

Allan Boukhebza

Tristan Six



# LICENCE PROFESSIONNELLE DEVELOPPEMENT D'APPLICATION WEB ET MOBILE

## THE QUIZ

Rapport de projet

## Sommaire :

Introduction.....	2
Chapitre 1 : Les Framework et langage utilisées.....	3
Chapitre 2 : Méthodologie .....	4
Chapitre 3 : Les DAO.....	6
Qu'est-ce qu'un DAO ? .....	6
Avantage du DAO : .....	6
Liste de nos DAO : .....	7
Exemples concrets :.....	7
Chapitre 4 : Les classes.....	8
.....	8
Liste de nos <b>classes</b> : .....	9
Exemples concrets :.....	9
Chapitre 5 : les contrôleurs .....	10
Chapitre 6 : Résultat obtenue .....	11
Chapitre 7 : Conclusion.....	13
Annexes .....	14
Annexe DAO : .....	14
Annexe Classe.....	15

## Introduction

Dans le cadre de notre formation en licence professionnelle Développement d'applications Web et Mobile, nous avons eu à réaliser le projet de notre choix. Dans le langage de notre choix, sur le support de notre choix. Nous avons donc choisi de réaliser un site web de divertissement qui se nomme « THE QUIZ ». Afin de réaliser ce projet, nous avons utilisé les technologies suivantes : PHP, JavaScript, Html, CSS et SQL, notre site permet de réaliser des quiz ludiques sur tous les thèmes, chaque bonne réponse rapporte des points. Tout est personnalisable dans notre jeu à partir de nos paquets de cartes, notamment la couverture, ou le fond d'écran de votre questionnaire. Tout le fonctionnement et les explications de l'application, vous seront donnés dans les prochains chapitres de ce rapport.

## Chapitre 1 : Les Framework et langage utilisées

**PHP** : Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. Nous avons utilisé le serveur WAMP afin de générer notre PHP sur la version 7.3.21 de celui-ci. C'est un choix évident pour faire de la programmation orienté objets sur le web.

**HTML** : Le *HyperText Markup Language*, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.

Ce langage permet : d'écrire de l'hypertexte, d'où son nom, de structurer sémantiquement la page, de mettre en forme le contenu, de créer des formulaires de saisie, d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques, de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS), Comme nous avons procédé ici.

**CSS** : Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium.

**JavaScript** : JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Langage obligatoire pour tout site dynamique c'est presque une norme aujourd'hui d'en avoir sur nos site web.

**JQuery** : jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. La première version est lancée en janvier 2006 par John Resig. Nous nous en somme servie pour les célèbres fonctions d'autocomplétions.

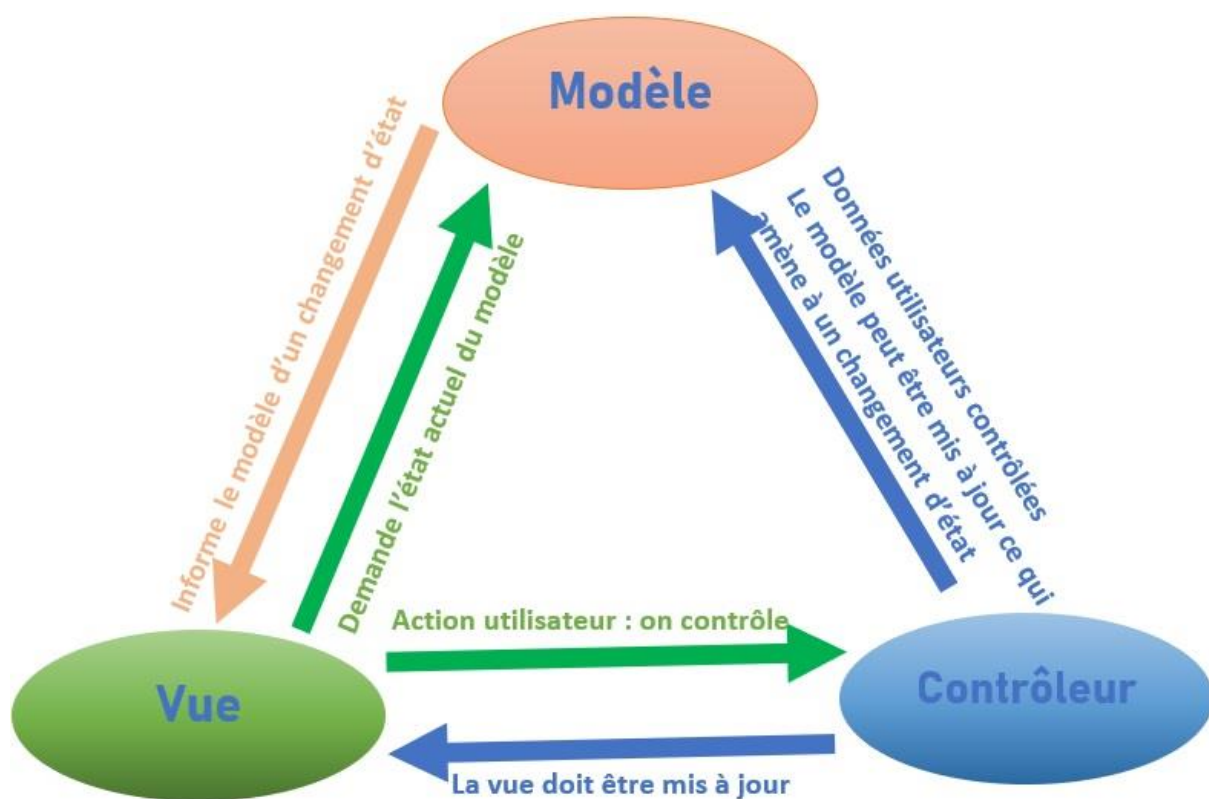
**Bootstrap** : Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. Toute la stylistique a de rare exception repose dessus, c'est un choix technique fait pour gagner du temps sur le développement et se concentrer sur le back-end.

