

Projet Technologies des Moteurs de Recherche :

DIVE  :

Moteur de Recherche des Sites Web et des Images



Groupe :

Fatima Ezzahra Assanfe

Achraf Elmaouaj

Amine Salim

Issam Zaidouhadou

Encadrants :

Mr. SENNOUNI Amine

Mr. SBIHI Boubker

Table des matières:

Introduction :	3
1. Conception :	4
1.1. Spécifications :	4
1.1.1. Spécifications fonctionnelles :	4
1.1.2. Spécifications liées à l'interface :	4
1.1.3. Spécifications liées à l'utilisateur :	5
1.2. Diagrammes UML :	5
1.2.1. Diagramme de classes :	5
1.2.2. Diagramme d'activité :	5
1.3. Architecture globale :	6
1.4. Architecture Applicative :	6
1.4.1. PHP :	6
1.4.2. HTML5 :	6
1.4.3. JavaScript :	7
1.4.4. MySQL :	7
1.4.5. WampServer :	7
2. Réalisation :	8
2.1. Interfaces :	8
2.1.1. Page d'accueil :	8
2.1.2. Page de recherche des sites :	9
2.1.3. Page de recherche des images :	10
2.1.4. Vue des images :	10
2.1.5. Pagination :	11
2.2. Scripts :	11
2.3. Base de données :	12
2.3.1. La table images :	12
2.3.2. La table sites :	12
2.3.3. Code SQL pour créer la base de données :	12
Conclusion :	15

Table des figures:

FIGURE 1 : DIAGRAMME DE CLASSES.....	5
FIGURE 2 : DIAGRAMME D'ACTIVITE.....	6
FIGURE 3 : PAGE DE RECHERCHE DES IMAGES.....	10
FIGURE 4 : LA VUE DES IMAGES	10
FIGURE 5 : PAGINATION.....	11
FIGURE 6 : LA BASE DE DONNEES DIVE	12
FIGURE 7 : LA TABLE IMAGES	12
FIGURE 8 : LA TABLE SITES.....	12

Introduction :

Un moteur de recherche est, comme son nom l'indique, un outil qui permet de rechercher sur le Web ainsi que sur un ordinateur personnel, des ressources, des contenus, des documents etc., à partir de différents attributs. Notre moteur de recherche DIVE permet la recherche des sites web à partir de leurs URL, titres, descriptions, mots clés et les classe en fonction de nombre des cliques.

1. Conception :

1.1. Spécifications :

1.1.1. Spécifications fonctionnelles :

- Le moteur de recherche doit permettre la recherche des sites web.
- Le moteur de recherche doit permettre la recherche des images.
- Le moteur de recherche doit être facile à maintenir.
- Le moteur de recherche doit permettre la recherche par un ou plusieurs termes.
- Le moteur de recherche doit avoir son propre crawler.
- Le crawler doit stocker toutes la pages web trouvées dans une base de données.
- Le moteur de recherche doit avoir un logo et un favicon.
- Le moteur doit indexer les pages web en se basant sur leurs titres, descriptions et mots clés.
- Les requêtes saisies par l'utilisateur seront traitées par les scripts *sitesearchprovider* et *imageeachprovider*.
- Le moteur doit indexer les images en se basant sur leurs titres, descriptions, mots clés et alts.

1.1.2. Spécifications liées à l'interface :

- L'interface du moteur de recherche doit être très simple.
- L'interface du moteur de recherche doit donner envie à l'utilisateur de rechercher.
- L'interface du moteur de recherche doit permettre la navigation entre les images.
- L'interface du moteur de recherche doit donner le titre de l'image, le lien vers l'image et le lien vers la source de l'image.
- L'interface du moteur de recherche doit permettre la visualisation du titre, lien et la description du site web trouvé.

- L'interface du moteur de recherche doit permettre la navigation entre les pages des résultats de la recherche.

1.1.3. Spécifications liées à l'utilisateur :

- L'utilisateur doit pouvoir rechercher par un ou plusieurs termes.
- L'utilisateur peut visualiser l'ensemble des sites web et images trouvés.
- L'utilisateur peut réeffectuer une autre recherche à partir de la page des résultats.
- L'utilisateur peut consulter les images dans la page des résultats.

1.2. Diagrammes UML :

1.2.1. Diagramme de classes :

Les diagrammes de classes expriment la structure statique d'un système basé sur des classes et les relations entre ces classes. Il ignore les aspects dynamiques et temporels et montre les composants qui collaborent au fonctionnement de notre moteur de recherche.

