



**The Faculty of Information and Communication Technology
Mahidol University**

ITCS223: Introduction to Web Development

Semester 2, 2024

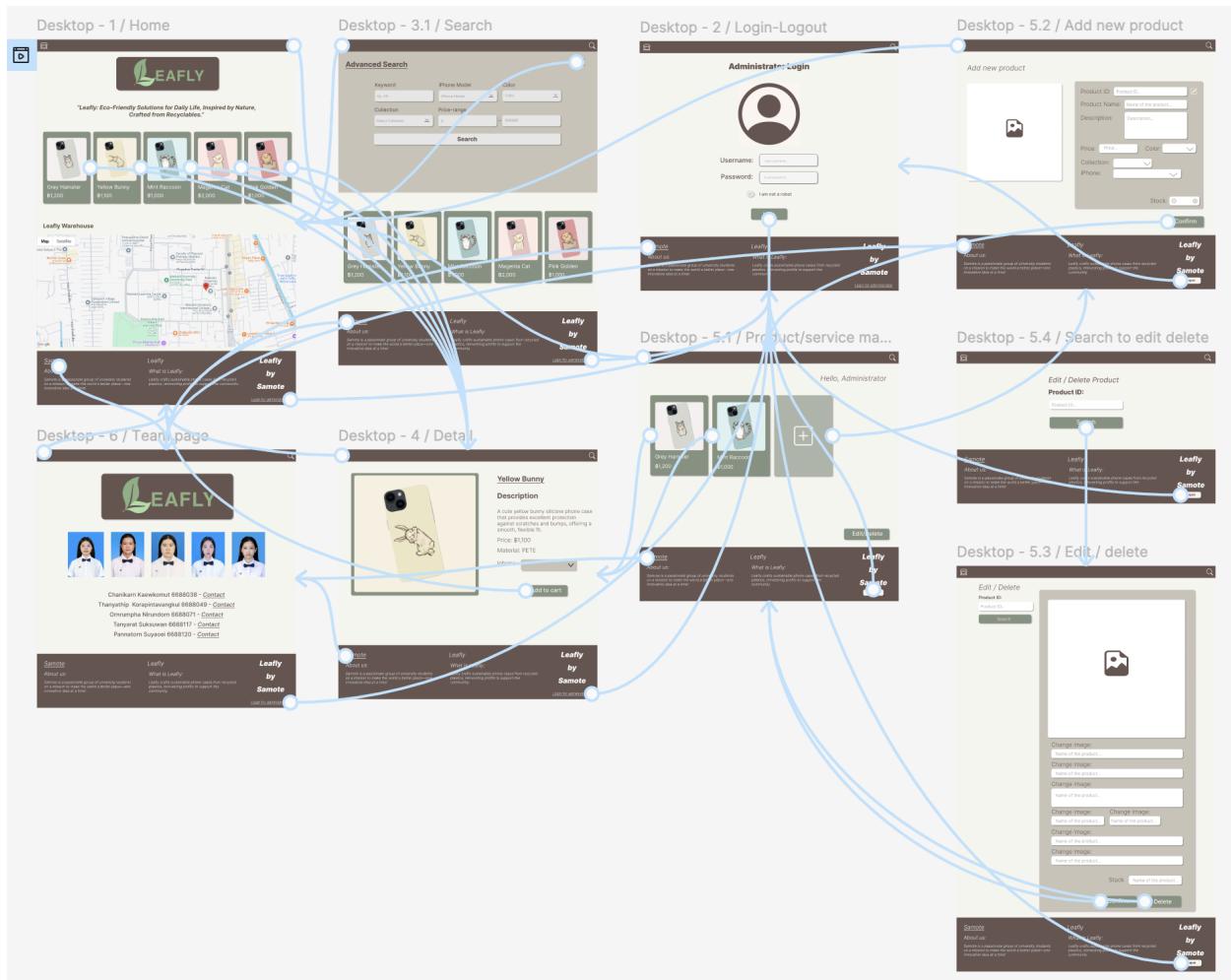
**Samote
Team Members**

| | | |
|-------------------|-------------------------|----------------|
| Chanikarn | Kaewkomut | 6688038 |
| Thanyathip | Korapintavangkul | 6688049 |
| Ornrumpa | Nirundorn | 6688071 |
| Thanyarat | Suksuwan | 6688117 |
| Pannatorn | Suyaoei | 6688120 |

Project Overview :

Our business focuses on providing a variety of high-quality phone cases, allowing the essence of the brand to inspire your own journey [1]. Established with the aim of combining style and protection, we offer customers a wide selection of cases that not only safeguard their devices but also express their personal style because we want to create something that's protective and beautiful at the same time [2]. Our product range includes protective cases, sleek minimalist designs, and vibrant custom prints. Customers can easily personalize their phone cases through our online platform, where they can choose from various colors, phone models, collections, key words, and price range. The platform is designed to offer a user-friendly shopping experience, ensuring secure transactions and quick delivery.

Navigation Diagram



Navigation and Footer: Each page will have a navigation bar at the top, with the left icon representing the home page. The footer at the bottom includes a link to the team page, information about our team, information about our business, and an administrator login link. However, the administrator will see the login link as logout.

The screenshot shows the Leafly website homepage. At the top, there is a navigation bar with a home icon and a search icon. The main header features the Leafly logo with a stylized leaf icon and the text "LEAFLY". Below the header is a tagline: "Leafly: Eco-Friendly Solutions for Daily Life, Inspired by Nature, Crafted from Recyclables." A row of five phone cases is displayed, each with a unique design and price: Grey Hamster (฿1200.00), Magenta Cat (฿2000.00), Mint Raccoon (฿1000.00), Pink Golden (฿1000.00), and Yellow Bunny (฿1100.00). Below this, a map titled "Leafly Warehouse" shows the location of the company's warehouse in Bangkok, Thailand. The map includes various landmarks such as Mahidol University, Starbucks (MU), and Luxe Taste Cafe. At the bottom of the page, there is a footer section for "Samote" and "Leafly", along with links for "About us:", "What is Leafly?", and "Leafly by Samote".

| Product Name | Price |
|--------------|----------|
| Grey Hamster | ฿1200.00 |
| Magenta Cat | ฿2000.00 |
| Mint Raccoon | ฿1000.00 |
| Pink Golden | ฿1000.00 |
| Yellow Bunny | ฿1100.00 |

Home page: The home page will include our logo and other information about our business. The home page displays best-selling products. There is a map of our warehouse. The user can view product details by clicking on each product.

[Advanced Search](#)

| | | |
|---------------------------------|--------------------------|---------|
| Keyword e.g. cat | iPhone Model.. | Color.. |
| Collection Select Collection | Price-range 0 – 99999 | |
| Search | | |



Grey Hamster
฿1200.00



Magenta Cat
฿2000.00



Mint Raccoon
฿1000.00



Pink Golden
฿1000.00



Yellow Bunny
฿1100.00



Banana Vibes
฿990.00



Lavender Dream
฿990.00



Sky Blue
฿990.00

Samote

About us:

Samote is a passionate group of university students on a mission to make the world a better place—one innovative idea at a time!

