CloudNova Landing Page

A modern, responsive landing page for a cloud infrastructure platform. This project demonstrates clean HTML and CSS structure, accessibility, and professional UI/UX principles.

Table of Contents

- CloudNova Landing Page
  - Table of Contents
  - Overview
  - Project Structure
  - How the Code Works
    - HTML Structure
    - Key Sections & Tags
    - Classes & IDs
    - Why Online Images and SVGs?
    - How to Modify or Extend
    - Accessibility & Responsiveness
  - For Developers & Colleagues
  - Code Examples: HTML & CSS With Explanations
    - 1. Header & Navigation
    - 2. Hero Section
    - 3. Platform (Security & Trust) Cards
    - 4. Pricing Cards
    - 5. Testimonials
    - 6. Contact Form

Overview

This landing page is designed for a fictional cloud platform, CloudNova. It features a hero section, solutions/features, a platform trust section, pricing, testimonials, about, and contact. The design is minimal, professional, and fully responsive.

Project Structure

```
index.html         # Main HTML file
styles.css         # Main CSS file
README.md          # This documentation
```

How the Code Works

HTML Structure

- `<header>`: Contains the navigation bar and brand.
- `<main>`: Contains all main content sections:
  - **Hero**: Main headline, highlights, and a side image.
  - **Solutions**: Feature cards for core services.
  - **Platform**: Security & trust features in a card layout.
  - **Pricing**: Simple, clear pricing cards.
  - **Testimonials**: Customer feedback cards.
  - **About**: Company mission and careers.
  - **Contact**: Contact form and company info.
- `<footer>`: Brand, copyright, and links.

Key Sections & Tags

- `<section>`: Used for each major content block, with descriptive `id` attributes for navigation and accessibility.
- `<div class="container">`: Centers and constrains content width.
- `<ul>`, `<li>`: Used for navigation, feature lists, and highlights.
- `<form>`: Contact form with labeled fields.
- `<svg>`: Inline icons for features and platform trust cards.
- `<img>`: Used for the hero side image (with online source for reliability and modern visuals).

Classes & IDs

- **Classes**: Used for layout, styling, and component structure. They enable modular, reusable, and responsive design.
- **IDs**: Used for navigation (anchor links) and accessibility (ARIA labels).

Why Online Images and SVGs?

- **Online images** (from Unsplash) are used for hero and media sections for high quality, fast loading, and no need for local storage.
- **SVG icons** are used for features and trust cards for scalability, clarity, and fast rendering.

How to Modify or Extend

- **Change content**: Edit the text in `index.html`.
- **Change images**: Replace the image URLs or SVGs in the HTML.
- **Change colors or layout**: Edit CSS variables in `:root` or the relevant class in `styles.css`.
- **Add features**: Copy an existing card/section and update its content and icon.

Accessibility & Responsiveness

- Semantic HTML and ARIA labels are used for accessibility.
- CSS grid and flexbox ensure the site works on all devices.
- Color contrast and focus styles are included for usability.

For Developers & Colleagues

This section is for anyone new to the codebase. It explains how to read, navigate, and extend the project:

- The code is organized into clear sections using semantic HTML tags (`<header>`, `<main>`, `<section>`, `<footer>`).
- Each section (hero, solutions, platform, pricing, testimonials, about, contact) is wrapped in a `<section>` tag with a unique `id` for navigation and clarity.
- Classes are used for layout and styling. IDs are used for navigation and accessibility.
- SVG icons and online images are used for modern visuals and scalability.
- To add or change features, copy an existing card/section and update its content and icon.
- Accessibility and responsiveness are built in by default.

Code Examples: HTML & CSS With Explanations

Below are key code snippets from the project, with comments on what each does.

1. Header & Navigation

HTML:

```html
<header class="site-header">
  <div class="container header-inner">
    <a class="brand" href="#">
      <span class="brand-mark">☁️</span>
      <span class="brand-text">CloudNova</span>
    </a>
    <nav class="site-nav">
      <ul class="nav-list">
        <li><a href="#solutions">Solutions</a></li>
        <li><a class="btn btn-small btn-primary" href="#contact">Get started</a>
</li>
      </ul>
    </nav>
  </div>
</header>
```

- `.site-header`: Sticky top bar for navigation.
- `.brand`: Logo and brand name.
- `.site-nav`, `.nav-list`: Navigation links. `.btn-primary` is styled for high contrast.

