

The Influence of Programming Language Tools on Web Development Education among Computer Science Students

Usman Ismail Abdulmalik¹, Shamsuddeen Sani Muhammed², Emmanuel Babajide Adekunle³,
Murtala Hashimu⁴, Yakubu Hassan Zali⁵ and Nura Abubakar Anka⁶
Department of Computer Science, Federal Polytechnic, Kaura Namoda, Zamfara State, Nigeria.
usmaniamailbox@yahoo.com¹

Abstract

The rapid evolution of programming language tools has transformed web development education, offering new opportunities and challenges for computer science students. This study examines the impact of programming language tools on web development education among computer science students at the Federal Polytechnic Kaura Namoda, Zamfara State, Nigeria. The study used survey research design. Data were collected from 120 students across various academic levels (ND II, HND I and HND II) using a structured questionnaire. Simple random sampling technique was used. Descriptive statistical tools and Chi-Square test were used for statistical analysis. The results show that a majority of students (42.50%) are in their fourth year (HND II), with a near-equal distribution between male (52.50%) and female (47.50%) students. Regarding the types of programming languages used for web development, C# was the most commonly used (45.00%), followed by PHP (21.67%) and Java (14.16%). However, Python had minimal usage among students (6.67%) while C++ had no usage (0.00%). In terms of learning methodologies, 36.67% of students combined departmental courses, self-learning and private classes, highlighting the importance of diverse learning environments. The proficiency level of students in web development was varied, with 46.67% reporting low proficiency, while 38.33% reported moderate proficiency. The primary challenges students faced included intensive training requirements (61.67%), complex syntax (17.50%) and hardware requirement (9.16%). Despite these challenges, 42.50% of students acknowledged the high earning potential of web development, while 25.83% stated it improved their thinking skills. Moreover, the study found that 44.17% of students felt programming inspired them to experiment with new ideas, thereby fostering creativity. More so the study reveals that there is significant influence of programming language tools on web development education among computer science students in the Polytechnic. The findings emphasise the importance of integrating diverse programming languages learning tools into the curriculum, addressing the challenges students face and enhancing their creative and problem-solving abilities. This research suggests that curriculum design should focus on a holistic approach that incorporates both theoretical and practical learning to better prepare students for the dynamic demands of the tech industry.

Keywords: Web development; Programming languages; Education; Students.

1. Introduction

The field of web development is characterised by rapid technological advancements, requiring students to adapt to new tools and languages continuously. Programming language tools ranging from integrated development environments (IDEs) to libraries and frameworks, play a crucial role in shaping students' experiences in learning web development. These tools simplify the web development process, making it accessible to learners with diverse backgrounds and skill levels. With features designed to enhance usability and foster collaboration,

programming language tools can significantly improve students' engagement and motivation. It enable students to focus on creativity and functionality, allowing them to build applications without getting overwhelmed by complex syntax or technical barriers. This increased accessibility encourages exploration and experimentation, making web development a more inviting field for programmers. Also, many tools align closely with industry standards, providing students with relevant skills that enhance their employability.

Traditionally, building a website required a comprehensive understanding of programming languages like HTML, CSS, and JavaScript (Mamatha, Gurukiran, Basavaraj & Meghana, 2023). The integration and learning of programming is expected to help students develop useful skills, such as computational thinking, logical thinking, problem solving and creativity (Zhang & Nouri, 2019). Computer science students in the polytechnic are taught programming languages from their first year to their final year. During this time, they are exposed to various languages and learn how to design and develop a range of applications, including software, mobile apps and web applications. According to Carbonnelle (2024) in the PYPL Popularity of Programming Language Index (November, 2024), the top seven most popular programming languages are Python, Java, JavaScript, C#, C/C++, R and PHP. Moreover, these programming languages are among the most commonly taught in computer science programmes, providing students with a solid foundation for their future careers.

Conversely, the advent of no-code and low-code development platforms has transformed the landscape of software development, allowing users with minimal technical skills to build applications quickly and efficiently. While these tools offer some advantages, such as increased accessibility and reduced time to market, they also present significant challenges, particularly in educational contexts. No-code is a software development approach that creates applications and websites using visual drag-and-drop interfaces instead of code while low-code is a visual method of developing software that needs less hand-coding and enables the delivery of apps more quickly (Yadav & Narayana, 2024). However, the disadvantages to students are tremendous, some of which include limited templates, learning curve and limited room for creativity, vendor lock-in dilemma and lack of code ownership. Additional concerns are the cost of no code platforms, platform limitations, limited security and lack of scalability. Other issues include limited functionality, challenges of team collaboration and developer experience and limitations, among others (Kleszcz, 2023; Kiguolis, 2024; Rana, 2024). Some popular no-code platforms include Wix Studio, Oracle APEX, Quickbase, Zoho Creator, Kissflow, Voiceflow, Bubble, Airtable, Zapier and Mendix (Strehlow, 2024).

