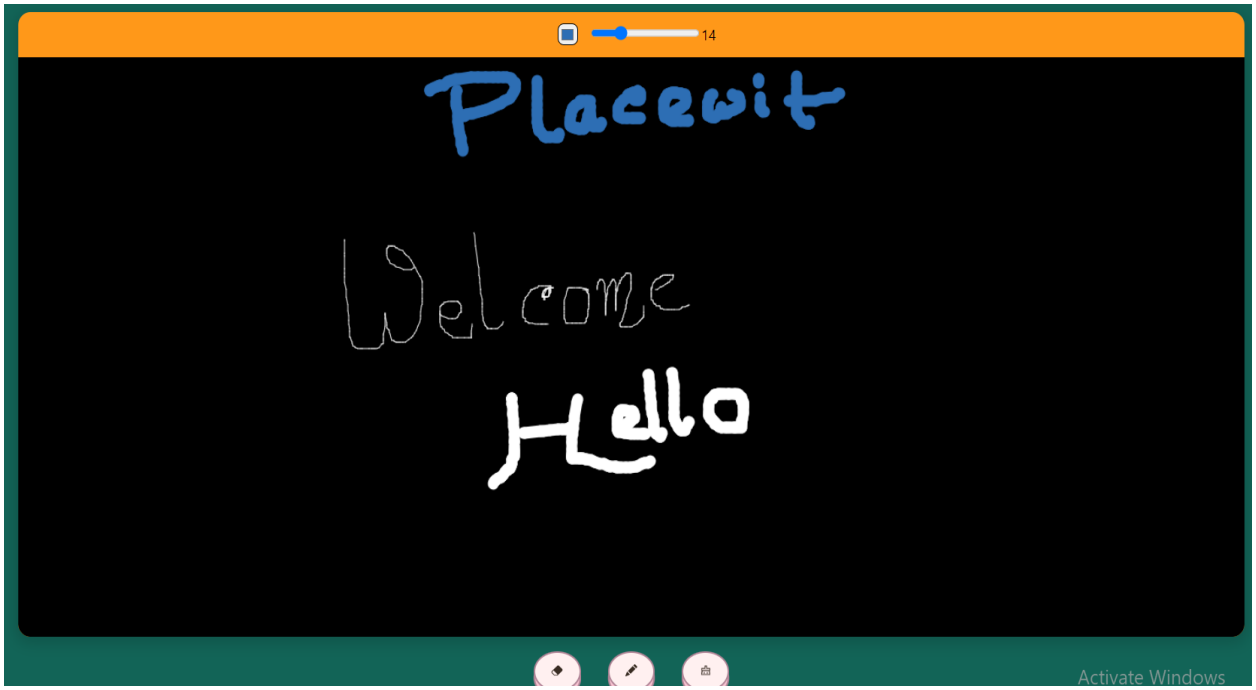

Realtime canvas

Project Link :
https://github.com/jaydip1235/placewit-js/tree/master/Realtime-Canvas

Jaydip Dey
Jadavpur University
https://linktr.ee/jaydipdey

The is a real-time, collaborative drawing application built on a stack of Socket.io, React, Express, and Node.js, featuring shared canvas functionality with capabilities such as drawing, erasing, dynamic brush size and color adjustment, and canvas clearing.

BACKEND

server1.js

- const express = require('express'); - Express.js is imported.
- const expressApp = express(); - An instance of an express application is created.
- const httpServer = require('http').createServer(expressApp); - An HTTP server is created using Express application as a listener for HTTP requests.
- const io = require('socket.io')(httpServer, {cors: { origin: true }}) - Socket.io is set up on the server created above. This allows us to use websockets to facilitate real-time communication. The CORS configuration allows any origin to connect to the server.
- const port = process.env.PORT || 5000; - The port for the server is set to 5000, or to the value provided in the environment variables.
- httpServer.listen(port, () => {console.log('Server running at', port)}) - The server is started on the specified port.
- io.on('connection', (socket) => {...}) - Socket.io listens for any new connections. A callback function is provided which is executed whenever a new user connects. This function logs that a user has come online, and sets up a listener for 'image-data' events.
- socket.on('image-data', (data) => {...}) - This is a listener for 'image-data' events. When an 'image-data' event is received from a client, it broadcasts the received data to all other connected clients.

FRONTEND

canvas1.js

Main Features:

1. Drawing - Users can draw on the canvas. The drawings are sent to a server using Socket.io and are also saved locally in state.
2. Color Picker - A color picker allows users to change the color of the brush.
3. Brush Size Control - A slider allows users to adjust the size of the brush.
4. Eraser - An eraser button allows users to erase parts of their drawings.
5. Clear Canvas - A button to clear the entire canvas.
6. Supports both mouse and touch events.

Code Breakdown:

The Canvas function is the main function of this code. It sets up the canvas, defines some helper functions for interacting with it, and renders the user interface.

- State Variables:
    - storeImgData & storeImgData1 - arrays used to store the image data.
    - brushColor - the current brush color.
    - brushRadius - the current brush size.
    - canvasDetails - contains information about the current state of the canvas including color, socketUrl, lineWidth.
- Utility Functions:
    - changeColor - change the brush color.
    - changeBrushSize - change the brush size.
    - eraser - change the brush to eraser.
    - pen - change the brush to pen.
    - clear - clear the entire canvas.
- Socket Setup: The useEffect hook is used to set up the socket.io connection and handle incoming 'image-data' events.
- Event Handlers: The second useEffect hook sets up event handlers for user input. It includes mouse and touch event handlers for drawing on the canvas.

- Draw and save: The draw and onSave functions handle the logic of drawing on the canvas and saving the drawing data. The drawn data is sent to the server via the socket.io connection and stored in storeImgData & storeImgData1.
- Rendering: The return statement renders the UI. It includes a color picker, a slider for changing the brush size, a canvas area for drawing, and buttons for erasing, switching to pen, and clearing the canvas.