Asynchronous API Wrapper Library

Name :K. Sandeep Kumar

Project Idea

- Build a Python library to connect with the GitHub API
- Use async/await for fast and efficient calls
- Handle limits, errors, and large data automatically
- Package it so other developers can install and use it

How It Works

1. Create a GitHubAPI client class
2. Use aiohttp for async HTTP requests
3. Handle rate limits (pause when API says slow down)
4. Manage multiple pages of results with async generator
5. Return clean, easy-to-use data

Features

- Async methods for GitHub users, repos, and issues
- Automatic handling of pagination
- Rate limit detection and retry support
- Clear error messages with custom exceptions
- Can be installed from PyPI

Skills Learned

- Writing async/await code in Python
- Using aiohttp for API requests
- Handling errors and retries safely
- Designing libraries for other developers
- Packaging and publishing to PyPI
- Writing with type hints and documentation

Example Workflow

1. Install the package from PyPI
2. Import GitHubAPI and create a client
3. Call methods like get_user() or get_repos()
4. The library handles rate limits, errors, and multiple pages
5. Developer gets clean and ready-to-use data

Conclusion

- Professional-grade async library
- Helps developers work with GitHub easily
- Handles real-world API challenges
- Shows skills in async programming, API design, and packaging