

```
<!DOCTYPE html>
<html lang="vi">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Smart Imposition Pro - Tool Bình Trang Danh Thiếp In Nhanh</title>

  <!-- Tailwind CSS -->
  <script src="https://cdn.tailwindcss.com"></script>
  <!-- Lucide Icons -->
  <script src="https://unpkg.com/lucide@latest"></script>
  <!-- PDF-lib for high quality PDF manipulation -->
  <script src="https://unpkg.com/pdf-lib@1.17.1/dist/pdf-lib.min.js"></script>
  <!-- PDF.js for rendering PDF pages onto HTML canvas -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.16.105/pdf.min.js"></script>

  <style>
    body {
      font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial,
sans-serif;
      background-color: #0f172a;
      color: #f8fafc;
    }
    .custom-scrollbar::-webkit-scrollbar {
      width: 6px;
      height: 6px;
    }
    .custom-scrollbar::-webkit-scrollbar-track {
      background: #1e293b;
    }
    .custom-scrollbar::-webkit-scrollbar-thumb {
      background: #475569;
      border-radius: 3px;
    }
    .canvas-container {
      background-color: #1e293b;
      background-image:
        radial-gradient(#334155 1px, transparent 1px);
      background-size: 16px 16px;
    }
  </style>
</head>
<body class="h-screen overflow-hidden flex flex-col">
```

```

<!-- TOP HEADER -->
<header class="bg-slate-900 border-b border-slate-800 px-6 py-3 flex items-center
justify-between shadow-md">
  <div class="flex items-center gap-3">
    <div class="bg-emerald-500 p-2 rounded-lg text-slate-950 font-bold">
      <i data-lucide="printer" class="w-6 h-6"></i>
    </div>
    <div>
      <h1 class="text-lg font-bold text-white tracking-wide flex items-center gap-2">
        SMART IMPOSITION PRO
        <span class="text-xs bg-emerald-500/20 text-emerald-400 border border-emerald-500/30
px-2 py-0.5 rounded-full font-mono">v2.0 Pre-press</span>
      </h1>
      <p class="text-xs text-slate-400">Hệ thống bình trang danh thiếp tự động - Tối ưu cho
xưởng In Nhanh</p>
    </div>
  </div>

  <!-- Quick Stats Header -->
  <div class="flex items-center gap-6 text-sm">
    <div class="flex items-center gap-2 bg-slate-800/80 px-3 py-1.5 rounded-md border
border-slate-700">
      <span class="text-slate-400">Layout:</span>
      <span id="stat-nup" class="font-bold text-emerald-400 font-mono">0-up</span>
    </div>
    <div class="flex items-center gap-2 bg-slate-800/80 px-3 py-1.5 rounded-md border
border-slate-700">
      <span class="text-slate-400">Số tờ cần in:</span>
      <span id="stat-sheets" class="font-bold text-amber-400 font-mono">0 tờ</span>
    </div>
    <button onclick="exportImposedPDF()" id="btn-export" disabled class="bg-emerald-600
hover:bg-emerald-500 text-slate-950 font-bold px-4 py-2 rounded-lg transition-all flex
items-center gap-2 shadow-lg disabled:opacity-50 disabled:cursor-not-allowed">
      <i data-lucide="download" class="w-4 h-4"></i>
      <span>XUẤT FILE PDF IN</span>
    </button>
  </div>
</header>

  <!-- MAIN APP CONTAINER (3 COLUMNS) -->
  <div class="flex-1 flex overflow-hidden">

```

```

<!-- COLUMN 1: SIDEBAR CÀI ĐẶT BÀI BÌNH (LEFT) -->
<aside class="w-80 bg-slate-900 border-r border-slate-800 flex flex-col p-4 space-y-5
overflow-y-auto custom-scrollbar">

  <!-- 1. UPLOAD FILE PDF -->
  <div class="space-y-2">
    <label class="text-xs font-semibold uppercase tracking-wider text-slate-400 flex
items-center gap-1.5">
      <i data-lucide="file-up" class="w-4 h-4 text-emerald-400"></i>
      1. Upload File PDF Đầu Vào
    </label>

    <div id="drop-zone" onclick="document.getElementById('file-input').click()" class="border-2
border-dashed border-slate-700 hover:border-emerald-500 bg-slate-800/50 hover:bg-slate-800
rounded-xl p-4 text-center cursor-pointer transition-all group">
      <input type="file" id="file-input" accept=".pdf" class="hidden"
onchange="handleFileSelect(event)">
      <i data-lucide="upload-cloud" class="w-8 h-8 mx-auto text-slate-500
group-hover:text-emerald-400 transition-colors mb-2"></i>
      <p id="upload-label" class="text-xs text-slate-300 font-medium">Kéo thả file PDF danh
thiếp vào đây hoặc click chọn file</p>
      <p class="text-[10px] text-slate-500 mt-1">Chỉ chấp nhận file .PDF chuẩn
vector/image</p>
    </div>

    <!-- Pre-flight File Info Box -->
    <div id="file-info-box" class="hidden bg-slate-800/80 border border-slate-700 rounded-lg
p-3 space-y-2 text-xs">
      <div class="flex justify-between items-center border-b border-slate-700/60 pb-1.5">
        <span class="font-medium text-slate-300 truncate max-w-[170px]"
id="info-filename">file_name.pdf</span>
        <span id="info-status" class="px-2 py-0.5 rounded text-[10px] font-bold
bg-emerald-500/20 text-emerald-400 border border-emerald-500/30">HỢP LỆ</span>
      </div>
      <div class="grid grid-cols-2 gap-2 text-slate-400 text-[11px]">
        <div>Số trang: <span id="info-pagecount" class="text-white font-mono">2</span></div>
        <div>TrimBox: <span id="info-trimsize" class="text-white font-mono">90 x 54
mm</span></div>
        <div>Bleed (Tràn lề): <span id="info-bleed" class="text-emerald-400 font-mono">1.5
mm</span></div>
        <div>MediaBox: <span id="info-mediasize" class="text-white font-mono">93 x 57
mm</span></div>
      </div>

