

# Cahier des charges :

## “Plan Genie AI”

## 1. Introduction

**Nom du projet :** “Plan Genie AI”

**Description :**

Conception et développement d'une application intelligente destinée à assister les utilisateurs dans la gestion de leurs tâches et événements quotidiens en exploitant des technologies avancées d'intelligence artificielle (IA).

**Durée estimée du projet :** 6 mois.

**Langages de programmation et outils prévus :**

- **Langages :** Python, JavaScript, Dart.
- **Frameworks et bibliothèques :** Django/DRF, FastAPI, Angular, React.js, Flutter.
- **Outils tiers :** Google Speech-to-Text API, Whisper, OpenAI GPT, spaCy, Hugging Face Transformers.
- **Bases de données :** PostgreSQL, MongoDB.
- **Visualisation des données :** Chart.js, ApexCharts, D3.js.

## 2. Objectifs du projet

### 2.1. Automatisation de la gestion des tâches et des événements :

Développer un système capable de convertir automatiquement les entrées des utilisateurs (notes textuelles ou messages vocaux) en tâches organisées et planifiées. Cela inclut l'extraction des informations essentielles telles que les priorités, les délais, et les dates clés.

### 2.2. Facilitation de la planification quotidienne :

Fournir un calendrier intégré qui synchronise les événements extraits des données utilisateur, permettant une vue claire et consolidée des activités à venir.

### 2.3. Amélioration de la productivité personnelle :

Utiliser l'IA pour générer des rapports d'activités détaillés, des analyses de performance, et des graphiques interactifs afin d'aider les utilisateurs à identifier les domaines où ils peuvent optimiser leur temps et leurs efforts.

### 2.4. Interface intuitive et accessible :

Proposer une interface utilisateur ergonomique et responsive, accessible aussi bien sur les plateformes web que mobiles. L'objectif est d'offrir une expérience utilisateur fluide et agréable, même pour les non-techniciens.

### 2.5. Notifications et rappels intelligents :

Développer un système de rappels et d'alertes basés sur les priorités des tâches et les échéances afin d'assurer que les utilisateurs ne manquent aucune activité importante.

### 2.6. Suivi des performances et statistiques :

Mettre en œuvre des fonctionnalités de suivi qui permettent aux utilisateurs de visualiser

leurs performances, y compris des analyses sur la répartition du temps et l'accomplissement des objectifs à travers des tableaux de bord interactifs.

### 2.7. Support multi-langues et flexibilité :

Garantir une accessibilité à une large audience en intégrant le support de plusieurs langues et en permettant des ajustements personnalisés pour répondre aux besoins spécifiques de chaque utilisateur.

## 3. Fonctionnalités principales

### 3.1. Importation de données

- **Importation de notes textuelles** : Permettre aux utilisateurs de saisir des informations directement sous forme de texte via l'application.
- **Conversion des messages vocaux** : Intégrer des technologies de reconnaissance vocale pour convertir des messages vocaux en texte exploitable.

### 3.2. Analyse et traitement des données

- **Extraction des informations pertinentes** : Utiliser des algorithmes de NLP pour identifier les priorités, délais, dates, et contextes à partir des données utilisateur.
- **Classification des tâches et événements** : Organiser automatiquement les données en catégories logiques.

### 3.3. Gestion des tâches et événements

- **Génération automatique de listes de tâches** : Créer des listes organisées avec priorités et délais en fonction des informations saisies.
- **Planification intégrée** : Synchroniser les tâches et événements avec un calendrier intégré.
- **Modification manuelle** : Permettre aux utilisateurs d'ajuster ou de compléter les tâches et événements générés.

### 3.4. Suivi des performances

- **Rapports quotidiens** : Générer une synthèse des activités réalisées et des performances.
- **Graphiques interactifs** : Proposer des graphiques pour visualiser l'accomplissement des tâches et le respect des délais.
- **Analyse de répartition du temps** : Offrir des statistiques détaillées sur la gestion du temps.

### 3.5. Notifications et rappels

- **Rappels personnalisés** : Alerter les utilisateurs sur les échéances imminentes et les tâches importantes.
- **Suggestions proactives** : Fournir des recommandations pour optimiser la journée en fonction des priorités.

### 3.6. Support multi-langues

- **Interface multilingue** : Permettre aux utilisateurs de choisir la langue de l'application pour une meilleure accessibilité.
- **Reconnaissance vocale multilingue** : Supporter plusieurs langues pour la conversion vocale.

## 4. Technologies proposées

### 4.1. Reconnaissance vocale

- Google Speech-to-Text API.
- Whisper.

### 4.2. Traitement du langage naturel (NLP)

- OpenAI GPT.
- spaCy.
- Hugging Face Transformers.

### 4.3. Backend

- Node.js.
- Python avec FastAPI ou Django REST Framework.

