Enhancing Programming Education Through Interactive Websites: A Focus on AJAX, JSON, and Java for Loops

Harika Dharavath
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

Jessica Felix
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

Suhair Amer
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

Ankita Maharjan
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

ABSTRACT
This paper discusses the development of interactive websites to teach AJAX, JSON, and for-loops in Java. The project integrates web technologies to create an engaging platform that enhances programming comprehension e-learning. A comparison is made between using HTML, CSS, and JavaScript versus the Wix platform for web development. HTML and CSS provide structure and visual design, while JavaScript enables interactivity, data handling, and real-time demonstrations of programming logic. Wix, although user-friendly and efficient for quick design, offers limited customization and scripting control. The findings show that traditional coding tools better support the learning of programming concepts, while Wix is suitable for simplified design and presentation. With both implementations, the interactive system effectively combines web development and Java programming to promote active learning and improve student understanding of key technical concepts.

Keywords
AJAX, JSON, Java, For Loops, HTML, CSS, JavaScript, Wix, Interactive Learning, Web Development, Programming Education, Human–Computer Interaction, Educational Technology

1. INTRODUCTION
In today's digital world, learning to code has become increasingly important. In this era of digital platforms, coding is the backbone of everything, from websites and mobile apps to software systems and even hardware operations [1]. However, when it comes to learning, many people find it challenging when lessons consist only of reading and watching videos. Imagine learning to code not by reading long manuals, but by clicking, typing, and seeing instant results. In recent times, interactive websites are transforming how we teach and learn, turning passive lessons into active experiences. This paper focuses on designing interactive platforms to teach key programming concepts, particularly Ajax (Asynchronous JavaScript and XML), JSON (JavaScript Object Notation), and control structures such as loops. Many studies show that interactive tools help students learn better. The educational benefits of interactive learning environments in computer science education are highlighted by recent studies. The usefulness of web-based and mobile technologies for teaching programming logic is highlighted in a systematic review by Ed.gov, which also emphasizes how elements like gamification and real-time feedback greatly improve learner motivation and comprehension [2]. Wang's course design highlights the wider benefits of web-based platforms in providing scalable, interactive learning experiences by illuminating the successful integration of web technologies into programming education [3].

Interactive e-learning systems let students practice and see programming principles right in the browser by offering an online editor with features like auto-complete, syntax highlighting, and live code execution. Additionally, it has performance tracking and communication capabilities that let teachers keep an eye on students' progress and modify their lessons. Particularly for novices studying web development, this type of integrated, web-based approach improves accessibility and engagement [4]. For younger students, web-based learning platforms have demonstrated efficacy in enhancing the accessibility and engagement of programming education. Codeable [5], a dynamic web tool that combines an interactive code editor, visualizations, and video tutorials to teach JavaScript principles, is one example. It turns abstract concepts like loops and functions into concrete experiences by letting students write and run code with instantaneous graphical feedback. This method is appropriate for middle school students since it not only encourages active learning but also lessens the anxiety that is frequently connected to instruction that is heavily reliant on syntax [6]. Some online courses already use these tools to teach coding. For example, websites like Coursera [7] and Codecademy [8] let users write code and see what it does. They provide dynamic, accessible, and structured learning environments for students of all skill levels, from total novices to highly skilled developers. While Codecademy concentrates on interactive, hands-on coding practice through embedded code editors and instant feedback, making it better suited for learners who prefer learning by doing and developing practical skills, Coursera offers structured, university-style courses with video lectures and certificates, perfect for students seeking academic depth and professional credentials [9] [10].

Online coding environments like HackerRank and Leetcode have become useful resources for remote programming instruction, building on the success of intelligent and gamified platforms. These systems provide situational activities, real-time feedback, and structured challenges that strengthen useful coding abilities. Without requiring elaborate exhibits or actual labs, instructors can improve student engagement and offer hands-on experience by incorporating them into university programs. Through

interactive, self-paced practice, this method not only facilitates remote learning but also develops professional competencies [11]. The assessment of online coding platforms in secondary and postsecondary education demonstrates their revolutionary role in programming instruction, extending the conversation on distant learning and skill development. These platforms, which include freeCodeCamp, HackerRank, and Codecademy, provide gamified, interactive settings that encourage community involvement, self-paced learning, and real-time feedback. The study highlights how these tools democratize access to programming instruction across a range of socioeconomic backgrounds while simultaneously improving conceptual knowledge and retention [12]. Jung et al. [13] and Eng et al. [14] both add to the expanding corpus of work supporting browser-based, interactive programming environments that facilitate active learning and remove installation obstacles. By recording live coding actions, such as typing and scrolling, and replaying them in the browser, Eng et al.'s ITSS platform gives students a dynamic, tutorial-like experience that boosts motivation and comprehension [14]. Similarly, Jung et al. suggest a dual-mode system that uses Blockly to combine syntax-based coding and visual programming, enabling students to construct logic using drag-and-drop blocks that translate into C code [13]. Their architecture strengthens both conceptual and practical skills by enabling real-time compilation and execution. Collectively, these studies demonstrate how cloud-based, integrated platforms can revolutionize programming education by increasing their pedagogical efficacy, accessibility, and engagement.

