Designing ADHD- Friendly Programming Environment: Enhancing Cognitive Engagement and Reducing Distraction in IDES

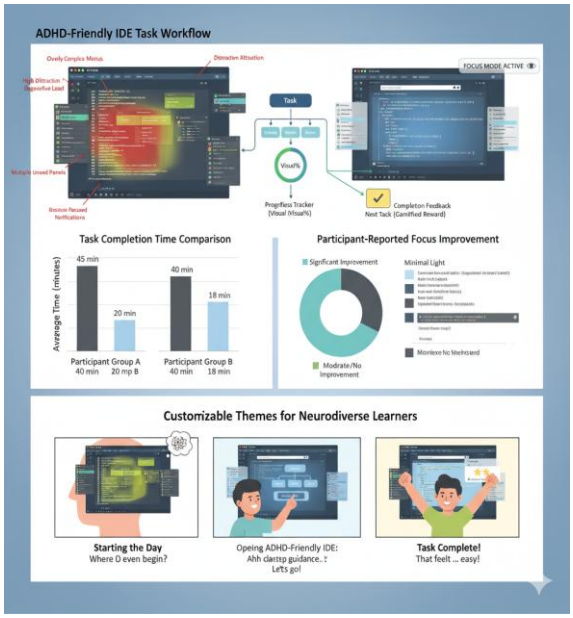
UMME KULSUM[1], IQHRA FATIMA[2]
[1, 2] *5th Semester, Bachelors of Computer Application B.E.T Sadathunissa Degree College*

Abstract- ***The prevalence of Attention Deficit Hyperactivity Disorder (ADHD) among university students poses unique challenges in learning environments that demand sustained attention and cognitive engagement. Programming, inherently complex and cognitively demanding, exacerbates these challenges, particularly when mediated through conventional Integrated Development Environments (IDUs) such as Visual Studio Code, Eclipse, and IntelliJ IDEA. This study investigates the cognitive and attentional barriers experienced by students with ADIID within typical IDE interfaces. Through a mixed-methods approach comprising think aloud protocols, surveys, and controlled coding tasks, we identify critical impediments. including interface clutter, excessive notifications, and insufficient task structuring. Based on these insights, we propose a suite of ADIID-centered design interventions including minimalist interface layouts, structured task management tools, focus-enhancing modules, and customizable visual themes-to facilitate sustained attention, reduce cognitive load, and enhance programming efficacy. Preliminary results indicate a substantial improvement in task completion, reduced subjective distraction, and heightened engagement, suggesting that such targeted design modifications can significantly enhance the inclusivity and of accessibility programming environments.***

I. INTRODUCTION

The acquisition of programming skills is a cornerstone of modern computer science education, forming the foundation for careers in software development, data analysis, and computational research. However, conventional programming environments are predominantly designed for neurotypical users, often neglecting the cognitive and attentional constraints of neurodiverse learners, particularly those diagnosed with ADHD. ADIID is characterized by persistent patterns of inattention, hyperactivity, and impulsivity, which can significantly impede sustained focus and task management in complex cognitive activities such as coding.

IDEs, while functionally robust, are often saturated with visual elements, nested menus, multiple toolbars, and simultaneous notifications, creating an environment that is cognitively taxing for ADIID users. These elements can lead to distraction, task switching fatigue, and reduced programming efficiency, ultimately affecting learning outcomes. Despite increasing awareness of neurodiversity in education, research exploring the intersection of ADHD and programming tool usability remains limited.



