**Documentation projet BDD**

1. **Présentation du projet**

Le projet consistait à créer, à partir d'une page blanche, un requêteur complet pour la base de données Sakila : location de films.

Le requêteur fonctionnera dans un navigateur et permettra d'interroger le contenu de la base de données (liste non exhaustive) : recherche d'un film par nom ou par catégorie, d'un acteur, de la disponibilité à la location, consultation de la fiche du film.

Nous pouvons faires l'analogie avec le site web Allociné.

1. **Installation**

Afin de pouvoir lancer ce projet, vous avez besoin de :

* Un serveur web capable d'interpréter du PHP ( Apache2, LAMPP, WAMMP, …)
* La base de données Sakila accessible en local où de l'extérieur ( pensez à changer les paramètres de connexion dans le fichier model/Manager.php)

Une fois ces deux choses réunies, passons à l’étape suivante :

* Déplacer le dossier Examen\_BDD dans le dossier www de votre serveur web ( ou le dossier htdocs pour ceux qui sont sous xampp).
* Assurez-vous d’avoir les droits nécessaires dans le dossier.
* C’est tout !!!!!!

1. **Architecture suivie**

Pour ce projet, nous avons utilisé les technologies suivantes :

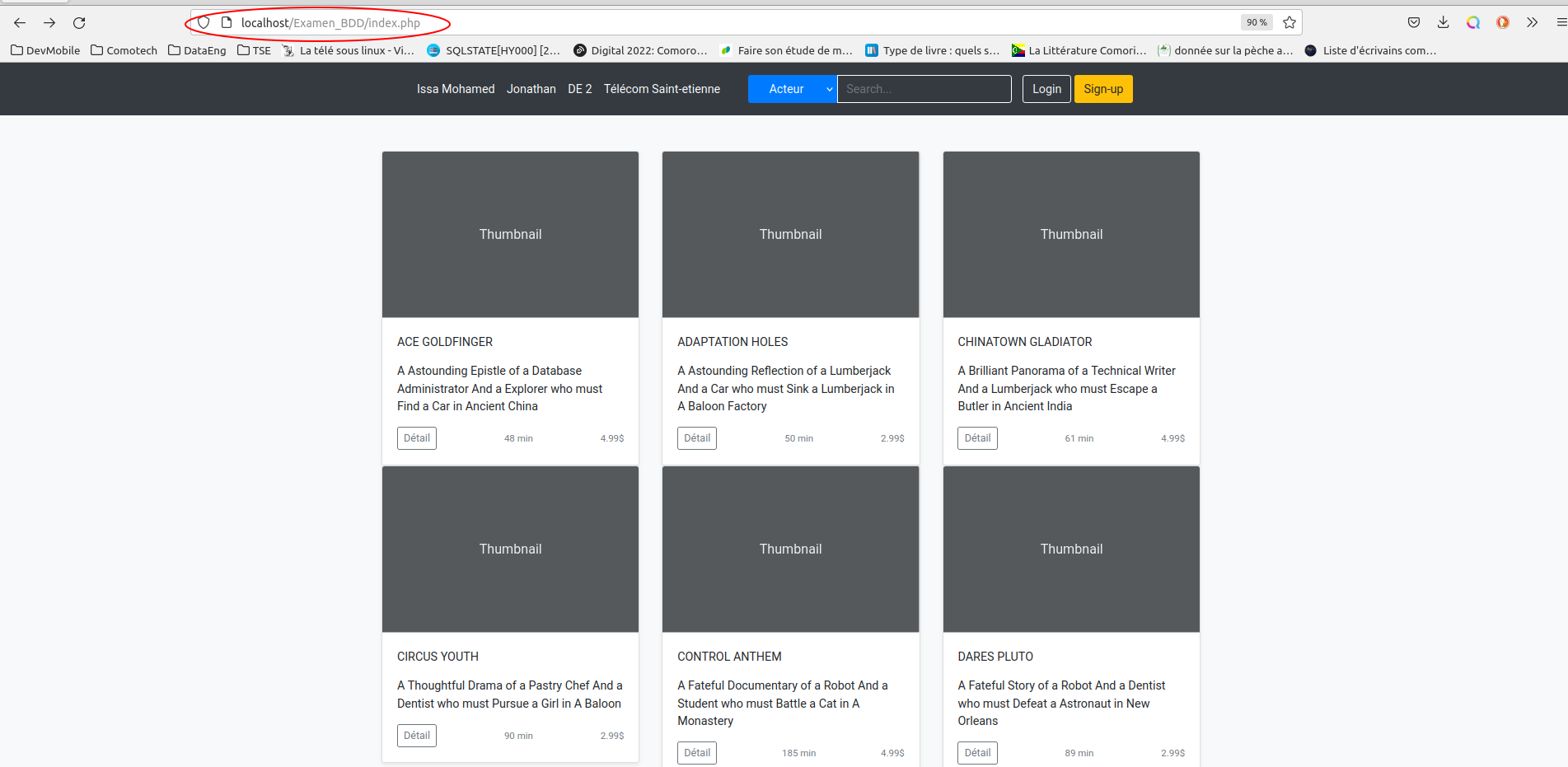
* PHP pour le backend
* HTML/CSS/Boostrap pour le font.

