

**Université de Cergy-Pontoise**

**RAPPORT**

pour le projet de Développement-Web  
**Licence d'Informatique deuxième année**

sur le sujet

**MUSIC**

rédigé par

**DAMODARANE Jean-Baptiste  
&  
ELUMALAI Sriguru**

**Année 2021-2022**



# Table des matières

1	Introduction.....	3
1.1	Contexte du projet.....	3
1.1.1	Description générale du site web.....	3
1.2	Les principaux objectifs du projet.....	3
1.3	Outils de Développement.....	4
1.4	Organisation du rapport.....	4
2	Informations Techniques.....	5
2.1	Plan du Site.....	5
2.2	Recherches.....	6
2.2.1	Maquette de la page Recherche.....	7
2.3	Informations.....	7
2.3.1	Maquette de la page Information.....	8
3	Architecture Client-Serveur.....	9
3.1	Bases de données (côté serveur).....	9
3.1.1	Maquette de la page Historique.....	10
3.1.2	Maquette de la page Statistique.....	10
3.2	Cookies (côté client).....	11
3.2.1	Maquette de la page d'accueil.....	11
4	GESTION DU PROJET.....	12
4.1	Planification de Projet.....	12
4.2	Répartitions des tâches.....	13
4.3	Répartitions des tâches entre membres de l'équipe.....	13
5	CONCLUSION.....	14
5.1	Défis rencontrés.....	14
5.2	Remerciements.....	14

## 1.1 Contexte du projet

Dans le cadre de notre deuxième année de Licence en Informatique à CY Cergy Paris Université, nous avons eu à réaliser, un projet d'environ 4 mois au sein du module Développement Web. Notre projet consiste à développer un site internet basé sur la Music, nommé "[Rythm Break](#)". Nous détaillerons dans ce dossier les fonctionnalités contenues dans le site et son architecture. Toutefois, l'objectif primordial de notre projet est de mettre en avant les grandes idées/options présentes dans les services de sites internet musicaux.

### 1.1.1 Description générale du site web

Notre site web met en avant les principaux fonctionnements d'un site internet musical. Inspiré de grands plateformes musicaux comme « Spotify » ou encore « Deezer », notre site permet aux utilisateurs de chercher des musiques, des albums et des artistes, et également de consulter les tendances actuels qui seront mises à jour régulièrement grâce aux APIs utilisés. Des écoutes des morceaux (de courte durée) des musiques y sont également présentes. Un système d'historique permettant de stocker les pages consultées est mis en place afin de rendre la navigation plus simple aux visiteurs.

## 1.2 Les principaux objectifs du projet

Les principaux objectifs de notre projet sont :

- Manipuler les format XML et JSON.
- Recherche de musiques, d'artistes et d'albums.
- Stockage de musiques consultés par les utilisateurs du site dans un fichier csv.
- Mise en place d'un compteur de fréquentation du site web.
- Utilisations de cookies pour changer la charte graphique lors du navigation.

## 1.3 Outils de Développement

- Langages de programmation : PHP (HTML/CSS)
- Versions des langages : PHP 7 / HTML 5 / CSS 3.
- Plateforme de Développement : Visual Studio Code.
- Gestion des versions : [Github](#).

## 1.4 Organisation du rapport

Afin de mener à bien ce rapport, nous commencerons par présenter la partie technique de notre portait sur l'architecture du site et ses principaux fonctionnements. Ensuite, nous aborderons la partie architecture client-serveur dans la quelle, nous présenterons les bases de données (côté serveur) et les cookies (côté client).

Nous présenterons ensuite les différentes tâches que nous avons pu identifiées ainsi que leurs répartitions au sein de l'équipe

Enfin, nous terminerons par faire un court bilan de notre travail et les difficultés rencontrés.

## 2 Informations Techniques

### 2.1 Plan du Site

Le site web est composé de cinq pages :

- La page « Recherche » permettant aux utilisateurs de chercher les chansons, les albums et les artistes.
- La page « Tendances » montrant les musiques, les albums et les artistes du moment.
- La page « Historique » affiche les musiques, les albums et les artistes qui ont été consultés par l'ensemble des utilisateurs récemment.
- La page « Statistiques » est composé d'Histogrammes des contenus consultés.
- La page « Annexe » permettant de montrer la localisation de l'utilisateur et l'image du jour par le service APOD.

