

Consultancy Firm Website — Comprehensive Feature Document

In-depth product specification & feature elaboration to attract users and stakeholders

Prepared by: VOIDSERVICE

Date: September 8, 2025

Executive Summary

This document provides a detailed, user-focused, and technically-informed description of the Consultancy Firm Website product.

The platform is designed to streamline candidate registration, paid activation, job application, medical testing workflows, and multi-party communication between candidates, administrative staff, and medical laboratories.

The goal is to present a clear, compelling narrative of value for prospective users and stakeholders, highlighting the system's capabilities, benefits, and implementation considerations.

Table of Contents

- 1. Product Overview & Value Proposition
- 2. User Journeys (Candidate, Admin, Medical Lab)
- 3. Core Functional Modules (Registration, Jobs, Medical Tests)
- 4. Communication & Chat Support System
- 5. Payments & Razorpay Integration
- 6. SEO & CMS (Add Pages from Dashboard)
- 7. Analytics & Reporting
- 8. Admin Portals & Roles (4 Portals)
- 9. QR Code Verification & Medical Reports
- 10. Technical Architecture & Security
- 11. Deployment Recommendations & Scalability
- 12. Appendix: Sample APIs & Acceptance Criteria

1. Product Overview & Value Proposition

This Consultancy Firm platform offers an end-to-end digital workflow for career-focused recruitment assistance. Key value elements:

- Seamless candidate onboarding with paid activation to ensure verified leads.
- Centralized job-posting, applicant tracking, and medical test coordination.
- Transparent, auditable communication across candidates, admins, and medical labs.
- Actionable analytics and SEO tools to drive organic traffic and measure conversion.

2. User Journeys

2.1 Candidate (User) Journey

A candidate visits the responsive front-end and registers with basic details. They pay the registration fee using the integrated Razorpay gateway to activate the account.

Once active, the candidate can browse jobs, apply, and follow each application's progress in their dashboard. When an application moves to medical verification, candidates receive notifications and can communicate using the in-app chat system. Completed medical results are viewable and downloadable from the profile, and a QR code provides quick verification.

2.2 Admin (Client) Journey

Admins (client representatives) can create and manage job postings, view applicants, and assign selected candidates to medical labs for required tests. Admins communicate with candidates and medical labs through the communication module and can access analytics for recruitment performance and conversion metrics.

2.3 Medical Lab Journey

Medical labs have dedicated portals to receive assigned candidates and medical test requests. Lab personnel manage sample collection, upload test results, update statuses, and communicate with admins and candidates. Each medical action is logged and reflected on the candidate dashboard in real time.

3. Core Functional Modules

3.1 Registration & Activation

Description:

Candidates register with email/phone and minimal KYC fields. Activation is gated by a registration fee to ensure lead quality.

Key UI Elements:

- Registration form (name, email, phone, DOB, country, ID upload)
- Razorpay payment modal and transaction status flow
- Activation email + SMS confirmation

Admin Controls:

- Configure registration fee amount, currency, tax settings
- View pending activations, manual activation/reversal

Data & Compliance:

- Store only essential KYC fields; encrypt sensitive files.
- Maintain transaction logs and invoices for audits.

Acceptance Criteria:

- User can register, pay via Razorpay, and be activated automatically on successful payment.
- Transaction receipt emailed; admin sees transaction in dashboard.

3.2 Job Listings & Applications

Description:

End-to-end job lifecycle: create job, publish, receive applications, review & shortlist.

Key UI Elements:

- Job listing cards with filters (location, category, salary, branch)
- Application modal with resume upload, cover note, and custom questionnaire
- Application timeline & status tracking (Applied → Shortlisted → Assigned → Medical → Completed)

Admin Controls:

- Create/edit/delete jobs, schedule openings per branch, import/export job lists
- Bulk actions for applicant status updates

Acceptance Criteria:

- Candidates can apply, view status, and receive notifications on status changes.
- Admins can assign candidates to a medical lab from the applicant list.

3.3 Medical Test Integration

Description:

Automatically trigger medical test workflow once an applicant reaches the assigned stage. Admin assigns tests and laboratories from the Super Admin/Client Admin interface.

