iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78302

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 14 Issue III Mar 2026- Available at www.ijraset.com

Web-Based Integrated Code Editor and Compiler Platform for Multiple Programming Languages

B. Sowjanya[1], SK. Rizwana[2], K. Sirisha[3], S. Preetham[4], R. Ramesh[5]
[1]*Assistant professor, Department of Computer science and Engineering (Artificial intelligence &Machine Learning),Avanthi institute of engineering of technology, India*
[2, 3, 4, 5]*Department of Computer science and Engineering (Artificial intelligence &Machine Learning),Avanthi institute of engineering of technology, India*

Abstract: ***Traditional programming environments require users to install Integrated Development Environments (IDEs) and configure compilers locally, which can be time-consuming and device dependent. This paper presents an Intelligent Web-Based Code Editor designed to allow users to write, compile, and execute programs directly from a web browser without requiring local installation. The system is developed using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. The platform supports multiple programming languages such as C, C++, Java, Python, and JavaScript. It provides features such as syntax highlighting, real-time output display, secure code execution through sandbox environments, and user authentication. The system improves accessibility for students, developers, and educational institutions by providing a centralized coding platform that can be accessed from any device with internet connectivity. The proposed solution reduces system dependency, improves coding efficiency, and enhances learning experiences for programming students.***

Keywords: Web-Based Code Editor, MERN Stack, Online Compiler, Multi-Language Programming, Cloud-Based Development.

I. INTRODUCTION

Programming is an essential skill in modern software development and education. Traditionally, programmers rely on desktop-based Integrated Development Environments (IDEs) such as Visual Studio, Eclipse, or NetBeans to write and execute code. However, these tools require installation, configuration of compilers, and significant system resources.

With the advancement of cloud computing and web technologies, web-based code editors have become an effective alternative to traditional development environments. These platforms allow developers to write and execute code directly in a web browser without installing software. The Intelligent Web-Based Code Editor proposed in this project aims to provide a simple and efficient environment for writing, compiling, and executing programs online. The system supports multiple programming languages and includes features such as syntax highlighting, secure code execution, and real-time output display.

The platform is designed using the MERN stack architecture where React is used for building the frontend interface, Node.js and Express.js handle backend logic, and MongoDB stores user data and code snippets. This approach ensures scalability, performance, and easy accessibility across different devices.

II. LITERATURE SURVEY

Earlier programming environments depended heavily on desktop-based Integrated Development Environments (IDEs). Tools such as Eclipse, NetBeans, and Visual Studio provided advanced features including debugging, code completion, and project management. However, these tools required installation and configuration, making them less accessible for users who needed quick programming environments [1].

With the emergence of lightweight editors such as Visual Studio Code and Sublime Text, developers gained faster and more flexible coding tools. Despite their advantages, these editors still required local installation and manual configuration of compilers [2].

The development of web-based code editors introduced a new approach to programming environments. Platforms such as Replit, CodePen, and JSFiddle allow users to write and execute code directly from browsers. These platforms provide accessibility and collaborative coding capabilities, especially for educational purposes [3].

Recent research also focuses on integrating secure execution mechanisms such as Docker-based sandbox environments to safely run user-submitted code without affecting server stability [4].

The use of modern web technologies such as React.js, Node.js, and MongoDB enables developers to build scalable and efficient web-based development platforms [5].

©IJRASET: All Rights are Reserved

These advancements highlight the potential of cloud-based coding platforms in improving accessibility and simplifying programming environments**.**

