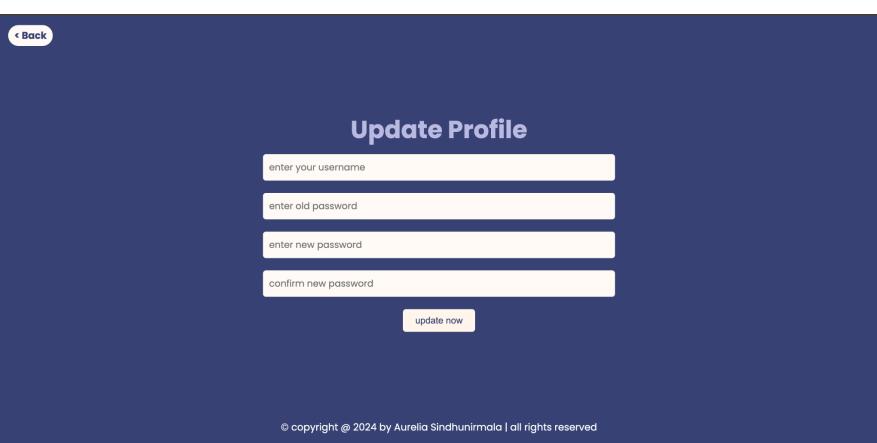
	<p>© copyright @ 2024 by Aurelia Sindhunirmala   all rights reserved</p>
About Us	<p>A virtual cafe where we sell numerous delicious toast. In addition, we also partner up with our team's other project, Toastology: A personality test that defines your specific toast.</p> <p><b>Creator</b></p> <p>Made by: Aurelia Sindhunirmala Second Year Computer Science Student</p> <p>Find me on:</p> <p>in Q M</p> <p>© copyright @ 2024 by Aurelia Sindhunirmala   all rights reserved</p>
Contact Us	<p>Leave Us A Message</p> <p>enter your name</p> <p>enter your email</p> <p>enter your phone number</p> <p>enter your message</p> <p>send message</p> <p>© copyright @ 2024 by Aurelia Sindhunirmala   all rights reserved</p>

Register	 <p>registered successfully, now please login</p> 
Log In	

Edit Account	 <p><b>Update Now</b></p> <p>enter your username enter your email enter your old password enter your new password confirm your new password</p> <p>update now</p> <p>© copyright @ 2024 by Aurelia Sindhunirmala   all rights reserved</p>
Shop Now	 <h3>Available Products</h3> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p><b>Benedict Egg And Avocado Toast</b></p> <p>Indulge in our signature Benedict Egg</p> </div> <div style="text-align: center;">  <p><b>Mushroom Tartar Toast</b></p> <p>Indulge in the earthy flavors of our Mushroom Tartar Toast, a harmonious</p> </div> <div style="text-align: center;">  <p><b>Sausage And Egg Toast</b></p> <p>Start your day off right with our savory Sausage and Egg Toast! Crispy,</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 20px;"> <div style="text-align: center;">  <p><b>Garlic Butter Toast</b></p> <p>Indulge in the rich, savory flavors of our Garlic Butter Toast, perfectly crafted to satisfy your cravings. Freshly baked bread is generously slathered with a decadent garlic butter spread, infused with the pungency of roasted garlic and the creaminess of melted butter. Each bite transports you to a world of comfort and delight, with the aromatic flavors of garlic mingling with the subtle sweetness of the bread.</p> <p>\$4    1    add to cart</p> </div> <div style="text-align: center;">  <p><b>Nutella Toast</b></p> <p>Indulge in the ultimate comfort food with our delectable Nutella Toast. This heavenly creation starts with a slice of perfectly toasted artisan bread, golden and crispy on the outside, yet soft and fluffy on the inside. Generously slathered with a rich layer of creamy Nutella, each bite offers the perfect balance of sweetness and texture. The velvety hazelnut spread melts into the warm toast, creating a mouthwatering experience that's both nostalgic and satisfying.</p> <p>\$9    1    add to cart</p> </div> <div style="text-align: center;">  <p><b>PB&amp;J Toast</b></p> <p>Indulge in the nostalgic charm of our Classic PB&amp;J Toast, a timeless treat that's sure to bring a smile to your face. Freshly toasted bread is generously slathered with a rich, creamy peanut butter and paired with a sweet and tangy grape jelly. The perfect harmony of flavors and textures, our PB&amp;J toast is a comforting reminder of childhood simplicity. Whether you're in the mood for a quick breakfast, a soothing snack, or a sweet treat, our PB&amp;J toast is the perfect companion.</p> <p>\$8    1    add to cart</p> </div> </div>

