

# **Librairie intelligente en ligne**

Rapport rédigé par  
RADOLANIRINA Yaël et LIGUORI Quentin  
dans le cadre du  
Projet d'Intégration  
au sein de  
CY Cergy Paris Université  
à destination de  
LIU Tianxiao

## **Remerciements**

Nous voudrions vivement remercier Monsieur LIU Tianxiao de nous avoir suivi et de nous avoir apporté son expertise tout au long de ce projet. Mais aussi messieurs LEMAIRE Marc et JEN Tao Yuan pour nous avoir fournis nos connaissances de base en développement web et base de données. Nous souhaitons aussi gratifier CY Cergy Paris Université pour son accueil et les enseignements qu'elle nous distribue. Et enfin nous souhaitons remercier l'association Génération Humaniste pour la mise à disposition de son local, ce qui nous a permis de travailler au calme et dans de bonnes conditions.

## Table des matières

Remerciements.....	2
1. Introduction.....	4
a. Contexte du projet.....	4
b. Objectif final.....	4
2. Spécifications globales.....	5
a. Plateforme web.....	5
b. Gestion des données.....	5
c. Fonctionnalités générales.....	6
d. Fonctionnement général de la suggestion.....	6
3. Conception technique.....	8
a. Site web.....	8
i. Architecture et design.....	8
ii. Dictionnaire de données.....	10
iii. Modèle conceptuel de données.....	13
iv. Modèle logique de données.....	13
b. Algorithme de suggestion.....	14
i. Renommée et succès.....	14
ii. Profil naturel de l'utilisateur.....	14
iii. Profil dynamique de l'utilisateur.....	15
iv. Paramètres globaux évolutifs.....	15
v. Suggestion finale.....	16
4. Déroulement du projet.....	17
a. Calendrier et organisation.....	17
b. Répartition des tâches.....	17
5. Conclusion et perspective.....	18
a. Problématiques identifiées et solutions apportées.....	18
b. Améliorations envisageables.....	18
Glossaire.....	19

# **1. Introduction**

## **a. Contexte du projet**

Dans le cadre de notre troisième année de Licence Informatique au sein de CY Cergy Paris Université, il nous a été demandé d'effectuer un Projet d'Intégration. Pour celui-ci, un ensemble de sujets furent proposés et nous avons choisi celui intitulé : « Librairie Intelligente en Ligne ».

Ce sujet consiste à implémenter un service de librairie en ligne permettant de suggérer automatiquement, et avec efficacité, des livres à chaque utilisateur en fonction de son profil et d'un certain nombre d'autres paramètres.

## **b. Objectif final**

Le but de ce projet est de concevoir et réaliser un site web qui permettrait à tout utilisateur de rechercher, de consulter et « d'acheter » des livres.

L'intérêt principal étant que, une fois implémenté, le site devrait être capable de suggérer cinq livres à chaque utilisateur en fonction de son profil, de son historique d'achat et d'un ensemble de facteurs globaux. Cette suggestion doit être automatique, personnalisée et évolutive en fonction du temps et des actions qu'effectueraient les utilisateurs sur le site.

## 2. Spécifications globales

### a. Plateforme web

Le service que nous avons implémenté est un site web accessible en intranet développé en *PHP*.

Sachant que nous avons à traiter un certain nombre de données, notre site a été conçu sur une architecture 3-tiers. En plus de permettre un meilleur stockage des données, la présence d'une base de données en *SQL* permet à l'administrateur de modifier celle-ci sans perturber le bon fonctionnement du site web.

### b. Gestion des données

Les données brutes en entrées ont principalement servies à établir l'algorithme que nous utilisons pour la suggestion de livres aux clients. Elles sont toutes stockées dans les tables d'une base de données.