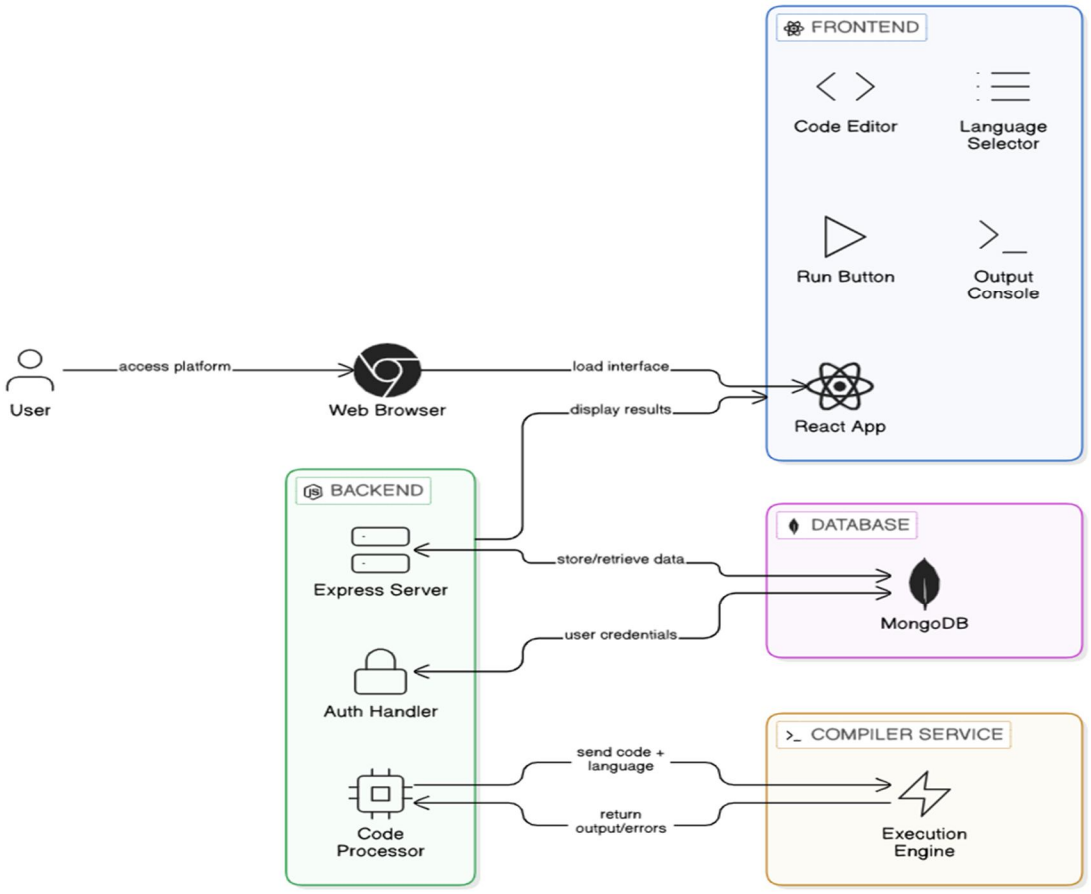
III. PROBLEM STATEMENT

Traditional programming environments require users to install software, configure compilers, and manage dependencies manually. These processes can be difficult for beginners and restrict access to programming environments across different devices.

Additionally, educational institutions often require dedicated computer laboratories with pre-installed development tools, which increases infrastructure costs.

The problem addressed in this project is the lack of a simple and accessible platform where users can write and execute code online without installation. The proposed Intelligent Web-Based Code Editor provides a cloud-based solution that allows users to code directly from their web browsers while ensuring secure code execution and efficient performance.

IV. SYSTEM ARCHITECTURE



The proposed system architecture follows a three-layer structure consisting of the frontend, backend, and database layers.

The frontend is developed using React.js, which provides an interactive user interface where users can write code, select programming languages, and view output results.

The backend is implemented using Node.js and Express.js, which handle API requests, process code execution requests, and manage authentication.

The database layer uses MongoDB to store user information, saved code snippets, and execution history.

When a user writes code and clicks the run button, the code is sent to the backend server through an API request. The backend then executes the code in a secure sandbox environment and returns the output or error message to the frontend interface.

This architecture ensures secure execution, scalability, and efficient communication between system components.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

V. METHODOLOGY

1) User Authentication: The system first allows users to register and log into the platform. Secure authentication is implemented using encrypted passwords and token-based authentication to protect user data.
2) Code Writing and Editing: Users can write code in an interactive code editor integrated into the web interface. The editor supports syntax highlighting, auto-indentation, and line numbering to improve readability and coding efficiency.
3) Language Selection: The system allows users to select the programming language from a dropdown menu. Supported languages include C, C++, Java, Python, and JavaScript.
4) Code Execution: When the user clicks the run button, the code is sent to the backend server. The backend identifies the selected programming language and executes the program in a sandbox environment.
5) Output Generation: After execution, the system generates the program output or error message. The result is sent back to the frontend and displayed in the output console

VI. TECHNOLOGY STACK

The system is developed using modern web development technologies.

