Task Management API - Project Documentation

Overview

The Task Management API is a web application that provides a backend system for managing tasks within an organization. It allows users to register, log in, create tasks, assign tasks, update task status, and manage task dependencies. The API is designed for use by managers and regular users, where managers have additional privileges.

Purpose

The purpose of this API is to facilitate task management in an efficient way, using a simple RESTful interface that can be integrated into various front-end systems or applications.

Features

- **User Authentication**: Register and log in functionalities with security considerations.
- **Task Management**: Creation, retrieval, updating, and of tasks.
- **Task Dependency Management**: Ability to add and manage dependencies among tasks.

Technologies and Stack

- **Programming Language**: Python
- **Framework**: Flask
- **Authentication**: Flask-JWT-Extended for JWT-based authentication
- **Database**: SQLite
- **Password Hashing**: Werkzeug for secure password handling

How to Set Up and Run the Application

1. **Environment Setup**:
    - Install Python 3.8 or above.
2. **Dependencies Installation**:
    - Install required dependencies via pip.
3. **Database Initialization**:
    - Run the initialization script to set up the database schema: `"from app.models import init_db; init_db()"`.
4. **Running the Server**:
    - Start the Flask server: `python run.py`.
    - The server will run on `http://127.0.0.1:5000`.

5. **Using Docker** :
    - Build the Docker image: `docker build -t task_management_api .`
    - Run the container: `docker run -d -p 5000:5000 task_management_api`
6. **Postman:**
    - `Import Task Management API.postman_collection in postman`

API Endpoints and Usage

- **POST /auth/register**: Register a new user.
    - **Body**: `{"username": "user", "password": "pass", "is_manager": true/false}`
- **POST /auth/login**: Log in and retrieve an access token.
    - **Body**: `{"username": "user", "password": "pass"}`
- **POST /api/tasks**: Create a new task (Managers only).
    - **Headers**: `Authorization: Bearer <JWT_TOKEN>`
    - **Body**: `{"title": "Task Title", "description": "Task Description", "due_date": "YYYY-MM-DD", "assignee_id": 1}`
- **GET /api/tasks**: Retrieve all tasks for managers, or assigned tasks for users.
    - **Headers**: `Authorization: Bearer <JWT_TOKEN>`
- **PUT /api/tasks/<task_id>**: Update a task (full update by managers, status update by assigned users).
    - **Headers**: `Authorization: Bearer <JWT_TOKEN>`
    - **Body**: `{...task details...}`
- **POST /api/tasks/<task_id>/dependencies**: Add dependencies to a task (Managers only).
    - **Headers**: `Authorization: Bearer <JWT_TOKEN>`
    - **Body**: `{"dependent_task_id": 2}`