Online Compiler

Semester- VII

Summer Internship Report

Submitted by

Name: Gabu Siddharth Mermabhai

Enroll No: 180170107030

COMPUTER ENGINEERING DEPARTMENT

VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE

CHANDKHEDA

Internal Guide:

Prof. Karan P. Bhatt

ASSISTANT PROFESSOR

VGEC, Chandkheda.

Gujarat Technological University

Academic Year 2021-22

Vishwakarma Government Engineering College, chandkheda

Computer Department

CERTIFICATE

Date: 10/06/2021

This is to certify that the Summer Internship Report entitled **Online compiler** Submitted by Enroll No: **180170107030** Name**: Gabu Siddharth Merambhai** towards the fulfillment of Subject: Summer Internship (3170001) of Gujarat Technological University is the record of work carried out by him under our supervision and guidance in the Academic Year 2021-22.

Internal Guide	Head of Department
Prof. Karan P. Bhatt	Prof M. T. Savaliya
Assistant Professor	Associate Professor
VGEC Chandkheda	VGEC Chandkheda

ACKNOWLEDGEMENT

With great pleasure, I take this opportunity to express my deep sense of gratitude and indebtedness to my renowned and esteemed guide **Prof. Karan P. Bhatt** Assistant Professor, Department of Computer Engineering, Vishwakarma Government Engineering College, Chandkheda for his consummate knowledge, due criticism, invaluable guidance and encouragement which has enabled us to give present shape to this work.

I am heavily indebted to HOD **M.T Savaliya**, Professor& Head, Department of Computer Engineering, Vishwakarma Government Engineering College, Chandkheda, for his everlasting willingness to extend his profound knowledge and experience in the preparation of this report. Any attempt to define this indebtedness would be incomplete.

Finally, I would like to thank our friends and family for their support and patience, and other faculty member of the department for his everlasting willingness to extend his support and help in the completion of this work. Especially to our parents who without their encouragement and financial support, this would not have been possible.

Yours Sincerely,

Gabu Siddharth (180170107030)

ABSTRACT

21’st Century is the era of technology. In this ever changing time’s our life are mostly governed by machines and electronic devices. And one of the factor which shaped the scenario of technology is ability to processes statements written in a particular programming language into machine language or code that a computer's processor uses. Compiler is a key ability required by most computer devices.

The research on this area has been making great progress. Compiler has a variety of uses. Preprocessing, lexical analysis, parsing, semantic analysis (syntax-directed translation), conversion of input programs to an intermediate representation, code optimization and code generation.

The principal idea behind this concept is Cloud computing and virtualization. With the recent advancement in this two field it has made possible to come up with maximum possible optimization required for online compiler.

A compiler that is to be installed manually on every system physically requires a lot of space and also configuring of it if not installed using default parameters. Also once a program is compiled it becomes platform dependent. It is also not easy to carry the same program code to multiple systems if situation doesn’t permit the usage of a single system. Another drawback is that we would need to install a different complier on each language on which we wish to work. I propose a solution to this in the form of a cloud based compiler.

List of Figures

List of Abbreviations

TABLE OF CONTENTS

CHAPTERS

1. Introduction

As Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort. I decided to make a project that aims to create an online compiler which helps to reduce the problems of portability of storage and space by making use of the concept of cloud computing.

The ability to use different compilers allows the programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors. Moreover a web based application can be used remotely through any network connection which is platform independent. The errors/Output of the compiled program can be display in a more convenient way.

Also the trouble of installing a compiler on each computer is avoided. Thus these advantages make this application ideal for conducting online examinations. Online compiler mainly deals with providing a platform to compile and execute programs that is not dependent on any platform related restriction or complication.

The compiler that I am going to implement would be a C, CPP, JAVA, Python, JavaScript compilers that is hosted on a private cloud implemented on Linux AWS platform. The compiler can be used to implement and run programs and directly view the output.

2. System Analysis

2.1 Study of current System

Compilers are used to run programs and convert them from a text format to executable format. A compiler that is to be installed manually on every system physically requires a lot of space and also configuring of it if not installed using default parameters. Also once a program is compiled it becomes platform dependent.

It is also not easy to carry the same program code to multiple systems if situation doesn't permit the usage of a single system. Another drawback is that we would need to install a different complier on each language on which we wish to work.

