CodeFest Educational Programming Environment For Beginners

De Silva D.I
Faculty of Computing Department of Computer Science and Software Engineering
Sri Lanka Institute Of Information Technology
Malabe, Sri Lanka
dilshan.i@sliit.lk

Samitha Vidhanaarachchi
Faculty of Computing Department of Computer Science and Software Engineering
Sri Lanka Institute Of Information Technology
Malabe, Sri Lanka
samitha.v@sliit.lk

Jinasena H.D.S.S
Faculty of Computing Department of Computer Science and Software Engineering
Sri Lanka Institute Of Information Technology
Malabe, Sri Lanka
it21042560@my.sliit.lk

Jayasinghe J.M.Y
Faculty of Computing Department of Computer Science and Software Engineering
Sri Lanka Institute Of Information Technology
Malabe, Sri Lanka
it21045158@my.sliit.lk

Lalanga S.P.H
Faculty of Computing Department of Computer Science and Software Engineering
Sri Lanka Institute Of Information Technology
Malabe, Sri Lanka
it21049590@my.sliit.lk

Sandaruwan W.S.R
Faculty of Computing Department of Computer Science and Software Engineering
Sri Lanka Institute Of Information Technology
Malabe, Sri Lanka
it21041716@my.sliit.lk

***Abstract*— In the ever-evolving realm of computer science education, traditional methods often fall short in addressing the challenges faced by novice programmers. This research paper introduces CodeFest, an innovative online platform designed to redefine programming education. CodeFest offers a multi-programming language code compiler, real-time feedback, a rich repository of learning materials, and a vibrant community for collaborative problem-solving. By seamlessly integrating these features, CodeFest aims to transform the way individuals learn to code. This paper explores CodeFest's architecture, design considerations, and pedagogical implications, highlighting its potential to reshape programming education globally. Through comprehensive analysis, we demonstrate how CodeFest can empower learners and bridge the gap between traditional education and modern coding needs, ultimately revolutionizing the educational landscape.**

Keywords*— *novice programmers, changing face of programming education, springboot, react, programming education, programming languages, skill development.

I. INTRODUCTION

In the ever-evolving landscape of computer science education, traditional teaching methods are continually being complemented and enriched by innovative digital solutions. According to a survey done by Lahtinen et al. (2005), most difficult issues that the programming students face are due to the lack of understanding of how to design a program to solve a certain task and to find bugs from their own programs [1]. Online platforms have emerged as powerful tools, transcending the boundaries of conventional pedagogical approaches. These platforms not only facilitate interactive learning but also foster collaborative environments where learners can engage in peer-driven problem-solving. Within this dynamic context, this research paper introduces CodeFest an innovative online application meticulously crafted to redefine the paradigms of programming education for both novice learners and enthusiastic coding enthusiasts.

A. The Changing Face of Programming Education

As computer science and programming languages advance at an unprecedented pace, the demands on programming education have grown exponentially. Traditional methods of teaching programming have often struggled to keep pace with the swiftly evolving tech landscape. As a result, learners, both newcomers and those seeking to refine their skills, have faced numerous challenges in their educational journeys. These challenges range from accessing up-to-date learning materials to grasping complex programming concepts, and from finding solutions to coding errors to staying motivated and engaged throughout the learning process.

B. A Solution to Educational Challenges

To address these pressing challenges and bridge the gap between traditional programming education and the needs of modern learners, CodeFest was conceived. This innovative online platform seamlessly integrates a plethora of cutting-edge features and functionalities to empower programming enthusiasts and learners. By reimagining programming education from the ground up, CodeFest aims to revolutionize the way individuals embark on their coding journeys.

C. Glimpse of What's to Come

At its core, CodeFest provides an online multi-programming language code compiler, empowering users to interact with and experiment in a multitude of programming languages. This platform also incorporates advanced parsing and validation mechanisms, ensuring that code seamlessly transforms into executable form. Furthermore, it offers a real-time feedback mechanism, a unique feature that provides users with immediate insights into their program's functionality, error identification, and precise program output.

