# Étape 1 : Définition des objectifs du projet (1 à 2 semaines)

**Objectifs du projet :**

L'objectif principal de SESAME Insights est d'analyser les résultats scolaires des étudiants du programme SESAME. Plus précisément, les objectifs du projet sont les suivants :

1. **Collecte de données :** Collecter et stocker les données des résultats scolaires des étudiants dans une base de données Oracle.
2. **Analyse des résultats :** Offrir une plateforme permettant d'analyser les résultats scolaires et de générer des informations précieuses sur les performances académiques des étudiants.
3. **Rapports et statistiques :** Générer des rapports détaillés, des graphiques et des statistiques pertinents pour évaluer les résultats scolaires des étudiants.
4. **Aide à la prise de décision :** Fournir aux responsables pédagogiques et aux enseignants des données utiles pour prendre des décisions éclairées en vue d'améliorer l'expérience d'apprentissage et le suivi des étudiants.
5. **Mise à jour régulière :** Permettre la mise à jour régulière des résultats scolaires dans la base de données Oracle pour suivre l'évolution des performances des étudiants au fil du temps.

**Fonctionnalités de base :**

Pour atteindre ces objectifs, l'application SESAME Insights devra proposer les fonctionnalités suivantes :

1. **Authentification et gestion des utilisateurs :**
   * Authentification des utilisateurs en tant qu'administrateur ou utilisateur simple.
   * Gestion des comptes utilisateurs, y compris la création, la modification et la suppression de comptes.
2. **Collecte de données :**
   * Importation de données de résultats scolaires dans la base de données Oracle.
   * Mise à jour régulière des données.
3. **Analyse des résultats :**
   * Calcul de la note moyenne générale des contrôles continus pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
   * Calcul de la note moyenne générale des contrôles terminaux pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
   * Calcul de la note moyenne générale pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
   * Classement général pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
4. **Génération de rapports et statistiques :**
   * Création de rapports détaillés sur les performances des étudiants.
   * Génération de graphiques et de statistiques pour visualiser les données.
5. **Interface utilisateur conviviale :**
   * Création d'une interface intuitive permettant aux utilisateurs d'accéder facilement aux fonctionnalités d'analyse.
6. **Mise à jour de la base de données :**
   * Automatisation de la mise à jour régulière des données de résultats scolaires dans la base de données Oracle.
7. **Sécurité et gestion des autorisations :**
   * Assurer la sécurité des données et la gestion des autorisations pour les administrateurs et les utilisateurs simples.

**Exigences de base :**

Les exigences de base pour le projet SESAME Insights sont les suivantes :

1. **Les technologies :**
   * Utilisation d'Angular pour la création de l'interface utilisateur.
   * Utilisation de Node.js pour le développement de l'API.
   * Utilisation d'Oracle Database pour le stockage des données.
2. **Interface utilisateur :**
   * Interface conviviale et intuitive pour une expérience utilisateur agréable.
3. **Sécurité :**
   * Mise en place de mécanismes de sécurité pour protéger les données sensibles.
4. **Mises à jour automatiques :**
   * Automatisation de la mise à jour régulière des données de résultats scolaires.
5. **Authentification :**
   * Authentification des utilisateurs avec des rôles (administrateur et utilisateur simple).
6. **Rapports et visualisations :**
   * Génération de rapports, de graphiques et de statistiques pour l'analyse des données.
7. **Performance :**
   * Assurer des performances optimales lors de l’analyse de grandes quantités de données.
8. **Documentation :**
   * Documentation détaillée de l'application, y compris les instructions d'installation et d'utilisation.
9. **Essais :**

Effectuer des tests unitaires et d'intégration pour garantir la fiabilité de l'application.

* Identifiez clairement les objectifs de votre projet.
* Définissez les fonctionnalités et les exigences de base de votre application.
* Établissez un budget et un calendrier préliminaire.

