Symbiosis

www.symbiosisonlinepublishing.com
ISSN Online: 2474-9257

Review Article Journal of Computer Science Applications and Information Technology Open Access

Intelligent Framework Selection: Leveraging MCDM in Web Technology Decisions

Rakesh Mittapally*

[*]Business Intelligence Architect/AI and ML Engineer, Virginia, USA.

Received: February 16, 2023; Accepted: February 27, 2023; Published: March 14, 2023

***Corresponding author:** Rakesh Mittapally, Business Intelligence Architect/AI and ML Engineer, Virginia, USA; E-mail: mittapallyrakesh@gmail.com*

Abstract

A combination of resources and tools known as a web development framework is used by software developers to create and manage websites, online services, and web applications as well as application programming interfaces (APIs). One of the key decisions that will have a significant impact on the project's short- and long-term performance is the choosing of the best development platform. You may judge its significance by the fact that, After selecting a platform, you must adhere to its constraints for the duration of the software development life cycle. Another challenge with multi-criteria decision-making (MCDM) is selecting the appropriate platform. We offer a COPRAS analytical hierarchy process model to address the MCDM problem. We try to integrate the popular and successful COPRAS modeling technique with the useful modeling capabilities of COPRAS logic. Sun's Java 2 Enterprise Edition (J2EE), Microsoft's ASP.NET, and Linux/Apache/MySQL/PHP (LAMP) are used as substitutes. taken into consideration as evaluation criteria Security, Compatibility, Performance, and Licensing Cost. The top four components are Linux/Apache/MySQL/PHP.

Introduction

Web services, resources, and APIs can all be created more easily with the aid of a software framework known as a web framework (WF) or web application framework (WAF). A standardized process for developing and deploying online applications on the Internet is provided by online frameworks. Web frameworks make an effort to lighten the administrative load brought on by typical web development duties. Many web frameworks, for instance, provide libraries for session management, template frameworks, and database access, and they frequently promote code reuse. Despite the fact that they typically concentrate on creating dynamic websites, they are also applicable to static websites. A web development framework is a collection of tools, pre-written code, and libraries that give programmers a place to start when developing web applications. By offering reusable components, abstractions, and conventions that streamline development tasks, frameworks make it simpler to build websites. Web development frameworks frequently solve concerns including handling routing, maintaining databases, processing user input, rendering templates, and handling security. They provide higher-level abstractions and abstract away low-level details, allowing developers to concentrate on creating logic specific to each application. Frameworks offer pre-made parts and libraries, minimizing the need for repetitive programming.

This makes it easier and more efficient for developers to create applications. Frameworks frequently have built-in functions, instruments, and utilities that make straightforward development tasks easier. They give developers guidelines and best practices for writing readable, maintainable code. As a result, there is no longer a need to start from scratch on every job, which increases productivity. Scalable apps can be developed with the help of frameworks. They provide capabilities like caching, session management, and database abstraction layers that make it possible to handle the rising user loads effectively. The communities for well-known web development frameworks are vast and vibrant. These groups contribute to the creation of plugins, extensions, and other tools that improve the framework's capabilities. The presence of a strong ecosystem guarantees that developers have access to a variety of tools, guides, and support. Inbuilt security features are found in many frameworks to guard against common web application vulnerabilities. These tools assist programmers in implementing secure coding techniques and safeguarding their apps from XSS, CSRF, and SQL injection attacks. Frameworks encourage code organization and maintainability by offering a structured architecture and adhering to recognized design patterns. Developer collaboration, maintenance, and long-term application expansion are all made simpler by this. It's crucial to remember that many frameworks support various programming languages, including JavaScript, Python, PHP, Ruby, and.NET. Consider aspects like the language you are most comfortable using, the project needs, the size and activity of the community, and the accessibility of tools and documentation when selecting a framework.

There are various frameworks for web development, each with their own set of features and advantages. Listed below are a few well-liked web development frameworks: Face book developed the JavaScript library React.js for building user interfaces. It enables you to efficiently update and render data as the application state changes and to build reusable UI components. Angular is a fully functional JavaScript framework developed