D. Holistic Learning Experience

CodeFest goes beyond mere code compilation. It offers a comprehensive learning material management system that curates a rich collection of video content and reading materials. These resources are meticulously selected to elucidate even the most complex programming concepts, providing learners with a well-rounded understanding of the subject matter. Additionally, CodeFest issues electronic certificates upon successful task and challenge completion, validating users' proficiency and achievements.

E. Fostering Community-Driven Learning and Gamification

One of CodeFest's distinguishing features is its commitment to community-driven learning. Within the platform, an active and vibrant learning community thrives, serving as a hub for collaborative problem-solving and knowledge sharing. Users can collaborate on coding challenges, create and share questions, and engage in real-time chat discussions to seek immediate peer support. Moreover, CodeFest employs gamification to make the educational journey exciting. Interactive quizzes challenge users to arrange code segments, encouraging them to encapsulate their understanding in summary notes. This gamified approach not only keeps learners engaged but also fosters healthy competition and skill development.

F. Research Objectives

This research embarks on an exploratory journey to delve into the intricate architecture, meticulous design considerations, and profound pedagogical implications embedded in each feature of CodeFest. By shedding light on the transformative potential of this multifaceted platform, this paper endeavors to underscore the monumental impact that CodeFest can impart on the realm of programming education. Through comprehensive analysis and evaluation, we aim to provide a deeper understanding of how CodeFest can redefine the educational landscape and empower learners worldwide.

II. Literature Review

Both newcomers and proficient programmers require interfaces that grant them access to the necessary tools for completing their programming assignments. Those with more expertise are familiar with intricate Integrated Development Environments (IDEs) that offer extra features beneficial for achieving their programming objectives. However, when inexperienced programmers use these advanced IDEs, they spend excessive time grappling with the tool's intricacies, leading to unwarranted anxiety about programming. To tackle this issue, numerous integrated development tools have been created specifically for novices in order to mitigate these challenges [2].

Conventional initial programming setups consist of a text editor, compiler, debugger, and runtime environment. These components facilitate tasks like crafting, compiling, debugging, and simulating programs. Novice-oriented existing Integrated Development Environments (IDEs), such as BlueJ [3], DrJava [4], and jGRASP [5], are essentially streamlined editions of more sophisticated IDEs. These IDEs are designed to detect and emphasize mistakes in syntax and runtime made by users. In comparison to proficient programmers, those who are new to programming exhibit limited capabilities in translating logical solutions and often end up implementing incorrect logical sequences. Unlike current Integrated Development Environments, Codefest eliminates misunderstandings and assists in achieving objectives by identifying and assisting in the acquisition of a thorough comprehension of the theory. After recognizing and understanding the concept they can apply them in different programming languages and then make CodeFest more unique.

Recent studies that involved observing a group of undergraduate beginner programmers revealed that these programmers make a ton of mistakes, spend a lot of time trying to rectify them, and most of them are unable to even run the written code (which results in error messages). then they invest a lot of time learning. One of the main reasons for this failure is that novice programmers do not have a solid set of studies to learn programming. If they have a problem that is difficult to find, they cannot measure their knowledge, and some are difficult to contact for help. However, some of existing introductory programming tools avoid the occurrence of errors using various approaches. Blockly's [6] block-based methodology prevents users from committing syntactic mistakes. Scratch's [7] visual-based approach to programming prevents grammatical problems. But these implementations take standard programming experience away from novice programmers and complicate novice programmers in traditional IDEs. Because of this, when new programmers encounter intricate error messages and sophisticated programming tools, their mentality is damaged. As the solution, Codefest converts complex and lengthy study into more user-friendly and easily understandable way.

