LLM-enhanced Learning Environments for CS: Exploring Data Structures and Algorithms with Gurukul

Ashwin Rachha
Department of Computer Science
Virginia Tech
Blacksburg, USA
ashwinr@vt.edu

Mohammed Seyam
Department of Computer Science
Virginia Tech
Blacksburg, USA
seyam@vt.edu

***Abstract*—In this Innovative Practice full paper, we introduce Gurukul, an innovative coding platform designed to support teaching Data Structures and Algorithm (DSA) course by integrating advanced Large Language Models (LLMs). LLMs have emerged as powerful tools in Computer Science Education (CSEd), offering unparalleled opportunities for enhancing student comprehension and engagement. However, their use in educational settings presents challenges, including tendencies toward hallucination, contextual inaccuracies, and the risk of undermining critical thinking by providing explicit solutions. To address these challenges, and to explore how specialized LLMs can bolster learner engagement, we present Gurukul, a platform featuring dual innovations: Retrieval-Augmented Generation (RAG) and Guardrails. Gurukul offers a hands-on practice feature where students can solve DSA problems within a code editor, supported by a dynamically Guardrailed LLM that prevents the delivery of explicit solutions. Additionally, the platform's study feature utilizes RAG, drawing from OpenDSA as a trusted source, to ensure accurate and contextually relevant information is provided. To assess the platform's effectiveness, we conducted a User Study with students, and a User Expert Review with faculty from a U.S. public state university specializing in DSA courses. Our analysis of student usage patterns and perceptions, along with insights from instructors, reveal that Gurukul positively impacted student engagement and learning in DSA, demonstrating the potential of specialized LLMs to enhance educational outcomes in this field.**

***Index Terms*—Large Language Models, Retrieval Augmented Generation, Guardrails, Computer Science Education, ChatGPT, AI in Education**

I. Introduction

Data Structures and Algorithms are fundamental components of computer science that equip students with the necessary skills to solve complex real world problems efficiently. Being skilled in DSA is not only pivotal for academic success but also critical for professional competence in software development and computational problem-solving. However, despite its importance, teaching and learning DSA present significant challenges, primarily due to the abstract nature of the concepts involved [1]. As students struggle to link theoretical principles with practical applications, there is another problem which is the one-size-fits-all [2] approach of traditional teaching relying solely on lecture based methods and static textbooks where students are passive recipients of knowledge rather than being active participants. This is exacerbated by the problem of reduced levels of engagement due to lack of interactive feedback [3].

The advent of Large Language Models like ChatGPT [4], Claude [5] and Llama [6] has changed the landscape of education as we know it. These advanced AI tools have the potential to enhance the learning experience by providing personalized, context-aware assistance that traditional educational methods often lack. In computer science education, LLMs can help bridge the gap between theoretical concepts and practical applications by providing interactive coding assistance, troubleshooting and instant clarification of doubts. Hence the proliferation of LLMs in education tools is inevitable. The application of LLMs, however, is not without challenges. Two primary issues inherent to the way LLMs are built are their propensity for "hallucinations" - generating plausible but incorrect or irrelevant content - and contextual inaccuracies. These issues can mislead learners, especially in complex subjects like DSA where precision is critical. Additionally, LLMs can be easily manipulated to provide direct answers, thus bypassing critical thinking and problem-solving processes essential in education [7], [8]. Other concerns include the reinforcement of biases and the lack of adaptability to different learning contexts, which could further undermine their educational value. Addressing these challenges while building LLM based educational tools is crucial to ensuring that LLMs enhance rather than detract from the educational experience.

In response to these challenges, we present the Gurukul platform, developed to explore design patterns and concepts for creating specialized LLM-based applications that enhance engagement and learning experiences in DSA courses. Gurukul's theoretical foundation is built on the premise that modern educational tools should provide interactive, personalized learning experiences that actively engage students and adapt to their individual needs. By incorporating RAG and Guardrails within its LLMs, Gurukul seeks to ensure that the educational content is not only accurate and relevant but also ethically aligned with educational standards and motives. This approach is meant to encourage the development of critical thinking skills rather than complete reliance on LLMs, fostering a more balanced and effective learning environment.

This study is guided by two principal research questions:

RQ1 - What are the user perceptions towards using Gurukul as an educational tool, integrated with specialized LLMs regarding learning comprehension and engagement in DSA courses?

RQ2 - How do Guardrails and Retrieval Augmented Generation work towards enhancing engagement and learning productivity, and what could be the design principles to consider while employing such applications?

II. Review of Literature

The utility of large language models like GPT4 [4], [9], Llama and Claude has opened up multiple avenues for educational use. In Computer Science Education (CSEd), recent studies have demonstrated the potential of these models to improve learning and teaching experiences in university-level programming courses [10]–[12]. Studies on the utilization of Artificial Intelligence have been an area of active research for many years. However, the investigation of the Large Language Models in educational settings has emerged more recently [13], [14]. As we examine the technological landscape in education, particularly within the confines of Computer Science and DSA, it is imperative to critically analyze existing digital tools and educational platforms that cater to these specialized subjects.

