Software Studio Homework 1 AI Reference

Name : Steward Melvin Tang 鄧智祥

Student ID: 112006223

1. I only use Chat GPT as the generative AI to brainstorm ideas, ask questions & solution on some of my issues. And I did not copy & paste directly the generated code

Code Segments that uses AI:

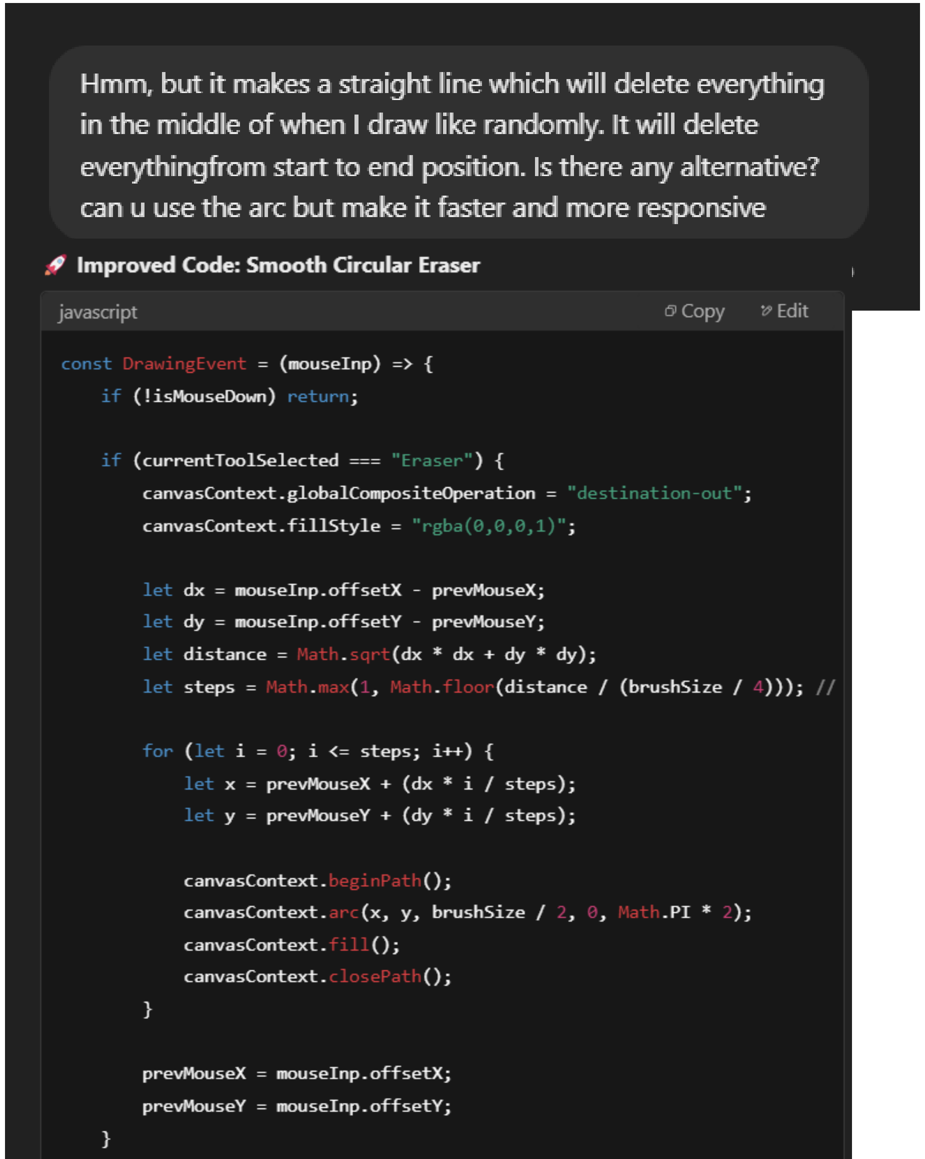
- **Eraser System (Line 351-368)**

As in the tutorial video I watched, the eraser uses a simple “white” brush to act as the eraser which is not allowed as stated in the guide pdf file. So I came across the thing called the “*globalCompositeOperation*” which I found on CanvasRenderingContext2D: globalCompositeOperation property - Web APIs | MDN

But at first, I uses the simple moveTo and lineTo mechanism (similar to brush) and it doesn’t really feel natural. Then I asked chat gpt for a better implementation and here is what I ask and what it gave me

- **Eraser System (Line 351-368)**

Here is what I asked and what it gave me:

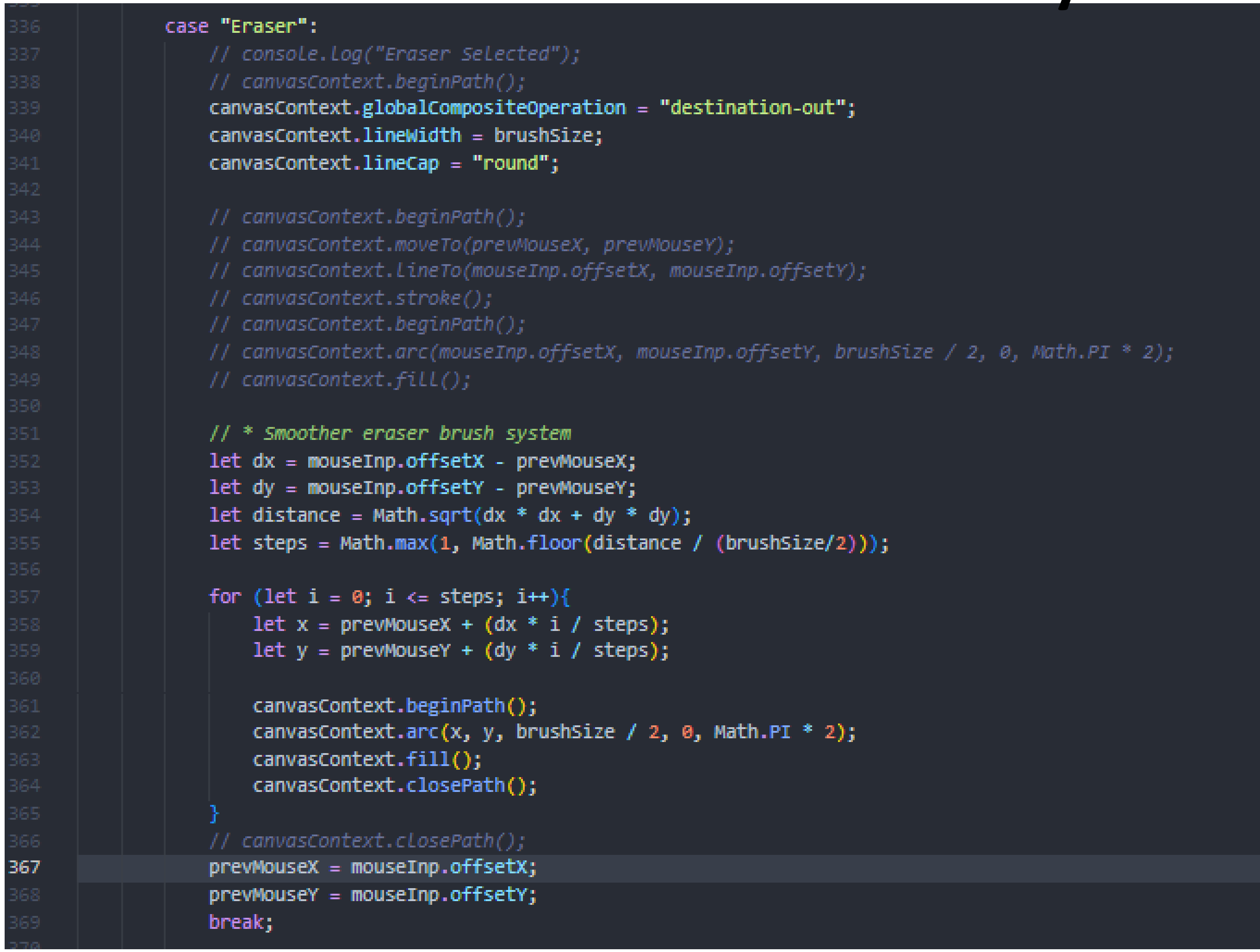


Hmm, but it makes a straight line which will delete everything in the middle of when I draw like randomly. It will delete everythingfrom start to end position. Is there any alternative? can u use the arc but make it faster and more responsive

🚀 **Improved Code: Smooth Circular Eraser**

```javascript
const DrawingEvent = (mouseInp) => {
    if (!isMouseDown) return;

    if (currentToolSelected === "Eraser") {
        canvasContext.globalCompositeOperation = "destination-out";
        canvasContext.fillStyle = "rgba(0,0,0,1)";

        let dx = mouseInp.offsetX - prevMouseX;
        let dy = mouseInp.offsetY - prevMouseY;
        let distance = Math.sqrt(dx * dx + dy * dy);
        let steps = Math.max(1, Math.floor(distance / (brushSize / 4))); //

        for (let i = 0; i <= steps; i++) {
            let x = prevMouseX + (dx * i / steps);
            let y = prevMouseY + (dy * i / steps);

            canvasContext.beginPath();
            canvasContext.arc(x, y, brushSize / 2, 0, Math.PI * 2);
            canvasContext.fill();
            canvasContext.closePath();
        }

        prevMouseX = mouseInp.offsetX;
        prevMouseY = mouseInp.offsetY;
    }
```

My code

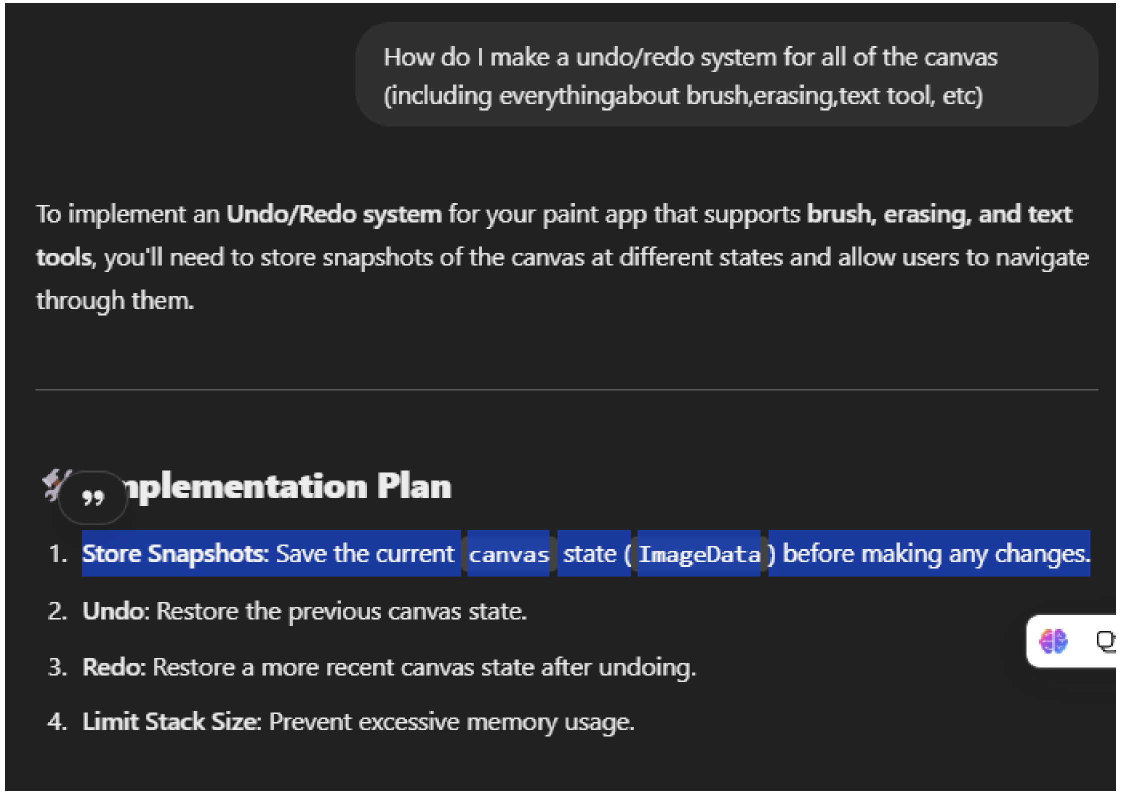


```javascript
336         case "Eraser":
337             // console.log("Eraser Selected");
338             // canvasContext.beginPath();
339             canvasContext.globalCompositeOperation = "destination-out";
340             canvasContext.lineWidth = brushSize;
341             canvasContext.lineCap = "round";
342 
343             // canvasContext.beginPath();
344             // canvasContext.moveTo(prevMouseX, prevMouseY);
345             // canvasContext.lineTo(mouseInp.offsetX, mouseInp.offsetY);
346             // canvasContext.stroke();
347             // canvasContext.beginPath();
348             // canvasContext.arc(mouseInp.offsetX, mouseInp.offsetY, brushSize / 2, 0, Math.PI * 2);
349             // canvasContext.fill();
350 
351             // * Smoother eraser brush system
352             let dx = mouseInp.offsetX - prevMouseX;
353             let dy = mouseInp.offsetY - prevMouseY;
354             let distance = Math.sqrt(dx * dx + dy * dy);
355             let steps = Math.max(1, Math.floor(distance / (brushSize/2)));
356 
357             for (let i = 0; i <= steps; i++){
358                 let x = prevMouseX + (dx * i / steps);
359                 let y = prevMouseY + (dy * i / steps);
360 
361                 canvasContext.beginPath();
362                 canvasContext.arc(x, y, brushSize / 2, 0, Math.PI * 2);
363                 canvasContext.fill();
364                 canvasContext.closePath();
365             }
366             // canvasContext.closePath();
367             prevMouseX = mouseInp.offsetX;
368             prevMouseY = mouseInp.offsetY;
369             break;
```

Code Segments that uses AI:

