

PDFViewer Component

Build a polished PDF viewer in Vue 3 with a two-page (desktop) layout and a flip animation. Show your engineering taste: structure, UX, and decisions matter.

Vue 3

Nuxt 3

pdf.js allowed

Responsive

Public Git repo

OVERVIEW

We'd like you to implement a **PDFViewer** in **Vue 3**. Given a PDF file (local asset or URL), the component should render:

- ✓ **Two pages per screen** on desktop / large screens.
- ✓ **One page per screen** on mobile / small screens.
- ✓ A **page-flip animation** inspired by StPageFlip (you may use any library or build it yourself).
- ✓ Graceful **loading** and **error** states.

You choose the controls and features (e.g., pagination, zoom, thumbnails, fit modes). Explain your choices in the **README**.

REQUIREMENTS

- **Framework:** Vue 3 with Nuxt 3. TypeScript is optional but appreciated.
- **Layout:** 2-up desktop, 1-up mobile. Use a clean, responsive breakpoint.
- **Animation:** Flip effect similar to StPageFlip (lib optional).
- **States:** Distinct loading and error UI (with retry).
- **Repository:** Public Git repository with clear, incremental commits.

Deliverables

- Public Git repository URL. The commits organization will be evaluated
- Source code (Nuxt 3) and lockfile.
- **README** with setup, run/build commands, architecture, trade-offs, known limits.
- Short note on the chosen controls/features and why.

WHAT WE'LL LOOK AT

Core Functionality

- PDF renders reliably; navigation works (next/prev, current page).
- Responsive: 2-up desktop, 1-up mobile with smooth flip animation.
- Clear loading/error handling and recoverability.

Code Quality & Architecture

- Project structure, component boundaries, composable.
- Readable code, meaningful names, no dead code.
- State handling without race conditions or leaks.

UX & Accessibility

- Intuitive controls and sensible defaults.
- Keyboard support (e.g., ←/→ to navigate).
- Labels/ARIA where appropriate.

Performance & Reliability

- Lazy rendering/caching of pages or bitmaps.
- No obvious memory leaks; workers disposed on unmount.
- Snappy interaction without jank.

NICE-TO-HAVES (OPTIONAL)

- Zoom & fit modes.
- Thumbnails / page strip.
- Go-to page / search in document.
- Basic tests (Vitest) and a simple CI (lint/test/build).

NOTES

- Keep it simple and focused—quality over feature count.
- Document your trade-offs. If something is out of scope, say so.
- Commits will be reviewed to understand your workflow and organization.