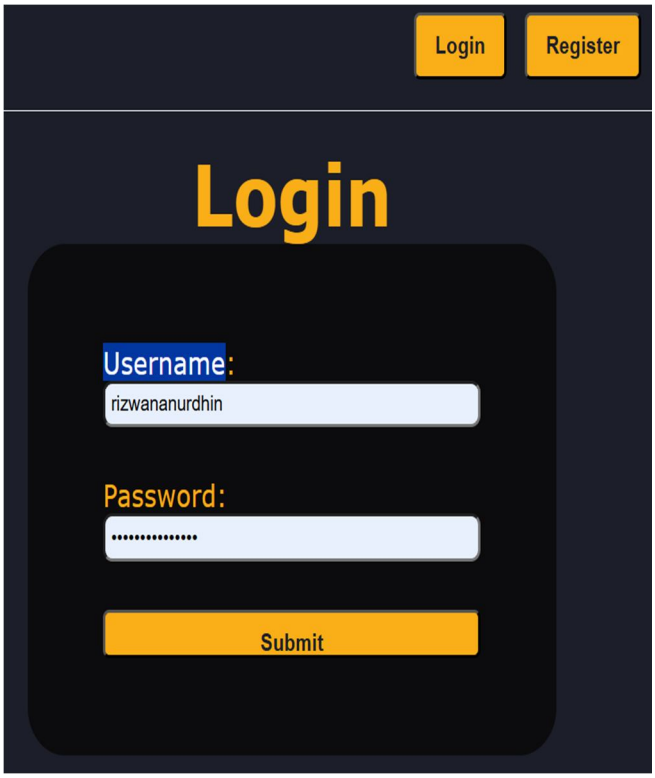
1) Frontend: React.js, HTML, CSS, JavaScript
2) Backend: Node.js, Express.js
3) Database: MongoDB
4) Code Execution Environment: Secure sandbox execution using server-side compilers
5) Development Tools: Visual Studio Code, GitHub, Postman

These technologies enable the system to provide a scalable, secure, and responsive web application.

