**ONLINE IDE, “EXECODE”**

# ABSTRACT

*This project is being developed by taking into consideration the developers working in a team who have to frequently install and upgrade software. The aim is to provide an online repository for storing their code and to carry out tasks such as compilation, debugging and execution at server-side. This will help the developers to view, edit, compile, debug and execute their code anytime anywhere even if they are working in a team. This includes support for versioning of the code developed in a team, uploading and downloading projects and permissions to access the code.*

**Keywords:** *Web server, Web browser,* Program editors, Web-based interaction

**Introduction:**

Developers in the computer field need to do a lot of initial installation before actually starting their coding task. Also if a developer needs to work at home or any other desktop available at that point of time then he/she needs to reinstall everything again. Another case is when a number of developers are working on the same project then each developer needs to do the same installation. This project is being developed to reduce these efforts.

In this project all the software resides at server-side; the compilation and execution takes place at server-side; the results are sent back to the client-side; hence it makes the developer’s task easy. Techniques such as web sockets and non-blocking server will provide an improved and faster working environment. The code is stored in an online repository. Since browser is used to access the code no issues of portability arises at client-side. Moreover if any software is supported only in some operating systems the server can be used to install the software and be made available to other users. All the team members can use the same software made available through the online IDE; hence there will be no issues related to having different versions of the same software. If particular software is upgraded then only a single copy must be updated. Uploading and downloading of projects for offline use. For novice programmers “ExeCode” provides sample codes of all languages.

**Platform:** Linux

The project is being developed in Linux platform using Python server and Tornado framework. Tornado framework is non-blocking server, can handle asynchronous requests, is highly scalable and can handle 1000 requests at a time. MySQL is being used as a database. We are using Github for storing the programs and will maintain the versions of the program.

Currently we are supporting C, C++, Java, Python and Ruby languages

**Block Diagram:-**

Client Machine

Server Machine

Download

Upload

View

Browser

Memory

Editor

1. Edit

Repository

1. Version Control

Send Data

Debug

Compile

Execute

Save

Set language

Set of Compiler

Commands

Results

**Fig 6.1 Block Diagram**

**Usage scenario-**

1. Owner – The creator of a project is termed as the owner.
2. Circle – If the project being developed using ExeCode consists of a team then the team members are said to join the Circle.
3. Public – If the project that is being developed using ExeCode is visible to anyone then it is termed as Public project.
4. Private – If the project that is being developed using ExeCode is visible to only the owner then it is termed as Private project.

**Software Interface Description**

The software interacts with a developer through a browser. The web application provides an editor for the user to type the code. It provides an option to set the language. It allows the user to send invitations to other developers. The user can create a new project, compile, debug and execute the project. Versioning of the project is provided by integrating with Github. The user can upload and download project and can use offline and later on integrate.

**Data Description**

The programs and user-related information are stored on the server side. There is a separate dedicated folder for each user with the name of folder same as the user-name created which gets created in the root directory along with two other folders named “Circle” and “Private” inside the user’s directory when he is registered in ExeCode. A separate file name “Userdetails.txt” is created and the user related information such as Github account information is kept in it. When the user creates a new project, a new directory with the project name is created and the project files are stored in that project directory. The owner of the project can invite other users for a team project.

**Major Functionalities:**

* **Create New Project**

This function enable user to select a language on which he is working and create a project with a name which is not already in use. This function is performed only by owner.

* **Send Invitations**

Owner is responsible for this function. For creation of a group, owner can send a invitation to programmers those will work on the same project and also can set the visibility mode.

* **Download Project**

User can perform this activity by selecting an existing project and download that project.

* **Upload Project**

Group can work on a project and modify it and upload a modified project on repository. Uploaded project can be downloaded by another member of group.

* **Run Project**

User will select a project he wants to execute and will call run function. After execution of this function system will display the desired result of project

* **Debug Project**

User will select a project he wants to debug and system will execute project line by line and will display the desired result of project.

* **Save Project**

User has an ability to modify a project any number of times. So user can save is modified project on server using this function.

* **Trash**

A Trash function is available that maintains all the files deleted by the user.

* **Notifications**

Notifications are sent by the admin if new languages are added or for any change.

* **Forum**

A forum is made available through the ExeCode that allows the user to put up their problems if any. Experts will be formulated by the ExeCode by watching all the replies sent by the users and suggesting the user to contact the expert for solution.

* **Chat Service**

Chat Service for users who wish to contact the team members through ExeCode.

* **COCOMO Estimates**

COCOMO estimates of the current project are displayed for Circle projects.

**Internal software data structure**

**Global data structure**

**Class Diagram and Design Patterns**

A observer pattern for -

**Components available**

* API for syntax highlighting.
* API for Treeview display of all projects
* API for getting access to Github.
* API for Chat Service.

**SWOT Analysis:**

|  |  |
| --- | --- |
| **Strengths**   1. Anytime, anywhere coding online. 2. Integration with Git for versioning and branching of projects (tracking project changes). 3. Use of event-driven non-blocking server for better performance. 4. Uniform packages and installations available for languages. | **Weaknesses**   1. Supports only Linux-based languages since the development environment is Linux. 2. Internet connection mandatory to utilize the product. |
| **Opportunities**   1. Useful for project group members to work on the project irrespective of place. 2. A log of the topics discussed available on the forum, which can be very useful to backtrack ideas brainstormed. | **Threats**   1. Slow internet connection may impede the performance of the product. |

**Limitations**

* Currently the number of languages supported is less. Also we cannot support Windows-based languages as our server is Linux where the compilation would take place.
* Data centralized architecture because we are supporting a single server currently. So as the number of users increase load on the server will increase. Also if the server crashes the whole system collapses.

**Future Framework**

We are planning to deploy ExeCode on Cloud to overcome its drawbacks of data centralized architecture. Also we plan to provide integration with other Version Control System like Joyent, BitBucket for user’s convenience.

**Conclusion:**