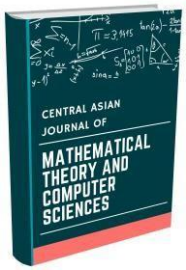

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.casjournal.org/index.php/cajmtcs
Volume: 07 Issue: 03 | July 2026 ISSN: 2660-5309

Article

Smart Compiler: An AI-Powered IDE for Intelligent Code Analysis and Execution

V. Chrysolite*[1], K. Senthamilselvan[2]

1. Department of Computer Science and Engineering, Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India
2. Department of Electronics and Communication Engineering, Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India
* Correspondence: v.chrysolite@dhaanishchennai.in

Abstract: Smart Compiler is an AI based Integrated Development Environment (IDE) to enhance the coding experience of both beginners and professionals. It supports a number of programming languages such as Java, C and Python so that users are able to write, compile and run programs all in one place. The system, powered by Google's Gemini API, analyses code intelligently, assisting users in identifying syntax errors and providing optimised solutions with detailed explanations for novices. This feature helps programmers to understand mistakes better and improve their coding skills over time. The IDE has a built-in terminal where you can see the output of your program as it happens, so testing and debugging is easy. The graphical user interface is developed with Java Swing providing interactive and user-friendly environment and OkHttp is used to communicate with APIs efficiently. Gemini AI improves code readability, suggests performance improvement and enforces better programming standards. The Smart Compiler connects learning platforms and professional development tools by integrating code execution with AI-powered feedback. Thus the system provides efficient coding, less time spent on debugging and continuous learning and is a valuable tool for modern software development.

Keywords: Smart Compiler, Intelligent Code Analysis, Java Swing, Real-Time Output, Built-in Terminal, Personalized Feedback, Code Correction Suggestions

Citation: Chrysolite, V., Senthamilselvan, K. Smart Compiler: An AI-Powered IDE for Intelligent Code Analysis and Execution. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(3), 115-129.

Received: 30th Apr 2026
Revised: 18th May 2026
Accepted: 05th Jun 2026
Published: 24th Jun 2026

1. Introduction

The Smart Compiler is a revolutionary and sophisticated AI driven Integrated Development Environment that aims to make programming easier, better code learning and higher coding efficiency for all users regardless of their skill levels. In today's fast-paced technological landscape, programming is a crucial skill across many industries. Yet, many learners and developers find it challenging to write efficient, error-free code [41]. The Smart compiler attempts to resolve these issues by providing a common platform where the user can develop, compile and execute the programs in different languages like Java, C, Python etc [54]. The system can provide real-time code analysis and suggestions for improvement with the integration of smart technologies such as Gemini AI, making it a powerful tool for both novice and experienced developers [34]. Unlike conventional IDEs that focus on code execution, the Smart Compiler emphasises learning and understanding,

providing users with the opportunity to develop solid programming fundamentals, and improving their productivity [61]. One of the key features of the Smart Compiler is that it gives you real-time feedback as you code [48]. As users write their code the system continually analyses it for syntax errors, logical issues, and inefficiencies. This immediate feedback allows users to catch and fix errors early, reducing the time spent debugging and improving the quality of the code.

This is further enhanced by the integration of Gemini AI, which provides intelligent suggestions for optimisation and best practices [66]. Such suggestions are provided in a beginner-friendly manner to allow users to not only correct errors but also understand the logic of correction. This method helps users to learn programming concepts better and to develop better coding habits over time [38]. It also comes with advanced debugging tools that assist users in locating logical errors, understanding program execution, and optimising performance, making it a comprehensive solution for coding and tackling problems. The Smart Compiler aims to solve the problem of missing integrated learning support and intelligent guidance in current development environments [52]. Many beginners have a hard time learning to program because they don't get immediate help or get clear explanations of their mistakes. The error messages generated by traditional IDEs tend to be technical and hard to understand, causing users to become confused and frustrated [60]. Moreover, developers often need to install many compilers/IDEs to support different programming languages, which is inefficient and complicates the system. The Smart Compiler solves these problems by providing a one-stop-shop platform to support a multitude of languages and intelligent AI-powered feedback [45]. It makes coding seamless, eliminates the need for multiple tools, and allows users to focus on learning and development rather than setup and configuration.

The goals of the Smart Compiler are to improve the coding experience, make programming more accessible and efficient [47]. The idea is to have an all-in-one platform that supports Java, C and Python programming so that users can work across different languages without switching environments [63]. Its purpose is to offer real-time feedback, error detection, and intelligent suggestions to enhance the quality of the code and decrease the chances of mistakes [65]. Another key goal is to increase learning efficiency by offering simple explanations, tutorials and examples that help users understand programming concepts [33]. The Smart Compiler aims to fill the gap between learning platforms and professional development tools by combining coding capabilities with educational resources. This way, users will not only write code, but will also acquire the skills needed to become proficient programmers [40]. The motivation and the scope of the Smart Compiler project is based on the motivation of simplifying programming education and making it accessible to a wider audience. To beginners, programming can be daunting, particularly when they come across errors that they can’t comprehend [55]. The Smart Compiler provides users a safe environment to experiment, learn and improve their skills without fear of failure to solve this problem. Real-time feedback and intelligent suggestions motivate users to code more frequently and experiment with new ideas [59].

Moreover, the user-friendly interface of the system makes it more accessible as users can easily and intuitively navigate and manage code [44]. The Smart Compiler reduces the entry barrier in conventional development environments and promotes more people to get engaged with programming and to build their technical skills. The current system, Smart Compiler, offers all the features that assist in coding and learning. It has a simple code editor that allows users to write, edit and organise their programs easily [37]. The editor features syntax highlighting, auto-indentation, and other features for readability and ease of use. After writing the code, users can compile and run the code within the IDE without any external tools. The system gives program output in real time, with any errors or warnings, so users can quickly see problems and fix them [62]. Coding and execution are integrated seamlessly, and the workflow is smooth, productivity rises and frustration declines. The system is comprised of several modules, each of which is designed for

specific functions and to contribute to the overall user experience [49]. The code editor module is the main interface for writing and editing programs, and offers essential features such as syntax highlighting, code formatting, and file management.

The compiler and executor module is responsible for compilation and execution of the code [42]. It ensures that the programs are executed correctly and efficiently. This module is multi-lingual, so you can switch between languages easily [58]. The AI assistant module is a key part of the system providing real-time feedback, error checking and intelligent recommendations. It works in the background to analyse code and suggest improvements to help users improve their programming skills and produce quality code [35]. The learning module contributes to this by providing tutorials, sample programs and explanations to help users understand programming concepts in detail [67]. These modules operate together to create a comprehensive learning, practice and development environment for users to enhance their coding skills. The AI assistant is especially important to this process, because it gives personalised feedback based on the user's code [51]. The feedback is customised according to the user's level of expertise, offering simple explanations for beginners and detailed insights and optimisation suggestions for more advanced users.