A. AI in Education

Historically, AI in education has seen significant interest, especially with the advent of Intelligent Tutoring Systems and Chatbots [16]. AI has become central in transforming educational paradigms, offering personalization and adaptability across diverse environments [17]. As show in Fig. 1, the interest in 'AI in Education' has seen a significant rise from 2004 to 2024, reflecting the growing importance of AI technologies in the educational sector.

Traditional CS education faces challenges, especially in foundational courses like DSA, due to the abstract nature of the content, diverse student backgrounds, and conventional teaching methods. Lecture-based delivery often fails to engage students or effectively convey complex concepts. Yadav et al. (2016) and Knobelsdorf and Vahrenhold (2013) highlight the isolation and lack of adequate CS background among teachers as significant barriers [1]. Webb et al. (2017) who argue for a curriculum that integrates real-world applications to better prepare students for professional challenges in CS. [18].

Innovative approaches like Problem-Based Learning (PBL) shift the focus from passive information reception to active problem-solving, helping students develop practical skills and deeper understanding. Kay et al. (2000) describe the effectiveness of PBL in improving student engagement and learning outcomes in CS courses [19].

Traditional DSA education relies heavily on theoretical lectures, which may not address all students' learning needs. This can lead to poor engagement and retention of information. Tamassia (1998) emphasizes the challenge of conveying abstract concepts and developing efficient implementations [20]. Steingartner et al. (2019) note that traditional methods often fail to account for the wide content scope and student variability, leading to inconsistent learning outcomes [21]. Su et al. (2021) discuss the difficulty students face in visualizing and understanding DSA concepts due to traditional teaching methods [22]. Tang et al. (2012) highlight the importance of incorporating experimental teaching to enhance engagement and learning efficiency [23].

B. Applications of Large Language Models in CS Education

Large Language Models (LLMs) offer transformative tools in CS education, providing novel approaches to teaching and learning. Models like GPT-3 and Codex facilitate the automatic generation of programming content, quizzes, and courses tailored to individual learning needs. Macneil et al. (2022) highlight LLMs' potential to generate code and explanations, aiding the learning process [24].

LLMs create personalized learning experiences that adapt to individual students' pace and style. Krüger and Gref (2023) show that LLMs enhance learning through interactive problem-solving scenarios [25]. Tran et al. (2023) demonstrate that LLMs can reduce educators' workload while enhancing the curriculum's cultural relevance [26].

However, LLMs come with inherent drawbacks. They are computationally expensive and require substantial hardware and energy resources, limiting their accessibility and scalability. Meyer et al. (2023) discuss ethical concerns surrounding LLM deployment, such as data privacy and bias [12]. In educational contexts, LLMs raise issues of accuracy and reliability. Hamaniuk (2021) emphasizes the need for careful oversight to ensure accurate and contextually appropriate information [27]. Macneil et al. (2022) advocate for a balanced approach that uses LLMs to enhance, not replace, traditional teaching methods [28]. Joseph et al. (2023) propose a Critical AI Literacy framework to help teachers adopt LLM-based tools [29]. Academic

Fig. 1. Google Trends results for the term "AI in Education" from 2004 to 2024 [15].

integrity is another concern, as LLMs can generate essays and answers that could be used for cheating.

C. Retrieval Augmented Generation and Its Uses in Education

Retrieval Augmented Generation (RAG) enhances AI's personalization and accuracy in education. RAG combines generative capabilities of models like GPT with retrieval functions similar to search engines, improving factual accuracy and reducing hallucination. Jiang et al. (2023) describe methods for actively deciding what and when to retrieve, enhancing the relevance and accuracy of generated content [30]. Wang et al. (2023) introduce Chat-Ed, a chatbot architecture that combines ChatGPT with a retrieval-based framework to support higher education [31].

In education, RAG creates dynamic and responsive learning materials. Li et al. (2022) survey RAG applications in education, highlighting its use in dialogue response generation and machine translation [32]. Mao et al. (2020) developed a Generation-Augmented Retrieval system for open-domain question answering, enriching query semantics and improving retrieval accuracy [33]. Manathunga and Illangasekara (2023) discuss RAG's use in medical education, integrating up-to-date knowledge into learning platforms [34].

D. LLMs with Guardrails and Their Uses

Large Language Models (LLMs) require guardrails to prevent unintended or inappropriate content. Guardrails ensure LLM outputs align with ethical standards and educational goals. Liffiton et al. (2023) explore CodeHelp, an LLM-powered tool providing programming assistance without revealing solutions, emphasizing the importance of guardrails to balance assistance and over-reliance [35].

Guardrails maintain the integrity and safety of LLM applications, particularly in sensitive areas like education. Sun et al. (2023) describe "Contrastive and Scenario-Guided Distillation" methods to tailor LLM outputs to predefined rules, ensuring content accuracy [36]. Rebedea et al. (2023) introduce NeMo Guardrails, a toolkit for adding programmable guardrails to LLM applications, providing flexibility and control over outputs [37].

