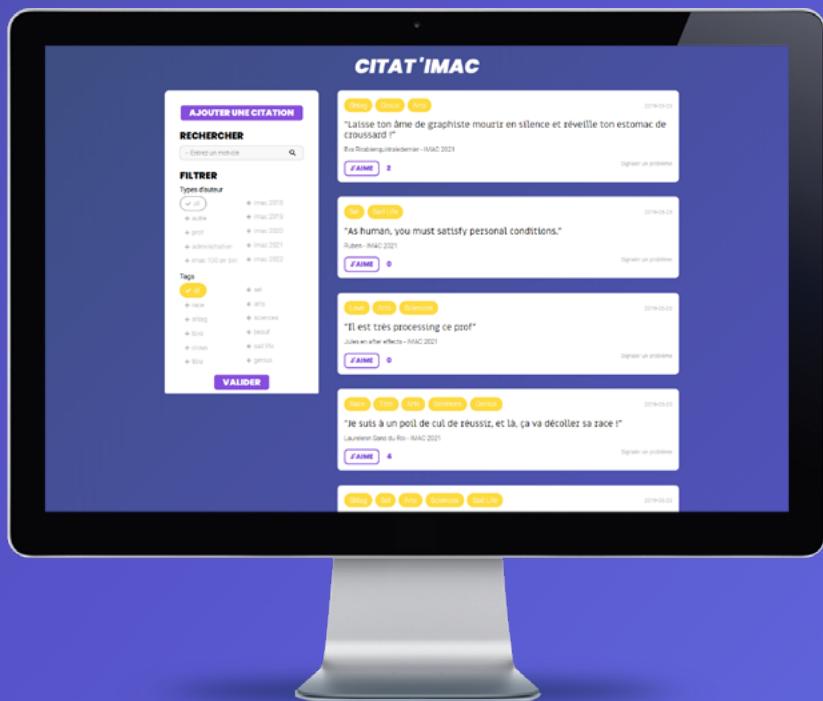


# CITAT'IMAC

PROJET PROGRAMMATION WEB



IMAC  
26.05.2019

Eva Benharira  
Amandine Kohlmuller  
Laurelenn Sangaré  
Margaux Vaillant

# INTRODUCTION

Dans le cadre du cours de web back-end du second semestre de l'IMAC, nous avons eu l'opportunité de réaliser un service Web Restful en utilisant PHP et SQL. L'objectif était de pouvoir manipuler une base de données avec les quatre principales actions : Create, Read, Update et Delete. Le thème du site internet étant libre, nous avons pris un moment pour réfléchir à un projet motivant, utile et répondant aux différents critères du sujet. C'est ainsi que le site Citat'IMAC est né.

**LIEN DU SITE**

<http://perso-etu.u-pem.fr/~akohlmul/citatimac/>

LIEN DU GIT

<https://github.com/EvaBien/Citat-imacs>

Nous reviendrons, dans un premier temps, sur les origines du projet, les caractéristiques du site sans oublier l'équipe qui l'a développé. Nous verrons ensuite les détails des différentes phases du projet et les éléments produits à ce moment de l'avancée. Nous prendrons également le temps de décrire notre méthodologie de travail et les différents outils que nous avons mis en place. Finalement, nous ferons un bilan et un retour d'expérience du projet.

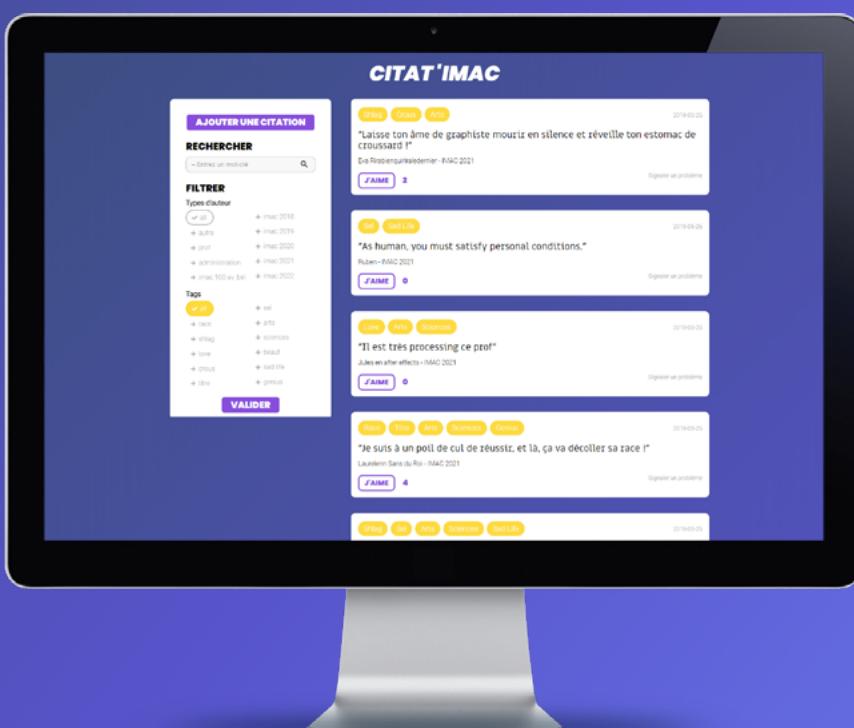


Illustration 1 : Capture d'écran du site Citat'IMAC

# SOMMAIRE

<b>1. CONTEXTE</b>	<b>04</b>
1.1 Origine du projet	04
1.2 Description et objectifs	05
1.3 Équipe	05
<b>2. CADRAGE</b>	<b>06</b>
2.1 Use case	06
2.2 Diagramme de classe	07
2.3 Maquettes	08
<b>3. PRODUCTION</b>	<b>10</b>
3.1 Structure du HTML	10
3.2 Base de données	11
3.3 Arborescence du projet	12
3.4 MVC	13
3.5 Endpoints	14
<b>4. MÉTHODOLOGIE</b>	<b>16</b>
4.1 Organisation	16
4.2 Répartition des tâches	18
4.3 Planning	19
<b>5. BILAN</b>	<b>20</b>
5.1 Synthèse récapitulative	20
5.2 Pistes d'améliorations	21
5.3 Retour d'expérience	22
<b>CONCLUSION</b>	<b>25</b>

# 1. CONTEXTE

## 1.1 ORIGINE DU PROJET

Ces quelques phrases résument assez bien les origines de notre projet :

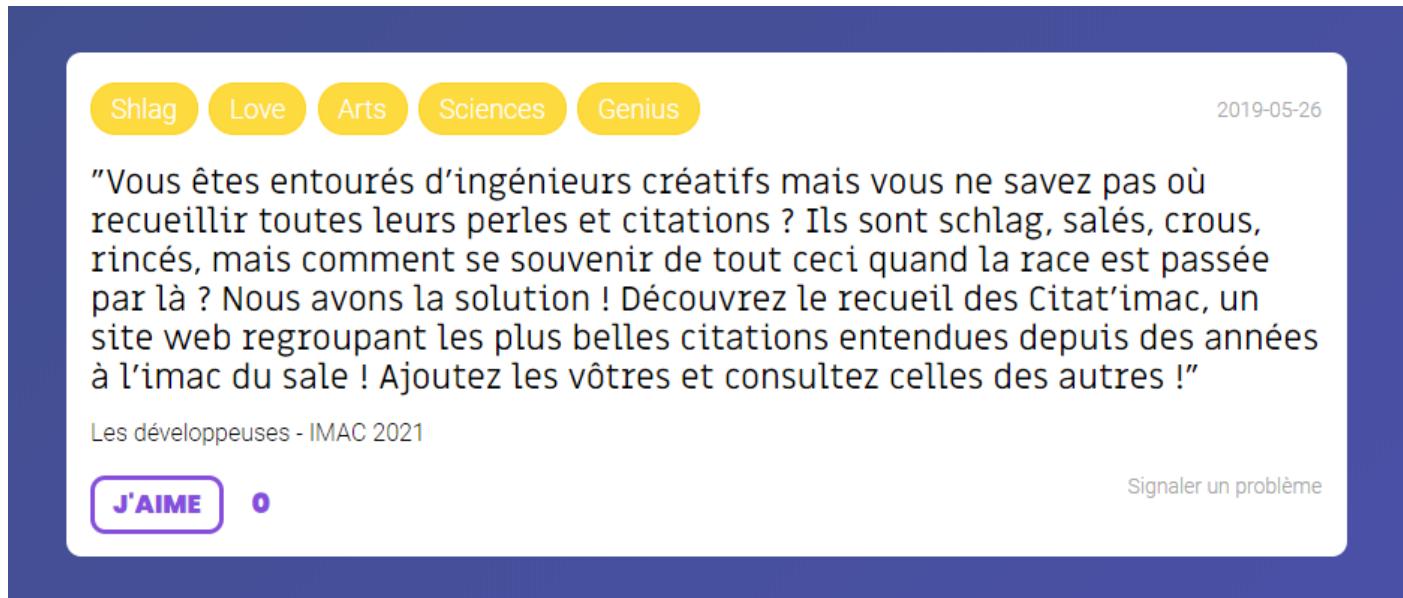


Illustration 2 : Capture d'écran, Pitch récapitulant le contexte et le concept du site ajouter en tant que citation sur le site Citat'IMAC

Par ailleurs, en discutant avec des élèves d'autres années, nous nous sommes rendues compte que divers documents Google doc existaient déjà pour rassembler les perles entendues par les uns et les autres. Certains possédaient également des notes dans leur téléphone. Cependant, il n'existe aucune centralisation de l'information et c'est à ce manque que nous souhaitons répondre avec le site Citat'IMAC.

## **1.2 DESCRIPTION ET OBJECTIFS**

Le site est donc un recueil de citations et de paroles des personnes de l'IMAC. Dans la première version du Citat'imax il y a seulement un affichage utilisateur et pas de gestion de sessions. En revanche, une page administrateur est mise en place afin de gérer les citations signalées par les utilisateurs.

