



Campus  
technique

HAUTE ECOLE  
EN HAINAUT

8a, avenue V. Maistriau B-7000 Mons  
Tél : +32 (0)65 33 81 54  
Fax : +32 (0)65 31 30 51  
E-mail : tech-mons@heh.be

[www.heh.be](http://www.heh.be)

# Projet PHP

Formations de Bacheliers et de Masters organisées par la Fédération Wallonie-Bruxelles

Aout 2019  
Version 1.0

Pittonet Benjamin



FÉDÉRATION  
WALLONIE-BRUXELLES



## Table des matières

Table des matières .....	2
1. Introduction.....	3
2. Présentation générale du projet .....	3
2.1. Cahier des charges.....	3
2.2. Schéma de données .....	4
2.3. Mockup.....	5
3. Evolution du projet.....	7
3.1. Mise en place:.....	7
3.2. Conception Web:.....	7
3.3. Connexions, Inscriptions et Profils .....	9
3.4. Le cœur du site : Les fiches de jeux .....	11
3.5. Options d'administration .....	13
4. Conclusion .....	14

## 1. Introduction

Ce rapport à pour but de consigner la méthodologie utilisée pour la conception d'un projet dans le cadre du cours de Développement Web de 2eme année Bachelier en Informatique.

Ce projet à pour objectif la conception d'un site web dynamique utilisant le langage PHP.

Pour ce projet, nous avons pu choisir un sujet et déterminer notre cahier des charges en fonctions des fonctionnalités voulues pour notre site.

## 2. Présentation générale du projet

J'ai personnellement choisis de faire un site web dédié aux collections de jeux vidéo.

Mon projet à pour but de fournir une plateforme web de type librairie de jeux vidéo sur laquelle tout utilisateur pourra consulter des fiches descriptives des jeux vidéo contenus dans la base de données.

L'administrateur de la base de données pourra quant à lui éditer la base de données via des options supplémentaire comme un bouton de suppression d'une fiche de jeux ainsi qu'une page accessible par lui seul lui permettant d'ajouter des jeux à la base de données.

Chaque utilisateur pourra consulter les fiches de jeux du site sans nécessité de créer un compte. Toutefois, l'une des fonctionnalités du site ; les listes de jeux personnel, ne seront accessible qu'au utilisateur ayant au préalable créé un compte.

Une fois le compte créé, chaque utilisateur pourra créer ses propres listes dans son profil et en consulter le contenu.

Afin de populer ces listes, l'utilisateur pourra lors de sa consultation d'une fiche de jeu, l'ajouter à l'une de ces listes.

### 2.1. Cahier des charges

#### Définition du projet :

Ce projet à pour but de mettre en place une librairie de jeux vidéo.

