MSA 2022 Phase 3

Front End Resources

- I would use next.JS as a framework, and host in on vercel, as this would cover off a requirement for the cloud hosting, and the API connection (next as a framework acts as an API as well as serving the frontend content).
- I would attempt to do both the mobile first development as well as UI scalability features together, as they are very closely related. Read this article on responsive layouts by material design. To implement this in material UI, you will want to read this article on breakpoints.
- I would attempt to do the unit testing alongside the storybook requirement, as they go pretty hand in hand. Both requirements are related to component isolation. You can find the storybook getting started here. You will also want the following addons:
  - Accessibility gives you action recording
  - Interactions gives you automation testing functionality
  - Chromatic gives you easy snapshot testing
- I would try to avoid attempting the other advanced features, as they are (generally) quite a bit more difficult, although any good attempt at achieving them will be considered a completed advanced feature, provided that you are able to explain where you think you went wrong, and how you would improve it next time 😃

Back End Resources

- Comprehensive unit testing is the easiest thing on this list. You will want to use: a mocking library, like NSubstitute; a testing library, like NUnit; and a fluent assertion library, like Fluent Assertions.
- Creating an onion structure is also surprisingly easy to accomplish. You will want to have a read through of this article here. For the project structure in the solution, I would recommend:
  - A API project to represent the application layer
  - A services project to represent the service and repository layer
  - A domain project to represent the domain layer
- I would also recommend using MediatR to help keep code both DRY and KISS and to help conform to the principle of inversion of control
- A lot of submissions for phase 2 were already using EF core - those of you who have a head-start (1 down, 2 to go!). Those of you who don't and need help, there's plenty of it going around and lots of people who can potentially help!
- If you want to implement caching, I would use something like Redis.
- There is an old MSA tutorial from 2020/2021 on how to implement a CI/CD pipeline to Azure using GitHub actions. Following this would get you another feature completed.
- I again, would try to avoid most of the other features on this list, unless you are looking for a challenge! As stated earlier, any good attempt at achieving these will also be considered a completed advanced feature.

Data Science Resources

Here is an example neural network implementation that should follow most the basic **requirements from 1 to 4** for Data Science section (using a different dataset and different goal)
https://github.com/rantonkoga/tensorflow-example-2022

Advanced Features (pick at least 2):

1) Hyperparameter Tuning: Feel free to look at this (Think this was a basic requirement in 2019 MSA Phase 2 AI):
https://www.tensorflow.org/tensorboard/hyperparameter_tuning_with_hparams

2) Deploying the model to Azure ML Studio and exposing the model as an API: This was actually a basic requirement in 2021 MSA Phase 2. Regardless of model deployment, feel free to look at
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-tensorflow for model deployment.

3) Make a python function that can convert an image to be usable by the model: I think opencv-python can do this, feel free to find other options. Also this can be integrated with advanced feature 2 as a pre-processing layer.

4) Integrate this with frontend/backend: People with frontend might have an easier time doing this.

PS: Also if you have a dedicated GPU, you have a significant advantage when it comes to training speed.