# Product Personalization Tool

CHANGE: sections/main-product.liquid

| **add**  **{% render 'custom-mockup-creator' %}**  **before**  **{% render 'delivery-date-calculation-block-product-page' %}**  **… and that one before**  **{%- when 'rating' -%}** |
| --- |

CREATE snippets/custom-product-media-gallery.liquid

| **48**  **class="product\_\_column-sticky"**  **=>**  **class="product\_\_column-sticky custom-product\_\_column-sticky"** |
| --- |

in snippets/custom-product-media-gallery.liquid

| **65**  **<li**  **id="Slide-{{ section.id }}-{{ featured\_media.id }}"**  **...**  **=>**  **<div id="mockup-wrapper" class="parent-container">**  **<div class="canvas-wrapper">**  **<canvas id="canvas"></canvas>**  **</div>**  **</div>** |
| --- |

IN snippets/buy-buttons.liquid

before

{%- if gift\_card\_recipient\_feature\_active -%}

add

| **{% if template contains 'ppt-' %}**  **<label id="message" class="form\_\_label">Minimum recommended size: 1000 × 1000px</label>**  **<input type="file" name="properties[Uploaded Design]" accept="image/\*" id="upload-input" />**  **<input type="file" name="properties[Mockup Screenshot]" accept="image/\*" id="screenshot-upload" hidden>**  **{% endif %}** |
| --- |

IN sections/cart-notification-product.liquid

before

| **{%- if item.image -%}**  **<div class="cart-notification-product\_\_image global-media-settings">**  **<img**  **src="{{ item.image | image\_url: width: 140 }}"**  **alt="{{ item.image.alt | escape }}"**  **width="70"**  **height="{{ 70 | divided\_by: item.image.aspect\_ratio | ceil }}"**  **loading="lazy"**  **>**  **</div>**  **{%- endif -%}** |
| --- |

after

| **{% assign mockup\_image = item.properties['Mockup Screenshot'] %}**  **{%- if mockup\_image or item.image -%}**  **<div class="cart-notification-product\_\_image global-media-settings">**  **<img**  **src="{% if mockup\_image %}{{ mockup\_image }}{% else %}{{ item.image | image\_url: width: 140 }}{% endif %}"**  **alt="{% if mockup\_image %}Custom Mockup Screenshot{% else %}{{ item.image.alt | escape }}{% endif %}"**  **width="70"**  **height="{{ 70 | divided\_by: item.image.aspect\_ratio | ceil }}"**  **loading="lazy"**  **style="object-fit: contain;"**  **>**  **</div>**  **{%- endif -%}** |
| --- |

AND

before

| **{%- for property in item.properties -%}**  **...**  **{%- endfor -%}** |
| --- |

after

| **{% if item.properties['Mockup Screenshot'] == blank %}**  **{%- for property in item.properties -%}**  **...**  **{%- endfor -%}**  **{%- endif -%}** |
| --- |

IN sections/main-cart-items.liquid

before

| **<td class="cart-item\_\_media">**  **{% if item.image %}**  **{% comment %} Leave empty space due to a:empty CSS display: none rule {% endcomment %}**  **<a href="{{ item.url }}" class="cart-item\_\_link" aria-hidden="true" tabindex="-1"> </a>**  **<div class="cart-item\_\_image-container gradient global-media-settings">**  **<img**  **src="{{ item.image | image\_url: width: 300 }}"**  **class="cart-item\_\_image"**  **alt="{{ item.image.alt | escape }}"**  **loading="lazy"**  **width="150"**  **height="{{ 150 | divided\_by: item.image.aspect\_ratio | ceil }}"**  **>**  **</div>**  **{% endif %}**  **</td>** |
| --- |

after

| **<td class="cart-item\_\_media">**  **<div class="cart-item\_\_image-container gradient global-media-settings">**  **<a href="{{ item.url }}" class="cart-item\_\_link" aria-hidden="true" tabindex="-1"> </a>**  **{% if item.properties['Mockup Screenshot'] %}**  **<img**  **src="{{ item.properties['Mockup Screenshot'] }}"**  **class="cart-item\_\_image"**  **alt="Custom Mockup Screenshot"**  **loading="lazy"**  **width="150"**  **style="max-height: 150px; object-fit: contain;"**  **>**  **{% elsif item.image %}**  **<img**  **src="{{ item.image | image\_url: width: 300 }}"**  **class="cart-item\_\_image"**  **alt="{{ item.image.alt | escape }}"**  **loading="lazy"**  **width="150"**  **height="{{ 150 | divided\_by: item.image.aspect\_ratio | ceil }}"**  **>**  **{% endif %}**  **</div>**  **</td>** |
| --- |

AND

before

| **{%- for property in item.properties -%}**  **...**  **{%- endfor -%}** |
| --- |

after

| **{% if item.properties['Mockup Screenshot'] == blank %}**  **{%- for property in item.properties -%}**  **...**  **{%- endfor -%}**  **{% endif %}** |
| --- |

