Little Shop: Project Documentation

Overview

Little Shop is a dynamic restaurant menu web application designed to provide users with an interactive shopping experience. It allows users to browse menu items, add them to a cart, manage their virtual balance, and complete purchases. The application focuses on simplicity, usability, and responsiveness utilizing JavaScript for functionality and Bootstrap for styling.

Features and Functionality

Core Features

1. **Menu Display**
    - A comprehensive list of menu items with images, descriptions, and prices.
    - Items are styled with Bootstrap for a visually appealing layout.
    - Clicking the **Add to Cart** button below each menu item adds the items to the cart.
2. **Cart Management**
    - Items in the cart are listed with their prices and a remove option.
    - When there is no list, some advertising sentences are dynamically displayed. They disappear as soon as an item is added.
3. **Balance Management**
    - Users start with a virtual balance of $100.
    - The balance decreases as items are added to the cart and increases when items are removed.
    - A **Reset Balance** button allows users to reset their balance to $100.
4. **Checkout Functionality**
    - The checkout page displays all selected items and the total price.
    - Provides a clear summary before completing the purchase.
    - Discount codes (which are shown below the checkout page) can be applied before checking out.

Additional Features

- **About Page**
    - An informative page detailing the concept behind Little Shop.
- **Responsive Design**

    - The application is fully responsive and works seamlessly across all screen sizes.
- **Navigation**
    - A navigation bar with links to Home, About, Cart, and Checkout pages.

How JavaScript Enhances the User Experience

Interactivity

- **Dynamic Cart Updates**
  JavaScript updates the cart in real time, displaying added items, their prices, and the updated total without refreshing the page.
- **Visual Enhancement**
  JavaScript enables dynamic sentences to appear on the Cart and Checkout pages when the cart is empty. In this way, visual attractiveness is enhanced when the user checks their cart.
- **Balance Management**
  JavaScript uses localStorage to save the user's balance, ensuring it remains accurate even after the page reloads.

Usability

- **Event Listeners**
  Buttons, such as "Add to Cart" and "Reset Balance," use event listeners to handle user interactions, making the application responsive and intuitive.
- **Error Prevention**
  JavaScript prevents users from overspending by ensuring the balance cannot fall below zero when adding items to the cart.

Challenges Faced During the Project

- The first challenge is unfamiliarity with bootstrap elements. I had to research a lot and watch the instructor's video repeatedly before the project was completed.
- Second is using the complicated JavaScript commands. The hardest command for me is adding the **Remove** button on the Cart page. Even with the help of the "Hamburger Page" tutorial, it took a good amount of time to configure it.
- The third challenge is caused by the **User Balance**. Since the cart is cleared after an order is confirmed, all costs are reset to $0, which makes the balance return to $100. I used localStorage to solve this. Then, I saved the balance as well as cart items to the local storage, making my website smoother than ever.

Plans for Additional Features or Backend Integration

I understand that it is not possible for the current me to add the following features, but I believe that they will make my website a usable commercial web page.

Additional Features

1. **User Account System**
    - Enable user registration and login functionality.
    - Allow users to save their cart and balance information in personalized accounts.
2. **Enhanced Checkout Page**
    - Integrate payment gateways (e.g., PayPal or Stripe) for real-world transactions.

Backend Integration

1. **Database Integration**
    - Replace localStorage with a database (e.g., MySQL or MongoDB) for centralized data storage.
    - Store menu items, user accounts, and cart details on the server.
2. **Order Management System**
    - Build an admin dashboard for managing menu items, tracking orders, and monitoring user activity.
3. **Real-Time Updates**
    - Use WebSocket or Firebase for real-time cart updates across multiple devices logged into the same account.