

ONLINE COMPILER USING CLOUD COMPUTING

Ms. Anchal Singh
anchalsanjay.singh@pace.edu

Mr. Tushar Rakholiya
tushar.m.rakholiya@pace.edu

Mr. Sanath Gholap
sanathsadanand.gholap@pace.edu

Mr. Ramesh Kyasaram
rameshtamanna.kyasaram@pace.edu

Ms. Harshada Chaudhari
harshadaanil.chaudhari@pace.edu

Mr. Brandon Mercado
brandon.c.mercado@pace.edu

Seidenberg School of Computer Science and Information Systems
Pace University, New York, NY, USA

ABSTRACT

In today's fast and competitive world everything is available on internet, and on web. So, we developed an online compiler using cloud computing. The main objective of this project is to develop a centralized compiler that helps to reduce problems like portability storage, cost, and space. It is the most convenient tool to compile code, remove errors and code. Moreover, we can run the web-based application remotely from any network connection that is independent of platform [6]. The challenge of installing a compiler on each machine is also avoided and therefore, these all benefits make this application suitable of cloud based online compiler make it suitable for performing online exams.

Keywords-- Centralized compiler, Cloud computing, Online compiler, Multiple Language Support, and Time Complexity.

I. INTRODUCTION

The idea for the Online Compiler came out of a need for a tool to give a successful use of web-based interface to compilers [3]. A compiler physically requires a lot of space and time to manually configure and install on each system. Although all necessary software is installed in user's computer, most users prefer to work on their computers at home or workplace and connect to the organization network. That situation creates unnecessary burden either for the network administrators who must install additional software on many machines of non-standard configuration, or in case of students who must purchase and install on their own several software packages along their full course of study.

When a program is compiled, it becomes platform dependent. Evidently, it is also not easy to carry the same program code to multiple machines if it does not permit the usage of a single system. Another drawback reveals that we would need to install different compilers for different programming languages on which its task is desired. These challenges impose the online compiler for multiple programming languages [4]. To solve the problem at least partially in the area of programming, we

proposed an approach of online compiler using cloud computing which allows the programmer to compile and store their code online. They can also create, open, save, delete and download their source and compiled code based on online.

In this project, we are including most demanding programming language compilers by the software developers mainly C, C++, Python, Java and C-Sharp. This web-based software will contain a system that has three text areas for entering source code, for entering pre-defined input and to get the output. The user would be given an option to select the languages in which he/she wants to write the source code and compile it. The software compiles the program and returns the output to the users.

II. LITERATURE REVIEW

Following research papers are the few of the research paper where we discovered the literature related to cloud compiler of different languages and are described below.

A. Science of Computer Programming: The basic concept of compiler is to produce machine understandable code from the human readable code (source Code). Even though the source code is free of syntax errors and compiler is generated successfully, it may still not do what is expected to do. Therefore, we must create test inputs for finding semantic errors while applying the inputs to the compiler for target expectation.

B. Multi-language cloud-based compiler: The author primarily focused on the compiler which executes the codes written in multiple languages. Traditionally you must install the compiler in the system for respective language and then execute the code.

Compiler for the languages are provided by the cloud because it is a centralized compiler. Everyone loves to get work done faster without spending money and without losing much space. The author made it all

possible by centralized compiler by saving the time and work of installing the compiler on the physical machine [6].

The portability is the main intention of the product. The web application can be supported by all the browser, so the developer has to use such libraries, files, and tools. The cloud compiler must provide the compatible platform to compile and execute the code in any browser with respect to the necessary constraints and dependencies.

III. PROJECT REQUIREMENTS

A. Functional Requirements:

- The primary function of the system is to take the code and input from the user for compilation.
- Users must select the compiler in which he/she wants to compile the code.
- After compilation, it returns the appropriate output if the code is executed successfully.
- Error message will be displayed if there is an error while compiling the code.

B. Usability Requirements:

- User interface will be used to compile the code as well as give the input to the code.
- The output window will be used to display the output of the compiled code.
- The login and registration are not required to compile the code
- If the user wants to create and save the project, the user must register.

