ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	10: Website Design & Development		
Submission date	Monday, May 2, 2022	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Pham Trung Kien	**Student ID**	GCH200815
Class	GGH1004	**Assessor name**	Dinh Duc Manh

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	Kien

Grading grid

P5	P6	P7	M4	M5	D2	D3

☼ **Summative Feedback:**		☼ **Resubmission Feedback:**	
Grade:	**Assessor Signature:**		**Date:**
Signature & Date:			

Table of Contents

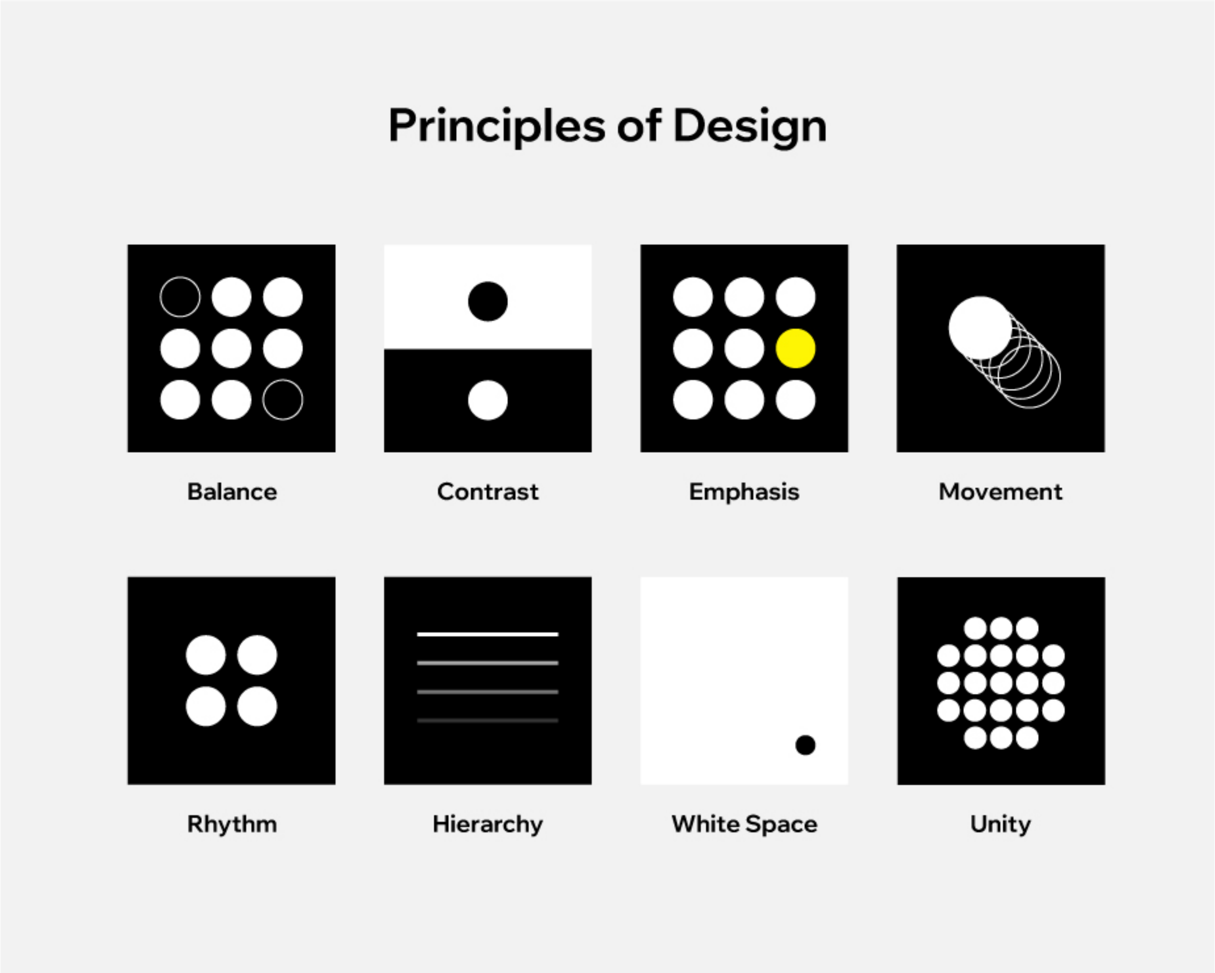
I. Introduction

The author was tasked with acting as a full-stack developer at the request of the MWS company, designing and developing an interactive website with a focus on producing a simple set of dynamic web pages that represent good design principles. This website will be used as an e-commerce platform for the subwayfood.co company's chain of Subway Struck restaurants. As a result, this is a report on the design process, website implementation, and results evaluation.

II. Website design principles, standards and guideline

1. Principle

(Marianne, 2021) states that an effective website design should accomplish its goal of communicating a specific message while also engaging the visitor. There are numerous key factors that influence how a website is perceived when it is designed. A well-designed website can aid in the development of trust and the prompting of visitors to take action. Making sure the website design is optimized for usability and ease of use is an important part of creating a great user experience. Some principles to consider when designing a website are listed below.



1.1 Website purpose

The website must satisfy the user's requirements. On all pages, having a simple, clear intention will assist the user in interacting with what it offers. Websites can serve a variety of functions, but there are a few core functions that all websites must fulfill (Marianne, 2021):

- Describing Expertise
- Building Your Reputation
- Generating Leads
- Sales and After Care

1.2. Simpicity

When it comes to user experience and website usability, simplicity is the way to go. The methods listed below can help to achieve simplicity through design (Marianne, 2021):

- Colour: Color has the ability to convey messages and elicit emotional responses.
- Type: On the website, typography is very important. It draws attention and serves as a visual representation of the brand's voice.
- Imagery: Imagery encompasses all visual aspects of communication. Still photography, illustration, video, and all forms of graphics fall under this category. All imagery should be expressive, capture the spirit of the company, and serve as the embodiment of the brand's personality.

1.3. Navigation

The navigation system used on websites allows visitors to interact and find what they are looking for. The ability to navigate a website is critical to retaining visitors. Visitors will abandon your website if the navigation is difficult to understand. It is critical to keep navigation simple, intuitive, and consistent across all pages (Marianne, 2021).

1.4. Visual hierarchy

The arrangement of elements in descending order of importance is known as visual hierarchy. This is accomplished through the use of size, color, imagery, contrast, typography, whitespace, texture, and style. The establishment of a focal point, which shows visitors where the most important information is, is one of the most important functions of visual hierarchy (Marianne, 2021).

1.5. Content

An effective website combines excellent design with excellent content. Great content can attract and influence visitors and convert them into customers by using compelling language (Marianne, 2021).

1.6. Grid based layout

Grids aid in the organization of your design and content. The grid assists in aligning and cleaning up elements on the page. The grid-based layout organizes content into a clean rigid grid structure with columns, sections that line up and feel balanced, and imposes order, resulting in an aesthetically pleasing website (Marianne, 2021).

1.7. Load time

Visitors will leave if they have to wait for a website to load. Nearly half of web visitors expect a site to load in two seconds or less, and they may abandon a site that does not load within three seconds. Optimizing the size of images will help the site load faster (Marianne, 2021).

1.8. Mobile friendly

More people are browsing the web on their phones or other mobile devices. It is critical to consider designing the website with a responsive layout that can adjust to different screen sizes (Marianne, 2021).

2. Standard and Guidline

According to (Anon., 2018), International Web design standards are rules and guidelines established by the World Wide Web Consortium (W3C) to promote consistency in the outline code that constitutes a web page. Essentially, it is the rule for the increase dialect that determines how a web page appears in a visitor's program window. The followings are a variety of rules of web design standards:

2.1. Logo on the Top left

One of the first and most important elements of any company's web design is its logo. Much is dependent on the style and aesthetic content of the logo type, but its placement on the site is also important. As a result, the top marketing websites all include a logo in the top left corner of every page on the site (Anon., 2020).

2.2. Contact information layout

The contact information on your site is critical because it allows users to contact the owner for further collaboration. The likelihood of future collaboration with the owner's company is determined by the location of their contact information. Almost half of all websites have a contact button or a link to them in the right corner of each page; this is considered common and standard placement (Anon., 2020).

2.3. Navigation layout

The website's navigation system is also important. The location of the layout is determined by how convenient it is for the user to be on the site and whether or not they wish to return to it. It is critical to consider a navigation system that will allow your website's visitors to easily navigate through the pages. In 88 percent of cases, the main navigation appears in the header at the top of every page, making horizontal top-level navigation a web design standard (Anon., 2020).

2.4. Homepage slideshow

Homepage slideshows are a fascinating way to make a first impression. It is a simple method of displaying any type of content that the owner wishes his or her users to see. It is frequently one of the most effective means of communicating information. A homepage slideshow (also known as a carousel) with a rotating series of images and messages can be found on more than 60% of websites (Anon., 2020).

2.5. Call to action

The call to action buttons can be as simple as a single click, a subscription, access to the profile, registering to receive a trial version of the product, or even purchasing. By creating the right design for these CTA buttons, you will generate significantly more traffic to the site than if they were not present. The call to action button is an essential component of any website design. During the opening of the home page, more than 70% of the websites had a call to action element within sight. This standard is strongly advised (Anon., 2020).

2.6. Site search

The ability to search for information on the site is one of the most important principles of usability. The search bar is already one of the standard provisions in the design development process. More than half of all websites include a search box in the profile header, allowing users to easily find something by simply typing a word into the search box (Anon., 2020).

2.7. Newsletter subscribe box

Users can receive more personalized information by subscribing to the site. Users can sign up for the newsletter and receive it by mail or phone. A sign-up box is used on more than 60% of websites, making it a web design standard (Anon., 2020).

2.8. Social media icons

In the footer of more than 70% of websites, there are icons for social media websites. This simplifies the user's task because instead of searching for the company's or the site's social networks, the user can simply click on these social networks and quickly look at them (Anon., 2020).

2.9. Responsive web design

According to the most recent data, mobile devices account for 52.6 percent of global website traffic. As a result, a website that can adapt to different screen sizes will be extremely beneficial in attracting new users (Anon., 2020).

3. Principles, Standards and Guidelines in the project

3.1. Principle

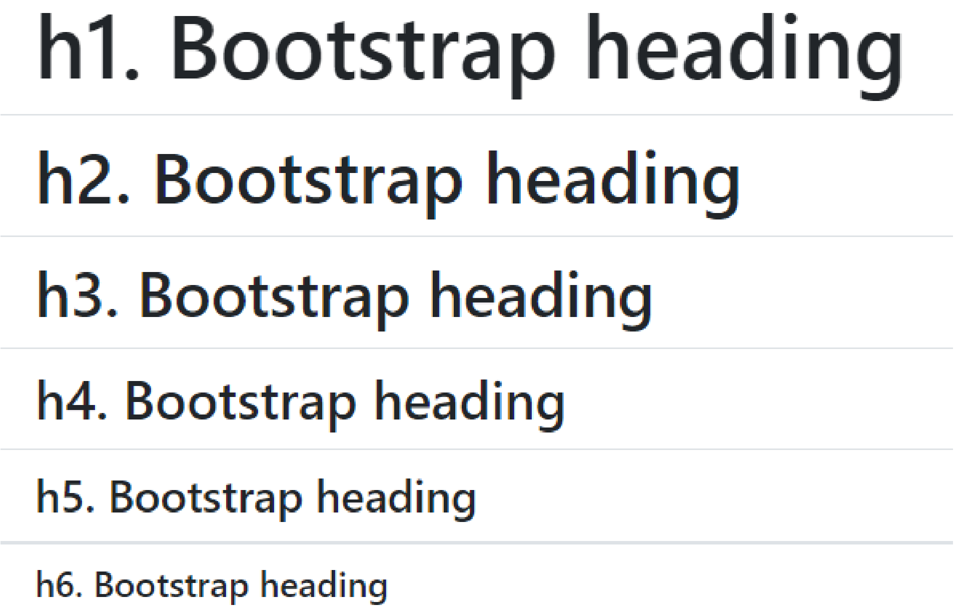
Website purpose

The website is designed as an e-commerce platform to meet the needs of the Subwayfood.co company. People can access and order the restaurant's dishes from home here.

Simplicity

- Colour: Color is one of the elements that tell the style of a website, so on subwayfood.co's website, we use three main color codes to promote simplicity and user-friendliness: #a56060, #f0f8ff, and #000000.

- Type: The font is another element that reveals a website's style, as the site does not use any fancy fonts other than Bootstrap's default.



h1. Bootstrap heading

h2. Bootstrap heading

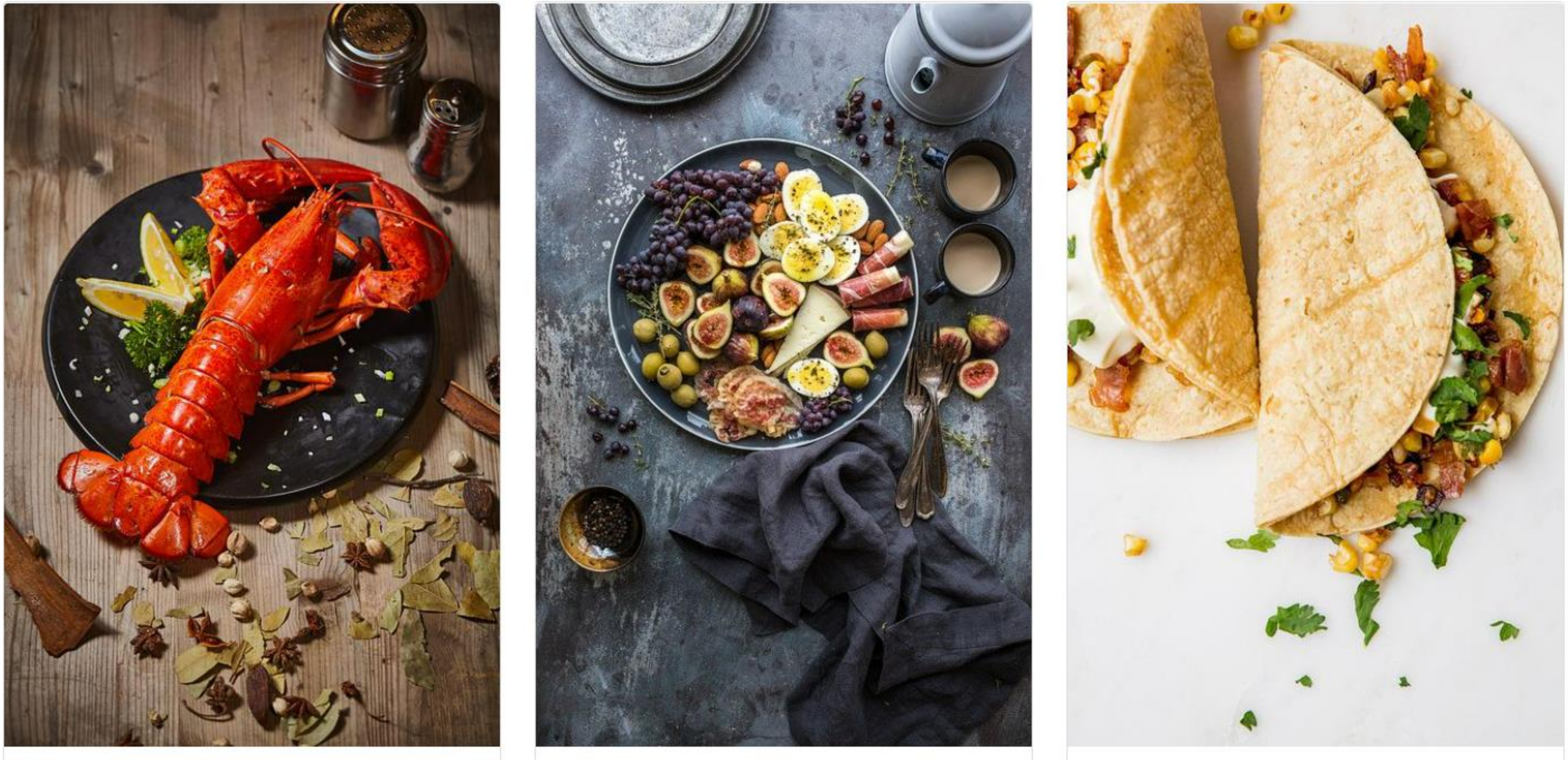
h3. Bootstrap heading

h4. Bootstrap heading

h5. Bootstrap heading

h6. Bootstrap heading

- Imagery: The display image is the final factor to consider; it is not an exaggeration to say that the image is the soul of any website. A web page will be spectacular if it is accompanied by high-quality images of the appropriate size. The author also designed the website, which features high-quality images taken in an artistic manner.



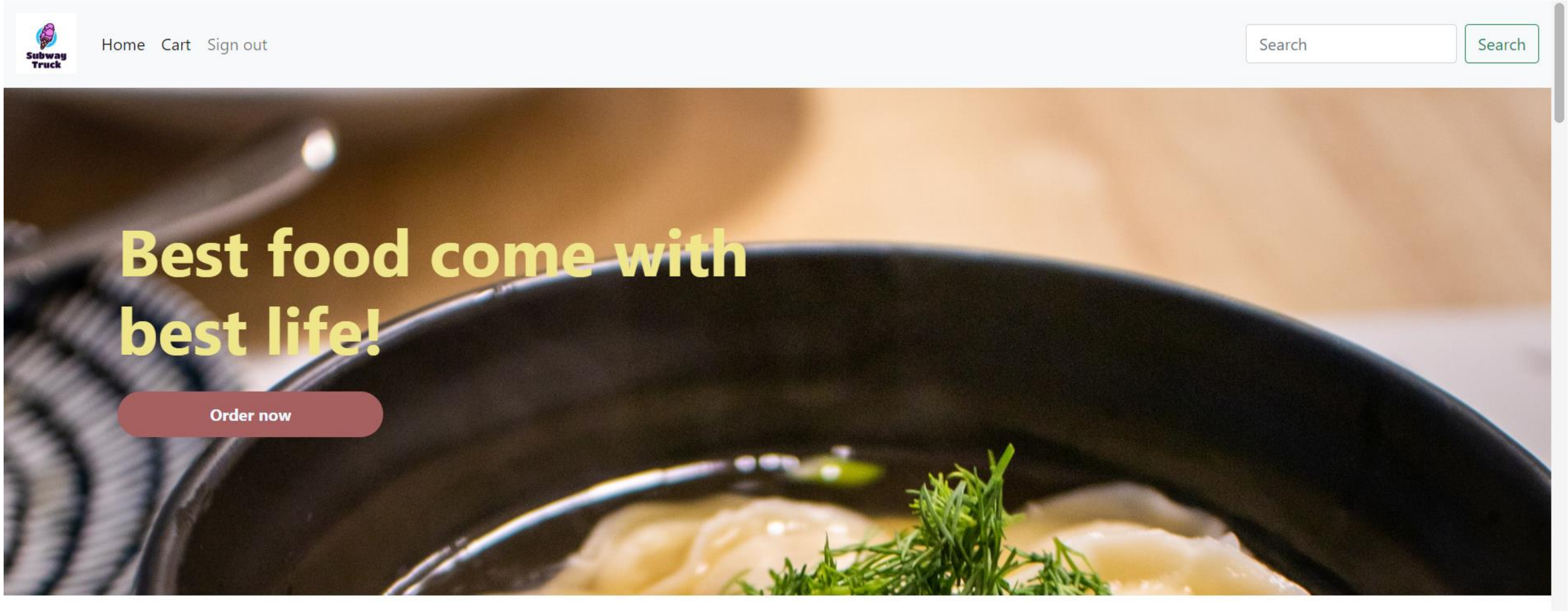
Grid-based layout

The website follows Bootstrap's grid principle, a CSS framework, the items will be divided into small cards in a large grid.

3.2. Standard and Guideline

Logo on the Top left

The website's logo is placed in the upper left corner, this position is easy to attract the attention of users thereby increasing brand recognition.



Contact information layout

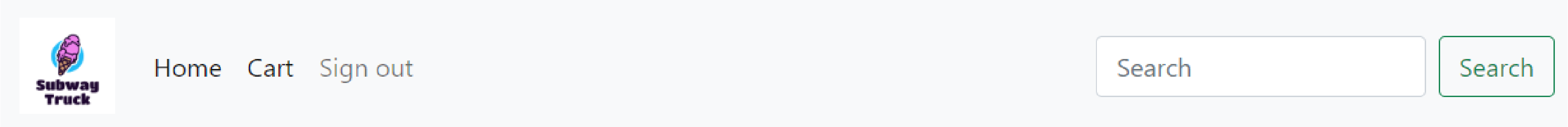
Users can search for information about the company and the subway truck restaurant chain to contact at the footer bar at the bottom of the website.

Contact us

Location	Telephone	Hours
Golden Park Tower, 2 Pham Van Bach, Yen Hoa, Cau Giay, Hanoi	098155808	8:00 - 22:00 Mon - Fri | 9:00 - 20:00 Sat & Sun

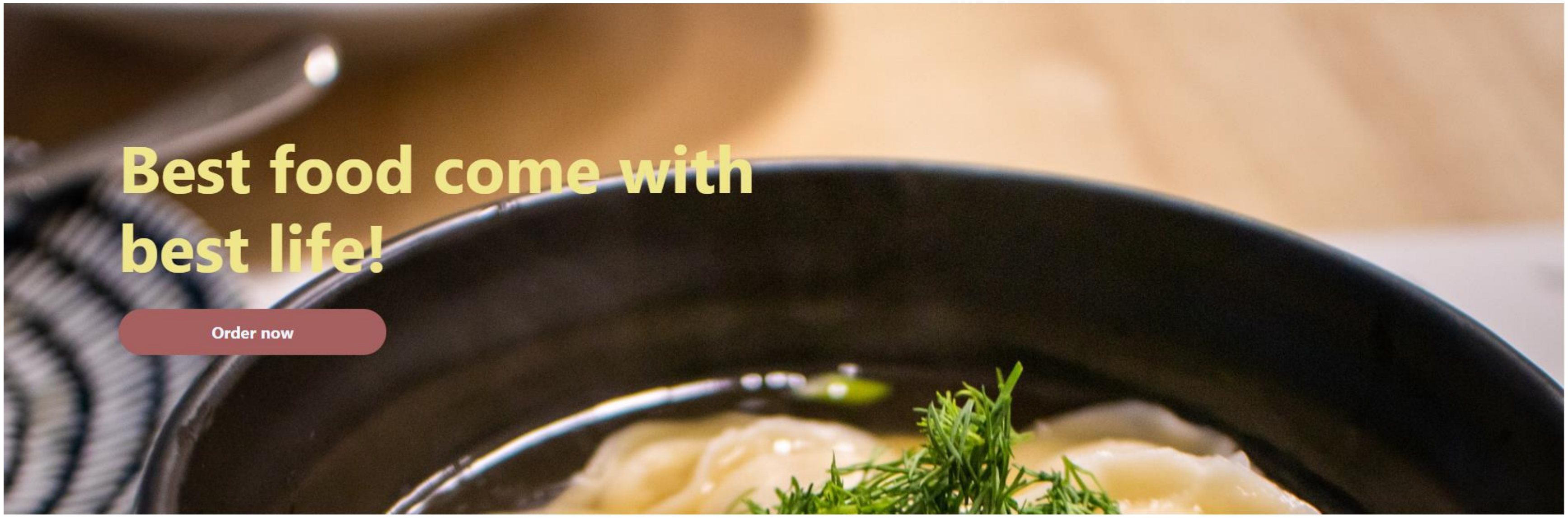
Navigation layout

The site has a navigation bar placed at the top, where the layout is arranged in the order of the logo first, then the items included in the site such as home button, shopping cart, sign out button, and search box.



Call to action

In the Subway Food website there is a call-to-action button that takes the user quickly to the restaurant's menu. This button is located in the slide bar.



III. Design Document

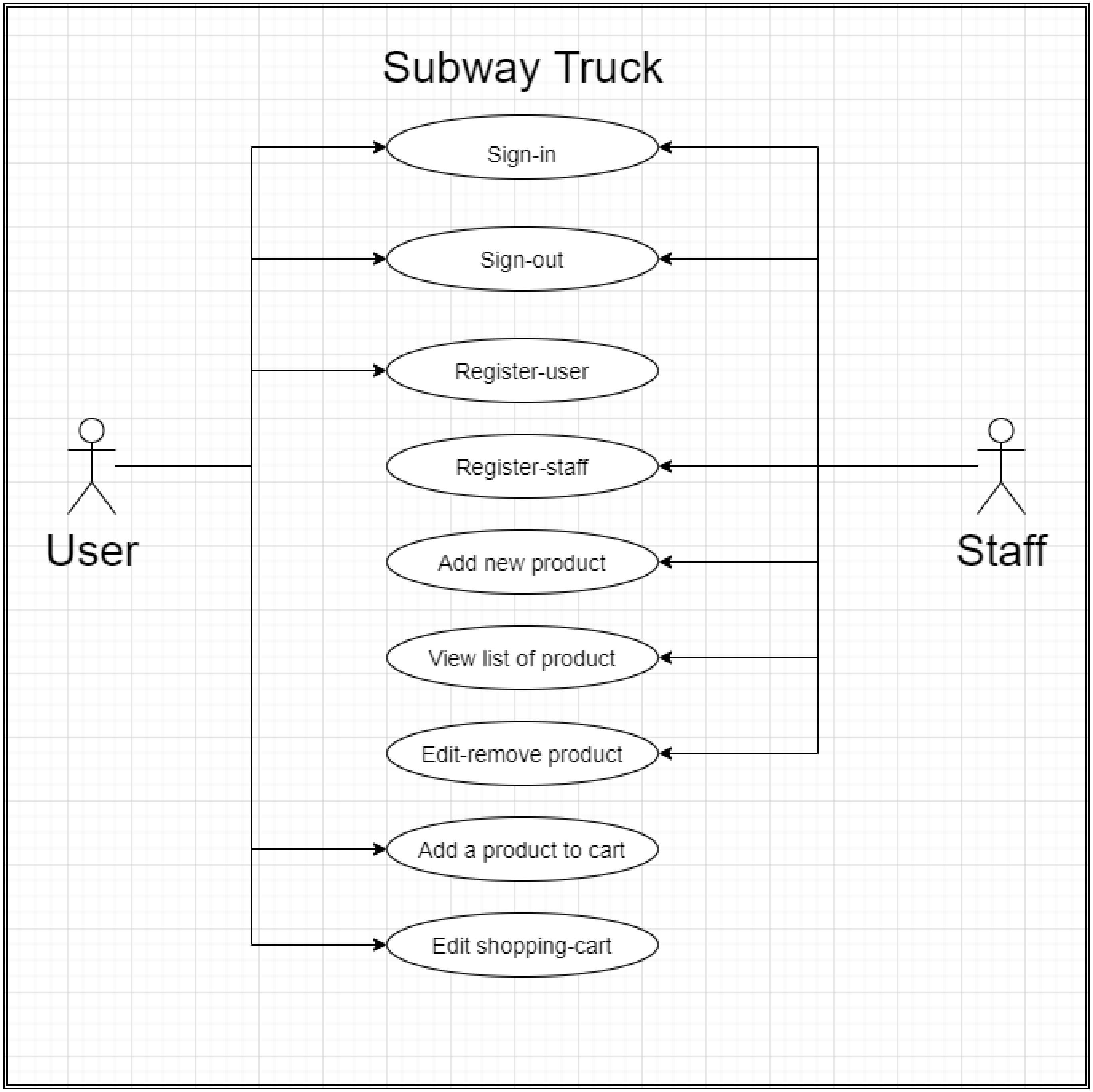
1. User requirement

- As a user, I want a website to order online so that I can save a significant amount of time to stay at home.

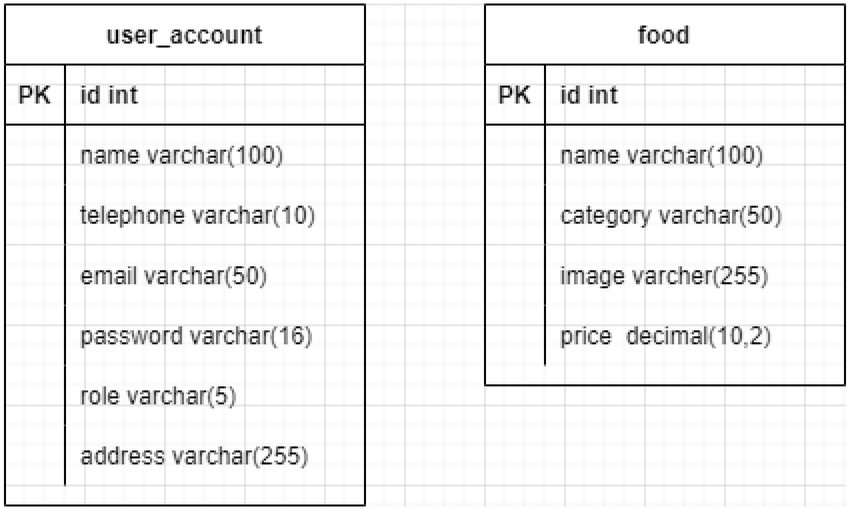
- As a user, I would like the website to have the login and register feature, so that the website will recognize who I am.
- As an admin or staff, I would like to have the private side for staff only to upload, update, remove item shown on the user page, so that staff work and user action will not cause unintentionally conflics.
- As a user I want to order right on the website of the restaurant and all the meals chosen will be stored in the cart. So that will be easy to shop around on the website and choose whatever I want.
- As a user, I want to remove the dish that I no longer want to make payment for so that will be so practical.
- As a user, I want to have the purchase feature so that I can make the payment for the items chosen.
- As a user, I want my shopping cart is displayed as a table form including price for each item, total quantity, and price. So that I can keep track at the total of the bill.
- As a staff, I want to have the register feature as usr but only for staff, so that I can create new account for new staffs.

