

A Comparative Analysis on Widely used Web Frameworks to Choose the Requirement based Development Technology

Nirjhor Anjum¹, Shamimul Alam²

Chief Analyst Officer, REVE Systems, Dhaka, Bangladesh¹

Technical Lead, Software Development Unit, Digital Business, ADN Telecom²

Abstract: In this modern world of technology there are hundreds of programming languages that are being used in the IT industry for building website and software. It is not a standard practice to use raw programming codes, rather everyone prefers to use Framework which is nothing but a big collection of ready libraries, packages and modules for making the development quicker. For this reason, experts have developed different Frameworks based on different popular programming languages. Now, it becomes really a confusing and challenging matter for companies, developers and managers while they want to choose a framework for developing their software products. As there are variety of frameworks of different programming languages, the technology decision makers get confused while choosing a framework. Considering this as a big problem of the software industry our aim is to test, analyze, assess, and propose a set of guidelines for choosing a framework based on the requirement of a software product.

Keywords: Web Framework, Programming, Choice, Requirement, Development Technology

I. INTRODUCTION

It would take a long time if an organization intends to develop any software product with the help of raw coding, because in case of raw coding the developers need to code each feature from scratch. To reduce the time of coding the same thing that was coded before, technology experts have introduced framework which is combination of different ready libraries, packages and modules that facilitate thousands of features to the developers. Now, the problem is, there are so many frameworks available using which a software or website can be developed; again there are different type of technology solutions like small website, medium website, large website, small software, medium software, large scale software, large web or software ecosystem; and choosing correct framework as per the project type and requirement becomes a matter of confusion. From industry surveys it is found that people choose technology either based on suggestion from their seniors, or based on their organization's previous practice, or based on local market practice, or based on client requirement. But, it is an important factor that, choice of framework should be based on the budget of development and management, performance requirement, product category, product size, end-user type etc. A project may fail due to choice of wrong framework even though the requirement engineering and development is properly done. Failure of software projects can occur due to choice of wrong framework because the manageability, usability, complexity, expandability etc varies from one framework to another.

As there is no methodology or guideline about which framework should be chosen for which type of product development, this paper considers this as a big problem of the software industry. The aim of this paper is to test, analyze, assess, and propose a set of guidelines on which framework is good for which sort of development. This paper will do research on the most popular and widely used web frameworks. Each framework will be tested, reported, analyzed, researched and finally a set of guideline will be provided so that industry decision makers can pick a suitable framework for their software product development. While analyzing and researching on the frameworks, the usability, manageability, reusability, complexity, expandability/customizability etc parameters will be addressed and assessed. Based on the outcome of the real test reports the decisions will be proposed as per the parameters.

II. OBJECTIVES OF THE RESEARCH

The objectives of this paper is:

- To test and analyze the popular web frameworks.
- To compare the frameworks based on a set of parameters that can identify the kind of frameworks.
- To propose a guideline which can be used to choose a framework based on requirement.

III.BACKGROUND STUDY

There are hundreds of websites over internet where people have compared web frameworks in terms of popularity, number of users, user review and rating etc. But there is no comparison or requirement based choice guideline on the web frameworks in terms of usability, manageability, reusability, complexity, performance, expandability or customizability etc. which are major factors before choosing a web framework for developing any web based project whether that is a software or a website. There are even researches on comparison of popular programming languages [1], but no research is found on comparison of web frameworks, while frameworks are widely used for website and software development. As in the software industry nobody uses raw code of any programming language to make software products but prefers to use frameworks [2], therefore doing research on the comparison of web frameworks in terms of performance parameters can help industry decision makers to choose the appropriate framework based on their requirements. In different researches [3], the technology specialists have mentioned “choice of wrong technology” as one of the major issue behind the failure of software projects. Different market-leaders companies also say [4] that, choosing proper technology is a big factor and is highly needed to avoid any kind of project failure. From all these research and feedback it is clear that, decision makers face difficulty and also do mistakes while choosing technology for their projects. This paper will provide a better comparison among the frameworks with guideline for the technology decision makers to choose the correct framework as per stakeholder needs.

