

AeroAspire -SDE Intern Training

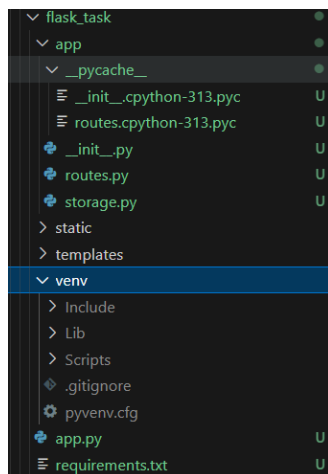
NIRANJAN C N

Week3-Day 1-Oct6

Setup Flask project; define folder structure (app folder, routes, static etc.); basic “Hello world” route, Implement GET /tasks and POST /tasks with dummy in-memory storage; test via Postman or curl

Questions:

1. What is the flow when a HTTP GET request comes to Flask → how Flask handles routing → response.
 - a. When we send a GET request to Flask it checks for the URL we have entered
 - b. It compares the route mentioned with the route present in the route file
 - c. Flask calls the function linked with that particular route
 - d. The response is sent back based on the parameters we mentioned
2. What is WSGI? How Flask sits on WSGI server (development vs production).
 - a. WSGI is Web Server Gateway Interface
 - b. Flask is a WSGI based application
 - c. Flask works inside a WSGI server that directs HTTP requests to Flask
 - d. In development Flask uses a simple WSGI server for development purpose
 - e. In production Flask runs on a robust WSGI server for better performance and reliability
3. How you'd structure a medium size API project (folders, modules).



- a.
 - b. So this would be the basic structure for functioning of Flask
 - c. The venv and pycache is added once we install and run flask respectively
4. What is REST: what makes an endpoint RESTful?
 - a. REST is Representational State Transfer
 - b. It is manner in which work and makes it easy to understand and use
 - c. An endpoint is RESTful by following these :

- i. By using GET,POST,PUT,DELETE for any operation
 - ii. By using valid URLs
 - iii. By ensuring that all the information is provided for a particular request
 - iv. And finally check for proper responses
- 5. What status codes should be returned and when?
 - a. 200 - The request sent is successful
 - b. 201 -The resource is created successfully
 - c. 400-The request sent has invalid data
 - d. 401-Authentication is needed
 - e. 404-The resource does not exist
 - f. 500-An Internal Server error
- 6. How to validate input and handle bad JSON or missing fields.
 - a. Making sure that the client sends JSON data if the type is application/json
 - b. Use request.get_json() to parse JSON body.
 - c. Check if all the required fields are satisfied
 - d. Else send a 400 error code
 - e. Check if the fields have the right type of data and satisfy the conditions
 - f. If any error send respective error codes