

Cahier des Charges

Academy Stats

Candice Giami et Thanina Benmammar

GIT : <https://github.com/Fjsksksk/Academy-Stats/>

Le Projet

Academy Stats est un projet destiné aux étudiants, enseignants et toute personne intéressée par les données sur l'éducation. Ce projet prendra la forme d'un site web accessible à tous, utilisant des données en open data disponibles sur la plateforme [Data.gouv.fr](https://data.gouv.fr/). Le but est de générer automatiquement des statistiques et rapports interactifs en fonction des filtres spécifiés par l'utilisateur, afin de répondre précisément à ses besoins.

Ces statistiques seront visualisées sous la forme de rapports interactifs, notamment via des intégrations avec Power BI ou d'autres outils de visualisation adaptés. Bien que nous nous spécialisons dans le domaine de l'éducation, ce projet est conçu de manière flexible pour être adapté à d'autres types de données, permettant ainsi la création de sites web sur le même principe.

Fonctionnalités

- **Système de filtrage avancé** : Les utilisateurs pourront affiner les résultats en fonction de plusieurs critères (niveau d'éducation, région, année, etc.).
- **Menu thématique** : Un menu proposant différents domaines liés à l'éducation (scolarité, université, formation continue, etc.).
- **Génération automatique de rapports interactifs** : Les rapports seront créés dynamiquement en fonction des données filtrées et pourront être visualisés sous forme de graphiques interactifs (Power BI, ou des alternatives open source).
- **Téléchargement et partage des rapports** : Les utilisateurs pourront télécharger les *rapports* en format PDF ou encore partager des liens vers les rapports interactifs.
- **Moteur de recherche** : Un moteur de recherche pour accéder rapidement aux statistiques précises en fonction de mots-clés spécifiques.

Contraintes

- **Dépendance aux données open data** : La qualité et la disponibilité des statistiques dépendent des données publiées en open data sur des plateformes comme Data.gouv.fr.
- **Limitation d'outils payants** : Le projet ne prévoit pas l'utilisation de logiciels ou licences payants (tels que Power BI Pro). Nous devons nous tourner vers des outils gratuits ou open source lorsque cela est possible.

- **Complexité des données** : Les jeux de données en open data peuvent être complexes à manipuler, ce qui nécessitera une architecture de traitement des données robuste.
- **Durée du Projet** : Le développement, la collecte et le traitement des données prendront du temps en raison de la complexité du projet

Technologies et outils

- **Langages** : Le développement du site web se fera principalement avec HTML5, CSS3, JavaScript pour la partie frontend, et Python pour la partie backend.
- **Bases de données** : Utilisation de MySQL pour stocker les données prétraitées.
- **Outils de visualisation** : Power BI pour la visualisation, si possible avec la version gratuite, ou des alternatives comme Google Data Studio, Grafana, ou Chart.js pour les graphiques.

Tâches

1. Réaliser la charte graphique du site web

- **Difficulté** : Facile
- **Durée estimée** : 1 jour
- **Détails** : Cette tâche consiste à concevoir l'identité visuelle du site web. Il est crucial de définir un ensemble cohérent de couleurs, de typographies, et d'éléments graphiques (boutons, icônes, etc.) qui vont structurer l'aspect esthétique du site. La charte graphique doit garantir une lisibilité optimale, tout en reflétant les valeurs du projet.
- **Étapes** :
 1. Choisir une palette de couleurs principale et secondaire en accord avec le thème du site.
 2. Sélectionner des typographies lisibles et esthétiquement adaptées.
 3. Concevoir des éléments graphiques comme les icônes.
 4. Valider l'ensemble graphique avec l'équipe projet avant implémentation.