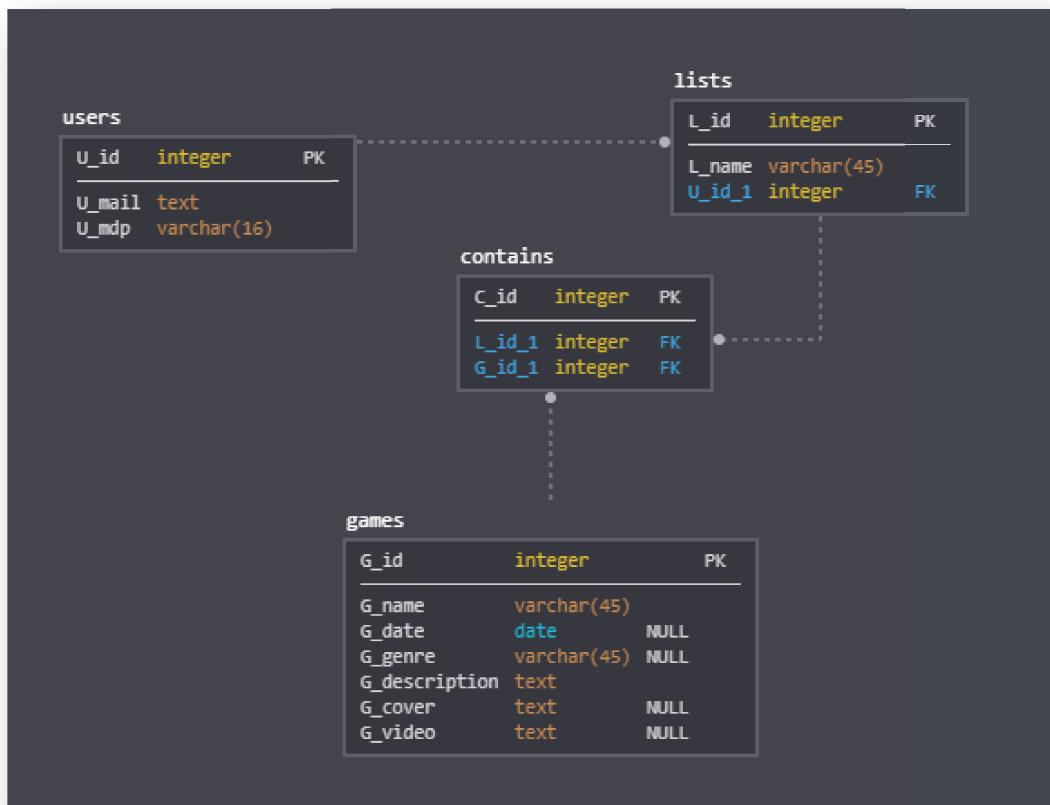
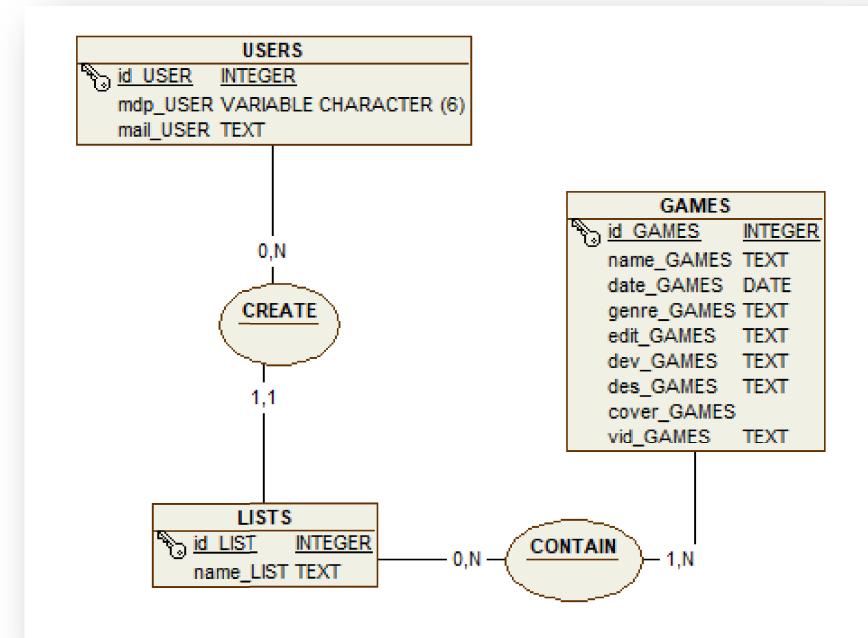
L'administrateur pourra ajouter de nouveaux jeux dans la base de données.

Les utilisateurs pourront chercher des jeux et consulter la fiche des jeux. Ils pourront aussi les ajouter à leur collection personnelle.

#### Fonctionnalité :

En tant que :	Je voudrais :	Afin de :
Administrateur	Avoir un compte sur le site avec des permissions spéciales	Pouvoir ajouter, éditer et gérer la base de données des jeux vidéo
Utilisateur	Rechercher des jeux vidéo	Voir un résumé du jeu
Utilisateur	Pouvoir me connecter sur le site	Créer ma collection de jeux personnelle
Utilisateur	Ajouter des jeux dans ma collection personnelle	Disposer d'une liste de tous les jeux que je possède