C. Technical Requirements:

- The compiler is a web-based compiler and so user interface will also be web based.
- Web-based application would be developed in JavaScript where the backend of the application which is the compiler will be deployed in the cloud.
- The backend will be built in the docker containers where the individual languages will have individual containers, and once the compiler is requested, the container will be allocated, and the container will be destroyed after the work is done.
- The plugins of the compiler are going to be used in the container for the compilation.
- Database connectivity is required when a unique identification is created for registration of the system and when user create, open, and delete the existed project in the system.
- The User table stores the registration and login data.
- The Saved Table is going to be accessed by the File Module for the access to the stored/saved code.

IV. TEAM ROLES

A strong, committed team can be behind each exceptional item or application. Parts and tasks were carefully accumulated as specialties to each member of our six-person team, in a way that played to each person's qualities, in order to express the prerequisites drawn within the previous field. While the team's tasks did see some interaction at times in order to provide the best possible results, the group members tended to have unexpected bonding in working together, interacting with one another at all times, grasping and obliging one another's thoughts, and making a difference one another at all times.

Harshada Chaudhary:

Frontend Developer/Designer

- Integrated HTML/CSS web pages in Flask
- Connected user database with Flask.
- Designed user interactions in web pages.

Anchal Singh:

Product Owner/Project Manager

- Defined stories and prioritized the product backlog.
- Evaluated product progress at each iteration.
- Maintained and authored technical documentation and GitHub.

Ramesh Kyasaram:

Frontend Developer/Tester

- Designed web page layout in HTML/CSS.
- Tested different compiler modules.
- Ensured quality of the developed product.

Brandon Mercado:

AWS Cloud Engineer, Solution Architect

- Designed, managed and maintained technological duties associated with AWS Cloud.

Sanath Gholap:

Scrum Master/Tester

- Conducted weekly standups with team members and delegated tasks.
- Assisted team in documentation, handling GitHub related tasks.
- Tested compiler modules and their integration with frontend.

Tushar Rakholiya:

Full stack Developer (Lead)

- Researched and implemented backend technologies.
- Monitored servers and databases for functionality.
- Ensured responsiveness of application.

V. SOFTWARE REQUIREMENTS

A. Python Flask: Flask is a web framework written in Python. It does not require any tools or libraries. Hence, it is considered as microframework. It has no database abstraction layer, type validation, or any other components where common functions are given by pre-existing third-party libraries. We developed our webapp using Python (Flask) with HTML and CSS for front-end. We imported the programming language plugins for the compilation of the code in Flask.

- **Why Python (Flask)?**

When it comes to creating web apps, Flask offers developers a lot of options. It provides resources, libraries, and mechanics that allow you to create a web application, but it doesn't impose any dependencies or tell you how the project should look.

B. Amazon Web Services (AWS): Amazon Web Services (AWS) is a secure platform for cloud services, providing computing resources, storage of databases, distribution of content and other features to help companies scale and expand. In this project, we will use AWS to create containers in which our webapp will be hosted. As the number of connections to the webapp exceeds the limit, new container will be automatically created to handle the connections. We used AWS services such as S3 bucket [7], EC2 [8], DynamoDB [9], IAM [10], Authy and Boto3 for Storage, Hosting, Database, User roles, Security and Integration purposes respectively.

- **Why Amazon Web Services (AWS)?**

AWS is designed to provide benefits such as Easy to use, Flexible, Cost-effective, Reliable, Scalable and high-Performance and Secured. Also, it provides low-cost migration services so that your existing infrastructure can be seamlessly moved over to AWS.

C. MongoDB: MongoDB is a document-oriented database that stores complex schema data in JSON-like records. This means that you can store the information without caring about the data structure, such as the number of fields or the type of fields to store the values. MongoDB is very much easy to use as it is a schema less databases. In this project, we used it primarily to store codes. In addition, we used saved table to store registration and authentication data. Also, we created tables to store user details and saved code.

- **Why MongoDB?**

MongoDB is schema-free, allowing you to create documents without having to define the structure of the document first. These documents can be easily changed by adding or deleting fields. In some cases, MongoDB performance is improved over MySQL because MongoDB does not use joins to connect data, improving performance.

D. DynamoDB: A fully managed proprietary NoSQL database that allows us flawless accessibility and seamless scalability.

- **Why DynamoDB?**

As we used AWS to create cloud containers for our project, hence we decided to use DynamoDB by AWS as our primary database. We used this database onwards for further database related operations. In addition, DynamoDB supports migration from another database. So, instead of creating all the database tables that we already created using MongoDB, we migrated from MongoDB to DynamoDB. This will save our time and efforts.