Ces derniers peuvent donc ajouter des citations, consulter, en fonction de plusieurs critères, celles enregistrés par les autres utilisateurs et signaler des citations. Cette dernière action envoie un mail aux développeuses qui peuvent ainsi agir en conséquence : modifier la citation, la supprimer ou ne rien faire.

L'affichage des citations est donc dynamique et un système de «like» est présent sur chacune d'entre-elles.

Nous proposons plusieurs critères de recherche et d'affichage des citations :

- par mots clé (dans le contenu)
- par type d'auteur (catégorie : prof, promotion)
- par tags (préalablement choisis par l'équipe de développeurs)

L'objectif du projet est donc de mettre en pratique les acquis de la formation dans un projet motivant et profitable aux élèves de l'IMAC. En effet, nous pensons que l'aspect ludique du site qui permet à chacun d'ajouter des citations ou de liker celles qu'il préfère est de nature à plaire aux étudiants.

## **1.3 ÉQUIPE**

L'équipe du projet est constituée de Eva Benharira, Amandine Kohlmuller, Laurelenn Sangaré et Margaux Vaillant. Amandine Kohlmuller est cheffe de projet et Laurelenn Sangaré est directrice technique. Nous détaillerons les rôles de chacunes dans la suite de ce rapport.

# 2. CADRAGE

Avant de nous lancer dans l'écriture du code, nous sommes passées par une phase de cadrage du projet. Cela nous a permis de mieux nous rendre compte des objectifs, de la structure du programme et de son fonctionnement.

## 2.1 USE CASE

Le use case, ou cas d'utilisation, fut la première étape de représentation du projet. Ce diagramme définit les différents cas d'utilisation et les exigences fonctionnelles du site.

Ainsi, à cette étape du projet, nous avions connaissance des différents types d'actions qu'un utilisateur ou un administrateur devra être capable de réaliser sur le site.

Concernant le signalement et la partie administration du site, nous avons trouvé un moyen de contourner la question des SESSIONS. En effet, n'ayant pas abordé ce point en cours et surtout manquant de temps, nous avons choisi de créer une page administrateur, accessible et visible par quiconque possédant l'url. Néanmoins, pour interagir avec les éléments de cette page, c'est-à-dire pour traiter le signalement d'une citation, il faut avoir connaissance d'un mot de passe, ce qui représente une première forme de sécurité. Ce mot de passe est bien évidemment "hashé" dans le code. Nous avons également concaténé une chaîne de caractères au mot de passe original afin d'éviter les attaques par dictionnaire.

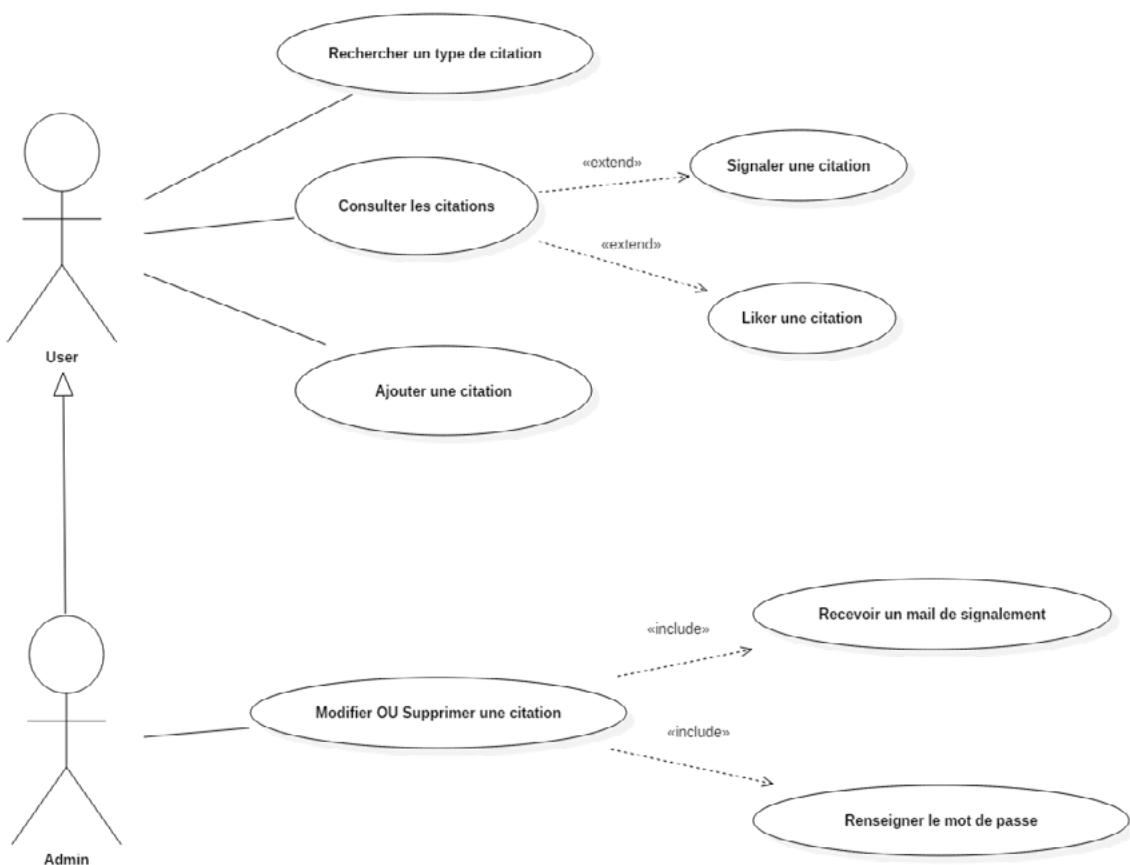


Illustration 3 : Use case du site Citat'IMAC

## 2.2 DIAGRAMME DE CLASSE

La seconde étape du cadrage concernait la structure de la base de données. Pour cela, nous avons réalisé un diagramme de classe avec l'application Star UML. Le schéma met en évidence les différentes tables de la base données, leurs attributs ainsi que les cardinalités.

On observe ainsi la centralité de la classe citation dont découlent les autres. Une table associative permet de faire le lien entre les classes citations et tags de manière à pouvoir associer plusieurs tags à une citation et dans le même temps plusieurs citations à un même tag. D'une façon similaire, la table signalement permet d'associer une type de signalement à une citation et inversement.

Ce diagramme de classe n'est pas très complexe en soi, il n'y a qu'une table association et l'ensemble des éléments est assez explicite. Nous avons par ailleurs ajouté certains attributs non ou peu exploités dans la première version du site, dans l'optique de faire une V2 quand le temps nous le permettra.

Par exemple, les tags et les auteurs ont aussi un attribut likes, qui n'est pourtant pas utilisé dans notre première version du site. En effet cela est dédié à l'ajout d'une éventuellement fonctionnalité de tri par Citations, Tags ou TypesAuteur les plus likés. De même pour TypeSignalement, nous avons une description pour chaque type qui pourrait apparaître, dans la V2, afin de mieux aiguiller l'utilisateur.

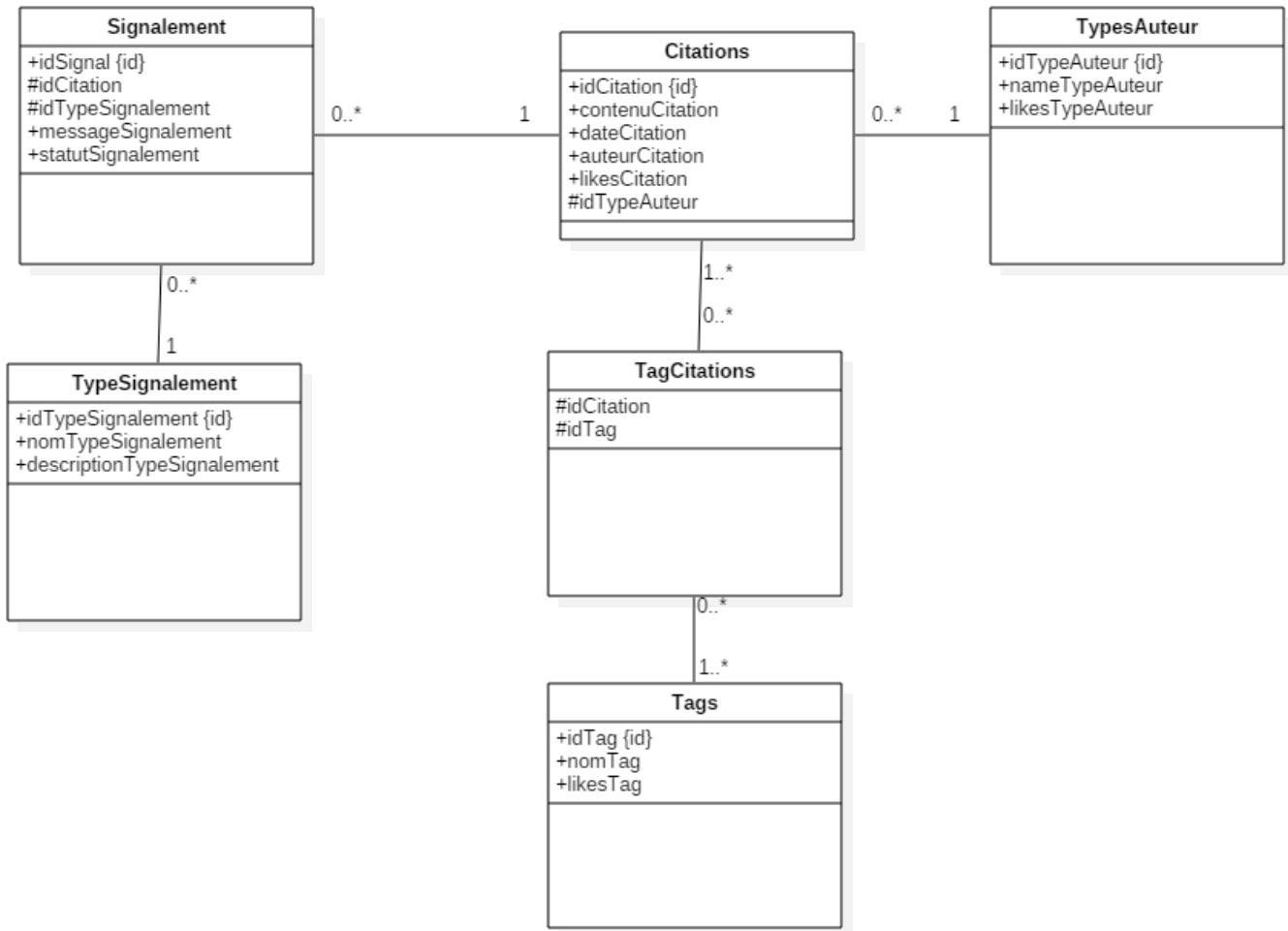


Illustration 4 : Diagramme de classe - Schéma base de données du site Citat'IMAC

## 2.3 MAQUETTES

En parallèle, nous avons conçu des wireframes puis directement des maquettes de la page d'accueil du site. Nous avons choisis un design, des couleurs, des typographies ainsi qu'une structure. L'objectif était de réaliser cela rapidement afin d'avoir une représentation visuelle du site et donc une meilleure compréhension des fonctionnalités.