## 2.2. Schéma de données



## 2.3. Mockup

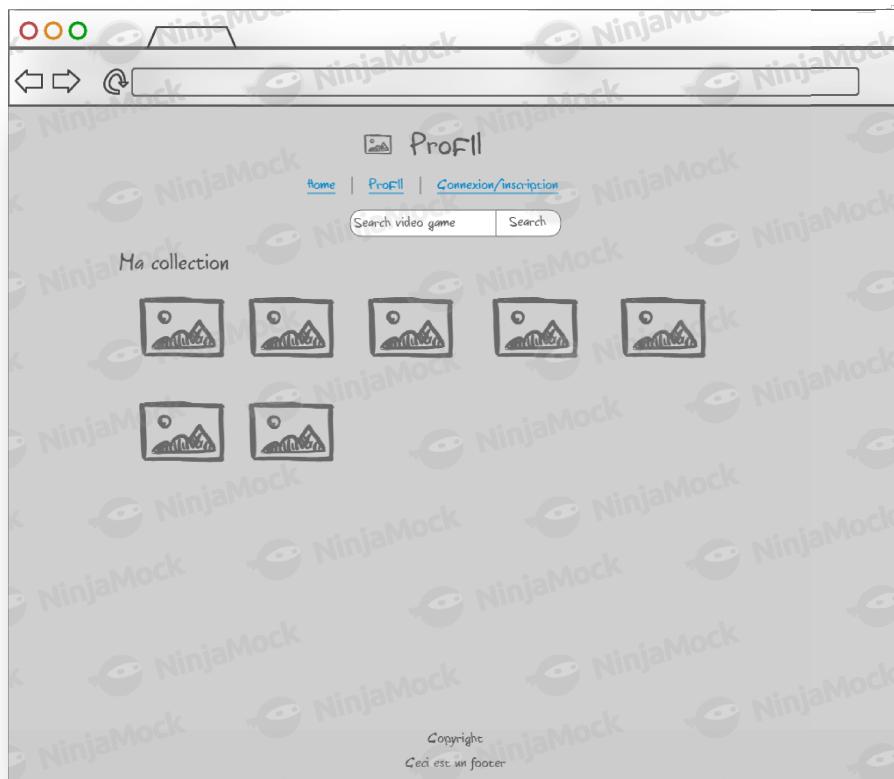
- Index :



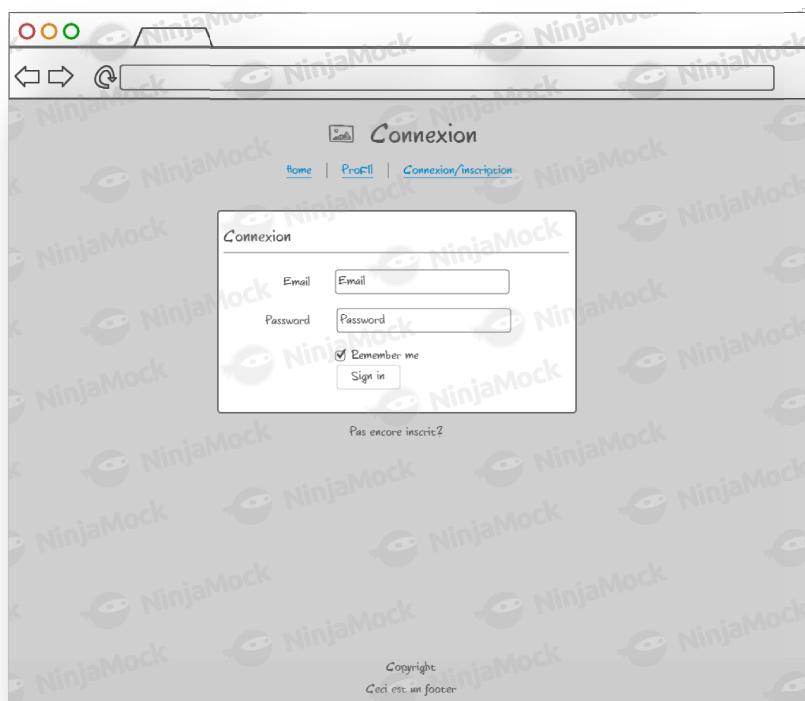
- Fiche :



- **Profil :**



- **Connexion/Inscription :**



## 3. Evolution du projet

### 3.1. *Mise en place:*

Ma première action fut la création de la base de données. Sans elle, il m'était en effet impossible de tester mes fonctionnalités, j'ai donc également ajouté manuellement les premières données afin d'obtenir une représentation visuel lorsque j'ajoutais une nouvelle fonctionnalité.