In contrast, the benefits of using programming language for web development are enormous and overwhelming. Learning programming offers numerous advantages. Coding is a highly valued skill in the job market and equips individuals with problem-solving abilities applicable to various careers. It involves providing instructions to computers in a language they can interpret. As the world becomes increasingly digital, many future jobs will require programming skills. Furthermore, learning to code enhances mathematical and technological skills that are relevant across multiple employment sectors (Ibrahim *et al.*, 2023). In today's digital era, programming is essential for students, especially as the economy evolves and more jobs are taken over by robots and artificial intelligence (AI). It remains a critical skill in both current and future job markets (Kim & Lee, 2016). While some believe programming skills are only useful for those in technical fields, basic programming knowledge can significantly enhance the value of team members who collaborate with developers. In higher education, programming teaches students how to decompose problems into manageable parts and devise step-by-step solutions using a language that computers can understand (Pang, Ishibuchi & Shang, 2020).

Thus, the purpose of this study is to examine the influence of programming language tools on web development education among computer science students, focusing on Federal Polytechnic, Kaura Namoda, Zamfara State, Nigeria. It aims to explore how these tools influence students' learning experiences, engagement and skill development in the context of web development. The study seeks to identify the programming languages used

by students, evaluate their proficiency levels, understand the methodologies they employ and examine the challenges they face. In addition, it aims to highlight the benefits of using programming languages for web development and how these skills contribute to students' overall educational and professional preparedness. By addressing these aspects, the study intends to provide insights that can enhance curriculum design and instructional strategies in web development education.

1.1 Research Questions

The following research questions were raised for the study:

- a) What are the programming languages used for web development by students?
- b) What are the proficiency levels of students in web development using programming languages?
- c) What are the methodologies employed by students in learning web development using programming language?
- d) What are the challenges students encountered in web development using programming language?
- e) What are the benefits of web development using programming language to students?

1.2. Research Hypothesis

The following research hypothesis were raised for the study:

- a) H 1: There is significant influence of programming language tools on web development education among computer science students in Federal Polytechnic, Kaura Namoda Zamfara, Nigeria.
- b) H 0: There is no significant influence of programming language tools on web development education among computer science students in Federal Polytechnic, Kaura Namoda, Zamfara, Nigeria.

2. Related Works

Nashiroh, Kamdi & Elmunsyah (2017) conducted research on the effectiveness of a web programming module designed using a scientific approach to enhance logical thinking skills among vocational high school students. The study aimed to (1) develop a web programming module that employs a scientific approach to improve students' logical thinking abilities and (2) assess the module's effectiveness in achieving this goal. The research resulted in a web programming module that incorporates a scientific approach into learning activities. The effectiveness evaluation revealed that the module was very effective in fostering logical thinking skills and enhancing learning outcomes. This conclusion was supported by several findings: (1) the average post-test score of students was 79.91, exceeding the minimum criterion; (2) the average logical thinking score among students was 82.98; and (3) students' overall response to the web programming module averaged 81.86%.

Medeiros, Ramalho & Falcão (2018) conducted a systematic literature review on teaching and learning introductory programming in higher education. The authors noted that, despite advancements in teaching methods and tools, dropout and failure rates remain high. Existing surveys and reviews often only cover studies published up to 2007 or focus primarily on teaching methods and tools. Their findings highlighted that the most commonly cited skills needed for learning programming include problem-solving and mathematical abilities. Problem-solving was also identified as a significant learning challenge, along with issues related to motivation, engagement and difficulties with programming syntax. The main teaching challenges include the lack of suitable methods and tools, as well as the need for scalable and personalised teaching approaches.

Annouychokanant, Boonlue, Chuathong & Thamwipat (2021) conducted a study on first-year students' attitudes toward programming in an Innovation in Educational Technology course. The aim was to explore the

relationship between students' attitudes toward programming, their gender and their learning performances. The study involved a survey of 90 freshmen who had some prior experience with block-based programming. The survey demonstrated a high reliability score of 0.93. The findings indicated no significant differences in attitudes toward programming between male and female students; however, notable differences were observed in learning performances regarding interest in programming, self-efficacy and creativity. Pairwise comparisons using Fisher's least significant difference (LSD) method revealed that low-performing students had the least favourable attitudes in these dimensions compared to their peers. This presents a challenge for educators to design learning activities that help improve the attitudes of low-performing students toward programming.

Sobral (2021) conducted a study focused on teaching and learning programming for students in science, technology, engineering and mathematics (STEM) degree programs. The author emphasised that introductory programming courses are essential for student success throughout their academic and professional journeys, despite traditionally high failure and dropout rates. The author noted the increasing number of publications and reviews on this topic and recognised the need for an umbrella review to synthesise previous studies and derive more comprehensive insights. Utilising databases such as ACM, Google Scholar, IEEE, SCOPUS and Web of Science, the formal search yielded 21 reviews related to programming fundamentals in higher education. The results provided bibliometric data on CS1 reviews, including trends over time, author information, sources, affiliations, languages, keywords and the most cited articles. The study also categorised the reviews based on four distinct purposes: general, specific, student group and teacher-directed. The author concluded that these catalogued findings could serve as valuable resources for future research and reviews concerning introductory programming, particularly in higher education contexts.

