Document d’avancement du projet

Tâches		03/12/2024	10/12/2024	17/12/2024	24/12/2024	31/12/2024	07/01/2025	14/01/2025	21/01/2024
Création du GitHub	Heidi								
Choix et validation du sujet	Heidi - Clarence								
Brain storming sur le type de visualisaitons	Heidi - Clarence								
Document de cadrage - Objectifs	Heidi - Clarence								
Demande de récupération de données personnelles Spotify	Heidi - Clarence								
Compréhension des fichiers et data exploitables	Heidi - Clarence								
Répartition des tâches	Heidi - Clarence								
Mise en plca de l'API Spotify	Heidi								
Graphe top 10 artistes	Clarence								
Bubble graphe genre écoutés	Heidi								
Graphe top 10 musiques	Clarence								
Carte du monde de la provenance des artistes	Heidi								
Heatmap	Clarence								
Graphe radar - suivis des périodes d'écoutes	Clarence								
Interface	Heidi - Clarence								
Mise en page et uptade du GitHub	Heidi								
Document de suivis des taches	Clarence								

1. Déroulement du projet

1.1. 03/12/2024 - 10/12/2024 : Mise en place du projet Visualisation de données Spotify

1.1.1. Création du repo GitHub "SpotifyDataViz"

Mise en place d'un repository pour présenter le projet, suivre ses avancées et le visualiser via GitHub Pages.

1.1.2. Brainstorming

Réflexion collaborative sur :

- Les données exploitables et leur mode d'accès (API Spotify, demande de données personnelles ou les deux).
- Les types de graphiques existants et les plus adaptés à chaque type de données.

1.1.3. Rédaction du document de cadrage

Définition des objectifs :

- Problème abordé et besoin visé.
- Public cible et tâches associées à la visualisation.
- Sources de données sélectionnées.
- Travaux existants liés au projet.
- Organisation et répartition des tâches.
- Esquisses des graphiques envisagés.

1.1.4. Demande des données personnelles

Requête auprès de Spotify pour récupérer les datasets des 12 derniers mois pour chaque membre du groupe, afin de maximiser les données disponibles.

1.1.5. Création du document d'avancement

Mise en place d'un diagramme de Gantt pour suivre les tâches.

1.2. 17/12/2024 : Réception et analyse des données

1.2.1. Réception des données personnelles Spotify

Téléchargement des fichiers fournis par Spotify.

1.2.2. Analyse des fichiers récupérés

- Format et contenu des fichiers examinés.
- Sélection des fichiers utiles : StreamingHistory_music_X.json.
    - Contenu : titre, artiste, date/heure d'écoute, durée d'écoute en millisecondes.
- Identification des graphiques et des observations à produire.

1.2.3. Répartition des tâches

Distribution des graphiques à concevoir entre les membres de l'équipe.

1.2.4. Utilisation de l'API Spotify

- Recherche sur l'utilisation de l'API pour combler les données manquantes (ex. : genres musicaux).
- Création d'une application sur Spotify Developers pour récupérer un ID et activer l'API.

1.2.5. Mise à jour du diagramme de Gantt

1.3. 24/12/2024 : Début de création des graphiques

1.3.1. Mise en place de l'API Spotify

- Récupération des tokens nécessaires via Spotify Developers.

1.3.2. Graphe Top 10 artistes (Version 1)

- Barres verticales : temps d'écoute par artiste.
- Trois graphiques distincts : par jour, par semaine, par mois.
- Tooltip : nom de l'artiste et durée d'écoute (convertie en heures/minutes/secondes).

1.3.3. Graphe radar (Top artistes par période)

- Répartition des mois autour du radar.

- Graduation : temps d'écoute par mois (en heures/minutes/secondes).
- Apparition du radar correspondant lorsqu'une barre du graphe Top 10 artistes est cliquée.

1.3.3. Mise à jour du diagramme de Gantt

1.4. 31/12/2024 : Conception des graphiques

1.4.1. Bubble graph (Genres écoutés)

- Données récupérées via l'API Spotify.
- Tooltip : genre musical correspondant à chaque bulle.
- Taille des bulles : proportion d'écoute du genre.

1.4.2. Graphe Top 10 artistes (Version 2)

- Barres horizontales : artiste le plus écouté en haut.
- Tooltip conservé.

