

# CAHIER DES CHARGES

---

## Bibliothèque de Prompts pour Intelligence Artificielle

<b>Organisme</b>	CNAV (Caisse Nationale d'Assurance Vieillesse)
<b>Service</b>	DIE - Direction des Services Informatiques
<b>Type de projet</b>	Application Web Interne
<b>Auteur</b>	Stagiaire BTS
<b>Date</b>	05/01/2026
<b>Version</b>	1.0

## **Table des matières**

- 1. Présentation du projet
- 2. Contexte et problématique
- 3. Objectifs du projet
- 4. Public cible et périmètre
- 5. Description fonctionnelle
- 6. Contraintes techniques
- 7. Architecture générale
- 8. Technologies utilisées
- 9. Gestion des données
- 10. Sécurité
- 11. Ergonomie et expérience utilisateur
- 12. Planning prévisionnel
- 13. Évolutions possibles
- 14. Conclusion

## **1. Présentation du projet**

Ce document constitue le cahier des charges de l'application « Bibliothèque de Prompts IA », développée dans le cadre d'un stage de BTS. Le projet consiste en la création d'une application web permettant de centraliser, organiser et partager des prompts optimisés pour l'utilisation d'intelligences artificielles.

L'application vise à résoudre une problématique concrète rencontrée par les collaborateurs : la difficulté à exploiter pleinement le potentiel des outils d'intelligence artificielle par manque de connaissance des techniques de prompting efficaces.

## **2. Contexte et problématique**

### **2.1 Contexte organisationnel**

La CNAV (Caisse Nationale d'Assurance Vieillesse) s'inscrit dans une démarche de transformation numérique et d'adoption des technologies émergentes, notamment l'intelligence artificielle. La Direction des Services Informatiques (DIE) joue un rôle central dans cette transformation en accompagnant les équipes vers l'utilisation optimale des nouveaux outils.

### **2.2 Problématique identifiée**

L'observation du terrain a révélé plusieurs difficultés :

- Méconnaissance des techniques de prompting efficace par la majorité des utilisateurs
- Résultats insatisfaisants obtenus avec les IA générant de la frustration et un abandon progressif des outils
- Absence de référentiel commun de bonnes pratiques en matière d'utilisation des IA
- Temps perdu à chercher comment formuler correctement ses requêtes
- Disparité dans la qualité des interactions avec les IA selon les collaborateurs

### **2.3 Opportunité identifiée**

La création d'une bibliothèque centralisée de prompts testés et validés représente une solution pragmatique permettant de :

- Capitaliser sur les bonnes pratiques existantes
- Accélérer la courbe d'apprentissage des nouveaux utilisateurs
- Harmoniser la qualité des interactions avec les IA au sein de l'organisation
- Favoriser le partage de connaissances entre collaborateurs

## **3. Objectifs du projet**

### **3.1 Objectif principal**

Développer une application web permettant aux collaborateurs de la CNAV d'accéder facilement à une bibliothèque de prompts optimisés pour améliorer leur utilisation des intelligences artificielles.

### **3.2 Objectifs spécifiques**

- Centraliser les prompts efficaces au sein d'une plateforme unique et accessible
- Organiser les prompts selon une taxonomie claire (frameworks, catégories, sous-catégories)
- Permettre aux utilisateurs de rechercher rapidement le prompt adapté à leur besoin
- Faciliter la réutilisation des prompts via un système de copie en un clic
- Identifier les prompts les plus populaires grâce à un système de likes
- Permettre aux utilisateurs de contribuer en créant de nouveaux prompts
- Offrir une expérience utilisateur moderne, intuitive et responsive

### **3.3 Indicateurs de succès**

- Taux d'adoption : au moins 70% des collaborateurs de la DIE utilisent l'application dans les 3 mois
- Satisfaction utilisateur : note moyenne supérieure à 4/5
- Création de prompts : au moins 100 prompts disponibles dans les 6 mois
- Engagement : moyenne de 10 likes par prompt populaire

## **4. Public cible et périmètre**

### **4.1 Utilisateurs cibles**

Phase 1 - Déploiement initial (mois 1-3) :

- Collaborateurs de la DIE (Direction des Services Informatiques)
- Nombre estimé : 60 utilisateurs
- Profils : développeurs, chefs de projet, architectes, responsables techniques