IV.RESEARCH METHODOLOGY

The Experimental Method of research is used for this paper, as the study requires real world test results for analyzing the frameworks and proposing a better guideline on choosing requirement based framework. At first, a powerful laptop computer is selected as test environment for testing all the frameworks. Multiple computers are not used for this research as it is required to test all frameworks on the same environment and the same hardware. Next, a set of parameters is selected so that these can be tested on each of the frameworks. It is believed that, test on same properties of different frameworks can bring a better comparison among the frameworks. After that, the tests are conducted on each of the selected frameworks. The test results on the parameters are listed in tabular format, and then analyzed to do research. There are some numeric test results on some parameters that were coming different in each test; in case of such results five (5) sample tests were conducted one after another, and then the average of the test results is considered as a final result. Finally, analyzing all the test results different guidelines are prepare for the technology decision makers so that they can choose a proper framework as per their requirements.

According to the research plan following frameworks are selected to test and analyze, because these are found as most popular frameworks worldwide:

- 1) Laravel (PHP)
- 2) CodeIgniter (PHP)
- 3) Symfony (PHP)
- 4) Zend (PHP)
- 5) ASP.Net Core (C#)
- 6) Java Spring Boot (Java)

The following properties are considered to be tested and analyzed on all of the above Frameworks:

Properties / Metrics	Measurement Unit
General Load Time	Micro Seconds
Data Create Time	Micro Seconds
Data Read Time	Micro Seconds
Data Update Time	Micro Seconds
Data Delete Time	Micro Seconds
Configuration Management	Easy, Average, Difficult
Database Handling Ability	Quantity of Database
Template Engine Availability for Design Management	Yes, No
Vendor Library	Quantity of Vendor Libraries
Design Pattern Acceptance	Quantity of Design Pattern
Routing Robustness	Easy, Average, Difficult
CRUD Generator Availability	Yes, No
Platform Dependency	Independent, Dependent
Source-code Safety (Encryption)	Yes, No

Multi-Threading	Yes, No
Enterprise Level Support	Yes, No
Open Community Support	Yes, No
Documentation for Guideline	Yes, No
Frequency of New Version Release	Frequent, Average, Slow
Feature Focus	Software, Website

To conduct test a single machine is selected. When tests will be running, to get accurate test result, no other application will be opened or used. Moreover, all the frameworks will be tested on same machine so that comparison can be done perfectly. The selected machine for this study is as follows:

Machine Type	Apple
Model	13.3" 2018
Processor	Intel(R) Core(TM) i5 CPU 2.3 GHz
Memory/RAM	8 GB
Operating System	macOS Mojave
Browser	Google Chrome

V.COMPARISON AND EMPIRICAL STUDY ON THE IDENTIFIED PROBLEM

As per research plan at first the CREATE, READ, UPDATE and DELETE activities (together known as CRUD) of each framework are tested. The test results were taken from network log of browsers while executing each action of CRUD:

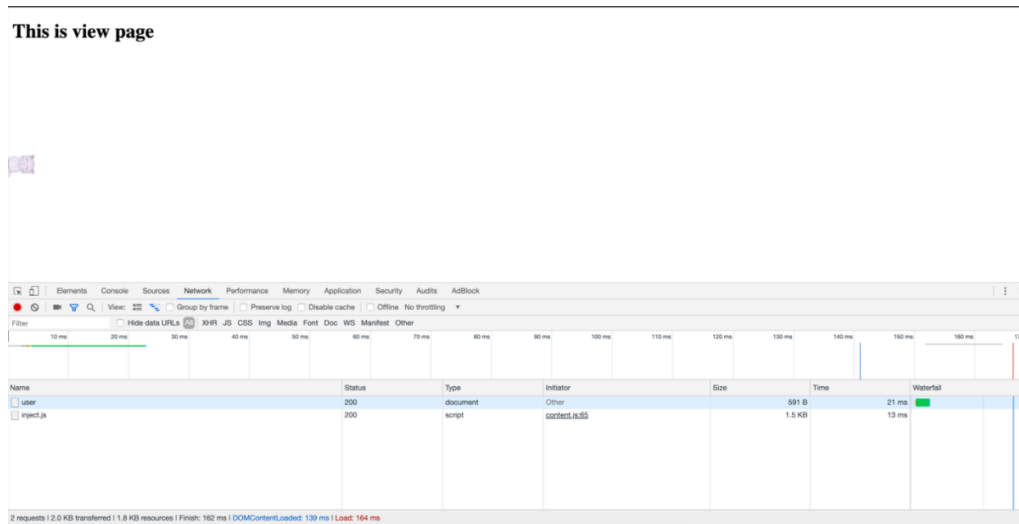


Figure: Image of Test Result Collection from Network Log of Browser

As each test on a single action of CRUD brings a bit different result, 5 tests for each action of CRUD was conducted and then from the average the execution time of that specific action is considered as final result for each framework:

Framework Name: Laravel 5.8

Serial	General Load Time	Data Selecting Time	Data Creating Time	Data Updating Time	Data Deleting Time
1	244000	0.0092079639434814	0.01018500328064	0.010699987411499	0.0095570087432861
2	234000	0.0087420940399171	0.0085890293121338	0.011168003082275	0.0085386196899411
3	271000	0.0080099105834961	0.0086350440979004	0.010379076004028	0.0086269378662109
4	230000	0.0090591907501221	0.0090408325195312	0.008944034576416	0.0095090866088867
5	236000	0.0087299346923828	0.0089678764343262	0.010523796081543	0.0096940994262695
Total	1215000	0.04374909	0.045417786	0.0517149	0.045925752
AVG.	243000	0.00874982	0.00908356	0.01034298	0.00918515

Framework Name: CodeIgniter 3.1

Serial	General Load Time	Data Selecting Time	Data Creating Time	Data Updating Time	Data Deleting Time
1	159000	0.00094389915466309	0.0013711452484131	0.0010130405426025	0.00047492980957031
2	155000	0.00081706047058105	0.0006721019744873	0.00057601928710938	0.00061702728271484
3	155000	0.00069189071655273	0.00056695938110352	0.00089907646179199	0.0004570484161377
4	147000	0.00090193748474121	0.00071501731872559	0.00077104568481445	0.00085997581481934
5	140000	0.00095891952514646	0.00072884559631348	0.00079894065856934	0.00057291984558105
Total	756000	0.004313707	0.00405407	0.004058123	0.002981901
AVG	151500	0.00086274	0.00081081	0.00081162	0.00081162

Framework Name: Zend Framework 3

Serial	General Load Time	Data Selecting Time	Data Creating Time	Data Updating Time	Data Deleting Time
1	237000	0.0070250034332275	0.0076901912689209	0.001967191696167	0.00088787078857422
2	223000	0.0070769786834717	0.0065488815307617	0.001339655303955	0.00064587593078613
3	248000	0.0065748691558838	0.0069198608398438	0.0012321472167969	0.00087904930114746
4	221000	0.0070540904998779	0.0069189071655273	0.0011899471282959	0.001215934753418
5	223000	0.006334973739624	0.0071380138397217	0.0013430118560791	0.00091409683227539
Total	1152000	0.034065916	0.035215855	0.007071953	0.004542828
AVG	230400	0.00681318	0.00704317	0.00141439	0.00090857

