Guided Project

Problem Statement

As we continue building the **ECommerce Application**, the next phase focuses on integrating **JavaScript** into our web pages. You will enhance the **interactivity** of the application by implementing **sign-in authentication**, performing **form data validation**, and **dynamically updating HTML tables** based on user input.

1. Sign-In Page

As the sign-in page is given in the source code,

- **Create** a file named **authentication.js** inside the **JS folder** and **link it** to the corresponding **HTML file** using the `<script>` tag.
- **Fetch the username and password** entered in the form when the user **submits** it.
- **Validate** that the username is "**great**" and the password is "**learning**" to allow access.
- If the credentials are **correct**, use location object to **redirect** to the admin page in the **same tab** and **prevent back navigation** to the login page.
- If the credentials are **incorrect**, **display an error message** asking the user to try again.
- **Allow 3 attempts** for the user to log in. After 3 attempts, do not allow the user to sign in and show a message why that is the case.

2. Add Product

- Create a file named **`add_product.js`** inside the **JS folder** and **link it** to the corresponding **HTML file** using the `<script>` tag.
- **Capture the form data** when the user **clicks Submit**.
- If the **Category** field is **empty**, show an **alert** prompting the user to fill this field.
- After capturing the data, **create an object** named **newProduct** with the following properties: **pname, price, units, desc, inStock, category, and prodimg**.
- **Reset the form** after the object is created.
- Use **URLSearchParams(object).toString()** to convert the object into a **query string**.
- **Redirect** the user to **list_products.html** and **pass the query string** along with the URL using **window.location.href**.

3. List Products

Go through the given list_foodItems.js file, you will see an Array of Objects.

- Link this file to the corresponding **HTML file** using the `<script>` tag.
- In this file, **create a function** named **addProduct** that takes two parameters — a **product** and a **number** (used as the serial number).

- Use **DOM manipulation** and **template strings** to **build a table row** that aligns with the table headers in **list_products.html**, and **append it to the <tbody>** of the table.
- Create another function called **loadAllProducts** that will **loop through the array** of products and, for each item, **call addProduct** with the item and its serial number.
- Use **URLSearchParams(window.location.search)** to **retrieve the query string**, and convert it into an object using **Object.fromEntries(params.entries())**.
- **Add this object** to the existing **array of products**, and **display it** in the table by calling **addFoodItem** with the new object.