In conclusion, integrating guardrails in LLM applications is crucial for safe and effective use in education. By aligning outputs with ethical standards and educational objectives, guardrails help leverage LLM benefits while minimizing risks. Ongoing development of guardrail technologies will enhance the reliability and safety of LLM applications in educational settings.

III. Methodology

This section outlines the system architecture and implementation of the Gurukul application, emphasizing both theoretical aspects and core functionalities.

A. Design

1) System Architecture: The Gurukul platform is built on a Hybrid Client-Server Architecture, utilizing Next.js for its flexibility in rendering strategies, such as Server Side Rendering (SSR) and Static Site Generation (SSG). This architectural choice enhances the application's performance and user experience by ensuring rapid loading times and effective Search Engine Optimization (SEO) while maintaining rich interactivity. The backend is powered by Node.js, which efficiently handles API requests, user authentication, and database operations.

2) Retrieval Augmented Generation: RAG forms the pedagogical backbone of the study feature, addressing the limitations of static LLMs by augmenting them with an external knowledge base. This process synthesizes information retrieval from a vast corpus of data with the generative capabilities of advanced language models. This feature was incorporated in the Study section of the application.

Mechanism of Action:

Fig. 2 illustrates the mechanism of RAG in the Gurukul application, showcasing how the system retrieves and generates contextually relevant educational content. The steps in which how RAG was implemented in gurukul are as follows:

- **Data Crawling and Preprocessing:** OpenDSA was selected as the primary source of documents

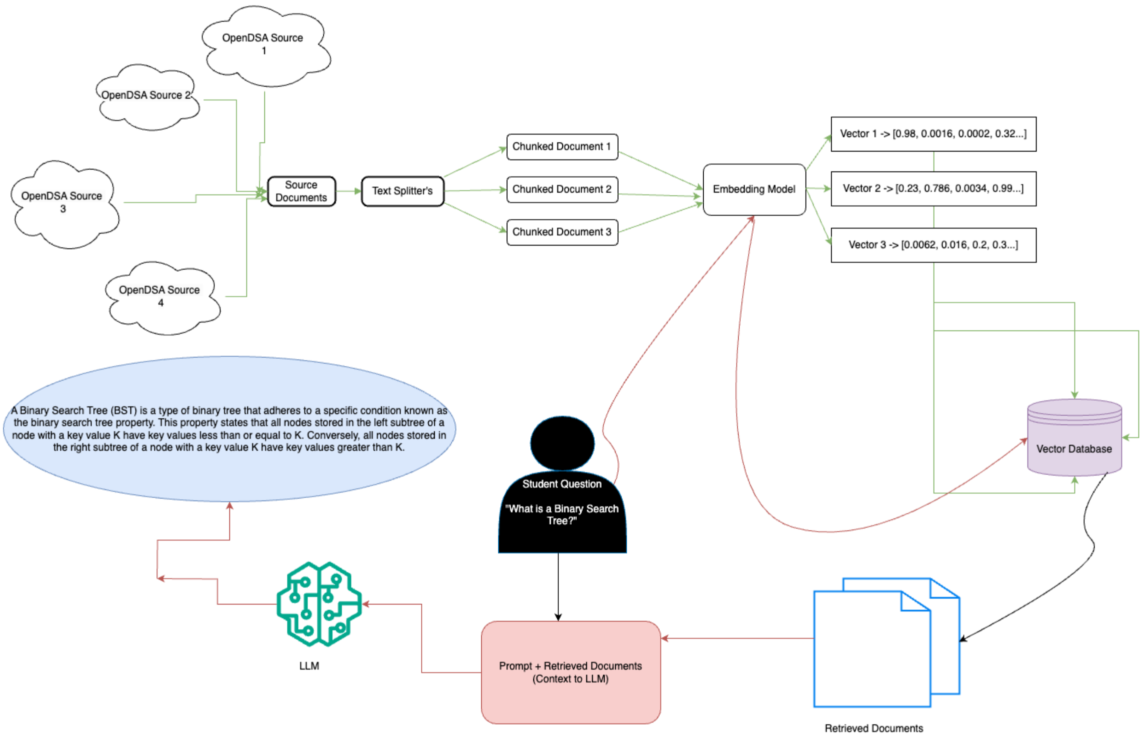

Fig. 2. A diagram illustrating the RAG process.

for the LLM to refer to, providing a comprehensive repository of educational content. The HTML documents from OpenDSA were converted into dense word vectors and stored in Supabase's PostgreSQL-based vector database (pgvector).

- **Document Ingestion** Collects a wide range of documents and data sources to serve as the knowledge base.
- **Document Chunking:** Breaks down large documents into smaller, manageable pieces to enhance retrieval efficiency.
- **Converting Documents to Vector Databases:** Transforms textual data into mathematical vectors using embedding models, storing these vectors in specialized databases for rapid retrieval. The process involves running the data through an embedding model to convert it into vectors. These vectors are stored in a vector database optimized for high-dimensional vector data, facilitating efficient retrieval during the RAG process.
- **Using Embeddings for Semantic Search:** Utilizes vector embeddings to understand the intent and contextual meaning behind queries, enhancing the relevance of fetched information.
- **Using LLMs with Vector Databases for RAG:** Integrates LLMs with vector databases to provide comprehensive, accurate, and contextually enriched responses. When a learner poses a query, the system vectorizes the query, searches the VectorDB for relevant information, and retrieves the most relevant content. This information is then synthesized by the LLM to generate accurate, contextually enriched responses.

