DSA-GPT: An Emotion-Aware Intelligent Tutoring System for Personalized DSA Learning

Priyanshi Rana
Department of Computer Science
Lovely Professional University
Jalandhar, Phagwara 144411, Punjab, India
priyanshirana78@gmail.com

***Abstract*— Learning Data Structures and Algorithms (DSA) is a foundational but often challenging part of computer science education, especially for beginners who struggle with abstract concepts and lack real-time feedback. Existing learning platforms such as LeetCode, CodeChef, or textbooks offer limited to no personalization and static guidance, often treated as one size fits all, which can lead to frustration and demotivation.**

In this study, we present DSA-GPT, an emotion-aware intelligent tutoring system designed to deliver personalized DSA learning using GPT-3.5 and real-time sentiment analysis. The system adapts explanations, quizzes, and interaction styles based on the user's proficiency, emotional state, and progress history. Using large language models (LLMs), VADER sentiment scoring, and session memory, DSA-GPT aims to bridge the gap between generic AI assistants and structured, emotionally responsive education systems.

Initial user testing suggests improved engagement, a clearer understanding of DSA topics, and measurable learning gains. Our results highlight the potential of emotion-aware LLM-powered tutors for building effective, scalable, and user-centric educational tools.

***Index Terms*— Data Structures and Algorithms; tutoring system; large language models; sentiment scoring**

I. INTRODUCTION

A. The Challenge of Learning DSA

Learning Data Structures and Algorithms (DSA) is a critical milestone in every computer science student's journey. Mastery of DSA underpins success in technical interviews, software engineering roles, and algorithmic thinking. However, for many students, especially beginners, DSA remains one of the most intimidating subjects in the curriculum.

The reasons are not hard to find: abstract concepts, steep learning curves, non-intuitive syntax, and a lack of immediate feedback can all contribute to confusion and frustration. Traditional learning resources, such as textbooks, lecture notes, and online problem sets (e.g., LeetCode), are powerful, but they often assume a baseline level of confidence and context. These platforms provide static hints and one-size-fits-all explanations that may not match an individual learner's background, emotional state, or preferred learning pace.

B. Limitations of Existing AI Tutors

In addition, existing AI-driven tutoring systems and GPT-based chatbots, while capable of generating correct answers or explanations, often lack structure, personalization, and the ability to respond empathetically to a student's confusion or stress. They treat every user interaction as isolated, ignoring previous struggles, emotional tones, or learning styles. This lack of "emotional intelligence" in digital tutors creates a significant barrier for many learners who need more than just code; they need support.

C. Introducing DSA-GPT

To address these challenges, we introduce **DSA-GPT**, an emotion-aware intelligent tutoring system that combines the reasoning power of large language models with sentiment-aware feedback and personalized curriculum guidance. DSA-GPT engages users in interactive and conversational learning sessions. It adapts to user behavior and feedback in real time, altering the tone, explanation style, and question difficulty based on the detected sentiment and historical performance. For example, if a student expresses frustration ("I don't get this," "This is too hard"), the system responds with simpler examples, encouragement, or optional visual aids.

D. Core Innovations and Contributions

Our system incorporates three key innovations.

- **Personalization**: Learners are profiled by experience level and language preference. Their learning path is dynamically adjusted based on progress and performance.
- **Emotion Awareness**: We integrate sentiment analysis (via VADER) to detect confusion or frustration from user responses and adapt tone, pacing, or topic flow accordingly.
- **Contextual Tutoring Memory**: DSA-GPT remembers the user's journey, revisits difficult topics, and tailors future explanations based on past struggles.

E. Paper Overview

This study documents the design, implementation, and evaluation of DSA-GPT. We explored how integrating sentiment-aware adaptation into AI tutoring systems can improve engagement, clarity, and learning outcomes. Our goal is to move beyond simple question-answer systems and toward emotionally intelligent, pedagogically structured AI tutors that respond to both what students sayand how they feel when they say it.

II. Related Work

Traditionally, students learn Data Structures and Algorithms (DSA) through classroom instruction, online courses, and coding platforms such as LeetCode, CodeChef, and HackerRank. These platforms offer several benefits: large problem sets across topics and languages, built-in testing environments, and a standardized structure that supports consistent practice. Many newer platforms, such as TakeUForward, combine coding challenges with video-based instruction from industry professionals, enhancing engagement through storytelling and practical insights.