La page administrateur n'a pas été designée au préalable parce qu'elle ne constituait pas une priorité. En effet, seul les administrateurs feront usage de cette page, son visuel peut donc rester très basique et fonctionnel.

Nous pouvons également remarquer que la partie navigation comprend un input "trier par" qui n'est pas présent sur la version en ligne du site. Cette dernière fonctionnalité a été abandonnée en cours de projet. En effet, elle représentait beaucoup de temps de travail en plus des autres fonctionnalités à gérer et nous avons fait le choix de nous concentrer sur le reste du site. En revanche, le fait de pouvoir faire apparaître les citations dans un ordre précis, par exemple par date d'ajout ou par nombre de "like", est un élément intéressant qui sera ajouté au site dans une version ultérieure.



Illustration 5 : Maquettes de la page d'accueil du site Citat'IMAC

Nous avons décidé de gérer l'ajout de citation et le signalement avec des pop-up, réduisant ainsi le nombre de pages à deux : index et admin.

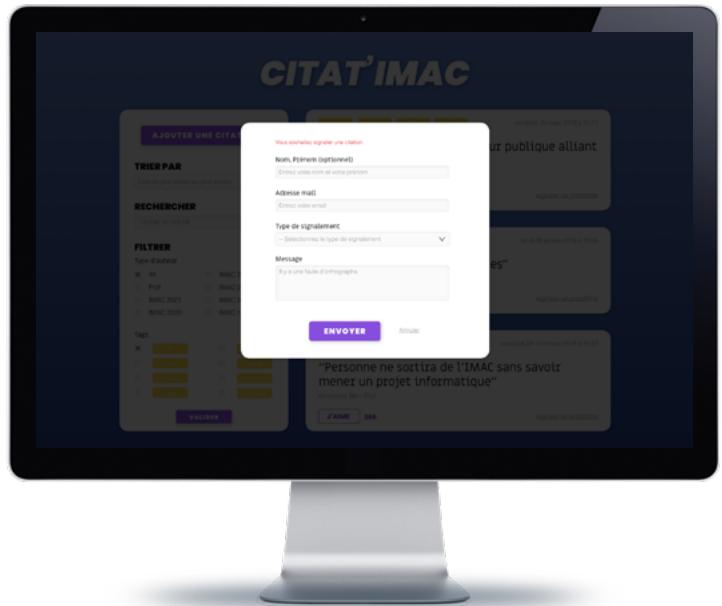


Illustration 6 : Maquette du pop-up « Signaler une citation »

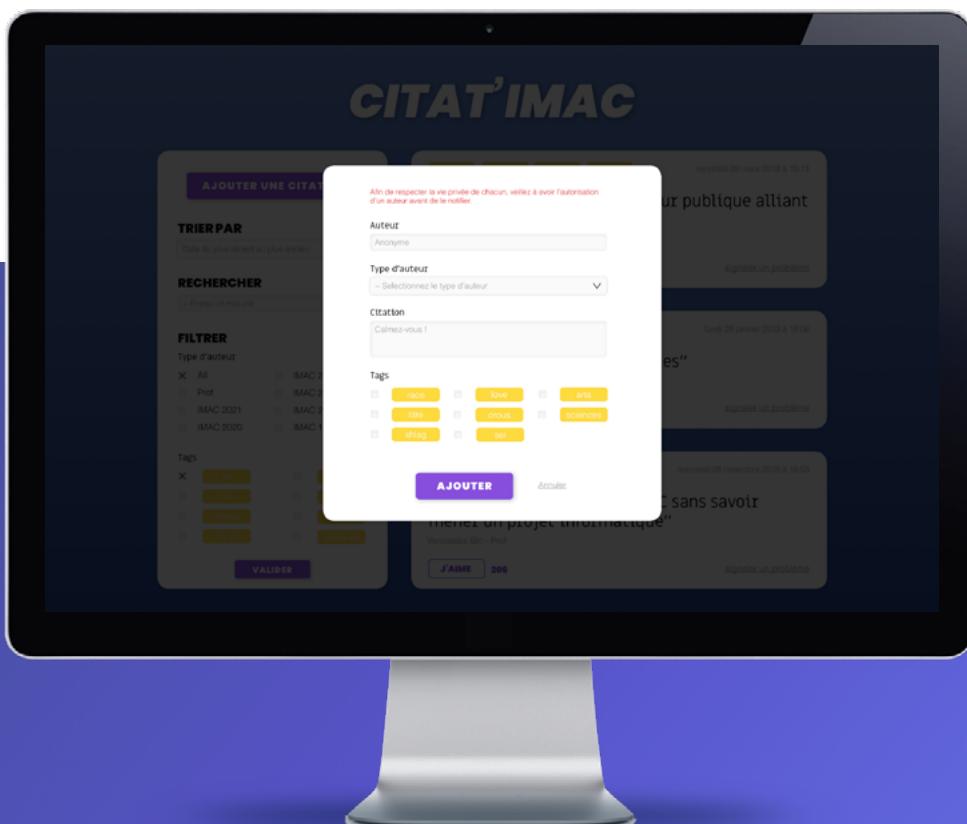


Illustration 7 : Maquettes du pop-up « Ajouter une citation »

# 3. PRODUCTION

Après avoir bien défini et cadré notre projet, nous sommes entrée en phase de production. Notre méthode de travail a également évolué tout au long du projet, nous reviendrons sur ce point plus en détail dans la partie méthodologie de ce rapport.

## 3.1 STRUCTURE DU HTML

A partir des maquettes et du diagramme de classe, nous avons construit la structure du code HTML. Il était important de bien penser et définir les id des div pour pouvoir aisément lier le back avec le front (css). Il nous fallait également penser qu'une grande partie si ce n'est la totalité du site serait en affichage dynamique. Cette étape réalisée, nous avons pu commencer à coder.

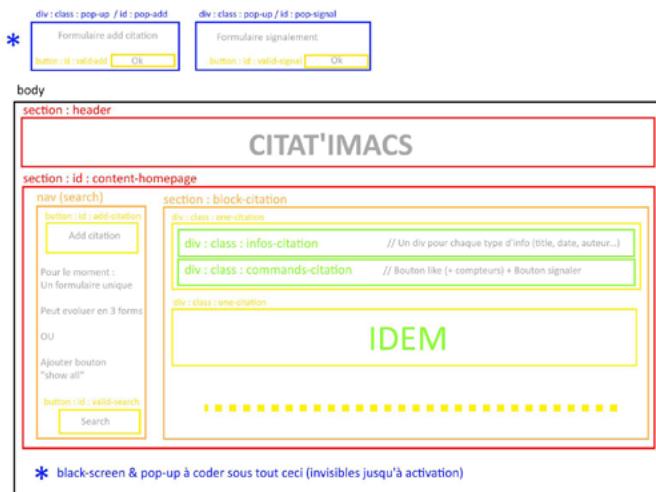


Illustration 8 : Découpage HTML de la page d'accueil du site Citat'IMAC

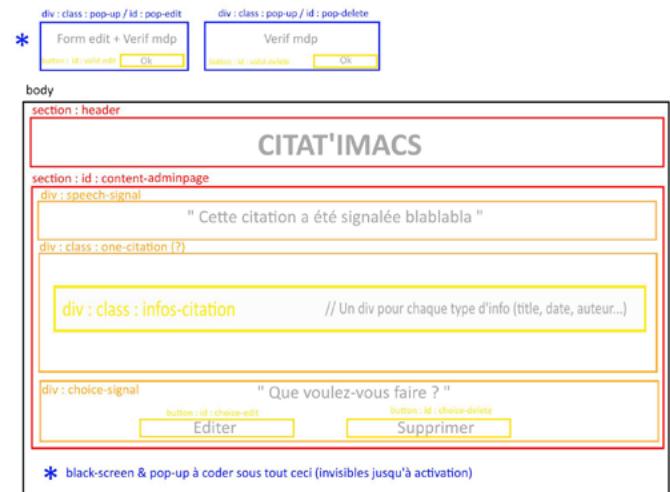


Illustration 9 : Découpage HTML de la page admin du site Citat'IMAC

## 3.2 BASE DE DONNÉES

En parallèle, nous avons créé la base de données à partir du diagramme de classe réalisé lors de la phase de cadrage. Il était important de faire attention à bien nommer chaque attributs en fonction de sa table.