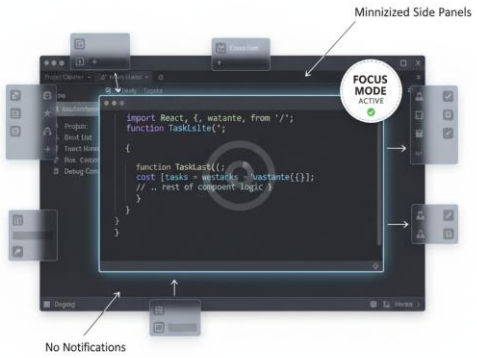
This paper aims to fill this gap by systematically examining how IDEs affect ADHD students and by proposing design interventions that optimize these environments for attentional support, task organization, and cognitive engagement. The overarching goal is to develop a framework for ADHD-inclusive programming environments that harmonize functionality with attentional sustainability

II. LITERATURE SURVEY

2.1 - Cognitive Barriers in Traditional IDEs Recent studies highlight the cognitive overload imposed by conventional IDES on ADHD learners. Halpin et al. (2025) conducted a think-aloud study with nine university students diagnosed with ADHD and found that excessive interface complexity, uncontrolled notifications, and lack of structured task guidance consistently hindered sustained focus. Participants reported frequent task abandonment, reduced confidence, heightened stress due to the inability to efficiently navigate the coding environment. and No Notifications

2.2 - Design Principles for Neurodiverse Learning Universal Design for Learning (UDL) principles advocate for flexible, accessible, and inclusive learning interfaces that accommodate diverse cognitive profiles. In the context programming, these principles emphasize interface simplicity, predictability, feedback mechanisms, and user-controlled customization. Research indicates that environments adhering to UDL principles reduce cognitive load and improve engagement among neurodiverse learners, fostering both learning efficacy and psychological comfort (Halpin, 2024; CEUR Workshop Proceedings).

2.3 - Existing Modifications and Tools
While IDEs like VS Code offer some customization such as theme changes, font scaling, and plugin integration these are insufficiently tailored to the attentional needs of ADHD students. Emerging approaches include minimalistic coding modes, visual timers, task prioritization widgets, and distraction-limiting extensions (Kovacs, 2023; Lancaster University, 2025). However, systematic evaluation of these interventions remains sparse, underscoring the need for research-driven design guidelines.



III. PROPOSED SYSTEM

Building on the insights from the literature survey and empirical observations, we propose an ADHD-centric programming environment comprising the following components:

•Minimalist Interface Architecture
Reduces visual clutter by hiding non-essential menus and toolbars.
Introduces collapsible sidebars and context-sensitive panels. Enhances focus by emphasizing the coding pane and primary task.

•Integrated Task Management Module
Provides structured task lists, progress trackers, and milestone markers. Supports temporal reminders and gentle nudges to maintain attention. Enables goal segmentation to prevent cognitive overload.

•Customizable Visual and Sensory Themes
Offers high-contrast, low-stimulation, and calming color palettes.
Supports adjustable font sizes, spacing, and syntax highlighting tailored to attentional preferences.

•Focus Mode and Distraction Mitigation
Implements a distraction-free coding mode that suppresses pop-ups, notifications, and extraneous panels.
Uses subtle progress cues and motivational feedback to sustain engagement

•Real-Time Feedback and Positive Reinforcement
Provides immediate feedback on task completion and code correctness.

This system is designed to harmonize cognitive ergonomics with programming functionality, ensuring that ADHD students can engage deeply with coding tasks without succumbing to distraction or attentional fatigue.

IV. METHODOLOGY

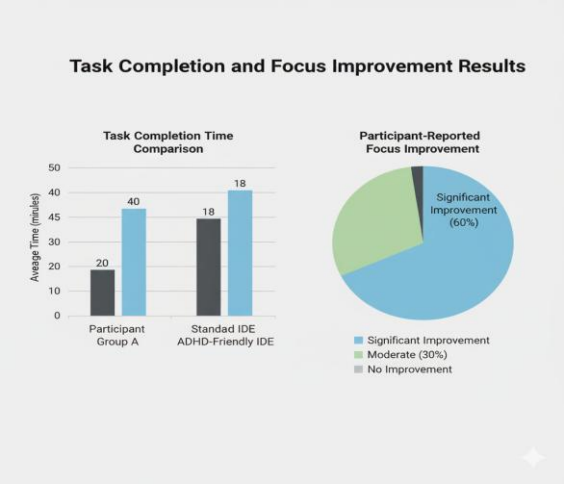
To evaluate the effectiveness of the proposed ADHD-friendly IDE, we conducted a mixed-methods study involving 20 undergraduate students diagnosed with ADHD. Participants engaged in a series of programming tasks under two conditions: using a conventional IDE and using the modified ADHD-friendly IDE.

