AeroAspire

SDE Intern

Goutham V

Week 3 – Day3 (09th October)

Task:

Add error handlers (404, 400); validate inputs (e.g. non-empty strings, valid JSON), Enable CORS; from React frontend make calls; test from Postman / browser

Reflection,

1. How do you test error flows (client sends invalid data)?

=I test error flows by sending wrong or missing data through Postman. For example, I try adding a task without a title or using an invalid ID. The API should return a clear message like *"Invalid data"* or *"Task not found"* with a proper status code such as **400** or **404**. This helps check if the backend handles wrong inputs properly.

2. Describe flow of exception in Flask: what happens if an unhandled exception occurs?

=If something goes wrong in Flask and it's not handled, Flask shows an error page or returns a **500 Internal Server Error**.
To avoid this, we usually use **try-except blocks** or **error handlers** to catch exceptions and send user-friendly error messages instead of breaking the app.

3. What is CORS? Why browsers block cross-origin requests; how to configure CORS in Flask.

=CORS (Cross-Origin Resource Sharing) controls which frontend sites can talk to the backend.
Browsers block requests from different origins (like a React app running on localhost:3000 trying to access Flask on localhost:5000) for security reasons.
In Flask, we can fix this easily using the **Flask-CORS** package:

```
from flask_cors import CORS
CORS(app)
```

This allows the React app to connect safely to the Flask API.

4. What is the flow of a fetch/Axios request from React to Flask → response → error handling?

=The React app sends a request (using **fetch** or **Axios**) to the Flask API.
Flask processes it, performs the action (like saving or fetching data), and sends a JSON response back.
React then updates the UI.
If there's an error (like wrong URL or server issue), React catches it and can show a message like *"Something went wrong"* to the user.

5. How to log or debug failed requests.

=I check the Flask terminal or console logs to see what error occurred. In React, I use **console.log** or **Axios error messages** to find the issue. Sometimes I also print variables in Flask using print() to trace what went wrong during a request.