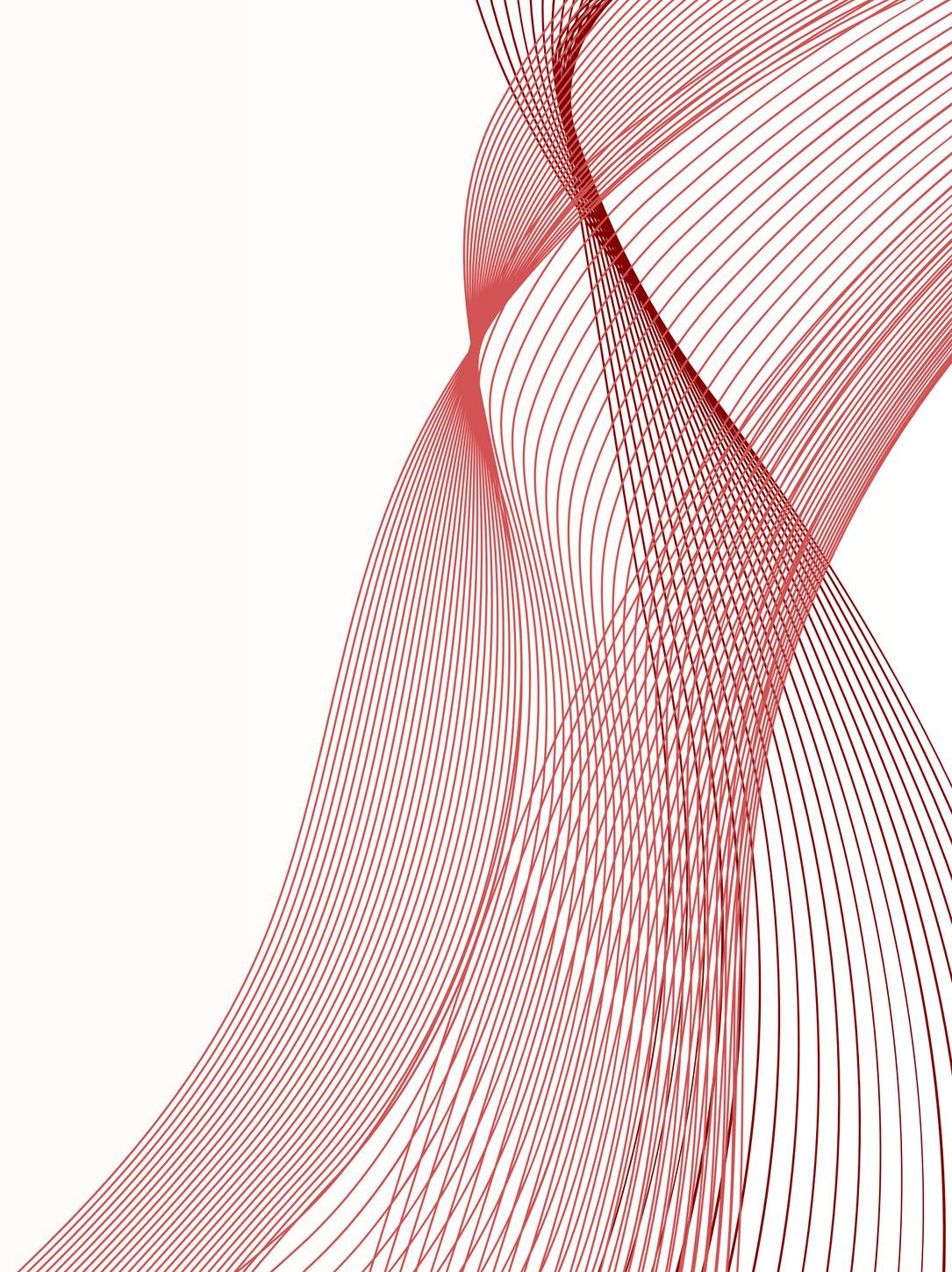
FRAMER
MOTION

~ Hunny Parmar



INTRODUCTION

What is Framer Motion?

