IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

Personalized Learning Chatbot for Algorithm Demonstration

[1]Naheed Sultana, [2]M. K. Charani, [3]C. Vaishnavi, [4]B. Jahnavi
Assistant Professor, Department of Information Technology[1]
Student, Department of Information Technology[2,3,4]
Stanley College of Engineering and Technology for Women, Hyderabad, Telangana, India,
naheed.ruquiya@gmail.com

Abstract: *Natural language processing (NLP) and artificial intelligence (AI) are revolutionizing education by making intelligent systems like chat-bots and massive language models possible, which facilitate interactive and individualized learning. Along with providing new avenues for educational progress, these technologies improve human-computer connection. But a lot of students still have difficulty in grasping abstract algorithmic ideas, especially in programming languages, because of the drawbacks of conventional, static teaching approaches. Combining conceptual clarity, engaging visual explanations, and natural language engagement is becoming more and more necessary in instructional approaches. Closing this gap can make programming education much more accessible and effective, especially for beginner.*

Keywords: Artificial Intelligence, NLP, Human-Computer Interaction (HCI), Personalized Learning, chat-bot, visual learning, sustainability

I. INTRODUCTION

AI is changing our daily lives by inventing and testing intelligent software and hardware, commonly referred to as intelligent agents. Intelligent computers may do a variety of tasks, from physical work to complicated operations. A chatbot is a simple sort of artificial intelligence, and it was one of the first and most widely used examples of human-computer interaction. They serve throughout fields that encompass education, information retrieval, business, and e-commerce[1]. A new era of learning and discovery, centred on chat-bots and artificial intelligence, is rapidly emerging. There has recently been a surge of interest in incorporating AI systems and chat-bots into educational settings. An additional benefit of AI systems and chat-bots in education is their capacity to follow learning pathways [2].This technology has a lot to offer education, especially in terms of personalizing language learning experiences. Large Language Models (LLMs), such as Large Language Model Meta AI (LLaMa), have grown in popularity as a result of the success of deep learning in natural language processing (NLP). They utilize robust computing to manage massive volumes of data utilizing a number of methodologies. Because of their ability to accurately capture the semantic links between words and sentences, these models, together with chat-bots, have become powerful tools that have inspired an unbeatable change in education [3]. The advancement of AI and NLP technology has resulted in a significant evolution in chat-bots, which can now do more complex activities and engage in interactions more similar to those of humans. In general, the development of chat-bots stems from the growing demand for more effective and practical ways to interact with technology, as well as advances in AI and NLP technologies [2].

Despite the increasing need for conceptual clarity in algorithmic thinking and data structure comprehension, many students find it difficult to understand abstract algorithmic logic when presented with textual code and static information. Conventional education approaches, like as lectures and textbooks, frequently lack interaction and don't show code behavior or offer real-time visual feedback. In addition, new students have trouble connecting algorithm theory to real-world application, particularly in lower-level languages like C. A number of the e-learning technologies available today offer a single platform that combines interactive chat-bot guidance, automatic code creation, and animated algorithm execution that is responsive to user input. Other platforms provide visualization or chat-bot support

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

separately. Personalized learning and practical experience are essential for understanding the foundations of programming and data structures, but they are restricted by this gap. As a result, an intelligent, interactive system is required that allows students to interact with algorithms through natural chatbot conversation, creates executable C code dynamically in response to user input, and offers animated logic execution in steps. Particularly for those new to computer science, such a platform would make learning more approachable, interesting, and customized.One of the most challenging aspects of learning programming is to find engaging examples that are not only fun and impressive but also illustrative of the subject matter at hand. That's why using the JavaScript library, an add-on called p5.js, is ideal for teaching and learning programming. p5.js will allow you to view the output of our scripts as graphics and gain a thorough comprehension of programming structures[4]. To bridge this gap, we developed an animation engine capable of producing flexible flicks for algorithm education. We created AAnim, an object-oriented library for animating algorithms. AAnim is built on Manim, an animation engine that generates simple animations and shapes. Users interact with AAnim using input commands. Then, AAnim runs the list of queries or the graph algorithm, animating the data structure and indicating the execution's current location in the pseudocode. Finally, AAnim produces a video file as its output [5].