## Réponses à Etape 1:

**Objectifs du projet :**

L'objectif principal de SESAME Insights est d'analyser les résultats scolaires des étudiants du programme SESAME. Plus précisément, les objectifs du projet sont les suivants :

1. **Collecte de données :** Collecter et stocker les données des résultats scolaires des étudiants dans une base de données Oracle.
2. **Analyse des résultats :** Offrir une plateforme permettant d'analyser les résultats scolaires et de générer des informations précieuses sur les performances académiques des étudiants.
3. **Rapports et statistiques :** Générer des rapports détaillés, des graphiques et des statistiques pertinents pour évaluer les résultats scolaires des étudiants.
4. **Aide à la prise de décision :** Fournir aux responsables pédagogiques et aux enseignants des données utiles pour prendre des décisions éclairées en vue d'améliorer l'expérience d'apprentissage et le suivi des étudiants.
5. **Mise à jour régulière :** Permettre la mise à jour régulière des résultats scolaires dans la base de données Oracle pour suivre l'évolution des performances des étudiants au fil du temps.

**Fonctionnalités de base :**

Pour atteindre ces objectifs, l'application SESAME Insights devra proposer les fonctionnalités suivantes :

1. **Authentification et gestion des utilisateurs :**
   * Authentification des utilisateurs en tant qu'administrateur ou utilisateur simple.
   * Gestion des comptes utilisateurs, y compris la création, la modification et la suppression de comptes.
2. **Collecte de données :**
   * Importation de données de résultats scolaires dans la base de données Oracle.
   * Mise à jour régulière des données.
3. **Analyse des résultats :**
   * Calcul de la note moyenne générale des contrôles continus pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
   * Calcul de la note moyenne générale des contrôles terminaux pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
   * Calcul de la note moyenne générale pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
   * Classement général pour chaque étudiant de chaque mention (L ou S) et chacun des 2 semestres (S1 et S2).
4. **Génération de rapports et statistiques :**
   * Création de rapports détaillés sur les performances des étudiants.
   * Génération de graphiques et de statistiques pour visualiser les données.
5. **Interface utilisateur conviviale :**
   * Création d'une interface intuitive permettant aux utilisateurs d'accéder facilement aux fonctionnalités d'analyse.
6. **Mise à jour de la base de données :**
   * Automatisation de la mise à jour régulière des données de résultats scolaires dans la base de données Oracle.
7. **Sécurité et gestion des autorisations :**
   * Assurer la sécurité des données et la gestion des autorisations pour les administrateurs et les utilisateurs simples.

**Exigences de base :**

Les exigences de base pour le projet SESAME Insights sont les suivantes :

1. **Les technologies :**
   * Utilisation d'Angular pour la création de l'interface utilisateur.
   * Utilisation de Node.js pour le développement de l'API.
   * Utilisation d'Oracle Database pour le stockage des données.
2. **Interface utilisateur :**
   * Interface conviviale et intuitive pour une expérience utilisateur agréable.
3. **Sécurité :**
   * Mise en place de mécanismes de sécurité pour protéger les données sensibles.
4. **Mises à jour automatiques :**
   * Automatisation de la mise à jour régulière des données de résultats scolaires.
5. **Authentification :**
   * Authentification des utilisateurs avec des rôles (administrateur et utilisateur simple).
6. **Rapports et visualisations :**
   * Génération de rapports, de graphiques et de statistiques pour l'analyse des données.
7. **Performance :**
   * Assurer des performances optimales lors de l’analyse de grandes quantités de données.
8. **Documentation :**
   * Documentation détaillée de l'application, y compris les instructions d'installation et d'utilisation.
9. **Essais :**

Effectuer des tests unitaires et d'intégration pour garantir la fiabilité de l'application.

# Étape 2 : Recherche et apprentissage (2 à 4 semaines)

* Apprenez les bases d'Angular, Node.js et Oracle si vous n'êtes pas déjà familier avec ces technologies.
* Explorez les meilleures pratiques en matière de développement web, d'architecture de base de données, de sécurité, etc.
* Découvrez les ressources en ligne, les cours ou les livres qui vous aideront à maîtriser ces technologies.

# Étape 3 : Planification détaillée (2 semaines)

* Décomposez votre projet en tâches et en fonctionnalités plus petites et plus gérables.
* Créez une liste des dépendances entre les tâches.
* Établissez un plan de développement détaillé avec des dates de début et de fin pour chaque tâche.

## Réponses à Etape 3:

Objectif de l'étape : Décomposer le projet en tâches et en fonctionnalités plus petites et plus gérables, établir un plan de développement détaillé avec des dates de début et de fin pour chaque tâche.