by Google for creating dynamic web applications. It supports features like two-way data binding, dependency injection, and routing and offers a strong framework for creating large-scale projects. Progressive JavaScript framework Vue.js is becoming more well-known. It offers a simple syntax and has capabilities like declarative rendering, component-based programming, and a virtual DOM for quick updates. Model-view-controller (MVC) is a design approach used by the high-level Python web framework Django. With capabilities like an ORM (Object-Relational Mapping) for database interaction and built-in management interfaces, it offers a practical solution to quickly construct online applications. The web application framework Ruby on Rails, popularly known as Rails, was developed in Ruby. It adheres to the MVC design and places more emphasis on convention than configuration. Scaffolding, database migration, and automated testing are functionalities offered by Rails. Popular PHP web framework Laravel employs the MVC design principle. Along with capabilities like routing, ORM, caching, and built-in authentication, it offers a simple and attractive syntax. Express.js: Express.js is a straightforward and flexible Node.js web application framework. It offers a comprehensive collection of capabilities, such as routing, middleware support, and template engines, for developing online applications and APIs. Microsoft created the web development framework known as ASP.NET. It enables the use of C# or Visual Basic.NET by developers to create dynamic web apps. Server controls, data binding, and authentication procedures are capabilities that are available in ASP.NET. These are only a few of the web development frameworks that are currently available. Your preferred programming language, project requirements, scalability requirements, and personal preference will all influence the framework you choose.

Choosing the best web development platform is a common challenge for companies. In the context of creating web applications, platform refers to the programming language as well as reusable software such as frameworks, libraries, tools, and auxiliary languages. The auxiliary languages are JavaScript, CSS, and HTML/DHTML. Numerous factors, including security, performance, license fees, compatibility, hardware prices, administration, and others, must be taken into account when making this decision. Although acceptance of information technology is an established study issue, According to Lin et al. (2011), evaluation of the acceptability of alternative items is frequently overlooked. They got to the conclusion that it is desirable to do a comparative research of numerous goods rather than rely simply on one product study after analyzing the Microsoft and Java web application platforms. We'll go into greater depth about the pertinent works that have been done currently. Direct exploratory tests were carried out by Prechelt (2011) to compare the Java EE, Perl, and PHP programming languages. Chen and Ma (2004) determined that Sun's Java 2 Enterprise Edition (J2EE), an open technology framework, is the best framework for server-side development. Ramana and Prabhakar examined the LAMP architecture's performance and developed a scale to evaluate it in 2005. Wu et al. (2008) proposed and created a pretty comprehensive security module for the LAMP application framework. LAMP wins when comparing the security requirements of various development platforms. However, ASP.NET from Microsoft was named by GU and TANG (2010) as one of the most representative and popular web development technologies. The framework should be chosen based on the actual development requirements, they concluded after comparing and contrasting the three ASP.NET frameworks (MVC, MVP, and Web Forms).

As was previously mentioned, selecting the best web development platform is a challenging, complex decision based on a number of factors. Therefore, the following three main web development platforms are compared using the COPRAS analytical hierarchy process-based approach presented in this research. • Linux/Apache/MySQL/PHP (LAMP) • Microsoft's ASP.NET • Sun's Java 2 Enterprise Edition (J2EE) The four main evaluation factors for these three platforms are Security, Compatibility, Performance, and Licensing Cost. Web development requires careful attention to information security, which must be maintained over the course of the project. Due to the web application's requirement to communicate with other distant services and distributed systems, information security is of the utmost importance. Due to the complexity of the web, security is exceedingly weak. Chen et al. (2008) also concurred that as online services are used more often, businesses are realizing that security is a major concern for actual business systems. Backward compatibility with earlier versions of the platform is essential for platform upgrades. This is done to ensure that programs created using an older language version can also be used in the newly upgraded environment. Backward compatibility must be accessible if a business wants to stay current with technology and benefit from new features added in the upgraded version of the platform. Zhong and Yang (2009) emphasized a stable and compatible evolution of software while proposing a collection of design strategies for developing enterprise web services. To increase compatibility, they suggested a method akin to versioning.

In a single study of a web development environment, there are numerous existing works. However, the COPRAS examination of the adoption of substitute items is largely disregarded. As mentioned above, some comparison research between "frameworks" in a single setting is conducted. However, choosing the best web development environment among LAMP, J2EE, and ASP.NET is a crucial choice that many managers today must make. Since selecting the best platform depends on a number of factors, including security, cost, performance, compatibility, etc., an efficient management tool for decision-making is required. Each institution prioritizes these factors differently, affecting the best decision. As a result, the current exploratory tests to compare technologies are insufficient. Additionally, there are extremely few exploratory studies that compare programming languages. In this article, we suggest the powerful management tool COPRAS to carry out a comparison study of several web technologies. The Managers present a method for comparing popular web

development platforms that can be used to new platforms in the future and base the analysis on additional criteria. This takes into account the IT industry's rapid expansion and technical advancement.