Pour permettre une meilleure suggestion possible de livre, ces données brutes sont séparées en 4 catégories distinctes :

- ◆ Les livres sont définis par la langue dans laquelle ils sont écrits, ce qui nous permet d'affiner les suggestions de l'utilisateur en fonction de son pays de résidence. Le genre, qui nous est utile afin de cibler les tranches d'âge ainsi que les intérêts des clients. Le prix nous donne une indication sur le type d'utilisateur ayant la capacité de l'acheter, de ce fait la suggestion propose des livres plus ou moins chers en fonction du client. Et enfin l'année de parution avec laquelle nous affinons les styles ainsi que les périodes littéraires appréciées,
- ◆ Le profil utilisateur naturel est défini par :
  - son âge, ce qui nous permet d'affiner la suggestion en sélectionnant des œuvres plus ou moins complexes,
  - son sexe, qui nous permet de mettre en avant certains genres de livres d'une manière stéréotypé,
  - son pays, qui nous donne une indication sur la langue parlée théorique de l'utilisateur afin de lui suggérer en priorité des livres correspondants avec celle-ci,
  - sa catégorie socioprofessionnelle, qui nous permettra de déterminer ses revenus et donc son pouvoir d'achat, pour estimer ce que le client pourrait dépenser en achetant des livres,
- ◆ Le profil utilisateur dynamique est défini par les actions de l'utilisateur. Il est composé des dépenses maximums de l'utilisateur selon son historique d'achat, ce qui nous permettra d'évaluer le prix que l'utilisateur est prêt à mettre pour acheter

un livre. L'historique d'achat de l'utilisateur nous permet d'établir les genres, les auteurs et les langues que l'utilisateur aime lire. Le profil dynamique est aussi constitué des livres consultés, ce qui constitue l'historique de visionnage, on obtiens donc les livres qui intéressent l'utilisateur mais qu'il n'a pas encore achetés.

- ◆ La statistique globale permet d'établir une temporalité de la suggestion (des livres d'épouvantes à Halloween, des livres spirituel pendant les Fêtes, ...). Chaque livre a une côte de popularité en fonction du nombre d'utilisateur qui ont achetés ce dit livre (les tendances). Chaque lecteur pourra évaluer ses achats, ses notes influencent le classement des livres. Les suggestions de chaque client sont aussi affectées par les historiques des profils similaires ce qui permet d'agréments le calcul.

### **c. Fonctionnalités générales**

D'une part, l'utilisateur peut procéder à son inscription, ou à sa connexion au site. Ensuite il a la possibilité de chercher un livre spécifique et de l'acheter, ou de le consulter. Pour cela il peut cliquer sur un livre sans l'acheter pour être redirigé sur une page dédiée où sont présentes toutes les informations du livre.

D'autre part, notre solution contient une partie administrateur qui offre aux responsables la possibilité d'ajouter ou de supprimer des livres.

Enfin, notre solution est capable de suggérer intelligemment des livres aux utilisateurs. Via un algorithme qui se sert des données brutes, nous attribuons une note à chaque livre ce qui établit un classement entre eux pour chaque client. À l'issue de ce calcul nous sélectionnons cinq livres que nous proposons à l'utilisateur correspondant.

### **d. Fonctionnement général de la suggestion**

L'algorithme, dans un premier temps, donne à chaque livre une note arbitraire basée sur la renommée de son auteur ainsi que sur son évaluation globale.

Dans un second temps, on compare les caractéristiques du livre avec celles du profil utilisateur naturel afin de tester la correspondance entre :

- la langue de rédaction du livre et le pays de résidence de l'utilisateur,
- le genre du livre et le sexe de l'utilisateur,
- l'année de sortie du livre et l'âge de l'utilisateur,
- le prix du livre et la catégorie socio-professionnelle de l'utilisateur.

En fonction de la correspondance entre ces caractéristiques, la note du livre augmente ou diminue.

Dans un troisième temps, nous vérifions :

- que le prix du livre ne dépasse pas la dépense maximale de l'utilisateur,

- que l'auteur, le genre et/ou l'année de parution soient représentatif de l'historique d'achat de l'utilisateur,
- si le livre est une des œuvres consultées par l'utilisateur ou non,
- si le livre est présent dans l'historique d'achat des profils similaires à celui de l'utilisateur.

Pour chaque vérification positive nous augmentons le score du livre.

Enfin, la période de l'année et les tendances font évoluer le score final des livres. Selon la période de l'année, des genres seront plus mis en avant que d'autres en augmentant artificiellement les scores des livres correspondants. Les tendances évoluent tout au long de l'année et sont calculées en fonction du nombre de livres vendu sur une période de temps définie.

