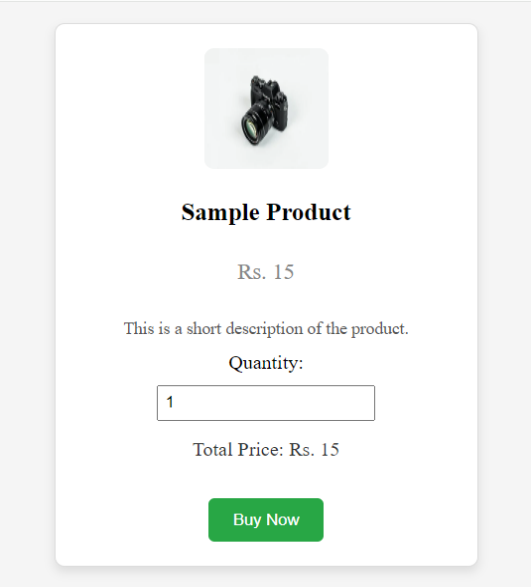
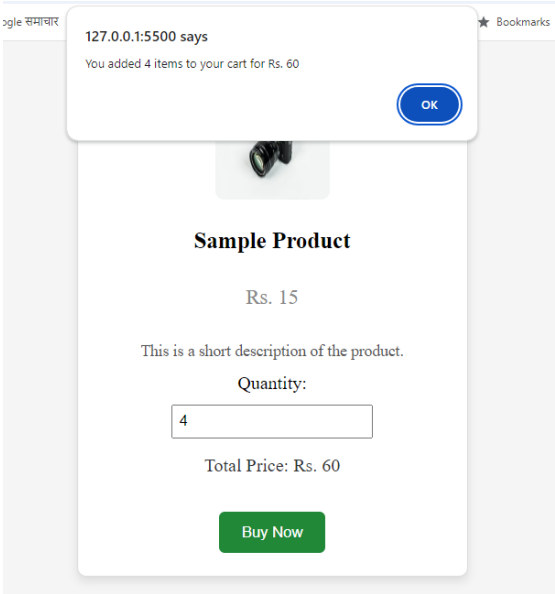
Total Marks: 100

Instructions: Design each section of the code step-by-step according to the guidelines below. Each section focuses on different parts of HTML, CSS, and JavaScript functionality. After completion of the code create a new GitHub Repository named (FSD-Test) and push the code in the repository.





Section A:

Product Card Layout *(20 Marks)*

Create the HTML structure for a **Product Card** to display the following details for a product:

1. **Product Image**: Add an image placeholder. (5 marks)
2. **Product Name and Price**: Use appropriate tags to show the product's name and price. (5 marks)
3. **Description**: Add a short description of the product. (5 marks)
4. **Buy Now Button**: Add a button with the text "Buy Now." (5 marks)

Section B:

Styling the Product Card *(20 Marks)*

Apply CSS styling to the **Product Card** created in HTML:

1. **Image Styling**: Set a fixed width and height for the product image and apply border-radius to make it slightly rounded. (5 marks)
2. **Button Styling**: Style the "Buy Now" button with a background color, padding, and a hover effect that changes the button color. (5 marks)
3. **Product Card Layout**: Use Flexbox to arrange the card elements neatly and align the product name, price, and button vertically. (10 marks)

Section C:

Product Quantity and Price Calculator *(40 Marks)*

For the **Product Card** created above, use JavaScript to add functionality:

- **Quantity Input**: Add an input field for the user to enter the quantity of the product. (5 marks)
- **Total Price Calculation**: Use JavaScript to calculate the total price based on quantity (e.g., `totalPrice = quantity * price`) and display it dynamically below the price. (10 marks)
- **Add to Cart Confirmation**: When the "Buy Now" button is clicked, display an alert showing the quantity and total price added to the cart (e.g., "You added 3 items to your cart for Rs.45"). (25 marks)

Section D:

Submission Guidelines for GitHub: [20 marks]

- Upload your output in Microsoft Form.
- Submit your code in a GitHub repository [Create a separate folder]

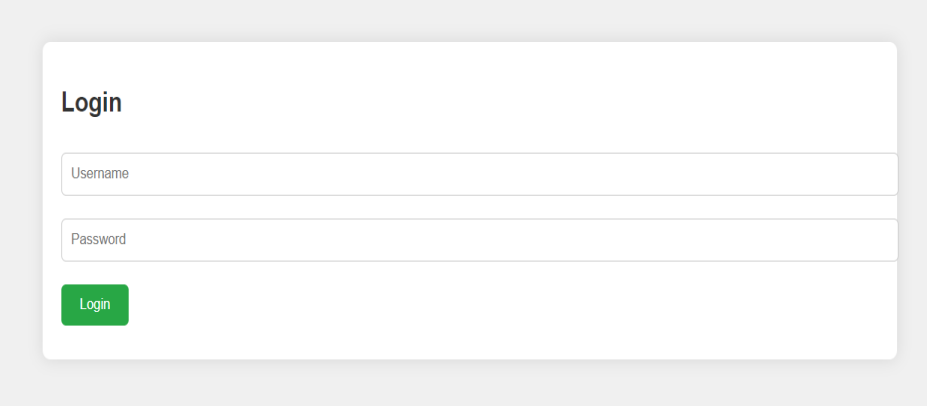
Total Marks: 100

Instructions: Design each section of the code step-by-step according to the guidelines below. Each section focuses on different parts of HTML, CSS, and JavaScript functionality in the given code. After completion of the code create a new GitHub Repository named (FSD-Test) and push the code in that repository.

