Python

# Extracting core design tokens for Lekh.ai PRD into a dictionary for reference  
lekhai\_design\_tokens = {  
 "colors": {  
 "primary": {"name": "Deep Crimson", "hex": "#D90429", "tailwind": "crimson"},  
 "secondary": {"name": "Graphite", "hex": "#2B2D42", "tailwind": "graphite"},  
 "accent": {"name": "Off-White", "hex": "#EDF2F4", "tailwind": "offwhite"}  
 },  
 "typography": {  
 "ui": "Inter",  
 "script": "Hind Siliguri"  
 },  
 "grid": "8pt"  
}

Here is the complete design system documentation and the dev-ready test layout for Lekh.ai.

### Part 1: Lekh.ai Design System Documentation

This system is built using Tailwind CSS for utility-first styling, Shadcn/UI as the accessible component foundation, and Lucide React for consistent, lightweight iconography.

#### 1. Typography

The typographic scale is optimized for high legibility in the UI, while accommodating complex Bengali Unicode characters in the script generation areas.

* **Font Families:**
  + **UI & Controls:** Inter, sans-serif (Clean, modern, highly readable for small UI text).
  + **Generated Script Output:** Hind Siliguri or Kalpurush (Tailored specifically for elegant Bengali text rendering).
* **Scale & Weights:**
  + h1 (Hero Titles): text-4xl (36px) to text-5xl (48px), font-bold (700), tracking-tight.
  + h2 (Section Titles): text-2xl (24px) to text-3xl (30px), font-semibold (600).
  + h3 (Card Headers/Labels): text-lg (18px), font-medium (500).
  + p (Body text): text-base (16px), font-normal (400), leading-relaxed (1.625).
  + small (Hints/Captions): text-sm (14px), text-muted-foreground.

#### 2. Color Palette (Tailwind Configuration)

The palette is injected directly into the Tailwind configuration (or globals.css as CSS variables for Shadcn) using the brand guidelines.

* **Primary (Deep Crimson - #D90429):** Used for primary actions (Generate button), active states, and focus rings.
  + *Tokens:* --primary / bg-crimson-600
* **Secondary (Graphite - #2B2D42):** Used for standard typography, borders, and secondary buttons.
  + *Tokens:* --secondary / text-graphite-900
* **Background / Accent (Off-White - #EDF2F4):** Used for the app background to reduce eye strain compared to pure white.
  + *Tokens:* --background / bg-offwhite-100
* **Surface / Card (Pure White - #FFFFFF):** Used for input fields, dropdowns, and the generated output box to provide high contrast against the off-white background.
* **Feedback States:**
  + *Error:* text-red-500 / border-red-500 (Used when >2 tones are selected).
  + *Success:* text-emerald-600 / bg-emerald-50 (For successful copy/export actions).
  + *Muted:* text-slate-500 (For placeholders and secondary text).

#### 3. Spacing, Layout & Shadows

Adhering strictly to an **8pt grid system** to ensure vertical and horizontal rhythm.

* **Spacing Scale:** Tailwind's default scale directly maps to the 8pt system (p-2 = 8px, p-4 = 16px, p-6 = 24px).
* **Layout:** \* Max container width: max-w-7xl (1280px) with mx-auto for centering.
  + Desktop uses a two-column structural split (e.g., grid-cols-12 where input is col-span-4 and output is col-span-8).
* **Border Radius:**
  + Buttons & Inputs: rounded-md (6px) for a crisp, professional SaaS feel.
  + Cards & Main Output Box: rounded-xl (12px) for softer, modern grouping.
* **Shadows:**
  + Cards/Output Box: shadow-sm (subtle elevation) resting on the off-white background.
  + Dropdown Menus: shadow-md to establish clear depth/z-index priority.

#### 4. Iconography (Lucide)

* **Style:** Outline only.
* **Stroke Width:** 1.5 for a refined, non-heavy appearance.
* **Sizes:** size-4 (16px) for inline button icons (e.g., Copy, Download), size-5 (20px) for standard UI elements.

#### 5. Shadcn/UI Component Customization Guidelines

* **Buttons:** Standard Shadcn button overrides. The default variant uses the Deep Crimson primary color. Added a pulse animation class for the "Generating..." loading state.
* **Inputs/Textareas:** Must use focus-visible:ring-1 focus-visible:ring-crimson-600 to maintain brand identity during user input.
* **Select/Dropdowns:** Limit visual clutter. Enforce error states (red borders + slight shake animation) natively via Tailwind arbitrary variants if the user exceeds the 2-tone limit.
* **Cards:** Used for the output display. Must have bg-white, border-slate-200, and relative positioning to house the absolute-positioned top-right action icons.

### Part 2: Interactive Preview (React + Tailwind)

Below is a single-file, self-contained React component that implements the MVP interface described in your PRD. It demonstrates the typography, color palette, custom Shadcn-style components, form inputs, feedback states, and the exact Bengali script output structure.