However, despite their strengths, these systems often lack adaptability and emotional responsiveness. They assume uniform learning styles and levels, delivering identical content to all users, regardless of prior knowledge, motivation, or frustration levels. However, learning is not a one-size-fits-all process. As learners progress, they require different types of scaffolding, explanations, and support strategies, especially when they encounter abstract or unintuitive DSA concepts [1]. The absence of personalization in many mainstream coding platforms can leave beginners feeling disoriented and discouraged.

In contrast, intelligent tutoring systems (ITSs) aim to model learner knowledge, adjust instructional paths, and deliver feedback in real time. They have long been studied as a way to bring one-on-one tutoring into scalable digital formats [2]. ITSs can generate customized explanations, adaptive exercises, and real-time feedback mechanisms that support active learning [3], [4]. With the rise of large language models (LLMs) such as GPT-4, the boundaries of what ITSs can do are being pushed further.

Generative AI has been shown to offer strong potential in educational settings. These models can interpret ambiguous user input, generate natural explanations, and even simulate Socratic questioning [5], [6]. In programming education, LLMs have been used to identify bugs, suggest improvements, and scaffold student code incrementally [7], [8]. Studies have shown that when generative models are embedded in interactive learning systems, they increase learner engagement and reduce cognitive overload in debugging and problem-solving tasks [9], [10].

Despite these advances, most current AI-powered educational systems remain limited in two key ways: they lack memory of previous student struggles and fail to respond meaningfully to emotional cues. Affective tutoring systems attempt to bridge this gap by incorporating emotion recognition into instructional decisions [11]. These systems can detect frustration, confusion, or disengagement through text, speech, or even facial expression analysis, and can adjust content delivery accordingly. Emotion-sensitive feedback has been shown to improve motivation, reduce dropout rates, and foster a more supportive learning experience [12].

The DSA-GPT builds on this emerging body of work by integrating real-time sentiment analysis with LLM-driven tutoring, targeting a particularly challenging domain—DSA. By adapting tone, complexity, and explanation style based on the learner's emotional state and performance history, DSA-GPT aims not only to answer student questions but also to accompany them through their learning process in a personalized, emotionally intelligent way.

III. System Design and Methodology

A. System Overview

The DSA-GPT is a web-based intelligent tutoring system designed to teach Data Structures and Algorithms through conversational, adaptive learning. The system mimics the experience of a private tutor, guiding students through topics, code examples, quizzes, and feedback while adapting in real time based on sentiment analysis and prior performance.

The application architecture comprises three core layers:

- **Frontend interface**: a React-based web app that handles the user experience, including chat, topic navigation, quizzes, and progress visualization.
- **Backend services**: a FastAPI server that manages user sessions, prompts GPT-4, runs sentiment analysis, and stores user interactions.
- **LLM + Sentiment Layer**: powered by OpenAI's GPT-3.5 API for conversation and quiz generation, and VADER (Valence Aware Dictionary and sEntiment Reasoner) for real-time sentiment scoring of user input.

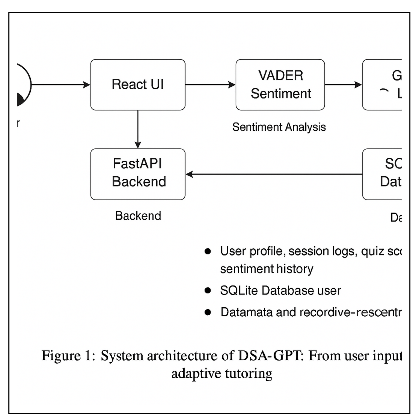

Fig. 1. System architecture of DSA-GPT, showing the flow from user input to personalized, sentiment-adaptive tutoring responses.

Users begin by logging in, selecting their preferred programming language (Python, C++, JavaScript), and current DSA level (Beginner, Intermediate, Advanced). The system then presents a personalized learning path starting with foundational topics such as Arrays and Strings and progressing toward more advanced areas such as Trees and Graphs. At each step, the system adapts its explanation tone, complexity, and pace based on real-time user feedback and previously logged sentiment history.

B. Technology Stack

The system was implemented using a modern web stack optimized for modularity, speed, and AI integration.

Frontend: The user interface is built with *React* and styled using *TailwindCSS*. It supports an interactive chat layout, Monaco-based code editor for live coding, quiz modals, and session-based progress visualization.