Task	User	Staff
Sign in	●	●
Sign up	●	●
Side based on role	●	●
Add a dish to cart	●	
Remove a dish from cart	●	
Add a new dish list list		●
Remove a dish from list		●
Make payment	●	
View list of dishes		●

2. Use-case diagram


Subway Truck
Sign-in
Sign-out
Register-user
Register-staff
Add new product
View list of product
Edit-remove product
Add a product to cart
Edit shopping-cart
User
Staff

3. ERD diagram

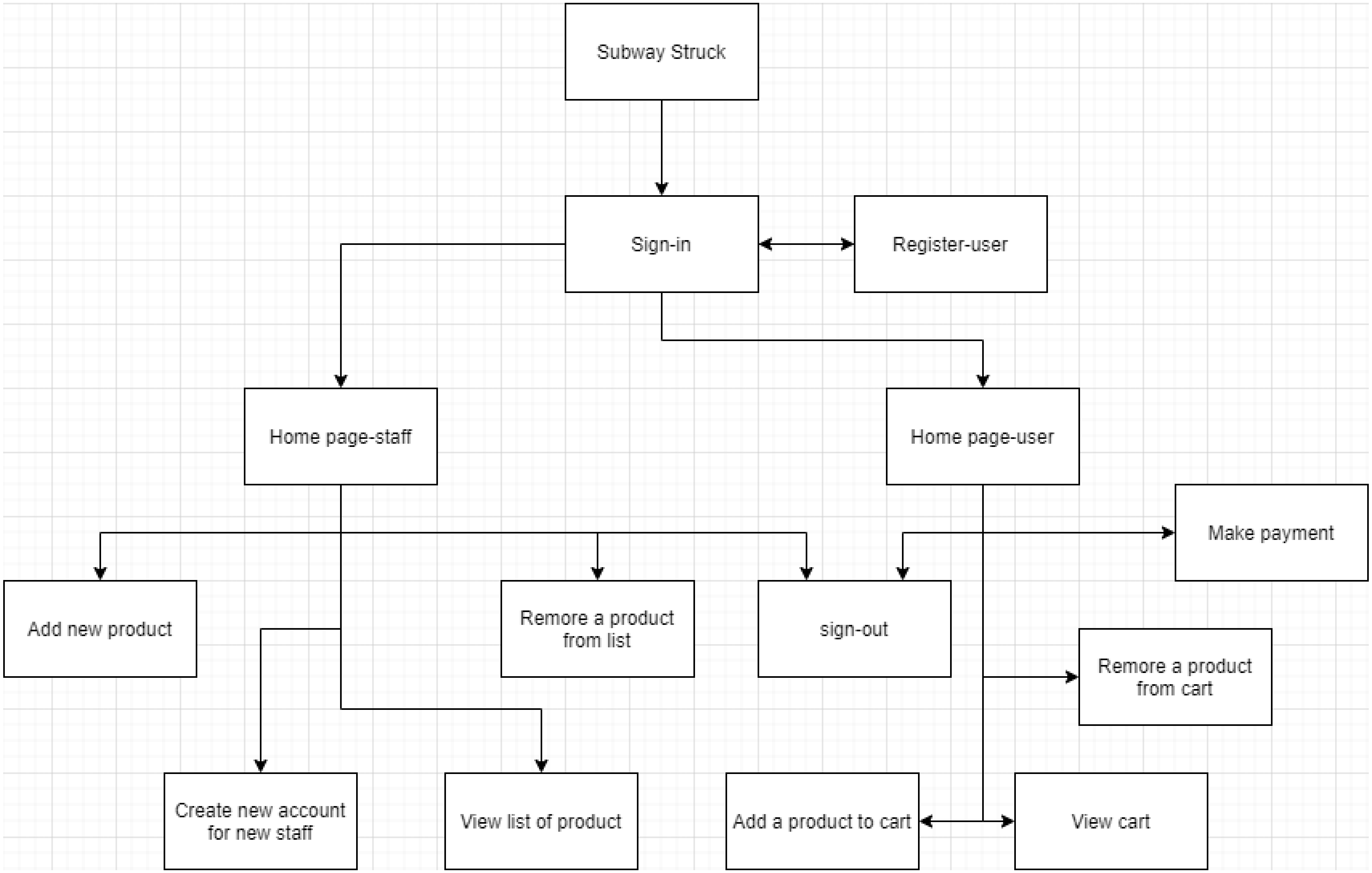


user_account	
PK	id int
	name varchar(100)
	telephone varchar(10)
	email varchar(50)
	password varchar(16)
	role varchar(5)
	address varchar(255)

food	
PK	id int
	name varchar(100)
	category varchar(50)
	image varcher(255)
	price decimal(10,2)

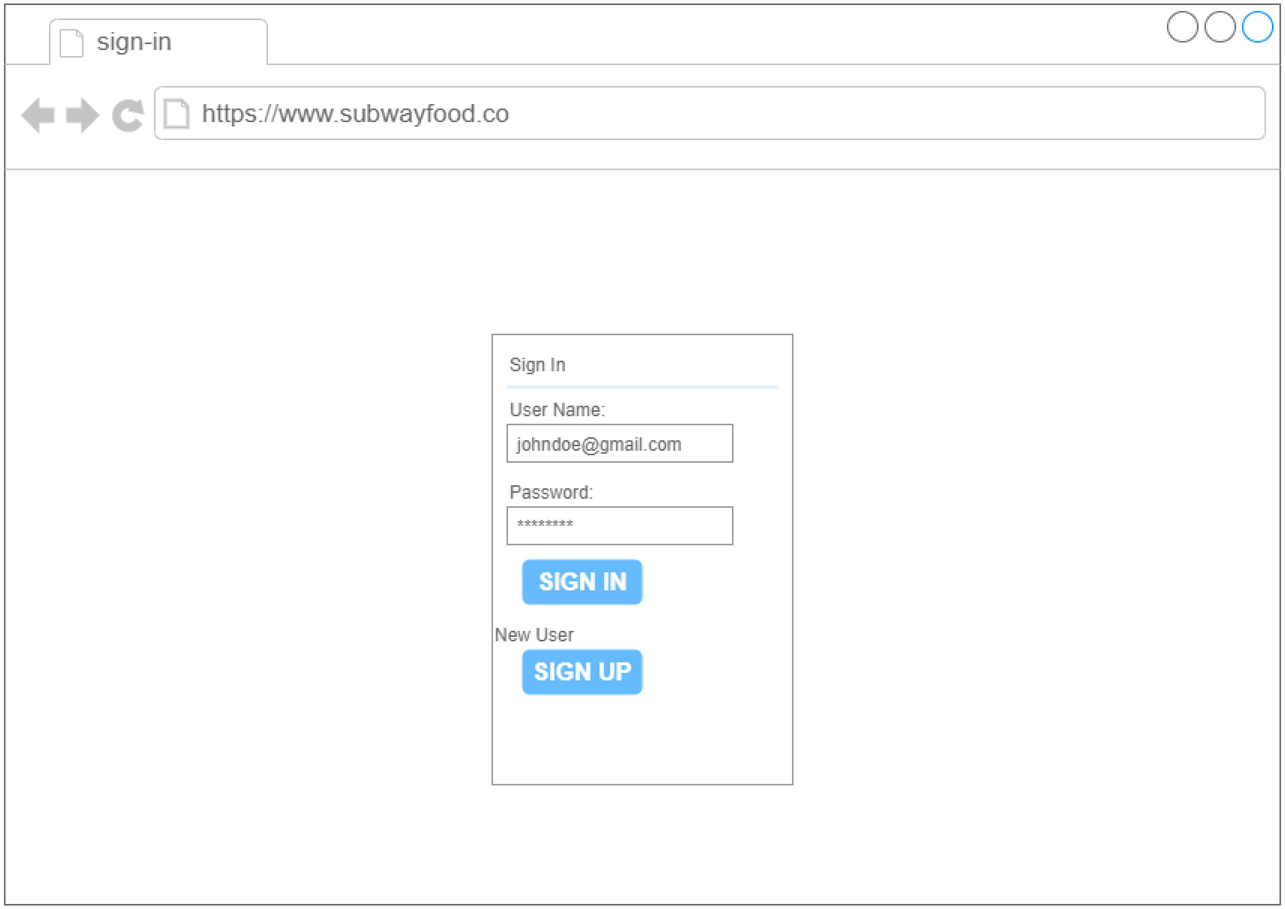
4. Sitemap

The author's schema's sitemap describes the interactions between the interfaces. Users who wish to use the service must first sign in with their account. They can create an account if they do not already have one. After logging in, the system will determine whether you are a customer or an employee. If they are customers, they will be directed to the homepage, where they can order the dishes served by the restaurant, remove items from the cart, or pay immediately. If they are employees, they will be directed to the employee home page, where they will have full administrative privileges. They have the ability to add, edit, or delete products, as well as create new accounts for employees. Check out the menu at the restaurant.


Subway Struck
Sign-in
Register-user
Home page-staff
Home page-user
Make payment
Add new product
Remore a product from list
sign-out
Remore a product from cart
Create new account for new staff
View list of product
Add a product to cart
View cart

IV. Website prototype

1. Sign-in



sign-in

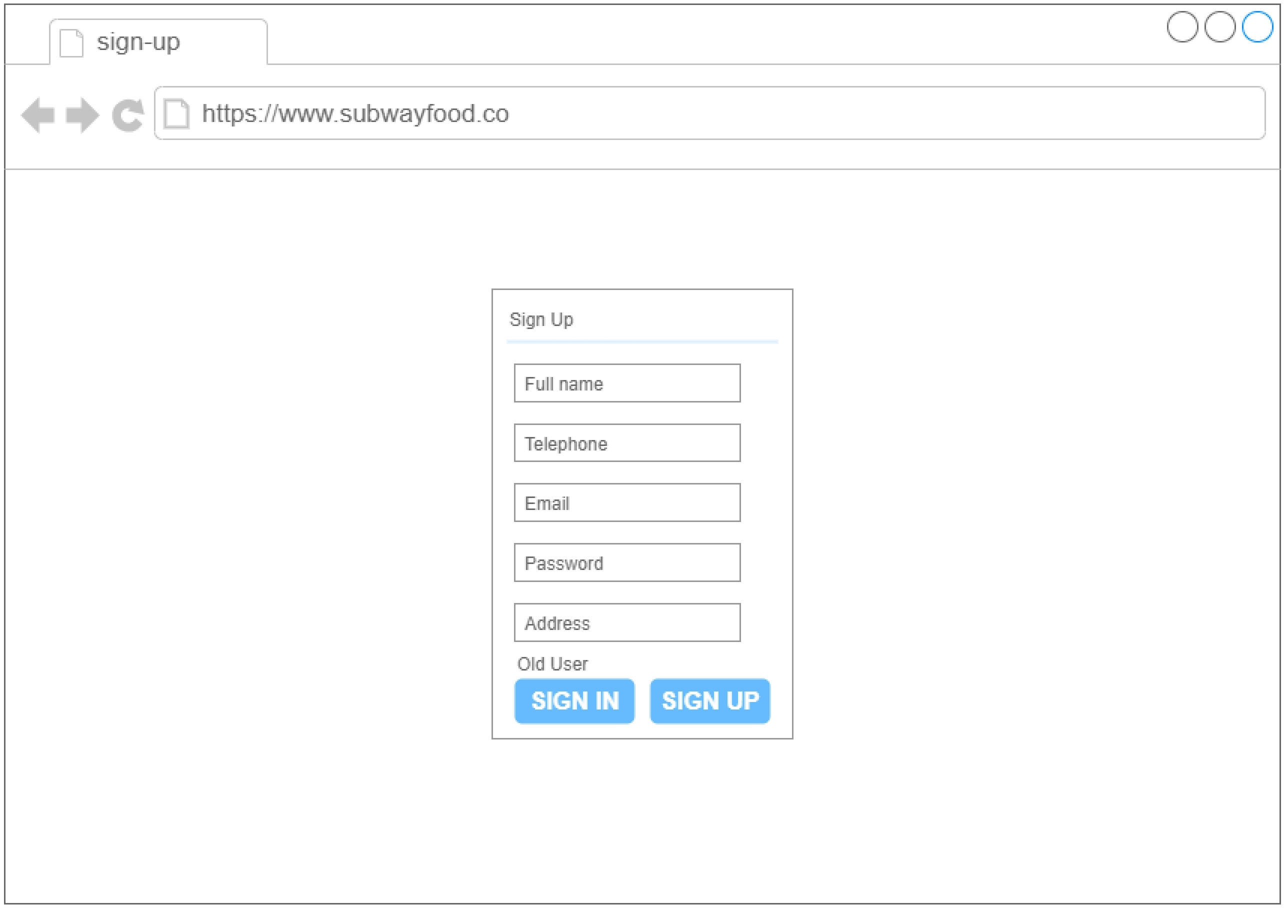
https://www.subwayfood.co

Sign In

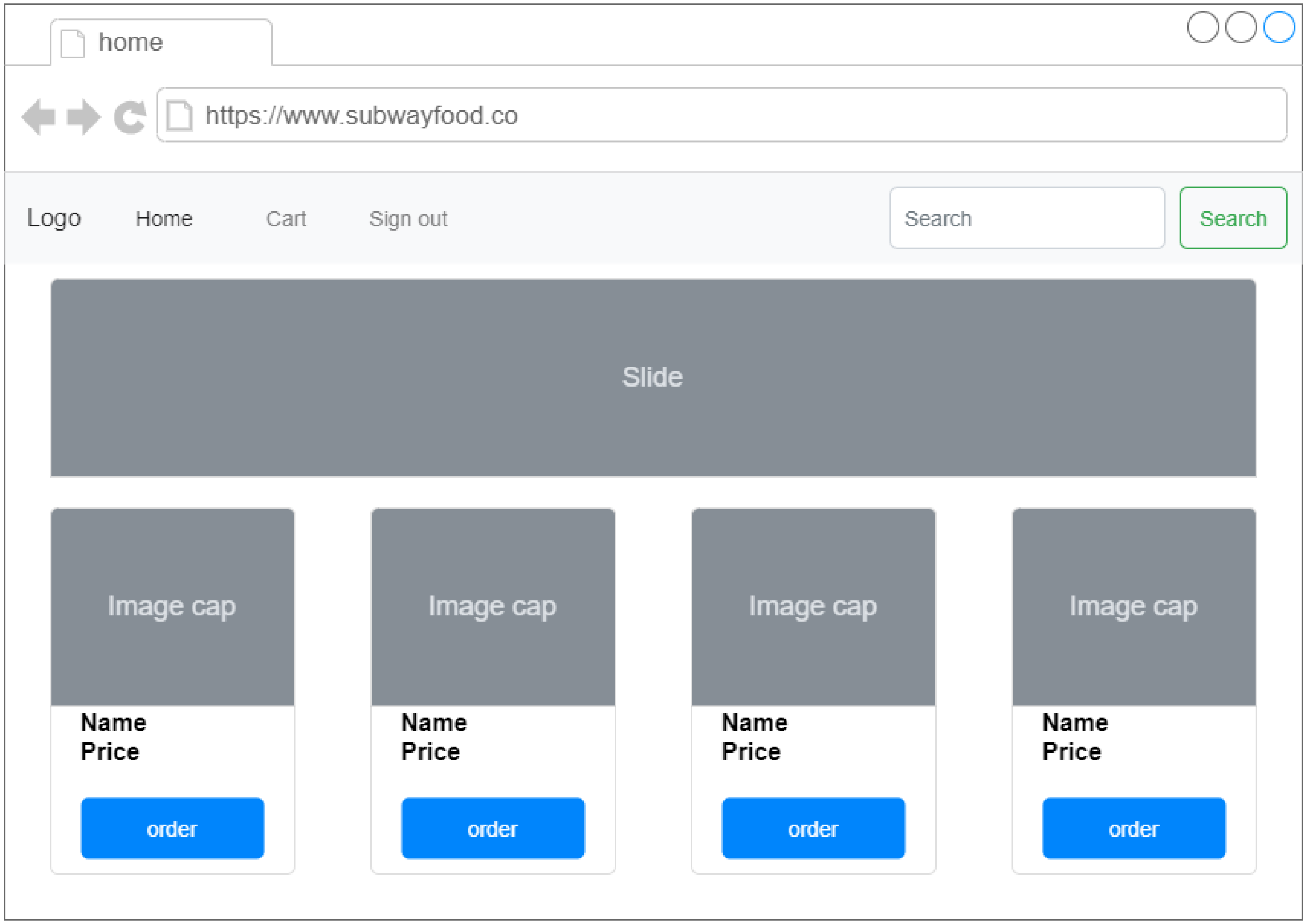
User Name: johndoe@gmail.com

Password: ********

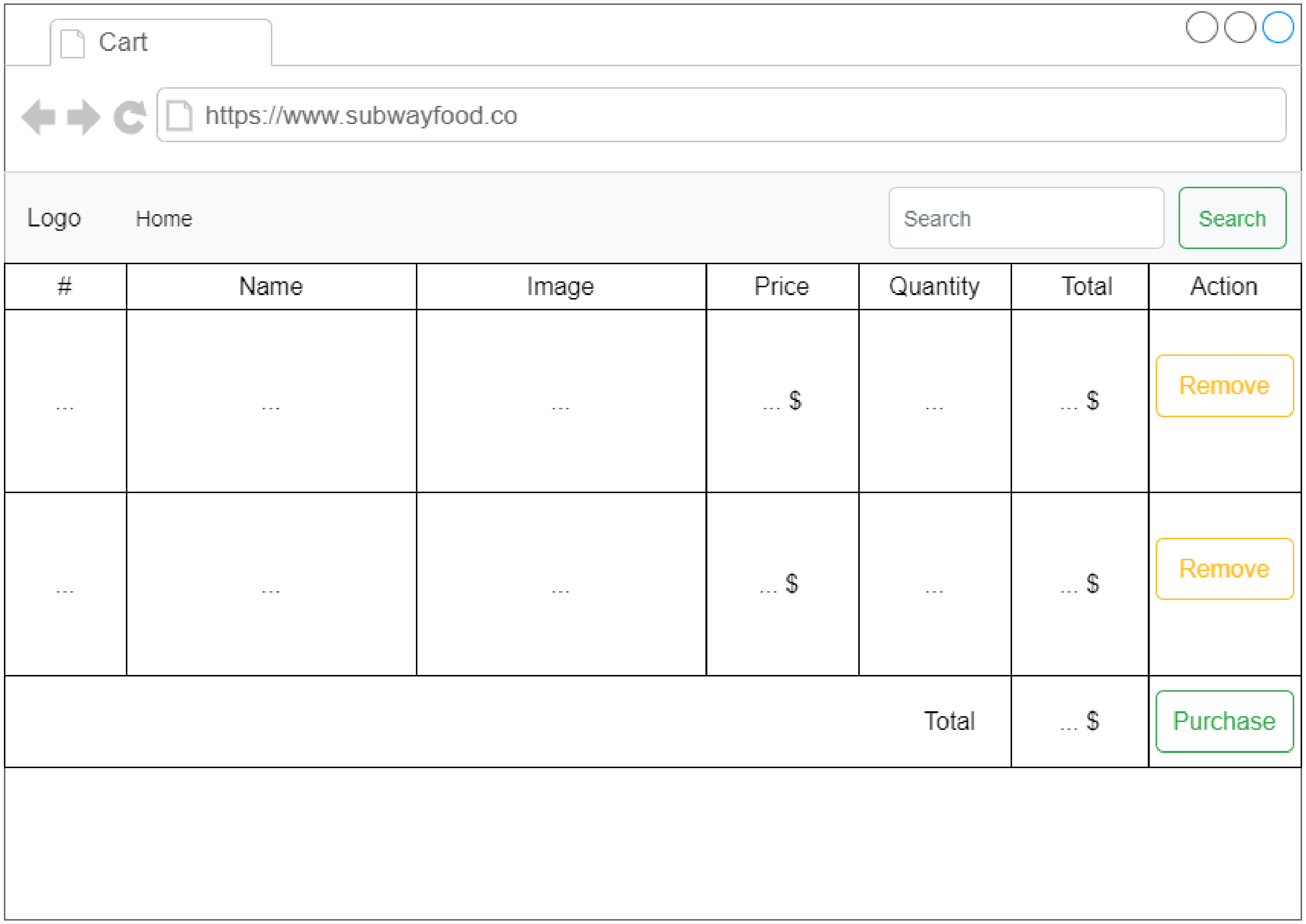
2. Sign-up


sign-up
https://www.subwayfood.co
Sign Up
Full name
Telephone
Email
Password
Address
Old User
SIGN IN
SIGN UP

3. User home

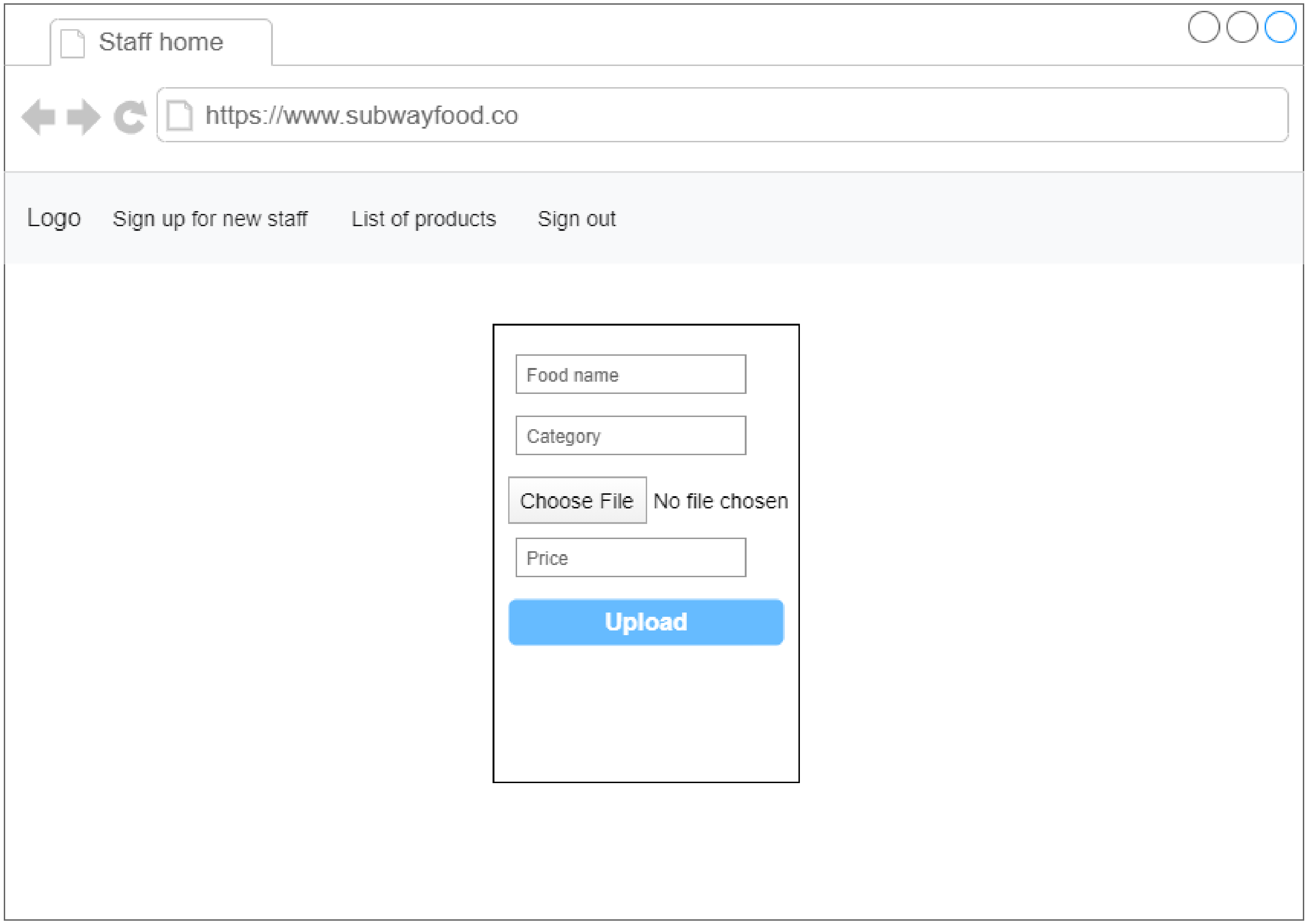

home
https://www.subwayfood.co
Logo
Home
Cart
Sign out
Search
Search
Slide
Image cap
Name
Price
order
Image cap
Name
Price
order
Image cap
Name
Price
order
Image cap
Name
Price
order

4. Shopping cart

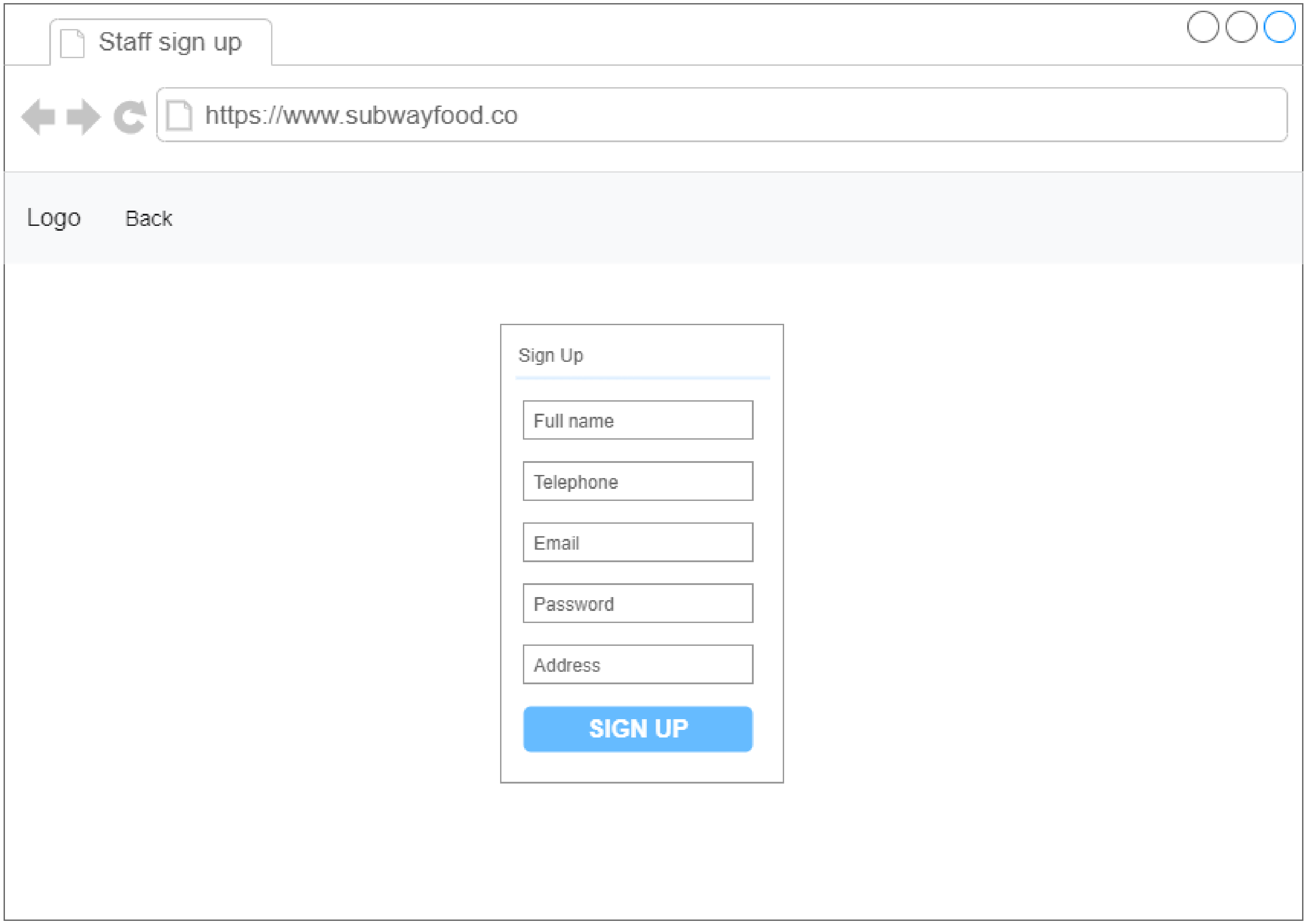


#	Name	Image	Price	Quantity	Total	Action
…	…	…	… $	…	… $	Remove
…	…	…	… $	…	… $	Remove
				Total	… $	Purchase

5. Staff home


Staff home
https://www.subwayfood.co
Logo
Sign up for new staff
List of products
Sign out
Food name
Category
Choose File
No file chosen
Price
Upload

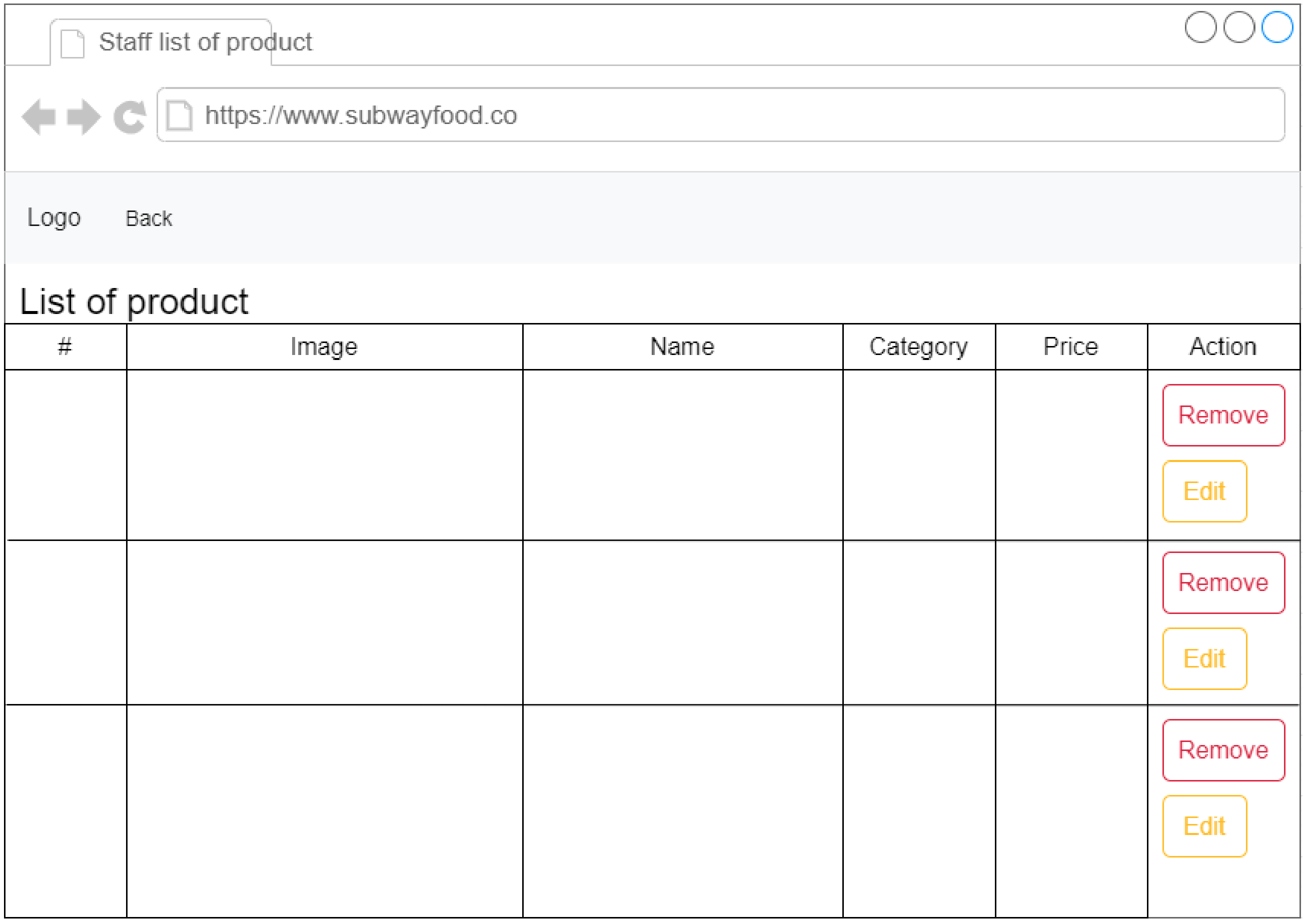
6. Staff sign-up



Staff sign up

https://www.subwayfood.co

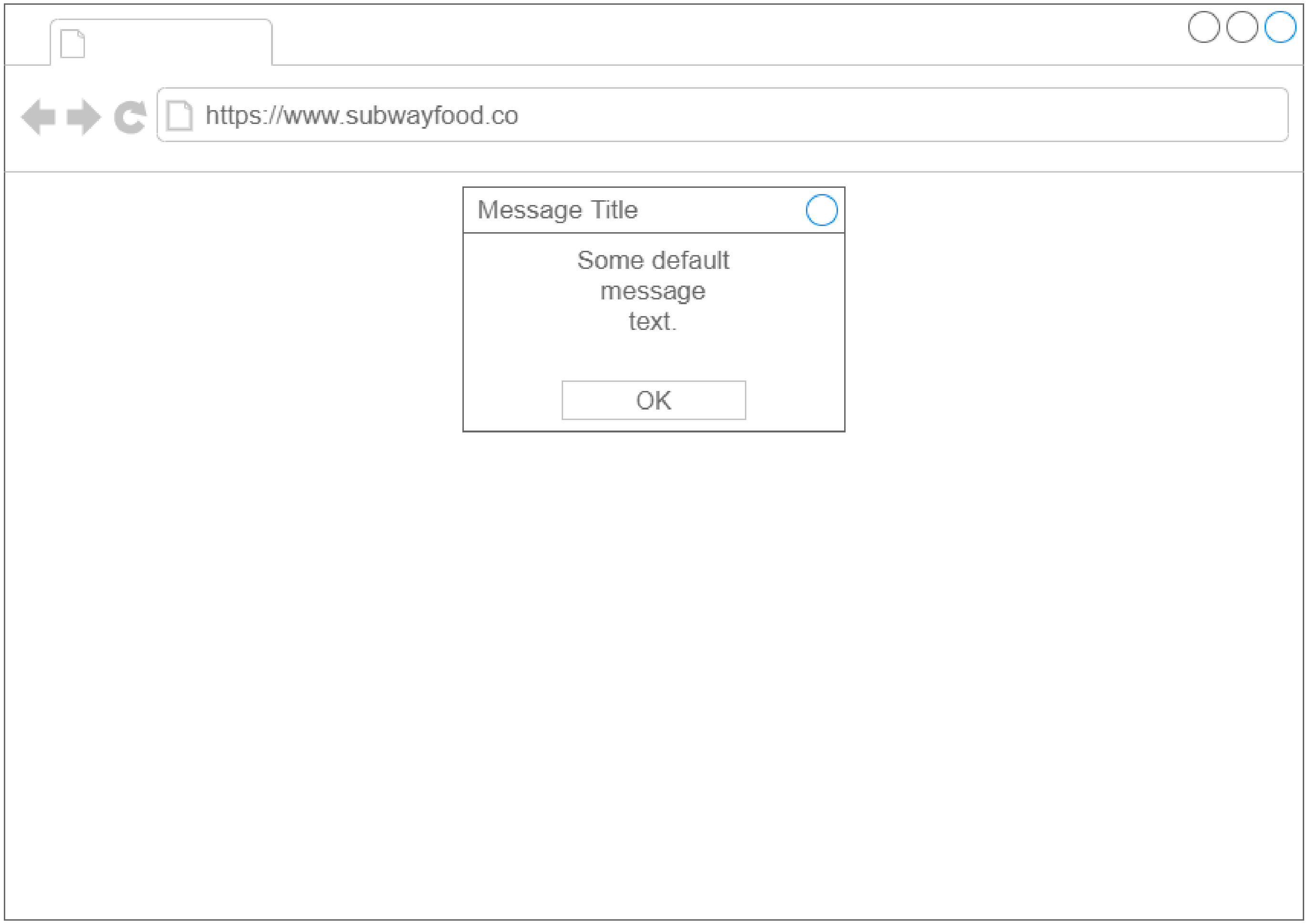
7. List of products



List of product

#	Image	Name	Category	Price	Action
					Remove Edit
					Remove Edit
					Remove Edit

8. Alert



https://www.subwayfood.co

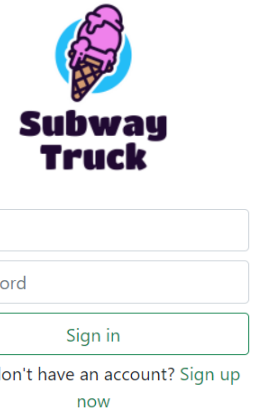
Message Title

Some default message text.