II. RELATED WORK

Recent advancements in artificial intelligence (AI) have spurred a significant shift in educational practices, particularly through the adoption of AI-powered chatbots, algorithm visualization platforms, and creative coding integrations. A systematic literature review (2018–2023) synthesized peer-reviewed studies, highlighting that AI chatbots like ChatGPT, Bard, Ada, and Replika enhance student learning by offering personalized support, step-by-step feedback, and interactive engagement. Educators also benefit from chatbot-driven automation in tasks such as grading, content generation, and administrative scheduling. However, the review also underscores critical challenges related to the accuracy of chatbot outputs, especially in domains like medical education, and broader concerns such as academic integrity, data privacy, algorithmic bias, and digital inequality.

To mitigate these issues, the study calls for institutional policy-making, teacher training, and ethical integration strategies [6].Expanding on practical design aspects, Aleedy et al. in their work “Using AI Chatbots in Education” categorize chatbot architectures into rule-based, machine learning-based, deep learning-based, and ensemble models, and present a case study involving the Hubert chatbot [7]. Deployment supported student engagement in a machine learning course by generating reflective prompts and collecting feedback via a dedicated dashboard. While the system demonstrated high participation, limitations such as poor language handling (especially Arabic), repetitive dialogue, lack of emotional understanding, and dashboard inefficiencies were noted, suggesting the need for hybrid models with enhanced context sensitivity and multi-domain handling capabilities.

Further addressing pedagogical challenges in computer science (CS) education, proposes a multi-role AI chatbot framework to overcome issues like high student-teacher ratios and limited personalization. Grounded in Self-Determination Theory, the system incorporates four specialized bots—Instructor, Social Companion, Career Advisor, and Emotional Supporter—each fulfilling distinct roles to improve learner autonomy, competence, and relatedness[8]. An experimental study involving 200 postgraduate CS students in Germany revealed that the multi-role chatbot system, powered by large language models (LLMs), significantly improved engagement and higher-order learning outcomes when compared to traditional and single-bot approaches .In parallel, the field of algorithm visualization has long aimed to enhance student understanding of abstract computational processes. A large-scale survey of 350 visualization tools indicates that most focus on simple algorithms (e.g., sorting), suffer from limited interactivity, and are hosted on unstable or outdated platforms such as Java applets [9]. The authors advocate for a collaborative, repository-driven approach with standardized quality metrics to improve sustainability and pedagogical utility.

Complementing this, Šimonák’s 2014 work introduces VizAlgo, a modular, plugin-driven Java platform designed to animate algorithm behavior. Survey feedback from 53 undergraduates confirmed that visualizing algorithms like HeapSort and RadixSort improved comprehension, and students emphasized the need for support in advanced topics like balanced trees and hashing, as well as features like step-back controls and online access [10].On a more creative frontier, the integration of generative AI with creative coding platforms like p5.js is reshaping how visual and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

computational thinking is taught. Traditional algorithmic art, while expressive, is often limited in stylistic diversity. New tools such as GenP5 and P52Style allow bidirectional interaction: GenP5 enhances p5.js sketches with AI-generated styles (e.g., abstract painting textures), while P52Style transfers algorithmic aesthetics to AI-generated imagery [11]. These systems support both procedural and learned representations, offering a powerful expansion of digital expression in educational and artistic domains. Collectively, these studies affirm that while AI-driven technologies offer immense promise in transforming education—through enhanced personalization, visualization, and creativity—they also demand thoughtful design, ethical foresight, and interdisciplinary collaboration to ensure scalable, inclusive, and meaningful learning outcomes.

III. REALIZATION OF THE PROPOSED FRAMEWORK

The proposed system is an innovative, AI-driven educational platform designed to revolutionize algorithm learning by combining chat-bot interactions, automatic code generation, and real-time visualizations. It is composed of several modular components, each responsible for a specific stage in the learning workflow—from understanding user queries to generating educational animations. These include,

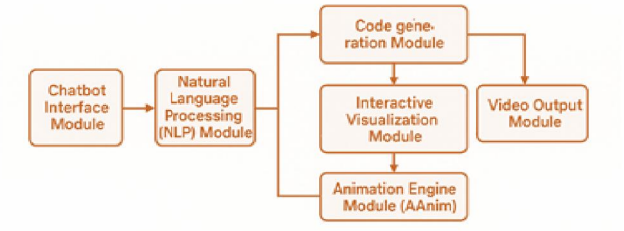

Fig. 1 Outline a broad architectural layout and go into great depth about the key elements of each component.