**Git** : Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Nous avons utilisé ce système afin de partager notre travail, ce qui est le plus pratique et la norme dans les grandes sociétés de développement.

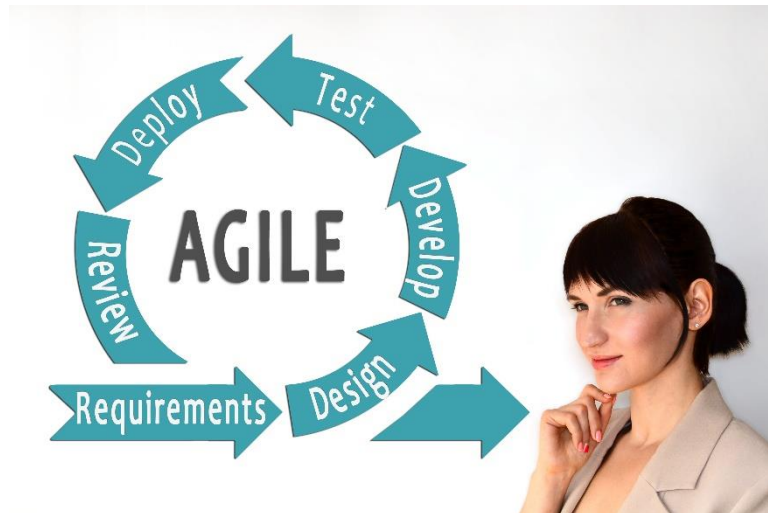
## Chapitre 2 : Méthodologie

Pour réaliser notre projet nous avons utilisé le model MVC qui est une norme web aujourd'hui, mais qu'est-ce que le modèle MVC ?

**Modèle-vue-contrôleur** ou **MVC** est un motif d'architecture logiciel destiné aux interfaces graphiques lancé en 1978, et il est très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs. Un modèle (Model) contient les données à afficher, une vue (View) contient la présentation de l'interface graphique, un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.



Grâce à ce modèle de développement, il est plus facile de travailler en groupe car les différents fichiers sont indépendants, ce qui permet de travailler sans écraser le travail de l'autre. La maintenance est surtout plus facile à l'avenir chaque dossier est explicite et plus petit donc beaucoup moins indigeste à lire.



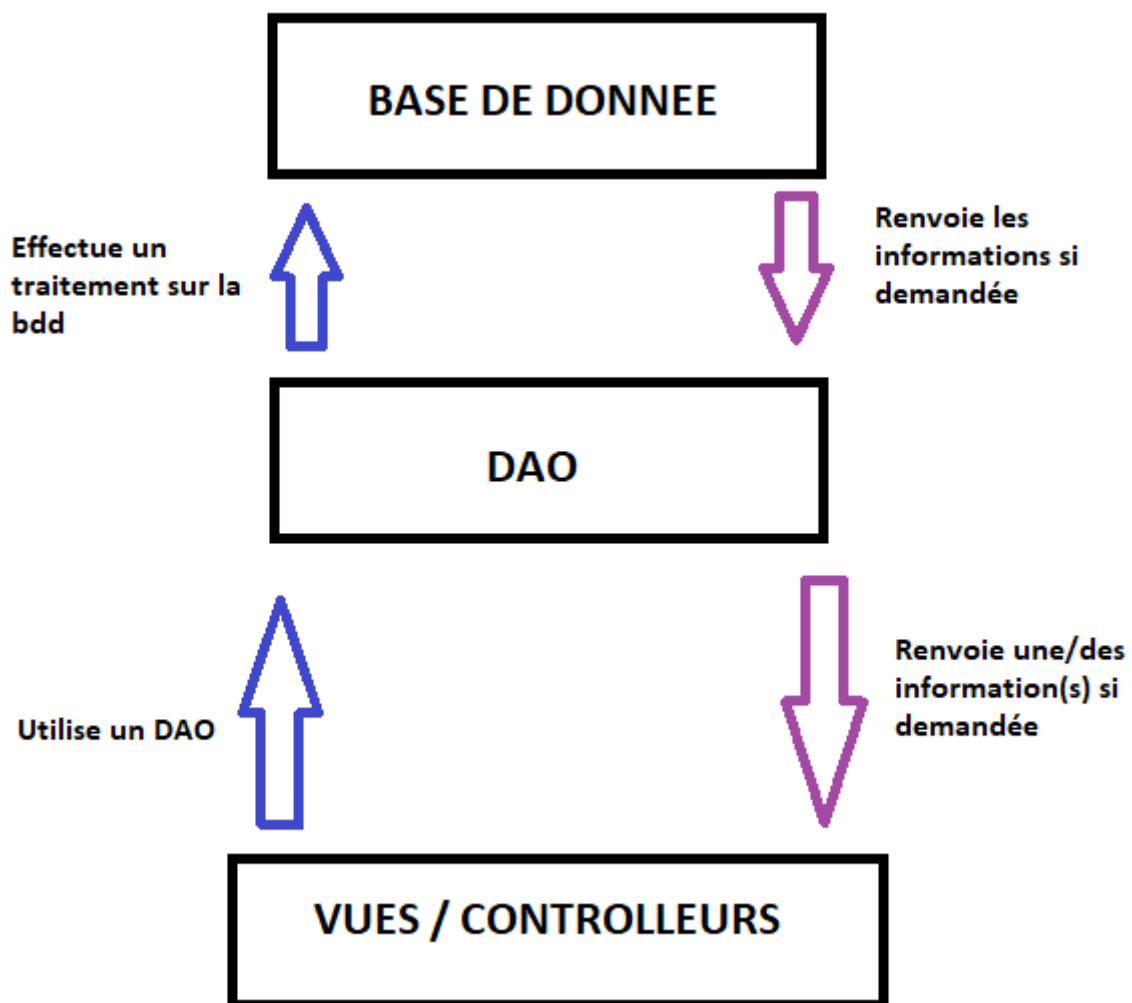
Nous avons opté pour la méthode de travail agile dans le développement de notre site web, mais d'abord qu'est-ce que la méthodologie agile :