Framework Name: Symfony 4

Serial	General Load Time	Data Selecting Time	Data Creating Time	Data Updating Time	Data Deleting Time
1	284000	0.04099702835083	0.17910003662109	0.0016570091247559	0.041982889175415
2	304000	0.039369821548462	0.016694068908691	0.001849889755249	0.040040969848633
3	296000	0.038915157318115	0.016907930374146	0.0020949840545654	0.042295932769775
4	316000	0.03825187631055	0.01699805259671784	0.0021500587463379	0.039408922195435
5	282000	0.038539886474609	0.018736124030696	0.0017178058624268	0.041155146194458
Total	1482000	0.19607377	0.248436213	0.009469748	0.20488386
AVG	294400	0.03921475	0.04968724	0.00189395	0.04097677

Framework Name: Java Spring Boot 2

Serial	General Load Time	Data Selecting Time	Data Creating Time	Data Updating Time	Data Deleting Time
1	596000	4000	16000	25000	5000
2	656000	4000	17000	15000	4000
3	623000	3000	17000	21000	3000
4	607000	4000	19000	17000	4000
5	595000	3000	15000	19000	4000
Total	3077000	18000	84000	97000	20000
AVG.	615400	3600	16800	19400	4000

Framework Name: ASP .NET 4.5

Serial	General Load Time	Data Selecting Time	Data Creating Time	Data Updating Time	Data Deleting Time
1	700000	4.1	290	1.9	11
2	702000	3.3	380	2.3	10.4
3	701000	4.5	310	1.7	12.4
4	752000	4.7	270	2	10.6
5	650000	2.9	386.1	2.1	16.6
Total	3505000	19.5	1636.5	10	61
AVG.	701000	3.9	327.3	2	12.2

The following parameters are bringing various result, so an average of the results are considered as final result:

- General Load Time (of 1.8 KB Sample View Content in Mic Second)
- Data Create Time
- Data Read Time
- Data Update Time
- Data Delete Time

However, as the following parameters have single result and are not testable from output the of any code block, but the results are found from reviewing frameworks, documentations and the framework architectures, therefore the result of the following parameters are provided in a table next to it:

- Configuration Management
- Database Handling Ability by Default
- Template Engine Availability for Design Management
- No of Vendor or Libraries
- Design Pattern Acceptance
- Routing Robustness
- CRUD Generator Availability
- Source-code Safety (Encryption)
- Enterprise Level Support
- Open Community Support
- Documentation for Guideline
- Feature Focus

After testing the executed codes, reviewing the frameworks, its feature and architecture the following results are found from all the selected frameworks:

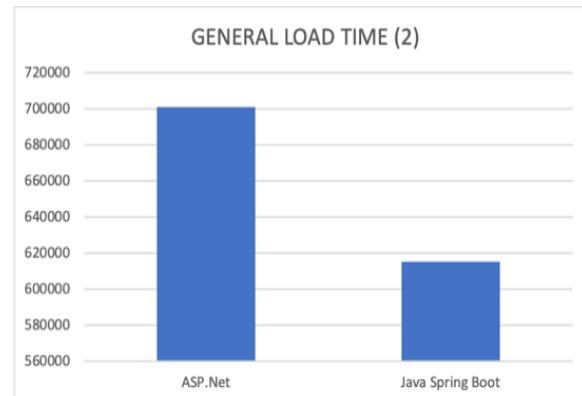
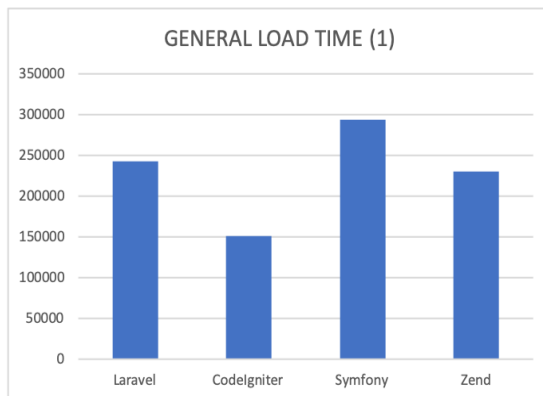
Metrics	Laravel ^[5] (PHP)	CodeIgniter ^[6] (PHP)	Symfony ^[7] (PHP)	Zend ^[8] (PHP)	ASP.Net Core ^[9] (C#)	Spring ^[10] (Java)
General Load Time of 1.8 KB Sample View Content (Mic Second)	243000	151500	294400	230400	701000	615400
Data Create Time	0.0091	0.0008	0.0497	0.0070	327.3	16800
Data Read Time	0.0087	0.0009	0.0392	0.0068	3.9	3600
Data Update Time	0.0103	0.0008	0.0020	0.0014	2	19400
Data Delete Time	0.0092	0.0006	0.0410	0.0009	12.2	4000
Configuration Management	Average	Easy	Average	Difficult	Difficult	Easy
Database Handling Ability	4	8	3	3	7	8
Template Engine Availability for Design Management	Yes	Yes	Yes	Yes	Yes	Yes