CSS:

```css
.site-header {
  position: sticky;
  top: 0;
  background: #fff;
  border-bottom: 1px solid var(--border);
}
.site-nav .btn-primary {
  color: #fff !important;
  background: var(--black);
}
```

- `.site-header`: Makes the header stick to the top.
- `.btn-primary`: White text, black background for nav button.

---

2. Hero Section

HTML:

```html
<section class="hero">
  <div class="container hero-inner">
    <div class="hero-copy">
      <h1>Enterprise Cloud Infrastructure That Scales With You</h1>
      <div class="hero-cta">
        <a class="btn btn-primary" href="#contact">Start free trial</a>
      </div>
    </div>
    <div class="hero-visual">
      <picture class="hero-img-wrap">
        <img class="hero-img" src="..." alt="Cloud illustration" />
      </picture>
    </div>
  </div>
</section>
```

- `.hero-inner`: CSS grid for two columns (text and image).
- `.hero-img-wrap`, `.hero-img`: Responsive image beside the text.

CSS:

```css
.hero-inner {
  display: grid;
  grid-template-columns: 1.2fr 1fr;
  gap: 32px;
}
.hero-img {
  width: 100%;
  border-radius: 16px;
}
```

- `.hero-inner`: Lays out hero text and image side by side.
- `.hero-img`: Makes the image responsive and rounded.

---

3. Platform (Security & Trust) Cards

HTML:

```html
<div class="platform-features-cards">
  <div class="platform-feature-card">
    <div class="platform-feature-icon">[SVG]</div>
    <div class="platform-feature-title">SOC 2 Type II</div>
    <div class="platform-feature-desc">Industry-standard security compliance.</div>
  </div>
  <!-- More cards... -->
</div>
```

- `.platform-features-cards`: Grid of trust cards.
- `.platform-feature-card`: Each card with icon, title, and description.

CSS:

```css
.platform-features-cards {
  display: grid;
  grid-template-columns: repeat(4, minmax(0, 1fr));
  gap: 32px;
}
.platform-feature-card {
  background: #fff;
  border-radius: 16px;
  box-shadow: var(--shadow);
  text-align: center;
}
```

- `.platform-features-cards`: Lays out cards in a grid.
- `.platform-feature-card`: Card styling for trust features.

---

4. Pricing Cards

HTML:

```html
<div class="pricing-grid">
  <div class="pricing-card">
    <h3>Starter</h3>
    <div class="price">$29/mo</div>
    <ul class="plan-features">
      <li>2 vCPUs, 4GB RAM</li>
    </ul>
  </div>
  <!-- More cards... -->
</div>
```

- `.pricing-grid`: Grid of pricing cards.
- `.pricing-card`: Each plan's card.

CSS:

```css
.pricing-grid {
  display: grid;
  grid-template-columns: repeat(3, minmax(0, 1fr));
  gap: 24px;
}
.pricing-card {
  background: #fff;
  border-radius: 12px;
  text-align: center;
}
```

- `.pricing-grid`: Lays out pricing cards in a row.
- `.pricing-card`: Card styling for each plan.

---

5. Testimonials

HTML:

```html
<div class="testimonials">
  <figure class="testimonial">
    <div class="avatar-placeholder">☁</div>
    <blockquote>CloudNova reduced our costs by 60%.</blockquote>
    <figcaption><strong>Sarah Chen</strong> · CTO, TechFlow</figcaption>
  </figure>
</div>
```

- `.testimonials`: Grid of testimonial cards.
- `.testimonial`: Each testimonial card.

CSS:

```css
.testimonials {
  display: grid;
  grid-template-columns: repeat(3, minmax(0, 1fr));
  gap: 24px;
}
.testimonial {
  background: #fff;
  border-radius: 12px;
  box-shadow: var(--shadow);
}
```

- `.testimonials`: Lays out testimonials in a grid.
- `.testimonial`: Card styling for each testimonial.

---

6. Contact Form

HTML:

```html
<form class="contact-form">
  <div class="form-field">
    <label for="name">Name</label>
    <input id="name" name="name" type="text" required />
  </div>
  <!-- More fields... -->
</form>
```

- `.contact-form`: Grid layout for the form.
- `.form-field`: Each input field and label.

CSS:

```css
.contact-form {
  display: grid;
  grid-template-columns: repeat(2, minmax(0, 1fr));
  gap: 12px;
}
.form-field {
  display: grid;
  gap: 6px;
}
```

- `.contact-form`: Lays out form fields in two columns.
- `.form-field`: Stacks label and input for clarity.