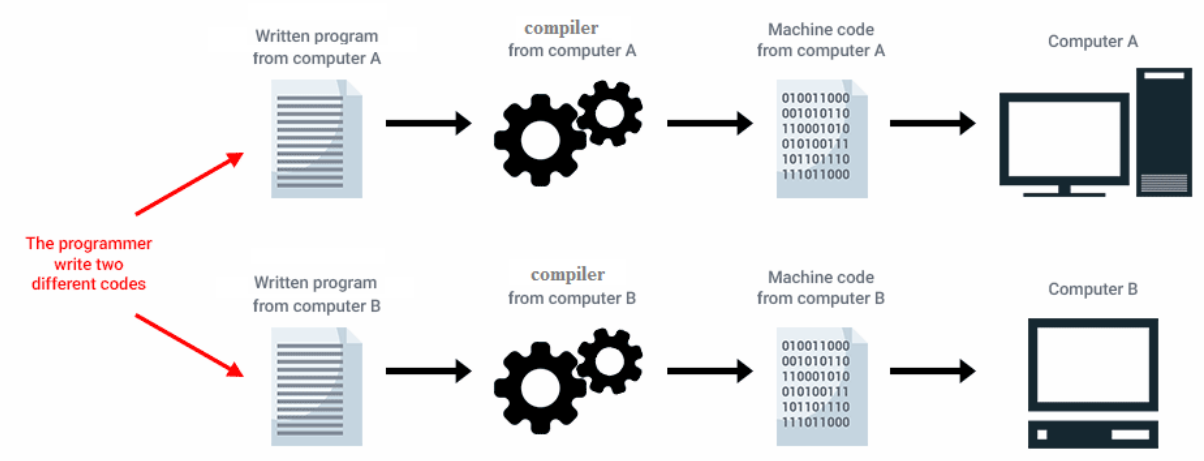

Fig 2.1 – current system

2.2 Problem and weakness of Current System

- A compiler that is to be installed manually on every system physically requires a lot of space and also configuring of it if not installed using default parameters.
- Also once a program is compiled it becomes platform dependent.
- It is also not easy to carry the same program code to multiple systems if situation doesn't permit the usage of a single system.
- We would need to install a different complier on each language on which we wish to work.
- Device and location dependence, users are unable to access systems using a web browser regardless of their location or what device they are using (e.g., PC, mobile phone).

2.3 Requirement analysis of New System

- Objectives
  - System is scalable and sustainable that enables sharing of resources and costs across a large pool of users.
  - Centralization of infrastructure in locations with lower costs (such as electricity etc.) due to centralization, increased security, focused resources, etc.
  - Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.
  - Device and location independence enable users to access systems using a web browser regardless of their location or what device they are using (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.
  - Virtualized and dynamic technology allows servers and storage devices to be shared and utilization be increased. Applications can be easily migrated from one physical server to another.

- Hardware requirements

    - Virtual Storage disk 10 GB
    - 1GB RAM
    - T2 micro ec2 instance

- Software Requirements

    - Linux Distribution (Ubuntu 18.04 used)
    - LAMP: Linux, Apache, MySQL, PHP.
    - Compilers for the languages to be supported.
        - g++ (for C++)
        - gcc (for C)
        - PyDEV (for python)
        - JDK (for Java)
        - Node (for JavaScript)

2.4 Brief literature review

The system is eventually intended for the software developers. Product will be deployed to web site and an android application. All users of this system will make use of the same via a website or an application. There will be cloud server where all the user data is kept and all the execution is done. Website and Android Application will only be the interface for the user data and the execution of provided functionalities.

Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort. Project aims to create an online compiler which helps to reduce the problems of portability of storage and space by making use of the concept of cloud computing.

The ability to use different compilers allows the programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors. Moreover a web based application can be used remotely through any network connection which is platform independent. The errors/Output of the compiled program can be stored in a more convenient way. Also the trouble of installing a compiler on each computer is avoided.

Thus these advantages make this application ideal for conducting online examinations. The software would be provided to the end user using a SAAS cloud. The software would contain a system that has a text editor. The user would be given an option to select the language in which he wants to compile the program. The software will compile the program and return the output to the user. Additional functionalities such as monitoring of the system, user usage, user forums, and collaborative development can be added as needed.