Une fois cette étape validé, J'ai commencé par réaliser un script de base permettant d'établir la connexion à cette même base de données car sans cela, aucune requête ne saurait aboutir.

### 3.2. *Conception Web:*

Ma première étape pour le projet fût la mise en place de la page d'index. L'algorithme utilisé sur l'index étant similaire à plusieurs de mes autres pages, il était donc crucial que son fonctionnement soit validé au plus tôt.

```
//Query will fetch the 15 last games added to DB
$query = $db->query('SELECT * FROM games ORDER BY `id` DESC LIMIT 15');
$game = $query->fetchAll();
foreach ($game as $game)
{//We loop to create the cards to display for each game and send the id of
those games trought the link
echo '.';
}
```

Cette méthodologie, avec une légère adaptation du code, mon permis de rapidement mettre en place le code du champ de recherche.

Je me suis par contre vite rendu compte que devoir inclure toute la partie header de l'html dans tout les pages allais vite devenir lourd compte tenu que la moindre modification sur cette partie, devrait apparaître sur chaque autres pages.

J'ai donc choisis de crée un fichier d'en tête séparer qui fût ensuite simplement ajouter à chaque pages du site, m'évitant ainsi de lourdes modification sur l'ensemble des pages.

```
//include the header file.
include 'header.php';
```

Un code tentaculaire et fragmenter offrait l'avantage d'avoir des pages PHP plus légère en code et donc plus facilement lisible. L'énorme inconvénient était cependant que le nombre de pages prévus par mon mockup allait bien vite être largement dépasser.

En effet, certaines pages ont largement bénéficié de l'ajout modulaire du code. C'est le cas notamment de la recherche de jeux dont le champ est inclus dans le fichier header. Une fois validé, ce champ passe lui-même les données de recherche à une page PHP qui fait appelle à un script PHP qui renvoie les résultats sur la page de recherche.

### Header.php :

```
<!-- Search form. Send POST to search.php via submit "Go"-->
<div style=" text-align: center;">
  <form action="search.php" method="post">
    <input name="search" placeholder="Search (Use spacebar then enter for all games)" autocomplete="on" type="text" style="width: 500px;">
    <button type="submit" name="Go">Go</button>
  </form>
</div>
```

### Search.php

```
<?php
  //include the header file.
  include 'header.php';
  //Include the searchq file that search the DB for games.
  include 'searchq.php';
?>
```

Le fichier de recherche permet donc de réappeler le script de recherche qui se trouve désormais « hors » de notre page.

Bien que pratique, cette méthode ne fut pas utiliser pour chaque fonctionnalité.

The screenshot shows the 'Gamer's Lair' website interface. At the top, there is a green circular icon with a white video game controller symbol. Below it, the text 'Gamer's Lair' is displayed. Underneath, there are links for 'Home | Profil | Sign-in/up' and a message 'Logged in as: admin@admin.com (logout)'. A button labeled 'Add games to Database' is visible. Below these, there is a search bar with the placeholder 'Search (Use spacebar then enter for all games)' and a 'Go' button. The main content area displays a search result for 'the'. Three game entries are shown in cards:

- da: Breath of the Wild**: An image of a character climbing a mountain, with the text 'da: Breath of the Wild' below it.
- Kill la Kill The Game: IF**: An image of a character with red hair and a black hoodie, with the text 'Kill la Kill The Game: IF' below it.
- GAROU: MARK OF THE WOLVES**: An image of a character with green hair and a red jacket, with the text 'GAROU: MARK OF THE WOLVES' below it.

At the bottom of the search results, the text 'Copyright benjamin pittonet.' is visible.

### 3.3. Connexions, Inscriptions et Profils

J'ai ensuite implémenté le profil afin de pouvoir tester la fonctionnalité utilisateur et administrateur.