2. Réaliser une maquette du site web

- **Difficulté** : Facile
 - **Durée estimée** : 15 jours
 - **Détails** : Cette tâche vise à concevoir une maquette détaillée de chaque page du site. Il s'agit de structurer visuellement le site, en anticipant les interactions utilisateur et l'organisation des données sur chaque page. La maquette sert de guide pour le développement du site.
 - **Étapes** :
 1. Définir la structure de la page d'accueil (menu de navigation, sections principales).
 2. Concevoir les pages de statistiques avec un design clair et facile à interpréter.
 3. Créer les pages avec les filtres interactifs et les options de recherche.
-

3. Récupérer les données intéressantes en open data

- **Difficulté** : Moyenne
 - **Durée estimée** : 15 jours
 - **Détails** : Cette tâche consiste à rechercher et télécharger les jeux de données nécessaires pour alimenter le projet. Les plateformes comme [Data.gouv.fr](https://data.gouv.fr) seront utilisées pour identifier des jeux de données relatifs à l'éducation en France. Il est important de vérifier que les données sont actuelles, complètes et qu'elles répondent aux besoins du projet.
 - **Étapes** :
 1. Identifier les sources de données open data pertinentes.
 2. Vérifier la disponibilité des données sous des formats compatibles (CSV, JSON, etc.).
 3. Téléchargez les jeux de données.
-

4. Prétraiter et nettoyer les données

- **Difficulté** : Moyenne
 - **Durée estimée** : 15 jours
 - **Détails** : Avant d'exploiter les données récupérées, il est essentiel de les nettoyer et de les organiser. Cette tâche implique la détection et la suppression des anomalies (données manquantes, valeurs aberrantes), ainsi que le formatage des données pour qu'elles soient prêtes à être utilisées dans le cadre des statistiques et des rapports.
 - **Étapes** :
 1. Analyser les données pour détecter les erreurs et les incohérences.
 2. Supprimer ou corriger les données manquantes et les valeurs aberrantes.
 3. Formater les données pour qu'elles puissent être intégrées dans des systèmes d'analyse comme Power BI.
-

5. Développer le système de génération de rapports Power BI

- **Difficulté** : Très difficile
 - **Durée estimée** : 48 jours
 - **Détails** : Le cœur du projet réside dans la génération automatique de rapports statistiques via Power BI. Cette tâche consiste à configurer Power BI pour qu'il traite les données nettoyées, puis à créer des rapports interactifs qui reflètent les résultats souhaités. Les rapports doivent être personnalisables via des filtres dynamiques.
 - **Étapes** :
 1. Importer les données nettoyées dans Power BI.
 2. Créer des tableaux de bord et des graphiques dynamiques en fonction des critères définis (dates, localisation, etc.).
 3. Ajouter des filtres interactifs permettant aux utilisateurs de personnaliser les rapports.
 4. Tester et valider les rapports pour assurer leur exactitude.
-

6. Développer le site web (frontend et backend)

- **Difficulté** : Difficile
 - **Durée estimée** : 64 jours
 - **Détails** : Cette tâche implique la création du site web, tant sur la partie **frontend** (interface utilisateur) que sur la partie **backend** (logique de traitement des données). Le **frontend** doit être ergonomique, intuitif, et refléter fidèlement la maquette. Le **backend** doit permettre de gérer les filtres appliqués aux données et d'intégrer les rapports Power BI.
 - **Étapes** :
 1. Coder la structure HTML/CSS des pages à partir de la maquette.
 2. Implémenter les interactions utilisateur avec JavaScript pour les filtres et les graphiques.
 3. Développer l'API ou le backend pour traiter les requêtes de filtrage des données.
 4. Tester l'interaction entre le backend et Power BI pour s'assurer de la bonne génération des rapports.
-

7. Intégrer la génération de Power BI sur le site web

- **Difficulté** : Très difficile
 - **Durée estimée** : 34 jours
 - **Détails** : Intégrer les rapports Power BI directement sur le site web en garantissant une fluidité d'usage. Cela permettra aux utilisateurs de générer des rapports personnalisés à partir du site, avec une mise à jour dynamique des données.
 - **Étapes** :
 1. Intégrer les rapports Power BI en iframe ou via API dans les pages du site.
 2. Assurer la compatibilité entre les filtres du site et ceux de Power BI.
 3. Optimiser l'affichage des rapports pour qu'ils s'adaptent aux différentes résolutions d'écran.
 4. Tester la génération de rapports sur des navigateurs et appareils variés.
-

8. Intégrer le filtrage avancé dans la génération des rapports

- **Difficulté** : Difficile
- **Durée estimée** : 15 jours
- **Détails** : Mettre en place un système de filtrage avancé qui interagit avec la génération des rapports Power BI. Ce système doit permettre aux utilisateurs de filtrer les données en fonction de critères précis (dates, catégories, localisation) et de générer des statistiques instantanément.
- **Étapes** :
 1. Développer un système de filtrage personnalisé en JavaScript ou via une API.
 2. Relier les filtres aux rapports Power BI de manière dynamique.
 3. Tester l'efficacité du filtrage et la rapidité de génération des rapports.