Chatbot Interface Module

The Chatbot [1, 2] Interface Module functions as a dynamic learning gateway, enabling users to interact with the system through natural, conversational language. Whether learners are seeking explanations or exploring specific algorithms, this module ensures a smooth and engaging experience. It is designed to accommodate a wide range of queries—both structured and unstructured—while adapting its responses to suit the user's level of understanding. With built-in mechanisms for identifying errors and prompting clarifications, the module creates a supportive environment that promotes exploration and effective learning.

Natural Language Processing (NLP) Module

The Natural Language Processing (NLP) Module acts as the system's interpreter, transforming user-friendly queries into precise, actionable components. Leveraging the capabilities of Large Language Models (LLMs), it accurately identifies user intent and extracts key parameters necessary for generating code and animations[3]. This module excels at handling nuanced queries, accommodating variations in phrasing and ambiguous language while maintaining contextual relevance. Additionally, it adapts to individual users by considering their past interactions and learning patterns, making the system more personalized and effective in an educational setting.

Code Generation Module

The Code Generation Module is responsible for transforming interpreted user input into executable C code that is both syntactically correct and pedagogically sound. It emphasizes clarity and structure, making the code accessible even to beginners. By incorporating user-defined inputs and constraints, the module tailors the output to specific learning scenarios. Additionally, it enriches the generated code with explanatory comments and utilizes dynamic templates to support a wide range of data structures and algorithms, ensuring alignment with academic standards and enhancing the learning experience.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

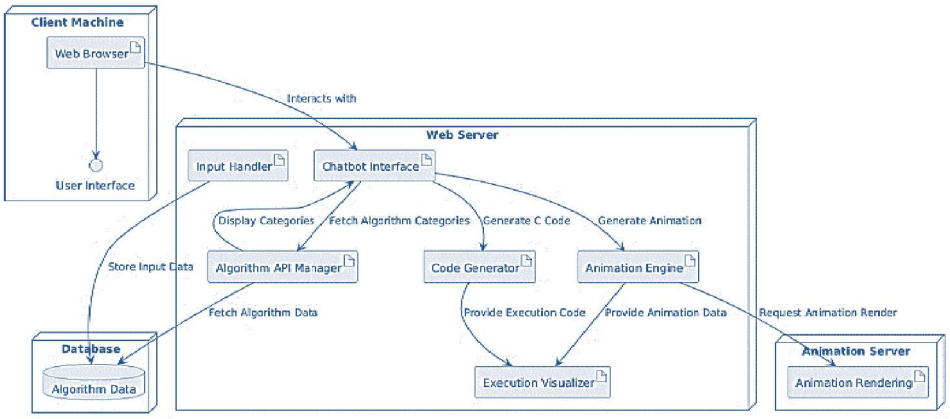

Fig. 2 Deployment diagrams, which are used to evaluate the effects of distribution and resource allocations, concentrate on the setup of the runtime processing nodes and their constituent parts and artifacts.

Interactive Visualization Module

The Interactive Visualization Module uses the p5.js JavaScript library to provide dynamic, real-time graphical representations that enhance conceptual understanding of programming logic and algorithmic behaviour[4]. Designed as a supportive visual engine to complement the animation module, it enables users to engage with interactive simulations such as sorting and tree traversals. This module delivers immediate feedback based on user-written scripts and operates directly within the browser, ensuring wide accessibility. By translating abstract code into intuitive visual forms, it fosters deeper insight and helps learners bridge the gap between theory and practice.

Animation Engine Module (AAnim)

AAnim is a robust, object-oriented animation engine built on top of Manim, designed to deliver precise, step-by-step visualizations of algorithm execution and data structure transformations. It synchronizes animations with highlighted pseudocode, allowing learners to follow the logical progression of operations in real-time[12]. From arrays and stacks to trees and graphs, AAnim brings various data structures to life with smooth transitions and execution pointer movements. It dynamically adjusts based on user input, offering an immersive and interactive experience that deepens comprehension through animated simulation.

Video Output Module

The Video Output Module serves as the concluding stage of the system, transforming the animations produced by AAnim[12] into polished, educational videos. These videos are exportable in high-definition MP4 or GIF formats and are tailored for versatility in learning environments. With options to customize duration, resolution, and frame rate, the module ensures that the final output meets varied presentation needs. Its built-in playback controls enhance classroom usability, and the generated content can be seamlessly reused in lectures, tutorials, or embedded into online learning platforms.

