

Online Code Editor

**Sanjana. S. Dindewar¹, Trupti. S. Wasade², Riya. G. Manusmare³, Rupali. N. Lohabade⁴,
Sakshi. G. Nandurkar⁵, Prof. Vanita Buradkar⁶**

Students, Department of Computer Science & Engineering^{1,2,3,4,5}

Guide, Department of Computer Science & Engineering⁶

Rajiv Gandhi College of Engineering, Research and Technology, Chandrapur, Maharashtra, India

Abstract: *Programming tools are important for programmers to develop software. If the developers have a good tool, it can help them develop system faster and more accurate. This paper proposed the Online Code Editor that was created for programmers or developers who want to write programs without any platform requirements or without any specific physical computers. It bases on web application running on the Private cloud computing. The features of the editor are performed on web programming languages, e.g., HTML, PHP, CSS, and JavaScript. The editor can isolate programming languages by highlighting syntax of programs. Users can create new projects and files, import and export files that they want on a server. Moreover, Save, Auto save, Delete, etc. are the additional functions of the editor. In this research of the text editor development, the open-source software called, "Ace" was used for some functions such as Undo, Redo, and Syntax highlight. The experimental results indicated that the proposed editor can be practically used on Private cloud computing.*

Keywords: online code editor, web-based tool, syntax highlighting, code auto completion, debugging.

I. INTRODUCTION

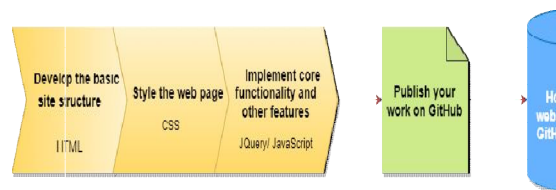
The online code editor is used to run multiple programming languages; however, many programmers who need to edit the source code urgently may not be able to access a convenient resource without installing any software on their computer or notebook. In our project, programmers can compile and run source code through a web browser, and the code is generated on the server. The output of the compilation will then be displayed in the client-side browser. Online code editors are designed to run on small resources such as PCs, tablets, Android devices, notebooks, and laptops. Online code editors are designed to run on small resources such as PCs, tablets, Android devices, notebooks, and laptops. Online code editors are designed to run on small resources such as PCs, tablets, Android devices, notebooks, and laptops. To edit program source code, computer operator must have at least one computer. If a programmer wants to run multiple languages, they must install the software for each one. However, if they use an online code editor, they can run all languages on the same platform, which eliminates the need for the programmer to install software for each language. Browser-based code editors and compilers significantly reduce both the hardware and software required by programmers when working on any given project by storing and executing source code online, allowing programmers and development teams to quickly begin projects for a wide range of platforms, devices, and operating systems. Code chef and code pad are two well-known online compilers. The generally accepted operation of our website is that users can write, or copy paste code into the Code editor and then use the submit button, which compiles the program in the backend and displays the result in the output window. In addition, users can add files that can be compiled using the backends' compiler.

II. CORE FEATURES ONLINE CODE EDITOR

- **Code Understanding or Code Completion:** The capacity of an IDE to know the keywords and the functions of a programming language is a must. The IDE may utilize this information to do things such as highlight typographic blunders, propose a rundown of accessible functions considering the proper circumstance, or offer a definition of a function from the official documentation
- **Proper Management of Resources:** When making applications, programming languages frequently depend on resources, like library or header documents, to be areas. An online IDE should have the capacity to deal

with these coding resources. It ought to be mindful of any needed resources so programming errors can be spotted at an early stage and not later.

- **Debugging Tools:** You need to be able to completely test your application before discharging a web IDE. The IDE may have the capacity to give you the value of the variables at specific points, unite them into distinctive information archives, or acknowledge diverse parameters.
- **Compile:** For programming languages that require a compile, the web IDE can decipher codes from high level programming languages to object code of a platform. Prerequisites for these elements fluctuate considerably from one language to another. This way, usually, a web-based IDE comes with one programming language or even a set of different types of languages. Some popular IDE editors are JBuilder for Java, Microsoft's Visual Studio for C# and Visual Basic, and more.