Material and Method

LAMP is a popular software stack for web development. Each component of the stack—Linux, Apache, MySQL, and PHP—serves a specific purpose. Let's examine each one in more detail: The LAMP stack's base operating system is Linux, an open-source program. Hosting web applications is made possible by the robust and secure environment it offers. Ubuntu, CentOS, and Debian are a few Linux distributions that are frequently used in LAMP configurations. A popular web server program is Apache HTTP Server. It is renowned for its robustness, adaptability, and stability. Clients send Apache incoming HTTP requests, and Apache responds to the clients with the desired web pages or resources. It supports functions like SSL/TLS encryption, URL rewriting, and virtual hosting, among others. Open-source MySQL is a well-known relational database management system (RDBMS). For online applications, it offers a dependable and effective means of storing and retrieving data. MySQL is renowned for its efficiency, scalability, and speed. Both database management and querying are available using SQL (Structured Query Language). The server-side programming language PHP (Hypertext Preprocessor) was created for web development. It enables developers to create online content dynamically and is incorporated within HTML. PHP is renowned for its ease of use, adaptability, and broad community support. It is a logical fit for the LAMP stack because it interacts well with Apache and MySQL. Developers can create and deploy dynamic websites and web apps using the LAMP stack as a whole. The operating system is Linux, while the web server operations are handled by Apache, the database operations are handled by MySQL, and the server-side scripting is powered by PHP. Because of its dependability, affordability, and interoperability with open-source technologies, the LAMP stack has gained widespread use.

The ASP.NET web development platform from Microsoft enables programmers to create strong and dynamic web applications. It is a component of the larger Microsoft.NET ecosystem. Here are some of the main ASP characteristics and elements.

Model-View-Controller (MVC) in NET Architecture: The MVC architectural pattern, which helps divide an application's concerns into various components, is followed by ASP.NET. The model provides the data and company logic, the view handles the display layer, and the controller manages the flow of messages and responses. An alternative programming model made available by ASP.NET is called Web Forms. It employs a more conventional event-driven programming methodology in which programmers build web forms with server-side controls and manage user interaction events.

Razor View Engine: ASP.NET uses the Razor view engine, which offers a simple and expressive syntax for creating dynamic web pages. To create dynamic content, Razor blends HTML markup with C# or Visual Basic code. Web API: A robust foundation for creating RESTful Web APIs is provided by ASP.NET. Web browsers, mobile devices, and other applications are just a few of the customers that developers can design APIs for. Microsoft provides Visual Studio, an advanced and feature-rich IDE made especially for creating ASP.NET applications.

The development process is streamlined by Visual Studio's capabilities for editing, debugging, testing, and deployment of code. Support for numerous programming languages, including C# (pronounced "C sharp") and Visual Basic.NET, is provided by ASP.NET. These.NET framework-compliant languages provide strong functionality for web development.

Cross-Platform Development: Thanks to the release of.NET Core, a cross-platform in nature free and open-source version of.NET, ASP.NET apps may now be created and used on Linux, macOS, and Windows.

Microsoft technology integration: ASP.NET works in harmony with other Microsoft products, including SQL Server databases, Azure cloud services, and Active Directory for authentication and authorization.

Security features: Applications should be protected from common online flaws like cross-site scripting (XSS) and SQL injection attacks. ASP.NET integrates a number of security features, including request validation, authentication, and authorization procedures.

A complete collection of tools and features are available with ASP.NET to help developers create scalable, enterprise-level web applications. Its integration with Microsoft technologies, cross-platform support, and strong community support make it a popular choice among developers and organizations using the Microsoft ecosystem.

For the creation and distribution of enterprise-level Java applications, Sun's Java 2 Enterprise Edition (J2EE), now known as Java Platform, Enterprise Edition (Java EE), was a popular platform. While it was initially developed by Sun Microsystems, it is now maintained by the Eclipse Foundation under the name Jakarta EE.

Here are some key aspects of J2EE/Java EE: Component-Based Architecture: J2EE/Java EE follows a component-based architecture, allowing developers to build applications using reusable components. These components include Enterprise JavaBeans (EJBs) for business logic, JavaServer Pages (JSP) for dynamic web content, and Java Servlets for handling web requests.

Scalability and High Performance: J2EE/Java EE provides features and services that support scalability and high performance, making it suitable for enterprise applications. These include connection pooling, distributed caching, and load balancing capabilities.

Java Servlets and JavaServer Pages (JSP): Java Servlets are server-side components that handle HTTP requests and responses. JSP allows developers to dynamically generate web content by embedding Java code within HTML templates.

Enterprise JavaBeans (EJB): EJB is a component model for building server-side business logic in Java. It provides features like transaction management, security, and remote method invocation (RMI) for distributed applications.

Java Persistence API (JPA): JPA is a standard API for object-relational mapping (ORM) in Java. It simplifies database interaction by allowing developers to map Java objects to relational database tables.

