ColabEdit

OVERVIEW

We observe different types of code editors in our daily life but many of them are specialized for personal use but very few of them have collaborative code editing features. Sometimes we want to draw some diagrams to understand code. So keeping in mind all the requirements we built a code editor which is collaborative and also has chat as well as a drawing panel.

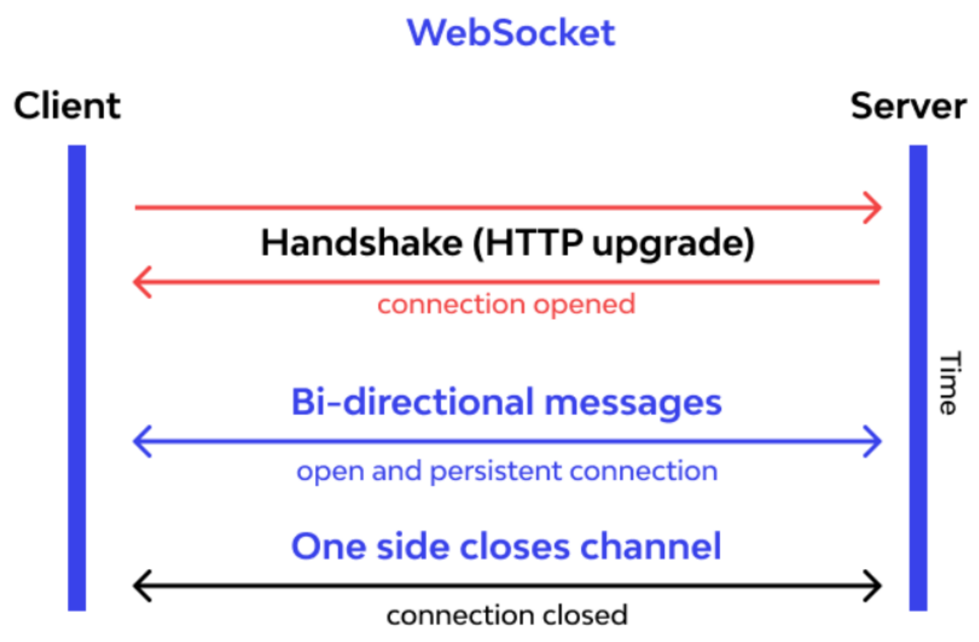
INTRODUCTION

We made a collaborative editor which has the following functionalities :

- Live Code Editor
- Chat Box
- Drawing Canvas

WEB SOCKET

WebSocket is bidirectional, a full-duplex protocol that is used in the client-server communication. It starts with ws:// or wss://.They need support from HTTP to initiate the connection. During connection setup phase the request message is sent to upgrade the protocol from HTTP to the Web Socket protocol.



SETUP AND USAGE

- First a user has to register himself in the database.
- Then after registering he/she needs to login using the registered name and password.
- Upon successful authentication, the user is asked to enter a valid room number.If the room is unoccupied then he/she can claim it by entering the password otherwise if the current room was already taken by the previous user then this user can only join this room if he has the password to this room.
- So we employed a security in our editor that only authenticated persons can only join the particular room. Without this security any random person can join the room and revert back or modify the working codes to a harmful extent.

