

# Cahier des charges — Projet StudyHub

Plateforme collaborative de gestion et de partage de connaissances pour étudiants

## 1. Contexte et objectifs

Dans le cadre du projet **BDC INGE1 2025-26**, l'équipe projet doit concevoir et développer une plateforme web permettant aux étudiants de centraliser, organiser et enrichir une base de connaissances commune.

L'objectif principal est de fournir un outil fiable, performant et sécurisé qui permette :

- le partage structuré d'informations académiques (fiches, méthodes, bonnes pratiques),
- la consultation rapide et efficace via un moteur de recherche avancé,
- la contribution collaborative (création, modification, validation, commentaires),
- la gestion centralisée des rôles, des utilisateurs et des contenus.

La période de développement s'étend du **16 septembre 2025 au 14 novembre 2025**.

## 2. Périmètre fonctionnel

### 2.1 Fonctionnalités principales (MVP)

- **Gestion des utilisateurs**
  - Inscription et authentification sécurisée
  - Gestion des rôles : étudiant, modérateur, administrateur
  - Profils utilisateurs personnalisables
- **Recherche et navigation**
  - Moteur de recherche par mots-clés, catégories, tags
  - Arborescence de navigation par catégories
  - Consultation des fiches avec historique des versions
- **Contribution collaborative**
  - Création de fiches de connaissances par les utilisateurs
  - Système de propositions de modifications avec workflow de validation
  - Gestion des commentaires sur chaque fiche
  - Signalement de contenu inapproprié
- **Administration et modération**

- Tableau de bord d'administration (gestion des utilisateurs, rôles, catégories, contenus)
- Outils de modération (validation, suppression, sanctions)
- Gestion des droits d'accès

## 2.2 Fonctionnalités optionnelles (bonus)

- Forum de discussion thématique
- Chat en temps réel entre utilisateurs connectés
- Calendrier collaboratif (examens, événements)
- Notifications intelligentes (nouveaux contenus pertinents)
- Système de gamification (badges, votes, likes)

## 3. Exclusions du périmètre

- Pas d'intégration avec des services propriétaires (Notion, Confluence, etc.)
- Pas d'utilisation de technologies non open-source
- Pas de gestion de médias lourds (streaming vidéo, fichiers volumineux)

## 4. Typologie des utilisateurs

- **Étudiants** : utilisateurs principaux, création et consultation de contenu.
- **Modérateurs** : contrôle qualité, validation des propositions, modération des échanges.
- **Administrateurs** : supervision générale, gestion des utilisateurs et des catégories, configuration globale.

## 5. Cas d'utilisation représentatifs

- **Recherche d'information** : consultation rapide via recherche par mots-clés.
- **Partage de contenu** : création et publication d'une nouvelle fiche.
- **Amélioration continue** : proposition de mise à jour sur une fiche existante.

- **Interaction sociale** : ajout de commentaires constructifs, échanges entre étudiants.

## 6. Architecture technique

- **Frontend**
  - Framework : React
  - Outil de build : Vite
  - Style : TailwindCSS (responsive, mobile-first)
- **Backend**
  - Environnement : Node.js
  - Framework : Express.js
  - API REST (JSON)
  - Authentification : JWT (JSON Web Tokens)
- **Base de données**
  - SGBD : MySQL
  - ORM : Sequelize

## 7. Modèle de données (entités principales)

- **User** : id, email, mot de passe (hashé), rôle, profil, date de création
- **Category** : id, nom, description
- **Article** : id, titre, résumé, contenu, catégorie, statut, auteur, dates
- **Proposal** : id, type (nouveau / modification), article associé, auteur, statut, contenu, date
- **Comment** : id, article, auteur, contenu, statut, date
- **Tag** : id, mot-clé (relation N-N avec Article)
- (*Bonus*) : Notification, Like

## 8. Planning prévisionnel

- **Phase 1 : Fondations (16-20 septembre)**  
Cahier des charges, maquettes, environnement de développement.

- **Phase 2 : Spécifications (22-26 septembre)**  
Spécifications fonctionnelles détaillées, MCD/MLD, définition de l'API.
- **Phase 3 : Développement principal (29 septembre – 18 octobre)**  
Authentification, gestion utilisateurs, recherche, fiches, propositions, commentaires, interface admin.
- **Phase 4 : Intégration et tests (20 octobre – 1er novembre)**  
Assemblage, tests fonctionnels, sécurité, validation.
- **Phase 5 : Livraison (3 – 14 novembre)**  
Documentation technique, ajout éventuel des fonctionnalités bonus, déploiement final.

## 9. Critères de validation

- **Succès minimal**
  - Authentification sécurisée et gestion des rôles fonctionnelle
  - Création, consultation et modification de fiches opérationnelles
  - Recherche et navigation efficaces
  - Interface d'administration utilisable et complète
  - Documentation technique disponible
- **Succès optimal**
  - Interface intuitive et fluide
  - Bonne performance avec forte volumétrie de contenu
  - Mise en œuvre des fonctionnalités bonus
  - Code documenté, testé et maintenable

## 10. Analyse des risques et mesures préventives

- **Dépassement du planning** → Priorisation stricte du MVP, bonus conditionnels
- **Manque de données pour tests** → Génération d'un jeu de données réaliste
- **Vulnérabilités de sécurité** → Bonnes pratiques + audits réguliers
- **Dette technique** → Revues de code, intégration continue

## 11. Conclusion

**StudyHub** est conçu comme un projet structurant pour la communauté étudiante. La démarche repose sur :

- une architecture moderne et maintenable,
- une méthodologie incrémentale (MVP puis bonus),
- une orientation utilisateur claire (simplicité, efficacité, collaboration).

La réussite du projet dépendra de la discipline d'exécution (respect des jalons, sécurité, qualité du code) et de la capacité de l'équipe à répondre aux besoins concrets des utilisateurs finaux.