



Université
de Technologie
d'Abidjan

PROJET

Plateforme de gestion des
anciens sujets d'examen

Préparé par

DABO ALI

AGHO CHRIS



07-78-74-86-02
07-10-14-58-64



agohchris90@gmail.com
alsondab9@gmail.com

Contents

Présentation du Projet.....	3
Objectif.....	3
Fonctionnalités.....	4
Fonctions secondaires (priorité moyenne) :.....	4
Technologies et Outils	4
Planning Prévisionnel.....	5
Modèles et Données.....	6
Idées Bonus à Ajouter	7
Prochaines étapes immédiates.....	7
Cyle de vie du projet.....	8
Diagramme de Gant.....	8
Diagramme de classe	8
Diagramme de cas d'utilisation.....	8
Diagramme de sequences.....	8
Conclusion	8

Cahier des Charges

Plateforme de Gestion des Anciens Sujets Universitaires

Présentation du Projet

Nom du projet : Plateforme de gestion des anciens sujets universitaires

Responsables du projet :

- ✚ -Chef de projet / Responsable technique :5 Dabo Ali
- ✚ Développeur Back-end : py
- ✚ Développeur Front-end :Dabo Ali
- ✚ Responsable design UI/UX : Dabo Ali et py

Objectif :

Créer une plateforme web sécurisée permettant aux étudiants d'accéder aux anciens sujets d'examens et corrigés via un espace personnel, et aux administrateurs d'ajouter et gérer ces ressources.

Objectifs Détaillés

- ✚ Centraliser et sécuriser les anciens sujets et corrigés.
- ✚ Faciliter la révision des étudiants via une interface de recherche intuitive.
- ✚ Mettre en place une gestion d'utilisateurs basée sur les rôles (étudiant/admin).

- ✚ Permettre une gestion simplifiée des sujets et des corrigés pour les administrateurs.

Fonctionnalités

Pour les étudiants :

- ✚ Connexion sécurisée avec identifiant unique et mot de passe.
- ✚ Accès à un tableau de bord personnel.
- ✚ Recherche de sujets par matière, année, semestre.
- ✚ Téléchargement des sujets et corrigés (PDF).
- ✚ Consultation de l'historique de téléchargement.

Pour les administrateurs :

- ✚ Connexion à un espace admin.
- ✚ Ajout de nouveaux sujets et corrigés via un formulaire (upload de fichiers, métadonnées associées).
- ✚ Modification / suppression de sujets et corrigés.
- ✚ Gestion des utilisateurs (ajout/suppression étudiants, réinitialisation de mot de passe).

Fonctions secondaires (priorité moyenne) :

- ✚ Statistiques : nombre de téléchargements par sujet, par utilisateur.
- ✚ Gestion des demandes de sujets non disponibles (boîte de suggestions).
- ✚ Authentification par rôles : distinction Étudiant / Admin.
- ✚ Historique de connexions et actions.

Technologies et Outils

Côté Client (Frontend)	Côté Serveur (Backend)	Base de Données	Outils Divers
Vite Js	Django (Python)	PostgreSQL / MySQL	Postman (tests API)
Figma (UI Design)	Django REST Framework		Git & GitHub (versioning)
Tailwind CSS / Bootstrap	JWT pour l'authentification		Netlify/Vercel (front)

Planning Prévisionnel

Phase	Responsable(s)	Durée estimée	Statut
Création du dépôt GitHub	Dabo Ali	1 jour	
Rédaction du cahier des charges	Dabo Ali	1 jours	
Conception des maquettes (Figma)	Dabo Ali & Py		
Mise en place environnement	Dabo Ali & Py		
Définition des modèles et migrations	Py		
Développement de l'API (auth, sujets)	Py		
Intégration du front React	Dabo Ali		

(connexion, tableau de bord)	
Intégration API / Front	Py
Tests et ajustements	Dabo Ali & Py
Déploiement	Dabo Ali & Py

Modèles et Données

Utilisateur (User)

Champ	Type	Obligatoire
id	Auto (UUID)	Oui
identifiant	String	Oui
email	String (unique)	Oui
mot_de_passe	Hashé	Oui
rôle	String (étudiant/admin)	Oui

Sujet

Champ	Type	Obligatoire
id	Auto (UUID)	Oui
matière	String	Oui

année	String	Oui
semestre	String	
fichier_sujet	Fichier (PDF)	Oui
fichier_corrige	Fichier (PDF)	
ajouté_par	Référence admin	Oui

Sécurité

- ✚ Authentification sécurisée avec JWT.
- ✚ Hashage des mots de passe avec bcrypt.
- ✚ Rôles et permissions clairement séparés (admin / étudiant).
- ✚ Données et fichiers protégés via des accès restreints et serveurs HTTPS.

Idées Bonus à Ajouter

Tu peux décider d'intégrer certaines de ces idées plus tard :

- ✚ Téléchargements limités: par nombre ou période (ex : 5 par jour).
- ✚ Commentaires ou notation des sujets: pour aider les autres étudiants.
- ✚ Demandes de sujets spécifiques: si non disponibles.
- ✚ Notifications email: lors de la mise en ligne de nouveaux sujets.

Prochaines étapes immédiates

1. Créer le dépôt GitHub commun
2. Finaliser ce cahier des charges ensemble
3. Designer la page d'accueil et page de recherche sous Figma
4. Définir les modèles Django et lancer les premières migrations
5. Créer l'API Django REST
6. Développer l'interface React et connecter l'API
7. Effectuer les tests et déployer

Cycle de vie du projet

Diagramme de Gant

Diagramme de classe

Diagramme de cas d'utilisation

Diagramme de sequences

Conclusion

Ce cahier des charges vous servira de guide pour structurer le projet, répartir les tâches et suivre votre avancement. Vous pouvez bien sûr l'adapter et l'ajuster selon l'évolution du projet.