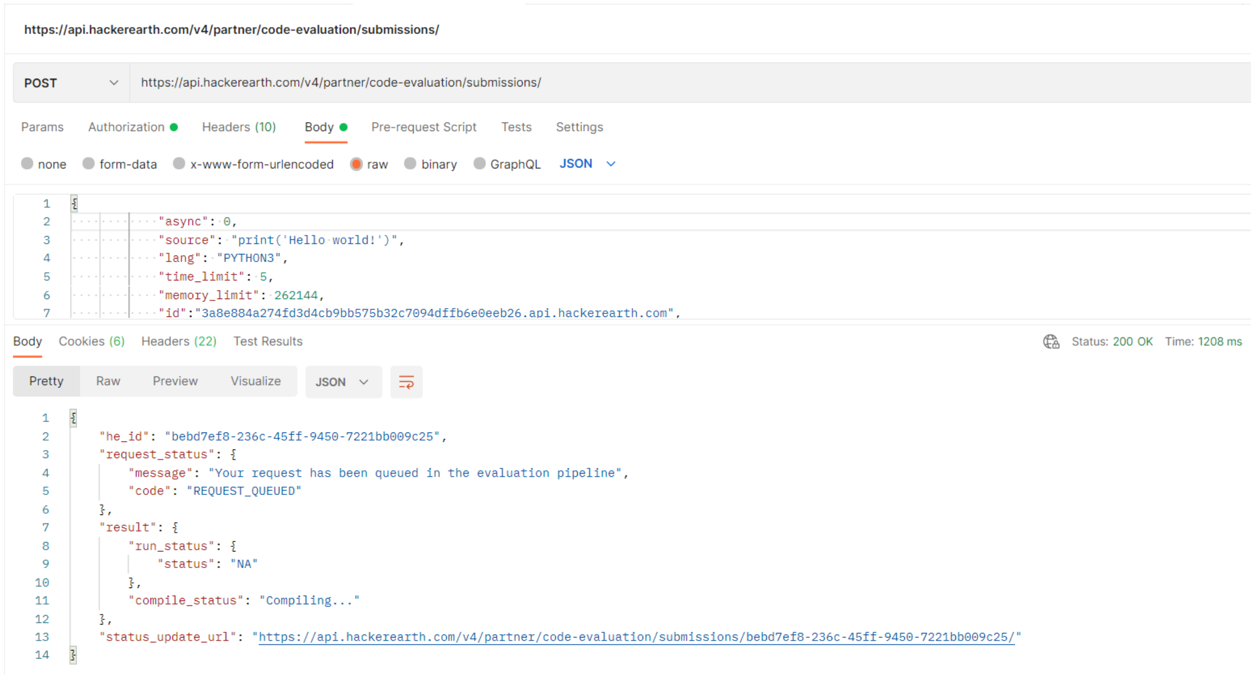
LIVE CODE EDITOR

Our live code editor supports python language.

We used Hackerearth API to compile and execute the code .

To compile the code we send a POST request to HackerEarth API and it responds 2 times , one after compilation and the other after execution of the code.

Compiling Stage:



https://api.hackerearth.com/v4/partner/code-evaluation/submissions/

POST https://api.hackerearth.com/v4/partner/code-evaluation/submissions/

```
{
        "async": 0,
        "source": "print('Hello world!')",
        "lang": "PYTHON3",
        "time_limit": 5,
        "memory_limit": 262144,
        "id":"3a8e884a274fd3d4cb9bb575b32c7094dffb6e0eeb26.api.hackerearth.com",
```