3) Guardrails and Their Implementation: Guardrails are essential for maintaining the integrity and safety of AI applications, particularly in educational settings. They prevent the model from generating harmful, biased, or inappropriate content that could mislead learners or propagate misinformation.

System Prompts: In the context of the Gurukul platform, the OpenAI API is configured with a system prompt to restrict the LLM's responses to educational content only. For instance:

```
const systemMessage = {
  role: "system",
  content: "You are a benevolent helping Teaching
        Assistant in Data Structures and Analysis. You
         only answer questions related to Data
        Structures and Algorithms, test cases, and
        imaginary test cases. You DO NOT OUTPUT ANY
        CODE."
};
```

The LLM assistant is designed to guide students through problem-solving without providing direct answers, encouraging critical thinking and problem-solving skills. The guardrails ensure that the assistant promotes ethical learning practices by preventing academic dishonesty.

Live Code Editor: The live code editor allows students to write, test, and debug code in a real-world coding environment. It features syntax highlighting, error detection, and code completion to promote a smoother learning curve. The monaco editor [38] component was used as the code editor and additional features included choice of programming language between Python, Java, C++, C, Go and Javscript. Fig 3 shows the code editor and llm assistant with a sample interaction.

Interactive Learning: The practice section integrates the LLM assistant to provide instant feedback, hints, explanations, and alternative solutions, facilitating immediate reinforcement and consolidation of new knowledge. Three static hints generated by ChatGPT were already provided in the code editor. These were meant to be seed points for understanding how to solve the problem. The hints were ordered in increasing level of verbosity with hint1 having least information and hint3 having most. The LLM assistant was embedded right next to the Code Editor for easy asking of questions. The editor was also incorporated with functionality to test and compile code with boiler plate code always rendered whenever users chose their appropriate programming langauge.

IV. Results

To evaluate Gurukul, we adopted a two-pronged approach. First, we conducted a User Expert Review involving experts from a public university in the USA. Second, we performed a User Study with students interested in or already enrolled in Data Structures and Algorithms courses. This dual approach provided a comprehensive and holistic evaluation from both pedagogical (instructor) and student perspectives. By incorporating insights from both groups, we aimed to assess the platform's effectiveness in facilitating learning and teaching in a balanced manner. Unfortunately, direct comparative data from previous years is unavailable. However, based on qualitative feedback from instructors who taught DSA courses in prior years, the Gurukul platform demonstrated improved student interaction and engagement compared to traditional teaching methods. Future studies will incorporate longitudinal data to provide more robust comparisons.

A. *User Expert Review*

The User Expert Review was conducted to evaluate the Gurukul platform's usability and educational effectiveness. The experts assessed various aspects of the platform, such as interface design, functionality, and overall user experience.The objectives of the Expert Review were to assess the usability and intuitive design of the platform, evaluate the effectiveness of the educational tool provided and gather expert feedback for potential improvements. The key features that were evaluated were

1) ***User Interface and Site Navigation****:* The professors found the platform visually appealing and professionally designed. The clean and structured layout facilitated ease of navigation, potentially reducing the learning curve for new users. Simplicity in user experience was emphasized, and the interface was described as intuitive with well-delineated sections for theory, practical exercises, and interactive tools. The ability to toggle to night mode stood well with the experts as a point was made specifically about how it might be beneficial for the visually impaired. Comments were made regarding formatting and font size for the LLM assistant. Suggestions for improvement included enhancing font size and contrast for users with visual impairments, and summarizing overly verbose error messages.

The logical flow between different sections was positively received, with recommendations for better labeling of interactive elements and a more detailed tutorial for first-time users. Enhancing accessibility features such as text-to-speech capabilities, better color contrasts, and comprehensive keyboard navigability was suggested. Some layout issues with the live code editor were noted. Overall some suggestions were also made to add more metadata into the systems to users know what each functionality is.

2) ***Content Evaluation****:* The use of RAG leveraging OpenDSA material was highly praised for generating dynamic, context-aware educational content. The LLM provided mostly correct and detailed answers, though responses to ambiguous prompts could be improved. The system's ability to reject queries outside its scope was commended, preventing the presentation of false information. The integration of verified sources for each response added a layer of trust. Recommendations included integrating a memory component in the LLM assistant and fine-tuning the balance between using OpenDSA and external knowledge bases.

Experts liked the idea that students would know where the information is sourced from. Guardrails effectively prevented the generation of misleading or inappropriate content, maintaining ethical standards in educational guidance. The LLM assistant provided accurate and relevant responses, aiding in the understanding of complex DSA concepts without giving direct answers, thus encouraging critical thinking.

Notes were made about incorporating a confidence score associated with the sources to establish confidence in the answers. The tags and problem metadata were helpful, though there was a noted lack of a filter option. Experts found the answers from the LLMs to be generally credible, although there were instances where the LLMs generated answers that seemed out of context or without referring to OpenDSA.

