

1-Page Skeleton Plan — Consulting Workflow Audit Portal (SaaS Admin Backbone)

Project Goal

Build a **client portal + audit delivery platform** that lets you onboard clients, collect intake data, run audits, deliver findings, track progress, and support retainers/subscriptions. This is both: **Your internal operating system** for consulting delivery

A **portfolio-grade .NET backend** proving real-world SWE skills (auth, multi-tenancy, deployments)

Industry-Standard Tech Stack

Backend

.NET 8 (ASP.NET Core Web API), C#

Entity Framework Core (ORM + migrations)

PostgreSQL (primary DB)

Auth: ASP.NET Core Identity + **JWT + refresh tokens**

Authorization: policy-based RBAC (Owner/Admin/ClientViewer)

Validation: FluentValidation

API Docs: Swagger / OpenAPI

Logging: Serilog

Background jobs: Hangfire (reminders, status nudges, scheduled reports)

Observability: OpenTelemetry + Health Checks

Frontend (simple + fast)

Next.js + TypeScript (or React + Vite)

UI: Tailwind + shadcn/ui

Auth client: JWT handling + refresh flow

DevOps / Delivery

Docker + docker-compose (API + DB + Redis optional)

CI/CD: GitHub Actions

Cloud: Azure (App Service or Container Apps) + Azure Database for PostgreSQL

Secrets: Azure Key Vault

Core Product Scope (MVP)

Tenancy + Users

Organization (Client)

Users belong to an org

Roles: **Owner (you)**, **Admin (you/team)**, **ClientManager**, **ClientViewer**

Workflow Audit Objects

Client Intake

Questionnaire responses (structured JSON + versioning)

Tool inventory (CRM/CMS/Ads/Sheets/etc.)

File uploads (screenshots, exports)

Audit

Status: Draft → In Review → Delivered → In Progress → Closed

Audit date, auditor, notes

Findings

Category (Automation, Data, Marketing, Security, Ops)

Severity (Low/Med/High), Effort (S/M/L), ROI estimate

Recommendation + Next steps

Status tracking per finding

Deliverables

Generated report export (PDF later) + stored version

Activity/Audit Log

Who changed what, when (basic compliance + credibility)

Data Model (minimum entities)

Organization

User (Identity)

OrganizationMember (role + join)

IntakeForm + IntakeResponse

Audit

Finding

Recommendation (or embedded in Finding)

Attachment (uploads)

ActivityLog

API Endpoints (minimum set)

Auth: register/login/refresh/logout, invite user

Orgs: create org, list members, assign roles

Intake: create form template, submit response, list responses

Audits: create/update, status transitions, assign auditor

Findings: CRUD, status updates, filtering/pagination

Attachments: upload/download (scoped to org + audit)

Logs: list activity by org/audit

Build Phases (from scratch)

Phase 0 — Setup (Day 1–2)

Repo + solution structure (Clean Architecture)

Docker compose: Postgres + API

EF Core migrations + Swagger

Base CI pipeline (build + test)

Phase 1 — Secure Multi-Tenant Foundation (Week 1)

Identity + JWT + refresh tokens

RBAC + org scoping (every query must be org-filtered)

Global error handling + validation

Phase 2 — MVP Workflow (Weeks 2–3)

Intake forms + file uploads

Audits + findings + status tracking

Activity log

Basic frontend portal screens

Phase 3 — “Consulting-grade” Polish (Weeks 4–6)

Filters/search/pagination everywhere

Background jobs (reminders, weekly summary emails optional)

Deploy to Azure + HTTPS + Key Vault

Integration tests (at least 10) + seeded demo data

Demo script + screenshots + README

Definition of Done (MVP launch)

You can onboard a client in 5 minutes:

Create org → invite client

Client completes intake

You create audit + findings

Client logs in and sees deliverables + statuses

Deployed to Azure with CI/CD

Logs + health checks working

Clear README + demo walkthrough

Optional Monetization Layer (after MVP)

Stripe subscriptions (retainers)

Plan entitlements (monthly audit, quarterly audit, advisory hours)

Usage limits (number of audits/findings per month)

White-label client branding