Key UI Elements:

- Medical test templates (blood test, ECG, X-Ray, custom tests)
- Assign tests to a candidate with scheduled dates, lab location, and responsible technician
- Medical portal dashboard to view assigned tests and candidate history
- QR-code generation for each assigned test to allow quick scan-based status checks

Admin Controls:

- Create & manage medical test templates, pricing, and lab availability
- Assign medical personnel and set test turnaround times

Data & Compliance:

- Medical results are sensitive — ensure encryption at rest and strict role-based access
- Audit trail for who accessed/updated the medical record and when

Acceptance Criteria:

- Medical lab receives assignment, updates status, and candidate sees results in dashboard with QR verification.

VOIDSERVICE

4. Communication & Chat Support System

Overview:

A multi-channel chat system is embedded into the platform providing real-time, contextual communication between candidates, admin users, and medical lab staff. The chat system is designed to be simple for users and powerful for admins, including attachments, status indicators, and searchable history.

Key Capabilities:

- Channels: Candidate ↔ Admin, Candidate ↔ Medical Lab, Admin ↔ Medical Lab
- Message types: text, images, PDF reports, structured messages (test results)
- Delivery features: read receipts, typing indicators, push notifications (mobile/web)
- Offline handling: messages stored and delivered when parties come online
- Moderation & canned responses: admins can use quick replies and pre-approved response templates

Implementation Options:

- WebSocket-based real-time channel (Socket.IO / Django Channels)
- Fallback to long-polling for legacy environments
- Store messages in a secure database with references to file storage (S3 or equivalent)

Privacy & Compliance:

- Secure attachments by signed URLs and access-limited endpoints
- Retain chat logs per retention policy configurable from Super Admin

Acceptance Criteria:

- All chat channels are functional, messages persist, and attachments are downloadable by authorized parties.

Featured Module: Communication Block

A highlighted block within the product emphasizes the platform's communication strength — direct channels, escalation flows, and admin moderation tools.

Core pieces:

- Unified inbox for admins to view all chats and filter by candidate, job, or lab.
- Escalation flags for urgent medical updates or payment issues.
- Exportable chat transcripts per candidate for audits and support.

5. Payments & Razorpay Integration

Overview:

Razorpay is integrated to manage registration fees, optional payment for premium services, refunds, and invoices. The integration supports both client-side checkout and server-side verification via webhooks.

Features:

- Razorpay Checkout (modal) for smooth in-page payments.
- Webhook listeners to verify payment status, update activation state, and generate invoices.
- Support for refunds, partial refunds, and transaction reversals under admin control.
- Transaction history, downloadable receipts/ invoices for users and admin.

Implementation Notes:

- Use server-side verification and capture modes to prevent tampering.
- Store only minimal payment metadata (transaction id, status, amount, user id) and avoid storing full card data.

Acceptance Criteria:

- Successful payments activate accounts automatically; failed payments return meaningful errors to users; admins can view and manage transactions.

6. SEO Controls & CMS (Add Pages from Dashboard)

Overview:

Non-technical admins can create landing pages, content pages, and blog posts directly in the dashboard using a WYSIWYG editor. Each page includes robust SEO controls to maximize organic visibility.

SEO Controls:

- Meta title & description per page (with character counters)
- Canonical URL and Robots meta tag controls
- Open Graph tags and Twitter Card settings for rich social previews
- Schema.org structured data insertion (JSON-LD) for job posting and organization schema
- Auto-generated sitemap.xml with a “Submit to Google” helper and manual sitemap regeneration

CMS Features:

- Page templates and drag-and-drop content blocks (hero, features, testimonials, CTA)
- Draft / Publish workflow and revision history
- Page-level access controls (public, members-only)
- Media library with automatic image optimization and alt-text prompt

Acceptance Criteria:

- Admins can create, preview, SEO-optimize, and publish pages without developer support. Search engines can index published pages and sitemap updates are generated automatically.

7. Analytics & Reporting

Overview:

Built-in analytics provides recruitment-specific KPIs and campaign insights. Data can be viewed on dashboards or exported for deeper analysis.

Key Dashboards:

- Overview: Registrations, Activations, Active Applicants, Applications per Job
- Conversion Funnel: Visitor → Registration → Activation → Application → Medical Completion
- Payment Insights: Total revenue, Refunds, Payment failures
- Medical Lab Performance: Avg turnaround time, pass/fail rates, sample backlog
- Communication Metrics: Average response time, unresolved chats

Integrations & Exports:

- Google Analytics (GA4) integration for traffic and event correlation
- CSV / Excel export, scheduled PDF reports, and API endpoints for BI tools

Acceptance Criteria:

- Admins can view real-time KPIs, filter by date/branch/lab, and export reports. Alerts can be configured for drop in conversion rate or payment failures.