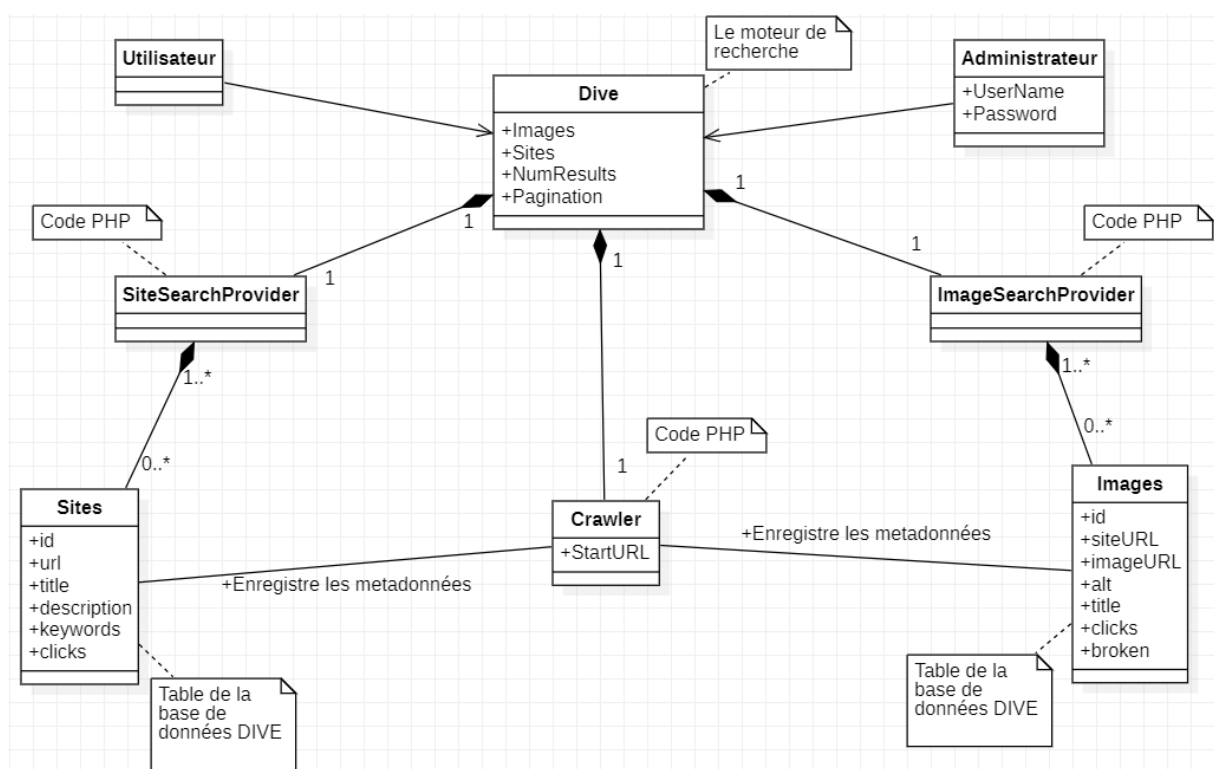


Figure 1 : Diagramme de classes

1.2.2. Diagramme d'activité :

Le diagramme d'activité ci-dessous permet de modéliser un processus interactif, global et partiel pour notre moteur de recherche.