The comprehensive set of DSA problems, ranging from basic to advanced levels, equipped with a live code editor, was commended. The live code editor's support for multiple programming languages was seen as beneficial, providing a real-world coding experience. The interactive environment, with instant feedback on code compilation and execution, enhanced learning by allowing students to learn from mistakes in real-time.

One user expert noted that adding the context of the

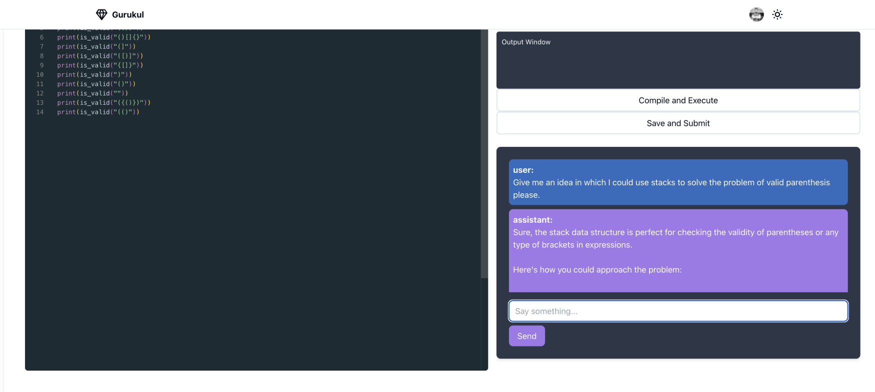

Fig. 3. User Interface : Practice section with Code Editor and LLM assistant

problem statement at hand in the LLM embedded in the practice section would help students stay on track when asking questions to the LLM.

*3) **Persuasion**:* The content's alignment with current DSA curricula was deemed highly beneficial for learners, demonstrating the platform's relevance and potential impact in educational settings. The User Expert Review highlighted the value of integrating the Gurukul platform into classroom environments as a supplementary tool, where hands-on practice and immediate feedback could significantly enhance the learning experience.

Despite these occasional inconsistencies, the experts were persuaded by the prospect of using such LLM-assisted educational tools in the classroom. They recognized the potential of these tools to provide personalized, context-aware assistance that traditional methods often lack, thus enhancing student engagement and comprehension.

This positive feedback from experts underscores the promise of LLM-assisted educational tools like Gurukul. The insights gained from the review will contribute to the continued development and refinement of these tools, ensuring they meet the high standards required for effective integration into educational curricula.

B. User Study

The User Study involved a diverse group of students enrolled in DSA courses. The study aimed to gather quantitative and qualitative data on user engagement, learning outcomes, and overall satisfaction with the Gurukul platform.

*1) **Quantitative Evaluation**:* Quantitative data was collected on user engagement metrics, such as the number of submissions, success rates, and average pass rates.

Data Collection and Questions Asked

Quantitative data was collected on user engagement metrics, such as the number of submissions, success rates, and average pass rates. Some data was also gathered through a structured questionnaire. By analyzing the ratings and responses, we can understand the overall user satisfaction and identify specific areas that need improvement.

This evaluation helps answer key questions about the platform's effectiveness in supporting students' learning experiences. It highlights the strengths of the platform, such as the positive feedback on enjoyability and usefulness, while also pointing out areas for enhancement, such as ease of use and overall quality. By addressing these insights, the Gurukul platform can be further refined to better meet the diverse needs of its users and enhance the overall learning experience.

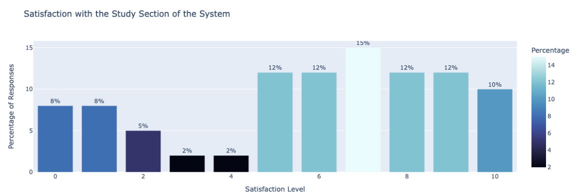

Fig. 4. Responses to the interview question "On a scale of 1-10 what do you think is the overall satisfaction with the study section?"

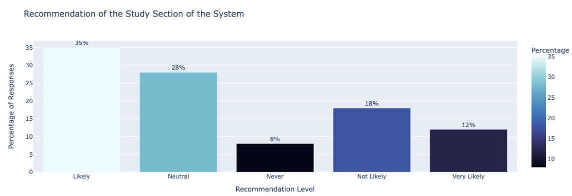

Fig. 5. Responses to the interview question "On a scale of 1-10 how much would you recommend the study section to other users?"