Ensuite, nous avons suivi le modèle de développement MVC pour l'implémentation du code. Ainsi, dans le répertoire Examen\_BDD vous trouverez :

* Un dossier modèle : ce dossier contient les programmes récupérant les données depuis la base de données.
  + Manager.php : contient toute les information nécessaire à la connexion de la base de données
  + FilmManger.php : Il s’agit de la classe répertoriant toutes les méthodes pour la récupération des données de la base SAKILA.
  + UserManager.php : Classe contenant la gestion des utilisateurs.
* Un dossier controller : Dossier faisant la liaison direct entre les données et l’affichages des données dans le HTML
  + frontendController.php : Gestion des liaison entre les données et les pages d’affichage de la partie utilisateur
  + backendController.php : Gestion des liaison entre les données et les pages d’affichage de la partie administration
* Un dossier Public :
  + CSS : Contient les CSS du site
  + images : Dossier contenant les images du sites
* Un dossier Views :
  + frontend : Dossier contenant tout les fichiers gérant l’affichage des données sur le site sur la partie frontend (utilisateur et non admin)
    - listFilmView.php : Gestion de l’affichage des films
    - Detailfilm.php : Gestion de la page détail d’un film
* Le fichier index.php : Ce fichier fait la liaison entre le l'utilisateur et le contrôleur. toutes les requêtes des utilisateurs sont arrivées sur ce fichier afin de réguler la circulation.

1. **Utilisation**
   1. **Accéder à la page d'accueil**

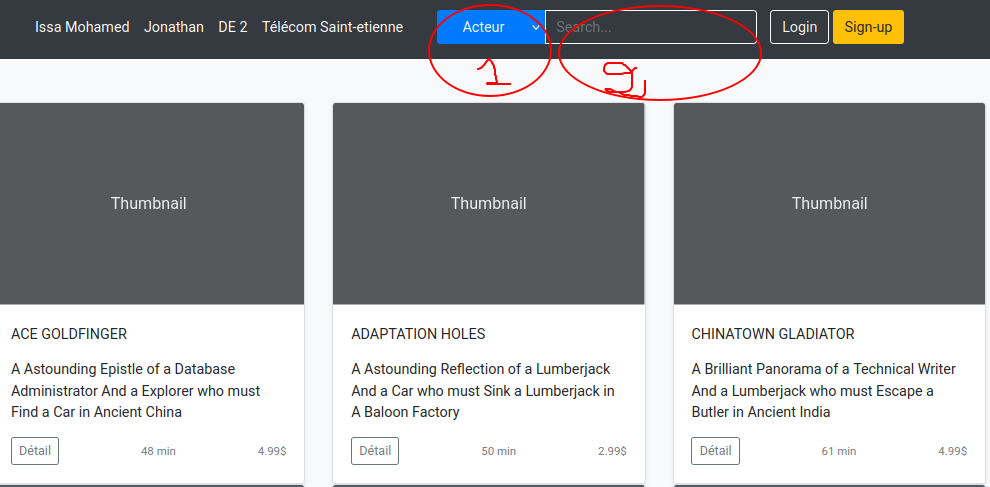
Pour accéder à la page d’acceuil, vous pouvez insérer dans la bare d’url de votre navigation le lien vers l’index.php se trouvant dans le dossier Examen\_BDD : <http://localhost/Examen_BDD/index.php>



Une fois cela réalisé, vous atterrissez sur la page d'accueil du site. Ici vous avez un affichage des films avec quelques informations.

