

ASP.NET vs PHP

Marius Olariu (B00350529)

Introduction

(500 words)

Nowadays, people expect personalized experiences and the Internet makes no exception to this rule. In order to fulfill this user requirement we need dynamic webpages (i.e. pages that can change automatically the content they display), few webpages currently are static (i.e. have the same content every time you open them). An example of a dynamic webpage could be *amazon.com* that displays on the home page products that you might be interested in (based on your previous purchases, search history) whereas an example of a static webpage could be the contents of a book published online (e.g. <https://natureofcode.com/book/>). Developers automatically wonder what are the best technologies to use in order to develop dynamic webpages. However, due to the fact that different dynamic webpages have different requirements this study-case is going to focus on comparing two web technologies suitable for developing a future version of *www.uws.ac.uk*, namely ASP.NET and PHP. One of the main requirements of this site is to be able to process a lot of data (e.g. a lot of applications for programmes provided by university) thus it uses a SQL database system to store it. This study-case is going to compare how the aforementioned technologies interact with SQL database systems.

ASP.NET is a framework from Microsoft for building dynamic web pages that was released in 2002 as a successor to **A**ctive **S**erver **P**ages (ASP). This technology is used in the server-side web development; in other words, it interacts with the "computer" processing the website request and produces web pages that contain only HTML, CSS and JavaScript. ASP.NET provides flexibility to the developers because they have multiple choices when it comes to what languages to use for the backend of the website, namely ASP.NET supports C#, Visual Basic .NET and J# (a language inspired by Java). Additionally, writing the backend in C# or J# speeds the process of providing the website to the client because these languages are compiled (the backend code does not have to be interpreted at runtime). In 2014 Microsoft made the .NET server stack open-source (ASP.NET, .NET compiler, .NET Core Runtime etc.), ASP.NET can be found on a GitHub repository (github.com/aspnet). Moreover, in the same year they started to support cross-platform development by making the tools needed available for platforms like Linux or macOS and deployment on a Linux web server.

PHP is one of the most used programming languages (Cass, 2014 2015) and it is mainly used to write the backend of a webpage, in other words, mainly a server-side language. PHP was released in 1994 and the acronym stands for *H*ypertext *P*reprocessor or *P*ersonal *H*ome *P*age *T*ools, the later variant being the original name. The main advantage of this language is the fact that it can be embedded in HTML code and after being interpreted the PHP code results in HTML code (that gets sent to the browser). It is worth mentioning that PHP development is now open-source. Websites developed using PHP for the backend are cross-platform, namely, they can be hosted on web servers running Linux or Windows. The widely adoption of this language is due to the fact that it is free and runs fast in combination with Apache and MySQL on a low-specifications web server (Converse et al., 2004). Another reason for the use of PHP is that it is easy to learn.

Body

(2500 words)

The above technologies are analysed based on how they interact with **Relational Database Management Systems** (RDBMS). In order to manipulate the data stored in a RDBMS the standard language is **Structured Query Language** (SQL) that is why these RDBMS are sometime referred as *SQL-databases*. There are many RDBMS systems out there, to mention but a few: Oracle, MySQL, Microsoft SQL Server, PostgreSQL etc. . PHP websites use mostly MySQL because these two technologies have been developed with each other in mind, are both open source projects with active web communities around them and (as mentioned above) are fast (Davis & Phillips, 2007). ASP.NET websites use mostly Microsoft SQL Server because both technologies are developed and maintained by Microsoft.

Storage

In the past database storage was quite expensive but as the hardware technology advances and new discoveries are put in practice, for example having data centers at the bottom of oceans (Simon, 2018), the price of storage is becoming cheaper. Nonetheless, there are few things that have to be taken into account when choosing a database system. The considerations will be briefly mentioned and then there is going to be analyzed how the two programming technologies interact.

Scalability

As the university keeps expanding, that is, opens new campuses, it is likely that a type of SQL-database cannot keep the pace with the new requirements for *uws.ac.uk* (e.g. store retrieve more data) thus it would be wise to pick a database system that offers two things: (i) easy migration of data or (ii) available to provide more storage and computing power (the server on which is hosted).

Database replication

No matter how scalable a database is at some point the database needs to be replicated in more than one server. This creates the need of keeping the servers synched, thus there is created a *master-slave* relationship between the database servers. Usually the inserts happen in the master database and the reads are supplied by the slave databases (they keep synched with the master database by periodically updating their content). Moreover, in case the master database fails one of the slaves databases could take the place to ensure availability of the website.

GUI Interface

Often there is a need to interact directly with the database so a GUI can be really helpful, whether a web interface or a native software for a specific OS.

Cost

Security

Maintenance available (or commercial license)

Conclusion

(1000 words)

References

- Cass, S. (2014), ‘The top 10 programming languages spectrum’s 2014 ranking’, IEEE Spectrum **51**(7), 68–68.
- Cass, S. (2015), ‘The 2015 top ten programming languages’, IEEE Spectrum, July **20**.
- Converse, T., Park, J. & Morgan, C. (2004), PHP5 and MySQL bible, Vol. 147, John Wiley & Sons.
- Davis, M. E. & Phillips, J. A. (2007), Learning PHP & MySQL: Step-by-Step Guide to Creating Database-Driven Web Sites, ” O’Reilly Media, Inc.”.
- Simon, K. (2018), ‘Project natick—microsoft’s self-sufficient underwater datacenters’, IndraStra Global **4**(6), 4.