SIA101 – System Integration and Architecture 1

FINAL OUTPUT

Final Grading Period

REST API WEBSITE & DOCUMENTATION

Overview

For your final output in System Integration and Architecture, you and a partner will collaborate to build a project that demonstrates your understanding of REST principles, web development, and API integration. This project will be a culmination of the concepts learned in class and should showcase creativity, technical proficiency, and clear documentation.

Objectives:

1. Develop a REST Website with a concept of your choice.
2. Create REST API Endpoints tailored to the services your website offers.
3. Write API Documentation to provide clear guidance on how to interact with your API.

Instructions:

1. Choose Your Partner and Concept

- Form a team of two members.
- Brainstorm and select a concept for your website (e.g., e-commerce, blog platform, booking system, task manager, etc.).
- The concept should include interactive services that will require API support (e.g., CRUD operations like adding and retrieving data).

2. Develop the REST Website

- Create a responsive website using HTML, CSS, JavaScript, or a front-end framework like React or Vue.js.
- The website must include:
    a. Homepage introducing the website and its purpose.
    b. Service Pages that demonstrate at least 3 different services/features provided by the website.
    c. Interactive Components that allow users to make HTTP requests (e.g., forms, buttons, or inputs).
- Technical Guidelines:
    a. Host the website locally or deploy it using services like Netlify, GitHub Pages, or Heroku.
    b. The website should consume the REST API endpoints you create in Step 3.

3. **Design REST API Endpoints**
    - Build a REST API using Python Flask, Node.js Express, Django REST Framework, or another backend framework.
    - Define and implement endpoints that align with the services provided by your website.
    - Example Endpoints:
        a. GET /items – Retrieve all items.
        b. POST /items – Add a new item.
        c. PUT /items/{id} – Update an item.
        d. DELETE /items/{id} – Delete an item.

    Technical Guidelines:

    - Include CRUD operations (Create, Read, Update, Delete) in your API.
    - Ensure the API adheres to RESTful principles:
    - Use appropriate HTTP verbs (GET, POST, PUT, DELETE).
    - Use clear and consistent URL patterns.
    - Return proper HTTP status codes.

4. **Create API Documentation**

    Your documentation must clearly explain how to use the API. Use tools like Swagger, Postman, or write the documentation manually in a well-formatted file.

    Include the following details:

    A. Overview: Brief description of the API and its purpose.
    B. Endpoint Descriptions:
        - URL
        - HTTP Method
        - Required parameters (query, path, or body).
        - Example Request/Response (JSON format).
    C. Authentication (if any):
        - Describe any required API keys or tokens.
    D. Error Handling:
        - List possible errors with their HTTP status codes and messages.

5. **Submission**

    You will submit the final project in the following format:

    A. GitHub Repository:
        - Upload all your project files (both website and REST API) to a shared GitHub repository.
        - Your repository should be well-organized and include the following:
            - Website files (HTML, CSS, JavaScript, etc.)
            - REST API code (Python, JavaScript, etc., depending on the framework used).

- API Documentation file named API_DOC.md.
- A README.md file containing:
    - A project description.
    - Instructions for running the website and API locally.
    - Link to the live/deployed website (if applicable).
    - Any other relevant setup details.

B. *Live Deployment (Preferably GitHub)*
- Deploy your website using platforms like Netlify, GitHub Pages, or Heroku.
- Provide the deployment link in your README.md file.

C. *Presentation:*
- Be ready to present your project during class on **January 08, 2025**.
- Your presentation should include:
    - A live demo of the website and its functionality.
    - Explanation of how the REST API integrates with your website.
    - Overview of the API documentation and its usage.

D. *Deadline:*
- Submit your project GitHub repository link by **January 06, 2025.**