Leafly

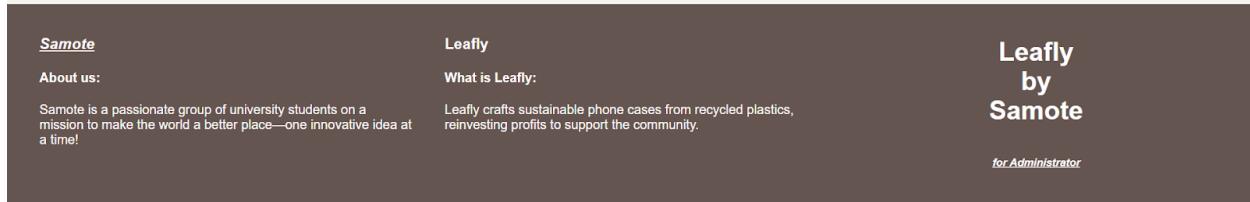
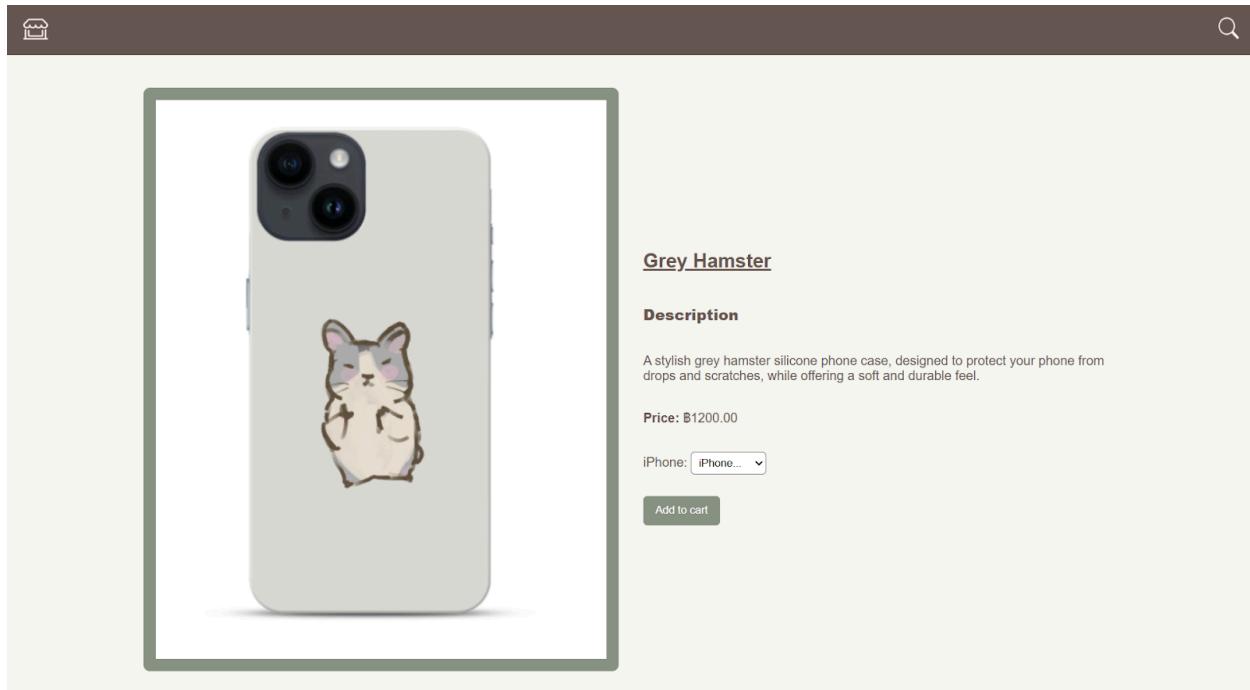
What is Leafly:

Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.

Leafly by Samote

for Administrator

Search page: The user will be able to search and filter based on their requirements. If the user does not search or filter, the results will be shown for all results. The user can search for and filter all or some of them. The results will be shown as the user specifies. The user can view product details by clicking on each product.



Product detail: This page includes the product's picture, name, description, price, iPhone model, and add-to-cart button.

LEAFLY

Chanikarn Kaewkomut 6688038 - [Contact](#)
 Thanyathip Korapintavangkul 6688049 - [Contact](#)
 Orrnrumpha Nirundorn 6688071 - [Contact](#)
 Tanyarat Suksuwan 6688117 - [Contact](#)
 Pannatorn Suyaoei 6688120 - [Contact](#)

Samote
About us:
 Samote is a passionate group of university students on a mission to make the world a better place—one innovative idea at a time!

Leafly
What is Leafly:
 Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.

Leafly by Samote
for Administrator

Team page: each member's name, student ID, and contact information will be displayed with an image of our team.

Administrator Login

Admin ID: Admin password:

I am not a robot

Leafly
What is Leafly:
 Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.

Leafly by Samote
for Administrator

Administrator login page: This page will have "I am not a robot" prompts for the administrator to enter their username and password in order to log in and a button for logging in to access the product service page.

Hello, Administrator

| | | | | | |
|--|--|---|--|--|--|
|  Grey Hamster ฿1200.00 |  Magenta Cat ฿2000.00 |  Mint Raccoon ฿1000.00 |  Pink Golden ฿1000.00 |  Yellow Bunny ฿1100.00 |  Banana Vibes ฿990.00 |
|  Lavender Dream ฿990.00 |  Sky Blue ฿990.00 |  | | | |

[Edit/Delete](#)

Samote

About us:
Samote is a passionate group of university students on a mission to make the world a better place—one innovative idea at a time!

Leafly

What is Leafly:
Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.

Leafly by Samote

[Logout](#)

Product Service page: All of the products are displayed on the Product Service page. The plus symbol on the card connects to the add product page. The edit/delete page has a button that connects to the edit/delete link.

Add new product

Product ID:

Product Name:

Description:

Price: Color:

Collection:

iPhone:

Stock:

Samote

About us:

Samote is a passionate group of university students on a mission to make the world a better place—one innovative idea at a time!

Leafly

What is Leafly:

Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.

Leafly by Samote

Add page: The administrator can enter the product ID, product name, price, colour, collection, iPhone model, stock, and description on this page in addition to uploading the product image. The product will be added to the product service page after you click the confirm button.

Edit / Delete Product

Product ID:

Samote

About us:

Samote is a passionate group of university students on a mission to make the world a better place—one innovative idea at a time!

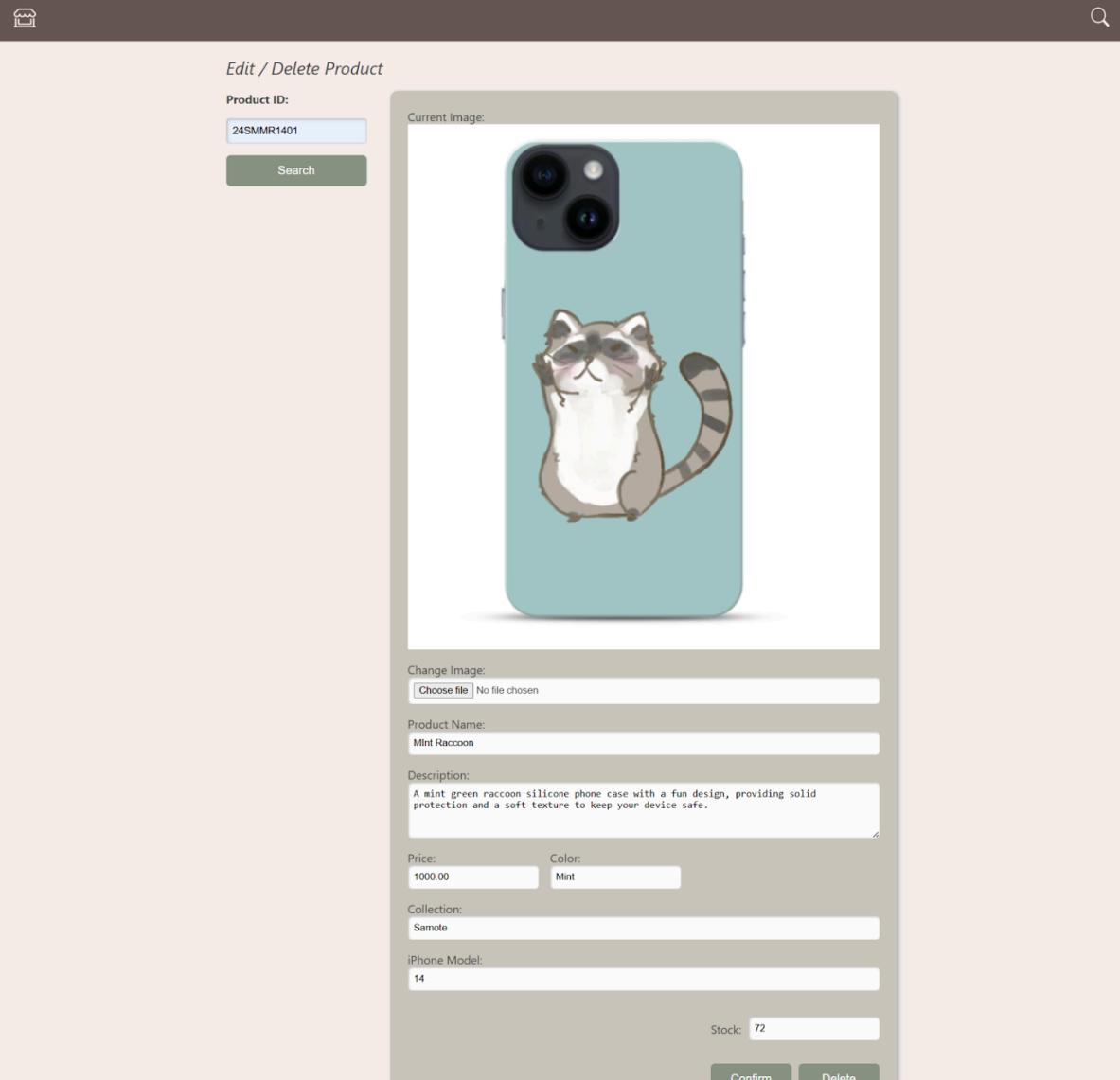
Leafly

What is Leafly:

Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.