Ibrahim *et al.* (2023) explored the significance of integrating programming languages into the secondary school curriculum in Nigeria, emphasising the increasing demand for coding skills in the job market and their contribution to improving problem-solving abilities. The study highlights the positive effects of learning programming on students' communication, creativity and confidence, preparing them for a digital future. Findings indicate that incorporating programming languages into the curriculum enhances creativity, problem-solving skills and academic performance, with peer group influence playing a supportive role. The study concludes that programming education is highly beneficial for students and advocates for sufficient resources to ensure effective implementation, particularly noting the proficiency gains among students learning languages such as C++.

However, despite the growing importance of programming languages tools in web development education, there is a noticeable gap in research that comprehensively examines the influence of these tools on students' learning experiences, particularly in the context of polytechnic education in Nigeria. While many studies have explored the general benefits of programming language education (Ibrahim *et al.*, 2023) or analyses challenges in introductory programming courses (Medeiros, Ramalho & Falcão, 2018), fewer have specifically focused on how programming language tools (e.g., IDEs, libraries and frameworks) impact the learning process in web development. This study, therefore, seeks to fill these gaps by focusing on the specific impact of programming language tools on web development education among computer science students focusing on Federal Polytechnic, Kaura Namoda, Nigeria. It aims to explore how these tools influence students' learning experiences, engagement and skill development, as well as the benefits and challenges students face. By addressing these aspects, the study intends to provide new insights that could guide the evolution of web development curricula in similar educational settings.

3. Methodology

This study adopted a survey research design to determine the influence of programming language tools on web development education among computer science students. The population of the study comprises students in National Diploma II (ND II), Higher National Diploma I (HND I) and Higher National Diploma II (HND II) who have completed at least two semesters. Data were collected using a self-structured questionnaire. The questionnaire contained close ended questions, a simple random sampling technique was used to select a sample of 120 students from computer science department at Federal Polytechnic, Kaura Namoda, Zamfara State, Nigeria. The questionnaires were administered through face-to-face interactions with the respondents in their classrooms to ensure effective feedback. Due to the nature of this direct interaction and the conditions established, all questionnaires were returned as valid. Data collection was conducted and subsequently coded and analysed using IBM SPSS version 22. Furthermore, descriptive statistics and Chi-Square test were used to analyse the data obtained.

4. Results and Discussion

This section aims to analyse and interpret the data gathered during the study. The data was collected using close-ended questionnaires, with respondents answering questions designed to address the research questions. Descriptive analysis was performed, using relevant percentages as shown in the charts.

Table 1: Demographic distribution of respondents

Characteristics	Variables	Frequency	Percentage (%)
Gender	Male	63	52.50
	Female	57	47.50
Year of study	ND II	26	21.67
	HND I	43	35.83
	HND II	51	42.50

Source: Fieldwork (2024)

a. Demographic distribution of respondents

The demographic distribution of respondents in Table 1 reveals the composition of the study's participants. In terms of gender, the distribution is nearly balanced, with 52.5% male and 47.5% female students. This suggests that both genders are equally represented in the sample, which is important for ensuring that the findings are relevant and applicable to all students, regardless of gender. The relatively small difference in gender distribution indicates that programming education is accessible to both male and female students in the study, supporting efforts to increase female participation in STEM (Science, Technology, Engineering and Mathematics) fields, including computer science.

Regarding the year of study, the distribution shows that the largest group of respondents are in their fourth year (HND II), representing 42.50% of the sample, followed by third-year students (HND I) at 35.83% and second-year students (ND II) at 21.67%. This distribution indicates a broad range of experience levels. Fourth-year students are expected to have the most advanced knowledge and proficiency in web development, having completed more complex courses and projects. In contrast, second-year students are still in the foundational stages of their education, focusing on basic programming languages and web development concepts. The variety in academic levels allows the study to capture how students' use of programming language tools evolves as they progress through their studies, with different levels of familiarity and expertise with industry-standard tools.

The demographic breakdown indicates that the sample is diverse in terms of both gender and academic experience. This diversity is valuable as it provides a comprehensive view of how different groups of students engage with programming languages and web development. Moreover, the gender balance and variation in year of study will allow the study to explore whether these factors influence students' experiences and proficiency in web development.