Phase 2 - Extension (mois 4-12) :

- Extension progressive à l'ensemble des collaborateurs de la CNAV
- Nombre estimé : jusqu'à 5 000 utilisateurs
- Profils : tous métiers confondus utilisant des outils d'IA

### **4.2 Périmètre fonctionnel**

Inclus dans le périmètre :

- Consultation et recherche de prompts
- Organisation par frameworks de prompting (PROMPT, RACE, COSTAR, etc.)
- Catégorisation thématique des prompts
- Système de likes et de popularité
- Création de nouveaux prompts par les utilisateurs
- Gestion des frameworks, catégories et sous-catégories (CRUD complet)
- Mode d'affichage clair/sombre

Exclus du périmètre (évolutions futures possibles) :

- Authentification et gestion des utilisateurs (non nécessaire pour v1.0)
- Intégration directe avec des APIs d'IA
- Historique personnel d'utilisation des prompts
- Système de notation détaillé (commentaires)
- Gestion des versions de prompts

## 5. Description fonctionnelle

### 5.1 Architecture fonctionnelle

L'application s'organise autour de cinq modules principaux permettant de couvrir l'ensemble des besoins identifiés.

### 5.2 Fonctionnalités détaillées

#### Module 1 : Consultation des prompts

Description :

Ce module constitue le cœur de l'application. Il permet aux utilisateurs de parcourir la bibliothèque de prompts et d'accéder rapidement à celui qui correspond à leur besoin.

Fonctionnalités incluses :

- Affichage de l'ensemble des prompts disponibles sous forme de cartes
- Informations affichées par prompt : titre, description, framework utilisé, catégorie, nombre de likes
- Vue détaillée d'un prompt avec l'intégralité de son contenu
- Bouton de copie rapide permettant de copier le prompt dans le presse-papier en un clic
- Feedback visuel confirmant la copie réussie

#### Module 2 : Recherche et filtrage

Description :

Pour faciliter la navigation dans une bibliothèque potentiellement volumineuse, ce module offre des outils de recherche et de filtrage avancés.

Fonctionnalités incluses :

- Barre de recherche textuelle permettant de chercher dans le titre et le contenu des prompts

- Filtrage par framework de prompting (PROMPT, RACE, COSTAR, etc.)
- Filtrage par catégorie thématique (Rédaction, Analyse, Code, etc.)
- Filtrage par sous-catégorie pour une granularité fine
- Tri par popularité (nombre de likes décroissant)
- Tri alphabétique
- Combinaison possible de plusieurs critères de filtrage

#### **Module 3 : Système d'interaction**

Description :

Ce module favorise l'engagement des utilisateurs et permet d'identifier les prompts les plus pertinents grâce à un système de likes.

Fonctionnalités incluses :

- Bouton 'like' sur chaque prompt
- Compteur de likes visible
- Identification visuelle des prompts les plus populaires
- Tri automatique par popularité disponible
- Persistance des likes en base de données

#### **Module 4 : Création de prompts**

Description :

Pour enrichir collectivement la bibliothèque, les utilisateurs peuvent créer et soumettre leurs propres prompts.

Fonctionnalités incluses :

- Formulaire de création guidé
- Sélection du framework de prompting désiré
- Génération automatique du format de prompt selon le framework choisi
- Champs de saisie : titre, description, contenu du prompt
- Sélection de la catégorie et sous-catégorie
- Validation et enregistrement en base de données
- Mise en ligne immédiate du prompt créé

#### **Module 5 : Administration de la taxonomie**

Description :

Ce module permet de gérer la structure organisationnelle de la bibliothèque (frameworks, catégories, sous-catégories).