TypeScript

import React, { useState } from 'react';  
import { Copy, Download, FileText, CheckCircle2, AlertCircle, Loader2 } from 'lucide-react';  
  
export default function LekhaiMVPPreview() {  
 const [isGenerating, setIsGenerating] = useState(false);  
 const [isCopied, setIsCopied] = useState(false);  
 const [toneError, setToneError] = useState(false);  
  
 // Simulated generation handler  
 const handleGenerate = () => {  
 setIsGenerating(true);  
 setToneError(false);  
 setTimeout(() => setIsGenerating(false), 2000);  
 };  
  
 const handleCopy = () => {  
 setIsCopied(true);  
 setTimeout(() => setIsCopied(false), 2000);  
 };  
  
 return (  
 // Background: Off-White (#EDF2F4)  
 <div className="min-h-screen bg-[#EDF2F4] text-[#2B2D42] font-sans p-6 md:p-12">  
 <div className="max-w-6xl mx-auto space-y-8">  
   
 {/\* Header / Hero Section \*/}  
 <header className="space-y-2">  
 <div className="flex items-center gap-2">  
 <div className="w-8 h-8 rounded bg-[#D90429] flex items-center justify-center text-white font-bold">  
 L  
 </div>  
 <h1 className="text-3xl font-bold tracking-tight">Lekh.ai</h1>  
 </div>  
 <p className="text-slate-500 text-lg max-w-2xl">  
 Agency-grade Bengali advertisement scripts generated in seconds. Configure your tone and industry, and let the hybrid structural engine do the rest.  
 </p>  
 </header>  
  
 {/\* Main Workspace Layout (Two Columns on Desktop) \*/}  
 <div className="grid grid-cols-1 lg:grid-cols-12 gap-8">  
   
 {/\* Left Column: Inputs & Controls \*/}  
 <div className="lg:col-span-4 space-y-6">  
 <div className="space-y-4 bg-white p-5 rounded-xl border border-slate-200 shadow-sm">  
   
 {/\* Prompt Input \*/}  
 <div className="space-y-2">  
 <label className="text-sm font-medium text-slate-700">Campaign Brief / Prompt</label>  
 <textarea   
 className="w-full h-32 px-3 py-2 text-sm border border-slate-300 rounded-md focus:outline-none focus:ring-1 focus:ring-[#D90429] transition-shadow resize-none"  
 placeholder="e.g., Write a 30s TVC for an empowering shoe brand..."  
 />  
 </div>  
  
 {/\* Tone Dropdown Simulation (with Error State) \*/}  
 <div className="space-y-2 relative">  
 <div className="flex justify-between items-center">  
 <label className="text-sm font-medium text-slate-700">Tone (Max 2)</label>  
 <span className="text-xs text-slate-400">Optional</span>  
 </div>  
 <div   
 className={`w-full px-3 py-2 text-sm border rounded-md flex justify-between items-center cursor-pointer ${toneError ? 'border-red-500 bg-red-50' : 'border-slate-300'}`}  
 onClick={() => setToneError(!toneError)} // Toggle error state for demo  
 >  
 <span className="text-slate-500">Select Tone...</span>  
 </div>  
 {/\* Success/Error Message Example \*/}  
 {toneError && (  
 <div className="flex items-center gap-1.5 text-red-600 text-xs mt-1 animate-in slide-in-from-top-1">  
 <AlertCircle className="w-3.5 h-3.5" />  
 <span>Maximum of 2 tones can be selected.</span>  
 </div>  
 )}  
 </div>  
  
 {/\* Industry Dropdown Simulation \*/}  
 <div className="space-y-2 relative">  
 <div className="flex justify-between items-center">  
 <label className="text-sm font-medium text-slate-700">Industry</label>  
 <span className="text-xs text-slate-400">Optional</span>  
 </div>  
 <select className="w-full px-3 py-2 text-sm border border-slate-300 rounded-md focus:outline-none focus:ring-1 focus:ring-[#D90429] appearance-none bg-white">  
 <option value="" disabled selected>Select Industry...</option>  
 <option value="fmcg">FMCG</option>  
 <option value="fashion">Fashion & Apparel</option>  
 <option value="real-estate">Real Estate & Construction</option>  
 </select>  
 </div>  
  
 {/\* Primary Button \*/}  
 <button   
 onClick={handleGenerate}  
 disabled={isGenerating}  
 className="w-full mt-4 flex items-center justify-center gap-2 bg-[#D90429] hover:bg-[#b00321] text-white px-4 py-2.5 rounded-md font-medium transition-all disabled:opacity-80"  
 >  
 {isGenerating ? (  
 <>  
 <Loader2 className="w-4 h-4 animate-spin" />  
 Generating Script...  
 </>  
 ) : (  
 'Generate Script'  
 )}  
 </button>  
 </div>  
 </div>  
  
