

# Knowledge Base: How We Built It (Frontend Tech Doc – Task 5)

## Objective

This project implements a simple flow using Next.js + React + TypeScript:

- Preloader → Dashboard (FormBuilder) → Floating Window
- Includes: Toolbar, FormBuilder, Preloader, FloatingWindow components
- Styling with Tailwind CSS
- Only CSS animations (no GSAP/Framer/Anime)

## Wireframes

App Flow:

Preloader (index.tsx) → Dashboard (dashboard.tsx) → FormBuilder → FloatingWindow

Component Map:

Dashboard → Toolbar, FormBuilder, FloatingWindow (conditional, after submit)

Data Flow:

- Local state (useState) holds form data
- Props passed from parent → child
- No global state manager used

## Tech Stack & Rationale

- Framework: Next.js (SSR + routing), React, TypeScript (type safety).
- Styling: Tailwind CSS (utility-first).
- Animations: CSS keyframes and Tailwind utilities.
- State management: React useState only.

## Scope & Constraints

- Built: Preloader, Dashboard with FormBuilder, Toolbar, FloatingWindow, CSS animations.
- Not built: Modal stacking, multi-window instance, server persistence.
- Performance constraints: instant interaction, bundle < 300KB, accessible forms.
- Responsiveness: Works across mobile, tablet, desktop with Tailwind classes.

## Key APIs & Libraries

- Animation: Tailwind CSS animations.
- DOM APIs: React synthetic events.
- Routing: Next.js page routing (/ → /dashboard).
- Testing: Optional Jest/RTL.
- Third-party: Tailwind CSS only.

## Risks, Trade-offs, and Future Work

- Shortcuts: Fake preloader (no real API), single floating window.

- Edge cases: invalid input blocked locally, no server validation.
- Future: persistence, multiple windows, theming, performance profiling, a11y audits.

## How to Run

- `git clone && cd my-app`
- `npm install`
- `npm run dev`
- Open `http://localhost:3000`