Recent studies continue to emphasize web-based platforms' effects on student performance, engagement, and accessibility, which serves to further support their worth in programming education. Alvarado and Rodríguez's assessment of Teloprogramo shows how real-time progress tracking and personalized education can be achieved using adaptive learning systems, which can result in better results and long-lasting motivation [15]. According to Alghamdi's research, students who use interactive e-learning tools perform better than their classmates in both practical coding tasks and conceptual knowledge, confirming the usefulness of digital platforms in formal educational contexts [16]. Together, the studied literature highlights how web-based platforms have revolutionized programming education. Iterative learning and skill improvement are supported by tools like HackerRank and LeetCode, which provide structured challenges and real-time feedback, especially for algorithmic thinking and technical interview preparation. With its interactive and visual design, Codeable shows how gamified, browser-based environments may help middle school students understand programming principles. Programming education is now more accessible and self-paced thanks to platforms like Codecademy and Coursera, which offer scaffolded, curriculum-driven learning with integrated examinations and community assistance. The incorporation of such technologies becomes crucial as educational institutions embrace remote and hybrid models more frequently. This is because doing so fosters professional preparation and computational thinking in addition to content delivery. Future studies into improving platform design, increasing access, and adapting instruction to different educational environments are made possible by this foundation.

2. HTML, CSS, AND JAVASCRIPT DEVELOPMENT OF AJAX & JSON (JAVASCRIPT INTERACTIVE QUIZ)

The project aims to create a web-based application that provides users with an interactive quiz experience, enhances their understanding of Ajax & JSON concepts, and showcases their knowledge through a scoring system. The webpage was developed using HTML, CSS and Java Script.

2.1 Design

The design of the project encompasses two main components.

For the programming component:

- It involves the implementation of server-side logic and data management.
- The back-end development will utilize technologies such as Node.js, Django, or Flask to handle requests and responses.
- A database or data storage mechanism will be employed to store quiz questions, their attributes, and user scores.
- A data model/schema will be designed to ensure efficient storage and retrieval of quiz-related information.
- Scoring logic will be developed to weigh the questions based on their difficulty levels.

For the interface component:

- It focuses on the front-end development, creating visually appealing and user-friendly web pages.
- HTML, CSS, and JavaScript will be used to design and structure the web pages.
- Attention will be given to responsive design, ensuring that the application works seamlessly across various devices and screen sizes.
- CSS styling will be applied to enhance visual aesthetics and provide engaging user experience.
- User interactions, such as navigating between questions, providing answers, and receiving feedback, will be implemented using JavaScript.

2.2 Implementation

For Front-End development:

- HTML, CSS, and JavaScript are used to design visually appealing web pages.
- A responsive layout is implemented to ensure the web application works well on different devices and screen sizes.
- Separate pages are created for the book details, quiz instructions, each question, and the final score.
- Navigation between questions is achieved using a "next" button/link.
- AJAX requests are made to fetch quiz questions and submit answers asynchronously.

For Back-End development:

- Node.js is chosen as the server-side framework to handle requests and responses.
- Express.js is used as the web application framework to define routes and endpoints.
- MongoDB is selected as the database to store quiz questions and their attributes.
- A QuizQuestion schema is created to define the structure of quiz questions in the database.
- Scoring logic is implemented to weigh questions based on their difficulty level.
- The AI assistant functionality is developed using a predefined set of questions and their answers.

For integration:

- Client-side JavaScript communicates with the server-side using AJAX requests and JSON format.
- The front-end sends requests to the back-end to fetch quiz questions and submit answers.

- The back-end processes the requests, interacts with the database, and sends appropriate responses.
- The AI assistant responds to predefined questions by fetching the corresponding answer from the json database.

For styling and user experience:

- CSS is used to style the web pages, making them visually appealing and consistent.
- The layout is optimized for different screen sizes and devices using responsive design techniques.
- User experience factors such as responsiveness, accessibility, and intuitive navigation are considered.
- Animations or transitions may be incorporated to enhance the user experience.

