📄 Design Document: 3-Panel Book Builder MVP (Cursor Project)

🚀 Goal

Create a web app for children to collaboratively build a storybook with a voice/chat assistant. The child interacts with a chat (via iframe), sees a visual preview of each page (image + text), and drags chosen pages into a sequence to export as a PDF.

🧱 App Structure

Layout

plaintext

CopyEdit

```
[ Left Panel ]    [ Center Panel ]    [ Right Panel ]
Book Sequence      Preview Area          Chat Interface (iframe)
```

Components

- `PageList` (left): drag-and-drop saved pages
- `Preview` (center): current image + text + save button
- `ChatGPTIframe` (right): child speaks/interacts
- `ExportButton`: converts sequence to PDF

📦 Tech Stack

Purpose	Tool
UI Framework	React + Tailwind CSS
Chat agent	ChatGPT iframe (openai.com/chat)
Image generation	OpenAI Image API (later)

Drag-and-drop	`react-beautiful-dnd` or `dnd-kit`
PDF export	`html2pdf.js` or `jspdf`
Hosting	Vercel, Netlify, or Glitch

📁 File Structure

bash
CopyEdit

```
src/
  App.jsx
  components/
    PageList.jsx
    Preview.jsx
    ChatGPTIframe.jsx
    ExportButton.jsx
  utils/
    pdfUtils.js
public/
  index.html
```

🧠 MVP User Flow

1. Child talks to GPT via iframe (right panel).
2. User (or code) copies output → populates `Preview`.
3. User clicks “Save Page” → adds to `PageList`.
4. Pages in `PageList` can be reordered.
5. User clicks “Export Book” → PDF generated.

🔧 Placeholder Logic

Save Page Example

js
CopyEdit

```
const addPage = ({ image, text }) => {
  setPages((prev) => [...prev, { id: uuid(), image, text }]);
};
```

Export to PDF

js
CopyEdit

```
import html2pdf from 'html2pdf.js';

export const exportToPDF = (pages) => {
  const content = document.getElementById('book-preview');
  html2pdf().from(content).save('storybook.pdf');
};
```

📌 Short-Term Trusting OpenAI UX

- Use `iframe` to embed ChatGPT:

jsx
CopyEdit

```
<iframe src="https://chat.openai.com/" className="w-full h-full" />
```

- Add instructions in the UI like:

> “Ask the assistant to help make a page! When you're done, copy the story and image prompt to the center.”

🔮 Future: Add Your Own Voice Agent

When you're ready:

- Replace `ChatGPTIframe` with your custom chat agent
- Add voice input using:
  - `Web Speech API` (for browser)

    - Or `Whisper` + `Recorder.js`

Then:

- Programmatically send prompt → GPT API
- Generate image → OpenAI Image API
- Auto-populate `Preview`

✅ Stretch Features

Feature	Tool
Login / saved books	Firebase Auth + Firestore
Gallery of books	Airtable or Supabase
Real-time voice	`whisper.cpp` + WebRTC
Audio narration	`ElevenLabs` or `gTTS`