Web based Code Editor A revolutionary coding platform

Dr. Dhananjay Maktedar[1], Kamani Monika[2]

[1]Professor, Computer Science and Engineering, Guru Nanak Dev Engineering College Bidar-585401, Karnataka

[2]PG Student, Computer Science and Engineering, Guru Nanak Dev Engineering College Bidar-585401, Karnataka

Abstract— **The Internet is expanding quickly, and many desktop applications are already being developed for the online. The Internet makes it simple to access several programmers from any location at any time. To design their applications, developers need many tools, including a code editor. In order to facilitate user collaboration while working on the project, the goal of this research is to design and construct a real-time code editor application employing web technology. This application offers the feature that allows users to work in a real-time project. Web code editor gives users a workspace where they can write code, run it, see the results in the terminal. The primary functions of the application include workspaces for creating, running, and building source code. The programming languages C, C++, Python and JAVA are supported by this application without worrying about the environmental setup.**

Keywords—**Web; Compilers; Editors; React; Web Editor; Web Technology; IDE; Code editors; JavaScript**

I.INTRODUCTION

The revolution of the world of information through the means of internet has been more than ever before. The internet has brought soul to the IT world. This has set the global possibility for the unparallel level of capability integration. The information can be accessed from any geographic location. With this setup, one should also be able to create his computer programs without any hassle of environment setup, package management etc.

The Internet creates the stage for sustained commitment to develop applications which help the people all over the world. This leads to a greater commitment to this research and development of IT infrastructure. Under this the Web Code editor is focused on writing the code on web without any third-party software or package download. The proposed code editor is focused to working with various major computer programming languages like C, C++, Java, Python etc. This makes easier for the user to develop the logic on the go.

This project not only targets PC/Laptop user but is also made to suit the Tablet and the mobile user through its Responsive UI and improved UX design. The concept of Code aesthetics, indentation, line breaks, line numbers are the key editor features.

The extension to this code editor is the code executor. The code written can then be executed on the web itself without any need for installing development kits, compilers, servers, runtime environments etc. The Project gives a feel of an IDE by the integration of the compiler to this code editor which is implemented as am extension.

This project used the core concepts of Web development i.e., frontend development for the development of code editor on the client side and the backend development for the compilation process. The compilation of the code is done at the backend where the actual execution of code is done on the compiler setup for the major programming languages.

This revolutionary idea can help the educational institutions or the individual code learner in trying out their code on the go without thinking of the environment setup. This makes the web app config free which leads to tense free learn environment.

II.REQUIREMENT ANALYSIS

A. *Software Requirement Specification*

- Online Compiler and code editor’s main objective is to implement Code without installing compiler in the system, directly the code can be compiled and run on the browser. An online compiler is revolutionized version of the conventional compiler with all the features of it inherited.
- However, The only observable difference is the working environment, here all the data is executed at the client side through a web browser. Thus, we

aim to develop a website where users can write programming code and paste into the code editor and compiler and then execute it.

- Under this section, an SRS document containing the details like the purpose of the application, functionality of the application is described. This document is can also becalled by various names like software document, SRS document etc. This consists of the scope, purpose, functional and nonfunctional requirements, software and hardware requirements of the project.

B. Purpose of the Project

The main agenda of this document is to build an online web code editor which is focused on writing the programming code on web without any third-party software or package download. The proposed code editor is focused to working with various major computer programming languages like C, C++, Java, Python etc. This makes easier for the user to develop the logic on the go. The Backend consumes the CPU Process Spawn to execute the code. The Editor also focuses on various editor features like indentation, line breaks, line numbers etc. The Web UI comes with various language support and also theme support. The application is highly responsive that this can be used on any browser. The latest frontend library React plays a major role in getting the right feature out.

C. Intended Audience

The Primary audience or the consumers of this application are the programmers or the code learner who are willing to work on the program logic then the configuration. The web editor provides on the go code editor which can be used any time on any web browser supported across various screen size. It would also be helpful to the school students who are excited to learn coding but are beginners to setup the coding environment.

D. Application Functionality

The following are the major functionalities prescribed for the project idea.

- Syntax Highlighting
- Keywords Suggestion
- Auto-Indentation for Python
- Line Numbers
- Multi Language Support
- Multi Theme Support

E. Application Features

The following are the application features which are mainly focused for the project idea.