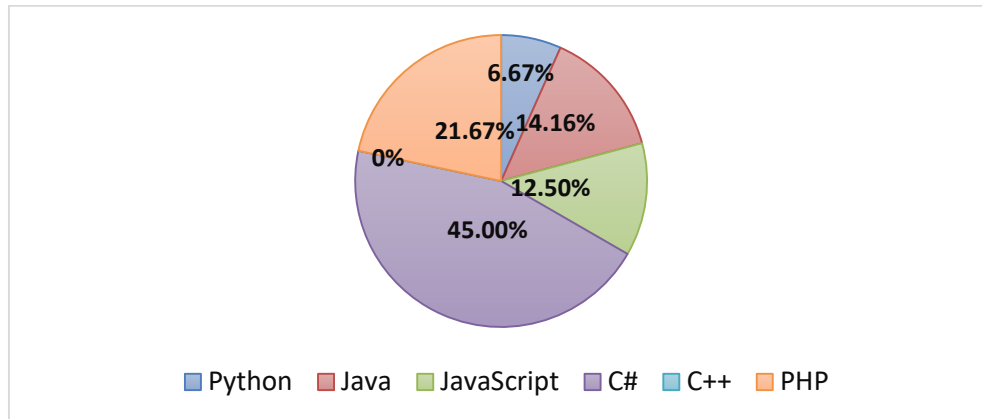


Figure 1: Type of programming language used by students

b. Type of programming language used by students for web development

The results presented in Figure 1 reveal the type of programming language used by students for web development. Among the participants, C# is the most commonly used language, with 45.00% of students indicating its use. This is followed by PHP, with 21.67% of students using it. Java ranks next, with 14.16%, while JavaScript is reported by 12.50% of students. Notably, Python is used by 6.67% of students and there are no respondents using C++. This distribution illustrates the preferences among students for certain programming languages, with C# being the predominant choice in web development tasks. The results show a strong preference for C# and PHP, with a moderate use of Java and JavaScript and minimal use of Python. The absence of C++ aligns with the expectation that this language is less relevant to web development.

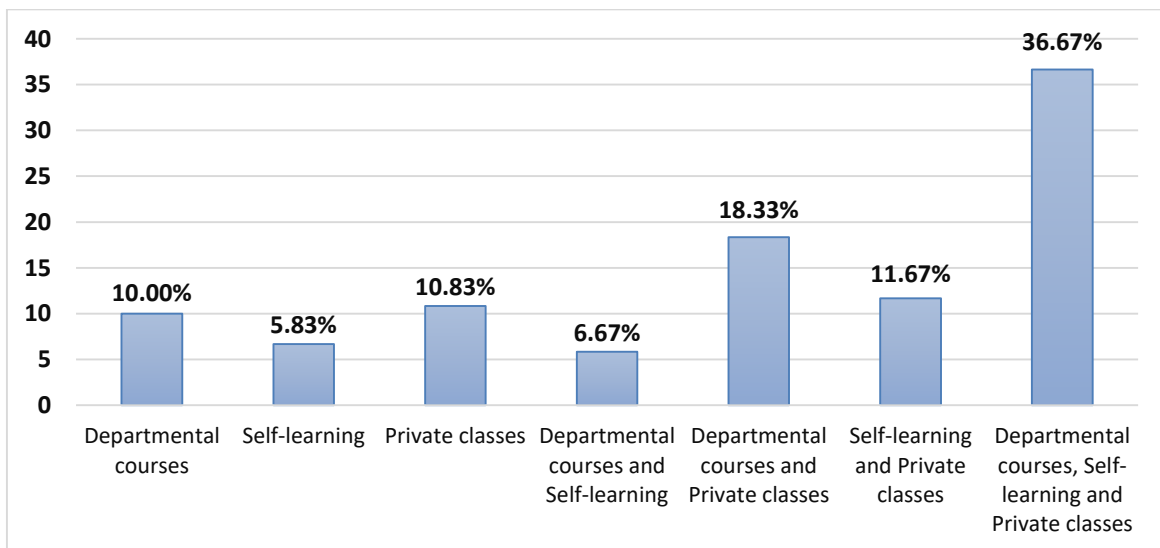


Figure 2: Web development using programming language learning methodology

c. Web development using programming language learning methodology

The results presented in Figure 2 indicate a diverse range of learning methodologies employed by students in learning web development using programming languages. The most common approach, used by 36.67% of students, is a combination of departmental courses, self-learning and private classes. The second most popular methodology, employed by 18.33% of students, is the combination of departmental courses and private classes. Self-learning and private classes together are used by 11.67% of students, which highlights that a significant portion of students prefer to take responsibility for their own learning and supplement their studies with private classes. Only 10.00% of students rely solely on departmental courses, which suggests that while formal education is fundamental, it may not always provide all the necessary tools and knowledge that students need to excel in web development. Similarly, 6.67% of students relies on a combination of departmental courses and self-learning. A smaller group, 5.83% of students are motivated enough to independently learn programming languages, they represent a smaller portion of the overall population and 10.83% of students opt for private classes alone, indicating that students are generally blending different approaches to meet their learning needs. The data reveals the importance of a blended approach in learning web development, suggesting that students thrive when they have access to various resources and methodologies.

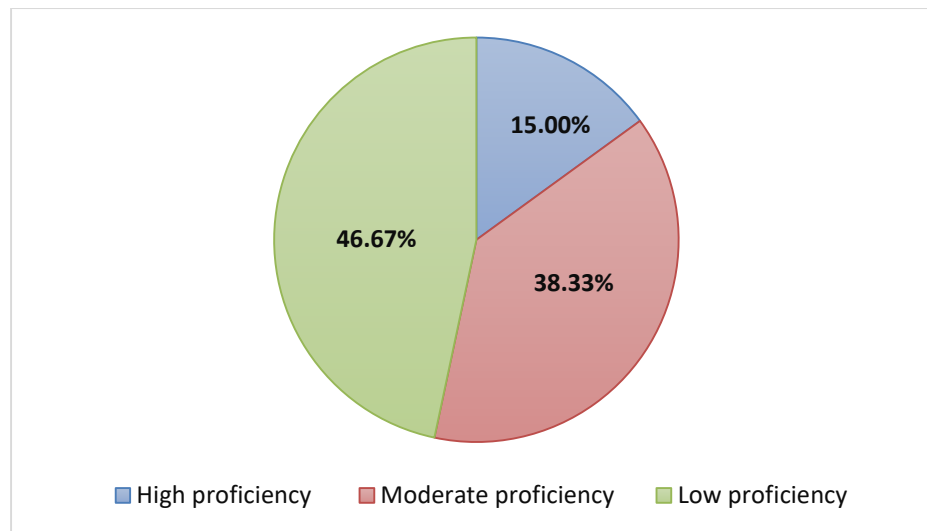


Figure 3: Students' proficiency level in web development using programming language