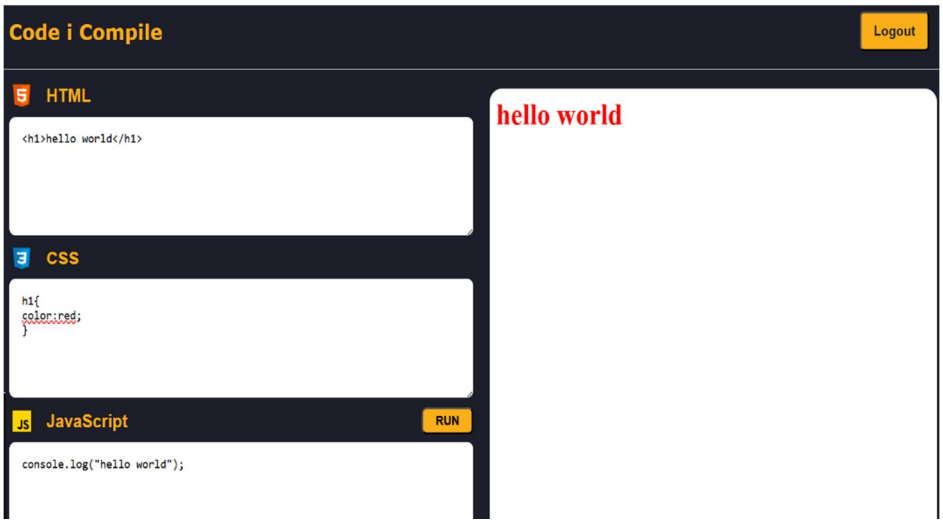
VII. INPUT & RESULT

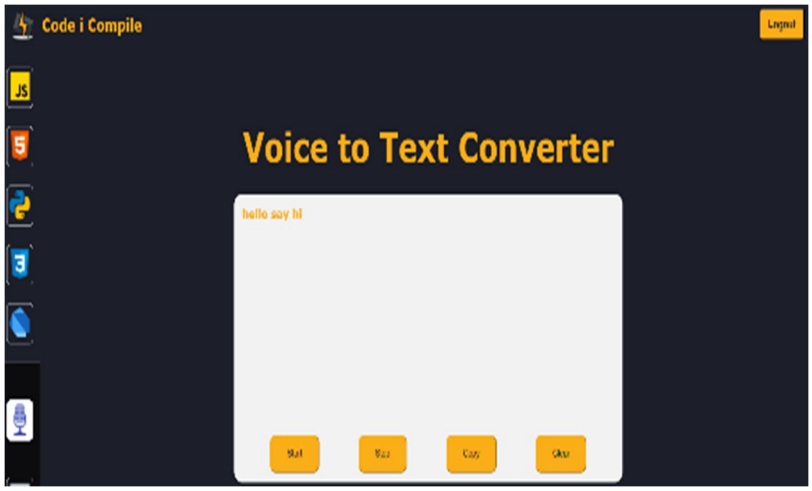
A. *Input*

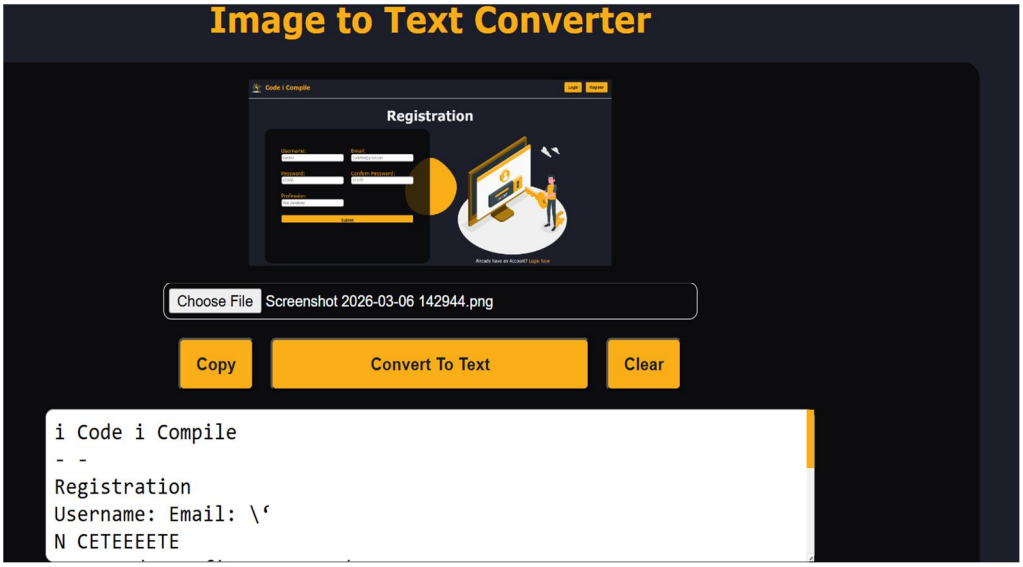
Users provide input in the form of programming code written in the code editor. They also select the programming language in which the program should be executed.

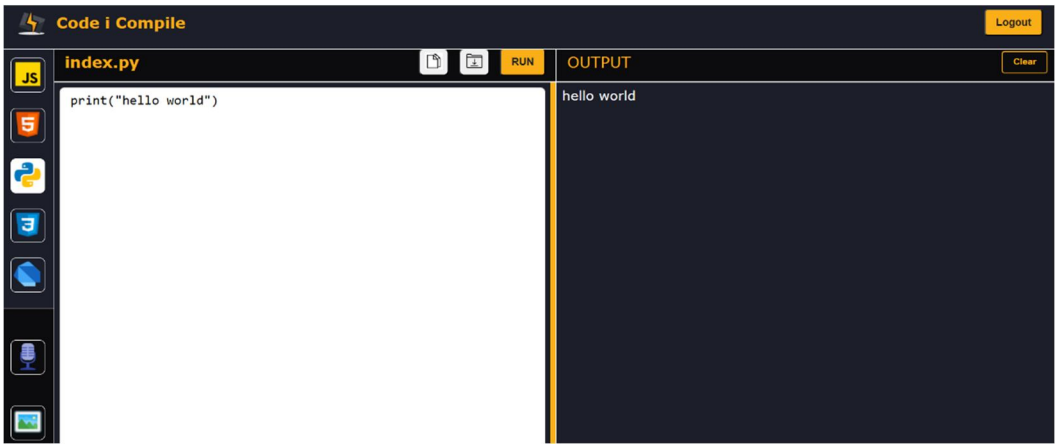


©IJRASET: All Rights are Reserved


Code i Compile
Logout
HTML
<h1>hello world</h1>
CSS
h1{
color:red;
}
JavaScript
RUN
console.log("hello world");
hello world