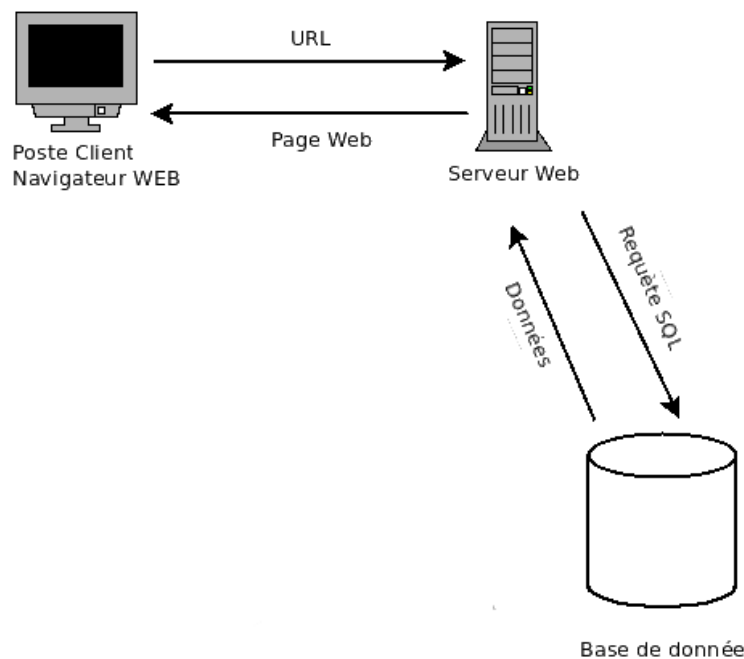
### 3. Conception technique

#### a. Site web

Le site web est le support principal de notre projet, il permet à l'utilisateur d'interagir avec la base de données. Dans cette partie nous allons présenter une description physique et technique du site et de sa base de données.

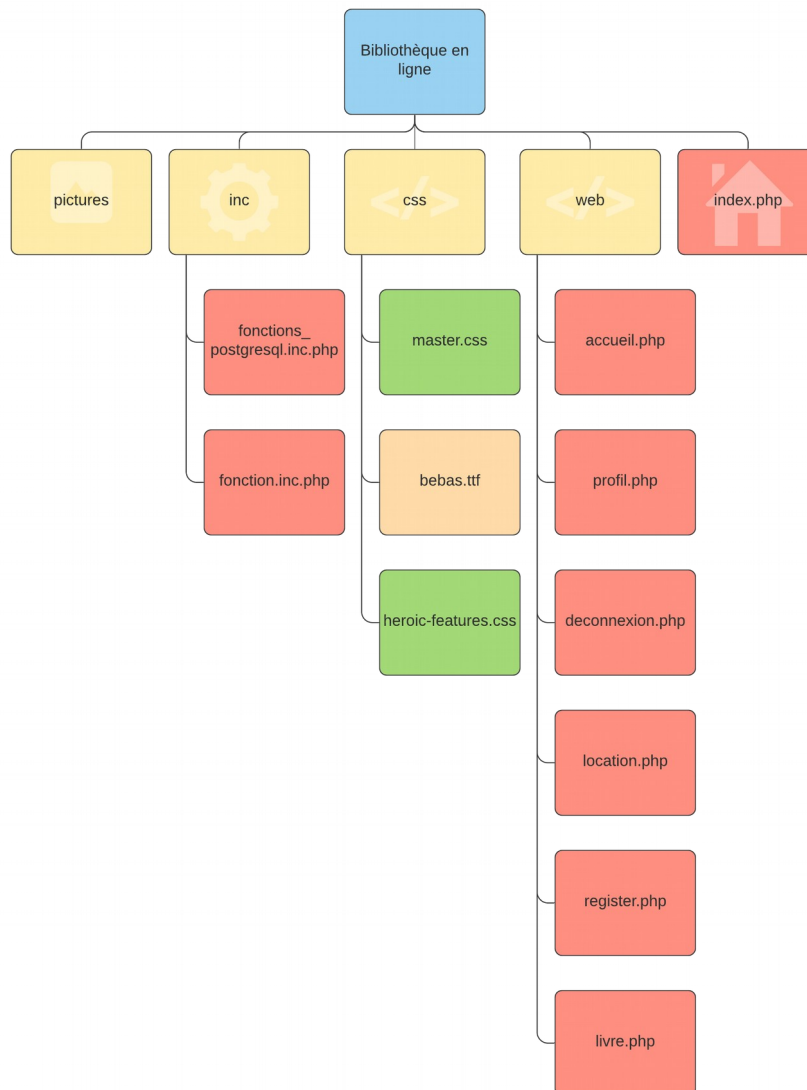
##### i. Architecture et design

Nous avons choisi une architecture 3-tiers afin de faciliter le stockage des nombreuses données que nous utilisons.