La base de données a été créée sur le portail MyAdmin fourni par l'université. Suite à des difficultés à héberger, notamment en raison du nom du serveur SQL, et la nécessité de passer par la plateforme FilleZilla à chaque modification et pour obtenir une meilleure visualisation en direct de l'avancée du projet, nous avons par la suite exporté cette base de donnée pour travailler en local.

The screenshot shows the phpMyAdmin interface for the database 'akohmul\_db'. The left sidebar displays the database structure with tables: s2\_citations, s2\_signalements, s2\_tagcitation, s2\_tags, s2\_typesauteur, and s2\_typesignalement. Each table has associated columns and indexes. The main area shows a table of 6 tables with the following details:

Table	Action	Lignes	Type	Interclassement	Taille	Perte
s2_citations	Afficher Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8_general_ci	32 Kio	-
s2_signalements	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8_general_ci	48 Kio	-
s2_tagcitation	Afficher Structure Rechercher Insérer Vider Supprimer	6	InnoDB	utf8_general_ci	48 Kio	-
s2_tags	Afficher Structure Rechercher Insérer Vider Supprimer	8	InnoDB	utf8_general_ci	16 Kio	-
s2_typesauteur	Afficher Structure Rechercher Insérer Vider Supprimer	10	InnoDB	utf8_general_ci	16 Kio	-
s2_typesignalement	Afficher Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8_general_ci	16 Kio	-
<b>6 tables</b>	<b>Somme</b>	<b>29</b>	<b>InnoDB</b>	<b>utf8_general_ci</b>	<b>176 Kio</b>	<b>0</b>

Below the table, there is a 'Nouvelle table' (New Table) dialog box with fields for 'Nom:' (Name) and 'Nombre de colonnes:' (Number of columns), set to 4. The bottom of the interface shows a 'Console de requêtes SQL' (SQL Query Console).

Illustration 10 : Capture d'écran de la base de données Citat'IMAC

### 3.3 ARBORESCENCE

L'arborescence de notre site, sous les conseils de notre enseignante, comporte en vérité deux grosses catégories : Les fichiers PHP accessibles et dédiés uniquement aux développeurs, il s'agit de la partie Back-end, regroupée dans le dossier "api". Le reste des fichiers, accessibles côté utilisateur, composent le Frontend. Il s'agit des fichiers html, javascript et css.

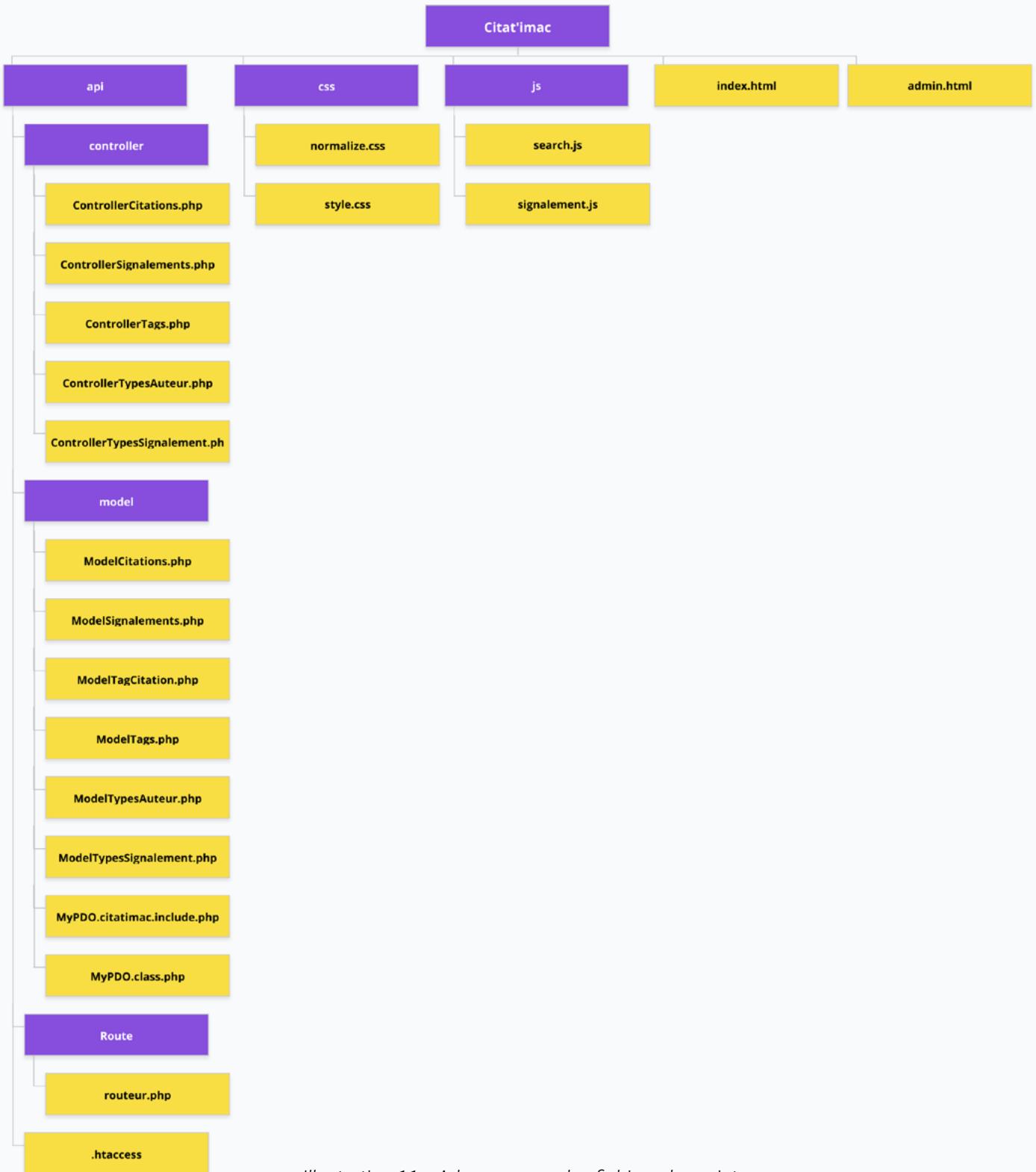


Illustration 11 : Arborescence des fichiers du projet

### 3.4 MVC

En réalité, notre arborescence repose sur le principe du MVC, bien que nous l'ayons quelque peu modifié.

MVC, ou Model-View-Controller, est un modèle d'organisation le code de notre site web. Il repose sur le principe suivant : La "View" est l'interface graphique visible par le client. Une interaction avec celle-ci, par le biais du javascript, va entraîner une requête. Cette requête sera envoyée (via un routeur ou non) à la partie "Controller" du site. C'est la partie intelligente et logique de notre projet, celle qui va traiter les cas particuliers entre autres. Une fois que le "Controller" a traité la requête reçue par le client, il demande au "Model" d'interagir avec la base de données en conséquence. Le "Model" peut ainsi récupérer, modifier, insérer ou supprimer des données de la base. Après cela, le "Model" envoie une réponse au "Controller", qui envoie lui-même l'information à la "View", qui sera modifiée en fonction des données reçues ou manipulées.

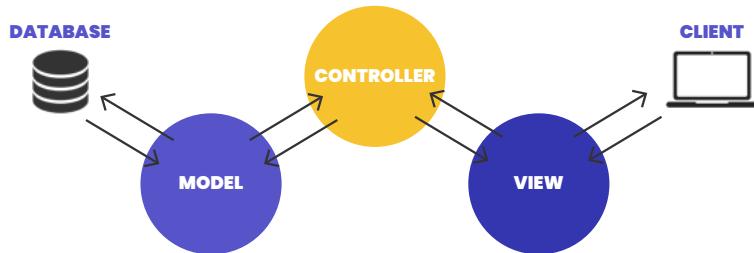


Illustration 12 : Schéma explicatif du MVC

Ce modèle est cependant à adapter en fonction du projet, s'il est trop conséquent, il faudra peut-être rediscuter l'ensemble de l'arborescence du projet. S'il est trop petit, la division du code peut paraître inutile, cependant cela permet aux novices de bien comprendre la communication et le rôle de chaque type de fichiers au sein d'un site ou programme. Dans notre cas, nous avons ajouté un dossier "Route" pour signaler que ce fichier allait gérer toutes les redirections des requêtes, ainsi qu'un dossier "Javascript", pour faire le lien entre la View et le Controller.

Ce diagramme de séquence illustre bien le fonctionnement du site avec l'exemple de l'ajout d'une citation.