Java Message Service (JMS): JMS is a messaging API that enables applications to exchange asynchronous messages. It supports the publish-subscribe and point-to-point messaging models, facilitating communication between different components of an application.

JavaServer Faces (JSF): JSF is a Java web application framework for building user interfaces. It provides a component-based model for creating dynamic web pages and supports features like event handling, validation, and navigation.

Web Services Support: J2EE/Java EE includes support for building and consuming web services using standards like SOAP (Simple Object Access Protocol) and XML (eXtensible Markup Language).

Security and Authentication: J2EE/Java EE offers various security mechanisms and authentication protocols, allowing developers to secure their applications. This includes support for SSL/TLS encryption, access control, and integration with authentication providers like LDAP (Lightweight Directory Access Protocol) or database-based authentication.

Java EE has been widely adopted in enterprise environments due to its robustness, scalability, and extensive ecosystem. It provides a standardized set of APIs and services that simplify the development of complex enterprise applications. However, it's important to note that Java EE has transitioned to Jakarta EE, with ongoing development and updates now happening under the Jakarta EE umbrella.

In 1994, Zavadskas, Kaklauskas, and Sarka published the initial version of the COPRAS (Complex Proportional Assessment) approach. This method evaluates the options by taking the relative weights of the relevant criteria. and establishes their priority in light of the competing criteria (Zavadskas et al., 2009). The relevance and utility level (priority) of the alternatives are assumed to be directly and proportionally correlated. According to Chatterjee and Chakraborty (2014), this method selects the best option by considering both the ideal and ideal-worst options. Due to its simplicity, the COPRAS approach has been used to solve a variety of issues in a variety of sectors, including economics, property management, and construction. In order to determine the optimal option, Zavadskas et al. (2001) introduced the COPRAS approach for evaluating building life cycles. Vilutien and Zavadskas (2003) used this method to discover the most efficient variant of a home maintenance task. A home credit access model was created by Zavadskas et al. in 2004 using the COPRAS methodology. Apartment building maintenance contractors were chosen by Zavadskas and Vilutiene (2004). Building design and renovation using the COPRAS approach was proposed by Kaklauskas et al. in 2005. Andrukevicius (2005) applied this technique to find the best builder for a trade and entertainment center. In order to replace the windows in the main building of Vilnius Gediminas Technical University, Kaklauskas et al. (2006) examined contractors. Using the COPRAS approach, Kaklauskas et al. (2007a) chose the optimum construction option. With the aid of the COPRAS approach, Kaklauskas et al. (2007b) ascertained the market value of real estate.

Result and Discussion

Table 1. DATA SET

	Security	Compatibility	Performance	Licensing cost
Linux/ Apache/ MySQL/ PHP	0.62	0.61	0.61	0.46
Microsoft's ASP.NET	0.006	0.16	0.16	0.09
Sun's Java 2 Enterprise Edition	0.376	0.23	0.23	0.46

A comparison of the three technologies' security, compatibility, performance, and license costs based on the given data yields some insightful results. With a security grade of 0.62, Linux/Apache/MySQL/PHP emerges as the most safe solution, outperforming both Microsoft's ASP.NET (0.006) and Sun's Java 2 Enterprise Edition (0.376). Microsoft's ASP.NET (0.16) performs worse in terms of compatibility than Linux/Apache/MySQL/PHP (0.61) and Sun's Java 2 Enterprise Edition (0.23). Performance-wise, all three technologies are fairly comparable, with Sun's Java 2 Enterprise Edition rating 0.23, Microsoft's ASP.NET scoring 0.61, and Linux/Apache/MySQL/PHP earning 0.61. However, when it comes to licensing prices, Sun's Java 2 Enterprise Edition (0.46) and Linux/Apache/MySQL/PHP (0.09) both rank lower than Microsoft's ASP.NET (0.09), making it the most affordable alternative.

Overall, the decision between these technologies will be based on the priorities and requirements of a given project. If security is your top priority, Linux/Apache/MySQL/PHP can be your best option. Microsoft's ASP.NET, on the other hand, might be more alluring for those looking for improved compatibility and lower license fees. Sun's Java 2 Enterprise Edition may be a good compromise option because it offers reasonable performance and license fees as well as a respectable level of security and compatibility. In order to make an informed choice, it is crucial to compare these characteristics to the demands and objectives of the project.