No of Vendor or Libraries	76	26	103	23	17 *	15 *
Design Pattern Acceptance	7	2	1	1	8	10
Routing Robustness	No	No	Yes	No	Yes	Yes
CRUD Generator Availability	No	No	No	No	No	No
Platform Dependency	Dependent	Dependent	Dependent	Dependent	Independent	Independent
Source-code Safety (Encryption)	No	No	No	No	Yes	Yes
Multi-Threading	No	No	No	No	Yes	Yes
Enterprise Level Support	No	No	No	No	Yes	Yes
Open Community Support	Yes	Yes	Yes	Yes	Yes	Yes
Documentation for Guideline	Yes	Yes	Yes	Yes	Yes	Yes
Feature Focus	Website, Software	Website, Software	Software	Software	Website, Software	Software

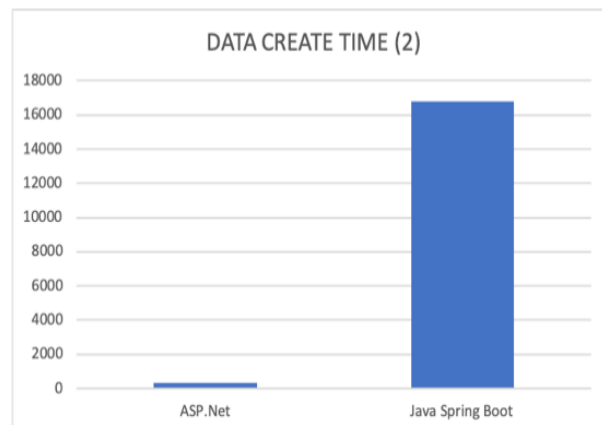
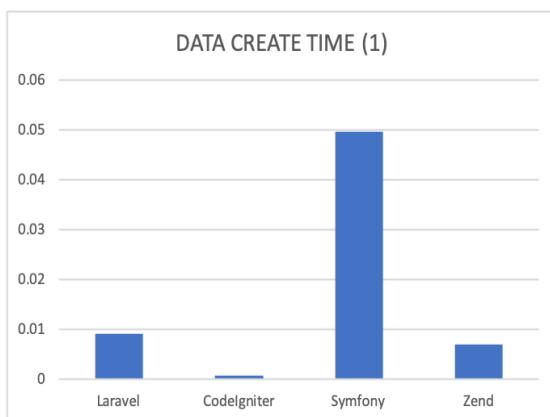
* By default this is the number of package, but the frameworks are open to access any package from open sources

As the Create Time, Read Time, Update Time, Delete Time and General Load Time comes with various numeric value, the following charts are presented to understand them better:

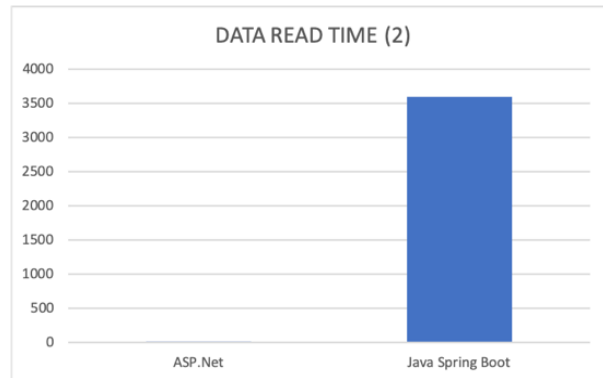
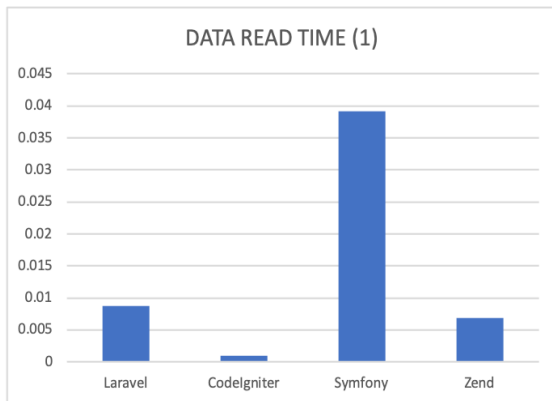
General Load Time of View



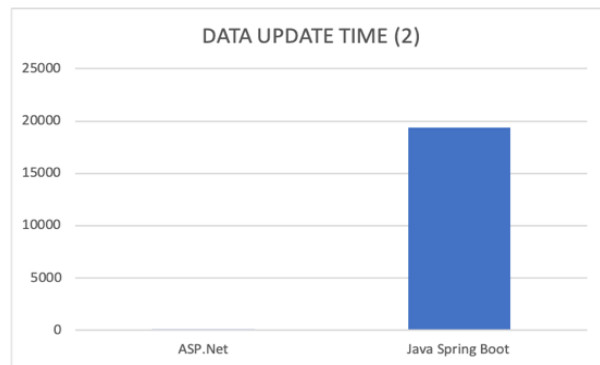
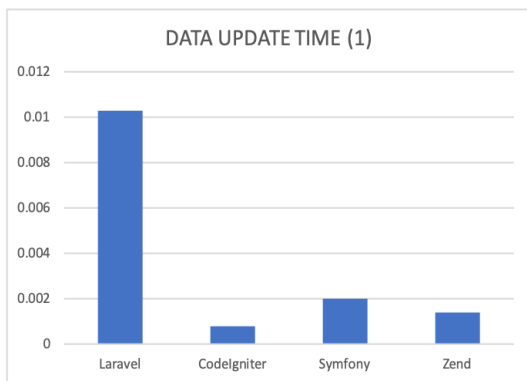
Data Create Time



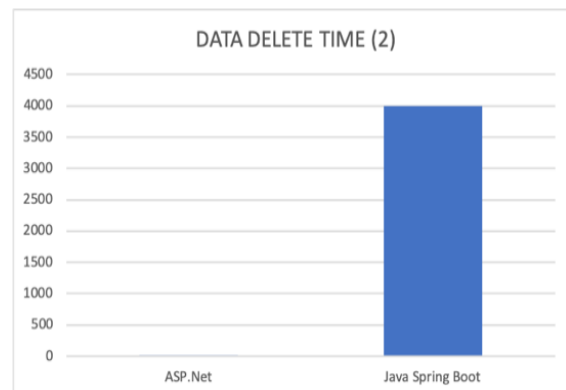
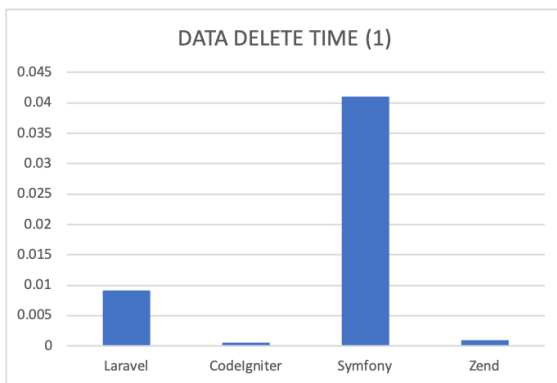
Data Read Time



