

Cahier des charges

Projet Web et base de données

Yunbei Zhang

M2 IM INALCO

Table des matières

1. Présentation du projet:
 - 1.1 Contexte
 - 1.2 Description de l'objectif
2. Données
 - 2.1 Description des données
 - 2.2 Organisation des données (Base de données)
3. Le site
 - 3.1 Architecture
 - 3.2 Design / charte graphique
4. Les fonctionnalités
 - 4.1 Description (côté utilisateur)
 - 4.2 Explicitation (côté développeur)
5. Bilan du travail effectué
 - 5.1 Où en est le projet ?
 - 5.2 Difficultés rencontrées
 - 5.3 Évolutions possibles
 - 5.4 Retour personnel

1. Présentation du projet

1.1 Contexte

Aujourd'hui, nous sommes dans une société de l'information. Les informations sont liées étroitement à notre vie quotidienne que nous ne pouvons pas vivre sans, cependant, nous devons passer à une étape intermédiaire qui concerne à extraire des informations utiles depuis une grande base de données. En conséquence, nous nous intéressons à manipuler et à extraire des données, et réaliser une interaction dynamique afin de faciliter la recherche d'information.

1.2 Description de l'objectif

Dans le cours Web et base de données, nous avons vu en cours comment établir une base de données à l'aide du logiciel MAMP, créer des jointures entre des différentes tables de données avec la fonctionnalité MAMP phpMyAdmin, écrire une page web avec le langage XHTML ainsi que l'utilisation de PDO pour la partie interaction PHP/MySQL, etc. Le but du projet est donc de réaliser un tel site web qui propose aux utilisateurs des fonctionnalités afin de réaliser l'interaction qui permet un accès à la base de données via PHP et extraire des informations conviennent à la requête.

Le choix de fonctionnalités n'est pas restreint, chacun peut établir ses propres fonctionnalités. En ce qui me concerne, mon choix du thème correspond à des films ayant obtenu un ou plusieurs Oscars depuis 2000. J'ai choisi à implémenter 4 fonctionnalités sur mon site final : un moteur de recherche permettant aux utilisateurs de trouver des informations sur les films qui leur intéressent selon de différents critères ; suppression de données visées ; ajout des commentaires aux films; visualiser les commentaires ajoutés.

2. Données

2.1 Description des données

Pour la partie l'extraction de données, j'ai consulté principalement les sites de Wikipédia au titre de "List of Academy Award-winning films"(cf. [Source](#)), il présente une liste de films ayant obtenu un nombre d'oscar, il s'agit de 4 colonnes : Le titre du film, année d'obtention d'oscar, le nombre d'oscar obtenus ainsi que le nombre de nomination. D'abord, j'ai extrait les 3 premières colonnes et les ai rangés dans un fichier CSV. Cependant il manque des données pour

les types d'Oscar qu'un film a remportés, je cherche par la suite la base de données sur le site officiel d'Oscars (cf. [Source](#)), et cette fois-ci, j'ai récupéré les types d'Oscars obtenus par chaque film. Ceci me donne une idée de faire la jointure de données d'Oscars avec des données de l'information de films. Malheureusement je n'ai pas pu trouvé un fichier seul qui comporte toutes ces informations, en conséquence j'ai retourné sur la page Wikipédia et récupérer des données unes par unes. Ce processus peut être un peu fastidieux et demande beaucoup de temps. Enfin les données que j'ai récupérées sont le titre du film, l'année d'obtention d'oscars, le nombre d'oscars obtenus, les types d'oscars obtenus, l'année de sortie, le pays de sortie, les catégories, directeurs, acteurs ainsi que le résumé.