4. Optimiser les performances pour garantir une réponse rapide même sur des jeux de données volumineux.
-

9. Développer un moteur de recherche et optimiser le filtrage

- **Difficulté** : Moyenne
 - **Durée estimée** : 13 jours
 - **Détails** : Mettre en place un moteur de recherche efficace sur le site, capable de rechercher des statistiques spécifiques. Cela nécessite également une optimisation du filtrage existant pour que l'utilisateur puisse trouver facilement les informations pertinentes.
 - **Étapes** :
 1. Développer un moteur de recherche en front-end avec des suggestions instantanées.
 2. Tester le moteur de recherche avec des cas d'utilisation variés.
-

10. Réaliser le README

- **Difficulté** : Facile
 - **Durée estimée** : 1 jour
 - **Détails** : Rédiger un fichier README clair et détaillé pour le dépôt GitHub. Ce document doit expliquer comment installer et utiliser le projet, décrire les principales fonctionnalités, et lister les dépendances. Il doit aussi inclure des instructions pour les contributions et les tests.
 - **Étapes** :
 1. Décrire le projet et ses objectifs.
 2. Fournir des instructions pour l'installation et la configuration.
 3. Lister les fonctionnalités principales et les dépendances requises.
 4. Expliquer comment exécuter les tests et contribuer au projet.
-

11. Tests unitaires et d'intégration

- **Difficulté** : Moyenne
- **Durée estimée** : 15 jours
- **Détails** : Développer des tests unitaires pour chaque composant du projet, ainsi que des tests d'intégration pour vérifier la cohérence des différents modules (filtres, génération de rapport, affichage des données). Ces tests doivent être automatisés pour garantir une maintenance efficace du code.
- **Étapes** :
 1. Créer des tests unitaires pour chaque fonction JavaScript (filtres, interactions).
 2. Créer des tests d'intégration pour vérifier la communication entre le backend et Power BI.
 3. Mettre en place un système d'intégration continue pour lancer les tests à chaque modification du code.

Conventions de nommage

1. Utilisation de la casse :

- Tous les noms de fonctions, variables et fichiers doivent être en **minuscule** et respecter la convention **snake_case** (chaque mot est séparé par un tiret bas `_`).
- **Exemples :**
 - Pour les fonctions : `generer_powerbi()`
 - Pour les variables : `nombre_utilisateurs`, `donnees_filtrees`

2. Pas de variables globales :

- Aucune variable globale ne doit être utilisée.

3. Noms de variables clairs et descriptifs :

- Les noms de variables doivent être suffisamment clairs et descriptifs pour indiquer leur rôle dans le code.
- Utiliser des noms explicites, même si cela les rend légèrement plus longs.
- **Exemple :** `nombre_etudiants` est préférable à `nb_etud`.

4. Noms des fonctions :

- Les noms de fonctions doivent être des **verbes** ou des expressions verbales, car elles représentent des actions.
- **Exemples :**
 - `calculer_statistiques()`
 - `filtrer_donnees()`
 - `generer_rapport()`

5. Noms des constantes :

- Les constantes doivent être en **majuscule** avec des underscores pour séparer les mots.
- **Exemple :** `TAILLE_MAX_FICHER = 10000`

6. Langue :

- Tout le code doit être écrit **en français** pour garantir la cohérence dans le projet. Cela inclut les noms de variables, les noms de fonctions et les commentaires.

7. Commentaires :

- Utiliser des commentaires clairs et concis pour expliquer des parties complexes du code ou les intentions derrière certaines décisions.
- Les commentaires doivent également être rédigés en français.

