FLASH-UI APP SUMMARY (REPO-BASED)

What it is
- Menu Flash is a React + Vite web app that uses Google Gemini models to generate printable restaurant menu HTML from user text (and optional reference images).
- It lets users iterate on multiple AI-generated design variants and print/save the result as PDF.

Who it's for
- Primary persona: restaurant operators/marketers/designers who need fast, editable, print-ready menu layouts without building templates manually.

What it does
- Generates 3 menu concepts per prompt and streams each HTML artifact live in the UI.
- Accepts optional reference image upload/paste to steer style/layout generation.
- Offers style controls: paper size (A3/A4/A5/Letter), color presets/custom colors, and font preference.
- Supports direct in-canvas editing (contenteditable text + floating formatting toolbar inside menu iframe).
- Includes element-tree editing: select, move, delete, and insert text/menu items/dividers/images/logo blocks.
- Generates additional menu variations for the currently focused design.
- Provides print preview at target paper size and opens browser print dialog for Print/Save PDF.

How it works (architecture from repo evidence)
- Frontend shell: single-page React app in index.tsx manages session state, prompts, artifacts, focus mode, and UI panels.
- AI service calls: @google/genai client in-browser calls Gemini models (gemini-3-flash-preview and gemini-2.5-flash-image) for style naming, HTML generation/streaming, element edits, and image edits.
- Rendering layer: each artifact is shown in sandboxed iframe (ArtifactCard) so generated HTML runs isolated from app chrome.
- Editing bridge: app injects INJECTED_EDITOR_SCRIPT into generated HTML; iframe and parent communicate via postMessage for tree data, element selection, HTML sync, and edit commands.
- Print/export flow: PrintPreviewModal sanitizes scripts/contenteditable attributes, injects print CSS with selected paper size, then opens window.print for PDF saving.
- Data/persistence: in-memory React state for sessions/artifacts; tutorial seen flag in localStorage; backend API, auth, and database: Not found in repo.

How to run (minimal)
1. Install dependencies: npm install
2. Add API key in .env.local: README says GEMINI_API_KEY, while code reads process.env.API_KEY (exact required key name is inconsistent in repo).
3. Start dev server: npm run dev