Cette architecture 3-tiers est composée, d'un coté, d'une base de données contenant toutes les informations des utilisateurs, des livres, des auteurs, etc ... Et, d'un autre coté, d'un site web programmé en *PHP* et *HTML* interprété par le navigateur de l'utilisateur.





Le design du site a été pensé pour que chaque page puisse être accessible en un maximum de trois clics. De ce design résulte l'architecture ci-dessus.

## ii. Dictionnaire de données

Dans le tableau suivant, nommé dictionnaire de données, on a inscrit toutes les données présentes dans les tables de notre base de données.

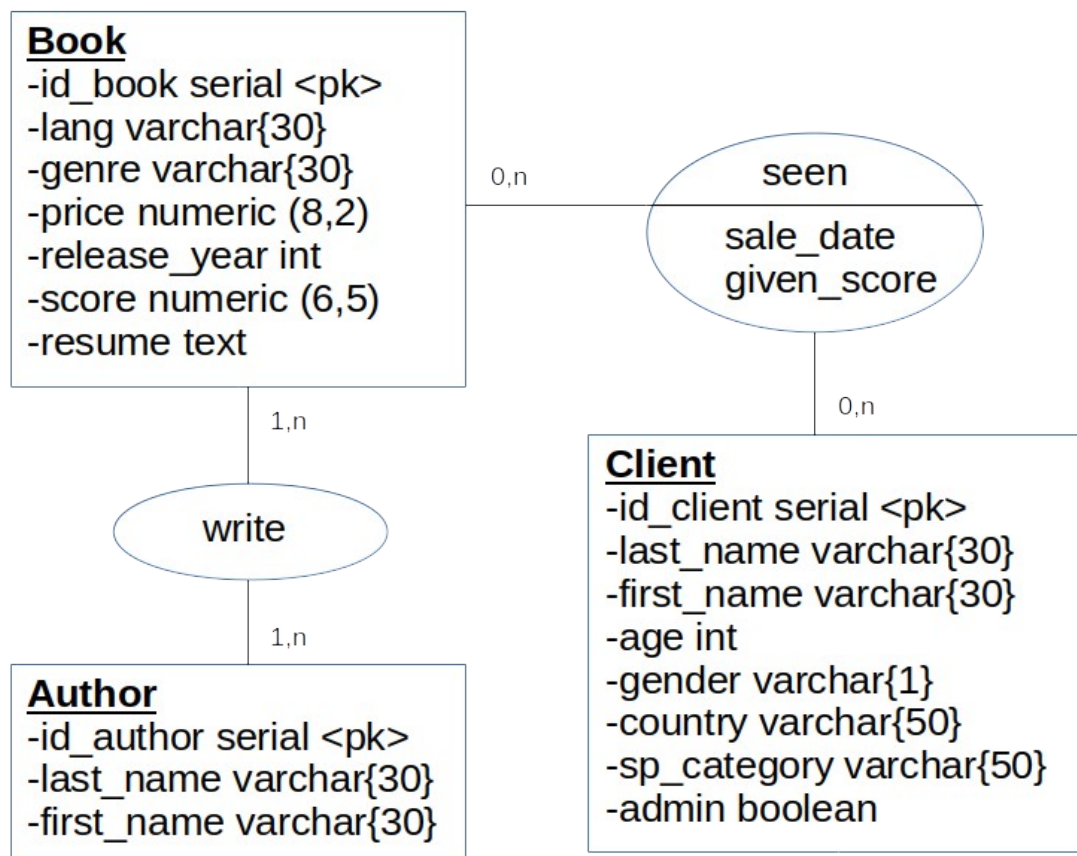
Nom de la table	Nom du champ	Type SQL	Taille	Description	Remarque
Book	id_book	serial		Identifiant numérique d'un livre	Clef primaire, unique, non nul
	title	varchar	50	Titre d'un livre	Non nul
	lang	varchar	2	Langue d'écriture d'un livre	Basé sur la norme ISO 639-1