Fonctionnalités incluses :

- Page répertoriant tous les frameworks disponibles
- Opérations CRUD (Create, Read, Update, Delete) sur les frameworks
- Opérations CRUD sur les catégories
- Opérations CRUD sur les sous-catégories
- Gestion des relations entre catégories et sous-catégories
- Interface d'administration accessible et intuitive

## 6. Contraintes techniques

### 6.1 Contraintes d'environnement

- Hébergement en interne sur l'infrastructure CNAV
- Compatibilité avec les navigateurs standards (Chrome, Firefox, Edge)
- Conformité aux politiques de sécurité de la CNAV
- Accessibilité depuis le réseau interne uniquement

### 6.2 Contraintes de performance

- Temps de chargement initial de la page < 2 secondes
- Temps de réponse aux recherches < 500 ms
- Support de 60 utilisateurs simultanés en phase 1
- Scalabilité jusqu'à 5 000 utilisateurs en phase 2
- Disponibilité cible : 99% pendant les heures ouvrées

### 6.3 Contraintes de compatibilité

- Responsive design : support desktop (1920x1080), tablette (768x1024), mobile (375x667)
- Support des navigateurs : Chrome, Firefox, Edge (2 dernières versions)

## 7. Architecture générale

### 7.1 Architecture applicative

L'application adopte une architecture MVC (Model-View-Controller) classique s'appuyant sur le framework .NET et ASP.NET Core.

Couches de l'application :

- Couche Présentation : vues CSHTML, JavaScript, CSS pour l'interface utilisateur
- Couche Contrôleurs : gestion des requêtes HTTP et orchestration de la logique métier
- Couche Modèles : classes C# représentant les entités métier

- Couche Accès aux données : Entity Framework pour la communication avec la base de données
- Base de données : stockage des prompts, frameworks, catégories, sous-catégories et statistiques

## 7.2 Modèle de données

Entités principales :

- Prompt : id, titre, description, contenu, framework\_id, catégorie\_id, sous\_catégorie\_id, nombre\_likes, date\_création
- Framework : id, nom, description, format\_template
- Catégorie : id, nom, description
- SousCatégorie : id, nom, description, catégorie\_id

Relations :

- Un Prompt appartient à un Framework (1:N)
- Un Prompt appartient à une Catégorie (1:N)
- Un Prompt appartient à une SousCatégorie (1:N)
- Un Prompt appartient à une SousCatégorie (1:N)

# 8. Technologies utilisées

## 8.1 Backend

- Langage : C# 8.0.0
- Framework : ASP.NET Core 8.0 (ou version ultérieure)
- ORM : Entity Framework Core pour la gestion de la base de données
- Architecture : MVC (Model-View-Controller)

## 8.2 Frontend

- Moteur de templates : Razor (CSHTML)
- JavaScript : JavaScript vanilla pour les interactions dynamiques
- Styles : CSS3 avec variables pour le mode clair/sombre
- Responsive design : Media queries CSS

## 8.3 Base de données

- Système : SQL Server (version compatible avec l'infrastructure CNAV)
- Gestion : Entity Framework Code First avec migrations
- Initialisation : seed data pour pré-remplir les données de référence

## 8.4 Outils de développement

- IDE : Visual Studio 2022 ou Visual Studio Code
- Contrôle de version : Svn

- Gestionnaire de packages : NuGet

## 9. Gestion des données

### 9.1 Stratégie de persistance

Entity Framework Core est utilisé pour gérer l'ensemble des opérations de persistance des données. L'approche Code First permet de définir le modèle de données directement en C# et de générer automatiquement la structure de base de données via des migrations.

### 9.2 Initialisation des données

Un système de seed data est mis en place pour :

- Pré-remplir les frameworks de prompting standards (PROMPT, RACE, COSTAR, etc.)
- Créer les catégories et sous-catégories de base
- Fournir des prompts exemples pour faciliter la prise en main
- Garantir une base de données opérationnelle dès le déploiement

### 9.3 Sauvegarde et restauration

Les sauvegardes de la base de données suivent la politique standard de la CNAV :

- Sauvegardes complètes quotidiennes
- Conservation sur 30 jours glissants
- Procédure de restauration documentée
- Tests de restauration trimestriels

## 10. Sécurité

### 10.1 Niveau de sécurité requis

L'application ne manipule aucune donnée sensible ou personnelle. Les prompts sont des contenus techniques destinés à l'optimisation de l'utilisation des IA. Par conséquent, le niveau de sécurité requis est standard pour une application interne.

### 10.2 Mesures de sécurité implémentées

- Hébergement sur le réseau interne CNAV uniquement (pas d'exposition Internet)
- Utilisation de connexions HTTPS pour les échanges
- Protection contre les injections SQL via Entity Framework et requêtes paramétrées
- Validation des entrées utilisateur côté serveur
- En-têtes de sécurité HTTP configurés (X-Frame-Options, X-Content-Type-Options, etc.)

