

Dossier MERISE complet (rétro-conception) — v2 (recréé)

Projet Django « portfolio » — Portfolio public + Administration + Users/Rôles/Menus

Généré le 06/02/2026 (recréation du fichier v2)

Table des matières (Word : clic droit > Mettre à jour le champ)

1. Présentation et périmètre

Ce dossier présente une rétro-conception MERISE du projet « portfolio » à partir des éléments fournis (code Django + templates). Il couvre : portfolio public, administration du contenu, gestion des utilisateurs, rôles/menus et permissions, contact, et statistiques de consultation (Démo live).

2. Livrables Données

2.1 Dictionnaire de données (structurant)

Les entités métier principales observées : ROLE, USER, MENU, ROLE_MENU_ACCES, CONTACT_MESSAGE, et les entités de contenu du portfolio (réalisations, témoignages, compétences, réseaux, photo, à propos, spécialité).

2.2 MCD — Modèle Conceptuel de Données (texte)

Entités :

- ROLE(id_role, nom_role)
- USER(id_user, nom, email, password_hash)
- MENU(id_menu, nom, code_menu)
- ROLE_MENU_ACCES(id_rma, est_visible)
- CONTACT_MESSAGE(id_contact, nom, email, message, date_envoi)
- PORTFOLIO_* (réalisations, témoignages, compétences, réseaux, photo, à propos, spécialité, etc.)

Associations :

- ROLE (1,n) — USER : un user a un seul rôle ; un rôle peut avoir plusieurs users.
- ROLE (n,n) — MENU via ROLE_MENU_ACCES(est_visible).

2.3 MLD — Modèle Logique (relationnel)

Tables : role, user_app, menu, role_menu_acces, contact_message, et tables portfolio_* avec un attribut est_visible pour la publication.

2.4 MPD — Modèle Physique (DDL indicatif)

```
CREATE TABLE role (
    id_role INTEGER PRIMARY KEY AUTOINCREMENT,
    nom_role TEXT NOT NULL UNIQUE
);
```

```
CREATE TABLE user_app (
    id_user INTEGER PRIMARY KEY AUTOINCREMENT,
```

```

    nom TEXT NOT NULL,
    email TEXT NOT NULL UNIQUE,
    password TEXT NOT NULL,
    id_role INTEGER NOT NULL,
    FOREIGN KEY (id_role) REFERENCES role(id_role)
);

```

```

CREATE TABLE menu (
    id_menu INTEGER PRIMARY KEY AUTOINCREMENT,
    nom TEXT NOT NULL,
    code_menu TEXT NOT NULL UNIQUE
);

```

```

CREATE TABLE role_menu_acces (
    id_rma INTEGER PRIMARY KEY AUTOINCREMENT,
    id_role INTEGER NOT NULL,
    id_menu INTEGER NOT NULL,
    est_visible INTEGER NOT NULL DEFAULT 0,
    FOREIGN KEY (id_role) REFERENCES role(id_role),
    FOREIGN KEY (id_menu) REFERENCES menu(id_menu),
    UNIQUE (id_role, id_menu)
);

```

3. Livrables Traitements

3.1 MCT — Modèle Conceptuel des Traitements

- Inscription / Connexion / Déconnexion (gestion de session).
- Administration du portfolio : CRUD contenu + bascule est_visible.
- Administration centrale : CRUD users + mise à jour matrice rôle×menu.
- Contact : création/consultation des messages.
- Statistiques : clic Démo live → compteur_demo_live++.

3.2 MOT/MLT/MPT (synthèse)

MOT : Visiteur (consultation/inscription/contact), Utilisateur connecté (navigation), Modérateur (gestion contenu), Administrateur (users + droits + contenu).

MLT : pseudo-algorithme : upsert rôle×menu ; filtre est_visible ; incrément atomique compteur.

MPT : vues Django et templates HTML implémentent ces traitements (routes des apps incluses au niveau projet).

4. IHM & navigation

Écrans majeurs : Index public, Inscription, Dashboard administration portfolio, Administration centrale (Utilisateurs/Permissions/Messages), Layout (header/footer) avec menus conditionnels selon session et rôle.

5. Pilotage

Livrables de pilotage (synthèse) :

- WBS : socle, auth, rôles/menus, contenu, contact, stats, tests/sécurité, déploiement.
- RACI : Admin (A/R users+droits), Modérateur (R contenu), Responsable saisie (R saisie), Utilisateur/Visiteur (I).
- Plan de tests : visibilité, droits, unicité email, compteur Démo live, export/import.
- Risques : contrôle de droits uniquement IHM, singletons non garantis DB, config prod (DEBUG/ALLOWED_HOSTS).