

# Dossier Coddity

## Une introduction et la présentation de votre équipe

Notre équipe est composée de deux étudiants : Clément Gambier et Thomas Mauran.

Nous sommes tous les deux étudiants en première année de DUT Informatique à l'IUT de Montpellier.

## L'analyse de la problématique et notre solution pour y répondre

A la lecture du sujet nous avons tout de suite pensé à réaliser un site web qui permettrait à tout le monde de déposer des informations sur lesquelles tout le monde pourrait débattre. Afin de combattre les fake news nous laissons chaque utilisateurs voter pour ou contre sur chaque information.

Pour connaître la fiabilité d'un article on peut également regarder le score de fiabilité de l'auteur qui est calculé selon le nombre de like et de dislike qu'il a reçu.

## La présentation et justification de nos choix techniques (archi, technos, design, conventions, etc.)

### Les technologies employées :

Notre site web a été réalisé en HTML, CSS, PHP, SQL.

Nous avons abordé au premier semestre la conception de site web avec l'HTML et le CSS donc nous avons quelques bases de plus nous avons participé ensemble à la Nuit de l'Info qui nous a également beaucoup appris.

Ensuite pour le back end nous ne connaissions aucun langage donc nous avons choisi le PHP qui nous semblait le plus simple à assembler avec les autres technologies que nous avons choisi. Nous n'avons aucune connaissance dans ce langage donc nous avons suivi le cours disponible sur Openclassroom et des tutoriaux plus spécifiques sur youtube.

A l'aide de nos cours de base de données nous savions écrire des requêtes SQL, il nous a fallu apprendre 2 syntaxe car sur certains tutoriels ils utilisaient mysqli et d'autres un PDO. Finalement, nous avons adapté nos fichiers et la majorité fonctionne avec un PDO car nous avons trouvé cela plus simple pour gérer plusieurs requêtes en même temps.

Ensuite pour la gestion des bases de données nous avons utilisé PHPmyadmin, nous n'avons pas eu de grandes difficultés car c'est un outil que nous trouvons très facile à prendre en main.

### Le serveur

Héberger un site web était aussi une expérience nouvelle pour nous.

Nous avons décidé de nous tourner vers les services de Pulseheberg qui proposent des serveurs avec des prix très abordables, une capacité de stockage largement suffisante pour notre projet et enfin une interface web très pratique pour des novices comme nous.

Une fois le serveur acquis nous avons acheté le nom de domaine wediscus.fr et grâce à l'interface plesk proposée par pulseheberg il nous a été très simple de relier notre serveur à cette url. Toujours grâce à la même interface il nous a été possible de générer un certificat SSH gratuitement pour rendre notre site https.

