

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Survey on Web Based Interface

Poreddy Jayaraju¹

Master of Engineering

Department of Computer science Eng.

Sri Vaishnav Instt. Of Tech. &science Indore.

RGPV University Bhopal. Madhya Pradesh, India.

Vijay Prakash²

Assistant Professor

Department of Computer science Eng.

Sri Vaishnav Instt. Of Tech. &science Indore.

RGPV University Bhopal. Madhya Pradesh, India.

Abstract: web based interface is an application which provides facilities to programmer for software development such as code completing and fixing, source code editing and management, automated testing, etc. Software is rapidly moving from the desktop to the Web. The Web provides a generic user interface that allows ubiquitous access, instant collaboration, integration with other online services, and avoids installation and configuration on desktop computers. Moving IDEs to the Web is not just a matter of porting desktop IDEs, a fundamental reconsideration of the IDE architecture is necessary in order to realize the full potential that the combination of modern IDEs and the Web can offer. This paper discusses implementation of Web based interface environment for compilation and execution of codes written in different languages like C, C++, C#, VB, Java, Perl, Python, Ruby, HTML, CSS, Java Script languages. Users can edit, write, compile, debug and store their code on server. Users need not to spend their time for finding and installing an IDE for different languages. User can use IDE in any device like PC, tablet and mobile devices which has browser with internet connection. Web based interface can be used in low configuration systems also

Index Terms: web based interface, Web, Compiler, programs.

I. BACK GROUND

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs have intelligent code completion. IDEs initially became possible when developing via a console or terminal. In the early hours systems could not support one, since programs were prepared using flowcharts, entering programs with punched cards (or paper tape, etc.) before submitting them to a compiler. Dartmouth BASIC was the first language to be formed with an IDE (and was also the first to be designed for use while sitting in front of a console or terminal). Its IDE was command-based, and therefore did not look much like the menu-driven, graphical IDEs prevalent at 2015. However it integrated editing, file management, compilation, debugging and execution in a manner steady with a modern IDE. Maestro I is a product from Soft lab Munich and was the world's first integrated development environment 1975 for software. Maestro I was installed for 22,000 programmers worldwide. Until 1989, 6,000 installations existed in the Federal Republic of Germany. Maestro I was arguably the world leader in this field during the 1970s and 1980s. Today one of the last Maestro I can be found in the Museum of Information Technology at Arlington.

A web IDE can be accessed from a web browser, such as Google Chrome or Internet Explorer, allowing for a portable work environment. A web IDE does not usually contain all of the same features as a traditional, or desktop, IDE, although all of the basic IDE features, such as syntax highlighting, are typically present.

A web IDE, like most websites, is commonly composed of two pieces: a frontend and a backend. The frontend is usually written in Java script, using AJAX methods to communicate with the backend using a HTTP API, although in some cases, a browser extension or desktop application serves as the frontend and communicates with the backend without the need for a

browser. The backend takes care of creating, saving, and opening files, as well as running any terminal commands if the IDE supports it. This setup allows for portability and continuity. The state of the IDE can be saved and reopened on another machine or machines and can be used to give the developer more control over the development environment [25].

II. INTRODUCTION

The goal of this project is to develop a web based Integrated Development Environment (IDE). The client only needs a web browser and an internet connection. By connecting to the server, an IDE looking page is downloaded by the browser that looks like and has the functionality of an IDE environment. The client is able to access previous code stored in the database or create new projects, compile the code and run it. Also, the client is able to download the executable output of the compiler from the server to local disk.

IDEs are designed to maximize programmers' productivity. They normally achieve this goal by consisting of a source code editor, a compiler and/or interpreter, built-in automation tools, and a debugger. Some modern IDEs even employ plug-in frameworks that support extension to the environment, hence meeting various needs of programmers. [21].

