iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70800

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Edu-Bot: A New Approach to Learning Python Effectively

Srisaila Nath (Asst. Prof)[1], M S Mohammed Zubair Darvesh[2], M. Akhil Sai[3], Ajay Kuche[4], Manishma Reddy M[5]
Dept. of Computer Science & Engg, Ballari Institute of Technology and Management, Ballari, India

Abstract: ***The evolution of programming education has highlighted the limitations of traditional learning methods, particularly in Python education. Edu-Bot is an interactive learning platform designed to bridge this gap by integrating AI-driven personalized learning, real-time coding environments, and adaptive video recommendations. Unlike existing platforms that primarily offer static content and generic exercises, Edu-Bot employs a chatbot powered by natural language processing (NLP) to provide real-time assistance, track learning progress, and tailor content based on user performance. Additionally, a face-tracking feature ensures user engagement by monitoring screen time and learning focus. The platform is built using Python (Flask), JavaScript, HTML, CSS, and MySQL, ensuring seamless interaction between the frontend and backend. This paper discusses the technical implementation of Edu-Bot, evaluates its impact through user engagement metrics, and compares it with existing learning platforms. The results indicate improved learning outcomes, increased retention rates, and enhanced user satisfaction, demonstrating the potential of AI-driven education in making Python learning more effective and interactive.***
Keywords: Artificial Intelligence, Edu bots, Online Tutors, Online Programming tools, Online Compilers.

I. INTRODUCTION

Python has emerged as one of the most popular programming languages, used in software development, artificial intelligence, and data science. Yet, most beginners find it difficult to learn Python because they do not have interactive and customized learning experiences. Conventional computer programming education is largely based on static text-based guidelines and traditional code editors, which do not often give immediate feedback, resulting in boredom and lack of practicality in applying theoretical principles to real-life situations. While tools such as Codecademy, Replit, and W3Schools have interactive coding environments, they are not backed by immediate help, adaptive learning features, and tracking of users' engagement. Students tend to be challenged when they try to learn complicated ideas without the benefit of immediate explanation, guided learning streams, and concentration-monitoring functionalities. Edu-Bot is designed to eliminate these challenges through a highly engaging and responsive platform for Python education. The platform combines artificial intelligence, live coding environments, and engagement-tracking functionality to boost the quality of learning. It features an AI-powered chatbot, built with Bot press, to offer real-time question answering, customized explanations, and interactive assistance. Edu-Bot also has a dynamic video suggestions system that recommends customized learning material based on user performance, providing an optimized and organized learning process. To increase engagement further, a face-tracking feature tracks user activity, promoting regular study routines and reducing distractions.

A built-in Python compiler enables students to code inside the platform without having to use third-party tools, providing an immersive learning environment. In contrast to current learning platforms, Edu-Bot offers not only coding exercises but also adaptive learning, real-time guidance, and activity tracking, improving the effectiveness and accessibility of Python education. Edu-Bot's effectiveness is measured by metrics on user engagement, reflecting gains in learning achievement and retention. The remainder of this paper is organized as follows: Section II provides related work, summarizing the shortfalls of the current solutions. Section III describes the proposed method, including the technical architecture, chatbot development, video recommendation algorithms, and engagement measurement mechanisms. Section IV reports experimental results, such as user feedback and performance assessments. Section V concludes the paper by providing an overview of key findings, limitations, and directions for future improvement.

II. RELATED WORK

The development of online learning platforms has revolutionized programming education to a great extent, offering learners interactive and accessible environments. Online platforms like Codecademy, W3Schools, and Replit provide coding exercises, reference guides, and real-time code execution, making programming more accessible to beginners. Nevertheless, these platforms lack personalized learning assistance, real-time feedback, and engagement-tracking features, which are essential for successful learning.