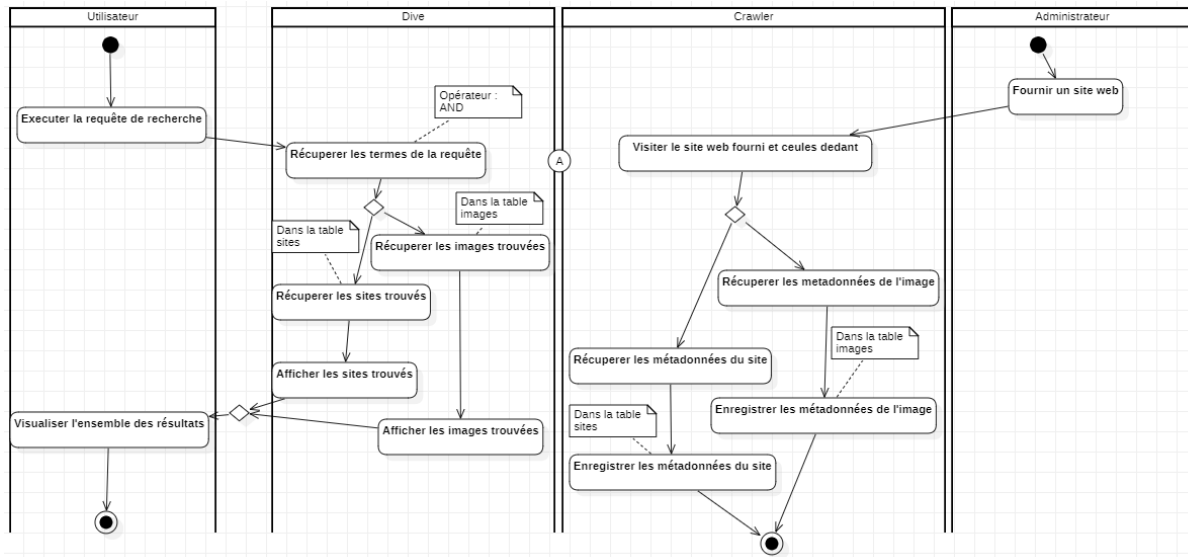


Figure 2 : Diagramme d'activité

1.3. Architecture globale :

Pour réussir dans la phase de conception, il est nécessaire de suivre un contexte conceptuel qui répond aux particularités et aux besoins de base du moteur de recherche, satisfaisant ainsi les utilisateurs. L'architecture client / serveur est l'architecture la plus adaptée dans notre cas. Cette architecture est basée sur l'utilisation de deux parties, à savoir la partie serveur et le partie client qui s'exécutent généralement sur deux ordinateurs différents. La partie contenant la base de données est la partie serveur. Le serveur accepte la demande, la traite et renvoie le résultat au demandeur. La partie client est le produit où le client envoie une requête et reçoit des résultats.

1.4. Architecture Applicative :

1.4.1. PHP :

PHP est un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.

1.4.2. HTML5 :

L'HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom.

1.4.3. JavaScript :

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. À chaque fois qu'une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, etc...

1.4.4. MySQL :

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

1.4.5. WampServer :

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

2. Réalisation :

2.1. Interfaces :

2.1.1. Page d'accueil :

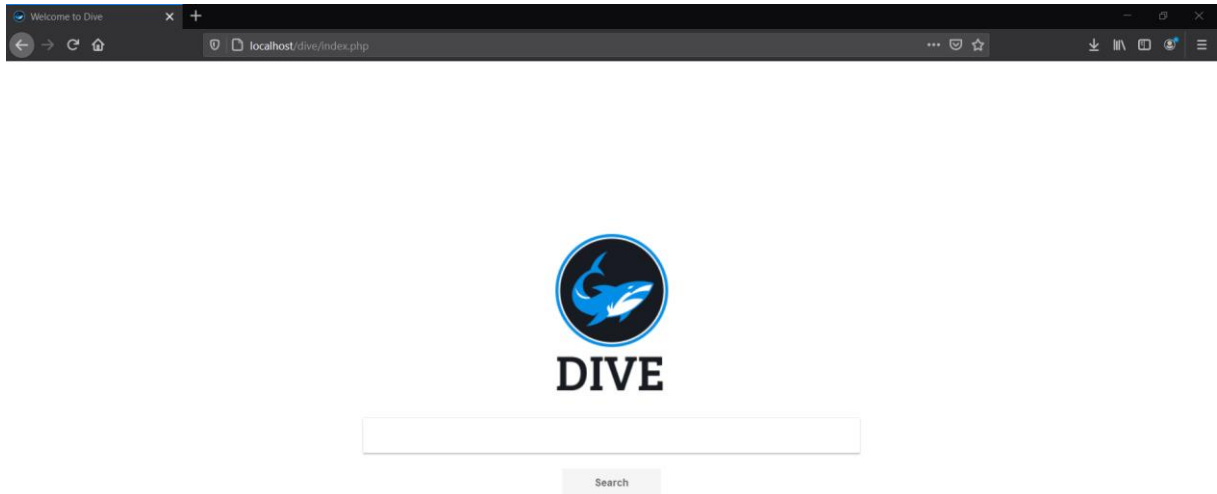


Figure 1 : La page d'accueil

La page d'accueil du moteur de recherche DIVE, c'est une interface simple avec des couleurs claires qui contient le logo et une barre pour saisir la requête de recherche par l'utilisateur ; le logo est un logo significatif d'un requin dans la mer noire, la mer noire signifie la grande quantité de l'information qui pourrait être dans la base de données du moteur de recherche, et le requin signifie l'information précise qu'on veut trouver. La signification de l'ensemble du logo signifie que le moteur de recherche DIVE est un moteur de recherche pertinent, c'est à dire à partir d'une énorme base de données, vous êtes capable de trouver l'information que vous voulez.