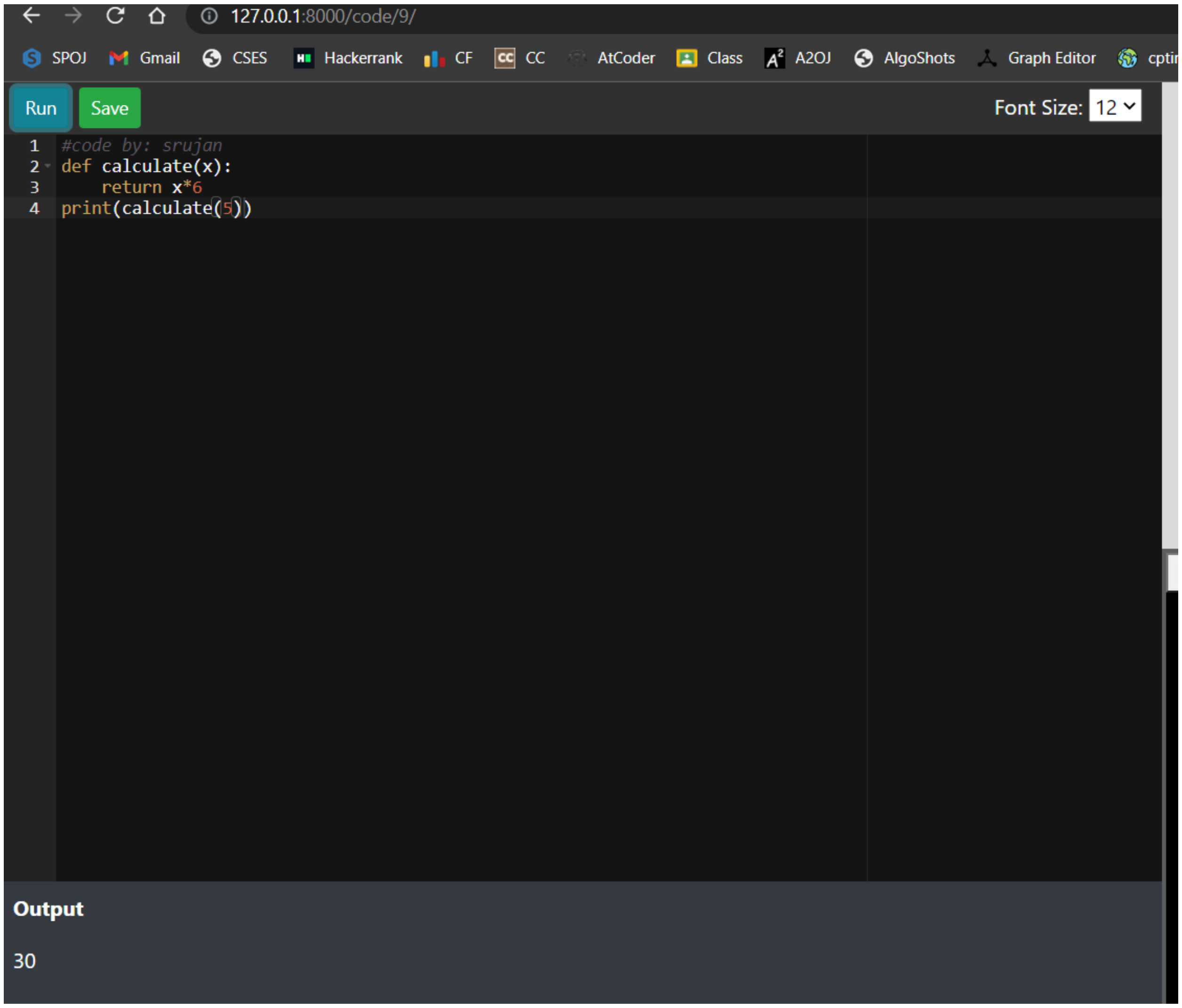
Status: 200 OK Time: 1208 ms

```
{
    "he_id": "bebd7ef8-236c-45ff-9450-7221bb009c25",
    "request_status": {
        "message": "Your request has been queued in the evaluation pipeline",
        "code": "REQUEST_QUEUED"
    },
    "result": {
        "run_status": {
            "status": "NA"
        },
        "compile_status": "Compiling..."
    },
    "status_update_url": "https://api.hackerearth.com/v4/partner/code-evaluation/submissions/bebd7ef8-236c-45ff-9450-7221bb009c25/"
}
```

Output after execution is in the form of json response returned by Hacker Earth API:

```
{

 'result': {'run_status': {'status': 'AC', 'output':
'https://he-s3.s3.amazonaws.com/media/userdata/AnonymousUser/code/b02119
4', 'time_used': 0.030336, 'memory_used': 2884, 'status_detail': 'NA'},
'compile_status': 'OK'},

'he_id': 'eff641db-ee88-4e4b-a077-5dede33c571c',

 'request_status': {'message': 'Your request has been completed successfully', 'code':
'REQUEST_COMPLETED'},

 'status_update_url':
'https://api.hackerearth.com/v4/partner/code-evaluation/submissions/eff641db-e
e88-4e4b-a077-5dede33c571c/'}
```

LIVE DEMO



```
#code by: srujan
def calculate(x):
    return x*6
print(calculate(5))
```

Output

```
30
```

Our editor allows multiple users in the same room to write and edit code simultaneously.

