

Product Requirement Document: Codalite Solutions Website

1. Introduction

This Product Requirement Document (PRD) outlines the vision, scope, features, and technical considerations for the Codalite Solutions website. This document serves as a guiding blueprint for the development team, ensuring a shared understanding of the project goals, user needs, and technical implementation details. Our aim is to create a robust, scalable, secure, and user-friendly platform that effectively showcases Codalite Solutions as a leading IT consultancy, software development, and training provider.

2. Vision and Goals

2.1 Vision Statement

To establish Codalite Solutions as the go-to partner for businesses seeking innovative IT solutions and individuals aspiring to build successful careers in technology, through a highly engaging and informative online presence.

2.2 Business Goals

- **Generate Leads:** Attract and convert potential clients for software development and IT consultancy services.
- **Increase Course Enrollments:** Drive registrations for both online and offline IT training programs.
- **Establish Thought Leadership:** Position Codalite Solutions as an authority in the IT industry through valuable content.
- **Enhance Brand Visibility:** Improve online presence and brand recognition.
- **Streamline Operations:** Provide tools for efficient management of client inquiries and training registrations.
- **Attract Talent:** Showcase company culture and career opportunities to attract skilled professionals.

3. Target Audience

The website will cater to two primary user groups:

- **Prospective Business Clients:**
 - Small to Medium-sized Enterprises (SMEs).
 - Decision-makers (CEOs, CTOs, IT Managers) looking for solutions to business problems, digital transformation, or custom software development.
 - Individuals seeking IT consultancy for strategic guidance.

- **Focus:** For now, the primary focus is on small to medium-sized businesses.
- **Prospective Students/Learners:**
 - Individuals from age 8 and above who can read and write, interested in acquiring or enhancing IT skills.
 - Those seeking structured, high-quality online or offline training programs.
 - **Focus:** For now, the primary focus is on individuals.

4. Branding and Design Guidelines

4.1 Logo

The primary logo for Codalite Solutions is the "C" with the "Codalite" wordmark, as provided (specifically `received_1708532489751492~2.jpeg`). It embodies a modern, tech-forward, and professional aesthetic.

4.2 Color Palette

The chosen color palette is derived from the primary logo, emphasizing a vibrant and professional feel.

- **Primary Purple:** #9933FF (Electric Purple) - Used for primary calls-to-action, key headings, and branding elements.
- **Accent Cyan/Aqua:** #00FFFF (Aqua/Cyan) - Used for highlights, interactive elements, and subtle design accents.
- **Dark Neutral:** #1A1A1A (Very Dark Grey) - Used for backgrounds, main text, and structural elements.
- **Light Neutral:** #F5F5F5 (White Smoke) - Used for contrasting text, backgrounds in lighter sections, and interactive element states.

4.3 Typography

- **Primary Font:** Inter (or a similar modern sans-serif for readability and a professional feel).
- **Headings:** Bold and impactful.
- **Body Text:** Clear, legible, and appropriately sized for various screen sizes.

4.4 Imagery

High-quality, professional imagery related to technology, teamwork, innovation, and learning will be used. Avoid generic stock photos where possible; prefer custom graphics or photos that reflect Codalite's unique brand.

5. Core Features and Functionality

5.1 Homepage

- **Hero Section:** Catchy headline, clear value proposition, primary call-to-action (e.g., "Request a Consultation," "Explore Courses").
- **Services Overview:** Brief sections highlighting IT Consultancy, Software Development, and Training.
- **Key Differentiators:** What makes Codalite unique? (e.g., expert team, innovative solutions, tailored training).
- **Client Testimonials/Logos:** Social proof.
- **Call-to-Action:** Prominent buttons leading to service pages, course catalog, or contact form.

5.2 Services Section

- **IT Consultancy:**
 - Detailed explanation of services (e.g., IT Strategy, Cloud Consulting, Cybersecurity Advisory).
 - Case studies/success stories.
 - "Request a Quote" or "Book a Consultation" form.
- **Software Development:**
 - Detailed explanation of services (e.g., Custom Web Apps, Mobile Apps, Enterprise Software, SaaS Development).
 - Portfolio/Project Showcase with descriptions, technologies used, and client benefits.
 - "Start Your Project" inquiry form.