Leafly by Samote

The Edit/Delete page: The page for editing or deleting: The product ID that the administrator wishes to modify will be entered.

A screenshot of a web-based admin interface for managing products. The main title is "Edit / Delete Product". On the left, there's a search bar with the product ID "24SMMR1401" and a "Search" button. The central part shows a "Current Image" of a mint green iPhone 14 case featuring a cartoon raccoon design. Below the image are several input fields: "Change Image" (with a "Choose file" button), "Product Name" ("Mint Raccoon"), "Description" (a text area containing a short product description), "Price" ("1000.00"), "Color" ("Mint"), "Collection" ("Samote"), "iPhone Model" ("14"), and "Stock" ("72"). At the bottom right are "Confirm" and "Delete" buttons. The footer contains links for "Samote" (About us, Leafly, What is Leafly), "Leafly by Samote" (Logout), and a "Logout" button.

After the admin search, the product details will be displayed, and the admin may edit any information, then click the confirm button. If the administrator wants to delete something, they can do it by clicking the delete button.

Details of web application and code

1. Home Page:

home.html

```
<body>
    <!-- Navbar -->
    <nav>
        <a href="/home"></a>
        <span class="search-icon"><a href="/search"></a></span>
    </nav>
    <!-- Header -->
    <header>
        
        <p class="tagline">"Leafly: Eco-Friendly Solutions for Daily Life, Inspired by Nature, Crafted from Recyclables." </p>
    </header>

    <!-- Product Section -->
    <!-- show our best seller products -->
    <div class="product-container" id="product-container"></div>

    <!-- GOOGLE MAP -->
    <h2 class="map">Leafly Warehouse</h2>
    <div id="map"></div>

    <!-- FOOTER -->
    <footer>
        <div class="footer-column">
            <h3><a href="/team">Samote</a></h3>
            <h4>About us:</h4>
            <p>Samote is a passionate group of university students on a mission to make the world a better place--one innovative idea at a time!</p>
        </div>
        <div class="footer-column">
            <h3>Leafly</h3>
            <h4>What is Leafly:</h4>
            <p>Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.</p>
        </div>
        <div class="footer-column">
            <div class="last-footer">
                <h1>Leafly <br> by <br> Samote</h1>
                <h5><a href="/admin-page"><br>for Administrator</a></h5>
            </div>
        </div>
    </footer>
    <script src="home.js"></script>
    <!-- Download Google Maps API -->
    <script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAmIbdkyjrg228uUkWH1-bSd0St0AGzuXM&callback=initMap">
    </script>
</body>
```

styles.css

```
/* home.html */
/* General Styles */
body {
    font-family: Arial, Helvetica, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #F5F5F0;
    color: #ffffff;
}

/* map section Leafly Warehouse*/
.map {
    text-align: left;
    margin: 30px;
    color: #685752;
}

#map{
    height: 500px;
    width: 100%;
}

/* Navigation Bar */
nav {
    /*makes two icons separate*/
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: #685752;
    padding: 15px 20px;
}

nav img {
    width: 30px;
}

.search-icon {
    font-size: 20px;
    color: white;
    cursor: pointer;
}

.search-icon a {
    text-decoration: none;
}

.search-icon img {
    width: 25px;
}

/* Header Section */
header {
    text-align: center;
    padding: 20px;
}

header img {
```

```
width: 230px;
height: 100px;
background-color: #685752;
object-fit: contain; /* Ensures the image fills the area properly*/
border-radius: 10px;
}

.logo {
  background: #685752;
  color: #C5D1B2;
  display: inline-block;
  padding: 10px 20px;
  border-radius: 5px;
}

.tagline {
  font-style: italic;
  color: #685752;
  font-weight: bold;
}

/* Product Section */
.product-container {
  display: flex;
  justify-content: center;
  flex-wrap: wrap;
  gap: 15px;
  padding: 20px;
}

.product {
  color: #ffffff;
  background-color: #879283;
  padding: 15px;
  text-align: center;
  width: 180px;
  border-radius: 10px;
}

.product img {
  width: 100%;
  border-radius: 10px;
  width: 170px;
  height: 200px;
  object-fit: cover;
}

.product-name {
  font-weight: bold;
  margin: 10px 0;
}

.a.product{
  text-decoration: none;
}

.product-price {
  color: #ffffff;
  font-size: 16px;
```

```
}

/* Footer Section */
footer {
    background-color: #685752;
    color: white;
    padding: 20px;
    margin-top: 20px;
    display: flex;
    justify-content: space-between;
    text-align: left;
}

.last-footer {
    text-align: center;
}

.footer-column {
    flex: 1;
    padding: 0 20px;
}

footer a {
    color: rgb(255, 255, 255);
    text-decoration: none;
    font-style: italic;
    text-decoration: underline;
}
```

home.js

```
function initMap() {
  console.log("initMap function called");

  // Leafly Warehouse Location
  const leaflyStore = { lat: 13.7947, lng: 100.3252 };

  // Maps
  const map = new google.maps.Map(document.getElementById("map"), {
    center: leaflyStore,
    zoom: 16,
  });

  // Maps marker
  const marker = new google.maps.Marker({
    position: leaflyStore,
    map: map,
    title: "Leafly Store",
  });

  // Message box when click Maps marker
  const infowindow = new google.maps.InfoWindow({
    content: "<strong>Leafly Store @ Salaya</strong>",
  });

  // Open message box when click Maps marker
  marker.addListener("click", () => {
    infowindow.open(map, marker);
  });
}

// Make initMap global function
window.initMap = initMap;
// Pull product information when Loading
function fetchProducts() {
  fetch('http://localhost:4000/api/products')
    .then(response => response.json())
    .then(products => {
      console.log('Products:', products);
      const productContainer = document.getElementById('product-container');
      const seenNames = new Set();

      const NewCollectionProducts = products
        .filter(product => product.Collection === "Samote")
        .filter(product => {
          if (seenNames.has(product.Product_Name)) {
            return false;
          }
          seenNames.add(product.Product_Name);
          return true;
        })
        .slice(0, 5);

      NewCollectionProducts.forEach(product => {
        console.log(`Product ID for ${product.Product_Name}: ${product.Product_ID}`);
        const productElement = document.createElement('a');
        productElement.classList.add('product');
        productElement.href = `detail.html?id=${product.Product_ID}`;
      });
    });
}
```

```
const productImage = document.createElement('img');

if (product.Product_Img) {

    productImage.src = product.Product_Img;
} else {
    // If no picture use placeholder instead
    productImage.src = 'placeholder.jpg';
}

productImage.alt = product.Product_Name;

const productName = document.createElement('p');
productName.classList.add('product-name');
productName.textContent = product.Product_Name;

const productPrice = document.createElement('p');
productPrice.classList.add('product-price');
productPrice.textContent = `฿${product.Price}`;

productElement.appendChild(productImage);
productElement.appendChild(productName);
productElement.appendChild(productPrice);

productContainer.appendChild(productElement);
});
})
.catch(error => console.error('Error fetching products:', error));
}

window.onload = fetchProducts;
```