2.5 Design: Analysis, Design Methodology and Implementation Strategy

To solve actual problems in an industry setting, we must incorporate a development strategy that encompasses the process, methods and tools. So development begins at the system level and progresses through analysis, design, coding, testing, and support.

Design. Software design is actually a multistep process that focuses on four distinct attributes of a program: data structure, software architecture, interface representations, and procedural (algorithmic) detail. The design process translates requirements into a representation of the software that can be assessed for quality before coding begins. Like requirements, the design is documented and becomes part of the software configuration.

Code generation. The design must be translated into a machine-readable form. The code generation step performs this task. If design is performed in a detailed manner, code generation can be accomplished mechanistically.

Testing. Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals; that is, conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

Support. Software will undoubtedly undergo change after it is delivered to the customer (a possible exception is embedded software). Change will occur because errors have been encountered, because the software must be adapted to accommodate changes in its external environment (e.g., a change required because of a new operating system or peripheral device), or because the customer requires functional or performance enhancements.

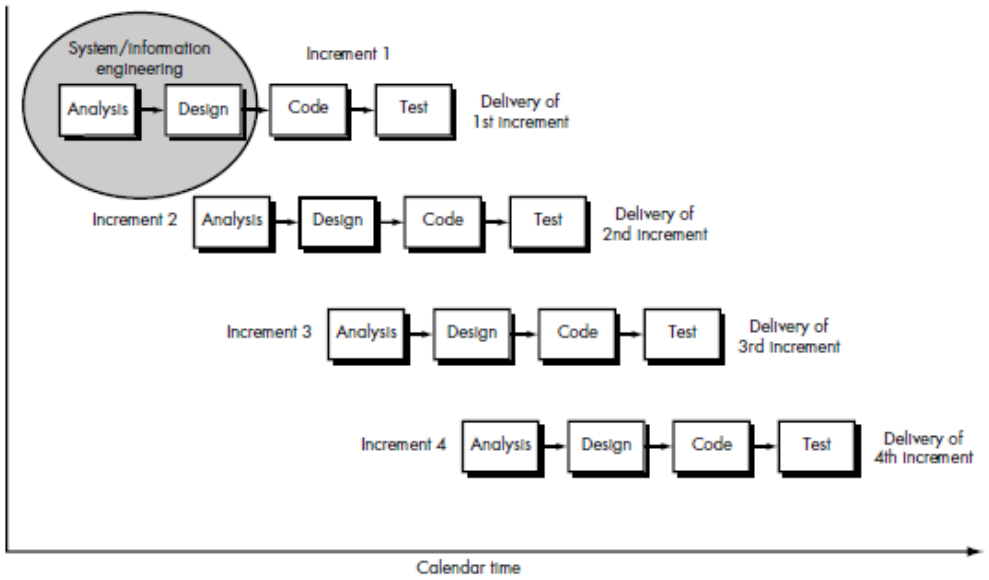

Fig 2.5 Incremental Design

The incremental model combines elements of the linear sequential model with the iterative philosophy of prototyping. The incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequence produces a deliverable "increment" of the software.

For example, word-processing software developed using the incremental paradigm might deliver basic file management, editing, and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and grammar checking in the third increment; and advanced page layout capability in the fourth increment. It should be noted that the process flow for any increment can incorporate the prototyping paradigm.

When an incremental model is used, the first increment is often a core product. That is, basic requirements are addressed, but many supplementary features remain undelivered. The core product is used by the customer.

As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality. This process is repeated following the delivery of each increment, until the complete product is produced.

