**Gabarit de proposition de projet**

**Cahier des Charges Simplifié – Nom de votre application**

**Partie 1 : Objectif du développement de votre application**

L'objectif principal de cette application est de [décrire l'utilité générale de l'application]. Cette application vise à [résoudre un problème spécifique / améliorer un processus / fournir un service unique] pour [public cible ou utilisateurs finaux].

**Partie 2 : Fonctionnalités principales**

Les fonctionnalités principales de l'application incluent :

1. **Fonctionnalité 1 :** [Description détaillée de la fonctionnalité 1]
2. **Fonctionnalité 2 :** [Description détaillée de la fonctionnalité 2]
3. **Fonctionnalité 3 :** [Description détaillée de la fonctionnalité 3]
4. **Fonctionnalité 4 :** [Description détaillée de la fonctionnalité 4]
5. **Fonctionnalité 5 :** [Description détaillée de la fonctionnalité 5]

**Partie 3 : Technologie utilisée pour la création de l’application et justification de votre choix**

Pour le développement de cette application, les technologies suivantes seront utilisées :

* **Langage de programmation :** [Nom du langage de programmation]
  + *Justification :* [Raison pour laquelle ce langage a été choisi, ex. facilité d'utilisation, performance, popularité]
* **Frameworks :** [Nom du/des frameworks]
  + *Justification :* [Raison pour laquelle ce/ces frameworks ont été choisis, ex. support communautaire, fonctionnalités intégrées]
* **Base de données :** [Nom de la base de données]
  + *Justification :* [Raison pour laquelle cette base de données a été choisie, ex. scalabilité, compatibilité avec le langage/framework]

**Partie 4 : Calendrier de répartition des tâches semaine par semaine (240 heures)**

Le travail sera réparti comme suit :

* **Semaine 1 (35 heures) :** Conception des fonctionnalités principales
  + *Activités :* Utilisation de diagrammes de séquence pour définir les interactions principales de l'application.
* **Semaines 2-3 (80 heures) :** Préparation et développement des interfaces de l’application et préparation de la base de données
  + *Activités :* Création des maquettes des interfaces utilisateurs, développement de l'interface utilisateur, configuration de la base de données.
* **Semaines 4-5 (70 heures) :** Implémentation des fonctionnalités de l’application
  + *Activités :* Codage des fonctionnalités principales, intégration des interfaces avec la logique métier.
* **Semaine 6 (35 heures) :** Test et validation, préparation de la documentation
  + *Activités :* Réalisation des tests unitaires et fonctionnels, correction des bugs, rédaction de la documentation utilisateur et technique.
* **Semaine 7 (20 heures) :** Révisions finales et déploiement
  + *Activités :* Revues de code, optimisations finales, préparation pour le déploiement.
* **Semaine 8 (20 heures) :** Déploiement et feedback
  + *Activités :* Déploiement de l'application, collecte de feedback utilisateur, ajustements finaux.

**Exemple de gabarit de proposition de projet**

**Cahier des Charges Simplifié – [Nom de l'application]**

### ****Partie 1 : Objectif du développement de votre application****

L'objectif principal de cette application est de [décrire l'utilité générale de l'application]. Cette application vise à [résoudre un problème spécifique / améliorer un processus / fournir un service unique] pour [public cible ou utilisateurs finaux]. En particulier, l'application permettra de :

* [Objectif spécifique 1]
* [Objectif spécifique 2]
* [Objectif spécifique 3]

Exemple : L'objectif de cette application est de faciliter la gestion des tâches et des projets pour les petites entreprises. Elle permettra aux utilisateurs de créer, attribuer et suivre l'avancement des tâches en temps réel, d'améliorer la collaboration entre les équipes et d'optimiser la productivité.

