Project Overview

This project contains two static website skeletons:

- `Agriculture_Consulting/`
- `AI_Consulting/`

Each site includes: `index.html`, `about/about.html`, `services/services.html`, `contact/contact.html`, and a shared `style.css` per site. Pages are lightweight, mobile-friendly, and designed for easy customization.

Site Structure and What to Edit

Agriculture_Consulting

- `index.html`: Home page with hero, CTAs, services preview cards, and testimonials. Includes basic SEO meta tags.
- `about/about.html`: Bio ("Who Am I"), FAQ, and Credentials. Add certifications, partner logos, and photos.
- `services/services.html`: Three service cards with bullet points and a CTA to contact.
- `contact/contact.html`: Contact form (name, email, phone, message) with a hidden honeypot to reduce spam. Action is a placeholder.
- `style.css`: Global styles for hero, buttons, cards, testimonials, sections, and responsive grid.

Customization tips - Update copy in each HTML file. Keep heading order logical (H1 → H2/H3) for accessibility/SEO. - Replace `<title>` and `<meta name="description">` per page. - Add images in an `assets/` folder and reference with `<img src="assets/…" alt="…">`. - Update footer text and year as needed.

AI_Consulting

- `index.html`: Dark hero with CTAs, high-impact use cases, testimonials, and an inline contact form (with honeypot). Basic SEO meta tags.
- `about/about.html`: "Who We Are," FAQ (regulated industries, engagement model), and Credentials.
- `services/services.html`: Service cards (Strategy, RAG Assistant, Forecasting/Automation) and CTA to contact.
- `contact/contact.html`: Contact form (name, email, message) with honeypot. Action is a placeholder.
- `style.css`: Styles for hero variants, cards, testimonials, and responsive grid.

Customization tips - Edit text to match offerings and industry verticals. - Duplicate `article.card` blocks to add more use cases or services. - Add secu-

rity/compliance notes (SOC2/HIPAA/ISO) if relevant. - Update `<title>` and `<meta name="description">` on the home page (and add OG image if desired).

Making the Contact Forms Work

The forms are static by default. Choose one of these wiring options:

- Formspree/Getform: Set `<form action="https://formspree.io/f/yourid" method="POST">` and follow provider docs. Add a redirect with a hidden `_redirect` field if desired.
- Netlify Forms: Host on Netlify and add `name="contact"` and `netlify` to the `<form>` element; optionally add `netlify-honeypot`.
- Serverless email: Use AWS Lambda + SES, Vercel Functions, or Netlify Functions. Point the form `action` to your function endpoint.
- CRM embed: Replace the `<form>` markup with your CRM's embed (HubSpot, Zoho, etc.) for direct lead capture.

Keep the honeypot input present but visually hidden—it helps reduce bot submissions.

Hosting Options (from simplest to most control)

1) GitHub Pages (simple, free)

1. Create a new GitHub repo and push this folder.
2. In repo Settings → Pages, select the branch and root (or `/docs`).
3. Your site will be live at `https://<username>.github.io/<repo>/`.
4. Optional: Add a custom domain and enable HTTPS in Pages settings.

2) Netlify (drag-and-drop or Git connect)

1. Create a Netlify account.
2. Drag the `Agriculture_Consulting/` or `AI_Consulting/` folder into Netlify, or connect the Git repo.
3. Netlify auto-deploys and provides a URL + free SSL.
4. Use Netlify Forms if you want native form handling.

3) Vercel (Git connect, previews)

1. Import the repo into Vercel.
2. Create two projects, setting the project root to each site folder.
3. Add a custom domain and SSL in Vercel.

4) Website Builders (Wix, Squarespace, GoDaddy, Canva)

Two approaches: - Rebuild with builder blocks: Create pages (Home, About, Services, Contact), use hero/cards/testimonials/form blocks, and paste the copy from these HTML files. This yields easier non-technical editing. - Embed custom HTML: Use the builder's HTML/Embed widget to insert code snippets from these files. Prefer the builder's native form blocks for easy submissions and spam protection.

5) AWS S3 + CloudFront (static, fast, low cost)

1. Create an S3 bucket and enable static website hosting.
2. Upload the site files (`index.html` must be at the root of the bucket for that site).
3. Create a CloudFront distribution pointing to the S3 website endpoint; set default root object to `index.html`.
4. Add a custom domain with an ACM certificate (us-east-1) and update DNS to point to CloudFront.

6) AWS EC2 with Nginx (full control)

1. Launch an EC2 instance (Amazon Linux/Ubuntu). Open ports 80/443.
2. Install Nginx: `sudo yum install -y nginx && sudo systemctl enable --now nginx` (Amazon Linux) or `sudo apt update && sudo apt install -y nginx` (Ubuntu).
3. Upload site files to, e.g., `/var/www/agriculture_consulting` and `/var/www/ai_consulting`.
4. Configure server blocks to serve each domain with `root` and `index index.html;`.
5. Test and reload: `sudo nginx -t && sudo systemctl reload nginx`.
6. Add SSL via Let's Encrypt (`certbot --nginx`). Point DNS A/ALIAS to your EC2 IP (use an Elastic IP).

7) cPanel/Shared Hosting

1. Upload the site folder into `public_html` (or an addon domain directory) via File Manager or SFTP.
2. Ensure `index.html` is present at the document root.
3. Enable SSL (AutoSSL or Let's Encrypt from your host).

Domains, SSL, and SEO Checklist

- Domains: Point DNS to your hosting (A/ALIAS for servers, CNAME for managed hosts).
- SSL: Use host-provided HTTPS (Netlify/Vercel/GitHub Pages) or Let's Encrypt for Nginx/cPanel.

- SEO:
  - Unique `<title>` and `<meta name="description">` per page.
  - One H1 per page; use H2/H3 for sections.
  - Add Open Graph tags and an OG image to the home page.
  - Add `robots.txt` and optionally `sitemap.xml` if you add many pages.
- Analytics: Paste GA4 or privacy-friendly analytics (e.g., Plausible) into `<head>` of your pages.

Editing Workflow

- Open files in a code editor (VS Code) and edit copy/styles in place.
- Preview locally: double-click `index.html` or run a simple server (`python -m http.server`).
- Keep navigation links updated if you move files.
- Deploy using your chosen host (drag-and-drop to Netlify, `git push` for GitHub Pages, S3 upload, etc.).
