Experiment No. 7

Title: REST APIs IN PHP

Batch: B-2 **Roll No.: 16010422234** **Experiment No.: 7**

Aim: REST APIs IN PHP

Resources needed: Windows OS, Web Browser, Editor, XAMPP Server

---

Pre Lab/ Prior Concepts:
Students should have prior knowledge of HTML/CSS/Basic Programming, web services.

Theory:

HTTP is designed considering the REST (Representational State Transfer) architecture which defines how to develop and consume the web service. REST API provides easy to implement without less complexity and stateless web service. SOAP (Simple Object Access Protocol architecture) based communication is another approach web services can adopt.

REST API supports multiple data formats such as Command Separated Value (CSV), JavaScript Object Notation (JSON), Extensible Markup Language (XML) as compared to SOAP [1][2].

Many REST APIs use JSON (or JavaScript Object Notation) to carry responses from REST API endpoints. PHP natively supports converting data to JSON format from PHP variables and vice versa through its json extension [1].

To get a JSON representation of a PHP variable, use json_encode():
$data = array(1, 2, "three");
$jsonData = json_encode($data); echo $jsonData;
Result will be : [1, 2, "three"]

Similarly, if you have a string containing JSON data, you can turn it into a PHP variable using json_decode():
<?php
$jsonData = "[1, \"KJSCE\", 2001]";
$data = json_decode($jsonData); print_r($data);
?>
Result will be:
Array(
[0] => 1
[1] => KJSCE [2] => 2001

)

There is no direct translation between PHP objects and JSON objects, these objects are an associative array.

Retrieving resources:
To retrieve the resource, GET method of HTTP is used. It uses the **curl** extension to format an HTTP request, set parameters on it, send the request, and get the returned information.
To retrieve information about the resource, first construct a URL representing the endpoint for that resource; it initializes a curl resource and provides the constructed URL to it. Finally, the curl object is executed, which sends the HTTP request, waits for the response, and returns it as demonstrated in the below code snippet.

```
<?php
$empId = '1';
$url = "http://localhost/test/rest1.php/{$empId}";
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
$response = curl_exec($ch);
$resultInfo = curl_getinfo($ch); curl_close($ch);
// decode the JSON
$empJson = json_decode($response); Print_r($empJson);
?>
```

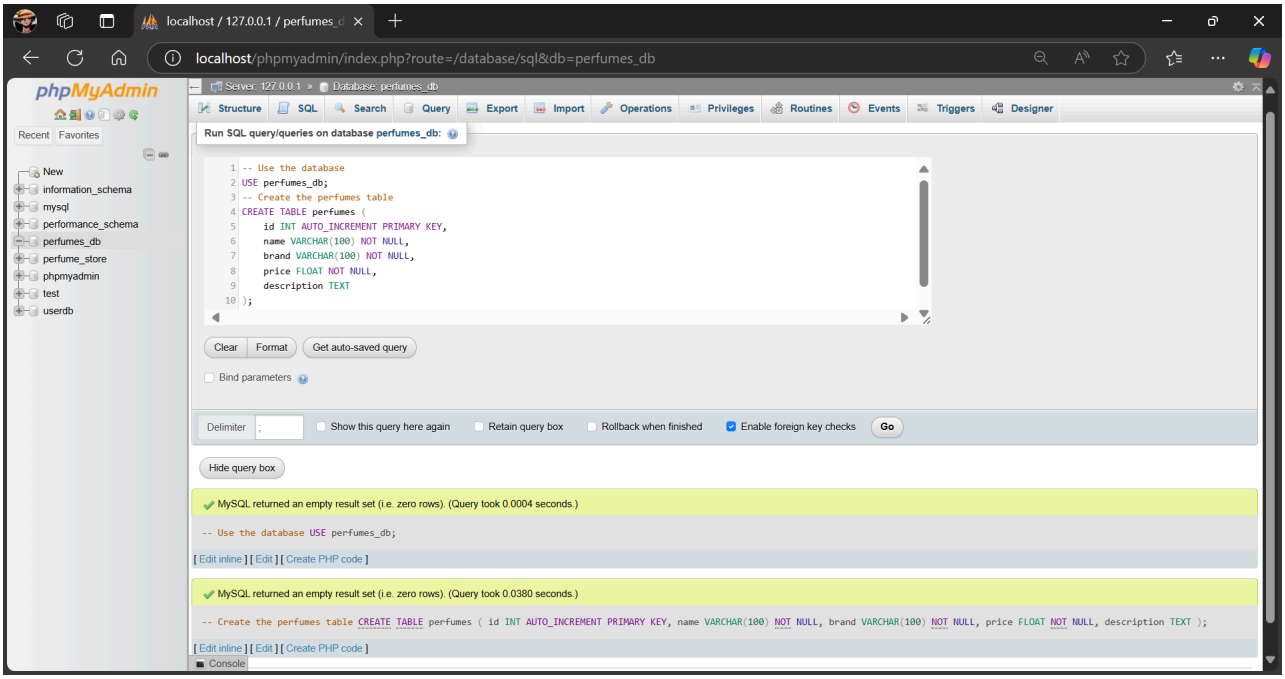
In this case, the response is JSON data, which is decoded and handed off for future processing. The PUT method of HTTP is used to update the resource and DELETE method of HTTP is used to delete the resource.

Activity:

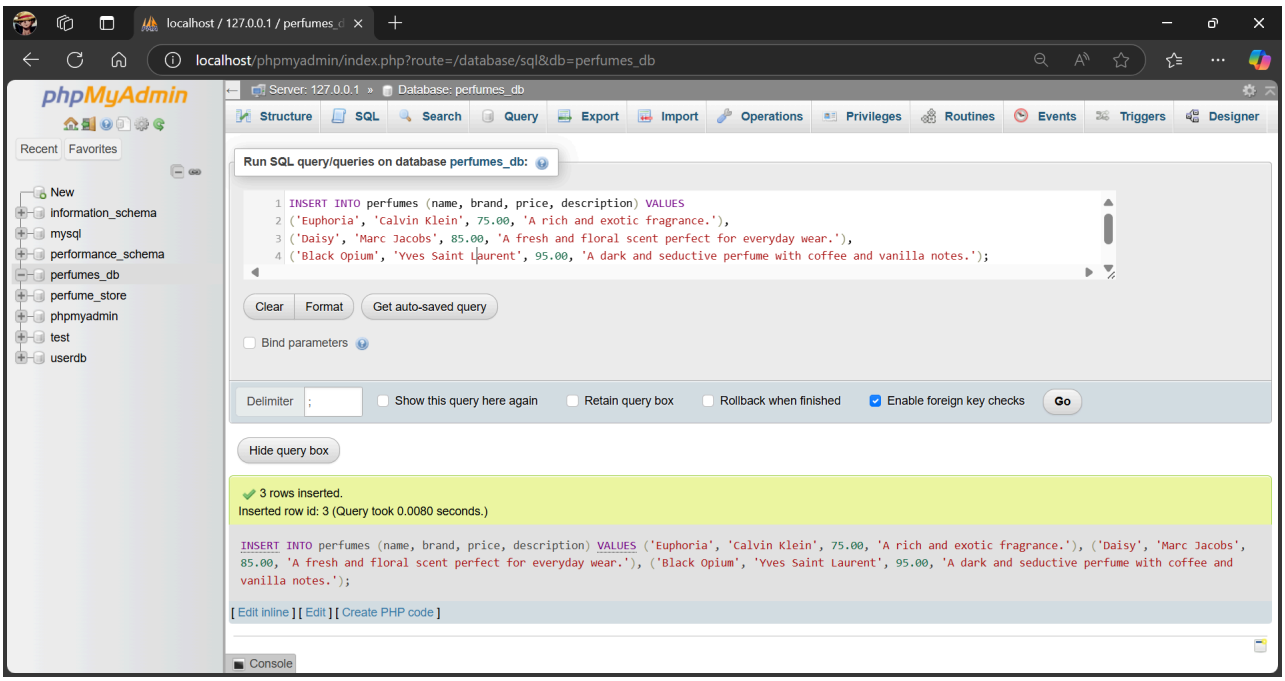
- Implement REST API to retrieve information of a resource
- Accept the required resource details from the user and using REST API retrieve information of the required resource.

