

Modèle de données (proposé)

Utilisateurs (Django User + Profile):

- `Profile: role (ADMIN, RESP_STAGE, TUTEUR_ECOLE, TUTEUR_ENTREPRISE, ETUDIANT), téléphone, département, etc.`

École & Programmes

- `Departement(name)`
- `Programme(name, departement)`
- `Promotion(annee, programme)`

Entreprises & Contacts

- `Entreprise(name, secteur, adresse, ville, pays, website, notes, niveau_confiance, convention_modele)`
- `ContactEntreprise(entreprise, nom, email, tel, role)`

Stages & Campagnes

- `Campagne(nom, date_debut, date_fin, programme, capacite, description)`
- `OffreStage(entreprise, titre, description, competences, remuneration, ville, type, campagne, nb_places, statut)`
- `Candidature(etudiant, offre, cv, lm, statut, date)`
- `Affectation(etudiant, offre, tuteur_ecole, tuteur_entreprise, date_debut, date_fin, convention_pdf, statut)`

Suivi & Livrables

- `Journal(etudiant, affectation, date, contenu, heures_travailles)`
- `Livrable(affectation, type {rapport, presentation, fiche_eval}, fichier, date_depot, statut)`
- `EvaluationTuteurEntreprise(affectation, competences, assiduite, commentaire, note)`
- `EvaluationTuteurEcole(affectation, livrables, soutenance, commentaire, note)`
- `NoteFinale(affectation, note_entreprise, note_ecole, coef_entreprise, coef_ecole, note_finale_calc)`

Système

- `Alerte(type, cible (user/affectation/livrable), echeance, statut)`
- `AuditLog(user, action, objet, before, after, ip, ua, ts)`

Flux fonctionnels clés (avec automatisations)

1. **Campagne** → publications d'**offres**, quotas.
 2. **Étudiant** : profil → candidatures → pièces jointes.
 3. **Tuteur entreprise** : valide la mission, suit présence → **évalue**.
 4. **Tuteur école** : suit livrables, **note**.
 5. **Rappels automatiques (Celery/beat)**:
 - J-7/J-3/J-1 avant **échéance** rapport.
 - Relances candidatures en attente.
 - Relances évaluations non remplies.
 6. **Reporting**: tableaux de bord (admin), export CSV/XLSX/PDF.
-

Stack & packages (recommandé)

- **Django 5+, PostgreSQL, Redis** (Celery broker), **Celery + celery-beat**
 - **django-jazzmin** (thème admin), **django-import-export**, **django-filter**, **django-crispy-forms** (si forms front), **django-storages** (si S3)
 - **django-environ**, **whitenoise** (si déploiement simple)
 - **drf (optionnel)** si tu veux exposer une API REST (mobile/SPA plus tard)
 - **WeasyPrint ou xhtml2pdf** pour PDF
-

Méthodologie projet (pragmatique et exécutable)

1) Cadrage & recueil des besoins (Semaine 1)

- **Acteurs & rôles**
 - **Admin École** (DSI/Scolarité) : paramétrage global, reporting.
 - **Responsable Stages** : création campagnes, appariements, validations, évaluations finales.
 - **Enseignant/Tuteur École** : suivi étudiant, validation livrables, notes.
 - **Tuteur Entreprise** : validation présence, évaluation missions.
 - **Étudiant** : préférences, candidatures, journaux, livrables, auto-évaluation.
- **Processus clés** (as-is → to-be)
 1. Gestion entreprises & conventions
 2. Campagne de stage (période, quotas, spécialités)
 3. Candidature / affectation / convention
 4. Suivi & livrables (rapport, journal, feuille de présence)
 5. Évaluations (tuteur entr., tuteur école)
 6. Soutenance & note finale
 7. Reporting (taux de placement, satisfaction, retours)
- **KPI**
 - Taux de placement par filière, **délai moyen d'affectation**, livrables reçus dans les temps, **notes moyennes**, **satisfaction** (NPS) entreprises/étudiants.

2) Conception (Semaine 2)

- **Modèle de données** (UML/ERD — voir plus bas)
- **Règles métier** : échéances, statuts, droits d'accès, périodes.
- **Maquettes rapides** (pour l'admin + exports) — on utilisera **Django Admin + thème Jazzmin**.

3) Implémentation (Semaines 3–4)

- **Back-office Django** (admin fortement customisé)
- **Automatisations** : emails, rappels, génération de PDF, imports(exports Excel/CSV).
- **Sécurité & RGPD** : permissions, journaux d'audit.
- **Tests** (unitaires + fonctionnels).

4) Recette & déploiement (Semaine 5)

- Jeux d'essais (fixtures), revue utilisateurs, corrections.
- Déploiement (Docker/Poetry ou pip + Gunicorn + Nginx, Postgres).

5) Évaluation & amélioration continue (Semaine 6)

- Sondages intégrés, suivi KPI, backlog d'évolutions.

Modèle de données (proposé)

Utilisateurs (Django User + Profile):

- `Profile: role (ADMIN, RESP_STAGE, TUTEUR_ECOLE, TUTEUR_ENTREPRISE, ETUDIANT),
téléphone, département, etc.`

École & Programmes

- `Departement(name)`
- `Programme(name, departement)`
- `Promotion(annee, programme)`

Entreprises & Contacts

- `Entreprise(name, secteur, adresse, ville, pays, website, notes,
niveau_confiance, convention_modele)`
- `ContactEntreprise(entreprise, nom, email, tel, role)`

Stages & Campagnes

- `Campagne(nom, date_debut, date_fin, programme, capacite, description)`
- `OffreStage(entreprise, titre, description, competences, remuneration,
ville, type, campagne, nb_places, statut)`

- Candidature(etudiant, offre, cv, lm, statut, date)
- Affectation(etudiant, offre, tuteur_ecole, tuteur_entreprise, date_debut, date_fin, convention_pdf, statut)

Suivi & Livrables

- Journal(etudiant, affectation, date, contenu, heures_travailles)
- Livrable(affectation, type {rapport, presentation, fiche_eval}, fichier, date_depot, statut)
- EvaluationTuteurEntreprise(affectation, competences, assiduite, commentaire, note)
- EvaluationTuteurEcole(affectation, livrables, soutenance, commentaire, note)
- NoteFinale(affectation, note_entreprise, note_ecole, coef_entreprise, coef_ecole, note_finale_calc)

Système

- Alert(e(type, cible (user/affectation/livrable), echeance, statut))
 - AuditLog(user, action, objet, before, after, ip, ua, ts)
-

Flux fonctionnels clés (avec automatisations)

1. **Campagne** → publications d'**offres**, quotas.
2. **Étudiant** : profil → candidatures → pièces jointes.
3. **Tuteur entreprise** : valide la mission, suit présence → **évalue**.
4. **Tuteur école** : suit livrables, **note**.
5. **Rappels automatiques (Celery/beat)**:
 - o J-7/J-3/J-1 avant **échéance** rapport.
 - o Relances candidatures en attente.
 - o Relances évaluations non remplies.
6. **Reporting**: tableaux de bord (admin), export CSV/XLSX/PDF.