©IJRASET: All Rights are Reserved

Research has indicated that learners learn more from dynamic, adaptive learning environments than from static tutorials since they need instant feedback and guided instruction to learn programming principles effectively. Some researchers have examined the contribution of artificial intelligence in enhancing online learning. Labadze et al. (2023) investigated the effect of AI-driven chatbots on student learning, highlighting their potential to offer personalized support, enhance engagement, and promote retention. Likewise, El Azhari et al. (2023) discussed the application of AI-based tutors in e-learning, pointing out their capacity to offer real-time assistance and adaptive content suggestions. These works are in line with the fundamental goals of Edu-Bot, which incorporates an AI chatbot built with Bot press to offer immediate query resolution and customized explanations. Nevertheless, available chatbot-driven platforms.Yet another challenge noticed in present-day educational platforms is the absence of engagement monitoring. Goodson et al. (2015) highlighted the necessity of time management in online learning, postulating that students usually find it challenging to stay concentrated and consistent in learning independently. While Udemy and Coursera offer video-based learning, they do not actively track user study patterns or engagement. Edu-Bot, on the other hand, uses a face-tracking system to measure user attention and screen time, enabling learners to improve their study habits and maintain active engagement during the learning process. In addition, research conducted by Chung Kwan (2023) discussed the implementation of AI-powered chatbots in flipped learning settings and indicated that models based on chatbots can facilitate improved student understanding and problem-solving abilities. Still, most available platforms do not have an embedded coding environment that enables students to implement theoretical knowledge in real time. Edu-Bot fills this gap by including a Python compiler within the platform, which lets learners code without having to jump between different tools. Despite the advancements in AI-driven education, ethical considerations and bias in chatbot interactions remain critical concerns, as discussed by Okonkwo and Ade-Ibijola (2023). Ensuring unbiased and ethical AI assistance is essential for maintaining educational integrity. Edu-Bot is designed to mitigate these issues by implementing a transparent, structured chatbot response system that prioritizes factual accuracy and inclusivity in learning support.

Through the examination of current platforms and research studies, it is clear that although there has been considerable advancement in online programming education, there are still gaps in personalized support, real-time tracking of engagement, and integration of coding. Edu-Bot fills these gaps by integrating AI-powered query answering, adaptive video suggestions, real-time coding environments, and engagement tracking, offering a more interactive and efficient learning experience.

III. PROPOSED SYSTEM

Edu-Bot is an interactive learning platform powered by AI that aims to enhance Python programming education through a more engaging and effective approach. In contrast to static tutorial-based e-learning platforms, Edu-Bot combines real-time support. The system is modular, web-based, and makes user interaction effortless, data handling efficient, and learning adaptive. Edu-Bot has a frontend interface, written in HTML, CSS, and JavaScript, with a user-friendly interface that consists of text-based tutorials, chatbot support, an inbuilt Python compiler, and video suggestion. The backend utilizes Flask to facilitate user authentication, interaction with chatbots, and database operations, with MySQL retaining user profiles, learning history, chatbot history, and user engagement data for enabling the platform to learn in line with the progress of the user. A real-time answering facility to programming questions is ensured through the deployment of a chatbot using Botpress, with a face-tracking facility implemented utilizing JavaScript to make users concentrate when they are undertaking learning. One of the key features of Edu-Bot is its AI-driven chatbot, a virtual tutor that offers instant query resolution and personalized support. It is different from conventional FAQ-based bots in that it responds dynamically to user queries by studying learning patterns and providing context-aware explanations. Coupled with Natural Language Processing (NLP), the chatbot ensures that responses are accurate and context-specific. Moreover, it recommends additional learning resources, including video tutorials and documentation, depending on the learner's current level. This aspect greatly minimizes learning gaps and enables users to get instant assistance while learning. Edu-Bot also has a customized video suggestion system, which provides students with relevant educational material based on their learning status. The site uses the YouTube Data API to dynamically retrieve video tutorials that are relevant to the user's level of learning, minimizing the learner's effort in searching for additional content. Through ongoing monitoring of user interaction and learning behavior, Edu-Bot makes real-time adjustments to content suggestions, offering a smooth and customized learning process.

For coding practice, Edu-Bot incorporates a built-in Python compiler driven by Code Mirror that enables learners to write, run, and debug Python programs right within the platform. The compiler supports syntax highlighting, instant error detection, and autocomplete prompts that make learning easier and facilitate the identification of errors quickly. This inbuilt compiler, as opposed to most other online coding environments, eliminates the need to shift between separate tools, so learners can effectively hone their programming skills in an organized and time-efficient way.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

To ensure sustained user engagement, Edu-Bot incorporates a face-tracking capability with JavaScript and WebRTC, which tracks screen time and attention levels. If the system finds that there is no activity, it sends subtle reminders to get back on track, nudging learners to remain focused and effectively manage their study time. This capability ensures high engagement levels, which are typically difficult to achieve in self-paced learning settings. With the incorporation of AI-powered query answering, tailored content suggestions, in-browser coding sessions, and user engagement tracking, Edu-Bot offers a well-rounded learning environment. Through real-time guidance, adaptive learning journeys, direct hands-on practice, and persistent tracking of user engagement, Edu-Bot facilitates more effective overall education in Python programming. It is a cutting-edge learning software offering students a user-friendly, optimized, and stimulating means to acquire Python programming proficiency.