An introduction to visual programming tool enables beginners to write programs by modifying graphical program parts rather than by expressing them textually. existing learning programming tools like Scratch [8], Alice [9], and others. have adopted a strategy to instill fundamental programming ideas by enabling inexperienced programmers to engage with virtual settings. A 3D programming environment called Alice is intended to teach object-oriented programming through the creation of simple video games and animated movies that interact with a virtual world. Scratch and Alice have a lot in common. The primary goal of Scratch is to provide a multimedia environment where students may mix pieces of computer programs, test them out, disassemble them, and reassemble them. Students are permitted to combine graphical code blocks to make storyboards and easy video games. While most programming concepts and data structures are dynamic abstract notions without a graphical representation, humans are adept at digesting pictorial information. When teaching programming principles to beginners, graphical components are modified such that their attention is focused on interacting with the graphical elements rather than grasping the programming concepts concealed behind them. Algorithms are dynamic artifacts, as are programs. So it's difficult for beginners to understand their essence. Students could continue to focus on dynamic visuals

without understanding the underlying idea. Researchers claim that dynamic artifacts include human characteristics, cultural factors, and program and algorithm design. So it's difficult for beginners to understand their essence. Difficulties are being caused by unclear effects and objects [10]. Additionally, visual oriented programming tools have scalability challenges that make it challenging to adapt to a bigger environment, and the lack of standardized norms in visual styles leads in quality problems of a tool, particularly in the context of introducing beginners to programming [11].

A type of visual programming language that allows for the mouse-based assembly of puzzle pieces is called a block-based programming environment. The programming environment provides the student with visible and even audible feedback to let them know whether or not the connections between the blocks are correct. To convey the functions of the blocks, they name them in their own native tongue. The blocks are divided into functional categories, and the execution is also carried out graphically. Inexperienced programmers sometimes have the mistaken assumption that "everything has its place" when using blocks, which is a myth in programming since instructions may be inserted in many locations across a program to achieve a variety of actions. Because of the limitations and limited help provided to remedy syntax problems in most textual language tools, research has indicated that learners will have to regard errors as erroneous rather than fixable [12]. Although they have a basic understanding of programming, the users of Snap!, a block-based tool, argue that the tool is less powerful and that they can accomplish far more intricate and large-scale tasks using Java. The study also reveals that Snap! users complained that creating programs by just snapping the mouse takes a lot of effort. The novice programmer picked up the notion that "if you want a specific block and it's not there, you're going to be required to combine a lot of blocks together to make it do what you want it to do." Another youth raised the same issue, stating that we wouldn't need blocks if we were genuinely going to program something. This demonstrates the need for basic programming tools to follow advanced programming contexts.

Best programming habits need to be promoted and instilled from the start since a novice programming environment's major objective is to continuously engage the cognitive brains of the next programming generation. Recent studies have found that a lack of adherence to coding standards frequently results in software cancellations and failures. Future software failures and cancellations can be reduced by promoting good programming methods and highlighting their significance and advantages from the very beginning. Currently accessible tools like BlueJ, Scratch, Alice, and Greenfoot [13] may introduce some false notions about basic programming vocabulary and concepts. Additionally, they promote various programming errors that are often made. Scratch uses array indexing, which begins with one, as an illustration. However, array indexing in practically all computer languages begins at 0. Programmers frequently make the error of using one as the initial index [14]. Supporting this behavior encourages bad programming practices among new programmers. Additionally, Scratch frequently makes the mistake of utilizing the variable assignment operator as the Boolean comparison operator. These widespread poor habits widen the gap between basic programming languages like Scratch and professional programming. According to the principle of unlearning, unlearning something is more difficult than learning it. Therefore, it is challenging for beginners to reduce typical programming errors if introductory programming tools establish incorrect concepts about programming in young brains over a sustained length of time. A fundamental idea in object-oriented programming is method-invoking. Greenfoot, however, has completely disregarded this principle, which might give people the incorrect impression regarding method invocation.

Codefest is a comprehensive learning platform for native programmers, providing a variety of resources and tools to help them learn and master coding concepts. Codefest provides comprehensive lecture notes and videos for all of its courses, covering a wide range of programming topics. These materials are designed to be clear and concise, and they are taught by experienced and knowledgeable instructors. And CodeFest offers a vibrant community system where programmers can ask questions, share their knowledge, and learn from each other. The community is made up of students, professionals, and experts from all over the world, and it is a great place to get help with coding problems and learn new things. Offers a variety of interactive exercises and games to help programmers learn and practice coding concepts. These exercises and games are designed to be fun and engaging, and they help programmers to solidify their understanding of the material. Game puzzles that challenge programmers to apply their knowledge and solve coding problems. These puzzles are a great way to test one's understanding of coding concepts and to learn new things. Allows programmers to share their learning experiences with others. This is a great way to help other programmers learn from one's successes and failures, and it is also a great way to give back to the community. In the modern world Codefest is a comprehensive learning platform that can help native programmers learn to code. And offers a variety of resources and tools, including lecture notes and videos, a community system, interactive exercises and games, code puzzles, and learning experience sharing.