8. Structure des classes (si applicable) :

- Les noms de classes doivent suivre la convention **PascalCase** (c'est-à-dire une majuscule pour chaque mot, sans espaces ni underscores).
- **Exemple :** `GenerateurRapport`, `AnalyseurDonnees`

9. Fichiers :

- Les noms de fichiers doivent être en **snake_case** et refléter clairement leur contenu.
- **Exemple :** `analyse_statistiques.py`, `generateur_rapport.py`

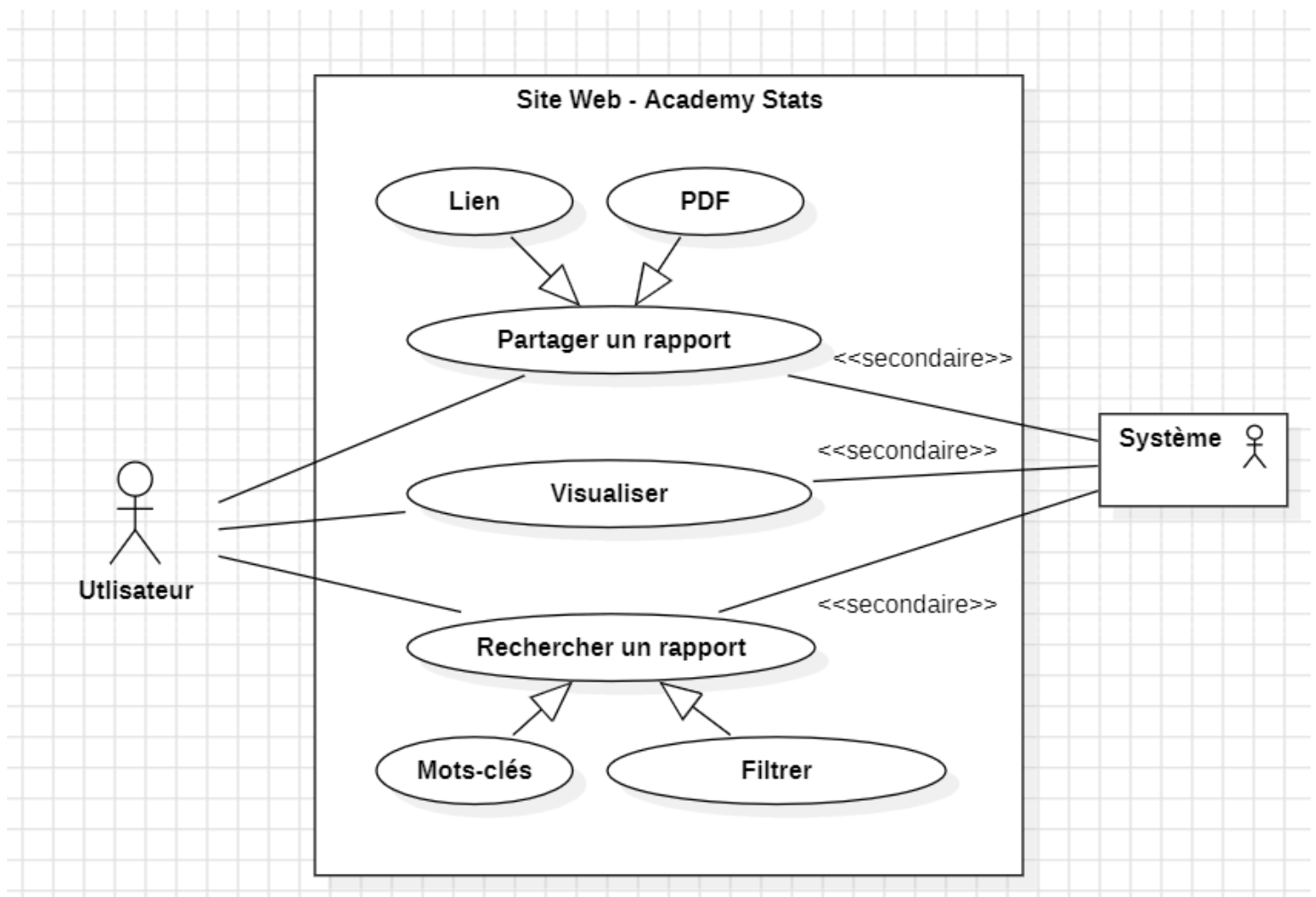
10. Limitation des longueurs :

- Les noms de variables ou de fonctions ne doivent pas être trop longs, mais suffisamment descriptifs. Entre 15 et 30 caractères est une bonne limite.

