

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Advanced Test Page</title>

  <!-- Meta -->
  <meta name="description" content="A sleek, high-performance landing page
with particles, typed headline, and responsive layout." />
  <meta name="theme-color" content="#121212" />
  <meta property="og:title" content="Advanced Test Page" />
  <meta property="og:description" content="Sleek, high-performance landing
page." />
  <meta property="og:type" content="website" />

  <!-- Favicon (Dark & Royal crown) -->
  <link rel="icon" type="image/svg+xml" href="data:image/svg+xml,%3Csvg
xmlns='http://www.w3.org/2000/svg' viewBox='0 0 64 64'%3E%3Crect width='64'
height='64' rx='12' fill='%230b0b0f'/%3E%3Cpath d='M12 24l8 7 6-11 6 12 6-12 6
11 8-7v22H12z' fill='%23cda434'/%3E%3Crect x='18' y='44' width='28'
height='6' rx='3' fill='%236b4eff'/%3E%3C/svg%3E" />
  <link rel="apple-touch-icon" href="data:image/svg+xml,%3Csvg xmlns='http://
www.w3.org/2000/svg' viewBox='0 0 64 64'%3E%3Crect width='64' height='64'
rx='12' fill='%230b0b0f'/%3E%3Cpath d='M12 24l8 7 6-11 6 12 6-12 6 11
8-7v22H12z' fill='%23cda434'/%3E%3Crect x='18' y='44' width='28' height='6'
rx='3' fill='%236b4eff'/%3E%3C/svg%3E" />

  <!-- Fonts -->
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
  <link rel="preconnect" href="https://fonts.googleapis.com" />
  <link href="https://fonts.googleapis.com/css2?
family=Poppins:wght@300;400;700&display=swap" rel="stylesheet" />

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/
bootstrap.min.css"
    rel="stylesheet"
    integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
    crossorigin="anonymous" />

  <style>
    :root {

```

```

--bg:#0b0b0f;    /* Deep charcoal */
--text:#f5f5f7;   /* Soft off-white */
--glass:rgba(255,255,255,0.06);
--gold:#cda434;   /* Royal gold */
--purple:#6b4eff; /* Royal purple */
--muted:#8a8fa3;  /* Muted text */
}
html { scroll-behavior: smooth; }
body {
  font-family: 'Poppins', sans-serif;
  background-color: var(--bg);
  color: var(--text);
  margin: 0;
  overflow-x: hidden;
}

/* Fullscreen particle canvas (tsParticles) */
#tsparticles {
  position: fixed;
  inset: 0;
  z-index: -1;
}

.hero {
  min-height: 100svh; /* better than 100vh on mobile */
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  text-align: center;
  padding: 0 20px;
}

h1 {
  font-size: clamp(2.5rem, 8vw, 5rem);
  font-weight: 700;
  margin-bottom: 20px;
  text-shadow: 0 0 10px rgba(255,255,255,0.5);
}

p.lead {
  font-size: clamp(1.05rem, 2.5vw, 1.5rem);
  max-width: 800px;
  margin: 0 auto 40px;
}

```

```

}

/* Color cycle text animation */
@keyframes colorCycle {
  0% { color:#ff4d4d; text-shadow:0 0 20px #ff4d4d; }
  25% { color:#4d79ff; text-shadow:0 0 20px #4d79ff; }
  50% { color:#4dff88; text-shadow:0 0 20px #4dff88; }
  75% { color:#ffd24d; text-shadow:0 0 20px #ffd24d; }
  100% { color:#ff4dff; text-shadow:0 0 20px #ff4dff; }
}
.color-cycle { animation: colorCycle 10s infinite alternate ease-in-out; }

.features {
  padding: 80px 0;
  background: linear-gradient(to bottom, rgba(18,18,18,0.8), var(--bg));
}
.feature-card {
  background: var(--glass);
  border-radius: 15px;
  padding: 30px;
  transition: transform .3s ease, box-shadow .3s ease;
}
.feature-card:hover {
  transform: translateY(-10px);
  box-shadow: 0 10px 20px rgba(0,0,0,0.2);
}

.cta { padding: 80px 0; text-align: center; background: var(--bg); }
.btn-primary { background: var(--gold); border-color: var(--gold); }
.btn-primary:hover { filter: brightness(1.05); }
a { color: var(--purple); }
.cta form { max-width: 520px; margin: 0 auto; }

footer { padding: 40px 0; text-align: center; background: #000; font-size: .9rem;
color: var(--muted); }

/* Reduced motion: disable animations/particles */
@media (prefers-reduced-motion: reduce) {
  .color-cycle, * { animation: none !important; transition: none !important; }
  #tsparticles { display: none !important; }
}
</style>
</head>
<body>

```

```

<!-- Particles -->
<div id="tsparticles" aria-hidden="true"></div>

<!-- Hero -->
<section class="hero" aria-labelledby="page-title">
  <!-- Accessible fallback + animated twin -->
  <h1 id="page-title">
    <span class="visually-hidden">Hello World 🙌</span>
    <span id="typed" class="color-cycle" aria-hidden="true">Hello World 🙌</
span>
  </h1>
  <p class="lead">
    This is your live, powerful GitHub Pages site—elevated with cutting-edge
design, interactive elements,
    and nuanced animations to captivate and convert.
  </p>
  <a href="#features" class="btn btn-lg btn-outline-light">Explore Features</a>
</section>