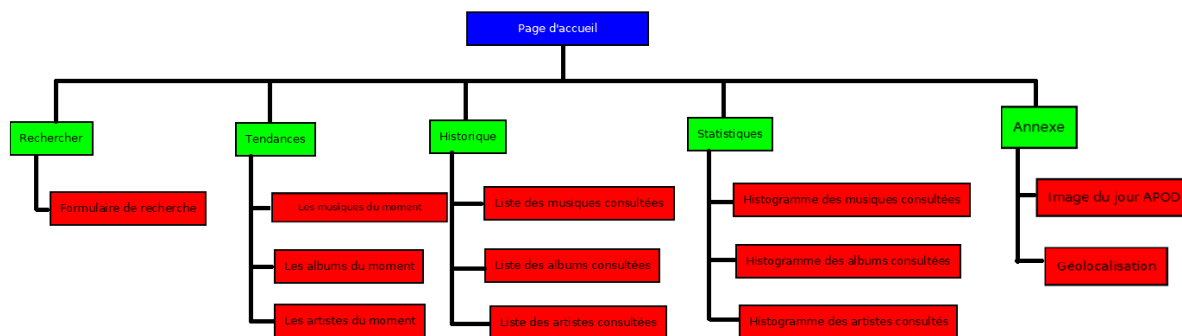


Figure 1: Image illustrant le plan du site

## 2.2 Recherches

Concernant l'interface de recherche de notre site, nous offrons la possibilité à l'utilisateur de choisir le type de recherche qu'il souhaite réaliser via une liste d'options prédéfinie. Il aura le choix d'effectuer une recherche sur un artiste, une musique ou un album, en sachant qu'il ne peut faire qu'un seul type de recherche à la fois. Après avoir fait son choix, le nom qu'il aura tapé sur la barre de recherche sera stocké dans une variable dans l'URL, pour d'éventuels requête plus tard.