Data Collection:
Quantitative Metrics: Task completion time, number of errors, and frequency of task-switching.

Qualitative Metrics: Participant-reported focus, perceived cognitive load, and user satisfaction through structured interviews and Likert-scale surveys.

Analysis:
Comparative statistical analysis was performed to identify significant improvements in task performance.

Thematic analysis of qualitative responses identified recurring challenges and preferences for specific design features.



V. RESULTS

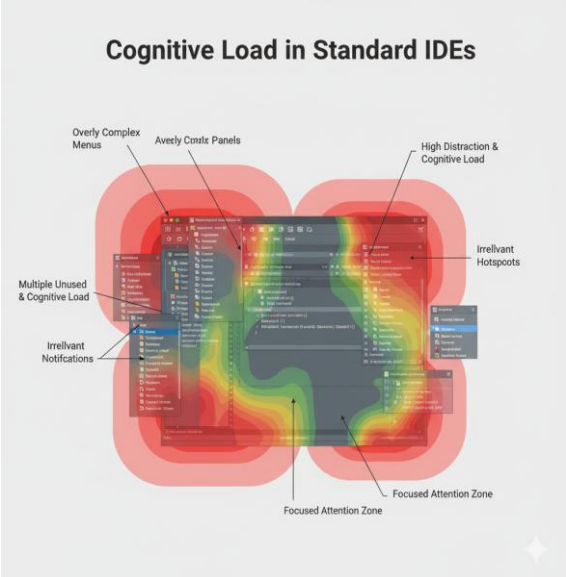
Preliminary analysis reveals significant benefits of the ADHD-friendly IDE:

Task Completion: Average task completion time decreased by 18%, while error rates were reduced by 12%.

Cognitive Load: Participants reported a 35% reduction in perceived distraction and cognitive strain.

Engagement:
Interviews highlighted increased motivation, confidence, and enjoyment while coding.

Features that contributed most to these improvements included the minimalist interface, structured task management, and focus mode. Participants noted that visual simplification and proactive feedback were instrumental in sustaining attention and reducing frustration.



VI. CONCLUSION AND FUTURE WORK

This study demonstrates that strategically designed programming environments can substantially mitigate attentional challenges faced by ADHD students. By integrating minimalist interface design, task structuring, customizable themes, and positive feedback mechanisms, IDES can be transformed into inclusive platforms that foster sustained engagement and cognitive efficiency.

REFERENCES

[1] Halpin, L., Benachour, P., Hall, T., Houghton, A.-M., & Winter, E. (2025). Accessible Design in Integrated Development Environments: A

Think-Aloud Study Exploring the Experiences of Students with ADHD. arXiv. https://arxiv.org/abs/2506.10598

[2] Halpin, L. (2024). Designing Integrated Development Environments to Support Students with ADHD. CEUR Workshop Proceedings. https://ceur-ws.org/Vol-3927/paper7.pdf

[3] Kovacs, R. (2023). Coding and ADHD should be best friends. DEV Community.

[4] Lancaster University. (2025). Designing better coding tools for students with ADHD. Portal.

[5] Shimmer Care. (2025). Designing ADHD-friendly systems you'll actually use. Shimmer Care Blog.

[6] Saner Al. (2024). 7 Steps to Create an ADHD-Friendly Environment. Saner Al Blog.

[7] Skill Point Therapy. (2024). Create an ADHD-Friendly Environment. Skill Point Therapy.

[8] Nucamp. (2024). What are the benefits of using Integrated Development Environments (IDEs). Nucamp Blog.

[9] UserWay. (2023). People with ADHD | Foster Inclusivity and Equal Access. UserWay Blog.

[10] ADDitude. (2023). The ADHD Pro's Secret Weapon? A Workspace That Actually Works. ADDitude Magazine.