FLAM Project Report

Project Title:

FLAM – Collaborative Drawing Canvas

1. Introduction

The FLAM Project is a simple web-based collaborative drawing app where more than one person can draw together on the same screen in real time.
The main idea is to allow users to connect to a shared canvas and see whatever others are drawing instantly.
It works just like a live whiteboard — when one user draws, every other connected user sees it happening at the same time.

This project helped me understand how real-time data exchange happens using **WebSockets**, and how to combine it with frontend technologies like HTML, CSS, and JavaScript.

2. Objectives

- To design an interactive drawing canvas using HTML5 and JavaScript.
- To connect multiple users using WebSockets for real-time updates.
- To include tools such as brush color, brush size, eraser, and clear options.
- To make the interface simple, clean, and responsive.
- To show how many users are currently active on the canvas.

3. Tools and Technologies Used

Component	Technology
Frontend	HTML5, CSS3, JavaScript
Backend	Node.js with Express
Real-Time Communication	WebSocket (`ws` library)

Code Editor	Visual Studio Code
Operating System	Windows 10

4. System Architecture

The project works on a **client-server model**.
Each user’s browser acts as a client, and all of them connect to one Node.js WebSocket server.
When someone draws, their browser sends the drawing data to the server, which immediately broadcasts it to all other users.

```
[User 1]  <-->  [WebSocket Server]  <-->  [User 2]
      ↑                 ↑                 ↑
   Draw/Erase     Broadcast to All    Draw/Undo/Clear
```

The same logic applies for clearing the screen or undoing actions.
Every user sees the same thing instantly.

5. Module Description

Module	Description
Drawing Module	Handles mouse movements and draws on the canvas.
Brush & Color Module	Lets users select color and brush thickness.
Eraser Module	Switches the brush color to white to erase drawings.
Undo/Redo Module	Keeps track of strokes and reverts/recovers them.
Clear Module	Clears the entire canvas for all users.
WebSocket Module	Manages real-time communication between users.
Active User Counter	Shows the number of users currently connected.

6. Results

- The project runs successfully on both local and multiple browser tabs.
- Drawing updates appear instantly for all connected users.

- Undo, redo, and clear functions reflect across all tabs.
- The interface is simple and works well on any screen size.

This proves that the WebSocket-based communication is implemented correctly and that multiple users can collaborate in real time.

7. Conclusion

The FLAM Project achieves its main goal of creating a **real-time collaborative drawing canvas**.
It clearly demonstrates the concept of two-way communication between server and clients using WebSockets.
Through this project, I learned about server-client communication, event handling, and state synchronization in JavaScript.

This can later be extended to add shapes, saving drawings, chat, or even voice features.

8. References

- Node.js Documentation — https://nodejs.org
- MDN WebSocket Guide — https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
- Express.js Documentation — https://expressjs.com

Submitted By

Name: Charitha Nalipireddy
Project Title: FLAM – Collaborative Drawing Canvas
Course: Computer Science and Technology
Institution: National Institute of Technology Silchar