La page de login et d'inscription appelle toutes les deux le même script afin d'effectuer l'opération voulue. Ce script test via conditions depuis quel formulaire vient les requêtes.

```
//User login. Called with sign in button.
if (isset($_POST['signin-btn']))
{//Check if user got on page via signin button.}

// User registration. Called with signup button.
if (isset($_POST['signup-btn']))
{//Check if user got on page via signup button}
```

Le code effectue alors l'opération voulue via d'autres tests afin de valider les informations. L'utilisateur est ensuite renvoyé vers son profil afin qu'il puisse gérer ces collections de jeux vidéo.

```
<?php
...
if (isset($_POST['list-btn']))
{//Check if user wants to add a list
    //Setting up variable and regex on the name to prevent unwanted characters
    $list = $_POST['postlist'];
    $list = preg_replace("/[^0-9a-zA-Z ]/", "", $list);
    if(empty($list)){...}
    else
    {
        $uid = $_SESSION['id'];
        //We insert the list into DB using name and user's id
        $sql = $db->query("INSERT INTO lists (id, name, U_id) VALUES (null,
'$list', '$uid')");
        mysqli_query($db, $sql);
        $_SESSION['add'] = "'$list' was added";
        header('location: profil.php');
    }
}
if (isset($_POST['rm-btn']))
{//Check if user wants to remove a list
    $list = $_POST['rmlist'];
    //Store query then execute
    $sql = $db->query("DELETE FROM lists WHERE id=$list");
    mysqli_query($db, $sql);
    $_SESSION['rmlist'] = "'$list' was removed";
    header('location: profil.php');
}
?>
```

```
...
<!-- This form allow user to create list -->
<form method="POST" action="profil.php" class="form-group col-md-12">
  <div class="form-group col-md-12" style="text-align: center;">
    <label for="list"><b>Create list</b></label>
    <input type="name" class="form-control" id="list" placeholder="Enter a name for the list" name="postlist" required="required" autocomplete="on">
  </div>
  <div class="form-group col-md-12" style="text-align: center;">
    <button type="submit" name="list-btn" class="btn btn-primary">Add to my lists</button>
  </div>
</form>

<?php
//We query user's list to display on his profil
$statement = $db->query('SELECT * FROM `lists` WHERE `U_id` = \''.
$_SESSION['id'].'\'');
$lists = $statement->fetchAll();
foreach ($lists as $lists)
{
  echo '
<div class="card" class="col-12" style="text-align: center;">
  <form method="POST" action="list.php" class="form-group col-md-12">
    <div class="form-group" style="text-align: center;">
      <a href="list.php?id='.$lists['id'].'" title="'.$lists['name'].'" name="'.$lists['name'].'">
        <p class="card-name">'.$lists['name'].'</p>
      </a>
    </div>
  </form>
</div>';
}
?>
```

La page profil va donc afficher les différentes listes de jeux de l'utilisateur mais aussi deux champs permettant d'ajouter des listes ou d'en supprimer.

Chaque option autorise une sélection via liste déroulante afin de choisir sur quelle liste exécuter l'action.

The screenshot shows a web application titled "GAMES LIST". At the top, there are links for "Home", "Profil", and "Sign-in/up". It displays a message "Logged in as: admin@admin.com (logout)". Below this is a button "Add games to Database". A search bar contains the placeholder "Search (Use spacebar then enter for all games)" with a "Go" button next to it. The main content area is titled "My lists of games:" and contains two sections: "Create list" and "Remove list". The "Create list" section has a text input "Enter a name for the list" and a blue "Add to my lists" button. The "Remove list" section has a dropdown menu set to "Solo games" and a blue "Remove from my lists" button. To the right, there are four categories: "Games to buy", "Versus Fighting", "Solo games", and "Racing games", each represented by a small box.