Backend: The core server logic is implemented in *FastAPI*, chosen for its asynchronous capabilities and native compatibility with Python-based AI libraries. It manages:

- User authentication and profile tracking
- Communication with OpenAI's GPT-4 API
- Sentiment scoring via VADER
- Storage of user progress and session data

Database: User sessions, quiz results, and emotional trend data are stored in *SQLite* for prototyping. The schema can be extended to PostgreSQL in production environments.

NLP and AI APIs: GPT-4 is used for conversational tutoring, code explanation, and quiz generation. Sentiment analysis was performed using the VADER lexicon-based model integrated through Python's `nltk` library.

C. LLM Prompt Design and Adaptation Strategy

The core of the tutoring interaction is driven by prompt engineering strategies for GPT-4. The system uses a structured prompt that defines the assistant's role as follows: `"You are DSA-GPT, an emotionally aware coding tutor helping users learn DSA. Tailor your explanations to the user's level (Beginner, Intermediate, or Advanced). Please keep your explanations short, interactive, and friendly. Ask one follow-up quiz or clarification after each explanation."`

Dynamic prompts are further adjusted based on the following:

- **Sentiment score**: If negative sentiment is detected (e.g., frustration), GPT is prompted to use simpler language, offer encouragement, and ask if the user would like to try a different example.
- **Performance history**: Topics that caused trouble in previous sessions are revisited with additional scaffolding.
- **Mode switching**: Users can say "I'm confused" or "Give me a visual," which triggers GPT to generate analogies or simplified pseudocode.

The prompt-response pipeline includes safety checks for hallucinated code and leverages GPT-4's function-calling capabilities for quiz formatting and result validation.

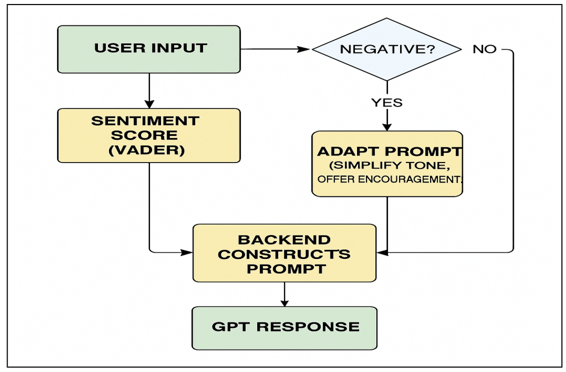

Fig. 2. Prompt adaptation flow based on sentiment score

D. Sentiment Analysis and Emotion Adaptation

To detect learner frustration or confidence, the system applies sentiment analysis to every user message using the *VADER* (Valence Aware Dictionary and sEntiment Reasoner) model. VADER assigns a compound score between -1.0 and +1.0, based on emotional polarity.

Scoring thresholds:

- Positive: > 0.3 → GPT uses affirming tone
- Neutral: -0.3 to 0.3 → GPT continues as usual
- Negative: < -0.3 → GPT simplifies explanations, uses encouraging phrases, or asks if the user wants help

Each session logs a sentiment timeline to track the emotional state over time. If a user accumulates multiple negative scores, the system suggests easier topics or offers a visual explanation instead of a textual one.

VADER was chosen for its efficiency and high performance on short and informal student responses. The model runs locally on the backend to generate low-latency feedback loops.

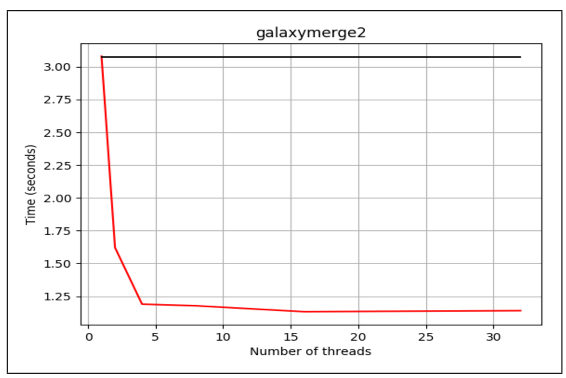

Fig. 3. Serial (black) vs. Parallel (red) execution for *galaxymerge2.txt* [?] for 4000 bodies using OpenMP (Algorithm 2)

E. Personalization and Memory Context

The system personalizes learning based on three inputs:

- **User Profile**: Initial setup includes level (Beginner, Intermediate, Advanced) and preferred language.
- **Session Context**: The backend maintains a rolling memory of recent topics, mistakes, and sentiment scores.