Output: (Code with result snapshot)

```
-- Use the database
USE perfumes_db;
-- Create the perfumes table
CREATE TABLE perfumes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    brand VARCHAR(100) NOT NULL,
    price FLOAT NOT NULL,
    description TEXT
);
```




```
INSERT INTO perfumes (name, brand, price, description) VALUES
('Euphoria', 'Calvin Klein', 75.00, 'A rich and exotic fragrance.'),
('Daisy', 'Marc Jacobs', 85.00, 'A fresh and floral scent perfect for everyday
wear.'),
('Black Opium', 'Yves Saint Laurent', 95.00, 'A dark and seductive perfume with
coffee and vanilla notes.');
```



home.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Perfume Paradise</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Our Perfume Collection</h1>

    <!-- Form to Add/Edit Perfume -->
    <div id="create-form">
        <h2 id="form-title">Add a New Perfume</h2>
        <form id="perfumeForm">
            <input type="hidden" id="perfumeId">
            <input type="text" id="name" placeholder="Name" required>
            <input type="text" id="brand" placeholder="Brand" required>
            <input type="number" id="price" placeholder="Price" required>
            <textarea id="description" placeholder="Description"
required></textarea>
            <button type="button" onclick="handleSave()">Add Perfume</button>
        </form>
```

```
    </div>

    <!-- Container to Display Perfume Collection -->
    <div id="perfumes-container" class="perfume-container"></div>

    <script>
        // Fetch and display perfumes
        function fetchPerfumes() {
            fetch('/WP-II/EXP7/api.php/perfumes')
                .then(response => response.json())
                .then(data => {
                    const container =
document.getElementById('perfumes-container');
                    container.innerHTML = '';  // Clear previous content

                    data.forEach(perfume => {
                        const item = document.createElement('div');
                        item.className = 'perfume-card';
                        item.innerHTML = `
                            <h2>${perfume.name}</h2>
                            <p><strong>Brand:</strong> ${perfume.brand}</p>
                            <p><strong>Price:</strong> $${perfume.price}</p>
                            <p><strong>Description:</strong>
${perfume.description}</p>
                            <button
onclick="deletePerfume(${perfume.id})">Delete</button>
                            <button onclick="editPerfume(${perfume.id},
'${perfume.name}', '${perfume.brand}', ${perfume.price},
'${perfume.description}')">Edit</button>
                        `;
                        container.appendChild(item);
                    });
                })
                .catch(error => console.error('Error fetching data:',
error));
        }

        // Function to handle saving (Add or Update based on ID presence)
        function handleSave() {
            const id = document.getElementById('perfumeId').value;
            if (id) {
                updatePerfume(id); // Update if ID exists
```

```
            } else {
                createPerfume(); // Add new if ID is empty
            }
        }

        // CREATE: Add a new perfume
        function createPerfume() {
            const name = document.getElementById('name').value;
            const brand = document.getElementById('brand').value;
            const price = document.getElementById('price').value;
            const description = document.getElementById('description').value;

            fetch('/WP-II/EXP7/api.php/perfumes', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({ name, brand, price, description })
            })
            .then(response => response.json())
            .then(data => {
                alert(data.message);
                resetForm();
                fetchPerfumes(); // Refresh the list
            })
            .catch(error => console.error('Error adding perfume:', error));
        }

        // DELETE: Remove a perfume
        function deletePerfume(id) {
            if (confirm("Are you sure you want to delete this perfume?")) {
                fetch(`/WP-II/EXP7/api.php/perfumes/${id}`, {
                    method: 'DELETE'
                })
                .then(response => response.json())
                .then(data => {
                    alert(data.message);
                    fetchPerfumes(); // Refresh the list
                })
                .catch(error => console.error('Error deleting perfume:',
error));
            }
```

```
        }

        // EDIT: Populate form with perfume data for editing
        function editPerfume(id, name, brand, price, description) {
            document.getElementById('perfumeId').value = id;
            document.getElementById('name').value = name;
            document.getElementById('brand').value = brand;
            document.getElementById('price').value = price;
            document.getElementById('description').value = description;

            document.getElementById('form-title').textContent = "Edit
Perfume";
            document.querySelector('#perfumeForm button').textContent =
"Update Perfume";
        }

        // UPDATE: Update a perfume's details
        function updatePerfume(id) {
            const name = document.getElementById('name').value;
            const brand = document.getElementById('brand').value;
            const price = document.getElementById('price').value;
            const description = document.getElementById('description').value;

            fetch(`/WP-II/EXP7/api.php/perfumes/${id}`, {
                method: 'PUT',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({ name, brand, price, description })
            })
            .then(response => response.json())
            .then(data => {
                alert(data.message);
                resetForm();
                fetchPerfumes(); // Refresh the list
            })
            .catch(error => console.error('Error updating perfume:', error));
        }

        // Reset the form and switch back to "Add" mode
        function resetForm() {
            document.getElementById('perfumeId').value = '';
```

```
            document.getElementById('name').value = '';
            document.getElementById('brand').value = '';
            document.getElementById('price').value = '';
            document.getElementById('description').value = '';
            document.getElementById('form-title').textContent = "Add a New
Perfume";
            document.querySelector('#perfumeForm button').textContent = "Add
Perfume";
        }

        // Load perfumes on page load
        fetchPerfumes();
    </script>
</body>
</html>
```

style.css

```
/* Basic Reset */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Serif', 'Times New Roman', sans-serif;
}

/* Body Styling */
body {
    background-color: #f5f5f5; /* Light background for contrast */
    color: #333333; /* Dark grey for primary text */
    font-family: 'Georgia', 'Times New Roman', serif;
    padding: 20px;
}

/* Header */
h1 {
    color: #333333;
    font-size: 2.5em;
    text-align: center;
    margin-bottom: 30px;
    font-weight: bold;
}
```

```css
/* Container for perfume cards */
.perfume-container {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    justify-content: center;
    max-width: 1200px;
    margin: 0 auto;
}

/* Perfume Card Styling */
.perfume-card {
    background-color: #1f2b46; /* Dark navy-blue background */
    color: #f5f5f5; /* Light text color */
    border-radius: 10px;
    padding: 20px;
    width: 250px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.15);
    text-align: center;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.perfume-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 8px 15px rgba(0, 0, 0, 0.3);
}

.perfume-card h2 {
    font-size: 1.8em;
    color: #d4af37; /* Gold color for headings */
    margin-bottom: 10px;
    font-family: 'Garamond', serif;
}

.perfume-card p {
    font-size: 1em;
    margin: 8px 0;
    color: #d1d1d1; /* Soft grey for descriptive text */
}

/* Button Styling */
```

```
button {
    background-color: #d4af37; /* Gold background */
    color: #1f2b46; /* Dark navy-blue text color */
    padding: 10px 20px;
    font-size: 1em;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    margin-top: 10px;
}

button:hover {
    background-color: #b69329; /* Darker shade of gold */
}

/* Styling for Form Container */
#create-form {
    background-color: #1f2b46;
    color: #f5f5f5;
    padding: 20px;
    border-radius: 10px;
    margin: 0 auto 30px;
    width: 100%;
    max-width: 500px;
    text-align: center;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.15);
}

#create-form h2 {
    color: #d4af37;
    margin-bottom: 20px;
}

#create-form form {
    display: flex;
    flex-direction: column;
}

#create-form input[type="text"],
#create-form input[type="number"],
#create-form textarea {
```

```
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #ddd;
    border-radius: 5px;
    font-size: 1em;
    color: #333;
    width: 100%;
}

#create-form input[type="text"]:focus,
#create-form input[type="number"]:focus,
#create-form textarea:focus {
    outline: none;
    border: 1px solid #d4af37;
    box-shadow: 0 0 5px rgba(212, 175, 55, 0.5);
}

/* Styling for Action Buttons inside Perfume Cards */
.perfume-card button {
    width: 45%;
    margin: 5px 2%;
    padding: 8px;
}

.perfume-card button:first-of-type {
    background-color: #d9534f; /* Red for delete */
    color: #fff;
}

