AeroAspire -SDE Intern Training

John Nikhil G

Week4-Day3-Oct15

- **Learning Topics:**
    - Filtering / search
    - Query parameters
    - Indexing basics
- **Task**
    - Add filters: by status, due-date
    - Implement search by title
    - Possibly pagination basics
- Questions:

1. How do you design efficient queries for filtering? What is an index; when should it be used?

- To make queries faster, you can narrow down the data by using conditions in your SELECT statement, so the database only fetches what's needed.
- Indexes are like a quick reference or a shortcut that the database can use to locate data without scanning the whole table.
- They are especially useful on columns that you often filter, sort, or join, because they reduce the amount of data scanned.
- Using primary keys is usually the fastest way to retrieve records, so always try to leverage them when possible.

---

2. How does pagination work (offset/limit etc.)?

- Pagination helps split a large dataset into smaller chunks, so you don't overwhelm the user with too many rows at once.
- The limit defines how many rows you want per page, while offset tells the database where to start fetching data.
- It's usually combined with ORDER BY so that the pages are consistent and predictable in their ordering.

---

3. What’s the flow of building these endpoints, receiving query parameters, applying them in SQL / ORM, returning results?

- First, you define endpoints in your backend (like GET, POST, PUT, DELETE) to handle specific actions.
- Users or clients send requests along with parameters, like filters or search terms.
- The backend takes these inputs and converts them into database queries, either using raw SQL or an ORM layer.
- The database executes the query and returns the requested data.
- Finally, the backend formats the response and sends it back to the client, which could be a web app, mobile app, or API tool like Postman or Swagger.