2. Login Page for Administrators:

login.html

```
<!-- Login Section -->
<div class="login-container">
    <div class="login-box">
        <h2>Administrator Login</h2>
        
        <br>
        <br>

        <!-- form -->
        <form id="loginForm">
            <!-- input admin id -->
            <label for="admin_id">Admin ID:</label>
            <input type="text" id="admin_id" name="admin_id" placeholder="Input Admin ID...">
            <br>
            <br>

            <!-- input admin password -->
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" placeholder="Input password...">
            <br>
            <br>

            <div class="captcha">
                <!-- checkbox for 'I am not a robot' -->
                <input type="checkbox" id="captcha">
                <label for="captcha">I am not a robot</label>
            </div>
            <br>
            <button type="submit" class="login-btn">Login</button>
        </form>
    </div>
</div>
```

login.css

```
/* Login section */
.login-box {
    color: #685752;
    text-align: center;
}

.login-box h2 {
    font-size: 30px;
}

.login-btn:hover {
    background-color: #6A6F4C;
}

.login-btn {
    color: white;
    background-color: #6A6F4C;
    border-radius: 5px;
    padding: 8px 17px;
    font-size: 18px;
    width: auto;
}
```

login.js

```
document.addEventListener('DOMContentLoaded', function () {

    const loginForm = document.getElementById('loginForm');

    loginForm.addEventListener('submit', async function (e) {
        e.preventDefault();

        // Get values from the form inputs
        const admin_id = document.getElementById('admin_id').value;
        const password = document.getElementById('password').value;
        const captcha = document.getElementById('captcha').checked;

        // Validate if checkbox is checked
        if (!captcha) {
            alert('Please confirm you are not a robot.');
            return;
        }

        try {
            // Send login request to the server
            const response = await fetch('http://localhost:4000/admin-login', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({ admin_id, password }),
                credentials: 'include'
            });
            const data = await response.json();
            if (data.success) {
                window.location.href = data.redirectUrl;
            }
        } catch (error) {
            console.error(error);
        }
    });
})
```

```
        } else {
            alert(data.message || 'Login failed');
        }

    } catch (error) {
    console.error('Error during login:', error);
    alert('Login failed. Please try again.');
}
});  
});
```

3. Search Page for Searching Products/Services

SearchPage.html

```
<!-- Search section -->
<div class="search-center-wrapper">
    <form class="search-form" action="/search" method="GET">
        <div class="form-grid-2rows">
            <!-- Row 1 -->
            <div class="form-item keyword">
                <!-- Keyword -->
                <label for="keyword">Keyword</label>
                <input type="text" id="keyword" name="keyword" placeholder="e.g. cat"/>
            </div>

            <div class="form-item model">
                <!-- Model (dropdown)-->
                <label for="model">iPhone Model</label>
                <select id="model" name="model">
                    <option value="">iPhone Model..</option>
                    <option value="14">14</option>
                    <option value="15">15</option>
                    <option value="16">16</option>
                </select>
            </div>

            <div class="form-item color">
                <!-- Color (dropdown)-->
                <label for="color">Color</label>
                <select id="color" name="color">
                    <option value="">Color..</option>
                    <option>Yellow</option>
                    <option>Red</option>
                    <option>Pink</option>
                    <option>Grey</option>
                    <option>Mint</option>
                    <option>Green</option>
                    <option>Orange</option>
                    <option>Purple</option>
                    <option>Brown</option>
                    <option>Blue</option>
                </select>
            </div>

            <!-- Row 2 -->
            <div class="form-item collection">
                <!-- Collection (dropdown)-->
                <label for="collection">Collection</label>
                <select id="collection" name="collection">
                    <option value="">Select Collection</option>
                    <option value="Samote">Samote</option>
                    <option value="Plain">Plain</option>
                </select>
            </div>

            <div class="form-item price">
                <!-- Price-range -->
```

```

        <label>Price-range</label>
        <div class="price-range">
            <input type="number" min="0" placeholder="0" name="price-min">
            <span>_</span>
            <input type="number" min="0" placeholder="99999" name="price-max">
        </div>
    </div>
<!-- Search Button -->
<div class="search-button-wrapper">
    <button type="submit" class="search_button" onclick="searchProduct()">Search</button>
</div>
</form>
</div>
</div>

<!-- Displayed products section -->
<div class="product-container" id="resultsContainer"></div>

```

Searchstyles.css

```

/*--- Search container ---*/
.container {
    background-color: #CAC5BC;
    padding: 20px;
    text-align: left;
}

/* Center wrapper for the whole search form */
.search-center-wrapper {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    padding: 40px 20px;
}

/* Button centered under grid */
.search-button-wrapper {
    margin-top: 30px;
    display: flex;
    justify-content: center;
}

/* Label & Input Styling */
label {
    color: #5B4C48;
    font-size: 18px;
    margin-bottom: 5px;
    display: block;
}

.keyword input {
    width: 87%;
}

input, select {

```

```
padding: 13px;
font-size: 16px;
background: #E7E7E7;
outline: none;
box-shadow: 3px 3px 6px rgba(0, 0, 0, 0.2);
border-radius: 5px;
border: none;
width: 100%;
}

.form-item select {
  color: gray;
  height: 45px;
  padding: 10px;
  border-radius: 5px;
  background: #E7E7E7;
  text-align: center;
}

/* Price Range */
.price-range span {
  font-size: 18px;
  color: #5B4C48;
}
.price-range {
  display: flex;
  align-items: center;
  gap: 10px;
}

/* Search Button */
.search_button {
  width: 100%;
  padding: 10px;
  background-color: #eae9e7;
  color: #5B4C48;
  font-size: 20px;
  font-weight: bold;
  border: none;
  border-radius: 12px;
  cursor: pointer;
  transition: background-color 0.2s ease;
  box-shadow: 3px 3px 6px rgba(0, 0, 0, 0.2);
  outline: none;
}

.search_button:hover {
  background-color: #879283;
}

.form-item {
  width: 100%;
}

.form-grid-2rows {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-areas:
    "keyword model color"
```

```
    "collection price price";
gap: 20px;
max-width: 900px;
margin: 0 auto;
}

.keyword      { grid-area: keyword; }
.model        { grid-area: model; }
.color         { grid-area: color; }
.collection   { grid-area: collection; }
.price        { grid-area: price; }

/* Product Section */
.product-container {
  display: flex;
  justify-content: left;
  flex-wrap: wrap;
  gap: 15px;
  padding: 20px;
}

.product {
  color: #ffffff;
  background-color: #879283;
  padding: 15px;
  text-align: center;
  width: 180px;
  border-radius: 10px;
}

.product img {
  width: 100%;
  border-radius: 10px;
  width: 170px;
  height: 200px;
  object-fit: cover;
}

.product-name {
  font-weight: bold;
  margin: 10px 0;
}

a.product{
  text-decoration: none;
}

.product-price {
  color: #ffffff;
  font-size: 16px;
}
```

search.js

```
document.addEventListener('DOMContentLoaded', () => {
  // Show when search with no criteria / show all product
  searchProduct();
});

// Function to go back to the previous page
function closeSearch() {
  window.history.back();
}

// Function to search for products based on the form input
function searchProduct(event) {
  if (event) event.preventDefault();

  const keyword = document.getElementById('keyword').value;
  const model = document.getElementById('model').value;
  const color = document.getElementById('color').value;
  const collection = document.getElementById('collection').value;
  const priceMin = document.querySelector('input[name="price-min"]').value;
  const priceMax = document.querySelector('input[name="price-max"]').value;

  const query = new URLSearchParams({
    keyword,
    model,
    color,
    collection,
    'price-min': priceMin,
    'price-max': priceMax
  });

  // Send a request to the back-end with the search query
  fetch(`http://localhost:4000/search-api?${query.toString()}`)
    .then(res => res.json())
    .then(data => {
      console.log(data);
      const container = document.getElementById('resultsContainer');
      container.innerHTML = '' // Clear previous results

      if (data.length === 0) {
        container.innerHTML = '<p>No products found.</p>';
        return;
      }

      data.forEach(product => {
        const productDiv = document.createElement('div');
        productDiv.className = 'product';
        productDiv.innerHTML =
          `<a href="detail.html?id=${product.Product_ID}">
            
          </a>
          <p class="product-name">${product.Product_Name}</p>
          <p class="product-price">฿${product.Price}</p>
        `;
        container.appendChild(productDiv); // Add to the container
      });
    })
    .catch(err => {
```

```
    console.error('Error fetching search results:', err);
  });
}

document.querySelector('.search-form').addEventListener('submit', searchProduct);
```