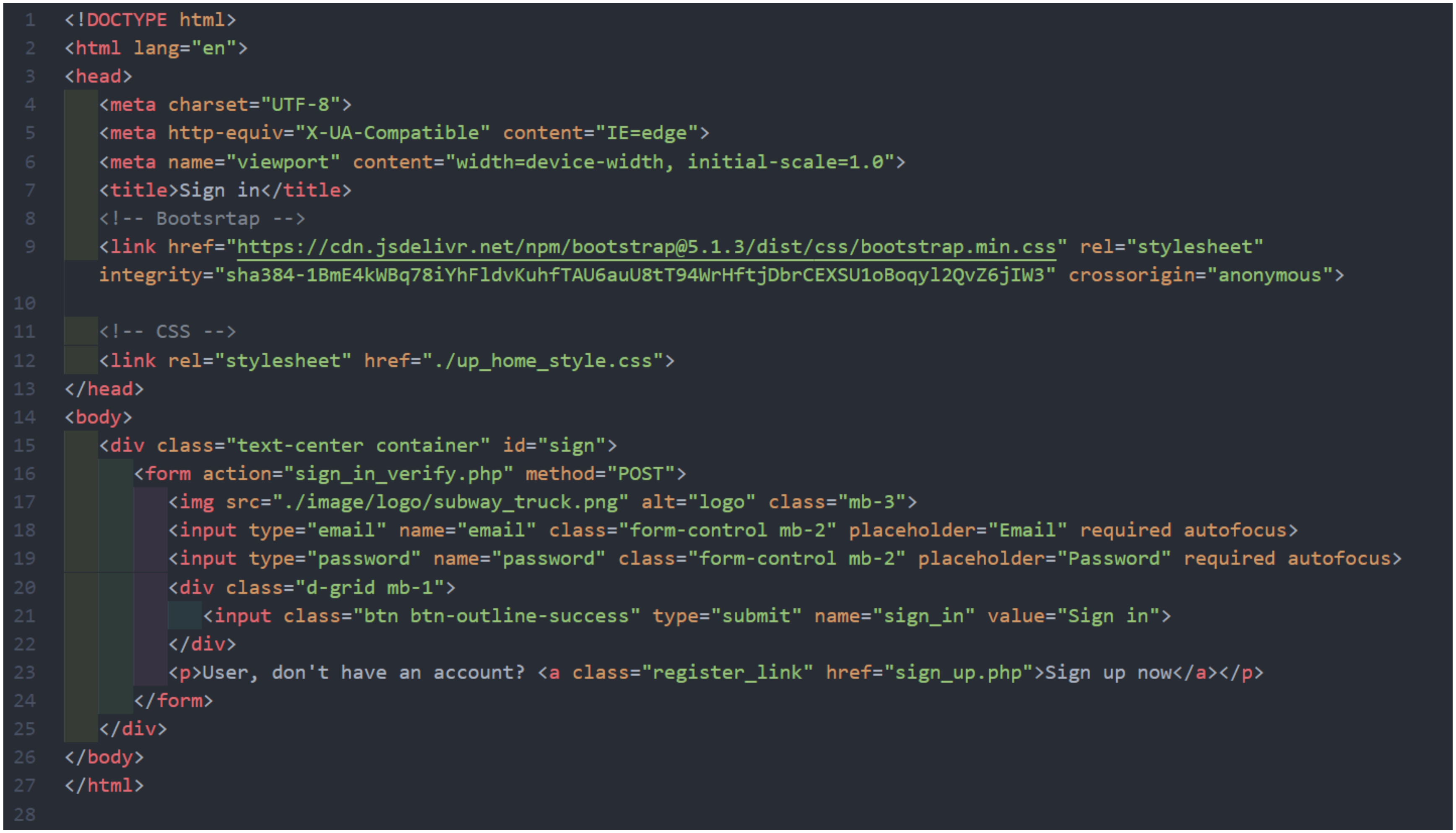
V. Screenshort

In this section, the author will provide actual screenshorts of the subway truck website, along with source code and detailed explanations. The entries will be divided into two sections, user and staff.

1. Sign in

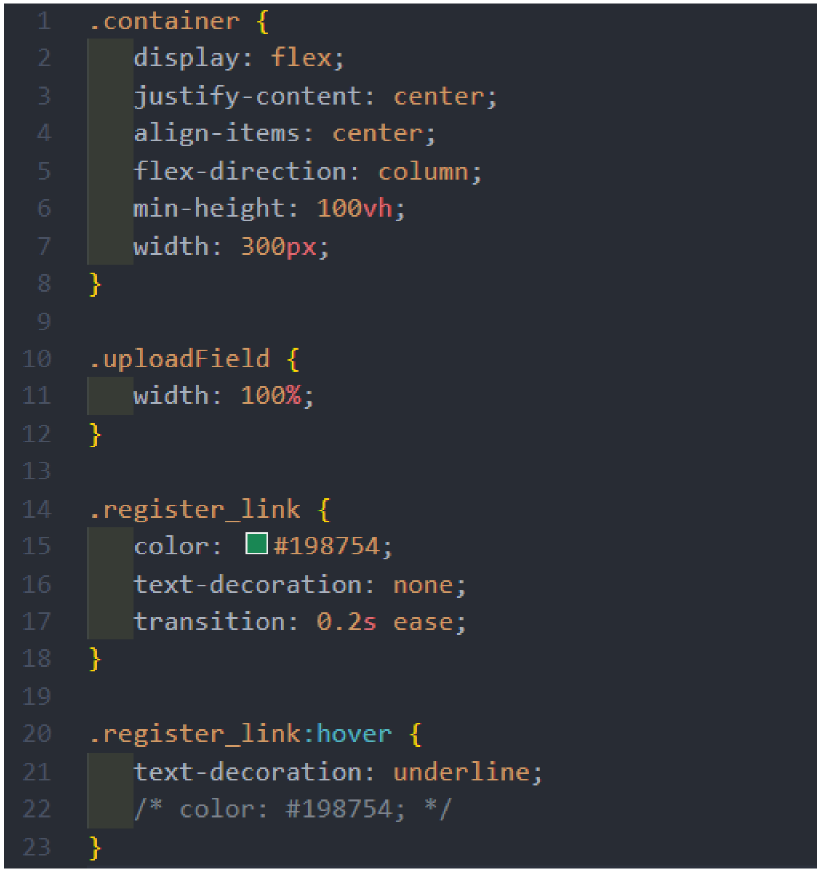


Anyone who visits the site will be taken to a login page. Here, they are required to use a personal account to log in, in case they do not have an account, they can press the register button below.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign in</title>
    <!-- Bootsrtap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <!-- CSS -->
    <link rel="stylesheet" href="./up_home_style.css">
</head>
<body>
    <div class="text-center container" id="sign">
        <form action="sign_in_verify.php" method="POST">
            <img src="./image/logo/subway_truck.png" alt="logo" class="mb-3">
            <input type="email" name="email" class="form-control mb-2" placeholder="Email" required autofocus>
            <input type="password" name="password" class="form-control mb-2" placeholder="Password" required autofocus>
            <div class="d-grid mb-1">
                <input class="btn btn-outline-success" type="submit" name="sign_in" value="Sign in">
            </div>
            <p>User, don't have an account? <a class="register_link" href="sign_up.php">Sign up now</a></p>
        </form>
    </div>
</body>
</html>

```

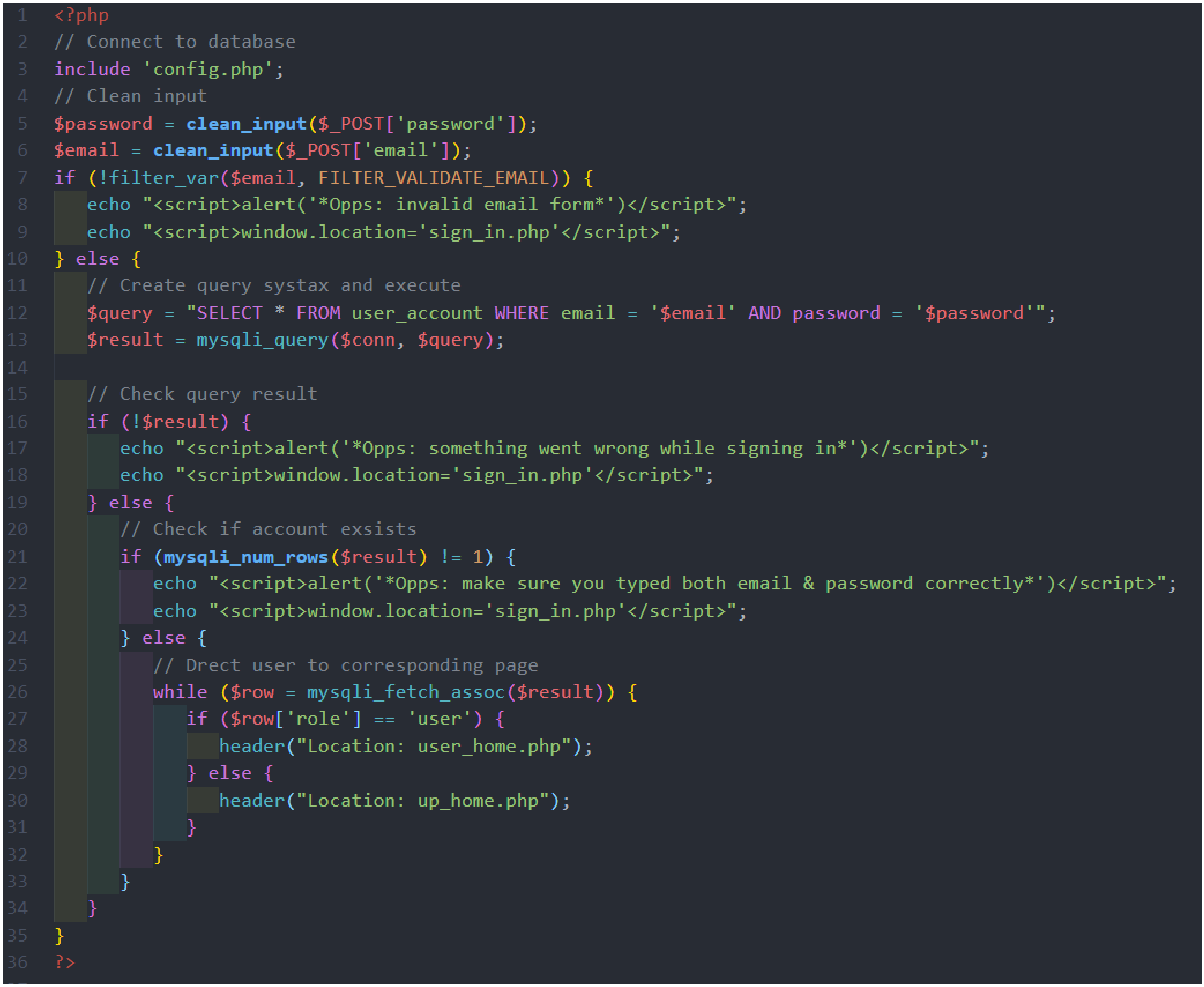
This is the source code used to display the login interface.



```
.container {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    min-height: 100vh;
    width: 300px;
}

.uploadField {
    width: 100%;
}

.register_link {
    color: #198754;
    text-decoration: none;
    transition: 0.2s ease;
}

.register_link:hover {
    text-decoration: underline;
    /* color: #198754; */
}
```

This is some CSS code to align item of the sign-in page and some other pages.

Just like its purpose, those part of code is only used for display, the whole process of handling the situation when the user enters the account and presses the login button will be executed in another file containing the following PHP code.



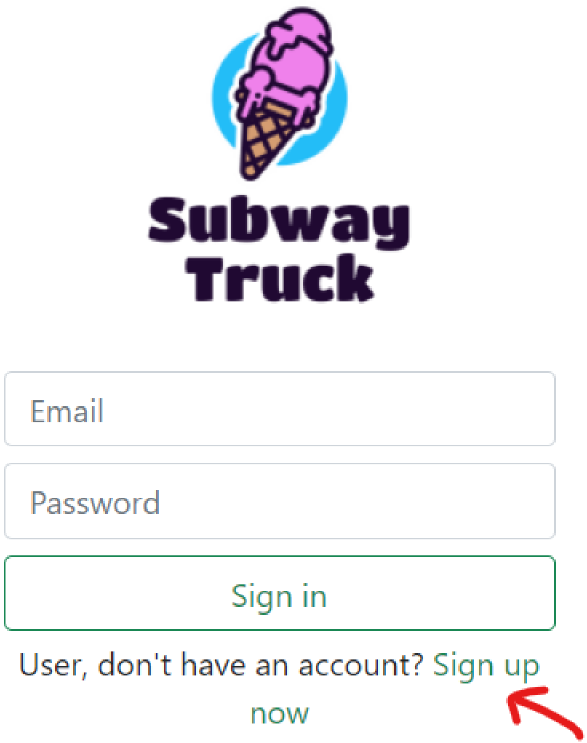
```php
<?php
// Connect to database
include 'config.php';
// Clean input
$password = clean_input($_POST['password']);
$email = clean_input($_POST['email']);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo "<script>alert('*Opps: invalid email form*')</script>";
    echo "<script>window.location='sign_in.php'</script>";
} else {
    // Create query systax and execute
    $query = "SELECT * FROM user_account WHERE email = '$email' AND password = '$password'";
    $result = mysqli_query($conn, $query);

    // Check query result
    if (!$result) {
        echo "<script>alert('*Opps: something went wrong while signing in*')</script>";
        echo "<script>window.location='sign_in.php'</script>";
    } else {
        // Check if account exsists
        if (mysqli_num_rows($result) != 1) {
            echo "<script>alert('*Opps: make sure you typed both email & password correctly*')</script>";
            echo "<script>window.location='sign_in.php'</script>";
        } else {
            // Drect user to corresponding page
            while ($row = mysqli_fetch_assoc($result)) {
                if ($row['role'] == 'user') {
                    header("Location: user_home.php");
                } else {
                    header("Location: up_home.php");
                }
            }
        }
    }
}
?>
```

When the user presses the login button, the system will connect to the database to start checking if the account exists or not. First, the system will check if the email the user entered is really an email, if not, it will notify, then the system will perform a query to the database to find the account corresponding to the user just entered. The system will use the function **mysqli_num_rows(**) to count the number of results returned, if there is a result returned, then the account is valid. the user will be taken to the home page of the user or employee according to the account permissions. If no results are returned, it means that the account does not exist, the system will notify the user.

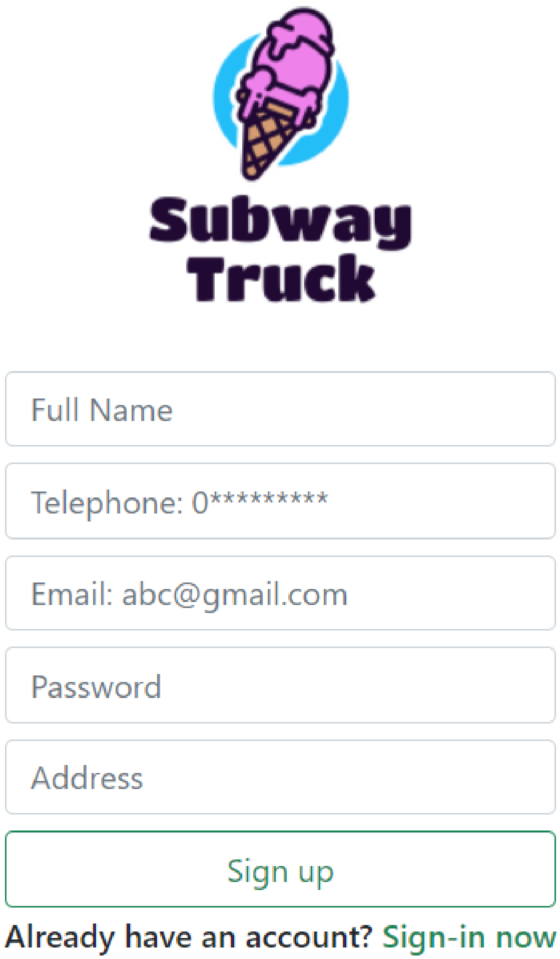
1. User page

1.1. Sign up

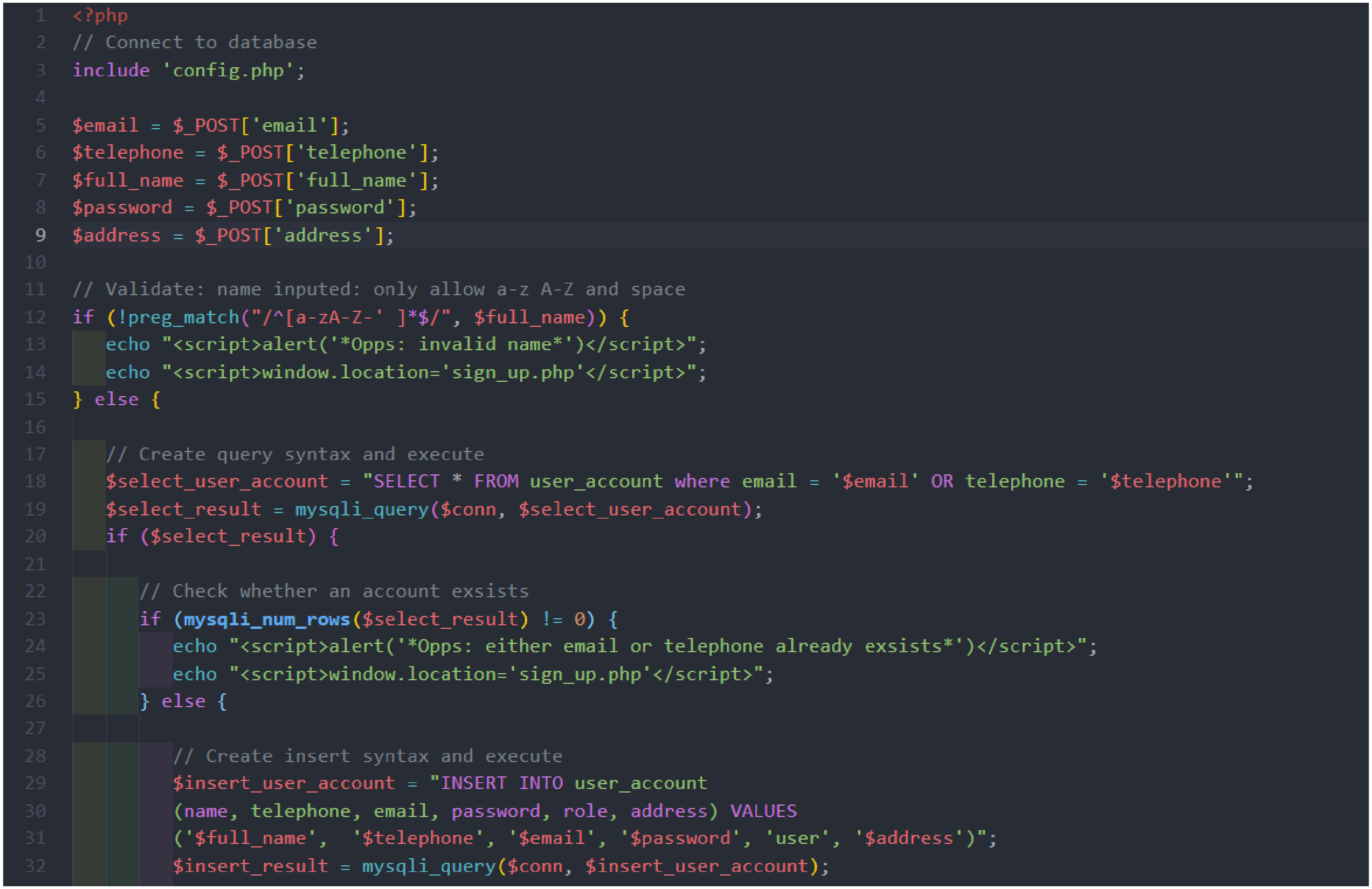
In the condition, a normal user does not have an account they can sign up by clicking the hypertext link below the login button.



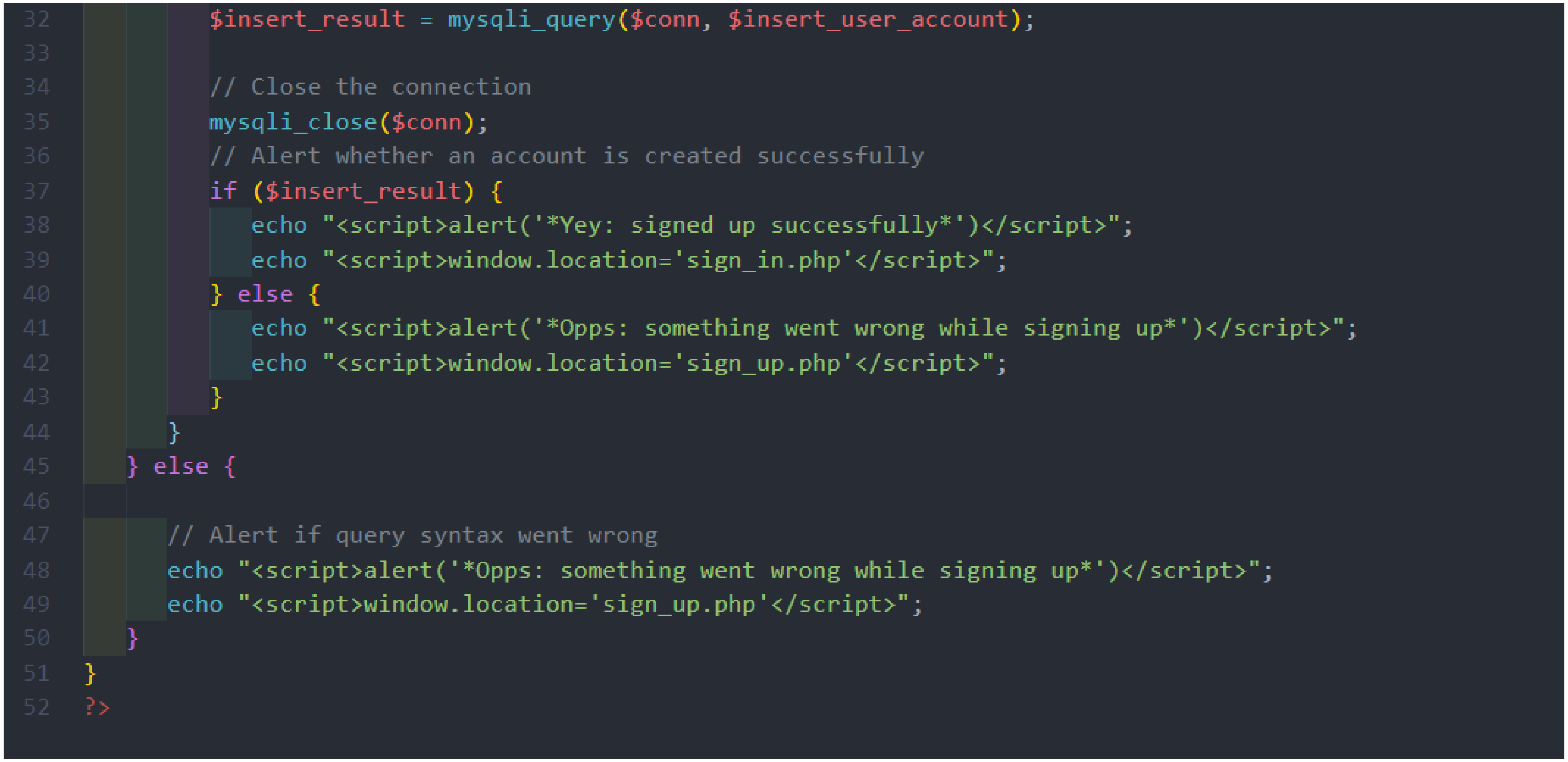
User, don't have an account? Sign up now



Here to register for an account, the user is required to enter all information such as email, phone number, email, password and address. After filling in that information, you can press the register button, the system will check if that account already exists, information such as email and phone number can only be used once for an account. If valid, the account will be created.



```php
<?php
// Connect to database
include 'config.php';

$email = $_POST['email'];
$telephone = $_POST['telephone'];
$full_name = $_POST['full_name'];
$password = $_POST['password'];
$address = $_POST['address'];

// Validate: name inputed: only allow a-z A-Z and space
if (!preg_match("/^[a-zA-Z-' ]*$/", $full_name)) {
    echo "<script>alert('*Opps: invalid name*')</script>";
    echo "<script>window.location='sign_up.php'</script>";
} else {

    // Create query syntax and execute
    $select_user_account = "SELECT * FROM user_account where email = '$email' OR telephone = '$telephone'";
    $select_result = mysqli_query($conn, $select_user_account);
    if ($select_result) {

        // Check whether an account exsists
        if (mysqli_num_rows($select_result) != 0) {
            echo "<script>alert('*Opps: either email or telephone already exsists*')</script>";
            echo "<script>window.location='sign_up.php'</script>";
        } else {

            // Create insert syntax and execute
            $insert_user_account = "INSERT INTO user_account
            (name, telephone, email, password, role, address) VALUES
            ('$full_name',  '$telephone', '$email', '$password', 'user', '$address')";
            $insert_result = mysqli_query($conn, $insert_user_account);
```



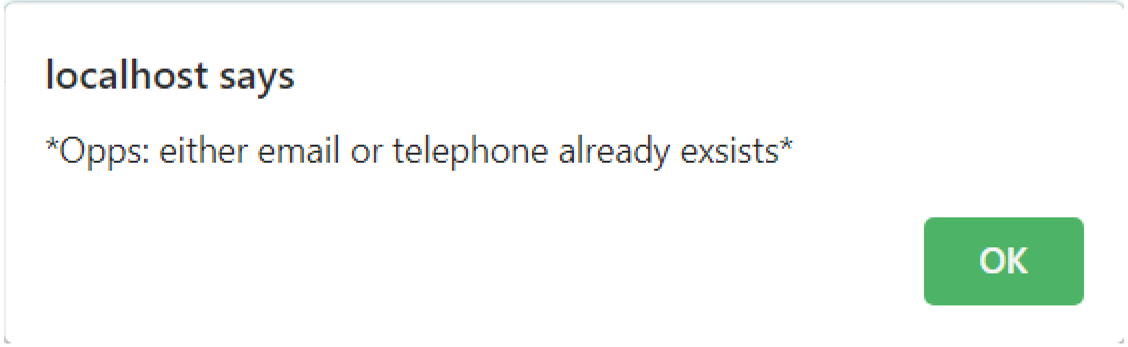
```php
            $insert_result = mysqli_query($conn, $insert_user_account);

            // Close the connection
            mysqli_close($conn);
            // Alert whether an account is created successfully
            if ($insert_result) {
                echo "<script>alert('*Yey: signed up successfully*')</script>";
                echo "<script>window.location='sign_in.php'</script>";
            } else {
                echo "<script>alert('*Opps: something went wrong while signing up*')</script>";
                echo "<script>window.location='sign_up.php'</script>";
            }
        }
    } else {