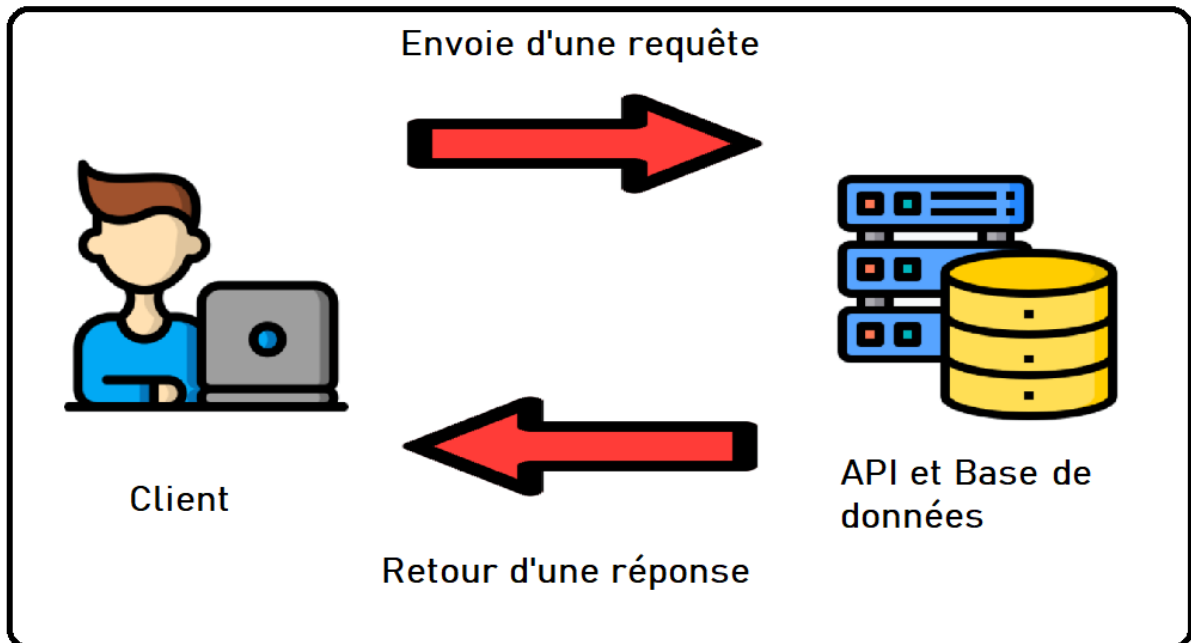


Figure 2: Image illustrant le fonctionnement de l'API

Pour l'affichage des résultats, nous avons utilisé l'API Deezer qui s'appuie sur plusieurs formats d'échanges dont les plus courant qui sont XML et JSON. Dans notre cas, pour pouvoir faire la requête correspondant à la recherche d'un artiste, d'une musique ou d'un album vers l'API Deezer, nous avons commencé par récupérer le nom de l'élément rechercher par l'utilisateur via la méthode GET du formulaire, qu nous avons ensuite mis dans la requête et en retour avec le format JSON nous avons pu récupérer l'ID de chaque résultat correspondant à la recherche de l'utilisateur, qu'on a stocker dans une variable dans l'URL, pour d'autres éventuelle requête plus tard. En parallèle de cela nous avons aussi pu afficher le noms et l'image pour chacun de ces résultats.

## 2.2.1 Maquette de la page Recherche

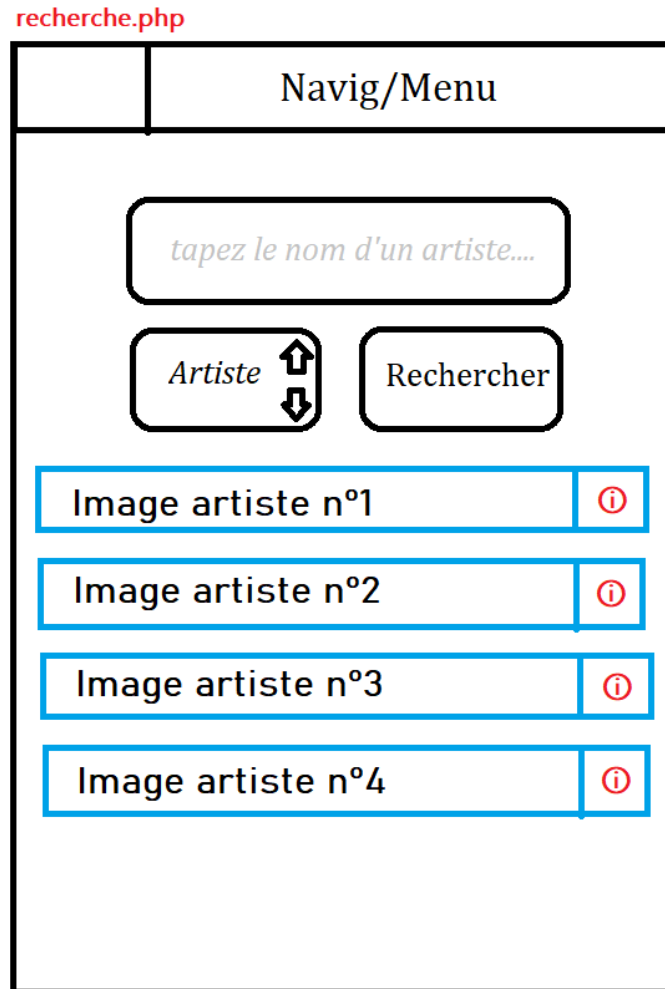


Figure 3: Page recherche lorsqu'un artiste est recherché

## 2.3 Informations

Pour pouvoir avoir plus d'informations sur un artiste, une musique ou un album, nous avons fait en sorte que lorsque l'utilisateur puisse cliquer sur l'une des images des résultats affichés précédemment dans la page de recherche, par lequel nous récupérerons l'ID du résultat courant, qu'on stocke dans l'url pour ensuite récupérer et entraîner une autre requête correspondante à la récupération de toute les autres informations complémentaire sur un artiste, un album ou une musique, (par exemple pour l'artiste : le nombre d'albums, le nombre de fans, les artistes similaires etc.) en utilisant toujours l'API Deezer avec le format JSON, en parcourant les différents indices du tableau.