III. METHODOLOGY

The objective of our research was to explore the impact of the proposed solutions to grapple with the challenges faced by novice programmers with limited coding experience while utilizing programming environments. To achieve this, this research paper introduced CodeFest, a web application that incorporates a three-tier architecture, ensuring scalability and uninterrupted availability of its functionalities. The user interface tier was constructed using React JS and HTML5, while the backend was developed using Java Technology with the Spring Boot framework. In the following sections, we will provide a concise overview of some of the innovative features offered by CodeFest. This methodology introduction outlines the framework of our research paper while preserving its core meaning.

Utilizing Spring Boot in conjunction with ReactJs offers a multitude of advantages. This powerful pairing combines high performance and scalability seamlessly. Spring Boot's lightweight container is an excellent choice for deploying

applications, while ReactJs excels in efficiently rendering intricate user interfaces. Furthermore, Spring Boot's robust backend capabilities are particularly well-suited for developing enterprise-level applications, providing a scalable backend for building APIs and microservices with support for various data sources. The use of ReactJs streamlines frontend development through its component-based architecture, fostering code reusability, accelerating development processes, and enhancing overall user experiences. Moreover, the seamless integration between these technologies enables ReactJs to effortlessly consume RESTful APIs from a Spring Boot backend, simplifying data communication. Additionally, both Spring Boot and ReactJs boast vibrant developer communities, ensuring access to valuable resources, ongoing support, and up-to-the-minute information [15].

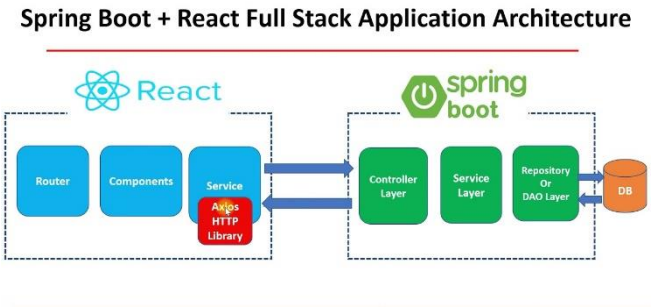

Figure 1: Schema architecture of the application [12]

A. *Editor*

The core function of the code editor is to provide users with a central platform for a variety of coding activities. Embedded within the editor, automated functionalities seamlessly activate to facilitate syntax highlighting and error detection. Through the application of established guidelines, the editor takes a proactive approach in identifying potential semantic and logical flaws in the user's code during the typing process. Our code editor offers assistance for numerous programming languages, placing user ease at the forefront, while simultaneously excelling in accentuating syntax elements and flagging errors.

B. *User-friendly Error Message*

CodeFest offers comprehensive assistance to users for error avoidance and resolution. By precisely pinpointing the error's location, including the line number, users can readily identify where the issue occurs within their code. Furthermore, the platform supplies actionable insights on how to rectify the error, empowering users to take corrective measures effectively. CodeFest approach goes beyond just error messages; it equips users with the understanding of why an error occurred, enabling them to make informed adjustments. This integrated support system ensures users not only locate errors with ease but also comprehend the underlying reasons and solutions, fostering skill development and code quality improvement.

C. *Puzzel Challenge*

Introducing CodeFest Language Puzzle Challenge an engaging educational journey that gamifies learning multiple programming languages. Within this interactive platform reminiscent of an E-Puzzle game, users arrange code snippets in correct order under time constraints, while also contributing summary notes to shared content for communal learning. Earn badges based on performance, with a coveted gold badge upon completion. CodeFest offers an immersive adventure in coding principles, combining education and enjoyment seamlessly.