        // Alert if query syntax went wrong
        echo "<script>alert('*Opps: something went wrong while signing up*')</script>";
        echo "<script>window.location='sign_up.php'</script>";
    }
}
?>
```

Here is the source code of the user account registration part. Here when the user has entered all required information and pressed register. The system will check the validity of the data they enter.

- Name: must contain only character and white space
- Email: must be abc@xxx.xxx
- Telephone: must contain 10 numbers and start with 0.
- No input filed can be empty

When the data is valid, the system will check the database to see if the data entered by the user has been used by another account. If such information already exists, the user will be asked to re-enter it, otherwise, an account will be created corresponding to the information provided by the user.

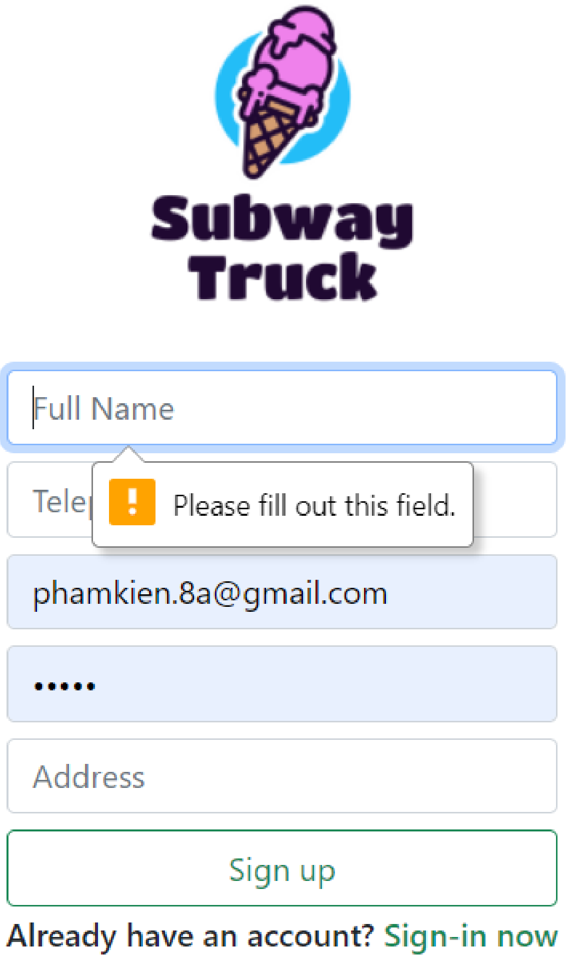
When entering an existing email or phone number, the following message will appear:



localhost says

Opps: either email or telephone already exsists

If a certain input cell is missing data, the following message will be displayed:



Subway Truck

Full Name

Tele

Please fill out this field.

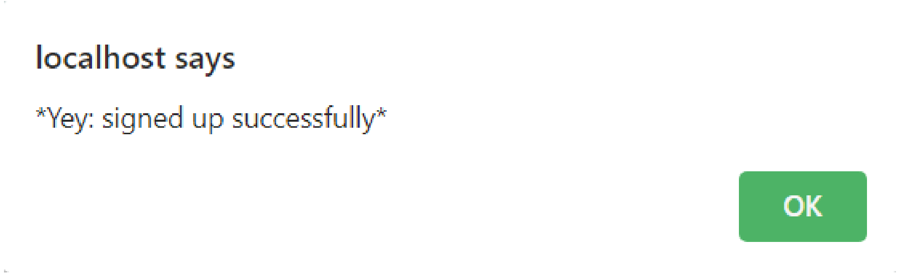
phamkien.8a@gmail.com

•••••

Address

Already have an account? Sign-in now

If the account creation is successful, there will be a message as below:

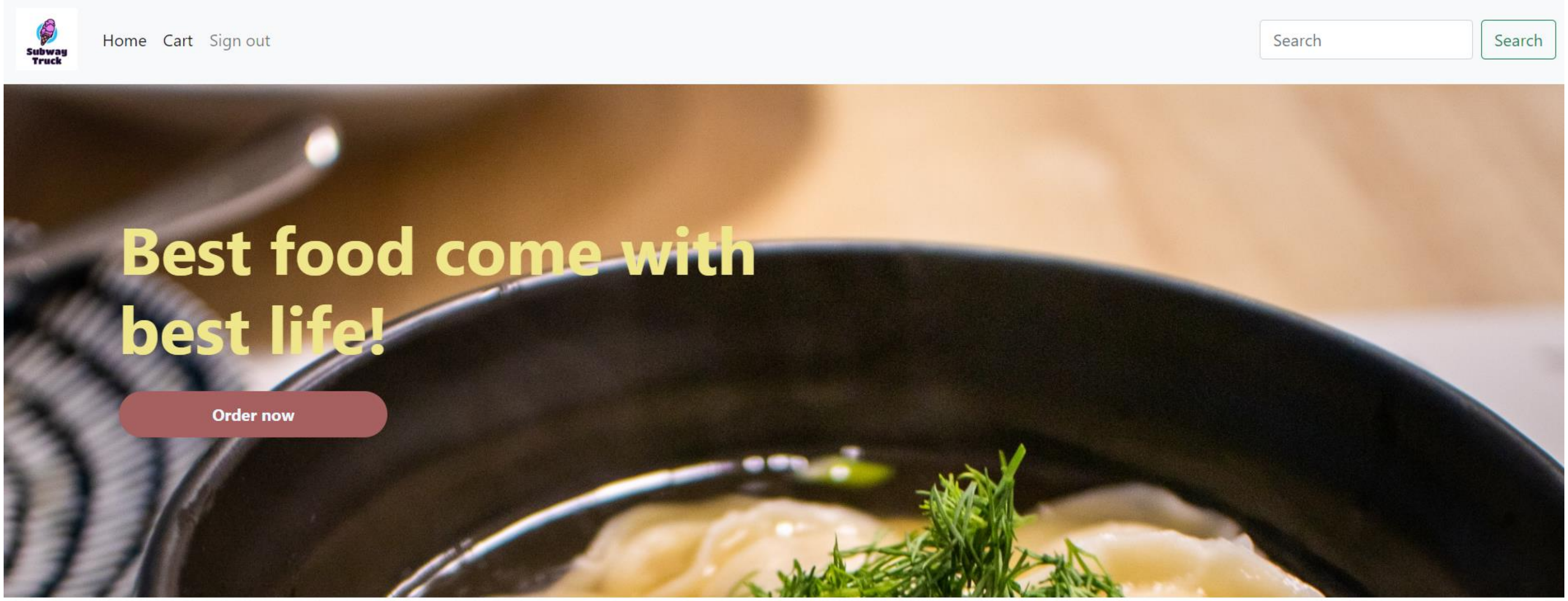


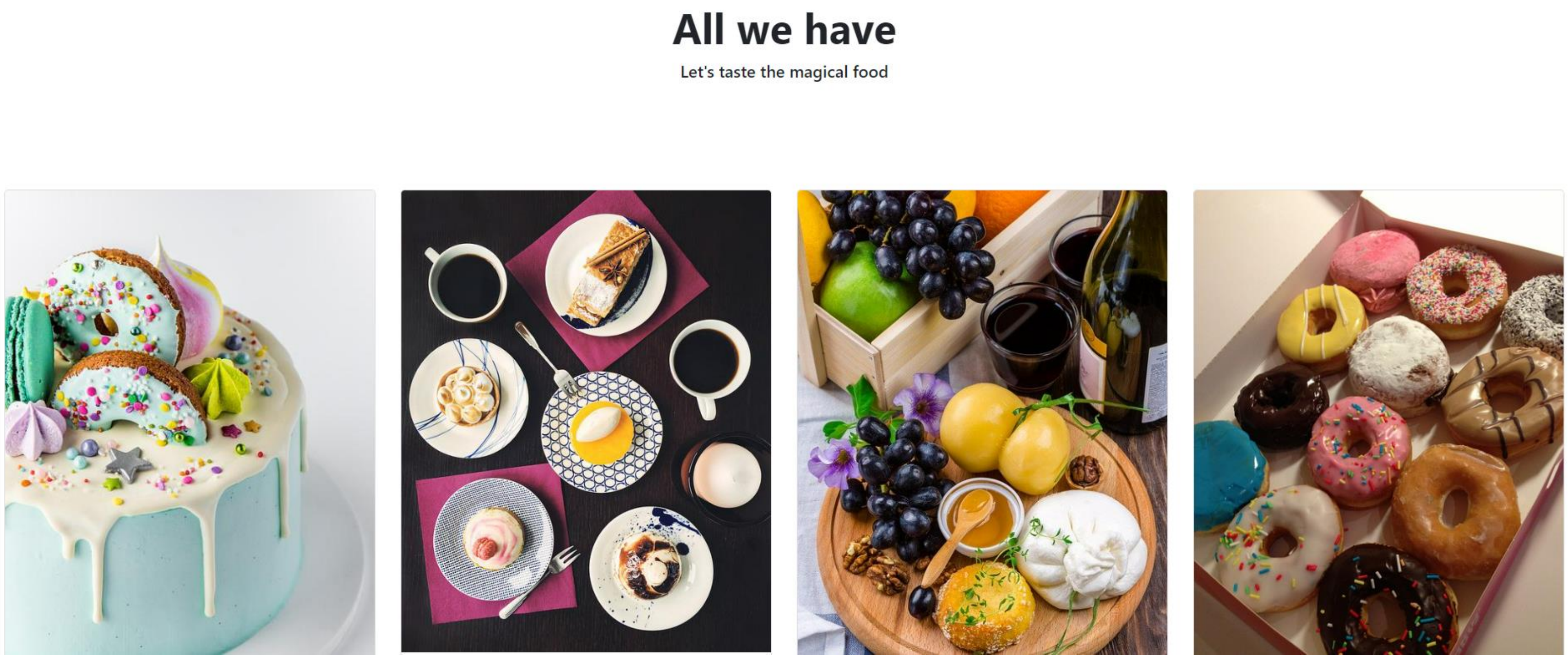
localhost says

Yey: signed up successfully

1.2. User home

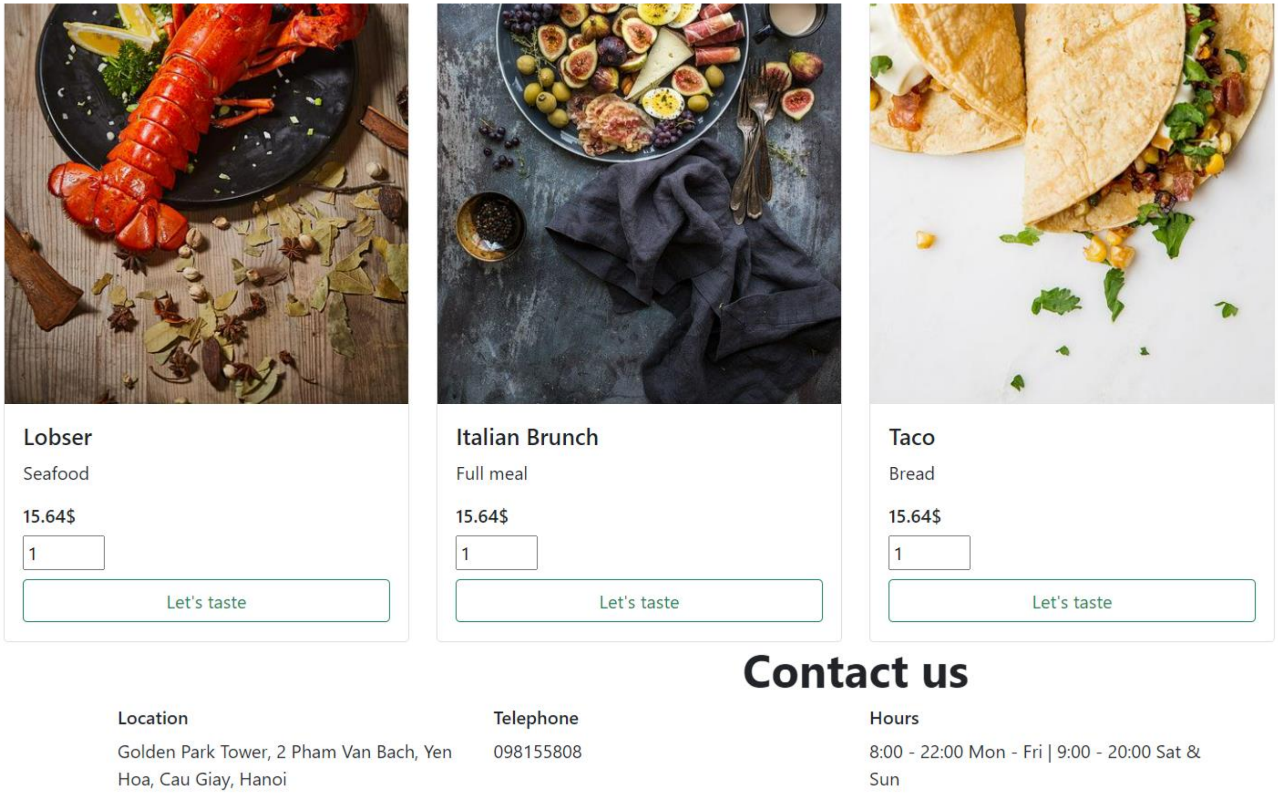
This is the part that will display the main screen displaying subwayfood's products. At this screen there will be a navigation bar at the top of the web page, followed by a slider containing a call-to-action button. The product list will be displayed from start to finish, each product will be displayed as a card. Each card will have a photo with the name and price of the corresponding dish. At the bottom of a website is a footer containing contact information, and the address of the restaurant.


Subway Truck
Home
Cart
Sign out
Search
Search
Best food come with best life!
Order now



All we have

Let's taste the magical food



Lobser

Seafood

15.64$

1

Let's taste

Italian Brunch

Full meal

15.64$

1

Let's taste

Taco

Bread

15.64$

1

Let's taste

Contact us

Location

Golden Park Tower, 2 Pham Van Bach, Yen Hoa, Cau Giay, Hanoi

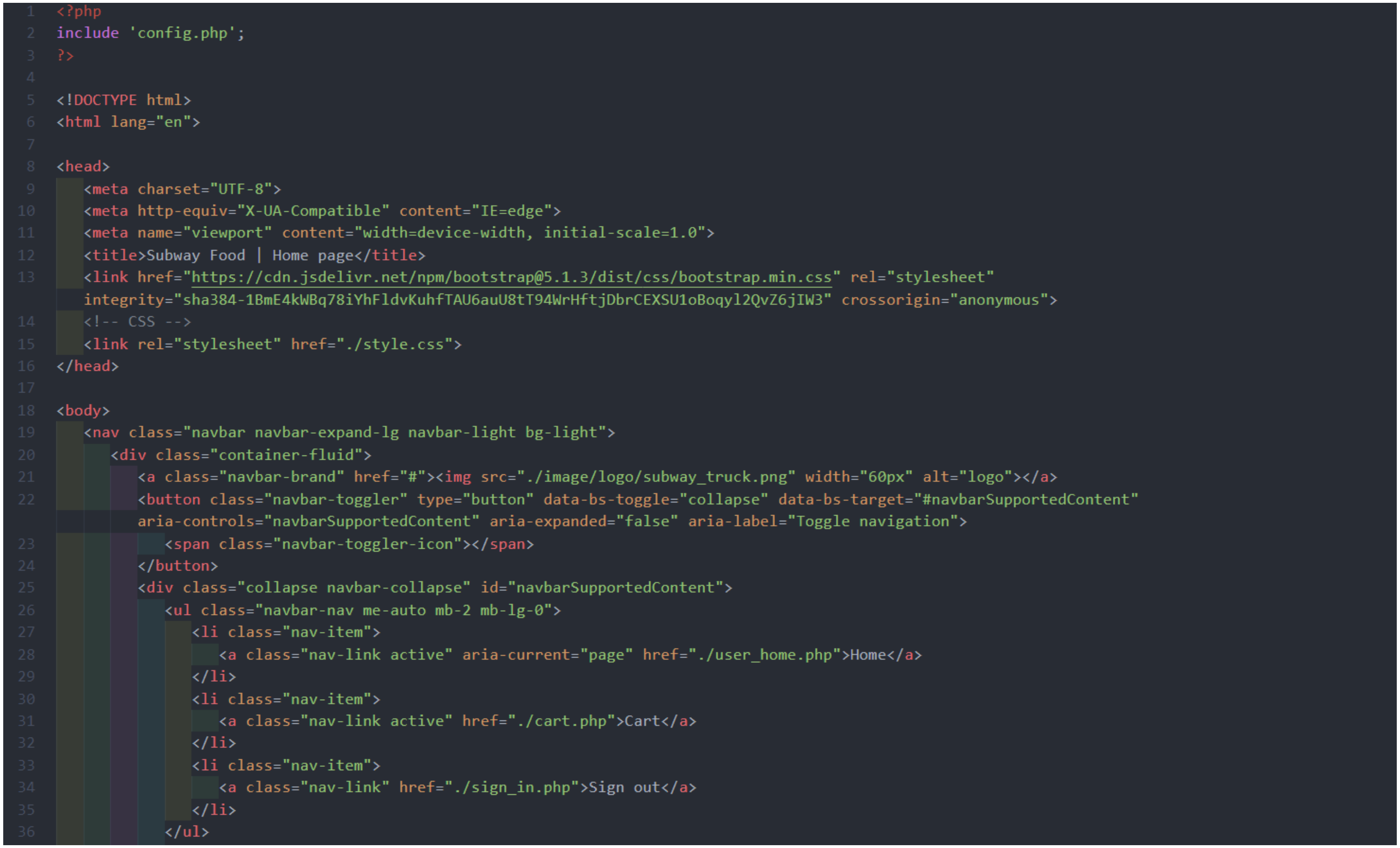
Telephone

098155808

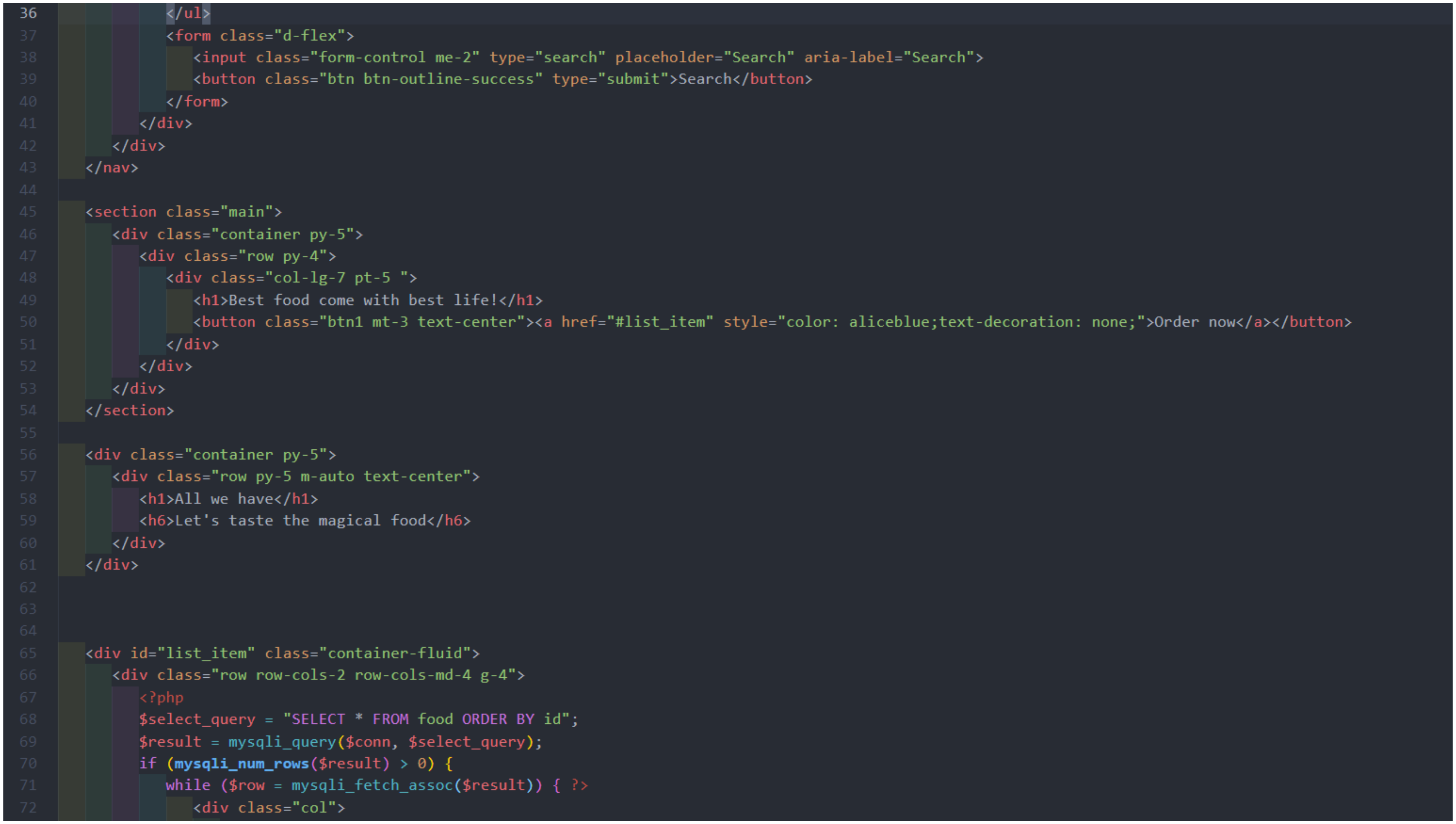
Hours

8:00 - 22:00 Mon - Fri | 9:00 - 20:00 Sat & Sun

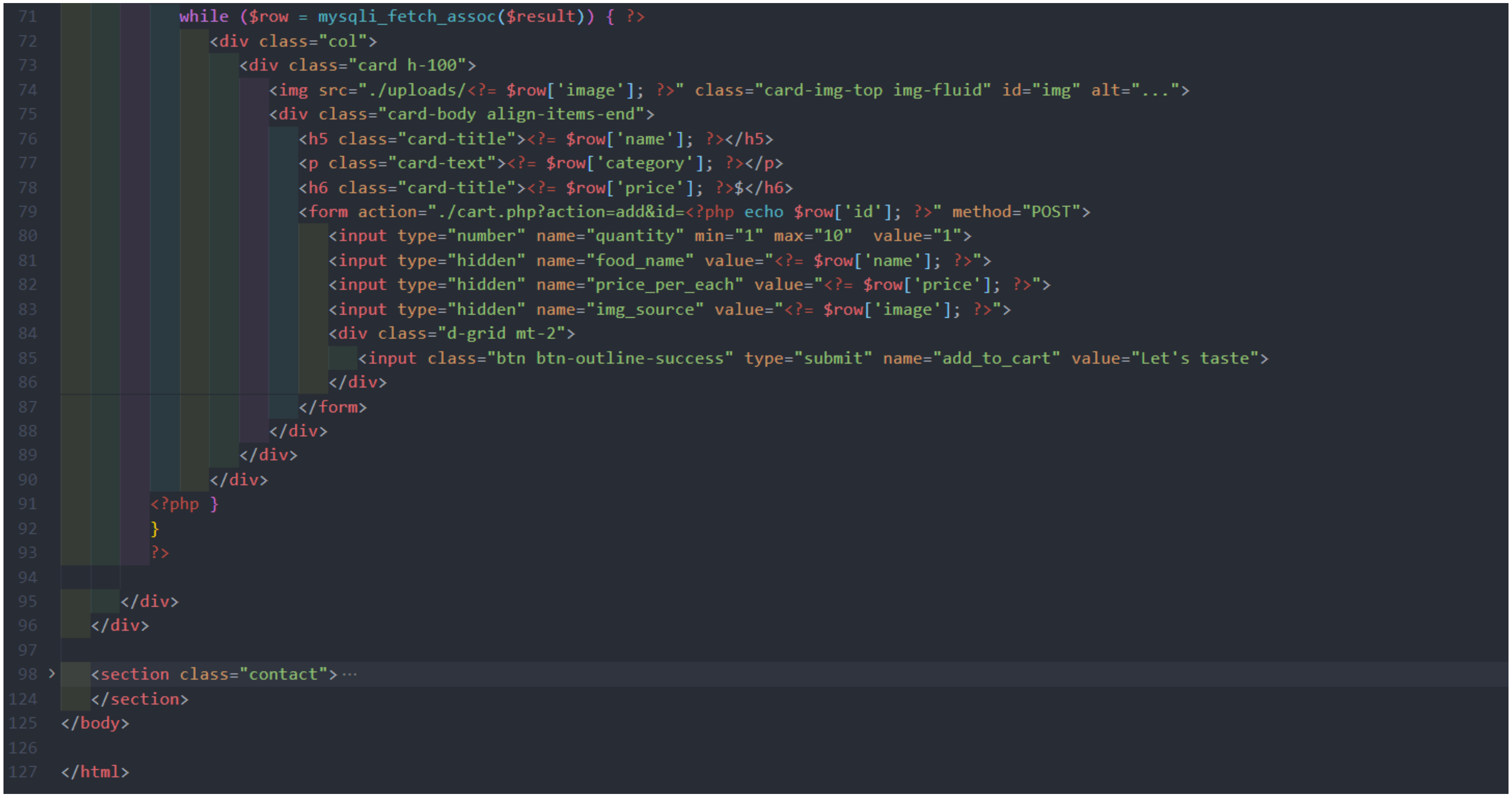
The source code that displays the user home page will be presented in the screenshorts below:



```
<?php
include 'config.php';
?>

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Subway Food | Home page</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
    <!-- CSS -->
    <link rel="stylesheet" href="./style.css">
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fluid">
            <a class="navbar-brand" href="#"><img src="./image/logo/subway_truck.png" width="60px" alt="logo"></a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="./user_home.php">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" href="./cart.php">Cart</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="./sign_in.php">Sign out</a>
                    </li>
                </ul>
```

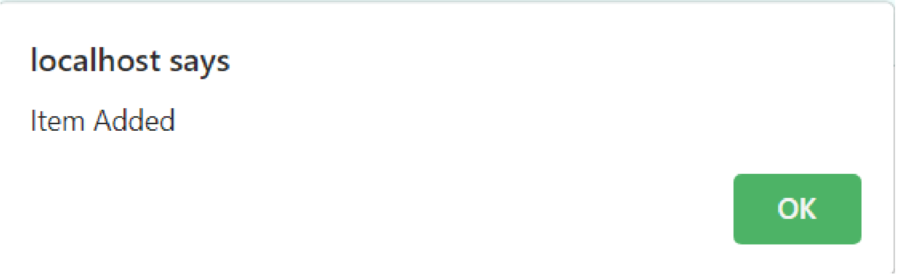


```
            </ul>
            <form class="d-flex">
                <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
                <button class="btn btn-outline-success" type="submit">Search</button>
            </form>
        </div>
    </div>
</nav>

<section class="main">
    <div class="container py-5">
        <div class="row py-4">
            <div class="col-lg-7 pt-5 ">
                <h1>Best food come with best life!</h1>
                <button class="btn1 mt-3 text-center"><a href="#list_item" style="color: aliceblue;text-decoration: none;">Order now</a></button>
            </div>
        </div>
    </div>
</section>

<div class="container py-5">
    <div class="row py-5 m-auto text-center">
        <h1>All we have</h1>
        <h6>Let's taste the magical food</h6>
    </div>
</div>



<div id="list_item" class="container-fluid">
    <div class="row row-cols-2 row-cols-md-4 g-4">
        <?php
        $select_query = "SELECT * FROM food ORDER BY id";
        $result = mysqli_query($conn, $select_query);
        if (mysqli_num_rows($result) > 0) {
            while ($row = mysqli_fetch_assoc($result)) { ?>
                <div class="col">
```



```
                while ($row = mysqli_fetch_assoc($result)) { ?>
                    <div class="col">
                        <div class="card h-100">
                            <img src="./uploads/<?= $row['image']; ?>" class="card-img-top img-fluid" id="img" alt="...">
                            <div class="card-body align-items-end">
                                <h5 class="card-title"><?= $row['name']; ?></h5>
                                <p class="card-text"><?= $row['category']; ?></p>
                                <h6 class="card-title"><?= $row['price']; ?>$</h6>
                                <form action="./cart.php?action=add&id=<?php echo $row['id']; ?>" method="POST">
                                    <input type="number" name="quantity" min="1" max="10"  value="1">
                                    <input type="hidden" name="food_name" value="<?= $row['name']; ?>">
                                    <input type="hidden" name="price_per_each" value="<?= $row['price']; ?>">
                                    <input type="hidden" name="img_source" value="<?= $row['image']; ?>">
                                    <div class="d-grid mt-2">
                                        <input class="btn btn-outline-success" type="submit" name="add_to_cart" value="Let's taste">
                                    </div>
                                </form>
                            </div>
                        </div>
                    </div>
            <?php }
            }
            ?>

        </div>
    </div>

    <section class="contact">···
    </section>
</body>

</html>
```

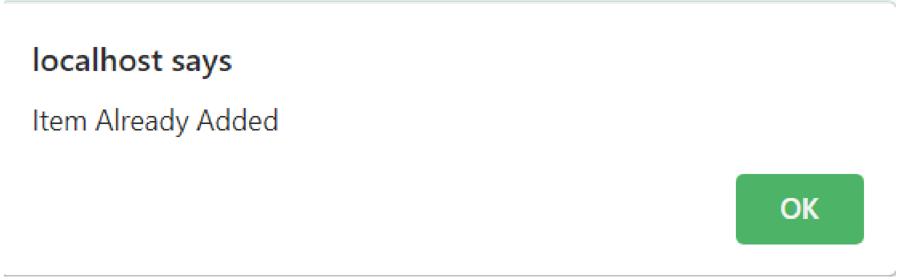
When the user likes a certain dish, they will press the order button below that dish, they can choose to order as much as they like, but the default is 1 and the maximum is 10. After pressing the order, the dish that will be saved in the shopping cart:


localhost says

Item Added

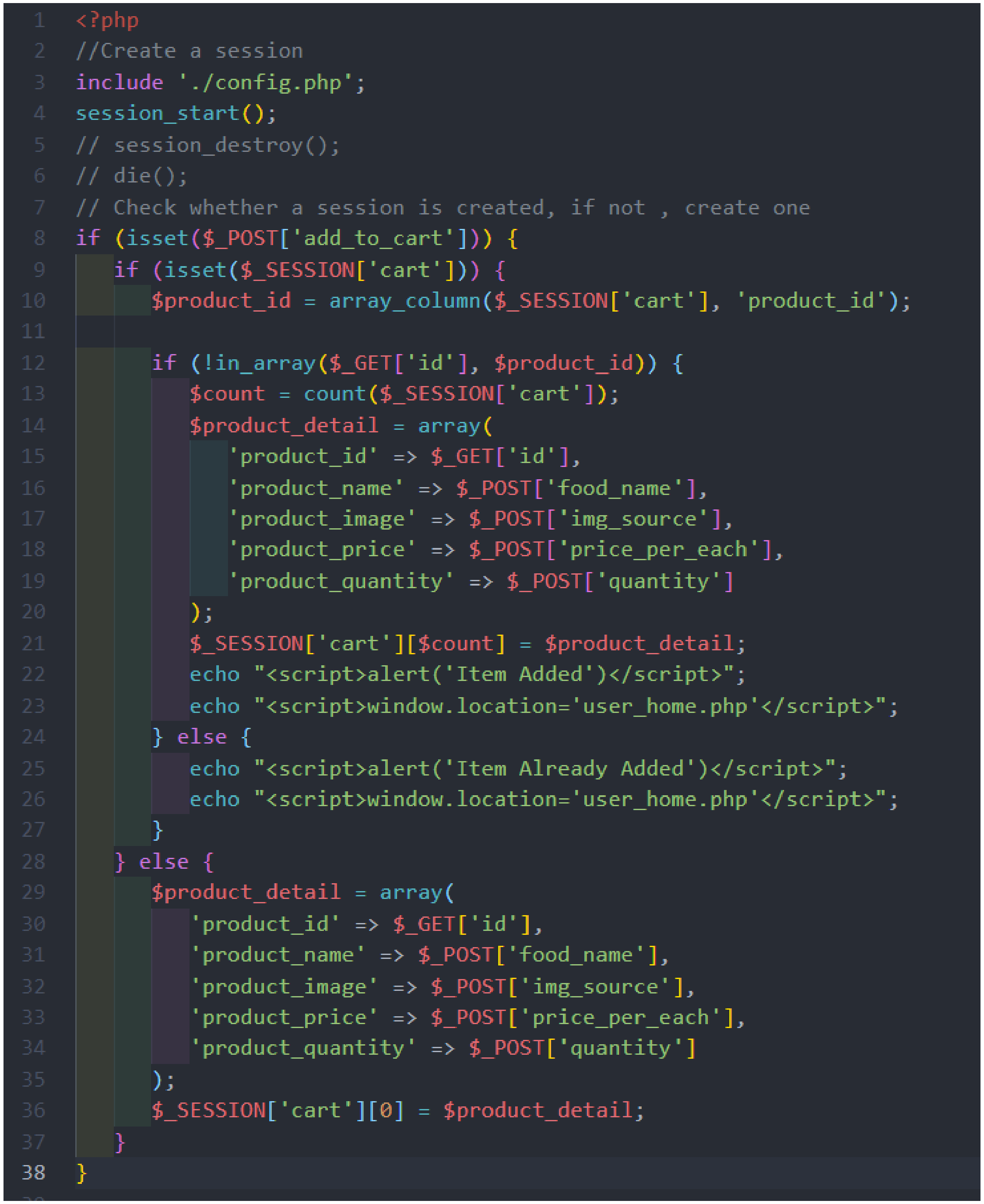
If the user wants to change the ordered item or order more, he or she will have to go to the shopping cart to remove the item and reorder it.

When an item already exists in the cart, the following message will appear:


localhost says

Item Already Added

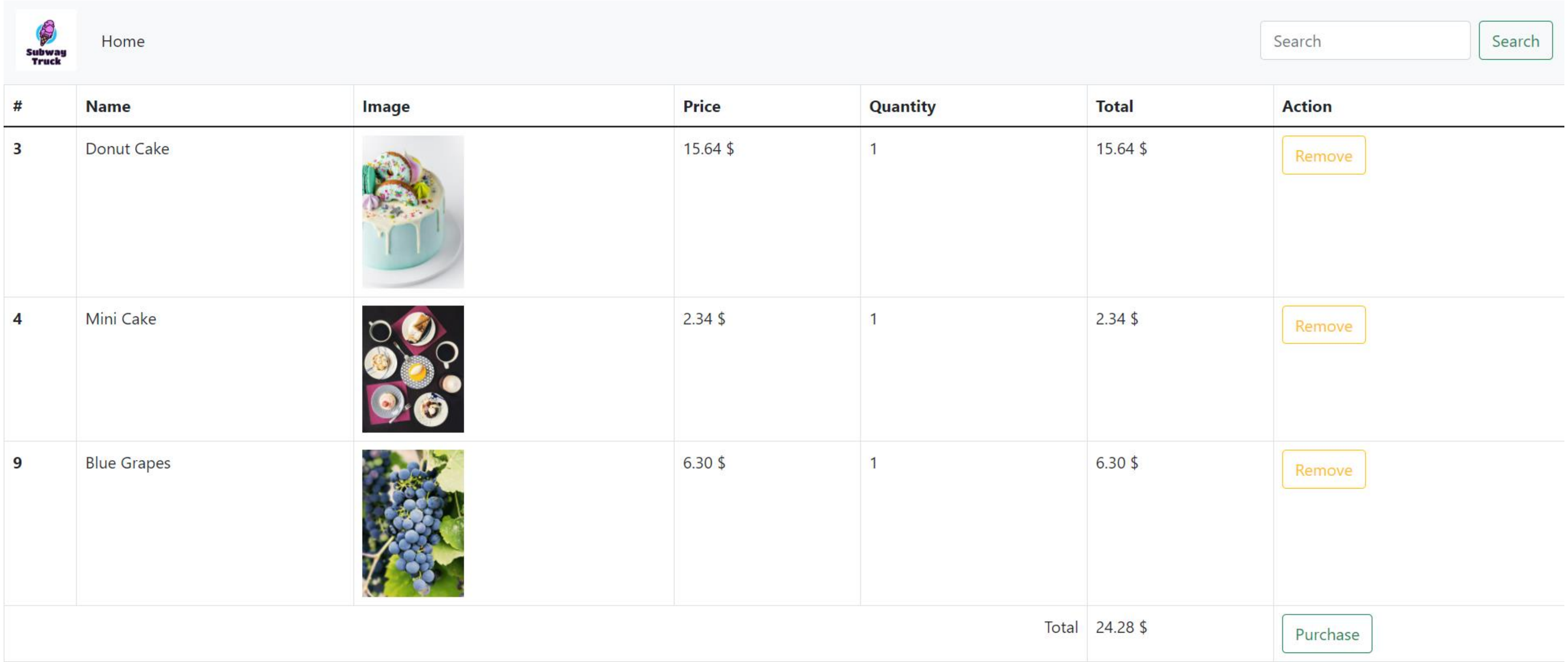
The screenshot below is the code that handles the action when the user chooses to add an item to the cart:



```php
<?php
//Create a session
include './config.php';
session_start();
// session_destroy();
// die();
// Check whether a session is created, if not , create one
if (isset($_POST['add_to_cart'])) {
    if (isset($_SESSION['cart'])) {
        $product_id = array_column($_SESSION['cart'], 'product_id');

        if (!in_array($_GET['id'], $product_id)) {
            $count = count($_SESSION['cart']);
            $product_detail = array(
                'product_id' => $_GET['id'],
                'product_name' => $_POST['food_name'],
                'product_image' => $_POST['img_source'],
                'product_price' => $_POST['price_per_each'],
                'product_quantity' => $_POST['quantity']
            );
            $_SESSION['cart'][$count] = $product_detail;
            echo "<script>alert('Item Added')</script>";
            echo "<script>window.location='user_home.php'</script>";
        } else {
            echo "<script>alert('Item Already Added')</script>";
            echo "<script>window.location='user_home.php'</script>";
        }
    } else {
        $product_detail = array(
            'product_id' => $_GET['id'],
            'product_name' => $_POST['food_name'],
            'product_image' => $_POST['img_source'],
            'product_price' => $_POST['price_per_each'],
            'product_quantity' => $_POST['quantity']
        );
        $_SESSION['cart'][0] = $product_detail;
    }
}
```

1.3. Cart

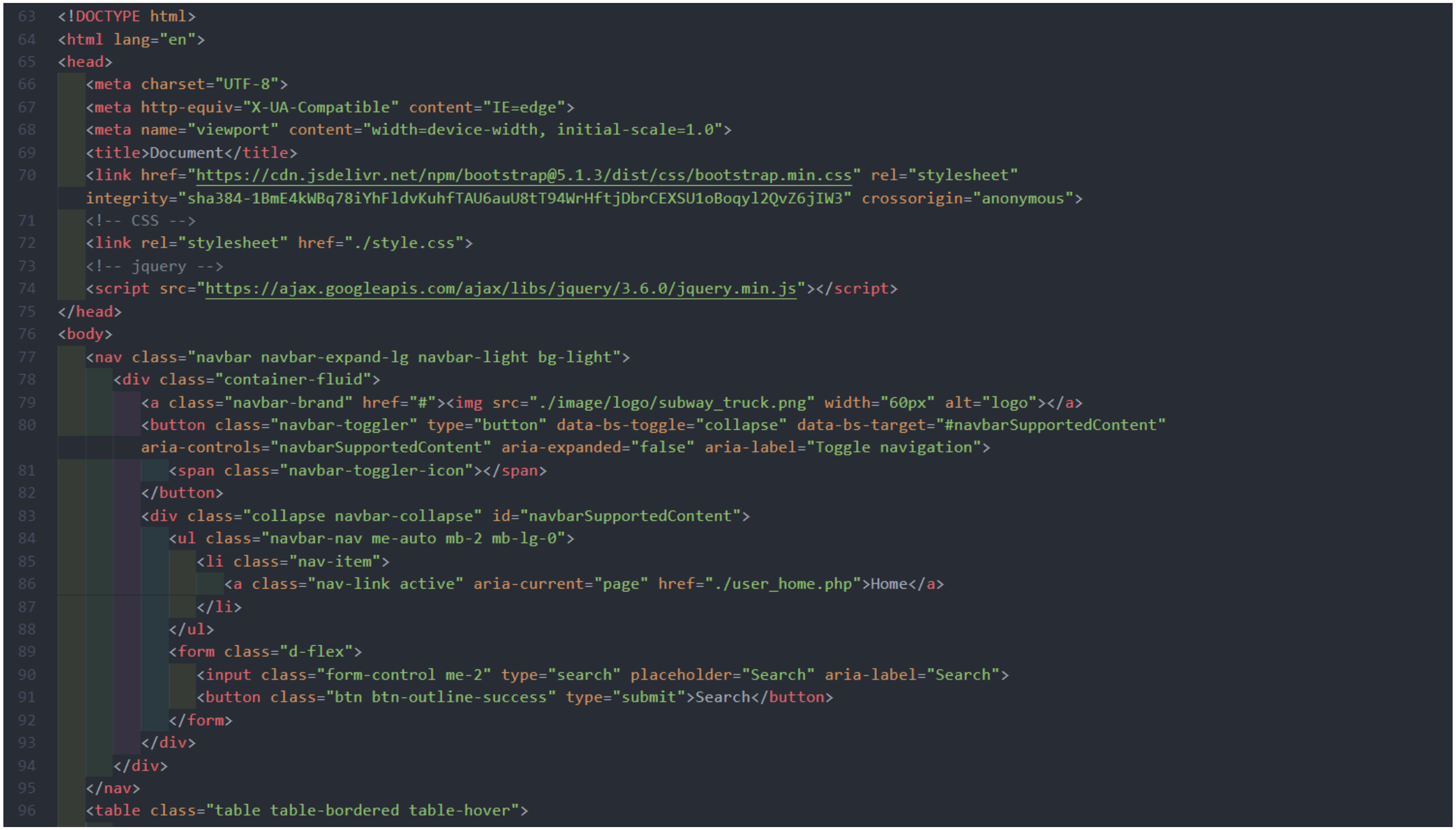
This is the page showing each person's purchase list. Here the purchase list will be arranged in a tabular form, the total amount of the order will be automatically calculated and displayed at the bottom of the table.



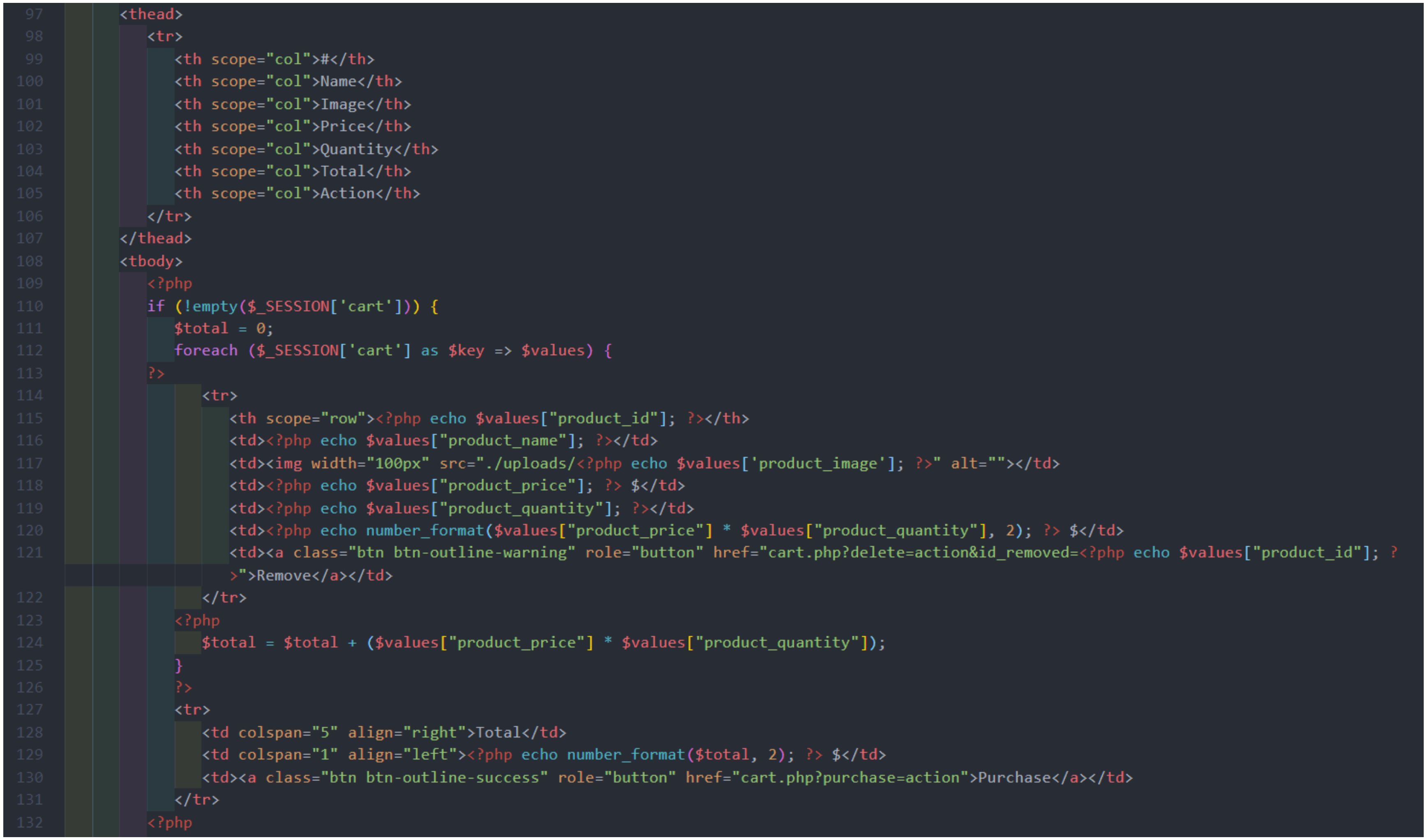
#	Name	Image	Price	Quantity	Total	Action
3	Donut Cake		15.64 $	1	15.64 $	Remove
4	Mini Cake		2.34 $	1	2.34 $	Remove
9	Blue Grapes		6.30 $	1	6.30 $	Remove
				Total	24.28 $	Purchase

Each item that is added will be added to the bottom of the table and the total price will automatically increase.

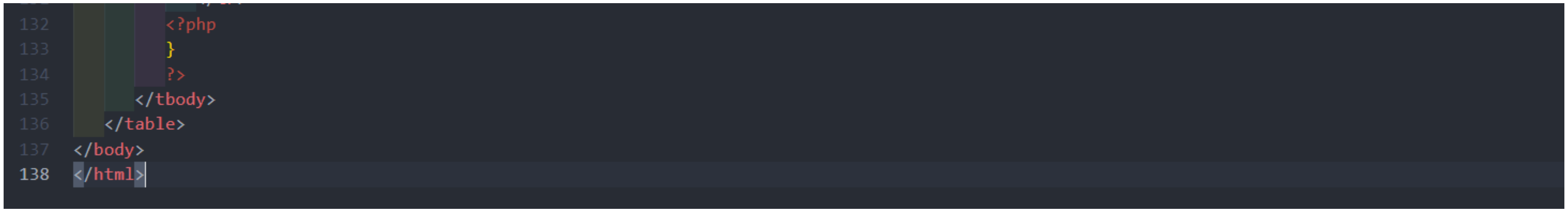
Below are screenshots of the source code used to display the cart page:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
    <!-- CSS -->
    <link rel="stylesheet" href="./style.css">
    <!-- jquery -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fluid">
            <a class="navbar-brand" href="#"><img src="./image/logo/subway_truck.png" width="60px" alt="logo"></a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="./user_home.php">Home</a>
                    </li>
                </ul>
                <form class="d-flex">
                    <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
                    <button class="btn btn-outline-success" type="submit">Search</button>
                </form>
            </div>
        </div>
    </nav>
    <table class="table table-bordered table-hover">
```

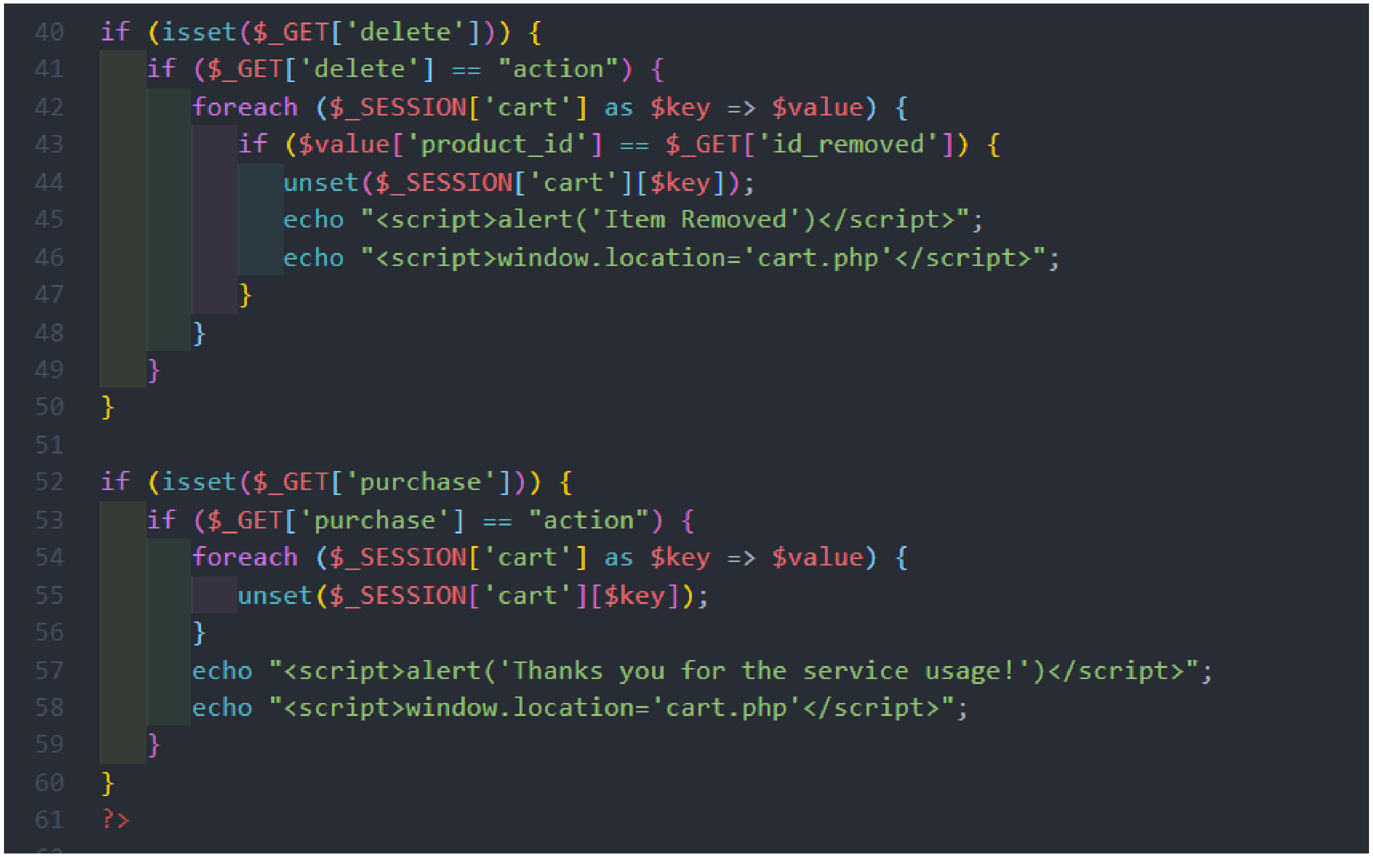


```
<thead>
    <tr>
        <th scope="col">#</th>
        <th scope="col">Name</th>
        <th scope="col">Image</th>
        <th scope="col">Price</th>
        <th scope="col">Quantity</th>
        <th scope="col">Total</th>
        <th scope="col">Action</th>
    </tr>
</thead>
<tbody>
    <?php
    if (!empty($_SESSION['cart'])) {
        $total = 0;
        foreach ($_SESSION['cart'] as $key => $values) {
    ?>
            <tr>
                <th scope="row"><?php echo $values["product_id"]; ?></th>
                <td><?php echo $values["product_name"]; ?></td>
                <td><img width="100px" src="./uploads/<?php echo $values['product_image']; ?>" alt=""></td>
                <td><?php echo $values["product_price"]; ?> $</td>
                <td><?php echo $values["product_quantity"]; ?></td>
                <td><?php echo number_format($values["product_price"] * $values["product_quantity"], 2); ?> $</td>
                <td><a class="btn btn-outline-warning" role="button" href="cart.php?delete=action&id_removed=<?php echo $values["product_id"]; ?>">Remove</a></td>
            </tr>
        <?php
            $total = $total + ($values["product_price"] * $values["product_quantity"]);
        }
        ?>
        <tr>
            <td colspan="5" align="right">Total</td>
            <td colspan="1" align="left"><?php echo number_format($total, 2); ?> $</td>
            <td><a class="btn btn-outline-success" role="button" href="cart.php?purchase=action">Purchase</a></td>
        </tr>
    <?php
```



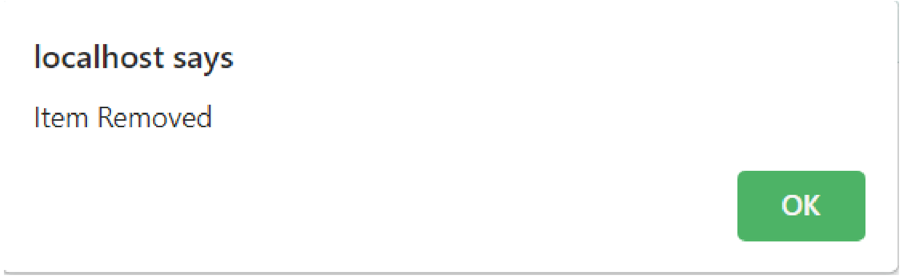
```
            <?php
            }
            ?>
        </tbody>
    </table>
</body>
</html>
```

Screenshot of the code below that handles when there are already a few items in the cart that the user has added, and then can delete some or choose to buy them all.



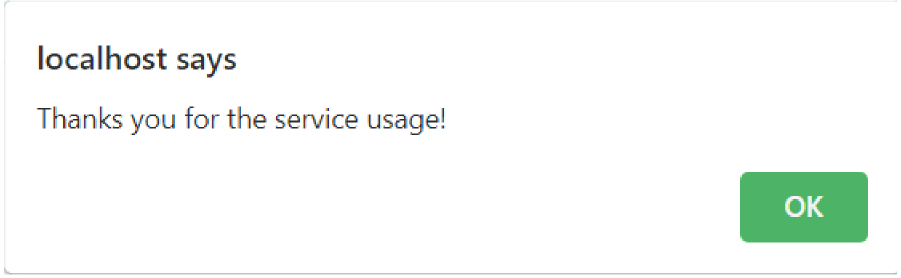
```
if (isset($_GET['delete'])) {
    if ($_GET['delete'] == "action") {
        foreach ($_SESSION['cart'] as $key => $value) {
            if ($value['product_id'] == $_GET['id_removed']) {
                unset($_SESSION['cart'][$key]);
                echo "<script>alert('Item Removed')</script>";
                echo "<script>window.location='cart.php'</script>";
            }
        }
    }
}

if (isset($_GET['purchase'])) {
    if ($_GET['purchase'] == "action") {
        foreach ($_SESSION['cart'] as $key => $value) {
            unset($_SESSION['cart'][$key]);
        }
        echo "<script>alert('Thanks you for the service usage!')</script>";
        echo "<script>window.location='cart.php'</script>";
    }
}
?>
```

The message returned after user removed an item from shopping cart:


localhost says

Item Removed

The message returned after user made payment:

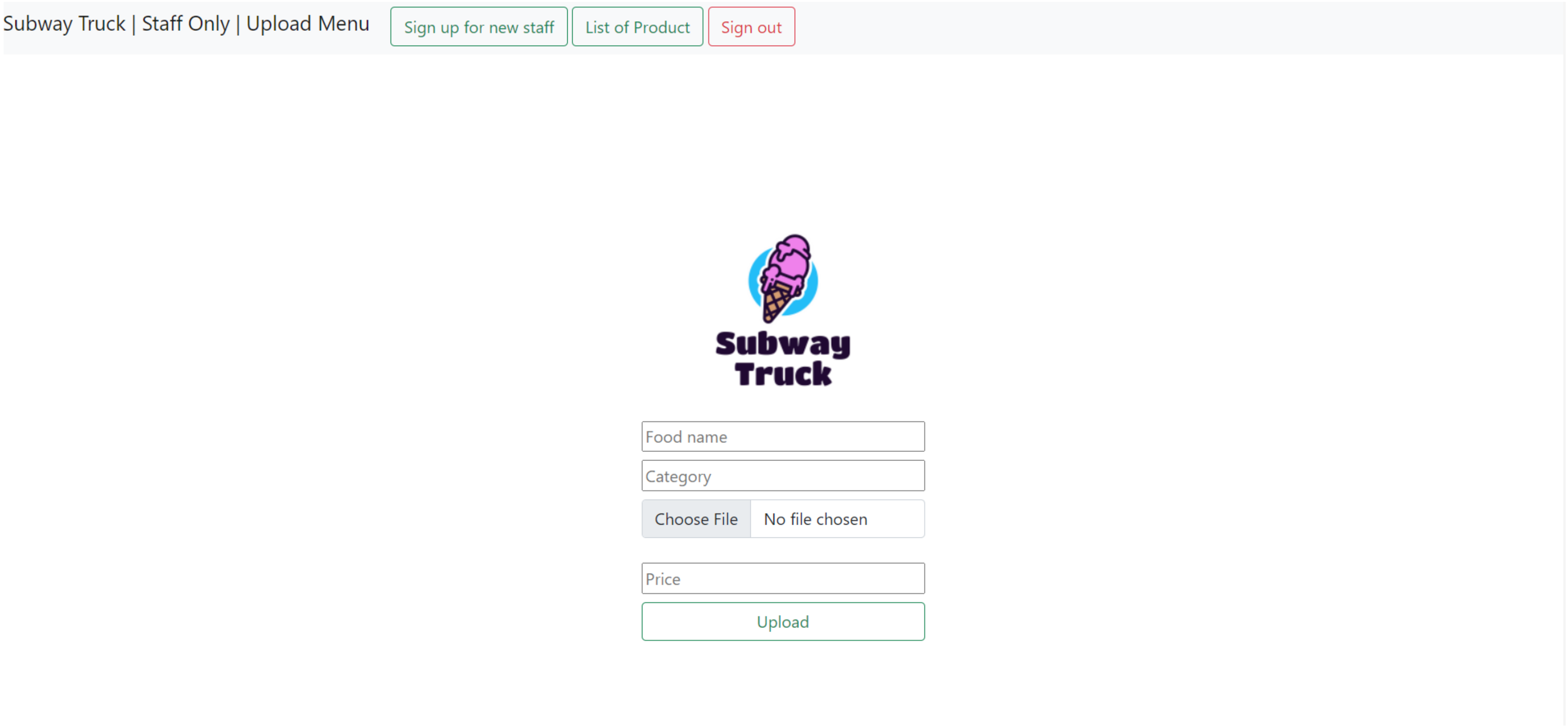

localhost says

Thanks you for the service usage!

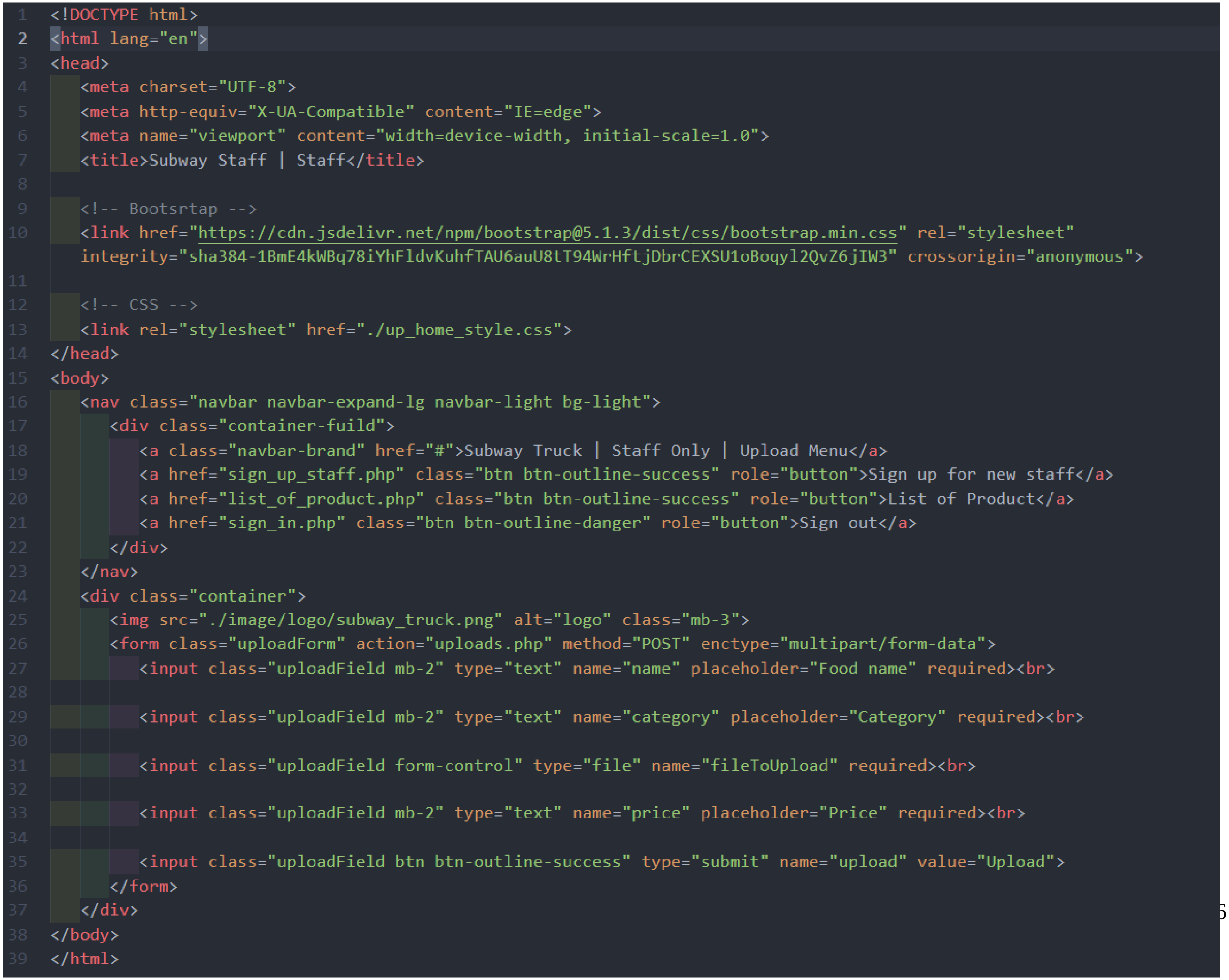
2. Admin (staff) page

2.1. Staff home page

This is the employee's main page after successful login. Here the website is designed with a navigation bar at the top. inside the navigation bar there will be buttons like sign up for a new account, view the dish list, and log out. In the center of the page there is an entry to add items to the list. Here to add a new dish, you will need to enter the full dish name, category, image, and price for that dish. If any item is missing, the system will not accept adding to the restaurant's database.

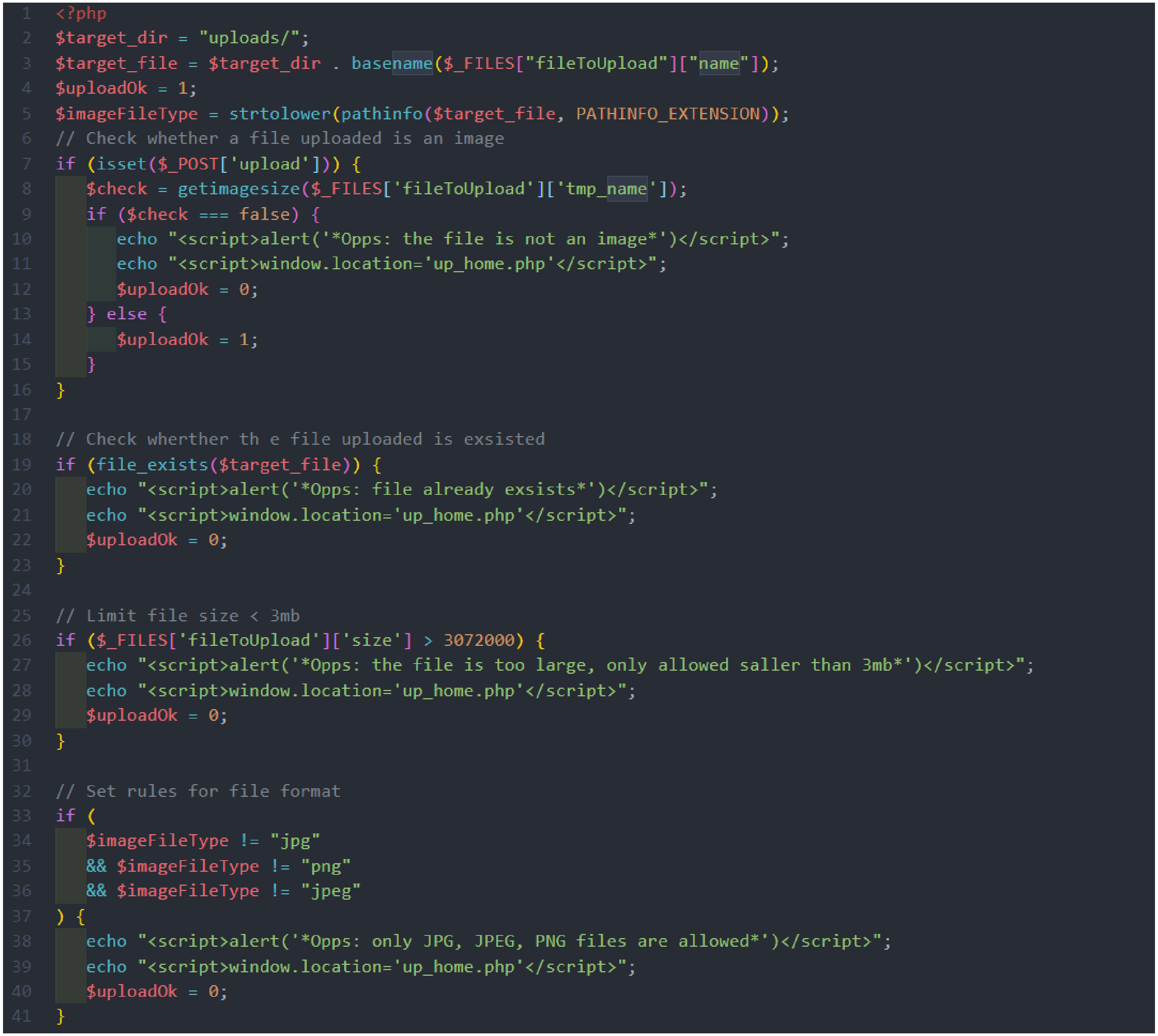


Below are snapshots of the entire system source HTML code used to display the website:

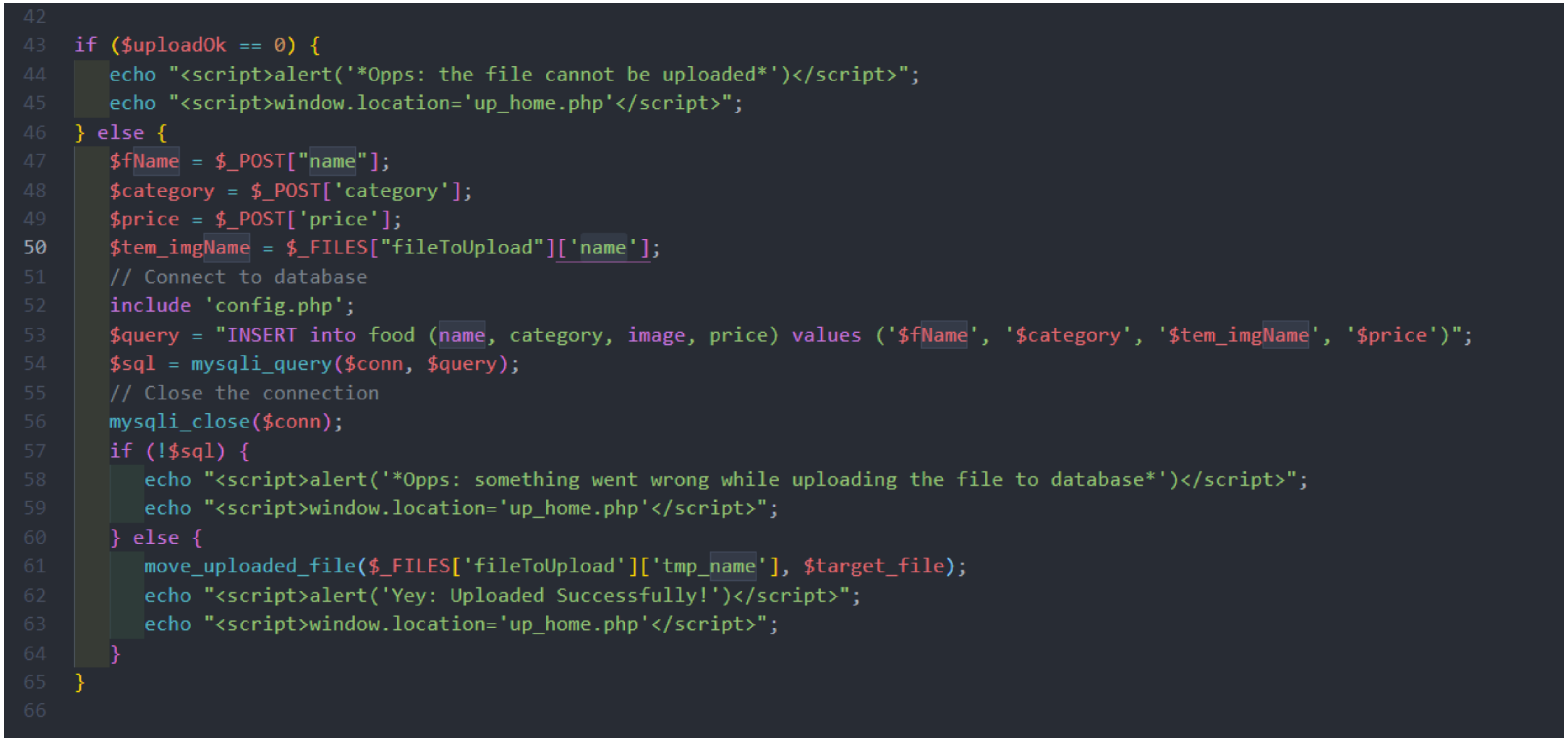


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Subway Staff | Staff</title>