### 2.3.1 Maquette de la page Information

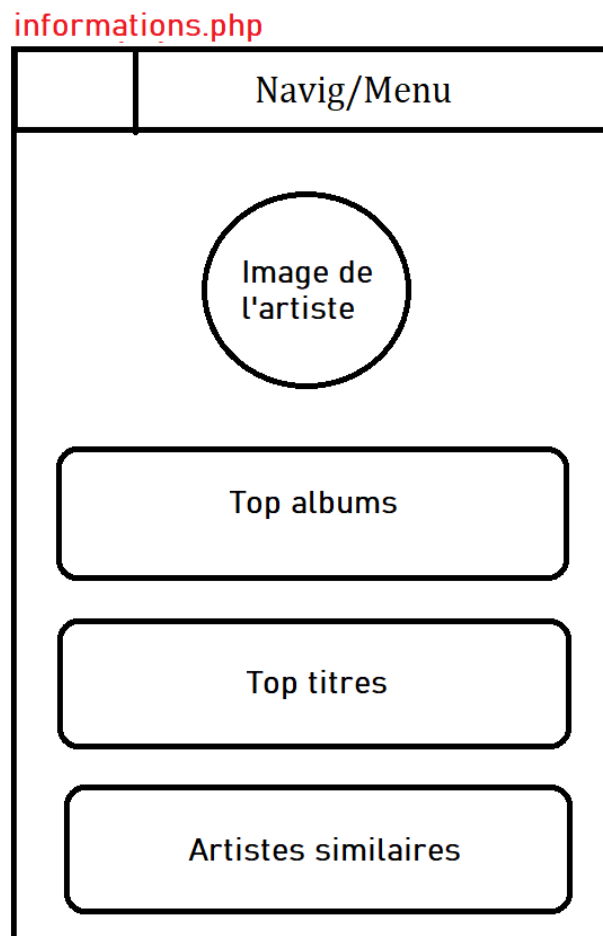


Figure 4: Page d'informations lorsqu'un artiste est recherché



### 3.1 Bases de données (côté serveur)

Nous avons créé 3 fichiers CSV :

- un pour les musiques consultées, qui contient 4 colonnes correspondant au nom de la musique, au nom de l'artiste, à l'ID et au nombre de vue.
- un pour les albums consultés, qui contient 4 colonnes correspondant au nom de l'album, au nom de l'artiste, à l'ID et au nombre de vue.
- un pour les artistes consultés, qui contient 3 colonnes correspondant au nom de l'artiste, à l'ID et au nombre de vue.

Prenons l'exemple du fichier CSV pour les musiques consultées, les 2 autres fonctionnent de la même manière.

Pour le 1<sup>er</sup> ajout de la musique dans le fichier CSV, on commence par vérifier avec la méthode GET du formulaire, si la variable correspondant au nom de la musique et la variable correspondant à son id existent dans l'url et s'ils sont vident. Lorsque c'est le cas, nous ajoutons dans le fichier CSV le nom de la musique, son ID, l'artiste de cette musique (en faisant une requête via l'ID vers l'API Deezer) et un compteur initialiser à 1 qui correspond au nombre de vue.

Pour les prochains ajouts dans le même fichier CSV, en plus de la vérification précédente avec la méthode GET du formulaire, nous vérifierons si la musique qui est en cours de consultation existe déjà dans le fichier CSV pour ce faire on comparera chaque ID de chaque musique présente dans le fichier CSV avec l'ID de la musique en cours de consultation et si ce sont les mêmes, on incrémentera juste le compteur (le nombre de vue) de 1 déjà existant dans le fichier. Sinon on fait l'ajout.