The experience is even better with the learning module, that provides the user tutorials and examples in a structure that guides the user through the different concepts of programming [46]. Interactive learning mixed with hands-on application provides users with a thorough understanding of programming. The Smart Compiler also has the important advantage of supporting multi-language projects, enabling developers to work on different programming languages within the same environment [56]. This feature is especially useful for projects that need multi-language support, saving the need to switch between different IDEs or tools. The multi-language support in the system also provides a flexible learning environment, as users can choose to explore different programming paradigms/techniques. It also includes advanced debugging tools to enhance the system's ability to help users quickly locate and fix problems [39]. They give insight into program execution, letting users see how their code is behaving and where improvements can be made [64]. In addition to technical features, Smart Compiler also focuses on user experience and accessibility. The interface is intuitive and easy to navigate, so users can focus on coding and not get bogged down in complex menus or configurations [53]. Real-time output display, error highlighting, and interactive suggestions make for a smooth and engaging user experience.

The system also contains inbuilt tutorials and examples which are useful for beginners and help them to gain confidence in their programming skills [43]. The Smart Compiler creates an environment that is supportive and efficient and encourages users to explore programming and develop their skills [57]. To sum up, the Smart Compiler is a major step forward in the field of programming tools, integrating the capabilities of a classic IDE and the strength of the artificial intelligence and educational helping [50]. It tackles the problems of the beginners and developers to provide a complete solution for enhanced learning, increased productivity and improved coding practices. The combination of real-time feedback, intelligent suggestions and multi-language support makes it a powerful platform for modern programmers [36]. Its innovative design and user-centric approach could revolutionise the way people learn and practise programming, making it more accessible, efficient, and enjoyable for users of all levels.

Review of Literature

But the brilliance of it is that it can do multiple programming languages [21]. The same platform allows users to write, compile and run programs in Java, C and Python without having to switch to different software environments. This removes the need for users to install separate compilers or IDEs for each language, making it easier to efficiently practise and experiment with multiple languages [3]. Students and developers benefit

from the integration of multiple languages, as they can work in a single space for learning and development. Instant Feedback: Smart Compiler utilises artificial intelligence to give feedback to the user while they write code [27]. This feature detects syntax mistakes, logic errors and common coding issues on the fly. The system flags errors as you type so you see what you've done wrong immediately and correct it right there [14]. It also provides real-time feedback, which helps avoid the accumulation of errors and reinforces learning by showing the user the right approach while they are coding.

Code Suggestions One important feature is the AI-powered code suggestion system [11]. The AI module reviews the structure and logic of the written code and suggests enhancements to optimise performance, improve readability, and follow best coding practices [9]. Such suggestions could be refactoring poor-performing code, proposing better data structures, or recommending simpler alternatives for complex code blocks. This feature helps users to write cleaner and efficient programs so it is suitable for both beginners and experienced developers [5]. Smart Compiler provides step-by-step tutorials and pre-written example programs to help you learn. For beginners, there are structured lessons, which can be used to learn programming concepts [32]. Advanced users can browse examples and learn efficient coding techniques [18]. The tutorials offer a blend of basic concepts, problem-solving methods and real-world applications to make the learning process interactive and practical.

Smart Compiler also has smart debug tools. The AI-assisted debugger helps users find and fix errors quickly and efficiently, providing hints and explanations for common problems [20]. This means that learners will find it less frustrating and development will be more efficient and accurate. These functional features together make Smart Compiler a comprehensive platform which not only allows executing code but also helps in learning, increases productivity and encourages best practices in programming [1]. With integrated AI support, users get guidance and feedback at every step of the coding process, making the IDE a wise learning companion. Besides the functional capabilities, Smart Compiler also emphasises a lot on non-functional features [23]. Non-functional features play vital role to provide smooth, efficient and user-friendly experience. Functional features are related to what the system can do, non-functional features are related to how well the system performs, how reliable is it and how easy is it for users to interact with it [15]. These aspects are crucial in building an IDE that is useful for coding and conducive to learning.

High performance is one of the main non-functional features of the Smart Compiler [13]. The system is designed to compile and execute code fast and thus the user is waiting less time. This is useful especially when writing complex programs or executing multiple code snippets [26]. Smart Compiler provides fast compilation and execution, which allows users to frequently test their programs and get immediate feedback, an important factor in learning and debugging. High performance also means no slow response times, so you can have more productive and enjoyable coding sessions. Another significant non-functional aspect of Smart Compiler is reliability [7]. The system can detect errors such as syntax errors, logical errors, and runtime exceptions with high accuracy. This makes the system less likely to crash and produce incorrect outputs [29]. Reliable performance is especially important for beginners, as frequent crashes or unexpected behaviour can confuse them and interfere with the learning process [17]. Smart Compiler offers a stable environment for users to focus on writing and understanding code without the worry of unexpected technical issues.

The IDE has a simple and intuitive user interface that makes it easy to use and navigate [22]. Menus, options and editor layouts The Smart Compiler, the current system, is architected in a well-defined manner integrating a front-end IDE with a back-end AI engine, offering seamless integration of coding, learning and intelligent assistance [30]. The architecture provides efficient compilation, execution and feedback processing while providing a user friendly interface. To analyse the core functions of Smart Compiler and

the advanced learning support it provides, it is necessary to understand the system architecture [10]. Smart Compiler is built at a high level as a client-server where the front-end IDE interacts with the back-end components for compilation/execution and AI-based analysis of the code [6]. The front end part includes the code editor, menus and user interface elements that allow the user to write and manage their programs. The code editor is built to accept exact user input and to interact in real-time with other parts of the system [24]. It provides syntax highlighting, indentation and basic error detection to enhance usability and code readability.

The code is written and run by the user. The input is sent to the compiler module on the backend [12]. The compiler's job is to transform a high-level programming language like Java, C or Python into machine-executable code. During this phase, the compiler checks for syntax errors, semantic correctness and other common programming errors [19]. If the code passes those checks, it is run and the output is returned to the front-end for display. The front-end and compiler work together in a seamless manner, enabling users to test their programs efficiently without changing environments. The AI engine runs in the background concurrently with the compilation process [2]. It provides real-time analysis of user code and suggests ways to improve code structure, efficiency, and detect logical or runtime errors. The AI engine's pattern recognition and pre-defined coding standards provide meaningful feedback [28]. The system enables continuous learning by piping this feedback back to the front-end in real-time so users can correct mistakes and learn best practices as they code.