2.2 Organisation des données (base de données)

Après avoir récupéré des données, il s'agit du pré-traitement et réorganisation de données. Le formatage de données est un processus important car cela doit être reconnu et accepté par PhpMyAdmin que l'on va employer plus tard. Il s'agit de 2 fichiers CVS dont l'un stocke les informations d'Oscars de chaque films. Il contient 4 colonnes : année d'obtention d'oscars, identifiant de film (commence par 1 puis incrémenté), nombre d'oscars obtenus, types d'oscars obtenus. Le séparateur de colonnes utilisé est '\t', et chaque ligne présente des données sur un film. Si un champ comprend plusieurs valeurs, par exemples un film a remporté à la fois l'Oscars de Comédiens et de Compositeurs, ces deux valeurs seront séparé par ' ';', chaque valeur pourra être détectée grâce à la grammaire de MySQL. L'autre fichier contient 7 colonnes, il enregistre des informations sur des films propres : identifiant de film, le titre, l'année de sortie, le pays de sortie, les catégories de film, les directeurs, les acteurs et le résumé. De la même manière, les données différentes sont séparées par '\t' et des données dans un même champs sont séparées par un ';'.

Jusqu'au présent, nous avons terminé le pré-traitement du corpus, la prochaine étape s'agit de construire la base de données avec l'outil MAMP. MAMP est un acronyme informatique de macOS comporte Macintosh, dont Apache (le serveur web), MySQL (le serveur de base de données) et Perl, Php ou Python(langage de script), il sera employé étant un serveur pour exécuter des sites Web dynamiques. MAMP nous propose un outil appelant phpMyAdmin (PMA), qui est une application web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL. Cette interface pratique permet

d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base de données sous forme de fichier .sql et d'y transférer ses données.

Pour construire une base de données et passez aux étapes suivantes, il faut d'abord connecter au serveur MAMP. Une fois connecté au serveur, nous nous rendons sur le WebStart page de MAMP, puis choisir l'outil phpMyAdmin. Nous pouvons commencer à manipuler les données et établir une base de données. Étant donné que mon corpus se compose de deux fichiers, j'ai créé une base de données nommé *Oscars*, et dans cette base de données, j'ai créé en suite deux tables pour structurer respectivement mes deux fichiers. La table *oscars_films* enregistre des données d'oscars, et l'autre, *films_info* stocke les données de l'information de films. La clé primaire de la table *oscars_films* est identique que celle de *films_info*, c'est *id_film*.

Maintenant Il ne reste qu'à insérer les fichiers. Voici l'aspect de la base de données après l'importation de données :

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	award_year	int(11)			No	None		
<input type="checkbox"/>	2	id_film 📽	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	3	nb_oscars_obtained	int(50)			No	None		
<input type="checkbox"/>	4	type_oscars_obtained	varchar(500)	utf8_general_ci		No	None		

(graphe 1 : structure de données de table *oscars_films*)

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id_film 📽	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	title	varchar(200)	utf8_general_ci		No	None		
<input type="checkbox"/>	3	release_year	int(11)			No	None		
<input type="checkbox"/>	4	country	varchar(200)	utf8_general_ci		No	None		
<input type="checkbox"/>	5	category	varchar(200)	utf8_general_ci		No	None		
<input type="checkbox"/>	6	directors	varchar(200)	utf8_general_ci		No	None		
<input type="checkbox"/>	7	actors	varchar(200)	utf8_bin		No	None		
<input type="checkbox"/>	8	plot	text	utf8_general_ci		No	None		

(graphe 2 : structure de données de table *films_info*)