Conclusion

Academy Stats a pour ambition de fournir un accès simplifié aux statistiques éducatives grâce à des outils de visualisation de données interactifs. En centralisant les données open data et en les rendant accessibles à travers des filtres personnalisables, ce projet permettra de démocratiser l'accès à l'information éducative et de fournir des outils puissants pour mieux comprendre les dynamiques du secteur éducatif.

Diagramme de Cas d'Utilisation



Explication du diagramme :

- **Acteurs :**
 - **Utilisateur :** L'utilisateur est l'acteur principal du système. Il représente toute personne interagissant avec la plateforme, comme un étudiant, un enseignant ou un autre type d'utilisateur.
 - **Système :** Il s'agit d'un acteur non-humain (externe) qui est responsable de la génération automatique des rapports. Ce système agit de manière autonome pour créer les rapports basés sur les données disponibles.
- **Cas d'utilisation :**

- **Visualiser** : L'utilisateur peut accéder aux rapports disponibles sur la plateforme pour les consulter.
- **Partager un rapport** : L'utilisateur peut partager les rapports qu'il a consultés, soit sous forme de **Lien**, soit sous forme de fichier **PDF**. Ces deux options sont des sous-cas d'utilisation (ou spécialisations) du partage de rapport.
- **Rechercher un rapport** : L'utilisateur peut effectuer une recherche d'un rapport particulier, soit en utilisant des mots-clés dans une barre de recherche, soit en appliquant des filtres. Ces deux méthodes représentent des spécialisations du cas d'utilisation Rechercher un rapport.
- **Définitions des symboles** :
 - **Flèche en triangle** : Utilisée pour représenter une **généralisation** ou une **spécialisation**. Ici, elle indique qu'il existe plusieurs options possibles pour accomplir une tâche, comme choisir entre partager un rapport via un lien ou un PDF.
 - **Ligne simple** : Représente une **relation** directe entre un acteur (Utilisateur ou Système) et un cas d'utilisation. C'est l'interaction entre l'utilisateur et une action du système.
 - **Bonhomme bâton** : Symbolise un **acteur humain**, une personne physique qui interagit avec le système.
 - **Bonhomme bâton dans un rectangle** : Représente un **acteur non-humain**, généralement un système ou une machine, comme ici avec le système générant les rapports automatiquement.

Diagramme de Gantt

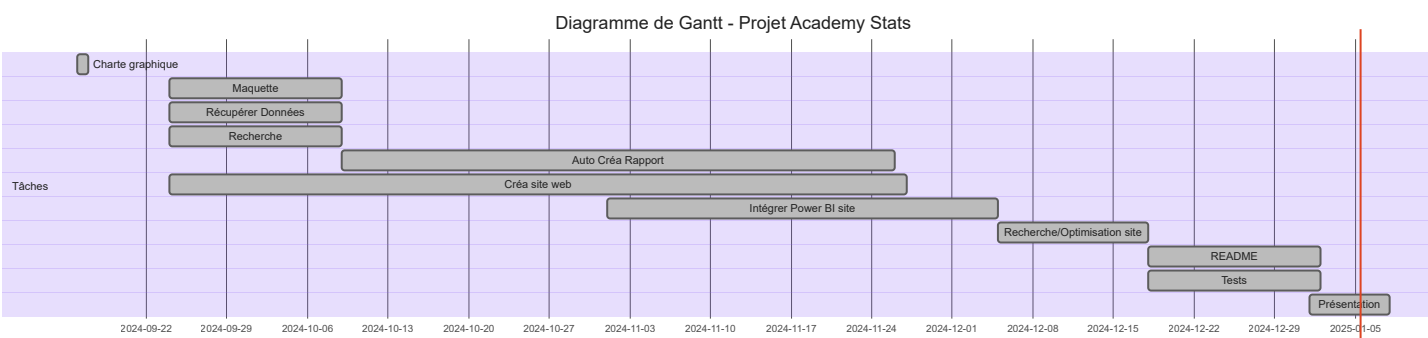


Tableau récapitulatif des Tâches

Légende des responsabilités

- **C = Candice**
- **T = Thanina**
- **C + T = Collaboration entre Candice et Thanina**