La méthodologie Agile s'oppose généralement à la méthodologie traditionnelle waterfall. Elle se veut plus souple et adaptée, et elle place les besoins du client au centre des priorités du projet. A l'origine, cette approche a été créée pour les projets de développement web et informatique. Aujourd'hui, la méthode Agile est de plus en plus répandue car elle est adaptable à de nombreux types de projets, tous secteurs confondus. Et cette manière de faire est d'autant plus facile à mettre en place, sachant que nous étions deux.

## Chapitre 3 : Les DAO

### Qu'est-ce qu'un DAO ?

Un DAO (*Data Access Object*) est un pattern de gestion de la communication avec une base de données de façon centralisée et sécurisée. Un DAO (qui sera sous la forme d'objet) pourra créer des données, mais aussi récupérer des données spécifiques avec différents critères et paramètres, et de mettre à jour des données existantes ainsi que de supprimer des données. Un DAO devra s'occuper que d'une table de la base de données en question.



### Avantage du DAO :

Cela permet de centraliser la communication avec la base de données qui en cas de changement permet une maintenance plus rapide. Cela permet aussi de sécuriser les appels et le traitement.

Liste de nos DAO :

**CardDAO** : Permet de gérer la connexion et les traitements avec la table « *card* »

**CardPackageDAO** : Permet de gérer la connexion et les traitements avec la table « *card\_package* »

**PermissionDAO** : Permet de gérer la connexion et les traitements avec la table « *permission* »

**UserDAO** : Permet de gérer la connexion et les traitements avec la table « *user* »

Chacun de nos DAO devant renvoyer une information complète de sa table le faisait sous-forme de classe correspondant au DAO (voir prochainement sur les Objets)

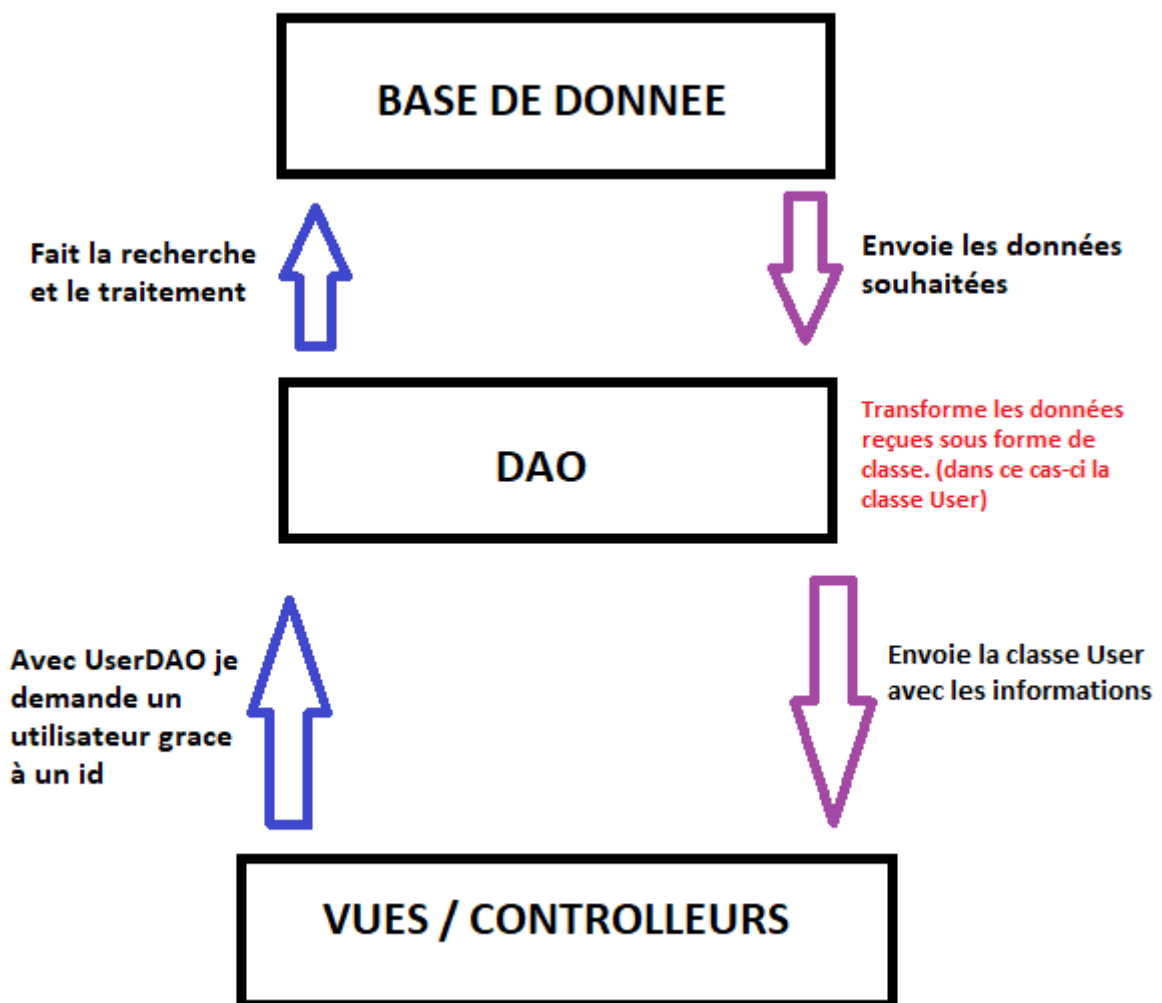
Exemples concrets :