## Notre organisation

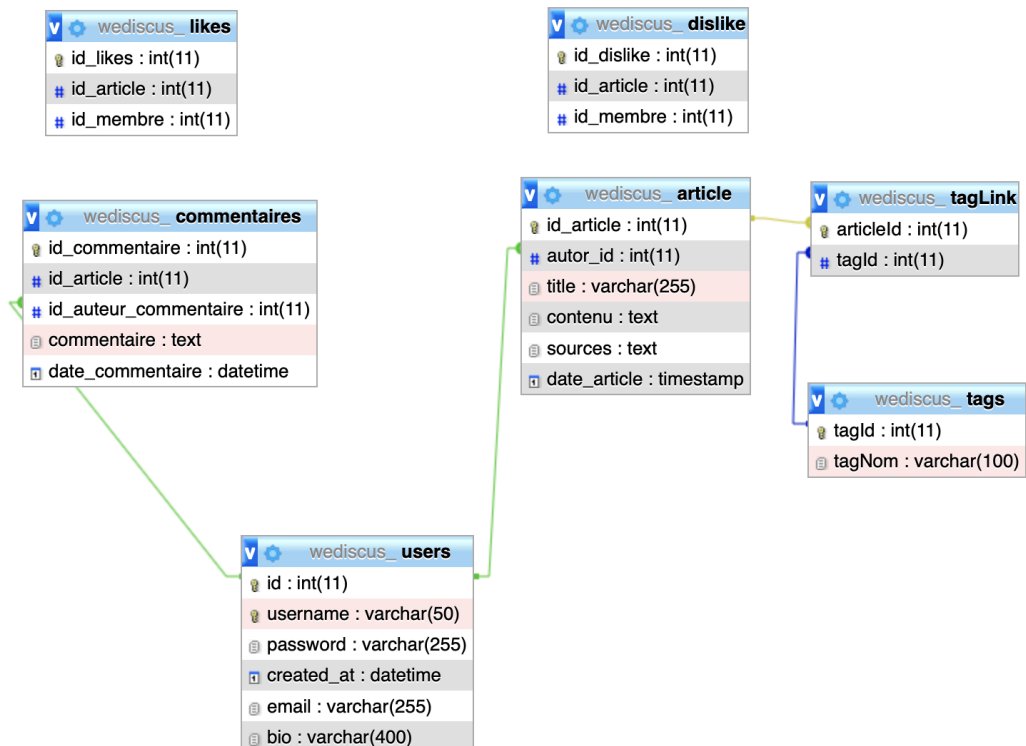
Pour ce qui est de l'organisation nous avons au début commencé avec github et l'IDE Atom mais n'étant absolument pas habitué à utiliser github nous avons vite commencé à directement modifier les fichiers sur le serveur, ce qui n'est pas une très bonne habitude mais nous permettait de toujours travailler sur une version à jour. Un collègue de travail nous a montré visual studio code et une extension ftp qui permet de modifier directement les fichiers sur le serveur en sauvegardant ce qui nous a permis d'être bien plus productif. Cependant si nous avions voulu être plus responsable il nous aurait fallu mettre en place un système de versionnage avec github.

## Le style du site

Pour ce qui est du style graphique du site étant donné notre faible niveau en design nous avons décidé de partir sur un style épuré avec des logos simples à comprendre, des animations minimalistes et comme couleur principale le blanc et le lavande.

## Les fonctionnalités de notre site web

Avant d'aborder les fonctionnalités, voici une représentation de notre base de données afin de mieux appréhender nos démarches ( les flèches représentant les clés étrangères présentes entre les tables).



## Fonctionnalités importantes :

### 1. Déposer des articles

Afin de pouvoir débattre d'un sujet, il faut tout d'abord créer une base à débattre. C'est pour cela que tous les utilisateurs de wediscuss peuvent ajouter un article sur lequel tous les utilisateurs pourront venir débattre. Pour cela on utilise une page html qui sert de formulaire qui lorsqu'elle est complétée lance une requête INSERT dans la table article.

### 2. Afficher des articles

Pour l'affichage, c'est très simple : une première page affiche tout les articles avec uniquement le titre, l'auteur et le nombre de commentaires (afin de connaître les articles où les débats sont les plus acharnés ! ). Ensuite, si on veut voir le contenu de l'article il suffit de cliquer sur la loupe pour accéder à la page la plus avancée du site. Cette page affiche le titre, l'auteur et son score de fiabilité, le contenu de l'article et ses sources.

Afin d'afficher ces informations une simple requête à la base de données suffit. Ensuite afin de donner une valeur numérique à l'avis des gens sur le sujet nous avons implémenté un système de like et dislike.

### 3. Commentaire et sources

La suite de la page afficher les articles se compose de deux champs de texte afin de rajouter une source ou encore un commentaire. Finalement on affiche l'espace commentaire qui est un endroit permettant à tout à chacun de débattre sur le sujet.

### 4. Le score de fiabilité

Pour savoir si un article est fiable, nous avons mis en place un score de fiabilité. Chaque utilisateur possède un score en pourcentage qui indique s'il est plus ou moins fiable. Ce score est calculé en fonction du nombre de like et de dislike que l'utilisateur a reçu sur ses articles.