```

```

<!-- Define Bleed Button if missing -->
<div id="bleed-action-box" class="hidden pt-1 border-t border-slate-700/60">
  <p class="text-[10px] text-amber-400 mb-1"> ⚠ Cảnh báo: File chưa khai báo Trần Lề
(Bleed)!</p>
  <div class="flex gap-1.5">
    <button onclick="setDefinedBleed(1.5)" class="flex-1 bg-amber-500/20
hover:bg-amber-500/30 text-amber-300 border border-amber-500/40 py-1 rounded text-[10px]
font-medium">Thêm 1.5mm</button>
    <button onclick="setDefinedBleed(2.0)" class="flex-1 bg-amber-500/20
hover:bg-amber-500/30 text-amber-300 border border-amber-500/40 py-1 rounded text-[10px]
font-medium">Thêm 2.0mm</button>
  </div>
</div>
</div>
</div>
</div>

<hr class="border-slate-800">

<!-- 2. KHỔ GIẤY IN & LỀ XƯỞNG -->
<div class="space-y-3">
  <label class="text-xs font-semibold uppercase tracking-wider text-slate-400 flex
items-center gap-1.5">
    <i data-lucide="layers" class="w-4 h-4 text-emerald-400"></i>
    2. Cấu Hình Khổ Giấy In
  </label>

  <div>
    <span class="text-xs text-slate-300 mb-1 block">Khổ giấy (Rộng x Cao mm):</span>
    <select id="sheet-preset" onchange="handleSheetPresetChange()" class="w-full
bg-slate-800 border border-slate-700 rounded-lg px-3 py-2 text-xs text-white focus:outline-none
focus:border-emerald-500">
      <option value="325x370">325 x 370 mm (Khổ chuẩn in nhanh)</option>
      <option value="325x280">325 x 280 mm</option>
      <option value="325x325">325 x 325 mm</option>
      <option value="325x350">325 x 350 mm</option>
      <option value="325x430">325 x 430 mm</option>
      <option value="325x480">325 x 480 mm</option>
      <option value="330x480">330 x 480 mm (Khổ lớn A3+)</option>
      <option value="custom">-- Tùy chỉnh kích thước --</option>
    </select>
  </div>

  <!-- Custom Sheet Inputs -->

```

```

<div id="custom-sheet-inputs" class="hidden grid grid-cols-2 gap-2">
  <div>
    <span class="text-[10px] text-slate-400">Rộng (mm):</span>
    <input type="number" id="sheet-custom-w" value="325" onchange="updateImposition()"
class="w-full bg-slate-800 border border-slate-700 rounded px-2 py-1 text-xs text-white">
  </div>
  <div>
    <span class="text-[10px] text-slate-400">Cao (mm):</span>
    <input type="number" id="sheet-custom-h" value="370" onchange="updateImposition()"
class="w-full bg-slate-800 border border-slate-700 rounded px-2 py-1 text-xs text-white">
  </div>
</div>

```

```

<div>
  <span class="text-xs text-slate-300 mb-1 block">Lề khổ giấy / Lề nhíp (mm):</span>
  <input type="number" id="sheet-margin" value="5" step="0.5"
onchange="updateImposition()" class="w-full bg-slate-800 border border-slate-700 rounded-lg
px-3 py-1.5 text-xs text-white">
</div>
</div>

```

```

<hr class="border-slate-800">

```

```

<!-- 3. KIỂU BÌNH & BON CẮT -->

```

```

<div class="space-y-3">
  <label class="text-xs font-semibold uppercase tracking-wider text-slate-400 flex
items-center gap-1.5">
    <i data-lucide="scissors" class="w-4 h-4 text-emerald-400"></i>
    3. Đặt Bon Cắt & Khoảng Cách Con
  </label>

```

```

<div class="grid grid-cols-2 gap-2">
  <div>
    <span class="text-[11px] text-slate-300 block mb-1">Kiểu In:</span>
    <select id="print-mode" onchange="updateImposition()" class="w-full bg-slate-800
border border-slate-700 rounded px-2 py-1.5 text-xs text-white">
      <option value="2">2 Mặt (Duplex)</option>
      <option value="1">1 Mặt (Single)</option>
    </select>
  </div>
  <div>
    <span class="text-[11px] text-slate-300 block mb-1">Trở mặt:</span>
    <select id="duplex-type" onchange="updateImposition()" class="w-full bg-slate-800

```

```

border border-slate-700 rounded px-2 py-1.5 text-xs text-white">
    <option value="horizontal">Trở ngang (Tumbling)</option>
    <option value="vertical">Trở đầu (Perfecting)</option>
</select>
</div>
</div>

<div class="grid grid-cols-2 gap-2">
    <div>
        <span class="text-[11px] text-slate-300 block mb-1">Khoảng cách 2 con (mm):</span>
        <input type="number" id="gutter-size" value="3" step="0.5" min="0"
onchange="updateImposition()" class="w-full bg-slate-800 border border-slate-700 rounded
px-2 py-1.5 text-xs text-white">
    </div>
    <div>
        <span class="text-[11px] text-slate-300 block mb-1">Độ dài Bon cắt (mm):</span>
        <input type="number" id="cropmark-len" value="3" step="0.5" min="1"
onchange="updateImposition()" class="w-full bg-slate-800 border border-slate-700 rounded
px-2 py-1.5 text-xs text-white">
    </div>
</div>
</div>
</div>

</aside>

<!-- COLUMN 2: LIVE CANVAS PREVIEW WORKSPACE (MIDDLE) -->
<main class="flex-1 bg-slate-950 flex flex-col overflow-hidden relative">

    <!-- Workspace Toolbar -->
    <div class="bg-slate-900/90 border-b border-slate-800 px-4 py-2 flex items-center
justify-between z-10 backdrop-blur-sm">
        <div class="flex items-center gap-3">
            <span class="text-xs text-slate-400 font-medium">Chế độ xem:</span>
            <div class="bg-slate-800 p-0.5 rounded-lg flex gap-1 border border-slate-700">
                <button onclick="setPreviewView('dual')" id="view-dual-btn" class="px-3 py-1
rounded-md text-xs font-medium bg-emerald-500 text-slate-950 shadow">Xem 2 Mặt Đồng
Thời</button>
                <button onclick="setPreviewView('front')" id="view-front-btn" class="px-3 py-1
rounded-md text-xs font-medium text-slate-400 hover:text-white">Mặt Trước</button>
                <button onclick="setPreviewView('back')" id="view-back-btn" class="px-3 py-1
rounded-md text-xs font-medium text-slate-400 hover:text-white">Mặt Sau</button>
            </div>
        </div>
    </div>