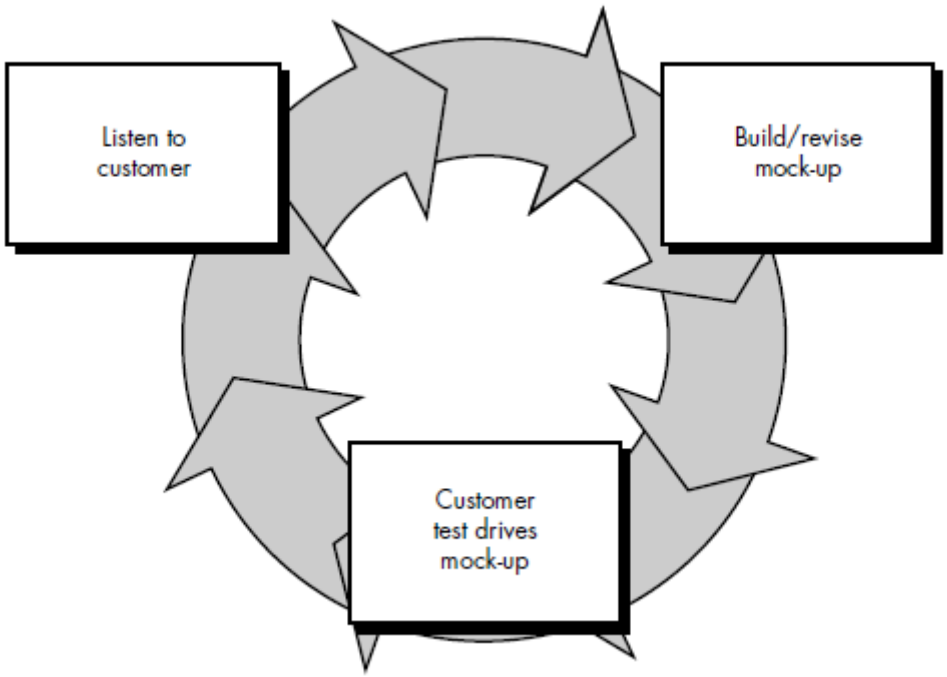

Fig 2.6 the prototyping

3. System Modeling

3.1 Dataflow diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. Often they are a preliminary step used to create an overview of the System which can later be elaborated.

It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. It can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO.

Dataflow Diagram for the system is as follows

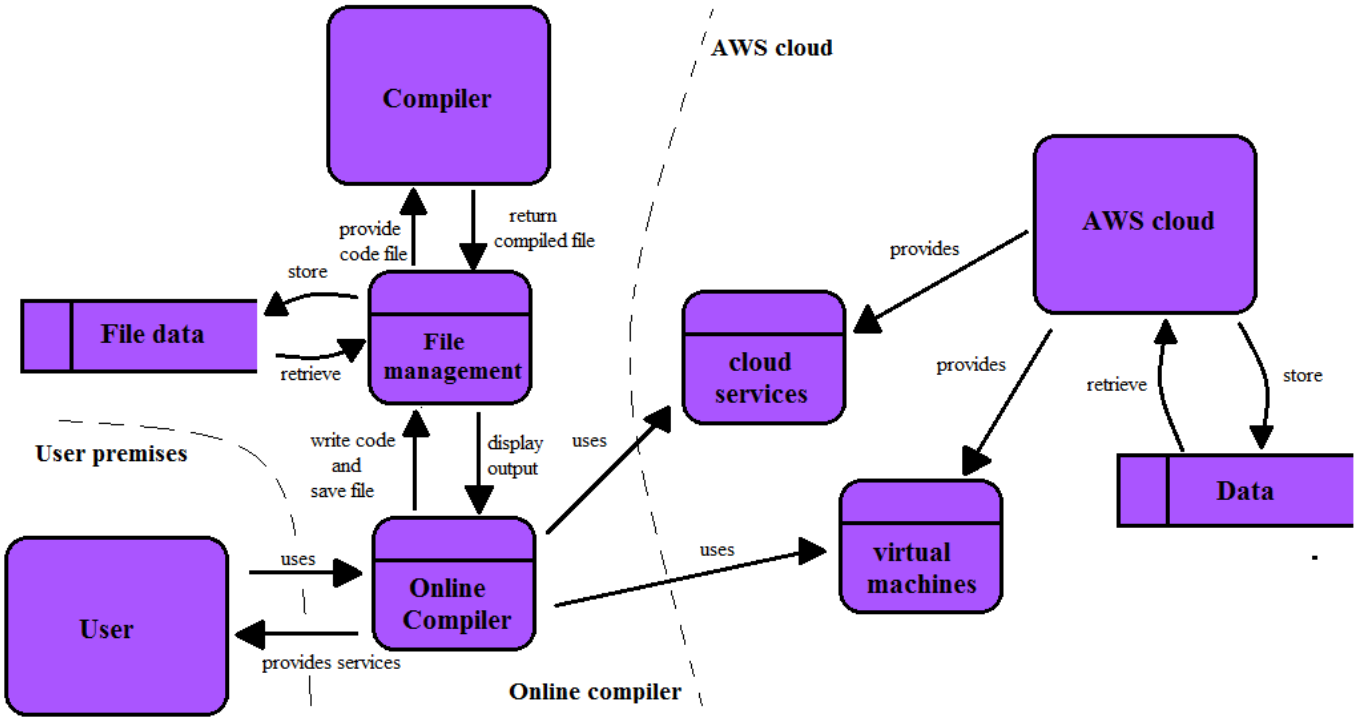

Fig 3.1 – Data Flow Diagram

3.2 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class Diagram for the system is as follows