1.4.3. Graphe Top 10 musiques (Version 1)

- Barres verticales : temps d'écoute par musique.
- Trois graphiques distincts : par jour, par semaine, par mois.
- Tooltip : nom de la musique, artiste, durée d'écoute (convertie).

1.4.4. Heatmaps (Heures et jours d'écoute)

- Trois heatmaps distinctes :
    - Heures d'écoute par jour.
    - Jours d'écoute par semaine.
    - Mois d'écoute.
- Intensité des couleurs : temps d'écoute moyen.
- Tooltip : durée d'écoute moyenne (convertie).

1.4.5. Design de l'interface avec Bootstrap

- Mise en place de la structure :
    - Compteur par graphe.
    - Bouton pour connexion/déconnexion avec Spotify.

1.4.6. Mise à jour du diagramme de Gantt

1.5. 07/01/2025

1.5.1. Assemblage des graphiques dans une seule visualisation avec un sélecteur pour choisir la période souhaitée et afficher le graphique correspondant.

1.5.2. Difficultés d'assemblage des heatmaps dans une visualisation unique. Modifications du code et des visualisations sans succès. Objectif : ajouter deux sélecteurs, un pour choisir une des trois heatmaps et un autre pour choisir la période.

1.5.3. Graphe Top 10 musiques (Version 2)

Deuxième version du graphique « Top 10 musiques » (camembert) avec un tooltip affichant :

- Le titre de la musique,
- Le nom de l'artiste,
- Le temps d'écoute converti en secondes, minutes et heures (au lieu de millisecondes). Ajout d'un sélecteur pour afficher les visualisations par semaine ou par mois. Objectif : diversifier les types de graphiques tout en gardant des visuels adaptés.

1.5.4. Carte du monde

- Carte récupérée via GeoJSON.
- Mapping manuel associant chaque artiste à son pays d'origine.
- Échelle de couleur : vert foncé (forte présence d'artistes) à gris (faible présence).
- Tooltip affichant le nom du pays et le nombre d'artistes écoutés.
- Objectif : personnaliser les visualisations pour mieux comprendre les goûts musicaux et les origines des artistes.

1.5.5. Interface utilisateur

- Ajout d'un bouton pour chaque utilisateur dans chaque conteneur (sauf pour la carte du monde et le graphique en bulles qui utilisent l'API Spotify).
- Boutons permettant de récupérer des données locales via un lien au repo GitHub.
- Ajout de sélecteurs de périodes pour chaque graphique.
- Intégration des graphiques : Top 10 artistes, radar graph, Top 10 musiques.

1.6. 14/01/2025 – Présentation et discussion avec le professeur

1.6.1. Interface :

- Difficultés d'intégration des heatmaps dans l'interface.

- Intégration des graphiques « bubble chart » pour les genres musicaux et la carte du monde (données provenant de l'API Spotify).
- Présentation au professeur de l'interface fonctionnelle sans les heatmaps.

1.6.2. Travail sur les heatmaps :

- Assemblage des heatmaps journalières et hebdomadaires dans une visualisation unique avec un sélecteur.
- Discussion avec le professeur : la heatmap des heures d'écoute par jour n'est pas pertinente.
- Nouvelle approche : une seule heatmap montrant les heures d'écoute par semaine.
- Visualisation des 4 semaines d'un même mois avec un sélecteur pour choisir le mois.
- Tooltip : temps d'écoute converti en heures, minutes et secondes.

1.6.3. Amélioration du radar graph :

- Axes en blanc (fond noir).
- Réduction de la taille et alignement avec le bar graph du Top 10 artistes.
- Ajout d'un conteneur plus grand ou création d'un nouveau conteneur pour afficher les deux graphiques sans scroll.
- Graduation linéaire des mois : affichage de tous les mois du dataset autour du radar, même ceux sans écoutes.

1.7. 21/01/2025 – Présentation finale et dernières tâches

1.7.1. Refonte de l'interface :

- Abandon du framework Bootstrap.
- Nouveau design épuré et homogène.
- Simplification et organisation du code.
- Uniformisation des couleurs et des tailles des graphiques.

