Question: How do you document your code?

I use visual assist in visual studio mostly to document my code. I am also aware of other tools like Doxygen and GhostDoc.

I enjoy adding a lot comment to my codes. Commenting my code makes support a lot easier for me.

For me code documentation is a necessity and not an option, in my previous life where I work on many projects within the same period, after a while supporting and navigating these code can be a bit difficult without proper documentation.

I also pay a lot of attention to variable naming, I ensure variable names are related to what the variable is to be used for within the code.

All these makes code modification easier for me and anyone modifying my code base. I have more than 10 applications currently running in different production server within my organization. Some of these applications are been modified with little or no help by different developers, some of them are new to the organization.

I also use swagger documentation to automatically generate documentation for my Web APIs.

Question: What are your thoughts on unit testing?

Unit testing is a necessity not an option. Unit test help us easily identify little avoidable mistakes within our code. It also makes regression testing a lot easier.

As a standard in my current organization we expect minimum code coverage of 70% for all code checked in into our repo.

Writing unit test can be a bit tedious sometimes and developers including me sometimes shy away from it. What we have done with my organization to encourage unit testing is advice developers to write unit test first before writing the code. We also factor unit testing time into the timeline.

We also configure our azure repository to calculate unit test code coverage for every code checked-in to the repo, check-in that does not meet the minimum 70% code coverage are rejected.

I also believe unit testing cannot replace functional testing or automated testing using tools like Selenium.

Question: What Design Patterns you have used in your projects?

I started out with repository pattern back in 2017. Introducing separation of concern and dependency injection into my coding was very exciting and it makes my life a lot easier.

I later started applying SOLID principle into my code, it was initially a bit difficult to adapt but with time I got used to it.

I am currently a big fan of clean architecture, which for me combines repository pattern and solid principle. The level of abstraction, code reusability and code readability I was able to achieve with clean architecture is very exciting.

Question: What are the most important performance issues in ASP.Net web applications?

Asp.net framework comes with a lot of issues, notable among them is the post back issues inherent in asp.net web form application. This is a big issue because for every action taken on the page the whole asp.net form is re-rendered, making the page very slow and causing a lot of overhead within the code. Another major issue is the view state that is used in storing the form state. All these drawbacks causes asp.net application to be heavy and very slow.

All these performance issues were completely eliminated in asp.net core. Asp.net core is platform independent, fast and easily configurable framework putting a lot of power in the hands of developers.

The first asp.net core version I used in a project was .net core 1.1 preview and since then asp.net core have been my preferred option for backend programing.

For me programing is fun, I can do it for free.