The architecture also has a learning module with tutorials, examples and exercises [16]. The module interacts with the front-end to show interactive learning content based on the user's coding activity. This ensures the theoretical learning is supported with practical experience and makes the system very effective for beginners as well as advanced users [25]. In summary, the Smart Compiler system architecture includes an easy-to-use IDE, a powerful compiler, a real-time AI analysis engine, and a systematic learning module [31]. This architecture guarantees that the platform offers a fluid coding and learning experience, integrating development, education and intelligent support into a cohesive system [8]. The workflow of the existing system, Smart Compiler, details the steps a user follows to write, compile, run, and improve code on the platform. To understand how the system integrates coding, learning, and AI-powered feedback into a seamless experience, it is important to understand the workflow [4]. The workflow allows users to concentrate on learning and development, while providing them with guidance and support at each stage.

2. Methodology

The workflow starts with writing code. When the user opens the Smart Compiler IDE, they are welcomed with a code editor that supports multiple programming languages including Java, C and Python. The editor provides syntax highlighting, auto-indentation and basic error detection [69]. These features let the user create code quickly, while reducing simple mistakes. Projects or files help users to organise their code so that they can manage multiple programs at the same time. Once the code is written, it is then compiled [73]. The code editor will submit the user's program to the compiler module which will check the code for syntax and semantic correctness. The compilation process identifies errors like missing semicolons, undefined variables, and incompatible data types [71]. If the code contains errors, the system will point them out right away and you can fix the mistakes before running the program. This step ensures that only error-free code is run, saving time and frustration.

The execution phase begins after a successful compilation. The compiler translates the code into machine instructions and runs the program. The program's output is shown in the IDE, so users can see if their code produces the expected results. This step also

provides users with the ability to try different scenarios and learn how their programs behave in real-time. At the same time, the AI feedback module is non-stop analysing the user's code [74]. This module provides suggestions for code optimisation, structural enhancements and potential logical improvements. Through the process of coding and execution, users are provided feedback that helps them learn best practices and develop a deeper understanding of programming concepts. It helps in debugging and gives hints and solutions for non-obvious errors [72]. Finally, the learning support phase combines tutorials, examples and exercises. Depending on the user activity, the system may suggest to the user relevant tutorials or sample programs to improve the understanding, and provide further practice. This process of writing, compiling, running, getting feedback and learning is iterative, making Smart Compiler a development environment and a continuous learning platform [68]. Smart Compiler's workflow is structured to lead users through every step of coding and learning effectively. The system combines an interactive code editor, compilation, execution, AI feedback and learning support to create a productive and educational programming experience [70].

3. Result and Discussion

It is a platform for coding and learning but the current system has some limitations which affect its usability and efficiency in certain conditions. It is important to identify these limitations in order to understand where the system needs to be improved and to guide the development of future versions [79]. One of the major disadvantages of Smart Compiler is that it supports only three programming languages which are Java, C and Python. These are popular languages but the platform cannot be accessed by users who would like to learn or work with other languages like C++, JavaScript, or Ruby [89]. This limitation impacts the systems' flexibility for users who need multi-language support for academic projects or professional development. Another limitation is the reliance on internet connectivity for some AI-driven features [95]. Basic coding, compiling and execution can be done offline but advanced features such as real-time code suggestions, AI-based debugging and intelligent optimisation require an active internet connection [84]. This can be problematic for users in areas with poor or limited access to the Internet, thus reducing the effectiveness of the system when used offline (Figure 1).

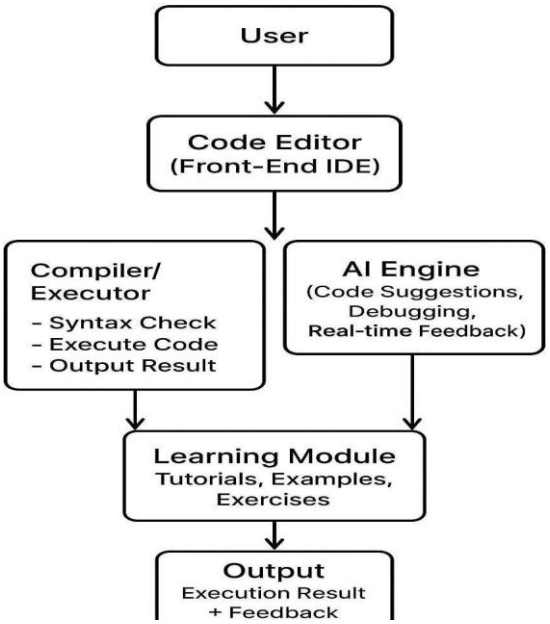

Figure 1. Block Diagram

The Smart Compiler may not function properly on large or complex projects [93]. It works best for small to medium size programs. If you run multiple large files at the same time, it can take time to compile and run. In addition, handling AI feedback can be time-consuming, affecting the experience of advanced users and their productivity when working with large codebases. Users have limited flexibility to customise the IDE to their preferences. The interface is user friendly and intuitive, but options for extensive theme, layout or feature configuration are limited [78]. This limitation might be problematic for users who want a customised coding environment, or who are used to highly configurable IDEs in professional development. tutorials and examples can be helpful, but can also foster a guided-learning mindset in users. Students who rely heavily on pre-written code or AI suggestions may not develop strong independent problem-solving skills if they do not practise coding without help [85]. To sum up, Smart Compiler is a powerful and intelligent platform to learn and develop, but it is not without limitations. Limited language support, internet needed for some features, performance issues with large projects, limited customisation and possible learning curve.

The system offers an advanced interactive environment where users can easily write, test and develop their programming skills. It offers a Code Editor that is the main environment for users to develop and manage their code efficiently. When editing, you have syntax highlighting to help you distinguish keywords, variables, and functions, so the code is easier to read and debug [82]. It also includes auto-completion, which helps users by predicting code elements as they are typed, thereby reducing syntax errors and speeding up coding [96]. When the user completes the coding, the Compiler or Executor performs a syntax check to check whether the code is correct according to the language structure, and contains no basic errors. Any syntax or logic errors are immediately recognised and displayed to the user in a clear and understandable manner for quick correction [75]. Once the code passes the validation phase, it is then run to produce the output of the program and users can see in real time how their program works and what are the results. The system also has an intelligent AI Engine that constantly monitors the code to make sure it adheres to the best programming practices and optimisation standards [99]. The AI also provides real-time debugging hints and suggestions, helping to identify subtle issues that may not stop the program from running but could affect performance or logic [90]. It also provides useful tips to improve the efficiency, readability and structure of the code, thus promoting better programming practices over time. This is further enhanced by a dynamic Learning Module for an enriched learning experience (Figure 2).