    <!-- Bootsrtap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <!-- CSS -->
    <link rel="stylesheet" href="./up_home_style.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fuild">
            <a class="navbar-brand" href="#">Subway Truck | Staff Only | Upload Menu</a>
            <a href="sign_up_staff.php" class="btn btn-outline-success" role="button">Sign up for new staff</a>
            <a href="list_of_product.php" class="btn btn-outline-success" role="button">List of Product</a>
            <a href="sign_in.php" class="btn btn-outline-danger" role="button">Sign out</a>
        </div>
    </nav>
    <div class="container">
        <img src="./image/logo/subway_truck.png" alt="logo" class="mb-3">
        <form class="uploadForm" action="uploads.php" method="POST" enctype="multipart/form-data">
            <input class="uploadField mb-2" type="text" name="name" placeholder="Food name" required><br>

            <input class="uploadField mb-2" type="text" name="category" placeholder="Category" required><br>

            <input class="uploadField form-control" type="file" name="fileToUpload" required><br>

            <input class="uploadField mb-2" type="text" name="price" placeholder="Price" required><br>

            <input class="uploadField btn btn-outline-success" type="submit" name="upload" value="Upload">
        </form>
    </div>
</body>
</html>
```

Below is a snapshot of the PHP code used to handle tasks when employees work with the website

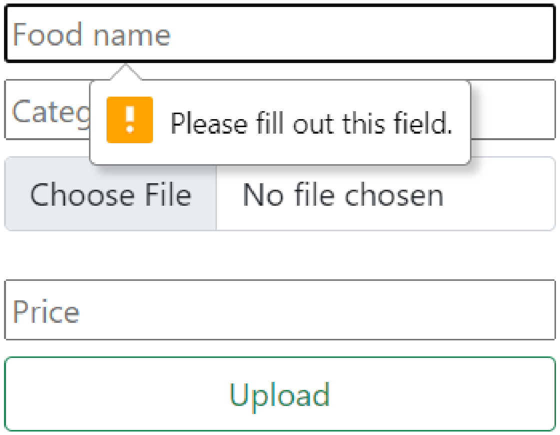


```php
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
// Check whether a file uploaded is an image
if (isset($_POST['upload'])) {
    $check = getimagesize($_FILES['fileToUpload']['tmp_name']);
    if ($check === false) {
        echo "<script>alert('*Opps: the file is not an image*')</script>";
        echo "<script>window.location='up_home.php'</script>";
        $uploadOk = 0;
    } else {
        $uploadOk = 1;
    }
}

// Check wherther th e file uploaded is exsisted
if (file_exists($target_file)) {
    echo "<script>alert('*Opps: file already exsists*')</script>";
    echo "<script>window.location='up_home.php'</script>";
    $uploadOk = 0;
}

// Limit file size < 3mb
if ($_FILES['fileToUpload']['size'] > 3072000) {
    echo "<script>alert('*Opps: the file is too large, only allowed saller than 3mb*')</script>";
    echo "<script>window.location='up_home.php'</script>";
    $uploadOk = 0;
}

// Set rules for file format
if (
    $imageFileType != "jpg"
    && $imageFileType != "png"
    && $imageFileType != "jpeg"
) {
    echo "<script>alert('*Opps: only JPG, JPEG, PNG files are allowed*')</script>";
    echo "<script>window.location='up_home.php'</script>";
    $uploadOk = 0;
}
```



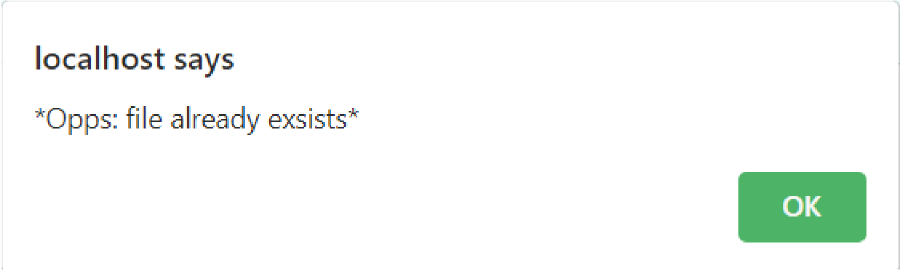
```
if ($uploadOk == 0) {
    echo "<script>alert('*Opps: the file cannot be uploaded*')</script>";
    echo "<script>window.location='up_home.php'</script>";
} else {
    $fName = $_POST["name"];
    $category = $_POST['category'];
    $price = $_POST['price'];
    $tem_imgName = $_FILES["fileToUpload"]['name'];
    // Connect to database
    include 'config.php';
    $query = "INSERT into food (name, category, image, price) values ('$fName', '$category', '$tem_imgName', '$price')";
    $sql = mysqli_query($conn, $query);
    // Close the connection
    mysqli_close($conn);
    if (!$sql) {
        echo "<script>alert('*Opps: something went wrong while uploading the file to database*')</script>";
        echo "<script>window.location='up_home.php'</script>";
    } else {
        move_uploaded_file($_FILES['fileToUpload']['tmp_name'], $target_file);
        echo "<script>alert('Yey: Uploaded Successfully!')</script>";
        echo "<script>window.location='up_home.php'</script>";
    }
}
```

The system will return a message if an extra data cell was empty:



Please fill out this field.

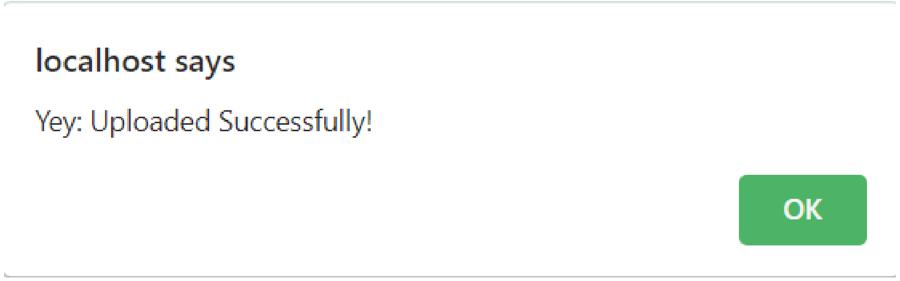
The system will return a notification if the image of the dish already exists:



localhost says

Opps: file already exsists

The system will return a message if adding dishes successfully:

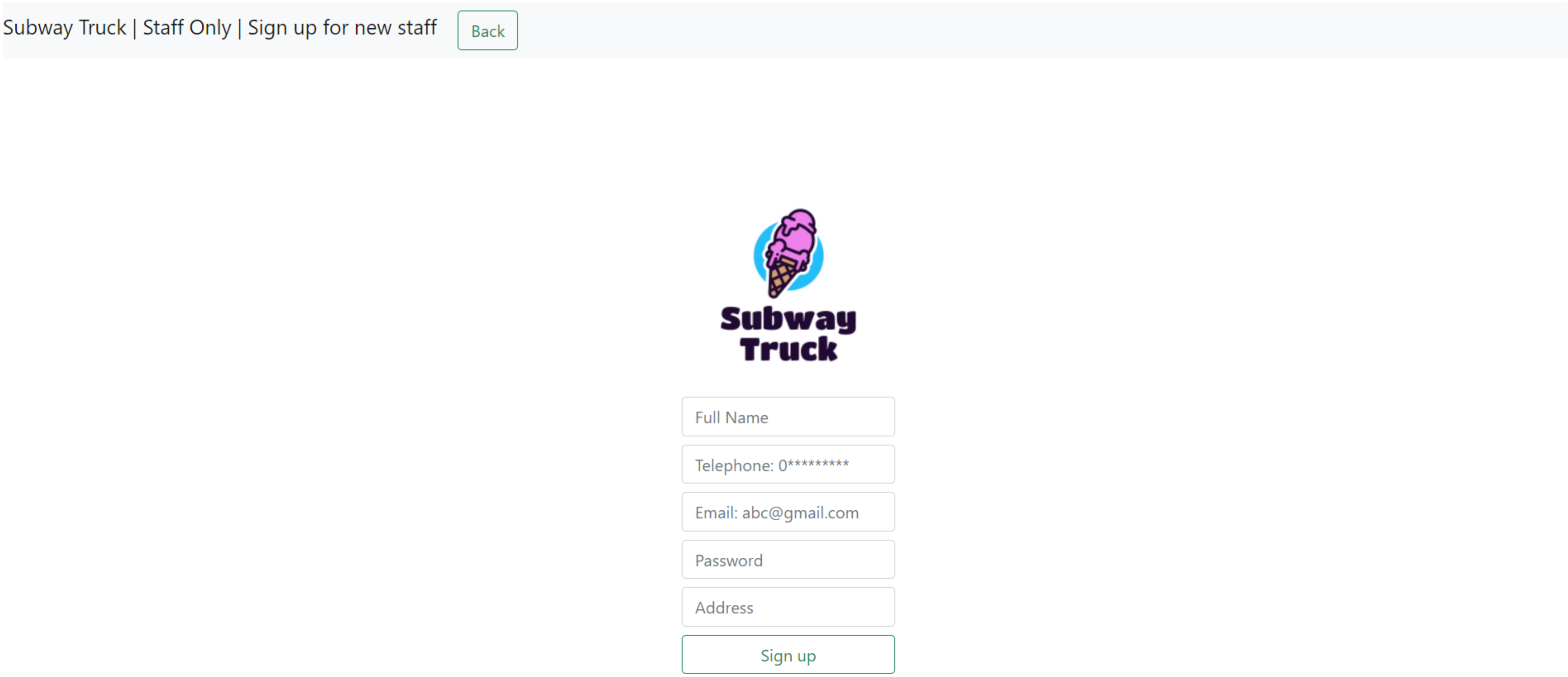


localhost says

Yey: Uploaded Successfully!

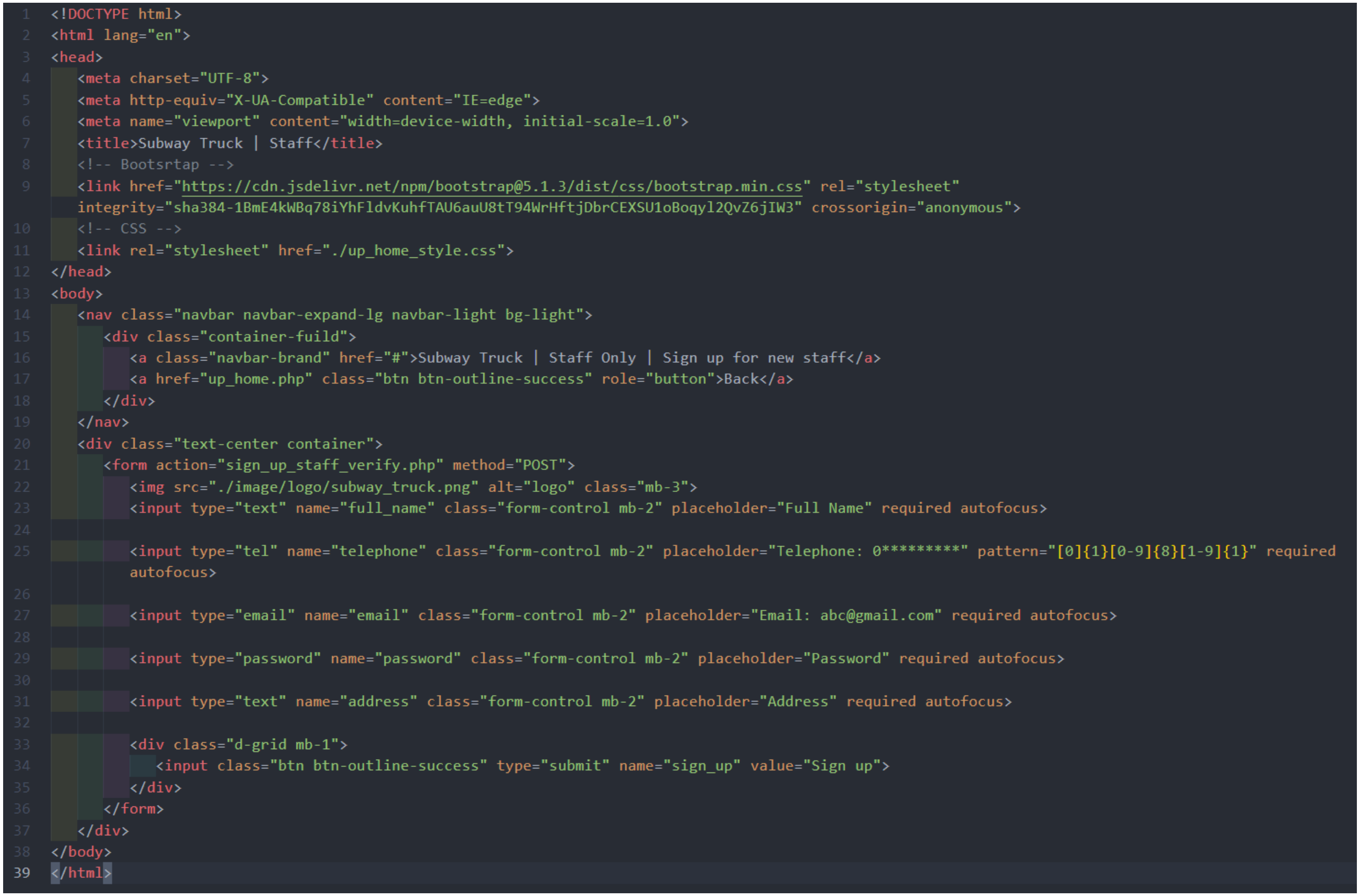
2.2. Sign up new staff account

This page works almost the same way as the guest registration page. Employees will have to enter the new employee's full name, email, phone number, address, and password:



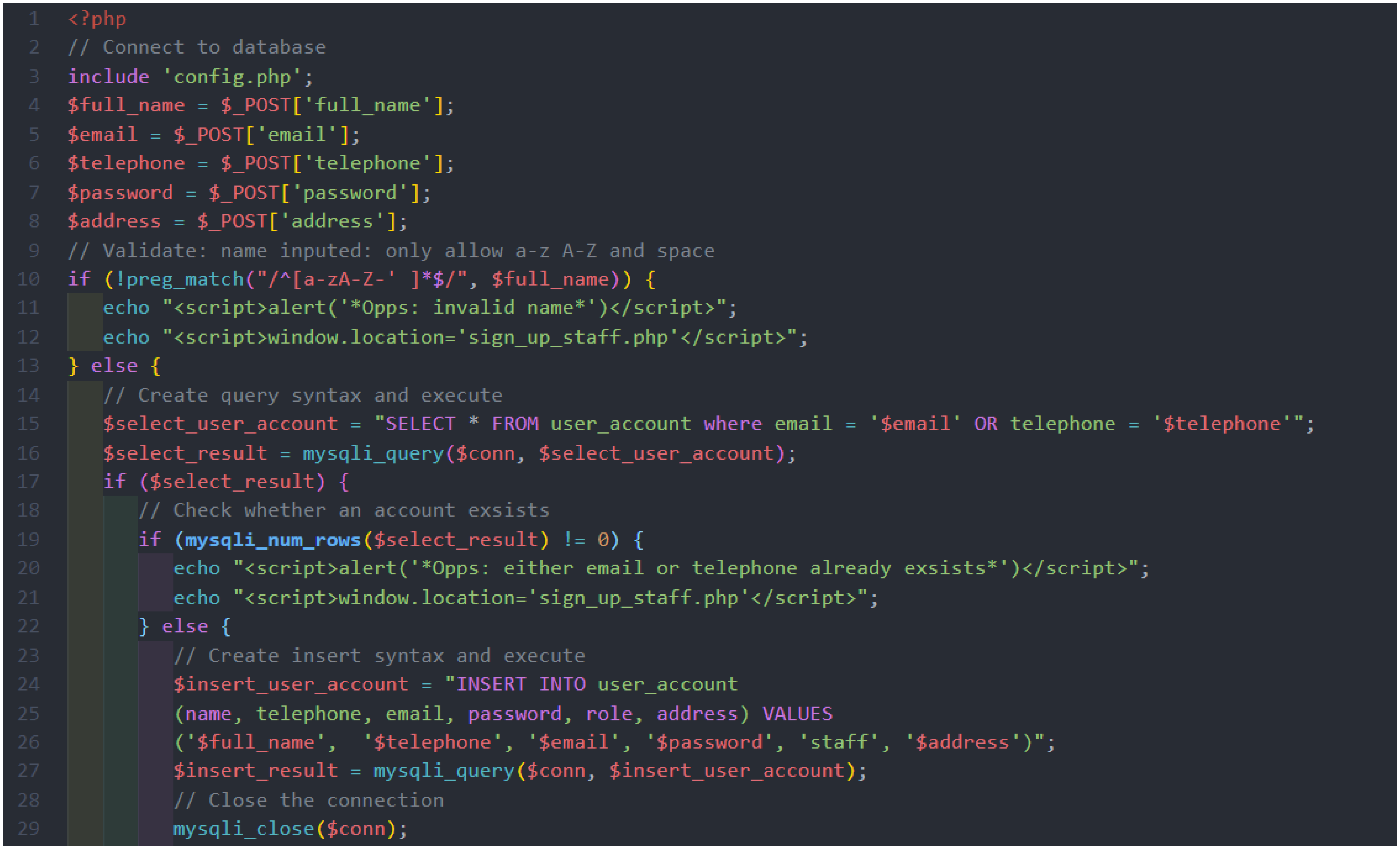
Subway Truck | Staff Only | Sign up for new staff

Below is a screenshot of the source code used to display the interface:

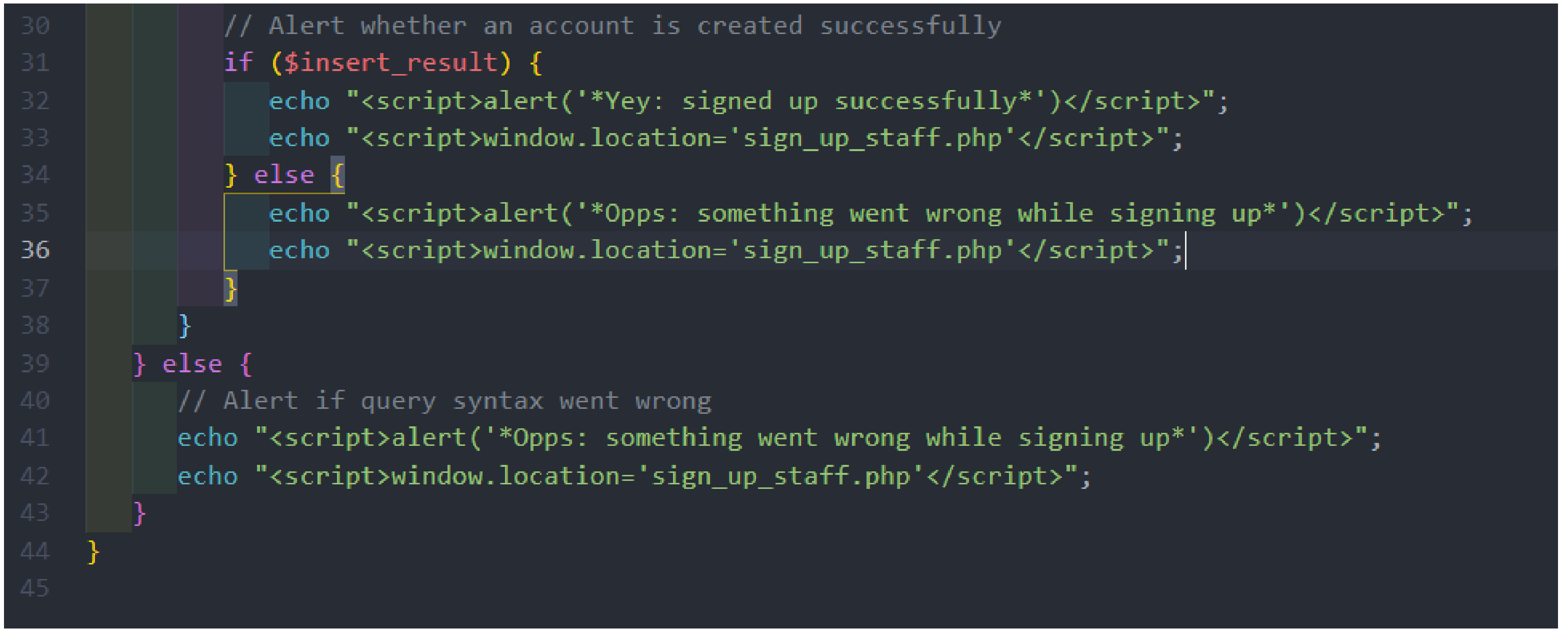


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Subway Truck | Staff</title>
    <!-- Bootsrtap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
    <!-- CSS -->
    <link rel="stylesheet" href="./up_home_style.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fuild">
            <a class="navbar-brand" href="#">Subway Truck | Staff Only | Sign up for new staff</a>
            <a href="up_home.php" class="btn btn-outline-success" role="button">Back</a>
        </div>
    </nav>
    <div class="text-center container">
        <form action="sign_up_staff_verify.php" method="POST">
            <img src="./image/logo/subway_truck.png" alt="logo" class="mb-3">
            <input type="text" name="full_name" class="form-control mb-2" placeholder="Full Name" required autofocus>

            <input type="tel" name="telephone" class="form-control mb-2" placeholder="Telephone: 0*********" pattern="[0]{1}[0-9]{8}[1-9]{1}" required autofocus>

            <input type="email" name="email" class="form-control mb-2" placeholder="Email: abc@gmail.com" required autofocus>

            <input type="password" name="password" class="form-control mb-2" placeholder="Password" required autofocus>

            <input type="text" name="address" class="form-control mb-2" placeholder="Address" required autofocus>

            <div class="d-grid mb-1">
                <input class="btn btn-outline-success" type="submit" name="sign_up" value="Sign up">
            </div>
        </form>
    </div>
</body>
</html>
```

When pressing the sign-up button, the system will check if the email, phone number and full name entered is in the correct format or is the same as another employee's account, if valid, the account will be created, otherwise the system will notice and have to re-sign-up. Below is the code:

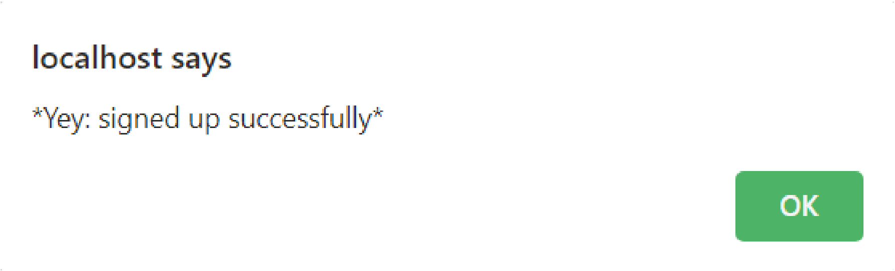


```php
<?php
// Connect to database
include 'config.php';
$full_name = $_POST['full_name'];
$email = $_POST['email'];
$telephone = $_POST['telephone'];
$password = $_POST['password'];
$address = $_POST['address'];
// Validate: name inputed: only allow a-z A-Z and space
if (!preg_match("/^[a-zA-Z-' ]*$/", $full_name)) {
    echo "<script>alert('*Opps: invalid name*')</script>";
    echo "<script>window.location='sign_up_staff.php'</script>";
} else {
    // Create query syntax and execute
    $select_user_account = "SELECT * FROM user_account where email = '$email' OR telephone = '$telephone'";
    $select_result = mysqli_query($conn, $select_user_account);
    if ($select_result) {
        // Check whether an account exsists
        if (mysqli_num_rows($select_result) != 0) {
            echo "<script>alert('*Opps: either email or telephone already exsists*')</script>";
            echo "<script>window.location='sign_up_staff.php'</script>";
        } else {
            // Create insert syntax and execute
            $insert_user_account = "INSERT INTO user_account
            (name, telephone, email, password, role, address) VALUES
            ('$full_name',  '$telephone', '$email', '$password', 'staff', '$address')";
            $insert_result = mysqli_query($conn, $insert_user_account);
            // Close the connection
            mysqli_close($conn);
```



```
            // Alert whether an account is created successfully
            if ($insert_result) {
                echo "<script>alert('*Yey: signed up successfully*')</script>";
                echo "<script>window.location='sign_up_staff.php'</script>";
            } else {
                echo "<script>alert('*Opps: something went wrong while signing up*')</script>";
                echo "<script>window.location='sign_up_staff.php'</script>";
            }
        }
    } else {
        // Alert if query syntax went wrong
        echo "<script>alert('*Opps: something went wrong while signing up*')</script>";
        echo "<script>window.location='sign_up_staff.php'</script>";
    }
}
```

Upon successful account creation, the following message will appear:

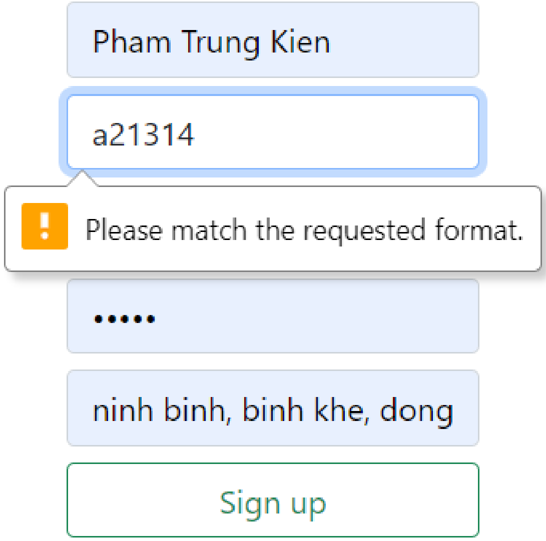


localhost says

Yey: signed up successfully

The system will return the following messages if there are errors during registration:

- Phone number is not in the correct form:



Pham Trung Kien

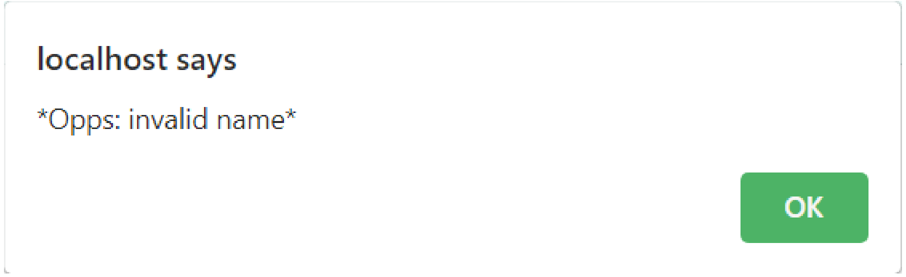
a21314

Please match the requested format.

•••••

ninh binh, binh khe, dong

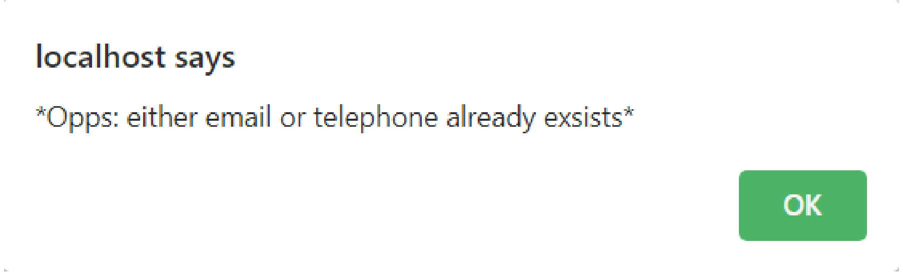
- Name dose not contain only letter and white space:



localhost says

Opps: invalid name

- Email or phone number already exists:

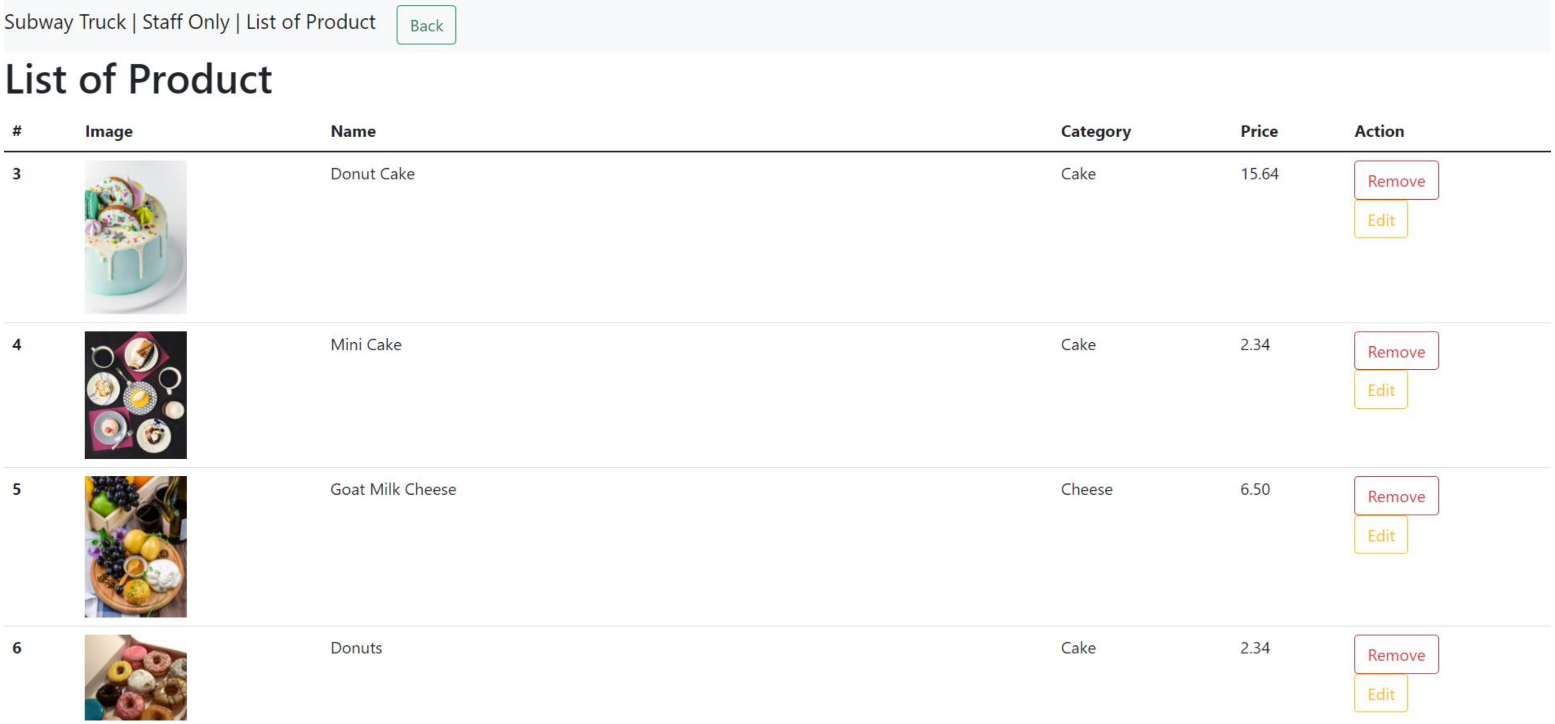


localhost says

Opps: either email or telephone already exsists

2.3. List of products

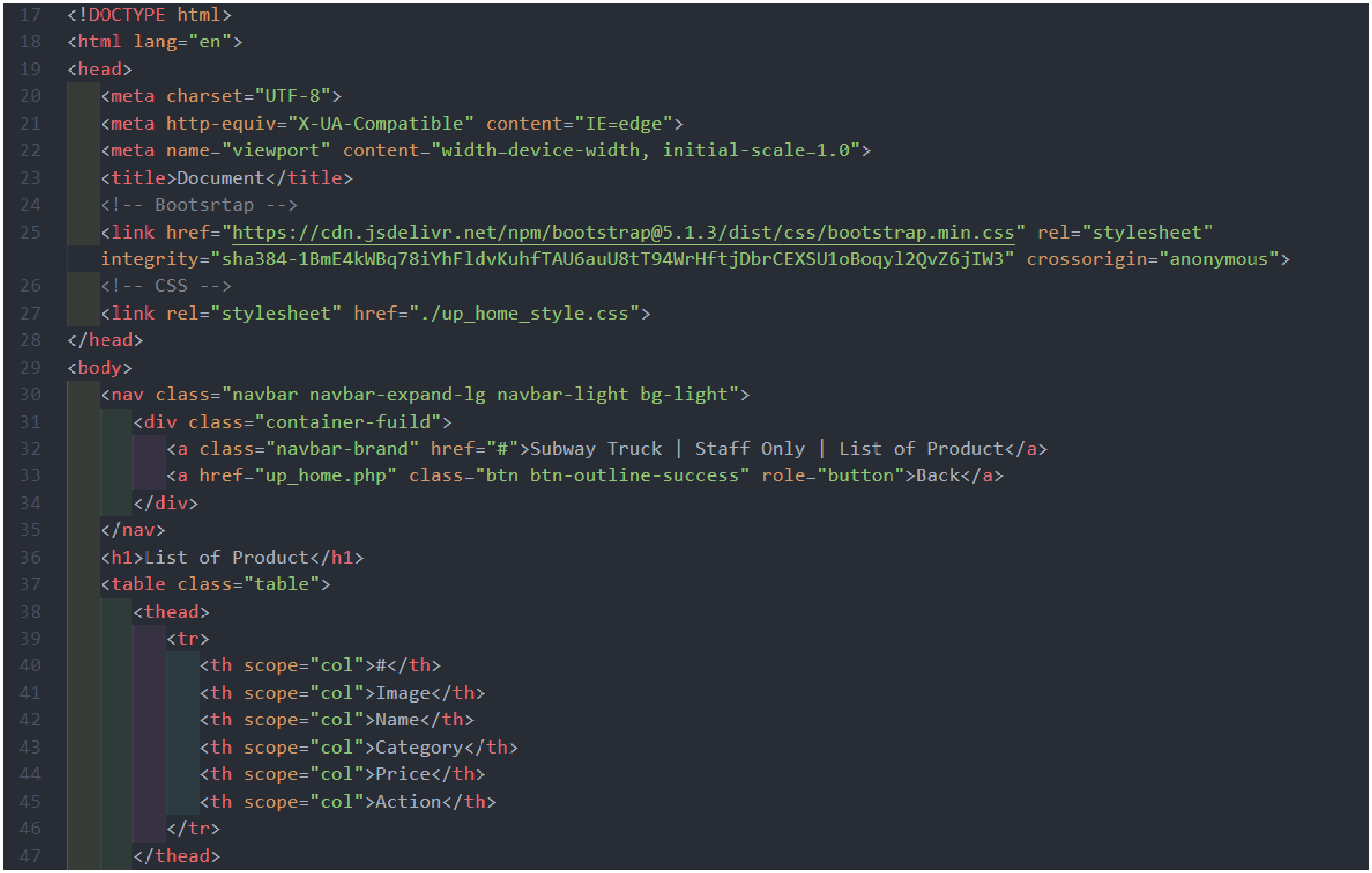
On this page, there will be a list containing information about the dishes that the restaurant serves displayed. Employees can see what food the restaurant they are working at serves, and they also have the right to delete that dish at the request of the boss. Although there is an item to edit dish information, this feature is still in development so it cannot be used yet.