The languages supported by our editor are python and python3.

In the above image we can see that a code to multiply the user input by 6 is written. If 5 is passed as input then output observed is 30 which is correct.

To perform the compilation and execution we took help of Hackerearth API , which allows to run compilation upto 1000 times and on exceeding the limit the administrator has to reset the key to obtain the more required number of executions permitted.

We also provided the basic functionalities such as adjusting font size. The font size adjusted by one user does not affect the font size of another user. Suppose if user 1 is working with font size of 12 then it’s not necessary for user2 to use the same font size . The user2 is free to use font size of its own.

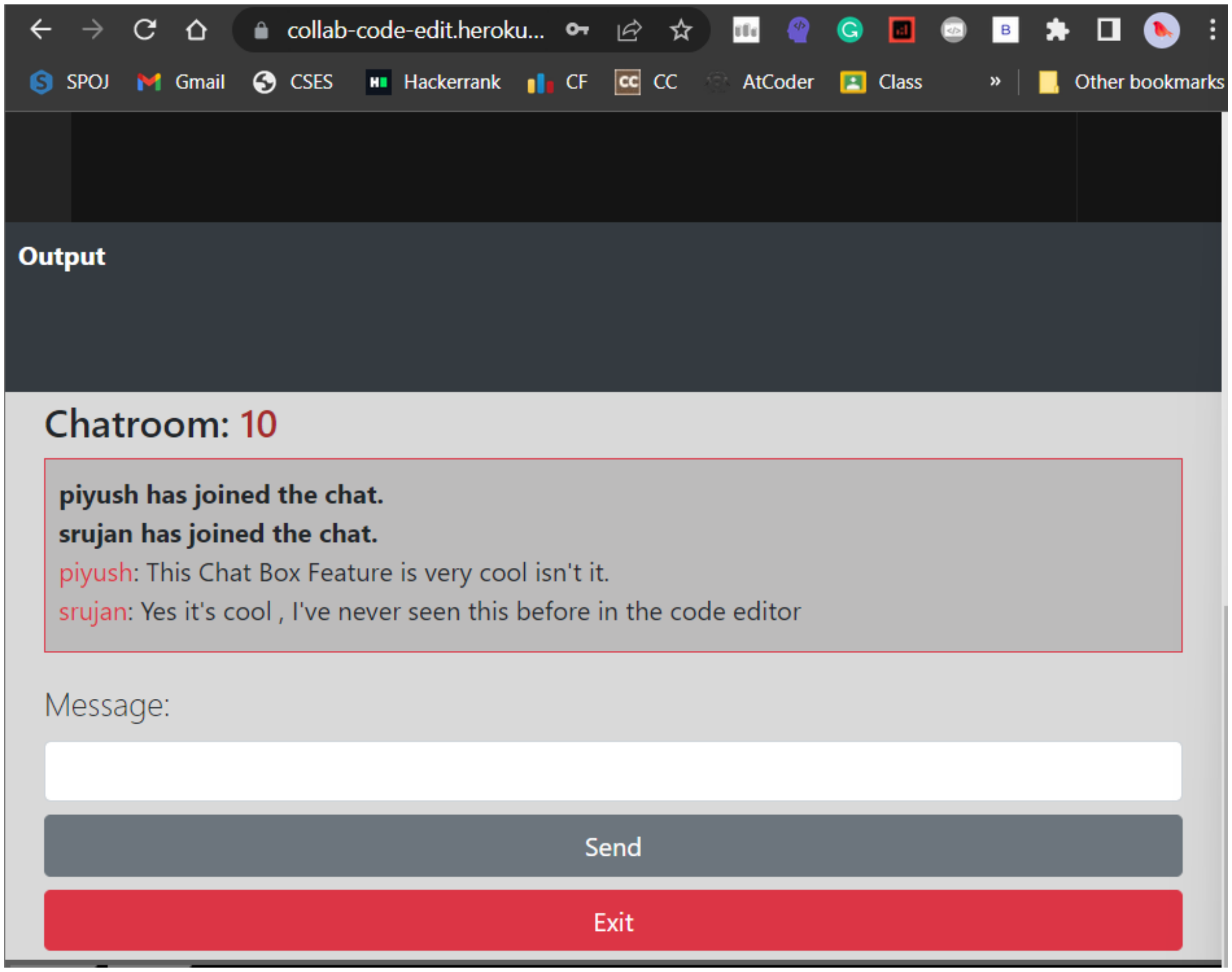
CHAT BOX

We designed this chat box so that people present in the same room can communicate with each other.

Having a chat box allows users to share their ideas easily with each other. People can either use it to explain the code part or simply can use the room for chatting with each other.

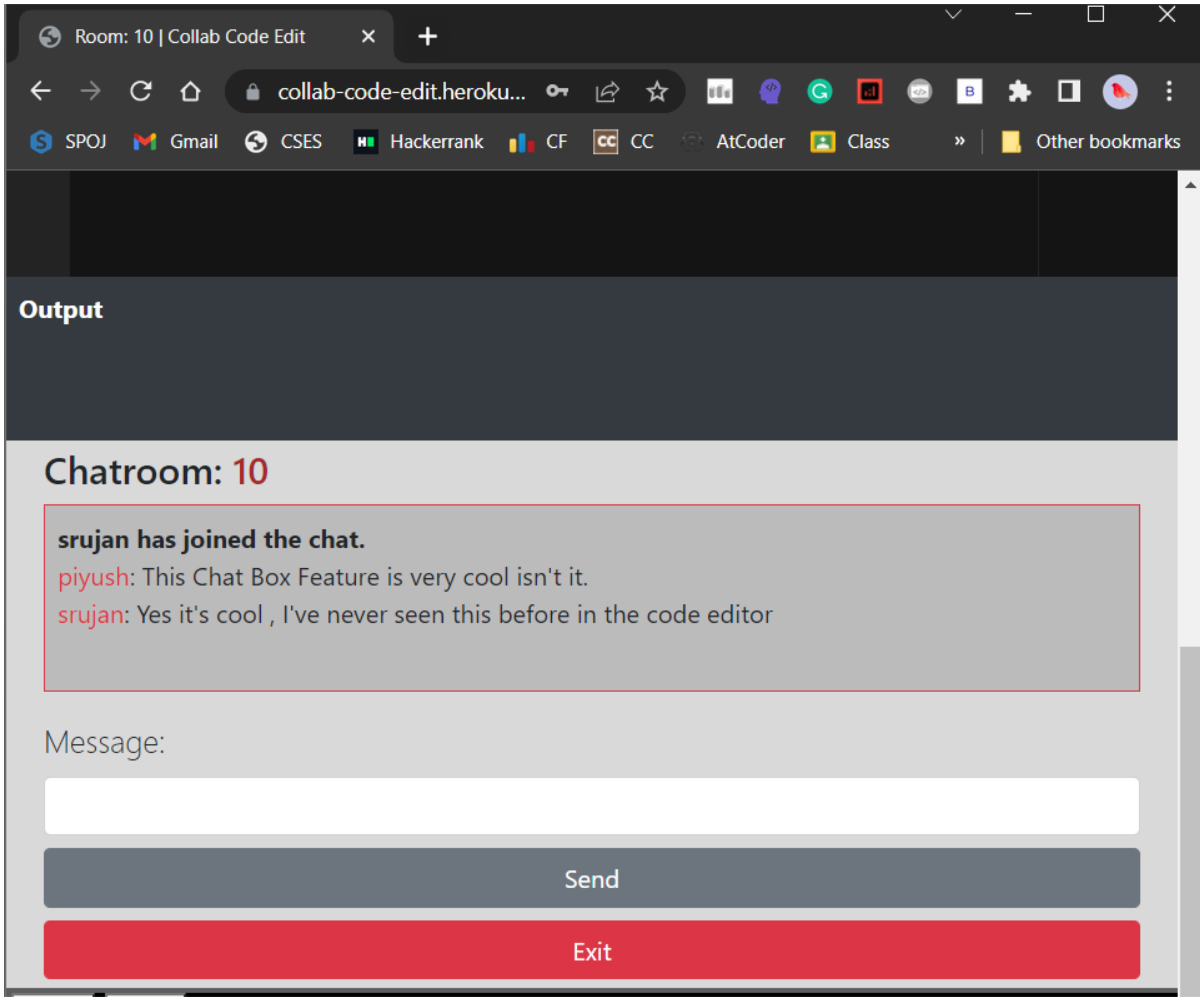
The messages are synchronized by setting a semaphore variable. When a user wants to send a message then first it acquires a semaphore and decreases it count by 1 , so the other user who wants to send the message can only write in chat but sending will be done once the semaphore or lock is released by the sending process of user1.This mechanism allows the synchronization and collaboration among users.