Vous pouvez voir des exemples concrets en [Annexe DAO](#)



## Chapitre 4 : Les classes

Dans notre projet, nous avons utilisés des classes PHP dans le but de stocker toutes les informations reçues d'un DAO (voir Chapitre précédent) et de les exploiter de façon très simple. Chacune des classes contient toutes les informations d'une table avec Getter/Setter permettant la manipulation de ses données.



Liste de nos **classes** :

Nous avons autant de classe qu'il y a de DAO soit :

**Card** : classe contenant toutes les informations de la table « card » reçu par « CardDAO »

**CardPackage** : classe contenant toutes les informations de la table « card\_package » reçu par « CardPackageDAO »

**Permission** : classe contenant toutes les informations de la table « permission » reçu par « PermissionDAO »

**User** : classe contenant toutes les informations de la table « user » reçu par « UserDAO »

Exemples concrets :

Vous pouvez voir des exemples concrets en [Annexe Classe](#)

## Chapitre 5 : les contrôleurs

Les contrôleurs sont essentiels dans le modèle MVC, c'est pour cela que l'on en retrouve beaucoup dans notre projet (annexe n°1). Ils régissent chaque action importante sur notre site web et évite les soucis plus simplement. En informatique, un contrôleur est un dispositif matériel ou un programme logiciel qui gère ou dirige le flux de données entre deux entités. Ici les deux entités seront nos pages web, comme par exemple les codes suivants :

- Celui-ci gère la connexion utilisateur

(Annexe n°2)

- Celui-ci les inscriptions

(Annexe n°3)

- Celui-ci lorsque l'utilisateur veut supprimer son compte

(Annexe n°4)

Nous devons faire appelle systématiquement pour chaque aspect logique de notre site à ces contrôleurs.

Si vous voulez découvrir chacun de nos contrôleurs je vous prie de vous référer au code fourni avec ce rapport.

## Chapitre 6 : Résultat obtenue

Notre application fonctionne, les méthodes mise en œuvre pour réaliser notre projet nous a fait finir dans les temps, et avec même la possibilité de plusieurs axes d'amélioration possible. Je vous prie donc de vous référer aux captures d'écran mentionnées dans les annexes afin de comprendre, les éléments évoqués.

Page d'accueil :

(Annexe n°5)

Comme vous pouvez le voir, il y a un système de classement mise en place pour nos joueurs sur notre site, celui en première position a une coupe et un fond différent. Le code qui permet ceux-ci est le suivant (annexe n°6), et nous pouvons nous connecter depuis n'importe quelle page comme dans l'exemple (annexe n°7). Une fenêtre va apparaître, elle fera office de page de connexion, une fois connecté la barre de navigation va se modifier comme dans cette image (annexe n°8) grâce au morceau de code suivant (annexe n°9) et (annexe n°10). Nous avons aussi sur toutes les pages sauf celle de jeu, une barre de recherche pour trouver soit tous les paquets de cartes liés à un thème, soit par nom (annexe n°11).

Page bibliothèque :

Sur cette page on trouve tous les paquets de cartes du site (annexe n°12), et les paquets sont affichés par tranche de 15 (annexe n°13), on peut naviguer entre ses tranches grâce à la pagination en bas de notre page web (annexe n°14). Et nous avons accès à un bouton « jouer » qui nous permet de lancer une partie (annexe n°15) s'il y a des cartes sinon le bouton « jouer » reste caché (annexe n°16), et si l'utilisateur n'est pas connecté un message s'affichera (annexe n°17).

Page création :

Comme son nom l'indique c'est depuis cette page que nous créons nos paquets de cartes et nos cartes (annexe n°18). Comme vous pouvez le constater il peut y avoir 3 ou 4 boutons, pour l'évidente raison que si un paquet de cartes possède au moins une carte il est jouable sinon c'est impossible donc le bouton jouer est enlevé. Chaque bouton respecte un code couleurs, orange pour la modification, rouge pour supprimer et vert pour jouer.

Page création et modification de paquets :

(Annexe n°19) (annexe n°20)

Comme vous pouvez le voir sur ses captures, ils sont en tout point pareils à l'exception des fonctions de ses pages l'une sert à créer et l'autre à modifier.

Page des cartes d'un paquet :

Cette page représente les cartes d'un paquet et respecte le même code couleurs que la page de création des paquets (annexe n°21).

Page création de paquets :

(Annexe n°22)

Comme vous pouvez le voir ici on peut créer une carte et ajouter des réponses à celle-ci jusqu'à une limite de 5.