### 3.4. Le cœur du site : Les fiches de jeux

La fonctionnalité de base de mon site étant presque finie, il me fallait désormais pouvoir consulter des jeux et les ajouter à mes listes. Pour récupérer les infos du jeu, nous récupérons l'id depuis le GET des pages précédente afin d'exécuter ensuite le code SQL et de récupérer les infos du jeu

```
//Connect to db, prepare and execute query
$db = Database::connect();
$statement = $db->prepare("SELECT * FROM games WHERE games.id = ?");
$statement->execute(array($id));
$gameview = $statement->fetch();
Database::disconnect();
```

La partie suivante du code permettra de tester l'utilisateur afin d'afficher les options d'administration si l'administrateur est logger.

```
if (!empty($_SESSION['id']))
{ // If user is logged in, we display option to add/remove game to our lists }
if ($_SESSION['mail'] == 'admin@admin.com' && $_SESSION['id'] == '1')
{ //If the user logged is the admin, we add the option to remove the game from
the DB}
```

La dernière partie de cette page affichera le résultat des informations du jeu en formatant l'affichage des données.

---

Document confidentiel - Usage interne uniquement.

### Exemple de page en tant qu'administrateur :

Logged in as: admin@admin.com ([logout](#))

[Add games to Database](#)

Search (Use spacebar then enter for all games)

**Kill la Kill The Game: IF**

[Back](#)

**KILL la KILL IF**

**Release:** 2019-07-26  
**Genre:** Fighting Game  
**Editor:** Arc System Works  
**Developer:** A+ Games  
**Description:** Kill la Kill The Game: IF is an upcoming arena battle-action fighting video game being developed by APLUS and published by Arc System Works; releasing in July 25, 2019 on the Playstation 4, Nintendo Switch, and PC via Steam.

**Kill La Kill IF - Opening (Story)**

Do you want to add "Kill la Kill The Game: IF" to your lists?  
  
 No lists? [Create one](#)

Remove "Kill la Kill The Game: IF" from your lists?

Remove "Kill la Kill The Game: IF" from the Database?

Copyright benjamin nittonet

### Exemple de page en tant qu'utilisateur simple :

Logged in as: test@test.test ([logout](#))

Search (Use spacebar then enter for all games)

**Dragon Ball FighterZ**

[Back](#)

**DRAGON BALL FighterZ**

**Release:** 2018-01-26  
**Genre:** Fighting Game  
**Editor:** Bandai Namco Entertainment  
**Developer:** Arc System Works  
**Description:** Dragon Ball FighterZ est un jeu de combat 2D développé par Arc System Works et édité par Bandai Namco. Cette nouvelle adaptation de la franchise Dragon Ball met en scène les personnages iconiques de la série dans des affrontements explosifs en 3 versus 3.

**DRAGON BALL FighterZ - Opening Cinematic | X1, PS4, PC**

Do you want to add "Dragon Ball FighterZ" to your lists?  
 test list  
 No lists? [Create one](#)

Remove "Dragon Ball FighterZ" from your lists?  
 test list

Document confidentiel - Usage interne uniquement.



