# S2 UE Informatique - Projet CIR1

## *ETUDE PRÉLIMINAIRE*


**LES GCC PROGRAMMERS (GROUPE 4)**

- **GANGUE Boris**
- **CORVAISIER Antoine**
- **CORNIC Maël**

***Responsables du module* : *François LEGRAS & Sylvain LEFEBVRE***


## *SOMMAIRE*


I. *Présentation du projet*

II. *Procédure de la réalisation du projet*

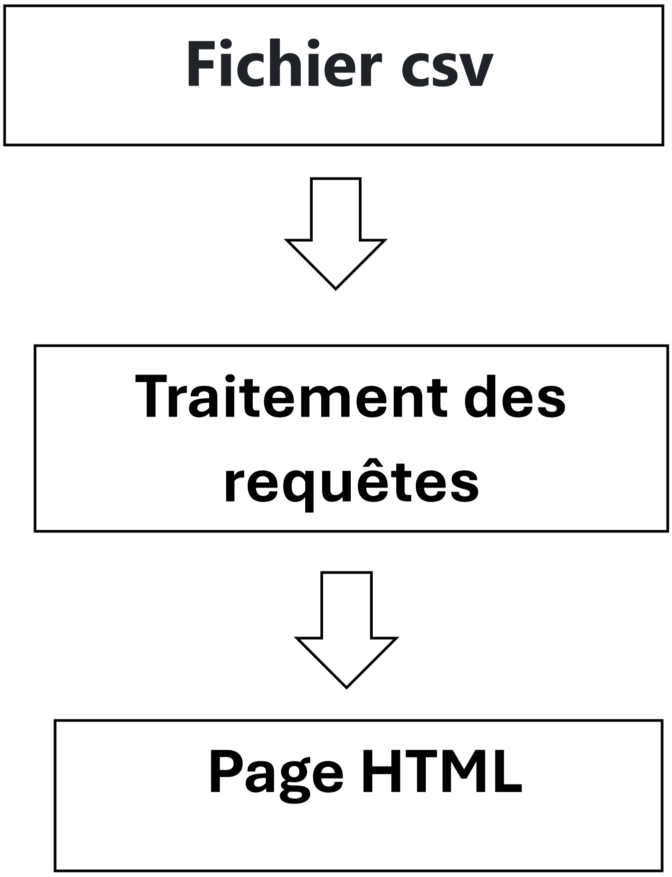
III. *Eventuelles points sensibles*

IV. *Planning*

V. *Démonstration mathématique*

## I. ***<u>Présentation du projet</u>***

L’idée principale du projet porte sur l’implémentation d’un **outil** capable de générer des *fichiers HTML* à partir de certaines requêtes soumit par l’utilisation.
Les pages seront générer en fonction des informations préformatées dans un *fichier csv*.



# II. *Procédure de la réalisation du projet*

Afin de répondre au mieux aux attentes en termes de rendu optimisé, nous avons choisi d'utiliser une table de hachage pour stocker le contenu du fichier CSV. Cela permet d'effectuer des opérations de recherche en temps constant, soit en O(1). Nous envisageons de diversifier les structures de données en sélectionnant les plus optimales pour le problème que nous souhaitons résoudre.

## ▪ Dépendances

- <stdio.h>
    - **printf** : Affiche un format de texte sur la sortie standard (console).
    - **scanf** : Lit des données depuis l'entrée standard (console).
    - **sprintf** : Écrit un format de texte dans un buffer.
    - **fprintf** : Écrit un format de texte dans un fichier.
    - **fscanf** : Lit des données depuis un fichier.
    - **fopen, fclose** : Ouvre et ferme des fichiers.
    - **fgets**: Lit et écrit des chaînes de caractères dans des fichiers.

- <stdlib.h>

- **`malloc, free`** : Alloue et libère de la mémoire dynamique.
- **`atoi`**: Convertit des chaînes de caractères en nombres entiers .
- **`exit`** : Termine le programme avec un code de sortie.

- <string.h>
  - **`strcpy`**: Copie des chaînes de caractères.
  - **`strcmp`** : Compare des chaînes de caractères.

## III. Eventuelles points sensibles

- Génération des fichiers html
- Implémentation des fonctions avancées
- Choix des structures de données optimales
- Intégration des différents portions de code

## IV. Planning

| | J-1 | J-2 | J-3 | J-4 | J-5 |
|---|---|---|---|---|---|
| *Boris* | **AM** : Définition des **struct** pour person.h, population.h et filemanager.h<br><br>**PM** : Implémentation des fonctions associées à population.c | **AM** : Code review de person.c et filemanager.c<br><br>**PM** : Recherche et implémentation des fonctions de advanced.c | **AM**: Terminer l'implémentation de advanced.c<br><br>**PM** : Code review de htmlexport.c | **AM** : Révision de l'ensemble du programme<br><br>**PM** : Préparation et rédaction de la présentation | **AM** : Dernère lecture<br><br>**PM** : Présentation |
| *Antoine* | **AM & PM** : Recherche sur comment implémenter les fonctions de htmlexport.c | **AM & PM** : Implémentation des fonctions de htmlexort.c | **AM** : Code review de advanced.c<br><br>**PM**: Rédaction du script | **AM** : Révision de l'ensemble du programme<br><br>**PM** : Préparation et rédaction de la présentation | **AM** : Dernère lecture<br><br>**PM** : Présentation |
| *Maël* | **AM & PM** : Implémentation des fonctions associées à person.c et filemanager.h | **AM & PM** : Code review de population.c | **AM & PM**: Implémentation du menu | **AM** : Révision de l'ensemble du programme<br><br>**PM** : Préparation et rédaction de la présentation | **AM** : Dernère lecture<br><br>**PM** : Présentation |

## V. Démonstration mathématique

Montrons que pour $n \geq 2, \sum_{\mathrm{i}=0}^{\mathrm{n}-1} 2^{\mathrm{i}} = 2^{\mathrm{n}} - 1$

**1. <u>Initialisation</u>**

Pour n = 2

$$\sum_{i=0}^{1} 2^i = 2^0 + 2^1 = 1 + 2 = 3$$

Et

$$2^2 - 1 = 4 - 1 = 3$$

Donc, la formule est vraie pour n = 2.

**2. <u>Hérédité</u>**

Supposons la proposition $k \geq 2, \sum_{\mathrm{i}=0}^{\mathrm{k}-1} 2^{\mathrm{i}} = 2^{\mathrm{k}} - 1$ vraie, montrons que $k \geq 2, \sum_{\mathrm{i}=0}^{\mathrm{k}} 2^{\mathrm{i}} = 2^{\mathrm{k}+1} - 1$

En utilisant l'hypothèse de récurrence, nous avons :
$\sum_{i=0}^{k} 2^i = \sum_{i=0}^{k-1} 2^i + 2^k$

Donc,

$$\sum_{i=o}^{k} 2^i = (2^k - 1) + 2^k$$

Simplifions l'expression :

$$\sum_{i=0}^{k} 2^i = 2^k - 1 + 2^k$$

$$\sum_{i=0}^{k} 2^i = 2^k + 2^k - 1$$

$$\sum_{i=0}^{k} 2^i = 2.2^k - 1$$

$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$$

Ainsi nous avons montrer que si la formule est vraie pour n = k, elle est également vraie pour n = k + 1.

### 3. Conclusion

La proposition est vraie au rang n = 2, elle est également vraie au rang n = k+1 donc par récurrence,

$$n \geq 2, \sum_{\mathrm{i}=0}^{\mathrm{n}-1} 2^{\mathrm{i}} = 2^{\mathrm{n}} - 1$$