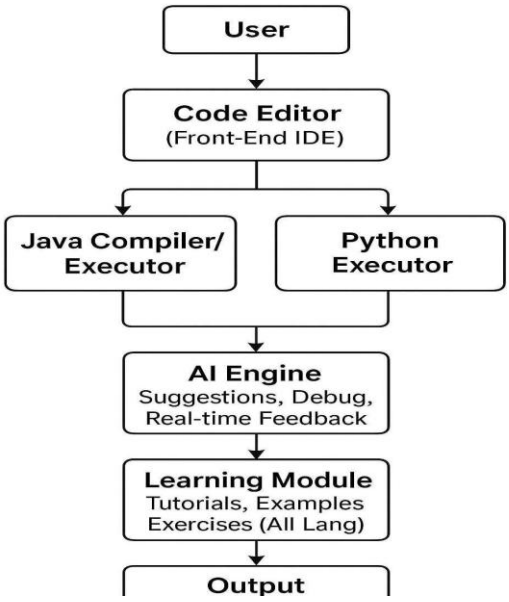

Figure 2. ER Diagram for Python Code

It gives you an advanced interactive environment to write, test and improve your programming skills with ease [88]. It offers a Code Editor that is the main environment for users to develop and manage their code efficiently. When editing, you have syntax highlighting to help you distinguish keywords, variables, and functions, so the code is easier to read and debug. The IDE also supports auto-completion feature that predicts code elements while typing which helps users to reduce the number of syntax errors and increase the speed of code writing [77]. Once the user has finished writing code, the Compiler or Executor checks the syntax to make sure that the code conforms to the correct language structure and has no basic errors [98]. If any errors in the syntax or logic are detected, the errors are immediately pinpointed and presented to the user in a simple and understandable form, so that they can be rectified quickly. Once the code passes the validation phase, it is then run to produce the output of the program and users can see in real time how their program works and what are the results [83]. The system also has an intelligent AI Engine that constantly monitors the code to make sure it adheres to the best programming practices and optimisation standards. The AI also provides real-time debugging hints and suggestions, helping to identify subtle issues that may not stop the program from running but could affect performance or logic. It also provides constructive feedback on how to improve code efficiency, readability and structure, promoting better programming habits over time [94]. This is complemented by an interactive Learning Module that enhances the learning experience (Figure 3).

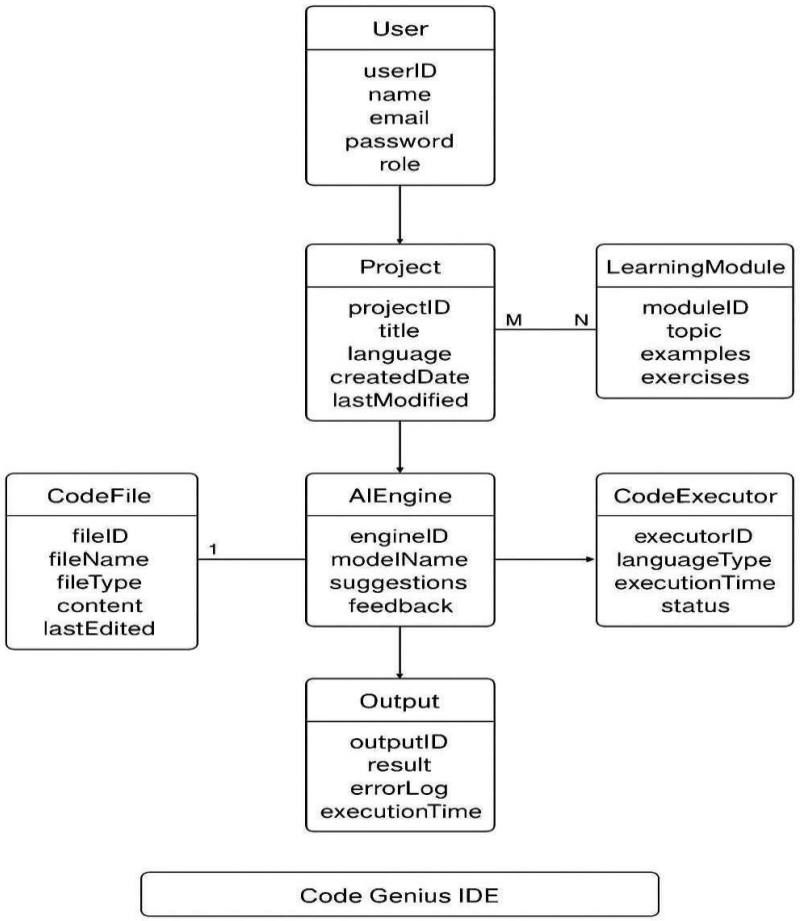

Figure 3. ER Diagram for Java

The system is intended to provide an intelligent interactive environment for a user to write, learn and improve his coding skill [86]. It begins with a Code Editor for efficient code writing and management. The editor features syntax highlighting to visually distinguish parts of the code and auto-completion to speed up typing and reduce common

errors. When the user writes the code the syntax is checked for structural or language-based mistakes [80]. A Compiler or Executor checks the syntax before the program is run. Any errors are identified and the user is presented with the opportunity to correct them [91]. If there are no errors in the code syntax, the code will be executed and you will see the output of the program (Figure 4).

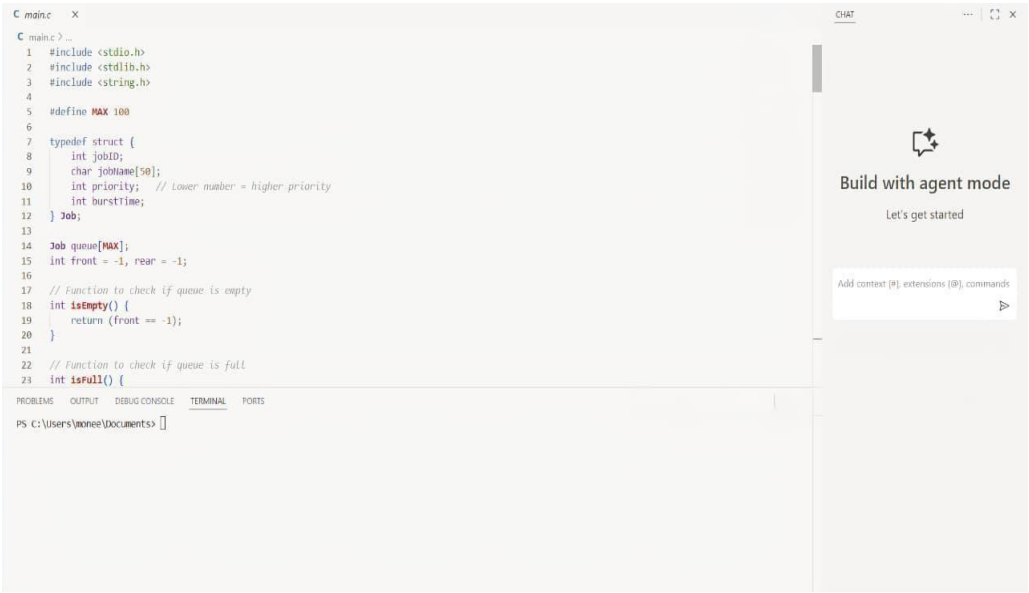

Figure 4. Result in Python

In addition, a built-in AI Engine is continuously analysing the code to ensure it follows best practices and is optimised for performance [81]. It also offers real-time assistance for debugging by offering possible fixes and explaining logical errors that don't necessarily cause your program to stop running but might affect your results. The system is further complemented by a Learning Module to assist learning [92]. This module provides tutorials, examples, and hands-on exercises specific to the user's current code or programming topic [76]. It enables the users to develop a better understanding of concepts and use them efficiently. And once the program is executed, it shows the combined output of the program and the AI feedback, so it is easy for the user to see how their code performed and how to improve it [87]. The whole integrated approach encourages the user to keep changing and reworking their code, learning from their mistakes and getting into better habits when it comes to coding [97].