8. Admin Portals & Roles (4 Portals)

The system provides four distinct portals, each tailored to a role with specific permissions and UX:

1) Super Admin:

- Full system control, global settings, user & role management, billing control, site-wide content, and audit logs.

2) Client Admin (Recruiter/Admin):

- Manage job postings, review applicants, assign labs, view analytics for their branch, and moderate chat.

3) Medical Lab Portal:

- Receive assignments, manage appointments, update medical results, and generate reports with QR verification.

4) Candidate (User) Portal:

- Apply for jobs, pay registration fees, view application timelines, chat, download medical reports, and manage profile details.

Role & Permission Highlights:

- RBAC (Role-Based Access Control) with granular permissions for create/read/update/delete per entity.
- Hierarchical management so Super Admin can delegate admin-level capabilities to client admins and lab managers.

Acceptance Criteria:

- Users only see actions and data appropriate to their role; admins can impersonate users for support (with logging).

VOIDSERVICE

9. QR Code Verification & Medical Reports

Overview:

Every completed medical report can be associated with a unique QR code that instantly resolves to a secure, read-only status page for that report. QR codes provide fast verification for employers and third parties without exposing sensitive data.

Features:

- QR generation per report, with short-lived signed tokens for public view
- Read-only verification page showing minimal required fields (name, test type, status, verification time)
- Downloadable PDF lab reports for authorized users

Security:

- Public QR views only show non-sensitive metadata; full report access requires authentication and RBAC checks

Acceptance Criteria:

- Scanning QR shows a concise, tamper-evident verification page; admins can revoke QR access if needed.

VOIDSERVICE

10. Technical Architecture & Security

Recommended Stack (example):

- Backend: Django + Django REST Framework OR Node.js + Express + TypeScript
- Database: PostgreSQL (encrypted at rest) with Redis for caching and job queues
- Real-time: Django Channels or Socket.IO for chat/websocket features
- Storage: Amazon S3 / compatible object storage for reports and uploads
- Payment: Razorpay Checkout and webhooks
- Containerization: Docker + Kubernetes (optional) for scalability

Security Measures:

- TLS everywhere (HTTPS), HSTS, secure cookies
- Authentication: JWT access tokens with refresh tokens or secure session auth
- RBAC and least privilege principles enforced in backend
- Encryption of PII and medical data at rest; patients' medical files stored encrypted
- Regular backups, database snapshots, and disaster recovery plan

Compliance:

- Provide configurable retention and deletion for medical records (GDPR/PDPA friendly)
- Audit logs for sensitive actions e.g., report access, status updates

Performance & Scalability:

- Horizontal scaling for stateless services, use worker pool for long-running tasks (file processing)
- CDN for static assets and image optimization

VOIDSERVICE

11. Deployment Recommendations & Scalability

Highlights:

- Use environment-specific config (staging/production) and infrastructure-as-code for repeatable deployment.
- Monitor with Prometheus + Grafana or hosted alternatives; log aggregation via ELK/Graylog.
- Autoscaling for web nodes, workers, and database read replicas for heavy traffic.

Operational Practices:

- Daily/weekly backups and verification, incident runbooks, and escalation matrix.
- Regular penetration testing and dependency vulnerability scanning.

VOIDSERVICE

12. Appendix: Sample APIs & Acceptance Criteria

Below are short samples and criteria to help developers integrate and test features quickly.

Sample Endpoints (conceptual):

- POST /api/v1/auth/register -> {name,email,phone} -> returns user_id, next steps to pay
- POST /api/v1/payments/razorpay/create -> server returns order_id -> client opens Razorpay Checkout
- POST /api/v1/payments/razorpay/webhook -> verify signature -> mark user active
- POST /api/v1/jobs/{id}/apply -> uploads resume, answers -> creates application record
- POST /api/v1/admin/assign-medical -> {app_id, lab_id, tests[]}
- GET /api/v1/medical/{report_id}/qr -> returns short URL to verification page

Acceptance & QA Checklist (sample):

- Secure payments — all successful transactions activate user without manual steps.
- Chat logs retained and exportable; attachments associated with records.
- SEO pages render server-side for crawlers or use prerendering for SPAs.
- Analytics show conversion funnel data and can export CSV/PDF.

For custom visuals (icon-rich brochure, colored layout, or printable flyer), request a 'Visual Brochure' and I will design a modern, attractive version.

OLD SERVICE