VI. SYSTEM ARCHITECTURE

An architecture of our implementation of the Online Compiler using Cloud Computing model is a prosperous and specific diagram, formed using to be had standard wherein the primary situation is to illustrate a particular set of tradeoffs built in the shape and layout of a device or environment. An architectural version is a look of a factor of view in software program architecture [2].

The overview of the system is presented in Fig. 1 shows the system architecture.

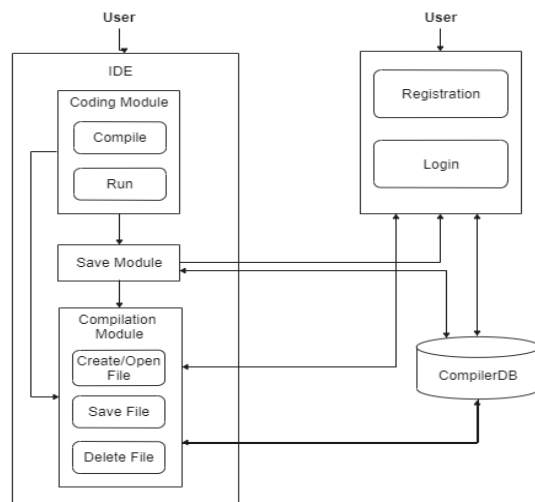


Fig.1 System Architecture

Following are the modules used in this system:

- **Registration/Login module:** This module is used for authenticating the user. Users can register and log in to the system. After registration system allocates a new space to the user.

- **Coding Module:** User can perform various operations like create, delete, upload the source code. GUI/IDE is provided to the user for coding.

- **Save Module:** User can save his/her code with different users.

- **Compilation module:** In this module, user compiles and run his/her program.

VII. PURPOSE & SCOPE

Cloud assemblies have digital compilation or built in the non-public cloud or externally hosted, promising digital compilation for all cloud computing assistance. Using cloud technology, clients using already present compilers, if getting rid of unneeded software program exemption from their circumstances. With no open financing, clients who set up Cloud compiler to decrease their month-to-month prices with the aid of using 50%, throughout the time accomplish extra structures controls, improved flexibleness, and the competence to quickly scale with minimum knowledge [1].

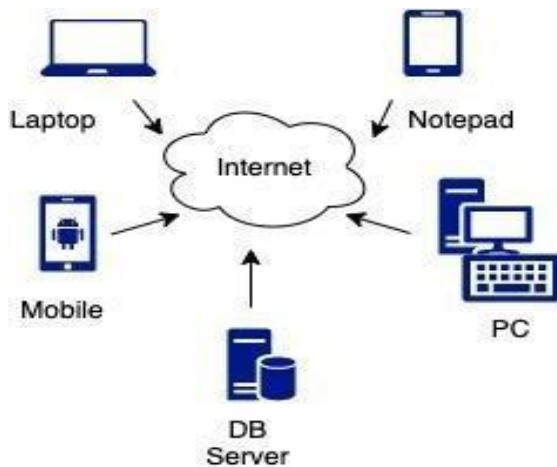


Fig.2 Cloud Computing

In Compilers, Cloud customers ought to now no longer download the compilers and upgrade the numerous instances of their systems. So that it enables us to investigate a couple of compilers, smooth to use. Using a software program as a provider can offer a couple of compilers together with C/ C ++, Java, C-Sharp and python. It gives online help, and error messages for consumer convenience.

VIII. RESULTS

A common front end is used for all compilers in the form of a HTML page. This HTML page allows for selecting a language in which the user wants to compile his code. User uses manual writing or copy and paste commands to enter the source code into the compiler front end.

The basic architecture of this HTML page consists of three text areas in which user can enter his code (input), pre-defined input and in the third text area, user can view the compiled code (output). (Fig. 1)

There is a drop-down menu located at the top of the page, using that menu we can change the programming language. Currently, the Online Compiler's package supports C, C++, C#, Java, and Python languages.

In addition, there is navigation bar which includes links for IDE, register and login pages. When the user login, he/she will be redirected to the Homepage and option for IDE, my code and logout will be available.