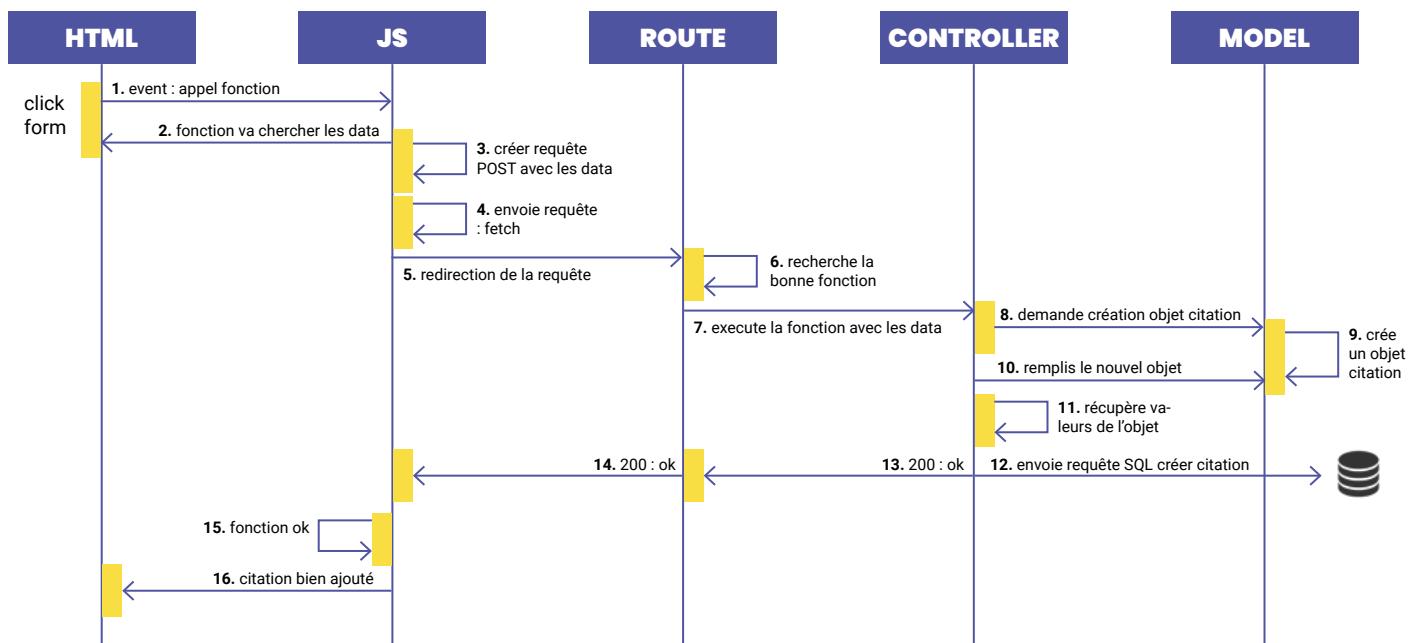


Illustration 13 : Diagramme de séquence système d'ajout d'une citation

## 3.5 ENDPOINTS

Les endpoints présentés ici sont en vérité les différentes url que peut recevoir le routeur afin d'appeler la fonction php associée dans le controller.

Dans un premier temps, nous cherchions à ce que le .fetch(url, data) dans le javascript prenne en premier argument les url présentées ci-dessous (nos endpoints). Nous avions créé un fichier .htaccess pour que toutes ces urls soient renvoyées vers le routeur. Malheureusement, après plusieurs jours de recherche, nous n'arrivions toujours pas à faire cette redirection vers routeur.php.

C'est pourquoi nous avons décidé d'intégrer l'url correspondant à la requête directement dans les données (data) du fetch, et de lui donner en premier argument le lien vers le routeur, à savoir :

```
function AllTags(){
    let formData = new FormData();
    let data = new Object();
    data['url'] = './tags/All';
    formData.append("url", './tags/All');

    data = JSON.stringify(data);
    formData.append('getData', data);

    fetch('./api/Route/routeur.php', {
        method: "POST",
        body: formData})
```

Illustration 14 : Exemple avec AllTags- lignes 6 à 17 dans search.js

Ce qui nous a renvoyé à un deuxième problème : même dans le cas où, à l'origine, nous faisions simplement une méthode GET, le fait d'envoyer des données (l'url correspondant à la requête) dans le fetch nous a contraint à utiliser une méthode POST même lors d'une simple récupération de données.

Le troisième problème auquel nous avons été confrontées : les méthodes PUT et DELETE.

Dans le routeur pour les POST (et GET), nous récupérions l'url transmise par le fetch dans une variable, et le reste de la requête dans l'autre grâce aux superglobales \$\_POST, \$\_GET.

Malheureusement, ces superglobales n'existent pas pour PUT et DELETE. Après maintes recherches pour récupérer, plus ou moins de la même manière, nos données via les méthodes PUT et DELETE (voir ci-dessous), nous avons finalement décidé d'utiliser à nouveau des méthodes POST.

```
parse_str(file_get_contents('php://input'), $_PUT);
```

Illustration 15 : Type de méthode que nous avons essayé d'appliquer pour récupérer les données avec PUT

Pour conclure sur les endpoints et le routage, nous avons rencontré beaucoup de difficultés sur cette partie, dues à notre manque de connaissance et de pratique sur le sujet. Nous avons tout de même réussi à créer le routeur. Néanmoins nous avons été contraintes de n'utiliser qu'une partie des fonctionnalités et méthodes que PHP et Javascript nous offrent sur la gestion de requêtes, particulièrement en n'utilisant que des méthodes POST.

## URL DE CITATIONS

**./citations/New**  
Crée une nouvelle citation.  
Appel : `apiCreateCitation($request);`

**./citations/All**  
Récupère toutes les citations du site  
Appel : `api GetAllCitation($request);`

**./citations/Id**  
Récupère les citations en fonction de leur ID.  
Appel : `apiGetCitationById($request);`

**./citations/Tags**  
Récupère les citations en fonction des tags qui leurs sont associés.  
Appel : `apiGetCitationByTags($request);`

**./citations/Keyword**  
Récupère les citations en fonction d'un mot que l'utilisateur recherche.  
Appel : `apiGetCitationByKeyword($request);`

**./citations/Typesauteur**  
Récupère les citations en fonction du type de leur auteur  
Appel : `apiGetCitationByTypeAuteur($request);`

**./citations/TagsKeyword**  
Récupère les citations en fonction des tags qui leurs sont associés ET d'un mot que l'utilisateur recherche  
Appel : `apiGetCitationByTagsAndKeyword($request);`

**./citations/TypesauteurKeyword**  
Récupère les citations en fonction du type de leur auteur ET d'un mot que l'utilisateur recherche  
Appel : `apiGetCitationByTypeAuteurAndKeyword($request);`

**./citations/TypesauteurTags**  
Récupère les citations en fonction du type de leur auteur ET en fonction des tags qui leurs sont associés  
Appel : `apiGetCitationTypeAuteurAndTags($request);`

**./citations/Allfactors**  
Récupère les citations en prenant en compte tous les différents facteurs (tags, auteurs et mot recherché par l'utilisateur)  
Appel : `apiGetCitationByAll($request);`

**./citations/Update**  
Met à jour la citation  
Appel : `apiUpdateCitation($request);`

**./citations/Delete**  
Supprime la citation  
Appel : `apiDeleteCitation($request);`

**./citations/GetLikes**  
Récupère le nombre de like d'une citation  
Appel : `getCitationLikes($request);`

**./citations/UpdateLikes**  
Met à jour les likes d'une citation  
Appel : `updateCitationLikes($request);`

## URL DES FILTRES

**./tags/All**  
Récupère tous les tags  
Appel : `api GetAllTags($request);`

**./typesAuteur/All**  
Récupère tous les types d'auteurs  
Appel : `api GetAllTypeAuteurs($request);`

## URL DE SIGNALEMENTS

**./typesSignalement/All**  
Récupère tous les types de signalements  
Appel : `api GetAllTypesSignalement($request);`

**./signalement/New**  
Crée un signalement  
Appel : `apiCreateSignalement($request);`

**./signalement/Id**  
Récupère le signalement en fonction de son ID  
Appel : `apiGetSignalementById($request);`

**./signalement/Nothing**  
"Ne fait rien" pour modifier une citation signalée.  
Appel : `NothingSignalement($request);`

# 4. MÉTHODOLOGIE

La méthode de travail doit être adaptée à tous les membres de l'équipe pour garantir la réussite d'un projet. Nous allons voir l'organisation mise en place et son fonctionnement, ce qui nous permettra de revoir les différentes étapes du projet.

## 4.1 ORGANISATION

Nous sommes passés par plusieurs phases dans l'organisation du projet. Dans un premier temps, Amandine Kohlmuller a mis en place les outils de communication interne, de partage des ressources et de suivie des tâches. Ainsi, nous avions une messagerie instantanée de groupe, permettant de transmettre des informations rapidement, un Google Drive pour stocker et partager les ressources comme les comptes rendus de réunions ou les diagrammes et un Trello. Ce dernier est une application qui propose de gérer un projet sous forme de cartes représentant des tâches et qui peuvent changer d'états (idée, à faire, en cours, en attente de validation, fait). Géré par la cheffe de projet et mis à jour par l'équipe, cet outils de gestion nous a fait gagné en efficacité.

Par ailleurs, Eva Benharira a mis en place un git partagé afin de pouvoir coder toutes en parallèle et ainsi avancer simultanément sur les différentes parties du projet. La plupart des membres étant novices dans l'utilisation de git, nous avons choisi de passer par l'interface GitKraken pour faciliter les commit via une interface simplifiée. De plus, pour ne pas embrouiller les membres dans leur découverte de git et puisque chacune travaillait sur des fichiers différents, nous n'avons en premier lieu pas mis en place de branche puisque les risques de conflits restaient limités. Suite à des remarques faite par notre professeure et les risques de conflit augmentant, nous avons cependant créé quelques branches sur la fin du projet. Bien que pour celles qui débutaient sur git, se coordonner en parallèle via la conversation de groupe restait encore une méthode privilégiée, elles ont ainsi pu être initiée au fonctionnement des branches et seront ainsi plus à même d'y recourir dans d'autres projets.

Concernant la méthodologie, nous aurions pu partir sur une méthode de gestion de projet classique avec un cahier des charges fonctionnel et un planning détaillé comme un Gantt. Cependant, et sous les conseils de notre professeur, nous avons opté pour une méthode agile. En effet, avoir le regard fixé sur le sommet de la montagne à gravir peut vite décourager et créer une baisse de motivation générale. En revanche, en ayant une brève idée de ce que l'on trouvera au sommet, mais en voyant clairement le chemin pas à pas, il est plus facile et plus efficace d'arriver jusqu'au sommet.

Une méthode agile permet de fixer des objectifs à court terme en adoptant un processus itératif et donc de donner davantage de visibilité sur le déroulé du projet. Le besoin n'étant jamais figé, il est possible d'adapter les livrables et la gestion du projet en fonction des besoins. Les membres de l'équipe et les interactions sont plus importantes que le processus et les outils. Cela incite à garder un rythme de travail durable et encourage à réfléchir aux moyens de devenir plus efficaces. L'équipe projet s'auto-organise et peut donc adapter le travail en fonction d'elle-même.