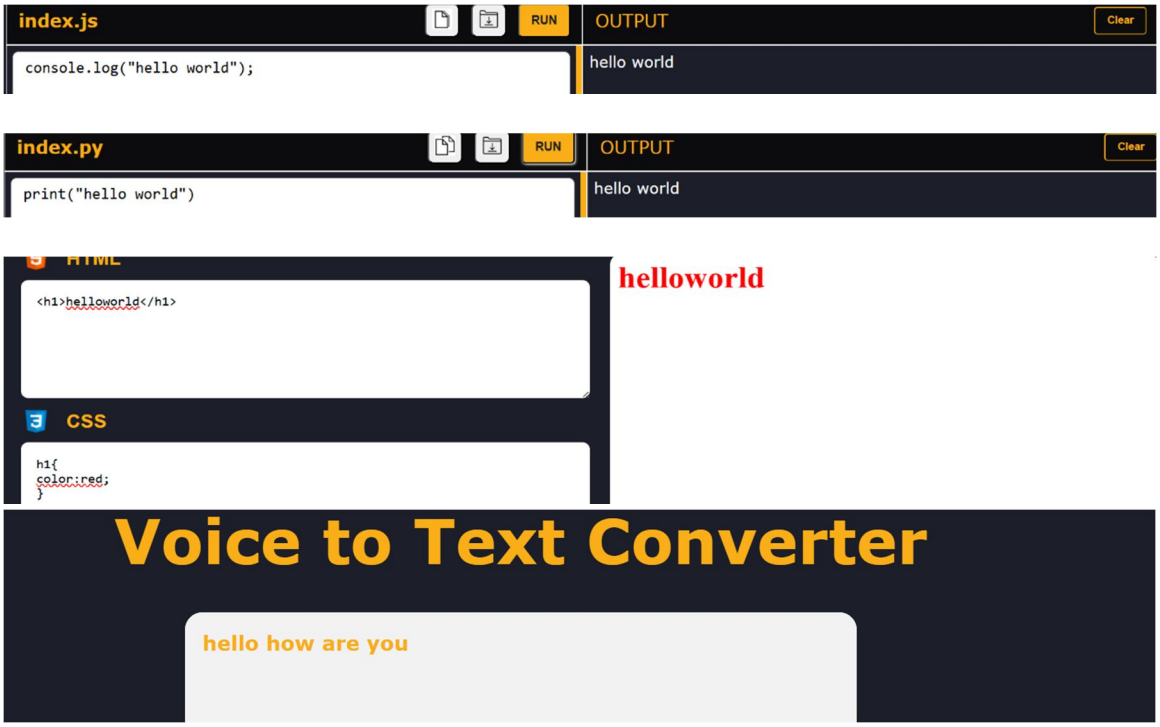

Code i Compile
Voice to Text Converter
hello say hi


Image to Text Converter
Registration
Choose File
Screenshot 2026-03-06 142944.png
Copy
Convert To Text
Clear
i Code i Compile
- -
Registration
Username: Email: \‘
N CETEEEETE



B. Result

After executing the code, the system displays the output or error message in the output console. This allows users to debug their programs and view results instantly.



VIII. ADVANTAGES

The proposed system provides several advantages compared to traditional programming environments.

The platform eliminates the need for installing compilers or IDEs. It provides accessibility from any device with internet connectivity. The system supports multiple programming languages and provides real-time output for programs.

The secure execution environment ensures that user-submitted code does not affect the server system. The user-friendly interface also makes the platform suitable for beginners and students learning programming.

IX. FUTURE SCOPE

The system can be further enhanced by integrating Artificial Intelligence features such as automated code suggestions and intelligent debugging assistance. Future improvements may also include real-time collaborative coding, competitive programming modules, and cloud-based project storage. Mobile application support and integration with version control systems such as GitHub can also enhance the functionality of the platform.

©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 14 Issue III Mar 2026- Available at www.ijraset.com

X. CONCLUSION

The Intelligent Web-Based Code Editor successfully provides an online platform for writing and executing programs without requiring local installation of development tools. The system uses the MERN stack architecture to provide a scalable and efficient web application.