- Portable: The proposed application can be consumed at any time on any environment irrespective of system software or hardware specification.
- Ready to Use: No extra configuration, plugin or installation is required.
- User friendly: User can easily operate the application as it is developed under the user experience consideration as the primary intent.
- IDE-Functionality: The proposed application only uses native resources and do live reload.
- Performance: User can execute the code at faster response time.

F. System requirements

- Software Requirements:

1. Operating System: Linux Operating System
2. Frontend Technologies: ReactJS, NextJS
3. Code Mirror Package
4. Nginx Server
5. NodeJS

- Hardware Requirements:

1. Processor with min. 1.2GHz Clock Speed
2. Min 512 MB RAM
3. 8GB Storage Space

III.SYSTEM DESIGN

The architecture for the proposed system requires a highly performing backend server with a minimum of 700 MHz clock speed. This server runs on the Linux based operating system which gives the facility to establish a cloud compiler server. Linux OS like Ubuntu, Fedora etc. can be used. The greater the performance of the CPU greater the compilation speed. The design is majorly divided into following layers:

1. Frontend Layer
2. Backend Layer

The Layers of our system architecture is as shown in Fig. 1

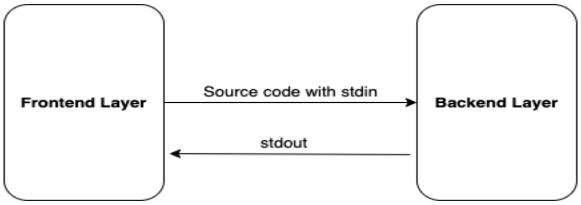


Fig 1: System architecture

G. Frontend Layer

The presentation layer of the system i.e., the user interface of the system is the playground for the user to give their inputs. The code editor with the functionality of Syntax highlighting, Line Number, Auto indentation etc. are designed to take the user input. User input here resembles the source code and the standard input.

H. Backend Layer

The backend here refers to the NodeJs server from where the compiler processes are spawned, and the execution is initiated. The Linux server with the compilers of various high level programming languages processes running, the NodeJs root module calls these processes and executes the program. The flow is as shown in Fig. 2.

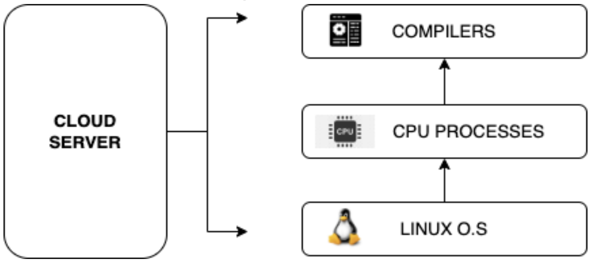


Fig. 2: Backend design

IV.DATA FLOW

The main agenda here is to compile the program written in various language over web without installing any compilers or packages. This can be achieved by creating a remote server with the capability of executing the programs and the User interface using which the user should be able to write his logics.

The flow of data through the multiple layers is as discussed below:

1. User selects the language and writes the code on the Web UI.
2. User gives standard input if required after clicking on the execute button.
3. An API Call is made to the remote server with the source code and standard input.
4. The Server fetches the compiler process available. If the process is ready to pick the input the code is compiled and executed. If the process is not ready, then the HTTP 503 error code is sent back to the user.
5. Once the program is executed and if it contains any kind of Compile time errors then the error stack trace is sent back to the user. If not, the Program output is sent back to the user. The flowchart for the system can be seen in the Fig. 3