Data Update Time



Data Delete Time



VIA COMPARISON STUDY ON THE SELECTED WEB FRAMEWORKS

From the analysis on the selected popular web frameworks some guidelines are prepared on choosing requirement-based web framework. The outcome of the study is described in the following sections:

Laravel (PHP): Laravel is very friendly to the developers. Use of composer makes the package management of different libraries easy in Laravel. Laravel has the 2nd largest quantity of vendor, packages, and libraries to facilitate different features for development. Four databases are accepted by this framework. The query execution speed of this framework is faster than any other PHP based web frameworks except CodeIgniter. Though the general load time or query execution time of Laravel a bit slower than CodeIgniter but still Laravel is more robust than CodeIgniter and provides a big number of features. Laravel can support multiple types of design patten which is a very interesting part of this framework. This framework has community forum and support, but they do not have any corporate support. The documentation and guideline is well and frequently updated. However, Laravel is not platform independent and doesn't support encrypting source codes. It also support limited number of databases.

CodeIgniter (PHP): As per test, CodeIgniter is the fastest framework. It has no template engine, which has made the framework faster. The installation and configuration process is so simple. Its documentation is well organized. Though it requires to write more codes in CodeIgniter than Laravel, still for beginners this framework is easy to learn. CodeIgniter allows a developer to write codes being less dependent on its standards which is good for beginners. However, CodeIgniter is not much rich like Laravel. The vendor library size of CodeIgniter is also not much high like Laravel.

Symfony (PHP): Symfony is a powerful framework from as found from the study. It has a big library of functionalities. A main drawback of Symfony is, it is comparatively slower than other frameworks, especially the PHP frameworks. Symfony encourages to code as per their standard which sometime becomes a bit complex for the beginners, but still it is good for the advanced developers. This framework has no corporate support, however it has an open community of people where guidelines can be found.

Zend (PHP): Zend is another robust framework. It is another faster framework compared to other frameworks. This framework has some corporate facilities and supports. As like Symfony, Zend strongly encourages to code as per their standard which sometime becomes a bit complex for the beginners, but still it is good for the advanced developers. The main drawback of Zend is its complex configuration management. Coding is also a bit complex than easy frameworks like CodeIgniter.

ASP.Net Core (C#): From the study it is found that ASP.Net is a faster and robust web framework. The framework is platform independent. Though by default it comes with limited library of features, however a developer can add any library or package of features from its huge collection. It has a good collection of features for web and software development. A very good part of ASP.Net Core is, many if its feature has corporate support beside community based guideline facility. A limitation of ASP.Net Core is, the documentation is not much easy for the beginners. Therefore it is a bit tough for a programmer to adopt ASP.Net Core so easily he is familiar with just raw coding on any language.

Spring (Java): The research finds Spring Framework as one of the most robust framework. By default it comes with less libraries, but a developer can add any library or package from its huge collection. For developing enterprise solutions it has a huge collection of features. Spring has a strong community support. The documentation is also well organized. The main focus of Spring is building web based software. Though this framework has a good documentation, still it is a bit complex for the beginners to learn Spring Framework directly unless learning Java well; the main complexity is its programming architecture which is a bit different from any PHP/C# framework.

VII. PROPOSING GUIDELINE ON CHOOSING REQUIREMENT BASED WEB FRAMEWORK

From the overall observation on all the frameworks some guidelines can be proposed based on the category of product/solution development. These suggestions are proposed considering the parameters like - usability, manageability, reusability, complexity, performance, expandability and customizability. However, the choice guideline is stated below:

Small/Medium Website: For small-medium website development Laravel and CodeIgniter is better. Both are light, faster and developer friendly frameworks that are few reasons for which small-medium website development can be preferred with this framework. If the development team is not much experienced, for them adopting this framework is a good choice. For handling a very big volume of user, query or data parallel Laravel or CodeIgniter is not a very good choice. For Small/Medium Website Spring or Zend is not a good choice at all. However, if the budget is good and the website owner expects to have premium i.e. corporate support they can go for ASP.Net in such case.

