

Task: Implementing JWT Authentication in Your OOP Python Project

Objective: Enhance the security of your existing OOP Python project by integrating JSON Web Token (JWT) authentication.

Instructions:

1. JWT Authentication:

- Implement JWT-based authentication to secure endpoints within your project.
- Create functionality to generate and validate JWTs for user authentication.
- Ensure that JWTs are used to protect sensitive routes and resources in your project.
- Implement token expiration and renewal mechanisms as necessary.

2. Documentation:

- Provide clear and comprehensive documentation on how JWT authentication is integrated into your project.
- Include explanations on generating tokens, validating them, and handling authentication failures.
- Document any configuration settings or environment variables related to JWT.

Requirements:

- Integrate JWT authentication for secure access control within your project.
- Ensure your JWT implementation adheres to security best practices.
- Test the authentication thoroughly to verify that tokens are correctly issued and validated.
- Keep the code clean, well-organized, and follow best practices for secure authentication.

Submit [here](#)