IN sections/main-product.liquid

before

| **{% render 'product-media-gallery', variant\_images: variant\_images %}** |
| --- |

after

| **{% if template contains 'ppt-' %}**  **{% render 'custom-product-media-gallery', variant\_images: variant\_images %}**  **{% else %}**  **{% render 'product-media-gallery', variant\_images: variant\_images %}**  **{% endif %}** |
| --- |

IN sections/personalization-settings.liquid

before

| **{% schema %}**  **{**  **"name": "Section name",**  **"settings": []**  **}**  **{% endschema %}** |
| --- |

after

| **<div**  **id="dimension-settings"**  **data-required-width="{{ section.settings.required\_width }}"**  **data-required-height="{{ section.settings.required\_height }}"**  **>**  **<p>**  **Minimum image size: {{ section.settings.required\_width }} × {{ section.settings.required\_height }} px**  **</p>**  **</div>**  **{% schema %}**  **{**  **"name": "Personalization Settings",**  **"settings": [**  **{**  **"type": "number",**  **"id": "required\_width",**  **"label": "Required image width (px)",**  **"default": 3000**  **},**  **{**  **"type": "number",**  **"id": "required\_height",**  **"label": "Required image height (px)",**  **"default": 4000**  **}**  **]**  **}**  **{% endschema %}**  **{% section 'personalization-settings' %}** |
| --- |