2.3 Testing and evaluation

Upon launching the application, user will be presented with the initial page displaying the book name, chapter number, chapter topic, and quiz instructions. To begin the quiz, simply click the "Start Quiz" button [Figure 1].

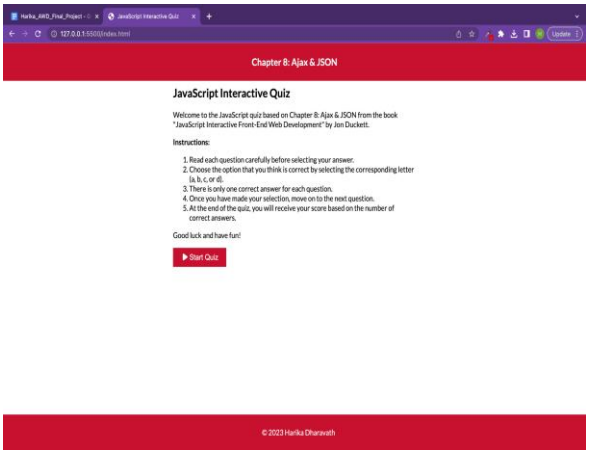

Fig 1. Initial page.

In general, to take the quiz, each question will be displayed on a separate page [Figure 2]. The user will read the question carefully and select the most appropriate answer from the options provided. After they choose an answer, they need to click the "Next" button to proceed to the next question. Once the user submits an answer, the application will provide immediate feedback indicating whether the answer is correct [Figures 3 and 4] or incorrect. If the answer is incorrect [Figures 5 and 6], the correct answer will be displayed, helping the user to learn from their mistakes. The user will continue answering questions until reaching the final page.

In specific, at first the page shows the book name, "JAVASCRIPT," the chapter number, "Chapter 8," the chapter topic, "Ajax & JSON," and the quiz instructions. A "Start Quiz" button is visible. After starting the quiz, the question-and-answer options are presented. A "Verify" button is provided to proceed to verify the question after selecting an answer, whether it is correct or not. A "Next" button is provided to proceed to the next question after selecting an answer. After answering each question, the screen shows feedback indicating whether the answer is correct or incorrect, along with the correct answer for incorrect responses.