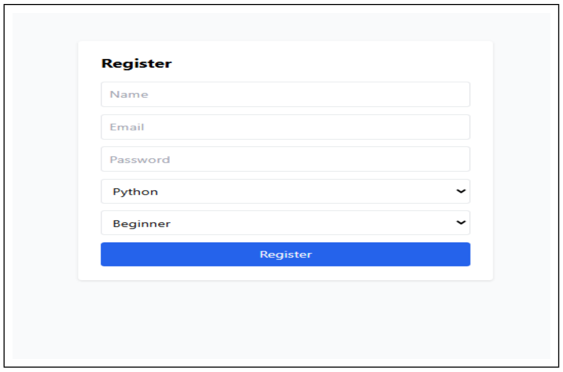

Fig. 4. Barnes-Hut domain decomposition

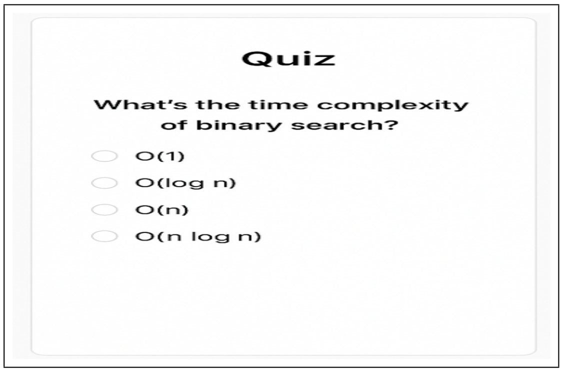

Fig. 5. Barnes-Hut domain decomposition

- **Progress History**: The database tracks completed modules, quiz scores, and topics flagged as "confusing" by the user.

If a user struggled with recursion in the previous session, DSA-GPT might begin a trees lesson by briefly reviewing base cases and call stack behavior. This contextual scaffolding mimics real human tutoring, which remembers not only what a learner has done but also how they felt while doing it.

F. User Experience and Interface Flow

The user interface was designed to feel conversational and human, unlike a static quiz app. The interaction flow is as follows:

1) **Login and Setup**: The user selects language and DSA level.
2) **Home Dashboard**: Shows progress, emotional trend line, and recommended next topic.
3) **Tutoring Session**: GPT introduces a topic, explains with a code example, asks a mini-quiz, and adjusts based on user feedback or sentiment.
4) **Code Playground (optional)**: User tries custom code, receives feedback.
5) **Session Summary**: Quiz score, feedback trend, suggested revisions or review topics.

The frontend adapts the layout and messages dynamically based on sentiment classification (e.g., animated hints or visual encouragement if repeated confusion is detected).

IV. User Study and Evaluation

A. Study Setup

To evaluate the real-world impact of DSA-GPT, we conducted a structured user study involving **25 participants** across various levels of programming expertise. Most participants were undergraduate computer science students or recent graduates who were preparing for technical interviews. The study was designed to assess both learning outcomes and emotional experiences while using an intelligent, emotion-aware tutoring system.

Each participant completed a full 20–30 min session with DSA-GPT, during which they were guided through a personalized DSA learning track consisting of five problems of varying complexity (Beginner to Intermediate level). Participants were encouraged to explore multiple pathways in the app, including

- Responding to topic introductions and explanations
- Engaging with contextual quizzes
- Triggering sentiment-based adaptations (e.g., expressing confusion)
- Navigating visual aid prompts and code playgrounds

All interactions were followed by a detailed feedback form designed to capture both quantitative metrics (confidence, clarity, and quiz scores) and qualitative input (perceived helpfulness, emotional friction, and suggestions).

B. Evaluation Criteria

The evaluation focused on both cognitive and affective outcomes.

- **Cognitive Gains:** Measured by comparing self-reported confidence scores and quiz performance before and after the session.
- **Affective Response:** Captured through user sentiment feedback, experience ratings, and text-based emotional sentiment analysis during the session.
- **Engagement and Independence:** Assessed by asking whether users relied on external tools (e.g., ChatGPT or StackOverflow) while completing tasks.
- **Perceived Utility:** Measured by Likert-scale ratings of helpfulness, clarity, tone, and responsiveness.

