

Cahier des Charges – Application Web d’Inscription en Ligne des Auditeurs

1. Présentation Générale du Projet

1.1 Contexte

Digitaliser le processus d’inscription des auditeurs via une plateforme web sécurisée et centralisée, avec :

- Automatisation des inscriptions, paiements et génération de documents.
- Interface de gestion administrative complète (validation, reporting, communication).
- Conformité RGPD et normes de sécurité des données.

1.2 Objectifs Généraux : Exigences Fonctionnelles

Auditeurs

- Création de compte (email/mot de passe + vérification par OTP).
- Formulaire dynamique avec sélection de formation, téléchargement de documents obligatoires (CNI, attestations), et intégration de passerelle de paiement (carte bancaire/virement).
- Dashboard personnel : historique des inscriptions, statut de paiement, téléchargement d’attestations/factures.

Administrateurs

- Backoffice sécurisé avec :
 - Validation manuelle des documents et inscriptions.
 - Gestion CRUD des formations (dates, capacités, descriptifs).
 - Reporting financier (export CSV/PDF) et suivi des impayés.
 - Notification automatisée par email (plugins SMTP/API).

2. Périmètre Fonctionnel du Projet

2.1 Modules Candidat

- Création/gestion de compte
- Formulaire d’inscription
- Téléversement de pièces (CNI, diplôme, photo, justificatif de paiement)
- Suivi de statut
- Notifications email/SMS

2.2 Modules Administration (Non prioritaire)

- Tableau de bord
- Validation/Rejet/Mise en attente
- Gestion des sessions (dates, quotas, tarifs)
- Exports Excel/PDF
- Messagerie groupée (email/SMS)

2.3 Modules Super Administration

- Gestion des rôles et accès
- Paramétrage (sessions, quotas, pièces exigées)
- Journalisation et rapports d'audit

3. Spécifications Techniques (Spring Boot & Angular)

3.1 Pile Technologique Recommandée (cible)

- Front-end : Angular (v16 ou 17), TypeScript, RxJS. Gestion d'état optionnelle (NgRx).
- Back-end : Java 21, Spring Boot 3.x (Spring Web, Spring Data JPA/Hibernate, Spring Validation, Spring Security).
- Base de données : PostgreSQL 16 (ou 16+).
- Authentification : Spring Security (JWT), possibilité d'OpenID Connect via Keycloak.
- Stockage fichiers : S3 compatible (MinIO) ou local sécurisé selon l'environnement.
- Notifications : Email (SMTP)
- Build/CI : GitHub Actions/GitLab CI).

3.2 Architecture & Intégration

- Architecture n-tiers : Front Angular SPA ↔ API REST Spring Boot ↔ PostgreSQL.
- Séparation des environnements : Dév / Recette / Prod avec variables d'environnement et secrets.
- Versionnement & migrations : Git (branche main/dev), migrations DB via Flyway ou Liquibase.

3.3 Sécurité (conformité RGPD & OWASP)

- HTTPS (TLS) obligatoire
- Politique JWT avec rafraîchissement et révocation
- Validation serveur & client des entrées
- Chiffrement des mots de passe (BCrypt)
- Rôles/permissions par profil (RBAC)
- Protection CSRF/XSS/Headers de sécurité
- Sauvegardes chiffrées et test de restauration
- Journalisation d'audit (accès, validations, export)

4. Architecture & Interface Utilisateur

4.1 Arborescence

- Accueil public
- Espace Candidat
- Espace Administration
- Espace Super Admin

4.2 UI/UX

- Design sobre, conforme à la charte de la Douane (bleu/blanc/doré)
- Responsive mobile-first
- Parcours en étapes (wizard) pour l'inscription

5. Planification Prévisionnelle

Phase 1 : Analyse & Conception (1 semaine)

Phase 2 : Développement (2 semaines)

Phase 3 : Tests

Durée totale estimée : 2 à 3 semaines

6. Équipe Projet

Chef de projet – Pilotage & coordination

Développeur back-end – Spring Boot, JPA, sécurité

Développeur front-end – Angular, UX

Designer – UI/Kit composants

7. Livrables Attendus

- Application web en production (version 1.0).
- Code source (architecture, API).

8. Critères de Validation & Réception

Respect du périmètre fonctionnel ; conformité sécurité & performance ; documentation complète ; UX validée ; disponibilité et sauvegardes conformes.