Section A: (20 Marks)

Container and Panel Layout

- Create a `<div>` with the class `container` to hold the main content of the page. Inside it, add two main sections: the "Login Panel" and the "Dashboard" with respective IDs `login-panel` and `dashboard`. *(05 Marks)*
- Set up `login-panel` with a heading, input fields for username and password, and a button for login. *(05 Marks)*
- Define `dashboard` with a heading, placeholder content for user details, buttons, a dropdown for sorting, and an unordered list for displaying users. *(10 Marks)*



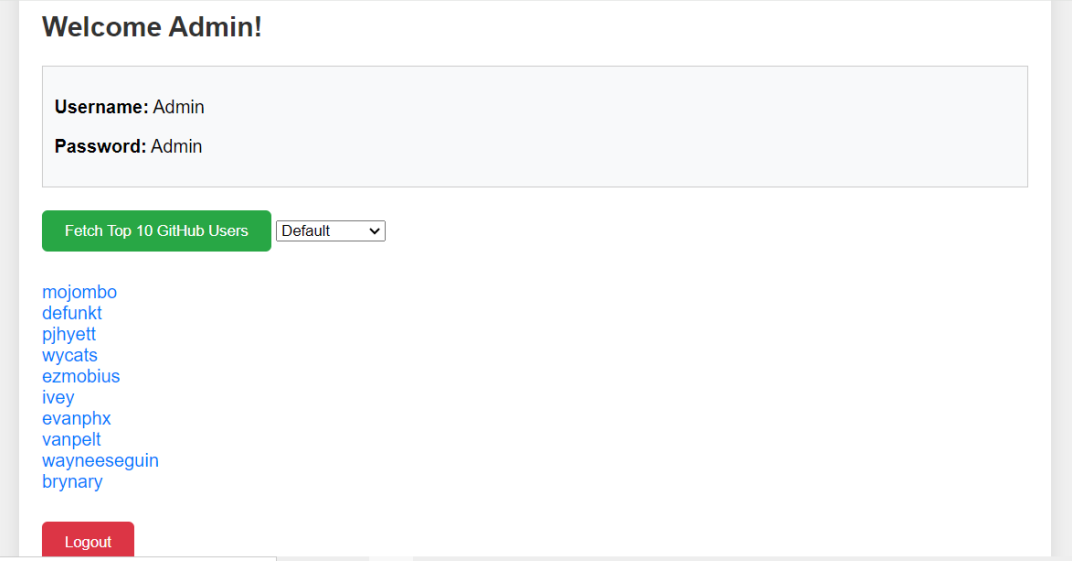
Section B: (20 Marks)

Form and Button Styling

- Style the input fields to be full width, with padding, a solid border, and rounded corners. Add margin space below each input field.
- Style the `button` class with padding, background color (`#28a745`), and white text. Add hover effects for buttons, as well as specific styles for `logout-button` using a red background color (`#dc3545`) and a separate hover color. *(10 Marks)*

Additional Styles

- Style the `.user-details` section with a light background (`#f8f9fa`), padding, and a border.
- Style the unordered list to remove bullets, padding, and add link styling for `ul li a`, including hover effects. *(10 Marks)*



Section C: (40 Marks)

Login Functionality

- Create a `login` function that retrieves values from the username and password fields.
- Validate if the username and password are both "admin." If valid, hide the `login-panel` and display the `dashboard`; otherwise, show an alert with "Invalid login details." *(05 Marks)*

Logout Functionality

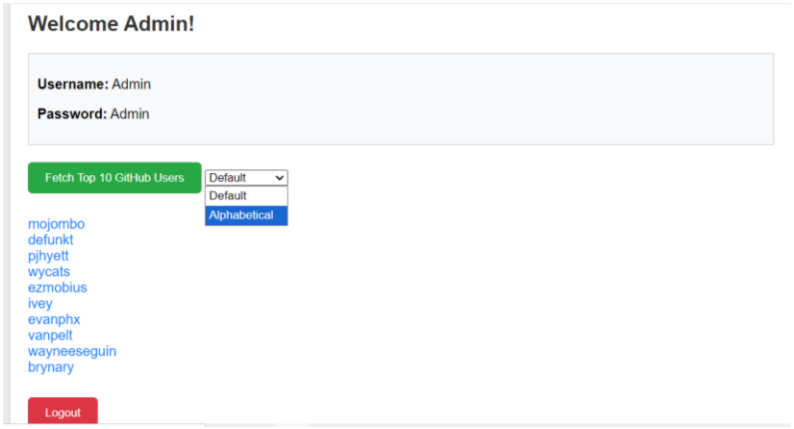
- Write a `logout` function to reset the form, hide the `dashboard`, and display the `login-panel`. *(05 Marks)*

GitHub User Fetching

- Write an `async` function `fetchGitHubUsers` that fetches user data from the GitHub API (`https://api.github.com/users`). Retrieve the top 10 users and save them in an array.
- Call `displayUsers` after fetching to display the users in `userList`. *(10 Marks)*

User List Display

- Write a function `displayUsers` to display users in the unordered list `userList`. For each user,create a list item with an anchor tag that links to the user's GitHub profile. *(10 Marks)*



Sorting Users

- Implement a `sortUsers` function that sorts users alphabetically by their GitHub username when the "Alphabetical" option is selected from the dropdown. *(10 Marks)*

Section D:

Submission Guidelines for GitHub: [20 marks]

- Upload your output in Microsoft Form.
- Submit your code in a GitHub repository [Create a separate folder]