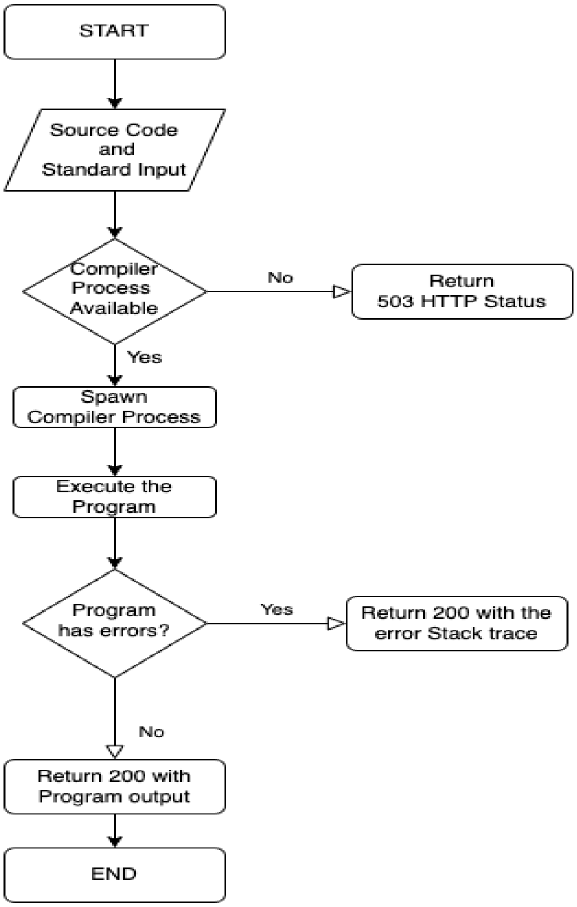

START
Source Code and Standard Input
Compiler Process Available
No
Return 503 HTTP Status
Yes
Spawn Compiler Process
Execute the Program
Program has errors?
Yes
Return 200 with the error Stack trace
No
Return 200 with Program output
END

Fig 3 Data Flow Chart

V.USER INTERFACE DESIGN

The User interface majorly consists of the coding space where the user writes his code. This coding space will take care of the user experience by providing him/her with the various code editor features like auto-indentation, Syntax highlighting, Line numbers, Block programs etc.

The navbar is designed in such a way that the language and the theme config should be accessible to the user easily. Under the more options feature in the navbar the use can create multiple sessions and maintain different tabs. The selected language and the theme are also displayed on the navbar for quick tooltip. The execute button at the bottom right of the screen is used for the execution purpose. Other major options like export file, Reset code and change config are also available at the button dropdown. This application should easily handle various screen sizes and resolution through its responsive design.

The design for Web UI and Mobile responsive UI can be seen in Fig 4. And Fig. 5

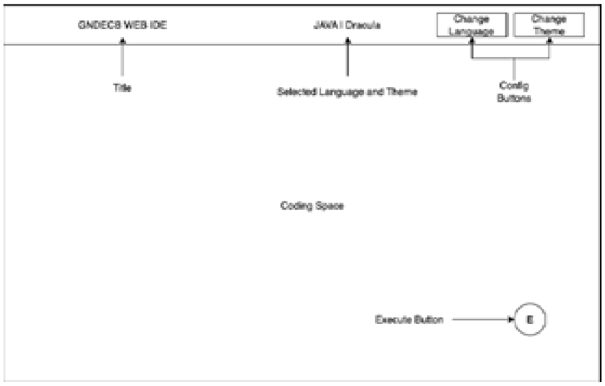


Fig 4: Web UI Design

The responsiveness of the application is also important so that the user can access the application using various devices of different screen resolution.

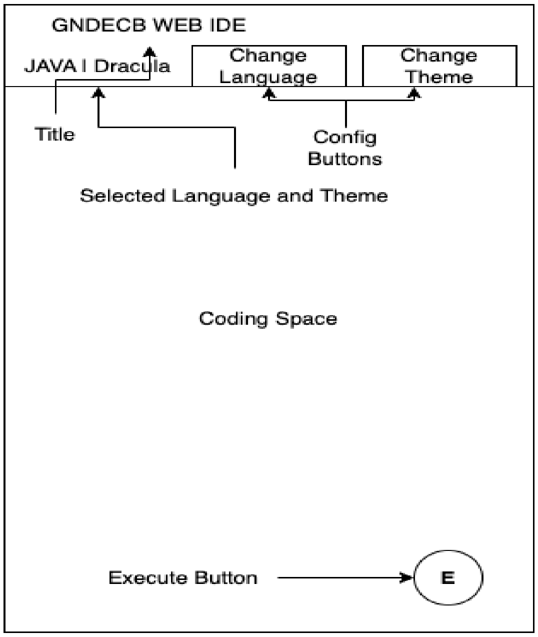


Fig 5: Mobile UI Design

VI.TESTING

The application needs to be tested at multiple level of development and post development. Unit level of testing should make sure that the each React component is rendered properly on successive flow of events like onClick, onView, onScroll etc. It's the developer's responsibility to check each component with check the props sent and the state initialized at the unit level.