La méthode agile divise le projet en plusieurs sprints, c'est-à-dire des cycles de travail très courts. De plus, différents types de réunions permettent de faire le point à intervalles réguliers sur les différentes avancées de la production et, en cas de dérives ou de nouvelles contraintes, une réadaptation peut être mise en place.

La méthode agile SCRUM est un bon exemple. Elle propose de mettre en place trois types d'acteurs principaux : le responsable produit, le scrum master et les membres de l'équipe. Dans notre cas, le responsable produit est notre enseignante, qui représente le client et peut ainsi proposer un recadrage du projet lorsqu'il dérive. Le scrum master est la cheffe de projet, Amandine Kohlmuller. Elle fait l'intermédiaire entre le responsable produit et l'équipe et élimine les obstacles qui pourraient empêcher l'équipe d'atteindre les objectifs qu'elle a fixé pour chaque sprint.

Enfin, le dernier acteur de la méthode SCRUM est l'équipe : Laurelenn Sangaré, Eva Benharira et Margaux Vailant. Elles sont responsables de la réalisation opérationnelle des travaux de chaque sprint et bénéficient d'une totale liberté dans l'exécution des tâches.

Avant chaque sprint, la scrum master établit une liste des prochains objectifs et des différentes tâches pour les atteindre. Lors de la phase de production, elle était épaulé par Laurelenn Sangaré, directrice technique, pour définir les tâches en question. Ces dernières se présentent donc sous la forme de cartes dans Trello, accessible par tous les membres de l'équipe. Une réunion de planification est alors organisée, au cours de laquelle les points de la liste sont évoqués un par un pour s'assurer d'une bonne compréhension. C'est également l'occasion pour que chacune émette son point de vue afin que la liste soit réadaptée si nécessaire. Cette réunion lance le début du sprint qui durera jusqu'à trois semaines.

Durant un sprint, les membres de l'équipe se retrouvent pour des stand-up meeting, de courtes réunions de 5 à 10 minutes, dans le but que chacune fasse part de l'avancée de ses tâches aux autres. Cela permet d'informer les acteurs du projet des travaux quotidiens mais également des éventuels obstacles rencontrés qui pourraient nécessiter une modification des objectifs du sprint. Par exemple un délai supplémentaire pour une tâche, une revue des objectifs à la baisse ou l'intervention de l'enseignante.

Une fois le sprint terminé une réunion est organisée de manière à ce que chacune présente les tâches qu'elle a réalisées. Dans notre cas, cette réunion permet également de faire une rétrospective de sprint. C'est le moment de faire le point sur ce qui a fonctionné et sur ce qu'il faut revoir. Ainsi, de nouveaux objectifs et de nouvelles tâches sont mises en place pour le prochain sprint.

Concernant l'application Trello, elle nous a été très utile et ces différents modules nous ont permis de faire évoluer notre méthode de travail au cours du projet. En effet, il est possible d'assigner des membres de l'équipe à une carte, d'ajouter une étiquette correspondant à un type d'élément du projet (gestion, back, css, design...) et de préciser une date limite pour effectuer la tâche. Lors de la phase de cadrage, nous utilisions ces trois fonctionnalités pour bien séquencer et segmenter les différentes missions. En phase de production, nous avons changé de manière de segmenter les tâches pour qu'elles correspondent davantage à la structure du code. Pendant cette phase, nous avons également beaucoup moins utilisé la fonctionnalité "d'assignation des tâches" afin d'être plus flexibles et plus organisées avec les autres projets du semestre.

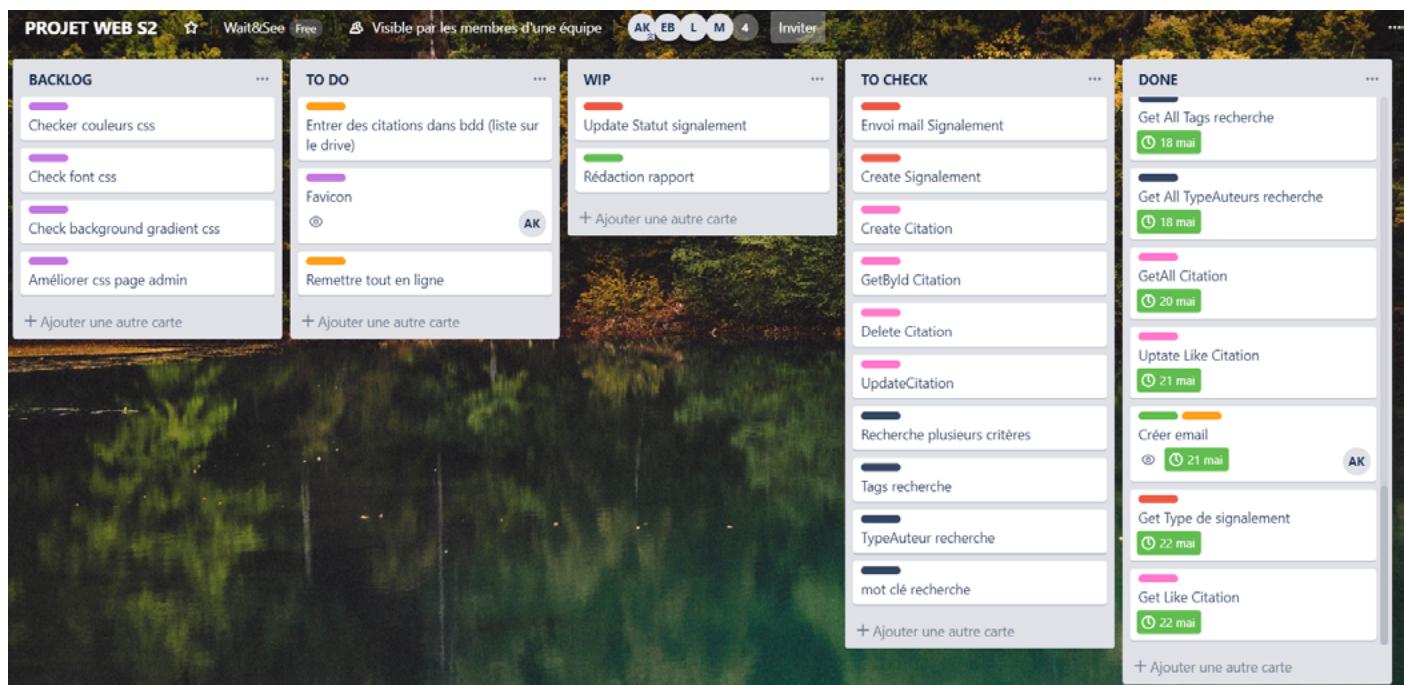


Illustration 16 : Capture d'écran du Trello, 22/02/2019

## **4.2 RÉPARTITION DES TACHES**

Après avoir toutes participé à la phase de recherche, brainstorming et réflexion autour du projet, nous avons commencé à répartir nos rôles.

Amandine Kohlmuller était chargée de la gestion de projet. C'est elle qui a mis en place la méthode agile et les outils du projet. Elle était responsable de la répartition des tâches dans le temps, du respect des délais des différents livrables, de l'organisation générale du projet et des échanges sur l'avancement du projet avec la professeure.

Concernant la production, elle s'est davantage occupée de la mise en place de l'arborescence, de la structure du html et de la documentation des différents fichiers. Elle a par exemple fait la documentation de tous les models. Elle était également responsable des tâches associées au design, de la réalisation des maquettes aux détails de proportions des éléments du css.

C'est aussi à la cheffe de projet qu'il a été confié la mission d'héberger la base de données et le site, avec le logiciel FilleZilla ainsi que de créer et gérer le mail de l'équipe : citatimacs@gmail.com

Enfin, elle s'est occupé de l'organisation et de la mise en forme du bilan du projet en produisant un rapport et en réunissant les différents documents du cadrage et de la production.

Laurelenn Sangaré, ayant le plus d'expérience dans le domaine, a été nommée directrice technique. Lors de la phase de production, elle était responsable de l'enchaînement des différentes étapes en se basant sur l'architecture logicielle et la structure du code discuté en amont avec l'équipe. Elle a partagé ses compétences et connaissances avec l'équipe en créant des exemples type de models et de controller (ModelCitation et ControllerCitation) pour permettre à l'équipe de faire de même avec les autres fichiers.

C'est également elle qui a créé les fichiers myPDO avec la base de données réalisée avec Eva Benharira. Elle a aussi pris en charge les fonctions particulières (section OTHER dans le code) et les appels api (fetch) du Javascript.

Enfin, elle a mis en place le routeur et a effectué les vérifications globales en collaboration avec les membres de l'équipe pour tester et mettre en ligne le site.

Margaux Vaillant était principalement responsable du front. Elle a travaillé sur le html et le css en se basant sur les maquettes et structure réalisées. Elle s'est également chargée de la partie affichage dynamique avec le Javascript.

Concernant le back, elle a créé des fonctions de recherche par critères dans ControllerCitations. Prenant exemple sur la fonction comportant tous les critères, elle a pu faire toutes les autres combinaisons possibles (Keyword, tags, typesAuteur ainsi que toutes les combinaisons doubles possibles). En vérité, Margaux ayant très bien compris la manipulation des requêtes SQL, le vrai défi pour elle dans cette partie était surtout d'adapter la syntaxe des fonctions pour qu'elles restent cohérentes avec le reste du code.

Eva Benharira était responsable du Git. Elle a créé le repo et a initié les membres de l'équipe à l'application GitKraken. Elle était responsable des branches et de la gestion des éventuels conflits.

Par ailleurs, elle a créé la base de données en ligne avec Laurelenn Sangare. Puis avec l'aide de celle-ci elle a pu créer les différents fichiers models. Elle a également mis en place les fonctions dans les controllers. En s'appuyant sur le modèle du fichier controller citation elle a créé les controllers TypeAuteur, TypeSignalement et Tags. Elle s'est également occupée des fonction new, update et get de controllerSignalement.

Enfin, elle a assisté Amandine Kohlmuller dans la rédaction du rapport et la relecture de celui-ci.