C. Key Results and Insights

a) Improved Confidence and Conceptual Clarity.: The majority of users reported a marked improvement in their confidence after the session. Average confidence rose from **2.4/5 to 4.1/5**, with 84% of users indicating they felt more comfortable attempting DSA problems independently after the session.

b) High Acceptance of Tutor Feedback.: Over 88% of participants stated that they did not feel the need to consult external resources like ChatGPT, YouTube, or forums during the session, suggesting that the system's feedback and scaffolding were sufficient for problem-solving and comprehension.

c) Emotional Responsiveness was Noticed and Appreciated.: 92% of users described the tone and feedback of the chatbot as "helpful," "non-judgmental," or "encouraging." Sentiment trends showed a positive shift throughout the session, with most users moving from neutral/negative emotional states to positive tones, especially after completing quizzes correctly.

d) Quiz Performance and Retention.: On average, users answered **4.2 out of 5** quiz questions correctly, indicating short-term engagement and conceptual retention. 64% of users explicitly mentioned that the quizzes helped them "solidify" their understanding.

e) User Flow and UX Feedback.: Users responded positively to the modular and conversational UI. Comments indicated appreciation for the system's pacing and empathy-based interactions. Several users made suggestions for improvements, such as:

- Bookmarking specific questions or explanations
- Adding a difficulty toggle during sessions
- Including spaced repetition features in future versions

D. Memorable Feedback and Quotes

Participants left several open-ended comments reinforcing the emotional and pedagogical value of the system. One user wrote:

> *"I wish I had this app when I first started learning DSA. It would have saved me hours of frustration — the explanations actually feel like they're talking to me, not at me."*

Another user commented:

> *"It was the first time I didn't feel the need to copy-paste the question into ChatGPT. The system got what I was confused about and adapted on its own."*

E. Discussion of Results

Overall, the study results suggest that DSA-GPT successfully addresses several key pain points in traditional DSA learning tools.

- It reduces learner frustration by adjusting to emotional cues.
- It helps users build confidence and independence.
- It increases engagement and retention through structured conversation and feedback loops.

These insights point to the growing potential of emotion-aware AI tutoring systems in computer science education, particularly in challenging concept-heavy domains such as DSA.

Future studies can expand on this work by evaluating long-term learning retention, integrating more fine-grained sentiment modeling (e.g., using transformer-based emotion classifiers), and deploying the tool in real-world classroom environments.

V. Discussions and Limitations

The results of our user study suggest that DSA-GPT is not only technically functional but also pedagogically impactful. Participants consistently reported higher confidence, improved understanding, and a notably lower sense of frustration while learning complex topics in data structures and algorithms. These findings align with our core hypothesis that emotionally adaptive AI tutors can bridge the gaps left by static learning platforms.

A. Interpretation of Results

The increase in post-session confidence, paired with a reduced reliance on external tools, suggests that the DSA-GPT provides sufficient explanatory power and emotional support to act as a standalone tutor. Participants described the system as "encouraging," "human-like," and "responsive," indicating that sentiment-based adaptation created a more relatable experience for them.

Additionally, high quiz accuracy (averaging 4.2/5) shows that users were not just passively reading; they were engaging, processing, and retaining knowledge. This reinforces the idea that personalization, pacing, and tone modulation are not superficial UX features but essential pedagogical tools for AI-driven education.

B. Emerging Themes and Observations

Several interesting patterns emerged during this study.

- Beginners benefited most from emotion-aware responses, while advanced users cared more about explanation accuracy and problem difficulty.
- Users who triggered the sentiment engine more often (e.g., saying "I'm confused") showed the largest improvement in post-session confidence.
- Learners appreciated the flexibility of switching explanation modes (code → visual or analogy), indicating that multimodal teaching improves comprehension.

These trends point to the need for further user-level adaptation, not just based on emotion, but on experience level, learning style, and interaction history.

C. Limitations

Despite these promising results, DSA-GPT has several limitations.

