Python Developer Challenge

Challenge Title: Python Backend Development with FastAPI for User Management and To-Do Application with API Integration.

Challenge Description:

Objective: Develop a backend application using FastAPI to provide APIs for user management and to-do management. Integrate the application with a database and implement authentication and CRUD operations.

Version Control Setup:

- Initialize a version control repository (e.g., Git) for the project.
- Set up a remote repository on a platform like GitHub, GitLab, or Bitbucket.

Modules:

- **Application Setup**:
  - Set up a FastAPI project for building the backend application.
- **Database Integration**:
  - Integrate a database (e.g., SQLite or PostgreSQL) for storing user and to-do data.
  - Create database tables and models for users and to-do items.
- **User Authentication API**:
  - Implement API endpoints for user authentication, including registration, login, and password reset.
  - Use JWT (JSON Web Tokens) for secure authentication.
- **User Profile API**:
  - Develop API endpoints for users to view and update their profiles.
  - Implement CRUD operations for user profiles.
- **To-Do Management API**:
  - Create API endpoints for to-do management, including creating, listing, updating, and deleting to-do items.

    - Implement CRUD operations for to-do items.
- **Authentication Middleware**:
    - Implement middleware to authenticate and authorize users using JWT for protected endpoints.

- **Error Handling**:
    - Develop error handling mechanisms to provide informative responses for various scenarios, such as invalid authentication attempts and API errors.
- **Testing**:
    - Write unit tests and integration tests for the API endpoints to ensure functionality and security.

- **Documentation**:
    - Create API documentation using tools like Swagger or ReDoc to describe the available endpoints, request/response formats, and authentication methods.

- **Completion Criteria**:
    - A functional FastAPI backend application with API endpoints for user management (authentication, profile) and to-do management (CRUD operations).
    - Integration of a database for storing user and to-do data, with tables and models defined.
    - Secure user authentication using JWT tokens for registration, login, and password reset.
    - Implementation of CRUD operations for user profiles and to-do items.
    - Authentication middleware that protects endpoints requiring authentication.
    - Effective error handling for informative API responses.
    - Unit tests and integration tests to validate the functionality and security of API endpoints.
    - Comprehensive API documentation describing available endpoints, request/response formats, and authentication methods.