LIVE DEMO



Chatroom: 10

piyush has joined the chat.
srujan has joined the chat.
piyush: This Chat Box Feature is very cool isn't it.
srujan: Yes it's cool , I've never seen this before in the code editor



Chatroom: 10

srujan has joined the chat.
piyush: This Chat Box Feature is very cool isn't it.
srujan: Yes it's cool , I've never seen this before in the code editor

In this example we can see that piyush created a room and entered the chat and later srujan joined.

They both can communicate by using a chat box as we can see in the above images.

DRAWING CANVAS

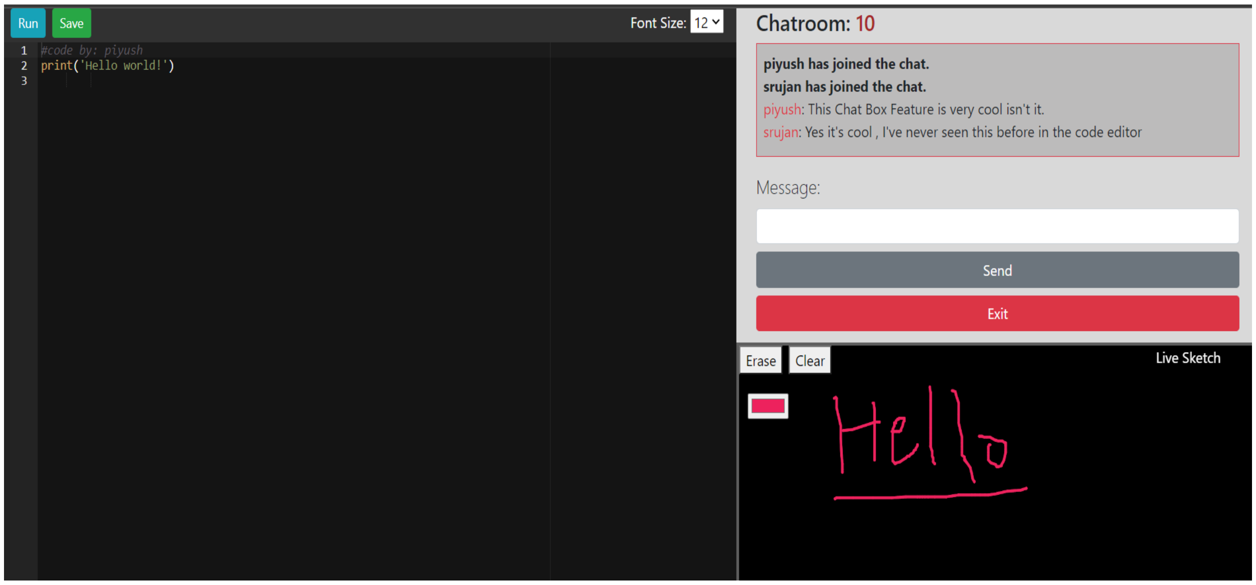
It is a black rectangular area present at the bottom right side of our colabedit. It provides users with the facility to draw or write text data.

