



Basic Drawing Whiteboard with Real-Time Collaboration

CSCI 663GVA

December 4, 2024

Ivan Velo Castaneda

Daniel Adesoji

Ronald Targbeh



Agenda

- Project Overview
- Problem Domain
- Motivation
- Proposed Solution
- Task Breakdown for the Group
- Future Extensions
- Technical Overview
- Challenges Faced
- Demo
- Conclusion



Project Overview

- Objective: Develop a collaborative drawing whiteboard for real-time user interaction.
- Key Features:
 - Multiple users can draw simultaneously.
 - Limitation: Only one user can draw on the whiteboard at a time.
 - Users can save their whiteboards and export them as images.
 - Synchronization of drawing actions using socket programming.
 - Basic drawing tools: pen, color selection, eraser, undo/redo functionality.



Problem Domain

- Need for Collaborative Tools: Essential in remote work and learning environments.
- Existing Solutions: Often complex or feature-heavy.
- Our Solution: A lightweight, user-friendly tool focused on speed, simplicity, and responsiveness.



Motivation

- Hands-On Experience: Gain practical skills in socket programming and real-time synchronization.
- Project Opportunities: Explore efficient data sharing across users, client-server architecture, and user interface design.



Proposed Solution

- Whiteboard Features:
 - Pen Tool: Select colors and pen sizes.
 - Eraser: Erase parts of drawings.
 - Undo/Redo: Manage drawing actions.
 - Real-Time Updates: Immediate synchronization across users.
 - Clear Board: Clear the whiteboard for all users.



Task Breakdown for the Group

- Ivan:
 - Set up the server using sockets to handle real-time communication.
 - Manage connections and broadcasting updates to all clients.
 - Keep track of user sessions to know who is currently drawing.
 - Implement Login and Sign-Up Functionality: Develop the authentication system to allow users to register and log in.
 - Implement Saving Feature: Develop functionality to allow users to save their whiteboards, including the name and image data.
 - Implement Exporting Feature: Enable users to export their whiteboards as image files (e.g., PNG) or PDFs for easy sharing and printing.
 - Implement Real-Time Synchronization: Ensure drawing actions are synchronized between multiple users in real time.
 - Add Undo/Redo Functionality: Create a system that allows users to undo and redo their drawing actions seamlessly.
- Daniel and Ronald:
 - Build the frontend UI using HTML/CSS/JavaScript to create an intuitive user experience.
 - Implement drawing tools such as the pen, color picker, and eraser.
 - Connect drawing actions with socket events to ensure real-time updates.
 - Provide feedback and testing support to ensure that the user experience is seamless and responsive.

Future Extensions

- Potential Features:
 - User avatars or color codes for identification.
 - Simple chat function for user communication while drawing.
 - Support multiple users effectively



Technical Overview

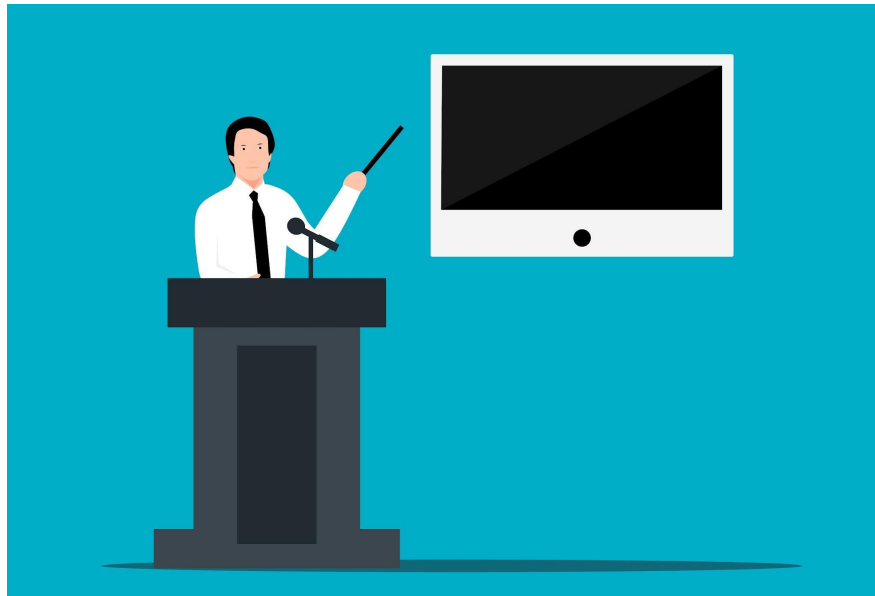
- User Interface Design:
 - Sign Up/Login Pages: User-friendly forms for authentication.
 - Dashboard: Overview of saved whiteboards with options to create new ones.
 - Drawing Interface: Tools for drawing, erasing, and managing sketches.
- Technical Stack:
 - Frontend: HTML/CSS for layout and design, JavaScript for interactivity.
 - Backend: Flask for web framework, Flask-SocketIO for real-time communication.
 - Database: SQLite for data persistence.
- Database Design:
 - Users Table: Stores user credentials (username, email, password).
 - Whiteboards Table: Stores whiteboard data (user_id, name, image, timestamp).
- Real-Time Collaboration:
 - Socket.IO Integration: Handles real-time drawing events.
 - Event Handling: Connection, disconnection, drawing events.
- Security Measures:
 - Password Hashing: Using Flask-Bcrypt for secure password storage.
 - CSRF Protection: Implemented using Flask-WTF to protect forms.
 - Input Validation: Ensures data integrity and prevents SQL injection.

Challenges Faced

- Real-Time Synchronization: Ensuring smooth collaboration without lag.
- User Interface Design: Creating an intuitive and responsive layout.
- Database Management: Handling concurrent access to whiteboard data.
- Synchronization of Drawing Actions: We were not able to fully synchronize drawing actions across multiple users effectively.



Demo



Conclusion

- Summary: A collaborative drawing tool enhancing teamwork and creativity.
- Areas for Improvement:
 - a. Synchronization: Work on better synchronizing drawing actions for one and multiple users to ensure a seamless experience.
 - b. Multi-User Support: Develop functionality to allow multiple users to draw simultaneously without conflicts.
 - c. Performance Optimization: Improve the performance of the application, especially under heavy load with multiple users.
 - d. User Interface Enhancements: Refine the UI for better usability and accessibility.

