Fiche du cours

Titre :

Web330 - Django REST API (focus) & React (intégration)

Description :

Parcours fullstack avec accent fort sur Django/DRF : modélisation, ORM, sérialisation, permissions & auth JWT, filtres/pagination, gestion d'erreurs, tests et documentation (OpenAPI). Côté front, React sert à consommer l'API (routing + formulaires + appels), sans aller trop loin dans l'écosystème React afin de consacrer l'essentiel du temps au backend.

Objectifs :

- Concevoir un schéma de données robuste et optimiser l'ORM (relations, agrégations, N+1).*
- Exposer une API REST propre avec DRF (serializers, viewsets, routers, filtres, pagination).*
- Sécuriser l'API : JWT (SimpleJWT), permissions, throttling, CORS, gestion d'erreurs normalisée.*
- Tester l'API (pytest) et documenter (OpenAPI/Swagger), préparer la config (.env).*
- Côté React : créer une SPA minimale consommant l'API (routing, formulaires, état léger, gestion chargement/erreurs).*
- Livrer un capstone cohérent (API solide + UI fonctionnelle) prêt à déployer.

Chapitres :

1. Architecture & Setup (Django/DRF) : apps, settings, env (.env), CORS, admin pro*
2. Modèles & ORM I : relations, migrations, requêtes, agrégations*
3. Modèles & ORM II : performances (select_related/prefetch_related), N+1, validations modèle*
4. DRF I : serializers (validation), viewsets/routers, conventions de réponse*
5. DRF II : filtres, recherche, pagination, ordering, erreurs (Problem Details*
6. Sécurité API : JWT (access/refresh), permissions, throttling, CORS fin, bonnes pratiques*
7. Qualité backend : tests (pytest & client DRF), OpenAPI/Swagger, structure des settings*
8. React – Fondamentaux pour consommer l'API : Vite, composants, useState/useEffect, appels API, gestion chargement/erreurs*
9. React – Routing & Formulaires : React Router, formulaires contrôlés, validation, intégration CRUD avec l'API*
10. Capstone & Industrialisation : intégration complète (API sécurisée + SPA), variables d'env, build front, README & démo

À la fin :

Vous saurez livrer une API DRF sécurisée, testée et documentée, performante côté ORM, et une SPA React légère qui la consomme (CRUD, formulaires, erreurs/chargements), avec une configuration propre et un livrable portfolio-ready. Vous maîtriserez JWT, pagination/filtres, CORS et une doc OpenAPI/Swagger claire, ainsi que des tests (pytest) essentiels. Enfin, vous préparerez le build (front + back), les variables d’environnement et un README professionnel pour un déploiement serein.