```

```

<!-- Zoom controls -->
<div class="flex items-center gap-2">
  <button onclick="adjustZoom(-0.1)" class="bg-slate-800 hover:bg-slate-700 p-1.5 rounded
border border-slate-700 text-slate-300">
    <i data-lucide="zoom-out" class="w-4 h-4"></i>
  </button>
  <span id="zoom-level" class="text-xs font-mono text-slate-300 w-12
text-center">100%</span>
  <button onclick="adjustZoom(0.1)" class="bg-slate-800 hover:bg-slate-700 p-1.5 rounded
border border-slate-700 text-slate-300">
    <i data-lucide="zoom-in" class="w-4 h-4"></i>
  </button>
  <button onclick="resetZoom()" class="bg-slate-800 hover:bg-slate-700 px-2 py-1 rounded
border border-slate-700 text-xs text-slate-300">Fit</button>
</div>
</div>

```

```

<!-- Canvas Scroll Area -->
<div class="flex-1 overflow-auto p-8 flex items-center justify-center canvas-container
custom-scrollbar" id="canvas-wrapper">
  <div id="canvas-holder" class="flex gap-8 items-center justify-center transition-transform
duration-150 origin-center">

```

```

<!-- Front Page Preview Card -->
<div id="front-card" class="bg-slate-900 border border-slate-800 rounded-xl p-3
shadow-2xl flex flex-col items-center">
  <div class="flex items-center justify-between w-full mb-2">
    <span class="text-xs font-bold text-emerald-400 flex items-center gap-1">
      <i data-lucide="eye" class="w-3.5 h-3.5"></i> MẶT TRƯỚC (FRONT SHEET)
    </span>
    <span class="text-[10px] text-slate-500 font-mono" id="front-sheet-dim">325 x 370
mm</span>
  </div>
  <div class="relative bg-white shadow-md overflow-hidden rounded-sm"
id="canvas-front-box">
    <canvas id="canvas-front" class="block"></canvas>
  </div>
</div>

```

```

<!-- Back Page Preview Card -->
<div id="back-card" class="bg-slate-900 border border-slate-800 rounded-xl p-3
shadow-2xl flex flex-col items-center">

```

```

<div class="flex items-center justify-between w-full mb-2">
  <span class="text-xs font-bold text-sky-400 flex items-center gap-1">
    <i data-lucide="eye" class="w-3.5 h-3.5"></i> MẶT SAU (BACK SHEET)
  </span>
  <span class="text-[10px] text-slate-500 font-mono" id="back-sheet-dim">325 x 370
mm</span>
</div>
<div class="relative bg-white shadow-md overflow-hidden rounded-sm"
id="canvas-back-box">
  <canvas id="canvas-back" class="block"></canvas>
</div>

</div>
</div>

<!-- Empty State Overlay -->
<div id="empty-state" class="absolute inset-0 bg-slate-950/90 flex flex-col items-center
justify-center z-20">
  <div class="bg-slate-900 border border-slate-800 p-8 rounded-2xl max-w-md text-center
shadow-2xl space-y-4">
    <div class="w-16 h-16 bg-emerald-500/10 text-emerald-400 rounded-full flex items-center
justify-center mx-auto border border-emerald-500/20">
      <i data-lucide="file-spreadsheet" class="w-8 h-8"></i>
    </div>
    <h3 class="text-lg font-bold text-white">Chưa Có File PDF Bài Bình</h3>
    <p class="text-xs text-slate-400 leading-relaxed">Vui lòng chọn hoặc kéo thả file danh
thiếp PDF vào bảng điều khiển bên trái để phần mềm tự động phân tích và bình trang live.</p>
    <button onclick="document.getElementById('file-input').click()" class="bg-emerald-500
hover:bg-emerald-400 text-slate-950 font-bold px-5 py-2.5 rounded-xl text-xs transition-all
shadow-lg inline-flex items-center gap-2">
      <i data-lucide="plus-circle" class="w-4 h-4"></i>
      Tải File PDF Danh Thiếp
    </button>
  </div>
</div>
</div>

</main>

<!-- COLUMN 3: THÔNG TIN BÀI IN & XUẤT FILE (RIGHT) -->
<aside class="w-80 bg-slate-900 border-l border-slate-800 flex flex-col p-4 space-y-5
overflow-y-auto custom-scrollbar">

```

```
<label class="text-xs font-semibold uppercase tracking-wider text-slate-400 flex items-center gap-1.5">
```

```
<i data-lucide="clipboard-list" class="w-4 h-4 text-emerald-400"></i>
```

```
4. Thông Tin Đơn Hàng In
```

```
</label>
```

```
<div class="space-y-3">
```

```
<div>
```

```
<span class="text-xs text-slate-300 mb-1 block">Tên Khách Hàng / Đơn Hàng:</span>
```

```
<input type="text" id="cust-name" value="Khách Hàng A" onchange="updateJobInfo()" class="w-full bg-slate-800 border border-slate-700 rounded-lg px-3 py-1.5 text-xs text-white focus:outline-none focus:border-emerald-500">
```

```
</div>
```

```
<div>
```

```
<span class="text-xs text-slate-300 mb-1 block">Loại Giấy In:</span>
```

```
<select id="paper-type" onchange="updateJobInfo()" class="w-full bg-slate-800 border border-slate-700 rounded-lg px-3 py-1.5 text-xs text-white">
```

```
<option value="Couche 300gsm">Couche 300 gsm</option>
```

```
<option value="Couche 250gsm">Couche 250 gsm</option>
```

```
<option value="Fort 250gsm">Fort 250 gsm (Giấy Ảnh)</option>
```

```
<option value="Kraft 300gsm">Kraft 300 gsm (Giấy Nâu)</option>
```

```
<option value="Giấy Mỹ Thuật">Giấy Mỹ Thuật Cao Cấp</option>
```

```
</select>
```

```
</div>
```

```
<div>
```

```
<span class="text-xs text-slate-300 mb-1 block">Cán Màng (Finishing):</span>
```

```
<select id="lamination" onchange="updateJobInfo()" class="w-full bg-slate-800 border border-slate-700 rounded-lg px-3 py-1.5 text-xs text-white">
```

```
<option value="Màng Mờ 2 Mặt">Cán Màng Mờ 2 Mặt</option>
```

```
<option value="Màng Bóng 2 Mặt">Cán Màng Bóng 2 Mặt</option>
```

```
<option value="Không Cán Màng">Không Cán Màng</option>
```

```
<option value="Khác">Gia công đặc biệt / Ép kim</option>
```

```
</select>
```

```
</div>
```

```
<div>
```

```
<span class="text-xs text-slate-300 mb-1 block">Số Lượng Thành Phẩm Cần (Con/Cái):</span>
```

```
<input type="number" id="order-qty" value="500" step="100" min="100" onchange="updateImposition()" class="w-full bg-slate-800 border border-slate-700 rounded-lg px-3 py-2 text-sm font-bold text-amber-400 focus:outline-none focus:border-emerald-500">
```

