Demarcus Johnson

Dr.Stanfield

Project 1: WebDesign

Overview

In this project, I developed a responsive web application using React. The application fetches and displays real-time data from two endpoints provided by the National Institute of Standards and Technology (NIST), specifically related to manufacturing processes.

Project Structure

- App.js: Contains the main component of the application.
- axios:Utilized for making HTTP requests to fetch data from the NIST endpoints.

Features

- **Real-time Data Update:The application automatically refreshes the displayed data every second.
- **Current Data Display: Displays the current manufacturing data obtained from the NIST API.
- **Sample Data Display: Presents the sample manufacturing data retrieved from the NIST API.
- **Error Handling: Logs errors to the console if there's any issue fetching data from the API.

Dependencies

- React: A JavaScript library for building user interfaces.
- axios: A promise-based HTTP client for browser and node.js.

- useState: React hook for managing state in functional components.

- useEffect: React hook for performing side effects in functional components.

Usage

1. Clone the repository.

2. Install dependencies using `npm install`.

3. Run the application using `npm start`.

4. Open the application in a web browser.

Additional Notes

- The application continuously retrieves data from the NIST API every second using `setInterval`.

- Fetched data is displayed in separate sections for current and sample data.

- Error handling is implemented to capture and log any errors occurring during data retrieval.

Testing Documentation

As part of the development process for Project 1, I conducted testing to ensure the functionality and reliability of the application. Below are the details of the test cases conducted and their results:

Test Case 1: Verify Data Display

Description: I verified that the application correctly displays the fetched data from the NIST API.

Test Steps:

* Opened the application in a web browser.

* Observed the "Current Data" section.

* Verified that the current manufacturing data is displayed.

* Observed the "Sample Data" section.

* Verified that the sample manufacturing data is displayed.

Expected Result: The application should display the current and sample manufacturing data fetched from the NIST API.

Actual Result: Upon testing, the application successfully displayed the current and sample manufacturing data as expected.

Test Case 2: Verify Settings Functionality

Description: I verified the functionality of the settings section, including the ability to update URLs and the data refresh interval.

Test Steps:

* Opened the application in a web browser.

* Navigated to the "Settings" section.

* Updated the current URL to a different valid URL.

* Updated the sample URL to a different valid URL.

* Updated the data refresh interval to a different value.

* Saved the changes.

* Observed the application behavior.

Expected Result: The application should update the URLs and refresh interval accordingly, fetching data from the new URLs and refreshing the data at the specified interval.

Actual Result: Upon testing, the application successfully updated the URLs and refresh interval, fetching data from the new URLs and refreshing the data at the specified interval.