Zephortech — Project Architecture & Technical Document

Prepared by: Muhammad Junaid
Project: Zephortech Corporate Website (Professional, CMS-driven, SEO-optimized)
Version: 1.0
Date: 2025-11-12

---

1. Executive Summary

Build a modern, maintainable, deployable corporate website for Zephortech using Next.js (App Router) frontend + Strapi CMS backend with PostgreSQL, CI/CD, and production-ready architecture. The system will support SEO, CMS-driven content, blogs, portfolio, contact forms, and analytics.

---

2. High-Level Architecture

Components:

1. **Client (Browser/Mobile)**: Next.js 14+ with TypeScript, Tailwind CSS, responsive design, SEO-optimized, SSR/ISR.
2. **Headless CMS (Strapi v5)**: REST & GraphQL endpoints, media library, role-based admin users.
3. **Database (PostgreSQL)**: Stores CMS content.
4. **Media/CDN**: Cloudinary/Imgix for optimized images.
5. **Email/Forms**: SendGrid/Mailgun for transactional emails.
6. **CI/CD**: GitHub Actions for linting, testing, and deployments.
7. **Monitoring**: Sentry for error tracking; UptimeRobot for availability.
8. **Security**: HTTPS, CSP, input sanitization, WAF, secret management.

Flow: Browser → Next.js → Strapi → PostgreSQL → Cloudinary → Email/Analytics → User

---

3. Technology Stack

- **Frontend**: Next.js 14+ (App Router) + TypeScript + Tailwind CSS
- **Backend/CMS**: Strapi v5 (self-hosted recommended)
- **Database**: PostgreSQL (production), SQLite (dev)
- **Image Hosting/CDN**: Cloudinary/Imgix
- **Hosting/Deployment**:
- Frontend: Vercel
- Backend/CMS: Render/DigitalOcean
- Domain/Email: Hostinger

- **Auth**: Strapi built-in; NextAuth.js if additional auth needed
- **CI/CD**: GitHub Actions
- **Monitoring**: Sentry, UptimeRobot

---

4. Repository & Folder Structure (Monorepo Recommended)

```
/apps/web/           # Next.js frontend
  app/               # App router pages
  components/        # UI components
  lib/               # utils, API clients, types
  styles/            # Tailwind & design tokens
/apps/cms/           # Strapi local config (dev)
/packages/ui/        # Shared React components
/packages/types/     # Shared TypeScript types
/infra/              # Docker-compose, k8s manifests (optional)
.github/workflows/   # CI/CD pipelines
README.md
package.json
```

---

5. Local Development

1. `pnpm install` at root
2. `pnpm dev` → starts frontend & CMS (Docker optional)
3. `.env.local` for secrets
4. `pnpm test`, `pnpm lint` for quality checks

---

6. CMS Schema (Strapi)

- **Service**: title, slug, summary, body, features, icon, heroImage, SEO meta
- **Portfolio**: title, client, industry, summary, problem, solution, techStack, images, outcome metrics
- **BlogPost**: title, slug, author, category, tags, content, excerpt, featuredImage, publishedAt
- **TeamMember**: name, role, bio, photo, social links
- **Testimonial**: clientName, role, company, quote, logo
- **Technology**: name, logo, category, description
- **ContactSubmission**: name, email, company, message, serviceInterest, budgetRange
- **SiteSettings**: siteTitle, defaultMeta, socialLinks, footerText, contactEmail

---

7. API Design & Caching

- `GET /services`, `GET /services/:slug`
- `GET /portfolio`, `GET /portfolio/:slug`
- `GET /blog?limit=10&page=1`
- `POST /contact` → form submissions with spam protection
- Next.js ISR for caching; Edge caching via Vercel + Cloudflare CDN

---

8. CI/CD & Deployment

- GitHub Actions for lint, test, build
- Vercel auto-deploy frontend
- Render/DigitalOcean deploy Strapi + Postgres
- Hostinger DNS to Vercel/Render
- SSL via hosting providers

---

9. Security

- HTTPS everywhere
- Input sanitization & rate limiting
- CSP headers, WAF (Cloudflare)
- Admin subdomain protected
- Secret rotation & DB backups

---

10. Testing & Quality

- Unit: Jest + React Testing Library
- Integration: mock CMS or dev instance
- E2E: Playwright
- Accessibility: axe-core + Lighthouse CI
- Linting & formatting: ESLint + Prettier + Husky
- TypeScript strict mode

---

11. Monitoring & Analytics

- Sentry: runtime errors
- UptimeRobot: availability
- Google Analytics 4 + GTM
- Search Console for indexing

---

12. Deployment Checklist

1. Deploy frontend → Vercel
2. Deploy CMS → Render/DigitalOcean
3. Configure DNS → Hostinger
4. Setup email → Hostinger or SendGrid
5. Verify SSL, analytics, sitemap
6. QA & performance tests

---

13. References

- Fronx Solutions
- ExtraSol