A white sign with green text

Description automatically generated

**CAHIER DES CHARGES**

**3ème année**

**Ingénierie Informatique et Réseaux**

Conception et realisation d'une application web E-learning

**Sous le thème**

**Réalisé par :**

Sanae Handaoui

Kenza Chouni

**Encadre par :**

1. Périmètre du projet

1.1 But

L’objectif de cette application web E-learning est de fournir une plateforme interactive et accessible qui permet aux étudiants d’apprendre en ligne, aux enseignants de partager des cours et aux administrateurs de gérer l’ensemble du système. Cette solution vise à faciliter l’apprentissage à distance en intégrant des fonctionnalités modernes comme des vidéos, des quiz interactifs, des forums de discussion et un suivi des performances des apprenants.

**Fonctionnalités principales :**

* **Gestion des formations** 📚
* **Gestion des professeurs** 👨‍🏫
* **Gestion des utilisateurs** 👥
* **Support multilingue** 🌍
* **Version web et mobile** 📱💻

**Choix des technologies :**

* **Frontend** : **React** – Interface réactive et dynamique.
* **Backend** : **Flask** – API légère et rapide pour la gestion des données.
* **Base de données** : **MySQL** – Gestion fiable des données.

1.2 Missions

● Etude technique

● Etude fonctionnelle

● Conception

● Réalisation

● Déploiement

● Documentation

1.3 Livrables

|  |  |  |  |
| --- | --- | --- | --- |
| **Livrable** | **Phase** | **Date de livraison** | **Etat** |
| Cahier des charges | Etude préalable |  |  |
| Dossier de  Conception | Etude détaillée |  |  |
| Code source/installable  (Par module) | Réalisation |  |  |
| Documentation API | Etude détaillée |  |  |
| Manuel d’utilisation | Déploiement  Rapport de test |  |  |

1.4 Réalisation

1.4.1 Contrainte technique

La réalisation de l’application web E-learning est soumise à plusieurs contraintes techniques qui doivent être prises en compte pour assurer la performance, la sécurité et la scalabilité du système

1.4.2 Contrainte de Sécurité

- Cryptage des données sensibles.

- Hachage des mots de passe.

- Utilisation de JWT pour l’authentification.

- Prévention des attaques XSS et CSRF.

- Contrôle d’accès basé sur les rôles.

- Surveillance des logs et gestion des incidents.

1.4.3 Contrainte de délais

Durée de 3 mois, incluant la gestion des risques.

1.4.4 Contrainte de maintenance

- Fournir une maintenance complète et assurer une bonne scalabilité

1.5 Equipe de travail et rôles

L’application est développée en solo par **Sanae Handaoui Kenza Chouni**

, qui prend en charge l’ensemble des aspects du projet :

👨‍💻 **Développeur Full-Stack**

* **Front-end (React.js)** : Conception et développement de l’interface utilisateur.
* **Back-end (Flask)** : Développement des API et logique métier.
* **Base de données (MySQL)** : Conception et gestion du stockage des données.
* **Sécurité et authentification** : Mise en place d’un système sécurisé pour la gestion des utilisateurs.

Ce projet est entièrement conçu et développé par **Sanae Handaoui Kenza Chouni**, assumant à la fois le rôle de **concepteur, développeur et intégrateur**.

# Etude de l’existant

## Concept métier

L’objectif de cette application web d’aide à l’apprentissage est de fournir une plateforme interactive et accessible qui permet aux étudiants d'apprendre en ligne, aux enseignants de partager des cours et aux administrateurs de gérer l’ensemble du système. Cette solution facilite l’apprentissage à distance en intégrant des fonctionnalités modernes telles que des vidéos, des quiz interactifs, des forums de discussion et un suivi des performances des apprenants.

Fonctionnalités principales :

1. **Gestion d’authentification 🔐**
   * Permettre aux utilisateurs de créer un compte sécurisé et de se connecter à la plateforme.
   * Implémentation de mécanismes de sécurité tels que le cryptage des données sensibles, le hachage des mots de passe et l'utilisation de JSON Web Tokens (JWT) pour l'authentification.
2. **Gestion des cours 📚**
   * Offrir des fonctionnalités pour télécharger, organiser et gérer les fichiers audio/vidéo ou textuels associés aux cours.
   * Stockage structuré des données dans une base MySQL.
3. **Gestion de commentaires :**

* Permet aux étudiants de laisser et modérer des commentaires.
* Interaction avec les enseignants.

1. **Gestion des professeurs :**

* Ajout, modification, et gestion des profils de professeurs.
* Suivi des performances et gestion des matières.

1. **Gestion des avis des utilisateurs :**

* Laisser des avis sur les cours et professeurs.
* Notes et commentaires avec modération par les administrateurs.

1. **Support multilingue 🌍**
   * Prendre en charge plusieurs langues pour répondre aux besoins internationaux.
2. **Gestion des chats (optionnelle) 💬**
   * Faciliter les interactions en temps réel entre utilisateurs (étudiants, professeurs) via une interface de chat intégrée.

## Logiciels similaires

* **Moodle**
  + **Fonctionnalités** : Plateforme open-source de gestion de l'apprentissage, gestion des cours, suivi des performances des étudiants, quiz et examens en ligne.
  + **Limites** : Interface utilisateur un peu complexe, nécessite des personnalisations pour certaines fonctionnalités avancées.
  + **Site officiel** : [Moodle](https://moodle.org)
* **Teachable**
  + **Fonctionnalités** : Création de cours en ligne, gestion des paiements, suivi des progrès des étudiants, création de quiz, vidéo et contenu multimédia.
  + **Limites** : Modèle payant avec des frais de plateforme pour les créateurs de contenu.
  + **Site officiel** : [Teachable](https://teachable.com)
* **Thinkific**
  + **Fonctionnalités** : Outils pour créer des cours interactifs, gestion des étudiants, intégration de vidéos et de quiz.
  + **Limites** : Tarification assez élevée pour les fonctionnalités premium.
  + **Site officiel** : [Thinkific](https://www.thinkific.com)

# Etude fonctionnelle

## Objectifs fonctionnels

## Besoins fonctionnels : Fonctionnalités (user stories)

### 2.1 Bloc fonctionnel : Gestion ...

**En tant qu’… :**

* Function 1
* Function 2.

### Bloc fonctionnel : Gestion ...

**En tant qu’… :**

* Function 1
* Function 2.

## Acteurs

* Guest
* Admin
* Professeur
* Etudiant

# Besoins non fonctionnels

# Stratégie de développement et plannification

## Méthodologie de développement

Décrire la méthode adoptée.

## Plan de test

Définir les tests unitaires, d’intégration et de validation utilisateur.

## Découpage en Lots de Travail

Présenter les différentes phases et responsabilités associées.

## Échéancier

Définir un calendrier avec les grandes étapes et le détail.

# Architecture du système

## Diagrammes UML

## Technologies utilisées

## Architecture logicielle

Définir les composants principaux et leur interaction.

# Annexes

Inclure toute documentation supplémentaire, glossaire, références, ...