.perfume-card button:first-of-type:hover {
    background-color: #c9302c;
}

.perfume-card button:last-of-type {
    background-color: #5bc0de; /* Light blue for edit */
    color: #1f2b46;
}

.perfume-card button:last-of-type:hover {
    background-color: #31b0d5;
}
```

```css
/* Additional Layout Elements */
.main-content {
    background-color: #f5f5f5;
    padding: 50px 20px;
    text-align: center;
    color: #333;
}

.introduction {
    background-color: #d4af37; /* Gold background */
    color: #1f2b46; /* Dark navy text */
    padding: 20px;
    border-radius: 8px;
    margin-bottom: 20px;
    max-width: 800px;
    margin: 0 auto;
}

.introduction h3 {
    font-size: 1.5em;
    font-weight: bold;
    margin-bottom: 10px;
}

.introduction p {
    font-size: 1em;
    line-height: 1.6;
}

/* Footer */
footer {
    background-color: #1f2b46;
    color: #f5f5f5;
    padding: 20px;
    text-align: center;
    font-size: 0.9em;
    border-top: 2px solid #d4af37;
    margin-top: 30px;
}

/* Scroll to Top Button */
.scroll-to-top {
```

```
    position: fixed;
    bottom: 20px;
    right: 20px;
    background-color: #d4af37;
    color: #1f2b46;
    padding: 10px;
    border-radius: 50%;
    cursor: pointer;
    font-size: 1.2em;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

.scroll-to-top:hover {
    background-color: #b69329;
}
```

api.php

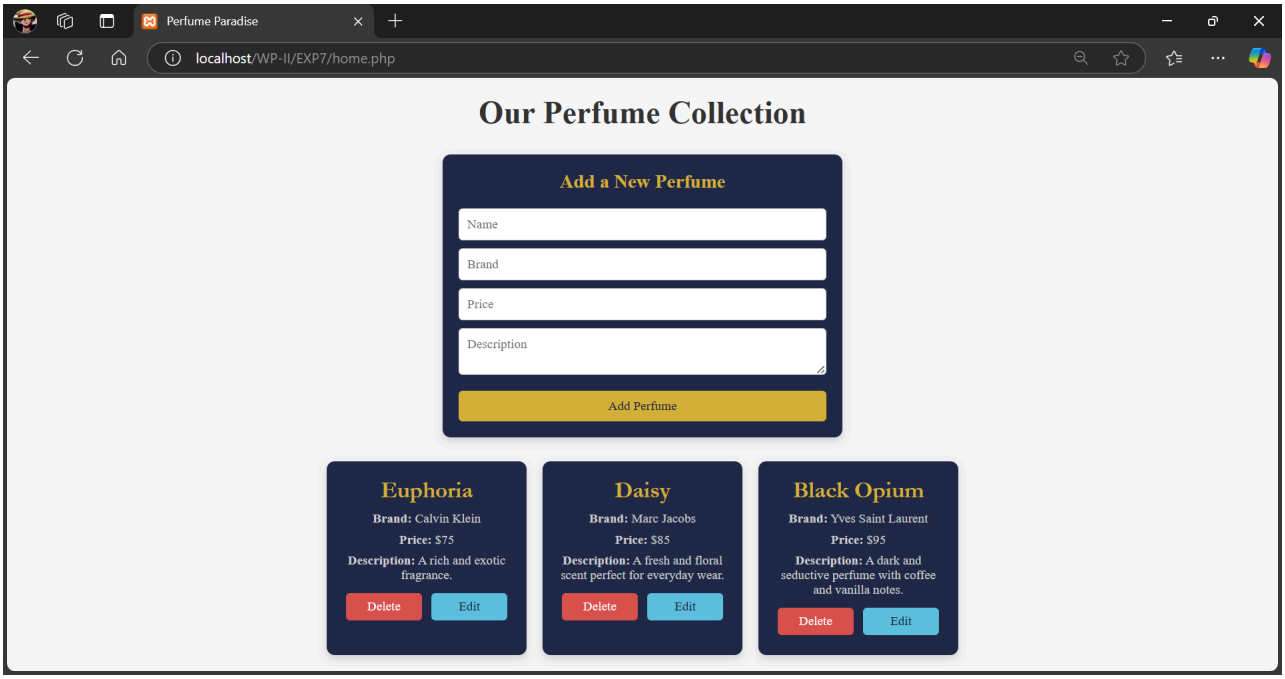
```
<?php
header("Content-Type: application/json");
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type,
Access-Control-Allow-Headers, Authorization, X-Requested-With");

$server = "localhost";
$username = "root";
$password = "";
$dbname = "perfumes_db";

// Connect to the database
$conn = new mysqli($server, $username, $password, $dbname);
if ($conn->connect_error) {
    die(json_encode(["error" => "Connection failed: " .
$conn->connect_error]));
}

// Helper function to get JSON input data
function getInput() {
    return json_decode(file_get_contents("php://input"), true);
}
```

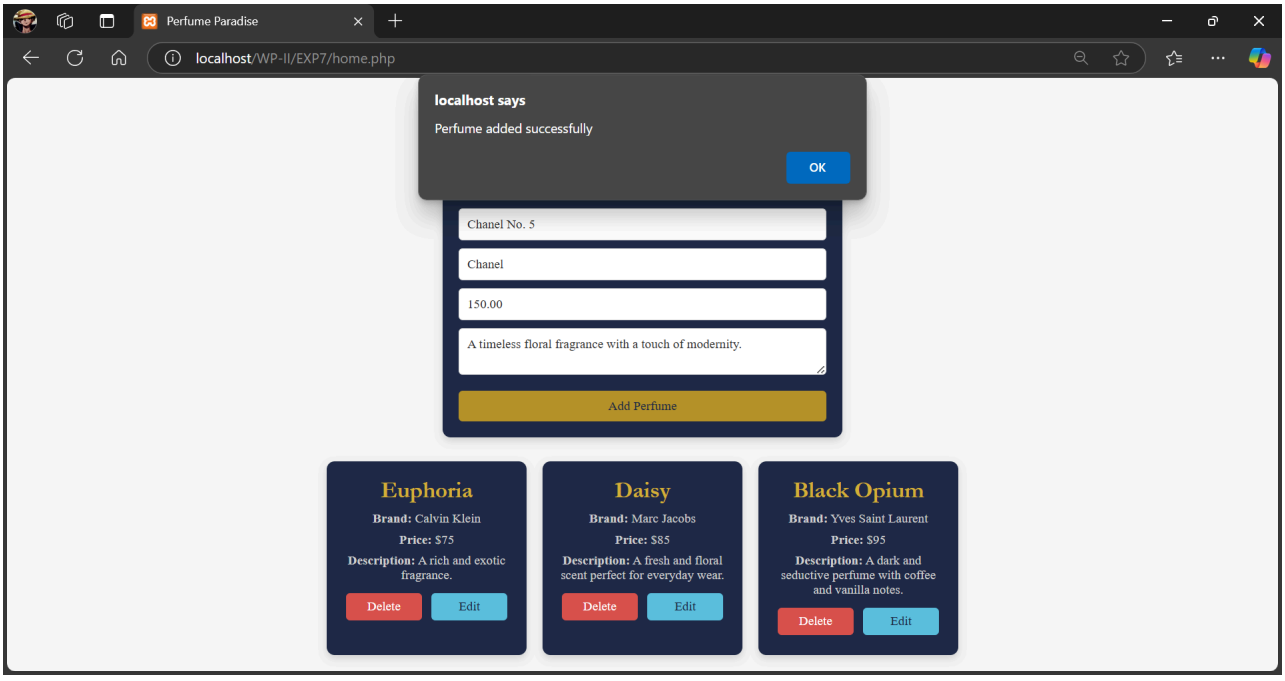
```php
$method = $_SERVER['REQUEST_METHOD'];
$path = explode('/', trim($_SERVER['PATH_INFO'], '/'));

// READ: Retrieve all perfumes or a specific perfume by ID
if ($method == 'GET' && $path[0] == 'perfumes') {
    $id = isset($path[1]) ? intval($path[1]) : 0;
    $sql = $id ? "SELECT * FROM perfumes WHERE id = $id" : "SELECT * FROM
perfumes";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        $perfumes = [];
        while($row = $result->fetch_assoc()) {
            $perfumes[] = $row;
        }
        echo json_encode($perfumes);
    } else {
        echo json_encode(["message" => "No perfumes found"]);
    }