Page modification de paquets :

(Annexe n°23)

Comme vous pouvez le voir ici on récupère les données de notre carte pour la modifier et ajouter des réponses à celle-ci jusqu'à une limite de 5.

Page utilisateur :

Lorsque l'on clique sur notre adresse mail une fois connecté nous arrivons sur la page utilisateur (annexe n°24), nous y trouvons toutes les informations de l'utilisateur. Elles peuvent être modifiées en cliquant sur modifier les informations ce qui nous fait atterrir sur la page dédiée (annexe n°25), ainsi, nous pouvons supprimer notre compte depuis la page utilisateur (annexe n°26).

Page jeu :

Lorsqu'on lance une partie, un chrono se lance pour que le joueur puisse se préparer à la partie (annexe n°27), puis les questions apparaissent (annexe n°28). Après avoir répondu à toutes les questions cette écran apparait et affiche le résultats (annexe n°29) puis la partie est terminer.

## Chapitre 7 : Conclusion

Ce projet nous a beaucoup apporté que ce soit en expérience personnelle, comme en travail d'équipe, cela nous a aussi permis d'approfondir nos connaissances dans les différents langages utilisés. Nous avons pu mettre en œuvre les connaissances théoriques acquises en cours, tout au long de ce projet, et cela nous a enseigné les exigences du monde du travail.

## Annexes

### Annexe DAO :

#### DAO/CardDAO.php

```
/**
 * @param int $id_card_package
 * @param string $question
 * @param string $answer
 * @param array $fake_answers
 * @return Card|false
 */
public function createNewCard(int $id_card_package, string $question, string $answer, array $fake_answers) {
    $fakeAnswers = implode( separator: " ", $fake_answers);

    $stmt = $this->connection->prepare( query: "INSERT INTO 'card' ('id_card', 'id_card_package', 'question', 'answer', 'fake_answers') VALUES (NULL, ?, ?, ?, ?)");
    $stmt->bind_param( types: "issss", &$var1: $id_card_package, &$var2: $question, $answer, $fakeAnswers);

    return $stmt->execute() == true ? new Card($this->connection->insert_id, $question, $answer, $fake_answers) : false;
}
```

#### controllers/cardCreate.php

```
}

include_once('../src/DAO/CardPackageDAO.php');

$packageDAO = new CardPackageDAO();
$package = $packageDAO->getCardPackageById($id_package);
if($package == false || $package->getCreator()->getId() != $compte->getId()) {
    $packageDAO->close();
    header( header: "location:http://quizjapan/vue/mesCreations");
    exit(0);
}

$packageDAO->close();

$cardDAO = new CardDAO();

$card = $cardDAO->createNewCard($id_package, $_POST['question'], $_POST['goodAnswer'], $badAnswers);
if($card == false) {
    $cardDAO->close();
    $_SESSION['erreur'] = "Erreur création, veuillez essayer de nouveau";
    header( header: "location:http://quizjapan/vue/creationCard?package=" . $id_package);
    exit(0);
}

$cardDAO->close();

$_SESSION['granted'] = "Création de carte réussi !";
header( header: "location:http://quizjapan/vue/creationCard?package=" . $id_package);
```

## Annexe Classe

### class/CardPackage.php

```
/**
 * @var User
 */
private $creator;

/**
 * CardPackage constructor.
 * @param int $id
 * @param string $name
 * @param string $description
 * @param string $background_name
 * @param string $image_name
 * @param Card[] $cards
 * @param string[] $themes
 * @param User $creator
 */
public function __construct(int $id, string $name, string $description, string $background_name, string $image_name, array $cards, array $themes, User $creator)
{
    $this->id = $id;
    $this->name = $name;
    $this->description = $description;
    $this->image_name = $image_name;
    $this->background_name = $background_name;
    $this->cards = $cards;
    $this->themes = $themes;
    $this->creator = $creator;
}

/**
 * @return int
 */
public function getId(): int
{
    return $this->id;
}
```

