Introduction – Technology Stack

Our group has decided to use Adobe XD for the planning and designing of the webpage, Adobe XD is a creative platform which allows for designing practice, planning, development and prototyping. The Adobe XD platform will be a major part of designing and implementing user experience methodologies so that we can see what elements work together and what don't. Adobe XD also has templates of smaller screens like mobile phones and tablets so we can see how the website looks on those smaller screens. Another designing method our group has planned on using is creating multiple wireframes all with different designs to see what works together. After the designing process, our group will decide on a Technology stack to implement the rest of the project.

Technology Stack A:

Frontend:

One of the first things our group thought about using for the development of the website is through Amazon Web services. Amazon Web Services is a cloud platform which provides different services across IaaS (Infrastructure as a Service), SaaS (Software as a Service), and PaaS (Platform as a Service). We would plan on using a cloud-based server so that the website will always be available.

A JavaScript framework will be used for the frontend of the website particularly using AngularJS. [1]

Backend:

A MEAN framework will be used for the backend of the website. MEAN stands for MongoDB, Express.js, Angular.js and Node.js. MongoDB is used as the NoSQL of the database, Express and Node run the backend while angular controls the interface.

Middle:

This will be done using C#. The goal of this is to enable communication between the front and back end of the technology stack and act as a glue between the two.

Security:

Password protection and unique ID validation will be used for verification of users on the website. Wrong credentials or non-existent IDs/ usernames will be marked as a threat and access will be denied. On top of this, cross scripting (XSS) by attackers will be obscured through encoding of any HTML used.

Server:

The Apache server will be used for this stack.

Technology Stack B:

Frontend:

Another web service we could use is Microsoft Azure. Like AWS, it is another cloud-based service which offers a wide range of different products which the user can pay to add. In the case of this project, we plan on using WIX for a base template, then branching off from there using HTML (Hypertext Mark-up Language) and CSS (Cascading Style Sheets). HTML will be used to design how the contents of the website are displayed on the browser while CSS will be used to aesthetically style said content.

Backend:

A LAMP framework will be used for the backend of this website. LAMP is an abbreviation of Linux, Apache, MySQL and PHP.

Middle:

This will be accomplished via Java.

Security:

Once again, password protection and unique ID validation will be used for verification of users on the website. Wrong credentials or non-existent IDs/ usernames will be marked as a threat and access will be denied. On top of this, Security Sockets Layer (SSL) will be used for added security. This is used to block interceptions from hackers in the form of man in the middle attacks by forming a one to one connection between the website and the user. [2]

Server:

The Nginx server will be used.

<u>Comparison of the two stacks:</u>

Frontend:

Using plain HTML, CSS and JavaScript is potentially easier to use in the early stages.

On the other hand, using a JavaScript framework, while supported by all modern web browsers and with more functionality, is generally more complicated than using just HTML, JavaScript and CSS. [3]. Using a JavaScript framework would also be more beneficial if the website scaled.

Backend:

On one hand, MEAN allows for JavaScript to be used for both front end and back end which arguably eases the development and interaction of the two which makes it easier for full stack development.

On the other hand, JavaScript slows down the browser and some ad blockers block JavaScript for security reasons which may hinder the website if it is heavily dependent on it.

LAMP is ideal for simple websites and APIs and can also accommodate bespoke development to the user’s specific needs due to its open-source nature. However, MEAN is also ideal for GUI focused projects and start-ups. In some situations, both may be used: LAMP for the API and MEAN for the GUI.[4]

Middle:

Both Java and C# are object-oriented programming languages. Java however is widely used by Linux developers whereas .Net in C# is widespread in the Windows community. This makes C# more straightforward and easier to work with. Nevertheless, Java is more ideal for simpler frameworks.[5]

Security:

XSS is ideal for protection against attacks on JavaScript heavy websites while SSL is ideal for any website. Both can be used in tandem in a situation where they are both required.