4. Detail Page for the Product/Service Result

Detail.html

```
<!-- Main Content-->
<div class="product-page-detail">
    <div>
        
    </div>
    <div class="center-detail">
        <div class="product-details-detail">
            <h2 class="product-title-detail"></h2>
            <h3><strong>Description</strong></h3>
            <p><br></p>
            <p><strong>Price:</strong></p>
            <p><label for="iphone">iPhone:</label>
                <select class="product-select-detail" id="iphone">
                    <option value="">iPhone...</option>
                    <option value="iphone14">iPhone 14</option>
                    <option value="iphone15">iPhone 15</option>
                    <option value="iphone16">iPhone 16</option>
                </select>
            </p>
            <button class="add-to-cart-detail">Add to cart</button>
        </div>
    </div>
</div>
```

DetailandProductService.css

```
/* Detail Page */
.product-page-detail {
    display: flex;
    justify-content: center;
    align-items: center;
    gap: 30px;
    padding: 40px;
    max-width: 1200px;
    margin: 0 auto;
}

.product-page-detail > div {
    flex: 1;
    display: flex;
    justify-content: center;
}

.center-detail {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: flex-start;
    text-align: center;
}

.product-image-detail {
```

```
    flex: 1;
    width: 500px;
    display: flex;
    justify-content: center;
    align-items: center;
    border-radius: 10px;
}

.product-detail {
    background-color: #879283;
    padding: 15px;
    text-align: center;
    border-radius: 8px;
}

.product-details-detail {
    flex: 1;
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    text-align: left;
}

.product-title-detail {
    color: #685752;
    font-weight: bold;
    text-decoration: underline;
}

.product-select-detail {
    padding: 5px;
    border-radius: 5px;
}

.add-to-cart-detail {
    margin-top: 10px;
    background-color: #879283;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

.add-to-cart-detail:hover, .edit-delete-btn-PS:hover {
    background-color: #6a7a63;
}
```

details.js

```
window.onload = async () => {
  const urlParams = new URLSearchParams(window.location.search);
  const productId = urlParams.get("id");

  // If no product ID is found in the URL, show an alert and stop
  if (!productId) {
    alert("Product not found");
    return;
  }

  try {
    const response = await fetch(`http://localhost:4000/api/product/${productId}`);

    // Check if the response from the server is successful
    if (!response.ok) {
      throw new Error(`Failed to fetch product: ${response.status} ${response.statusText}`);
    }

    const product = await response.json();

    // Update the page with the product information
    document.querySelector(".product-image-detail").src = product.Product_Img; // Set product image
    document.querySelector(".product-title-detail").innerText = product.Product_Name; // Set product name
    document.querySelector(".product-details-detail p:nth-of-type(1)").innerHTML = product.Description
    || "No description." // Set product description
    document.querySelector(".product-details-detail p:nth-of-type(2)").innerHTML =
`<strong>Price:</strong> ${product.Price}` // Set product price
  } catch (err) {
    console.error("Error loading product:", err);
    alert("Error loading product details.");
  }
};
```

5. Product/Service Management Page for Administrators

ProductService.html

```
<!-- Main Content-->
<h2 class="admin-greeting-PS"><em>Hello, Administrator</em></h2>

<div class="product-grid-PS" id="product-list"> </div>

<div class="button-container-PS">
    <div class="edit-delete-btn-PS"><a href="/delete-edit-page">Edit/Delete</a></div>
</div>

<!-- FOOTER -->
<footer>
    <div class="footer-column">
        <h3><a href="/team">Samote</a></h3>
        <h4>About us:</h4>
        <p>Samote is a passionate group of university students on a mission to make the world a better place--one innovative idea at a time!</p>
    </div>
    <div class="footer-column">
        <h3>Leafly</h3>
        <h4>What is Leafly:</h4>
        <p>Leafly crafts sustainable phone cases from recycled plastics, reinvesting profits to support the community.</p>
    </div>
    <div class="footer-column">
        <div class="last-footer">
            <h1>Leafly <br> by <br> Samote</h1>
            <button type="submit" class="logout_btn" id="logoutBtn">Logout</button>
        </div>
    </div>
</footer>
```

DetailandProductService.css

```
/* Product Service Page */
.admin-greeting-PS {
    text-align: right;
    margin-right: 20px;
    font-style: italic;
}

.product-grid-PS {
    display: flex;
    flex-wrap: wrap;
    justify-content: flex-start;
    margin-left: 40px;
    gap: 20px;
    margin-top: 20px;
}
```

```
.product-card-PS, .add-product-card-PS {
    background-color: #8B9C89;
    width: 180px;
    padding: 10px;
    border-radius: 10px;
    text-align: center;
    box-shadow: 2px 2px 8px rgba(0, 0, 0, 0.1);
    flex: 0 0 auto;
}

.product-card-PS img {
    width: 100%;
    height: 200px;
    object-fit: cover;
    border-radius: 10px;
}

.add-product-card-PS {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 300px;
}

.add-product-card-PS:hover {
    background-color: #6a7a63;
}

.add-icon-PS {
    font-size: 40px;
    color: white;
    border: 2px solid white;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
    text-decoration: none;
}

.add-icon-PS a {
    text-decoration: none;
    color: white;
}

.button-container-PS {
    display: flex;
    justify-content: flex-end;
    margin-right: 40px;
}

.edit-delete-btn-PS {
    display: inline-block;
    padding: 10px 20px;
    background-color:#8B9C89;
    color: white;
    text-decoration: none;
    border-radius: 5px;
    text-align: center;
}
```

```
.edit-delete-btn-PS a {
  text-decoration: none;
  color: white;
}
```

ProductService.js

```
document.addEventListener('DOMContentLoaded', async () => {
  try {
    // Check admin Login status by calling backend API
    const response = await fetch('http://localhost:4000/check-admin', {
      method: 'GET',
      credentials: 'include', // Send session cookie with the request
    });

    if (response.status === 401) {
      // If not Logged in as admin, redirect to Login page
      window.location.href = '/login';
    } else {
      // If admin is authenticated, continue to Load the admin page
      const data = await response.json();
      console.log('Welcome admin:', data);
    }
  } catch (error) {
    console.error('Error checking admin status:', error);
  }

  // Load product data and render product cards
  try {
    const productResponse = await fetch('http://localhost:4000/search-api?');
    const products = await productResponse.json();
    const productList = document.getElementById('product-list');

    products.forEach(product => {
      const card = document.createElement('div');
      card.className = 'product-card-PS';

      card.innerHTML = `
        <a href="detail.html?id=${product.Product_ID}">
          
        </a>
        <p class="product-name">${product.Product_Name}</p>
        <p class="product-price">฿${product.Price}</p>
      `;

      productList.appendChild(card);
    });

    // Add card
    const addCard = document.createElement('div');
    addCard.className = 'add-product-card-PS';
    addCard.innerHTML = '<div class="add-icon-PS"><a href="/add-page">+</a></div>';
    productList.appendChild(addCard);
  } catch (err) {
```

```
        console.error('Error loading products:', err);
    }

// Set up event listener for the Logout button
document.getElementById('logoutBtn').addEventListener('click', async function () {
    try {
        const response = await fetch('http://localhost:4000/admin-logout', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            credentials: 'include', // Send session cookie with the request
        });

        const data = await response.json();

        if (data.success) {
            console.log('Logout successful');
            // Redirect to Login page after successful Logout
            window.location.href = '/login';
        } else {
            alert('Logout failed: ' + data.message);
        }
    } catch (error) {
        console.error('Error during logout:', error);
        alert('Logout error, please try again.');
    }
});
```