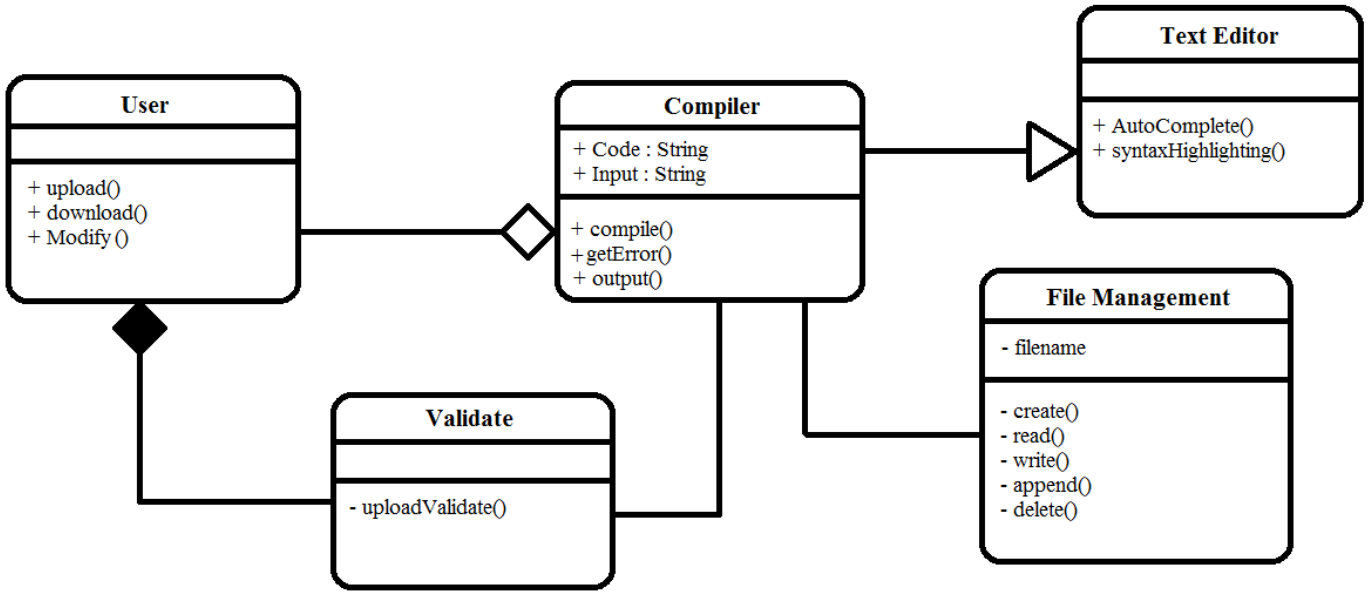

Fig 3.2 – Class Diagram

3.3 System Architecture

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system. A diagram much like a picture is worth a thousand words. In other words, an architectural diagram must serve several different functions.

To allow relevant users to understand a system architecture and follow it in their decision-making, we need to communicate information about the architecture. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