1.7.2. Tri des données uniquement par semaine (tri quotidien et mensuel jugés non pertinents).

1.7.3. Intégration des heatmaps dans l'interface.

1.7.4. Unification de la sélection des utilisateurs :

- Un bouton pour l'utilisateur 1 et un autre pour l'utilisateur 2.
- Les boutons appellent à la fois l'API Spotify et les fichiers JSON locaux stockés dans le repo GitHub.

1.7.5. Finalisation des détails :

- Ajustements de l'interface et des visualisations.

- Rédaction du document de cadrage et mise à jour de la Gantt chart.

1.7.6. Dernier push sur le repo GitHub et pull request finale.

---

2. Problèmes rencontrés

- **Effectif réduit** : Le projet a été réalisé uniquement par deux personnes, ce qui a limité la capacité à avancer sur plusieurs tâches en parallèle.
- **Manque de temps** : Le calendrier serré a restreint la possibilité d'approfondir certaines fonctionnalités ou optimisations.
- **Sous-estimation du temps nécessaire pour l'interface** : La conception de l'interface utilisateur a pris bien plus de temps que prévu, en particulier pour intégrer les graphiques de manière interactive et intuitive.
- **Mapping manuel des origines des artistes** : La création d'une carte montrant les origines des artistes a été un défi. Le processus manuel de collecte et de traitement des données a été long et fastidieux. Faute de temps, ce travail n'a été réalisé que sur un échantillon réduit d'artistes. Si plus de temps avait été disponible, nous aurions pu explorer des solutions automatisées pour améliorer cette partie.

---

3. Ressources utilisées

1. Bootstrap

- **Définition** : Framework CSS conçu pour faciliter le développement d'interfaces web responsives et modernes. Il propose des grilles, des composants prédéfinis et des styles réutilisables pour accélérer le design des sites web.
- **Utilité** : Création rapide et harmonieuse de l'interface utilisateur, assurant une mise en page responsive adaptée aux écrans de différentes tailles.
- **Lien** : https://getbootstrap.com/

2. GeoJSON

- **Définition** : Format basé sur JSON utilisé pour représenter des données géospatiales, comme des points, des lignes ou des polygones.
- **Utilité** : Permet de structurer et de visualiser des données géographiques, comme le mapping des origines des artistes sur une carte interactive.
- **Lien** : https://geojson.org/

3. CodePen

- **Définition** : Plateforme en ligne pour tester, partager et collaborer sur des extraits de code HTML, CSS et JavaScript.
- **Utilité** : Prototypage rapide et test de petites fonctionnalités ou designs spécifiques avant de les intégrer au projet principal.
- **Lien** : https://codepen.io/

4. Git

- **Définition** : Système de gestion de versions décentralisé permettant de suivre les modifications du code et de collaborer efficacement.
- **Utilité** : Gestion des différentes versions du projet via GitHub, permettant un suivi clair et un travail collaboratif.
- **Lien** : https://git-scm.com/

5. Spotify API

- **Définition** : Interface de programmation proposée par Spotify permettant d'accéder à des données sur les utilisateurs, les morceaux, les artistes et bien plus.
- **Utilité** : Récupération des genres musicaux des artistes et autres données nécessaires à l'analyse et la visualisation.
- **Lien** : https://developer.spotify.com/documentation/web-api/

6. Visual Studio Code (Live Server)

- **Définition** : Éditeur de code léger et puissant, avec une extension "Live Server" permettant de visualiser un projet local en temps réel dans le navigateur.
- **Utilité** : Développement et tests rapides de l'interface et des fonctionnalités en temps réel grâce à l'extension Live Server.
- **Lien** : https://code.visualstudio.com/

7. 3DS

- **Définition** : Logiciel permettant de modéliser, animer et visualiser en 3D.
- **Utilité** : Pas utilisé directement dans le projet, mais mentionné comme une ressource annexe potentiellement utile pour des visualisations en 3D avancées.
- **Lien** : https://www.autodesk.com/products/3ds-max/

8. Node.js

- **Définition** : Environnement d'exécution JavaScript permettant de développer des applications côté serveur.
- **Utilité** : Utilisé pour des scripts nécessaires à la récupération des données via des API ou pour le traitement des données.
- **Lien** : https://nodejs.org/