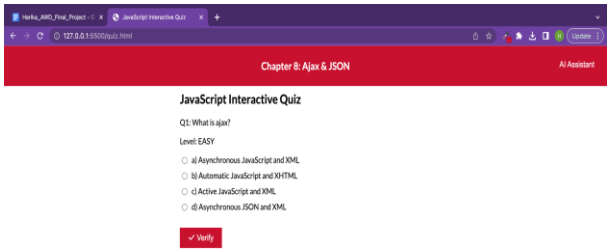

Fig 2. Quiz page

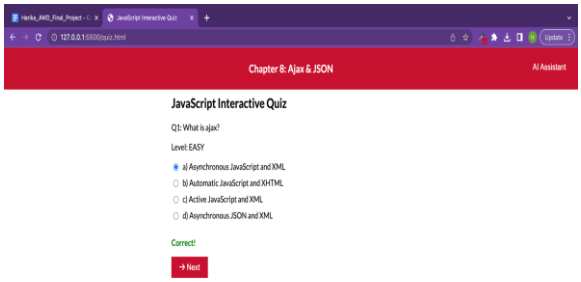

Fig 3. Taking the quiz and choosing a correct answer 1/2

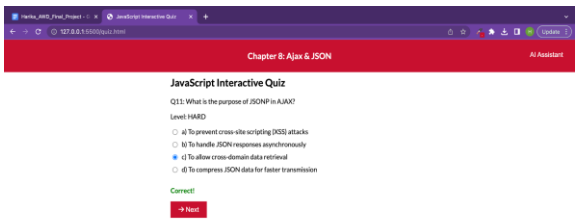

Fig 4. Taking the quiz and choosing a correct answer 2/2

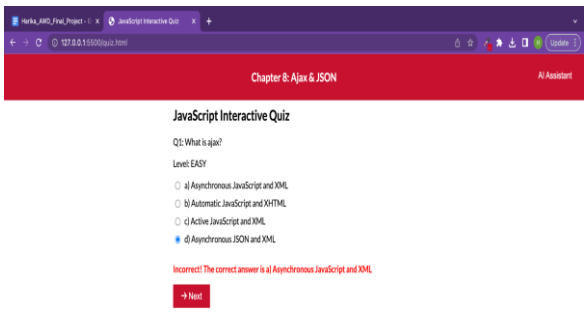

Fig 5. Taking the quiz and choosing an incorrect answer 1/2

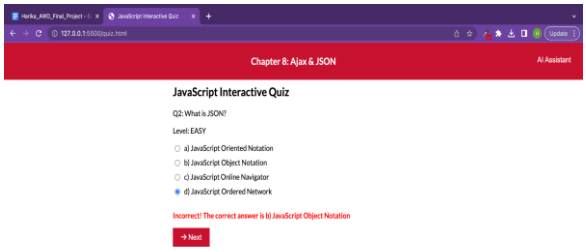

Fig 6. Taking the quiz and choosing an incorrect answer 2/2

Upon reaching the final page, the user will see the total score based on the number of correctly answered questions. A message will be displayed congratulating the user on completing the quiz and thanking them for participation [Figures 7, 8 and 9]. To retake the quiz, the user can simply click the HOME button.

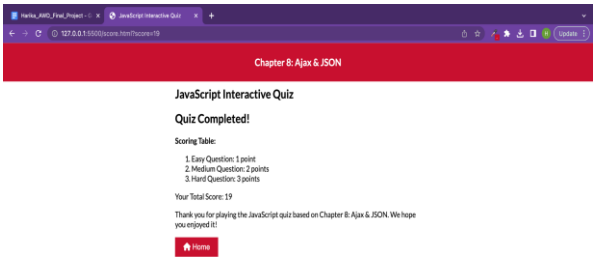

Fig 7. Completed quiz and thank you page. Example 1/3

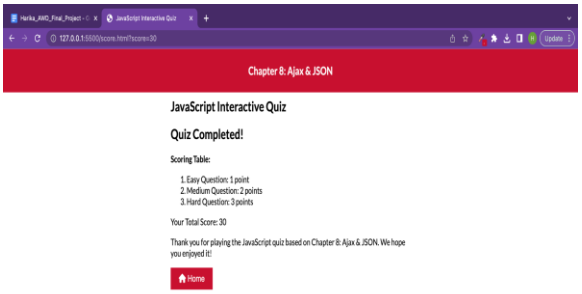

Fig 8. Completed quiz and thank you page. Example 2/3

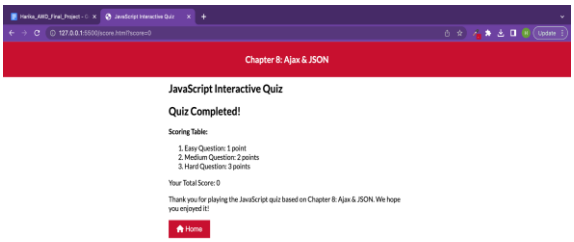

Fig 9. Completed quiz and thank you page. Example 3/3

The interactive quiz system was evaluated through a series of structured, non-user testing methods. For code quality and functional accuracy, static analysis was performed, and no critical syntax or style violations were identified. Unit tests covering core quiz logic that included answer validation, scoring, state transitions, timing functions, and JSON parsing were executed successfully. Integration tests confirmed that the system consistently retrieved and processed quiz data from the server using AJAX, properly handled valid and invalid JSON responses, and maintained stable quiz state even during simulated errors or delays. For performance, the system displayed responsive loading indicators and recovered gracefully from temporary AJAX failures by showing descriptive error messages and retry options. JSON payload sizes remained small, enabling the quiz to load efficiently even under constrained bandwidth. For accessibility, all interactive elements were keyboard-accessible, tab order was logical, and labels were correctly used. A manual keyboard-only walkthrough confirmed that all quiz functions were operable without a mouse. Input and output validation were tested. The system requires minimal text input reducing errors that may result from intentionally malformed JSON, special-character injection attempts, and simulated cross-site scripting (XSS) payloads.

3. WIX WEBSITE BUILDER DEVELOPMENT OF A WEBPAGE TO TEACH JAVA FOR LOOPS

The project aims to create a web-based application that provides users with an interactive quiz experience, enhances their understanding of Java For-loop concepts, and showcases their knowledge through a scoring system. The webpage was developed using Wix Website Builder.