System Architecture for the system is as follows

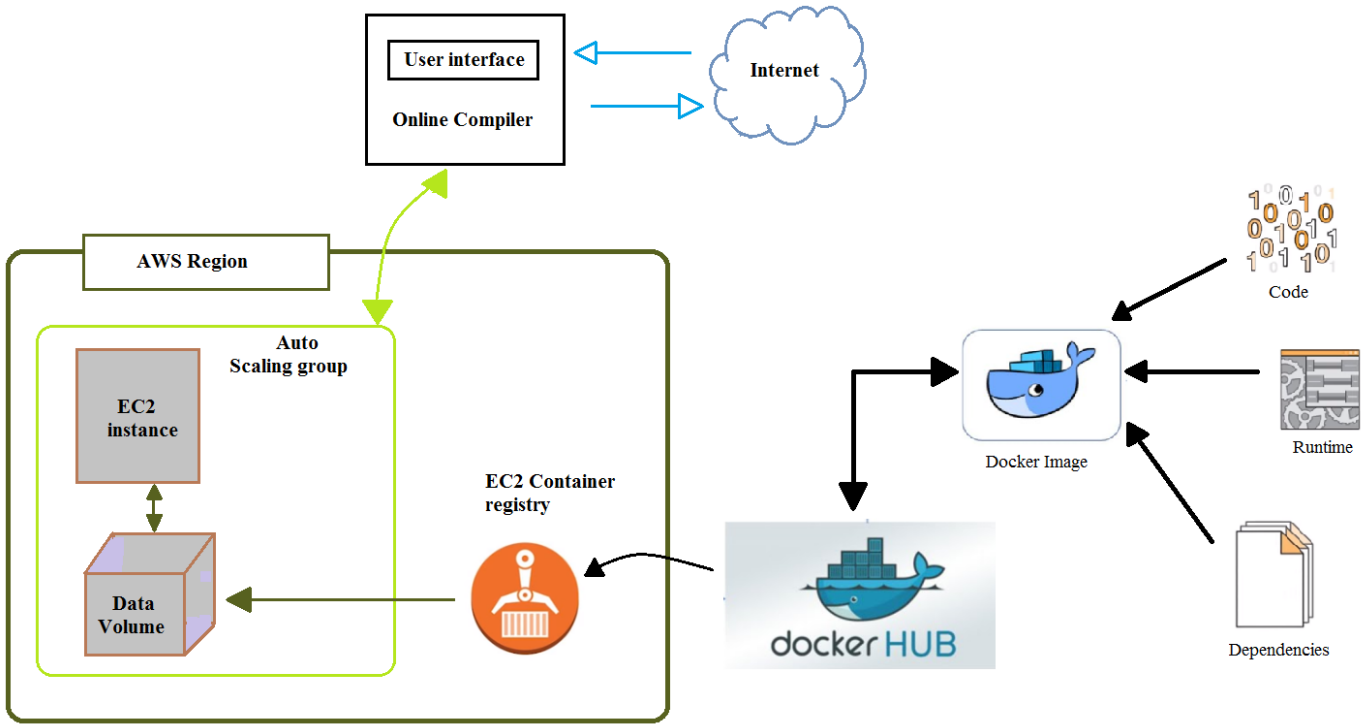

Fig 3.3 – System Architecture

4. Implementation

The extensive support for deploying PHP based applications in AWS triggered the idea of building the application using PHP. Also the web application to create an online compiler was developed. The code is a server side script. The compilers are hosted on virtual machines created in our Linux AWS cloud account.

