

PORFOLIO 2.0

Cahier des charges - Site public + Dashboard Admin

Périmètre: Golden Scope (MVP Pro)

Date: 15 février 2026

Langues: EN / FR / AR (RTL)

Backend: NestJS + PostgreSQL | Frontend: (au choix) Angular/React/Next

Objectif	Un portfolio professionnel, administrable, multilingue, orienté recrutements & clients.
Livrables	Site public + Dashboard admin + API + Base de données + Stockage médias + SEO/OG.
Utilisateurs	Visiteurs (public) / Admin (vous).
Priorités	Clarté des projets (case studies), conversion (contact), crédibilité (témoignages), SEO.

1. Contexte & objectifs

Ce projet vise à transformer votre portfolio en une vitrine professionnelle orientée résultats: présenter vos projets sous forme de "case studies", faciliter le contact client/recruteur, et vous permettre de tout administrer (contenu, médias, CV, témoignages) via un dashboard sécurisé.

2. Portée (Golden Scope)

Le périmètre ci-dessous correspond au MVP Pro: assez complet pour convaincre, mais maîtrisé pour livrer rapidement.

2.1 Fonctionnalités - Côté public

- Page d'accueil: hero + projets en vedette + stack + témoignages + CTA contact.
- Catalogue projets avec filtres: catégorie, tags, vedette, statut.
- Page projet détaillée (case study): problème, rôle, solution, challenges, résultats, galerie, liens (live/GitHub).
- Page CV: sections dynamiques + téléchargement PDF + génération d'un CV 1 page (PDF).
- Témoignages: afficher uniquement les avis approuvés, badge "Vérifié".
- Contact: formulaire (catégorie, budget optionnel, deadline optionnelle) + confirmation.
- SEO: meta, OG, sitemap/robots.
- i18n: EN/FR/AR avec support RTL pour l'arabe + préférence sauvegardée.

2.2 Fonctionnalités - Dashboard Admin

- Authentification admin (JWT, bcrypt) + routes protégées.
- Gestion projets: CRUD + case study + tags + médias + vedette + statut + état (brouillon/publié/archivé) + prévisualisation.
- Gestion CV: informations profil + expériences/éducation/skills/certifs/langues (CRUD + réordonnement) + upload/remplacement PDF.
- Génération PDF "One-page CV" depuis les données (modèle propre).
- Témoignages: liste (pending/approved/rejected), validation, correction, badge vérifié, association optionnelle à un projet.
- Messages contact: inbox + read/unread + statuts (new/in progress/replied/closed) + notes internes.
- Mini-CRM leads: conversion message -> lead, source, valeur estimée, date de relance.
- Recherche globale admin: projets, CV, témoignages, messages, leads.

3. Personas & parcours

Les parcours clés sont optimisés pour la conversion (contact) et la crédibilité (case studies, avis).

3.1 Visiteur (recruteur / client)

- Arrive sur la home, comprend votre valeur en 10 secondes (headline + preuves).
- Filtre et explore 2-3 projets pertinents; lit la section "Résultats".
- Télécharge le CV ou envoie un message avec contexte (budget/deadline si freelance).

3.2 Admin (vous)

- Se connecte au dashboard.
- Ajoute ou met à jour un projet (brouillon -> preview -> publié).
- Met à jour le CV (et régénère le one-page).
- Traite les messages (statuts + notes) et crée un lead si opportunité.

4. Exigences non fonctionnelles (qualité)

Critères indispensables pour un rendu professionnel et une base solide.

- Performance: pages rapides (optimisation images, cache, lazy-load).
- Accessibilité: navigation clavier, contrastes, labels formulaires.
- Sécurité: bcrypt, JWT, rate limiting (login + contact), validation des inputs, protection upload.
- Qualité UX: responsive complet, mode sombre/clair (facultatif côté public si souhaité), transitions propres.
- SEO: URLs propres, meta/OG par page, sitemap, canonical.
- Internationalisation: gestion RTL, contenu traduisible par entité/champ.
- Observabilité: logs applicatifs, audit log admin (recommandé).

5. Architecture & stack recommandée

Architecture simple, maintenable, compatible hébergements gratuits.

