Your name

2025-10-19

Below is a **step-by-step recipe** for building a minimal but fully functional Astro blog that:

1. Uses **Tailwind CSS** for styling
2. Has a **sticky header** that stays on top while scrolling
3. Shows images that **fade in/out (or slide)** as the user scrolls down the page

Feel free to copy-paste the snippets, tweak the styles, and expand the content as you grow your blog.

1. Project Setup

```
# Create a new Astro project (if you haven't already)
npm create astro@latest my-astro-blog
cd my-astro-blog

# Install Tailwind CSS and its dependencies
npm install -D tailwindcss postcss autoprefixer

# Initialise Tailwind
npx tailwindcss init -p   # creates tailwind.config.js & postcss.config.cjs

# Install a small utility for scroll animations (optional)
npm install -D @tweenjs/tween.js
```

tailwind.config.cjs

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: [
    "./src/**/*.{astro,js,jsx,ts,tsx}",
    // If you add more directories, list them here
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

src/styles/tailwind.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

/* Custom utility for the sticky header */
.sticky-header {
  @apply fixed top-0 w-full z-50 shadow-md bg-white dark:bg-gray-900;
}
```

src/layouts/Base.astro

```
---
const { title } = Astro.props;
---

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>{title ? `${title} | My Astro Blog` : "My Astro Blog"}</title>
    <link rel="stylesheet" href="/src/styles/tailwind.css" />
  </head>
  <body class="bg-gray-50 dark:bg-gray-800 text-gray-900 dark:text-gray-100">
    <slot />
  </body>
</html>
```

2. Sticky Header Component

Create `src/components/Header.astro`:

```
---
const { navLinks } = Astro.props;
---

<header class="sticky-header flex items-center justify-between px-6 py-4">
  <h1 class="text-xl font-bold text-indigo-600 dark:text-indigo-400">My Astro
Blog</h1>
  <nav class="space-x-4">
    {navLinks.map(link => (
      <a href={link.href} class="hover:underline">
        {link.text}
      </a>
    ))}
  </nav>
</header>
```

Add a few links in the main page:

```
---
import Header from '../components/Header.astro';

const navLinks = [
  { href: '/', text: 'Home' },
  { href: '/about', text: 'About' },
  { href: '/blog', text: 'Blog' },
];
---
<Header navLinks={navLinks} />
```

Because the header uses `fixed top-0`, it will stay visible as you scroll.

3. Scroll-Triggered Image Animations

We'll use **IntersectionObserver** (built-in, no extra libs) to add a CSS class when an image enters the viewport. The class will apply a transition (fade/slide).

3-1. Utility Hook

Create `src/utils/useIntersection.ts`:

```
import { onMount } from 'solid-js';

export function useIntersection(
  element: HTMLElement,
  options: IntersectionObserverInit = { threshold: 0.2 }
) {
  let observer: IntersectionObserver | null = null;

  onMount(() => {
    if (!element) return;
    observer = new IntersectionObserver((entries) => {
      entries.forEach((entry) => {
        if (entry.isIntersecting) {
          element.classList.add('animate-visible');
        } else {
          // If you want images to animate back out when scrolling up:
          element.classList.remove('animate-visible');
        }
      });
    }, options);

    observer.observe(element);
  });

  return () => {
    if (observer) observer.disconnect();
  };
}
```

3-2. Image Component

Create `src/components/ScrollImage.astro`:

```
---
import { useIntersection } from '../utils/useIntersection';
import type { ComponentProps } from 'solid-js';

const props = Astro.props as {
  src: string;
  alt?: string;
  class?: string;
};
---

<img
  ref={useIntersection}
  src={props.src}
  alt={props.alt ?? ''}
  class={`transform opacity-0 transition-all duration-700 ease-out
${props.class}`}
  loading="lazy"
/>
```

3-3. Tailwind Animations

Add to `src/styles/tailwind.css`:

```
/* Fade in from the bottom */
@keyframes fadeInUp {
  0% {
    opacity: 0;
    transform: translateY(20px);
  }
  100% {
    opacity: 1;
    transform: translateY(0);
  }
}

.animate-visible {
  animation: fadeInUp 1s forwards;
}
```

4. Putting It All Together

Create a simple page, e.g., `src/pages/index.astro`:

```
---
import Header from '../components/Header.astro';
import ScrollImage from '../components/ScrollImage.astro';
```

```
const navLinks = [
  { href: '/', text: 'Home' },
  { href: '/about', text: 'About' },
  { href: '/blog', text: 'Blog' },
];
---

<Base title="Home">
  <Header navLinks={navLinks} />

  <!-- Hero section -->
  <section class="pt-20 bg-indigo-50 dark:bg-indigo-900 py-12">
    <div class="max-w-4xl mx-auto text-center space-y-6">
      <h2 class="text-3xl font-bold">Welcome to My Astro Blog</h2>
      <p class="text-lg text-gray-700 dark:text-gray-300">
        Dive into articles about web dev, design, and more.
      </p>
    </div>
  </section>

  <!-- Content with images -->
  <main class="max-w-4xl mx-auto py-12 space-y-20">
    {Array.from({ length: 6 }).map((_, i) => (
      <section class="space-y-4">
        <h3 class="text-xl font-semibold">Article {i + 1}</h3>
        <p class="text-gray-600 dark:text-gray-400">
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
          luctus.
        </p>
        <ScrollImage
          src={`/images/sample-${i + 1}.jpg`}
          alt={`Sample image ${i + 1}`}
        />
      </section>
    ))}
  </main>

  <footer class="py-6 text-center text-sm text-gray-500 dark:text-gray-400">
```

```
    © 2025 My Astro Blog
  </footer>
</Base>
```

Tip: *Put placeholder images in `public/images/` or use unsplash URLs.*

5. Run & Build

```
# Development server
npm run dev

# Production build
npm run build
```

Open `http://localhost:3000` and scroll. The header stays fixed, and each image fades in as it enters the viewport.

6. Enhancements & Variations

Idea	How to Implement
Lazy-load images	Add `loading="lazy"` (already included).
Slide-in from left/right	Change the `@keyframes` to translateX instead of Y.
Animate on scroll up	Keep the `else` block in IntersectionObserver to remove the class.
Use a library (GSAP, ScrollMagic)	Import GSAP and use its `ScrollTrigger` for more complex sequences.
Dark mode toggle	Add a button that toggles `class="dark"` on `<html>`.
Pagination / Infinite scroll	Use Astro's `client:load` components or fetch API data.

7. Recap

- **Astro** gives you a fast, framework-agnostic site generator.
- **Tailwind CSS** handles all styling; the sticky header uses `fixed top-0`.
- **IntersectionObserver** triggers a CSS animation class on images, creating a smooth fade/slide effect as the user scrolls.

Happy coding! If you run into any snags or want to add more features, just let me know.