5.3 Training Section

- **Course Catalog:**
 - Browseable list of all courses (online and offline).
 - Filters by category (e.g., Web Development, Data Science, Cloud, Cybersecurity), difficulty, and format (online/offline).
 - Search functionality.
- **Course Detail Pages:**
 - Comprehensive course description, learning objectives, curriculum outline.
 - Instructor profiles.
 - Prerequisites, duration, format (online/offline, live/self-paced).
 - Pricing and enrollment options.
 - Student testimonials.
 - "Enroll Now" button with payment integration.
- **Online Learning Platform (for enrolled students):**
 - User dashboard to track progress.
 - Access to course materials (videos, notes, assignments).
 - **Interactive Quizzes:** Quizzes to test understanding.
 - Discussion forums/Q&A.
 - Assignment submission and grading (if applicable).
 - **Progress Tracking:** Ability for students to track their completion and performance.

- **Certificate Generation:** Automated generation of certificates upon course completion.
- Live video streaming will be considered for a later phase after initial features are implemented.
- **Offline Class Information:**
 - Schedule and location details.
 - Registration process for in-person classes.

5.4 About Us

- **Our Story/Mission:** Codalite's journey and core values.
- **Team Page:** Profiles of key team members (photos, roles, brief bios).
- **Company Culture:** What it's like to work with/for Codalite.

5.5 Contact Us

- Contact form (Name, Email, Subject, Message).
- Phone number, email address, physical address (if applicable).
- Map integration (if physical office).
- Social media links.

6. "Cool Features" and Enhancements

6.1 Blog / Knowledge Base

- Regularly updated articles, tutorials, industry insights, and company news.
- Categorization and tagging for easy navigation.
- Search functionality.
- Comment section (moderated).
- Subscription option for new posts.

6.2 Client Case Studies / Success Stories

- In-depth narratives of successful client projects, highlighting problems solved, solutions implemented, and measurable results.
- Filterable by industry or service type.

6.3 Interactive "Problem Solver" / Needs Assessment Tool

- A guided questionnaire (e.g., "What's your biggest IT challenge?") that helps potential clients identify their needs and suggests relevant Codalite services.
- Captures user input and generates a lead for the sales team.

6.4 Live Chat / Chatbot Integration

- Provide instant support for common queries.
- Route complex inquiries to human agents during business hours.
- Collect leads outside business hours.

6.5 Resource Library

- Downloadable assets: eBooks, whitepapers, industry reports, checklists.
- Requires email signup for access (lead generation).

6.6 Career Opportunities / Job Board

- List open positions within Codalite Solutions.
- Application submission form.

6.7 Newsletter Subscription

- Simple form to subscribe to company news, updates, and special offers.

7. Technical Architecture and Stack

7.1 Core Technologies

- **Backend Framework:** Python (Django)
 - Chosen for its "batteries-included" approach, robust ORM, admin panel, and strong security features. Ideal for rapid development and scalability.
- **Frontend Interactivity:** HTMX and Alpine.js
 - **HTMX:** Allows for dynamic HTML updates without writing much JavaScript, leveraging Django's templating engine for server-side rendering. Excellent for partial page updates and form submissions.
 - **Alpine.js:** Provides lightweight JavaScript interactivity directly in HTML, perfect for UI components like toggles, tabs, modals, and simple state management without a full SPA framework.
- **Styling:** Tailwind CSS
 - A utility-first CSS framework for rapidly building custom designs. Ensures consistency, responsiveness, and avoids custom CSS bloat.
- **Database:** PostgreSQL
 - A powerful, open-source relational database known for its reliability, feature richness, and performance, making it an excellent choice for Django applications.

7.2 Suggested Additional Technologies & Infrastructure

- **Asynchronous Task Queue:** Celery with Redis
 - **Celery:** Distributed task queue for handling long-running operations (e.g., sending emails, processing video uploads for online courses, generating complex reports) asynchronously, preventing blocking of the main web server.