## 4.3 PLANNING

Afin de faire un compte-rendu global de notre projet, nous avons réalisé un diagramme de Gantt représentant les différentes tâches importantes et leur répartition dans le temps. Cela permet de bien mettre en évidence les différents sprints et l'enchaînement des étapes. Par conséquent, il apparaît clairement que la partie cadrage du projet et mise en place de la structure du site est tout aussi essentielle que la partie production et code.

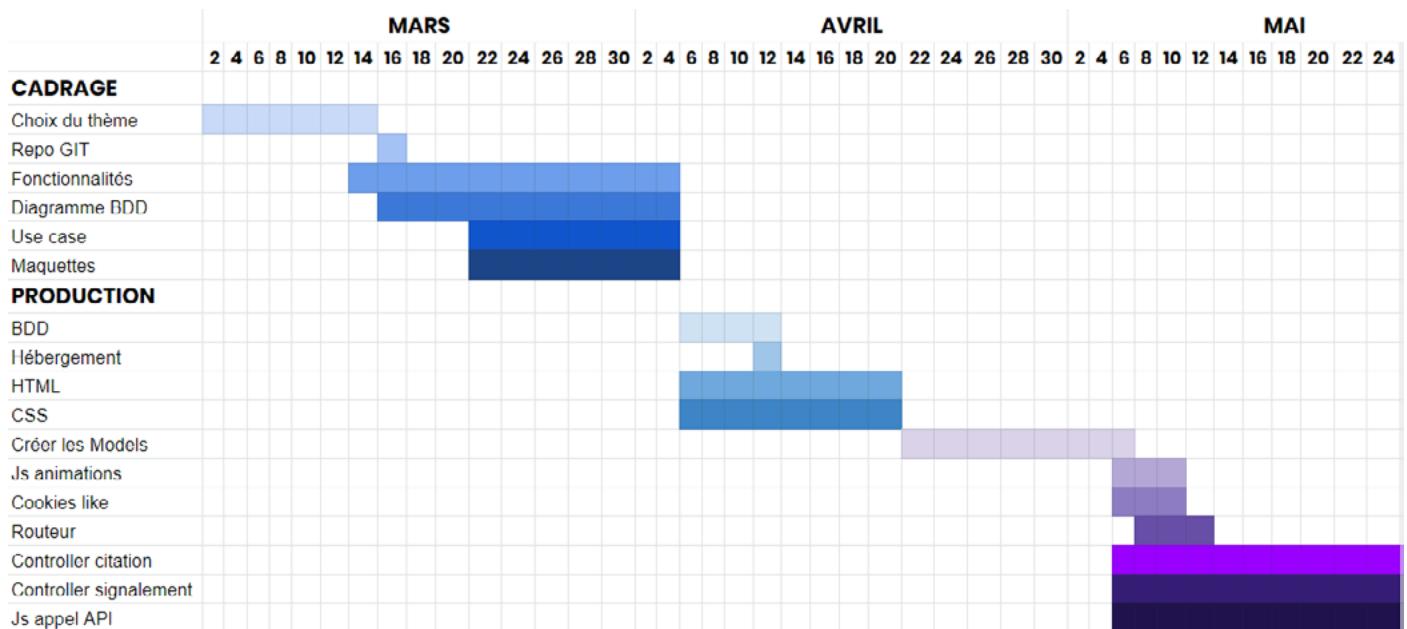


Illustration 17 : Planning de Gantt revue bilan du projet

# BILAN

## 5.1 SYNTHÈSE RÉCAPITULATIVE

FONCTIONNALITÉS SOUHAITÉES	FAIT ?
<b>AFFICHAGE DYNAMIQUE</b> Afficher les citations Recherche par mot-clés Trie pas like Trie par date Filtre par types d'auteur Filtre par tags	oui oui non non oui oui
<b>AJOUTER UNE CITATION</b> Choisir l'auteur Choisir le type d'auteur Choisir les tags	oui oui oui
<b>SIGNALER UNE CITATION</b> Choisir un type de signalement Ajouter un message Envoyer un mail	oui oui oui
<b>TRAITER UN SIGNALLEMENT</b> Mail redirection vers une page admin Modifier l'auteur d'une citation Modifier le type d'auteur Modifier le contenu d'un citation Modifier les tags d'une citation Supprimer une citation Ne rien faire Valider n'importe quel traitement d'un signalement avec un mot de passe	oui oui oui oui non oui oui oui

## CITAT'IMAC

Cette citation a été signalée avec le message suivant :

"LAURELENN TOUCHE PAS A CE SIGNALLEMENT JE VEUX TESTER LA PAGE ADMIN"

Love

2019-05-25

"miaaouu"

Sun - AUTRE

[EDITER LA CITATION](#)

[SUPPRIMER LA CITATION](#)

[NE RIEN FAIRE](#)

## **5.2 PISTES D'AMÉLIORATIONS**

Notre objectif principal, pour cette première version, était avant tout de faire un site fonctionnel et d'en développer les fonctionnalités premières. Nous avons également eu la possibilité d'améliorer l'expérience utilisateur en développant l'interface graphique du site, afin qu'elle soit plus intuitive et agréable.

Ainsi, concernant les améliorations que nous aimerais faire, elles reposent surtout sur l'amélioration de l'expérience utilisateur et l'affichage du site. Cela implique par exemple de traiter le cas où nous aurons beaucoup de citations ou tags à charger, mais aussi de nouveaux critères de recherche et tri (nom de l'auteur, likes, date), l'affichage d'informations supplémentaires, ou encore une meilleure gestion des signalements.

Voici donc la liste des fonctionnalités que nous développerons dans la prochaine version :

### **AFFICHAGE UTILISATEUR :**

- Ne charger qu'une partie des citations avec un possibilité d'en afficher davantage
- Rendre le site responsive

### **FONCTIONNALITÉS UTILISATEUR :**

- Trier l'apparition des citations par nombre de "like" ou par date
- Valider la recherche par mot-clés en pressant le bouton entrée
- Effectuer la recherche par nom d'auteurs
- Vérifier si la citation existe déjà

### **AFFICHAGE ADMIN :**

- Signifier quand un signalement a été traité

### **FONCTIONNALITÉ SIGNALEMENT :**

- Pouvoir aussi modifier les tags
- Faire un mail avec plus d'informations voire un feedback à l'utilisateur quand la citation a été modifiée

## **5.3 RETOUR D'EXPÉRIENCE**

### **MARGAUX VAILLANT :**

Participer à ce projet m'a permis d'avoir des bases plus solides en javascript et PHP que celles déjà acquises en licence. En effet, mes bases de PHP s'arrêtaient à des notions syntaxiques, puisque nous ne le liions pas à une base de données. J'avais appris de mon côté le javascript, jQuery et langage SQL, cependant, apprendre en autodidacte signifie se focaliser sur les aspects du langage qui nous intéresse et non sa globalité. Avec les contraintes imposées par le sujet, et celles que chacune, codant d'une façon différente, met en place, il a fallu être créative et réfléchie face à plusieurs difficultés.

Étant très à l'aise avec le CSS, j'ai également pu "coder" les mockup qu'Amandine avait fait. Encore une fois, les objectifs étant déjà fixés, se coller à un template et s'y adapter, plutôt que faire le CSS "dans le noir" a été enrichissant.

Il a aussi fallu penser le code HTML en sachant que plus tard il allait être écrit dynamiquement par le PHP. Le CSS a donc dû parfois s'adapter à des contraintes mises en place par le HTML.

Je me suis également occupée des requêtes SQL mélangeant les choix (par exemple citation avec un tag "sel" et un auteur de la promo "imac2021"). Les TD m'avaient déjà beaucoup enrichie et plu, car faire des requêtes complexes c'est relever un défi, et je trouve cela motivant. Utiliser donc des requêtes complexes dans ce projet m'a permis d'y voir un plus grand but et une utilisation plus large que lors d'un simple TD.

Coder le javascript sans le jQuery a également été un défi pour moi, qui n'avais appris à lier le PHP et le javascript qu'avec celui-ci.

Travailler sur ce projet m'a donc permis de repenser mes connaissances et a été très instructif.

### **EVA BENHARIRA :**

L'IMAC représente pour moi une première expérience de la programmation. Ce projet m'a donc été très bénéfique dans la mesure où il me permet de mettre les connaissances acquises à l'épreuve tout en étant épaulée par mes camarades.

En effet, j'ai appris grâce aux cours d'architecture logicielle ainsi qu'aux cours de web de ce second semestre ce qu'était une base de données. Travailler avec Laurenn m'a permis de lui faire des propositions qu'elle pouvait confirmer ou infirmer me montrant les bons raisonnements à mener ainsi que mes sources d'erreur. J'ai ainsi pu commencer à acquérir certains réflexes.

J'ai également pu utiliser une seconde fois le logiciel PHPMyAdmin et pousser plus loin l'usage que j'en avais eu en cours lors des TDs.

La réflexion globale sur le projet m'a aussi beaucoup apporté en terme de travail de groupe. En effet, cadrer le projet et se répartir les tâches est quelque chose de très important d'autant plus que je m'intéresse à la gestion de projet. J'ai donc beaucoup appris ici, notamment grâce à l'emploi de Trello qui, s'il n'a pas été utilisé tout au long de notre processus de travail, s'est révélé très intéressant. Je l'ai ainsi déjà réutilisé pour la gestion d'autres projets.

En terme de code, il ne s'agit pas de quelque chose de naturel pour moi. C'est pourquoi l'aide de mes camarades s'est révélée importante pour me permettre d'avancer et je suis ainsi contente d'avoir pu réaliser plusieurs fichiers et d'en comprendre le fonctionnement. Le système Model-View-Controller m'a bien aidée à cerner l'usage qui est fait de chaque fichier. De plus, mener ce projet m'a permis de me familiariser avec la programmation objet qui me semblait vraiment difficile au premier abord.