Cart	<p><b>toastlover's Shopping Cart</b></p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <b>Mushroom Tartar Toast</b>  \$6   1  Sub Total : \$6  <a href="#">delete item</a> </div> <div style="text-align: center;"> <b>Banana Peanut Butter Toast</b>  \$7   1  Sub Total : \$7  <a href="#">delete item</a> </div> <div style="text-align: center;"> <b>S'mores Toast</b>  \$7   1  Sub Total : \$7  <a href="#">delete item</a> </div> </div> <div style="background-color: #2e3436; color: white; padding: 5px; text-align: center;"> Grand Total : \$20  <a href="#" style="color: inherit; text-decoration: none;">Continue Shopping</a>  <a href="#" style="color: inherit; text-decoration: none;">Empty Cart</a>  <a href="#" style="color: inherit; text-decoration: none;">Checkout</a> </div> <p style="font-size: small; margin-top: 5px;">© copyright @ 2024 by Aurelia Sindhuirmala   all rights reserved</p>																				
Checkout	<p><b>Your Orders</b></p> <p>Mushroom Tartar Toast (\$6 x 1)  Banana Peanut Butter Toast (\$7 x 1)  S'mores Toast (\$7 x 1)  Grand Total : \$20</p> <p><b>PLACE YOUR ORDERS</b></p> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Name :</td> <td style="width: 50%;">Phone Number :</td> </tr> <tr> <td><input type="text" value="enter your name"/></td> <td><input type="text" value="enter your number"/></td> </tr> <tr> <td>Email :</td> <td>Payment :</td> </tr> <tr> <td><input type="text" value="enter your email"/></td> <td><input type="text" value="Cash On Delivery"/></td> </tr> <tr> <td>Address line 01 :</td> <td>Address line 02 :</td> </tr> <tr> <td><input type="text" value="e.g. Flat number"/></td> <td><input type="text" value="Street name"/></td> </tr> <tr> <td>City :</td> <td>State:</td> </tr> <tr> <td><input type="text" value="Riverside"/></td> <td><input type="text" value="California"/></td> </tr> <tr> <td>Country :</td> <td>Zip Code :</td> </tr> <tr> <td><input type="text" value="United States"/></td> <td><input type="text" value="e.g. 56400"/></td> </tr> </table> <p style="text-align: center;"><a href="#">place order</a></p> <p style="font-size: small; margin-top: 5px;">© copyright @ 2024 by Aurelia Sindhuirmala   all rights reserved</p>	Name :	Phone Number :	<input type="text" value="enter your name"/>	<input type="text" value="enter your number"/>	Email :	Payment :	<input type="text" value="enter your email"/>	<input type="text" value="Cash On Delivery"/>	Address line 01 :	Address line 02 :	<input type="text" value="e.g. Flat number"/>	<input type="text" value="Street name"/>	City :	State:	<input type="text" value="Riverside"/>	<input type="text" value="California"/>	Country :	Zip Code :	<input type="text" value="United States"/>	<input type="text" value="e.g. 56400"/>
Name :	Phone Number :																				
<input type="text" value="enter your name"/>	<input type="text" value="enter your number"/>																				
Email :	Payment :																				
<input type="text" value="enter your email"/>	<input type="text" value="Cash On Delivery"/>																				
Address line 01 :	Address line 02 :																				
<input type="text" value="e.g. Flat number"/>	<input type="text" value="Street name"/>																				
City :	State:																				
<input type="text" value="Riverside"/>	<input type="text" value="California"/>																				
Country :	Zip Code :																				
<input type="text" value="United States"/>	<input type="text" value="e.g. 56400"/>																				

## Admin's Portal

Log In	 <p><b>Login Now</b> Default username = admin &amp; password = 1212</p> <p>enter your username enter your password login now</p> <p><a href="#">Back to User View</a></p> <p>© copyright @ 2024 by Aurelia SindhuNirmala   all rights reserved</p>						
Home Page	 <p><b>Admin Portal</b></p> <table border="1"> <tbody> <tr> <td><b>Welcome!</b> admin <a href="#">Update Profile</a></td> <td><b>14</b> Products added <a href="#">See products</a></td> <td><b>6</b> Users <a href="#">See Users</a></td> </tr> <tr> <td><b>2</b> Admin <a href="#">See admins</a></td> <td><b>7</b> New messages <a href="#">See messages</a></td> <td><b>Bye!</b> Switch to access user's page <a href="#">Logout</a></td> </tr> </tbody> </table> <p>© copyright @ 2024 by Aurelia SindhuNirmala   all rights reserved</p>	<b>Welcome!</b> admin <a href="#">Update Profile</a>	<b>14</b> Products added <a href="#">See products</a>	<b>6</b> Users <a href="#">See Users</a>	<b>2</b> Admin <a href="#">See admins</a>	<b>7</b> New messages <a href="#">See messages</a>	<b>Bye!</b> Switch to access user's page <a href="#">Logout</a>
<b>Welcome!</b> admin <a href="#">Update Profile</a>	<b>14</b> Products added <a href="#">See products</a>	<b>6</b> Users <a href="#">See Users</a>					
<b>2</b> Admin <a href="#">See admins</a>	<b>7</b> New messages <a href="#">See messages</a>	<b>Bye!</b> Switch to access user's page <a href="#">Logout</a>					
Update Profile	 <p><b>Update Profile</b></p> <p>enter your username enter old password enter new password confirm new password update now</p> <p>© copyright @ 2024 by Aurelia SindhuNirmala   all rights reserved</p>						