IV. PROGRAMMING CONSTRUCTS AND PSEUDOCODE

```
function main():
  while true:
    raw_input←InputHandler.receive_from_browser() store_user_input(raw_input)

     intent,params←ChatbotInterface.parse_intent(raw_input)

    if intent == "LIST_CATEGORIES" then
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

```
    categories←AlgorithmAPIManager.fetch_categories()
    ChatbotInterface.display_categories(categories)

else if intent == "SELECT_ALGORITHM" then
    algo_name ← params.algorithm
    // fetch the algorithm's definition, pseudocode, sample data, etc.
    algo_data←AlgorithmAPIManager.fetch_algorithm_data(algo_name)

    // generate C code from the fetched algorithm spec
    c_code←CodeGenerator.generate_code(algo_data)
    ChatbotInterface.display_code(c_code)

    // execute or simulate the code to get step-by-step traces
    exec_trace←ExecutionVisualizer.provide_execution(c_code)
    ChatbotInterface.display_execution(exec_trace)

    // prepare animation data (steps, visuals) for rendering
    anim_payload←AnimationEngine.provide_animation_data(algo_data, exec_trace)
    AnimationServer.render(anim_payload)

    // send back a link or embedded object for the rendered animation
    animation_link←AnimationServer.get_rendered_url(anim_payload)
    ChatbotInterface.display_animation(animation_link)