<!-- Features -->
<section id="features" class="features" aria-labelledby="why-title">
  <div class="container">
    <h2 id="why-title" class="text-center mb-5">Why This Site Stands Out</h2>
    <div class="row">
      <div class="col-md-4 mb-4">
        <div class="feature-card h-100">
          <h3>Dynamic Particles</h3>
          <p>True animated color cycling powered by tsParticles—smooth, subtle,
and performant.</p>
        </div>
      </div>
      <div class="col-md-4 mb-4">
        <div class="feature-card h-100">
          <h3>Seamless Animations</h3>
          <p>Color cycling with easing for premium feel, with automatic fallbacks for
reduced motion.</p>
        </div>
      </div>
      <div class="col-md-4 mb-4">
        <div class="feature-card h-100">
          <h3>Responsive & Interactive</h3>
          <p>Fully responsive layout, hover polish, and typed headline with an
accessible text fallback.</p>
        </div>
      </div>
    </div>
  </div>

```

```
</div>
</div>
</div>
</section>
```

```
<!-- CTA -->
```

```
<section class="cta" aria-labelledby="cta-title">
```

```
<div class="container">
```

```
<h2 id="cta-title">Join the Revolution</h2>
```

```
<p>Sign up for updates and make this your million-dollar asset.</p>
```

```
<!-- Default: client-side success (launchable without a backend) -->
```

```
<!-- To use Mailchimp: replace the <form> action with your list's POST URL
and keep method="POST".
```

```
    To use ConvertKit: replace action with your form endpoint (e.g., https://
api.convertkit.com/forms/XXXX/subscriptions). -->
```

```
<form id="subscribe-form" action="https://api.convertkit.com/forms/
YOUR_FORM_ID/subscriptions" method="POST" novalidate>
```

```
<div class="input-group mb-2">
```

```
<label class="visually-hidden" for="email">Your Email</label>
```

```
<input id="email" name="email" type="email" class="form-control"
placeholder="Your Email" aria-label="Your Email" required />
```

```
<button id="subscribe-btn" class="btn btn-primary"
type="submit">Subscribe</button>
```

```
</div>
```

```
<small class="text-secondary">No spam. Unsubscribe any time.</small>
```

```
<div id="form-msg" class="mt-3" role="status" aria-live="polite"
style="display:none;"></div>
```

```
</form>
```

```
</div>
```

```
</section>
```

```
<footer>
```

```
<p>&copy; 2025 Your Powerful Site. All rights reserved.</p>
```

```
</footer>
```

```
<!-- Vendor scripts (deferred) -->
```

```
<script defer src="https://cdn.jsdelivr.net/npm/tsparticles@3/
tsparticles.bundle.min.js"></script>
```

```
<script defer src="https://cdn.jsdelivr.net/npm/typed.js@2.1.0/dist/
typed.umd.js"></script>
```

```
<!-- App script -->
```

```
<script>
```

```

window.addEventListener('DOMContentLoaded', function () {
  const reduceMotion = window.matchMedia('(prefers-reduced-motion:
reduce)').matches;

  /* tsParticles with animated hue (true color cycling) */
  if (!reduceMotion && window.tsParticles) {
    const isSmall = window.matchMedia('(max-width: 576px)').matches;
    tsParticles.load('tsparticles', {
      fpsLimit: 60,
      background: { color: { value: '#121212' } },
      particles: {
        number: { value: isSmall ? 24 : 56, density: { enable: true, area: 800 } },
        move: { enable: true, speed: 2, random: true, outModes: { default: 'out' } },
        opacity: { value: 0.25, random: { enable: true, minimumValue: 0.1 } },
animation: { enable: true, speed: 1, minimumValue: 0.1 } },
        size: { value: { min: 1, max: 3 } },
        shape: { type: 'circle' },
        color: {
          value: { h: 0, s: 100, l: 60 },
          animation: { enable: true, speed: 10, sync: false }
        },
      },
      links: {
        enable: true,
        distance: 150,
        opacity: 0.2,
        width: 1,
        color: '#ffffff'
      },
    },
    interactivity: {
      events: {
        onHover: { enable: true, mode: 'grab' },
        onClick: { enable: true, mode: 'push' },
        resize: true
      },
      modes: {
        grab: { distance: 200, links: { opacity: 0.5 } },
        push: { quantity: 3 }
      }
    },
    detectRetina: true
  });
} else {
  const p = document.getElementById('tsparticles');

```

```

    if (p) p.style.display = 'none';
  }

  /* Typed headline (keeps accessible fallback visible to AT/SEO) */
  if (!reduceMotion && window.Typed) {
    new Typed('#typed', {
      strings: ['Hello World 🙌'],
      typeSpeed: 100,
      backSpeed: 0,
      showCursor: false
    });
  }

  /* Form: launchable by default (shows success message) */
  const form = document.getElementById('subscribe-form');
  const email = document.getElementById('email');
  const btn = document.getElementById('subscribe-btn');
  const msg = document.getElementById('form-msg');

  form.addEventListener('submit', async (e) => {
    const hasRealAction = form.getAttribute('action') &&
    form.getAttribute('action').length > 0;
    if (hasRealAction) { return; } // Let ConvertKit/Mailchimp handle it when action
    is set

    e.preventDefault();
    if (!email.checkValidity()) {
      msg.style.display = 'block';
      msg.className = 'mt-3 text-danger';
      msg.textContent = 'Please enter a valid email.';
      email.focus();
      return;
    }

    // DEFAULT: client-only success
    btn.disabled = true;
    msg.style.display = 'block';
    msg.className = 'mt-3 text-info';
    msg.textContent = 'Submitting...';

    await new Promise(r => setTimeout(r, 500));
    msg.className = 'mt-3 text-success';
    msg.textContent = 'Success! You're on the list — check your inbox.';
    form.reset();
  });

```

```
        btn.disabled = false;
    });
});
</script>

<noscript>
    <style>#tsparticles{display:none} .color-cycle{animation:none}</style>
</noscript>
</body>
</html>
```