**Products**

Add Product

Toast Name (required)

Product Price (required)

Image (required)

Toast description (required)

[add product](#)

**Products Added**

**Products Added**



**Benedict Egg and Avocado Toast**  
\$6

Indulge in our signature Benedict Egg and Avocado Toast, a harmonious blend of flavors and textures. Freshly toasted sourdough bread is topped with a perfectly poached egg, crispy bacon, and creamy avocado slices. Finished with a sprinkle of red pepper flakes and a drizzle of zesty lemon aioli, this toast is a brunch lover's dream come true.

[update](#) [delete](#)



**Mushroom Tartar Toast**  
\$6

Indulge in the earthy flavors of our Mushroom Tartar Toast, a harmonious blend of sautéed wild mushrooms, tangy tartar sauce, and fresh herbs on toasted sourdough bread. Each bite transports you to a forest floor, with the richness of the mushrooms balanced by the brightness of the tartar sauce. Finished with a sprinkle of microgreens and a drizzle of truffle oil, this toast is a true umami delight.

[update](#) [delete](#)



**Update Name**

**Update Price**

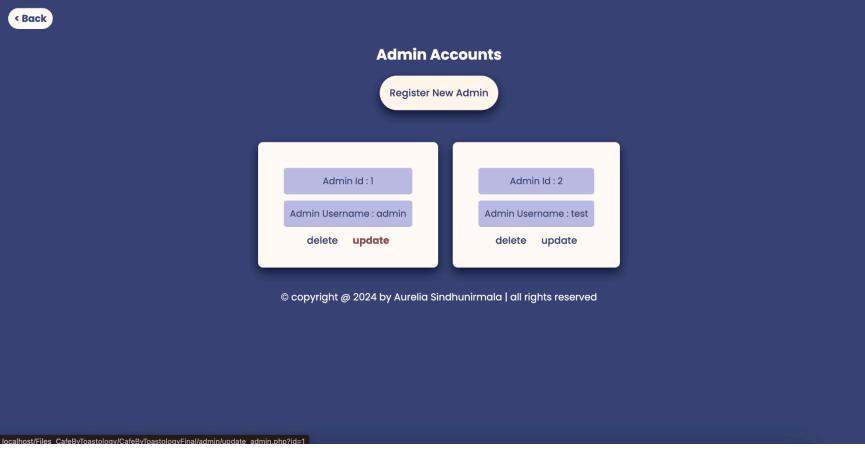
**Update Details**  
Indulge in our signature Benedict Egg and Avocado Toast, a harmonious blend of flavors and textures. Freshly toasted sourdough bread is topped with a perfectly poached egg, crispy bacon, and creamy avocado slices. Finished with a sprinkle of red pepper flakes and a drizzle of zesty lemon aioli, this toast is a brunch lover's dream come true.

**Update Image**

[Update](#) [Cancel](#)

[View All Products](#)

© copyright © 2024 by Avocado Sourdough Inc. All rights reserved.