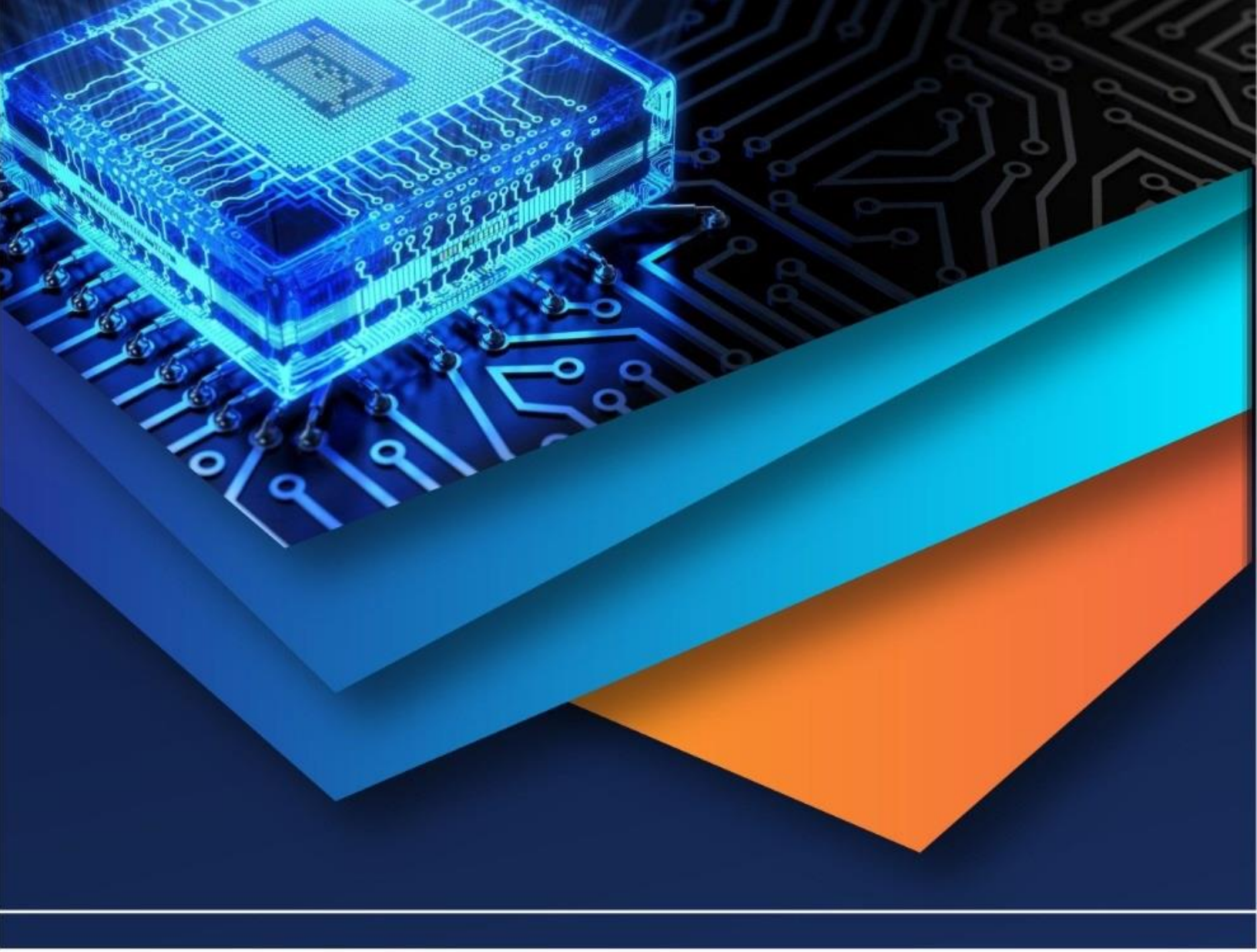
The platform improves accessibility, simplifies coding environments, and supports multiple programming languages. It is particularly useful for students, developers, and educational institutions that require an easy-to-use programming platform.

By integrating modern web technologies with secure code execution mechanisms, the system demonstrates the potential of cloud-based development tools in modern programming environments..

REFERENCES

[1] Node.js Documentation. Available: https://nodejs.org/
[2] Express.js Documentation. Available: https://expressjs.com/
[3] MongoDB Documentation. Available: https://www.mongodb.com/docs/
[4] Microsoft, “Monaco Editor Documentation.” Available: https://microsoft.github.io/monaco-editor/
[5] D. Spinellis, Code Quality: The Open Source Perspective, Addison-Wesley, 2006.
[6] M. Fowler, Refactoring: Improving the Design of Existing Code, Addison-Wesley, 2018.

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



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)