d. Students' proficiency level in web development using programming language

The results presented in Figure 3 show students' proficiency level in web development using programming language. The results indicates that a significant portion of students, 46.67%, report having low proficiency in web development. A smaller group of students, 38.33%, possess moderate proficiency, indicating that these students have a reasonable understanding of web development concepts and programming languages but may still need to refine their knowledge and gain more practical experience. Moreover, only 15% of students report high proficiency in web development, meaning that a relatively small proportion of the respondents have reached an advanced level in using programming languages for web development. These students are likely to have a strong grasp of the fundamental concepts and may be capable of building sophisticated web applications.

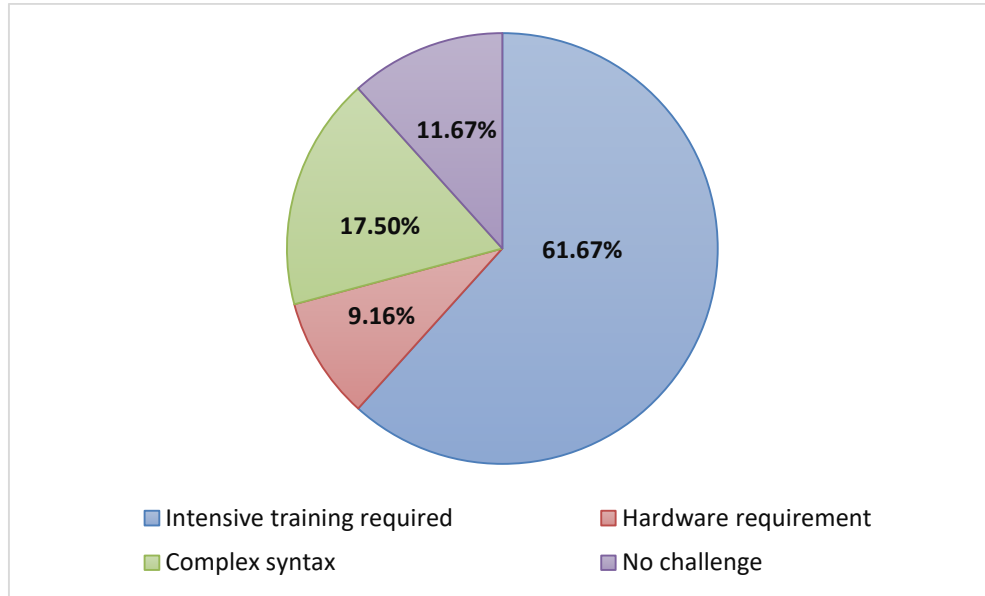


Figure 4: Challenges students encountered in web development using programming language

e. Challenges students encountered in web development using programming language

The results presented in Figure 4 reveal the challenges students encountered in web development using programming language. The results indicated that the most significant challenge identified by 61.67% of the students is the need for intensive training. Another notable challenge, reported by 17.5% of students, is complex syntax. A smaller percentage of students, 9.16%, cited hardware requirements as a challenge. Interestingly, 11.67% of students reported having no challenges in web development using programming languages. This group might have experienced fewer difficulties or found the learning process relatively straightforward, which could reflect prior experience or a greater affinity for programming.

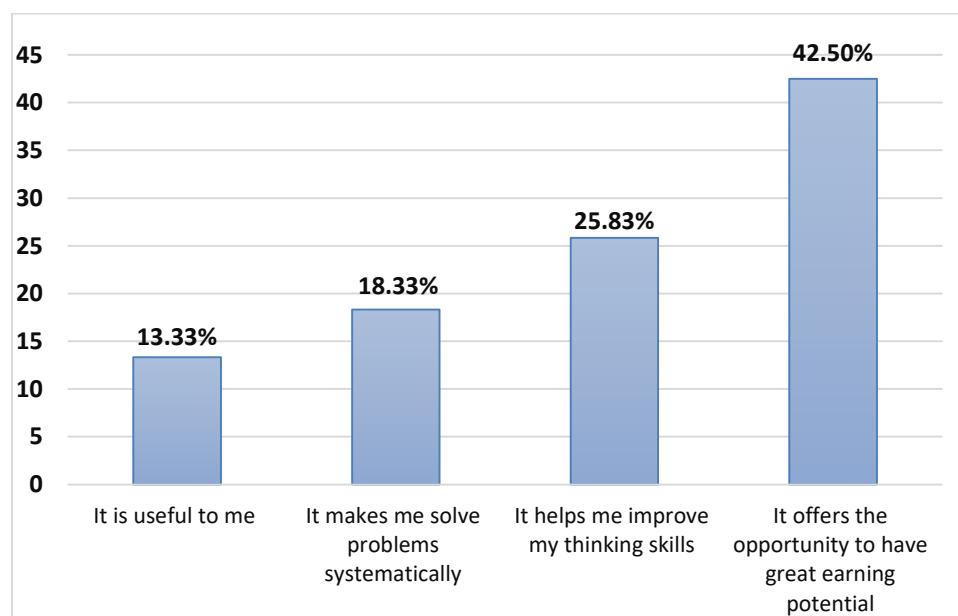


Figure 5: How meaningful is web development using programming language to students?

f. How meaningful is web development using programming language to students?