Users	 <p><b>User accounts</b></p> <table border="1"> <tbody> <tr> <td>User id : 1 Username : test Email : test@gmail.com</td> <td>User id : 3 Username : jake Email : jake@bighit.ent</td> <td>User id : 4 Username : wonwoo Email : wonwoo@bighit.ent</td> </tr> <tr> <td><a href="#">delete</a></td> <td><a href="#">delete</a></td> <td><a href="#">delete</a></td> </tr> <tr> <td>User id : 5 Username : lila Email : lila@ygp.ent</td> <td>User id : 7 Username : winter Email : winter@sm.ent</td> <td>User id : 8 Username : lisa Email : lisa@yg.ent</td> </tr> <tr> <td><a href="#">delete</a></td> <td><a href="#">delete</a></td> <td><a href="#">delete</a></td> </tr> </tbody> </table>	User id : 1 Username : test Email : test@gmail.com	User id : 3 Username : jake Email : jake@bighit.ent	User id : 4 Username : wonwoo Email : wonwoo@bighit.ent	<a href="#">delete</a>	<a href="#">delete</a>	<a href="#">delete</a>	User id : 5 Username : lila Email : lila@ygp.ent	User id : 7 Username : winter Email : winter@sm.ent	User id : 8 Username : lisa Email : lisa@yg.ent	<a href="#">delete</a>	<a href="#">delete</a>	<a href="#">delete</a>
User id : 1 Username : test Email : test@gmail.com	User id : 3 Username : jake Email : jake@bighit.ent	User id : 4 Username : wonwoo Email : wonwoo@bighit.ent											
<a href="#">delete</a>	<a href="#">delete</a>	<a href="#">delete</a>											
User id : 5 Username : lila Email : lila@ygp.ent	User id : 7 Username : winter Email : winter@sm.ent	User id : 8 Username : lisa Email : lisa@yg.ent											
<a href="#">delete</a>	<a href="#">delete</a>	<a href="#">delete</a>											
Admins	 <p><b>Admin Accounts</b></p> <p><a href="#">Register New Admin</a></p> <table border="1"> <tbody> <tr> <td>Admin Id : 1 Admin Username : admin</td> <td>Admin Id : 2 Admin Username : test</td> </tr> <tr> <td><a href="#">delete</a> <a href="#">update</a></td> <td><a href="#">delete</a> <a href="#">update</a></td> </tr> </tbody> </table> <p>© copyright @ 2024 by Aurelia SindhuNirmala   all rights reserved</p>	Admin Id : 1 Admin Username : admin	Admin Id : 2 Admin Username : test	<a href="#">delete</a> <a href="#">update</a>	<a href="#">delete</a> <a href="#">update</a>								
Admin Id : 1 Admin Username : admin	Admin Id : 2 Admin Username : test												
<a href="#">delete</a> <a href="#">update</a>	<a href="#">delete</a> <a href="#">update</a>												
Messages	 <p><b>Messages</b></p> <table border="1"> <tbody> <tr> <td>User id : 1 Name : asd Email : asdg@gmail.com Number : 123 Message : asdfaf</td> <td>User id : 1 Name : qewr Email : test@gmail.com Number : 123 Message : qerw</td> </tr> <tr> <td><a href="#">Delete</a></td> <td><a href="#">Delete</a></td> </tr> <tr> <td>User id : 1 Name : xx Email : adsf@dsf Number : 123</td> <td>User id : 1 Name : fasd Email : fdsaf@fda Number : 1234</td> </tr> </tbody> </table>	User id : 1 Name : asd Email : asdg@gmail.com Number : 123 Message : asdfaf	User id : 1 Name : qewr Email : test@gmail.com Number : 123 Message : qerw	<a href="#">Delete</a>	<a href="#">Delete</a>	User id : 1 Name : xx Email : adsf@dsf Number : 123	User id : 1 Name : fasd Email : fdsaf@fda Number : 1234						
User id : 1 Name : asd Email : asdg@gmail.com Number : 123 Message : asdfaf	User id : 1 Name : qewr Email : test@gmail.com Number : 123 Message : qerw												
<a href="#">Delete</a>	<a href="#">Delete</a>												
User id : 1 Name : xx Email : adsf@dsf Number : 123	User id : 1 Name : fasd Email : fdsaf@fda Number : 1234												