De plus étant la plus expérimentée dans l'usage de GitHub et de GitKraken j'ai pu aider mes camarades en leur montrant les bases de l'utilisation et les rassurant en cas de gestion de conflit. J'ai de plus pris sur moi d'initier la création de branche lorsque notre professeure nous en a fait remarquer la nécessité. Je suis dorénavant capable d'en créer et de gérer leur raccordement au master sans problème ce qui j'en suis sûre me sera très utile dans de futurs projets pour éviter de me trouver confrontée à de trop gros conflits. Je pense que ce projet a ainsi pu permettre à mes camarades de découvrir GitKraken et l'aspect pratique du fonctionnement de git tout en nous initiant toutes à la gestion de branches. De ce fait, je pense que nous serons à même de mener nos futurs projets de manière plus cadrée en répartissant les différentes parties de code branche par branche plutôt que de travailler directement sur le master.

Enfin, je suis très heureuse du produit final de notre collaboration. Je me suis sentie valorisée et accompagnée tout au long du projet par mes camarades et notre professeure et j'ai pu développer de nombreuses compétences et découvrir de nouveaux logiciels et méthodes de travail.

Citat'IMAC est un site fonctionnel et ludique qui, je l'espère, rencontrera son public !

### **AMANDINE KOHLMULLER :**

Cette expérience m'a été très bénéfique et enrichissante sur plusieurs points. Dans un premier temps, j'avais très peu d'expérience en back-end et travailler sur un tel projet m'a permis d'acquérir des compétences et connaissances techniques essentielles au métier de cheffe de projet. En effet, ma mission principale sur ce projet correspondait parfaitement à mon objectif professionnel. Mon but est de pouvoir interagir avec tous les corps de métiers liés au numérique et c'est ce que j'ai dû faire au cours de ces derniers mois.

J'ai déjà eu la possibilité de manager une équipe, avec la méthode agile SCRUM, pour un projet tuteuré lors de ma précédente formation. Bien que je me sois appuyée sur ce vécu pour appréhender ce projet, je me suis adaptée aux membres de l'équipe, aux enjeux et aux objectifs qui étaient bien différents. Par exemple, en DUT, mon équipe et moi-même avions un niveau de compétence égale, il m'était donc plus aisément de comprendre un problème concernant un certain aspect du projet et de trouver une solution. En revanche, pour le site Citat'IMAC, nous n'avions pas toutes la même expérience et le même niveau de compétence. Au début du projet, je craignais d'être freinée par ces lacunes. Finalement, nos différences furent un véritable atout et j'ai beaucoup appris auprès de mes collègues.

Par ailleurs, et de manière plus générale, j'ai été très heureuse de pouvoir partager et mettre en pratique mes connaissances en terme de gestion de projet. Très peu d'enseignements nous encouragent à faire cela dans la formation et je pense que c'est pourtant un aspect essentiel à aborder pour tendre vers une professionnalisation.

Finalement, ce projet résume assez bien la raison de ma venue dans la seule formation publique d'ingénieur alliant sciences et arts. J'ai eu la possibilité de toucher à des domaines de compétences variés (back-end, front-end, web design...), d'échanger et de travailler avec des personnes davantage spécialisées dans ces domaines et d'organiser et planifier un projet de la naissance de l'idée à la livraison.

## **LAURELENN SANGARÉ :**

Durant mes deux années d'études précédentes, j'ai eu l'occasion de faire du PHP et du web plus généralement. Cependant, mon premier projet a été quelques peu chaotique et je n'avais au final que très peu appris sur le PHP. J'ai eu la chance par la suite de faire un stage de 3 mois en web, utilisant un framework (Symfony) et jQuery, ce qui m'a permis de fortifier mes bases, et de me sentir plus à l'aise avec le PHP.

Au moment de commencer le projet de PHP de cette année, en IMAC, j'étais un peu plus en confiance que l'année passée, et je savais déjà plus ou moins quelle serait la logique et le fonctionnement global du site.

Cependant, les méthodes recommandées dans ce projet n'étaient pas les mêmes que celles que j'avais apprises, ce qui m'a donné l'occasion de découvrir une nouvelle facette du PHP, une nouvelle logique grâce à de nouveaux outils.

Ce projet m'a donc permis, évidemment, de gagner en expérience, mais surtout en flexibilité quant aux nouvelles ressources que j'ai pu manipuler par rapport aux années passées (fetch, javascript sans jquery, input dynamique dans la page sans framework etc...). Et en ce qui concerne les compétences déjà acquises, j'ai pu fortifier celles-ci (requêtes SQL, manipulation du JSON, lien View-Controller etc...) j'ai pu bien évidemment les améliorer par la pratique et les différents problèmes rencontrés.

Pour ce qui est de mon retour par rapport au travail du groupe de manière générale, il s'est avéré que j'ai été "reconnue" comme étant la personne la plus expérimentée en PHP et web en général au sein du groupe. J'ai donc dû développer un esprit pédagogique pour, dans un premier temps, discuter du fonctionnement global et de l'architecture du site selon mon expérience et mes quelques connaissances; et dans un second temps, pour expliquer le fonctionnement du langages aux camarades moins expérimentées.

A première vue, mon travail peut paraître plus conséquent que celui de certains membres de l'équipe. Cependant, nous nous sommes assurées que chacune d'entre nous ait l'opportunité, dans un premier cas, de manipuler le php, et dans un second cas, d'apprendre réellement quelque chose grâce à ce projet.

En effet, nous n'avons pas eu des parcours égaux et nos compétences sont bien différentes les unes des autres, mais chacune a été utile au sein du groupe, a appris et a développé des fonctionnalités plus ou moins complexes en fonction de leurs connaissances et leur niveau.

Mais ayant une vision globale sur l'ensemble du projet, je peux affirmer que ce que chacune a fait a été très bien fait, réussi, et compris, sans qu'il n'y ait forcément de correction derrière.

Au-delà de ça, les compétences de mes camarades m'ont permis de me décharger de certaines tâches qui m'incombaient dans ma formation précédente (css, rapport, gestion de projet), pour pouvoir rester concentrée sur les tâches où j'étais le plus utile. Bien sûr, chacune a participé aux différents aspects du projet, mais de manière plus ou moins importante suivant nos compétences et préférences.

Pour ainsi conclure, je suis très fière du projet que nous avons réalisé. D'une part pour le sujet que nous avons choisi, c'est important d'aimer ce pour quoi nous faisons ce projet. Mais aussi pour le développement technique de celui-ci, pour son aspect général. Et surtout parce que je suis très heureuse de savoir que j'ai pu aider mes amies à améliorer certaines compétences, de savoir qu'elles ont tiré un enseignement de ce projet, et que chacune ait su trouver sa place, se sentir utile, et se sentir grande.

# CONCLUSION

Pour conclure sur l'ensemble du projet, nous voulions en premier lieu un sujet qui nous plaise et qui soit ludique, dans lequel l'utilisateur est un véritable acteur du site.

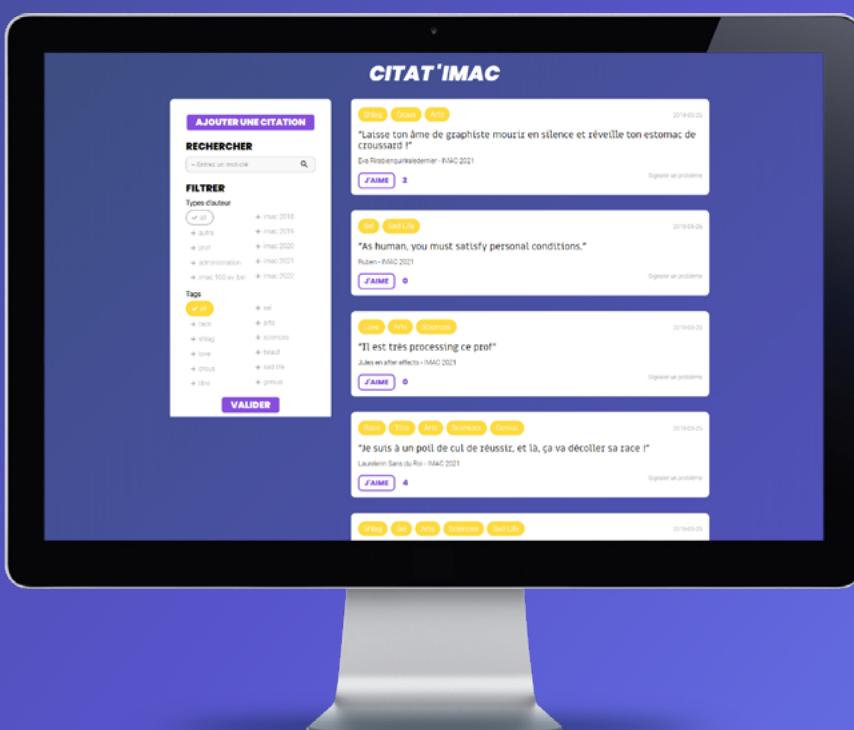
L'IMAC est si riche de personnalités différentes que nous voulions créer un site autour des plus belles paroles entendues au sein de la formation.

Malgré la quantité de travail qu'a nécessité la conception du site, nos compétences diverses nous ont permis de n'être prises de court à aucun moment. Bien sûr le délai était une contrainte importante et nous avons fini juste dans les temps, mais nous ne doutions pas de l'aboutissement de la première version de notre site.

La communication au sein du groupe, l'organisation, les séances de travail et les brainstorming nous ont permis d'avancer dans notre projet sainement et en surmontant les difficultés.

Ainsi, ce projet nous a apporté des compétences techniques, nous a appris à interagir au sein d'une équipe et nous a permis de concevoir un projet de bout en bout.

Nous sommes très fiers d'avoir réalisé ce site web, et nous espérons l'améliorer dans des versions ultérieures.



*Illustration 18 : Capture d'écran du site Citat'IMAC*