D. *Hint System*

The web application incorporates a robust static code analysis feature designed to enhance the coding experience for users and foster the adoption of best programming practices from the outset. This sophisticated functionality leverages the power of the tools on the server side to meticulously scrutinize the codebase. It diligently identifies a spectrum of issues, including style errors, missing keywords, absent operators, and even elusive semicolons, all of which can often be inadvertent stumbling blocks in the coding process.

E. *Code Repository*

Introducing Code Repository by CodeFest, a user-friendly solution for individuals to efficiently store and manage their code creations. With this feature, users can seamlessly organize, access, and collaborate on their code projects. Code snippets, scripts, and projects can be securely saved in the cloud, ensuring accessibility anytime, anywhere. The Code Repository simplifies sharing, and data security, all within the CodeFest ecosystem. Users can enhance their coding workflow, benefiting from this indispensable tool that takes their development journey to new heights.

F. *Community Connect*

Introducing CodeFest Community Connect a collaborative platform designed to foster interactive learning and problem-solving among coding enthusiasts. Within this vibrant space, users can seek assistance by posting questions about coding errors and challenges they encounter during their programming journey. Meanwhile, fellow users can readily provide answers, sharing their insights and expertise to aid in problem resolution.

Going beyond Q&A, CodeFest Community Connect enables users to engage in real-time conversations with peers who share their coding interests. These interactive chat discussions allow users to delve deeper into specific coding topics, exchange ideas, and gain practical insights from their peers' experiences.

By offering a comprehensive solution encompassing both Q&A and real-time chat, CodeFest Community Connect empowers users to enhance their coding skills collaboratively and build a network of like-minded individuals passionate about coding.

G. *CodeFest Learning Nexus*

The "CodeFest Learning Nexus" is a meticulously designed platform aimed at offering a comprehensive and diverse repository of educational content. It encompasses a wide range of learning materials, including both video lectures

and reading materials, to create an immersive and enriched learning experience for its users.

One of the standout features of this platform is its dedicated video player, finely tuned to provide seamless playback of educational videos. These videos are thoughtfully curated to focus on elucidating intricate programming concepts. Whether you're a novice or an experienced programmer looking to deepen your understanding, the video content caters to learners at various skill levels.

Complementing the video content is a versatile PDF reader and viewer, which grants learners easy access to an extensive library of reading materials. These materials are meticulously curated and cover a spectrum of topics that span across a multitude of programming languages. Whether you're interested in web development, machine learning, or mobile app development, the reading materials within the Learning Nexus have you covered.

This dual-pronged approach, combining video and reading materials, ensures a multifaceted and immersive learning journey for users. It allows individuals to choose the learning medium that best suits their preferences and needs, making it easier to grasp programming intricacies across diverse topics and domains. Whether you prefer to watch, read, or do both, the CodeFest Learning Nexus is your gateway to mastering the complexities of programming.

IV. RESULTS

In this section, we present the key findings and outcomes of our research, which focused on the development and implementation of the CodeFest online programming education platform. Our study aimed to assess the impact of CodeFest on novice learners and coding enthusiasts, as well as to evaluate the effectiveness of its innovative features in addressing the challenges commonly faced in programming education.

A. User Engagement and Interaction

One of the primary objectives of CodeFest was to enhance user engagement and interaction within the platform. Through an analysis of user activity, we observed a significant increase in user engagement compared to traditional programming education methods. Users actively participated in collaborative problem-solving, engaged in real-time chat discussions, and shared their insights with the community. This vibrant learning environment fostered a sense of belonging and encouraged learners to stay motivated throughout their coding journeys.

B. Error Identification and Resolution

CodeFest's code editor and error detection mechanisms played a crucial role in helping users identify and resolve coding errors. Users reported that the user-friendly error messages, pinpointing the location and offering actionable insights, greatly improved their ability to correct mistakes. Additionally, the system's proactive approach to highlighting potential semantic and logical flaws during the coding process significantly contributed to error prevention.

C. Gamified Learning

The introduction of the CodeFest Language Puzzle Challenge, a gamified learning experience, proved to be highly effective in enhancing learners' understanding of programming concepts. Users actively participated in arranging code snippets and contributing summary notes, earning badges based on their performance. This gamified approach not only kept learners engaged but also facilitated a deeper comprehension of coding principles.