## Database

Forms	<p>The screenshot shows a tree view of the database structure under 'cafeDatabase'. The nodes listed are: admins, cart, categories, messages, orders, products, users, and xref_product_category. The 'xref_product_category' node is highlighted with a yellow background.</p>																																																																																																																																																								
Admins	<p>The screenshot shows a table with columns: id, name, and password. There are two rows of data:</p> <table border="1"> <thead> <tr> <th></th> <th>id</th> <th>name</th> <th>password</th> </tr> </thead> <tbody> <tr> <td>▶</td> <td>1</td> <td>admin</td> <td>618dcdfb0cd9</td> </tr> <tr> <td></td> <td>2</td> <td>test</td> <td>7110eda4d09</td> </tr> <tr> <td>+</td> <td>&lt;Auto </td> <td></td> <td></td> </tr> </tbody> </table>		id	name	password	▶	1	admin	618dcdfb0cd9		2	test	7110eda4d09	+	<Auto																																																																																																																																										
	id	name	password																																																																																																																																																						
▶	1	admin	618dcdfb0cd9																																																																																																																																																						
	2	test	7110eda4d09																																																																																																																																																						
+	<Auto																																																																																																																																																								
Cart	<p>The screenshot shows a table with columns: id, user_id, pid, name, price, quantity, and image. There are 25 rows of data, representing various items in a cart. The table header includes sorting and filtering icons.</p> <table border="1"> <thead> <tr> <th></th> <th>id</th> <th>user_id</th> <th>pid</th> <th>name</th> <th>price</th> <th>quantity</th> <th>image</th> </tr> </thead> <tbody> <tr> <td>▶</td> <td>1</td> <td>3</td> <td>2</td> <td>Mushroom</td> <td>6</td> <td>5</td> <td></td> </tr> <tr> <td></td> <td>3</td> <td>5</td> <td>2</td> <td>Mushroom</td> <td>6</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>6</td> <td>5</td> <td>1</td> <td>Benedict E</td> <td>6</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>7</td> <td>5</td> <td>2</td> <td>Mushroom</td> <td>6</td> <td>3</td> <td></td> </tr> <tr> <td></td> <td>8</td> <td>5</td> <td>3</td> <td>Sausage</td> <td>6</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>9</td> <td>5</td> <td>1</td> <td>Benedict E</td> <td>6</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>10</td> <td>1</td> <td>3</td> <td>Sausage</td> <td>6</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>11</td> <td>1</td> <td>3</td> <td>Sausage</td> <td>6</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>12</td> <td>1</td> <td>6</td> <td>Broccoli</td> <td>5</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>17</td> <td>1</td> <td>8</td> <td>Cinnamon</td> <td>5</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>18</td> <td>1</td> <td>5</td> <td>Banana P</td> <td>7</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>19</td> <td>1</td> <td>7</td> <td>S'mores T</td> <td>7</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>20</td> <td>1</td> <td>7</td> <td>S'mores T</td> <td>7</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>21</td> <td>5</td> <td>18</td> <td>Strawberry</td> <td>11</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>23</td> <td>10</td> <td>2</td> <td>Mushroom</td> <td>6</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>24</td> <td>10</td> <td>5</td> <td>Banana P</td> <td>7</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>25</td> <td>10</td> <td>7</td> <td>S'mores T</td> <td>7</td> <td>1</td> <td></td> </tr> <tr> <td></td> <td>+</td> <td>&lt;Auto </td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		id	user_id	pid	name	price	quantity	image	▶	1	3	2	Mushroom	6	5			3	5	2	Mushroom	6	1			6	5	1	Benedict E	6	1			7	5	2	Mushroom	6	3			8	5	3	Sausage	6	1			9	5	1	Benedict E	6	1			10	1	3	Sausage	6	1			11	1	3	Sausage	6	1			12	1	6	Broccoli	5	1			17	1	8	Cinnamon	5	1			18	1	5	Banana P	7	1			19	1	7	S'mores T	7	1			20	1	7	S'mores T	7	1			21	5	18	Strawberry	11	1			23	10	2	Mushroom	6	1			24	10	5	Banana P	7	1			25	10	7	S'mores T	7	1			+	<Auto					
	id	user_id	pid	name	price	quantity	image																																																																																																																																																		
▶	1	3	2	Mushroom	6	5																																																																																																																																																			
	3	5	2	Mushroom	6	1																																																																																																																																																			
	6	5	1	Benedict E	6	1																																																																																																																																																			
	7	5	2	Mushroom	6	3																																																																																																																																																			
	8	5	3	Sausage	6	1																																																																																																																																																			
	9	5	1	Benedict E	6	1																																																																																																																																																			
	10	1	3	Sausage	6	1																																																																																																																																																			
	11	1	3	Sausage	6	1																																																																																																																																																			
	12	1	6	Broccoli	5	1																																																																																																																																																			
	17	1	8	Cinnamon	5	1																																																																																																																																																			
	18	1	5	Banana P	7	1																																																																																																																																																			
	19	1	7	S'mores T	7	1																																																																																																																																																			
	20	1	7	S'mores T	7	1																																																																																																																																																			
	21	5	18	Strawberry	11	1																																																																																																																																																			
	23	10	2	Mushroom	6	1																																																																																																																																																			
	24	10	5	Banana P	7	1																																																																																																																																																			
	25	10	7	S'mores T	7	1																																																																																																																																																			
	+	<Auto																																																																																																																																																							