award_year	id_film	nb_oscars_obtained	type_oscars_obtained
2016	1	6	ACTRESS IN A LEADING ROLE;CINEMATOGRAPHY;DIRECTING...
2016	2	3	ACTOR IN A SUPPORTING ROLE;BEST PICTURE;WRITING (A...
2016	3	2	FILM EDITING;SOUND MIXING
2016	4	2	ACTOR IN A LEADING ROLE;WRITING (Original Screenpl...
2016	5	1	SOUND EDITING
2016	6	1	COSTUME DESIGN
2016	7	1	ACTRESS IN A SUPPORTING ROLE
2016	8	1	DOCUMENTARY (Feature)
2016	9	1	SHORT FILM (Animated)
2016	10	1	SHORT FILM (Live Action)
2016	11	1	MAKEUP AND HAIRSTYLING
2016	12	1	VISUAL EFFECTS
2016	13	1	FOREIGN LANGUAGE FILM
2016	14	1	DOCUMENTARY (Short Subject)
2016	15	1	ANIMATED FEATURE FILM
2015	16	6	COSTUME DESIGN;FILM EDITING;MAKEUP AND HAIRSTYLING...
2015	17	3	ACTOR IN A LEADING ROLE;CINEMATOGRAPHY;DIRECTING
2015	18	2	BEST PICTURE;WRITING (Original Screenplay)

(graphe 3 : contenu de table *oscars_films*)

id_film	title	release_year	country	category	directors	actors	plot
1	La La Land	2016	United States	musical romantic comedy-drama	Damien Chazelle	Ryan Gosling;Emma Stone;John Legend;Rosemarie DeWi...	While stuck in traffic on a Los Angeles highway, M...
2	Moonlight	2016	United States	coming-of-age comedy-drama	Barry Jenkins	Trevante Rhodes;André Holland;Janelle Monáe;Ashton...	In Liberty City, Miami, Cuban drug dealer Juan fin...
3	Hacksaw Ridge	2016	United States;Australia	biographical war drama	Mel Gibson	Andrew Garfield;Sam Worthington;Luke Bracey;Teresa...	As a child, Desmond Doss nearly kills his younger ...
4	Manchester by the Sea	2016	United States	drama	Kenneth Lonergan	Matt Damon;Kimberly Steward;Chris Moore;Kevin J. W...	As a child, Desmond Doss nearly kills his younger ...
5	Arrival	2016	United States	science fiction drama	Denis Villeneuve	Amy Adams;Jeremy Renner;Forest Whitaker;Michael St...	Twelve extraterrestrial spacecraft appear at twelv...
6	Fantastic Beasts and Where to Find Them	2016	United Kingdom;United States	fantasy	David Yates	Eddie Redmayne;Katherine Waterston;Dan Fogler;Alis...	In 1926, British wizard and "magizoologist" Newt S...
7	Fences	2016	United States	period drama	Denzel Washington	Denzel Washington;Viola Davis;Stephen Henderson;Jo...	In 1950s Pittsburgh, Troy Maxson (Denzel Washingto...
8	O.J.: Made in America	2016	United States	documentary	Ezra Edelman	Kareem Abdul-Jabbar;Mike Albanese;Muhammad Ali	The documentary explores race and celebrity throug...
9	Piper	2016	United States	computer- animated short	Alan Barillaro	None	A flock of sandpipers hunt for food at a seashore,...
10	Sing	2016	Hungary	short	Kristóf Deák	Dorka Gáspárfalvi;Dorottya Hais;Zsófia Szamosi	The story takes place in Budapest in 1991. Zsófi (...)

(graphe 4 : contenu de table *films_info*)

Il s'agit de 249 lignes de données par table, les deux tables sont reliées par la clé *id_film*.

3. Le site

3.1 Architecture

La phase du site web est une interface graphique pour représenter les fonctionnalités dynamiques en PHP. Dans la perspective graphique, il s'agit de 4 rubriques donc 4 pages web, dont la page d'Accueil représente une introduction de l'objectif et une explication de la fonction du site.

La page "Rechercher un film" inclut un moteur de recherche qui se divise en 5 sous-moteurs de recherche, ceci dit que l'utilisateur peut lancer une recherche approximative (par chaque sous-moteurs de recherche), ou une recherche précise (recherche par sélectionner 5 critères sur les films) ; il inclut aussi un fonctionnalité de suppression de données selon le choix de critères.

Par la suite, la page "Ajouter un commentaire" est comme son nom indique, permettant aux utilisateurs de choisir un film et mettre leurs commentaires, ils peuvent aussi visualiser des commentaires ajoutés sans doute.