4. Conclusion

This paper "Smart Compiler – AI Powered IDE with Gemini Integra on" proves the success of blending traditional code execution environments with the power of Artificial Intelligence. The smart coding platform lets users write, compile and run programs in multiple languages such as Java, C and Python and get AI-powered code analysis and suggestions from Google's Gemini. In this project, I have successfully implemented the Gemini API through HTTP communication, using the OkHttp library in Java. This allowed for a smooth interaction between the IDE and the AI model, providing developers with immediate insights, debugging tips, and code optimisation advice without having to leave their workspace. The user interface is developed with Java Swing and has a clean, intuitive and resizable layout with separate sections for code editing, AI suggestions and output display. This makes the platform functional and user-friendly even for beginners. This IDE bridges the gap between human understanding and the machine by combining AI capabilities with a local compiler setup. In conclusion, Smart Compiler is a step towards the future of intelligent development environments, where AI acts as a coding companion

to boost productivity, lower learning curves and foster innovation in software development. Potential enhancements include support for more programming languages, dark mode themes, cloud-based saving and more advanced AI-assisted code completion features.

References

[1] M. M. Reddy Chinthala and M. Kalloji, "Policy-Oriented Zero Trust Microsegmentation for East-West Traffic Governance in Hybrid Cloud Architectures," 2025 6th International Conference on Smart Electronics and Communication (ICOSEC). IEEE, pp. 1330–1335, Sep. 24, 2025.

[2] M. M. R. Chinthala, H. Apuri, and K. Bitra, "Behaviour-Aware Hybrid Deep Networks for Detecting Zero-Day and Ransomware Threats," IJITEE, vol. 15, no. 5, pp. 1–10, Apr. 2026.

[3] S. Venkatasubramanian, "G-MAPQR: GNN-assisted multi-agent predictive quality of service routing for high-mobility MANETs," IETE Journal of Research, pp. 1–15, 2026.

[4] S. Venkatasubramanian, S. Raja, V. Sumanth, J. N. Dwivedi, J. Sathiaparkavi, S. Modak, and M. L. Kejela, "Fault diagnosis using data fusion with ensemble deep learning technique in IIoT," Mathematical Problems in Engineering, vol. 2022, Art. no. 1682874, pp. 1–8, 2022.

[5] S. Raja, J. Logeshwaran, S. Venkatasubramanian, M. Jayalakshmi, N. Rajeswari, N. G. Olaiya, and W. D. Mammo, "OCHSA: Designing energy-efficient lifetime-aware leisure degree adaptive routing protocol with optimal cluster head selection for 5G communication network disaster management," Scientific Programming, vol. 2022, Art. no. 5424356, pp. 1–11, 2022.

[6] S. Venkatasubramanian, J. N. Dwivedi, S. Raja, N. Rajeswari, J. Logeshwaran, and A. Praveen Kumar, "Prediction of Alzheimer's disease using DHO-based pretrained CNN model," Mathematical Problems in Engineering, vol. 2023, Art. no. 1110500, pp. 1–11, 2023.

[7] H. Apuri and C. Yepuri, "Design of multi-agent autonomous workflow systems using agentic AI frameworks," in Proc. IEEE Int. Conf. Advances in Urban Computing (ICAUC), 2026.

[8] H. Apuri, M. Aurangabadkar, S. Goel, M. M. R. Chinthala, and C. Yepuri, "Advancing infrastructure-as-code resilience through generative AI agents for predictive remediation and autonomous security enforcement," Int. J. Engineering and Advanced Technology (IJEAT), vol. 15, no. 4, Mar. 2026.

[9] H. Apuri et al., "Self-healing infrastructure: Autonomous LLM agents for real-time remediation of configuration drift and security misconfigurations in IaC deployments," Int. J. Innovative Technology and Exploring Engineering (IJITEE), vol. 15, no. 4, Mar. 2026.

[10] H. Pandian, "AI-based capacity forecasting models for elastic cloud and hybrid enterprise systems," Journal of Information Systems Engineering and Management, vol. 10, no. 63s, pp. 1648–1658, Dec. 2025.

[11] H. Pandian, "AI-driven predictive performance bottleneck detection in mission-critical financial systems," Journal of Computational Analysis and Applications, vol. 30, no. 2, pp. 1019–1033, Mar. 2022.

[12] H. Pandian, "Architectural optimization techniques for high-volume batch processing in Hadoop ecosystems," International Journal of Intelligent Systems and Applications in Engineering, vol. 8, no. 4, pp. 428–439, Dec. 2020.

[13] H. Pandian, "Embedding performance engineering into CI/CD pipelines for regulated financial systems," Journal of Computational Analysis and Applications, vol. 33, no. 8, pp. 7892–7910, Dec. 2024.

[14] H. Pandian, "Performance engineering as a first-class cybersecurity control," in Proc. International Conference on Cyber Security, IoT, Data & Information Technology, vol. 459, pp. 1–12, Jun. 2024.

[15] V. T. Manne, "Post-quantum cryptography migration framework for real-time payment gateways," in 2026 Second International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), 2026, pp. 1–7.

[16] V. T. Manne, "Adaptive graph-based risk scoring for real-time instant payment systems," in 2026 Second International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), 2026, pp. 1–7.

[17] S. B. Venkata, "Risk-aware rework prevention in personalized hearing aid manufacturing," Journal of Computer Science and Technology Studies, vol. 4, no. 2, pp. 215–230, 2022.

[18] S. B. Venkata, "Predictive infrastructure orchestration in azure using terraform and dynatrace for medical systems," in 2025 International Conference on Data, Energy and Communication Networks (DECoN), 2025, pp. 1–6.

[19] K. Sharma and B. Goswami, "Rental market of pump-sets in the central and western parts of Nepal plains," Asia-Pacific Journal of Rural Development, vol. 30, no. 1–2, pp. 226–243, 2020.

