Republic of the Philippines
Pamantasan ng Cabuyao
(UNIVERSITY OF CABUYAO)

Academic Affairs
Division College of
Computing Studies

Katapatan Mutual Homes, Brgy. Banay-banay, City of Cabuyao, Laguna 4025

MIDTERM ASSESSMENT TASK
ITP110 – Web Technologies
First Semester, Academic Year 2024-2025

Ramos, Joshua
4IT-B

Introduction

Server-side scripting has transformed websites from simple, static pages to the dynamic, interactive experiences we expect today. This journey started in the mid-1990s and has evolved over decades, driven by advancements in technology and a growing demand for richer user experiences. In this paper, we'll explore the major turning points in server-side scripting, breaking down each era: its origins in the 1990s, the expansion of dynamic web development, the rise of scalable frameworks, and the modern full-stack frameworks that power today's web. By looking back at these innovations, we can see how server-side scripting has continually adapted to shape the modern web.

The Birth of Server-Side Scripting (Mid-1990s)

In the early days of the web, most sites were static, meaning they displayed the same content to every visitor without any customization or interaction. This changed when the Common Gateway Interface (CGI) arrived, allowing servers to execute scripts and create dynamic content. CGI scripts, often written in Perl, were the first step toward making websites interactive. Around this time, companies like Netscape started to push web technology forward, laying the groundwork for a more interactive web (National Center for Supercomputing Applications, n.d.). A breakthrough came in 1995 when Rasmus Lerdorf introduced PHP. Originally designed as a simple tool to track visits to his online resume, PHP's ability to embed directly within HTML made it easy for developers to create dynamic pages without needing to know complex programming languages. As a result, PHP quickly gained traction as a go-to for creating dynamic websites (Narasimhan, 1996).

The Expansion of Dynamic Web Development (1998-2005)

Between 1998 and 2005, server-side scripting really took off, especially with the release of Microsoft's Active Server Pages (ASP) in 1998. ASP allowed developers to build dynamic, database-driven applications within the Windows environment, which was appealing to many businesses. PHP also kept evolving, with PHP 3 bringing new functionality that helped it thrive in the open-source community.

Around this time, other options like JavaServer Pages (JSP) and ColdFusion became popular, especially among enterprise-level users who needed robust tools for web applications. Database integration also became standard, enabling websites to handle user data more effectively. This paved the way for e-commerce and user-centered sites, marking a shift where people could interact with websites through features like shopping carts, user accounts, and personalized content (McLaughlin, 2002; Welling & Thomson, 2008).

The Rise of Web Frameworks and Scalability (2005-2010)

As web applications grew more complex and user bases expanded, developers needed faster ways to build and scale sites. Frameworks like Ruby on Rails, launched in 2005, offered a solution. Rails introduced the Model-View-Controller (MVC) structure, which helped developers organize code and create applications more efficiently. This framework quickly gained popularity, with companies like Twitter and Airbnb using it to scale their platforms. Around the same time, Django for Python and Express.js for JavaScript also became popular, giving developers options for building scalable apps with less repetitive work. Frameworks allowed developers to reuse components and follow best practices, making it easier to manage large, complex projects and speeding up development timelines (Hansson, 2008; Holovaty & Kaplan-Moss, 2009).

Modern Server-Side Scripting and Full-Stack Frameworks (2010-Present)

With the release of Node.js in 2009, JavaScript expanded beyond front-end development, allowing developers to use it for server-side scripting. This made it possible to build full-stack applications entirely in JavaScript, which was a game-changer for real-time applications like messaging apps and collaborative tools. Since then, JavaScript frameworks like React and Angular have emerged, making it easier to build interactive, user-friendly interfaces. With the introduction of ECMAScript 6 (ES6), JavaScript gained new features that made it more robust and scalable for large applications (Haverbeke, 2018). JavaScript now powers not only web applications but also mobile and desktop apps through tools like React Native and Electron. It's even expanding into areas like machine learning with TensorFlow.js and high-performance computing with WEB Assembly, allowing JavaScript to handle tasks once reserved for other languages (Mozilla Developer Network, 2022).

Conclusion

Server-side scripting has come a long way, starting with CGI scripts and evolving into today's powerful full-stack frameworks. Each phase in its development has added new capabilities, allowing developers to build more interactive and responsive websites. Today's server-side technology enables complex, data-driven applications that can handle massive amounts of user data, personalize content, and respond to users in real-time. As server-side scripting continues to evolve, it remains central to the growth of the internet, helping developers meet the increasing demands of modern web users. Looking back, it's clear that each step in this journey has shaped the web as we know it today, and server-side scripting will undoubtedly keep pushing the web forward.

References

- National Center for Supercomputing Applications. (n.d.). *Common Gateway Interface (CGI)*. https://www.ncsa.illinois.edu

- Narasimhan, R. (1996). *The development of PHP*. https://www.php.net
- McLaughlin, B. (2002). *Java & XML Data Binding*. O'Reilly Media. https://www.oreilly.com
- Welling, L., & Thomson, L. (2008). *PHP and MySQL Web Development*. Addison-Wesley Professional. https://www.pearson.com
- Hansson, D. H. (2008). *Ruby on Rails 3.0*. https://rubyonrails.org
- Holovaty, A., & Kaplan-Moss, J. (2009). *The Django Book*. Apress. https://djangobook.com
- Haverbeke, M. (2018). *Eloquent JavaScript: A Modern Introduction to Programming*. No Starch Press. https://eloquentjavascript.net
- Mozilla Developer Network. (2022). *WebAssembly Concepts*. https://developer.mozilla.org/en-US/docs/WebAssembly