- Framer Motion is a powerful React animation library.
- Designed to make complex animations easy with a declarative API.
- Built on top of Framer, a prototyping tool.
- Provides smooth, high-performance animations.
- Alternative to CSS animations, GSAP, and React Spring.

WHY USE FRAMER MOTION?

- **Simple & Declarative:**

  Easy-to-read syntax.

- **Optimized Performance:**

  Uses animation techniques like layout projection.

- **Rich Features:**

  Includes drag, scroll, and gesture-based animations.

- **React-Friendly:**

  Works seamlessly within React components.

- **Reusable Variants:**

  Allows reusable animation presets.

CORE FEATURES OF FRAMER MOTION

- **Animate & Initial:**

  Define starting and ending states.

- **Variants:**

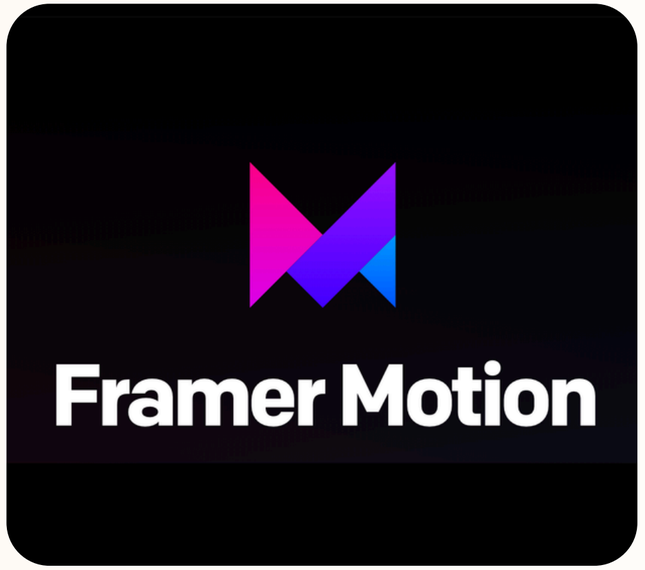
  Reusable animation objects.

- **Transitions**

  Control duration and easing.

- **Gestures (Hover & Tap):**

  Control duration and easing.



- **Drag & Drop:**

  Built-in draggable elements.

- **Scroll Animations:**

  Animate elements based on scroll.

- **Keyframe Animations:**

  Multi-step animations.

Efficient Image Loading

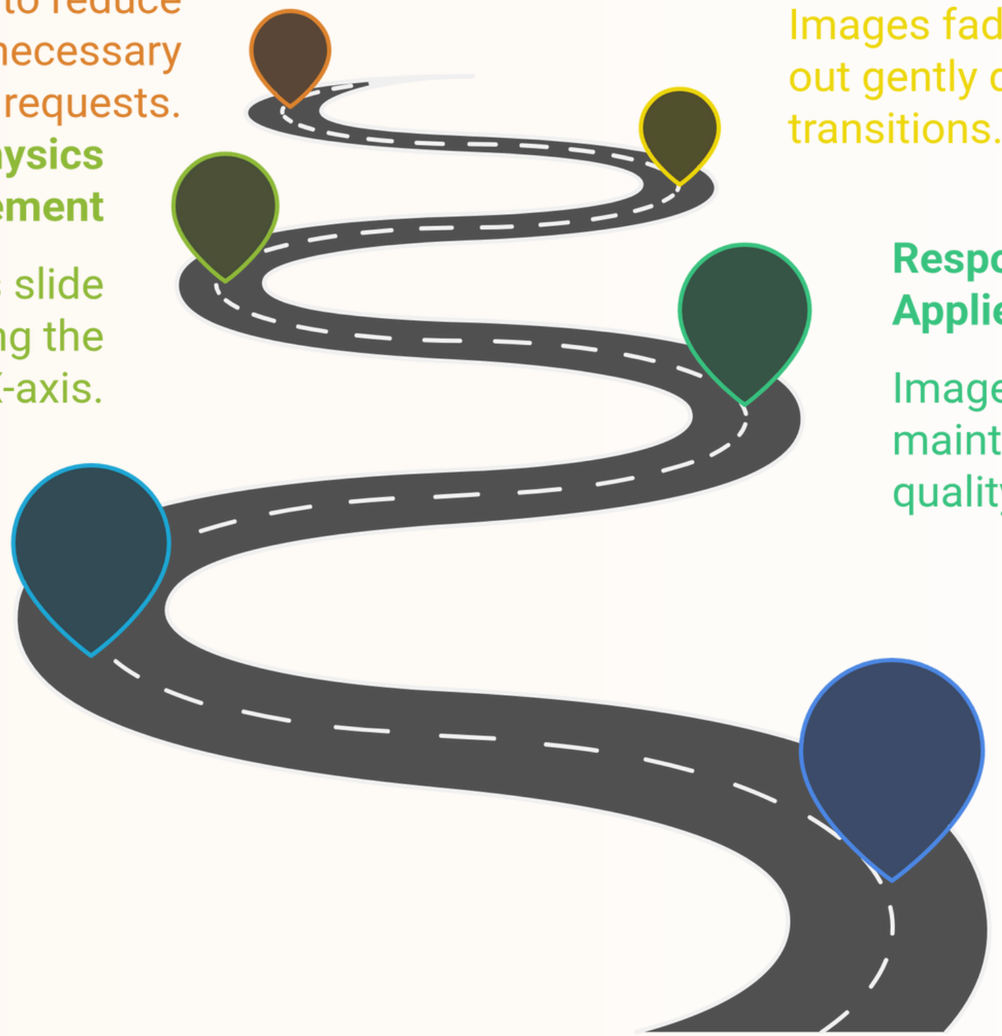
Loading is optimized to reduce unnecessary requests.