[20] K. Sharma, "Mechanization without ownership: Market structure and pricing in Nepal's Terai," CABI Agriculture and Bioscience, vol. 7, no. 1, p. 0020, Feb. 2026.

[21] K. Sharma, P. Basnet, and K. R. Bhatt, "Social media discussion and short-horizon stock returns: Evidence from a retail coordination episode," Digital Finance, vol. 8, no. 1, pp. 1–17, 2026.

[22] K. Sharma and R. N. Shrestha, "The migration paradox: Why remittances fail to stimulate agricultural investment in Nepal's Terai plains," Economics Bulletin, vol. 45, no. 4, pp. 1649–1657, Dec. 2025.

[23] N. Chawla, P. Kotla, S. R. Venna, and M. B. Patel, "Comprehensive analysis of robotic process automation for software project management," in 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), IEEE, 2025.

[24] P. Kotla and D. Sen, "Robotic process automation in business process modelling using improved sine cosine algorithm with chaotic theory," in Proc. 2025 3rd International Conference on Data Science and Network Security (ICDSNS), 2025.

[25] P. Kotla, "Accelerating shared services with UiPath: Lessons from early automation centres of excellence (CoEs)," PowerTech Journal, vol. 46, no. 2, pp. 27–39, May 2022.

[26] P. Kotla, "Adaptive learning in UiPath: Enhancing RPA for continuous improvement and scalability," PowerTech Journal, vol. 47, no. 2, pp. 174–191, Apr. 2023.

[27] B. Kumar, "Blockchain-based authentication model for education data storage," International Journal of Industrial and Systems Engineering, vol. 51, no. 4, pp. 498–515, 2025.

[28] V. Ramalingam, B. Kumar, S. K. Gupta, D. M. Alsekait, and D. S. AbdElminaam, "A hybrid federated learning framework with generative AI for privacy-preserving and sustainable security in IoT-enabled smart environments," Scientific Reports, vol. 16, Art. no. 3071, 2026.

[29] B. Kumar, W. Shamas, J. Sandeep, and D. Albalushi, "Developing an advanced cybersecurity framework and blueprint: A contemporary approach to counter hacking through reverse engineering techniques," in Smart Cyber Physical Systems, Proceedings of ICSCPS 2024, Singapore: Springer, 2024, pp. 27–37.

[30] B. Kumar, N. B. Najmusseher, P. K. Nizar Banu, and R. Dwivedi, "Epileptic seizure detection contribution in healthcare sustainability," in AI and IoT: Driving Business Success and Sustainability in the Digital Age, B. Awwad, Ed., Studies in Systems, Decision and Control, vol. 601, Cham, Switzerland: Springer, 2025, pp. 225–235.

[31] B. Kumar and O. Al Falhi, "Digital transformation through APIs," in Proc. Int. Conf. on Communication, Information Technology and Internet of Things (COM-IT-CON), 2022.

[32] K. Al Aufi and B. Kumar, "Security testing of Android application using Drozer," in Proc. Int. Conf. on Computational Sciences and Sustainable Technologies, Springer CCIS, 2024, pp. 8–18.

[33] B. Al Barwani, E. Al Maani, and B. Kumar, "IoT-enabled smart cities: A review of security frameworks, privacy, risks, and key technologies," in Proc. 1st Int. Conf. on Innovation in Information Technology and Business (ICIITB 2022), Advances in Computer Science Research, vol. 104, Springer, 2022, pp. 169–181.

[34] H. Pandian, "Performance-driven development (PDD): A new software engineering paradigm," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 9, no. 12, pp. 278–286, Dec. 2021.

[35] H. Pandian, "Quantifying business risk and financial loss from performance failures in enterprise systems," in Proc. International Conference on Computer Science, Engineering and Applications, vol. 978, pp. 160–170, Jul. 2021.

[36] H. Pandian, “Self-healing performance architectures for large-scale banking and payment platforms,” International Journal of Intelligent Systems and Applications in Engineering, vol. 11, no. 11s, pp. 1019–1029, Dec. 2023.

[37] H. Pandian, “Unified observability framework for enterprise performance, capacity, and reliability,” Journal of Electrical Systems, vol. 14, no. 4, pp. 96–106, Apr. 2018.

[38] H. Pandian, “Workload characterization models for distributed enterprise systems,” International Journal on Recent and Innovation Trends in Computing and Communication, vol. 7, no. 12, 2019.

[39] D. Sumathi, S. Ayyappan, A. Sivasangari, B. Ramakrishna, P. Kanagaraju, and F. B. Kunwar, “Random forest with SHAP analysis for identifying environmental determinants of air quality in urban regions,” Journal of Environmental Protection and Ecology, vol. 26, no. 8, pp. 2963–2973, 2025.

[40] D. Sumathi, S. Ankam, P. P. Adivarekar, K. S. S. Sandeep, R. Gomathi, and R. Shobarani, “Early plant disease detection using graph isomorphic networks: Enhancing crop yield through leaf analysis,” Journal of Computer Science, vol. 21, no. 9, pp. 2065–2073, Oct. 2025.

[41] D. Sumathi, A. Mishra, and D. K. Jha, “AI-based stress prediction: Integrating psychological and behavioral data using deep learning,” in Proc. International Conference on Innovative Computing and Communication (ICICC), Lecture Notes in Networks and Systems, vol. 1435, vol. 6, no. 1, pp. 145–158, 2025.

[42] D. Sumathi and P. Poongodi, "Scheduling Based on Hybrid Particle Swarm Optimization with Cuckoo Search Algorithm in Cloud Environment," IIOAB JOURNAL, vol. 7, no. 9, pp. 358-366, 2016.

[43] T. P. Krishna Kumar, M. Ramachandran, and V. Saravanan, “Risk assessment of emergency management using WASPAS MCDM method,” Recent Trends in Management and Commerce, vol. 2, no. 3, pp. 36–43, 2021.

[44] T. P. Krishna Kumar, M. Ramachandran, and V. Saravanan, “Candidate selection for a project using weight sum method,” Data Analytics and Artificial Intelligence, vol. 1, no. 1, pp. 53–59, 2021.

[45] T. P. Krishna Kumar, M. Ramachandran, and S. Sivaji, “Supplier selection analysis using multi-criteria decision-making VIKOR method,” Data Analytics and Artificial Intelligence, vol. 1, no. 1, pp. 48–52, 2021.

[46] T. P. Krishna Kumar, M. Ramachandran, S. Chinnasamy, and M. Mani, “Matrix organization analysis using grey relational analysis,” REST Journal on Banking, Accounting and Business, vol. 1, no. 1, pp. 64–71, 2022.

[47] T. P. Krishna Kumar, M. Ramachandran, and K. Ramu, “Emergency management investigation using COPRAS method,” Recent Trends in Management and Commerce, vol. 2, no. 1, pp. 32–37, 2022.