La dernière page "À Propos" présente l'information sur l'auteur, les fonctionnalités du site et les sources de données.

Au niveau de la structuration de balise html, chaque page Web se divise par 4 intervalles : le bandeau de page, le menu, le contenu du page et enfin le pied de page, chacune de ses pages a été validée par le site du W3C Markup Validation Service.

Dans la partie contenu du fichier *accueil.html* et *apropos.html*, il ne contient que la balise du type text (*<h2>*, *<h3>* *<p>*), car ces 2 pages compose de l'introduction ou des explications du site, on n'a pas besoin d'autres balises dans le contenu.

Le fichier *recherche.html* a été bien formalisé en ces 4 intervalles, dans la partie du contenu, il s'agit d'un *<form>* composé de 5 *<filedSet>*, Chaque *<filedSet>* inclut une formulaire de type *<select>* qui est un champ du choix de critères pour la recherche, puis les deux boutons qui suivent servent à soumettre le résultat.

Le fichier *commentaire.html* a un *<form>* compose d'un *<filedSet>* qui compose de 17 *<select>* qui suivent respectivement un bouton de type ration, un *<h3>*, un formulaire de type *<textarea>*, et deux bouton de soumise.

3.2 Design / charte graphique

Pour la partie de charte graphique, j'ai régularisé d'abord le body du site, pour que les 4 pages a une vue et un format unifié. Dans le fichier *site.css*, à l'intérieur de *body {}*, l'image d'arrière plan a été paramétrisé à "*background.jpeg*", puis d'autres propriétés comme *margin* spécifie la marge de chaque côté de page, *width* pour le largeur de page, et l'affichage du texte est centré en défaut. J'ai aussi fixé le pied de page en bas de la fenêtre avec *position: fixed;*.

J'ai ajouté une image "*transparent.png*" sur l'arrière plan, et *transparent url('transparent.png')* permet de rendre l'image plus transparente. Cela facilite l'écriture et enrichit le CSS.

Le paramétrage des hyperlinks est réalisé par *a:link*, *a:visited*, *a:hover*, *a:active*.

4. Les fonctionnalités

4.1 Description (côté utilisateur)

Nous avons indiqué antérieurement que le site propose 4 fonctionnalités aux utilisateurs : rechercher des informations sur des films ayant obtenu un ou plus Oscars depuis 2000 ; supprimer des données non pertinentes ; ajouter des commentaires à un film choisi ; consulter tous les commentaires existants.

Rendez-vous sur la page "Rechercher un film" en tant qu'utilisateur, les premières choses qui sautent aux yeux sont les cinq cadres de recherche. Les titres sur le cadre de recherche sont : Recherche par année d'obtention d'Oscar, Recherche par le type d'Oscar obtenu, Recherche par le nombre d'Oscar obtenu, Recherche par l'année de sortie du film et Recherche par le pays. Dans chaque cadre de recherche, il s'agit d'un menu déroulant et deux boutons indiquant "rechercher" et "supprimer". Tout en bas de ces cadres de recherche, il y a un autre bouton indiquant "recherche par détail".

Nous commençons par le test de recherche approximative. Si l'on veut rechercher tous les films ayant remporté 5 oscars ou plus, il fallait aller à la deuxième cadre, choisir nombre ≥ 5 , et cliquer sur la touche "rechercher". Voici le résultat de recherche :

Recherche par le nombre d'Oscars obtenu

Nombre d'Oscars obtenus

✓ >=5

<5

rechercher

supprimer

(graphe 5 : recherche des films ayant 5 oscars ou plus)