IV. LITERATURE SURVEY

Author Name	Description	Implementation	Remarks
Labadze et al. (2023) – [1]	Investigated the effect of AI-driven chatbots on student learning, highlighting their potential to offer personalized support, enhance engagement, and promote retention.	Implemented an AI-powered chatbot to assist students with queries in an online learning environment.	Found that chatbots improve learning efficiency but require fine-tuning for better contextual understanding.
El Azhari et al. (2023) – [2]	Discussed the application of AI-based tutors in e-learning, pointing out their capacity to offer real-time assistance and adaptive content suggestions.	Integrated an AI-driven tutor into an online education platform to provide instant feedback and personalized learning recommendations.	Demonstrated improved learner engagement but noted challenges in maintaining contextual accuracy in chatbot responses.
Goodson et al. (2015) – [3]	Highlighted the necessity of time management in online learning, postulating that students usually find it challenging to stay concentrated and consistent in learning independently.	Conducted an analysis on time management strategies in digital learning environments.	Found that engagement tracking tools can help students develop better study habits but require further optimization.
Chung Kwan (2023) – [4]	Discussed the implementation of AI-powered chatbots in flipped learning settings and indicated that models based on chatbots can facilitate improved student understanding and problem-solving abilities.	Developed an AI-based chatbot to support self-paced learning in a flipped classroom model.	Noted improvements in student performance but emphasized the need for better AI adaptability to different learning styles.
Yedjou, Clement G. et al. (2023) –(5)	Lead author, involved in outlining the advancements of AI in education and ethical concerns.	Helped frame the scope of AI integration in educational environments and potential outcomes.	Played a central role in emphasizing the societal and ethical balance of AI use.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

V. METHODOLOGY

Edu-Bot was built upon an organized methodology combining artificial intelligence, live coding capabilities, and user interaction monitoring to develop an interactive and efficient Python learning platform.

Conventional learning sites usually lack tailored advice, which complicates understanding intricate programming principles for newcomers. Edu-Bot resolves the shortcomings by merging AI-based support, adaptive content suggestions, and a dynamic coding interface. This part gives an overview of the system architecture, key functionalities, and implementation approaches utilized in developing Edu-Bot.

System Architecture:-Edu-Bot is a web application developed with HTML, CSS, JavaScript, Python (Flask), and MySQL to provide seamless user interaction and optimal data management. The system has three main layers:

Frontend Interface: Developed with HTML, CSS, and JavaScript, the frontend offers a friendly and user-oriented interface. With text-based guides, a Python compiler, AI chatbot, and video suggestions, learning becomes interactive and interactive.

Backend Processing : Backend is driven by Flask, with user authentication, processing of chatbot interactions, and request management. MySQL is used as the database to store user profiles, learning progress, and chatbot conversation logs in order to customize the learning experience.

AI and Engagement Monitoring : The chatbot, powered by Botpress, uses Natural Language Processing (NLP) to answer user questions in real-time. Also, a face-tracking system, using JavaScript and WebRTC, keeps track of user engagement, measuring how long they actively spend on learning.

AI-Powered Chatbot for Real-Time Support Edu-Bot incorporates an AI-powered chatbot built on Botpress that offers immediate query resolution and learning support.

As opposed to FAQ-based bots that are static, this chatbot is proactive and interacts with students by reading their queries and giving them responses in line with their needs. It assists users in understanding Python topics, debugging errors, and best practices in coding through the application of NLP and pre-set knowledge bases.

Personalized Video Recommendation System Rewards-Centric Recommender System of Edu-Bot guarantees that students are provided with appropriate Python tutorials relevant to their progress and learning history. Through usage of the YouTube Data API, the system derives and suggests educational videos aligned with the user's immediate learning requirements dynamically. Rather than dumping students with universal resources, the recommending engine delivers personalized content, enabling learners to expand upon their existing knowledge base effectively.

Integrated Python Compiler for Hands-On Practice:- Edu-Bot has an integrated Python compiler powered by Code Mirror, where learners can write code, test, and debug it in the platform itself.

Code Mirror improves learning compared to typical code editors through syntax coloring, real-time error identification, and auto-completion. This integration allows learners to test Python code with ease without the need to switch between different environments, facilitating learning through instant feedback.

Face-Tracking for Engagement Monitoring:- Edu-Bot presents a face-tracking feature to observe user attention and learning patterns. Built with JavaScript and WebRTC, the feature identifies whether or not the user is actively engaged with the screen, monitoring lesson time spent.

If the user is inactive for a long duration, the system provides subtle reminders to stay on track. This promotes self-regulation and effective time management to ensure learners remain dedicated to progress.