- **Redis:** Used as a message broker for Celery.
- **Containerization:** Docker
 - Ensures consistent development, testing, and production environments. Simplifies deployment and scaling.
- **Cloud Hosting:** AWS / Google Cloud Platform (GCP) / Azure
 - Provides scalable infrastructure (compute, database, storage, CDN) for high availability and performance.
 - **Specific Services:**
 - **Compute:** AWS EC2 / GCP Compute Engine / Azure VMs (or managed services like AWS Elastic Beanstalk, GCP App Engine for simpler deployments).
 - **Database:** AWS RDS (PostgreSQL) / GCP Cloud SQL (PostgreSQL) / Azure Database for PostgreSQL.
 - **Storage: Local File System (Development)** / Cloudflare R2 (Production) / GCP Cloud Storage (Production) / Azure Blob Storage (Production) for static files (images, videos, course materials). This setup allows for local development without external dependencies, while leveraging Cloudflare R2 or other cloud storage for production scalability and performance.
 - **Content Delivery Network (CDN):** AWS CloudFront / GCP Cloud CDN / Azure CDN for faster content delivery globally. (Note: Cloudflare R2 often integrates with Cloudflare's CDN capabilities, which should be leveraged when using R2).
- **Payment Gateway Integration:** Stripe / PayPal
 - For secure and reliable processing of online course payments.
- **Email/SMS Notifications:** SendGrid / Twilio
 - Reliable services for transactional emails (e.g., course enrollment confirmations, password resets) and SMS notifications.
- **Web Server:** Nginx
 - Used as a reverse proxy to serve static files and forward dynamic requests to Django's WSGI server (e.g., Gunicorn).
- **WSGI Server:** Gunicorn
 - A Python WSGI HTTP Server for UNIX, used to run Django applications in production.

8. Technical Requirements and Edge Cases

8.1 Performance

- **Page Load Time:** Target sub-2-second load times for critical pages (Homepage, Services, Course Catalog).
- **Image Optimization:** Implement image compression and responsive image loading (`srcset`).
- **Lazy Loading:** Apply lazy loading for images and videos below the fold.
- **Caching:** Implement server-side caching (e.g., Redis caching for Django), browser caching, and CDN caching.

- **Database Query Optimization:** Ensure efficient database queries to minimize response times.

8.2 Scalability

- **Horizontal Scaling:** Design the application to be stateless where possible to allow for easy scaling by adding more web server instances.
- **Database Scalability:** Utilize managed database services that offer read replicas and vertical scaling options.
- **Asynchronous Processing:** Offload heavy tasks to Celery workers.
- **CDN Usage:** Distribute static and media files globally.

8.3 Security

- **Authentication & Authorization:**
 - Secure user registration and login (Django's built-in auth system).
 - Password hashing (PBKDF2, bcrypt).
 - Multi-factor authentication (MFA) as an optional enhancement for admin/instructor accounts.
 - Role-based access control (RBAC) for different user types (admin, instructor, student, client).
- **Data Protection:**
 - **HTTPS Everywhere:** Enforce SSL/TLS for all traffic.
 - **Input Validation:** Strict server-side validation for all user inputs to prevent injection attacks (SQL Injection, XSS).
 - **Sensitive Data Encryption:** Encrypt sensitive data at rest and in transit.
 - **Regular Backups:** Implement automated, encrypted database and file backups.
- **OWASP Top 10 Mitigation:**
 - **Broken Access Control:** Implement strict permission checks on all views.
 - **Cryptographic Failures:** Use strong encryption algorithms and secure key management.
 - **Injection:** Django ORM protects against SQL injection; template escaping for XSS.
 - **Insecure Design:** Conduct threat modeling and security reviews during design.
 - **Security Misconfiguration:** Secure server configurations, remove default credentials, disable unnecessary services.
 - **Vulnerable and Outdated Components:** Regularly update Django, Python, and all third-party libraries.
 - **Identification and Authentication Failures:** Strong password policies, rate limiting login attempts.
 - **Software and Data Integrity Failures:** Validate uploaded files, ensure data integrity checks.
 - **Security Logging and Monitoring:** Implement comprehensive logging and monitoring for suspicious activities.
 - **Server-Side Request Forgery (SSRF):** Validate URLs and restrict outbound requests.

- **CSRF Protection:** Django's built-in CSRF middleware will be enabled.
- **Clickjacking Protection:** Implement `X-Frame-Options` header.
- **Content Security Policy (CSP):** Implement a strict CSP to mitigate XSS and data injection attacks.
- **Regular Security Audits:** Conduct periodic security audits and penetration testing.

8.4 Accessibility (WCAG 2.1 AA Compliance)

- **Semantic HTML:** Use appropriate HTML tags for structure and meaning.
- **Keyboard Navigation:** Ensure all interactive elements are navigable via keyboard.
- **ARIA Attributes:** Use ARIA roles and attributes where necessary for complex widgets.
- **Color Contrast:** Maintain sufficient color contrast ratios for text and interactive elements.
- **Alt Text:** Provide descriptive alt text for all images.
- **Responsive Design:** Ensure the website is fully usable on various devices and screen sizes.