Server:

Nginx is event driven and handles multiple requests within one thread. It fully supports all Unix OS and supports Windows partially while Apache is process driven and creates a new thread for each request and is widely used alongside LAMP. It fully supports Windows and Unix. Apache has better OS support and is faster while Nginx is more secure and supports multiple client requests simultaneously albeit with limited resources.[6]

<u>SWOT Analysis of both stacks:</u>

<u>Technology Stack A:</u>

Strengths:

- Its frontend is more widely supported and has more functionality.
- Using the same language for both front and backend allows for quicker development and easier integration.
- C# is widely used and better supported by windows which is arguably the more used than Linux.
- Nginx and more secure, faster and handles multiple requests simultaneously.

Weaknesses:

- JavaScript is more complicated than HTML + CSS.
- JavaScript can slow down the browser and is less trusted by security measures such as ad blockers.
- Java is mostly used by Linux developers whose target audience are significantly less as compared to .NET.
- Nginx only partially supports Windows.

Opportunities:

- A need for Windows based websites with better security creates a profitable gap in the market for this stack.
- A conspicuous need for a website for drivers and their cars makes this ideal.

- Ability to integrate both technologies such as API using LAMP and GUI using MEAN.[7]

Threats:

- Multiple similar websites made by colleagues poses huge competition.
- Constant need for improvement to satisfy user needs creates faster entropy.
- Threats to security due to usage of JavaScript which would create need for integration of SSL in stack B.

Technology Stack B:

Strengths:

- HTML + CSS is easier to use and integration of the two creates for a better-looking GUI and website in general.
- LAMP is more ideal for simpler websites and is able to be modified according to specific user needs.
- Java is more ideal for simpler frameworks.
- Apache has better OS support and is faster.

Weaknesses:

- JavaScript is more widely supported by all web browsers and provides more functionality than HTML + CSS.
- LAMP is less ideal for GUIs.
- C# and .NET are more widespread than Java.
- Apache is unable to simultaneously handle multiple user requests.

Opportunities:

- It is possible to create a more favourable platform for Linux users through a more secure website albeit less favoured.
- The potential need for both API and GUI makes using both stacks possible, specifically the front and back ends.

Threats:

- PHP is arguably outdated and less favoured hence posing a threat for the stack to be less accepted.
- More security threats due to Apache being less secure than Nginx.

References:

[1] Hackernoon. (2017). *How To Pick the Right Web Technology Stack for Your Product* [Website]. Available: https://hackernoon.com/how-to-pick-the-right-web-technology-stack-for-your-product-f6d94440af2f

[2] BRALIN Technology solutions. *9 Security Tips To Protect Your Website From Hackers* [Website]. Available: https://www.bralin.com/9-security-tips-to-protect-your-website-from-hackers

[3] Hubspot. (2018). *Web Design 101: How HTML, CSS and JavaScript work* [Website]. Available: https://blog.hubspot.com/marketing/web-design-html-css-javascript#:~:text=HTML%20provides%20the%20basic%20structure,the%20behavior%20of%20different%20elements.

[4] Linkedin. (2015). *LAMP vs MEAN, Deciding the right stack for your start-up* [Website]. Available: https://www.linkedin.com/pulse/lamp-vs-mean-deciding-right-stack-your-startup-robert-roose/

[5] Decipherzone. (2019). *Java vs .NET for Web Application Development* [Website]. Available: https://www.decipherzone.com/blog-detail/Java-vs--NET-for-Web-Application-Development

[6] Serverguy. (2020). *Apache Vs NGINX – Which Is The Best Web Server for You?* [Website]. Available: https://serverguy.com/comparison/apache-vs-nginx/#:~:text=The%20main%20difference%20between%20Apache,multiple%20requests%20within%20one%20thread.

[7] Last Call. (2020). *LAMP vs MEAN: What is the right stack for you?* [Website]. Available: https://rapidapi.com/blog/lamp-vs-mean/