

Guide to Adding New Pages and Animations to Portfolio Website

Overview

This guide explains how to add new pages and animations to an existing Next.js + Tailwind + Framer Motion portfolio website. It is structured for easy reference and written in the same instructional style as the master documentation.

1. Adding a New Page

Step 1: Create the Page File

Go to the `app/` folder.

Create a new folder for your page, for example: `app/blog`.

Inside that folder, create a `page.tsx` file:

...

```
// app/blog/page.tsx

export default function BlogPage() {

  return (
    <div className="container mx-auto py-12">
      <h1 className="text-3xl font-bold mb-4">Blog</h1>
      <p className="text-gray-400">This is the blog page.</p>
    </div>
  )
}
```

Step 2: Add to Navigation (Optional)

Open `components/layout/navbar.tsx`

Add a new link object to the `navLinks` array:

```
  { href: "/blog", label: "Blog", icon: Book },
```

Make sure to import the icon (e.g., `Book`) from `lucide-react`

2. Adding Animations with Framer Motion

Step 1: Install Framer Motion (already done in this project)

```
npm install framer-motion
```

Step 2: Animate Elements

You can wrap any element with `motion` to animate it.

Example: Fade in a heading

```
<motion.h1  
  initial={{ opacity: 0 }}  
  animate={{ opacity: 1 }}  
  transition={{ duration: 0.5 }}  
  className="text-3xl font-bold">  
  Welcome  
</motion.h1>
```

Example: Slide in from the bottom

```
<motion.div
  initial={{ opacity: 0, y: 20 }}
  animate={{ opacity: 1, y: 0 }}
  transition={{ duration: 0.5, delay: 0.2 }}
>
<p className="text-gray-400">This paragraph slides in.</p>
</motion.div>
...
```

3. Creating Animated Page Transitions (Optional Advanced)

Step 1: Create a wrapper component like `AnimatedPage.tsx`

```
...
```

```
import { motion } from 'framer-motion'

export function AnimatedPage({ children }: { children: React.ReactNode }) {
  return (
    <motion.div
      initial={{ opacity: 0 }}
      animate={{ opacity: 1 }}
      exit={{ opacity: 0 }}
      transition={{ duration: 0.4 }}
    >
      {children}
    </motion.div>
  )
}
```

Step 2: Use it in your pages

...

```
import { AnimatedPage } from '@/components/animated-page'
```

```
export default function BlogPage() {
```

```
    return (
```

```
        <AnimatedPage>
```

```
            <h1 className="text-3xl">Blog Page</h1>
```

```
        </AnimatedPage>
```

```
    )
```

```
}
```

...

4. Notes

- Use motion sparingly to maintain performance.
- Keep all animations under `0.5s` unless for overlays.
- Use the same Tailwind structure and layout styling conventions.