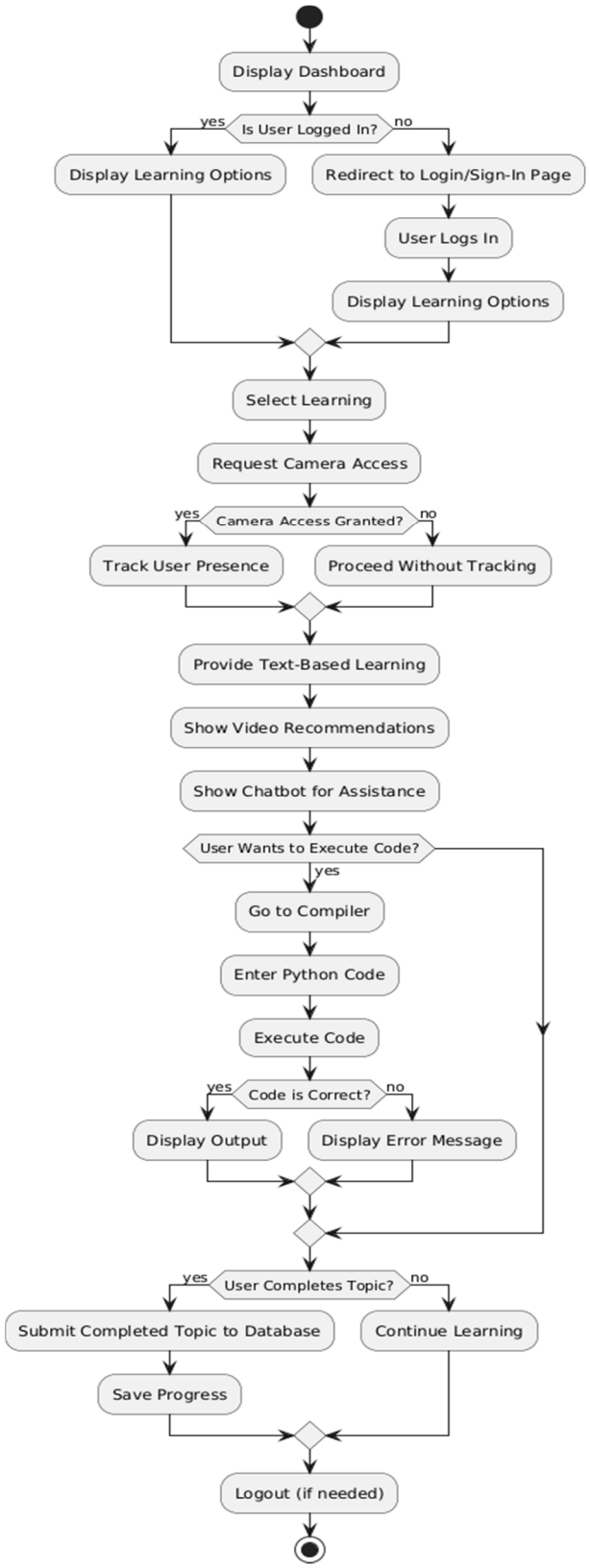
Data Storage and User Progress Tracking:- Edu-Bot uses MySQL to store and track user profiles, learning history, and engagement data. All activity of each user, such as completed tutorials, viewed videos, and coding exercises, is tracked to tailor their learning experience. This organized data storage allows Edu-Bot to dynamically change content suggestions and offer in-depth progress insights, allowing users to monitor and enhance their learning process.

Testing and Optimization:- For reliability and efficacy, Edu-Bot was thoroughly tested and optimized. Unit testing was done on major components such as the chatbot, video recommendation system, and Python compiler to ensure their functionality. User testing was carried out to determine usability, engagement levels, and learning efficacy. Testers' feedback was used to fine-tune system features, improve performance, and enhance the overall user experience.Fig. 1 Example of an unacceptable low-resolution image.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. FLOWCHART

Figure 1: Flowchart representing the learning process in Edu-Bot, from login to code execution and progress tracking.



