Internship Assessment: Backend Development

Assignment: Develop a RESTful API for a Task Management System

Objective: Assess the intern's ability to design, implement, and document a fully functional RESTful API using backend technologies taught in the curriculum.

Assessment Criteria

1. **Technical Requirements:**
    - Utilize any of the following backend technologies:
        - Node.js
        - Python Flask/Django
        - Java Spring Boot
    - Ensure the API adheres to RESTful principles.
2. **Functional Requirements:**
    - Implement CRUD operations for managing tasks:
        - **Create:** Add a new task to the database.
        - **Read:** Retrieve task(s) from the database. Support both single-task and all-tasks retrieval.
        - **Update:** Modify details of an existing task.
        - **Delete:** Remove a task from the database.
    - Include the following fields for each task:
        - Task ID (auto-generated)
        - Title (string)
        - Description (string)
        - Status (enum: Pending/Completed)
        - Due Date (date)
    - Store task data in a database such as MongoDB or MySQL.

3. **Documentation:**
    - Use Swagger or Postman to document the API endpoints.
    - Ensure documentation includes:
        - Endpoint descriptions.
        - Request and response formats.
        - Example inputs and outputs.
4. **Design and Implementation:**
    - Follow best practices for API design and coding standards.
    - Handle edge cases (e.g., invalid inputs, non-existent task IDs).
    - Implement appropriate HTTP status codes for responses.
    - Use environment variables for sensitive configurations (e.g., database credentials).
5. **Evaluation Metrics:**
    - **Functionality:** Does the API meet the functional requirements?
    - **Code Quality:** Is the code modular, readable, and maintainable?
    - **Error Handling:** Are errors handled gracefully with meaningful messages?
    - **Documentation:** Is the API well-documented and easy to understand?
    - **Performance:** Is the API performant and scalable?
6. **Submission Guidelines:**
    - Provide the source code in a compressed folder.
    - Include a README file with:
        - Instructions to set up and run the API.
        - A list of features implemented.
        - Any challenges faced during development.
    - Share the API documentation link or file (e.g., Swagger URL or Postman collection).
7. **Bonus Points:**
    - Implement authentication (e.g., JWT) to secure the API.
    - Add filtering or sorting functionality for tasks (e.g., filter by status, sort by due date).
    - Use Docker to containerize the application.
    - Write unit tests for critical API endpoints.

8. **Timeline:**
    - Deadline for submission: [Specify Deadline]

Rubric for Grading (Out of 100 Points)

Criteria	**Weightage**
Functionality	30 points
Code Quality	20 points
Documentation	20 points
Error Handling	20 points
Submission Guidelines	5 points
Bonus Features	5 points

Expected Outcome

- A fully functional and documented RESTful API for managing tasks.
- The project serves as a practical demonstration of backend development skills and provides insights into the intern's technical capabilities and problem-solving approach.