The results presented in Figure 5 show the meaningfulness of web development using programming languages to students. A total of 42.50% students highlighted the potential for significant earning opportunities as a key benefit, indicating a strong economic motivation for their interest in this field. In addition, 25.83% of students reported that web development helps improve their thinking skills, while 18.33% of students noted that it enables them to solve problems systematically. Lastly, 13.33% of students found web development to be useful in general.

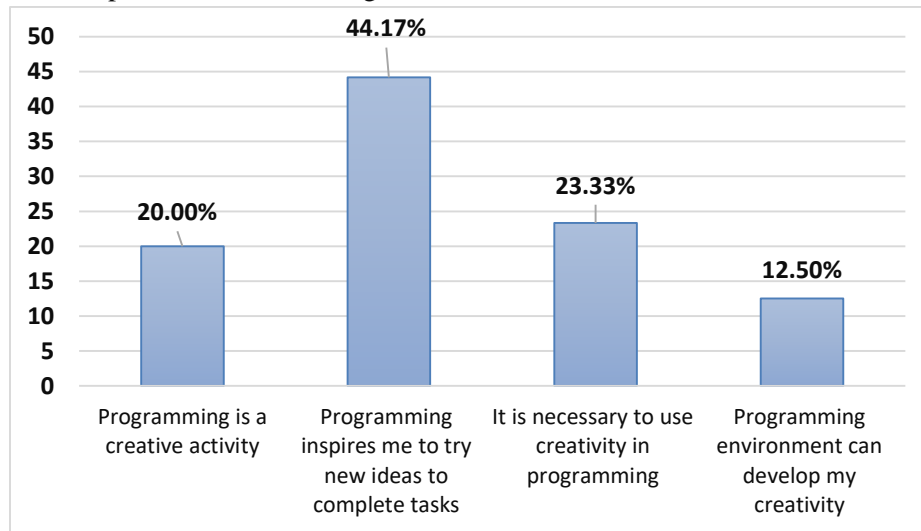


Figure 6: How web development using programming language improve students' creativity?

g. How web development using programming language improve students' creativity?

The results presented in Figure 6 reveal how web development using programming languages enhances students' creativity. A notable 44.17% of students indicated that programming inspires them to try new ideas when completing tasks, highlighting the role of programming as a catalyst for creative exploration. Moreover, 23.33% of students believe that creativity is essential in the programming process, suggesting that innovative thinking is integral to effective coding. Furthermore, 20.00% of students see programming as a creative activity in itself, while 12.50% of students recognise that the programming environment can foster their creative development.

Test of Hypothesis

- a) H_1 : There is significant influence of programming language tools on web development education among computer science students in Federal Polytechnic, Kaura Namoda Zamfara, Nigeria.
- b) H_0 : There is no significant influence of programming language tools on web development education among computer science students in Federal Polytechnic, Kaura Namoda, Zamfara, Nigeria.

Table 2: Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	181.562	9	.000

Likelihood Ratio	148.078	9	.000
Linear-by-Linear Association	96.372	1	.000
N of Valid Cases	120		

Source: Fieldwork (2024)

From the Chi-Square tests in Table 2, the results of the test of hypothesis indicated a rejection of the null hypothesis at $P < 0.000$ level implying that there is significant influence of programming language tools on web development education among computer science students in Federal Polytechnic, Kaura Namoda, Zamfara State, Nigeria.

The results discussed in line with each of the research questions. The first research question was “What are the programming languages used for web development by students?” The results reveal that there is significant influence of programming languages tools in web development education among computer science students in Federal Polytechnic, Kaura Namoda, Zamfara State, according to the probability value of one percent statically significant.

The second research question was “What are the proficiency levels of students in web development using programming languages?” The results show that 46.67% of respondents report low proficiency, raising concerns about the effectiveness of current educational methods and potentially limiting students' readiness for real-world web development tasks. In contrast, only 15.00% of students achieve high proficiency, highlighting a critical shortage of advanced skills. Meanwhile, 38.33% of students exhibit moderate proficiency, suggesting a foundational understanding but a need for further support to advance to more complex aspects of web development. These findings point to the necessity for educational institutions to reassess their curriculum and teaching strategies. Incorporating more hands-on projects, real-world applications and collaborative learning opportunities could help bridge the gap between theory and practice. In addition, mentorship programs and peer collaboration could foster a more supportive and dynamic learning environment. The low proficiency levels may also be influenced by factors such as limited access to resources, insufficient practice opportunities, or challenges within the learning environment. Addressing these issues is crucial for improving student outcomes in web development. The study suggests that targeted interventions, including innovative teaching methods and enhanced support mechanisms, are essential to better prepare students for careers in web development. Future research could explore the underlying causes of low proficiency and identify strategies for improvement.