III. SYSTEM DISCRIPTION

User Interface:

The user interface (UI) is the visual representation of the online code editor that allows users to interact with the system. It includes components like menus, toolbars, code editor view, side panels, and output/console area. The UI provides a means for users to write, edit, and manage their code.

Code Editor Functionality:

The core functionality of the online code editor revolves around code editing. This includes features like syntax highlighting, code completion, code folding, indentation, error highlighting, code formatting, and navigation tools. The code editor allows users to write and edit code efficiently, providing tools to enhance readability and productivity.

File Management:

The online code editor includes functionality for managing code files. Users can create new files, open existing files, save files, rename files, and organize them into directories or folders. File management allows users to work on multiple code files within the editor and maintain a structured project.

Collaboration:

Many online code editors support real-time collaboration, enabling multiple users to work on the same code simultaneously. Collaboration functionality includes features like real-time synchronization, shared cursors and selections, and communication channels for collaborating on code projects.

Version Control Integration:

Integration with version control systems like Git is a common feature in online code editors. It allows users to commit changes, create branches, merge code, view commit history, and manage repositories within the editor. Version control integration enables efficient code management and collaboration with other developers.

Execution and Testing:

Some online code editors provide the ability to execute code within the editor environment. This includes features like running code snippets, executing scripts, and displaying the output or error messages generated by the code. Testing frameworks may also be integrated to facilitate code testing and debugging.

These components and functionalities work together to create a comprehensive online code editor system that enables developers to write, edit, manage, collaborate on, and execute code efficiently.

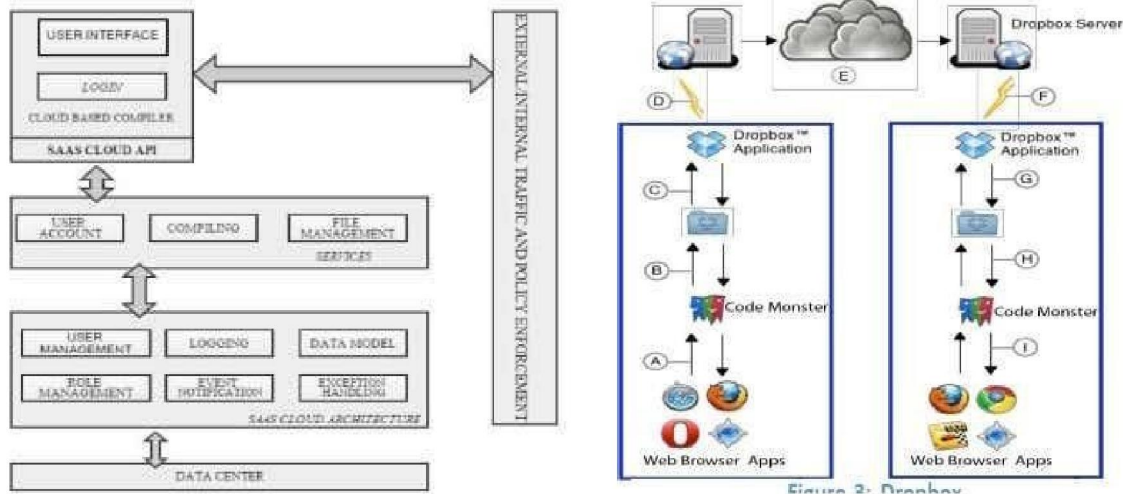


Figure 2: Dropbox

IV. METHODOLOGY

Requirements Gathering

The first step is to gather requirements by understanding the target audience, their needs, and the desired functionalities of the online code editor. This involves analysing user expectations, researching existing code editors, and identifying the core features and capabilities required.

Design and Architecture:

Based on the gathered requirements, the next step is to design the architecture and user interface of the online code editor. This includes creating wireframes, mock-ups, and design prototypes to visualize the overall structure, layout, and flow of the editor.