</div>

</div>

<hr class="border-slate-800">

<!-- Imposition Summary Card -->

<div class="bg-slate-800/80 border border-slate-700 rounded-xl p-4 space-y-3">

<h4 class="text-xs font-bold text-emerald-400 uppercase tracking-wider border-b border-slate-700 pb-2">Tóm Tắt Thông Số Bài Bình</h4>

<div class="space-y-2 text-xs">

<div class="flex justify-between">

Số con / tờ (N-up):

18 con / tờ

</div>

<div class="flex justify-between">

Số tờ in máy cần:

28
tờ

</div>

<div class="flex justify-between">

Bố cục Lưới (Cols x Rows):

3 cột x 6 hàng

</div>

<div class="flex justify-between">

Xoay bài 90°:

Không

</div>

</div>

</div>

<!-- Action Button -->

<button onclick="exportImposedPDF()" id="btn-export-side" disabled class="w-full
bg-emerald-500 hover:bg-emerald-400 text-slate-950 font-bold py-3 rounded-xl transition-all flex
items-center justify-center gap-2 shadow-xl disabled:opacity-50 disabled:cursor-not-allowed">

<i data-lucide="printer" class="w-5 h-5"></i>

XUẤT PDF BÀI BÌNH IN

</button>

</aside>

</div>

```
<!-- SCRIPT & CORE ENGINE LOGIC -->
<script>
  // Set PDF.js worker URL
  pdfjsLib.GlobalWorkerOptions.workerSrc =
'https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.16.105/pdf.worker.min.js';

  // GLOBAL STATE
  const state = {
    pdfFile: null,
    pdfBytes: null,
    pdfDocLib: null,
    pdfDocJs: null,
    pageCount: 0,

    // Card dimensions in mm
    cardTrimW: 90,
    cardTrimH: 54,
    bleed: 1.5,
    hasExplicitBleed: false,

    // Sheet config
    sheetW: 325,
    sheetH: 370,
    sheetMargin: 5,

    // Layout & Imposition
    printMode: 2, // 1 side or 2 side
    duplexType: 'horizontal',
    gutter: 3,
    cropMarkLen: 3,
    cropMarkOffset: 1.5,

    // Job metadata
    customerName: 'Khách Hàng A',
    paperType: 'Couche 300gsm',
    lamination: 'Màng Mờ 2 Mặt',
    orderQty: 500,

    // Calculated values
    nUp: 18,
    cols: 3,
    rows: 6,
    isRotated: false,
```

```

requiredSheets: 28,

// View settings
previewView: 'dual', // 'dual', 'front', 'back'
zoomScale: 1.0,

// Rendered Image Data URLs for fast canvas preview
frontPageImgData: null,
backPageImgData: null
};

// Initialize Lucide Icons
lucide.createIcons();

// 1. FILE SELECTION & PREFLIGHT PARSER
async function handleFileSelect(event) {
  const file = event.target.files[0];
  if (!file) return;
  if (file.type !== 'application/pdf' && !file.name.endsWith('.pdf')) {
    alert("Vui lòng chọn file đúng định dạng .PDF!");
    return;
  }
}

state.pdfFile = file;
document.getElementById('upload-label').innerText = file.name;

const arrayBuffer = await file.arrayBuffer();
state.pdfBytes = arrayBuffer;

try {
  // Load with PDF.js for preview rendering
  state.pdfDocJs = await pdfjsLib.getDocument({ data: arrayBuffer.slice(0) }).promise;
  state.pageCount = state.pdfDocJs.numPages;

  // Load with PDF-lib for manipulation
  state.pdfDocLib = await PDFLib.PDFDocument.load(arrayBuffer);

  // Analyze First Page Box Dimensions
  const page1 = state.pdfDocJs.getPage(1);
  const pageObj = await page1;
  const viewport = pageObj.getViewport({ scale: 1.0 });

  // Convert points to mm (1 pt = 25.4 / 72 mm = 0.352777778 mm)

```

```

const ptToMm = 25.4 / 72;
const pageW = viewport.width * ptToMm;
const pageH = viewport.height * ptToMm;

// Check if page fits standard business card proportions
if (pageW < 60 || pageH < 35 || pageW > 160 || pageH > 110) {
  alert('Cảnh báo: Kích thước PDF (' + pageW.toFixed(1) + 'x' + pageH.toFixed(1) + 'mm)
ngoài phạm vi danh thiếp tiêu chuẩn!');
}

// Determine Bleed & TrimBox
// Standard namecard trim is usually around 90x54, 90x55, 89x54, 85x50
if (Math.abs(pageW - 93) < 3 && Math.abs(pageH - 57) < 3) {
  state.cardTrimW = 90;
  state.cardTrimH = 54;
  state.bleed = 1.5;
  state.hasExplicitBleed = true;
} else if (Math.abs(pageW - 96) < 3 && Math.abs(pageH - 60) < 3) {
  state.cardTrimW = 90;
  state.cardTrimH = 54;
  state.bleed = 3.0;
  state.hasExplicitBleed = true;
} else {
  // Default assumes file has no bleed or user defines it
  state.cardTrimW = Math.round(pageW);
  state.cardTrimH = Math.round(pageH);
  state.bleed = 0;
  state.hasExplicitBleed = false;
}

// Render source page thumbnails for fast grid draw
state.frontPageImgData = await renderPdfPageToDataURL(state.pdfDocJs, 1);
if (state.pageCount > 1) {
  state.backPageImgData = await renderPdfPageToDataURL(state.pdfDocJs, 2);
} else {
  state.backPageImgData = state.frontPageImgData;
}

// Update UI Info Box
document.getElementById('file-info-box').classList.remove('hidden');
document.getElementById('empty-state').classList.add('hidden');
document.getElementById('info-filename').innerText = file.name;
document.getElementById('info-pagecount').innerText = state.pageCount;

