Single Page Application

Objective:

Create a single-page website for a product line, like the one developed in CSD 1103 (you may use the same product line). This website should include a shopping cart feature, fetch content dynamically, and be stored and published on GitHub using GitHub Pages.

Purpose:

This assignment is designed to help you practice the following skills essential for a front-end developer:

- Developing a single-page application without relying on frameworks like React, Angular, or Vue.
- Utilizing GitHub for version control and hosting.
- Dynamically fetching and displaying data from an external source.
- Implementing a shopping cart with local storage to manage the user's selections and pricing calculations.
- Presenting your project and explaining your development process and choices.

Instructions:

1. Set Up GitHub Repository:

- Create a new repository on GitHub for this project.
- Clone the repository to your local machine.

2. HTML Structure:

- Create a single HTML file that will serve as your entire website. This file should include sections for your product line, shopping cart, and other necessary content.

3. Styling and Layout:

- Use CSS to style your website. Ensure the design is user-friendly and responsive.

4. Fetching Content:

- Store your product data as a JSON file (or another format if you like) in your GitHub Pages.
- Use JavaScript to fetch this data and display it dynamically on your website.

5. Shopping Cart Functionality:

- Allow users to select items to buy from your product line.
- Calculate prices, including tax.
- Store the cart information in local storage so that it persists across sessions.

6. Publishing:

- Push your code to GitHub.
- Set up GitHub Pages for your repository and ensure your website is accessible through the GitHub Pages URL.
- Excel

7. Presentation:

- Prepare a ~5-minute presentation to demonstrate your website.
- Discuss the following points:
  - Overview of the project and its features.
  - Your development process and any challenges you encountered.
  - Key decisions you made in terms of design and implementation.
  - A brief walkthrough of your code and how it achieves the required functionality.
- Online, via teams
  - Be prepared to present 2nd last class (August 8th)
  - Spill over is (August 15th)
  - Volunteers first, random order after

Technical Requirements:

- **HTML:** One single HTML file for the entire website.
- **CSS:** External CSS for styling.
- **JavaScript:** Use plain JavaScript or jQuery (no other frameworks) for fetching data and implementing the shopping cart.
- **Data Storage:** file(s) stored on GitHub Pages for product data.

- **Local Storage:** Utilize local storage to maintain the state of the shopping cart.

Submission:

- Submit the GitHub repository link and pages link.
- Ensure your GitHub Pages site is live and accessible.

Additional Notes:

- Focus on code readability and maintainability.
- Ensure that all links, images, and resources are correctly linked and functional.
- Test your website thoroughly before submission to avoid any broken functionality.

Grading Rubric:

Criteria	Excellent (90-100%)	Good (75-89%)	Satisfactory (60-74%)	Needs Improvement (below 60%)	Weight
HTML Structure	Clear, well-organized, and semantic HTML structure	Mostly clear and organized HTML structure	HTML structure is somewhat clear, with minor issues	HTML structure is unclear and disorganized	**5**
CSS Styling and Layout	Attractive, user-friendly, and fully responsive design	Generally attractive and mostly responsive design	Adequate design, some responsiveness issues	Poor design and lack of responsiveness	**5**
JavaScript Functionality	Efficient, well-organized code, implements all features	Functional code, implements most features	Code functions, but some features are incomplete	Code is inefficient, buggy, and missing key features	**10**
Data Fetching	Successfully fetches and displays data dynamically	Mostly successful data fetching, minor issues	Basic data fetching, some issues in display	Data fetching is unsuccessful or incomplete	**10**
Shopping Cart Implementation	Fully functional cart, accurate calculations, persists	Mostly functional cart, minor calculation issues	Cart works but has several issues with calculations	Cart is not functional or lacks key features	**10**
Presentation	Clear, concise, and engaging; covers all key points	Clear and concise; covers most key points	Covers main points but lacks clarity or detail	Unclear, disorganized, and missing key points	**5**
Deployment on GitHub Pages	Successfully deployed, accessible, and fully functional	Deployed and accessible, minor functionality issues	Deployment successful but some functionality issues	Deployment unsuccessful or major functionality issues	**5**
Criteria	**Excellent (90-100%)**	**Good (75-89%)**	**Satisfactory (60-74%)**	**Needs Improvement (below 60%)**	**50**