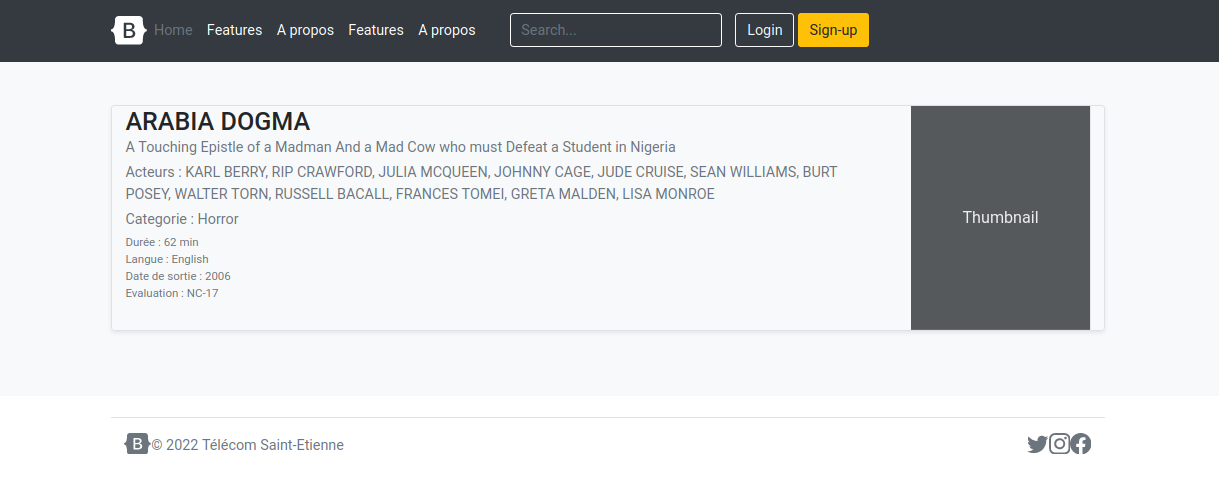
* 1. **Utilisation des filtres**

L’utilisation des filtres est assez simple.



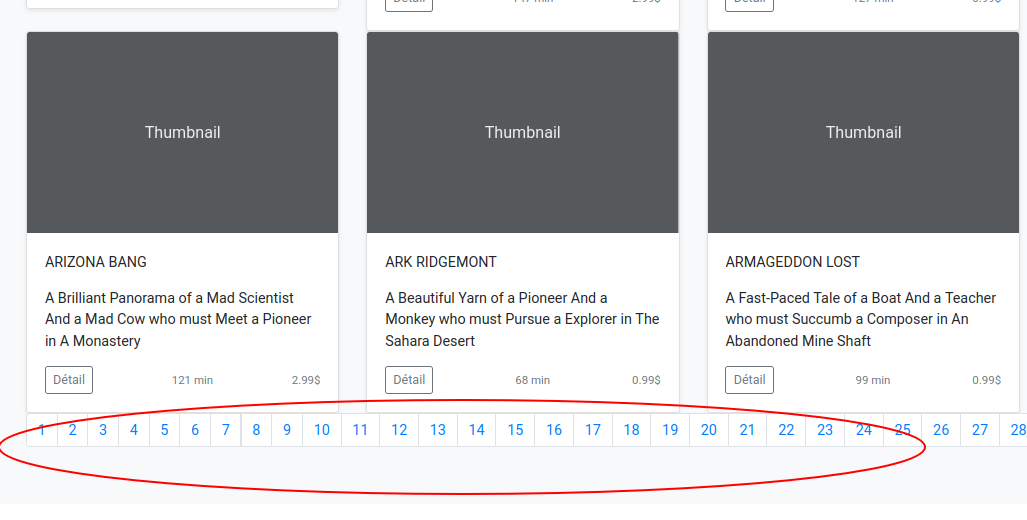
En 1 vous avez la catégorie de filtre que vous voulez réaliser. en deux le texte et appuyer sur entrée pour valider. c’est avec cette combinaison que nous avons construit notre système de filtre. Ainsi, comme vous pouvez le constater, il n’y a pas la possibilité de faire une combinaison de catégorie de filtre (acteur + catégorie ou autre)

* 1. **Page Détail**

****

* 1. **La pagination**

Afin d'alléger la lisibilité. Nous avons fait une affiche de 30 films et mis en place un système de pagination.



1. **Défis et difficulté rencontrée**

Il s’agit d’un projet très intéressant et très enrichissant. Tout au long de ce projet nous avons affronté des difficultés techniques pour sa réalisation.

En effet, le php et le html/css n'étaient pas des langages que nous avons eu à faire sur des projet assez conséquent comme celui-ci. Ainsi le défi numéro un était de s’approprier ces langages. Une tâche qui n’était pas forcément facile vu le timing et les différents projets à réaliser.

Ensuite, nous n’avons pas eu assez de temps pour travailler sur la partie admim. Enfin sous windows, les filtre ne marche pas. On a aucune idée du pourquoi. Cependant, sous linux et mac, tout fonctionne correctement.

Ainsi, en réalisant ce projet, nous avons mis en avant la compétence d’autodidacte.

1. **Recommendation**

Afin de tirer l’essentiel de ce projet à savoir la programmation web, il serait intéressant que ce projet soit réalisé après le module Technologie du web. Cela nous donnera la possibilité d’appliquer ce qu’on a pris et de le mettre en application. Ça nous permettra de faire la liaison directe entre les langages web et les bases de données en général.