Categories	A screenshot of a database table titled "Categories". The table has three columns: "id", "category_id", and "categories". There are two data rows: one with id 1, category_id 1, and categories "breakfast"; and another with id 2, category_id 2, and categories "dessert". The table is displayed in a dark-themed interface with standard column headers.
Messages	A screenshot of a database table titled "cafeDatabase.messages - cafeDatabase". The table has six columns: "id", "user_id", "name", "email", "number", and "message". There are eight data rows. The first four rows belong to user_id 1, with names asd, qewr, xx, and fasd respectively. The next three rows belong to user_id 3, with names jake, jake, and xx. The last row is a placeholder with id 8 and user_id <Auto . The table is displayed in a dark-themed interface with standard column headers.
Orders	A screenshot of a database table titled "cafeDatabase.orders - cafeDatabase". The table has ten columns: "id", "user_id", "name", "number", "email", "method", "address", "total_products", "total_price", "placed_on", and "payment". There is one data row for user_id 4, named Wonwoo, with details: number 909567785, email wonwoo@credit card, method flat no. BigHi, address Benedict Egg and /12, total_products 12, total_price 06/09/24, and payment pending. The table is displayed in a dark-themed interface with standard column headers.
Products	A screenshot of a database table titled "cafeDatabase.products - cafeDatabase". The table has five columns: "id", "name", "details", "price", and "image_1". There are 19 data rows, each representing a type of toast. The first few rows include: Benedict Egg and Avocado Toast (details Indulge in our signature, price 6, image 1.PNG); Mushroom Tartar Toast (details Indulge in the earthy fla, price 6, image 2.PNG); Sausage and Egg Toast (details Start your day off right, price 6, image 3.PNG); Tuna Salad Toast (details Dive into the freshness, price 5, image 4.PNG); Banana Peanut Butter Toast (details Indulge in the classic, price 7, image 5.PNG); Broccoli and Hummus Toast (details Get your daily dose of, price 5, image 6.PNG); Smores Toast (details Indulge in the classic, price 7, image 9.PNG); Cinnamon Butter Toast (details Indulge in the warm, co, price 5, image 7.PNG); Tomato Avocado Toast (details Indulge in the warm, co, price 8, image 8.PNG); Garlic Butter Toast (details Indulge in the rich, sav, price 4, image 10.PNG); Nutella Toast (details Indulge in the ultimate, price 9, image 12.PNG); PB&J Toast (details Indulge in the nostalgic, price 8, image 14.PNG); Strawberry Cream Cheese Toast (details Savor the delightful fu, price 11, image 11.PNG); and Blueberry Cheese Toast (details Indulge in the perfect b, price 11, image 13.PNG). The table is displayed in a dark-themed interface with standard column headers.

Users				
	id	name	email	password
▶	1	test	test@gma...@gma...	7110eda4d09...
	3	jake	jake@bigl...	12748ff9b92c...
	4	wonwoo	wonwoo@...	c1777d8e848...
	5	lia	lia@jyp.er...	4a5b9bbe3c6...
	7	winter	winter@si...	8d3c3084a84...
	8	lisa	lisa@yg.e...	850b36862fd1...
	10	toastlover	toastlover...	6216f8a75fd5...
	11	admin	jake@bigl...	6216f8a75fd5...
+	<Auto			

## VII. Demo Video

User View:  [User.mov](#)

Admin View:  [Admin.mov](#)

## VIII. Preview Code

<Index.php>

```
<?php

    include 'attributes/connect.php';
    session_start();

    if(isset($_SESSION['user_id'])){
        $user_id = $_SESSION['user_id'];
    }else{
        $user_id = '';
    }

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cafe by Toastology</title>
```

```

<!-- Icon -->
<link rel="icon" href="pic/icon.png">
<link rel=" shortcut icon" href="pic/icon.png">

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

<!-- Font Awesome -->
<link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.2/css/all.min.css"
      integrity="sha512-SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9QevSCWr3W6A==" crossorigin="anonymous"
      referrerPolicy="no-referrer" />
<link rel="stylesheet"
      href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" >

<!-- Font -->
<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=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
</head>

<body>
<?php include ('attributes/header.php') ?>

<!-- <div class="home-bg"> -->
<section class="home">
    <div class = "home">
        <h3> Welcome to </h3>
        <img src = "pic/name.png" alt="" class ="logo">
        <p>Find out which toast perfectly complements your taste!</p>
    </div>

    <div class = "test">
        <h3> Having trouble which toast to purchase?</h3>
        <p> Take this test and try your personal toast!</p>
        <div class="toast">

```

```

        
        
    </div>
    <div class = "test-btn">
        <a href = "https://bit.ly/toastology"> <btn> Take the test</btn>
    </a>
    </div>
</div>

</section>
<!-- </div> -->

<?php include ('attributes/footer.php')?>
<script src="js/script.js"></script>

</body>
</html>

```

<User\_login.php>

```

<?php
    include ('attributes/connect.php');
    session_start();

    if (!isset($_SESSION['user_id'])) {
        header('location:user_login.php');
        exit();
    }

    ini_set('display_errors', 1);
    ini_set('display_startup_errors', 1);
    error_reporting(E_ALL);

    if(isset($_SESSION['user_id'])){
        $user_id = $_SESSION['user_id'];
    }else{
        $user_id = '';
    };

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


```

```

$email = $_POST['email'];
// if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
//     $message[] = "Invalid email format";
//
$pass = sha1($_POST['pass']);

$select_user = $conn->prepare("SELECT * FROM `users` WHERE email = ? AND
password = ?");
$select_user->execute([$email, $pass]);
$row = $select_user->fetch(PDO::FETCH_ASSOC);

if($select_user->rowCount() > 0) {
    $_SESSION['user_id'] = $row['id'];

    setcookie("email", $row['email'], time() + (86400 * 30), "/");
    // expires in 1 hour

    header('Location:index.php');
    exit();
} else{
    $message[] = 'incorrect username or password!';
}
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cafe by Toastology</title>

    <!-- Icon -->
    <link rel="icon" href="pic/icon.png">

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

    <!-- Font Awesome -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.2/css/all.min.css"
" data-bbox="171 875 880 900"/>