```

```

    document.getElementById('info-trimsize').innerText = `${state.cardTrimW} x
    ${state.cardTrimH} mm`;
    document.getElementById('info-mediasize').innerText = `${pageW.toFixed(1)} x
    ${pageH.toFixed(1)} mm`;
    document.getElementById('info-bleed').innerText = `${state.bleed} mm`;

    if (!state.hasExplicitBleed || state.bleed === 0) {
        document.getElementById('bleed-action-box').classList.remove('hidden');
    } else {
        document.getElementById('bleed-action-box').classList.add('hidden');
    }

    // Enable Export Buttons
    document.getElementById('btn-export').disabled = false;
    document.getElementById('btn-export-side').disabled = false;

    // Run Imposition calculation & Draw Canvas
    updateImposition();

} catch (err) {
    console.error(err);
    alert('Lỗi đọc file PDF: ' + err.message);
}
}

// Define Bleed Button Action
function setDefinedBleed(bleedVal) {
    state.bleed = bleedVal;
    state.hasExplicitBleed = true;
    document.getElementById('info-bleed').innerText = `${state.bleed} mm`;
    document.getElementById('bleed-action-box').classList.add('hidden');
    updateImposition();
}

// Helper: Render PDF Page to Data URL Image
async function renderPdfPageToDataUrl(pdfDoc, pageNum) {
    const page = await pdfDoc.getPage(pageNum);
    const viewport = page.getViewport({ scale: 2.0 }); // Higher scale for crisp preview
    const canvas = document.createElement('canvas');
    const ctx = canvas.getContext('2d');
    canvas.width = viewport.width;
    canvas.height = viewport.height;

```

```

    await page.render({ canvasContext: ctx, viewport: viewport }).promise;
    return canvas.toDataURL('image/png');
}

```

// 2. SHEET PRESET CHANGE HANDLER

```

function handleSheetPresetChange() {
    const val = document.getElementById('sheet-preset').value;
    const customInputs = document.getElementById('custom-sheet-inputs');

    if (val === 'custom') {
        customInputs.classList.remove('hidden');
    } else {
        customInputs.classList.add('hidden');
        const [w, h] = val.split('x').map(Number);
        state.sheetW = w;
        state.sheetH = h;
    }
    updateImposition();
}

```

// 3. CORE IMPOSITION ALGORITHM & GRID CALCULATION

```

function updateImposition() {
    // Read UI inputs
    if (document.getElementById('sheet-preset').value === 'custom') {
        state.sheetW = Number(document.getElementById('sheet-custom-w').value) || 325;
        state.sheetH = Number(document.getElementById('sheet-custom-h').value) || 370;
    }
    state.sheetMargin = Number(document.getElementById('sheet-margin').value) || 5;
    state.printMode = Number(document.getElementById('print-mode').value) || 2;
    state.duplexType = document.getElementById('duplex-type').value;
    state.gutter = Number(document.getElementById('gutter-size').value) || 0;
    state.cropMarkLen = Number(document.getElementById('cropmark-len').value) || 3;
    state.orderQty = Number(document.getElementById('order-qty').value) || 500;

    const cardW = state.cardTrimW + 2 * state.bleed;
    const cardH = state.cardTrimH + 2 * state.bleed;

    const usableW = state.sheetW - 2 * state.sheetMargin;
    const usableH = state.sheetH - 2 * state.sheetMargin;

    // Orientation 1: Portrait / Standard
    const cols1 = Math.floor((usableW + state.gutter) / (cardW + state.gutter));
    const rows1 = Math.floor((usableH + state.gutter) / (cardH + state.gutter));
}

```

```

const nUp1 = Math.max(0, cols1 * rows1);

// Orientation 2: Landscape / Rotated 90 deg
const cols2 = Math.floor((usableW + state.gutter) / (cardH + state.gutter));
const rows2 = Math.floor((usableH + state.gutter) / (cardW + state.gutter));
const nUp2 = Math.max(0, cols2 * rows2);

if (nUp2 > nUp1) {
    state.isRotated = true;
    state.cols = cols2;
    state.rows = rows2;
    state.nUp = nUp2;
} else {
    state.isRotated = false;
    state.cols = cols1;
    state.rows = rows1;
    state.nUp = nUp1;
}

// Calculate required printed sheets
state.requiredSheets = state.nUp > 0 ? Math.ceil(state.orderQty / state.nUp) : 0;

// Update UI Stats & Summary
document.getElementById('stat-nup').innerText = `${state.nUp}-up`;
document.getElementById('stat-sheets').innerText = `${state.requiredSheets} tờ`;
document.getElementById('sum-nup').innerText = `${state.nUp} con / tờ`;
document.getElementById('sum-sheets').innerText = `${state.requiredSheets} tờ`;
document.getElementById('sum-grid').innerText = `${state.cols} cột x ${state.rows} hàng`;
document.getElementById('sum-rotated').innerText = state.isRotated ? 'Có (90°)' : 'Không';

document.getElementById('front-sheet-dim').innerText = `${state.sheetW} x ${state.sheetH} mm`;
document.getElementById('back-sheet-dim').innerText = `${state.sheetW} x ${state.sheetH} mm`;

// Redraw Live Preview Canvas
drawPreviewCanvas();
}

function updateJobInfo() {
    state.customerName = document.getElementById('cust-name').value || 'Khách Hàng A';
    state.paperType = document.getElementById('paper-type').value;
    state.lamination = document.getElementById('lamination').value;

