**Developer Task Reflections**

For the solution I choose to present to the user a dropdown from which they can choose the product they want to see the safety sheet of. For the backend, the api is capable of querying the database using EF core and retrieving the data for each product. There is also an extra table called BackupSheets made to store the binary data of each pdf.

When the request for getting a pdf is sent the backend checks if sending a request to that specific url throws a Http exception. If it does then it will return the binary data of the pdf stored in BackupSheets. If the url is valid then it will check its binary content against the data in BackupSheets and then return it to the user.

As for elements I would like to improve, the front-end is the first one; I would like to be able to detect in the AngularJs controller if the data came from the valid Url or BackupSheets (send an extra parameter saying that in the response) so I could signal to the user that Url was not working. Also if the Url is valid and the data in it is different than what is in the backup I would like to signal that to user as well on the front-end.

Furthermore, I did not have a chance to code the unit tests as they are empty right now and I would have also liked to setup a recurring job in Hangfire that runs every 3 other days using CheckContentChange class to see if the data in the pdfs changed.