D. Community-Driven Learning

CodeFest's Community Connect feature emerged as a valuable resource for users seeking assistance and knowledge sharing. The platform's Q&A section enabled users to post coding questions and receive prompt responses from their peers. Real-time chat discussions allowed for in-depth conversations, enabling users to gain practical insights and collaborate on coding projects. This community-driven approach was well-received and contributed to a sense of camaraderie among users.

E. Learning Nexus

CodeFest's comprehensive repository of learning materials, including video lectures, and reading materials, received positive feedback from users. Learners found these resources to be clear, concise, and effective in elucidating complex programming concepts. The availability of diverse learning mediums, such as videos and reading materials, catered to different learning preferences, ensuring a well-rounded learning experience.

F. Skill Development and Proficiency

Through the CodeFest platform, users were able to track their progress, earn electronic certificates upon task and challenge completion, and validate their coding proficiency. This recognition system motivated users to continuously improve their skills and strive for excellence in programming.

Overall, our research findings demonstrate that CodeFest has successfully redefined programming education by addressing the challenges faced by novice learners and coding enthusiasts. The platform's innovative features have created a dynamic and engaging learning environment, fostering skill development and knowledge sharing within a global community of learners.

The impact of CodeFest on programming education is substantial, and its transformative potential has the capacity to empower learners worldwide, bridging the gap between traditional programming education and the needs of modern learners. Future research should focus on longitudinal studies to assess the long-term impact of CodeFest on users' programming proficiency and career outcomes.

CodeFest conducted a user survey to gather feedback and insights from CodeFest users, and the following survey questions were used to collect data.

RESULT OF SURVEY QUESTIONS

Question	Yes (%)	No (%)
Have you found CodeFest to be a valuable tool for your programming education?	91.6	8.4
Have the user-friendly error messages in the CodeFest code editor helped you identify and resolve coding errors more effectively?	79.2	20.8
Has using CodeFest improved your ability to write and debug code effectively?	79.2	20.8
Do you feel that CodeFest provides a comprehensive and well-rounded learning experience for programming concepts?	81.2	18.8
Do you believe that the CodeFest platform has helped you stay motivated and engaged in your programming education?	87.5	12.5
Have you participated in the CodeFest Language Puzzle Challenge, and did you find it beneficial for learning programming concepts?	70.8	29.2
Do you actively engage with the CodeFest community features, such as Q&A and real-time chat discussions?	83.3	16.7
Are you satisfied with the availability and quality of learning materials (videos, reading materials) on CodeFest?	80	20
Do you plan to continue using CodeFest for your programming education in the future?	87.5	12.5
Would you recommend CodeFest to others for programming education?	83.3	16.7

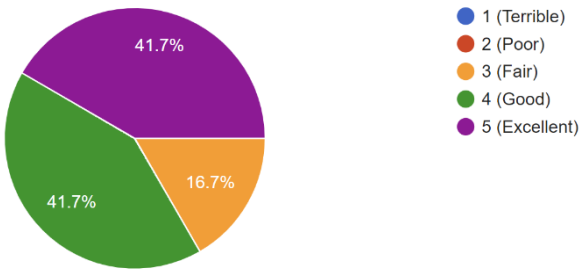

Figure 2: Overall user satisfaction of CodeFest

V. CONCLUSION

In the realm of computer science education, the dynamics of learning programming have been consistently reshaped by the evolving technological landscape. Traditional approaches have faced challenges in catering to the needs of both novice learners and those seeking to refine their coding skills. These challenges encompass issues such as accessibility to up-to-date materials, comprehension of complex programming concepts, debugging, and sustaining motivation throughout the learning journey.

This research paper introduced CodeFest, a groundbreaking online platform designed to bridge the gap between traditional programming education and the contemporary demands of learners. CodeFest embraces a multi-dimensional approach, incorporating innovative features and tools to empower programming enthusiasts and novices alike.