The process of compilation is performed after setting the desired language option and pasting the source code into the appropriate text area. The user must press the run button to execute the code. As a result, the user can see the output in the second text area. User can also enter the pre-defined input in pre-defined text area.

Everybody can inspect and utilize the Online Compiler by visiting the web site. There is no need for registration if the user wants to compile his/her code. Any user can access the web site and compile his/her code. If the user wants to save his code online, only then he must register. By doing this, the user can save his source code and access the previous compiled source codes associated with his account.

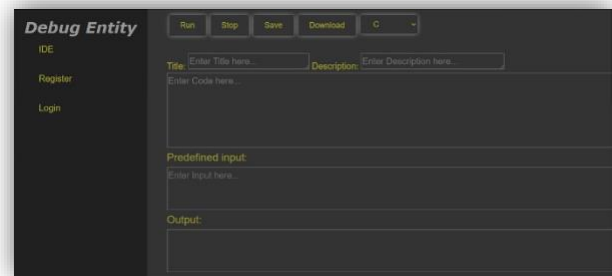


Fig. 3 Screenshot of the Homepage.

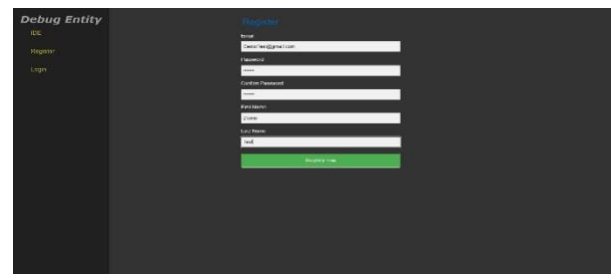


Fig. 4 Screenshot of the Registration Page.



Fig. 5 Screenshot of the Login Page.



Fir. 6 Screenshot of the Saved Code Page.

IX. CONCLUSION

Thus, we have shown that how our project can be used to eliminate the necessity of a separate compiler for every other language the user wants to compile and run. Another benefit of such a project is that if the compiler package is to be updated it can be easily achieved without installing it again on each machine. It also helps user to integrate the code and eliminate the errors on our application itself.

X. FUTURE SCOPE

- **Adding more compilers:** The application can be added with more compiling abilities for other languages such as COBOL, FORTRAN etc.
- **Shared editing:** Multiple people can work on a single project using this application in a simple way.
- **Including meter for efficiency:** The application can specify the user about the complexity of the code which will enhance their coding skills.
- **Providing security:** Security is a major concern on online platform. Security measures can be improved to keep the user data safe from any malicious activities.

XI. REFERENCES

- [1] Surya Chandra.V et al, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.8, August- 2015, pg. 348-355
- [2] Sajid Abdulla, Srinivasan Iyer, Sanjay Kutty, "Cloud based Compiler", International journal of student research in Technology Management, vol1(3), May 2013.
- [3] Arjun Datta, Arnab Kumar Paul, "Online Compiler as a Cloud- Service", International Conference on Advanced Communication Control and Computing technologies (ICACCCT), IEEE 2014.
- [4] Parag Chaudhari, Ritesh Manjarkar, Akhilesh Kulkarni, Sawre Vellaswami," Multi-Language Cloud Based Compiler", International journal of informative and futuristic Research review paper, vol2, Issue8, April 2015.
- [5] Aamir Nizam Ansari, Siddhart Patil, Arundhati Navada, Aditya Peshave, Venkatesh Borole, "Online C/C++ Compiler Using Cloud Computing", Pune Institute of Computer Technology, Pune University, 2011 IEEE.
- [6] Ratnadip Kawale, Pooja Soni, Gaurav Suryawanshi, Prof. Pradip Balbudhe, Online Editor for Compiling and Executing Different Languages Source Code, Volume 6, Issue 3, March 2016.
- International Journal of Advanced Research in Computer Science and Software Engineering (IJARCS)
- [7] Amazon Web Services, Inc., "Amazon S3 bucket" Online.Available: <https://aws.amazon.com/s3/>.
- [8] Amazon Web Services, Inc., "Amazon EC2" Online Available: www.aws.amazon.com/ec2/
- [9] Amazon Web Services, Inc., "Amazon DynamoDB" Online Available: <https://aws.amazon.com/dynamodb/>
- [10] Amazon Web Services, Inc., "Amazon IAM" Online Available: <https://aws.amazon.com/iam/>