Phase 1:Analyse des besoins

Oui, absolument, vous devriez inclure la **Phase 1 : Analyse des besoins** dans votre rapport, car c'est une étape essentielle pour montrer que vous avez bien défini les objectifs et les attentes avant de commencer le développement. Voici comment vous pouvez structurer cette section dans votre rapport :

Exemple d'ajout de la Phase 1 dans le rapport :

1. Phase 1 : Analyse des besoins

Objectif : Comprendre les besoins des utilisateurs et définir les fonctionnalités principales de l'application TaskFlow.

1.1. Objectifs de TaskFlow

- Permettre aux utilisateurs de gérer efficacement leurs tâches quotidiennes.
- Offrir une interface intuitive et simple d’utilisation.
- Ajouter des fonctionnalités comme la gestion des priorités, les notifications, et la synchronisation des données.

1.2. Public cible

- **Étudiants** : Pour gérer leurs travaux scolaires, révisions et projets.
- **Professionnels** : Pour organiser leurs tâches professionnelles et rendez-vous.
- **Individus organisés** : Toute personne cherchant à améliorer sa productivité.

1.3. Fonctionnalités principales

- **Ajout de tâches** : Création de tâches avec un titre, une description, une date d’échéance, une priorité et une catégorie.
- **Modification et suppression** : Gestion des tâches existantes.
- **Notifications** : Rappels automatiques pour les échéances.
- **Catégorisation** : Classement des tâches par catégories personnalisées.

- **Recherche et tri** : Filtrer et trier les tâches par priorité, date ou catégorie.

1.4. Livrables de cette phase

- Liste des fonctionnalités définies.
- Personas utilisateurs (facultatif mais recommandé).
- Wireframes des interfaces principales (facultatif mais utile pour enrichir votre rapport).

Pourquoi inclure cette phase dans le rapport ?

1. **Démontrer votre approche méthodique** : Montrez que vous avez analysé les besoins avant de coder.
2. **Valoriser la compréhension des utilisateurs** : Cela met en avant que vous avez pensé aux utilisateurs finaux.
3. **Structurer votre rapport** : Chaque phase correspond à une étape logique du projet.

---

1. Diagramme de séquence

Qu'est-ce que c'est ?

Un diagramme de séquence montre l'ordre des interactions entre les utilisateurs, le système, et ses composants. Il illustre **comment** et **dans quel ordre** les actions se déroulent.

Pourquoi est-il utile ?

Pour **TaskFlow**, un diagramme de séquence peut :

- Décrire le flux des événements pour des scénarios clés comme :
  - Connexion d’un utilisateur.
  - Ajout/modification d’une tâche.
  - Consultation des tâches depuis le tableau de bord.

- Clarifier les interactions entre :
  - L’utilisateur.
  - L’interface utilisateur (Frontend).
  - Le backend (API).
  - La base de données.

Exemple : Ajout d’une tâche

Voici un résumé des étapes dans un diagramme de séquence :

1. L'utilisateur clique sur **"Ajouter une tâche"**.
2. Le front-end envoie une requête au backend via une API (`POST /tasks`).
3. Le backend enregistre la tâche dans MongoDB.
4. Le backend renvoie une réponse au front-end.
5. Le front-end met à jour l’interface utilisateur avec la nouvelle tâche.

Représentation graphique :

- Acteurs : Utilisateur, Frontend, Backend, Base de données.
- Messages échangés : Actions comme `POST /tasks`, réponses (`200 OK`).

---

2. Diagramme UML URL (ou des routes API)

Qu'est-ce que c'est ?

Un **diagramme UML pour les routes API** montre comment chaque URL est associée aux opérations du backend. Cela donne une vision claire des points d’accès RESTful.

Pourquoi est-il utile ?

Pour **TaskFlow**, un tel diagramme peut :

- Montrer toutes les routes disponibles :
  - `GET /tasks` pour récupérer les tâches.
  - `POST /tasks` pour ajouter une tâche.

    - `PUT /tasks/:id` pour modifier une tâche.
    - `DELETE /tasks/:id` pour supprimer une tâche.
- Décrire les méthodes HTTP, les paramètres, et les réponses.
- Faciliter la compréhension pour les développeurs.

Exemple pour TaskFlow :

Routes principales :

Méthode	URL	Action
GET	`/tasks`	Récupérer toutes les tâches.
POST	`/tasks`	Ajouter une nouvelle tâche.
PUT	`/tasks/:id`	Modifier une tâche existante.
DELETE	`/tasks/:id`	Supprimer une tâche.

Graphique UML simplifié :

- Un rectangle pour chaque route.
- Liens entre les composants (Frontend → Backend → Base de données).