4.1 Snapshots of project

Live Link: http://ec2-54-82-89-25.compute-1.amazonaws.com/

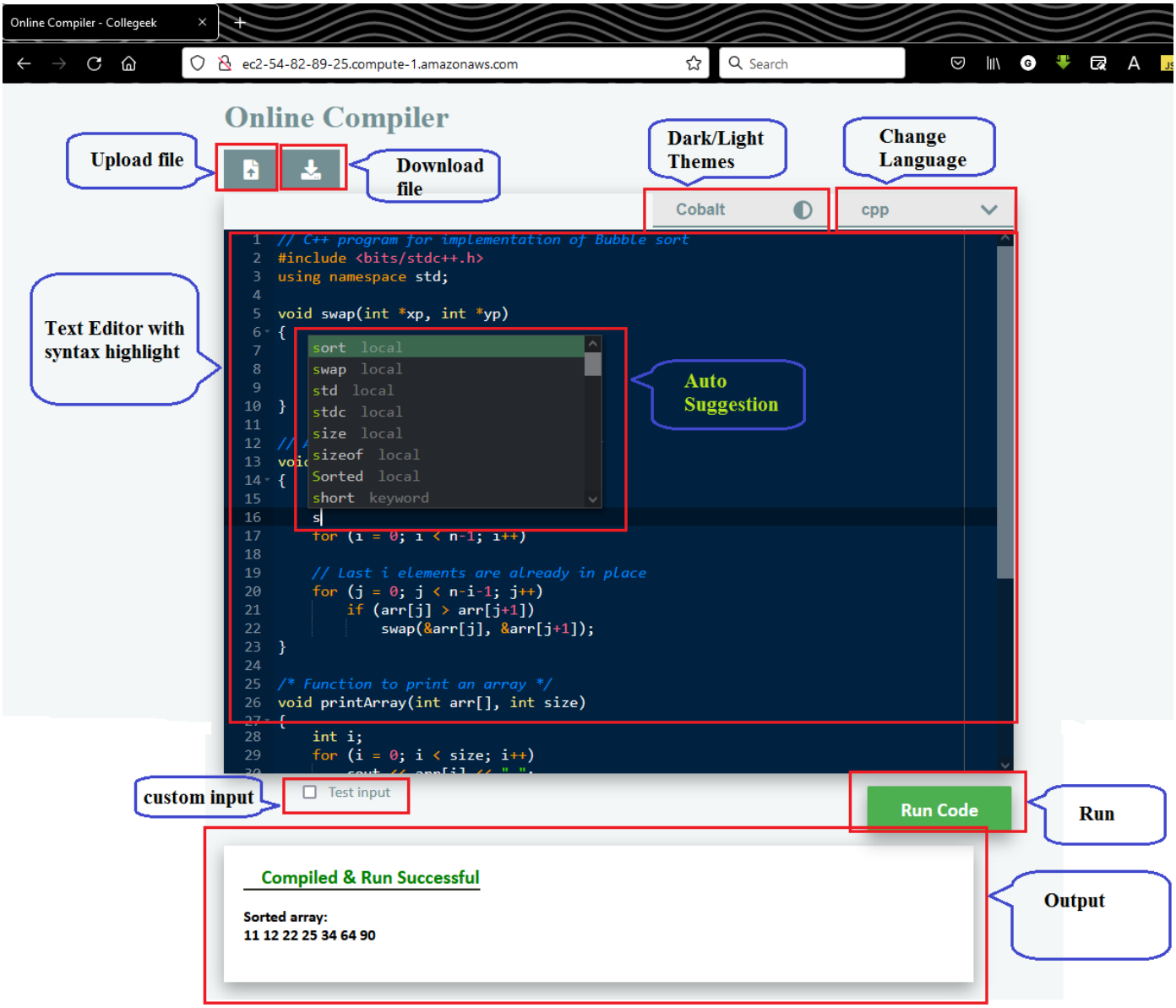

Fig 4.1 – Online compiler

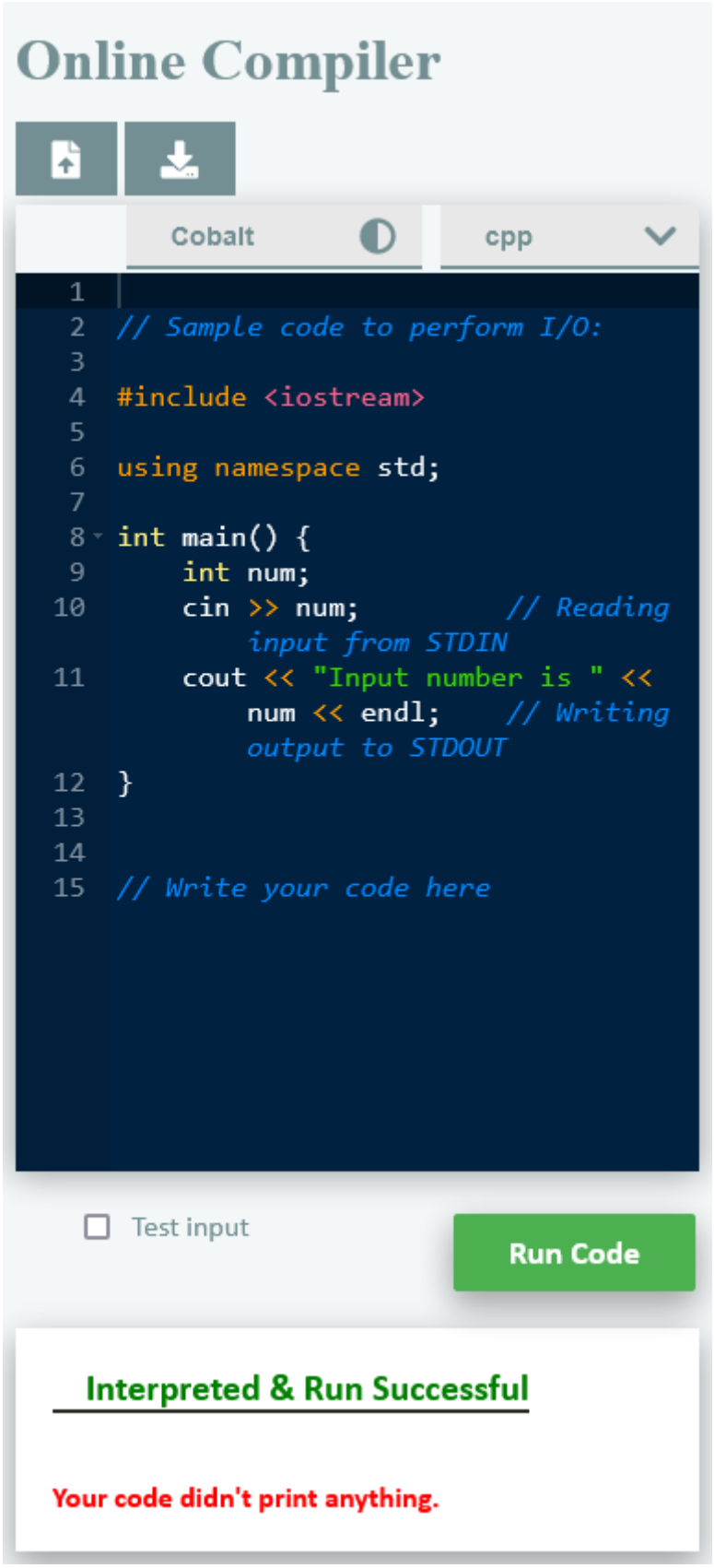

Fig 4.2 Compiler on Mobile devices

5. CONCLUSION

The main reason for creating the project is to provide a centralized compiling scheme. Also, it will act as a centralized repository for all the codes written. The other major advantage that this system will have over the others is that it will make the users system lightweight i.e. there will be no need to maintain separate compilers at the client side.

A compiler, which is the heart of any computing system, transforms source code from a higher level language to a lower, machine level language. This is mainly done in order to create executable files which can then be run in order to execute the program and its instructions. As compared to the current scenario where each compilers required to be installed on each machine separately this would eliminate the need to install compilers separately. So we can check our code at the centralized server. Another advantage of such project is that whenever the compiler package is to be upgraded it can be done easily without again installing it on each and every machine.

6. Future Scope

- **Provide more compilers:** The application can be extended to provide compilers for J#, FORTRAN, COBOL, C#, Ruby etc.
- **Collaborative Editing:** Collaborative editing features can be added so that large project groups can work on the project online and with ease.
