**Lecture 4 Project - Note Manager**

**Part 1**

Here's a simple "Personal Notes Manager" project - a backend API that lets users create, read, update, and delete notes. It's perfect for beginners since it uses just one table but covers essential backend concepts.

Database Table (notes):

* id (primary key)
* title
* content

notes APIs:

* GET /notes
* POST /notes
* PUT /notes/:noteId
* GET /notes/:noteId
* DELETE /notes/:noteId

**Part 2**

Add 2 more columns to the database table

* created\_at
  + the time of the note creation
* updated\_at
  + the time of the note update

**Part 3**

Modify the GET /notes api to support searching by note title or content.

For example, user can search by:

GET /notes?title=test

OR

GET /notes?content=test

**Part 4**

Add another table “authors”. A note can have only 0 or 1 author, while an author can have many notes.

* id (primary key)
* name
* created\_at
* modified\_at

Add the author's CRUD APIs.

* GET /authors
* POST /authors
* PUT /authors/:authorId
* GET /authors/:authorId
* DELETE /authors/:authorId

**Part 5**

Modify the notes APIs to support this new author concept:

* GET /notes: should return author details of a note
* POST /notes should takes author id as part of a note payload
* PUT /notes/:noteId should takes author id as part of a note payload
* GET /notes/:noteId should return author details of a note

**Part 6**

Add error handling to each APIs.