SUNSTONE

“TO-DO-LISTS”

Task - 4

TO-DO-LISTS

A to-do list project typically aims to help users organize tasks, manage priorities, and increase productivity by providing a user-friendly platform for creating, editing, and tracking tasks or activities. Features often include task categorization, due dates, reminders, and sometimes collaboration options for team projects. The primary goal is to streamline task management and enhance efficiency in personal or professional settings.

LMSUsername	Name	Batch
au910020104001	AATHAVAJAYANTH A.S	CC2
au910020104303	DHARSHINI A	CC2
au910020104308	PRAVEEN M	CC2

Task-4

Evaluation Metric:

User Authentication and Authorization:

Assess the API's methods for user authentication and authorization to ensure secure access control.

Task Management Functions:

Evaluate the completeness and flexibility of task-related functionalities, including creating, updating, deleting, and organizing tasks..

Data Synchronization:

Assess the API's ability to synchronize tasks across different devices and platforms to ensure a seamless user experience.

Webhooks and Notifications:

Evaluate the availability of webhooks or notification mechanisms to inform users about updates or changes to their tasks in real-time.

Error Handling and Logging:

Evaluate how the API handles errors, and check if it provides detailed error messages and logs for debugging purposes.

Security Measures:

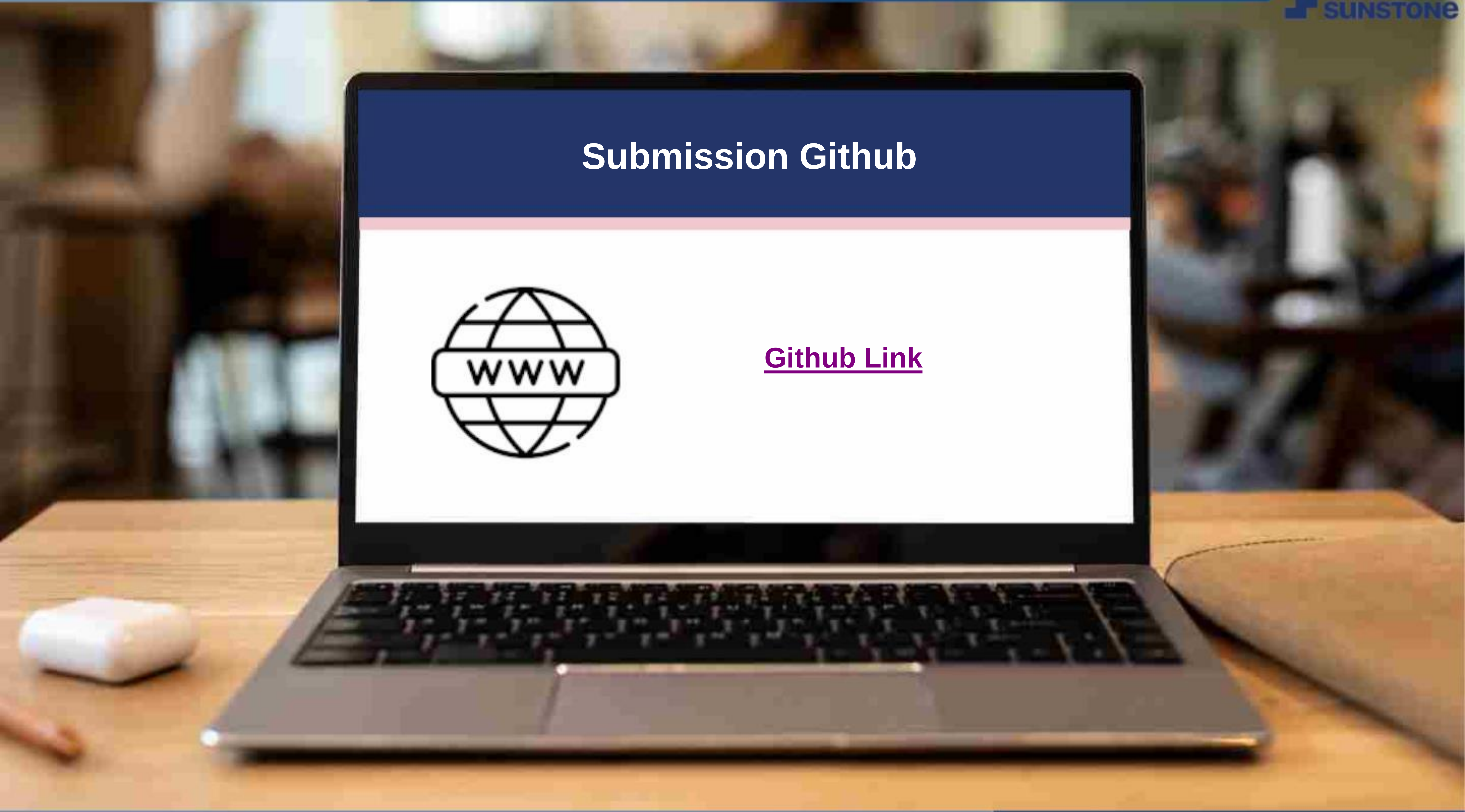
Assess the security features implemented in the API, including encryption, secure connections, and protection against common vulnerabilities.

Step-WiseDescription

- **Define Requirements:**
  -Identify the features you want in your to-do list (e.g., create tasks, complete, delete tasks).
  - Decide on the data structure for tasks
- **Choose Tech Stack:**
  - Select a backend framework (e.g.Springboot).
  - Choose a database (e.g.MYSQL) for storing tasks.
- **Create API Endpoints:**
  Define endpoints for CRUD operations (Create, Read, Update, Delete).
  Example:
    - `POST /tasks` for creating a task.
    - `GET /tasks` for fetching all tasks.
    - `GET /tasks/{id}` for fetching a specific task.
    - `PUT /tasks/{id}` for updating a task.
    - `DELETE /tasks/{id}` for deleting a task.
- **Implement CRUD Operations:**
  - Write functions/handlers for each endpoint to perform CRUD operations.
  - Ensure proper validation and error handling.
- **Frontend Integration:**
  - Develop a frontend (react.js) that interacts with your API.
  - Use frameworks like React, Angular, or Vue for web, or React Native/Flutter for mobile.

Task Summary

Create a feature-rich To-Do List application with a comprehensive API for efficient task management. The API supports essential CRUD operations, allowing users to create, read, update, and delete tasks with attributes such as title, description, due date, and completion status. Optional user authentication enhances security, and thorough error handling ensures robust responses. Detailed documentation facilitates seamless integration, while optional frontend development provides a user-friendly interface. Deployment to a hosting service, scalability considerations, thorough testing, monitoring tools, and adherence to security best practices contribute to a reliable and user-friendly To-Do List solution. User feedback and iterative improvements drive the project towards an enhanced user experience.



Submission Github

Github Link

SUNSTONE

Thank you!