Année d'Oscars	Identifiant du film	Titre du film	Nombre d'Oscars obtenus	Type_oscars_obtenus	Année de sortie	Pays	Catégorie	Directeurs	Acteurs	Résumé
2003	194	The Lord of the Rings: The Return of the King	11	ART DIRECTION;COSTUME DESIGN;DIRECTING;FILM EDITING;MAKEUP;MUSIC (Original Score);MUSIC (Original Song);BEST PICTURE;SOUND MIXING;VISUAL EFFECTS;WRITING	2003	New Zealand;United States	epic high fantasy adventure	Peter Jackson	Elijah Wood;Ian McKellen;Viggo Mortensen	Many years ago, two Hobbits, Smeagol and Déagol, are playfully fishing when Déagol discovers the One Ring beneath the river. The two quickly begin to fight over the ring, and Smeagol kills his friend. Retreating into the mountains, Smeagol gradually becomes an animalistic creature called Gollum, forgetting his old identity and becoming obsessed with the ring.
2008	117	Slumdog Millionaire	8	CINEMATOGRAPHY;DIRECTING;FILM EDITING;MUSIC (Original Score);MUSIC (Original Song);BEST PICTURE;SOUND MIXING;WRITING (Adapted Screenplay)	2008	United Kingdom	drama	Danny Boyle	Dev Patel;Freida Pinto;Madhur Mittal	18-year-old Jamal Malik, an Indian Muslim from the Juhu slum, is a contestant on the Indian version of Who Wants to Be a Millionaire?, and is one question away from the grand prize. However, before the 20 million (US\$310,000) question, he is detained and tortured by the police, who suspect him of cheating because of the impossibility of a simple "slumdog" with very little education knowing all the answers. Jamal recounts, through flashbacks, the incidents in his life which provided him with each answer.
										Dr. Ryan Stone (Bullock), a biomedical engineer from Lake Zurich, Illinois, is

(graphe 6 : résultat de la recherche par le nombre d'oscars obtenu >= 5)

Après avoir soumis la réponse, nous sommes envoyés à une autre page qui présente toutes informations des films, y compris année d'oscars, identifiant du film, titre du film, nombre oscars obtenus, type d'oscars obtenus, année sortie du film, pays de sortie, catégorie de film, directeurs, acteurs et le résumé. Nous pouvons constater que dans la colonne Nombre d'oscars obtenus, les chiffres apparues sont classées en ordre descendant, le résultat est tout à fait.

Nous testons en même temps la recherche précise, c'est à dire que dans chacun de ces 5 cadre, on choisit un critère de recherche, puis cliquer sur la touche de soumission tout en bas. Voici le résultat obtenu :

Recherche par année d'obtention d' Oscar

2016

Recherche par le type d' Oscar obtenu

PICTURE

Recherche par le nombre d' Oscars obtenu

<5

Recherche par l'année de sortie du film

2016

Recherche par le pays

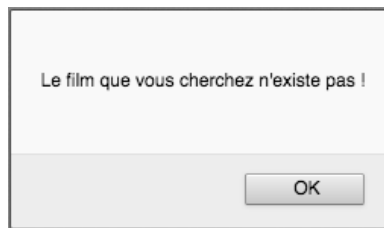
United States

(graphe 7 : recherche précise en choisissant 5 critères)

Année d'Oscars	Identifiant du film	Titre du film	Nombre d'Oscars obtenus	Type_oscars_obtenus	Année de sortie	Pays	Catégorie	Directeurs	Acteurs	Résumé
2016	2	Moonlight	3	ACTOR IN A SUPPORTING ROLE;BEST PICTURE;WRITING (Adapted Screenplay)	2016	United States	coming-of-age comedy-drama	Barry Jenkins	Trevante Rhodes;André Holland;Janelle Monáe;Ashton Sanders;Jharrel Jerome;Naomie Harris;Mahershala Ali	In Liberty City, Miami, Cuban drug dealer Juan finds Chiron, a withdrawn child who goes by the nickname "Little", hiding from a pack of bullies in a crackhouse. Juan allows Chiron to spend the night with him and his younger girlfriend Teresa before returning Chiron to his mother Paula.