[48] T. P. Krishna Kumar, D. R. Pallavi, M. Ramachandran, and C. Raja, “Evaluation of techno-economic using decision making trial and evaluation laboratory (DEMATEL) method,” Recent Trends in Management and Commerce, vol. 3, no. 2, pp. 101–110, 2022.

[49] T. P. Krishna Kumar, M. Ramachandran, C. Raja, and A. Murugan, “Understanding of e-learning programs using WPM MCDM method,” REST Journal on Banking, Accounting and Business, vol. 1, no. 2, pp. 13–19, 2022.

[50] T. P. Krishna Kumar, V. Saravanan, M. Ramachandran, and M. Selvam, “A market segmentation assessment weighted scoring for using WSM method: A study for different market,” REST Journal on Banking, Accounting and Business, vol. 1, no. 3, pp. 1–8, 2022.

[51] T. P. Krishna Kumar, M. Ramachandran, K. Ramu, and A. Murugan, “Analysis of reverse logistics system using COPRAS MCDM method,” REST Journal on Banking, Accounting and Business, vol. 1, no. 4, pp. 31–37, 2022.

[52] T. P. Krishna Kumar, M. Ramachandran, C. Sivaji, and C. Raja, “Financing practices of micro and small entrepreneurs using WSM MCDM method,” REST Journal on Data Analytics and Artificial Intelligence, vol. 1, no. 4, pp. 18–25, 2022.

[53] T. P. Krishna Kumar, M. Ramachandran, V. Prasanth, and C. Raja, “Developing business services using IBM SPSS statistics,” REST Journal on Banking, Accounting and Business, vol. 2, no. 1, pp. 40–50, 2023.

[54] T. P. Krishna Kumar, M. Ramachandran, K. Ramu, and A. Murugan, “Using DEMATEL for corporate social responsibility (CSR) analysis,” REST Journal on Banking, Accounting and Business, vol. 2, no. 1, pp. 51–59, 2023.

[55] S. D. Khan, T. P. Krishna Kumar, A. Anjutha, S. Santhanalaxmi, K. Sasirekha, and T. Srihari, "Entrepreneurship, innovation, and technological change: Catalysts of economic evolution—A descriptive study," Migration Letters, vol. 21, no. S1, pp. 962–971, 2024.

[56] T. P. Krishna Kumar, R. Suriakala, A. Krishnaprasad, and D. Nair, "The impact of Internet of Things (IoT) on supply chain optimization: A review of existing literature," The National Research Journal of Information Technology and Information Science, vol. 12, no. 1, pp. 5–13, 2025.

[57] T. P. Krishna Kumar, R. Suriakala, A. Krishnaprasad, and D. Nair, "Bridging psychological foundations: Transforming Indian education through NEP-2020 and NCF-2023," Academe Journal of Education & Psychology, vol. 15, no. 1, pp. 1–16, 2025.

[58] S. Venkatasubramanian, V. Mohan, A. Subasri, S. H. Prasath, A. Thenmozi, and M. A. D. Thirumanraj, "Decentralized IoT frameworks: Blockchain-enabled trust in smart ecosystems," in Edge Computing and Applications, N. Ramalingam, Y. El Allioui, and S. Bhattacharyya, Eds., Cham, Switzerland: Springer, 2026.

[59] S. Venkatasubramanian, S. K. B. Pradeep Kumar CH, and M. Aurna Devi, "Optimization of IoT communication using low-power wide-area networks," in Proc. 6th Int. Conf. Advances in Electrical, Computing, Communications and Sustainable Technologies (ICAECT), Bhilai, India, 2026, pp. 1–7.

[60] S. Venkatasubramanian, "Real-time intrusion detection in IoT networks: A novel approach combining modified salp swarm feature selection and GLIRU model," in Improving Threat Detection, Network Security, and Incident Response with AI, Hershey, PA, USA: IGI Global, 2025, pp. 307–332.

[61] S. Nagar, "An impact of performance of companies with sustainability goal on stock market movements," European Economic Letters, vol. 13, no. 3, pp. 417–420, 2023.

[62] S. Nagar and P. Mahajan, "An assessment of effectiveness of remote work mode in job internships," The Online Journal of Distance Education and E-Learning (TOJDEL), vol. 11, no. 2, p. 1154, 2023.

[63] S. Nagar, "Augmentation of insurance business in India: Role of commercial banks," in India Banking and Finance Report 2024, National Institute of Bank Management (NIBM), pp. 81–94, 2024.

[64] S. Nagar, "Cointegration of Indian stock market with global stock markets: An empirical analysis," European Economic Letters, vol. 14, no. 2, pp. 2457–2464, 2024.

[65] L. Jose, "Contextual Diachronic Semantic Framework: Advancing Literary Analysis and Pedagogy through the Semantic Study of Shirley Jackson's 'The Lottery'," Educational Process: International Journal, vol. 17, Art. no. e2025377, Aug. 14, 2025.

[66] P. Jothilingam, "Towards autonomous commissioning: Integrating digital twins, artificial intelligence and smart sensors for next-generation process control systems," Certified Journal of International Research (CJIR), vol. 5, no. 1, pp. 1-8, Mar. 2025.

[67] P. Jothilingam, "Edge computing for industrial automation and control: Enabling real-time processing, scalable architectures and secure operations," Certified Journal of International Research (CJIR), vol. 5, no. 1, pp. 1–8, Mar. 2025.

[68] P. Jothilingam, "Advancing cybersecurity in industrial control systems: Frameworks, threat modeling, and resilience strategies," International Journal of Supportive Research (IJSR), vol. 2, no. 2, pp. 69–75, Jul. 2024.

[69] P. Jothilingam, "Cybersecurity in water and wastewater systems: Protecting critical infrastructure from emerging threats and ensuring operational resilience," in Proc. International Conference on Recent Advances in Science, Engineering, Technology and Management, India, Mar. 2024, pp. 550–558.

[70] P. Jothilingam, "Digital twin technologies for ICS: Leveraging virtualization and sensor data for FAT/SAT, commissioning and predictive risk detection," International IT Journal of Research, vol. 1, no. 1, pp. 45–49, Oct. 2023.

[71] I. Ganie and S. Jagannathan, "Online continual safe reinforcement learning–based optimal control of mobile robot formations," Proc. 2024 IEEE Conf. Control Technol. Appl. (CCTA), Newcastle upon Tyne, U.K., 2024, pp. 519–524.

[72] I. Ganie and S. Jagannathan, "Online learning-driven human intent estimation and control for human–robot interaction," Proc. 2025 Amer. Control Conf. (ACC), Denver, CO, USA, 2025, pp. 5160–5165.

[73] I. Ganie and S. Jagannathan, “Online lifelong optimal tracking control of nonlinear continuous-time strict-feedback systems using deep neural networks,” Neural Netw., vol. 191, 107793, 2025.