The third research question was “What are the students' proficiency level in web development using programming language?” The results reveal that a combination of departmental courses, self-learning and private classes is the most common approach, used by 36.67% of students. This suggests that students value a holistic learning experience that blends formal instruction with independent study and additional support. A significant portion (18.33%) also relies on a combination of departmental courses and private classes, emphasising the importance of structured learning and personalised instruction to fill knowledge gaps. Moreover, self-learning alone is less common, with only 5.83% of students using this method. In addition, the results show that 10.00% of students depend solely on departmental courses, which highlights the importance of a strong, comprehensive curriculum. However, relying only on formal education may limit exposure to the latest tools and technologies typically discovered through self-learning. The findings indicate that a blended learning approach is favoured by students, combining various methodologies to enhance their understanding and skills in web development. As the field continues to evolve, educators should consider integrating more interactive and flexible learning options to cater to diverse learning styles. Future research could explore the

effectiveness of these various methodologies in terms of student outcomes and satisfaction, providing insights into how to optimise web development education further.

The fourth research question was “What are the challenges students encountered in web development using programming language?” A significant 61.67% of respondents expressed the need for intensive training, highlighting a major gap in their skills and knowledge. This suggests that the current curriculum may not adequately prepare students for the complexities of web development, signalling a need for more comprehensive, hands-on learning experiences. To address this, educational institutions could integrate practical workshops, project-based assignments and mentorship opportunities to give students real-world exposure and boost their confidence. In contrast, fewer students reported issues with hardware requirements (9.16%) and complex syntax (17.50%), indicating these are less significant barriers, although they may still present challenges for some students. A small group of 11.67% reported facing no challenges, which suggests that a subset of students may have developed effective strategies or support systems, potentially offering valuable insights for peer mentorship. The results suggest that educational programmes should focus on providing more intensive, experiential learning opportunities and strengthening support systems to better equip students for the demands of web development. Future research could explore specific training needs and effective teaching strategies to overcome the challenges students encounter.

The fifth research question was “What are the benefits of web development using programming language to students?” Focusing on its meaningfulness and impact on students' creativity. In terms of meaningfulness, the majority of students recognise significant value in learning web development, particularly for its career and cognitive benefits. Notably, 42.20% of students view web development as a pathway to high earning potential, reflecting the growing demand for web developers and the economic incentives driving students toward this field. In addition, 25.83% of students believe that web development enhances their thinking skills, indicating that engaging with programming promotes critical thinking and analytical abilities. Moreover, 18.33% noted that it helps them solve problems systematically fostering structured thinking and effective problem-solving strategies. These cognitive benefits are valuable not only in tech fields but also in a wide range of other professions. However, 13.33% of students reported that web development is useful to them, suggesting that while a smaller group finds the subject directly beneficial to their current academic or professional endeavours, many students are still motivated by the broader advantages of acquiring programming skills. The findings suggest that web development is perceived as a meaningful endeavour by students, with a clear emphasis on its economic advantages and its role in fostering essential cognitive skills. Educational institutions should leverage this insight to design curricula that not only enhance technical skills but also highlight the long-term benefits of web development, ultimately preparing students for successful careers in an increasingly digital world.

Furthermore, in terms of impact on students' creativity, 44.17% of students report that programming inspires them to explore new ideas when completing tasks, suggesting that web development fosters an innovative mind set and encourages creative problem-solving. This indicates that engaging with programming goes beyond technical skills, enabling students to think outside the box and experiment with different solutions. Furthermore, 23.33% of students recognise the necessity of creativity in programming itself, reinforcing the idea that coding is not just a mechanical process but one that requires imaginative thinking to create effective and unique solutions. As programming becomes more integrated across various industries, fostering creativity in students is vital to prepare them for careers that demand both technical expertise and innovative thinking. While 20.00% of students view programming as a creative activity, there is still potential to further cultivate this perception. Educators could enhance this by incorporating more project-based learning and creative assignments that encourage students to express their creativity through code such as designing visually appealing websites or developing interactive applications. The fact that only 12.50% of students believe the programming

environment itself can develop their creativity suggests room for improvement in how programming spaces are designed. More engaging and flexible physical and digital environments could better stimulate creativity and enhance the learning experience. The findings emphasise the importance of web development as a means to boost creativity. By integrating both technical and creative skill development into curricula, educational institutions can better prepare students for the evolving demands of the tech industry, fostering well-rounded individuals capable of meeting both technical and innovative challenges.