Spring Physics Movement

Images slide smoothly along the X-axis.

Size Throttling Activates

Throttling alters ıages to maintain clarity.



Opacity Transition

Images fade in and out gently during transitions.

Responsive Width Applied

Images adapt to maintain optimal quality.

Container Resizin

The carousel container adjusts i size.

INSTALLATION & SETUP

- **Installing Framer Motion**

```bash
npm install framer-motion
```

- **Basic Import in a Component**

```jsx
import { motion } from "framer-motion";
```

- **Basic Animated Component**

```jsx
<motion.div animate={{ opacity: 1 }} initial={{ opacity: 0 }} />
```

UNDERSTANDING ANIMATE & INITIAL

- **animate:**

  Defines the final state.

- **initial:**

  Defines the starting state.

- **Example:**

```jsx
<motion.div animate={{ opacity: 1 }} initial={{ opacity: 0 }} />
```

- **Result:**

The `div` fades in when mounted.

UNDERSTANDING MOTION COMPONENTS

- motion.div, motion.span, motion.button, etc

Works the same as regular HTML elements but with animation properties.

Example:

```
<motion.button whileHover={{ scale: 1.2 }}>Hover me</motion.button>
```

USING VARIANTS (REUSABLE ANIMATIONS)

- **Variants help reuse animations efficiently.**

  Example:

jsx Copy Edit

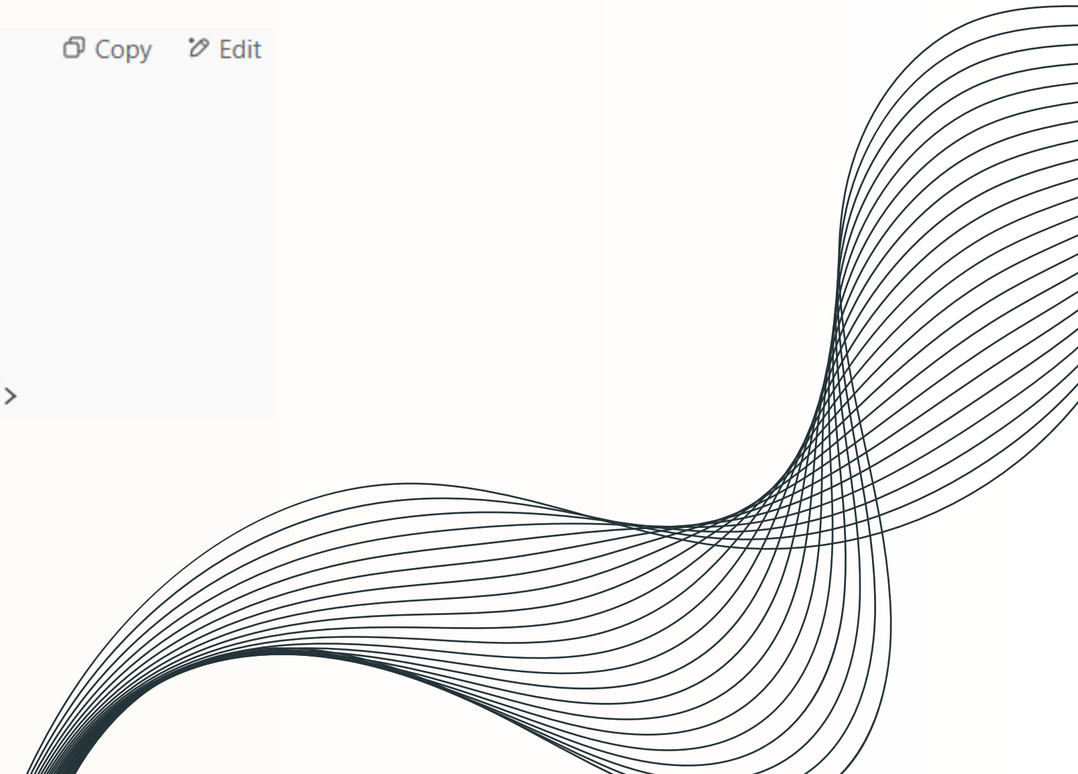
```jsx
const boxVariants = {
  hidden: { opacity: 0, y: -50 },
  visible: { opacity: 1, y: 0 }
};

<motion.div variants={boxVariants} initial="hidden" animate="visible" />
```

- **Why Use Variants?**

  Cleaner and reusable animation logic.

  Simplifies complex animations.



TRANSITION EFFECTS IN FRAMER MOTION

- **Purpose:**

  Controls the speed and behavior of animations.

- **Types:**

ease, spring, keyframes

- **Example:**

  Using spring for a bouncing effect:

```
<motion.div animate={{ y: [0, -30, 0] }} transition={{ type: 'spring' }} />
```

INTERACTIVE ANIMATIONS (GESTURES)

- **Hover Effects:**

  Elements grow or change color on hover.

- **Tap Effects:**

  Buttons respond visually when clicked.

- **Drag & Drop:**

  Enables moving UI elements interactively.

- **Example:**

```
<motion.div drag dragConstraints={{ left: -100, right: 100 }} />
```

SCROLL ANIMATIONS

- **Definition:**

  Triggers animations when an element enters the viewport.

- **Use Cases:**

  Fade-in effects for sections, progressive reveals.

- **Impact:**

  Improves storytelling and user engagement.

- **Example:**

```
import { useInView } from 'framer-motion';
function Component() {
  const ref = useRef(null);
  const isInView = useInView(ref);
  return <motion.div ref={ref} animate={{ opacity: isInView ? 1 : 0 }} />;
}
```