```

```
}
```

// 4. PREVIEW CANVAS RENDERING ENGINE

```
function drawPreviewCanvas() {  
  if (!state.pdfFile) return;  
  
  const scale = 2.0; // Render resolution scale  
  const mmToPx = 3.7795275591 * scale; // 1mm in pixels at scale  
  
  const canvasFront = document.getElementById('canvas-front');  
  const canvasBack = document.getElementById('canvas-back');  
  
  const sheetPxW = state.sheetW * mmToPx;  
  const sheetPxH = state.sheetH * mmToPx;  
  
  canvasFront.width = sheetPxW;  
  canvasFront.height = sheetPxH;  
  canvasBack.width = sheetPxW;  
  canvasBack.height = sheetPxH;  
  
  // Set CSS display size  
  const cssWidth = state.sheetW * 1.8; // mm to display px approx  
  const cssHeight = state.sheetH * 1.8;  
  
  canvasFront.style.width = cssWidth + 'px';  
  canvasFront.style.height = cssHeight + 'px';  
  canvasBack.style.width = cssWidth + 'px';  
  canvasBack.style.height = cssHeight + 'px';  
  
  const ctxF = canvasFront.getContext('2d');  
  const ctxB = canvasBack.getContext('2d');  
  
  // Fill Sheet Background White  
  ctxF.fillStyle = '#ffffff';  
  ctxF.fillRect(0, 0, sheetPxW, sheetPxH);  
  ctxB.fillStyle = '#ffffff';  
  ctxB.fillRect(0, 0, sheetPxW, sheetPxH);  
  
  // Draw Sheet Margins (Dashed Red Lines)  
  drawSheetMargin(ctxF, sheetPxW, sheetPxH, state.sheetMargin * mmToPx);  
  drawSheetMargin(ctxB, sheetPxW, sheetPxH, state.sheetMargin * mmToPx);  
  
  if (state.nUp <= 0) return;
```

```

// Calculate Grid Start Position (Centered on Sheet)
const finalCardBleedW = (state.isRotated ? state.cardTrimH : state.cardTrimW) + 2 *
state.bleed;
const finalCardBleedH = (state.isRotated ? state.cardTrimW : state.cardTrimH) + 2 *
state.bleed;

const gridWidth = state.cols * finalCardBleedW + (state.cols - 1) * state.gutter;
const gridHeight = state.rows * finalCardBleedH + (state.rows - 1) * state.gutter;

const startX = (state.sheetW - gridWidth) / 2;
const startY = (state.sheetH - gridHeight) / 2;

// Preload images for fast drawing
const imgFront = new Image();
const imgBack = new Image();
imgFront.src = state.frontPageImgData;
imgBack.src = state.backPageImgData || state.frontPageImgData;

imgFront.onload = () => {
  // Draw Cards Grid on Front & Back Sheets
  for (let r = 0; r < state.rows; r++) {
    for (let c = 0; c < state.cols; c++) {

      // Front Position
      const posX = startX + c * (finalCardBleedW + state.gutter);
      const posY = startY + r * (finalCardBleedH + state.gutter);

      // Back Position (Handle Tumbling / Perfecting Duplex Offset)
      let backCol = c;
      let backRow = r;
      if (state.duplexType === 'horizontal') {
        backCol = state.cols - 1 - c; // Flip horizontally
      } else {
        backRow = state.rows - 1 - r; // Flip vertically
      }
      const backPosX = startX + backCol * (finalCardBleedW + state.gutter);
      const backPosY = startY + backRow * (finalCardBleedH + state.gutter);

      // Draw Card Front
      drawSingleCard(ctxF, imgFront, posX * mmToPx, posY * mmToPx, finalCardBleedW *
mmToPx, finalCardBleedH * mmToPx, state.bleed * mmToPx, state.isRotated);
    }
  }
}

```

```

        // Draw Card Back
        drawSingleCard(ctxB, imgBack, backPosX * mmToPx, backPosY * mmToPx,
finalCardBleedW * mmToPx, finalCardBleedH * mmToPx, state.bleed * mmToPx,
state.isRotated);
    }
}

// Draw Professional Pre-press Crop Marks Vector Grid
drawCropMarksGrid(ctxF, startX, startY, state.cols, state.rows, finalCardBleedW,
finalCardBleedH, state.bleed, state.gutter, state.cropMarkLen, mmToPx);
drawCropMarksGrid(ctxB, startX, startY, state.cols, state.rows, finalCardBleedW,
finalCardBleedH, state.bleed, state.gutter, state.cropMarkLen, mmToPx);

// Draw Slug Information Text on Sheet Margin
drawSlugInfo(ctxF, sheetPxW, sheetPxH, mmToPx, 'MẶT TRƯỚC (FRONT)');
drawSlugInfo(ctxB, sheetPxW, sheetPxH, mmToPx, 'MẶT SAU (BACK)');
};
}

function drawSheetMargin(ctx, sheetW, sheetH, marginPx) {
    ctx.save();
    ctx.strokeStyle = 'rgba(239, 68, 68, 0.4)';
    ctx.setLineDash([6, 6]);
    ctx.lineWidth = 1;
    ctx.strokeRect(marginPx, marginPx, sheetW - 2 * marginPx, sheetH - 2 * marginPx);
    ctx.restore();
}

function drawSingleCard(ctx, img, x, y, w, h, bleedPx, isRotated) {
    ctx.save();
    ctx.translate(x + w / 2, y + h / 2);
    if (isRotated) {
        ctx.rotate(90 * Math.PI / 180);
    }
    ctx.drawImage(img, -w / 2, -h / 2, w, h);
    ctx.restore();

    // Draw Trim Box Border (Solid Cyan Line)
    ctx.save();
    ctx.strokeStyle = '#06b6d4';
    ctx.lineWidth = 1;
    ctx.strokeRect(x + bleedPx, y + bleedPx, w - 2 * bleedPx, h - 2 * bleedPx);
    ctx.restore();
}