(graphe 8 : résultat de la recherche par le nombre d'oscars obtenu >= 5)

On peut constater que plus la recherche précise, moins de résultat pertinent. Cela pourra être un défaut de la recherche précise, et si l'on ne trouve pas de l'information pertinente, on sera informé comme par un message d'alerte :

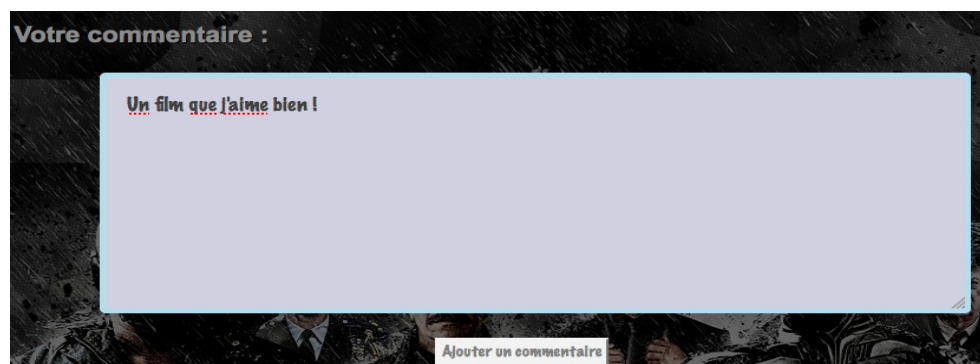


Après plusieurs testes, le fonctionnalité de supprime marche bien aussi, si l'on a choisit pour un film déjà supprimé, on ne pourra pas le supprimer une autre fois.

Nous nous rendons sur la page "Ajouter un commentaire", il s'agit de 17 menus déroulants, chacun suit d'un bouton de type ratio, il faut d'abord sélectionner l'année Oscar, puis le titre de films dans ce champs. Après la sélection, on peut mettre un commentaire à un film dans la base de donnée. Voici un le résultat :



(graphe 9 : l'année choisie est 2016 et le titre choisi est La La Land)



(graphe 10 : ajout d'un commentaire)



(graphe 11 : message retourné)

Identifiant du film	Titre du film	Commentaires
1	La La Land	Un film que j'aime bien !

(graphe 12 : enregistre du commentaire réussie et la visualisation)

Enregistrement du commentaire est réussi, graphe 12 est une visualisation des commentaires après avoir cliquer sur la touche "visualiser" les commentaires".

4.2 Explicitation (côté développeur)

En fait ce qu'on a vu précédemment concerne une interaction dynamiques, elle est réalisé par l'utilisation de l'extension PDO (PHP Data Objects) qui propose une interface permettant un accès à mysql via PHP.

Certaines fois après que l'on soumet la réponse, nous sommes envoyés vers une autre page ou nous recevons un message dans une petite fenêtre, tout ça a été écrit dans un programme PHP. Le php a été relié avec la page courante, car dans le fichier html, il s'agit d'une balise `<form>`, et on écrit le nom du fichier PHP et la méthode dans ses attributs :

```
<form action="test_projet.php" method="get" id="recherche">
```

Pour récupérer des données dans la base de données, on a besoin d'utiliser l'extension PDO, les étapes sont à respecter :

- 1) Connexion à la base de données
- 2) Définition de la requête
- 3) Envoi de la requête

4) Affichage des résultats

- Recherche par année Oscar :

D'abord je vais expliquer la fonctionnalité Rechercher et Supprimer. Pour la Recherche, j'ai écrit 5 menus déroulants, si le nom de `<select>` est "annee_oscar", il va récupérer *value* de `<option>`, puis définir la requête : chercher toutes les informations de table *oscars_films* et *films_info* si *award_year* (année Oscar) égale à *value* de `<select>`. Puis il s'agit d'envoyer la requête et afficher les résultats.

- Recherche par le type d'Oscars obtenus :

Si *value* de `<option>` est présenté dans la colonne *type_oscars_obtained*, on envoie la requête et affiche le résultat de l'information. Étant donné qu'il peut y avoir plusieurs types d'Oscars séparés par un délimiteur ';', j'ai utilisé le mot-clé LIKE dans la requête sql qui est en fait comme une expression régulière qui cherche des données incluant le motif.