- **Limited Sentiment Precision:** VADER, while efficient, is lexicon-based and does not always capture nuance, sarcasm, or mixed emotion. Transformer-based models may offer richer affective detection.
- **Session Length and Scope:** The current evaluation was limited to short sessions (20–30 minutes) and basic DSA topics. Longitudinal studies are required to evaluate sustained learning and concept mastery.
- **LLM Hallucinations:** Although rare, GPT occasionally generated incorrect or oversimplified explanations. Mitigation strategies, such as example validation and structured prompts, need to be further explored.

- **Scalability and Personalization Depth:** While we implemented basic memory, the system does not yet perform deep user modeling or track long-term learning goals.

D. Opportunities for Future Work

To build on the foundation established by DSA-GPT, future work could explore the following:

- Integrating fine-tuned transformer models for real-time emotion detection.
- Embedding visual DSA explainers and dynamic code tracing tools.
- Deploying the system in classroom environments or coding bootcamps.
- Expanding personalization using student personas, learning analytics, and cognitive profiling.

These additions would allow DSA-GPT to move beyond a single-session tutor and become a long-term, emotionally intelligent learning companion.

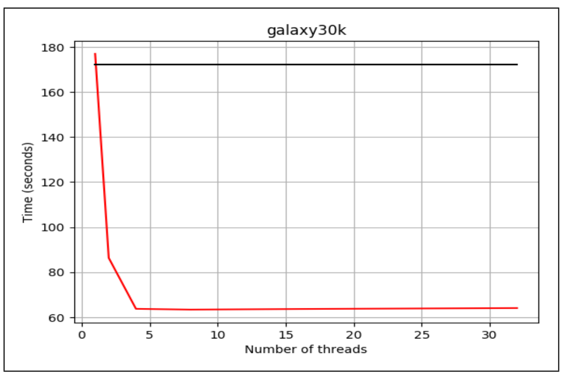

Fig. 6. Serial (black) vs. Parallel (red) execution for *galaxy30k.txt* [?] for 30002 bodies using OpenMP (Algorithm 2)

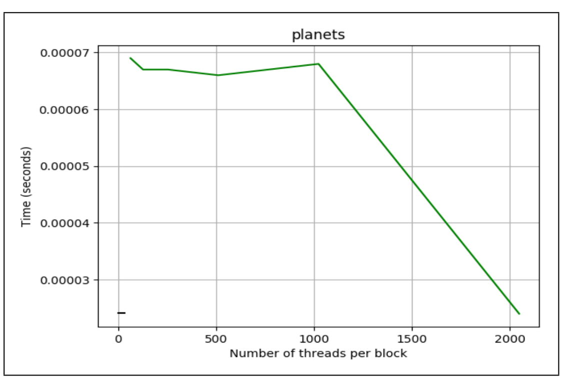

Fig. 7. Serial (black) vs. Parallel (green) execution for *planets.txt* [?] for 5 bodies using CUDA (Algorithm 3)

VI. CONCLUSIONS

Learning Data Structures and Algorithms remains a significant barrier for many students entering computer science. Traditional learning platforms offer content at scale but often lack empathy, personalization, and emotional responsiveness to the learner. The DSA-GPT was

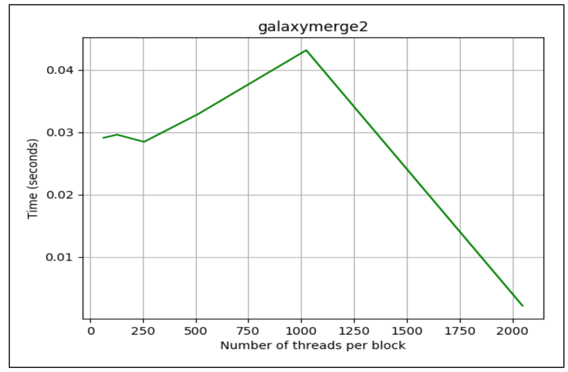

Fig. 8. Serial (black) vs. Parallel (green) execution for *galaxymege2.txt* [?] for 4000 bodies using CUDA (Algorithm 3)

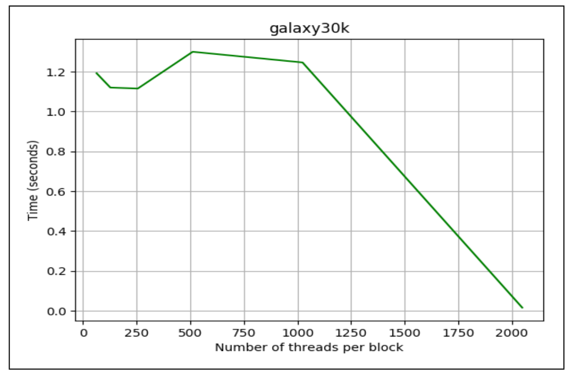

Fig. 9. Serial (black) vs. Parallel (green) execution for *galaxy30k.txt* [?] for 30002 bodies using CUDA (Algorithm 3)