```

```
}
```

```
// PRE-PRESS CROP MARKS GRID ALGORITHM
```

```
function drawCropMarksGrid(ctx, startX, startY, cols, rows, cardBleedW, cardBleedH, bleed, gutter, markLen, mmToPx) {
```

```
    ctx.save();  
    ctx.strokeStyle = '#000000';  
    ctx.lineWidth = 1.2;
```

```
    const cardTrimW = cardBleedW - 2 * bleed;  
    const cardTrimH = cardBleedH - 2 * bleed;
```

```
    for (let r = 0; r < rows; r++) {  
        for (let c = 0; c < cols; c++) {  
            const cardX = startX + c * (cardBleedW + gutter) + bleed;  
            const cardY = startY + r * (cardBleedH + gutter) + bleed;
```

```
            const left = cardX * mmToPx;  
            const right = (cardX + cardTrimW) * mmToPx;  
            const top = cardY * mmToPx;  
            const bottom = (cardY + cardTrimH) * mmToPx;  
            const len = markLen * mmToPx;  
            const gap = (bleed > 0 ? bleed : 1) * mmToPx;
```

```
            // Corner / Column / Row Crop Marks Logic
```

```
            // Left Horizontal Marks
```

```
            if (c === 0) {  
                drawLine(ctx, left - gap - len, top, left - gap, top);  
                drawLine(ctx, left - gap - len, bottom, left - gap, bottom);  
            }
```

```
            // Right Horizontal Marks
```

```
            if (c === cols - 1) {  
                drawLine(ctx, right + gap, top, right + gap + len, top);  
                drawLine(ctx, right + gap, bottom, right + gap + len, bottom);  
            }
```

```
            // Top Vertical Marks
```

```
            if (r === 0) {  
                drawLine(ctx, left, top - gap - len, left, top - gap);  
                drawLine(ctx, right, top - gap - len, right, top - gap);  
            }
```

```
            // Bottom Vertical Marks
```

```
            if (r === rows - 1) {  
                drawLine(ctx, left, bottom + gap, left, bottom + gap + len);
```

```

        drawLine(ctx, right, bottom + gap, right, bottom + gap + len);
    }

    // Inner Gutter Marks (Single straight lines between cards)
    if (c < cols - 1 && gutter > 0) {
        const nextLeft = (cardX + cardTrimW + gutter) * mmToPx;
        drawLine(ctx, right + gap, top, nextLeft - gap, top);
        drawLine(ctx, right + gap, bottom, nextLeft - gap, bottom);
    }
    if (r < rows - 1 && gutter > 0) {
        const nextTop = (cardY + cardTrimH + gutter) * mmToPx;
        drawLine(ctx, left, bottom + gap, left, nextTop - gap);
        drawLine(ctx, right, bottom + gap, right, nextTop - gap);
    }
}
}
}
ctx.restore();
}

```

```

function drawLine(ctx, x1, y1, x2, y2) {
    ctx.beginPath();
    ctx.moveTo(x1, y1);
    ctx.lineTo(x2, y2);
    ctx.stroke();
}

```

// DRAW SLUG INFORMATION TEXT ON SHEET MARGIN

```

function drawSlugInfo(ctx, sheetPxW, sheetPxH, mmToPx, sideLabel) {
    ctx.save();
    ctx.fillStyle = '#334155';
    ctx.font = 'bold 11px sans-serif';
    const marginPx = state.sheetMargin * mmToPx;

    const infoText = `SMART IMPOSITION PRO | ${state.customerName} | ${state.paperType} |
    ${state.lamination} | SL: ${state.orderQty} con -> In ${state.requiredSheets} Tò | ${sideLabel}`;
    ctx.fillText(infoText, marginPx, marginPx - 6);
    ctx.restore();
}

```

// 5. VIEW TOGGLE & ZOOM CONTROLS

```

function setPreviewView(mode) {
    state.previewView = mode;
    const frontCard = document.getElementById('front-card');
}

```

```

const backCard = document.getElementById('back-card');

document.getElementById('view-dual-btn').className = mode === 'dual' ? 'px-3 py-1
rounded-md text-xs font-medium bg-emerald-500 text-slate-950 shadow' : 'px-3 py-1
rounded-md text-xs font-medium text-slate-400 hover:text-white';
document.getElementById('view-front-btn').className = mode === 'front' ? 'px-3 py-1
rounded-md text-xs font-medium bg-emerald-500 text-slate-950 shadow' : 'px-3 py-1
rounded-md text-xs font-medium text-slate-400 hover:text-white';
document.getElementById('view-back-btn').className = mode === 'back' ? 'px-3 py-1
rounded-md text-xs font-medium bg-emerald-500 text-slate-950 shadow' : 'px-3 py-1
rounded-md text-xs font-medium text-slate-400 hover:text-white';

if (mode === 'dual') {
  frontCard.classList.remove('hidden');
  backCard.classList.remove('hidden');
} else if (mode === 'front') {
  frontCard.classList.remove('hidden');
  backCard.classList.add('hidden');
} else if (mode === 'back') {
  frontCard.classList.add('hidden');
  backCard.classList.remove('hidden');
}
}

function adjustZoom(delta) {
  state.zoomScale = Math.min(3.0, Math.max(0.4, state.zoomScale + delta));
  document.getElementById('canvas-holder').style.transform = `scale(${state.zoomScale})`;
  document.getElementById('zoom-level').innerText = `${Math.round(state.zoomScale *
100)}%`;
}

function resetZoom() {
  state.zoomScale = 1.0;
  document.getElementById('canvas-holder').style.transform = 'scale(1.0)';
  document.getElementById('zoom-level').innerText = '100%';
}

// 6. EXPORT HIGH-QUALITY VECTOR PDF FOR PRINTING
async function exportImposedPDF() {
  if (!state.pdfDocLib || !state.pdfBytes) return;

  alert('Đang tiến hành tạo file PDF bài bình in chất lượng cao vector...');