Table 2. Normalized Data				
	Security	Compatibility	Performance	Licensing cost
Linux/Apache/MySQL/PHP	0.62	0.61	0.61	0.46
Microsoft's ASP.NET	0.01	0.16	0.16	0.09
Sun's Java 2 Enterprise Edition	0.38	0.23	0.23	0.46

The normalized data makes it possible to more clearly comprehend how the various technologies differ from one another. With a security grade of 0.62, it appears that Linux/Apache/MySQL/PHP continues to hold the top spot for being the most secure solution. With a security rating of 0, Microsoft's ASP.NET is still considerably less safe despite improvements from the prior data. With a security score of 0.38, Sun's Java 2 Enterprise Edition lies somewhat in the middle of the two. Linux/Apache/MySQL/PHP maintains its high compatibility rating of 0.61, indicating good compatibility. Sun's Java 2 Enterprise Edition and Microsoft's ASP.NET both have lesser compatibility scores of 0.16 and 0.23, respectively. All three technologies score similarly in terms of performance, with ratings of 0.61 and 0.23 for Sun's Java 2 Enterprise Edition, Microsoft's ASP.NET, and Linux/Apache/MySQL/PHP, respectively. Both Linux/Apache/MySQL/PHP and Sun's Java 2 Enterprise Edition have ratings of 0.46 for licensing costs, indicating comparable license costs. Microsoft's ASP.NET, which had a lower grade of 0.09 (see figure 1), is still the most economical choice.

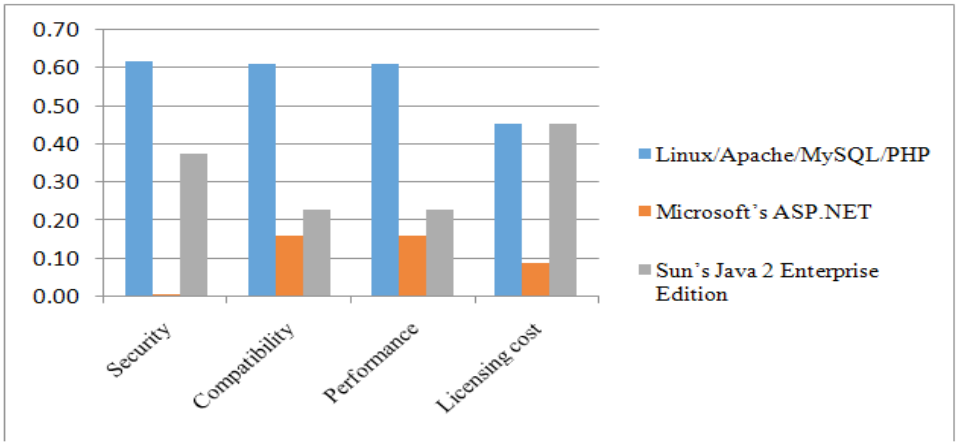

Figure 1 : Normalized Data

The general comparison of the technologies is largely comparable with the earlier research when taking into account the normalized data. The best technology to choose will ultimately depend on the particular priorities and requirements of a project.

Table 3: Weighted data gives each of the four factors				
	Security	Compatibility	Performance	Licensing cost
Linux/Apache/MySQL/PHP	0.48	0.1	0.33	0.09
Microsoft's ASP.NET	0.48	0.1	0.33	0.09
Sun's Java 2 Enterprise Edition	0.48	0.1	0.33	0.09

Table 3's weighted data gives each of the four factors—security, compatibility, performance, and licensing cost—equal weight or relevance. As a result, all three technologies—Linux/Apache/MySQL/PHP, ASP.NET from Microsoft, and Java 2 Enterprise Edition from Sun—receive equal marks for each of these factors. All technologies have security ratings of 0.48, which indicates a comparatively high level of security attention during the evaluation. Similar to this, all technologies have compatibility ratings of 0, indicating that they are all comparable in terms of interoperability with other platforms or operating systems. All three technologies have a performance rating of 0.33, indicating comparable performance characteristics. It is significant to note that a number of variables, including hardware, optimization, and particular use cases, can affect performance. Each technology receives a licensing cost rating of 0.09, suggesting that they are all thought to have minimal licensing costs. This implies that there is no discernible difference in licensing costs between the three technologies based on the weighted data.

Table 4: provide an evaluation based on relative relevance and scores				
	Security	Compatibility	Performance	Licensing cost
Linux/Apache/MySQL/PHP	0.30	0.06	0.06	0.15
Microsoft's ASP.NET	0.00	0.02	0.02	0.03
Sun's Java 2 Enterprise Edition	0.18	0.02	0.02	0.15

In order to provide an evaluation based on relative relevance and scores, the weighted normalized decision matrix combines the weights and the normalized data. By taking into account the normalized scores as well as the relative relevance of each component, the weighted normalized decision matrix offers a thorough evaluation. Based on the precise weights given to each criterion and the priorities of the current project, it can help in making judgments that are well-informed.