### vue/mesCreations.php

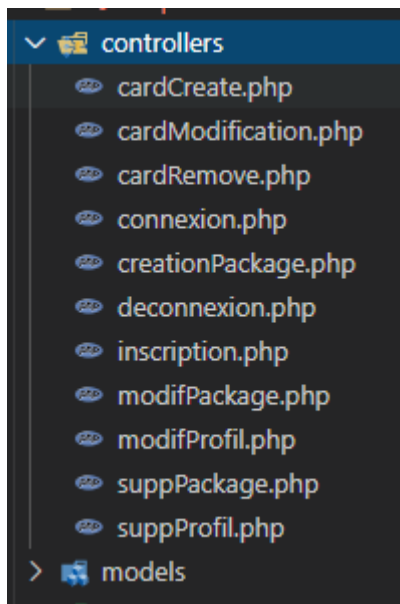
```
$list = $dao->getCardPackagesByCreator($compte->getId(), $limit, $offset);
$list = $list == false ? array() : $list;

echo "<div class='row'>";
foreach ($list as $package) {
    if($i >= $package_by_rows) {
        echo "</div>";
        echo "<div class='row'>";
        $i = 0;
    }
    $i = $i + 1;
}
?>

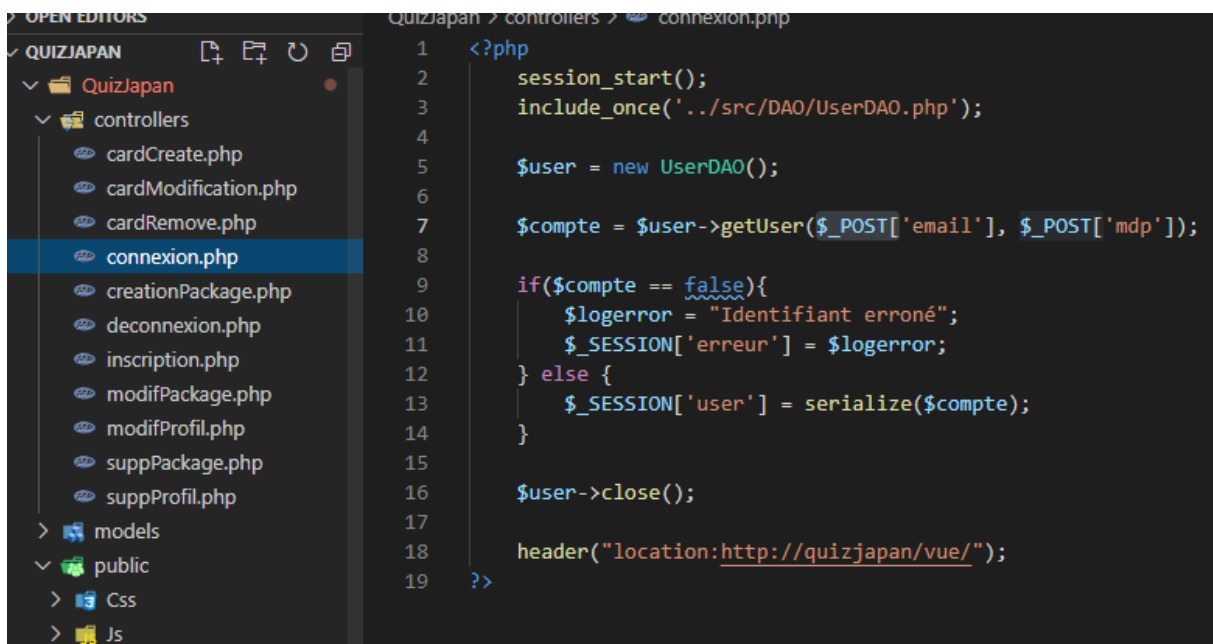
<div class="col mt-2 mb-4 text-center">
    <div class="card" style="...">
        
        <div class="card-body text-center">
            <h5 class="card-title"><?=$package->getName()></h5>
            <p class="card-text"><?=$package->getDescription()></p>
            <div class="row">
                <div class="col-6">
                    <button onClick=window.location.href='../vue/modifPackage.php?package=<?=$package->getId()>' type='button' class='card-link btn btn-warning mt-2'>Packet</button>
                    <?php
                        if(count($package->getCards()) > 0){
                            echo"<button onClick=window.location.href='../vue/intermede.php?package=". $package->getId() . "' type='button' class='card-link btn btn-success mt-2 mr-3'>";
                        }
                    ?>
                </div>
                <div class="col-6">
                    <button onClick=window.location.href='../vue/displayCard?package=<?=$package->getId()>' type='button' class='card-link btn btn-warning mt-2'>Cartes</button>
                    <button onClick=window.location.href='../controllers/suppPackage?package=<?=$package->getId()>' type='button' class='card-link btn btn-danger mt-2'>Supprimer</button>
                </div>
            </div>
        </div>
    </div>
</div>
```



## Annexe n°1



## Annexe n°2



### Annexe n°3



```
1 <?php
2 session_start();
3 include_once('../src/DAO/UserDAO.php');
4
5 $user = new UserDAO();
6
7 if($user->pseudoIsAlreadyTaken($_POST['pseudo']) == true){
8     $pseudoError = "Ce pseudo est déjà pris !";
9     $_SESSION['erreur'] = $pseudoError;
10    header("location:http://quizjapan/vue/inscription.php");
11 }
12 if($user->emailIsAlreadyTaken($_POST['email']) == true){
13     $mailErr = "Ce mail est déjà pris !";
14     $_SESSION['erreur'] = $mailErr;
15    header("location:http://quizjapan/vue/inscription.php");
16 }
17 else {
18     $compte = $user->createNewUser($_POST['nom'], $_POST['prenom'], $_POST['pseudo'], $_POST['date'], $_POST['email'], $_POST['mdp'], UserDAO::$USER_PERMISSION, 0);
19     $_SESSION['user'] = serialize($compte);
20     $user->close();
21 }
22 header("location:http://quizjapan/vue/");
23 }
24 >
```

### Annexe n°4



```
<?php
    session_start();
    include_once('../src/DAO/UserDAO.php');

    $user = new UserDAO();

    $compte = unserialize($_SESSION['user']);

    $user->removeUser($compte->getId());

    $user->close();

    session_destroy(); //Paf la session meurt
    unset($_SESSION);

    header("location:http://quizjapan/vue/");
?>
```

## Annexe n°5

[The Quiz](#) [Accueil](#) [Bibliothèque](#)  [Recherche](#) [Connexion](#)

Bienvenue sur The Quiz un site éducatif pour petit et grand !

### Classement des joueurs :

Tarskan 🏆 0 points
Mirza 0 points

[A propos](#) [Vie privée](#) [Condition d'utilisation](#)  
© 2021 The Quiz, tout droit réservé.

