Interview Task: RESTful API for Task Management

Overview

Your task is to develop a RESTful API using a Python framework of your choice (e.g., Django, FastAPI) that manages a simple task management system. This system will allow users to create, read, update, and delete tasks. Each task will have a title, description, creation date, due date, and status (e.g., pending, completed).

Requirements

- **Framework**: Choose either FastAPI or Django (preferably) for building the API.
- **Database**: SQLite
- **Task Model**: Define a task model with the following fields: title, description, creation date, due date, and status. The status field should only allow predefined values (e.g., pending, in progress, completed).
- **CRUD Operations**: Implement endpoints to create, read, update, and delete tasks. Ensure you validate input data and handle possible errors gracefully.
- **Filtering and Sorting**: Implement functionality to filter tasks by status and due date, as well as to sort tasks by creation date or due date in both ascending and descending orders.

Unique Challenge

Smart Task Suggestions: Implement an endpoint that provides "smart" task suggestions. This feature should analyze the existing tasks and suggest new tasks based on the following criteria:

- Frequency of similar task titles or descriptions (suggest variations or related tasks).
- Tasks that are commonly marked as completed together or in sequences (suggest tasks that follow up on completed tasks).

For instance, if a user frequently creates tasks related to "Project A Review", the system might suggest creating tasks for "Project A Follow-up Meeting" or "Project A Finalization".

Evaluation Criteria:

- **Functionality**: The API meets all functional requirements, including CRUD operations, filtering, sorting, and the smart task suggestions feature.
- **Code Quality**: The code is well-structured, clean, and follows best practices for the chosen Python framework.
- **Creativity and Problem-Solving**: The implementation and effectiveness of the smart task suggestions feature.
- **Documentation**: The API is well-documented, with clear instructions on how to set it up and call its endpoints. Include examples of request and response bodies.

Submission Instructions:

- Provide the source code for your API in a public Git repository.
- Include a README file with setup instructions, endpoint documentation, and examples of how to use the API.
- Document any significant design decisions or assumptions you made during development.