function validateAndSubmit() {

       if (validateForm()) {

          document.forms[0].submit();

        }

        }

        function validateForm() {

        let imageDatas = false

           console.log("validate form");

           clearErrorMessages();

           const name = document.getElementsByName('productName')[0].value;

           const description = document.getElementsByName('descriptionData')[0].value;

           const brand = document.getElementsByName('brand')[0].value;

           const price = document.getElementsByName('regularPrice')[0].value;

           const saleprice = document.getElementsByName('salePrice')[0].value;

           const color = document.getElementsByName('color')[0].value;

           const category = document.getElementsByName('category')[0].value;

           const images = document.getElementById('input1')

           const quantity=document.getElementsByName('quantity')

           imageDatas = document.getElementById("imageDatas")?.value;

           let isValid = true;

           if (name.trim() === "") {

   displayErrorMessage('productName-error', 'Please enter a product name.');

   isValid = false;

}

           if (description=== '') {

               displayErrorMessage('description-error', 'Please enter a product description.');

               isValid = false;

           }

           if ( parseInt(quantity) < 0) {

               displayErrorMessage('quantity-error', 'Please enter a valid non-negative quantity.');

               isValid = false;

           }

           if (!/^\d+(\.\d{1,2})?$/.test(price) || parseFloat(price) < 0) {

               displayErrorMessage('regularPrice-error', 'Please enter a valid non-negative price.');

               isValid = false;

           }

          if (!/^\d+(\.\d{1,2})?$/.test(saleprice) || parseFloat(saleprice) < 0) {

               displayErrorMessage('salePrice-error', 'Please enter a valid non-negative price.');

               isValid = false;

           }

           if (color.trim() === "") {

               displayErrorMessage('color-error', 'Please enter a color.');

               isValid = false;

           }

           if (!imageDatas && images.files.length === 0) {

               displayErrorMessage("images-error",'Please select an image.');

               isValid = false;

           }

           return isValid;

       }

       function displayErrorMessage(elementId, message) {

           var errorElement = document.getElementById(elementId);

           errorElement.innerText = message;

           errorElement.style.display = "block";

       }

       function clearErrorMessages() {

           const errorElements = document.getElementsByClassName('error-message');

           Array.from(errorElements).forEach(element => {

               element.innerText = '';

           });

           const errorMessage = document.getElementById('errorMessage');

       }

    function viewImage1(event) {

           document.getElementById('imgView1').src = URL.createObjectURL(event.target.files[0])

       }

       function viewImage2(event) {

           document.getElementById('imgView2').src = URL.createObjectURL(event.target.files[0])

       }

       function viewImage3(event) {

           document.getElementById('imgView3').src = URL.createObjectURL(event.target.files[0])

       }

       function viewImage4(event) {

           document.getElementById('imgView4').src = URL.createObjectURL(event.target.files[0])

       }

       function viewImage(event, index) {

           let input = event.target;

           let reader = new FileReader();

           reader.onload = function () {

               let dataURL = reader.result;

               let image = document.getElementById('imgView' + index);

               image.src = dataURL;

               let cropper = new Cropper(image, {

                   aspectRatio: 1,

                   viewMode: 1,

                   guides: true,

                   background: false,

                   autoCropArea: 1,

                   zoomable: true

               });

               let cropperContainer = document.querySelector('#croppedImg' + index).parentNode;

               cropperContainer.style.display = 'block';

               let saveButton = document.querySelector('#saveButton' + index);

               saveButton.addEventListener('click', async function () {

                   let croppedCanvas = cropper.getCroppedCanvas();

                   let croppedImage = document.getElementById("croppedImg" + index);

                   croppedImage.src = croppedCanvas.toDataURL('image/jpeg', 1.0);

                   let timestamp = new Date().getTime();

                   let fileName = `cropped-img-${timestamp}-${index}.png`;

                   await croppedCanvas.toBlob(blob => {

                       let input = document.getElementById('input' + index);

                       let imgFile = new File([blob], fileName, blob)

                       const fileList = new DataTransfer();

                       fileList.items.add(imgFile);

                       input.files = fileList.files

                   });

                   cropperContainer.style.display = 'none';

               });

           };

           reader.readAsDataURL(input.files[0]);

       }

       const selectedImages = [];

   document.getElementById("imageInput").addEventListener("change", handleFileSelect);

       function handleFileSelect(event) {

           const addedImagesContainer = document.getElementById("addedImagesContainer");

           addedImagesContainer.innerHTML = "";

           const files = event.target.files;

           for (let i = 0; i < files.length; i++) {

               const file = files[i];

               selectedImages.push(file);

               const thumbnail = document.createElement("div");

               thumbnail.classList.add("thumbnail");

               const img = document.createElement("img");

               img.src = URL.createObjectURL(file);

               img.alt = "thumbnail";

               img.style.width = "50px";

               img.style.height = "auto";

               const removeIcon = document.createElement("span");

               removeIcon.classList.add("remove-icon");

               removeIcon.innerHTML = "&times;";

               removeIcon.addEventListener("click", function () {

                   const index = selectedImages.indexOf(file);

                   if (index !== -1) {

                       selectedImages.splice(index, 1);

                   }

                   thumbnail.remove();

               });

               thumbnail.appendChild(img);

               thumbnail.appendChild(removeIcon);

               addedImagesContainer.appendChild(thumbnail);

           }

       };