NAME-Aryan Chowdhury
Education-Jadavpur University(2021-2025)
Class-BE-IT-UGIII

REPORT FOR FAST-API PROJECT

Introduction:

The MongoDB schema design was approached with simplicity and flexibility in mind. Two collections, 'asset' and 'metrics', were created to store documents representing assets and performance metrics. Each document contains various fields providing comprehensive information about assets and their associated metrics.

API Functionality:

Comprehensive CRUD operations for assets and performance metrics were provided through well-organized RESTful endpoints. Authentication was implemented using HTTP Basic authentication. Endpoints for fetching individual assets and performance metrics by their IDs were included. Aggregate data endpoints allowed for the calculation of average downtime, total maintenance costs, and identification of assets with high failure rates.

Challenges Faced and Solutions Implemented:

Challenges:

1. Authentication: The challenge was to implement the authentication securely.
2. Error Handling: Robust error handling, particularly in asynchronous code, presented a challenge.

Solutions:

1. Authentication: HTTP Basic authentication with a secure username and password combination ('admin/admin') simplified authentication.(only for one user both the username and the password were given “**admin**”)
2. Error Handling: Effective error handling was achieved by implementing custom exception handlers and using appropriate HTTP status codes for different error scenarios.

Self-Assessment Against Evaluation Criteria:

Criteria:

1. Schema Design: The MongoDB schema design was deemed appropriate for the application's requirements, providing flexibility and scalability.
2. API Functionality: The API offered comprehensive functionality for managing assets and performance metrics, including CRUD operations and aggregate data retrieval.
3. Authentication and Security: Basic authentication was implemented securely, though there was potential for enhancement with more secure authentication methods.

4. Error Handling: The application featured robust error handling, offering informative error messages and appropriate HTTP status codes.

Overall Assessment:

The project demonstrated a solid understanding of MongoDB schema design, API development, and best practices for building scalable and efficient web applications. With further refinement in authentication and security aspects, the application would be well-equipped for production use.

Conclusion:

The project showcased effective MongoDB schema design and API functionality. Challenges were addressed through appropriate solutions, resulting in a robust and scalable application. Continued improvements in authentication and security would further enhance its suitability for production deployment.