* 1. Définir les tâches principales : Commencez par identifier les grandes tâches ou les fonctionnalités principales de votre projet. Celles-ci pourraient inclure la création de l'interface utilisateur, le développement de l'API, la mise en place de la sécurité, la génération de rapports, etc.

### Réponses à 3-1:

**Création de l'interface utilisateur :**

* + 1. Conception de l'interface utilisateur en collaboration avec les parties prenantes.
    2. Création des maquettes (wireframes) pour chaque page ou écran de l'application.
    3. Développement des composants Angular pour l'interface utilisateur en se basant sur les maquettes.
    4. Mise en place de la navigation entre les pages.
    5. Intégration de formulaires pour la saisie de données.
    6. Gestion des interactions utilisateur (clics, soumissions de formulaire, etc.).
    7. Validation et tests de l'interface utilisateur.

**Développement de l'API :**

* + 1. 3-1-8 Création du serveur Node.js avec Express.js.
    2. Définition des routes pour gérer les demandes HTTP liées à l'API.
    3. Intégration du pilote Oracle pour interagir avec la base de données.
    4. Développement des fonctionnalités d'accès aux données (lecture, écriture, mise à jour, suppression).
    5. Validation et tests de l'API (tests unitaires et tests d'intégration).
    6. Documentation de l'API, y compris les points d'accès et les formats de données.

**Mise en place de la sécurité :**

* + 1. Mise en place de l'authentification des utilisateurs (inscription, connexion, déconnexion).
    2. Gestion des sessions utilisateur et des jetons d'authentification.
    3. Gestion des autorisations d'accès aux fonctionnalités en fonction du rôle de l'utilisateur (administrateur ou utilisateur simple).
    4. Validation des données entrantes pour prévenir les attaques.
    5. Configuration de mécanismes de sécurité pour protéger les données sensibles.

**Génération de rapports :**

* + 1. Conception des rapports et des visualisations à générer.
    2. Développement des fonctionnalités pour générer des rapports basés sur les données.
    3. Création de graphiques (histogrammes, diagrammes circulaires, etc.) pour visualiser les données.
    4. Affichage de statistiques pertinentes pour évaluer les résultats scolaires.
    5. Intégration des rapports dans l'interface utilisateur.
    6. Tests des fonctionnalités de génération de rapports.
  1. Décomposition des tâches : Pour chaque grande tâche, décomposez-la en tâches plus petites et plus gérables. Par exemple, si vous avez la tâche "Création de l'interface utilisateur", vous pourriez la décomposer en sous-tâches telles que "Conception de la page d'accueil", "Création du formulaire de connexion", "Mise en place de la navigation", etc.