// CREATE: Add a new perfume
} elseif ($method == 'POST' && $path[0] == 'perfumes') {
    $data = getInput();
    $name = $data['name'];
    $brand = $data['brand'];
    $price = $data['price'];
    $description = $data['description'];

    $sql = "INSERT INTO perfumes (name, brand, price, description) VALUES
('$name', '$brand', '$price', '$description')";
    if ($conn->query($sql) === TRUE) {
        echo json_encode(["message" => "Perfume added successfully"]);
    } else {
        echo json_encode(["error" => "Error: " . $conn->error]);
    }

// UPDATE: Update an existing perfume by ID
} elseif ($method == 'PUT' && $path[0] == 'perfumes' && isset($path[1])) {
    $data = getInput();
    $id = intval($path[1]);
    $name = $data['name'];
    $brand = $data['brand'];
```

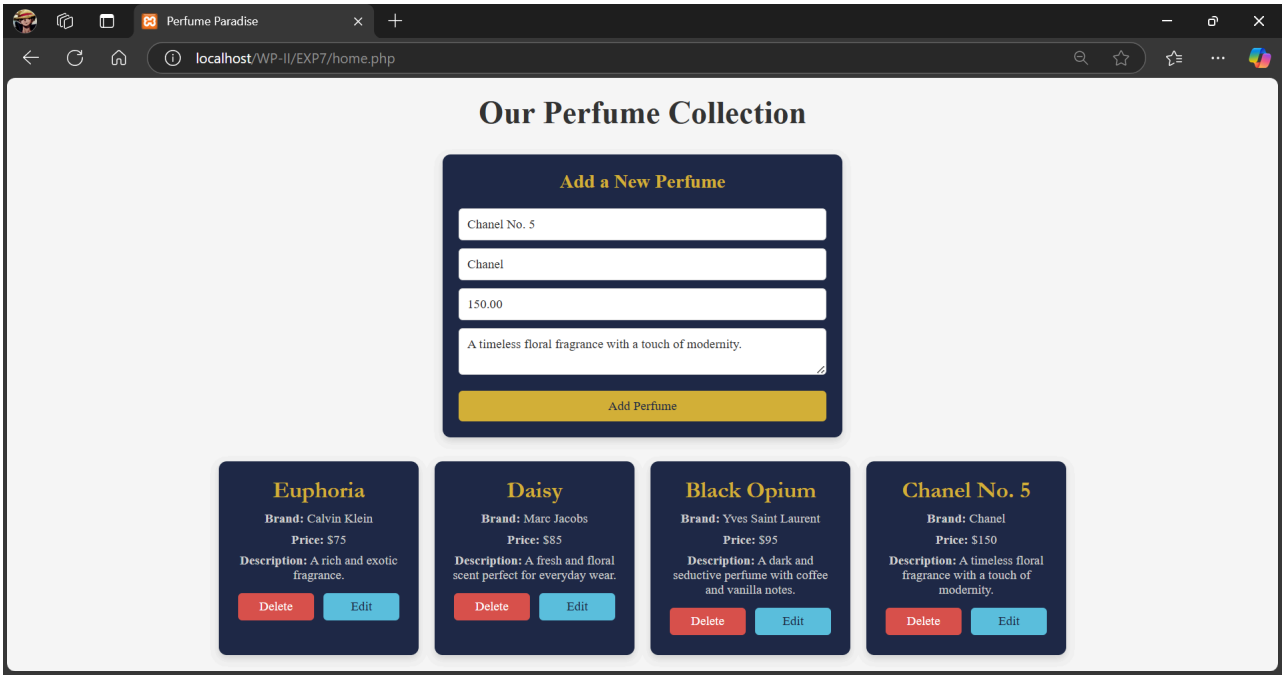
```
    $price = $data['price'];
    $description = $data['description'];

    $sql = "UPDATE perfumes SET name='$name', brand='$brand', price='$price',
description='$description' WHERE id=$id";
    if ($conn->query($sql) === TRUE) {
        echo json_encode(["message" => "Perfume updated successfully"]);
    } else {
        echo json_encode(["error" => "Error: " . $conn->error]);
    }

// DELETE: Delete a perfume by ID
} elseif ($method == 'DELETE' && $path[0] == 'perfumes' && isset($path[1])) {
    $id = intval($path[1]);
    $sql = "DELETE FROM perfumes WHERE id=$id";
    if ($conn->query($sql) === TRUE) {
        echo json_encode(["message" => "Perfume deleted successfully"]);
    } else {
        echo json_encode(["error" => "Error: " . $conn->error]);
    }

// Invalid request
} else {
    echo json_encode(["error" => "Invalid request"]);
}