[74] I. Ganie and S. Jagannathan, “Optimal trajectory tracking of uncertain nonlinear continuous-time strict-feedback systems with dynamic constraints,” Int. J. Control, pp. 1–15, 2024.

[75] H. R. Laskar, “Adoption of fintech and digital financial services (DFS) by young professionals,” Int. J. Adv. Res. Eng. Technol., vol. 11, no. 1, pp. 537–561, 2020.

[76] H. R. Laskar, “Factors influencing saving and investment behavior of government and private sector employees,” Indian Journal of Economics and Business, vol. 20, no. 1, pp. 1168–1192, 2021.

[77] S. Roushon and H. R. Laskar, “Influence of Neural Behaviour on Decision Making,” IOSR Journal of Humanities and Social Science, vol. 29, no. 5, ser. 13, pp. 35–42, May 2024.

[78] Z. Alam and H. R. Laskar, “The Influence of Neural Behavior on Individuals’ Financial Decisions,” Journal of Economics, Finance and Management Studies, vol. 7, no. 6, pp. 3298–3306, Jun. 2024.

[79] S. Laskar, H. R. Laskar, and M. N. I. Barbhuyan, “Perception of Women Entrepreneurs Regarding Social Media Marketing,” Bangladesh Journal of Multidisciplinary Scientific Research, vol. 9, no. 5, pp. 10–18, Nov. 2024.

[80] V. K. Nomula, A. S. Mohammed, A. R. Neravetla and S. Dhanasekaran, "Leveraging Deep Learning in Implementing Efficient Healthcare Processes," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6.

[81] I. A. Mohammed, "Artificial Intelligence in Supplier Selection and Performance Monitoring: A Framework for Supply Chain Managers," Educational Administration: Theory and Practice, vol. 29, no. 3, pp. 1186–1198, 2023.

[82] I. A. Mohammed, "The Role of Artificial Intelligence in Enhancing Business Efficiency and Supply Chain Management," Journal of Information Systems Engineering and Management, vol. 10, no. 10s, pp. 509–518, Feb. 2025.

[83] I. A. Mohammed, "AI-Powered Risk Management Frameworks for Ensuring Supplier Quality in Carbon Capture and Energy Storage Supply Chains," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 1, pp. 854–, Dec. 2023.

[84] I. A. Mohammed, "Optimizing Carbon Capture Supply Chains with AI-Driven Supplier Quality Management and Predictive Analytics," Journal of Next-Generation Research 5.0, Dec. 2024.

[85] I. A. Mohammed, "Machine Learning-Driven Predictive Models for Enhancing Supplier Reliability in Renewable Energy Storage Supply Chains," International Journal of Intelligent Systems and Applications in Engineering, pp. 767–770, 2022.

[86] N. Gupta, M. Adawadkar, I. A. Mohammed, S. Verma, and M. Dubey, "Predictive Insights: Leveraging Artificial Intelligence for Strategic Business Decision-Making," Advances in Consumer Research, vol. 2, pp. 98–105, 2025.

[87] U. A. K. Yokubbaeva, P. P. Devi, S. Mahadevan, and D. S. U. Sharipov, "Words and algorithms: The intersection of linguistic and artificial intelligence," AIP Conf. Proc., vol. 3306, p. 050005, 2025.

[88] V. Devi Vanniarajan and S. Shfmkari, "Service quality of life insurance companies at Salem," Global Business Review, vol. 2, no. 2, pp. 23-31, 2008.

[89] V. Kumar, P. P. Devi, T. N. Babu, A. S. Nader, A. A. S. Mohammed and R. Saravanakumar, "AI-Powered Recruitment Marketing Enhancing Candidate Experience and Employer Branding," 2025 IEEE International Conference on Emerging Technologies and Applications (MPSec ICETA), Gwalior, India, 2025, pp. 1-6.

[90] V. P. Rameshkumaar, D. Ganesan, S. Revathy, R. Karthikeyan, and P. P. Devi, "Evaluating the impact of artificial intelligence on logistics and supply chain efficiency," Indian Journal of Natural Sciences, vol. 15, no. 88, pp. 88823–88828, Feb. 2025.

[91] Y. A. Abduvakhob Kizi, P. Praba Devi, S. Mahadevan, and S. D. S. Ugli, "Words and algorithms: The intersection of linguistic and artificial intelligence," AIP Conference Proceedings, vol. 3306, no. 1, p. 050005, 2025.

[92] S. G. K. Peddireddy, "Advancing Threat Detection in Cybersecurity through Deep Learning Algorithms," FMDB Transactions on Sustainable Intelligent Networks., vol.1, no. 4, pp. 190–200, 2024.

[93] S. G. K. Peddireddy, "Integrating AI for Proactive Network Defense against Emerging Security Vulnerabilities," FMDB Transactions on Sustainable Computer Letters., vol. 2, no. 4, pp. 232–241, 2024.

[94] S. G. K. Peddireddy, "Optimizing Resource Allocation in Multi-Cloud Environments for Cost Efficiency and Scalability," FMDB Transactions on Sustainable Computing Systems., vol. 2, no. 4, pp. 167–177, 2024.

[95] V. S. A. Anala, A. R. Pothu, and S. Chintapalli, "Enhancing Preventive Healthcare with Wearable Health Technology for Early Intervention," FMDB Transactions on Sustainable Health Science Letters., vol.2, no.4, pp. 211–220, 2024.

[96] V. S. A. Anala and S. Chintapalli, "Scalable Data Partitioning Strategies for Efficient Query Optimization in Cloud Data Warehouses," FMDB Transactions on Sustainable Computer Letters., vol. 2, no. 4, pp. 195–206, 2024.

[97] O. Alimbaeva, A. Joshi, G. Saritha, L. H. Alzubaidi, K. Senthamil Selvan, and A. Chaudhary, "Novel Materials for High-Performance Energy Storage Devices," in E3S Web of Conferences 13th International Conference on Power and Energy Systems (ICPES 2023), Chengdu, China, 2024.

[98] B. Jayaprakash, D. Bordoloi, P. Mehta, A. Amudha, P. Marwaha, and K. S. Selvan, "A Network for Medical Segmentation of Images with Multiple Centers That Preserves Privacy," in Proceedings of the 2024 Global Conference on Communications and Information Technologies (GCCIT 2024), Bangalore, India, 2024.

[99] M. Vigenesh, M. Grover, K. Senthamilselvan, A. Singla, M. Chethan, and S. B. Patil, "Secure Data Aggregation in Wireless Sensor Ad Hoc Networks Using Homomorphic Encryption," in Proceedings of the 15th International C9onference on Computing, Communication and Networking Technologies (ICCCNT 2024), Himachal Pradesh, India, 2024.