- **Undo & Redo System (Brainstorm)**

My code

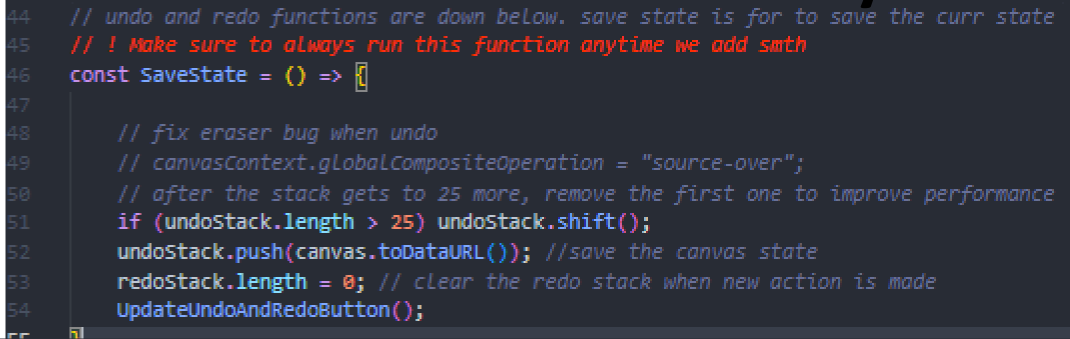


How do I make a undo/redo system for all of the canvas (including everythingabout brush,erasing,text tool, etc)

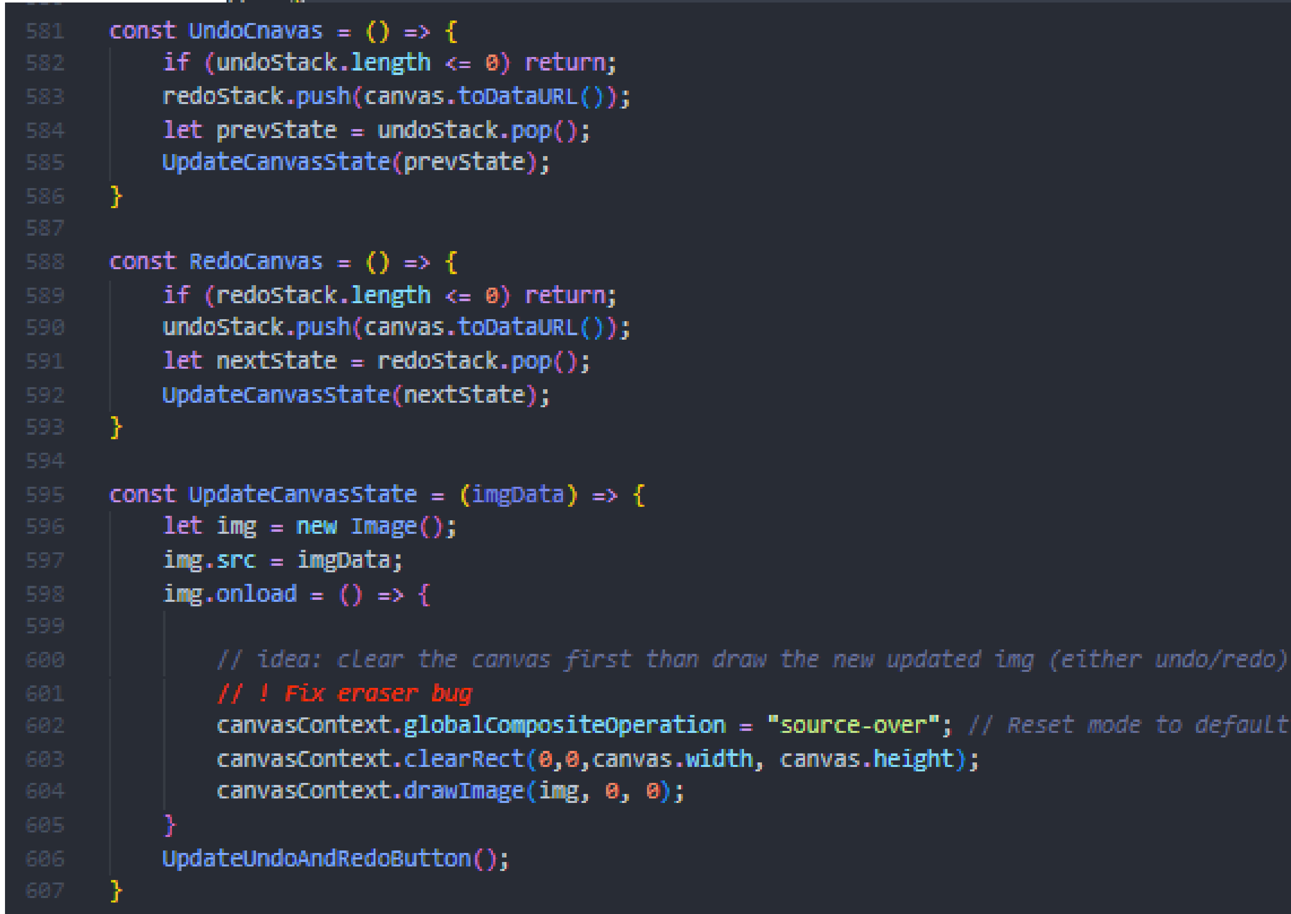
To implement an **Undo/Redo system** for your paint app that supports **brush, erasing, and text tools**, you'll need to store snapshots of the canvas at different states and allow users to navigate through them.

🛠 Implementation Plan

1. **Store Snapshots:** Save the current `canvas` state (`ImageData`) before making any changes.
2. **Undo:** Restore the previous canvas state.
3. **Redo:** Restore a more recent canvas state after undoing.
4. **Limit Stack Size:** Prevent excessive memory usage.