6. Team Page

team.html

```
<!-- picture -->
<div class="teampic-container">
  <div class="teampic">
    
  </div>
  <div class="teampic">
    
  </div>
  <div class="teampic">
    
  </div>
  <div class="teampic">
    
  </div>
  <div class="teampic">
    
  </div>
</div>

<!-- name & member contact-->
<div class="teamname">
  <p>Chanikarn Kaewkomut 6688038 - <a href="https://www.instagram.com/tampchnk?igsh=bHp3dDB6cTVqbXAz&utm_source=qr" target="_blank">Contact</a></p>
  <p>Thanyathip Korapintavangkul 6688049 - <a href="https://www.instagram.com/_diiaaa_?igsh=MWJ3b25mb3J1bmM1ZA==" target="_blank">Contact</a></p>
  <p>Ornrumpha Nirundorn 6688071 - <a href="https://www.instagram.com/shisha_sushi?igsh=c3phZTM4dmZleWFw&utm_source=qr" target="_blank">Contact</a></p>
  <p>Tanyarat Suksawan 6688117 - <a href="https://www.instagram.com/tyr_ploy?igsh=MXY4NHQycGtza2RjcA==" target="_blank">Contact</a></p>
  <p>Pannatorn Suyaoei 6688120 - <a href="https://www.instagram.com/betterbimon?igsh=enVrbHg2cXlzaWI3&utm_source=qr" target="_blank">Contact</a></p>
</div>
```

styles.css

```
/* for team page */
.teamname {
    text-align: center;
    color: #685752;
    font-weight: bold;
    font-size: large;
}

.teamname a{
    text-align: center;
    color: #685752;
    font-weight: bold;
    font-size: large;
    font-style: italic;
}

.teampic-container {
    display: flex;
    justify-content: center;
    flex-wrap: wrap;
    gap: 30px;
    padding: 20px;
}

/* .teampic {
    background-color: #685752;
    padding: 15px;
    text-align: center;
    width: 120px;
    border-radius: 10px;
} */

.teampic img {
    width: 120px;
    object-fit: cover;
}
```

Details of web services and code

1. Authentication Web Service for Administrators

```
// Admin Login
router.post("/admin-login", function (req, res) {
  const loginTime = getBangkokDateTime();
  const { admin_id, password } = req.body;

  if (!admin_id) {
    console.log("No admin_id in session.");
    return res.status(400).json({ success: false, message: "Missing credentials" });
  }

  // Get the first name and last name from the Admin_info table where that admin_id is that Login.
  const adminInfoSql = `SELECT * FROM Admin_info WHERE admin_id = ?`;
  connection.query(adminInfoSql, [admin_id], (err, adminResults) => {
    if (err) {
      console.error("Error fetching admin info:", err);
      return res.status(500).json({ success: false, message: "Server error" });
    }
    if (adminResults.length === 0) {
      console.log("Admin ID not found");
      return res.status(404).json({ success: false, message: "Admin not found" });
    }

    const adminName = `${adminResults[0].First_Name} ${adminResults[0].Last_Name}`;

    // Verify admin credentials
    const sql = `SELECT * FROM Admin_login WHERE admin_id = ? AND Password = ?`;
    connection.query(sql, [admin_id, password], (error, results) => {
      if (error) {
        console.error("Login check error:", error);
        return res.status(500).json({ success: false, message: "Server error" });
      }

      if (results.length > 0) {
        req.session.admin_id = admin_id; // Store admin ID in session
        // Insert admin Login data
        const loginsql = `
          INSERT INTO Admin_login (admin_id, Password, Login_Time)
          VALUES (?, ?, ?)
        `;
        connection.query(loginsql, [admin_id, password, loginTime], (err) => {
          if (err) {
            console.error("Insert login failed:", err);
          } else {
            console.log(`[LOGIN] by Admin: ${adminName} at ${loginTime}`);
          }
        });

        res.json({ success: true, message: "Login successful", redirectUrl: "/admin-page" });
      } else {
        return res.status(401).json({ success: false, message: "Invalid Admin ID or Password" });
      }
    });
  });
});
```

```

// Admin Logout
router.post('/admin-logout', function (req, res) {
  const logoutTime = getBangkokDateTime();
  const admin_id = req.session.admin_id;

  // Check if admin is Logged in
  if (!admin_id) {
    console.log("No admin_id in session.");
    return res.status(400).json({ success: false, message: "No admin session" });
  }

  // Get admin name for Logging
  const getAdminNameSql = `SELECT First_Name, Last_Name FROM Admin_info WHERE admin_id = ?`;
  connection.query(getAdminNameSql, [admin_id], (error, results) => {
    if (error) {
      console.error("Error fetching admin name:", error);
      return res.redirect('/');
    }

    const adminName = results.length > 0 ? `${results[0].First_Name} ${results[0].Last_Name}` : 'Unknown Admin';
    console.log(`[LOGOUT] by Admin: ${adminName} at ${logoutTime}`);

    // Get Latest Login time to update in database
    const latestLoginSql = `SELECT MAX(login_time) AS latest_login_time FROM Admin_login WHERE admin_id = ?`;
    connection.query(latestLoginSql, [admin_id], (error, loginResults) => {
      if (error || !loginResults[0].latest_login_time) {
        console.error("Error fetching latest login time:", error || 'No login found');
        return res.status(500).json({ success: false, message: 'Logout failed' });
      }
      const latestLoginTime = loginResults[0].latest_login_time;
      // Update Logout time in Admin_login table
      const updateLogSql = `
        UPDATE Admin_login
        SET logout_time = ?
        WHERE admin_id = ? AND login_time = ?
      `;
      connection.query(updateLogSql, [logoutTime, admin_id, latestLoginTime], (error) => {
        if (error) {
          console.error("Logout log update failed:", error);
        }
        // Destroy session and clear cookie
        req.session.destroy((err) => {
          if (err) {
            console.error('Session destroy error:', err);
            return res.status(500).json({ success: false, message: 'Logout failed' });
          }
          res.clearCookie('connect.sid');
          res.json({ success: true, message: 'Logged out successfully' });
        });
      });
    });
  });
});

```

2. Product/Service Search and Details

```
// Search Product
//based on optional criteria (keyword, iPhone model, color, collection, price range)
router.get('/search-api', async (req, res) => {
  const {
    keyword,
    model,
    color,
    collection,
    'price-min': priceMin,
    'price-max': priceMax
  } = req.query;

  let sql = 'SELECT * FROM Product WHERE 1=1';
  const params = [];

  if (keyword) {
    sql += ' AND Product_Name LIKE ?';
    params.push(`%${keyword}%`);
  }

  if (model) {
    sql += ' AND Iphone_Model LIKE ?';
    params.push(`%${model}%`);
  }

  if (color) {
    sql += ' AND Color LIKE ?';
    params.push(`%${color}%`);
  }

  if (collection) {
    sql += ' AND Collection LIKE ?';
    params.push(`%${collection}%`);
  }

  if (priceMin && priceMax) {
    sql += ' AND Price BETWEEN ? AND ?';
    params.push(Number(priceMin), Number(priceMax));
  } else if (priceMin) {
    sql += ' AND Price >=?';
    params.push(Number(priceMin));
  } else if (priceMax) {
    sql += ' AND Price <=?';
    params.push(Number(priceMax));
  }

  connection.query(sql, params, async (err, results) => {
    if (err) {
      console.error('Search query failed:', err);
      return res.status(500).json({ error: 'Query failed' });
    }

    const seenNames = new Set();
    const mimeType = 'image/jpeg';

    // Prepare results with Base64 images and remove duplicate products
    const processedResults = await Promise.all(results.map(async (product) => {
```

```

    if (product.Product_Img && !product.Product_Img.startsWith('data:image')) {
      product.Product_Img = `data:${mimeType};base64,${product.Product_Img}`;
    }
    return product;
  }));
}

const uniqueResults = processedResults.filter(product => {
  if (seenNames.has(product.Product_Name)) {
    return false;
  }
  seenNames.add(product.Product_Name);
  return true;
});

res.json(uniqueResults);
});
});

```

```

// Views product
router.get("/api/product/:id", async (req, res) => {
  const productId = req.params.id;
  const sql = "SELECT * FROM Product WHERE Product_ID = ?";

  connection.query(sql, [productId], async (err, results) => {
    if (err) {
      console.error("Query error:", err);
      return res.status(500).json({ error: "Internal Server Error" });
    }

    if (results.length === 0) {
      return res.status(404).json({ error: "Product not found" });
    }

    const product = results[0];

    // Check if the product image exists and convert to Base64 format
    if (product.Product_Img && !product.Product_Img.startsWith('data:image')) {
      try {
        const mimeType = 'image/jpeg';
        product.Product_Img = `data:${mimeType};base64,${product.Product_Img}`;
      } catch (err) {
        console.error('Error reading image file:', err);
        product.Product_Img = null; // Set to null if there's an issue
      }
    }
    res.json(product);
  });
});