$conn->close();
?>
```

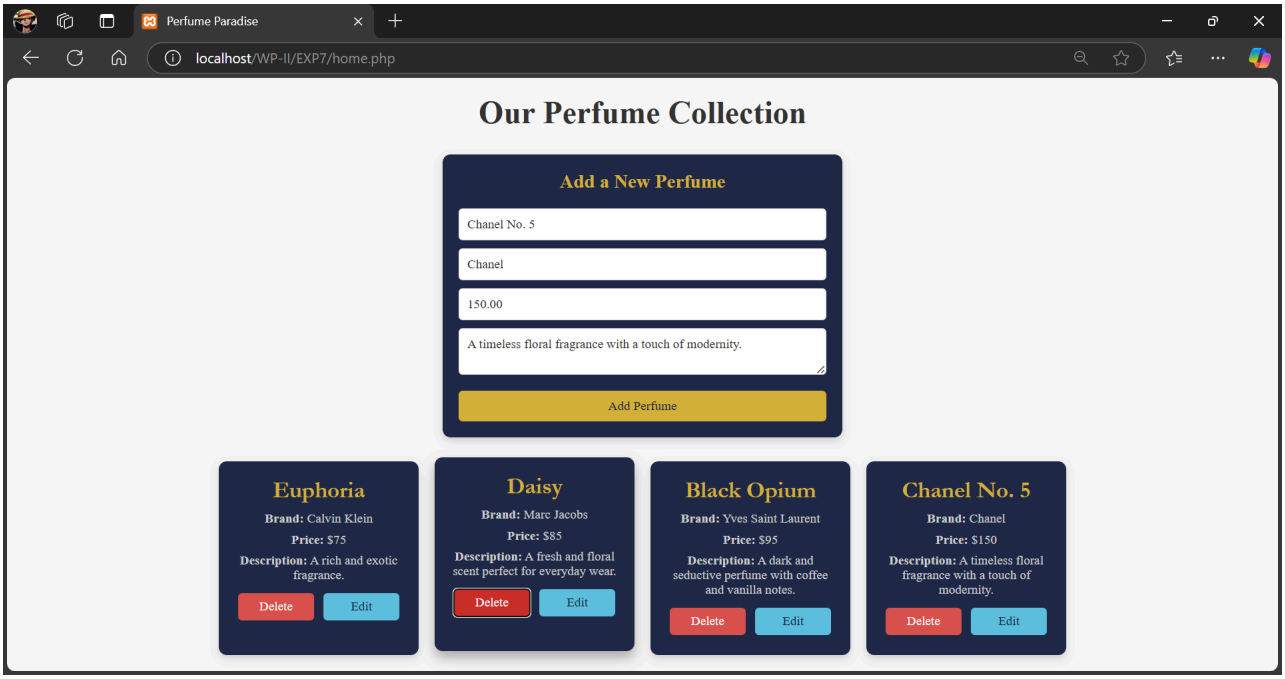
.htaccess

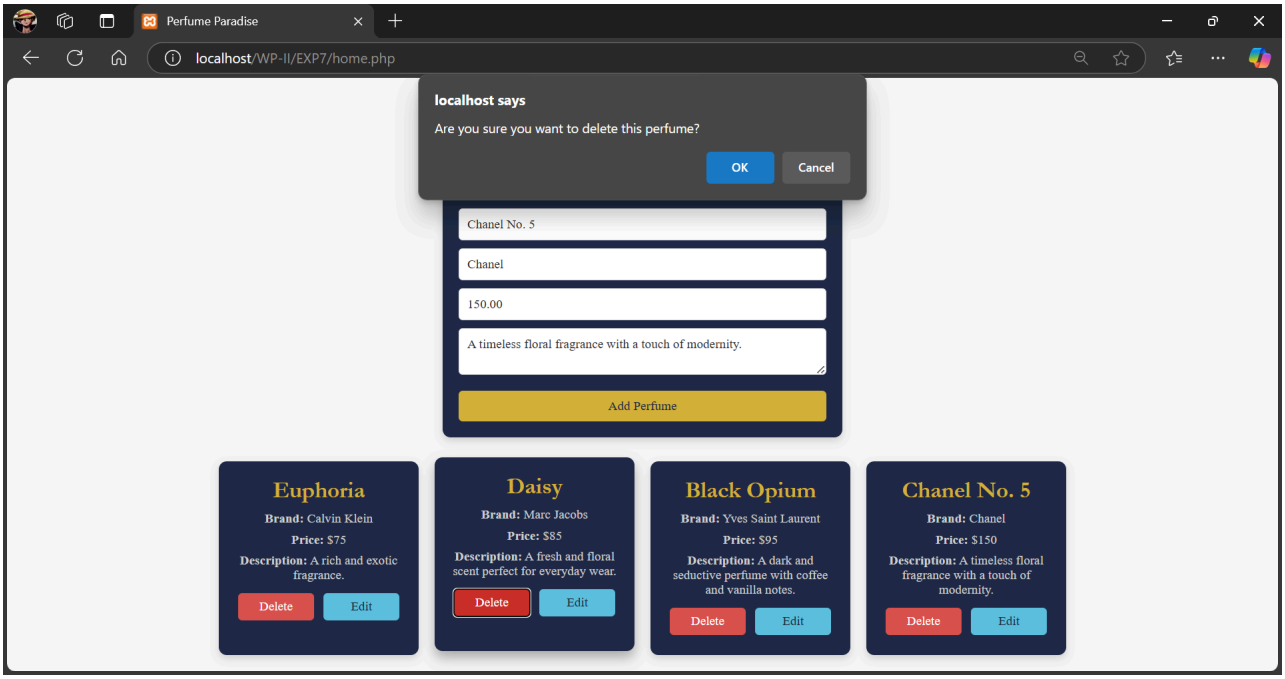
```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^([^\.]+)$ $1.php [NC,L]
```

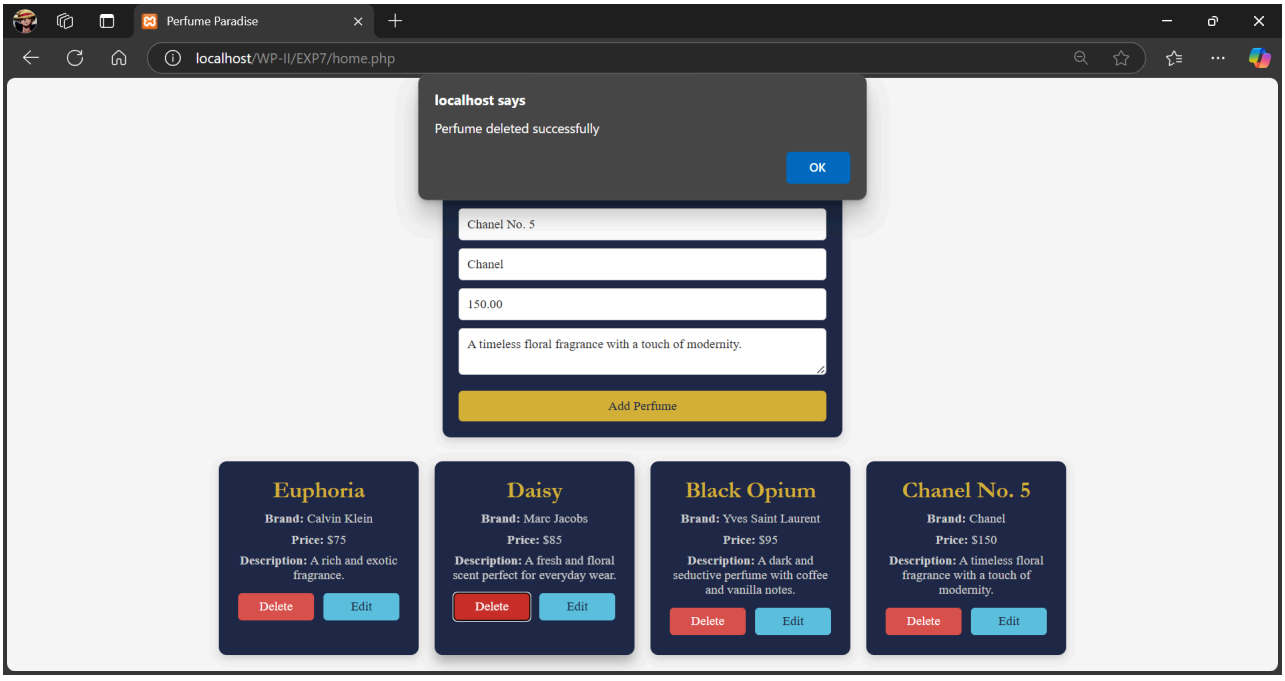

Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Add a New Perfume
Name
Brand
Price
Description
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Daisy
Brand: Marc Jacobs
Price: $85
Description: A fresh and floral scent perfect for everyday wear.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit

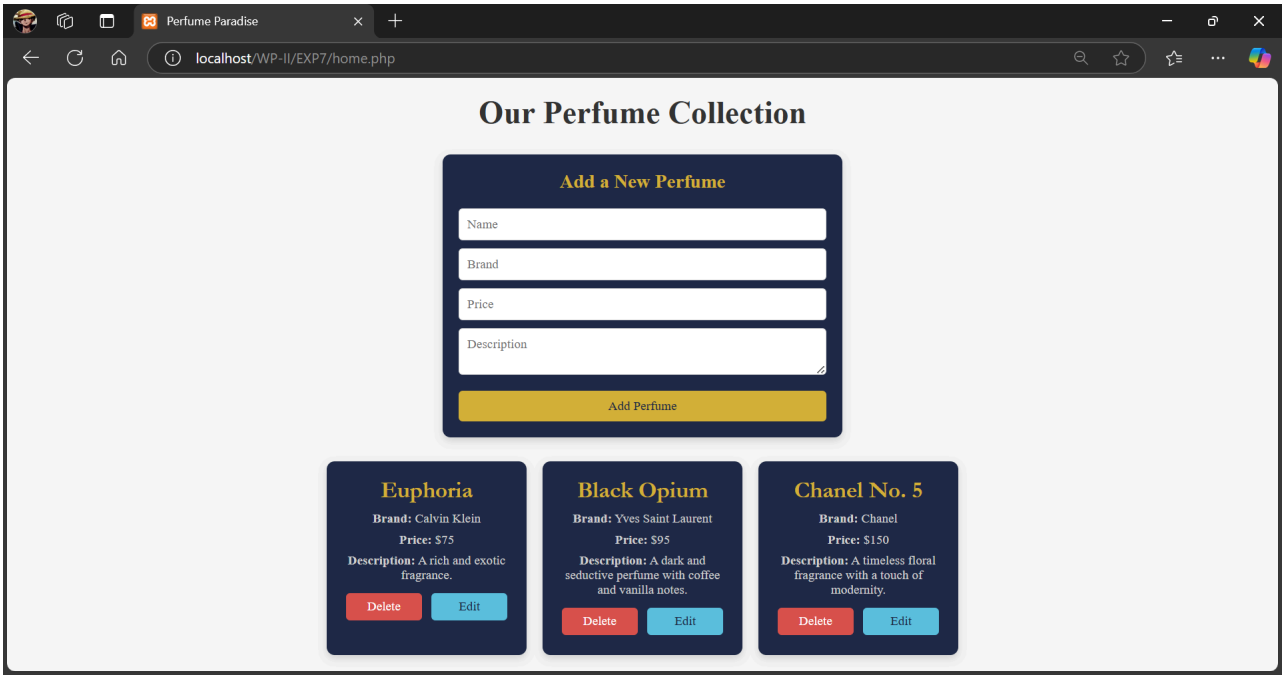

Perfume Paradise
localhost/WP-II/EXP7/home.php
localhost says
Perfume added successfully
OK
Chanel No. 5
Chanel
150.00
A timeless floral fragrance with a touch of modernity.
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Daisy
Brand: Marc Jacobs
Price: $85
Description: A fresh and floral scent perfect for everyday wear.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit

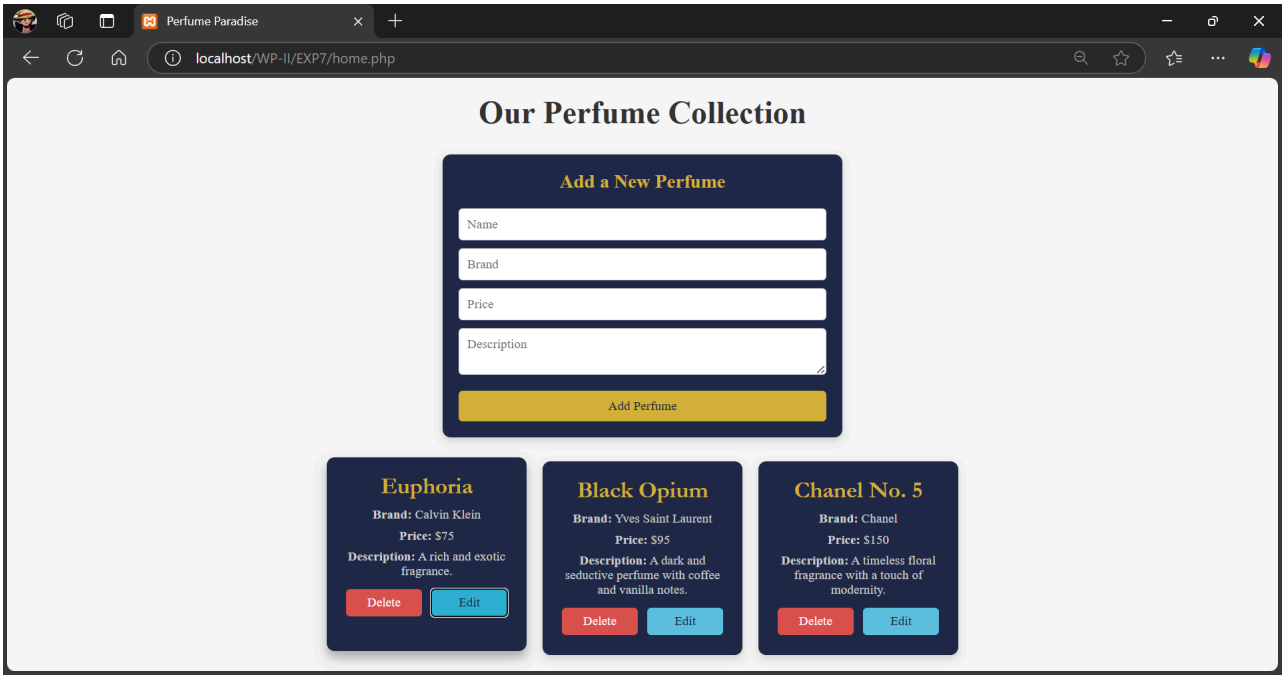

Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Add a New Perfume
Chanel No. 5
Chanel
150.00
A timeless floral fragrance with a touch of modernity.
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Daisy
Brand: Marc Jacobs
Price: $85
Description: A fresh and floral scent perfect for everyday wear.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit

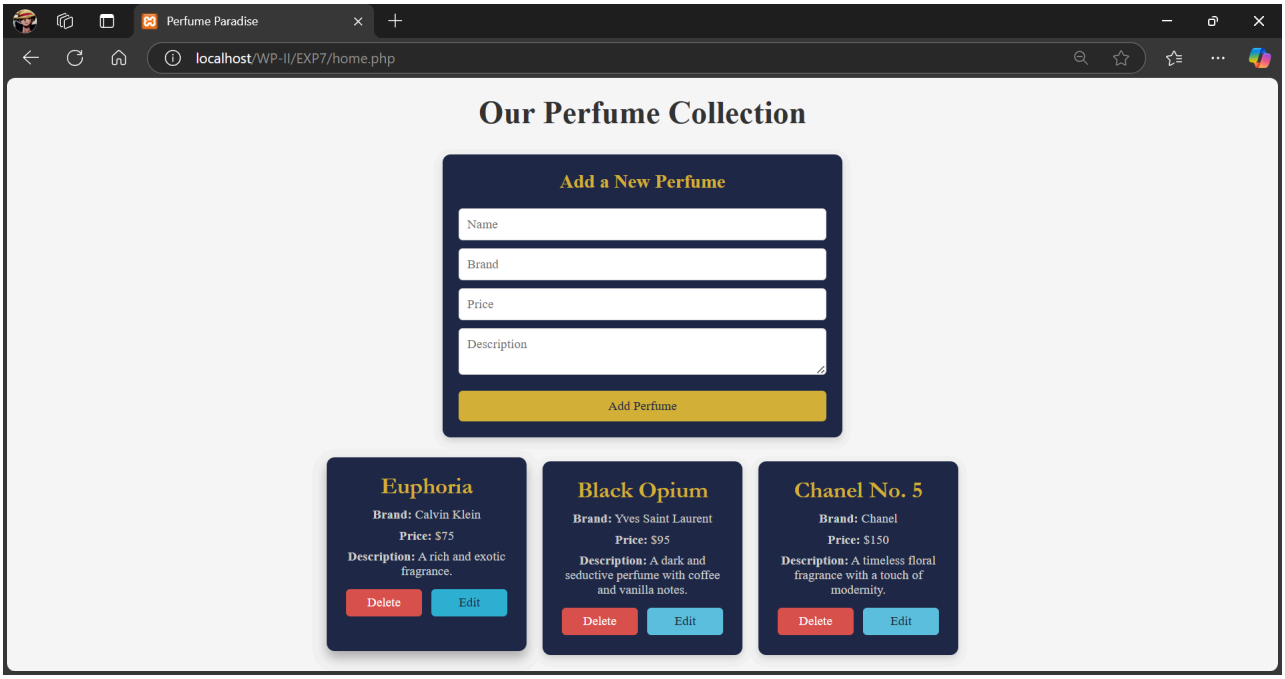

Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Add a New Perfume
Chanel No. 5
Chanel
150.00
A timeless floral fragrance with a touch of modernity.