VII. RESULT AND DISCUSSIONS

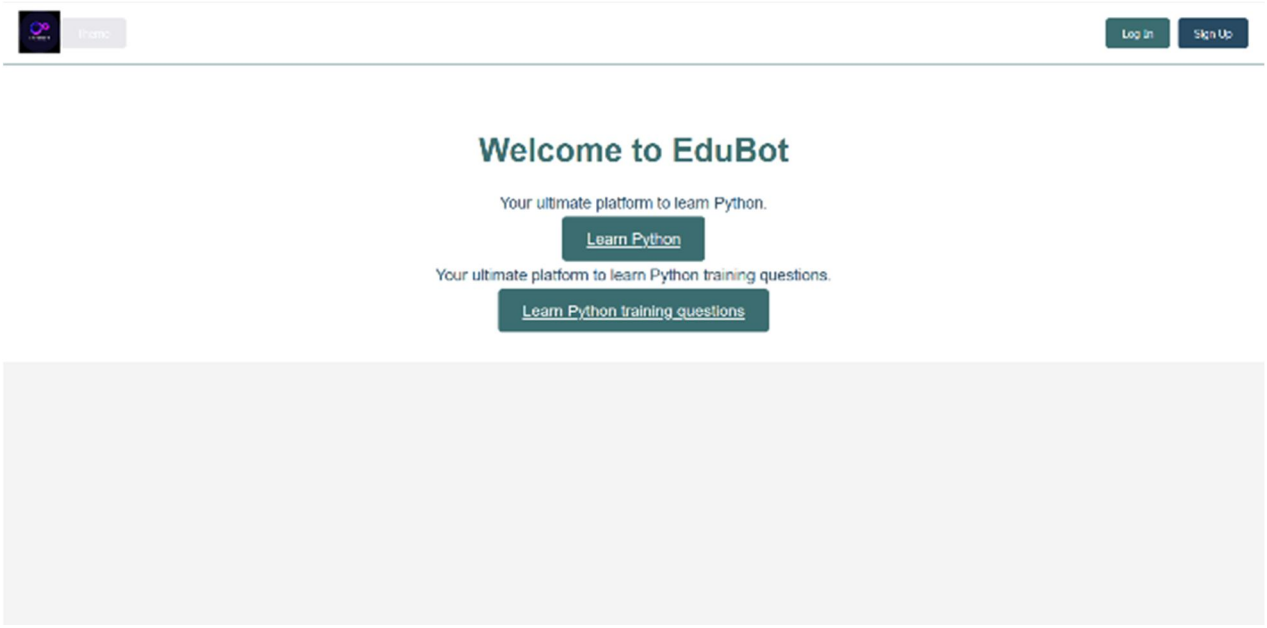


Figure 2 : Edu-Bot Home Page

The home page serves as the entry point for learners, providing easy navigation to Python tutorials, chatbot assistance, coding exercises, and video recommendations.

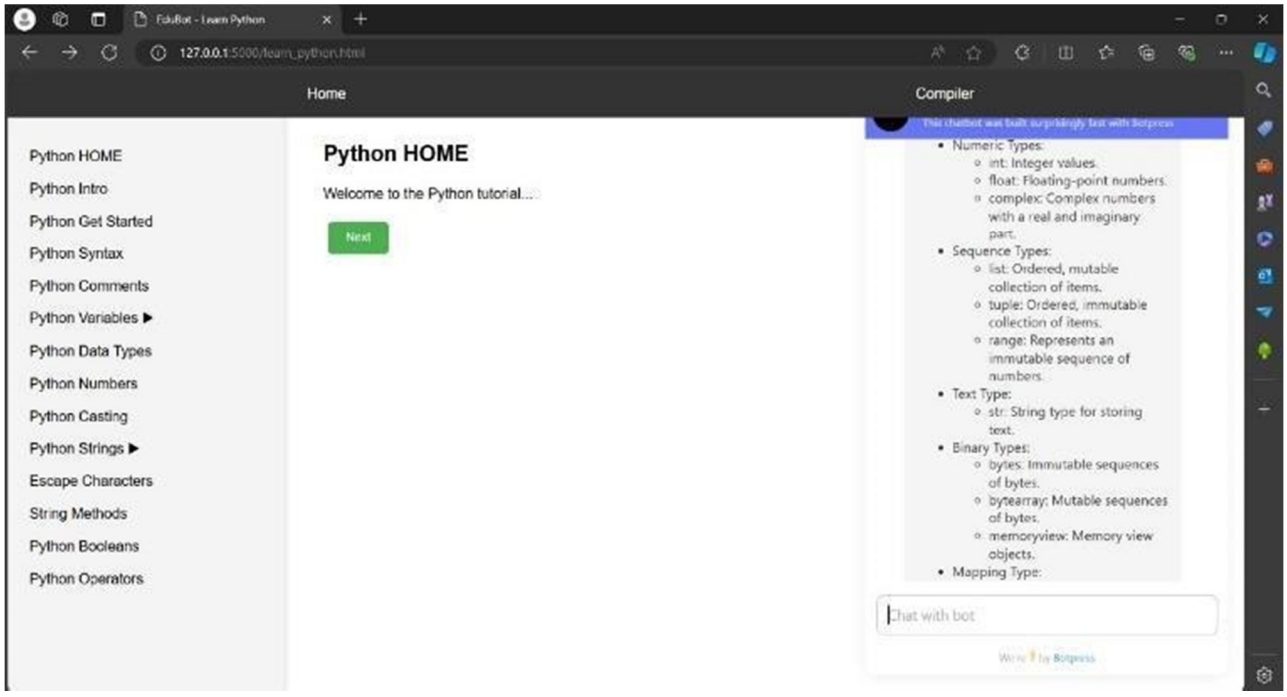


Figure 3 : AI Chatbot for Real-Time Assistance

The chatbot, powered by Botpress, allows users to ask Python-related questions and receive instant feedback. Unlike static FAQ pages, it provides dynamic responses based on user input.