## Annexe n°6

```
$="container">
class="row">

```

## Annexe n°7

The screenshot shows a login modal titled "Connexion" centered on a website. The modal has a white background and a thin border. It contains two input fields: the first is labeled "Email" with a person icon, and the second is labeled with six asterisks "\*\*\*\*\*" with a lock icon. Below these fields is a prominent blue button labeled "Connexion". Underneath the button, there is a link that says "Vous avez perdu votre mot de passe ?" and another link labeled "Inscription". The background of the website is visible through the modal, showing a search bar at the top with the text "Par thème/nom" and a green "Recherche" button. At the bottom of the background, the text "Mirza 0 points" is visible.

## Annexe n°8

This screenshot shows the header of a website. On the left, there is a navigation menu with links: "The Quiz", "Accueil", "Bibliothèque", and "Mes créations". In the center, there is a search bar with the placeholder text "Par thème/nom" and a green "Recherche" button. On the right side of the header, it displays "Nombre de point : 0" followed by the email address "tarkanfrip@gmail.com" and a power icon.

## Annexe n°

```
<a class="nav-link" href="http://quizjapan/vue/bibliotheque">Bibliothèque</a>
</li>
<?php
    if(isset($compte)){
        echo ('<li class="nav-item">');
        echo ('<a class="nav-link" href="http://quizjapan/vue/mesCreations">Mes créations</a>');
        echo ('</li>');
    }
?>
</ul>
<ul class="nav nav-pills">
```

## Annexe n°9

```
if(isset($compte)){
    echo('<span class="nav-link">Nombre de point : ');
    echo $compte->getPoints();
    echo('</span>');
}
?>
</li>
<li class="nav-item">
<?php
if(isset($compte)){
    echo('<a class="nav-link" href="../../vue/menuUtilisateur.php">');
    echo $compte->getEmail();
    echo('</a>');
} else {
    echo('<a class="nav-link" data-toggle="modal" data-target="#exampleModal">Connexion</a>');
}
?>
</li>
<?php
if(isset($compte)){
    echo('<li class="nav-item">');
    echo('<a class="nav-link" id="deconnexion" name="deconnexion" href="../../controllers/deconnexion.php"><span class="fa fa-power-off fa-2x"></span></a>');
    echo('</li>');
}
?>
</ul>

<!-- connexion -->
<?php
if(isset($compte)){
} else {
    include_once('../../public/template/AffichageConnexion.php');
}
?>
```