3.1 Design

Some of the main usability goals that concern this project are the memorability, learnability, utility, and efficiency of the included content. The information also needs to be easy to remember, as it is a how-to site that contains instructive details, and if users are not able to retain what is learned through this site, the purpose is defeated. Assuming that the majority of the audience does not have previous exposure to programming, the material needs to be simplified and guided away from overly technical messages. When considering the user experience goals, the content provided should be able to attract and maintain the user's attention; if the learning process comes across as boring or unattractive, users will not be motivated to remain on the site and work through the information and pre/post quizzes. The overall aesthetic needs to avoid a crowded, overwhelming sensation, as the influx of information while learning programming is a notable deterrent. Feelings of encouragement and cognitive stimulation should be elicited by the site, and users should feel a sense of reward or accomplishment by participating in the quizzes and being able to see how their knowledge has improved by interacting with the blog content.

The user requirements are the factors that contribute to and allow users to complete each aspect of the site efficiently and effectively. This includes the pre- and post- quizzes, as well as the blog section that instructs and provides specific information about a Java statement/data type. A specific factor that needs to be considered is the inclusion of user-friendly phrases and explanations that do not require previous knowledge. If the terms used throughout are too technical or high level, users will not be subject to optimal user experience. The user should:

- Have the ability to work through the content in an efficient manner that is not overwhelming.
- Be able to obtain tangible results/proof that demonstrates expansion of knowledge/increased skill level.
- Be given the option to navigate freely among displayed content as needed.
- Have the opportunity to navigate additional resources if further clarification or additional information is desired.

The following user requirements for the site take into consideration the broadness of audience skill level and the necessity of user-friendly dialogue and interaction methods.

- The site needs to provide users with a comprehensive how-to section that explains the Java statement/data type clearly ("for" loops).
- The content on the site needs to be accurate and up to date, with proper syntax and modern practices.
- Novice through advanced users will gain and/or solidify knowledge from the site, and the information that may be previously known should be framed in an attractive manner that maintains attention.
- The pre- and post- quizzes will be scaled appropriately, with the latter covering more complex topics to evaluate what users absorbed from the blog content.

- The quizzes need to provide feedback to the user regarding score and explanations for incorrect answers.
- User errors should be prevented as much as possible, especially during the quizzes; if they occur, concise language instructs the user how to correct the issue.

The main tasks for the system include completion of the initial quiz, successful navigation through the site content, and a passing grade on the post quiz. In specific:

- Complete pre-test/quiz.
- Read and comprehend the blog content that instructs the user how-to use a Java statement/data type ("for" loops).
- Complete the post-test/quiz with a passing grade (>= 70%)

Nex is a use case scenario:

- The user searches for information about Java "for" loops.
- The user sees the site in the results and follows the link.
- The user arrives on the landing page and reads that there are pre- and post- quizzes surrounding the how-to blog.
- The user completes the pre-test.
- The system scores the quiz and alerts the user of their score.
- The user advances to the blog content.
- The user reads the material and views the examples.
- The system prompts the user to advance to the post-quiz.
- The user completes the quiz.
- The system grades the quiz and displays the user's score.
- The user exits the site.

3.2 Implementation

Webpage was developed with the Wix Website builder. The Web site titled "Java Coding" begins with a main landing page [Figures 10, 11 and 12] that informs the user why they should get started with learning Java and what the blog site contains. This information is above the fold, as is a general layout of how the user will proceed through the content: pre-test, main module, then exit quiz. The pre-test is located on the landing page so the user can get started as soon as possible [Figures 13, 14, 15 and 16]. OpinionStage's free resources were used to embed quizzes into the site, both the pre- and post- versions. Each question is automatically graded as the user progresses through them, and a final score is displayed after the final question is submitted. Each quiz is 10 questions, and the pre-test features true/false and multiple-choice options. The exit quiz possesses programming questions in addition to multiple-choice and true/false.

The main how-to section [Figures 17, 18, 19, and 20] of the site first introduces the idea of for loops and gives the user basic background information on their application. Then, the general form of a loop is provided and labeled accordingly. Each component is explained in further detail under their own subheading, and brief examples are shown of what the part would appear as in a working loop. Further down the page, full loop examples are given, and individual components are underlined and elaborated to explain to the user a real application of a loop and why certain parts are written in specific ways. Variable scope is also covered in these first few examples. After the user is made aware of how for loops are constructed, the concept of nested loops is explored. Examples are provided and explanation of what happens while the loop is being iterated are displayed next to the code. Finally, there are additional examples of for-loops that combine and/or pick from the covered information and demonstrate loops that solidify what the user learned from the how-to content. The user can now click on Exit quiz (post-test) and take it [Figures 21, 22, 23, 24 and 25]

There is a brief "Thank you" message included towards the bottom of the Exit Quiz page [Figures 26 and 27. It also features links to external resources that provide more information and examples of Java for loops. Some have more advanced examples so the user can see how they can expand their knowledge base and continue reading about loops.