### Réponses 3-2 :

**Développement de l'Interface Utilisateur (UI) :**

* + 1. Conception de l'UI :
  + Réalisation de maquettes (wireframes) pour chaque écran de l'application, y compris la page d'accueil, le tableau de bord, la page de détails de l'étudiant, etc.
  + Conception de la structure de navigation de l'application.
    1. Création des composants Angular :
  + Création des composants Angular spécifiques à chaque écran ou à chaque section de l'application, tels que HomeComponent, DashboardComponent, StudentDetailsComponent, etc.
  + Création de composants réutilisables pour les éléments d'interface communs, tels que HeaderComponent, FooterComponent, etc.
    1. Mise en place de la navigation :
  + Configuration de la navigation entre les différentes pages de l'application en utilisant le module de routage Angular.
  + Création de routes pour chaque écran de l'application.
    1. Gestion des formulaires :
  + Création de formulaires pour la saisie de données, par exemple, la création d'un formulaire de connexion avec les champs "Nom d'utilisateur" et "Mot de passe".
  + Ajout de validation des données dans les formulaires.
    1. Styling et CSS :
  + Application de styles CSS pour chaque composant et écran de l'application.
  + Utilisation de frameworks CSS tels que Angular Material ou Bootstrap pour une apparence cohérente.
  + Personnalisation des styles pour correspondre à la charte graphique de l'application.
    1. Gestion des interactions utilisateur :
  + Gestion des interactions utilisateur telles que la navigation entre les écrans, la soumission de formulaires, le tri des données, etc.
  + Gestion des événements (clics, survols de souris, etc.) pour améliorer l'expérience utilisateur.
    1. Intégration des graphiques et des visualisations :
  + Intégration de graphiques et de visualisations pour afficher les données de manière visuelle, par exemple, la création de graphiques à barres pour les résultats scolaires.
  + Personnalisation des graphiques en fonction des besoins.
    1. Tests de l'UI :
  + Réalisation de tests d'interface utilisateur pour s'assurer que chaque écran fonctionne correctement et est conforme aux maquettes.
  + Tests de réactivité pour garantir que l'UI est adaptée aux différents appareils et tailles d'écran.
    1. Internationalisation et localisation :
  + Mise en place de fonctionnalités d'internationalisation (i18n) pour prendre en charge plusieurs langues si nécessaire.
  + Gestion des traductions des éléments d'interface pour chaque langue prise en charge.
    1. Optimisation des performances :
  + Optimisation de la vitesse de chargement de l'UI en minimisant les requêtes réseau et en utilisant la mise en cache si nécessaire.
  + Gestion de la réactivité de l'UI pour assurer une expérience utilisateur fluide.
    1. Accessibilité :
  + Vérification de l'accessibilité de l'UI pour les utilisateurs ayant des besoins spécifiques, conformément aux normes WCAG.
    1. Documentation de l'UI :
  + Documentation des éléments d'interface, des composants réutilisables et des modèles de données.
  + Création de guides d'utilisation pour les utilisateurs finaux.
    1. Intégration avec l'API :
  + Intégration des fonctionnalités de l'API pour récupérer et afficher les données dans l'UI.

Ces sous-tâches détaillées pour le développement de l'UI devraient vous aider à mieux organiser le processus de création de l'interface utilisateur de votre application SESAME Insights. Chaque sous-tâche représente une étape spécifique du développement de l'UI, ce qui facilitera sa mise en œuvre et son suivi.

* 1. Estimation du temps : Estimez la durée nécessaire pour accomplir chaque tâche. Cela peut être basé sur votre expérience passée ou sur des estimations approximatives. Assurez-vous d'être réaliste dans vos estimations.
  2. Priorisation : Identifiez les tâches critiques qui doivent être accomplies en premier. Celles-ci peuvent être les dépendances de base nécessaires pour démarrer d'autres parties du projet.
  3. Dépendances entre les tâches : Identifiez les dépendances entre les tâches. Certaines tâches doivent être terminées avant que d'autres ne puissent commencer. Par exemple, vous devez d'abord avoir une base de données fonctionnelle avant de pouvoir développer des fonctionnalités d'importation de données.
  4. Création d'un diagramme de Gantt : Utilisez un diagramme de Gantt ou un outil de gestion de projet pour organiser visuellement les tâches, leurs dépendances et leurs durées. Cela vous permettra de voir le calendrier général du projet.
  5. Attribution des ressources : Si vous travaillez en équipe, attribuez des responsables à chaque tâche. Assurez-vous que chaque membre de l'équipe sait quelles sont ses responsabilités.
  6. Définition des jalons : Identifiez des jalons importants dans le projet, tels que des points de contrôle pour évaluer les progrès. Les jalons vous aident à suivre votre avancement.
  7. Création d'un calendrier : Établissez un calendrier avec des dates de début et de fin pour chaque tâche. Assurez-vous que le calendrier est réaliste et tient compte des contraintes de temps.
  8. Révision et ajustement : Une fois que vous avez établi votre plan de développement, révisez-le pour vous assurer qu'il est complet et réaliste. N'hésitez pas à ajuster les dates ou les priorités si nécessaire.

# Étape 4 : Configuration de l'environnement de développement (1 à 2 semaines)

* Installez Angular CLI, Node.js et les dépendances nécessaires sur votre ordinateur.
* Configurez un environnement de développement pour Oracle Database.

# Étape 5 : Création de l'application Angular (4 à 6 semaines)

* Commencez à créer l'interface utilisateur de votre application Angular en fonction des spécifications définies dans l'étape 3.
* Créer des composants, des services et des modèles Angular.
* Utilisez des appels HTTP pour interagir avec l'API Node.js que vous créez par la suite.

