AeroAspire

SDE Intern

Goutham V

Week 3 – Day5 (11th October)

Task:

Bonus: authentication (token-based), logging, testing (unit tests)

1. Explain full request-response cycle including network, HTTP headers.

=When a user performs an action (like submitting a form), the **frontend (React)** sends a **request** to the **backend (Flask)** using HTTP.
This request travels over the **network** with some extra information called **HTTP headers** — for example, content type, authentication token, or browser details.
The Flask server processes the request, interacts with the **database** if needed, and then sends a **response** back to the frontend with a status code (like 200 OK or 404 Not Found).
Finally, the frontend updates the UI based on the response received.

2. What do you need to consider for security (e.g. input validation, sanitization, auth) even if simple app.

=Even for small apps, security is very important. Some key things to consider are:

- **Input validation:** Make sure the data entered by users is valid (e.g., not empty or invalid type).
- **Sanitization:** Clean inputs to prevent harmful data like script tags (to avoid XSS attacks).
- **Authentication:** Use token-based login or API keys to secure endpoints.
- **Error handling:** Don’t show internal errors or stack traces to users.
- **HTTPS:** Always use secure connections in production.

3. How would you monitor errors in production?

=To monitor errors in a live app, we can:

- Use **logging** in Flask to record errors and important events.
- Store logs in a file (like app.log) or send them to monitoring tools.
- Use services like **Sentry** or **New Relic** to track real-time errors and performance issues.
- Regularly review logs to detect repeated problems early and fix them quickly.