Report

# CA1

In CA1, we use XML to read and write data which is a sort of file-based system. In CA1 we append the XML file with the new record on the submit button and delete the selected record from the XML on the delete button and this is how CA1 works.

# CA2

In CA2, which is an updated version of the CA1. In CA2 we use Node JS and MongoDB for server-side programming, to making Rest full API.

We actually updating the previous version of CA1 by exchanging the file-based XML approach with the NoSQL database and we are using separate API endpoints to perform CRUD functions.

System call insert record API (which inserts the record into the database) on the submit button provided on the front end and call the get records API (which fetch all the records from the database) on render. And system calls the delete API (which delete the respective record) when the user clicks the delete button after selecting the particular row of the table provided at the web page. These are all the new things we adding the existing CA1.

# Better Approach in My Opinion

According to me, the better approach is CA2 because it is a systematic way to handle things. It also improves the efficiency, modularity, and usability of the system. You can call API endpoints from another system and by this approach, you can easily modify the particular part of the system without disturbing the flow of another part because we made the separate API endpoint to perform each task of the system.

The reason to update the CA1 is:

in XML there is no identifier for record and when we want to get the particular record then we have to traverse XML file which will be much hectic approach and when we want to delete the particular record then we again have to traverse the file line by line, but in CA2 we made the MongoDB database and every document in the collection has its own object id. And by this object id, we can fetch that document directly, is faster than fetching the record from the XML file.

So that’s why CA2 is a better approach than CA1.

# Technologies used and Reason

As you know, we used Node JS and MongoDB at the backend to make the API which interacts with our MongoDB database and jQuery (JavaScript library) at the front end to interact with the API endpoints.