else
    ChatbotInterface.display_error("Sorry, I didn't understand that.")
```

V. RESULTS & OUTPUTS

Fig. 3 Welcome Page enabling users to interact with the system ensures a smooth and engaging experience

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

Fig. 4 User interfaces (UI) combine tactile input with visual and aural output.

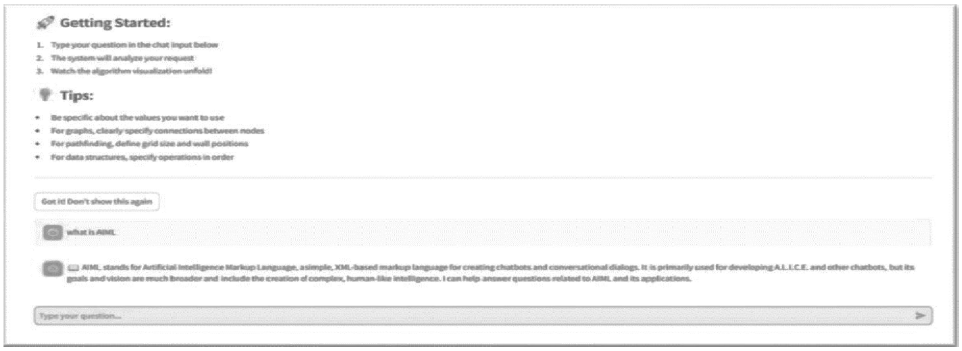

Fig. 5 Assist people with their quests by offering a "Getting Started" guide, practical advice on how to write effective inquiries, and a text input box where users may enter their own questions.

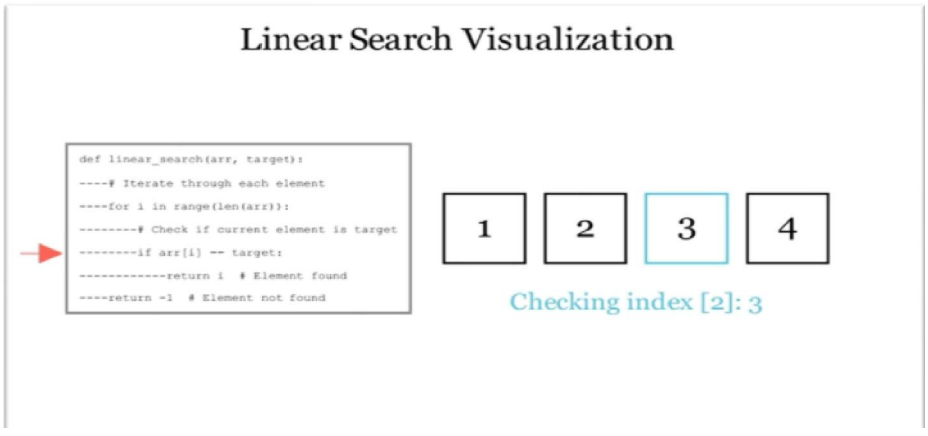

Fig. 6 Using both code and a visual array representation, the image demonstrates how the Linear Search algorithm operates, assisting students in making the connection between the algorithm's control flow and its actual data processing, hence enhancing comprehension and debugging abilities.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

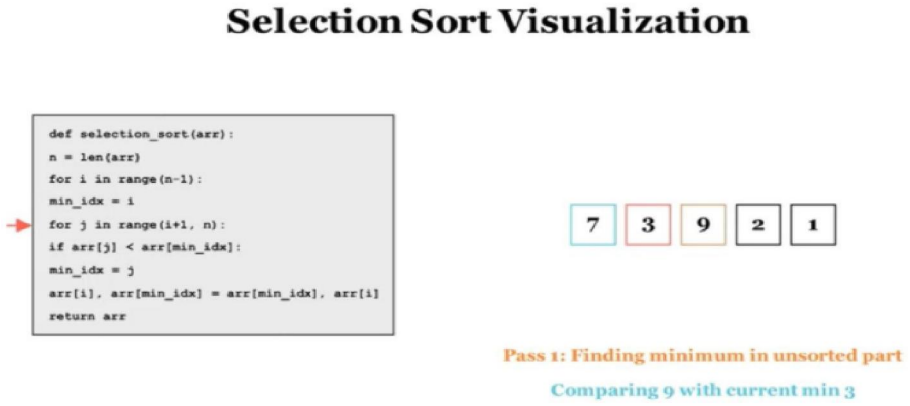

Fig. 7 It helps them understand the swapping and selection mechanisms of the selection sort, as well as the logic behind each sorting phase. It also provides a detailed explanation of the Selection Sort method by combining code logic with a graphical array illustration to improve comprehension.

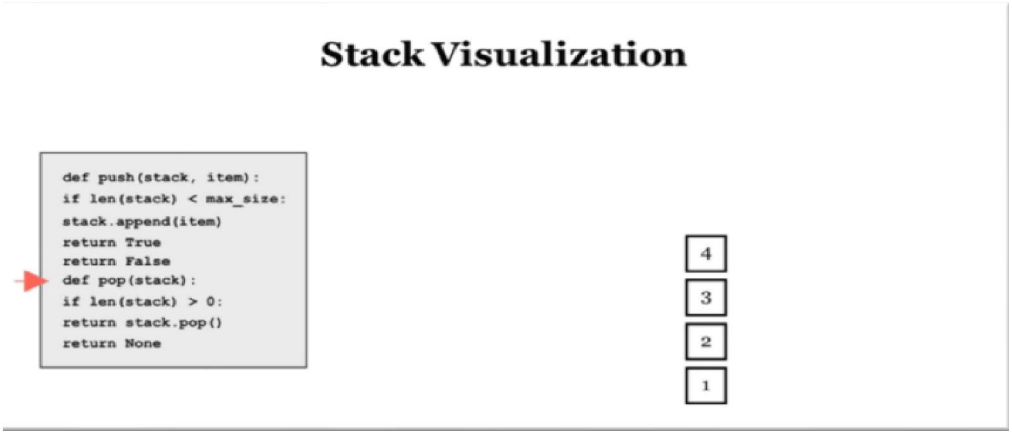

Fig. 8 The provided graphic simplifies the learning of stack operations like push and pop by illuminating the concept of a stack both visually and through code. The code defining the rationale for these operations is on the left. When the push ( ) method determines that the stack's current size is less than a predetermined maximum size, it appends the item to the stack and returns True; if not, it returns False. In the event that the stack is not empty, the pop () function returns None; otherwise, it removes and returns the topmost element. A vertical stack element arrangement is shown in the graphic depiction on the right side, with one element at the bottom and four at the top. The LIFO (Last In, First Out) principle of stacks is reinforced by this graphic, which makes it evident that element 4 was added most recently and will be the first.

VI. CONCLUSION

In this study, we introduced a comprehensive, AI-powered learning platform that improves the learning experience of algorithmic concepts, especially for unfit programmers, by utilizing the combined powers of conversational agents, natural language processing, code generation, and dynamic visualization. By enabling real-time feedback, customized assistance, and understandable interactions, the suggested framework tackles important pedagogical issues. Additionally, it offers executable and visual representations of complex conceptual platforms using its scalable and modular architecture. The framework bridges the gap between abstract algorithm theory and real-world comprehension by incorporating technologies like Manim-based AAnim, p5.js, and Large Language Models (LLMs)[8][9][11]. It encourages greater engagement and retention by transforming traditional, stuffed with text training into a multimodal

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

learning environment tailored to different learner needs. Its automatic generation of annotated C code, synchronized animation of pseudocode execution, and exportable visual components further establish this platform as a versatile tool for both in-class and self-directed learning situations.

The results support the increasing potential of AI and chatbot-driven solutions that will transform technical education by lowering the cognitive burden related to abstract computational thinking, fostering learner autonomy, and boosting interactivity [12].Future research is going to focus on improving adaptive learning paths via learner analytics, adding emotional and sentiment-aware answers, broadening language support, and connecting with cloud-based learning management systems to guarantee wider accessibility and large-scale implementation. In the end, this study advances the continuous development of intelligent educational systems by showing how AI may effectively enhance and support the teaching and learning of fundamental computer science ideas in a way that is impactful, inclusive, and interactive.