Table 5. Bi and Ci

	Bi	Ci
Linux/Apache/MySQL/PHP	0.419	0.150
Microsoft's ASP.NET	0.035	0.029
Sun's Java 2 Enterprise Edition	0.226	0.150
	min(Ci)*sum(Ci)	0.0097

Table 5 lists the values for Bi and Ci for each of the following technologies: Microsoft's ASP.NET, Sun's Java 2 Enterprise Edition, and Linux/Apache/MySQL/PHP. Without more details, it is unclear what these indicators actually imply or are used for. The values are Bi values of 0.419 and Ci values of 0.150 for Linux/Apache/MySQL/PHP. Bi values for Microsoft's ASP.NET are 0.035 and Ci values are 0.029. Bi values for Sun's Java 2 Enterprise Edition are 0.226 and Ci values are 0.150. Calculated as 0.0097, the value min(Ci)*sum(Ci) denotes the minimum value of Ci multiplied by the total number of Ci values.

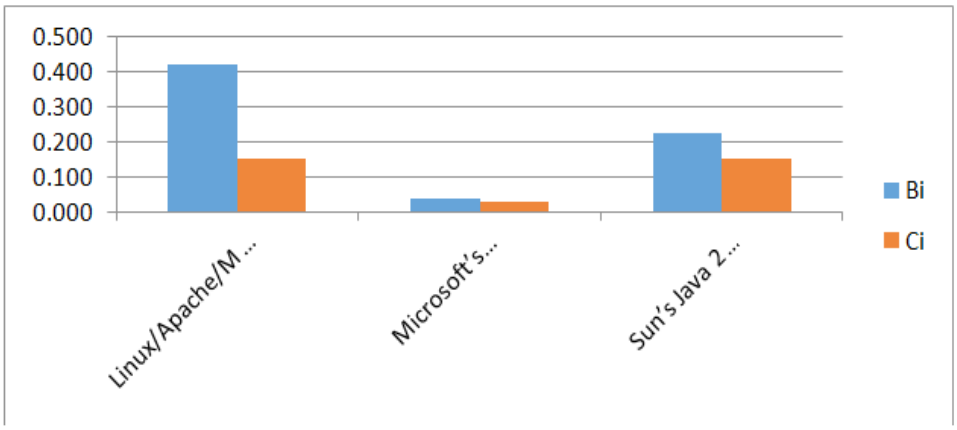

Figure 2 : Bi and Ci graph

Figure 2 Gives Linux/Apache/MySQL/PHP, the values are Bi values 0.419 and Ci values 0.150. Microsoft's ASP.NET has Bi values 0.035 and Ci values 0.029. Sun's Java 2 Enterprise Edition has Bi values 0.226 and Ci values 0.150.

Table 6. Min(Ci)/Ci values

	Min(Ci)/Ci
Linux/Apache/MySQL/PHP	0.1957
Microsoft's ASP.NET	1.0000
Sun's Java 2 Enterprise Edition	0.1957
SUM	1.3913

Table 6 lists the values for Min(Ci)/Ci for each of the following technologies: Sun's Java 2 Enterprise Edition, Microsoft's ASP.NET, and Linux/Apache/MySQL/PHP. The Min(Ci)/Ci values are totaled in the "SUM" row.The value of Min(Ci)/Ci for Linux, Apache, MySQL, and PHP is 0.1957. The Min(Ci)/Ci value for Sun's Java 2 Enterprise Edition is also 0.1957. Microsoft's ASP.NET, on the other hand, has a Min(Ci)/Ci ratio of 1.0000. The cumulative relative comparison of all the Ci values to the minimum value is represented by the Min(Ci)/Ci values added together, which equals 1.3913.

Table 7. Qi, Ui and Rank

	Qi	Ui	Rank
Linux/Apache/MySQL/PHP	0.465	100%	1
Microsoft's ASP.NET	0.272	58%	3
Sun's Java 2 Enterprise Edition	0.273	59%	2

The Qi value for each technology is represented by Qi. The Qi value of Sun's Java 2 Enterprise Edition is 0.273, Microsoft's ASP.NET is 0.272, and Linux/Apache/MySQL/PHP is 0.465. The Ui value for each technology is represented by Ui. According to the greatest Qi value, it shows the proportionate percentage score for each technology. The ranking order determined by the Qi values is represented by Rank. With a rank of 1, Linux/Apache/MySQL/PHP is ranked first. With a rank of 2, Sun's Java 2 Enterprise Edition is ranked second. As seen in figure 3, Microsoft's ASP.NET is ranked third with a rank of 3.