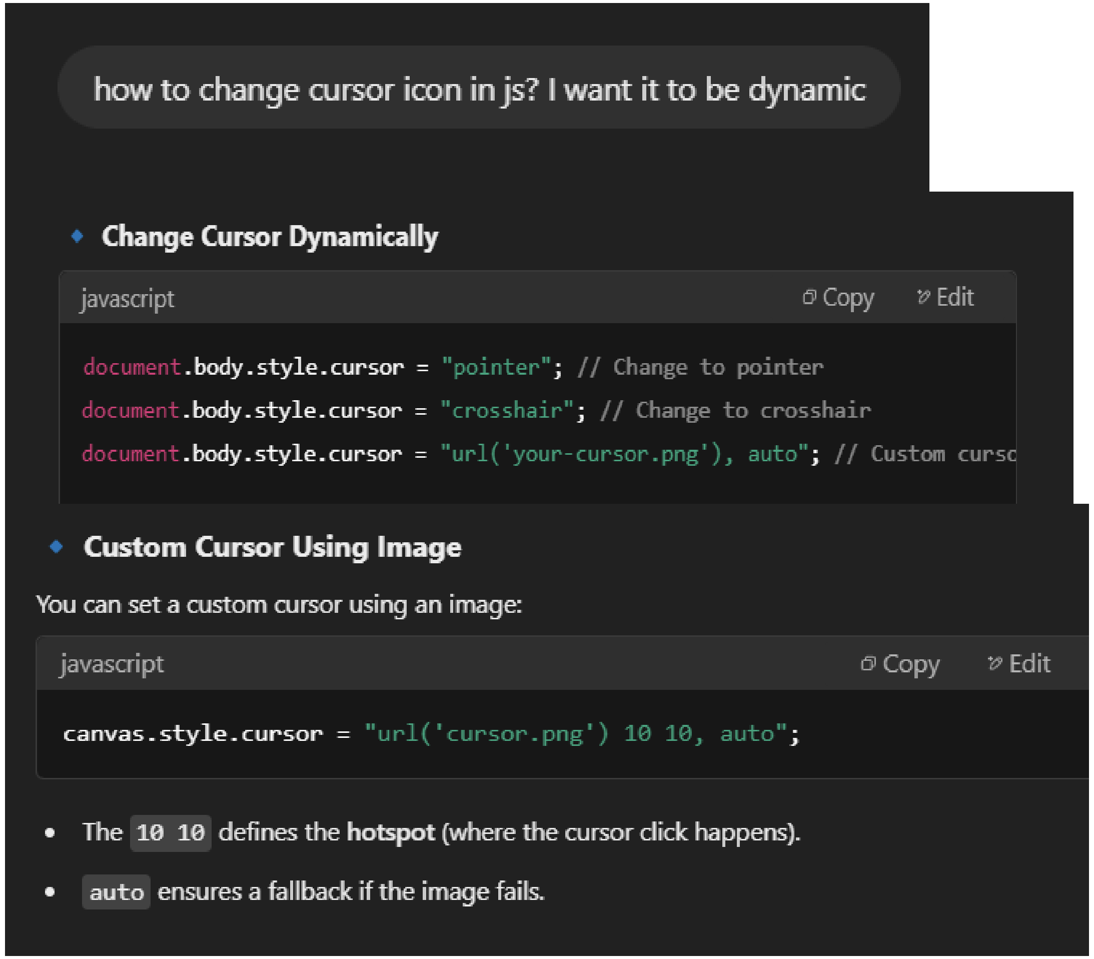
```
// undo and redo functions are down below. save state is for to save the curr state
// ! Make sure to always run this function anytime we add smth
const SaveState = () => {

    // fix eraser bug when undo
    // canvasContext.globalCompositeOperation = "source-over";
    // after the stack gets to 25 more, remove the first one to improve performance
    if (undoStack.length > 25) undoStack.shift();
    undoStack.push(canvas.toDataURL()); //save the canvas state
    redoStack.length = 0; // clear the redo stack when new action is made
    UpdateUndoAndRedoButton();
}
```



```
const UndoCnavas = () => {
    if (undoStack.length <= 0) return;
    redoStack.push(canvas.toDataURL());
    let prevState = undoStack.pop();
    UpdateCanvasState(prevState);
}

const RedoCanvas = () => {
    if (redoStack.length <= 0) return;
    undoStack.push(canvas.toDataURL());
    let nextState = redoStack.pop();
    UpdateCanvasState(nextState);
}

const UpdateCanvasState = (imgData) => {
    let img = new Image();
    img.src = imgData;
    img.onload = () => {

        // idea: clear the canvas first than draw the new updated img (either undo/redo)
        // ! Fix eraser bug
        canvasContext.globalCompositeOperation = "source-over"; // Reset mode to default
        canvasContext.clearRect(0,0,canvas.width, canvas.height);
        canvasContext.drawImage(img, 0, 0);
    }
    UpdateUndoAndRedoButton();
}
```

Code Segments that uses AI:

- **Custom Cursor (Dynamic Size) - brainstorm**



how to change cursor icon in js? I want it to be dynamic

◆ **Change Cursor Dynamically**

```javascript
document.body.style.cursor = "pointer"; // Change to pointer
document.body.style.cursor = "crosshair"; // Change to crosshair
document.body.style.cursor = "url('your-cursor.png'), auto"; // Custom curso
```

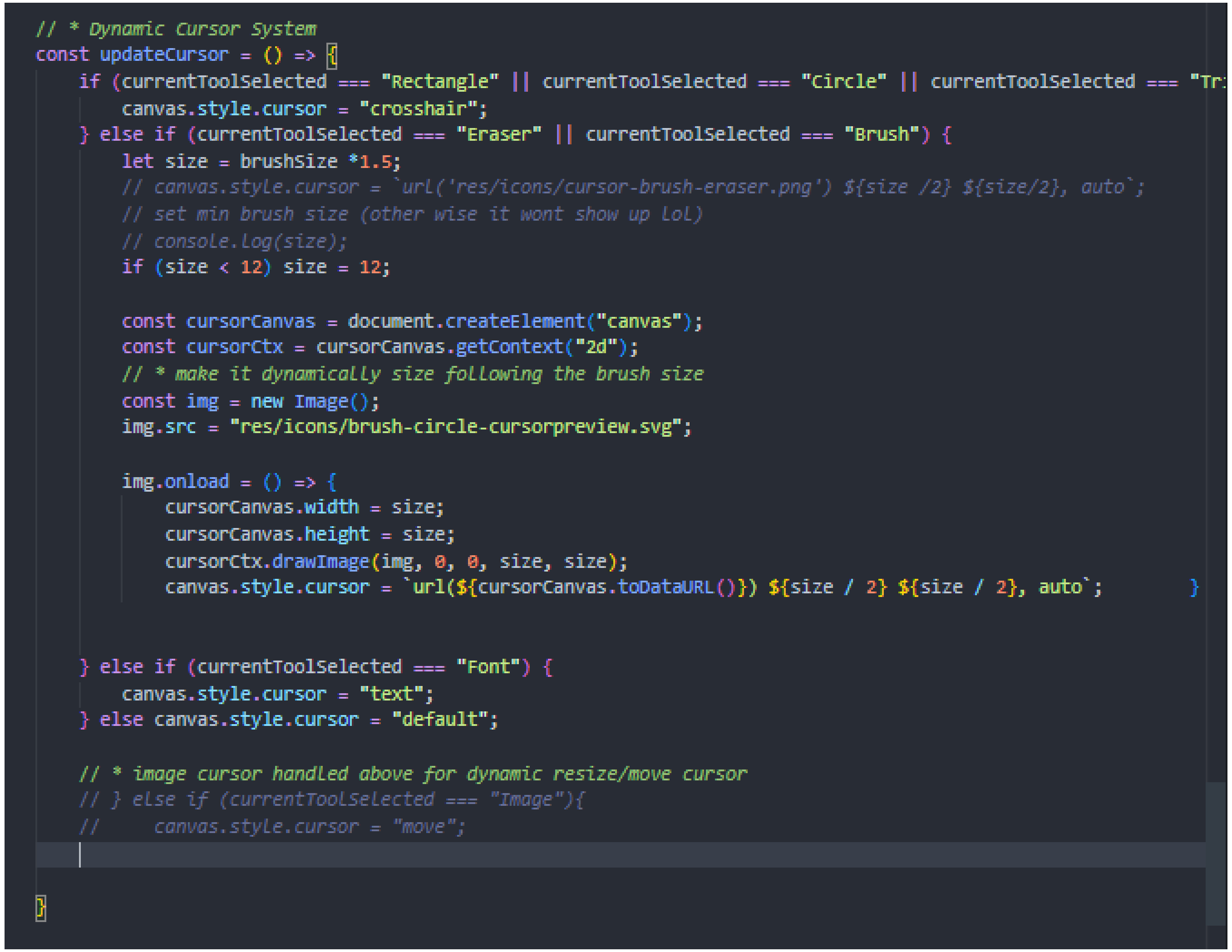
◆ **Custom Cursor Using Image**

You can set a custom cursor using an image:

```javascript
canvas.style.cursor = "url('cursor.png') 10 10, auto";
```

- The `10 10` defines the **hotspot** (where the cursor click happens).
- `auto` ensures a fallback if the image fails.