I. Unit and Integration Testing

The API call at the backend is the key functionality of the backend server. The working of backend is necessary for the program execution. It should be tested by with various possible scenarios. The API should be called with proper request body containing source code, standard input, language and authToken to check the following scenarios.

a. The server should return 503 if the server couldn't find the child process for a programming language.

b. The server should return 200 with the error stack trace in the standard error param if the program executed has compile time errors.

c. The server should return 200 with the program output if the program is executed successfully.

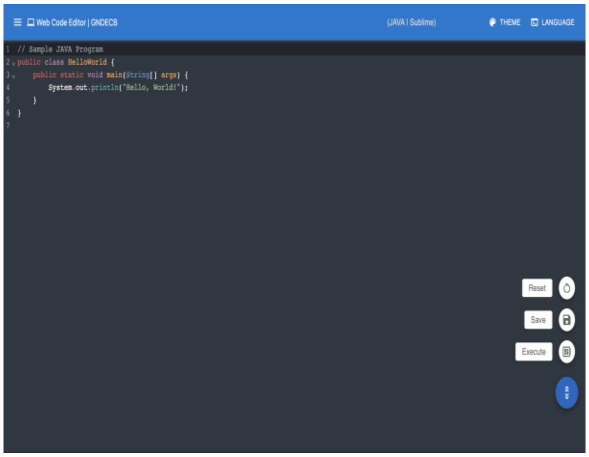
d. Performance testing should be done to check the integrity and the capacity to handle the multiple API calls.

e. The server should return 400 if any request param is missing in the request body.

f. The server should return null in the standard input if the server is down.

VII.INTERPRETATION OF RESULTS

As reported in the earlier chapters, The design and implementation of the proposed application. The process from requirement analysis to the testing part, we conclude here to the result part. The output can be discussed under the test environment along with sample outputs with the help of screenshots captured. The initial homepage of the web editor portal is as shown in Fig. 6



```
// Sample JAVA Program
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Fig 6. Editor Page

VIII.CONCLUSION

As shown in the research study, this application can be a very essential tool for the beginners who is eager to learn coding and should not worry about the complex environment setup like compiler installation, package management etc. This tool could be useful to the educational institutions to make their students focus on learning the programming the algorithms or code logics then the project setup or the environment setup.

This application helps the coder with various key features like Syntax highlighting which help the user in identifying the identifiers like keywords, variables etc. It also helps with the features like line numbers, Auto-indentation, Blocks etc. This has the support for various major programming languages like JAVA, C, C++, Python, JavaScript. It also has the multi theme support where one can change the color code for the identifiers of their choice.

J. Future enhancements

The following are the some of the enhancements that can be made which can bring some more features to the application.

- Currently the support for 5 programming languages is present. More languages can be integrated.
- Currently multiple sessions can be created in one go without saving any data. Save code feature can be implemented.
- Multiuser programming which will enable multiple programmers to code on the same session.
- Add support for more themes.
- Live interaction with the console of the compiler child process can be implemented to include live console interaction and avoid single stdin execution.

ACKNOWLEDGMENT

The satisfaction and euphoria that accompany the successful completion of any work would be incomplete without mentioning the people who have made it possible, I would like to thank Dr. Dayanand J, for giving us the guidance and encouragement to complete our project successfully. It gives us immense pleasure in expressing our heartfelt gratitude for boosting our confidence and standing with us in our tough times.

REFERENCES

[1] CodeR: Real-time Code Editor Application for Collaborative Programming by Christine Soesanto, Bina Nusantara University, Indonesia

[2] Browser based Code Editor by Terna Engineering College, Mumbai

[3] Portable Code Compiler published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 |Issue-4, June 2020, pp.135-138

[4] A Study and Analysis of Compiler Optimization by Authors: Prathibha A. Ballala, Dr. Sarojadevi H.

[5] Compiler and Runtime Support for Programming in Adaptive Parallel Environments Published as: University of Maryland Technical Report CSTR-3510 and UMIACS-TR-95-83

[6] Banking Interprocedural Compilation of Irregular Applications for Distributed Memory Machines Published in: Supercomputing '95, Department of Computer Science Technical Reports CR-TR-3447

[7] Interprocedurally Partial Redundancy Elimination and Its Application to Distributed Memory Compilation Published in: Conference on Programming Language Design and Implementation '95