## **10.3 Authentification et autorisation**

Version 1.0 (actuelle) : Pas de système d'authentification

L'application est accessible à tous les utilisateurs du réseau interne sans nécessité de se connecter.

Ce choix se justifie par :

- Absence de données sensibles ou personnelles
- Public cible restreint (réseau interne uniquement)
- Volonté de faciliter l'adoption en supprimant toute friction
- Contexte de preuve de concept en phase initiale

Évolution prévue (Phase 2) : Intégration avec Active Directory

En cas d'extension à l'ensemble de la CNAV, l'authentification via Active Directory sera intégrée pour :

- Identifier les contributeurs de prompts
- Personnaliser l'expérience utilisateur
- Gérer des permissions différencierées (administrateurs, contributeurs, lecteurs)

## **11. Ergonomie et expérience utilisateur**

### **11.1 Principes d'ergonomie**

L'interface utilisateur est conçue selon les principes suivants :

- Simplicité : navigation intuitive sans formation préalable
- Clarté : information visible et hiérarchisée
- Efficacité : accès rapide aux fonctionnalités principales
- Cohérence : design uniforme sur l'ensemble de l'application
- Feedback : retour visuel immédiat sur chaque action

### **11.2 Design visuel**

- Palette de couleurs professionnelle et sobre
- Espacement généreux pour une lecture confortable
- Mise en avant des actions principales (boutons d'action)
- Icônes universelles pour guider l'utilisateur

### **11.3 Mode clair / Mode sombre**

L'application propose deux thèmes visuels :

- Mode clair : fond blanc, texte sombre (par défaut)
- Mode sombre : fond sombre, texte clair (confort visuel en environnement peu éclairé)

- Commutateur accessible en permanence dans l'interface
- Préférence mémorisée dans le navigateur de l'utilisateur
- Transition fluide entre les deux modes

## 11.4 Responsive design

L'application s'adapte automatiquement aux différentes tailles d'écran :

- Desktop ( $\geq 1200\text{px}$ ) : affichage en grille multi-colonnes
- Tablette (768px - 1199px) : affichage en grille 2 colonnes
- Mobile ( $< 768\text{px}$ ) : affichage en une seule colonne
- Menu de navigation adaptatif (hamburger menu sur mobile)
- Taille de police et espacement ajustés selon l'écran

## 12. Planning prévisionnel

Le projet s'est déroulé sur une période de stage BTS de 8 semaines, organisée en phases successives.

### 12.1 Phase 1 : Cadrage et conception (Semaines 1-2)

- Analyse du besoin et rencontres avec les utilisateurs potentiels
- Rédaction du cahier des charges
- Conception de la base de données (modèle entité-relation)
- Maquettage des écrans principaux
- Choix des technologies et mise en place de l'environnement de développement

### 12.2 Phase 2 : Développement du socle technique (Semaines 3-4)

- Mise en place du projet ASP.NET Core
- Configuration d'Entity Framework et création de la base de données
- Développement des modèles de données
- Création des migrations et seed data
- Mise en place de l'architecture MVC

### 12.3 Phase 3 : Développement des fonctionnalités principales (Semaines 5-7)

- Affichage de la liste des prompts
- Système de recherche et filtrage
- Détail d'un prompt et fonction de copie
- Système de likes
- Création de nouveaux prompts
- Génération automatique du format selon le framework

### 12.4 Phase 4 : Fonctionnalités avancées et administration (Semaine 8)

- Page de gestion des frameworks
- CRUD complet pour les catégories et sous-catégories

- Implémentation du mode clair/sombre
- Optimisation du responsive design

## **12.5 Phase 5 : Tests et documentation (Semaine 9)**

- Tests fonctionnels sur différents navigateurs et appareils
- Corrections de bugs identifiés
- Optimisation des performances
- Rédaction de la documentation utilisateur
- Rédaction de la documentation technique

## **12.6 Phase 6 : Déploiement et clôture (Semaine 10)**

- Déploiement sur l'environnement de test
- Session de démonstration auprès de la DIE
- Formation des premiers utilisateurs
- Recueil des premiers retours
- Présentation finale du projet

## **13. Évolutions possibles**

Plusieurs axes d'amélioration ont été identifiés pour faire évoluer l'application au-delà de sa version initiale.