Fig. 10. Home page 1/3

Fig.11. Home page 2/3

Fig. 12. Home page 3/3

Fig. 13. Pre-test start page

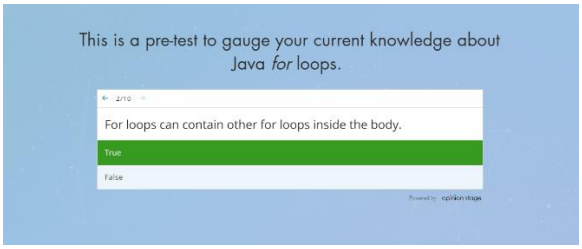

Fig. 14 selecting correct answer in pre-test

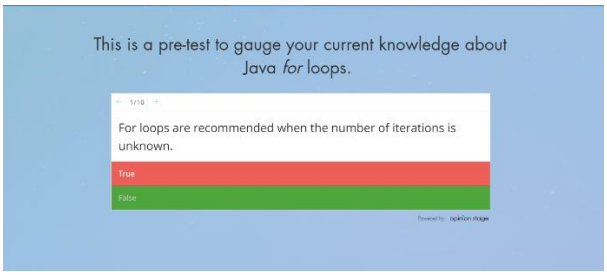

Fig. 15 selecting incorrect answer in pre-test

Fig. 16 displaying result of pre-test

Fig. 17 How-To Section, samples of lesson teaching for loops 1/3

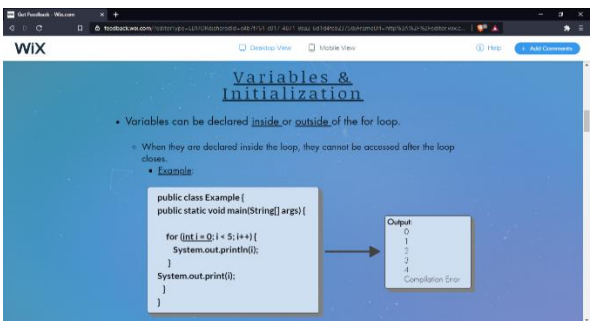

Fig. 18 How-To Section, samples of lesson teaching for loops 2/3

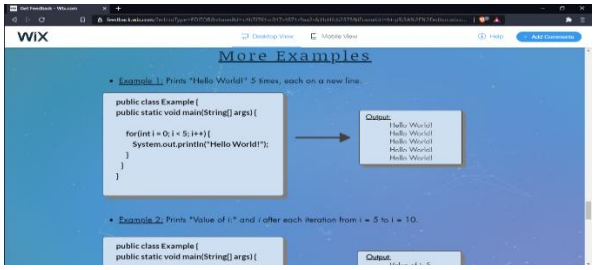

Fig. 19 How-To Section, samples of lesson teaching for loops 3/3

Fig. 20 End of the How-To Section

Fig. 21 post-test/ Exit Quiz Start Page 1/2

Fig. 22 post-test/ Exit Quiz Start Page 2/2

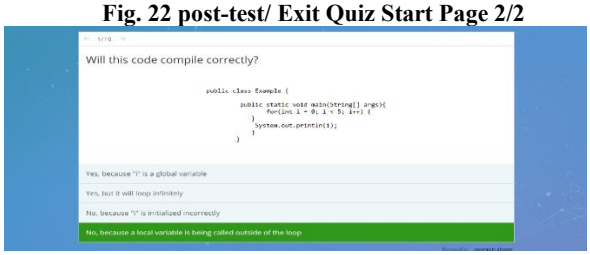

Fig. 23 choosing the correct answer in the post-test /Exit Quiz

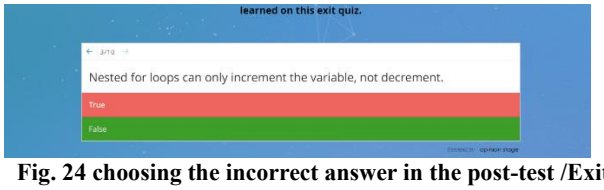

Fig. 24 choosing the incorrect answer in the post-test /Exit Quiz

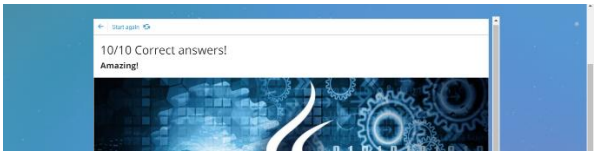

Fig. 25 Post-test /exit quiz page displaying final score

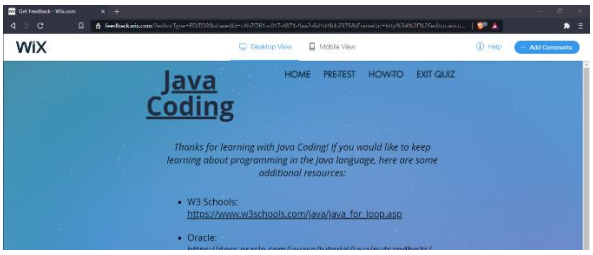

Fig. 26 Thank You Page 1/2

Fig. 27 Thank You Page 2/2

3.3 Testing and Evaluation

The main goal of this evaluation is to determine the usability and efficacy of the Java how-to blog that instructs users on the basics of creating programs using *for* loops. The site was constructed with the Wix Web site builder, and the evaluation seeks to analyze how well the final product allows users to expand their knowledge from the time of taking the pre-test to that of the exit quiz. Additionally, the layout of the site is going to be explored to determine if it progresses in an appropriate manner that encourages learnability and good utility. Buttons, hyperlinks, and other multimedia elements that are integrated throughout the interaction period will be evaluated as well, especially on factors such as placement, size, attractiveness, and other factors that impact the user experience.

After reading the how-to blog and completing both quizzes, each participant was asked to complete a questionnaire. They responded to a set of questions to rate various aspects of the site. Each statement will be based on a scale of 1 to 5, where 1 is "Strongly Disagree" and 5 is "Strongly Agree". Options 2 through 5 correspond to "Disagree", "Neutral", and "Agree", respectively. The questions are as follows:

1. I learned something from this site.
2. This site was confusing and hard to read.