```

```

try {
  const mmToPt = 72 / 25.4; // 1mm = 2.83464567 pt
  const sheetPtW = state.sheetW * mmToPt;
  const sheetPtH = state.sheetH * mmToPt;

  const outPdf = await PDFLib.PDFDocument.create();

  // Embed Source Pages from uploaded PDF
  const [sourceFrontPage] = await outPdf.copyPages(state.pdfDocLib, [0]);
  let sourceBackPage = sourceFrontPage;
  if (state.pageCount > 1) {
    [sourceBackPage] = await outPdf.copyPages(state.pdfDocLib, [1]);
  }

  // Embed as XObjects
  const embeddedFront = await outPdf.embedPage(sourceFrontPage);
  const embeddedBack = await outPdf.embedPage(sourceBackPage);

  // Page 1: Front Sheet
  const frontPage = outPdf.addPage([sheetPtW, sheetPtH]);

  // Page 2: Back Sheet (if 2-sided)
  let backPage = null;
  if (state.printMode === 2) {
    backPage = outPdf.addPage([sheetPtW, sheetPtH]);
  }

  const cardBleedW = (state.isRotated ? state.cardTrimH : state.cardTrimW) + 2 *
state.bleed;
  const cardBleedH = (state.isRotated ? state.cardTrimW : state.cardTrimH) + 2 *
state.bleed;

  const gridW = state.cols * cardBleedW + (state.cols - 1) * state.gutter;
  const gridH = state.rows * cardBleedH + (state.rows - 1) * state.gutter;

  const startX = (state.sheetW - gridW) / 2;
  const startY = (state.sheetH - gridH) / 2;

  // Draw Imposed Grid Pages
  for (let r = 0; r < state.rows; r++) {
    for (let c = 0; c < state.cols; c++) {

      const posX = (startX + c * (cardBleedW + state.gutter)) * mmToPt;

```

```
// PDF-lib coordinate system Y starts from Bottom-Left!
const posY = (state.sheetH - (startY + r * (cardBleedH + state.gutter) + cardBleedH)) *
mmToPt;
```

```
// Draw Front Cell
frontPage.drawPage(embeddedFront, {
  x: posX,
  y: posY,
  width: cardBleedW * mmToPt,
  height: cardBleedH * mmToPt,
  rotate: state.isRotated ? PDFLib.degrees(-90) : PDFLib.degrees(0)
});
```

```
// Draw Back Cell (if 2-sided)
if (backPage) {
  let backCol = c;
  let backRow = r;
  if (state.duplexType === 'horizontal') {
    backCol = state.cols - 1 - c;
  } else {
    backRow = state.rows - 1 - r;
  }
  const backPosX = (startX + backCol * (cardBleedW + state.gutter)) * mmToPt;
  const backPosY = (state.sheetH - (startY + backRow * (cardBleedH + state.gutter) +
cardBleedH)) * mmToPt;
```

```
  backPage.drawPage(embeddedBack, {
    x: backPosX,
    y: backPosY,
    width: cardBleedW * mmToPt,
    height: cardBleedH * mmToPt,
    rotate: state.isRotated ? PDFLib.degrees(-90) : PDFLib.degrees(0)
  });
}

}
}
```

```
// Draw Crop Marks Vectors on PDF
drawPdfVectorCropMarks(frontPage, startX, startY, state.cols, state.rows, cardBleedW,
cardBleedH, state.bleed, state.gutter, state.cropMarkLen, state.sheetH, mmToPt);
if (backPage) {
  drawPdfVectorCropMarks(backPage, startX, startY, state.cols, state.rows, cardBleedW,
```

```

cardBleedH, state.bleed, state.gutter, state.cropMarkLen, state.sheetH, mmToPt);
    }

    // Save PDF to Bytes & Download
    const pdfBytesOut = await outPdf.save();
    const blob = new Blob([pdfBytesOut], { type: 'application/pdf' });
    const link = document.createElement('a');
    link.href = URL.createObjectURL(blob);

    const safeCustName = state.customerName.replace(/[^a-zA-Z0-9_]/g, '_');
    link.download =
`${safeCustName}_DanhThiep_${state.cardTrimW}x${state.cardTrimH}_${state.nUp}Up_${state
.sheetW}x${state.sheetH}_${state.requiredSheets}To.pdf`;
    link.click();

    } catch (err) {
        console.error(err);
        alert('Lỗi xuất PDF: ' + err.message);
    }
}

// Helper: Draw Crop Marks in PDF Vector Space
function drawPdfVectorCropMarks(page, startX, startY, cols, rows, cardBleedW, cardBleedH,
bleed, gutter, markLen, sheetH, mmToPt) {
    const cardTrimW = cardBleedW - 2 * bleed;
    const cardTrimH = cardBleedH - 2 * bleed;

    for (let r = 0; r < rows; r++) {
        for (let c = 0; c < cols; c++) {
            const cardX = startX + c * (cardBleedW + gutter) + bleed;
            const cardY = startY + r * (cardBleedH + gutter) + bleed;

            const left = cardX * mmToPt;
            const right = (cardX + cardTrimW) * mmToPt;

            // Convert Y from top-down to PDF bottom-up
            const top = (sheetH - cardY) * mmToPt;
            const bottom = (sheetH - (cardY + cardTrimH)) * mmToPt;

            const len = markLen * mmToPt;
            const gap = (bleed > 0 ? bleed : 1) * mmToPt;

            // Outer Horizontal / Vertical Marks

```

```

        if (c === 0) {
            page.drawLine({ start: { x: left - gap - len, y: top }, end: { x: left - gap, y: top }, thickness:
0.5 });
            page.drawLine({ start: { x: left - gap - len, y: bottom }, end: { x: left - gap, y: bottom },
thickness: 0.5 });
        }
        if (c === cols - 1) {
            page.drawLine({ start: { x: right + gap, y: top }, end: { x: right + gap + len, y: top },
thickness: 0.5 });
            page.drawLine({ start: { x: right + gap, y: bottom }, end: { x: right + gap + len, y: bottom },
thickness: 0.5 });
        }
        if (r === 0) {
            page.drawLine({ start: { x: left, y: top + gap + len }, end: { x: left, y: top + gap }, thickness:
0.5 });
            page.drawLine({ start: { x: right, y: top + gap + len }, end: { x: right, y: top + gap },
thickness: 0.5 });
        }
        if (r === rows - 1) {
            page.drawLine({ start: { x: left, y: bottom - gap }, end: { x: left, y: bottom - gap - len },
thickness: 0.5 });
            page.drawLine({ start: { x: right, y: bottom - gap }, end: { x: right, y: bottom - gap - len },
thickness: 0.5 });
        }
    }
}
}
}

</script>
</body>
</html>

```