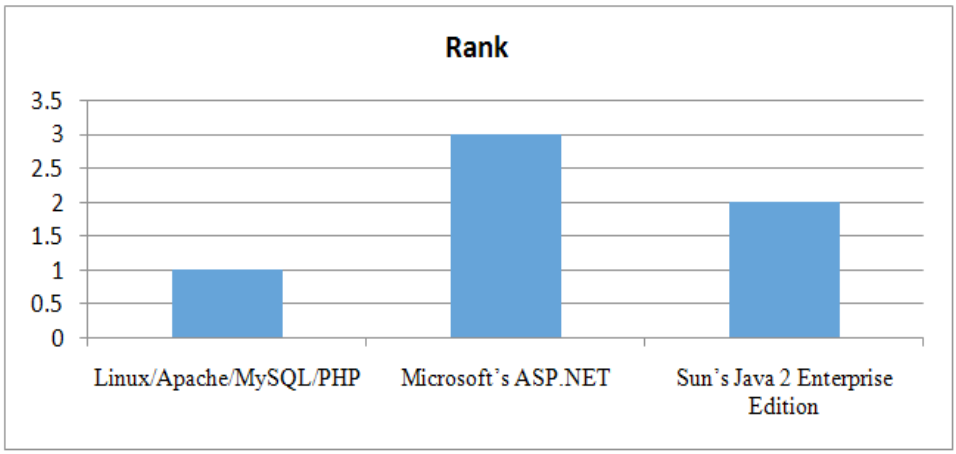

Figure 3 : Ranking

LAMP is the web development platform of choice based on the scores provided in this research. This methodology only considers four factors when deciding on the optimal platform. However, more standards might be introduced dependent on the business and its needs. Additionally, since there are various frameworks for each of the aforementioned programming languages, there are more options that can be considered. A framework is a collection of repeatable procedures that are applicable to web development. The availability of frameworks, the availability of components, the ease of configuration, the standard of manpower necessary, the language supporting the current database, etc. are other factors that a corporation may want to assess based on the industry. The values for pair wise comparisons may potentially deviate from the model shown above depending on the circumstances and policies of the firm. For instance, a young business can place more value on performance than compatibility. On the other hand, a well-known company that only makes newer software updates can place more value on backward compatibility. As a result, businesses can utilize the aforementioned model as a starting point when choosing the appropriate platform to use for constructing their web application.

Conclusion

Finding the best platform for web application development has shown to be a multi-criteria decision-making challenge that has been effectively solved using fuzzy AHP. This issue with decision-making has been solved by a successful management tool. The majority of earlier research tended to concentrate on single platform studies or exploratory trials. Managers needed a precise and dependable tool to choose the best development platform based on aspects of corporate culture, finances, and objectives given the vast volume of web development projects conducted each year. In this example, the criteria for 'Security' contribute the most to the ultimate success of the web development project, followed by 'Performance', 'Compatibility', and 'Licensing cost', based on overall score. The finest platform for developing online applications is the Linux/Apache/MySQL/PHP (LAMP) platform, which consists of the Apache online Server, MySQL database query language, and PHP scripting language. Additionally, the Managers can expand on this methodology in the future to compare-evaluate different platforms and base their research on additional factors described in the "Discussion and analysis" section. This takes into account the IT industry's quick development and ongoing technical changes.

References

1. Sarfaraz, A., P. Mukerjee, and K. Jenab. "Using fuzzy analytical hierarchy process (AHP) to evaluate web development platform." Management Science Letters 2, no. 1 (2012): 253-262.
2. Bodin, Lawrence D., Lawrence A. Gordon, and Martin P. Loeb. "Evaluating information security investments using the analytic hierarchy process." Communications of the ACM 48, no. 2 (2005): 78-83.
3. Cao, Jian, Feng Ye, and Gengui Zhou. "A new web-based method for partner selection in the agile virtual enterprise." In 2006 6th World Congress on Intelligent Control and Automation, vol. 2, pp. 6920-6924. IEEE, 2006.
4. Chang, Da-Yong. "Applications of the extent analysis method on fuzzy AHP." European journal of operational research 95, no. 3 (1996): 649-655.
5. Sarfaraz, A., P. Mukerjee, and K. Jenab. "Using fuzzy analytical hierarchy process (AHP) to evaluate web development platform." Management Science Letters 2, no. 1 (2012): 253-262.
6. Zhi-Guo, Chen, Xu Ya-Feng, Wang Yong-Gang, and Li Hang. "Construction of a Safe Web Service Based on WSE." In 2008 International Conference on Cyberworlds, pp. 481-486. IEEE, 2008.
7. Gu, Ming-xia, and Keming Tang. "Comparative analysis of WebForms MVC and MVP architecture." In 2010 The 2nd Conference on Environmental Science and Information Application Technology, vol. 2, pp. 391-394. IEEE, 2010.
8. Lin, Julian, Hock Chuan Chan, and Kwok Kee Wei. "Understanding competing web application platforms: An extended theory of planned behavior and its relative model." IEEE Transactions on Engineering Management 58, no. 1 (2010): 21-35.
9. Nakatani, Kazuo, and Ta-Tao Chuang. "A web analytics tool