3. The quizzes were too long.
4. The pre-test was too hard.
5. The site is attractive.
6. I want to learn more after completing this site.
7. I understand each part of a *for* loop.

There will also be a brief interview to provide open-ended questions and hear more feedback from the participants in an unstructured manner. The questions are as follows:

- Did you like the layout of the site?
- Do you think the explanations for different terms were explicit enough?
- Was the exit quiz reflective of the content in the how-to section?
- What needs to be improved about the site, if anything?

The main limitation of this project is that users may have varying levels of prior programming knowledge. Experienced users will find Java for-loops easier to understand and may score higher on the pre-test, while beginners may struggle with unfamiliar terms. Also, the site uses simple, foundational explanations of loop components and their functions. However, learning preferences differ, so the site's format may not suit all users or support every learning style equally.

All users showed score improvement after using the site [Figure 28]. The average completion time was 13.64 minutes (max 15:25, min 11:22). Questionnaire responses (1–5 scale) indicated that most users did not find the site confusing, hard, or too long, and that it was attractive, informative, and encouraged further learning [Figure 29]. Thematic analysis of open-ended responses showed overall positive perceptions with two main areas for improvement.

With regard to usability: Students consistently noted that the site was easy to navigate, visually clear, and simple to follow. With regard to content clarity: Most found the initialization and condition sections easy to understand. With regard to key difficulty and in specific Nested loops: Several students reported confusion with nested loops, suggesting a need for clearer explanations, examples, or added scaffolding. With regard to breadth of skills: some users felt the site covered a good range of skills. While some felt no improvements were needed, others requested more examples or an advanced option for additional challenge. A few responses were neutral or vague. In summary, students found the site easy to use and the core content accessible. The main improvement needed is clearer instruction on nested loops, along with optional advanced materials for learners seeking more depth.