The survey results for the study section of the system showed the following average ratings, each based on a scale from 1 to 10, with 10 being the highest possible rating. Figure 4 shows the overall satisfaction levels with the study section, indicating a generally positive response from users. Figure 5 presents the distribution of recommendations for the study section, highlighting its perceived value among users. The average satisfaction level with the study section was

TABLE I
QUANTITATIVE ANALYSIS: LLM-RELATED FEEDBACK QUESTIONS

Question	**Very Well (%)**	**Well (%)**	**Neutral (%)**	**Not Well (%)**	**Bad (%)**
Address learning needs	27.5	30.0	27.5	10.0	5.0
Match learning style, pace, preferences	27.5	37.5	30.0	0.0	5.0
Align with curriculum, syllabus	15.0	27.5	40.0	10.0	7.5
Prepare for practice section	15.0	47.5	30.0	2.5	5.0
Apply knowledge and skills	22.5	55.0	17.5	2.5	2.5
Recommend to others	12.5	35.0	27.5	17.5	7.5

5.90, indicating a moderate level of satisfaction among respondents. The usefulness of the study section for achieving learning goals received an average rating of 5.55, suggesting that users found it somewhat helpful but with room for improvement. The overall quality of the study section was rated at an average of 5.90, demonstrating a generally moderate perception of its effectiveness and quality. Additionally, the extent to which users felt they learned from the study section was rated at an average of 6.80, indicating that respondents believed the study section contributed fairly well to their learning. Overall, these results suggest that while the study section is considered moderately satisfactory and useful, there are opportunities for enhancements to better meet user expectations and learning objectives.

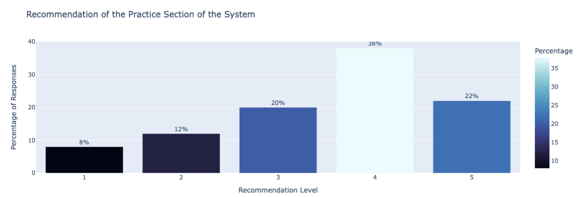

Fig. 6. Responses to the user study: "On a scale of 1-10 how much would you recommend the practice section to other users?"

The survey results for the practice section of the system showed the following average ratings, each based on a scale from 1 to 10, with 10 being the highest possible rating. The average satisfaction level with the practice section was 7.25, indicating that most respondents were quite satisfied with this component of the system. The ease or difficulty of using the practice section received an average rating of 3.58, suggesting that users generally found it somewhat easy to navigate and use. The enjoyment or frustration experienced while using the practice section also averaged 3.58, which implies that users had a moderately positive experience. Finally, the usefulness of the practice section for achieving learning goals was rated at an average of 6.94, demonstrating that respondents found it to be fairly effective in helping them meet their learning objectives. Overall, these results indicate a generally positive perception of the practice section, with room for improvements in user experience and usability.

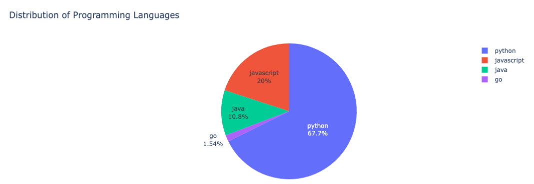

Fig. 7. Distribution of Programming Languages used.

Fig 7 shows the distribution of the percentage of programming language submissions.

*2) **Qualitative Evaluation**:* Students provided feedback on various aspects of the platform, including ease of use, helpfulness of the LLM-generated content, and overall learning experience.

Study Section Feedback The study section of the Gurukul app, utilizing RAG, received diverse feedback from users. A majority of students appreciated the comprehensive explanations and tailored learning resources provided by RAG, which helped deepen their understanding of theoretical concepts. The ability of the system to reference credible sources added a layer of trust to the learning process, though some users felt references were overused, especially when the AI seemed unsure of the answers. 24 students responded positively towards the educational value of RAG, emphasizing its effectiveness in delivering theoretical knowledge and facilitating access to a broad spectrum of learning resources. Users appreciated Gurukul's focus on specific DSA topics, which provided a more targeted learning experience compared to broader AI systems like ChatGPT. The platform's clean and engaging design received positive feedback, though there were calls for a more structured learning approach, integrating direct instructional content alongside the interactive AI-driven query system. Feedback on the platform's impact on motivation and confidence varied. Some users reported a boost in both areas, praising the immediate and interactive feedback that helped refine their problem-solving strategies. Others saw no noticeable change in their learning drive or self-assurance. There was a consistent desire for more content depth and a greater variety of problem types. Many users wished for more challenging questions to cater to different skill levels and foster a comprehensive understanding of the field.

Practice Section Feedback The practice section received both positive and negative feedback. Users appreciated the AI integration, which allowed them to ask questions and receive help while practicing. The flexibility to ask questions repeatedly and get responses aiding problem-solving was highly valued. The variety and relevance of problems were well-received, with users finding them similar to those on platforms like Leetcode. However, some users found the AI limited in its discussion capabilities and noted frequent conversation restarts. The coding area was criticized for being small, and the compilation process for causing annoying popups. Users wished for expected output for test cases to better understand code failures and clearer feedback on debugging, including syntax error highlighting. The practice section generally met user expectations, but feedback highlighted areas for improvement. Users wanted better UI, including adjustable text boxes, resizable windows, and improved dark mode readability. More helpful and clear feedback on code compilation and testing was requested, along with features for sorting problems by difficulty or topic and viewing all problems under a specific topic at once. Users also desired customizable and personalized practice options. In summary, while the Gurukul platform effectively facilitated learning and engagement in DSA courses, the feedback underscores the importance of balancing AI capabilities with user needs. Enhancements in interactive and comprehensive lesson formats, content diversity, and UI improvements could further elevate the platform's educational value.