### **13.1 Court terme (3-6 mois)**

- Intégration de l'authentification via Active Directory
- Système de favoris personnels pour chaque utilisateur
- Historique des prompts utilisés
- Système de notation avec commentaires (en complément des likes)
- Export de prompts en PDF ou fichier texte
- Statistiques d'utilisation pour les administrateurs

### **13.2 Moyen terme (6-12 mois)**

- Gestion des versions de prompts (historique des modifications)
- Système de validation / modération avant publication
- Rôles et permissions (administrateur, contributeur vérifié, utilisateur)
- Suggestions de prompts basées sur l'historique d'utilisation
- Tags supplémentaires pour affiner la recherche
- API REST pour permettre l'intégration avec d'autres outils internes

### **13.3 Long terme (> 12 mois)**

- Intelligence artificielle pour suggérer des améliorations de prompts
- Analyse de la performance des prompts (taux de réutilisation, satisfaction)
- Intégration directe avec des plateformes d'IA (ChatGPT, Claude, etc.)
- Génération assistée de prompts via formulaires intelligents

- Communauté d'utilisateurs avec badges et gamification
- Version mobile native (iOS/Android)

### **13.4 Évolutions techniques**

- Migration vers une architecture API RESTful + Frontend séparé (React, Vue.js)
- Mise en place d'un système de cache (Redis) pour améliorer les performances
- Containerisation avec Docker pour faciliter le déploiement
- Mise en place de tests automatisés (unitaires, intégration)
- Pipeline CI/CD pour l'automatisation des déploiements

## **14. Conclusion**

La Bibliothèque de Prompts IA répond à un besoin réel identifié au sein de la CNAV : faciliter et optimiser l'utilisation des intelligences artificielles par les collaborateurs. En centralisant des prompts testés et validés, l'application permet de réduire significativement la courbe d'apprentissage et d'harmoniser les pratiques.

Le projet démontre plusieurs atouts majeurs :

- Simplicité d'utilisation : interface intuitive ne nécessitant aucune formation
- Pragmatisme : réponse directe à une problématique terrain
- Évolutivité : architecture permettant d'envisager sereinement les évolutions futures
- Scalabilité : capacité à passer de 60 à 5 000 utilisateurs
- Adoption : engagement des utilisateurs favorisé par le système de likes et la contribution

Ce projet de stage constitue une base solide qui pourra être enrichie progressivement selon les retours utilisateurs et l'évolution des besoins organisationnels. L'approche itérative retenue permet de valider rapidement la pertinence du concept tout en gardant la flexibilité nécessaire pour adapter la solution.

Au-delà de sa dimension technique, ce projet s'inscrit dans une dynamique de transformation digitale et d'adoption des nouvelles technologies au sein de la CNAV, contribuant ainsi à la modernisation des pratiques et à l'amélioration continue de la productivité.

## Annexes

### Annexe A : Glossaire

**Prompt :** Instruction ou requête formulée pour interroger une intelligence artificielle

**Framework de prompting :** Méthode structurée pour construire des prompts efficaces (ex: PROMPT, RACE, COSTAR)

**CRUD :** Create, Read, Update, Delete - opérations de base sur les données

**Seed data :** Données initiales injectées dans la base de données au premier déploiement

**Responsive design :** Conception d'interface s'adaptant automatiquement à toutes les tailles d'écran

**Entity Framework :** ORM (Object-Relational Mapping) permettant de manipuler la base de données via des objets C#

**MVC :** Model-View-Controller - pattern architectural séparant les données, la logique et l'affichage

### Annexe B : Exemples de frameworks de prompting

**PROMPT :** Problem, Role, Objective, Method, Parameters, Test

**RACE :** Role, Action, Context, Expectation

**COSTAR :** Context, Objective, Style, Tone, Audience, Response

### Annexe C : Contacts du projet

Rôle	Contact
<b>Maître de stage</b>	Grégory Vincent
<b>Responsable DIE</b>	Stéphane David
<b>Stagiaire</b>	Théo OGER