5. Conclusion

This study examines the significant impact of programming language tools on web development education for computer science students at Federal Polytechnic, Kaura Namoda, Zamfara State, Nigeria. The findings reveal a variety of programming languages in use for web development, with C# being the most prevalent (45.00%), followed by PHP (21.67%) and Java (14.16%), pointing to a diverse engagement with various tools. However, proficiency levels are relatively low, with 46.67% of students self-assessing as having low proficiency. Furthermore, the exploration of learning methodologies revealed that many students rely on a combination of departmental courses and self-learning, with 38.67% utilising all three learning approaches together, underscoring the importance of diverse educational strategies. The main challenges faced include the intensive training required (61.67%) and complex syntax (17.50%). Despite these challenges, students recognise the value of web development in terms of career potential, with 42.50% acknowledging its earning potential and 25.83% noting improvements in their thinking skills. Moreover, students view programming as a creative process, with 44.17% stating that it inspires them to experiment with new ideas. The result further reveals that there is significant influence of programming language tools on web development education among computer science students in the Polytechnic. These findings suggest that while students face difficulties in mastering web development, they value its benefits and see it as an essential skill for their future careers. It is recommended that educational institutions provide more hands-on training, diverse learning resources and personalised support to enhance students' proficiency and foster creativity.

The researchers will like to recommend the following:

1. Educational institutions should revise their web development curricula to incorporate a broader range of programming languages and modern development tools. This should include hands-on training with popular programming languages and frameworks to align with industry standards.
2. Institutions should invest in resources such as tutorial sessions, workshops and access to online learning platforms to provide students with additional support in mastering programming languages.
3. Educators should adopt innovative teaching methodologies that emphasis active learning, collaboration and problem-solving, allowing students to engage more deeply with the material.
4. Institutions should implement mentorship initiatives that pair students with experienced professionals can provide guidance and enhance practical understanding of web development concepts.
5. Institutions should actively seek to understand and address the specific challenges students face, such as complex syntax and the need for intensive training, by providing targeted support and resources.
6. Institutions should encourage group projects and peer-to-peer learning can foster collaboration and enhance students' learning experiences, allowing them to share knowledge and skills.
7. Regular assessments of student proficiency and feedback mechanisms should be established to monitor learning outcomes and make necessary adjustments to teaching strategies and resources.

References

- Amnouychokanant V., Boonlue, S., Chuathong, S. and Thamwipat, K. (2021). A study of first-year students' attitudes toward programming in the innovation in educational technology course. *Hindawi Education Research International*, 2021, 1-10. <https://doi.org/10.1155/2021/910534>
- Carbannelle, P. (2024). *PYPL (Popularity of Programming Language)*. Retrieved November 01, 2024 from <https://pypl.github.io/PYPL.html>
- Ibrahim, A. A., Haruna, S. A., Ayuba, D., Usman, B., Abdullahi, M. I. (2023). Adoption of programming language in nigerian secondary schools curriculum. *Nigerian Journal of Computing, Engineering and Technology*, 2(2), 289-300. <https://nijocet.fud.edu.ng/wp-content/uploads/2024/02/NIJOCET-VOL2-ISSUE-2-019.pdf>
- Kiguolis, L. (2024). *7 No code drawbacks to consider. codeornocode*. Retrieved November 01, 2024 from <https://codeornocode.com/no-code/no-code-is-not-a-good-idea/>
- Kim, S. W. and Lee, Y. (2016). The effect of robot programming education on attitudes towards robots. *Indian Journal of Science and Technology*, 9(24), 1-11. 10.17485/ijst/2016/v9i24/9610
- Kleszcz, M. (2023). *The top 5 drawbacks of using no-code and low-code tool*. Apptension. Retrieved November 01, 2024 from <https://www.apptension.com/blog-posts/no-code-and-low-code-limitations>
- Mamatha, S. K., Gurukiran, A. C., Basavaraj and Meghana (2023). No-code web development. *International Research Journal of Modernization in Engineering Technology and Science*, 5(6), 2959-2965. <https://www.doi.org/10.56726/IRJMETS42320>
- Medeiros, R. P., Ramalho, G. L. and Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 1-14. http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
- Nashiroh, P. K., Kamdi and W., Elmunsyah, H. (2017). The effectiveness of web-programming module based on scientific approach to train logical thinking ability for students in vocational high school. *The Proceedings of AIP Conference. Green Construction and Engineering Education for Sustainable Future* 1887, 020068 (2017). <https://doi.org/10.1063/1.5003551>
- Pang, L. M., Ishibuchi, H. and Shang, K. (2020). Decomposition based multi-objective evolutionary algorithm design under two algorithm frameworks. *IEEE Access*, 8, 163197–163208. 10.1109/ACCESS.2020.3022164
- Rana, K. (2024). *Disadvantages of no-code apps*. 3veta.com. Retrieved November 01, 2024 from <https://3veta.com/blog/business-advice/advantages-and-disadvantages-of-no-code-apps-and-platforms/>
- Sobral, S. R. (2021). Teaching and learning to program: umbrella review of introductory programming in higher education. *Mathematics MDPI* 9(15), 1-23. <https://doi.org/10.3390/math9151737>
- Strehlow, R. (2024). *Low-code no-code web development, explained*. Wix Studio. Retrieved November 01, 2024 from <https://www.wix.com/studio/blog/low-code-no-code>
- Yadav, D. C. and Narayana, P. (2024). Low code/no code development. *International Journal for Research in*

Science Engineering & Technology, 11(5), 1-5. <https://ijrset.in/index.php/ijrset/article/view/650>

Zhang, L. and Nouri, J. (2019). A systematic review of learning computational thinking through scratch in K-9. *Computers & Education*, 141 (2019), 1-25. <https://doi.org/10.1016/j.compedu.2019.10360>