# Étape 6 : Développement de l'API Node.js (6 à 8 semaines)

* Développez l'API Node.js pour gérer les demandes de l'application Angular.
* Intégrez la base de données Oracle pour stocker et récupérer des données.
* Mettez en place des mécanismes d'authentification et de sécurité.

# Étape 7 : Tests et débogage (4 à 6 semaines)

* Effectuez des tests unitaires et d'intégration pour chaque composant de l'application.
* Identifiez et corrigez les erreurs et les problèmes de performance.
* Assurez-vous que l'application fonctionne correctement sur les différents navigateurs.

# Étape 8 : Documentation (2 semaines)

* Rédigez une documentation détaillée pour votre application, y compris les instructions d'installation, d'utilisation et de maintenance.
* Documentez le code source pour faciliter la collaboration future.

# Étape 9 : Déploiement (2 à 4 semaines)

* Déployez votre application Angular sur un serveur web accessible depuis Internet.
* Configurez votre serveur Oracle pour autoriser les connexions depuis l'application Node.js.
* Effectuez des tests de déploiement pour vous assurer que tout fonctionne correctement en ligne.

# Étape 10 : Suivi et maintenance (suite)

* Surveillez l'application en production et corrigez rapidement les problèmes qui surviennent.
* Prévoyez des mises à jour et des améliorations futures en fonction des retours des utilisateurs et de l'évolution de

**Étape 3 : Planification détaillée (2 semaines)**

* Décomposez votre projet en tâches et en fonctionnalités plus petites et plus gérables.
* Créez une liste des dépendances entre les tâches.
* Établissez un plan de développement détaillé avec des dates de début et de fin pour chaque tâche.

Réponses :

**Objectif de l'étape :** Décomposer le projet en tâches et en fonctionnalités plus petites et plus gérables, établir un plan de développement détaillé avec des dates de début et de fin pour chaque tâche.

**1. Définir les tâches principales :** Commencez par identifier les grandes tâches ou les fonctionnalités principales de votre projet. Celles-ci pourraient inclure la création de l'interface utilisateur, le développement de l'API, la mise en place de la sécurité, la génération de rapports, etc.

**2. Décomposition des tâches :** Pour chaque grande tâche, décomposez-la en tâches plus petites et plus gérables. Par exemple, si vous avez la tâche "Création de l'interface utilisateur", vous pourriez la décomposer en sous-tâches telles que "Conception de la page d'accueil", "Création du formulaire de connexion", "Mise en place de la navigation", etc.

**3. Estimation du temps :** Estimez la durée nécessaire pour accomplir chaque tâche. Cela peut être basé sur votre expérience passée ou sur des estimations approximatives. Assurez-vous d'être réaliste dans vos estimations.

**4. Priorisation :** Identifiez les tâches critiques qui doivent être accomplies en premier. Celles-ci peuvent être les dépendances de base nécessaires pour démarrer d'autres parties du projet.

**5. Dépendances entre les tâches :** Identifiez les dépendances entre les tâches. Certaines tâches doivent être terminées avant que d'autres ne puissent commencer. Par exemple, vous devez d'abord avoir une base de données fonctionnelle avant de pouvoir développer des fonctionnalités d'importation de données.

**6. Création d'un diagramme de Gantt :** Utilisez un diagramme de Gantt ou un outil de gestion de projet pour organiser visuellement les tâches, leurs dépendances et leurs durées. Cela vous permettra de voir le calendrier général du projet.

**7. Attribution des ressources :** Si vous travaillez en équipe, attribuez des responsables à chaque tâche. Assurez-vous que chaque membre de l'équipe sait quelles sont ses responsabilités.

**8. Définition des jalons :** Identifiez des jalons importants dans le projet, tels que des points de contrôle pour évaluer les progrès. Les jalons vous aident à suivre votre avancement.

**9. Création d'un calendrier :** Établissez un calendrier avec des dates de début et de fin pour chaque tâche. Assurez-vous que le calendrier est réaliste et tient compte des contraintes de temps.

**10. Révision et ajustement :** Une fois que vous avez établi votre plan de développement, révisez-le pour vous assurer qu'il est complet et réaliste. N'hésitez pas à ajuster les dates ou les priorités si nécessaire.