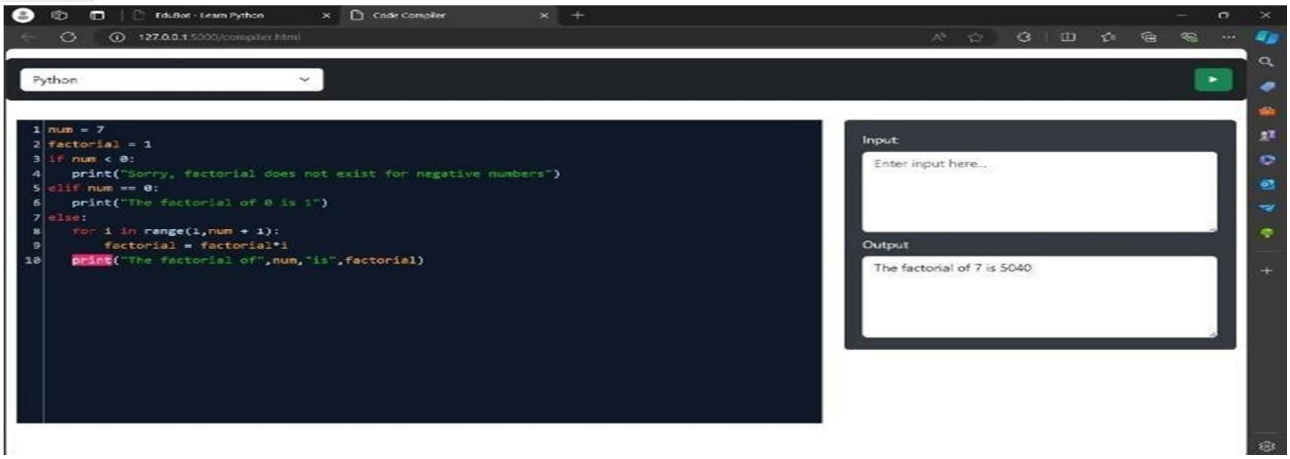


Figure 4 : Integrated Python Compiler

The Python compiler, built with Code Mirror, allows learners to write and execute Python code within the platform. Features include syntax highlighting, real-time error detection, and autocomplete suggestions.

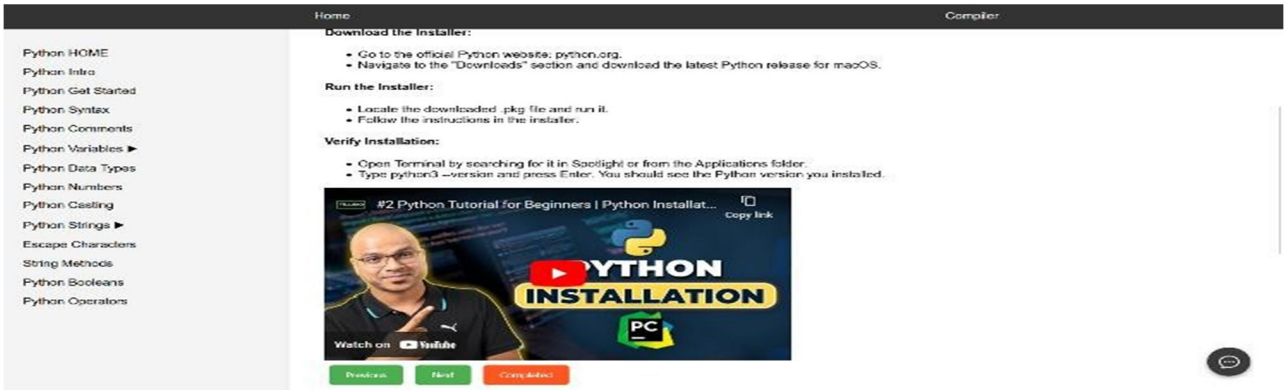


Figure 5 : Adaptive Video Recommendation System

Edu-Bot suggests personalized video tutorials using the YouTube Data API, helping learners find relevant explanations for Python concepts without leaving the platform.