### 3.1.1 Maquette de la page Historique

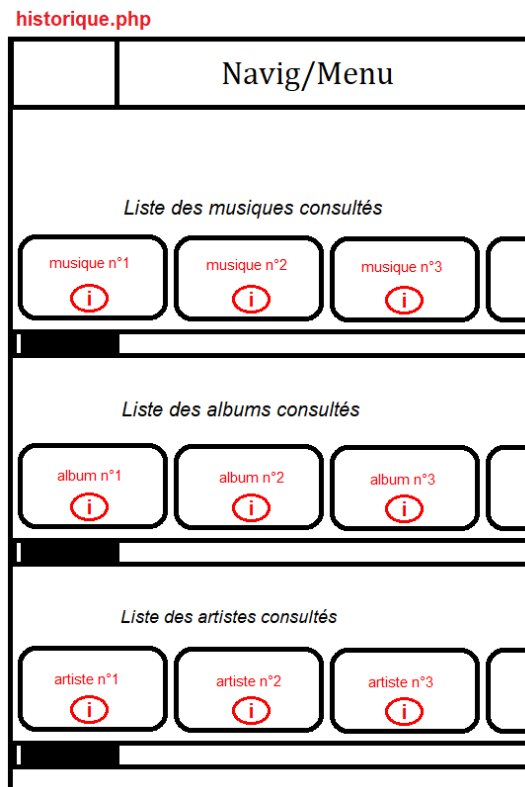


Figure 5: Page historiques avec l'ensemble des éléments consultés

### 3.1.2 Maquette de la page Statistique

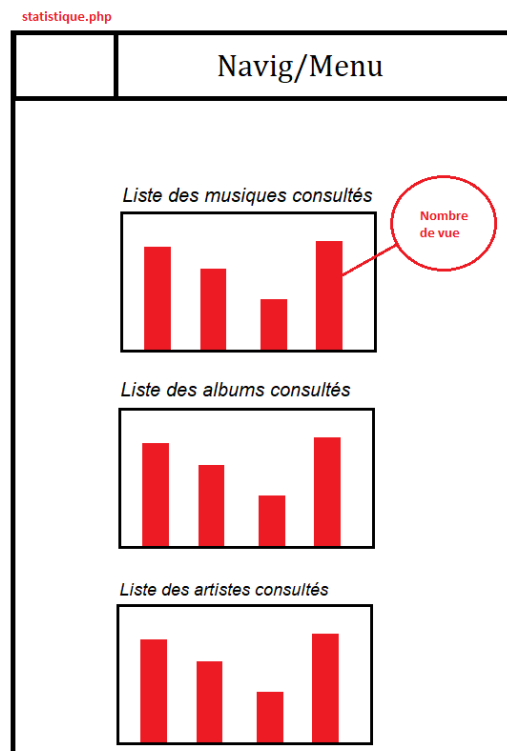


Figure 6: Page Statistiques avec les histogrammes des éléments consultés

## 3.2 Cookies (côté client)

On a mis en place 3 cookies dans notre site :

- Un cookie pour alterner entre un style sombre et un style jour : Lorsque l'utilisateur clique sur le logo mode nuit ou le logo mode jour, on enregistre son choix dans le cookie jusqu'à ce qu'il reclique sur un logo mode nuit ou jour.
- Un cookie pour la dernière musique consultés sur le site par l'utilisateur : On récupère le nom de la musique en cours de consultation et son ID dans l'URL via la méthode GET puis on l'enregistre dans le cookie jusqu'à ce que l'utilisateur consulte une autre musique. La récupération de ces 2 variables nous servira à faire une requête vers l'API Deezer pour afficher l'image de la musique et son nom dans la page d'accueil.
- Un cookie pour la date de la dernière personne ayant visité notre site : Lorsque l'utilisateur est entrain de consulté notre site, on enregistre dans le cookie, l'heure et le jour de sa visite qu'on affichera dans le bas de page. Ce cookie sera modifié lors de sa prochaine visite dans notre site.

### 3.2.1 Maquette de la page d'accueil

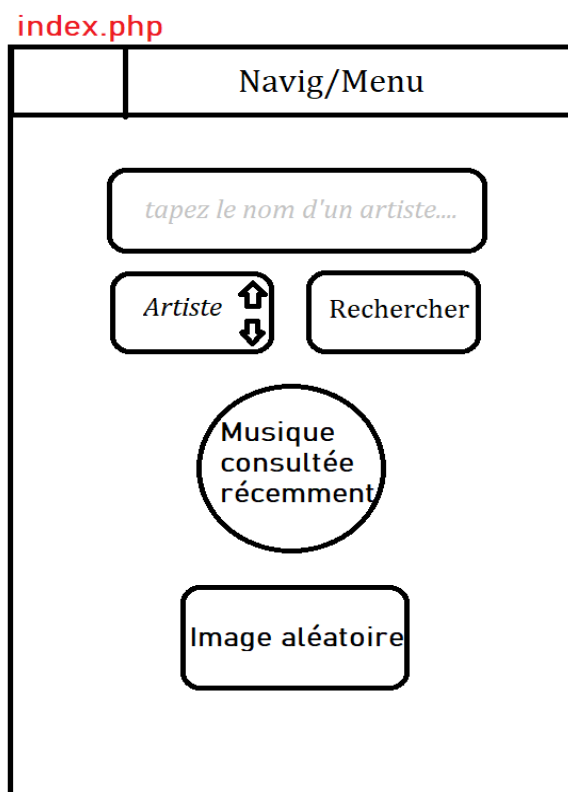


Figure 7: Page d'accueil du site

## 4 GESTION DU PROJET

La gestion du projet est la première chose que nous avons fait avant même de commencer le projet, car elle permet de répartir les travaux à réaliser entre les membres du groupe et nous a permis d'assurer la stabilité et la continuité du projet. Cela a permis également un meilleur avancement du Projet en surveillant au fur et à mesure son évolution.