2.1.2. Page de recherche des sites :

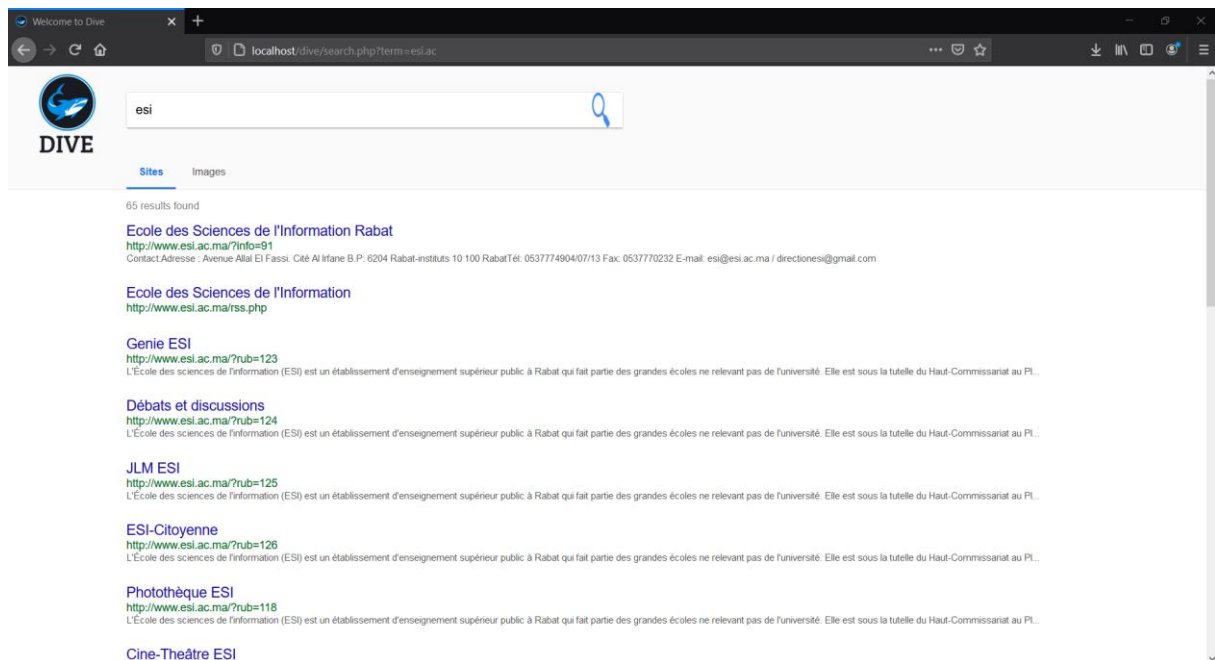
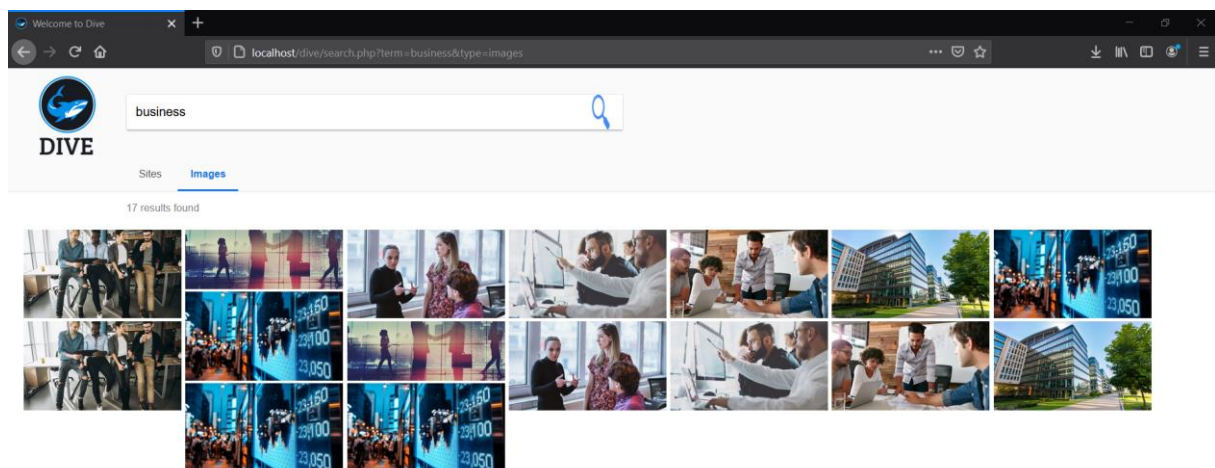