Figure 6 : Face-Tracking for Engagement Monitoring

This feature ensures learners stay focused by tracking screen presence using JavaScript and WebRTC. If a user becomes inactive, they receive gentle reminders to maintain study consistency.

VIII. CONCLUSIONS

Edu-Bot is a cutting-edge and innovative way of learning Python that aims to make programming education more interactive, engaging, and efficient. In contrast to conventional learning platforms that use static tutorials and independent coding environments, Edu-Bot integrates everything into a smooth, AI-driven experience. With its real-time chatbot support, built-in Python compiler, customized video suggestions, and engagement monitoring, the platform keeps learners on track, motivated, and actively engaged in the learning process. The platform has been found to generate encouraging improvements in user interaction and learning achievements. The chatbot built on Botpress gives immediate responses, minimizing frustration and making learning simpler and more engaging. The embedded Python compiler with CodeMirror enables learners to code in real time, with detection of errors and syntax highlighting to assist them in correcting errors promptly. Also, the video learning system provides the learner with content that is pertinent to their level of progress, making it unnecessary to look for additional resources. The face-tracking feature also promotes engagement by assisting students in staying focused and having improved study habits.

Nevertheless, inasmuch as Edu-Bot offers a number of benefits, there are areas for improvement. The chatbot, though efficient, may have trouble processing intricate programming questions, suggesting the importance of a more sophisticated NLP model. Certain users may experience the face-tracking aspect as invasive, hence introducing an optional feature to turn it on or off would allow for greater flexibility in use. Moreover, the platform is presently centered on Python only, and this might restrict its use for students who are interested in other programming languages. Ahead lies more enhancement towards more chatbot functionalities, implementing support for more languages such as Java, C++, and JavaScript, and supporting features of gamification such as quizzes, challenge questions, and reward points in order to engage the users. Additions including voice-operated chatbot sessions and a dedicated app for smartphones will make it still more popular and accessible to its users. In summary, Edu-Bot is a robust and innovative learning tool that enhances access, personalization, and experiential learning of Python. Through ongoing development, it can revolutionize programming education to make learning smarter, quicker, and more interactive for learners globally

REFERENCES

[1] Labadze, L., Grigolia, M. & Machaidze, L. Role of AI chatbots in education: systematic literature review. Int J Educ Technol High Educ 20, 56 (2023). https://doi.org/10.1186/s41239-023-00426-1

[2] El Azhari, K., Hilal, I., Daoudi, N., & Ajhoun, R. (2023). SMART Chatbots in the E-learning Domain: A Systematic Literature Review. International Journal of Interactive Mobile Technologies (iJIM), 17(15), pp. 4–37. https://doi.org/10.3991/ijim.v17i15.40315.

[3] Goodson, C.E., Miertschin, S.L., & Stewart, B.L. (2015). TIME MANAGEMENT SKILLS AND STUDENT PERFORMANCE IN ONLINE COURSES. DOI 10.18260/p.24921

[4] Chung Kwan Lo and Khe Foon Hew .A review of integrating AI-based chatbots into flipped learning: new possibilities and challenges Front. Educ., 22 May 2023 Sec. Digital Education Volume 8 - 2023 | https://doi.org/10.3389/feduc.2023.1175715

[5] Yedjou, Clement G, Tchakoua, Clauditte T, Tchounwou, Martha, Latinwo, Lekan, Eidahl, Kyle, Alo, Richard A, and Liu, Hong. RECENT ADVANCES IN ARTIFICIAL INTELLIGENCE (AI) IN EDUCATION, ETHICAL CONCERNS AND IMPLICATIONS. Retrieved from. International Journal of Science Academic Research https://par.nsf.gov/biblio/10531733

