AeroAspire

SDE Intern

Goutham V

Week 3 – Day2 (08th October)

Task:

Add endpoints: DELETE /tasks/<id>, PUT /tasks/<id>; optional filtering via query - e.g. ?completed=true

Reflection,

1. What is the difference between path parameters and query parameters?

=Path parameters are a part of the URL itself, usually used to identify a specific resource.

For example:

/tasks/5

Here, 5 is a path parameter that points to a particular task.

On the other hand, query parameters are added to the URL after a "?" symbol and are generally used for filtering or searching.

Example:

/tasks?status=completed

This doesn’t point to one task, but instead filters tasks based on their status.

In short —
Path parameters → identify a specific item.
Query parameters → refine or filter a collection.

2. How to handle errors if the client passes an invalid ID or resource not found?

=If a client tries to access a task that doesn’t exist (for example, /tasks/999), the backend should not crash or return a confusing message.

Instead, we can handle it gracefully using simple checks. For instance, in Flask we can write:

```
task = Task.query.get(task_id)

if not task:

    return jsonify({"error": "Task not found"}), 404
```

This way, the user clearly understands what went wrong, and our API stays stable.
It’s also a good idea to handle invalid IDs or missing fields with proper status codes like 400 (Bad Request) or 404 (Not Found).

3. Describe the flow of updating a resource: client sends JSON → Flask handler → modify data → return response.

=The update process follows a very logical flow:

1. **Client sends a request — usually a PUT or PATCH request to something like /tasks/3 with a JSON body containing the updated data.
Example:**
2. **{**
3. **"status": "completed",**
4. **"description": "Finished writing the report"**
5. **}**
6. **Flask receives the request — the backend reads the JSON data and finds the corresponding task in the database.**
7. **Modify and save — once the task is found, the backend updates only the changed fields and commits the changes using SQLAlchemy.**
8. **Send response — after saving, the backend sends a confirmation message (for example, "Task updated successfully") or returns the updated task in JSON format.**

**So the entire flow is like a conversation between the frontend and backend —
frontend says *"please update this data"*, backend says *"got it, updated successfully"*.**