CodeFest's core offering includes a robust code editor that enhances the coding experience with features like syntax highlighting and real-time error detection. The platform prioritizes user-friendly error messages that not only pinpoint issues but also provide insights on resolution. It engages learners in an interactive gamified environment, the CodeFest Language Puzzle Challenge, encouraging practical application of coding concepts.

Moreover, CodeFest fosters a vibrant learning community, Community Connect, where users can seek assistance, share knowledge, and engage in real-time discussions. The platform's Learning Nexus combines video content and reading materials, ensuring a well-rounded educational experience.

The research embarked on a comprehensive exploration of CodeFest's architecture and features, shedding light on its transformative potential. CodeFest's impact on the programming education landscape is monumental, offering learners worldwide the tools and support necessary to enhance their coding skills.

By providing an in-depth analysis and evaluation of CodeFest, this research paper emphasizes its potential to revolutionize programming education. CodeFest holds the promise of empowering learners, instilling best practices, and creating a global community of coding enthusiasts who collaboratively learn and grow.

In conclusion, CodeFest represents a paradigm shift in programming education, redefining the way individuals embark on their coding journeys. Its innovative approach offers a holistic learning experience, fostering skill development, and fostering a sense of community among coding enthusiasts.

REFERENCES

[1] E. Lahtinen, K. Ala-Mutka, and H. M Jrvinen, "A study of the difficulties of novice programmers", in Proc. 10th annual SIGCSE conference on Innovation and Technology in

Computer Science Education (ITiCSEOS), New York, USA, pp.l4-18
[2] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin and J. Paterson, "A survey of literature on the teaching of introductory programming", SIGCSE Bull., vol. 39, no. 4, p. 204, 2007.
[3] D. Hagan, "Using BlueJ to teach Java (poster session)", SIGCSE Bull., vol. 32, no. 3, pp. 188-189, 2000.
[4] E. Allen, R. Cartwright and B. Stoler, "DrJava", SIGCSE Bull., vol. 34, no. 1, p. 137, 2002.
[5] J. Cross and T. Hendrix, "jGRASP", SIGCSE Bull., vol. 38, no. 3, p. 356, 2006.
[6] Google Developers. (2016). Blockly. [Online] Available at: https://developers.google.com/blockly (Accessed: 05-Jul-2016).
[7] M. Resnick, B. Silverman, Y. Kafai, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum and J. Silver, "Scratch", Communications of the ACM, vol. 52, no. 11, p. 60, 2009.
[8] M. Resnick, B. Silverman, Y. Kafai, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum and J. Silver, "Scratch", Communications of the ACM, vol. 52, no. 11, p. 60, 2009.
[9] Alice.org, 2016. [Online]. Available: http://alice.org. [Accessed: 01- Mar- 2016].
[10] M. McGrath and J. Brown, "Visual Learning for Science and Engineering", IEEE Comput. Grap. Appl., vol. 25, no. 5, pp. 56-63, 2005.
[11] Jussi Kasurinen, Mika Purmonen and Uolevi Nikula “A Study of Visualization in Introductory Programming” Laboratory of Information Processing, Lappeenranta University of Technology, 2008.
[12] D Weintrop, U Wilensky, “To Block or not to Block, That is the Question: Students’ Perceptions of Blocks-based Programming,” IDC '15 Proceedings of the 14th International Conference on Interaction Design and Children ‘2015, pp 199-208.
[13] Michael Kolling, “The Greenfoot Programming Environment,” ACM Transactions on Computing Education, v10, no. 4, Nov., pp. EJ908803, 2010.
[14] Scratch.mit.edu, "Scratch - Imagine, Program, Share", 2016. [Online]. Available: https://scratch.mit.edu. [Accessed: 01- Mar- 2016].
[15] reflectoring.io, 2023. How to Build Responsive Web Apps with Spring Boot and React: A Step-by-Step Guide. [Online] Available at: https://reflectoring.io/build-responsive-web-apps-with-springboot-and-react-tutorial/ [Accessed 19 09 2023].
[16] S. J. Whittall, W. A. C. Prashandi, G. L. S. Himasha, D. I. De Silva, T. K. Suriyawansa, "CodeMage: Educational programming environment for beginners," Nineth International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, February 1-4, 2017, pp. 311-316.