Proposal for RESTful Web Service to Manage Employee Data Using Spring Boot and HPE GreenLake Cloud Platform

This project proposal outlines a RESTful web service to manage employee data with essential CRUD operations, using Spring Boot as the framework and HPE GreenLake as the private cloud platform for secure and scalable data storage.

This service will provide endpoints for retrieving, adding, updating, and deleting employee records, ensuring seamless integration with client applications.

Requirements

1. **Endpoints and HTTP Methods:**
    a. **GET `/employees`: Returns the complete list of employees in JSON format, including `first_name`, `last_name`, `employee_id`, `email`, and `title`.**
    b. **POST `/employees`: Accepts a JSON payload to create a new employee entry. Required fields include `first_name`, `last_name`, `employee_id`, and `email`.**
    c. **DELETE `/employees/{employee_id}`: Deletes an employee record based on the unique `employee_id`.**
    d. **PUT `/employees/{employee_id}`: Updates the employee data for the specified `employee_id`. Only fields provided in the JSON payload will be updated.**

2. **Data Model:**
    a. **Each employee record will include:**
        i. **`first_name` (String, required)**
        ii. **`last_name` (String, required)**
        iii. **`employee_id` (Integer, unique identifier, required)**
        iv. **`email` (String, required, validated for proper email format)**
        v. **`title` (String, optional)**

    b. **The `employee_id` serves as a primary key, guaranteeing uniqueness across employee records.**

3. **Data Storage with HPE GreenLake:**
   - ☑ **HPE GreenLake will be used as the primary cloud platform, providing a secure and scalable environment for data storage and management. This private cloud solution aligns with data security and compliance requirements, enabling encryption for both data at rest and in transit.**
   - ☑ **Using HPE GreenLake's data services, we will implement secure, high-performance storage, with scalability to handle growing datasets without compromising response times.**

4. **Implementation Using Spring Boot:**

   Framework: Spring Boot will be used to build the RESTful web service due to its robustness, ease of setup, and comprehensive support for RESTful architecture.

   Spring Data JPA: This will be used for database interactions, enabling efficient querying and object-relational mapping.

   Validation and Exception Handling: Spring Boot's validation framework will enforce data integrity by validating fields such as `email` and `employee_id` uniqueness, and provide clear error responses.