### 5. Barre de recherche

Il nous semblait obligatoire de permettre aux utilisateurs de trouver un article. C'est pour cela que notre barre recherche a été faite. Grâce à elle vous pouvez chercher un utilisateur, cela vous montrera les articles qu'il a écrit. Mais encore plus utile un nom d'article. Une chose qui n'est pas encore aboutie c'est que si vous tapez un mot le système vous propose les articles où ce mot est présent, pour le futur nous voudrions améliorer la précision de ce système.

D'ailleurs vous ne serez pas déçu si vous tapez n'importe quoi dans la barre de recherche (ne fonctionne pas sur safari ...).

### 6. Profil

Nous avons mis en place une page de profil qui permet à tous les utilisateurs de modifier leurs biographies ainsi que leurs pseudonymes. Ils peuvent aussi consulter

leurs date d'inscription, leurs nombre d'articles et leur score de fiabilité sur cette même page. En cliquant sur la photo de profil il est possible d'éditer les informations de son compte

#### 7. Nous avons décidé de forcer la connexion

En effet, nous avons forcé la connexion afin de faciliter l'utilisation du site. Si vous n'étiez pas connecté vous ne pourrez pas accéder à toutes les fonctionnalités du site (car il vous faut un username afin de pouvoir écrire un article ou un commentaire). Il aurait donc fallu rediriger l'utilisateur sur la page connexion et bloquer l'envoi des articles et des commentaires.

#### 8. Les tags

Pour organiser chaque article, nous avons mis en place un système de tag qui permet de les différencier en fonction de leurs catégories. Voici la liste des tags présent sur le site :

- Informatique
- Mathématique
- Science
- Culture
- Droit
- Actualité
- Nouvelle technologie
- Autre

## Conclusion

### Les difficultés :

L'apprentissage du PHP, la syntaxe est vraiment difficile à prendre en main.

L'implémentation du CSS dans les fichiers PHP. Certaines fois cela fonctionnait, d'autres fois non, il se pourrait qu'il y ait un peu de CSS inline ... pardon ...

### Ce que l'on a appris :

Tout le long de ce projet on a énormément appris, alors tout dire prendrait du temps mais voici les choses les plus marquantes :

- Le PHP
- La gestion de base de données.
- Prise en main d'outils qui permettent de faciliter le développement.
- Faire des briefing pour voir l'avancée du projet.

### Autocritique :

Nous sommes très fiers de notre site web. Comme précisé plus tôt, nous sommes en première année à l'IUT de Montpellier. Ce projet nous a permis de découvrir comment mettre en pratique ces requêtes sql et les adapter à une page web. Il manque quelques

fonctionnalités mineures que nous n'avons pas eu le temps d'implémenter en raison d'un manque de temps. Comme le fait de pouvoir consulter le profil des autres utilisateurs, faire une recherche par tag ou modifier sa photo de profil. Nous nous sommes concentré sur les fonctionnalités majeures car chaque fonctionnalité nous a pris beaucoup de temps en raison de notre faible expérience.

#### Les erreurs :

Commencer par le front a été une erreur majeure qui nous a fait perdre beaucoup de temps, nous aurions dû commencer par concevoir toutes les fonctionnalités puis appliquer un style graphique à notre site.

Faire un diagramme d'attributs pour la base de données aurait aussi permis de gagner du temps. Car nous avons dû recréer plusieurs fois des tables.

Le fait de ne pas utiliser github nous a valu plusieurs fois des problèmes de version et donc une perte de temps considérable.

#### Nos réussites

Nous avons rempli la plupart des fonctionnalités désirées au départ qui étaient : pouvoir créer des articles, un système de like / dislike, la barre de recherche et le score de fiabilité.

#### Les perspectives de notre application.

A l'avenir nous aimerions implémenter les fonctionnalités suivantes :

- Séparer toutes les informations en différentes pages.

- Accéder au profil d'une personne

- Pouvoir modifier sa photo de profil

Merci de nous avoir permis de participer à ce projet, ce fut très enrichissant et on a pris beaucoup de plaisir à le réaliser.

Thomas Mauraan et Clément Gambier