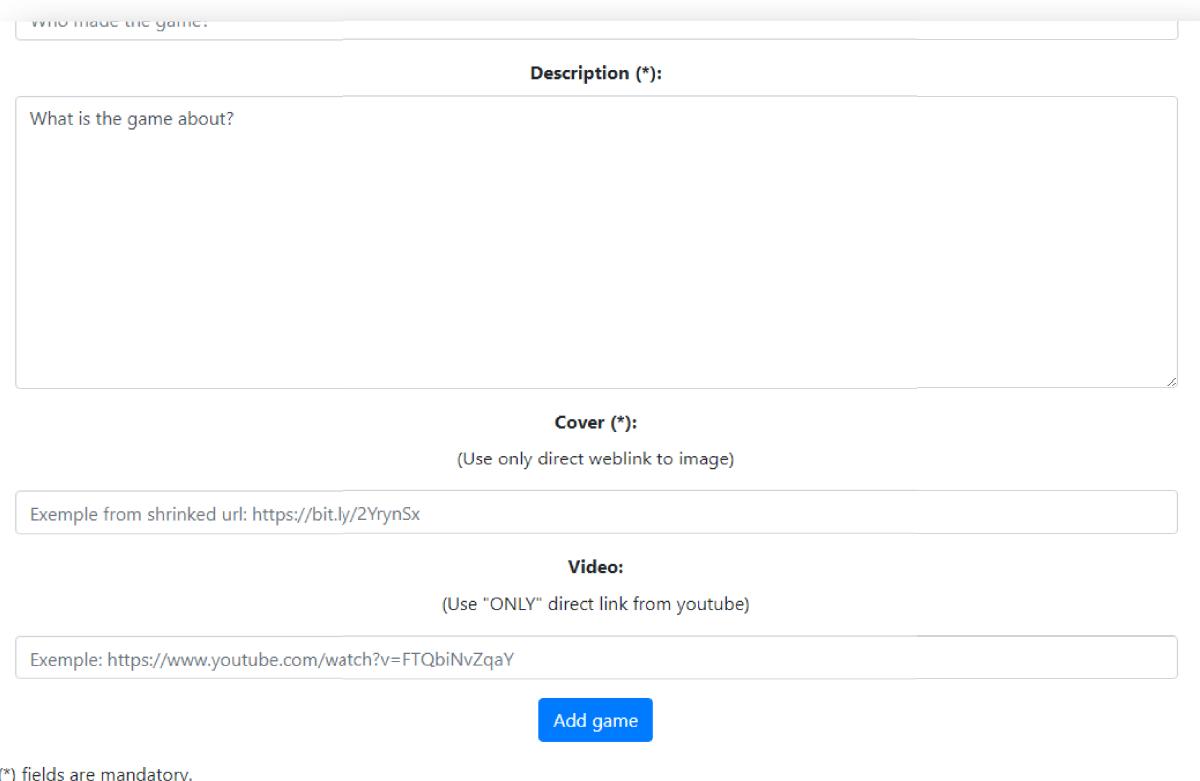
On peut remarquer que l'administrateur aura l'option d'ajouter des jeux à la base de données et ce depuis toutes les pages mais également de pouvoir supprimer un jeu depuis la fiche de ce dernier.

### 3.5. *Options d'administration*

L'option de pouvoir ajouter des jeux ne sera disponible que pour l'administrateur. Pour cela, il nous suffit simplement de vérifier qu'elle utilisateur est logger sur le site et d'afficher les options en conséquence.

Le bouton d'ajout permettra alors à l'admin d'accéder à une page sur laquelle nous vérifieront que la requête d'ajout vient bien du bouton réservera à l'admin.

Elle affichera ensuite un formulaire que l'admin devra remplir en fonction des champs demandées et qui ajoutera alors les données voulues à notre base de données.



The screenshot shows a web-based form for adding a game. At the top, there is a placeholder text 'What is the game about?' followed by a large text input area. Below this, there is a section labeled 'Cover (\*)' with a note '(Use only direct weblink to image)' and a text input field containing an example URL: 'Exemple from shrinked url: https://bit.ly/2YrynxS'. Underneath is a 'Video:' section with a note '(Use "ONLY" direct link from youtube)' and a text input field containing an example URL: 'Exemple: https://www.youtube.com/watch?v=FTQbiNvZqaY'. At the bottom right of the form is a blue 'Add game' button. A note at the bottom left states '(\*) fields are mandatory.'

## 4. Conclusion

Bien que complet selon mon cahier des charges, ce projet m'a permis de comprendre que l'ajout de fonctionnalité, aussi basique et minime soit-elle, à d'énorme implication au niveau du code PHP et du formatage de la page en elle-même.

J'avais au départ envisagé beaucoup plus d'options que j'ai abandonné au fil du projet, adaptant mon cahier de charges en fonction, afin d'obtenir un produit un temps assez correcte et viable au terme du projet.

Il y'aurait donc bien des choses à ajouter au projet afin de le rendre encore plus agréable, chercher des utilisateurs pour établir des listes d'amis, pouvoir laisser des commentaires sur les jeux, partager ces jeux avec ces amis, etc.

J'envisage d'ailleurs de changer l'orientation du projet pour mon intérêt personnel et d'en faire un site dédié aux jeux de combat pour lesquels les utilisateurs pourraient consulter et ajouter leurs propres combos pour chaque personnage des jeux.

Ce projet c'est donc montrer très complet en ce qui concerne l'apprentissage de la matière, la compréhension de la méthodologie de conception d'un site dynamique, mais aussi au niveau des idées possibles d'évolution même d'un projet.