 {/\* Right Column: Output Card \*/}  
 <div className="lg:col-span-8">  
 <div className="bg-white rounded-xl border border-slate-200 shadow-sm overflow-hidden h-full min-h-[500px] flex flex-col relative group">  
   
 {/\* Output Toolbar (Hover reveal icons) \*/}  
 <div className="absolute top-4 right-4 flex items-center gap-2 opacity-40 group-hover:opacity-100 transition-opacity bg-white/90 p-1 rounded-md backdrop-blur-sm shadow-sm border border-slate-100">  
 <button   
 onClick={handleCopy}  
 className="p-1.5 text-slate-500 hover:text-[#D90429] hover:bg-slate-50 rounded transition-colors"   
 title="Copy Text"  
 >  
 {isCopied ? <CheckCircle2 className="w-4 h-4 text-emerald-600" /> : <Copy className="w-4 h-4" />}  
 </button>  
 <button className="p-1.5 text-slate-500 hover:text-[#D90429] hover:bg-slate-50 rounded transition-colors" title="Copy as Markdown">  
 <FileText className="w-4 h-4" />  
 </button>  
 <button className="p-1.5 text-slate-500 hover:text-[#D90429] hover:bg-slate-50 rounded transition-colors" title="Download Word Doc">  
 <Download className="w-4 h-4" />  
 </button>  
 </div>  
  
 {/\* Dynamic Generation State vs Result \*/}  
 {isGenerating ? (  
 <div className="flex-1 flex flex-col items-center justify-center space-y-4 text-slate-400">  
 <div className="w-8 h-8 rounded-full border-2 border-slate-200 border-t-[#D90429] animate-spin" />  
 <p className="animate-pulse text-sm">Structuring table formatting via Qwen 1.5B...</p>  
 </div>  
 ) : (  
 <div className="flex-1 p-6 overflow-y-auto">  
 {/\* Generated Script using Bengali Typography \*/}  
 <div className="font-['Hind\_Siliguri',sans-serif] space-y-6">  
 <h2 className="text-2xl font-bold text-[#2B2D42]">## কনসেপ্ট: সীমানা ছাড়িয়ে (Beyond Boundaries)</h2>  
 <p className="text-slate-600 text-[17px] leading-relaxed">  
 <strong>দৃশ্যপট:</strong> একটি পাহাড়ের চূড়া বা সমুদ্রের পাড়। একজন ফ্রিল্যান্সার বা ডিজিটাল নোম্যাড (অনিক) ল্যাপটপ কোলে নিয়ে বসে আছে। সূর্য ডুবছে।  
 </p>  
   
 {/\* Markdown Table Simulation \*/}  
 <div className="overflow-x-auto rounded-lg border border-slate-200">  
 <table className="w-full text-left border-collapse">  
 <thead>  
 <tr className="bg-slate-50 text-slate-600 border-b border-slate-200 text-sm">  
 <th className="p-3 font-semibold">Scene</th>  
 <th className="p-3 font-semibold border-l border-slate-200">Visual</th>  
 <th className="p-3 font-semibold border-l border-slate-200">Audio / Dialogue</th>  
 </tr>  
 </thead>  
 <tbody className="text-[15px]">  
 <tr className="border-b border-slate-200">  
 <td className="p-3 align-top font-medium">Scene 1</td>  
 <td className="p-3 align-top border-l border-slate-200 text-slate-700">  
 অনিক টাইপ করছে। স্ক্রিনে জটিল কোডিং বা গ্রাফিক্স রেন্ডারিং চলছে। ব্যাকগ্রাউন্ডে বাতাসের শোঁ শোঁ শব্দ।  
 </td>  
 <td className="p-3 align-top border-l border-slate-200 text-slate-700">  
 <span className="font-bold">VO:</span> অফিস মানেই কি চার দেয়াল? ডেস্ক মানেই কি একঘেয়েমি?  
 </td>  
 </tr>  
 <tr>  
 <td className="p-3 align-top font-medium">Scene 2</td>  
 <td className="p-3 align-top border-l border-slate-200 text-slate-700">  
 ক্লোজ আপ: ল্যাপটপের স্লিক ডিজাইন। অনিক এক হাতে ল্যাপটপটি ধরে উঠে দাঁড়িয়ে হাঁটা শুরু করে।  
 </td>  
 <td className="p-3 align-top border-l border-slate-200 text-slate-700">  
 <span className="font-bold">VO:</span> শক্তি যখন হাতের মুঠোয়, তখন পুরো পৃথিবীটাই আমার ওয়ার্কস্পেস।  
 </td>  
 </tr>  
 </tbody>  
 </table>  
 </div>  
 </div>  
 </div>  
 )}  
 </div>  
 </div>  
  
 </div>  
 </div>  
 </div>  
 );  
}