snippets/custom-mockup-creator.liquid

| **<script src="https://cdn.jsdelivr.net/npm/html2canvas@1.4.1/dist/html2canvas.min.js"></script>**  **<script src="https://cdnjs.cloudflare.com/ajax/libs/fabric.js/4.4.0/fabric.min.js"></script>**  **<script>**  **document.addEventListener("DOMContentLoaded", function () {**  **const mockup = document.getElementById("tshirt-base");**  **let productData = {{ product | json }};**  **const deleteIcon = new Image();**  **deleteIcon.src = 'data:image/svg+xml;utf8,' + encodeURIComponent(`**  **<svg width="60" height="60" viewBox="0 0 60 60" xmlns="http://www.w3.org/2000/svg">**  **<defs>**  **<mask id="mask-x">**  **<rect width="60" height="60" fill="white"/>**  **<line x1="15" y1="15" x2="45" y2="45" stroke="black" stroke-width="6"/>**  **<line x1="45" y1="15" x2="15" y2="45" stroke="black" stroke-width="6"/>**  **</mask>**  **</defs>**  **<circle cx="30" cy="30" r="30" fill="#b2ccff" mask="url(#mask-x)"/>**  **</svg>**  **`);**  **function initCanvas() {**  **const canvasElement = document.getElementById("canvas");**  **const canvas = new fabric.Canvas("canvas");**  **fabric.Object.prototype.setControlsVisibility({ mt: false, mb: false, ml: false, mr: false });**  **fabric.Object.prototype.cornerSize = 30;**  **fabric.Object.prototype.cornerStyle = 'circle';**  **canvas.setWidth(700);**  **canvas.setHeight(700);**  **canvasElement.style.width = '100%';**  **canvasElement.style.height = '100%';**  **fabric.Image.fromURL('{{ product.selected\_or\_first\_available\_variant.featured\_image | image\_url: width: 700 }}', function (img) {**  **img.scaleToWidth(700);**  **canvas.setBackgroundImage(img, canvas.renderAll.bind(canvas));**  **});**  **let uploadedImage;**  **document.getElementById('upload-input')?.addEventListener('change', function (e) {**  **const label = document.querySelector('label#message');**  **if (!label) return;**  **const file = e.target.files[0];**  **if (!file) {**  **if (uploadedImage) {**  **canvas.remove(uploadedImage);**  **canvas.renderAll();**  **uploadedImage = null;**  **}**  **label.innerHTML = `{{ 'translations.product\_page.ppt.min\_size\_label' | t }}: 1000 × 1000px`;**  **return;**  **}**  **const reader = new FileReader();**  **reader.onload = function (event) {**  **const img = new Image();**  **img.onload = function () {**  **const width = img.naturalWidth;**  **const height = img.naturalHeight;**  **const isValid = width >= 1000 && height >= 1000;**  **label.innerHTML = `{{ 'translations.product\_page.ppt.uploaded\_size\_label' | t }}: <span style="color:${isValid ? 'green' : 'red'}">${width} × ${height}px</span>`;**  **fabric.Image.fromURL(event.target.result, function (fabricImg) {**  **if (uploadedImage) canvas.remove(uploadedImage);**  **const maxWidth = canvas.getWidth() \* 0.5;**  **const maxHeight = canvas.getHeight() \* 0.5;**  **if (fabricImg.width > maxWidth || fabricImg.height > maxHeight) fabricImg.scaleToWidth(maxWidth);**  **fabricImg.set({**  **left: (canvas.getWidth() - fabricImg.getScaledWidth()) / 2,**  **top: (canvas.getHeight() - fabricImg.getScaledHeight()) / 2,**  **hasBorders: true**  **});**  **canvas.add(fabricImg);**  **canvas.renderAll();**  **uploadedImage = fabricImg;**  **});**  **};**  **img.src = event.target.result;**  **};**  **reader.readAsDataURL(file);**  **});**  **document.addEventListener('change', function () {**  **const selectedVariantId = document.querySelector('[name="id"]')?.value;**  **const selectedVariant = productData.variants.find(v => v.id == selectedVariantId);**  **if (selectedVariant?.featured\_image?.src) {**  **fabric.Image.fromURL(selectedVariant.featured\_image.src, function (img) {**  **img.scaleToWidth(700);**  **canvas.setBackgroundImage(img, canvas.renderAll.bind(canvas));**  **});**  **}**  **});**  **const isMobile = window.innerWidth <= 1200;**  **fabric.Object.prototype.controls.deleteControl = new fabric.Control({**  **x: 0,**  **y: 0.5,**  **offsetY: isMobile ? 40 : 22,**  **offsetX: 0,**  **cursorStyle: 'pointer',**  **cornerSize: isMobile ? 66 : 33,**  **mouseUpHandler: function (\_, transform) {**  **const target = transform.target;**  **const canvas = target.canvas;**  **canvas.remove(target);**  **canvas.requestRenderAll();**  **const label = document.querySelector('label#message');**  **if (label) label.innerHTML = `{{ 'translations.product\_page.ppt.min\_size\_label' | t }}: 1000 × 1000px`;**  **const uploadInput = document.getElementById('upload-input');**  **if (uploadInput) uploadInput.value = '';**  **},**  **render: function (ctx, left, top, \_, fabricObject) {**  **const size = this.cornerSize;**  **if (deleteIcon.complete) {**  **ctx.save();**  **ctx.beginPath();**  **ctx.arc(left, top, size / 2, 0, 2 \* Math.PI);**  **ctx.clip();**  **ctx.drawImage(deleteIcon, left - size / 2, top - size / 2, size, size);**  **ctx.restore();**  **} else {**  **deleteIcon.onload = function () {**  **fabricObject.canvas.requestRenderAll();**  **};**  **}**  **}**  **});**  **return canvas;**  **}**  **function rebindEvents(canvas) {**  **const fileInput = document.getElementById('screenshot-upload');**  **document.body.addEventListener('click', function (e) {**  **const button = e.target.closest('button[name="add"]');**  **if (!button) return;**  **const designInput = document.getElementById('upload-input');**  **if (!designInput.files.length) {**  **e.preventDefault();**  **alert('Please upload a design and wait for the mockup screenshot before adding to cart.');**  **return;**  **}**  **e.preventDefault();**  **const form = button.closest('form');**  **const mockup = document.getElementById('mockup-wrapper');**  **if (!mockup || !form || !fileInput) return;**  **canvas.getObjects().forEach(obj => { obj.hasControls = false });**  **canvas.renderAll();**  **html2canvas(mockup, { useCORS: true, scale: 2 }).then(canvasScreenshot => {**  **canvas.getObjects().forEach(obj => { obj.hasControls = true });**  **canvas.renderAll();**  **canvasScreenshot.toBlob(function (blob) {**  **const file = new File([blob], 'mockup-screenshot.png', { type: 'image/png' });**  **const dataTransfer = new DataTransfer();**  **dataTransfer.items.add(file);**  **fileInput.files = dataTransfer.files;**  **submitViaAjax(form);**  **}, 'image/png');**  **});**  **});**  **function submitViaAjax(form) {**  **const submitButton = form.querySelector('[name="add"]');**  **const spinner = submitButton.querySelector('.loading\_\_spinner');**  **const originalText = submitButton.querySelector('span');**  **submitButton.disabled = true;**  **submitButton.classList.add('loading');**  **spinner.classList.remove('hidden');**  **originalText.textContent = 'Adding...';**  **const formData = new FormData(form);**  **fetch('/cart/add.js', {**  **method: 'POST',**  **body: formData,**  **headers: { 'Accept': 'application/json' }**  **}).then(response => response.json()).then(data => {**  **fetch(`/cart?sections=cart-notification-product,cart-notification-button,cart-icon-bubble&key=${data.key}`).then(res => res.json()).then(sectionData => {**  **document.querySelector('cart-notification')?.renderContents({ key: data.key, sections: sectionData });**  **submitButton.disabled = false;**  **submitButton.classList.remove('loading');**  **spinner.classList.add('hidden');**  **originalText.textContent = 'Add to cart';**  **});**  **}).catch(() => {**  **submitButton.disabled = false;**  **submitButton.classList.remove('loading');**  **spinner.classList.add('hidden');**  **originalText.textContent = 'Add to cart';**  **});**  **}**  **}**  **const canvas = initCanvas();**  **rebindEvents(canvas);**  **});**  **</script>** |
| --- |

# 