Software is moving from the desktop to the Web. Online services are rapidly replacing traditional downloadable software products. Based on the latest developments in Ajax technologies, vastly improved JavaScript engines, and the introduction of HTML5, there is now even a small but growing collection of browser-based code editors. Fully fledged integrated development environments (IDEs) are still lagging behind in this pull towards the Web. Modern, desktop-based IDEs integrate a wide range of software engineering tools, and provide a platform for writing, maintaining, testing, building, running, debugging, and deploying software. They increase developer productivity by incorporating many different kinds of editor services specific to the syntax and semantics of a language. These services assist developers in understanding and navigating through the code, they direct developers to inconsistent or incomplete areas of code, and they even help with editing code by providing automatic indentation, bracket insertion, and content completion.

The integration of complete tool suites for software development and the development of language-specific editor services took a tremendous effort for the current generation of IDEs such as Eclipse and Visual Studio [19]. There are many IDEs available for programmers, such as Eclipse, Net Beans, maces, J Developer, text mate and intellijs. However, these IDEs are desktop-based applications. These desktop IDEs take some time in loading the entire application and opening the editor. To avoid all these complications, Web-based IDEs are emerging which make life easier for any software developer [22].

This paper discusses the implementation of Web Based Integrated Development Environment (IDE) for the different languages to code, compile and run the code. The Web based interface will allow easy development and testing of applications. The users have the privilege to register on to the system write, save and manage programs online. After the language is chosen by the user the request is forwarded to the respective compiler. Multiple users can write programs in different programming languages and also can compile and run the program. a source code editor, a compiler and/or interpreter, built-in automation tools, and a debugger Some modern IDEs even employ plug-in frameworks that support extension to the environment hence meeting various needs of programmers[21].

Why Web based interface.

Software development is an important activity in today's world. In old days, programmers used to write codes into the text files and then by using compiler and similar tools which are command line based, these written codes were turned into software programs. As the computers evolve, size and the complexity of software production increased. With this increasing complexity, accomplishing tasks such as code editing, build automation and debugging started getting more and more difficult. Solution for this problem of programmers is found to be Integrated Development Environments which are commonly referred as IDE's. Although IDE's are life saver for programmers, these software applications have couple of drawbacks.

Local systems IDE's are installed on a system and one need to use that computer to use features of IDE and develop the software. Stand alone IDE high computer resources, as IDE's supported more facilities to the programmer they require much more computer resources, especially memory and CPU, which may not be available all the time. Most of the desktop based IDE's require the development environment to be set up on their machines.

This development environment requires language specific integrated development environment like eclipse or visual studio to be downloaded and configured within the user's machine. If the user decides to work on a different machine the entire development kit and IDE has to be installed in the new system which makes the process tedious and extremely inconvenient. Web based interface thus provides a solution to the given problem and gives user the flexibility to start a web browser and open his/her project.

The basic requirement here is that the user must have access to an internet connection to be able to connect to the Web IDE. We will install all programming development environment on a server.

Important questions developers might have such as: Which of my team mates worked on this piece of code before? What other parts has this person changed in the past? How many of the co-developers are working on the project code at this moment? Who is modifying what part of the system? Can I get real-time feedback on the changes they are making to the code? While answers to these developer questions may be available in the minds of certain team members, the underlying knowledge is often left implicit and unavailable to other team members [23]. Most of the reasoning leading to a particular piece of code gets lost, leaving only the new code itself as result of a complex program comprehension process.

This leads to a number of research hypotheses. The first is that making the knowledge leading to code modifications explicit, and sharing it among all team members, will lead to a significant increase in productivity and software quality. Web-based IDE encourages live collaboration between team members, because its structure requires all documents and codes to be modified and saved on the servers, which will release the problems of the implicit code knowledge. The most exciting advantage of web-based IDE is that, with the new generation of smart phones and PDAs that support AJAX and Java programming jobs can be done movably anywhere anytime.

Following points justifies the importance of Web based interface System:

1. Available 24×7 days on anywhere.
2. Access from anywhere or any computer (with Internet access).
3. Minimal configuration needed (or only needed once).
4. Centralized workspace.
5. No need to install a lot of software locally.
6. Allows for development from inexpensive machines, such as mobile phones.