- Recherche par le nb d'Oscars obtenues :

Ici j'ai donné deux choix de critères : nombre ≥ 5 ou nombre < 5 , dont `<option value="1">>=5</option>` et `<option value="2">&<5</option>`. Donc on peut faire de requêtes différentes selon *value* récupéré ("1" ou "2").

- Recherche par année sortie du film :

Cette recherche est comme la recherche par année Oscar, sauf quand on définit la requête sql, on va se rendre dans la table *films_info* en regardant si le contenu dans la colonne *release_year* convient à la condition, et si oui, on affiche toutes les informations pertinentes des films.

- Recherche par le pays d'origine :

Cette recherche a été implémentée comme la recherche par le type d'Oscars obtenus, l'application de mot-clé LIKE est indispensable.

- Recherche totale :

Sélectionner tous les critères et combiner ces 5 dernières conditions.

- La suppression travaille comme la recherche approximative, sauf que dans la grammaire de

sql, on doit utiliser le mot-clé DELETE.

– Ajout d'un commentaire

Pour ajouter un commentaire, il va falloir choisir d'abord l'année d'obtention d'Oscars, puis le titre de film. On regarde value de `<input>` qui est "ajouter" en tant que name de `<select>`, et on peut obtenir la value de `<option>` par la suite, qui est défini comme `id_film` en fait. Il faut créer une table *commentaire* dans la base de données pour qu'on puisse y insérer des commentaires, donc j'ai créé une table dans la base de données Oscars sur phpMyAdmin. Elle a 3 colonnes : `id_film`, `id_comment`, `comment`. Donc on peut insérer `id_film` et des commentaires dans la colonne `id_film` et dans `comment`.

À noter qu'il faut éviter le message d'erreur. Comme c'est une interaction dynamique, l'utilisateur ne connaît pas bien la base de données, donc il faut donner des instructions pour les diriger et aider afin de trouver de résultats pertinents. Par exemple, pour la fonctionnalité rechercher, si l'utilisateur ne choisit aucun critère pour une recherche approximative, il sera signalé qu'il faut choisir un critère dans le champs ; ensuite pour la recherche précise, si l'un des 5 champs n'est pas rempli, il sera informé aussi. Pareil pour la fonctionnalité supprimer, et de plus, si un film a été supprimé, il ne pourra pas le supprimer une autre fois et sera informé. Pour la fonctionnalité ajouter des commentaires et affichage, si un critère parmi le choix de l'année Oscars, le titre du film et ajout du commentaire n'est pas sélectionné ou rempli, il sera aussi signalé.

5. Bilan du travail effectué

Le projet que j'ai travaillé a réalisé 4 fonctionnalités et la construction de site a été finalisée. Pendant ces temps j'ai appris comment mieux extraire et structurer des données, établir une base de données, faire la jointure entre des tables, utiliser le PDO ainsi qu'employer des techniques de web. J'ai rencontré des difficultés à la phase 1, la récupération et formatage de données. Je n'ai pas trouvé un fichier qui comporte toutes les données en jeux, en conséquence, j'ai dépensé beaucoup de temps à rédiger deux fichiers cvs valides de données.

Le travaux est toujours améliorable, d'abord pour la partie d'extraction de données, il vaut mieux d'écrire un programme qui récupère automatiquement des données, ceci peut réduire le temps de travail. Ensuite, dans la phase du code php et XHTML, j'ai écrit beaucoup de code, cela aurait pu être plus propre et moins lourd. D'ailleurs pour la partie page web, ce sera plus

intéressant d'y mettre des codes de javascript.

Bref, le site que j'ai rédigé permettant aux utilisateurs de consulter des informations des films, et de supprimer des informations, d'ajouter et visualiser des commentaires, finalement, l'interaction dynamique est réussie.