```

```

integrity="sha512-SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3p
BnMFcV7oQPJkl9QevSCWr3W6A==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
<link rel="stylesheet"
href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" >

<!-- Font -->
<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=Poppins:ital,wght@0,100;0,200;0,
300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,7
00;1,800;1,900&display=swap" rel="stylesheet">
</head>
<body>
<?php include ('attributes/header.php')?>

<section class="user-form2">
<form action="" method="post">
<h3>Login Now</h3>
<input type="email" name="email" required placeholder="enter your
email" maxlength="50" class="box" oninput="this.value =
this.value.replace(/\s/g, '')">
<input type="password" name="pass" required placeholder="enter your
password" maxlength="20" class="box" oninput="this.value =
this.value.replace(/\s/g, '')">
<input type="submit" value="login now" class="btn" name="submit">
<p>Don't have an account?</p>
<a href="user_reg.php" class="optionn-btn">Register Now</a>
<div class = "admin-login">
<a href="admin/login.php" class="optionn-btn">Admin? Click
here</a>
</div>
</form>
</section>

<?php include ('attributes/footer.php')?>
<script src="js/script.js"></script>
</body>
</html>

```

<Shop.php>

```

<?php
    include ('attributes/connect.php');
    session_start();

    if(isset($_SESSION['user_id'])){
        $user_id = $_SESSION['user_id'];
    }else{
        header('Location: user_login.php');
    };
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cafe by Toastology</title>

    <!-- Icon -->
    <link rel="icon" href="pic/icon.png">

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

    <!-- Font Awesome -->
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.2/css/all.min.css"
    "integrity="sha512-SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJk19QevSCWr3W6A==" crossorigin="anonymous"
    referrerPolicy="no-referrer" />
    <link rel="stylesheet"
    href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" >

    <!-- Font -->
    <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=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
</head>

```

```

<body>
    <?php include ('attributes/header.php')?>

    <section class="products">
        <h1 class="heading">Available Products</h1>
        <div class="box-container">
            <?php
                $select_products = $conn->prepare("SELECT * FROM `products`");
                $select_products->execute();
                if($select_products->rowCount() > 0){
                    while($fetch_product =
$select_products->fetch(PDO::FETCH_ASSOC)) {
                        ?>
                        <form action="" method="post" class="box">
                            <input type="hidden" name="pid" value="<?= $fetch_product['id'];
?>">
                            <input type="hidden" name="name" value="<?=
$fetch_product['name']; ?>">
                            <input type="hidden" name="details" value="<?=
$fetch_product['details']; ?>">
                            <input type="hidden" name="price" value="<?=
$fetch_product['price']; ?>">
                            <input type="hidden" name="image" value="<?=
$fetch_product['image']; ?>">
                            <button class="fas fa-heart" type="submit"
name="add_to_wishlist"></button>
                            <a href="#" class="fas fa-eye"></a>
                            
                            alt="">
                            <div class="name"><?= $fetch_product['name']; ?></div>
                            <div class="details"><?= $fetch_product['details']; ?></div>
                            <div class="flex">
                                <div class="price"><span>$</span><?=
$fetch_product['price']; ?></div>
                                    <input type="number" name="qty" class="qty" min="1" max="99"
onkeypress="if(this.value.length == 2) return false;" value="1">
                                </div>
                                <input type="submit" value="add to cart" class="btn"
name="add_to_cart">
                            </form>
                        <?php
                    }
                }
            </div>
        </section>
    </body>

```

```

        }else{
            echo '<p class="empty">no products found!</p>';
        }
    ?>
</div>

</section>

<?php
if(isset($_POST['add_to_cart'])){
    if($user_id == ''){
        header('user_login.php');
    } else{
        $pid = $_POST['pid'];
        $name = $_POST['name'];
        $price = $_POST['price'];
        $qty = $_POST['qty'];
        $image = $_POST['image'];

        $insert_cart = $conn->prepare("INSERT INTO `cart` (user_id, pid,
name, price, quantity, image) VALUES(?, ?, ?, ?, ?, ?)");
        $insert_cart->execute([$user_id, $pid, $name, $price, $qty,
$image]);

        $message[] = 'added to cart!';
    }
}
?>

<?php include ('attributes/footer.php')?>
<script src="js/script.js"></script>
</body>
</html>

```

<Products.php>

```

<?php
include ('../attributes/connect.php');
include ('../attributes/admin_msg.php');
include ('back.php');
session_start();