- **Mobile Applications**: Mobile Applications can be developed so that the users can create and execute their applications using their mobiles with greater ease.
- **Create Web API's**: Could also provide the above project by using API's in the cloud. This helps to create a more interactive way of providing software as a service.

7. Reference

- Jajodia, P. (2020, June 2). How We Built Our Online Python Compiler. Programiz Blog. https://www.programiz.com/blog/online-python-compiler-engineering/
- programiz. Online C Compiler. https://www.programiz.com/c-programming/online-compiler/
- Dockerfile reference. (2021, June 10). Docker Documentation. https://docs.docker.com/engine/reference/builder/
- M. (2019, March 21). Quickstart - Deploy Docker container to container instance - Azure CLI - Azure Container Instances. Microsoft Docs. https://docs.microsoft.com/en-us/azure/container-instances/container-instances-quickstart
- M. (2018, March 21). Tutorial - Prepare container image for deployment - Azure Container Instances. Microsoft Docs. https://docs.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-prepare-app
- AWS. (2017, March 12). Amazon Elastic Container Service https://aws.amazon.com/ecs/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc&ecs-blogs.sort-by=item.additionalFields.createdDate&ecs-blogs.sort-order=desc
- Scribbr. (2021, June 2). APA Citation Generator (Free) | References & In-text Citations. https://www.scribbr.com/apa-citation-generator/new/webpage/