### ****Partie 2 : Fonctionnalités principales****

Les fonctionnalités principales de l'application incluent :

1. **Gestion des utilisateurs :**
   * Création, modification et suppression des comptes utilisateurs.
   * Authentification et autorisation sécurisées (inscription, connexion, gestion des rôles).
2. **Gestion des tâches :**
   * Création, modification et suppression de tâches.
   * Attribution des tâches aux utilisateurs.
   * Suivi de l'avancement des tâches avec des statuts (à faire, en cours, terminé).
3. **Gestion des projets :**
   * Création, modification et suppression de projets.
   * Association des tâches aux projets.
   * Visualisation de l'état d'avancement des projets.
4. **Collaboration en temps réel :**
   * Messagerie instantanée entre les membres de l'équipe.
   * Notifications en temps réel pour les mises à jour des tâches et des projets.
5. **Rapports et analyses :**
   * Génération de rapports sur l'avancement des tâches et des projets.
   * Statistiques sur la productivité des utilisateurs et des équipes.

### ****Partie 3 : Technologie utilisée pour la création de l’application et justification de votre choix****

Pour le développement de cette application, les technologies suivantes seront utilisées :

* **Langage de programmation :** JavaScript (Node.js pour le backend, React pour le frontend)
  + Justification : JavaScript est un langage populaire et polyvalent, idéal pour le développement full-stack. Node.js permet un développement backend rapide et efficace, tandis que React offre une expérience utilisateur dynamique et réactive.
* **Frameworks :**
  + **Backend :** Express.js
    - Justification : Express.js est un framework minimaliste et flexible pour Node.js, facilitant le développement de l'API RESTful.
  + **Frontend :** React
    - Justification : React est une bibliothèque de choix pour le développement d'interfaces utilisateur interactives et réactives. Sa composabilité et sa communauté active en font un choix idéal pour ce projet.
* **Base de données :** MongoDB
  + Justification : MongoDB est une base de données NoSQL flexible et évolutive, parfaite pour gérer des données non structurées et des changements fréquents dans les schémas de données.
* **Outils de développement :**
  + **Environnement de développement :** Visual Studio Code
    - Justification : Visual Studio Code est un éditeur de code puissant et extensible, largement utilisé dans l'industrie du développement web.
  + **Gestion de version :** Git
    - Justification : Git est le système de contrôle de version le plus utilisé, offrant une gestion efficace des versions du code et facilitant la collaboration entre les développeurs.

### ****Partie 4 : Calendrier de répartition des tâches semaine par semaine (240 heures)****

Le travail sera réparti comme suit :

#### **Semaine 1 (35 heures) : Conception des fonctionnalités principales**

* **Activités :**
  + Analyse des besoins et spécifications fonctionnelles.
  + Création des diagrammes de séquence pour définir les interactions principales de l'application.
  + Élaboration de l'architecture de l'application.

#### **Semaines 2-3 (80 heures) : Préparation et développement des interfaces de l’application et préparation de la base de données**

* **Activités :**
  + Conception des maquettes des interfaces utilisateurs (wireframes).
  + Développement des composants de l'interface utilisateur avec React.
  + Configuration et initialisation de la base de données MongoDB.
  + Création des schémas de données et des modèles.

#### **Semaines 4-5 (70 heures) : Implémentation des fonctionnalités de l’application**

* **Activités :**
  + Développement des fonctionnalités backend avec Node.js et Express.js.
  + Intégration des interfaces utilisateurs avec le backend (API RESTful).
  + Implémentation de l'authentification et de l'autorisation des utilisateurs.
  + Développement des fonctionnalités de gestion des tâches et des projets.

#### **Semaine 6 (35 heures) : Test et validation, préparation de la documentation**

* **Activités :**
  + Réalisation des tests unitaires et fonctionnels pour chaque fonctionnalité.
  + Correction des bugs identifiés durant les tests.
  + Rédaction de la documentation utilisateur et technique.

#### **Semaine 7 (20 heures) : Révisions finales et déploiement**

* **Activités :**
  + Revues de code pour assurer la qualité et la conformité aux normes de codage.
  + Optimisations finales des performances de l'application.
  + Préparation de l'environnement de production.

#### **Semaine 8 (20 heures) : Déploiement et feedback**

* **Activités :**
  + Déploiement de l'application sur un serveur de production.
  + Collecte de feedback des utilisateurs finaux.
  + Ajustements finaux et corrections basées sur le feedback.