Figure 2 : Page de recherche des sites

En effectuant une recherche dans le moteur de recherche DIVE, DIVE nous donne le nombre des résultats trouvés avec la requête de recherche saisie, les résultats sont soit sous forme de sites ou sous forme d'images. Par exemple dans notre cas, la requête 'ESI', DIVE nous a donné 65 résultats sous forme de sites. (Nous avons indexé le site de l'ESI en utilisant le crawler), le logo du moteur de recherche reste toujours près de la barre de recherche du moteur pour que le logo s'aggrave dans la mémoire de l'utilisateur ainsi que pour le côté esthétique du moteur de recherche.

2.1.3. Page de recherche des images :



D I VE

Figure 3 : Page de recherche des images

Dans ce cas, la requête business nous a donné 17 résultats dans Images.

Remarque : vous pouvez voir la pagination dans le bas de la page.

2.1.4. Vue des images :

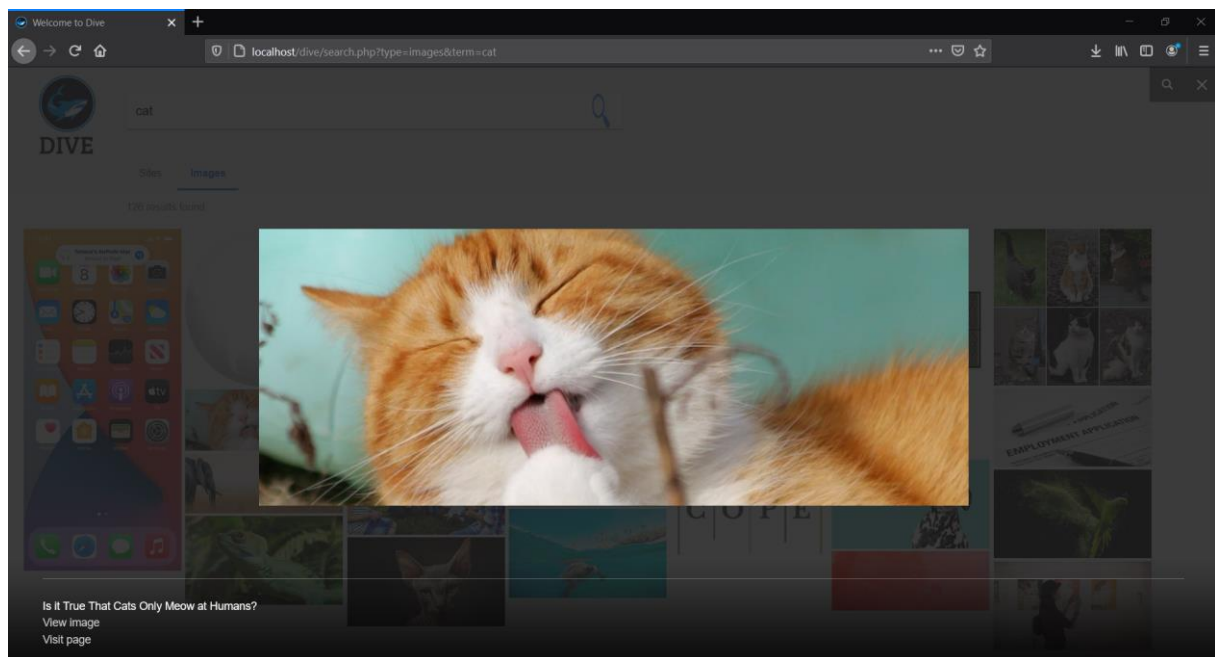


Figure 4 : La vue des images

En cliquant sur une image dans les résultats, la vue de l'image se positionne en avant tandis que les autres images du résultat restent en arrière-plan.

Nous avons ainsi la possibilité de voir l'image dans le site où il apparaîtrait.

2.1.5. Pagination :



Figure 5 : Pagination

Voici la forme de pagination dans le moteur de recherche DIVE.

Le 'I' en bleu indique la page actuelle sélectionnée. Par exemple dans cette image on est dans la page 5.

