

Question 1

Source control (or version control) is the practice of tracking and managing changes to code. The codebase is usually stored in a SCM (Source Control Management) system which will help to track and manage code that is uploaded/pushed from multiple sources/developers. The SCM system will help to track code changes, historical changes/revision, different versions of the codebase at different points in the development process and enable collaboration between multiple developers – each with their own versions of the codebase which can later be merged together after detectable conflicts are resolved. Developers will typically work on their own forks and branches and commit their code for review by senior developers.

Question 2

Amongst what would be an endless list of aspects that encompasses “quality software”, here is a list of 5 general aspects that would be of high priority when it comes to quality software.

- Testing

Making sure that the application (and/or individual components/functions of the application) works as expected involves the use of a testing framework such as Jest. This ensures that function output and behaviour is as expected during and after the development process. User testing would also help to catch bugs, logic errors and facilitate User Interface adjustments based on user feedback and product alpha/beta testing.

- Scalability

The application needs to be engineered to be capable of growing to support the size of its increasing userbase. The foundation architecture of the application may have an effect in this outcome. Is it a Monolith application? ... or does the application use microservices? - perhaps both server side and client side. Database architecture is also an important consideration in terms of scalability. A document-based database (such as Mondo DB) is considered to be more easily and quickly deployable and scalable than a table/relational-based database. Mongo DB requires less planning and it is not as rigid in the way that table/relational-based database are to implement.

- Maintainability

The application needs to be easily maintainable by its engineers. The code base needs to be refactorable and re-implemented without causing significant interruption to users. This may perhaps come down to how the application was architecturally implemented. This is also one of many reasons that microservice architecture is generally used with larger applications, as Microservices allow parts of an application to be totally offline without effecting the rest of the application or its users. Another factor would be workflow, source control process and internal company procedures.

- Performance

As the userbase increases and/or new features implemented, the application needs to have been engineered for maximum speed, no matter how large the userbase. Server processing power will also play a major role here, as well as server bandwidth and also whether the application has been deployed using CDN technology (depending if global access is required).

- Security

Any software application needs to be secure against cyber-attacks and service disruption, information theft and identity theft. Keeping application dependency package versions current and up to date helps in maintaining security vulnerabilities. Also, maintaining server operating systems and security utilities can help to prevent malware and spyware script injections into hosted software applications.

Question 3

A MERN stack application is comprised of four main technologies – Mongo, Express, React and Node.

MongoDB Mongo is a document-orientated “NoSQL” database program. It is used to store information in documents (in JSON format) – as opposed to a traditional SQL database which uses relational tables.

Node & Express Node is a JavaScript environment with libraries that make it easier to write software. Express extends Node specifically to make it easier to create web servers. Therefore, Express is a server-side framework which is used for building web/mobile applications and API's.

React React is a JavaScript library/framework for building client-side user interfaces. This is the part of the application that is rendered in the user web browser and is how the user interacts with the application.

As stated, the user (client) interacts with the application via the user interface which is rendered to the users screen via the React framework. When the user/client triggers certain events, React will send or request data to/from the applications Server/Backend (Node/Express) API/s which are listening for incoming requests. The Server will then send a response/s back to the client – perhaps querying and saving data to the Mongo database first.

Question 4

Firstly, the dev team (or project manager or business analyst) would need to meet with the client in order to assess the business requirements of the website or web application.

This would help determine what technologies need to be used to develop a website or application so that it would meet the businesses requirements - now and also perhaps into the future (referring to future scalability).

Depending on what the website or web application requirements are – a static website could easily be done with HTML, CSS and JavaScript, perhaps even using a Framework/Library such as React.

If a web application is required, full knowledge of MERN stack or similar server-side / client-side application design would be required. The team would need to produce technical documentation such as ERD charts, User-flow diagrams as well as UI and UX designs wireframes and low or high-fidelity designs.

Working as a team, a Source Control Process and Source Control Management solution would need to be implemented so as to keep the development process and codebase managed.

Also, depending on the size/scope of the project, the team would also need to determine what development methodology to be suitable for the project. A smaller project might benefit more from 'Rapid Application Development' methodology as opposed to a 'DevOps Deployment' methodology.

Question 5

A project that I worked on earlier in the course was a web application built with Ruby On Rails. Of course, the biggest aspect of this project was learning the Object Orientated language 'Ruby'. Being my first programming language also meant that it was my biggest obstacle.

Apart from Ruby, learning Rails was also quite an undertaking and a challenge. Having only ever worked with HTML and CSS, I really found the concepts of OOP difficult and making sense of how it all fit into the Rails Framework.

One concept and technology I believe I had a good handle on was Postgres and SQL relational databases. Having had previous experience with deploying Content Management Systems such as WordPress meant that I have prior knowledge of relational database design and implementation.

Question 6

With reference to one of your own projects, evaluate how effective your knowledge and skills were for this project and suggest changes or improvements for future projects of a similar nature.

One of the very first projects that I worked on during the Gen Tech Bootcamp was the Personal Portfolio website. This was primarily a personal website which was written using only HTML and CSS.

Having had many years of previous experience working on my own hobby website, I found that I excelled during this project. I received a lot of affirmations that the way I was coding HTML and CSS was indeed how it was supposed to be done by professional developers. I found this to be very rewarding after spending many years learning by myself.

One improvement I have made – and definitely something I have changed – is that I now code SCSS and strict HTML5. I find this to be much more efficient than using older HTML syntax and plain CSS.

Another improvement I have made is the addition of using more JavaScript in the client browser for small things that I previously had tried to do with CSS. I have been able to make my static websites much more visually dynamic.