It is quite similar to google whiteboard.It has functionalities such as color palette which provides (255*255*255) RGB color combinations.

Another functionality is the draw/erase button using which we can toggle between writing/deleting the data. The third feature provided was the reset button which cleared the window instantly for the user. Choosing a particular color from a color palette is individual operation , i.e. different users can use different colors for drawing on the canvas.

In this functionality the sync between the users is maintained by the drawing queue which acts as a temporary buffer to store the data until the other user has kept the canvas busy. We also take care of mouse position , if the cursor is outside the canvas boundary then drawing by dragging the mouse does not affect the canvas area.

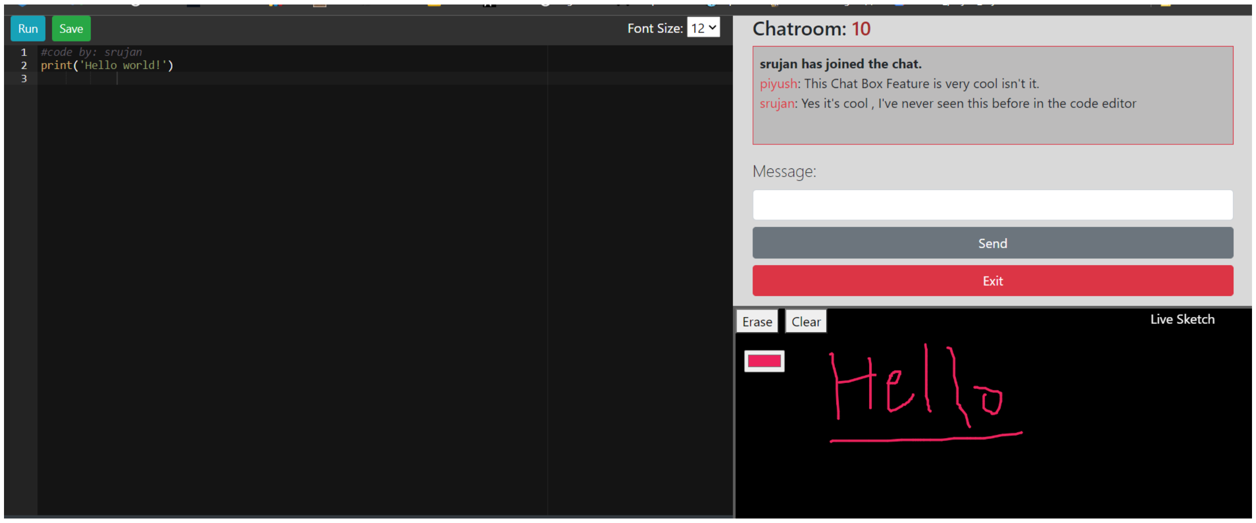
LIVE DEMO



```
#code by: piyush
print('Hello world!')
```

Chatroom: 10

piyush has joined the chat.
srujan has joined the chat.
piyush: This Chat Box Feature is very cool isn't it.
srujan: Yes it's cool , I've never seen this before in the code editor



```
#code by: srujan
print('Hello world!')
```

Chatroom: 10

srujan has joined the chat.
piyush: This Chat Box Feature is very cool isn't it.
srujan: Yes it's cool , I've never seen this before in the code editor

Here we can observe that in the case of both the users, namely piyush and srujan drawing board   is in sync with each other.

FUTURE WORK

Currently we support python language but in the future we want to support other languages like c++ , java,HTML,Javascript .

We want to add video calling feature in this editor so that it becomes easier for the people to communicate and stay connected with each other.

We will add more functionalities to our canvas such as changing brush strokes , providing different brush options,adjusting the size of eraser so that a large area can be erased easily.