```

```

$admin_id = $_SESSION['admin_id'];

if(!isset($admin_id)){
    header('location:admin_login.php');
}

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

    $name = $_POST['name'];
    $price = $_POST['price'];
    $details = $_POST['details'];

    $image_01 = $_FILES['image_1']['name'];
    $image_size_01 = $_FILES['image_1']['size'];
    $image_tmp_name_01 = $_FILES['image_1']['tmp_name'];
    $image_folder_01 = '../uploaded_img/'. $image_01;

    $select_products = $conn->prepare("SELECT * FROM `products` WHERE name = ?");
    $select_products->execute([$name]);

    if($select_products->rowCount() > 0){
        $message[] = 'product name already exist!';
    }else{
        if (move_uploaded_file($image_tmp_name_01, $image_folder_01)) {
            $insert_products = $conn->prepare("INSERT INTO `products` (name, details, price, image_1) VALUES(?, ?, ?, ?)");
            $insert_products->execute([$name, $details, $price, $image_01]);
            if ($insert_products) {
                // echo "Product inserted successfully.";
                $message[] = 'Product inserted successfully';
            } else {
                // echo "Insert product failed";
                $message[] = 'Failed to insert product';
            }
        } else {
            $message[] = 'Failed to upload image. Ensure the images/directory is writable.';
        }
    }
}

```

```

if(isset($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    $delete_product_image = $conn->prepare("SELECT * FROM `products` WHERE id = ?");
    $delete_product_image->execute([$delete_id]);

    $fetch_delete_image = $delete_product_image->fetch(PDO::FETCH_ASSOC);
    unlink('../uploaded_img/'.$fetch_delete_image['image_1']);

    $delete_product = $conn->prepare("DELETE FROM `products` WHERE id = ?");
    $delete_product->execute([$delete_id]);

    $delete_cart = $conn->prepare("DELETE FROM `cart` WHERE pid = ?");
    $delete_cart->execute([$delete_id]);
    header('location:products.php');
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cafe by Toastology - Products</title>

    <!-- Icon -->
    <link rel="icon" href="../pic/icon.png">

    <!-- CSS -->
    <link rel="stylesheet" href="../css/admin.css">

    <!-- Font Awesome -->
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.2/css/all.min.css"
        integrity="sha512-SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9QevSCWr3W6A==" crossorigin="anonymous"
        referrerPolicy="no-referrer" />
    <link rel="stylesheet"
        href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" >

    <!-- Font -->

```

```

<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=Poppins:ital,wght@0,100;0,200;0,
  300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,7
  00;1,800;1,900&display=swap" rel="stylesheet">
</head>
<body>
  <section class = "inventory">
    <section class="add-products">
      <h1 class="heading">Add Product</h1>

      <form action="" method="post" enctype="multipart/form-data">
        <div class="flex">
          <div class="inputBox">
            <span>Toast Name (required)</span>
            <input type="text" class="box" required maxlength="100"
placeholder="enter product name" name="name">
          </div>
          <div class="inputBox">
            <span>Product Price (required)</span>
            <input type="number" min="0" class="box" required
max="9999999999" placeholder="enter product price"
onkeypress="if(this.value.length == 10) return false;" name="price">
          </div>
          <div class="inputBox">
            <span>Image (required)</span>
            <input type="file" name="image_1" accept="image/jpg,
image/jpeg, image/png" class="box" required>
          </div>
          <div class="inputBox">
            <span>Toast description (required)</span>
            <textarea name="details" placeholder="enter product
details" class="box" required maxlength="500" cols="30" rows="5"></textarea>
          </div>
        </div>
        <input type="submit" value="add product" class="btn"
name="add_product">
      </form>
    </section>

    <section class="show-products">

```

```

<h1 class="heading">Products Added</h1>

<div class="products">
    <?php
        $select_products = $conn->prepare("SELECT * FROM
`products`");
        $select_products->execute();
        if($select_products->rowCount() > 0){
            while($fetch_products =
$select_products->fetch(PDO::FETCH_ASSOC)) {
                ?>
                <div class="box">
                    
                    <div class="name"><?= $fetch_products['name']; ?></div>
                    <div class="price">$<span><?= $fetch_products['price'];
?></span></div>
                    <div class="details"><span><?=
$fetch_products['details']; ?></span></div>
                    <div class="flex-btn">
                        <a href="update_product.php?update=<?=
$fetch_products['id']; ?>" class="option-btn">update</a>
                        <a href="products.php?delete=<?=
$fetch_products['id']; ?>" class="delete-btn" onclick="return confirm('delete
this product?');">delete</a>
                    </div>
                </div>
            <?php
                }
            } else{
                echo '<p class="empty">no products added yet!</p>';
            }
        ?>
    </div>
</section>
</section>

<?php include ('../attributes/footer.php')?>
<script src="../js/admin_script.js"></script>
</body>
</html>

```