Tâche	Date de début	Date de fin	Durée	Responsable
Charte graphique	16-09-24	16-09-24	1 jour	C
Maquette	24-09-24	08-10-24	15 jours	C
Récupérer Données	24-09-24	08-10-24	15 jours	C

Tâche	Date de début	Date de fin	Durée	Responsable
Recherche	24-09-24	08-10-24	15 jours	T
Auto Créa Rapport	09-10-24	25-11-24	48 jours	T
Créa site web	24-09-24	26-11-24	64 jours	C
Intégrer Power BI site	01-11-24	04-12-24	34 jours	C + T
Recherche/Optimisation site	05-12-24	17-12-24	13 jours	C
README	18-12-24	01-01-25	15 jours	T
Tests	18-12-24	01-01-25	15 jours	C
Présentation	01-01-25	07-01-25	7 jours	C + T

Charte graphique

Responsable : Candice

Description : Réalisation de la charte graphique du site web, incluant la définition des couleurs, typographies, et autres éléments visuels. Cette étape permet de garantir une cohérence graphique à travers toutes les pages du site.

Maquette

Responsable : Candice

Description : Conception de la maquette du site web. Il s'agit de définir visuellement les différentes pages, notamment la page d'accueil et les sections liées aux statistiques et aux filtres. Utilisation de l'outils comme Canva.

Récupérer Données

Responsable : Candice

Description : Recherche, sélection et récupération des données pertinentes en open data sur des plateformes comme [Data.gouv.fr](https://data.gouv.fr). Cette tâche consiste à identifier les données qui seront utilisées pour générer les statistiques sur le site.

Recherche

Responsable : Thanina

Description : Identifier les meilleures pratiques et technologies pour mener à bien le projet.

Quelle technique utiliser ?

Objectif : Rechercher les algorithmes ou techniques adaptés pour l'analyse des statistiques et la génération des rapports.

Quelle API utiliser ?

Objectif : Rechercher les API nécessaires pour l'accès aux données ou pour d'autres fonctionnalités (API Open Data, visualisation, etc.).

Quelles fonctions développer ?

Objectif : Définir les fonctions clés à développer (génération de rapports, filtrage des données, etc.).

Auto Créa Rapport

Responsable : Thanina

Description : Automatisation de la création des rapports à partir des données. Le but est de générer dynamiquement des rapports interactifs (via Power BI ou des alternatives open source) en fonction des critères choisis par l'utilisateur.

Créa site web

Responsable : Candice

Description : Développement complet du site web, incluant la structure HTML, la mise en forme CSS, et les interactions JavaScript. Cette tâche englobe aussi l'intégration du backend en Python pour traiter les requêtes et les filtres de données.

Intégrer Power BI site

Responsables : Candice & Thanina

Description : Intégration des rapports automatisés dans le site web. Il s'agit de rendre les rapports Power BI accessibles via l'interface utilisateur du site, en permettant aux utilisateurs de générer et visualiser les rapports personnalisés.

Recherche/Optimisation site

Responsable : Candice

Description : Amélioration des performances du site, en particulier l'optimisation du filtrage des données et la création d'un moteur de recherche performant. Cette fonctionnalité doit permettre aux utilisateurs de trouver rapidement les statistiques pertinentes.

Faire le README

Responsable : Thanina

Description : Rédaction du fichier README pour le dépôt GitHub du projet. Ce document doit inclure une description claire du projet, les instructions d'installation et d'utilisation, ainsi que les détails techniques pour les contributeurs.

Tests

Responsable : Candice

Description : Mise en place des tests unitaires et d'intégration pour vérifier que toutes les fonctionnalités du site web fonctionnent correctement. Cela inclut les tests du système de filtrage, de la génération de rapports, et des interfaces utilisateur.

Présentation

Responsables : Candice & Thanina

Description : Finalisation du projet et préparation de la présentation pour le 7 janvier. Cela inclut la mise en forme de la documentation, la préparation des diapositives, et une démonstration fonctionnelle du site web.

[Diagramme de Gantt](#)

[Diagramme de Cas d'Utilisation](#)