Extras

* Challenges faced
  + Time constraints
    - The exam was given to me last Monday. From Tuesday to Thursday, I participated in the knowledge transfer session conducted for the outsourced team so they can be versed in using UMBRACO 8.
    - Then on Friday, I had a family emergency at home.
    - I was able to work on the programming test just this last weekend. Then I also have other interviews on Monday and Tuesday this week.
  + New in .Net Core
* Should I have more time to work
  + What I would have done is to add class libraries for the data layer, for the class models, and the utilities layer (named common in most projects). This is necessary to separate the concerns, and worked only in the project that would require changes.
  + Create a UI that would connect/consume the API
* Quality and best practices
  + As I have mentioned, the separation of concerns would be a start. This would also help in terms of flexibility, and accessibility as developers only need to access the program that would require changes.
  + Make sure that the code is loosely coupled wherein the objects are not so dependent on each other. Mechanism such as dependency injection helps you accomplish this as you can easily pass objects as parameters. The use of interface helps you include behavior from multiple sources in a class.
  + Implement SOLID principle
* Making the code scalable
  + Write codes that are:
    - Less dependent on using server state sessions in saving data.
    - Instead using alternatives such as caching or even cookies.
    - Have a consistent open and close connection so as to handle the use of network bandwidth.
  + Make class libraries and projects accessible specifically limited to the right people
    - Use sealed classes so these classes can be used only in a particular module
    - In Azure devops, create repositories that are role based. Example all the frontend projects or components are accessible only to frontend developers and vice versa for backend developers.