The purpose of this application is for me to record all of notes. These are created by whenever I work on courses in Coursera or Udemy, or complete exercises in Hackerrank or CodeWars.

I currently do all of this in word documents. I want to move away from word documents for several reasons.

1. It is tedious to go through again. Essentially, the default formatting does not look too good on word.
2. I want to review how often I revise stuff. I want to record data whenever I revise notes I have created. This revision needs to be record so I can see some data analysis on the work I have completed.
3. I want to create tests (automatically or not) to help when revising notes. I also want the results recorded. Altogether, this can give me a good idea of whether I have mastered some concepts and how away from mastering it am I.
4. I can group these notes content under a course type structure and have them open for others to benefit from them.
5. It can help me organise my studies, particularly when I am studying similar things (like SQL) but on different platforms.

To get a good description of possible data structures lets look at some examples:

* I am learning React which is a JavaScript library.
* JavaScript is a language needed for Front-end web development. It can also be used for Backend Web development / Full-stack web development / Mobile development.
* Each of those development form part of Software development, separated by platform to an extent.
* Software development can be considered a sub aspect of Software Engineering.

How would we model this data?

JavaScript has various aspects to it, such Basics (variables, control and loop statements, operators, etc), Object Oriented Programming, DOM Manipulation, Data Structures and Algorithms, etc.

React is built on top of both JavaScript and HTML.

Since React has multiple lower-level technologies, it cannot directly connect to them (1 to 1).

We can consider React Development as being a SUBJECT, Front-End Web Development is a CAREER which has many SUBJECTs, one of which is React. A SUBJECT can also be connected to many CAREERs, so this is a many-to-many relationship.

A CAREER can also be a sub-career i.e., a specialization, of another CAREER. Hence, there will be a self-one-to-many relationship.

A CAREER can also be associated with many TECHNOLOGIES. Front-end Development can be associated with HTML, CSS and JAVASCRIPT. This is a one-to-many relationship.

A SUBJECT can also be associated with 1 or more TECHNOLOGIES, another one-to-many relationship.

A SUBJECT can have 1 or more TOPICs, another one-to-many relationship. Also, a TOPIC can be have various sub-TOPICs, representing via a self-one-to-many relationship.

Finally, a TOPIC can have 1 or more NOTEs. Each NOTE can have 1 or more NOTE\_ATTACHMENTS.

Entities are CAREERs, SUBJECTs, TECHNOLOGIES, TOPICs, NOTES.

Note, each of these will have timestamp related fields, as well as user related fields, such as created by and updated by. Also, each entity will have an audit log to record all changes to the data in the tables.

Basic user flow description:

When I login to the application, I can see a basic list of SUBJECTs I am currently working on. When clicking on a SUBJECT, I will see a list of highest-level TOPICs. If I click on a TOPIC, I should see the lower-level TOPICs, and so on, until we finally see the NOTEs.

Here I can click on a NOTE, to view the relevant NOTEs. I can also add a new set of NOTEs.