2.2. Scripts :

Index : est le fichier contenant la page d'accueil de notre moteur de recherche.

ImageSearch : permet la recherche des images.

Sitesearch : permet la recherche des sites web.

Search : est le fichier contenant le script permettant de la recherche, ce script est basé sur une méthode GET permettant d'extraire les résultats en se basant sur la requête entrée par l'utilisateur.

Style : fichier contenant la feuille de style CSS. Nous avons utilisé un style FlexBox afin que le logo et la barre de recherche puissent s'adapter avec le changement de la taille de la fenêtre. Nous avons choisi les mesures en se basant sur google. Cette feuille de style prend en charge l'ensemble des pages de notre moteur de recherche.

Crawl : on donne au Crawler un lien, ensuite le Crawler visitera ce lien, et il récupérera tous les liens disponibles dans le lien donné, par la suite il insérera ces liens dans une table (*sites* ou *images*) dans la base de données *dive*, ensuite pour chacun de ses liens il va suivre la même procédure. Le web crawler va commencer par le site suivant. Nous modifions à chaque le lien de départ afin d'enrichir la base de données.

2.3. Base de données :

Dive et la base de données où sont stocker les pages web et les images indexées, et elle contient deux table *sites* et *images* avec une somme de 3628 éléments.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> images	Parcourir Structure Rechercher Insérer Vider Supprimer	3 015	MyISAM	latin1_swedish_ci	637,1 kio	-
<input type="checkbox"/> sites	Parcourir Structure Rechercher Insérer Vider Supprimer	613	MyISAM	latin1_swedish_ci	230,9 kio	-
2 tables	Somme	3 628	MyISAM	latin1_swedish_ci	868,0 kio	0 o

Figure 6 : La base de données dive

2.3.1. La table images :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/> 1	id	int(11)			Non	Aucun(e)		AUTO_INCREMENT
<input type="checkbox"/> 2	siteUrl	varchar(512)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 3	imageUrl	varchar(512)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 4	alt	varchar(512)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 5	title	varchar(512)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 6	clicks	int(11)			Oui	NULL		
<input type="checkbox"/> 7	broken	tinyint(4)			Oui	NULL		

Figure 7 : La table images

2.3.2. La table sites :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/> 1	id	int(11)			Non	Aucun(e)		AUTO_INCREMENT
<input type="checkbox"/> 2	url	varchar(1024)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 3	title	varchar(512)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 4	description	varchar(2048)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 5	keywords	varchar(512)	latin1_swedish_ci		Non	Aucun(e)		
<input type="checkbox"/> 6	clicks	int(11)			Oui	NULL		

Figure 8 : La table sites

2.3.3. Code SQL pour créer la base de données :

```
-- Base de données : `dive`
```

```
--
```

```
CREATE DATABASE IF NOT EXISTS `dive` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
```

```
USE `dive`;
```

--

-- Structure de la table `images`

--

```
DROP TABLE IF EXISTS `images`;
CREATE TABLE IF NOT EXISTS `images` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `siteUrl` varchar(512) NOT NULL,
  `imageUrl` varchar(512) NOT NULL,
  `alt` varchar(512) NOT NULL,
  `title` varchar(512) NOT NULL,
  `clicks` int(11) DEFAULT NULL,
  `broken` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

--

-- Structure de la table `sites`

--

```
DROP TABLE IF EXISTS `sites`;
CREATE TABLE IF NOT EXISTS `sites` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `url` varchar(1024) NOT NULL,
```

```
`title` varchar(512) NOT NULL,  
`description` varchar(2048) NOT NULL,  
`keywords` varchar(512) NOT NULL,  
`clicks` int(11) DEFAULT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;  
COMMIT;
```

Conclusion :

Tout au long de la préparation de notre projet nous avons essayé de mettre en pratique les connaissances acquises durant notre cours Technologie des Moteurs de Recherche et cela dans le but de mettre en place un moteur de recherche performant en utilisant PHP, HTML5, JavaScript, CSS et MySQL.

En cours de ce projet, nous avons conceptualisé et nous avons développé le back-end et le front-end de DIVE, ensuite nous avons créé la base de données, finalement, nous avons exécuté le crawler sur plusieurs sites afin de pouvoir indexer plus de 3000 éléments (sites et images).

Comme perspective, nous souhaiterons que DIVE soit un commencement dans le domaine des moteurs de recherche dans le but de l'améliorer et l'adapter aux changements des besoins des utilisateurs.