V. Discussions

The overall evaluation data shows positive reception of the Gurukul platform. High success and pass rates indicate effective learning of DSA concepts. Users valued the LLM's clear explanations. The integration of guardrails and RAG improved engagement and learning productivity, contributing to the platform's educational effectiveness.

Engagement and Satisfaction The total number of submissions per user and the frequency of submissions over time demonstrated varied engagement levels, suggesting that the platform was able to cater to different learning preferences and needs. Users frequently engaged with the platform, particularly around assignment deadlines and exam preparations, indicating its relevance and utility in supporting their academic activities.

Learning Comprehension The success rates and average pass rate provide quantitative evidence of effective learning outcomes. Additionally, the thematic analysis of user feedback identified several key areas where the platform excelled, such as concept clarification and detailed guidance. Users particularly valued the interactive learning components, including the live code editor and practice questions, which facilitated a hands-on approach to learning.

User Feedback Qualitative feedback further reinforced the positive user perceptions. Users highlighted the effectiveness of the LLM in providing accurate and relevant responses, which aided in their understanding of DSA concepts. However, there were also suggestions for improvement, such as increasing the variety of practice questions and offering more detailed explanations for complex topics. Addressing these areas could further enhance user satisfaction and learning outcomes.

Effectiveness of Guardrails The guardrails effectively prevented the generation of misleading or inappropriate content, ensuring that users received accurate and ethically aligned educational guidance. This is particularly important in maintaining trust and reliability in AI-driven educational tools. The expert reviews corroborated these findings, with positive evaluations of the guardrails' ability to provide safe and relevant responses.

Role of Retrieval Augmented Generation The RAG system enhanced the LLM's capability by integrating external knowledge sources, providing users with more comprehensive and contextually relevant answers. This feature was particularly valuable in addressing complex queries and offering detailed guidance. The expert reviews and user feedback both highlighted the RAG system's effectiveness in enriching the learning experience.

VI. Conclusions and Future Work

The Gurukul platform has demonstrated significant potential in enhancing DSA learning, as evidenced by user studies and expert reviews. By integrating innovative concepts like RAG and Dynamic Guardrails, the platform paves the way for LLM-enabled educational tools, effectively bridging the gap between theoretical knowledge and practical application. The positive impact on student engagement, understanding, and motivation underscores the platform's strengths. To further improve, the platform should focus on expanding its knowledge base with additional authoritative sources and advancing guardrail technology for more nuanced guidance. Conducting longitudinal studies with a larger, diverse sample will provide deeper insights into the platform's long-term effectiveness. Additionally, integrating with popular Learning Management Systems will increase accessibility and streamline adoption in various educational settings. Gurukul represents a significant advancement in DSA education, offering a comprehensive solution that combines theory with practical application through innovative technologies. By continuously evolving to address user feedback and meet the needs of students and educators, the platform has the potential to become an indispensable tool in computer science education. Future work will focus on expanding capabilities, improving scalability, and maintaining an effective, engaging learning experience for all users.

REFERENCES

[1] A. Yadav, S. Gretter, S. Hambrusch, and P. Sands, "Expanding computer science education in schools: understanding teacher experiences and challenges," *Computer science education*, vol. 26, no. 4, pp. 235–254, 2016.

[2] S. Sentance and A. Csizmadia, "Computing in the curriculum: Challenges and strategies from a teacher's perspective," *Education and information technologies*, vol. 22, pp. 469–495, 2017.

[3] D. Dicheva, K. Irwin, and C. Dichev, "Oneup: Engaging students in a gamified data structures course," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pp. 386–392, 2019.

[4] OpenAI, "Chatgpt (mar 14 version)," 2023. Large language model.

[5] Anthropic, "Claude: Large language model by anthropic," 2024. Accessed: 2024-08-18.

[6] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models." https://ai.facebook.com/blog/large-language-model-llama-meta-ai/, 2023. Accessed: 2024-08-18.

[7] A. E. Wolf, "The plagiarism question of ai: How teachers have responded to llms in the classroom," 2024.

[8] S. Balloccu, P. Schmidtová, M. Lango, and O. Dušek, "Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms," *arXiv preprint arXiv:2402.03927*, 2024.

[9] OpenAI, "Gpt-4 (specific version)," 2023. Large multimodal model.

[10] M. Abdullayeva and Z. M. Musayeva, "The impact of chat gpt on student's writing skills: An exploration of ai-assisted writing tools," in *International Conference of Education, Research and Innovation*, vol. 1, pp. 61–66, 2023.

[11] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, p. 102274, 2023.

[12] J. G. Meyer, R. J. Urbanowicz, P. C. Martin, K. O'Connor, R. Li, P.-C. Peng, T. J. Bright, N. Tatonetti, K. J. Won, G. Gonzalez-Hernandez, *et al.*, "Chatgpt and large language models in academia: opportunities and challenges," *BioData Mining*, vol. 16, no. 1, p. 20, 2023.