Large Website: For large website development frameworks like Symfony especially Zend is a good choice. Library of both Symfony and Zend are rich enough to build large scale websites. If the development team is well experienced in programming and design patterns, these frameworks are good for them. For handling an average volume of user, query or data parallel Symfony or Zend can be a choice.

Large Scale Web Ecosystem: For building large scale web ecosystem ASP.Net Core or Spring Framework can be the first choice. For expert level web specialists choosing Spring or ASP.Net Core is a good option. For handling a very big volume of user, query or data parallel Spring or ASP.Net is a very good choice. However, in case of Large Scale Web Ecosystem it may happen like the website front might be developed on one Framework, and the back-end analytics and management panel might be developed on another Framework.

Small/Medium Software: For small-medium software, Symfony or Zend or ASP.Net Core can be a good choice. However, Laravel can be an option for small software development as well. For expert level web specialists choosing Spring or ASP.Net Core is a good option.

Large Scale Software: For large scale Software, using Spring Framework can be a better choice; and second choice can be ASP.Net Core. For expert level software specialists choosing Spring or ASP.Net Core is a good option. For handling complex query, big volume of data, large number of user, or processing big number of requests simultaneously Spring or ASP.Net is a very good choice as these frameworks allow multi-threading.

VIII.FUTURE WORK

In this research other frameworks are not compared. Moreover, other parameters of the frameworks like 'security' is not studied in this research. However, security is not considered as a parameter to be tested and studied in this research as that is a very large area of analysis.

IX.CONCLUSION

This paper has presented a guideline for the technology decision makers to choose appropriate framework based on their project type and requirements. From this research it is found that, all the frameworks are good in their own specific areas. However, from overall analysis it is found that, Laravel and CodeIgniter is good for small-medium website or small software. Symfony and Zend is good for small-medium software, or large websites. At the other side for medium to large websites ASP.Net and Spring is good. And, for medium or large software Spring is best, and ASP.Net is the second best choice. It is expected that, this comparison will help the industry decision makers to take better decision on selecting development technology as per their need.

REFERENCES

- [1]. Parveen, Z. & Fatima, N. (2016). Performance Comparison of Most Common High Level Programming Languages. International of Computing Academic Research. 5 (5). Retrieved from: <http://www.meacse.org>
- [2]. Why should I use a framework? (2019). Retrieved from: <https://symfony.com/why-use-a-framework>
- [3]. Mandal, A. & Pal S. C. (2015). Identifying the Reasons for Software Project Failure and Some of their Proposed Remedial through BRIDGE Process Models. International Journal of Computer Sciences and Engineering. 3(1), 118-126.
- [4]. Eriksson, U. (2017). Common Reasons why IT Projects Fail. REQtest, ReQtest Article; URL: <https://reqtest.com/agile-blog/common-reasons-projects-fail/>, 28 March 2017
- [5]. Laravel Documentation (2019). Retrieved from: <https://laravel.com/docs/6.0>
- [6]. CodeIgniter Documentation (2019). Retrieved from: <https://codeigniter.com/docs>
- [7]. Symfony Documentation (2019). Retrieved from: <https://symfony.com/doc/current/index.html#gsc.tab=0>
- [8]. Zend Framework Documentation (2019). Retrieved from: <https://framework.zend.com/learn>
- [9]. ASP.Net Core Documentation (2019). Retrieved from: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2>
- [10]. Spring Documentation (2019). Retrieved from: <https://docs.spring.io/spring/docs/current/spring-framework-reference/index.html>
- [11]. Worldwide Usage of Server Side Languages and their Frameworks (2019). Retrieved from: https://w3techs.com/technologies/overview/programming_language/all