Daily Assignment on FETCH API

Imagine you are developing a blogging platform where users can create and publish their posts. Additionally, users can manage their tasks with a todo list feature. Your task is to implement the frontend functionality for displaying a list of blog posts and a todo list retrieved from remote servers using Fetch API. Ensure proper error handling in case the servers are unreachable or return unexpected data. Finally, refactor your code using a suitable JavaScript design pattern to improve code organization and scalability.

Objective:

Gain practical experience in working with Fetch APIs, handling JSON data, manipulating the DOM, implementing error handling using JavaScript, and understanding JavaScript design patterns through practical application.

Instructions:

Set Up:

- Access the provided sample public APIs: JSONPlaceholder API for Posts and JSONPlaceholder API for Todos.
- Familiarize yourself with the available endpoints and data structures.

Task 1: Fetch and Display Posts:

- Use the Fetch API to asynchronously retrieve blog post data from the provided public API (`/posts` endpoint).
- Handle the JSON response and dynamically display the fetched posts on the web page using DOM manipulation.

Task 2: Fetch and Display Todos:

- Use the Fetch API to asynchronously retrieve todo list data from the provided public API (`/todos` endpoint).
- Handle the JSON response and dynamically display the fetched todo list items on the web page using DOM manipulation.

Task 3: Error Handling:

- Implement error handling mechanisms using JavaScript to manage potential errors during the API requests for both posts and todos.

- Display relevant error messages to the user in case of failed requests or unexpected responses.

Task 4: JavaScript Design Patterns:

- Choose a simple JavaScript design pattern (e.g., Module Pattern or Revealing Module Pattern).
- Refactor your code to incorporate the chosen design pattern for better structure and maintainability.

Task 5: Integration and Testing:

- Integrate Fetch API calls, error handling, and design pattern implementation into a cohesive application.
- Thoroughly test the functionality of your application to ensure proper data retrieval, error handling, and adherence to the chosen design pattern.

Submission Guidelines:

1. Document your approach, code implementation, and any challenges faced during the assignment.
2. Ensure your code is well-commented and follows best practices for readability and maintainability.
3. Submit your completed assignment within the allotted time frame according to the submission guidelines provided.
4. Provide a brief reflection on your learning experience and any insights gained from completing the assignment.

Sample Public APIs:

- Posts API Endpoint: JSONPlaceholder API for Posts
- Todos API Endpoint: JSONPlaceholder API for Todos
- Documentation: JSONPlaceholder API Documentation