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Daisy
Brand: Marc Jacobs
Price: $85
Description: A fresh and floral scent perfect for everyday wear.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit

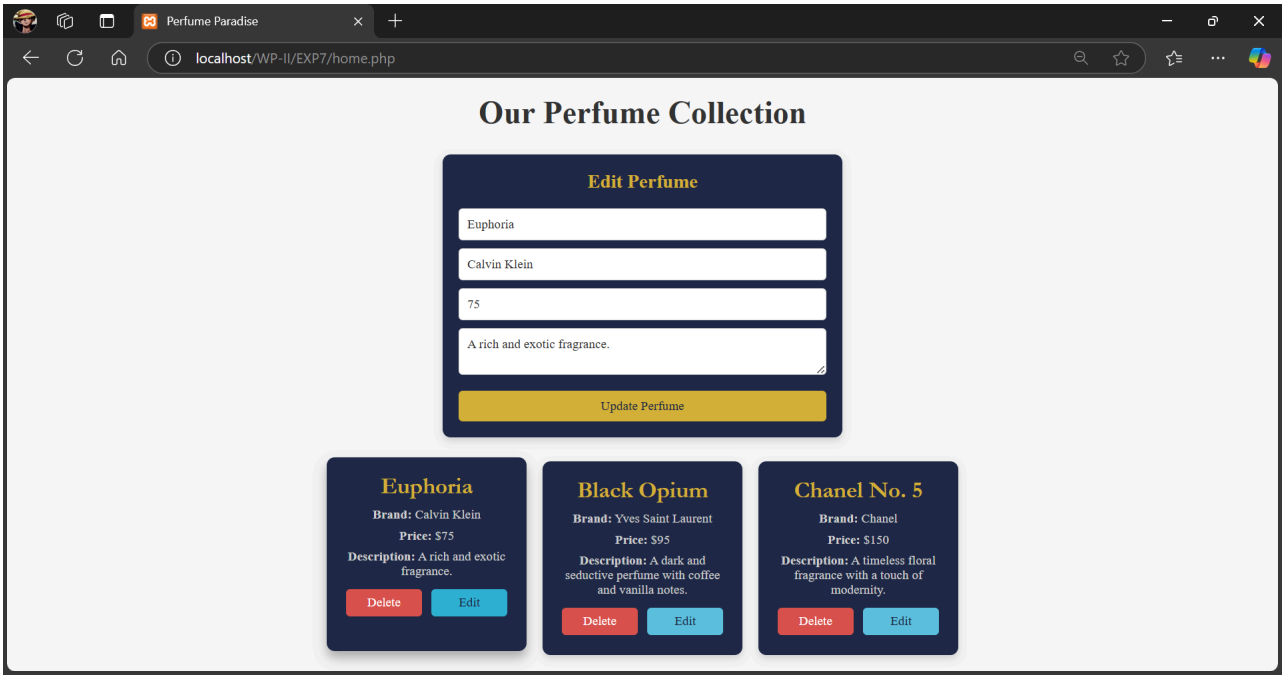

Perfume Paradise
localhost/WP-II/EXP7/home.php
localhost says
Are you sure you want to delete this perfume?
OK
Cancel
Chanel No. 5
Chanel
150.00
A timeless floral fragrance with a touch of modernity.
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Daisy
Brand: Marc Jacobs
Price: $85
Description: A fresh and floral scent perfect for everyday wear.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit

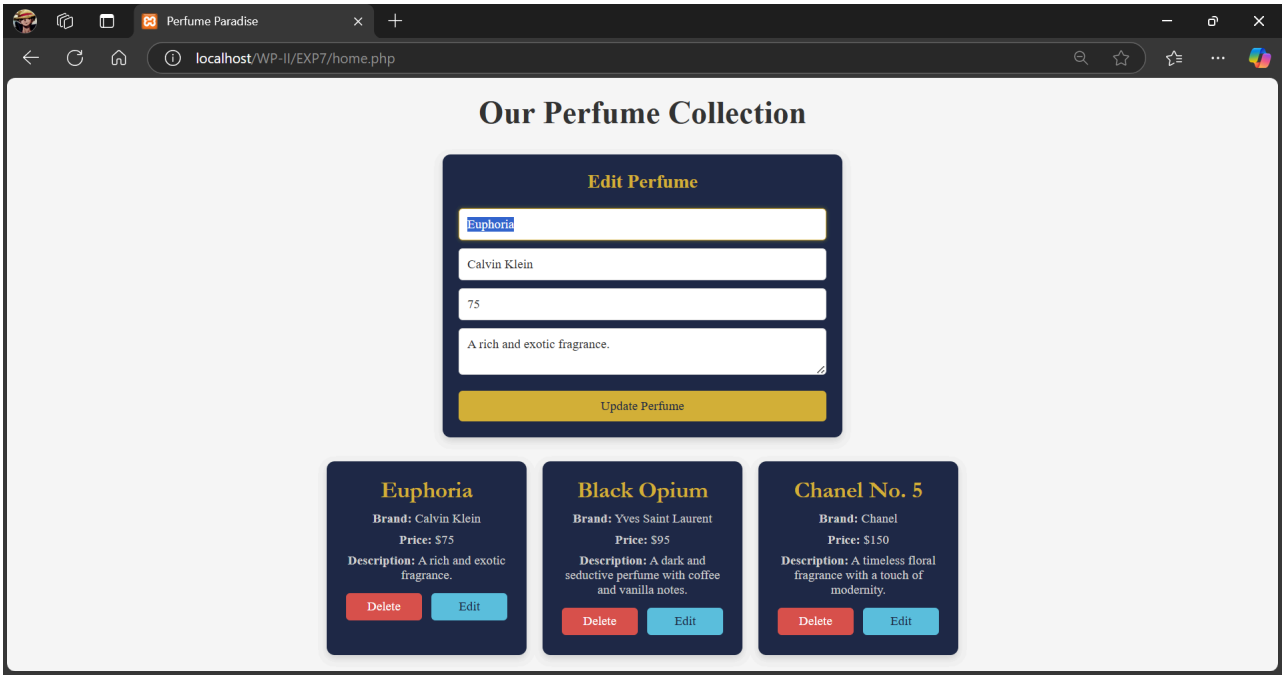

Perfume Paradise
localhost/WP-II/EXP7/home.php
localhost says
Perfume deleted successfully
OK
Chanel No. 5
Chanel
150.00
A timeless floral fragrance with a touch of modernity.
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Daisy
Brand: Marc Jacobs
Price: $85
Description: A fresh and floral scent perfect for everyday wear.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit

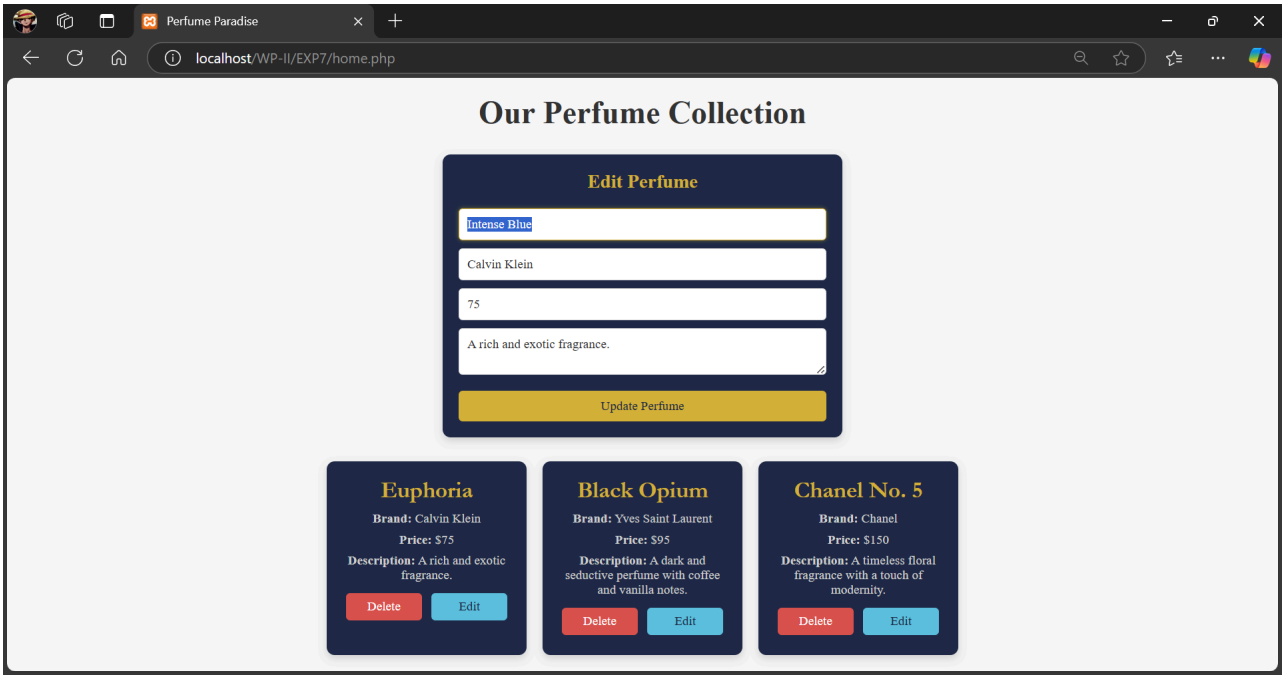

Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Add a New Perfume
Name
Brand
Price
Description
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit

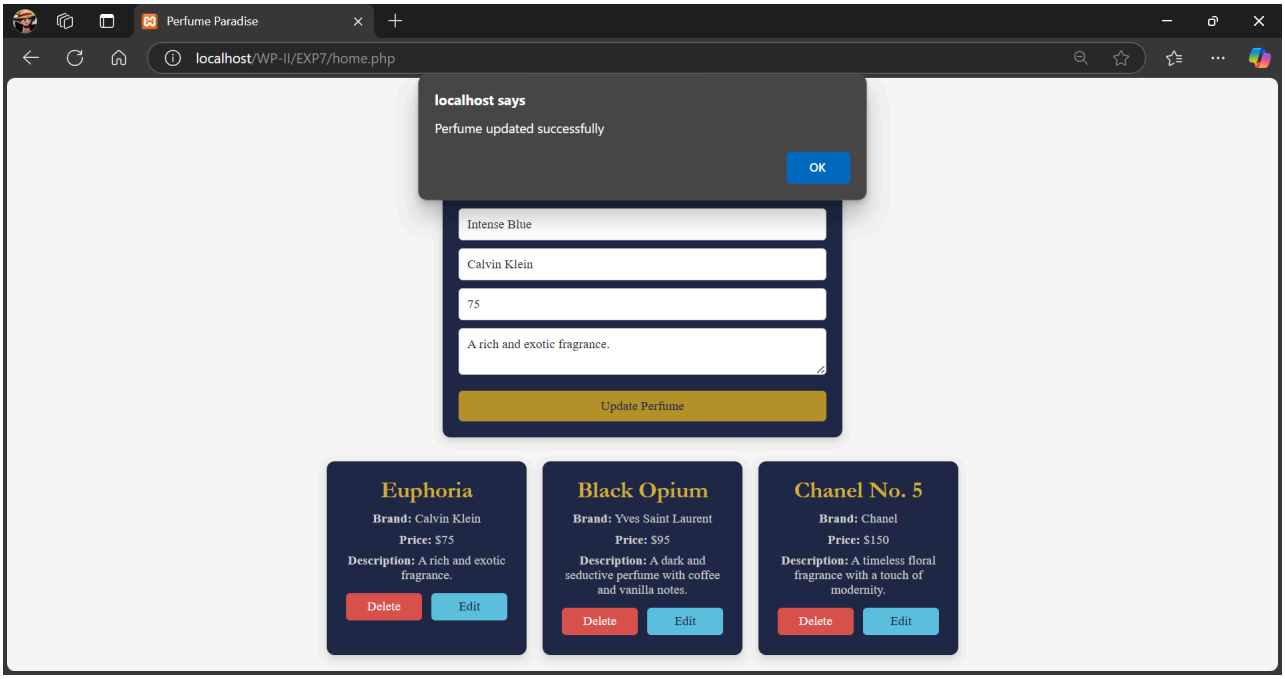

Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Add a New Perfume
Name
Brand
Price
Description
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit

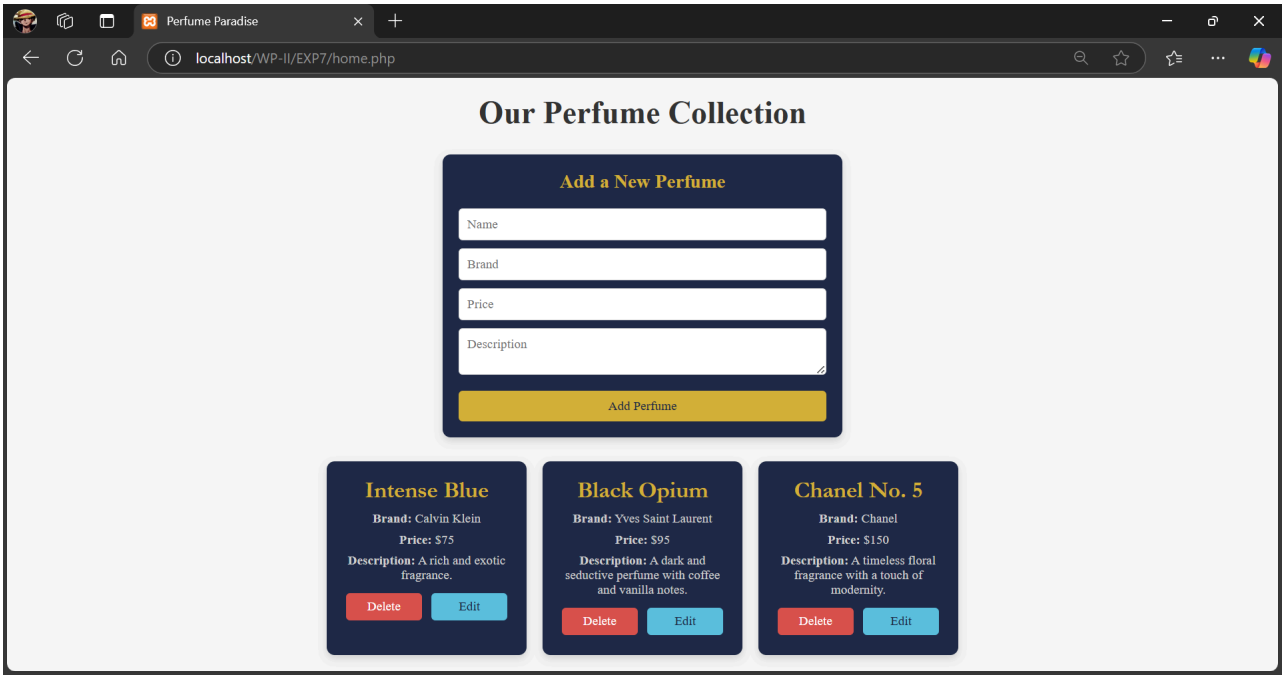

Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Add a New Perfume