RADOLANIRINA Yaël, LIGUORI Quentin

	genre	varchar	30	Genre littéraire d'un livre	
	count_note	int		Nombre d'avis donnés à u livre	Non nul, 0 par défaut
	release_year	int		Année de parution d'un livre	AAAA
	score	numeric	(6, 5)	Note d'un livre	Non nul, à 2.5/5 par défaut
	resume	text		Résumé d'un livre	
	price	numeric	(8,2)	Prix d'un livre	
Client	login	varchar	20	Identifiant alpha-numérique d'un client	Clef primaire, unique, non nul
	password_client	varchar	60	Mot de passe d'un utilisateur	Crypté, non nul
	last_name	varchar	30	Nom d'un client	
	first_name	varchar	30	Prénom d'un client	
	age	int		Âge d'un client	>0
	gender	varchar	1	Sexe d'un client	« f » ou « m »
	country	varchar	50	Pays de résidence d'un client	
	sp_category	varchar	50	Catégorie socio-professionnelle d'un client	Une des 8 catégories définies par l'I.N.S.E.E : -Agriculteurs exploitants -Artisans, commerçants, chefs d'entreprise -Cadres et professions intellectuelles supérieures -Professions intermédiaires -Employés -Ouvriers -retraités -autres sans activité professionnelle
	admin	boolean		Vrai si c'est un admin Faux sinon	Par défaut en Faux
Author	id_author	serial		Auteur d'un livre	Non nul, clef primaire, unique
	last_name	varchar	30	Nom d'un auteur	Non nul

	first_name	varchar	30	Prénom d'un auteur	Non nul
Write	id_author	serial		Auteur d'un livre	Non nul, clef étrangère
	id_book	serial		Identifiant numérique d'un livre	Non nul, clef étrangère
Seen	id_client	serial		Identifiant numérique d'un client	Clef étrangère, non nul
	id_book	serial		Identifiant numérique d'un livre	Non nul, clef étrangère
	sale_date	date		Date d'achat	Au format jj-mm-aaaa
	given_score	numeric	(1,0)	Note de l'utilisateur donné au livre	Compris entre 1 et 5

### iii. Modèle conceptuel de données



#### iv. Modèle logique de données

*book* (#id\_book, title, lang, genre, release\_year, score, resume, price)

*client* (#id\_client, last\_name, first\_name, age, gender, country, sp\_category, admin)

*author* (#id\_author, last\_name, first\_name)

*write* (#id\_author, #id\_book)

*seen* (#id\_client, #id\_book, sale\_date, given\_score)

#### b. Algorithme de suggestion

La suggestion de livre est l'essence même de notre projet. Nous allons donc, dans cette partie, décrire la manière dont les livres sont notés et triés. Pour cela, nous allons détailler l'ensemble des tests effectués sur un livre, que l'on nommera « A » pour l'exemple, par soucis de clarté.

##### i. Renommée et succès

Dans un premier temps, nous allons donner une note de base à chaque livre.

À l'aide d'une requête SQL imbriquée et de la fonction *COUNT*, nous allons interroger la base de données afin de récupérer le nombre de livres vendus par l'auteur de « A ». Notre note de base sera la moyenne entre ce dernier nombre et la note donnée à « A » par l'ensemble des utilisateurs qui l'ont acheté.

##### ii. Profil naturel de l'utilisateur

Dans un second temps, nous allons comparer une partie des caractéristiques de « A » avec le profil naturel de l'utilisateur. Ce profil est, pour notre réalisation, composé du pays de résidence, du sexe, de l'âge et de la catégorie socio-professionnelle de l'utilisateur.

Nous avons donc créé une variable arbitraire qui sera incrémenté de 10 unité à chaque critère validé.

Pour commencer nous allons récupérer, par une simple requête, le pays de résidence de l'utilisateur dans la base de données et vérifier que la langue de rédaction de « A », stockée dans la base au format *ISO 639-1*, correspond bien à une des langues principales du pays. Pour cela on s'aide d'un tableau *PHP* contenant les correspondances entre les langues (au format *ISO 639-1*) et un ensemble de pays où cette langue est principalement parlée.

Pour continuer, on récupère dans la base de donnée le sexe de l'utilisateur et on le compare, grâce à un tableau *PHP* qui fait une correspondance stéréotypée entre sexe et genre littéraire, au genre littéraire de « A ». Ainsi, par exemple, si « A » est un livre de romance, il sera valorisé pour une femme et pas pour un homme. Et inversement s'il est un livre de science-fiction.

Ensuite, nous définissons un intervalle de 20 ans avant et après l'année de naissance de l'utilisateur, puis on examine si l'année de parution du livre est présente ou non dans cet intervalle.