```

3. Product/Service management web service for administrators

```
// Add new product (admin only)
router.post('/add-product', async (req, res) => {
  const {
    product_id,
    product_name,
    description,
    color,
    price,
    stock,
    collection,
    iphone_model,
    product_image_base64
  } = req.body;

  let imageData = null;
  // Validate Base64 image format
  if (product_image_base64) {
    const regex = /^data:image\/(jpeg|jpg|png|gif);base64,/;
    if (regex.test(product_image_base64)) {
      imageData = product_image_base64;
    } else {
      return res.status(400).json({ success: false, message: 'Invalid image format' });
    }
  }

  // Check if product ID already exists
  const checkSql = 'SELECT COUNT(*) AS count FROM Product WHERE Product_ID = ?';
  connection.query(checkSql, [product_id], (err, results) => {
    if (err) {
      console.error('Error checking product ID:', err);
      return res.status(500).json({ success: false, message: 'Error checking product ID' });
    }

    if (results[0].count > 0) {
      return res.status(400).json({ success: false, message: 'Product ID already exists' });
    }

    // Insert new product into database
    const sql = `
      INSERT INTO Product
      (Product_ID, Product_Name, Description, Color, Price, Stock_Quantity, Collection, Iphone_Model,
      Product_Img)
      VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
    `;

    connection.query(
      sql,
      [product_id, product_name, description, color, price, stock, collection, iphone_model, imageData],
      (err, result) => {
        if (err) {
          console.error('Error inserting product:', err);
          return res.status(500).json({ success: false, message: 'Error inserting product' });
        }
        console.log(`Add new product : [${product_id}] Product name: ${product_name}, iPhone model: ${iphone_model}, Collection: ${collection}, Color: ${color}`);
      }
    );
  });

  res.json({ success: true, message: 'Product added successfully!', redirect: '/admin-page' });
}
```

```
        }
    );
});
});
```

```
// Edit product (admin only)
router.put('/edit-product', async (req, res) => {
  console.log('Received edit-product request:', req.body);

  const {
    product_id,
    product_name,
    description,
    price,
    color,
    collection,
    iphone_model,
    stock,
    product_image_base64
  } = req.body;

  if (!product_id) {
    return res.status(400).json({ success: false, message: 'Product ID is required' });
  }

  let imageData = product_image_base64 || null;
  // Validate Base64 image format
  if (imageData) {
    const regex = /^data:image\/(jpeg|jpg|png|gif);base64,/;
    if (!regex.test(imageData)) {
      return res.status(400).json({ success: false, message: 'Invalid image format' });
    }
  }

  // Update product in database
  const sql = `
    UPDATE Product
    SET Product_Name = ?, Description = ?, Price = ?, Color = ?, Collection = ?, Iphone_Model = ?,
    Stock_Quantity = ?, Product_Img = ?
    WHERE Product_ID = ?
  `;
  console.log('Executing SQL:', sql, 'Params:', [product_name, description, price, color, collection, iphone_model, stock, imageData, product_id]);

  connection.query(
    sql,
    [product_name, description, price, color, collection, iphone_model, stock, imageData, product_id],
    (err, result) => {
      if (err) {
        console.error('Error updating product:', err);
        return res.status(500).json({ success: false, message: 'Error updating product', details: err.message });
      }
      if (result.affectedRows === 0) {
        return res.status(404).json({ success: false, message: 'Product not found' });
      }
    }
  );
});
```

```
        console.log(`Updated product: [${product_id}]`);
        res.json({ success: true, message: 'Product updated successfully!' });
    }
);
});
```

```
// Delete product (admin only)
router.delete('/delete-product/:product_id', isAdminLoggedIn, (req, res) => {
  const { product_id } = req.params;
  const query = `DELETE FROM Product WHERE Product_ID = ?`;
  connection.query(query, [product_id], (err, result) => {
    if (err) {
      console.error('Error deleting product:', err);
      return res.status(500).json({ success: false, message: 'Database error', details: err.message });
    }
    if (result.affectedRows === 0) {
      return res.status(404).json({ success: false, message: 'Product not found' });
    }
    console.log(`Delete product : [${product_id}]`);
    res.json({ success: true, message: 'Product deleted successfully' });
  });
});
```

The testing results of web services using Postman

1. Authentication Web Service for Administrators

TESTING ADMIN LOGIN

Test Case 1 : Admin login

Method: POST

URL: http://localhost:4000/admin-login

Body: raw JSON

```
{  
  "admin_id": "AD67093",  
  "password": "Mantrajennie444"  
}
```

The screenshot shows the Postman application interface. At the top, there's a header bar with 'Getting started' and a 'POST http://localhost:4000/i...' button. To the right is a dropdown for 'No environment'. Below the header, the URL 'http://localhost:4000/admin-login' is entered in the 'HTTP' field. On the right, there are 'Save' and 'Send' buttons. The main area has tabs for 'Params', 'Authorization', 'Headers (9)', 'Body' (which is selected), 'Scripts', and 'Settings'. Under 'Body', the 'raw' option is selected, and the JSON payload is displayed:

```
1 {  
2   "admin_id": "AD67093",  
3   "password": "Mantrajennie444"  
4 }
```

At the bottom, the response details are shown: '200 OK', '32 ms', '583 B', and a copy icon. Below this, there are tabs for 'Body', 'Cookies (1)', 'Headers (11)', 'Test Results', and a refresh icon. The 'Body' tab shows the JSON response:

```
1 {  
2   "success": true,  
3   "message": "Login successful",  
4   "redirectUrl": "/admin-page"  
5 }
```

TESTING ADMIN LOGOUT

Test Case 2 : Admin logout

Method: POST

URL: http://localhost:4000/admin-logout

Body: raw JSON

```
{  
  "admin_id": "AD67093",  
  "password": "Mantrajennie444"  
}
```

The screenshot shows the Postman application interface. At the top, there is a header bar with 'Getting started' and 'POST http://localhost:4000/'. Below the header, the URL 'http://localhost:4000/admin-logout' is entered in the address bar. The main workspace has tabs for 'Params', 'Authorization', 'Headers (9)', **Body**, 'Scripts', and 'Settings'. The 'Body' tab is selected and set to 'raw' mode with 'JSON' selected. The request body is a JSON object:

```
1 {  
2   "admin_id": "AD67093",  
3   "password": "Mantrajennie444"  
4 }
```

At the bottom of the workspace, there are tabs for 'Body', 'Cookies (1)', 'Headers (11)', 'Test Results', and a refresh icon. To the right, the response status is shown as '200 OK' with a green background, followed by '18 ms', '466 B', and a globe icon. There are also icons for copy, share, and search.

2. Product/Service Search and Details

TESTING SEARCH PRODUCT

Test Case 1 : No-Criteria Search

Method: GET

URL: <http://localhost:4000/api/products>

Test Case 2 : Criteria Search - Search by Color "Pink"

Method: GET

URL: <http://localhost:4000/search-api?color=Pink>

Test Case 3 : Criteria Search - Search by all criteria

Method: GET

URL:

<http://localhost:4000/search-api?keyword=Banana&model=14&color=Yellow&collection=Plain&price-min=500&price-max=1000>

TEST VIEW PRODUCT DETAILS

Test Case 1 : View product details : Grey Hamster case

Method: GET

URL: <http://localhost:4000/api/product/24SMGH1401>

Test Case 2 : View product details : Sky Blue case

Method: GET

URL: <http://localhost:4000/api/product/25PLSB1501>

3. Product/Service management web service for administrators

** MAKE SURE YOU ARE LOGGED IN FIRST **

We collect images as Base64. It's too long to include in comments, so we provide a sample Base64 string for testing.