### 4.4. Frontend

- Angular ou React.js pour la version web.
- Flutter/Dart pour la version mobile.

### 4.5. Base de données

- PostgreSQL ou MySQL pour les données structurées.
- MongoDB pour les données semi-structurées.

### 4.6. Visualisation des données

- Chart.js.
- ApexCharts.
- D3.js.

## 5. Déroulement du projet

### 5.1. Phase 1 : Conception

- **Analyse des besoins utilisateurs** : Identifier précisément les attentes et exigences des utilisateurs cibles à travers des ateliers et des questionnaires.
- **Conception des maquettes et wireframes** : Élaborer des prototypes visuels interactifs pour valider l'ergonomie et les fonctionnalités de l'application.

- **Validation des spécifications fonctionnelles et techniques :**

Documenter en détail les fonctionnalités prévues ainsi que l'architecture technique du projet, incluant les flux de données et les intégrations.

## 5.2. Phase 2 : Développement

- **Mise en place du backend et des bases de données :** Créer une architecture robuste pour gérer les données utilisateur et les processus serveur, en garantissant performance et sécurité.
- **Développement du frontend (web et mobile) :** Concevoir des interfaces utilisateurs intuitives et responsives avec une attention particulière à l'expérience utilisateur (UX).
- **Intégration des modules de reconnaissance vocale et NLP :** Déployer et configurer les API et bibliothèques nécessaires pour la conversion vocale et l'analyse linguistique.
- **Tests fonctionnels continus :** Vérifier chaque fonctionnalité implémentée à travers des tests automatisés et manuels pour s'assurer de la qualité.

## 5.3. Phase 3 : Tests et validation

- **Tests unitaires :** Vérifier individuellement chaque composant pour garantir leur bon fonctionnement.
- **Tests d'intégration :** S'assurer que les différents modules interagissent correctement entre eux.
- **Validation des performances :** Effectuer des tests de charge et de stress pour garantir la stabilité de l'application sous des conditions réelles.
- **Évaluation de la satisfaction utilisateur :** Récolter les retours d'un panel de testeurs sur l'utilisabilité et l'efficacité des fonctionnalités.

# 6. Livrables

## 6.1. Documentation fonctionnelle et technique :

Inclut une description complète des fonctionnalités, des choix technologiques, et de l'architecture logicielle. Comprend également les spécifications API et les diagrammes de flux.

## 6.2. Maquettes et wireframes validés :

Prototypes visuels interactifs validés par les parties prenantes, détaillant l'ergonomie et les flux utilisateurs.

## 6.3. Code source complet et structuré :

Inclut l'ensemble des modules de backend, de frontend, et des scripts de base de données, accompagné de commentaires détaillés.

## 6.4. Application fonctionnelle (version web et mobile) :

Logiciel prêt à être utilisé, intégrant toutes les fonctionnalités spécifiées.

## 6.5. Rapports de tests et validations :

Documentations détaillant les résultats des tests unitaires, fonctionnels, et de performance, avec les éventuelles améliorations apportées.

#### 6.6. Tableaux de bord et graphiques interactifs :

Modules de visualisation des données prêts à l'emploi, avec une présentation claire des performances et des statistiques.

## 7. Contraintes et exigences

#### 7.1. Compatibilité multi-plateformes :

L'application doit fonctionner sans heurts sur les plateformes web, Android et iOS, garantissant une expérience utilisateur uniforme.

#### 7.2. Performance et scalabilité :

Le système doit être capable de traiter des données volumineuses tout en maintenant une rapidité d'exécution et une disponibilité élevée.

#### 7.3. Sécurité des données :

Des mesures de sécurité robustes doivent être mises en œuvre, notamment le chiffrement des données sensibles, la protection contre les attaques courantes (SQL injection, XSS, etc.), et une gestion sécurisée des accès utilisateur.

#### 7.4. Conformité réglementaire :

Respect des lois et règlements en vigueur, comme le RGPD pour la protection des données personnelles.

#### 7.5. Accessibilité et ergonomie :

L'interface utilisateur doit être intuitive et accessible aux personnes ayant des besoins spécifiques, comme les options pour les malvoyants ou les outils de lecture vocale.

## 8. Planning estimé

Phase	Durée estimée
Conception	4 semaines
Développement	16 semaines
Tests et validation	4 semaines

## 9. Conclusion

Le projet "Plan Genie AI" a pour ambition de simplifier la gestion quotidienne des utilisateurs grâce à l'intégration d'une intelligence artificielle performante et accessible.

En respectant les contraintes et exigences définies, ce projet représente une opportunité significative pour les étudiants de développer des compétences avancées en IA, tout en répondant à des problématiques concrètes.

Ce cahier des charges constitue une feuille de route détaillée garantissant le succès du projet.