**Cyber Form Back-End Project Explanation**

The **Cyber Form Back-End Project** is a web-based cybersecurity tool designed to provide users with practical functionalities such as password strength checking, encryption, and decryption. The goal of this project was to create an interactive and educational platform that allows users to explore fundamental cybersecurity concepts safely, while demonstrating professional web development and server-side programming practices.

This project is built on **Node.js** with **Express.js** as the server framework. We incorporated key security and performance features such as **Helmet** for HTTP header security, **CORS** to manage cross-origin requests, and **rate limiting** to prevent abuse from excessive requests. The encryption and decryption functionalities leverage the **AES-256-GCM** algorithm, ensuring a robust, industry-standard cryptographic method for securing data.

One of the key aspects of this project is the management of sensitive information. All master encryption keys are stored in a separate **environment file** (.env), which is never committed to the repository. This ensures that users who clone the project will not have access to the keys used in the demonstration, thereby maintaining security. To make the project functional on their local machine, users are instructed to generate their own cryptographic key and store it in the .env file, along with a secure API token. This approach allows the project to operate without exposing sensitive material while giving users practical experience with server-side encryption.

The project structure is straightforward yet effective. Static HTML, CSS, and JavaScript files for the front-end interface are served directly by Express using express.static(). The server listens for API requests at designated endpoints (/api/encrypt and /api/decrypt) and processes them using Node’s **crypto** module. The encryption and decryption operations are performed entirely on the server side, meaning that user data is securely handled and never exposed in an unsafe manner. This also allowed us to practice handling JSON requests, error management, and asynchronous operations in Node.js.

The password strength checker is another critical component of the project. This feature evaluates a user’s password based on several criteria, including character diversity, length, and entropy, which is calculated based on the number of possible combinations. Users are provided with actionable feedback to help them generate stronger passwords. This serves both an educational purpose and a demonstration of how front-end and back-end systems can interact seamlessly.

Setting up the project locally is simple. Users must install the necessary dependencies using **npm**, create their .env file with a newly generated master key and API token, and run the server with node server.js or using a process manager like **nodemon**. Once the server is running, users can access the front-end pages in their browser and fully utilize the encryption, decryption, and password evaluation functionalities.

This project demonstrates the practical application of cybersecurity principles in web development while adhering to best practices for security, modularity, and maintainability. It provides a solid foundation for understanding how encrypted communication and password security can be implemented in real-world scenarios without compromising sensitive information.

**Contact Information**

For questions, feedback, or collaboration opportunities related to the Cyber Form projects, feel free to reach out or connect via GitHub.

You can explore more of my cybersecurity and web development projects here:

**GitHub:** https://github.com/MarioHUBB

**Email:** [dumbamari02@gmail.com](mailto:dumbamari02@gmail.com)

**Instagram:** @modestmarioo

I welcome contributions, suggestions, and discussions about improving the platform or developing new educational cybersecurity tools.