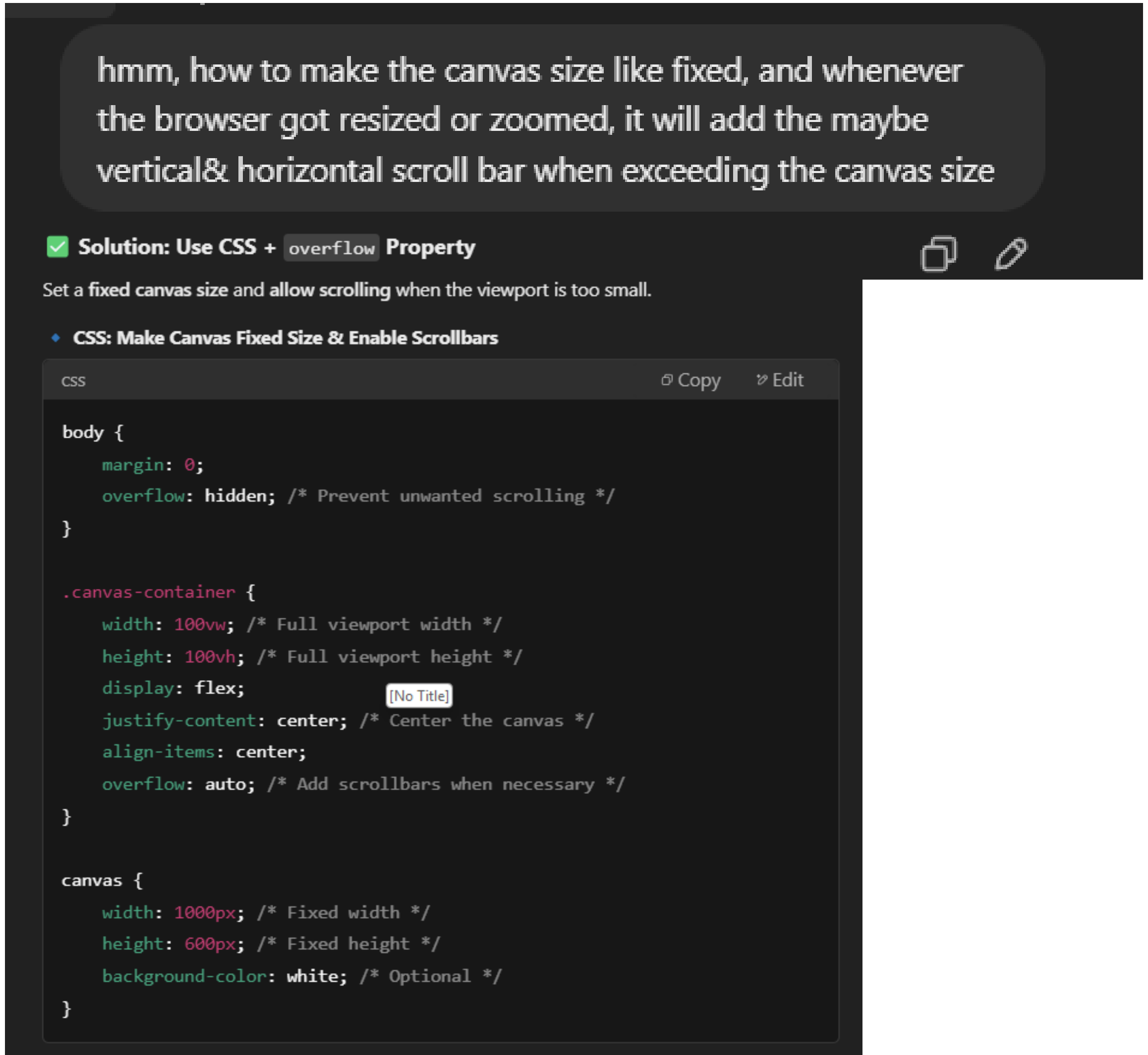
My code



```javascript
// * Dynamic Cursor System
const updateCursor = () => {
    if (currentToolSelected === "Rectangle" || currentToolSelected === "Circle" || currentToolSelected === "Tr
        canvas.style.cursor = "crosshair";
    } else if (currentToolSelected === "Eraser" || currentToolSelected === "Brush") {
        let size = brushSize *1.5;
        // canvas.style.cursor = `url('res/icons/cursor-brush-eraser.png') ${size /2} ${size/2}, auto`;
        // set min brush size (other wise it wont show up lol)
        // console.log(size);
        if (size < 12) size = 12;

        const cursorCanvas = document.createElement("canvas");
        const cursorCtx = cursorCanvas.getContext("2d");
        // * make it dynamically size following the brush size
        const img = new Image();
        img.src = "res/icons/brush-circle-cursorpreview.svg";

        img.onload = () => {
            cursorCanvas.width = size;
            cursorCanvas.height = size;
            cursorCtx.drawImage(img, 0, 0, size, size);
            canvas.style.cursor = `url(${cursorCanvas.toDataURL()}) ${size / 2} ${size / 2}, auto`;        }


    } else if (currentToolSelected === "Font") {
        canvas.style.cursor = "text";
    } else canvas.style.cursor = "default";

    // * image cursor handled above for dynamic resize/move cursor
    // } else if (currentToolSelected === "Image"){
    //     canvas.style.cursor = "move";


}
```

Code Segments that uses AI:

- **Fixed Canvas Size (Responsive) - Brainstorm**



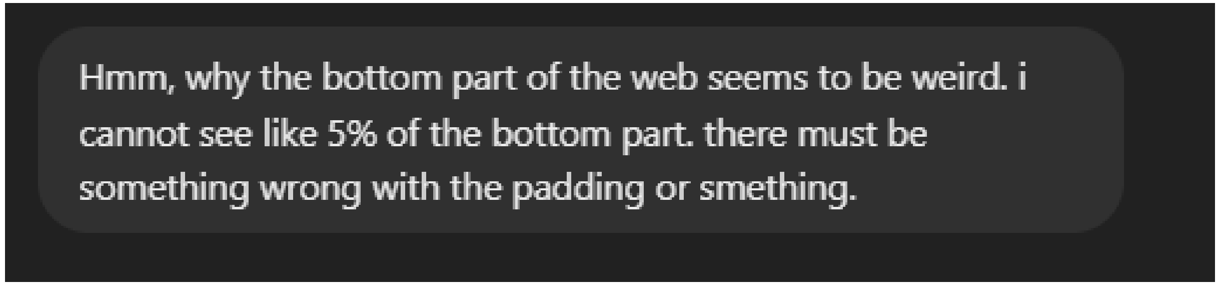
hmm, how to make the canvas size like fixed, and whenever the browser got resized or zoomed, it will add the maybe vertical& horizontal scroll bar when exceeding the canvas size