REFERENCES

[1]. Artificial Intelligence Applications and Innovations. 2020 May 6;584:373–383. doi: 10.1007/978-3-030-49186-4_31]

[2]. Kooli, Chokri. "Chatbots in Education and Research: A Critical Examination of Ethical Implications and Solutions." Sustainability, vol. 15, no. 7, 2023, p. 5614, https://doi.org/10.3390/su15075614.

[3].Cabezas, Darío S., et al. "Integrating a LLaMa-based Chatbot with Augmented Retrieval Generation as a Complementary Educational Tool for High School and College Students." Proceedings of the 19th International Conference on Software Technologies (ICSOFT 2024), SCITEPRESS, 2024, pp. 395–402. https://doi.org/10.5220/0012763000003753.

[4]. https://www.apress.com/gp/blog/all-blog-posts/javascript-and-p5-js/15548356

[5]. Z. J. Liu and T. Sun, "AAnim: An Animation Engine for Visualizing Algorithms and Data Structures for Educators," in Proc. SIGCSE 2021, San Francisco State University, 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3408877.3432449

[6]. Labadze, L., Grigolia, M., & Machaidze, L. (2023). Role of AI chatbots in education: Systematic literature review. International Journal of Educational Technology in Higher Education, 20(56). https://doi.org/10.1186/s41239-023-00426-1

[7]. Aleedy, M., Atwell, E., & Meshoul, S. (2021). Using AI Chatbots in Education: Recent Advances, Challenges and Use Case. Presented at the International Conference, November 2021. Retrieved from https://www.researchgate.net/publication/357606102

[8]. Müller, T. Schmidt, and J. Weber, “AI Chatbots as Multi-Role Pedagogical Agents: Transforming Engagement in CS Education,” Proc. 15th Int. Conf. on Educational Technology (ICET), pp. 112–121, 2023.

[9]. Shaffer, C. A., Cooper, M., & Edwards, S. H. (Year). Algorithm visualization: A report on the state of the field. Department of Computer Science, Virginia Tech.

[10]. Šimoňák, S. (2014). Using algorithm visualizations in computer science education. Open Computer Science, 4(1), 54–65. https://doi.org/10.2478/s13537-013-0105-2

[11]. Exploring Bridges Between Algorithmic and AI-generated Art Jiaqi Wu 1 and Eytan Adar 1 University of Michigan, Ann Arbor MI 48109, USA {wujiaq, eadar}@umich.ed

[12]. Liu, Z. J., & Sun, T. (n.d.). AAnim: An animation engine for visualizing algorithms and data structures for educators. San Francisco State University. Retrieved from https://zhuozhuocs.github.io/AAnim/.

[13]. K. Mehlhorn, P. Sanders, Algorithms and Data Structures (Springer-Verlag, Berlin Heidelberg, 2008)

[14]. S. Diehl (Ed.), Software Visualization, Lecture Notes in Computer Science 2269, 2002

[15]. M.H. Brown, R. Sedgewick, A system for algorithm animation, Proceedings of the 11th annual conference on

[16]. Computer graphics and interactive techniques, SIGGRAPH’84 (ACM New York, NY, USA, 1984)

[17]. D.J. Jarc, M.B. Feldman, R.S. Heller, Assessing the benefits of interactive prediction using Web-based algorithm

[18]. Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). Programming is hard – or at least it used to be: Educational opportunities and challenges of AI code generation. In Proceedings of the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, May 2025
Impact Factor: 7.67

2023 ACM Conference on Global Computing Education (CompEd '23). ACM. https://doi.org/10.1145/3576885.3618486

[19]. Heidy Khlaaf, Pamela Mishkin, Joshua Achiam, et al. 2022. A Hazard Analysis Framework for Code Synthesis Large Language Models. https://doi.org/10.48550/arxiv.2207.14157

[20]. Juho Leinonen, Arto Hellas, Sami Sarsa, et al. 2022. Using Large Language Models to Enhance Programming Error Messages. https://doi.org/10.48550/arxiv.2210.11630