Name
Brand
Price
Description
Add Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit


Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Edit Perfume
Euphoria
Calvin Klein
75
A rich and exotic fragrance.
Update Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit


Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Edit Perfume
Euphoria
Calvin Klein
75
A rich and exotic fragrance.
Update Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit


Perfume Paradise
localhost/WP-II/EXP7/home.php
Our Perfume Collection
Edit Perfume
Intense Blue
Calvin Klein
75
A rich and exotic fragrance.
Update Perfume
Euphoria
Brand: Calvin Klein
Price: $75
Description: A rich and exotic fragrance.
Delete
Edit
Black Opium
Brand: Yves Saint Laurent
Price: $95
Description: A dark and seductive perfume with coffee and vanilla notes.
Delete
Edit
Chanel No. 5
Brand: Chanel
Price: $150
Description: A timeless floral fragrance with a touch of modernity.
Delete
Edit





Questions:

1) **What is the purpose of the REST API?**

   The purpose of the REST API is to provide a simplified, scalable, and stateless mechanism for web services to communicate over HTTP. REST APIs allow for easy retrieval, creation,

update, and deletion of resources, supporting a variety of data formats like JSON and XML, making them ideal for web applications that need to exchange data with minimal overhead.

2) **What are the other alternatives to REST API?**
   Alternatives to REST API include:
   - SOAP (Simple Object Access Protocol): A protocol-based web service that provides more stringent standards and built-in error handling but can be more complex and heavy compared to REST.
   - GraphQL: A query language for APIs that allows clients to request only the specific data they need, often used for more flexible and efficient data fetching.
   - gRPC (gRPC Remote Procedure Calls): A high-performance, open-source framework developed by Google that uses HTTP/2 and protocol buffers, making it suitable for low-latency communications.

Outcomes: Demonstrate the use advanced features such as REST API, email handling, localization and internationalization in PHP

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)
In this experiment, we implemented a REST API in PHP to retrieve information about a specific resource. We explored how RESTful architecture, which is stateless and flexible, facilitates web services by supporting various data formats like JSON. Through hands-on practice with PHP functions like json_encode, json_decode, and curl, we successfully retrieved and processed JSON data. This activity demonstrated the ease of implementing REST APIs and highlighted PHP's capability to interact with web services, contributing to our understanding of PHP's role in modern web development.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge with date

References:

Books:

1) Thomson PHP and MySQL Web Development Addison-Wesley Professional, 5th Edition 2016.
2) Peter MacIntyre, Kevin Tatroe Programming PHP O'Reilly Media, Inc, 4th Edition 2020
3) Frank M. Kromann Beginning PHP and MySQL: From Novice to Professional, Apress 1st Edition, 2018