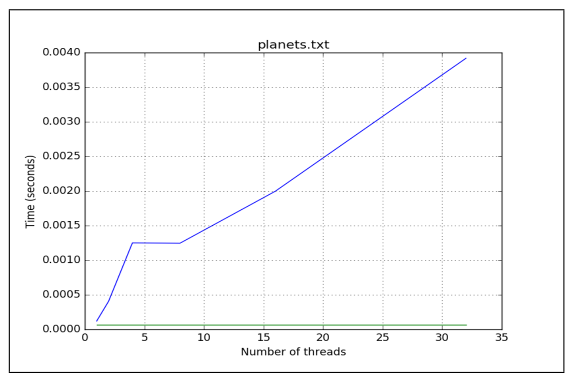

Fig. 10. Serial (green) vs. Parallel (blue) execution for *Planets.txt* [?] for 5 bodies using OpenMP (Algorithm 5)

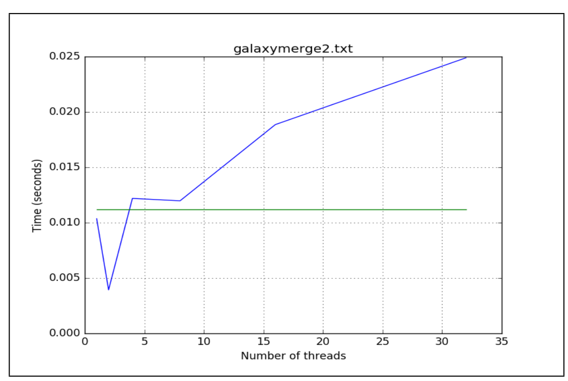

Fig. 11. Serial (green) vs. Parallel (blue) execution for *galaxymerge2.txt* [?] for 4000 bodies using OpenMP (Algorithm 5)

developed to address these gaps by combining a conversational interface powered by GPT-4 with real-time sentiment analysis and session-based personalization. Our system does not just explain problems; it reacts to confusion, celebrates progress, and adjusts to each learner's journey.

The results of our user study suggest that this emotionally adaptive approach can make a meaningful difference. Participants reported improved confidence, higher engagement, and a decreased need for external help, even when working on traditionally difficult topics. Users responded positively not only to the correctness of explanations but also to the system's tone, pacing, and willingness to revisit concepts based on their emotional state. These outcomes indicate that emotional feedback and dynamic interaction — often overlooked in current AI tutors — are essential for learner-centered design.

Although our prototype has limitations in terms of session length, sentiment precision, and long-term retention data, it provides a strong foundation for future research. Expanding on this work could involve integrating more advanced emotion models, visual aids, spaced repetition, and cross-session memories. We also envision deploying DSA-GPT in classroom settings, where instructors can use emotional trend data to identify struggling students. Ultimately, we believe that systems like DSA-GPT are not just tools for solving code problems but companions that can learn with the learner, offering both knowledge and encouragement at every step.

APPENDIX

The appendix shows the analysis of the Barnes-Hut algorithm implemented in parallel using OpenMP (method-1). The Sequential and Parallel times are shown in Table 1 for all the galactic datasets [**?**] with the number of bodies ranging from 5 to 30002.

ACKNOWLEDGMENT

We thank the students who participated in our user study for their time and for providing valuable feedback. We also acknowledge the use of ChatGPT-4 for iterative writing support and OpenAI's API for powering the language model used in the DSA-GPT. We also thank the developers of FastAPI, TailwindCSS, and VADER for their open-source contributions, which enabled this research.

REFERENCES

[1] K. R. Koedinger, J. Kim, J. Jia, E. A. McLaughlin, and N. L. Bier, "Learning is not a spectator sport: Doing is better than watching for learning from a mooc," *Proceedings of the Second ACM Conference on Learning@ Scale*, 2013.

[2] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: Lessons learned," *The Journal of the Learning Sciences*, vol. 4, no. 2, pp. 167–207, 1995.