✅ **Solution: Use CSS + `overflow` Property**

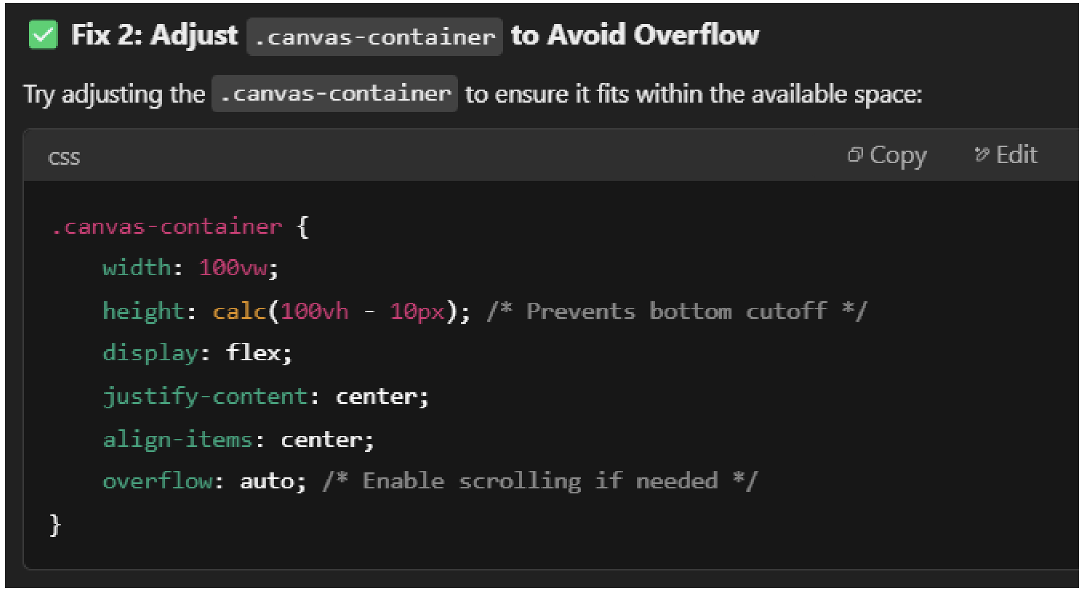
Set a **fixed canvas size** and **allow scrolling** when the viewport is too small.

🔹 **CSS: Make Canvas Fixed Size & Enable Scrollbars**

```css
body {
    margin: 0;
    overflow: hidden; /* Prevent unwanted scrolling */
}

.canvas-container {
    width: 100vw; /* Full viewport width */
    height: 100vh; /* Full viewport height */
    display: flex;
    justify-content: center; /* Center the canvas */
    align-items: center;
    overflow: auto; /* Add scrollbars when necessary */
}

canvas {
    width: 1000px; /* Fixed width */
    height: 600px; /* Fixed height */
    background-color: white; /* Optional */
}
```



Hmm, why the bottom part of the web seems to be weird. i cannot see like 5% of the bottom part. there must be something wrong with the padding or smething.



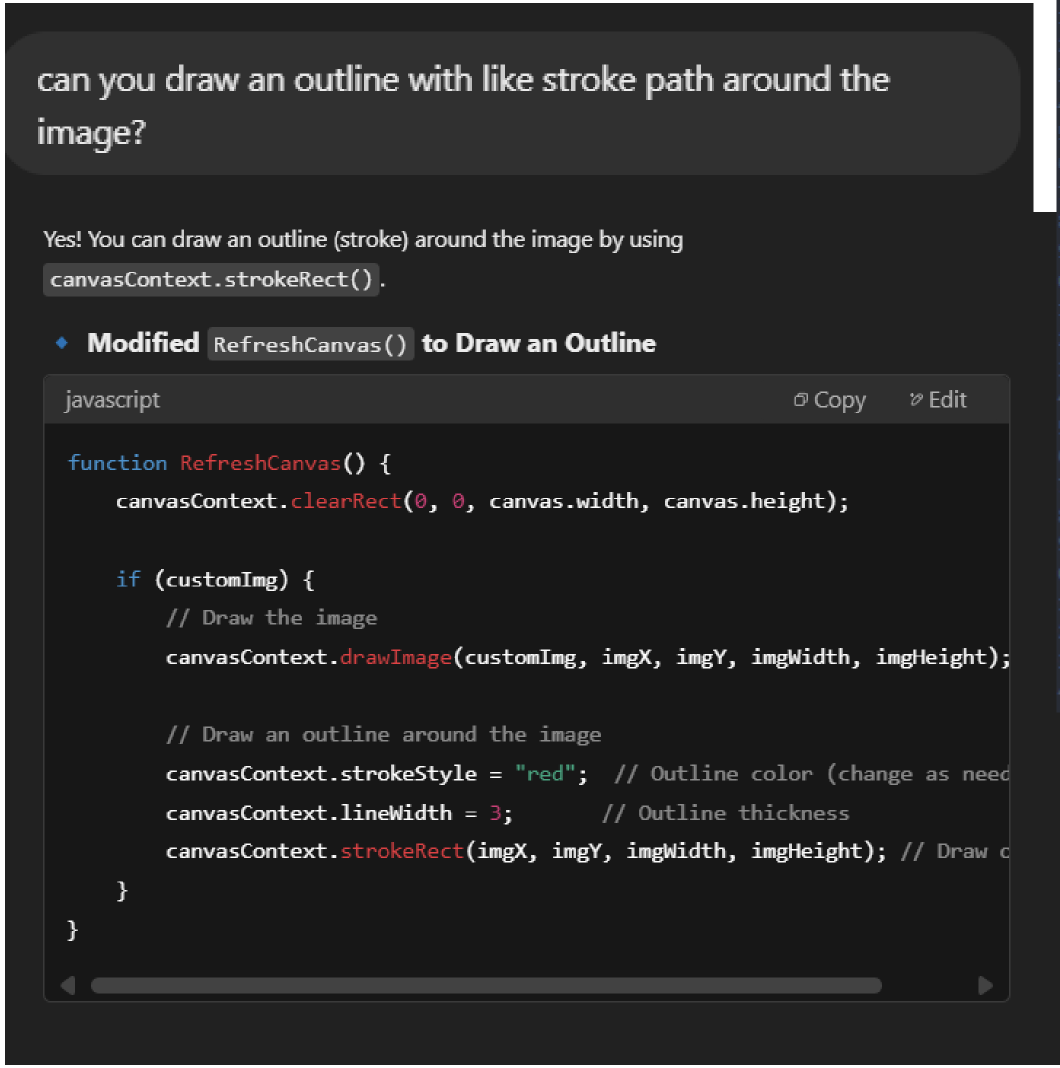
✅ **Fix 2: Adjust `.canvas-container` to Avoid Overflow**

Try adjusting the `.canvas-container` to ensure it fits within the available space:

```css
.canvas-container {
    width: 100vw;
    height: calc(100vh - 10px); /* Prevents bottom cutoff */
    display: flex;
    justify-content: center;
    align-items: center;
    overflow: auto; /* Enable scrolling if needed */
}
```

Code Segments that uses AI:

- **Image Transformation Tool - Brainstorm**



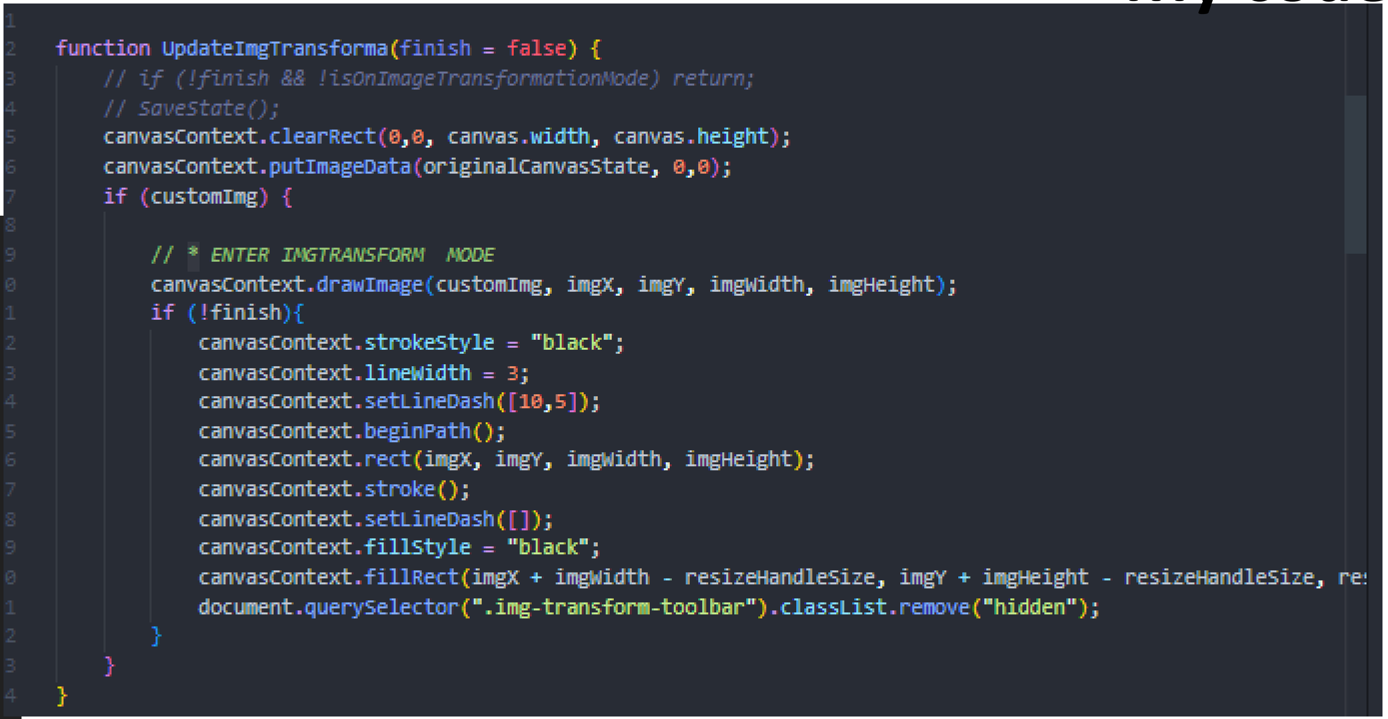
can you draw an outline with like stroke path around the image?

Yes! You can draw an outline (stroke) around the image by using `canvasContext.strokeRect()`.

◆ **Modified `RefreshCanvas()` to Draw an Outline**

```javascript
function RefreshCanvas() {
    canvasContext.clearRect(0, 0, canvas.width, canvas.height);

    if (customImg) {
        // Draw the image
        canvasContext.drawImage(customImg, imgX, imgY, imgWidth, imgHeight);

        // Draw an outline around the image
        canvasContext.strokeStyle = "red";  // Outline color (change as need
        canvasContext.lineWidth = 3;       // Outline thickness
        canvasContext.strokeRect(imgX, imgY, imgWidth, imgHeight); // Draw c
    }
}
```

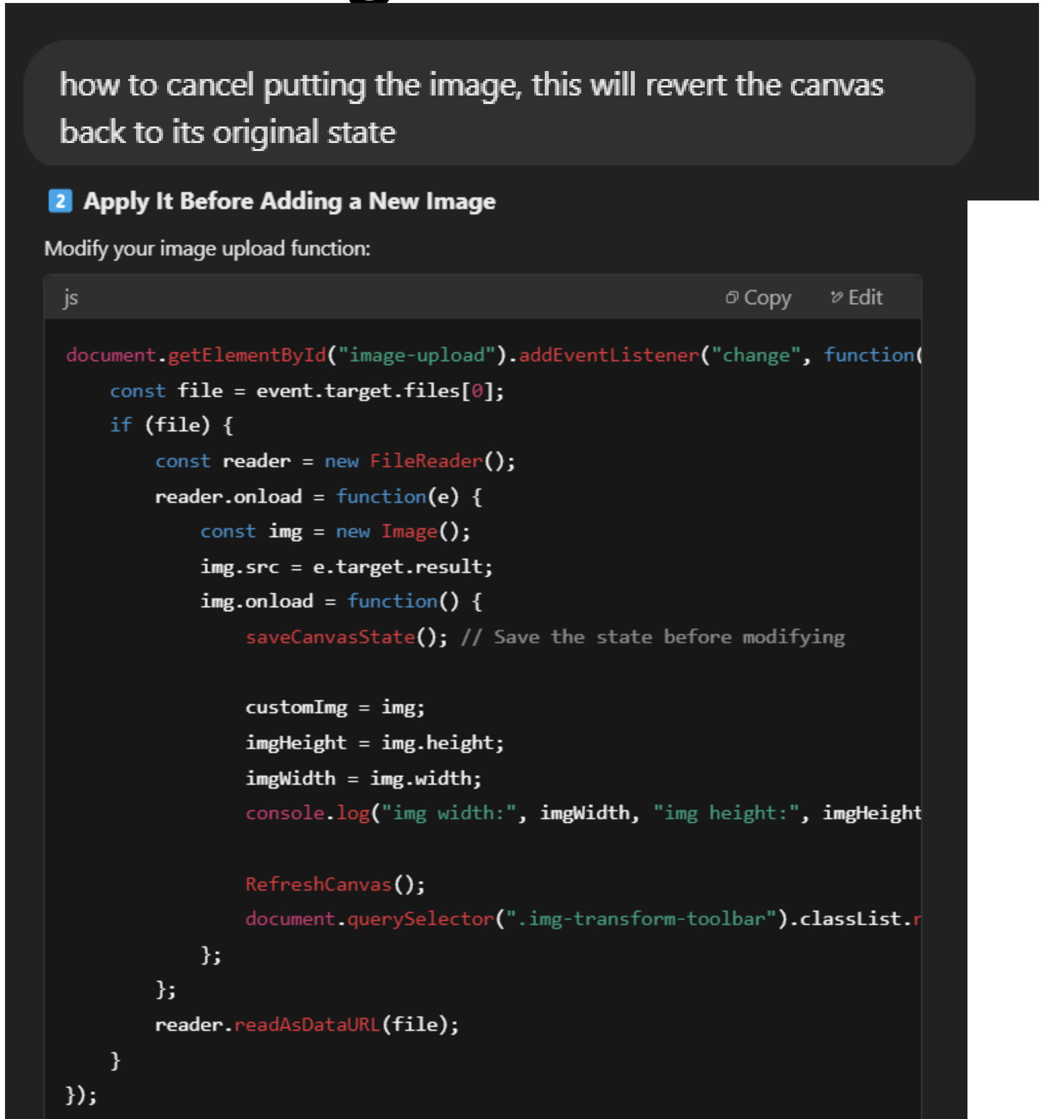
My code



```javascript
function UpdateImgTransforma(finish = false) {
    // if (!finish && !isOnImageTransformationMode) return;
    // SaveState();
    canvasContext.clearRect(0,0, canvas.width, canvas.height);
    canvasContext.putImageData(originalCanvasState, 0,0);
    if (customImg) {

        // * ENTER IMGTRANSFORM  MODE
        canvasContext.drawImage(customImg, imgX, imgY, imgWidth, imgHeight);
        if (!finish){
            canvasContext.strokeStyle = "black";
            canvasContext.lineWidth = 3;
            canvasContext.setLineDash([10,5]);
            canvasContext.beginPath();
            canvasContext.rect(imgX, imgY, imgWidth, imgHeight);
            canvasContext.stroke();
            canvasContext.setLineDash([]);
            canvasContext.fillStyle = "black";
            canvasContext.fillRect(imgX + imgWidth - resizeHandleSize, imgY + imgHeight - resizeHandleSize, re
            document.querySelector(".img-transform-toolbar").classList.remove("hidden");
        }
    }
}
```