5.1 Backend

- NestJS (REST) avec modules: Auth, Projects, Upload, CV, Testimonials, Contact, Leads, Search, i18n, SEO.
- PostgreSQL + migrations (Prisma / TypeORM / Drizzle au choix).
- Stockage fichiers: Cloudinary ou Supabase Storage (images + PDF).

5.2 Frontend

- Option A: Next.js (SEO excellent, pages statiques + API).
- Option B: Angular (dashboard très solide) + SSR/SSG si besoin.
- i18n: next-intl / ngx-translate / i18next selon stack.
- UI: Tailwind + composants (shadcn/ui si React).

5.3 Déploiement

- Frontend: Vercel/Netlify (gratuit).
- Backend: Render/Railway (gratuit selon limites).
- DB: Supabase Postgres (free tier).

6. Modèle de données (vue d'ensemble)

Le schéma complet PostgreSQL est livré dans un fichier .sql séparé. Ci-dessous, la structure fonctionnelle principale.

Domaine	Tables principales
Auth	users
Profil	profile, social_links
Projets	projects, project_media, tags, project_tags
CV	experiences, education, skills, certifications, spoken_languages, resume_files
Témoignages	testimonials
Contact/CRM	contact_messages, leads, lead_notes
i18n	translations (entity_type, entity_id, field, lang, value)
Audit (optionnel)	audit_logs

6.1 Règles clés

- Projects: publish_state = draft/published/archived; published_at requis si publié.
- Testimonials: status = pending/approved/rejected; is_verified contrôlé par admin.
- Contact: is_read + status de traitement; conversion optionnelle en lead.
- i18n: traduction par champ (ex: project.title, project.case_problem) pour EN/FR/AR.

7. Endpoints API (résumé)

Liste indicative (REST). Les détails (DTO/validation) seront implémentés côté NestJS.

7.1 Auth

- POST /auth/login
- POST /auth/logout (optionnel côté backend)
- POST /auth/change-password (optionnel)

7.2 Projects

- GET /projects (public, filtres, pagination)
- GET /projects/:slug (public)
- POST /admin/projects
- PATCH /admin/projects/:id
- DELETE /admin/projects/:id
- POST /admin/projects/:id/media
- DELETE /admin/media/:id

7.3 CV

- GET /cv (public)
- GET /cv/pdf (public - CV complet)
- GET /cv/one-page (public - PDF généré)
- PATCH /admin/profile
- CRUD /admin/experiences, /education, /skills, /certifications, /spoken-languages
- POST /admin/resume/pdf (upload)

7.4 Testimonials / Contact / Leads

- POST /testimonials (public submit)
- GET /testimonials (public approved only)
- PATCH /admin/testimonials/:id (approve/reject/verify/link)
- POST /contact (public)
- GET /admin/messages
- PATCH /admin/messages/:id (read/status/notes)
- POST /admin/leads (create / convert from message)

8. Critères d'acceptation

Le projet est considéré comme livré lorsque les critères suivants sont atteints.

- Le site public est responsive, multilingue (EN/FR/AR) et SEO-ready (meta/OG + sitemap).
- Les projets affichent une page "case study" complète et filtrable.
- Le dashboard admin permet CRUD complet + gestion médias + workflow draft/publish avec preview.
- Le CV est administrable et téléchargeable en PDF, avec génération one-page fonctionnelle.
- Les témoignages ont un workflow pending/approved/rejected + badge verified.
- Le contact inbox gère read/unread + statuts et permet de créer des leads.
- Sécurité minimum: JWT + bcrypt + validation + rate limiting sur login/contact.

9. Roadmap proposée (phases)

Découpage recommandé pour livrer vite sans perdre la qualité.

Phase	Livraison	Contenu
1	Base + Projets	Auth, CRUD Projects + case studies, upload images, pages publiques + filtres, i18n base (EN/FR/AR).
2	CV + Témoignages	CV dynamique, upload PDF, one-page generator, workflow témoignages + verified.
3	Contact + CRM + SEO	Inbox messages, statuts, leads, recherche globale, SEO/OG + sitemap.

Annexe: Schéma PostgreSQL complet

Le fichier SQL livré: [portfolio_schema_postgresql.sql](#)