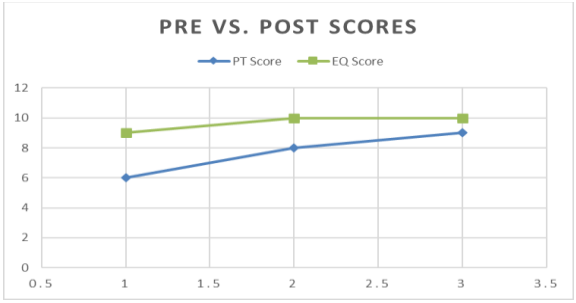

Fig. 28 pre and post scores comparison

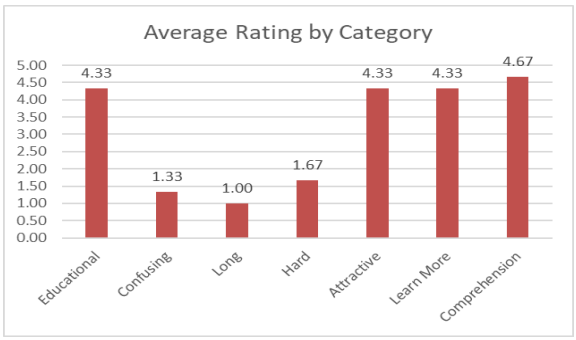

Fig. 29 average rating by category

4. CONCLUSION

The development of an interactive website to teach AJAX, JSON, and for-loops in Java demonstrates the value of combining web technologies with programming education. Through the use of HTML, CSS, and JavaScript, students gain practical experience in creating dynamic and responsive web applications that reinforce theoretical concepts. The comparison with Wix highlights that while Wix simplifies design and layout, it lacks the flexibility and coding depth offered by traditional development tools. Therefore, using HTML, CSS, and JavaScript provides a more effective environment for learning programming logic and web interaction. Overall, the project shows that interactive website development not only improves technical skills but also enhances engagement, creativity, and problem-solving abilities among students. Future work may expand the platform to cover additional Java topics and include assessment tools to track learner progress. Enhancing accessibility, mobile support, and adaptive learning features could further improve user experience. Future research could also evaluate learning outcomes with larger user groups and explore hybrid development approaches that combine ease of design with greater customization.

5. REFERENCES

[1] Staff Writer, "The importance of codes and coding in today's digital world," *Reference.com*, May 19, 2025. [Online]. Available: https://www.reference.com/science-technology/importance-codes-coding-today-s-digital-world

[2] Coelho, R. C., Marques, M. F. P., & de Oliveira, T. (2023). *Mobile learning tools to support in teaching programming logic and design: A systematic literature review*. Informatics in Education, 22(4), 589–612. https://eric.ed.gov/?id=EJ1410469

[3] Wang, X. (2006). *A practical way to teach web programming in computer science*. *Journal of Computing Sciences in Colleges*, 22(1), 211–220. https://dl.acm.org/doi/abs/10.5555/1181811.1181841

[4] Elgamal, A. F., Abas, H. A., & Baladoh, E.-S. M. (2013). *An interactive e-learning system for improving web programming skills*. Education and Information Technologies, 18(1), 29–46. https://doi.org/10.1007/s10639-011-9175-7

[5] Codeable. (n.d.). *Hire the best freelance WordPress developers*. https://www.codeable.io/

[6] Bhatti, S., Dewani, A., Maqbool, S., & Memon, M. A. (2019). *A web-based approach for teaching and learning programming concepts at middle school level*. International Journal of Modern Education and Computer Science, 11(4), 46–53. https://doi.org/10.5815/ijmecs.2019.04.06

[7] Coursera. (n.d.). *Coursera: Online courses & credentials from top educators*. https://www.coursera.org

[8] Codecademy. (n.d.). *Learn to code - for free*. https://www.codecademy.com

[9] Brown, Michael, Samruddhi. "Coursera vs Codecademy (2025): A Detailed Comparison." *Open2Study*, 20 Sept. 2025, https://www.open2study.com/comparison/coursera-vs-codecademy/.

[10] *BitDegree*. "Coursera vs Codecademy: Which One Is Better?" https://www.bitdegree.org/online-learning-platforms/comparison/coursera-vs-codecademy/.

[11] I S Zinovieva *et al* 2021 *J. Phys.: Conf. Ser.* **1840** 012029**DOI** 10.1088/1742-6596/1840/1/012029

[12] Okoh, F. O., & Grace, I. (2022). *Evaluating the impact of online coding platforms on programming skill acquisition in secondary and tertiary education*. Acta Electronica Malaysia, 6(1), 16–23. https://doi.org/10.26480/aem.01.2022.16.23

[13] Jung, I., Choi, J., Kim, I., & Choi, C. (2016). *Interactive learning environment for practical programming language based on web service*. In 2016 15th International Conference on Control, Automation and Systems (ICCAS) (pp. 1215–1220). IEEE. https://ieeexplore.ieee.org/document/7760705

[14] Eng Lieh Ouh, Benjamin Kok Siew Gan, and David Lo. 2022. ITSS: Interactive Web-Based Authoring and Playback Integrated Environment for Programming Tutorials. In 44nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '22), May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3510456.3514142

[15] Alvarado, J., & Rodríguez, M. (2025). *Impact of an Interactive Learning Platform on Programming Student Performance*. IEEE Xplore. https://ieeexplore.ieee.org/document/10767622.

[16] Alghamdi, M. (2025). *Improving Programming Skills: The Impact of Interactive E-Learning Tools on Student Success*. CLEI Electronic Journal, 28(1). https://doi.org/10.19153/cleiej.28.1.9