III. RELATED WORK

Many efforts have been made to implement online compiler and runtime environments in past few years. In this section we briefly discuss recent developments. Codeine [24]. It seems exciting in the beginning. However, when writing something more complicated in its editor, it fails miserably. For example, it only supports simple statements in C++ no other than "print", but not the core concepts of Object-oriented programming. Python Fiddle [7] is a code editor and code execution environment that allows programmers to run snippets and debug scripts on the go. It supports a plethora of third-party packages, boasts superb documentation, comes with a wide array of built-in hot keys and is also open source to boot. Code run Studio [2] offers users a cross-platform tool for writing ASP.NET, JavaScript, C#, HTML and CSS. Its default Visual Studio compatibility is a nice touch and should have Microsoft-focused coders feeling right at home. It comes equipped with the usual bells and whistles like

code completion and syntax highlighting. Remote Application Platform [1] Project is an open-source software project under the Eclipse Technology Project which aims to enable software developers to build Ajax-enabled rich Internet applications by using the Eclipse development model, plugging and a Java-only application programming interface (API). It can be considered a counterpart for web development to the Rich Client Platform (RCP). Source kit is a lightweight, browser-based alternative to bloated desktop development suites.

Supported languages include all the majors you'd expect, such as C/C++/C#, PHP, Python, Java script, Java and Ruby. Odin [3] allows developers to code in PHP, Python, Perl and Java script while working with popular frameworks like Jingo, Ruby on Rails and Node.js. We Scheme [6] is an educational programming environment, embedding Code Mirror [4] for syntax highlighting and bracket matching. However, these can useful tools for coding small program; they do not provide a comprehensive environment with all the facilities that are especially important for productivity in larger projects. They also do not offer any support for collaboration. Another IDE, specialized to Iron Python, is provided by Void Space, and uses Silver Light for its implementation [8]. There is currently one open source initiative for creating an extensible IDE for the Web, allowing developers to add new components using JavaScript. The Cloud9 project [9] integrates the Mozilla Sky Writer [10] and ACE editors, and provides a plugging based IDE architecture in HTML5 and JavaScript. Compile online [17] provides more than 50 language compilers online but does not provide facility for store the program. Using compiler [18] we can learn programming languages and execute programs online. An Online Programming Tutors named Problems by Kumar [11] is designed as a tutoring system for students to learn Java, C++, and C# OO programming.

It mainly covers Java programming and basic programming constructs. In Cloud9 IDE [9] front end is all Java script, while the back end relies on the popular Node JS framework. It has syntax highlighting for C#, C++, Python, Perl, Ruby, Scale and a few others as well. Code anywhere [20] is a code editor in a browser with an integrated ftp client, and all popular web formats are supported (HTML, PHP, JavaScript, CSS, and XML). Cloud IDE [5] is a solid Web IDE which supports the usual languages like Java script, Ruby, Groovy, Java and Html.

[Erbil](#) [26] if you think JavaScript is not so popular among the developers; you might be unaware about the current scenario. Today, JavaScript has gained immense popularity and respect from developers across the globe. Erbil is all about JavaScript — it provides tools to create JavaScript applications for online business productivity. [Orion](#) [27] — Orion brings Eclipse's considerable experience to the Web IDE. It is extensively used these days for front-end development, which restricts it to the HTML and JavaScript. , its team is trying hard to improve it and you can expect a lot more features in the near future. [Ideate](#) [28] this is on online compiler and debugging tool that allows you to compile source code and execute it online in more 60 different programming languages. Choose a programming language of your choice, enter the source code

IV. CONCLUSION

In this paper we have discussed Web based interface from the stand alone desktop to the Web. The proposed research questions arose from placing ourselves in the seat of the software developer who already develops for the Web, but now wants to transition his daily development activities to the Web, and take advantage of the trademarks of the Web, pervasive collaboration, zero deployment, instant-access from anywhere, and huge computational resources. We discussed how we can move the software development environment from one paradigm (the desktop) to another (the Web). This can be used for E-learning platform that lets students write, execute and test programs entirely in their browse. With Web based interface allows the user to write and manage their programs on the server. The programs stored on server, the compilation of the programs will be managed by the server by forwarding the request to the required processor. Based on the programming language in which program/code is written and sends that program/code to the respected compiler. The proposed system showed how Web based interface can be used to eliminate the problem of storage. Many more applications are possible when taking into account.