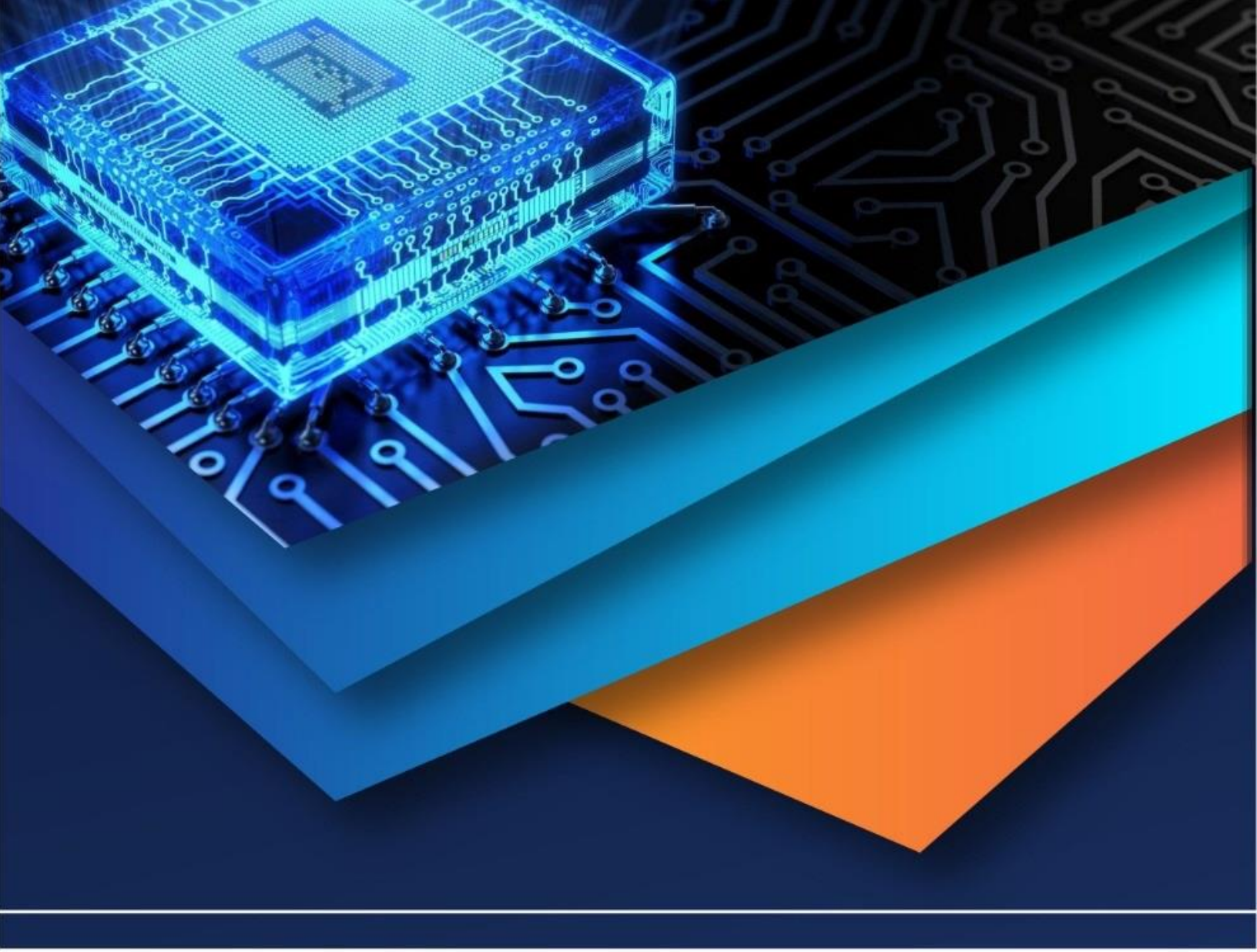
[6] Wienand, M., Wulfert, T. & Hoang, H. Design principles for e-learning platforms featuring higher-education students' enterprise systems end-user training. Discov Educ 3, 82 (2024). https://doi.org/10.1007/s44217-024-00165-z

[7] AlAfnan, M. A., Samira Dishari, Marina Jovic, & Koba Lomidze. (2023). ChatGPT as an Educational Tool: Opportunities, Challenges, and Recommendations for Communication, Business Writing, and Composition Courses. Journal of Artificial Intelligence and Technology, 3(2), 60–68. https://doi.org/10.37965/jait.2023.0184

[8] Bassam Alsanousi, Abdulmohsen S. Albesher, Hyunsook Do and Stephanie Ludi, "Investigating the User Experience and Evaluating Usability Issues in AI-Enabled Learning Mobile Apps: An Analysis of User Reviews" International Journal of Advanced Computer Science and Applications(IJACSA), 14(6), 2023. http://dx.doi.org/10.14569/IJACSA.2023.0140602

[9] Yu Chen & Scott Jensen & Leslie J. Albert & Sambhav Gupta & Terri Lee, 2023. "Artificial Intelligence (AI) Student Assistants in the Classroom: Designing Chatbots to Support Student Success," Information Systems Frontiers, Springer, vol. 25(1), pages 161-182, February. DOI: 10.1007/s10796-022-10291-4.

[10] el Azhari, Khadija & Hilal, Imane & Najima, Daoudi & Ajhoun, Rachida. (2023). SMART Chatbots in the E-learning Domain: A Systematic Literature Review. International Journal of Interactive Mobile Technologies (iJIM). 17. 4-37. 10.3991/ijim.v17i15.40315.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)