Code Segments that uses AI:

- **Image Transformation Tool - Brainstorm**



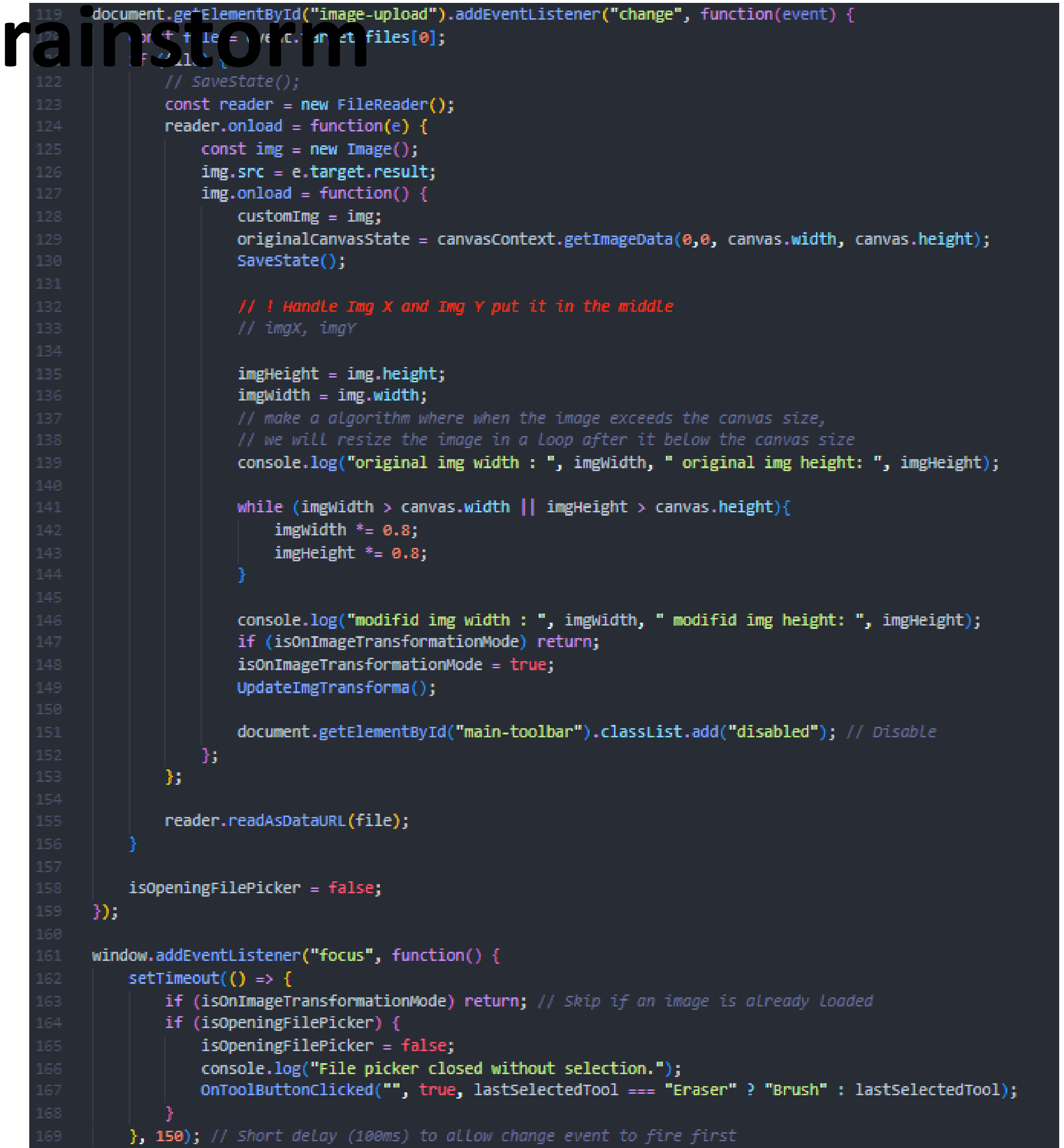
how to cancel putting the image, this will revert the canvas back to its original state

2 Apply It Before Adding a New Image

Modify your image upload function:

```js
document.getElementById("image-upload").addEventListener("change", function(
    const file = event.target.files[0];
    if (file) {
        const reader = new FileReader();
        reader.onload = function(e) {
            const img = new Image();
            img.src = e.target.result;
            img.onload = function() {
                saveCanvasState(); // Save the state before modifying

                customImg = img;
                imgHeight = img.height;
                imgWidth = img.width;
                console.log("img width:", imgWidth, "img height:", imgHeight

                RefreshCanvas();
                document.querySelector(".img-transform-toolbar").classList.r
            };
        };
        reader.readAsDataURL(file);
    }
});
```

My code



```js
document.getElementById("image-upload").addEventListener("change", function(event) {
    [illegible] files[0];
    [illegible] {
        // SaveState();
        const reader = new FileReader();
        reader.onload = function(e) {
            const img = new Image();
            img.src = e.target.result;
            img.onload = function() {
                customImg = img;
                originalCanvasState = canvasContext.getImageData(0,0, canvas.width, canvas.height);
                SaveState();

                // ! Handle Img X and Img Y put it in the middle
                // imgX, imgY

                imgHeight = img.height;
                imgWidth = img.width;
                // make a algorithm where when the image exceeds the canvas size,
                // we will resize the image in a loop after it below the canvas size
                console.log("original img width : ", imgWidth, " original img height: ", imgHeight);

                while (imgWidth > canvas.width || imgHeight > canvas.height){
                    imgWidth *= 0.8;
                    imgHeight *= 0.8;
                }

                console.log("modifid img width : ", imgWidth, " modifid img height: ", imgHeight);
                if (isOnImageTransformationMode) return;
                isOnImageTransformationMode = true;
                UpdateImgTransforma();

                document.getElementById("main-toolbar").classList.add("disabled"); // Disable
            };
        };

        reader.readAsDataURL(file);
    }

    isOpeningFilePicker = false;
});

window.addEventListener("focus", function() {
    setTimeout(() => {
        if (isOnImageTransformationMode) return; // Skip if an image is already loaded
        if (isOpeningFilePicker) {
            isOpeningFilePicker = false;
            console.log("File picker closed without selection.");
            OnToolButtonClicked("", true, lastSelectedTool === "Eraser" ? "Brush" : lastSelectedTool);
        }
    }, 150); // Short delay (100ms) to allow change event to fire first
```