TESTING INSERT PRODUCT

Test Case 1 : Insert Green Forest

Method: POST

URL: <http://localhost:4000/add-product>

Body: raw JSON

```
{  
    "product_id": "25PLGF1401",  
    "product_name": "Green Forest",  
    "description": "Protect your phone with the Green Forest Phone Case, designed with vibrant, eco-friendly materials. Featuring a unique forest design, it provides durability, style, and protection for your device.",  
    "color": "Green",  
    "price": 999.00,  
    "stock": 50,  
    "collection": "Plain",  
    "iphone_model": "iPhone 14",  
    "product_image_base64":  
        "data:image/jpg;base64,/9j/4AAQSkZJRgABAgAAAQABAAD"  
}
```

The screenshot shows the Postman application interface. At the top, there is a header bar with 'Getting started' and 'POST http://localhost:4000/'. Below the header, the URL 'http://localhost:4000/add-product' is entered in the address bar. To the right of the address bar are 'Save' and 'Send' buttons. Underneath the address bar, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body' (which is selected), 'Scripts', 'Settings', 'Cookies', and 'Beautify'. The 'Body' tab has several options: 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (which is selected), 'binary', 'GraphQL', and 'JSON'. The JSON content is displayed in a code editor-like area with line numbers from 1 to 12. Lines 1 through 11 represent the JSON object, and line 12 represents the closing brace. Below the JSON content, there is a preview section showing the response status '200 OK', response time '9 ms', response size '422 B', and a copy button. At the bottom, there are tabs for 'Body', 'Cookies (1)', 'Headers (10)', 'Test Results', and a refresh icon. The 'Body' tab is selected, showing the JSON response: { "success": true, "message": "Product added successfully!", "redirect": "/admin-page" }.

Test Case 2 : Insert Red Lion

Method: POST

URL: <http://localhost:4000/add-product>

Body: raw JSON

{

```
"product_id": "24SMRL1501",
"product_name": "Red Lion",
"description": "Unleash boldness with the Red Lion Phone Case. Designed for strength and style, this case offers rugged protection with a fierce lion emblem, perfect for those who dare to stand out.",
"color": "Red",
"price": 1800.00,
"stock": 70,
"collection": "Samote",
"iphone_model": "iPhone 15",
"product_image_base64":
"data:image/jpg;base64,/9j/4AAQSkZJRgABAgAAAQABAAD/"
}
```

The screenshot shows the Postman application interface. At the top, the URL is `http://localhost:4000/add-product`. The method is set to `POST`. The response status is `200 OK`, time taken is `8 ms`, and size is `422 B`. The response body is:

```
{ "success": true, "message": "Product added successfully!", "redirect": "/admin-page" }
```

TESTING EDIT PRODUCT

Test Case 1 : Update New Price

Method: PUT

URL: http://localhost:4000/edit-product

Body: raw JSON

```
{  
    "product_id": "25PLGF1401",  
    "product_name": "Green Forest",  
    "description": "Protect your phone with the Green Forest Phone Case, designed with vibrant, eco-friendly materials. Featuring a unique forest design, it provides durability, style, and protection for your device.",  
    "color": "Green",  
    "price": 1999.00,  
    "stock": 50,  
    "collection": "Plain",  
    "iphone_model": "iPhone 14",  
    "Product_image_base64":  
        "data:image/jpg;base64,/9j/4AAQSkZJRgABAgAAAQABAAD"  
}
```

The screenshot shows the Postman application interface. At the top, there's a header bar with 'Getting started' and a 'PUT http://localhost:4000/edit-product' button. To the right, there are environment dropdowns and a 'Save' button.

The main area has a 'PUT' method selected and the URL 'http://localhost:4000/edit-product'. Below this, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body (raw)', 'Scripts', and 'Settings'. The 'Body (raw)' tab is active, showing the JSON payload from the code block above. There are also 'Cookies' and 'Beautify' buttons.

The JSON payload is displayed in a code editor-like format:

```
1 {  
2     "product_id": "25PLGF1401",  
3     "product_name": "Green Forest",  
4     "description": "Protect your phone with the Green Forest Phone Case, designed with vibrant, eco-friendly materials. Featuring a unique forest design, it provides durability, style, and protection for your device.",  
5     "color": "Green",  
6     "price": 1999.00,  
7     "stock": 50,  
8     "collection": "Plain",  
9     "iphone_model": "iPhone 14",  
10    "Product_image_base64": "data:image/jpg;base64,/9j/4AAQSkZJRgABAgAAAQABAAD"  
11 }  
12
```

At the bottom, there are tabs for 'Body', 'Cookies (1)', 'Headers (10)', 'Test Results', and a refresh icon. The 'Test Results' tab is active, showing a green '200 OK' status with a response time of 7 ms, a response size of 399 B, and a link icon. The response body is shown in a code editor-like format:

```
1 {  
2     "success": true,  
3     "message": "Product updated successfully!"  
4 }
```

Test Case 2 : Update Stock

Method: PUT

URL: http://localhost:4000/edit-product

Body: raw JSON

```
{  
  "product_id": "24SMRL1501",  
  "product_name": "Red Lion",  
  "description": "Unleash boldness with the Red Lion Phone Case. Designed for strength  
and style, this case offers rugged protection with a fierce lion emblem, perfect for those  
who dare to stand out.",  
  "color": "Red",  
  "price": 1800.00,  
  "stock": 100,  
  "collection": "Samote",  
  "iphone_model": "iPhone 15",  
  "product_image_base64":  
  "data:image/jpg;base64,/9j/4AAQSkZJRgABAgAAAQABAAD/"  
}
```

The screenshot shows the Postman application interface. At the top, it displays a 'PUT' method and the URL 'http://localhost:4000/edit-product'. The 'Body' tab is selected, showing the raw JSON payload from the previous code block. The 'Send' button is highlighted in blue. Below the request, the response section shows a '200 OK' status with a response time of 7 ms, a response size of 399 B, and a copy icon. The response body is displayed in JSON format:

```
{  
  "success": true,  
  "message": "Product updated successfully!"  
}
```

TESTING DELETE PRODUCT

Test Case 1 : Delete product 25PLGF1401

Method: DELETE

URL: <http://localhost:4000/delete-product/25PLGF1401>

The screenshot shows the Postman application interface. At the top, there's a header bar with 'Getting started' and 'DEL http://localhost:4000/d'. On the right, it says 'No environment' with a dropdown arrow. Below the header, the URL 'http://localhost:4000/delete-product/25PLGF1401' is entered. To the right of the URL is a 'Save' button and a dropdown menu. A large blue 'Send' button is at the bottom right of the main input area. The main body of the interface has tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Scripts', and 'Settings'. The 'Body' tab is selected and has a radio button for 'raw' which is checked, and another for 'JSON' which is also checked. Below these tabs is a code editor containing the following JSON:

```
1 {  
2   "success": true,  
3   "message": "Product deleted successfully"  
4 }
```

At the bottom of the interface, there are tabs for 'Body', 'Cookies (1)', 'Headers (10)', 'Test Results', and a refresh icon. To the right of these tabs, the response status '200 OK' is displayed along with performance metrics: 5 ms, 398 B, and a globe icon. There are also icons for copy, search, and refresh.

Test Case 2 : Delete product 24SMRL1501

Method: DELETE

URL: <http://localhost:4000/delete-product/24SMRL1501>

This screenshot shows the same Postman interface as the previous one, but for a different test case. The URL is 'http://localhost:4000/delete-product/24SMRL1501'. The response status is '200 OK' with a response time of 6 ms and a response size of 398 B. The JSON response body is identical to the first one:

```
1 {  
2   "success": true,  
3   "message": "Product deleted successfully"  
4 }
```

References

- [1] “Urban Sophistication,” *Urban Sophistication*, 2015.
<https://urbansophistication.com/pages/about-us> (accessed Feb. 26, 2025).
- [2] Casetify, “Show Your Colors | CASETiFY,” *CASETiFY*, 2020.
https://www.casetify.com/?utm_campaign=228232&irpid=44240&irmpname=AB_C%20Consulting&sharedid=&utm_medium=smarterchaos&utm_source=affiliate&clickid=RIAWQ-RotxyPRFgXPfSdJy7nUkswgrRM%3A0r5RA0 (accessed Feb. 26, 2025).