### 4.1 Planification de Projet

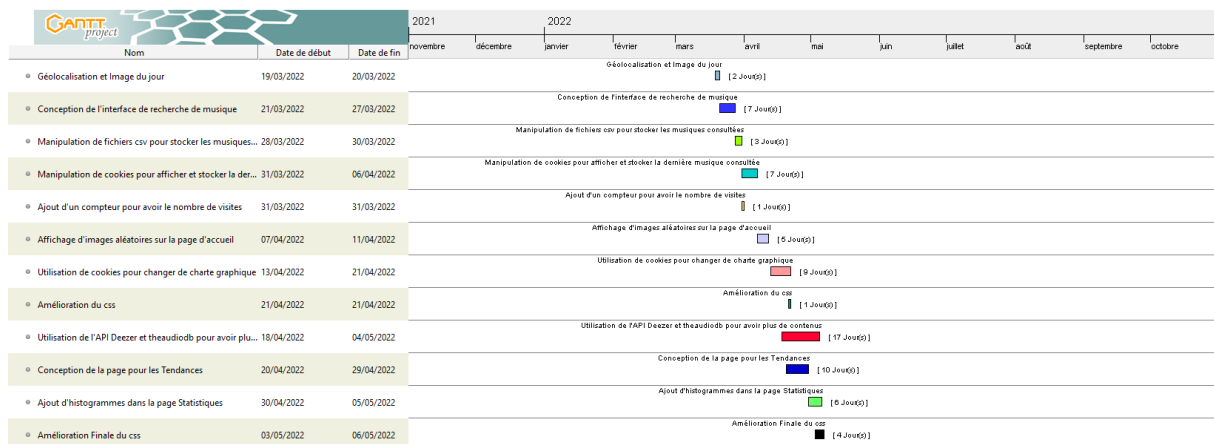


Figure 8: Diagramme de Gantt

Le diagramme de Gantt nous a permis d'avoir une vision plus globale et claire des objectifs du projet, leurs responsables et leurs niveaux d'importance.

Nous avons fait un diagramme de Gantt après avoir étudié les niveaux de faisabilités, pour ne pas avoir du retard.

