Software Engineering 102

Conceptual Knowledge

1. Which programming language did you use?

2. Which books did you read to be a better software engineer?

3. Which blogs, websites do you follow and what is the frequency?

4. What is an abstract class in C# and for which purpose we use it?

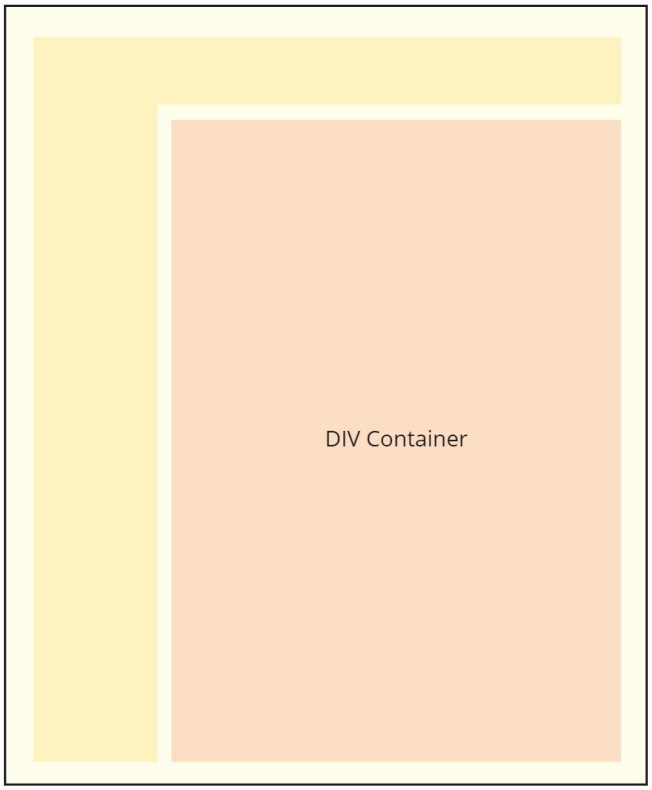
Practical Knowledge

Your are being provided with the following requirements:

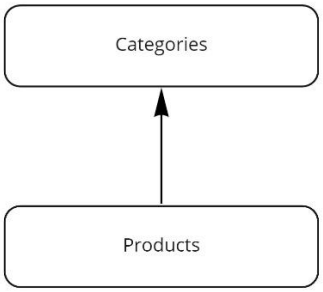
- Visitors of an e-commerce application want to able to see products within a category. As an example, following page is a PLP (product listing page) of Hepsiburada: https://www.hepsiburada.com/laptop-notebook-dizustu-bilgisayarlar-c-98?filtreler=isletimsistemi:Windows%E2%82%AC2010%E2%82%AC20Home,Windows%E2%82%AC2010%E2%82%AC20Pro&siralama=coksatan

- Your team lead tells you that the list of the products should be loaded asynchronously. The page itself will be loaded from cache and a javascript code on you page will request the product list on page onload event and will put the result in to a div control.

  Following drawing shows the sections coming from the cache with the yellow colour tones and the div container where you should put the product list with the light orange colour.

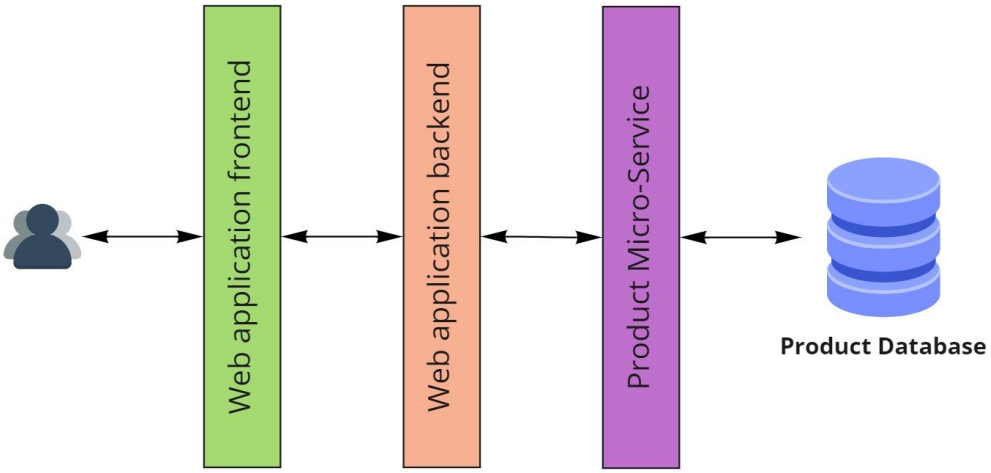


- Till this moment the database did not contain product and category data so you have to create a very simple database with two tables: Products and Categories. You are free to add and use any fields necessary.



Again your team lead tells you to use code-first approach to create the database but if it will take to much of your time you can use db-first approach too.

- Because of the overall architectural strategy the product data should be provided by micro-service which will be responsible for all product related data. So the information flow should be like following:



Here are the details about the layers;

Assuming you will have a Visual Studio solution including two projects:

  - Web application (ASP.NET Web Application project)
  - Micro-service (ASP.NET Web API project)

Web application frontend; (should be in the web application project) which is the Javascript code block on the view will call a controller function in the same application and puts the result (which should be in HTML format) in to the DIV container on the same view.

Web application backend; (should be in the web application project) which is a controller function in the same web application will make a HTTP request to the product microservice. The request to the micro-service should be in JSON format which should be a serialized version of a C# object (**ProductQuery**). The JSON response from the micro-service should be deserialized to a **List<Product> collection.**

As a tip; this requirement says that you have to have following objects:

- Product
- ProductQuery

After getting the response as an object collection, **the controller should render this in a view and send the response as HTML string.**

Product micro-service; which is a ASP.NET Web API project in the same Visual Studio solution should get the JSON query from the backend controller, populate the results by making a query to the product database. The response should be deserializable to **List<Product>** by the backend controller.

What we expect:

- A Visual Studio solution where we'll be able to see
  - Two projects mentioned above
  - How your web application project calls your micro-service project
  - How you make a JS async call to an URL within the same application handled by a backend controller
  - How you've structured your solution from a N-Layer perspective
  - How your data model is implemented (USE CODE-FIRST approach if possible)
  - Clean code with inline comments in English