REAL-WORLD APPLICATIONS OF FRAMER MOTION

- **Websites:**

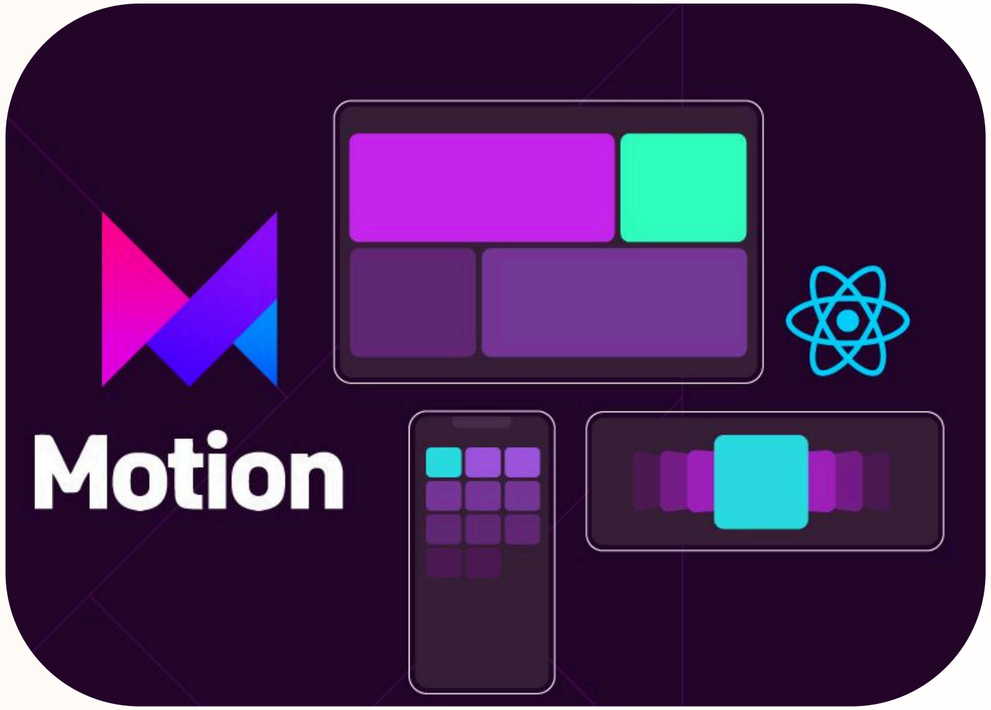
  Engaging landing pages and UI components.

- **Dashboards:**

  Smooth data visualization transitions.

- **Mobile Apps:**

  Enhancing interactivity and responsiveness.

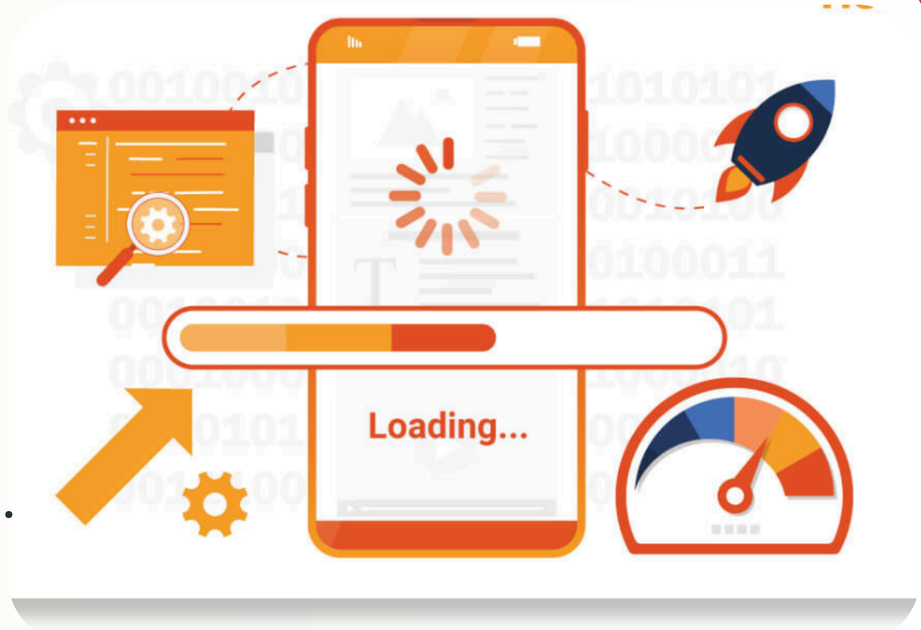


BEST PRACTICES FOR USING FRAMER MOTION

- **Keep animations subtle** to avoid overwhelming users.
- **Optimize for performance** by limiting unnecessary animations.
- **Use Variants** for consistency and maintainability.
- **Test responsiveness** to ensure smooth transitions on all devices.

