# Why choose this solution

I choose ASP.NET core as it is a cross platform technology so it is easier to integrate it with other systems.

In addition, Entity framework core has so many enhancements when it comes to performance than Entity framework and when dealing with documents performance is very important.

Therefore, the solution is:

1. An ASP.Net core web application (BE)

As the application is not complicated I did not separate it in multitier architecture, however I tried to respect the separation of concerns principle in order to make change easier.

1. AngularJs App (FE)

# How To Run:

1. Clone Project from URL:

<https://github.com/MarcoYacoub/ChemiClean.git>

1. Create 2 Databases:
   1. ChemiClean
   2. ChemiCleanTest
2. Open solution as administrator
3. Update connection string in the web project “ChemiClean” in files with the connection string of DB “ChemiClean”:
   1. appsettings.json
   2. appsettings.Development.json
4. From package manager run command update-database
5. Run the script InsertProductsDocuments
6. Repeat steps 4,5,6 for DB ChemiCleanTest instead of “ChemiClean”
7. Update connection string on the “ChemiCleanTest” in files:
   1. ProductRepository
   2. ProductsServiceTest
8. Make sure to update back the connection string in files appsettings to ChemiClean DB connection string
9. Build solution and Run.

Elements to improve:

Issues:

* + 1. The filtration has some issues for example: load more does not work fine when I change search criteria and then click load more it takes the new search criteria even If I did not click search
    2. I fetch the data to the DB in the first time to use the application so the first time to use the application it’s very slow, so maybe do this from an admin page or using a scheduled job instead.
    3. When a User tries to download a file I always try to get the online version if not found I get the local one from the DB (I think this behavior needs to be validated and this depends on how often there are changes on the documents because if there are not that much changes maybe I should check if the last modification date is less than say a month I then get the local version I have)

App:

May be use another technology for the frontend App may be React.js or Angular.

Code Quality:

* + 1. Revisit the queries and algorithms used in order to enhance performance if possible.
    2. Write more comments in code.
    3. Add More Unit tests
    4. Use mock data instead of another DB for testing.