Fetch API Assignment Documentation

1. Approach

The objective of this assignment was to demonstrate the use of the Fetch API, asynchronous JavaScript (async/await), and the Revealing Module Pattern for structuring the code. The approach focused on modularity, readability, and maintainability. The project was divided into three key modules:

1. ApiModule — Responsible for making API calls using Fetch API with proper timeout handling and error validation.
2. UIModule — Managed all DOM-related operations including rendering data, loaders, and error messages.
3. AppController — Orchestrated interactions between modules and handled user actions such as refreshing and filtering.

Additionally, the code was organized into separate files (index.html, styles.css, app.js) to maintain a clean separation of concerns. The UI was kept minimal and responsive, following a card-based layout for displaying posts and todos.

2. Code Implementation

The code was implemented using the Revealing Module Pattern, which enhances encapsulation and code readability. Each module exposes only the necessary public methods while keeping internal details hidden.

- index.html: Structured the webpage layout and linked external JS and CSS files.
- styles.css: Defined a clean, modern look using CSS variables, flexbox, and grid.
- app.js: Included three self-contained modules for API, UI, and control logic.

Error handling included timeout management (using AbortController), HTTP status validation, and safe rendering using HTML escaping to prevent XSS vulnerabilities.

3. Challenges Faced

- Implementing proper error handling for both network and logical errors.
- Managing timeouts and ensuring the UI provides feedback during slow network calls.
- Structuring the app cleanly within a single script while maintaining modularity.
- Ensuring that the UI remains responsive and visually organized when dealing with large API datasets.

4. Reflection

This assignment strengthened my understanding of asynchronous programming, modular architecture, and clean coding practices in JavaScript. I learned how to handle real-world scenarios such as network failures, data validation, and user experience during data

fetching.

Working with the Fetch API and applying the Revealing Module Pattern improved my confidence in building scalable front-end applications. Additionally, documenting and commenting the code helped me focus on clarity, making the project easier to understand and maintain for others.

5. Submission Notes

- The code is fully commented and organized according to best practices.
- Hosted Website Link : https://blogtrackr.netlify.app/
- The documentation and final ZIP file (fetch_api_assignment_commented.zip) are included as part of the submission package.