PERFORMANCE OPTIMIZATION IN FRAMER MOTION

- **Avoid excessive re-renders** by optimizing component updates.
- **Use Layout Animations** for smooth transitions.
- **Lazy load animations** for better performance.
- **Compare performance** with CSS animations.



COMPARISON WITH OTHER ANIMATION LIBRARIES

- **Framer Motion vs. GSAP:**
  Framer is easier for React, GSAP is more flexible.

- **Framer Motion vs. React Spring:**
  Framer is declarative, Spring is physics-based.

- **Framer Motion vs. Anime.js:**
  Framer integrates better with React, Anime.js is JS-based.

CONCLUSION

Framer Motion makes animation in React easier and more powerful.

- **Key Takeaways:**
  - Simple syntax and ease of use.
  - High performance and optimized rendering.
  - Built-in gestures and interactive animations.
  - Ideal for modern UI/UX improvements.
- **Final Thoughts:**
  - **Framer Motion** is a must-have for React developers aiming for engaging interfaces.
  - Continue exploring advanced features and best practices for better animations.

```jsx
import { motion } from "framer-motion";

export default function App() {
  return (
    <div className="min-h-screen flex flex-col items-center bg-gradient-to-r from-pink-200 via-purple-200 to-blue-200 text-gray-800">
      <section className="h-screen flex flex-col justify-center items-center text-center space-y-6 px-6">
        <motion.h1
          initial={{ opacity: 0, y: -30 }}
          animate={{ opacity: 1, y: 0 }}
          transition={{ duration: 1 }}
          className="text-[100px] font-extrabold text-gray-900"
        >
          Framer Motion
        </motion.h1>
        <motion.p
          initial={{ opacity: 0, y: 30 }}
          animate={{ opacity: 1, y: 0 }}
          transition={{ duration: 1, delay: 0.5 }}
          className="text-3xl text-gray-700 max-w-xl"
        >
          Simplifying React animations
        </motion.p>
        <motion.div
          animate={{ y: [0, 10, 0] }}
          transition={{ repeat: Infinity, duration: 1 }}
          className="text-gray-600 text-sm mt-6"
        >
          ↓ Scroll Down to See More ↓
        </motion.div>
      </section>

      <section className="min-h-screen flex flex-col justify-center items-center text-center space-y-12 px-6">
        <h2 className="text-4xl font-bold text-gray-900">Why Use Framer Motion?</h2>
        <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
```

```jsx
        <motion.div
          initial={{ opacity: 0, y: 0 }}
          whileInView={{ opacity: 1, y: 0 }}
          transition={{ duration: 0.8, delay:"0.4", ease: "easeOut" }}
          viewport={{ amount: 0.5 }}
          className="bg-white p-6 rounded-xl shadow-lg hover:shadow-2xl transition duration-300"
        >
          <h3 className="text-xl font-semibold text-gray-900">✨ Smooth UI Animations</h3>
          <p className="text-gray-600">Create high-performance animations effortlessly.</p>
        </motion.div>
        <motion.div
          initial={{ opacity: 0, y: 0 }}
          whileInView={{ opacity: 1, y: 0 }}
          transition={{ duration: 0.8, delay:"0.4", ease: "easeOut" }}
          viewport={{ amount: 0.5 }}
          className="bg-white p-6 rounded-xl shadow-lg hover:shadow-2xl transition duration-300"
        >
          <h3 className="text-xl font-semibold text-gray-900">💡 Interactive Motion Effects</h3>
          <p className="text-gray-600">Use gestures, hover, and scroll-based animations.</p>
        </motion.div>
        <motion.div
          initial={{ opacity: 0, y: 0 }}
          whileInView={{ opacity: 1, y: 0 }}
          transition={{ duration: 0.8, delay:"0.4", ease: "easeOut" }}
          viewport={{ amount: 0.5 }}
          className="bg-white p-6 rounded-xl shadow-lg hover:shadow-2xl transition duration-300"
        >
          <h3 className="text-xl font-semibold text-gray-900">🚀 Optimized for Performance</h3>
          <p className="text-gray-600">Built for smooth, efficient animations in React.</p>
        </motion.div>
      </div>
    </section>
  </div>
);
```

THANK YOU