Technology Selection:

Choose the appropriate technologies and frameworks for developing the online code editor. This includes selecting programming languages, libraries, and tools for the frontend (client-side) and backend (server-side) development. Consider factors like performance, scalability, compatibility, and the desired features of the editor.

Frontend Development:

Develop the frontend components of the online code editor. This involves implementing the user interface, code editor functionality, syntax highlighting, code completion, and other code-related features. Use HTML, CSS, and JavaScript (along with libraries like CodeMirror or Monaco Editor) to create a responsive and interactive user interface.

Backend Development:

Develop the server-side components that handle user authentication, file storage, collaboration, version control integration, and other backend functionalities. Choose an appropriate server-side technology stack (e.g., Node.js, Ruby on Rails, Django) and develop the necessary APIs and services to facilitate communication between the frontend and backend.

Integration and Testing:

Integrate the frontend and backend components, ensuring seamless communication and proper functionality. Conduct thorough testing to identify and fix any bugs, issues, or compatibility problems. Test various scenarios, user interactions, and edge cases to ensure the online code editor performs as expected.

User Feedback and Iteration:

Release the online code editor to a limited group of users or a beta testing phase. Gather feedback from users, monitor system performance, and collect suggestions for improvement. Use this feedback to iterate on the editor, addressing issues, adding new features, and enhancing the user experience.

Maintenance and Updates:

Regularly maintain and update the online code editor to ensure it remains secure, stable, and up to date with the latest technologies and frameworks. Fix bugs, address security vulnerabilities, and provide ongoing support to users.

V. RESULT

The result of an online code editor is a user-friendly and accessible coding environment that enhances productivity, collaboration, and code quality. It provides efficient code editing features such as syntax highlighting, code completion, and indentation guides, making the coding process more streamlined and efficient. With real-time collaboration and version control integration, users can collaborate with team members, track code changes, and manage code revisions seamlessly. Additionally, the ability to execute code within the editor environment allows for quick testing and debugging. Overall, an online code editor empowers developers to write, edit, and manage their code more effectively, fostering a productive and collaborative coding experience.

VI. CONCLUSION

In conclusion, an online code editor offers a versatile and accessible platform for developers to write, edit, and manage code efficiently. It provides a user-friendly interface with features such as syntax highlighting, code completion, and indentation guides, enhancing coding productivity and readability. Real-time collaboration and version control integration enable seamless teamwork and code management. The ability to execute code within the editor environment allows for quick testing and debugging. With cross-platform compatibility and cloud-based hosting, online code editors offer flexibility and accessibility from any device with an internet connection. Overall, online code editors empower developers with a convenient and productive coding environment that fosters collaboration, code quality, and efficient development workflows.

REFERENCES

- [1]. M. Doernhoefer, "Surfing the Net for Software Engineering Notes", ACM SIGSOFT Software Engineering Notes, Vol. 24, No. 3, (1999), pp. 15–24.
- [2]. L. C. L. Kats, R. G. Vogelij, K. T. Kalleberg, and E. Visser, "Software development environments on the web", in Proceedings of the ACM international symposium on new ideas, new paradigms, and reflections on programming and software - Onward! '12, (2012), pp. 99.
- [3]. M. Goldman, "Role-based interfaces for collaborative software development", in Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology - UIST '11 Adjunct, (2011), pp. 23.
- [4]. F. Fröbner, "A Practice Theoretical Analysis of Real Time Collaboration Technology: Skype and Sametime in Software Development Projects", Göttingen: Cuvillier, (2008).
- [5]. S. Klein, N. Vehring, and M. Kramer, "Introducing Real Time Communication: Frames, Modes & Rules", in Proceedings 23rd Bled eConference eTrust: Implications for the Individual, (2010), pp. 591–606.
- [6]. Bau, D.A. Droplet, ABlock-Based Editor for Text Code.journal of computer science in colleges.30,6(Jun. 2015).