Enfin, nous nous attardons sur le prix du livre et la catégorie socioprofessionnelle de l'utilisateur. Cette dernière, par un souci de réalisme, correspond à un des groupes socioprofessionnels définis par la *PCS* de l'*INSEE* en 2003. Nous avons défini, pour chaque catégorie, un prix limite au-delà duquel le livre n'est pas valorisé sur ce critère. Le test à effectuer revient à vérifier que le prix de « A » ne dépasse pas le seuil correspondant à la catégorie socioprofessionnelle préalablement sélectionnée.

### iii. Profil dynamique de l'utilisateur

Dans un troisième temps, nous allons comparer une partie des caractéristiques de « A » avec le profil dynamique de l'utilisateur. Ce profil correspond à l'ensemble des actions effectuées par les utilisateurs sur le site, c'est-à-dire principalement les historiques de visionnage et d'achat.

Ici aussi nous créons une variable arbitraire, nommée « *\$dynScore* » qui sera incrémentée de vingt unités par critère validé, ainsi la validation de ces tests aura une plus grande influence sur le résultat de la suggestion que les tests sur profil naturel de l'utilisateur.

Tout d'abord nous vérifions que le prix du livre ne dépasse pas la dépense maximale de l'utilisateur. Pour cela nous sélectionnons le prix du livre le plus cher acheté par l'utilisateur grâce à la fonction *SQL MAX*, que nous comparons au prix de « A ». Si ce dernier est inférieur ou égale à la somme maximum dépensée par l'utilisateur, alors le critère est validé.

Ensuite, on va sélectionner, par une requête imbriquée, tous les livres achetés par l'utilisateur. Pour chacun d'eux on vérifie si l'auteur, le genre et/ou l'année de parution correspondent à ceux de « A ». Pour chacun de ces trois tests validés on incrémente « *\$dynScore* » de vingt.

Puis nous vérifions si « A » est un des livres consultés par l'utilisateur en utilisant une simple requête.

Enfin, nous allons essayer de voir si le livre est présent dans l'historique d'achat des profils similaires à celui de l'utilisateur. Pour cela on va récupérer l'ensemble des profils ayant le même âge, le même pays de résidence, le même sexe et la même catégorie socioprofessionnelle que l'utilisateur, puis on les place dans un tableau *PHP*. On parcourt ensuite l'ensemble des profils et, pour chacun d'entre eux, on vérifie si « A » est présent dans leur historique d'achat. Si c'est le cas on incrémente un compteur, sinon on incrémente un autre compteur et une fois l'ensemble du tableau parcouru on compare ces deux compteurs. On considère le critère validé lorsque le « A » est présent dans plus de 50 % des historiques d'achat.

#### iv. Paramètres globaux évolutifs

Dans un quatrième temps, nous comparons « A » à deux paramètres temporels.

Ici aussi nous créons une variable arbitraire, qui sera incrémenté de 10, ce qui permettra de compter le nombre de critères validés par le livre.

D'une part nous allons mettre en valeur des genre littéraire en fonction de la période de l'année. Nous récupérons donc le mois de l'année via la fonction *PHP date()*, puis nous vérifions si le genre de « A » correspond à une des règles mises en place. Celles-ci sont :

- si nous sommes en février, alors on valorise les livres d'érotisme ou de romance,
- si nous sommes en juin, alors on valorise les livres de jeunesse,
- si nous sommes en septembre, alors on valorise les livres d'art et d'éducation,
- si nous sommes en octobre, alors on valorise les livres de fantasy et terreur,
- si nous sommes en décembre, alors on valorise les livres de spiritualité.

D'autre part, on va sélectionner via une requête complexe les cinq livres les plus vendus du mois, et vérifier si « A » en fait parti ou non.

#### v. Suggestion finale

Pour terminer, à la fin de chacune de ces quatre fonctions, on récupère la valeur de la variables arbitraire qu'on additionne dans une variable nommée « *\$bookSScore* ». Elle correspond au score de « A » pour l'utilisateur. On la place ensuite dans un tableau *PHP* avec pour clef l'identifiant numérique du livre auquel elle correspond.

Ce tableau sera donc composé des scores de chaque livre pour un utilisateur donné. On trie ensuite le tableau dans l'ordre décroissant et à l'affichage on ne retourne que les informations des cinq premiers livres, c'est à dire les livres ayant les plus grands scores, soient les livres qui correspondent le plus à l'utilisateur.

