ASP.NET vs PHP

Marius Olariu (B00350529)

1 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 suitabale 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 sytem to store it. This study-case is going to compare how the aforementioned technologies interract 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 *Hypertext Preprocessor* or *Personal Home Page Tools*, 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 is 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 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.

2 Body

(2500 words)

The above technologies are analysed based on how they interract with **R**elational **D**atabase **M**anagement **S**ystems (RDBMS). In order to manipulate the data stored in a RDBMS the standard language is **S**tructured **Q**uery **L**anguage (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.

2.1 Storage

In the past database storage was quite expensive but as the hardware technology advances and new discoveries are put in practice, for example (if cloud storage is chosen) 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 interract.

2.1.1 Scalability

As the university keeps expanding, for example - 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 easy migration of data to another database management system.

2.1.2 Database replication

No matter how scalabale 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 thei content). Moreover, in case the master database fails one of the slaves databases could take the place to ensure availability of the website.

2.1.3 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.

PHP offers native support for the following types of databases.

Table 1: Main databases supported by PHP through native APIs (PHP: Database Extensions - Manual, n.d)

Name	**Type**	**Language**
Cubrid	Relational	SQL
DB++	Relational	AQl
dBase	Relational	dBase
InterBase	Relational	SQL
FrontBase	Relational	SQL
IBM Db2	Relational	SQL
IBM Informix	Relational	SQL
Ingres	Relational	QUEL
MaxDb	Relational	SQL
MongoDB	Document-oriented database	Mongo Shell
mSQL	Relational	SQL
Microsoft SQL Server	Relational	SQL
MySQL	Relational	SQL
Oracle	Relational	SQL
PostgreSql	Object-Relational	SQL
Paradox	Relational	PAL
Ingres	Relational	QUEL

ASP.NET developers can work with a database using **E**ntity **F**ramework **C**ore (EFC), a object-relational mapping that uses .NET objects to manipulate data. ASP.NET supports the following databases.

Table 2: Main databases supported by ASP.NET through EFC (ASP.NET: Database Providers, n.d)

Name	**Type**	**Language**
MariaDB	Relational	SQL
MyCAT Server	Relational	SQl
Firebird	Relational	SQL
Progress OpenEdge	Relational	OpenEdge ABL
IBM Db2	Relational	SQL
IBM Informix	Relational	SQL
Ingres	Relational	QUEL
Microsoft SQL Server	Relational	SQL
MySQL	Relational	SQL
Oracle	Relational	SQL
PostgreSql	Object-Relational	SQL

As can be seen above, for both technologies there is a large variety of databases available. Some databases are available for both technologies like Oracle, PostgreSql, MySQL, Microsoft SQL Server, Ingres, IBM Informix, IBM Db2. There can be said that PHP offers support for almost every database that ASP.NET supports and some more. Therefore, if one is looking for database scalability (see 2.1.1) it is better to develop the backend in PHP.

When it comes to GUI Interfaces (see 2.1.2.) most of them provide at least a web GUI. One could look further to see which type of database provides the best GUI or the largest number of GUIs (e.g. web, native application etc.) but this is beyond the purpose of this work. Again, it seems that PHP is more likely to be the right choice.

2.2 Cost

2.3 Security

2.4 Maintenance available (or commercial license)

2.5 Comparison from which type of devices they were accessed

2.6 Speed

According to *similartech.com*, a website which performs comparisons between technologies, ASP.NET was used in the development of about 2.5 million websites whereas PHP was used for the developemnt of about 7.5 million websites (ASP.NET vs PHP, n.d). These numbers suggests that both of them are valid technologies used by industry. One way to compare the two technologies would be by the webpage load speed. As can be found on Alexa's website, a company owned by Amazon, the current version of *uws.ac.uk* loads pretty slow, on average 3.64 seconds and 83% of websites load faster (uws.ac.uk Traffic Statistics, n.d). It is generally believed that ASP.NET is faster than PHP because it can use for the backend programming languages that are compiled, like C#, whereas PHP is an interpreted language.

There has been found a study (Mirzoev & Sack, 2014) doing a comparis on from this point of view. The authors created two small websites using both technologies: test.aspx, test.php, about.aspx and about.php. The first type of website, *test.**, called 3 JavaScript files, 1 style sheet, 4 images and 1 favic on. The second type of website, *about.**, read a text file with 10.000 lines (each line being 15 characters long) and displayed it in the webpage. The webpage load speed was recorded using an addon in Mozilla Firefox browser, the results can be seen in Figure **??**. In both sessions PHP ran faster than ASP.NET (session 1 -> 0.189 s, session 2 -> 1.281), however, PHP also had one the slowest loading time in session 1 -> 0.304 seconds. In *Session 1* ASP.NET average load time was faster with 11.2 miliseconds than PHP while in *Session 2* PHP website loaded faster on average with 13.56 miliseconds than the one using in ASP.NET. It seems from this study that there is not a big difference between ASP.NET and PHP when it comes to webpage load speed. However, the author belives that a more thorough testing scenario could be designed in order to validate the results of the above mentioned study.

	Testing Session 1		Testing Session 2	
Iteration	about.aspx	about.php	test.aspx	test.php
1	0.222	0.202	1.438	1.317
2	0.198	0.209	1.339	1.337
3	0.294	0.190	1.304	1.337
4	0.214	0.226	1.323	1.296
5	0.193	0.198	1.294	1.301
6	0.243	0.181	1.370	1.317
7	0.209	0.208	1.305	1.313
8	0.197	0.231	1.288	1.332
9	0.228	0.229	1.312	1.381
10	0.195	0.287	1.305	1.310
11	0.189	0.295	1.304	1.281
12	0.189	0.267	1.318	1.370
13	0.229	0.225	1.328	1.328
14	0.249	0.304	1.313	1.314
15	0.200	0.203	1.371	1.302
16	0.191	0.214	1.299	1.314
17	0.197	0.209	1.405	1.298
18	0.245	0.197	1.388	1.291
19	0.202	0.270	1.404	1.335
20	0.205	0.237	1.329	1.331
21	0.245	0.257	1.385	1.337
22	0.224	0.207	1.379	1.300
23	0.223	0.223	1.319	1.427
24	0.207	0.229	1.386	1.332
25	0.239	0.209	1.300	1.366
Mean	0.21708	0.22828	1.34024	1.32668
Standard Deviation	0.0252915	0.0335006	0.042762	0.032263
Min	0.189	0.181	1.288	1.281
Max	0.294	0.304	1.438	1.427
Max & Min Difference	0.105	0.123	0.15	0.146

Figure 1: Speed comparisons between two webpages written using both technologies

3 Conclusion

(1000 words)

References

ASP.NET: Database Providers (n.d). [Online] Available: `https://docs.microsoft.com/en-us/ef/core/providers/index` [Accessed: 07 February 2019].

ASP.NET vs PHP (n.d). [Online] Available: `https://www.similartech.com/compare/asp-net-vs-php` [Accessed: 07 February 2019].

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.".

Mirzoev, D. & Sack, L. (2014), 'Webpage load speed: Asp. net vs. php', arXiv preprint arXiv:1404.2163 .

PHP: Database Extensions - Manual (n.d). [Online] Available: `http://php.net/manual/en/refs.database.php` [Accessed: 07 February 2019].

Simon, K. (2018), 'Project natick—microsoft's self-sufficient underwater datacenters', IndraStra Global **4**(6), 4.

uws.ac.uk Traffic Statistics (n.d). [Online] Available: `https://www.alexa.com/siteinfo/uws.ac.uk` [Accessed: 07 February 2019].