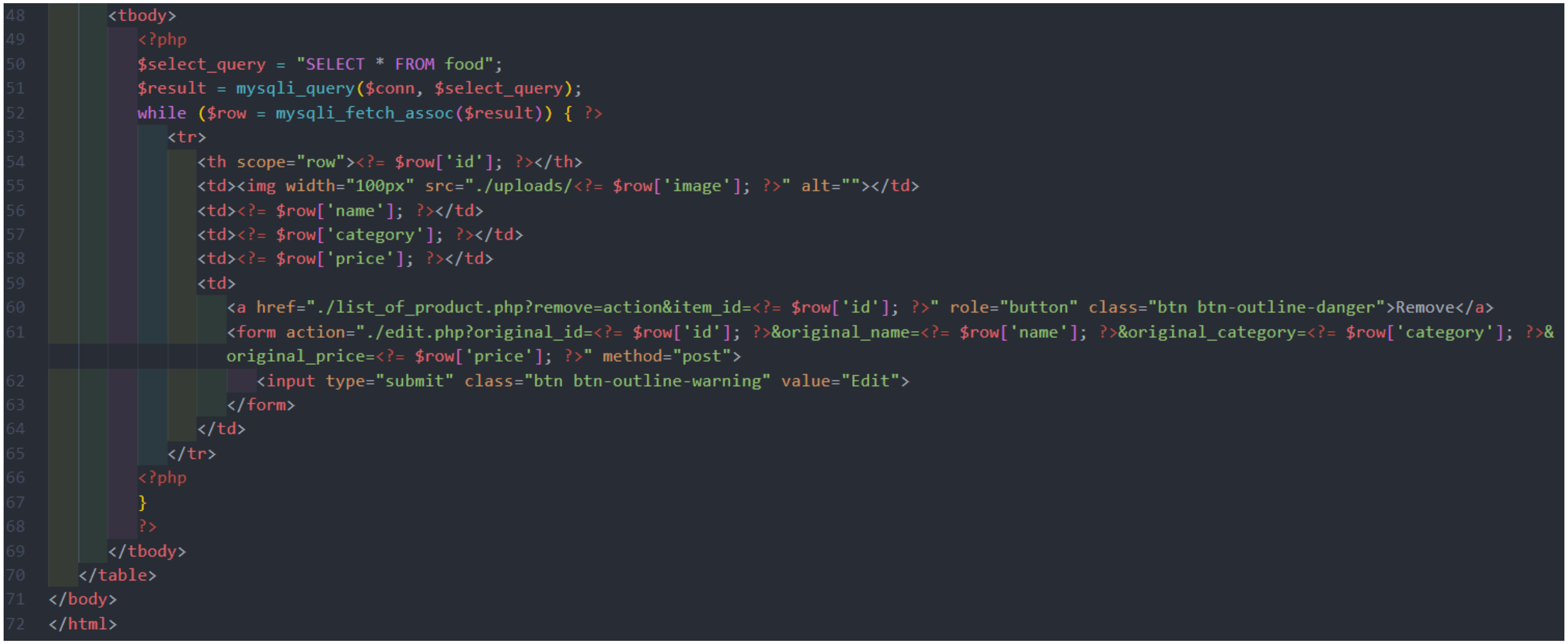
List of Product

#	Image	Name	Category	Price	Action
3		Donut Cake	Cake	15.64	Remove Edit
4		Mini Cake	Cake	2.34	Remove Edit
5		Goat Milk Cheese	Cheese	6.50	Remove Edit
6		Donuts	Cake	2.34	Remove Edit

The source code to display the page is shown in the screenshot below:

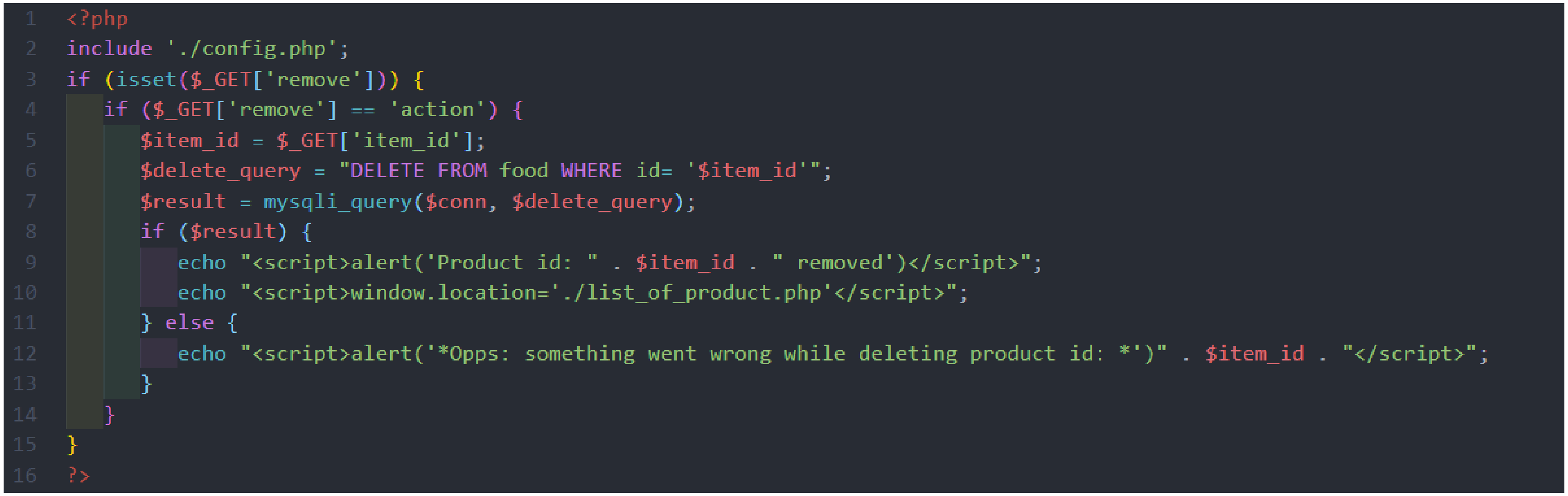


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <!-- Bootsrtap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
    <!-- CSS -->
    <link rel="stylesheet" href="./up_home_style.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fuild">
            <a class="navbar-brand" href="#">Subway Truck | Staff Only | List of Product</a>
            <a href="up_home.php" class="btn btn-outline-success" role="button">Back</a>
        </div>
    </nav>
    <h1>List of Product</h1>
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Image</th>
                <th scope="col">Name</th>
                <th scope="col">Category</th>
                <th scope="col">Price</th>
                <th scope="col">Action</th>
            </tr>
        </thead>
```



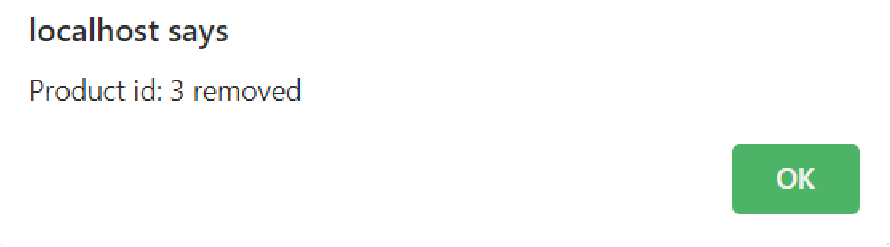
```
        <tbody>
            <?php
            $select_query = "SELECT * FROM food";
            $result = mysqli_query($conn, $select_query);
            while ($row = mysqli_fetch_assoc($result)) { ?>
                <tr>
                    <th scope="row"><?= $row['id']; ?></th>
                    <td><img width="100px" src="./uploads/<?= $row['image']; ?>" alt=""></td>
                    <td><?= $row['name']; ?></td>
                    <td><?= $row['category']; ?></td>
                    <td><?= $row['price']; ?></td>
                    <td>
                        <a href="./list_of_product.php?remove=action&item_id=<?= $row['id']; ?>" role="button" class="btn btn-outline-danger">Remove</a>
                        <form action="./edit.php?original_id=<?= $row['id']; ?>&original_name=<?= $row['name']; ?>&original_category=<?= $row['category']; ?>&original_price=<?= $row['price']; ?>" method="post">
                            <input type="submit" class="btn btn-outline-warning" value="Edit">
                        </form>
                    </td>
                </tr>
            <?php
            }
            ?>
        </tbody>
    </table>
</body>
</html>
```

When an employee wants to remove an item from the list, they just need to press the delete button, then the system will receive the request and delete that dish. Below is the code used to process the request to delete the dish:



```php
<?php
include './config.php';
if (isset($_GET['remove'])) {
    if ($_GET['remove'] == 'action') {
        $item_id = $_GET['item_id'];
        $delete_query = "DELETE FROM food WHERE id= '$item_id'";
        $result = mysqli_query($conn, $delete_query);
        if ($result) {
            echo "<script>alert('Product id: " . $item_id . " removed')</script>";
            echo "<script>window.location='./list_of_product.php'</script>";
        } else {
            echo "<script>alert('*Opps: something went wrong while deleting product id: *')" . $item_id . "</script>";
        }
    }
}
?>
```

When successfully deleting a dish, there will be a message as follows:



localhost says

Product id: 3 removed

VI. Test plan

#	Scope	Detail	Expected result	Actual result	Status
1.	Sign-in	Sign-in with user account	• Signed in successfully • User is taken to user home page	• Signed in successfully • User is taken to user home page	Passed

2.		Sign-in with staff account	• Signed in successfully • Staff is taken to staff home page	• Signed in successfully • Staff is taken to staff home page	Passed
3.		Sign-in with incorect account	• Error message is displayed	• Error message is displayed	Passed
4.		Sign-in with incorect account (email or password is missed)	• Error message is displayed	• Error message is displayed	Passed
5.	Sign-up (user)	Sign-up with valid data	• Signed up successfully message	• Signed up successfully message	Passed
6.		Sign-up with data exsists	• Error message is displayed	• Error message is displayed	Passed
7.		Sign-up with data missed	• Error message is displayed	• Error message is displayed	Passed
8.		Sign-up with inlvaid name	• Error message is displayed	• Error message is displayed	Passed
9.		Sign-up with inlvaid phone number	• Error message is displayed	• Error message is displayed	Passed
10		Sign-up with inlvaid email	• Error message is displayed	• Error message is displayed	Passed
11	User home page	Add an item to cart (first time)	• Added successfully message displayed	• Added successfully message displayed	Passed
12		Add an item to cart (second time)	• Error message is displayed	• Error message is displayed	Passed
13	Cart	Remove an item	• Removed successfully message displayed	• Removed successfully message displayed	Passed
14		Make payment	• Made payment successfully message displayed	• Made payment successfully message displayed	Passed
15	Add dish (staff)	Add new dish with normal data	• Added successfully message displayed	• Added successfully message displayed	Passed
16		Add new dish with data missed	• Error message is displayed	• Error message is displayed	Passed
17		Add new dish with file is not image	• Error message is displayed	• Error message is displayed	Passed
18		Add new dish with image is not JPG, JPEG, PNG	• Error message is displayed	• Error message is displayed	Passed
19		Add new dish with image exsists	• Error message is displayed	• Error message is displayed	Passed
20	Sign-up staff account	Sign-up with valid data	• Signed up successfully message displayed	• Signed up successfully message displayed	Passed
21		Sign-up with data exsists	• Error message is displayed	• Error message is displayed	Passed
22		Sign-up with data missed	• Error message is displayed	• Error message is displayed	Passed

23		Sign-up with inlvaid name	• Error message is displayed	• Error message is displayed	Passed
24		Sign-up with inlvaid phone number	• Error message is displayed	• Error message is displayed	Passed
25		Sign-up with inlvaid email	• Error message is displayed	• Error message is displayed	Passed
26	List of products	Remove a product	• Removed successfully message displayed	• Removed successfully message displayed	Passed

VII. Test evaluation

Based on user requirements and the website after being built, the author has performed some tests on the website's features to check if it works smoothly and stably. In general, the website's functions have been implemented very well by the author, he has also foreseen a lot of situations where users can make mistakes and prevent them, so the website works very smoothly and stably. determined.

The features of the website have been properly tested, analyzed and run for the most part. By carefully checking the input, incorrect input of data format or duplicate data is completely prevented. Therefore, the consistency in the data of subwayfood.co's database can be ensured.

VIII. Website evaluation

After completing the website's development, the developer created a website that met the needs of the users while also meeting the requirements of the e-commerce website. Users and staff members will be able to perform the majority of the specific tasks with simple user rights. And here is a post-development assessment of the site's strengths and weaknesses, as well as plans for future development.

1. Advantage

- There are animations used to highlight the object that the user is pointing the mouse at.
- The style is light and unfussy, while sticking to website design principles and paying attention to details.
- Meet all of the user's specific requirements.
- User rights are clear.
- Minimalist design.
- Able to ensure that the user input data is safe and does not harm the website.

2. Disavnatge

- There are not many roles.
- The number of animations is limited.
- Cannot align the product display information.
- Missing some functions like searching or sorting products.
- No payment method yet.
- There is no order management function for employees yet.
- The order cannot be saved when the user exits the site.

3. Future improvement

In the future, in order for the website to grow stronger, it needs to be further optimized as well as add many other useful features to become a true e-commerce site.

- Integrate more payment gateways such as mono and vnpay e-wallets. Transfer via bank card, visa, mastercard or COD.
- Adjust the display information of the dishes.
- Add order management function for employees.
- Add the function of adding or reducing the number of products in the cart as well as choosing the items to pay instead of having to buy them all in the cart.

IX. Compare and contrast the multipage website created to the design documents

1. User requirement

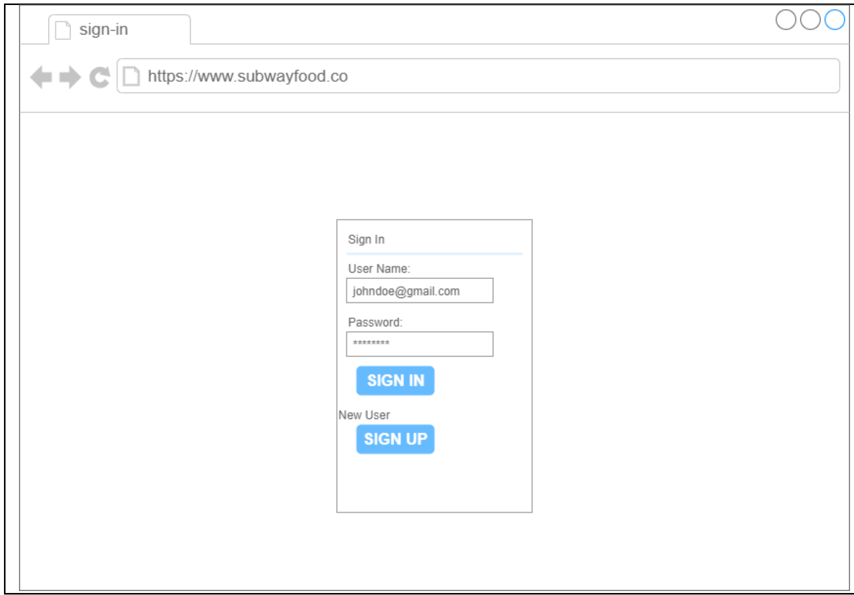
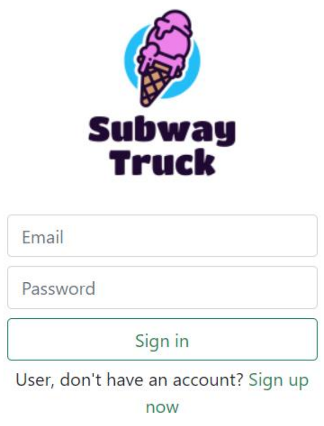
In summary, the comparison between user requirements and actual results that the author has implemented can see that he has met the website requirements very well, he even added some features. help increase the security of the website. From these things, it can be seen that he has performed the assigned task very well.

Task	User	Staff	Status
Sign in	●	●	Passed
Sign up	●	●	Passed
Side based on role	●	●	Passed
Add a dish to cart	●		Passed
Remove a dish from cart	●		Passed
Add a new dish list list		●	Passed
Remove a dish from list		●	Passed
Make payment	●		Passed
View list of dishes		●	Passed

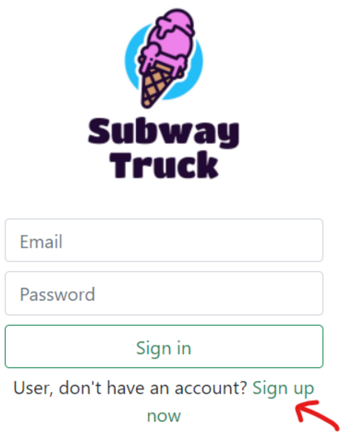
2. Website design prototype and result

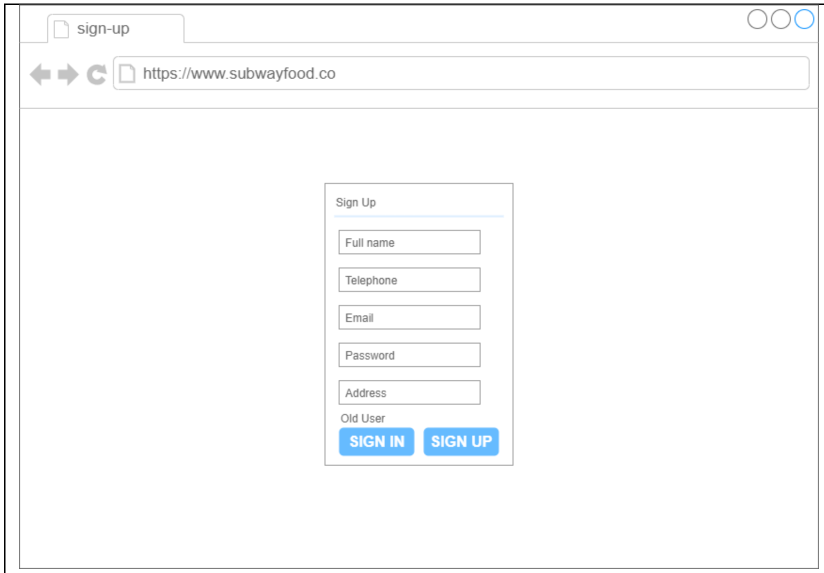
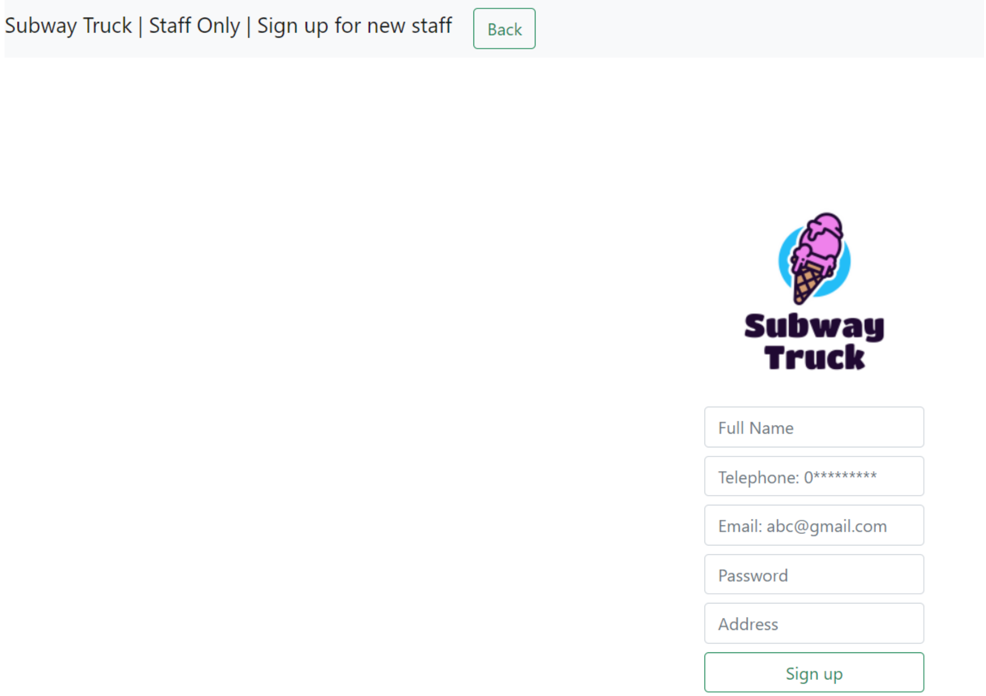
2.1. Sign-in

Prototype	Actual result

 sign-in https://www.subwayfood.co Sign In User Name: johndoe@gmail.com Password: ******** SIGN IN New User SIGN UP	 User, don't have an account? Sign up now

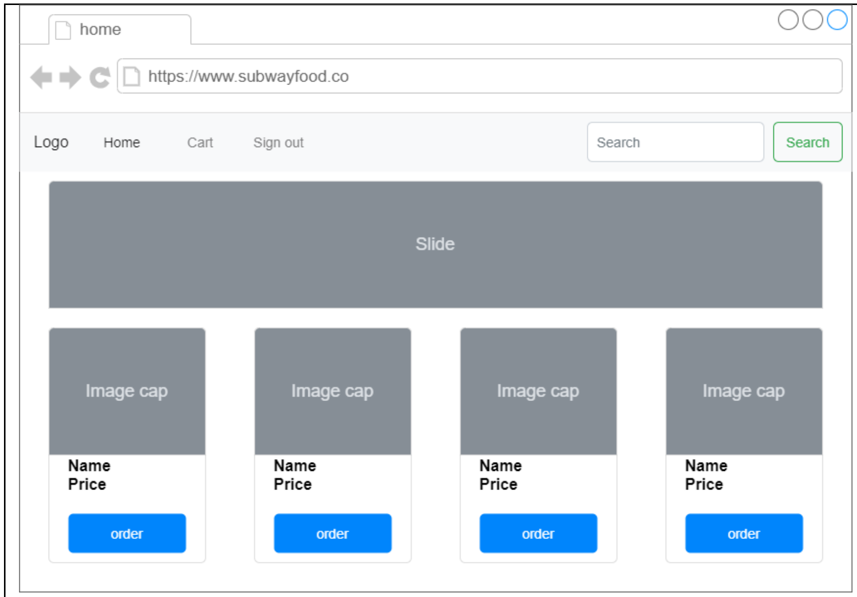
2.2. Sign-up

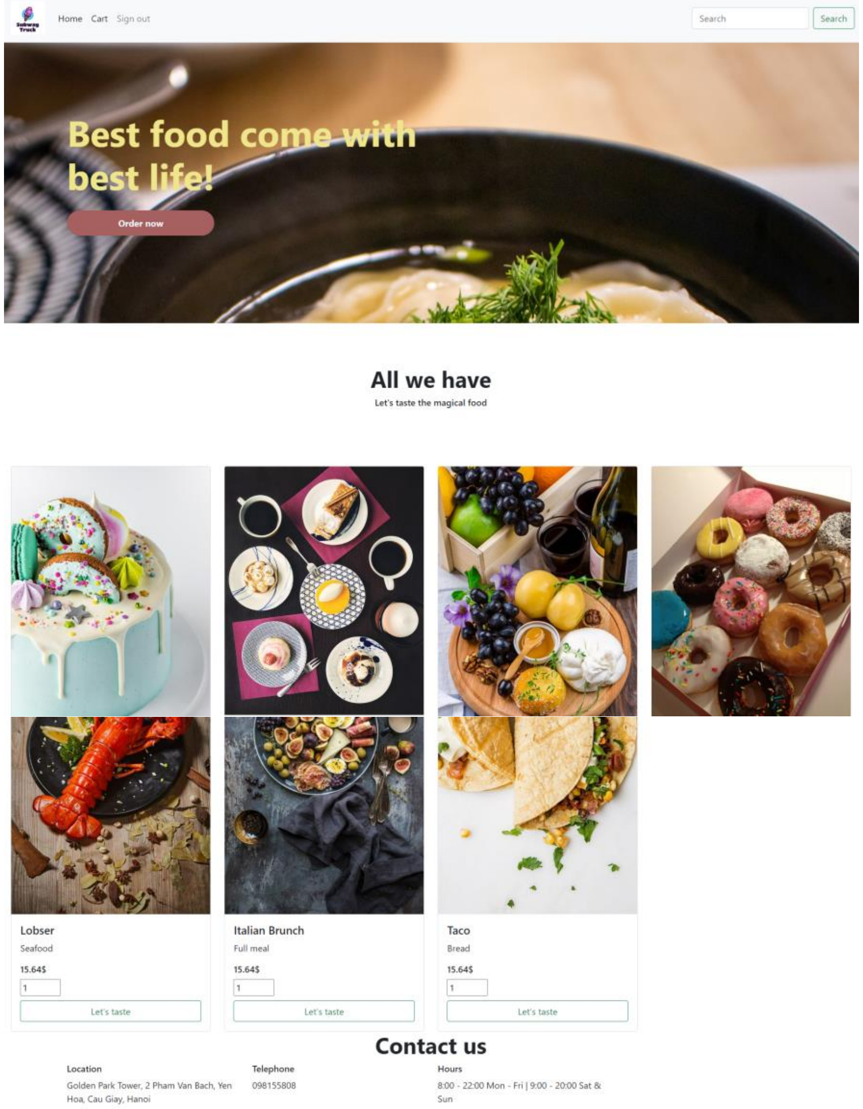
Prototype	Actual result	Side
	 User, don't have an account? Sign up now	User

 sign-up https://www.subwayfood.co Sign Up Full name Telephone Email Password Address Old User SIGN IN SIGN UP	 Subway Truck | Staff Only | Sign up for new staff Back Subway Truck Full Name Telephone: 0********* Email: abc@gmail.com Password Address Sign up	Staff

2.3. User home

Prototype	Actual result





Best food come with best life!

All we have

Let's taste the magical food

Lobser
Seafood
15.64$
1
Let's taste

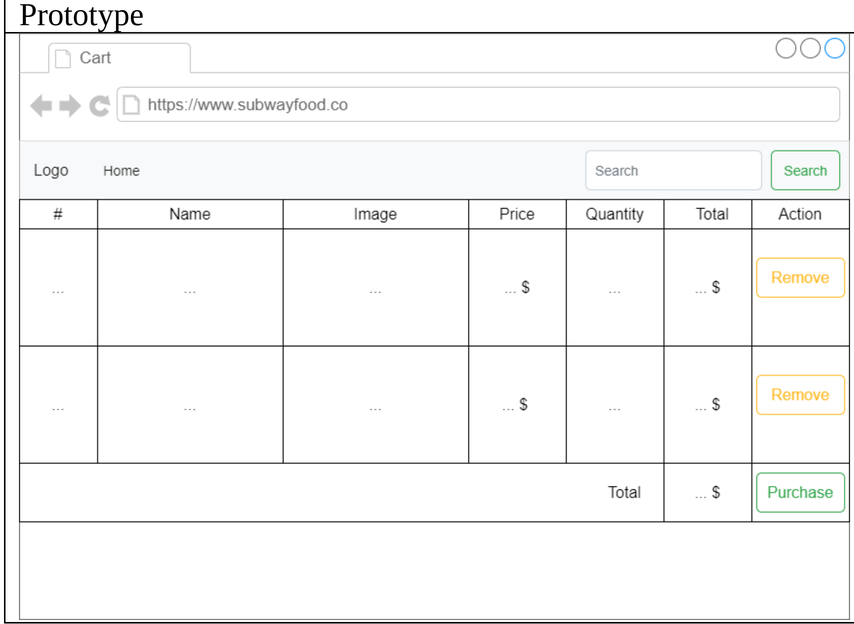
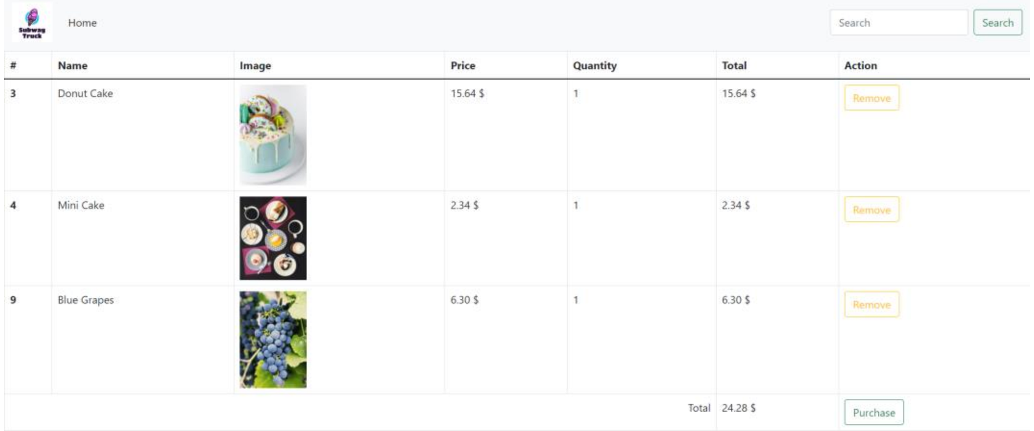
Italian Brunch
Full meal
15.64$
1
Let's taste

Taco
Bread
15.64$
1
Let's taste

Contact us

Location	Telephone	Hours
Golden Park Tower, 2 Pham Van Bach, Yen Hoa, Cau Giay, Hanoi	098155808	8:00 - 22:00 Mon - Fri | 9:00 - 20:00 Sat & Sun

2.4. Shopping cart

Prototype	Actual result
	

Cart

https://www.subwayfood.co

Logo Home Search Search

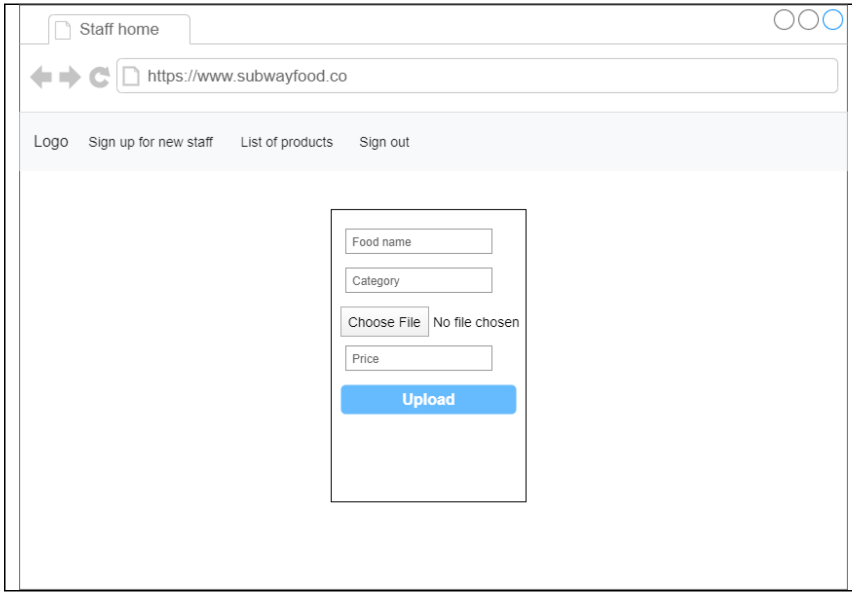
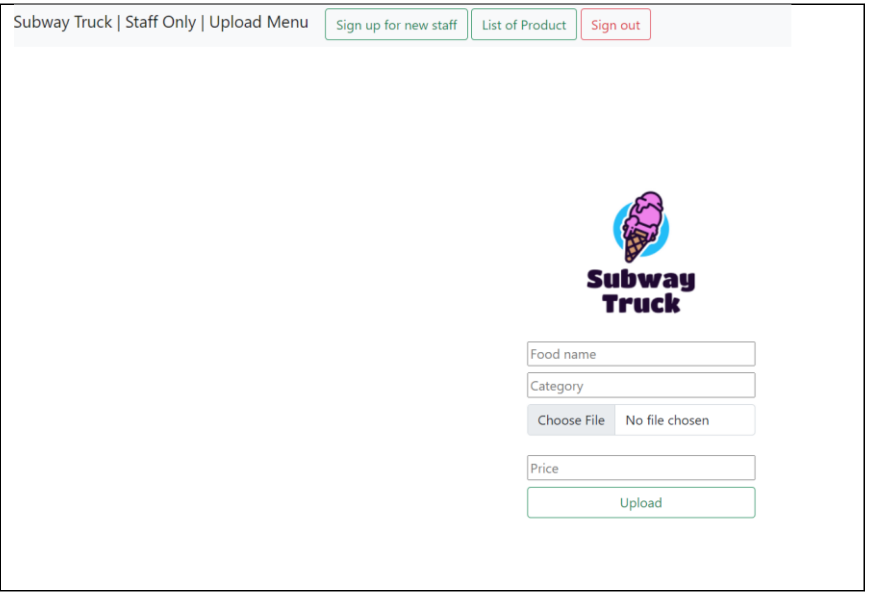
#	Name	Image	Price	Quantity	Total	Action
...	...	...	... $	...	... $	Remove
...	...	...	... $	...	... $	Remove
				Total	... $	Purchase

Home Search Search

#	Name	Image	Price	Quantity	Total	Action
3	Donut Cake		15.64 $	1	15.64 $	Remove
4	Mini Cake		2.34 $	1	2.34 $	Remove
9	Blue Grapes		6.30 $	1	6.30 $	Remove
				Total	24.28 $	Purchase

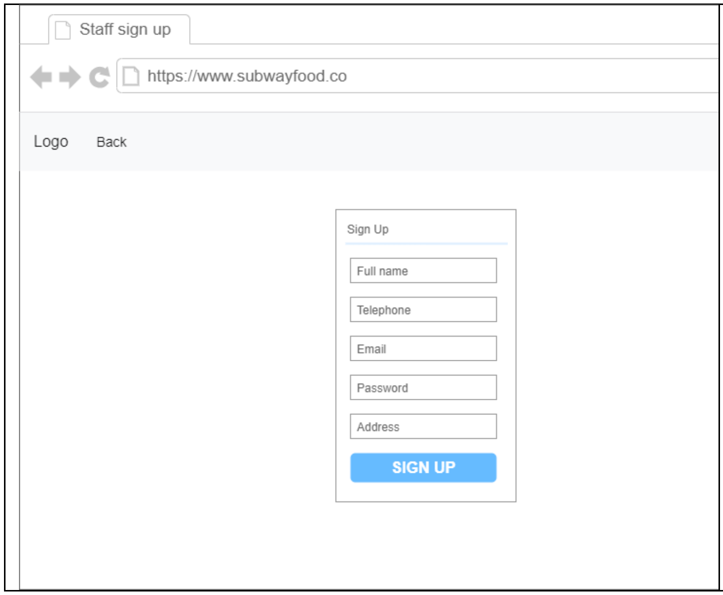
2.5. Staff home

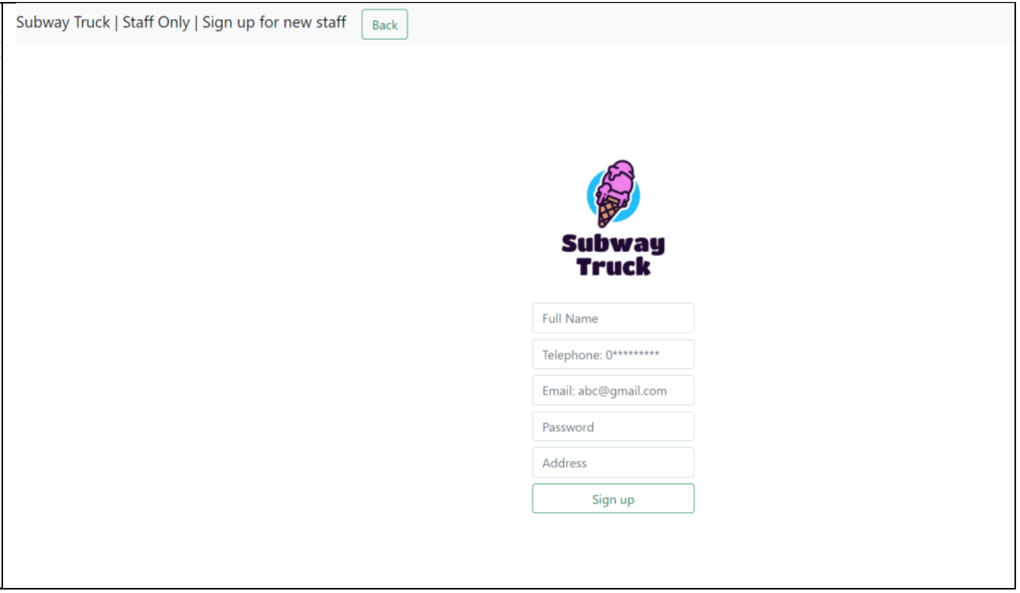
Prototype	Actual result

 Staff home https://www.subwayfood.co Logo Sign up for new staff List of products Sign out Food name Category Choose File No file chosen Price Upload	 Subway Truck | Staff Only | Upload Menu Sign up for new staff List of Product Sign out Subway Truck Food name Category Choose File No file chosen Price Upload

2.6. Staff sign-up

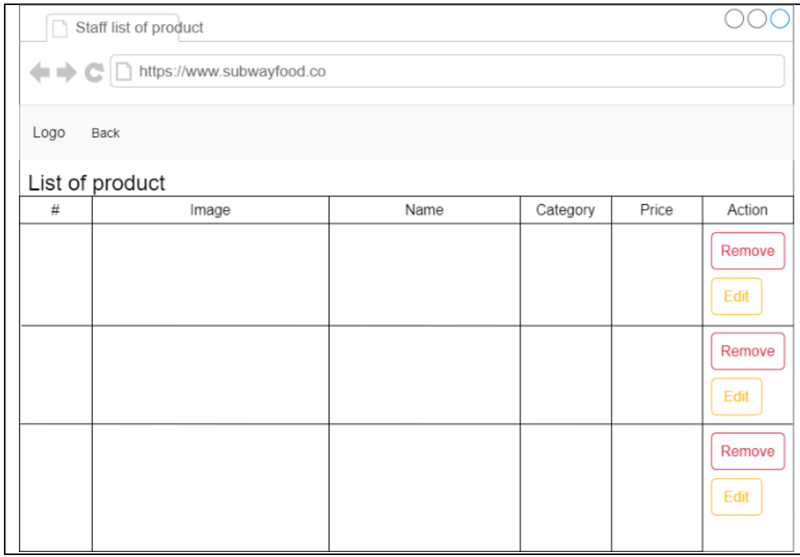
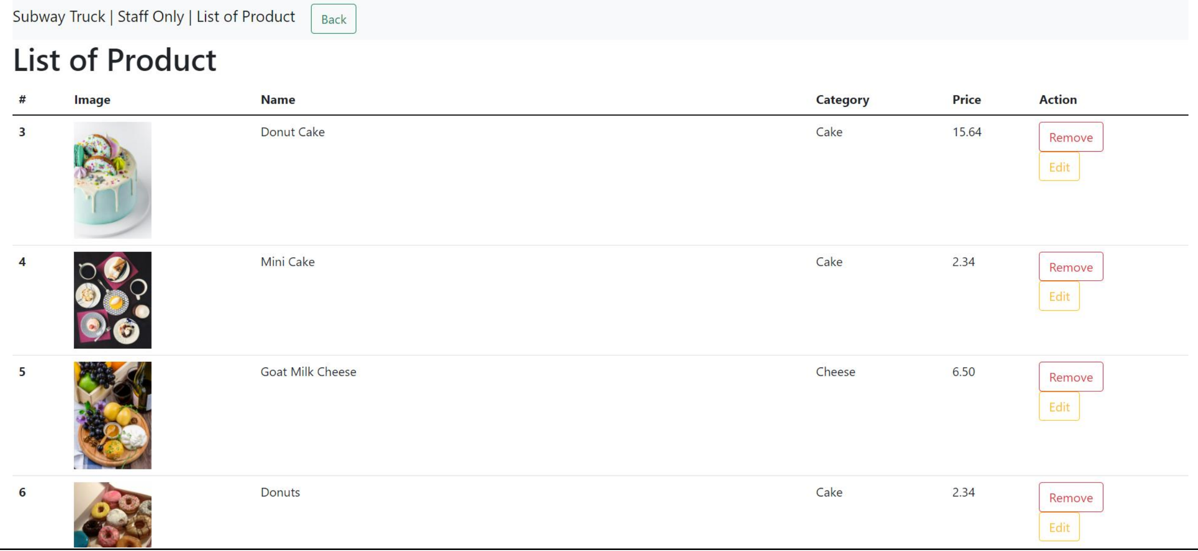
Prototype	Actual result





2.7. List of products

Prototype	Actual result

Staff list of product

https://www.subwayfood.co

Logo Back

List of product

#	Image	Name	Category	Price	Action
					Remove Edit
					Remove Edit
					Remove Edit

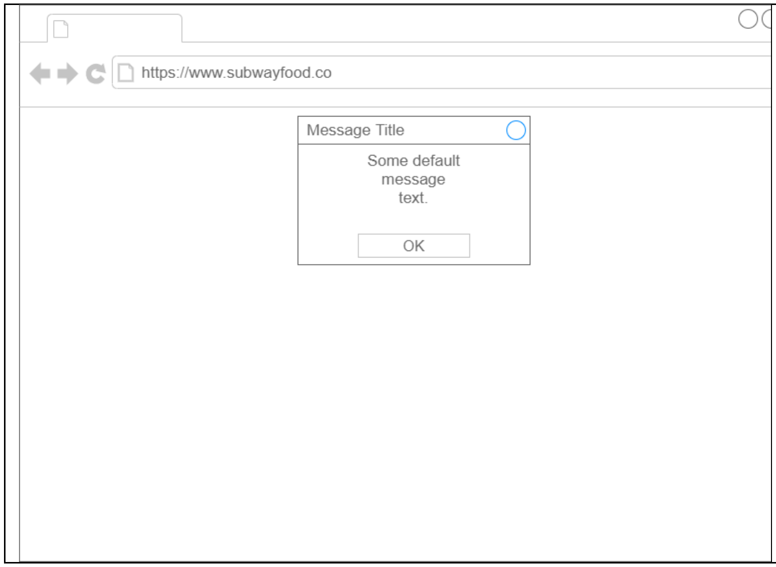
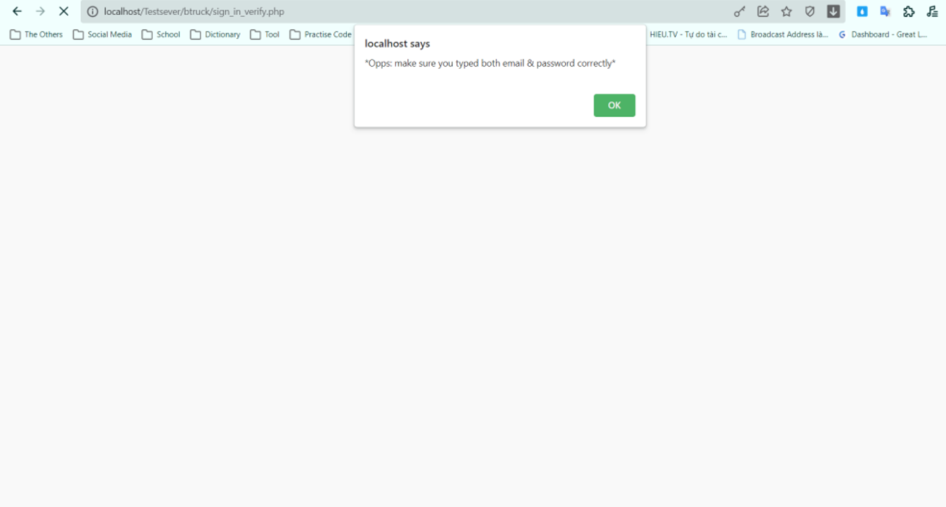
Subway Truck | Staff Only | List of Product Back

List of Product

#	Image	Name	Category	Price	Action
3		Donut Cake	Cake	15.64	Remove Edit
4		Mini Cake	Cake	2.34	Remove Edit
5		Goat Milk Cheese	Cheese	6.50	Remove Edit
6		Donuts	Cake	2.34	Remove Edit

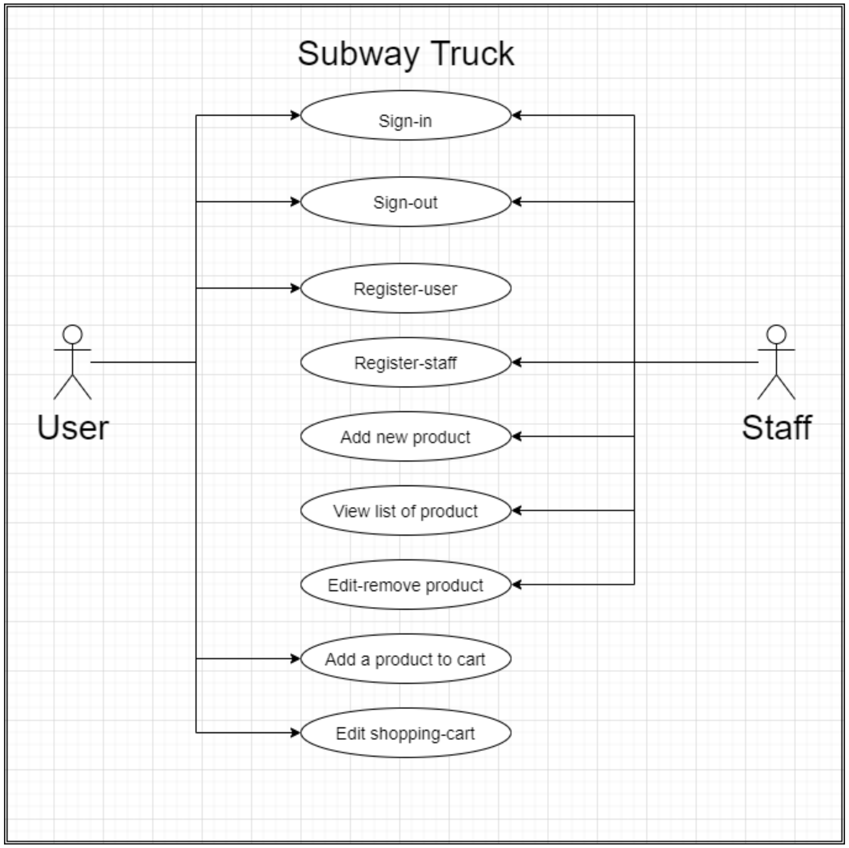
2.8. Alert

Prototype	Actual result

 https://www.subwayfood.co Message Title Some default message text. OK	 localhost/Testsever/btruck/sign_in_verify.php localhost says *Opps: make sure you typed both email & password correctly* OK

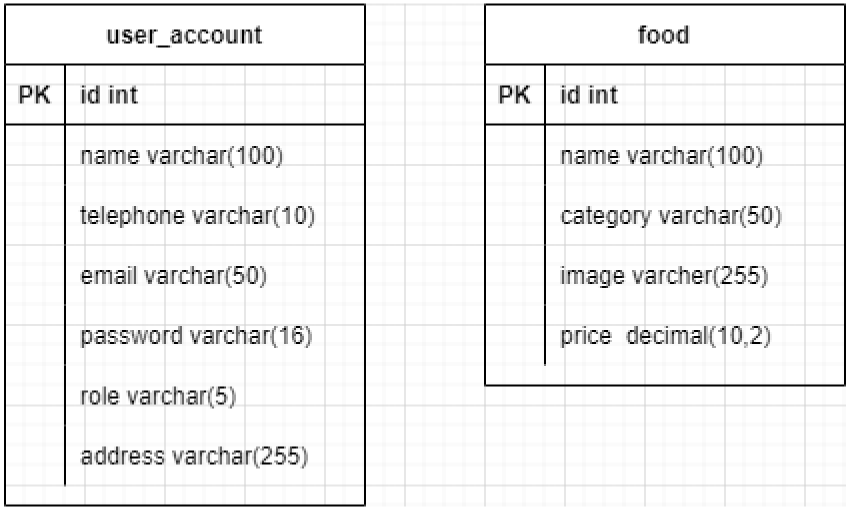
3. Design

3.1. Use-case diagram



3.2. ERD

This is the entity diagram of the website's database. It includes 2 tables to save the accounts of employees as well as guests, and store information about dishes:

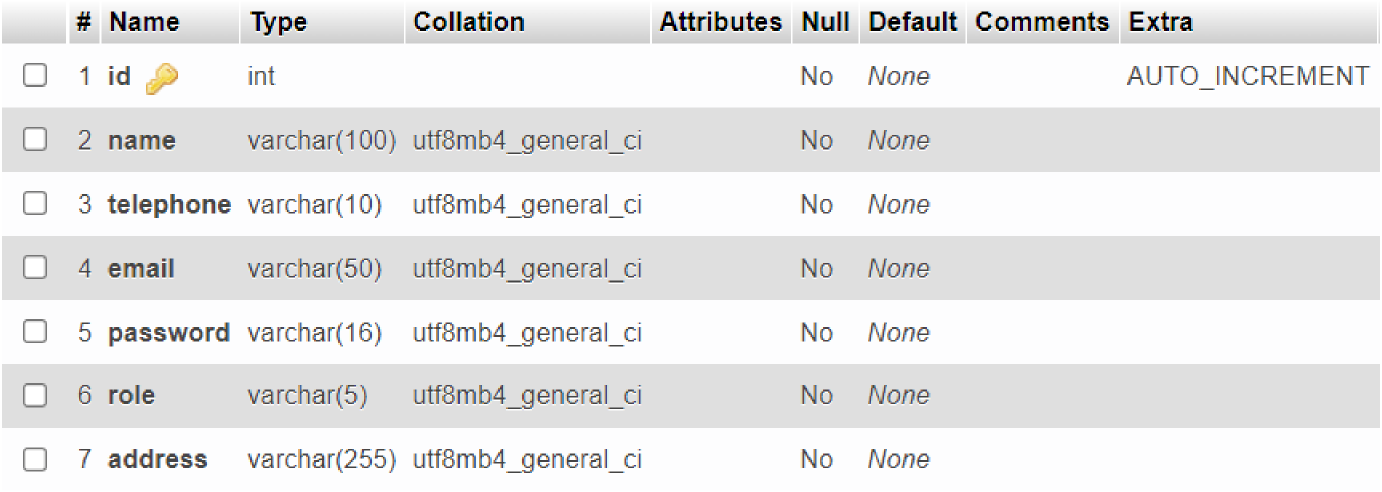


user_account	
PK	id int
	name varchar(100)
	telephone varchar(10)
	email varchar(50)
	password varchar(16)
	role varchar(5)
	address varchar(255)

food	
PK	id int
	name varchar(100)
	category varchar(50)
	image varcher(255)
	price decimal(10,2)

When the user successfully registers an account, the account information will be saved in the user_account table. This table includes fields such as:

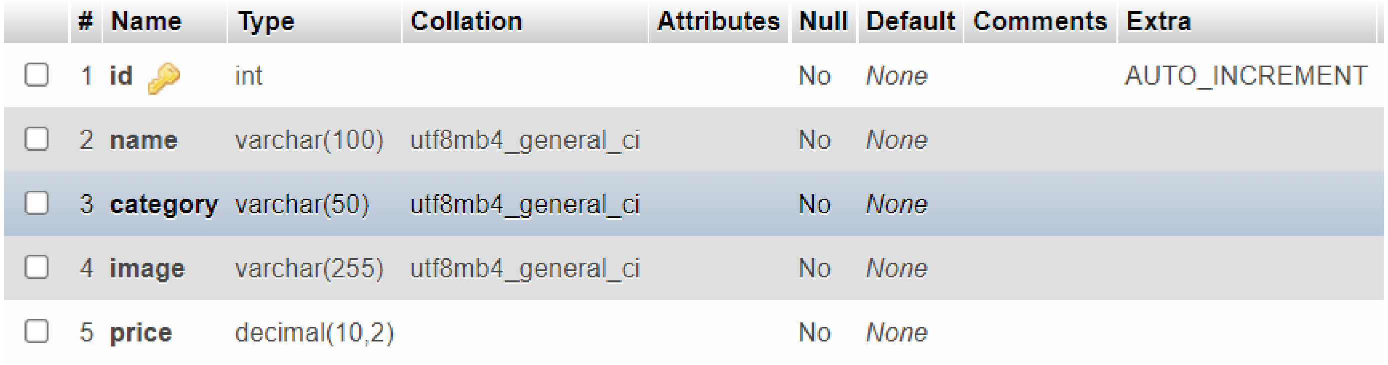
- id: it houses the id of each person, this field will automatically increase every time a new account is registered.
- name: save the name of the account owner.
- telephone: stores the account owner's phone number
- email: stores the account owner's email.
- password: stores the password
- role: stores the role of each user (2 stages: user or staff)
- address: Stores the address



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**id** 🔑	int			No	*None*		AUTO_INCREMENT
□	2	**name**	varchar(100)	utf8mb4_general_ci		No	*None*		
□	3	**telephone**	varchar(10)	utf8mb4_general_ci		No	*None*		
□	4	**email**	varchar(50)	utf8mb4_general_ci		No	*None*		
□	5	**password**	varchar(16)	utf8mb4_general_ci		No	*None*		
□	6	**role**	varchar(5)	utf8mb4_general_ci		No	*None*		
□	7	**address**	varchar(255)	utf8mb4_general_ci		No	*None*		

When adding a new dish to the list, the staff will have to add all the information for that dish and enter it correctly, then the dish will be saved in the food table of the database.

- id: it houses the id of each dish, this field will automatically increase every time a new dish is added.
- name: holds the dish’ name.
- category: stores the types of dishes.
- image: holds the address of the image.
- price: stores the price of each dish.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**id** 🔑	int			No	*None*		AUTO_INCREMENT
□	2	**name**	varchar(100)	utf8mb4_general_ci		No	*None*		
□	3	**category**	varchar(50)	utf8mb4_general_ci		No	*None*		
□	4	**image**	varchar(255)	utf8mb4_general_ci		No	*None*		
□	5	**price**	decimal(10,2)			No	*None*		

X. Evaluate the Quality Assurance process and review how it was implemented during your design and development stages

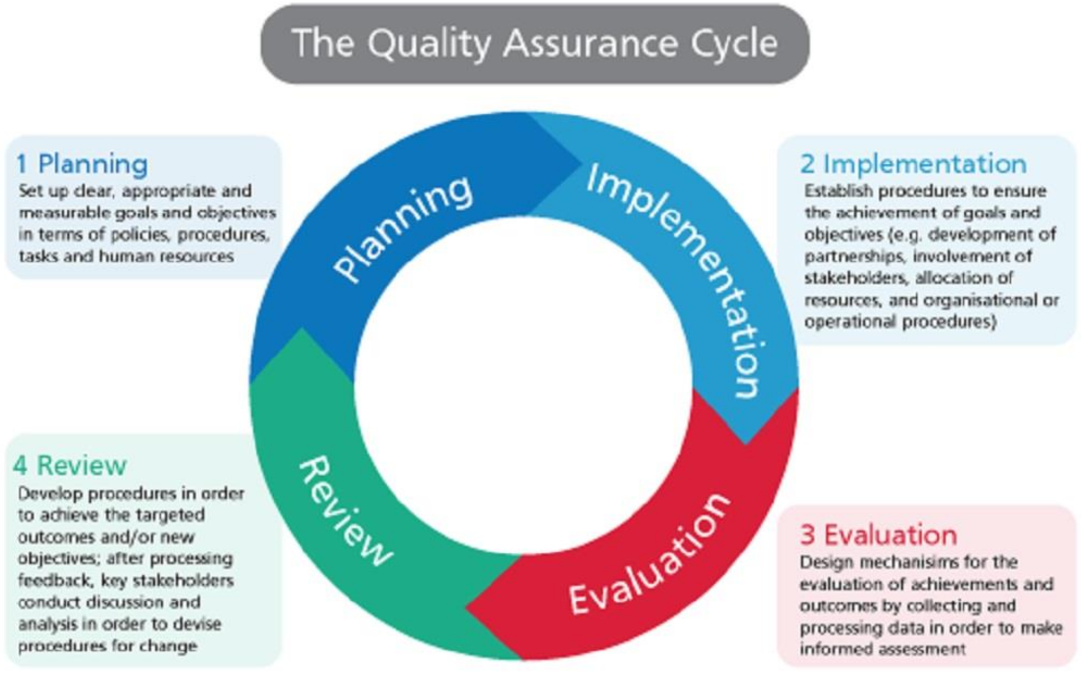
1. Quality Assurance

According to (Hamilton, 2022), Quality Assurance in Software Testing is defined as a procedure used to ensure the quality of software products or services provided to customers by a company. Quality assurance is concerned with making the software development process more efficient and effective in accordance with the quality standards established for software products. QA Testing is another term for Quality Assurance.

Four steps of Quality Assurance process are:

- Plant: People must plan and establish process-related objectives, as well as identify the processes needed to deliver a high-quality end product.
- Do: Development and testing of Processes and also “do” changes in the processes
- Check: Monitoring processes, modifying processes, and determining whether they meet predetermined goals

- Act: A Quality Assurance tester should carry out the actions required to improve the processes.



1 Planning

Set up clear, appropriate and measurable goals and objectives in terms of policies, procedures, tasks and human resources

2 Implementation

Establish procedures to ensure the achievement of goals and objectives (e.g. development of partnerships, involvement of stakeholders, allocation of resources, and organisational or operational procedures)

3 Evaluation

Design mechanisms for the evaluation of achievements and outcomes by collecting and processing data in order to make informed assessment

4 Review

Develop procedures in order to achieve the targeted outcomes and/or new objectives; after processing feedback, key stakeholders conduct discussion and analysis in order to devise procedures for change

2. Importance of QA

(Anon., 2022) states that there are 7 reasons of why QA is crucial:

- Saves Money and Effort: A flawed application can cost clients and users a lot of money. As a result, developers must make amends or risk losing their clients' trust. They can identify any potential bugs or issues that may arise in the future through a thorough QA testing process. This will save them a significant amount of time and effort once they have deployed.
- Prevents Emergencies: The stakes would be high with so many software companies. A simple bug in corporate software could cause system blackouts, communication breakdowns, or even data loss. So, if developers intend to use software throughout a company or deal with sensitive data, make sure to implement quality assurance testing to ensure that there is no room for error.

- Boosts Client’s Confidence: By focusing on QA testing, developers communicate to their clients that they want their application to run smoothly and without errors. This is especially important when it comes to building long-term working relationships and increasing customer loyalty.
- Enhances User Experience: It is self-evident that user experience can be a deciding factor in an IT product's success or failure. If a company's software is slow or prone to errors, its customers or users may become dissatisfied and switch to a competitor's product. As a result, it is critical that experienced employees meticulously test their product to ensure that the user will run it smoothly in their daily job or task.
- Creates More Profit: If developers are creating an application to sell or market, one of the most important factors that determines whether they can sell it at a higher price is the quality assurance process. Nothing is worse than disgruntled customers who paid for a product that did not perform as advertised.
- Improves Customer Satisfaction: Aside from profits, quality assurance can improve customer satisfaction, thereby enhancing the company's reputation. A satisfied customer will tell their friends or family about the product, which helps the company expand its client base without spending too much money on marketing.
- Promotes Efficiency and Productivity: A faulty piece of software can result in hurried fixes or frantic communication, which can exacerbate the situation. Obviously, everyone can work better when they don't have to deal with constant errors that are time-consuming and difficult to correct. Organizing quality assurance testing from the start of the project will allow the company to run more smoothly and productively.

3. QA in the project

3.1. Plan

After receiving and reviewing user requests, I organize them into tasks that I need to complete. I started by analyzing and categorizing them based on my own work experience. I drew a shift diagram to model the things I needed to do and what I needed to do to get them done. According to the development of the user object. I will check out the individuals and partnerships that will be on the site. Then, for the site, I create an appropriate entity relationship diagram.

Then I build wireframe websites based on the requirements of the user's role in the site as well as the design.

And to make sure that the information I collect is relevant to the needs of the client. I had to tell them what we were going to do.

3.2. Do

After planning what needs to be done for the project. The author has started implementing the project in the order originally set out. During his work, he also relied on work experience and personal experience to help the website after completion will have better quality. Although he knows that it is impossible to make a perfect product from the beginning, so he still tries to make the end result of the website the best possible quality.

3.3. Check

In the process of developing any project, even if it is not a software development, errors are inevitable. Every time I run a website and encounter an error, the author always carefully checks the source of the error by applying his problem-solving skills. Once he has identified the problem causing the error, he will go back to the Do step to work on fixing the error until the error is completely gone and the site is running smoothly.

3.4. Act

After the above steps, especially testing to ensure the website works smoothly, the author has also proposed a report on the future development of the website such as what additional functions are needed, what functions need to be added further improvements, to contribute to website development

XI. Conclutiom

To sum up, throughout this report the author has presented design principles, standards and guidelines for developing a website. He also presented the design of the website he built for the company subwayfood.co. He also provides hands-on site shots, source code, and explains how the site works and user permissions. To prove the quality of his website, the author also wrote and performed a round of website quality testing by executing test cases based on user requirements. Based on the results, the website works very smoothly and stably. The author also mentioned QA and pointed out how important it is in making a project and how he applies it to the project. Therefore, from his personal judgment, he deserves a M grade.

References

Anon., 2018. *pixselo.com.* [Online]
Available at: https://pixselo.com/top-10-web-design-standards-for-website/
[Accessed 30 Aoril 2022].

Anon., 2020. *gapsystudio.com.* [Online]
Available at: https://gapsystudio.com/blog/web-design-standards/
[Accessed 30 april 2022].

Anon., 2022. *www.3pillarglobal.com.* [Online]
Available at: https://www.3pillarglobal.com/insights/why-is-quality-assurance-so-important/#:~:text=Quality%20Assurance%20Saves%20You%20Money,might%20occur%20in%20the%20future.
[Accessed 02 may 2022].

Hamilton, T., 2022. *www.guru99.com.* [Online]
Available at: https://www.guru99.com/all-about-quality-assurance.html
[Accessed 02 May 2022].

Marianne, 2021. *www.feelingpeaky.com.* [Online]
Available at: https://www.feelingpeaky.com/9-principles-of-good-web-design/
[Accessed 30 April 2022].