



॥ सा विद्या या विमुक्तये ॥

**भारतीय प्रौद्योगिकी संस्थान धारवाड**  
**Indian Institute of Technology Dharwad**

Software Systems Lab (CS213)  
**Programming Drawboard**

**Team Members**

Saksham Chhimwal(210010046)  
Shubh Agarwal(210020046)  
Shivesh Pandey(210020044)

# Contents

<b>1</b>	<b>Issues</b>	<b>3</b>
1.1	Need for visualization . . . . .	3
1.2	Boilerplate . . . . .	3
1.3	Cleaner Code . . . . .	3
<b>2</b>	<b>Proposed Solution</b>	<b>4</b>
2.1	The DrawBoard . . . . .	4
2.2	Online Shell(s) . . . . .	4
2.2.1	Python Shell . . . . .	4
2.2.2	C/C++ Shell . . . . .	5
<b>3</b>	<b>Technologies Used</b>	<b>6</b>

# 1 Issues

## 1.1 Need for visualization

Its much of a standard operational procedure in programming to keep debugging the code, but a specific area that most of the programmers find hard to check is the logical flow of functions inside the program. While writing a program it is necessary for a programmer to understand how the program is working and how the member functions are being called inside of the program(**Call Stack**). While debuggers of many **IDEs** do provide a interface of how the program is being executed but for a beginner it requires quite a bit of effort to understand how to use the debugger to its full potential.

The major drawback is it renders all in a textual manner which sometimes is not appealing to the user and hence they tend to avoid it and trace the program manually which is a great loss of manpower.

## 1.2 Boilerplate

Many programming languages that are being used in modern world do not require writing boilerplate but programming languages like **C**, **C++** and **Java** to list a few demand the user to write boilerplate even to do the least of all the tasks intended. This might not seem to be a problem in large projects but its not always the case where we use them. As a student sometimes it is much more convenient to solve a problem using scripts rather than using the calculators, it does save a lot of time. But in programming languages that are typed statically there is no way where we can run a file without saving it.

If a student wants to run a **C/C++** code to test a algorithm then he has to do a lot of work even before he implements the actual working piece of code thus creating unnecessary delays and increasing the scope of human error.

## 1.3 Cleaner Code

Many programmers fail to write a clean code which arises a problem when working in a team project. In a team a programmers main purpose is to make the code as readable as possible but its not something that one could learn in a day or two it needs experience. When someone fails to write a code that is not readable the its almost impossible for the co-worker to implement his logic without actually understanding what the other person has actually done.

But this can be avoided if the other person transports his code in terms of titled blocks which the other person can join to create what he actually wants without even looking into the code that is working behind it. This also provides the much needed intuition that is required when one is asked to work on a specific piece of the code of a much larger project.

## 2 Proposed Solution

We intend to create an Online Shell and a Drawboard that will act as both a quick alternative to run the code without even writing the boilerplate and a Drawboard in which the programmer can lay out his ideas in pieces and connect them to execute his program.

Thus our solution can be separated into two different parts:

1. **DrawBoard**

This will act as a place for the programmer to control the flow of the functions in program.

2. **Online Shell**

This will act as a quick way for the programmer to write down the code quickly without writing the unnecessary boilerplate.

- (a) **Python Shell**

- (b) **C++ Shell\***

- (c) **C Shell\***

*\* If time permits.*

### 2.1 The DrawBoard

It will provide the user with a **GUI** in which he will be able to see all the functions he has created and will be able to connect them in any way possible. The functions will then be executed in the manner the user connected them and eventually the program will execute.

Apart from this the user will be provided with some functionalities that can be implemented directly using the menu in the **Drawboard** like **Loops**, **GoTo**, etc.. these will help making the online execution unit **Turing Complete**.

The user will write code in the shell unit and will convert it into a function which in turn will be converted into a function block that will be called when connected to another function block. The user will be able to himself control the flow of the code and see it in real time both simultaneously.

Moreover the user can print his plan directly in the form of PDF document that can be presented to other people.

### 2.2 Online Shell(s)

#### 2.2.1 Python Shell

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec

ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### **2.2.2 C/C++ Shell**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### 3 Technologies Used

- **HTML,CSS,JS**  
To create the webpage and making it responsive.
- **Node.js**  
This acts as the backend server of our Web Application.
- **Express**  
This is the framework being used with our backend server.
- **GitHub**  
This is serving two purposes in the project
  1. For the Developers, it is being used for the collaborative work and a version control of our project.
  2. For the User, it acts as the database to store their shell commands that they give in the form of **Gists**.
- **Python**  
This is being used as the primary shell languages that the user can work with.
- **C/C++** (*if time permits.*)  
These are being used as another shell language(s) that the user can work with.
- **MongoDB**  
This is our primary database that will hold the user login credentials into the shell in-order to avoid logging in into GitHub each time the user opens the shell.