8.5 Search Engine Optimization (SEO)

- **Semantic URLs:** Clean, human-readable URLs.
- **Meta Tags:** Implement dynamic meta titles and descriptions.
- **Structured Data (Schema.org):** Use JSON-LD for services, courses, organization, and contact information.
- **Sitemap.xml & Robots.txt:** Proper configuration for search engine crawling.
- **Fast Load Times:** (As mentioned in Performance).
- **Mobile-First Design:** Ensure excellent mobile experience.

8.6 Cross-Browser and Device Compatibility

- Test across major browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).
- Ensure consistent user experience and functionality.

8.7 Error Handling and Logging

- Implement robust error handling for both frontend (user feedback) and backend (server-side logging).
- Centralized logging system (e.g., ELK stack, Sentry) for monitoring and debugging.
- User-friendly error pages (e.g., 404, 500).

8.8 Data Backup and Recovery

- Automated daily backups of the database and media files.
- Defined recovery point objectives (RPO) and recovery time objectives (RTO).
- Regular testing of backup and recovery procedures.

9. Information Architecture / Sitemap

- **Homepage** (/)
- **Services** (/services/)
 - IT Consultancy (/services/it-consultancy/)
 - Software Development (/services/software-development/)
 - (Potentially other sub-services)
- **Training** (/training/)
 - Course Catalog (/training/courses/)
 - Individual Course Page (/training/courses/<slug>/)
 - Online Learning Platform (/training/my-courses/ - authenticated)
 - Offline Classes (/training/offline-classes/)
- **Portfolio** (/portfolio/)
 - Case Study (/portfolio/<slug>/)
- **About Us** (/about/)
 - Team (/about/team/)
 - Culture (/about/culture/)
- **Blog** (/blog/)
 - Blog Post (/blog/<slug>/)
- **Resources** (/resources/)
- **Careers** (/careers/)
- **Contact Us** (/contact/)
- **Privacy Policy** (/privacy-policy/)
- **Terms of Service** (/terms-of-service/)
- **Login/Register** (/accounts/login/, /accounts/register/)

10. Phased Development / Milestones (Agile Approach)

We will adopt an Agile development methodology, breaking the project into manageable sprints and delivering incremental value.

Phase 1: Minimum Viable Product (MVP) - Core Marketing Site

- **Goal:** Launch a foundational website to establish online presence and capture basic leads.
- **Features:**
 - Homepage
 - Basic Services pages (overview, no detailed case studies yet)
 - Basic Training overview (catalog list, no enrollment/LMS)
 - About Us (basic)
 - Contact Us form
 - Blog (basic functionality, a few initial posts)
- **Technical Focus:** Core Django setup, Tailwind CSS integration, HTMX/Alpine.js for basic interactivity, deployment to cloud.

Phase 2: Enhanced Marketing & Lead Generation

- **Goal:** Improve lead capture and showcase expertise.
- **Features:**
 - Detailed Service pages with initial Case Studies/Portfolio items.
 - Interactive "Problem Solver" tool.
 - Resource Library (with email capture).
 - Newsletter subscription.
 - Live Chat/Chatbot integration.
 - Enhanced SEO.

Phase 3: Online Learning Platform (LMS)

- **Goal:** Enable online course delivery and management.
- **Features:**
 - Full Course Detail Pages.
 - User registration and login for students.
 - Payment Gateway integration (Stripe/PayPal).
 - Student Dashboard.
 - Course content delivery (video, text, files).
 - **Interactive Quizzes.**
 - **Progress Tracking.**
 - Discussion forums.
 - **Certificate Generation.**
 - Live video streaming will be considered for a later phase after these features are implemented.
- **Technical Focus:** Celery for background tasks, robust user management, secure payment processing.

Phase 4: Continuous Improvement & Advanced Features

- **Goal:** Refine existing features and introduce advanced functionalities.
- **Features:**
 - Advanced LMS features (assignments, grading, live sessions).
 - Personalized content recommendations.
 - Career opportunities section.
 - Client portal (for active project clients, future consideration).
 - Integrations with CRM/Marketing Automation tools.

11. Key Performance Indicators (KPIs)

Success will be measured by the following KPIs:

- **Website Traffic:** Unique visitors, page views.
- **Lead Conversion Rate:** Percentage of visitors who complete a contact form or inquiry.

- **Course Enrollment Rate:** Number of course registrations.
- **Bounce Rate:** Percentage of visitors who leave after viewing only one page.
- **Time on Site:** Average duration visitors spend on the website.
- **SEO Rankings:** Position in search engine results for target keywords.
- **User Engagement:** Interactions with interactive tools, blog comments, resource downloads.
- **Customer Satisfaction:** Feedback from clients and students (post-service/course surveys).