## 4.2 Répartitions des tâches

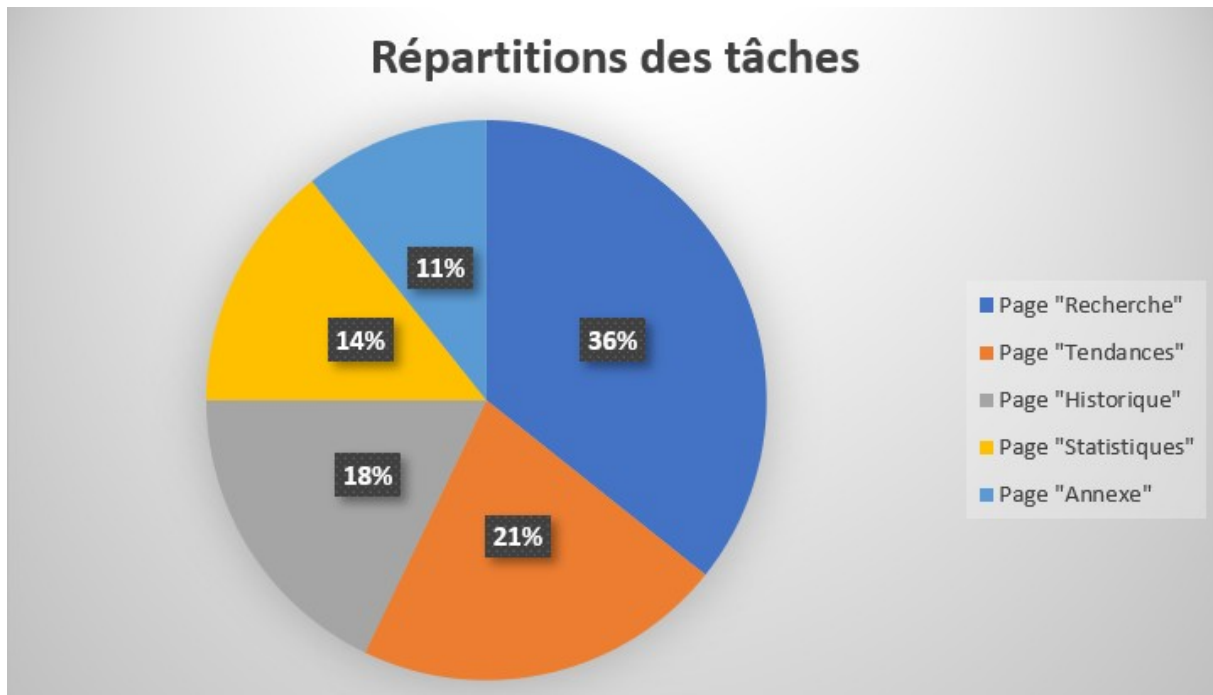


Figure 9: Diagramme de répartitions de tâches

## 4.3 Répartitions des tâches entre membres de l'équipe

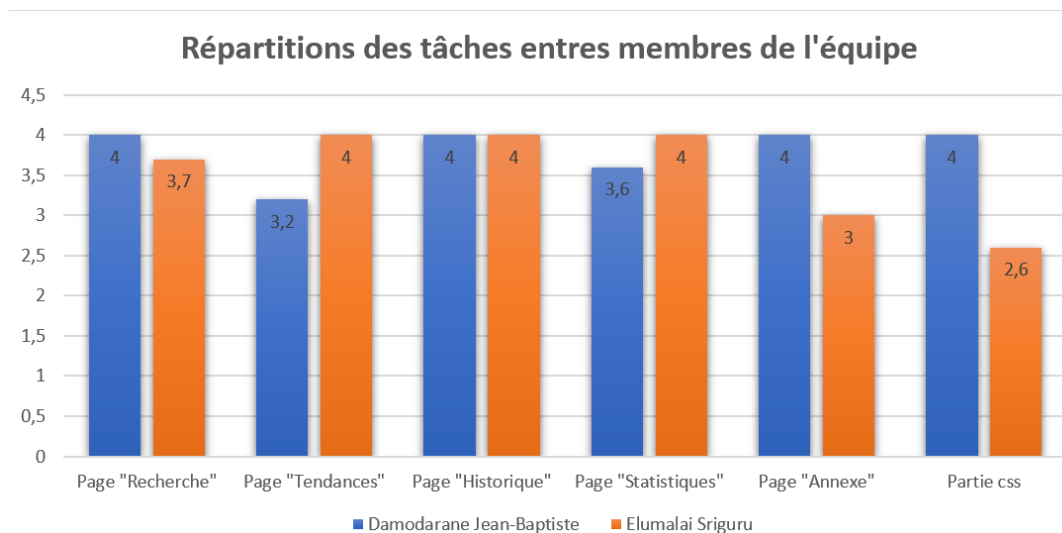


Figure 10: Diagramme de répartitions des travaux au sein de l'équipe

## 5 CONCLUSION

« Music » a été un sujet très intéressant et ce projet nous a permis de mieux comprendre le langage php, html et css.

Notre site permet de chercher des musiques, des albums et ainsi des artistes dont nous pouvons avoir des informations (biographie de l'artiste, informations sur des musique, des artistes, écoutes des extraits, etc.)

### 5.1 Défis rencontrés

La partie la plus complexe était d'afficher les informations sur les musiques, les artistes et les albums. Mais après quelques jours de recherches approfondies sur le sujet, nous avons su surmonter ce défi.

### 5.2 Remerciements

Nous tenons à remercier très sincèrement et à témoigner toute notre reconnaissance à notre professeur Mr.Marc Lemaire, qui nous a enseigné lors des CM et des TD, les bases du langage PHP, HTML et CSS lors du semestre 4, et cela nous a permis de nous introduire dans une dimension différente du monde informatique.

Nous remercions encore une fois, notre professeur, qui nous a guidé et conseillé lorsque nous avons rencontré des problèmes pendant que nous développons notre projet.

### 5.3 Liens pour les consultations du projet/site

- [Rythm Break](#)
- [Rythm Break \(damodaranejb.alwaysdata.net\)](#)
- [Rythm Break \(elumalai.alwaysdata.net\)](#)