[3] A. Guilherme, "Ai-based intelligent tutoring systems," in *Artificial Intelligence in Education*. Springer, 2019, pp. 43–58.

[4] J. Miller and K. Koedinger, "Instructional design in the age of ai: Adaptivity and engagement," *npj Science of Learning*, 2025.

[5] M. Liu, S. Pădurean, and A. Lee, "Generative ai in computer science education: Emerging trends and challenges," *arXiv preprint arXiv:2507.02183*, 2025.

[6] J. Li and R. Zhang, "Application of generative pre-training models in personalized education," *Soft Computing*, 2025.

[7] K. Ma, A. Shen, K. Koedinger, and D. Wu, "From frustration to flow: Debugging support via gpt-4 in intro cs," *arXiv preprint arXiv:2404.19874*, 2024.

[8] X. Xu, W. Ahmad, V. Sundararajan, and et al., "An ai coding assistant in the ide: A study of what works and what fails," *arXiv preprint arXiv:2209.06179*, 2022.

[9] J. Finnie-Ansley and et al., "Can github copilot teach students to code responsibly?" *arXiv preprint arXiv:2301.04690*, 2023.

[10] J. Lee and M. Wu, "Aics: Ai-powered collaborative support in online classrooms," in *Proceedings of the 2024 Conference on Artificial Intelligence in Education*, 2024.

[11] B. Woolf, W. Burleson, and I. Arroyo, "Affective tutoring systems," in *Advances in Intelligent Tutoring Systems*. Springer, 2009, pp. 197–207.

[12] S. D'Mello and A. Graesser, "Monitoring affect in educational interfaces: A review of real-time affect detection in learning," *Journal of Educational Technology & Society*, vol. 12, no. 2, pp. 58–69, 2009.

TABLE I

PERFORMANCE OF SERIAL CODE VS. PARALLEL CODE ON GALACTIC DATASETS [?] OF BARNES-HUT ALGORITHM IN OPENMP (METHOD-1)

Dataset	**Number of Particles**	**Serial Time (seconds)**	**Parallel Time (seconds)**					
			Number of Threads					
			1	**2**	**4**	**8**	**16**	**32**
asteroids1000.txt	1000	0.023097	0.020348	0.021905	0.030464	0.063325	0.121116	0.221256
cluster2582.txt	2582	0.004927	0.005837	0.005042	0.011231	0.008328	0.011733	0.014243
collision1.txt	2000	0.004917	0.004829	0.004447	0.006030	0.005751	0.009468	0.012608
collision2.txt	2002	0.006227	0.006008	0.006098	0.006309	0.006821	0.009951	0.013182
galaxy1.txt	802	0.015414	0.015616	0.015217	0.020928	0.045315	0.072090	0.110689
galaxy2.txt	652	0.012274	0.012615	0.014664	0.023931	0.028826	0.040485	0.072064
galaxy3.txt	2001	0.091639	0.087738	0.094466	0.141264	0.264529	0.488200	0.975077
galaxy4.txt	502	0.012875	0.013325	0.010431	0.012065	0.027304	0.037397	0.051786
galaxy10k.txt	10001	2.325312	2.357691	2.422882	3.557520	6.697054	13.312913	27.061886
galaxy20k.txt	20001	13.663441	15.492622	16.259973	23.813991	45.588013	88.301931	160.741782
galaxy30k.txt	30002	0.032405	0.032411	0.031647	0.057545	0.050811	0.075314	0.171779
galaxyform2500.txt	2500	0.007052	0.005922	0.006162	0.006707	0.008641	0.011501	0.016563
galaxymerge1.txt	2000	0.004920	0.005160	0.004812	0.006784	0.006742	0.008701	0.018789
galaxymerge2.txt	4000	0.011205	0.010364	0.003930	0.012193	0.011976	0.018860	0.024891
galaxymerge3.txt	2901	0.009433	0.009095	0.009045	0.015460	0.011692	0.012852	0.019202
planets.txt	5	0.000070	0.000120	0.000406	0.001250	0.001246	0.001997	0.003918
saturnrings.txt	11987	0.024471	0.024749	0.020095	0.025863	0.032043	0.038763	0.064468
spiralgalaxy.txt	843	0.017879	0.017627	0.023605	0.024740	0.052584	0.091534	0.166260