[13] W. Gan, Z. Qi, J. Wu, and J. C.-W. Lin, "Large language models in education: Vision and opportunities," in *2023 IEEE International Conference on Big Data (BigData)*, pp. 4776–4785, IEEE, 2023.

[14] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, "Challenges and applications of large language models," *arXiv preprint arXiv:2307.10169*, 2023.

[15] Google, "Google trends - explore data for "ai in education"," 2023.

[16] M. J. Timms, "Letting artificial intelligence in education out of the box: educational cobots and smart classrooms," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 2, pp. 701–712, 2016.

[17] V. Devedžić, "Web intelligence and artificial intelligence in education," *Educational Technology & Society*, vol. 7, no. 4, pp. 29–39, 2004.

[18] M. Webb, T. Bell, N. Davis, Y. J. Katz, N. Reynolds, D. P. Chambers, M. M. Sysło, A. Fluck, M. Cox, C. Angeli, *et al.*, "Computer science in the school curriculum: Issues and challenges," in *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing: 11th IFIP TC 3 World Conference on Computers in Education, WCCE 2017, Dublin, Ireland, July 3-6, 2017, Revised Selected Papers 11*, pp. 421–431, Springer, 2017.

[19] J. Kay, M. Barg, A. Fekete, T. Greening, O. Hollands, J. H. Kingston, and K. Crawford, "Problem-based learning for foundation computer science courses," *Computer Science Education*, vol. 10, no. 2, pp. 109–128, 2000.

[20] R. Tamassia, "Implementing algorithms and data structures: An educational and research perspective: (invited presentation)," in *International Symposium on Algorithms and Computation*, pp. 4–8, Springer, 1998.

[21] W. Steingartner, J. Eged, D. Radaković, and V. Novitzká, "Some innovations of teaching the course on data structures and algorithms," in *2019 IEEE 15th International Scientific Conference on Informatics*, pp. 000389–000396, IEEE, 2019.

[22] S. Su, E. Zhang, P. Denny, and N. Giacaman, "A game-based approach for teaching algorithms and data structures using visualizations," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pp. 1128–1134, 2021.

[23] Y. Tang, Z. Xu, and Q. Li, "Probe into e-education of "algorithm and data structure"," in *Information Engineering and Applications: International Conference on Information Engineering and Applications (IEA 2011)*, pp. 1576–1581, Springer, 2012.

[24] S. MacNeil, A. Tran, J. Leinonen, P. Denny, J. Kim, A. Hellas, S. Bernstein, and S. Sarsa, "Automatically generating cs learning materials with large language models," *arXiv preprint arXiv:2212.05113*, 2022.

[25] T. Krüger and M. Gref, "Performance of large language models in a computer science degree program," in *European Conference on Artificial Intelligence*, pp. 409–424, Springer, 2023.

[26] M. Tran, "Prompt engineering for large language models to support k-8 computer science teachers in creating culturally responsive projects," in *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 2*, pp. 110–112, 2023.

[27] V. A. Hamaniuk, "The potential of large language models in language education," *Educational Dimension*, vol. 5, pp. 208–210, 2021.

[28] S. MacNeil, J. Kim, J. Leinonen, P. Denny, S. Bernstein, B. A. Becker, M. Wermelinger, A. Hellas, A. Tran, S. Sarsa, *et al.*, "The implications of large language models for cs teachers and students.," in *SIGCSE (2)*, p. 1255, 2023.

[29] S. Joseph, "Large language model-based tools in language teaching to develop critical thinking and sustainable cognitive structures.," *Rupkatha Journal on Interdisciplinary Studies in Humanities*, vol. 15, no. 4, 2023.

[30] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," *ArXiv*, vol. abs/2305.06983, 2023.

[31] K. Wang, J. Ramos, and R. Lawrence, "Chated: A chatbot leveraging chatgpt for an enhanced learning experience in higher education," *arXiv preprint arXiv:2401.00052*, 2023.

[32] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, "A survey on retrieval-augmented text generation," *ArXiv*, vol. abs/2202.01110, 2022.

[33] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen, "Generation-augmented retrieval for open-domain question answering," pp. 4089–4100, 2020.

[34] S. S. Manathunga and Y. A. Illangasekara, "Retrieval augmented generation and representative vector summarization for large unstructured textual data in medical education," *ArXiv*, vol. abs/2308.00479, 2023.

[35] M. Liffiton, B. E. Sheese, J. Savelka, and P. Denny, "Codehelp: Using large language models with guardrails for scalable support in programming classes," in *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, pp. 1–11, 2023.

[36] A. Y. Sun, V. Nair, E. Schumacher, and A. Kannan, "Conscendi: A contrastive and scenario-guided distillation approach to guardrail models for virtual assistants," *arXiv preprint arXiv:2304.14364*, 2023.

[37] T. Rebedea, R. Dinu, M. N. Sreedhar, C. Parisien, and J. Cohen, "Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails," pp. 431–445, 2023.

[38] Microsoft, "Monaco editor: The code editor that powers visual studio code," 2023. Available at: https://microsoft.github.io/monaco-editor/.