$('#wp2023-apply-crop').on('click', function() {

        if (currentCropper && typeof currentImageIndex !== 'undefined') {

            var croppedCanvas = currentCropper.getCroppedCanvas();

            var cmToPixels = 37.7952755906; // Conversion factor from cm to pixels

            // Assuming you have a way to determine the current size index

            var selectedSizeDiv = $('.wrap-size .size.selected');

            var marginCm = parseFloat(selectedSizeDiv.data('margin')) ||

                1.5; // You need to define this function

            var totalMarginAndBorderPixels = marginCm \*

                cmToPixels; // Total size of margin and border in pixels

            var borderThickness = 1; // Thin but visible border thickness in pixels

            // Calculate the margin size excluding the border

            var marginPixels = totalMarginAndBorderPixels - borderThickness;

            // Calculate total additional space needed for margin and border

            var totalExtraSpace = 2 \* totalMarginAndBorderPixels;

            // Adjust the canvas size to include margin and border

            var canvasWithMarginAndBorder = document.createElement('canvas');

            canvasWithMarginAndBorder.width = croppedCanvas.width + totalExtraSpace;

            canvasWithMarginAndBorder.height = croppedCanvas.height + totalExtraSpace;

            var ctx = canvasWithMarginAndBorder.getContext('2d');

            // Draw the thin black border

            ctx.fillStyle = "#000000"; // Border color (black)

            ctx.fillRect(0, 0, canvasWithMarginAndBorder.width, canvasWithMarginAndBorder.height);

            // Draw the white margin inside the black border

            ctx.fillStyle = "#ffffff"; // Margin color (white)

            ctx.fillRect(borderThickness, borderThickness,

                canvasWithMarginAndBorder.width - 2 \* borderThickness,

                canvasWithMarginAndBorder.height - 2 \* borderThickness);

            // Draw the original image inside the margin

            ctx.drawImage(croppedCanvas, totalMarginAndBorderPixels, totalMarginAndBorderPixels);

            var currentEdits = imageEdits[currentImageIndex] || {};

            // Capture the current index before incrementing

            let capturedIndex = currentImageIndex;

            // Convert canvas to Blob and update the validFiles array

            canvasWithMarginAndBorder.toBlob(function(blob) {

                croppedImages.push({

                    blob: blob,

                    index: capturedIndex

                });

                // Optionally, update the thumbnail preview

                updateThumbnailPreview(canvasWithMarginAndBorder, capturedIndex);

            }, 'image/png'); // Adjust the MIME type as needed

            if (capturedIndex < imageFiles.length - 1) {

                currentImageIndex++;

                startEditingImage(currentImageIndex); // Start editing the next image

            } else {

                // All images have been processed, now you can close the modal and clean up

                $('#image-edit-modal').hide();

                if (currentCropper) {

                    currentCropper.destroy();

                    currentCropper = null;

                }

                currentImageIndex = null; // Reset the index or set it to the initial state

                // console.log('All images have been edited.');

            }

        }

    });