selection method: an analytical hierarchy process approach." Internet research 21, no. 2 (2011): 171-186.

10. Benmoussa, Khaoula, Majida Laaziri, Samira Khoulji, Kerkeb Mohamed Larbi, and Abir El Yamami. "A new model for the selection of web development frameworks: application to PHP frameworks." International Journal of Electrical and Computer Engineering 9, no. 1 (2019): 695.
11. Plekhanova, Julia. "Evaluating web development frameworks: Django, Ruby on Rails and CakePHP." Institute for Business and Information Technology 20 (2009): 2009.
12. Ignacio Fernández-Villamor, José, Laura Díaz-Casillas, and Carlos Á. Iglesias. "A comparison model for agile web frameworks." In Proceedings of the 2008 Euro American Conference on Telematics and Information Systems, pp. 1-8. 2008.
13. Laaziri, Majida, Khaoula Benmoussa, Samira Khoulji, and Mohamed Larbi Kerkeb. "A Comparative study of PHP frameworks performance." Procedia Manufacturing 32 (2019): 864-871.
14. Prechelt, Lutz. "Plat_Forms: A web development platform comparison by an exploratory experiment searching for emergent platform properties." IEEE Transactions on Software Engineering 37, no. 1 (2010): 95-108.
15. Xing, YongKang, JiaPeng Huang, and YongYao Lai. "Research and analysis of the front-end frameworks and libraries in e-business development." In Proceedings of the 2019 11th International Conference on Computer and Automation Engineering, pp. 68-72. 2019.
16. Borissova, Daniela, Zornitsa Dimitrova, Vasil Dimitrov, Radoslav Yoshinov, Magdalena Garvanova, and Ivan Garvanov. "Multi-Attribute Decision-Making Model for Ranking of Web Development Frameworks." In 2021 25th International Conference on Circuits, Systems, Communications and Computers (CSCC), pp. 3-8. IEEE, 2021.
17. Hezer, Seda, Emel Gelmez, and Eren Özceylan. "Comparative analysis of TOPSIS, VIKOR and COPRAS methods for the COVID-19 Regional Safety Assessment." Journal of infection and public health 14, no. 6 (2021): 775-786.
18. Kundakcı, Nilsen, and A. Işık. "Integration of MACBETH and COPRAS methods to select air compressor for a textile company." Decision Science Letters 5, no. 3 (2016): 381-394.
19. Organ, Arzu, and Engin Yalçın. "Performance evaluation of research assistants by COPRAS method." European Scientific Journal 12, no. 10 (2016): 102-109.
20. Kumari, Reetu, and Arunodaya Raj Mishra. "Multi-criteria COPRAS method based on parametric measures for intuitionistic fuzzy sets: application of green supplier selection." Iranian journal of science and technology, Transactions of Electrical Engineering 44, no. 4 (2020): 1645-1662.
21. Amoozad Mahdiraji, Hannan, Sepas Arzaghi, Gintaras Stauskis, and Edmundas Kazimieras Zavadskas. "A hybrid fuzzy BWM-COPRAS method for analyzing key factors of sustainable architecture." Sustainability 10, no. 5 (2018): 1626.
22. Zheng, Yuanhang, Zeshui Xu, Yue He, and Huchang Liao. "Severity assessment of chronic obstructive pulmonary disease based on hesitant fuzzy linguistic COPRAS method." Applied Soft Computing 69 (2018): 60-71.
23. Keshavarz Ghorabaee, Mehdi, Maghsoud Amiri, Jamshid Salehi Sadaghiani, and Golnoosh Hassani Goodarzi. "Multiple criteria group decision-making for supplier selection based on COPRAS method with interval type-2 fuzzy sets." The international journal of advanced manufacturing technology 75 (2014): 1115-1130.
24. Roozbahani, Abbas, Hamed Ghased, and Mehdi Hashemy Shahedany. "Inter-basin water transfer planning with grey COPRAS and fuzzy COPRAS techniques: A case study in Iranian Central Plateau." Science of the Total Environment 726 (2020): 138499.