



CHAHIER DES CHARGES

PROJET : << TRANSFER DE FICHER >>

I. INTRODUCTION

➤ Objectif

Développer une application multiplateforme permettant aux étudiants de pouvoir avoir les sujets de années antérieurs.
Chaque étudiant sera enregistré, aura un profil,

➤ Technologie

- Web : Laravel pour le backend et Angular pour le frontend
- Mobile : React Native
- API : RESTful
- Base de données : MySQL
- Stockage de fichiers : S3

II. DESCRIPTION FONCTIONNELLE

a) Inscription et authentification

- **Enregistrement :**

L'enregistrement se fait par l'administration lors de l'inscription, après l'enregistrement l'étudiant peut avoir accès à son profil et modifier uniquement que le mot de passe, les autres informations sont propres à son identité et ne peut pas être modifié.

- **Connexion :**

Les étudiants pourront de connecter avec leurs noms d'utilisateur et leur mot de passe. Les mots de passes et nom d'utilisateur seront par défaut les différents matricules de chacun.

- **Réinitialisation et oublie de mot de passe :**

Ils pourront modifier, récupérer leur profil en cas d'oubli de mot de passe ou identifiant par le biais de leur email si le problème persiste, l'administration prend le relais.

b) Gestion de profils

- **Mise à jour :**

Les étudiants pourront mettre à jour **uniquement** l'email, le nom d'utilisation, le mot de passe.

c) Publication de fichier

- **Publication de fichier :**

Les étudiants et l'administration pourront publier ses sujets et documents en précisant la filière, l'années, le module et les autres pourrons les télécharger.

- **Extension de fixe :**

Les fichiers pourront être téléchargés sous forme de fichiers PDF.

d) Consultation et Téléchargement de fichier

- **Liste des fichiers :**

Les étudiants tout comme l'administration pourront voir une liste de devoirs publiés, filtrable par module et filière.

- **Téléchargement :**

Les étudiants pourront télécharger les fichiers sous forme de fichiers PDF.

III. DESCRIPTION TECHNIQUE

a) Architecture

- **Backend :**

Développé avec Laravel pour gérer la logique métier, l'authentification, et les API RESTful.

- **Frontend :**

Développé avec Angular pour une interface utilisateur interactive.

- **Mobile :**

Développé avec React Native pour une application mobile multiplateforme (iOS et Android).

- **Base de données :**

Utilisation de MySQL pour stocker les données utilisateurs, les devoirs, etc.

- **Stockage de fichier :**

Utilisation d'un service de stockage cloud AWS S3 pour les fichiers

b) APIs RESTfull

- **Auth API :**

Gestion de connexion, et réinitialisation de mot de passe.

- **User API :**

Gestion des profils.

- **Assignement API :**

Gestion de la publication, consultation, et téléchargement des devoirs.

c) Sécurité

- **Validation des Entrées :**

Validation des données utilisateur côté serveur pour prévenir les injections SQL et autres attaques.

- **Auth APIs :**

Utilisation de tokens JWT pour sécuriser les APIs.

- **HTTPS :**

Chiffrement des communications entre le client et le serveur.

IV. DEVELOPPEMENT ET DEPLOIEMENT

a) Environnement de développement

L'éditeur de texte **visual studio code**

b) Déploiement

V. PLANNING ET BUDGET

a) Etape du projet

b) Budget prévisionnel

LIVRABLE :

✓ WEB

✓ MOBILE (ANDROID ,IOS)