References

1. Remote Application Platform Available: <http://www.eclipse.org/rap/>
2. Code runs Studio Available: <http://coderun.com>.
3. Coding Available: <https://koding.com/>
4. Code Mirror Available: <http://codemirror.net/>.
5. Cloud IDE Available: <https://codenvy.com/>
6. D. You, E. Schnauzer, S. Krishnamurthy, and K. Filer, "We-Scheme: the browser is your programming environment."
7. Python Fiddle <http://pythonfiddle.com/>
8. The Eclipse Foundation. Void space python online Available: <http://www.voidspace.org.uk/ironpython/silverlight/index.shtml>.
9. Cloud9 IDE. <https://c9.io>.
10. M. Labs. Mozilla labs: Skywriter.
11. Problems by Amrita N. Kumar <http://problets.org/>
12. Suryawanshi Hershel, Roader Chakrapani Ambhore Ajay, Rathod Sharad, "Compiler as Service over Cloud", International Journal of Computer Applications (0975 – 8887) Volume 70– No.1, May 2013
13. An sari Mohr. Arshad, Khan Arshiya, Shaikh Sana, Mirza Zainab, "Compilers on Cloud", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 9, September - 2013
14. M.Pabitha, T.Selvakumar, S.Punitha Devi "An Effective C, C++, PHP, Perl, Ruby, Python Compiler using Cloud Computing", International Journal of Computer Applications (0975 – 8887) Volume 69– No.7.
15. Mayank Patel "Online Java Compiler Using Cloud Computing", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-2.
16. A.Rabiyathul and Basariya k. Tamil Selvi, "Centralized C# Compiler Using CloudComputing", IJCE Volume 06– No.6, Issue:02, March2012.
17. compile Online IDE Available: <http://www.compileonline.com>.
18. Online Compiler Available: <http://www.Compile.com>.
19. Lennart C. L. Kats, Richard Vogelij, Karl Trygve Kalleberg, Eelco Visser "Software Development Environments on the Web: A Research Agenda"
20. Codeanywhere Available: <https://codeanywhere.net/>.
21. Jingwen Ou Mahdi Tayarani Najaran Mushfiqur Rouf *Aurora SDK: A Web Based Integrated Development Environment*.
22. Web based interface for Interfacing View Controller by Tejasvi Palvai.
23. Adinda: A Knowledgeable, Browser-Based IDE Martin Pinzger Deursen , Arie van Deursen ,Anja Guzzi Delft University of Technology.
24. code ide <http://codeide.com/>
25. Wikipedia. Integrated development environment. http://en.wikipedia.org/wiki/Integrated_development_environment
26. _Erbix ide <http://Erbixide.com/>
27. Orion Available: <https://orion.com/> [28]Ideone Available.

AUTHOR(S) PROFILE



Poreddy Jayaraju has Received The B.Tech Degree In Computer Science Engg From MNR College Of Engg And Tech In 2012, Affiliated To JNTU Hyderabad. He Is Also Pursuing Master Of Engineering In Computer Science From SVITS College, Indore affiliated to RGPV University Bhopal, Madhya Pradesh, India. His Research Interests Include Database Management System, Object Oriented And Web Technology.



Vijay Prakash An Assistant Professor in Computer Science & Engineering department, working in SVITS college, Indore since 7 years. He has academic experience of UG & PG Courses at SVITS. He believes in ethical values to contribute his Technical & Professional skills to the society. Coordinator of Computer Society of India CSI-SB, SVITS since 2011. Working as Coordinator of Indore Chapter since 2012 and also he Got National award by CSI in 3 categories at 49th Golden Jubilee year celebration.