## 4. Déroulement du projet

### a. Calendrier et organisation

Pendant ce projet nous nous sommes organisé afin de pouvoir remplir les contraintes demandées durant les points d'avancement. Nous avons utilisé *GitHub* afin de nous permettre de partager les fichiers du projet à tout moment. Nous avons aussi favorisé la communication dans ce projet en nous tenant au courant des avancées de chacun via une conversation *WhatsApp*. Nous avons choisi d'héberger notre site et notre base de données sur l'intranet de CY Cergy Paris Université, car nous sommes familier avec celui-ci, ce qui nous a permis de commencer à travailler rapidement et avec efficacité.

	Implémentation Du site web	Implémentation De la BDD	Premier point d'avancement	Implémentation de l'algorithme de suggestion	Finalisation du site	Second Point d'avancement	Documentation (présentation + rapport)	Rendu + présentation
23/12/19 au 19/01/2020								
20/01/20								
21/01/2020 au 23/02/2020								
24/02/20								
25/02/2020 au 8/03/2020								
09/03/20								

### b. Répartition des tâches

Nous avons répartis les tâches équitablement, mais nous nous sommes entr aidés afin de maximiser notre production. Yaël s'est principalement occupé de l'implémentation de l'algorithme, et Quentin du développement du site ainsi que de la base de données. L'ajout de données a été effectué par l'ensemble du groupe pour que cela impacte le moins possible le reste du développement.

## **5. Conclusion et perspective**

### **a. Problématiques identifiées et solutions apportées**

D'un point de vue général, ce projet est une réussite.

D'une part nous avons réussi à créer une librairie intelligente en ligne, permettant de rechercher des livres présents dans une base de données par leurs titres, de les « acheter » tout en les associant au compte sur lequel l'utilisateur est connecté.

D'autre part, notre solution permet aussi l'édition de la base de données depuis l'interface administrateur du site qui permet de gérer l'ajout et la suppression de livres et l'ajout d'auteurs.

Enfin, le site comprend un système intelligent qui va permettre la suggestion des cinq livres correspondant le plus à l'utilisateur.

### **b. Améliorations envisageables**

Malgré les réussites du projet, notre programme n'en est pas moins développable. En premier lieu, nous pourrions ajouter la couverture de chaque livre. De plus il serait intéressant de permettre à l'utilisateur d'effectuer une recherche par auteur, genre ou encore année de parution. Enfin nous pourrions ajouter des critères lors du calcul de la suggestion, afin de perfectionner cette dernière.



## Glossaire

- ◆ COUNT : En SQL, la fonction d'agrégation COUNT() permet de compter le nombre d'enregistrement dans une table (<https://sql.sh/fonctions/agregation/count>).
- ◆ date() : Fonction PHP qui formate une date/heure locale (<https://www.php.net/manual/fr/function.date.php>).
- ◆ GitHub : est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de version Git (<https://github.com/>).
- ◆ HTML : (Hypertext Markup Language) est le langage de balisage conçu pour représenter les pages web.
- ◆ INSEE : L'Institut national de la statistique et des études économiques collecte, produit, analyse et diffuse des informations sur l'économie et la société françaises (<https://www.insee.fr/fr/accueil>).
- ◆ ISO 639-1 : est une norme internationale de l'Organisation internationale de normalisation (ISO) qui définit des codes pour la représentation des noms de langues.
- ◆ MAX : dans le langage SQL, cette fonction permet de retourner la valeur maximale d'une colonne dans un set d'enregistrement (<https://sql.sh/fonctions/agregation/max>).
- ◆ PCS : les professions et catégories socioprofessionnelles sont une nomenclature statistique permettant de classer des métiers (<https://www.insee.fr/fr/information/2497952>) .
- ◆ PHP : (officiellement *Hypertext Preprocessor*) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML (<https://www.php.net/manual/fr/intro-whatis.php>).
- ◆ SQL : est un langage informatique normalisé servant à exploiter des bases de données relationnelles(<https://sql.sh/>).
- ◆ WhatsApp : est une application mobile multiplateforme qui fournit un système de messagerie instantanée chiffrée de bout en bout aussi bien par Internet que par les réseaux mobiles (<https://www.whatsapp.com/?lang=fr>).