## Annexe n°10

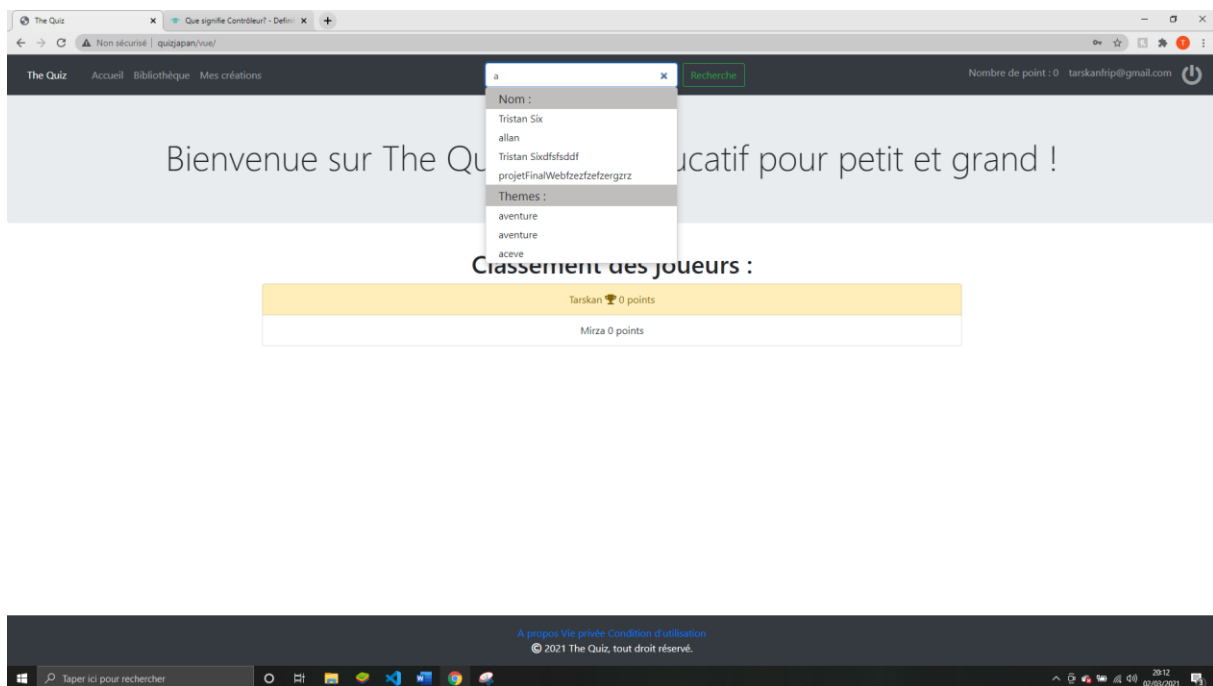
```
<ul class="navbar-nav mr-auto">
<li class="nav-item">
<form method="post" action="http://quizjapan/vue/bibliotheque" class="form-inline my-2 my-lg-0">
<input id="navbar_search_button" name="navbar_search_button" class="form-control mr-sm-2" type="search" placeholder="Par thème/nom" aria-label="Search" size="35">
<button name="navbar_submit_button" class="btn btn-outline-success my-2 my-sm-0" type="submit">Recherche</button>
</form>
</li>
</ul>
```

```

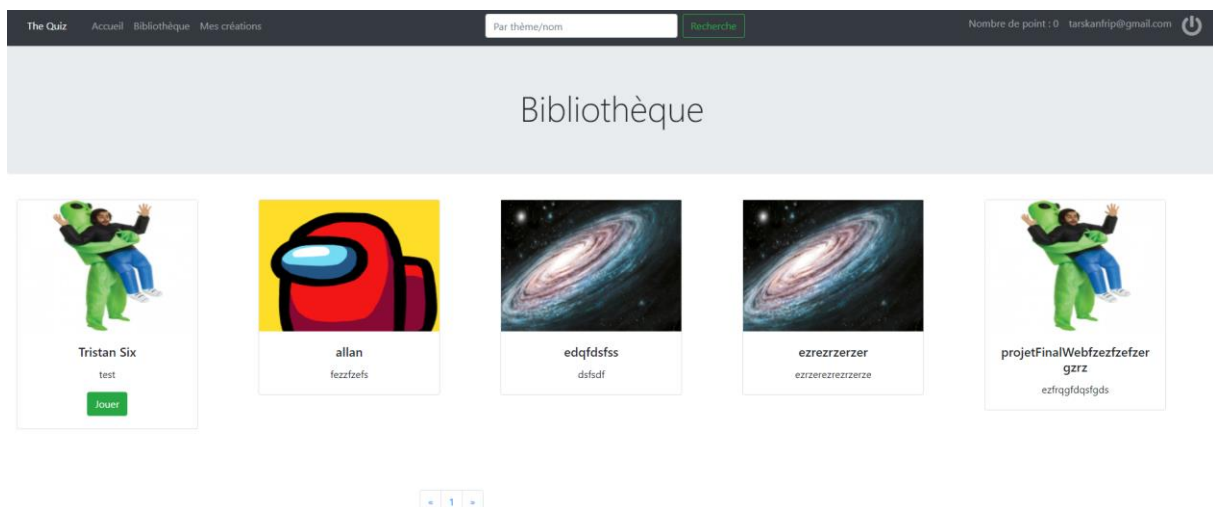
zJapan > public > Js > JS autocompleteTheme.js > $() callback > source
1  $( function() {
2      function split( val ) {
3          return val.split( /\s*/ );
4      }
5      function extractLast( term ) {
6          return split( term ).pop();
7      }
8
9      $( "#themes" )
10         // don't navigate away from the field on tab when selecting an item
11         .bind( "keydown", function( event ) {
12             if ( event.keyCode === $.ui.keyCode.TAB &&
13                 $( this ).data( "autocomplete" ).menu.active ) {
14                 event.preventDefault();
15             }
16         })
17         .autocomplete({
18             source: function( request, response ) {
19                 $.getJSON( "../src/Completion/ThemeSearch.php", {
20                     term: extractLast( request.term )
21                 }, response );
22             },
23             search: function() {
24                 // custom minLength
25                 var term = extractLast( this.value );
26                 if ( term.length < 2 ) {
27                     return false;
28                 }
29             },
30             focus: function() {
31                 // prevent value inserted on focus
32                 return false;
33             },
34             select: function( event, ui ) {
35                 var terms = split( this.value );
36                 // remove the current input
37                 terms.pop();
38                 // add the selected item
39                 terms.push( ui.item.value );
40                 // add placeholder to get the comma-and-space at the end
41                 terms.push( "" );
42                 this.value = terms.join( ", " );
43                 return false;
44             }
45         });
46     });

```

## Annexe n°12



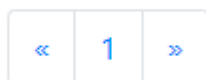
## Annexe n°13




## Annexe n°14

```
$page = isset($_GET['page']) ? $_GET['page'] : 1;
$page = is_numeric($page) ? $page : 1;
$limit = 15;
$package_by_rows = 5;
```

Annexe n°15



Annexe n°16



**Tristan Six**

test

**Jouer**

A promotional card for a game. At the top is a photograph of a man with a beard, wearing a black long-sleeved shirt and blue trousers, sitting inside a large, inflated green alien suit. The alien suit has a large head with two large black eyes and a small mouth. The man's arms are raised. Below the image, the name 'Tristan Six' is written in a bold, black, sans-serif font. Underneath the name, the word 'test' is written in a smaller, regular, black, sans-serif font. At the bottom of the card is a green rectangular button with the word 'Jouer' in white, sans-serif font.



Annexe n°17



allan

fezzfzefs

Annexe n°18



Tristan Six


test

Il faut te connecter pour jouer

## Annexe n°19

Mes créations


Crée un packet de carte



Tristan Six  
test

PacketCartes


JouerSupprimer



allan  
fezzfzefs

PacketCartes


Supprimer



edqdfsfss  
dsfsdf

PacketCartes


Supprimer



ezrezrzerzer  
ezrezrezrzerze

PacketCartes

Supprimer



projetFinalWebfzezfzefzer  
gzzz  
ezfrqgfdqpfyds

PacketCartes

Supprimer

## Annexe n°20

Retour

Données personnelle du clients

Email

tarskanfrip@gmail.com

Mot de passe

Nom

Six

Prénom

Tristan

Pseudo

Tarskan

Date de naissance

21/01/2021

Modifier les informations

## Annexe n°21

[Retour](#)

### Création du packet de cartes :

Nom du jeu de cartes :

Image du packet de cartes (favorisé des images avec une resolution 360x360)

[Choisir un fichier](#) Aucun fichier choisi

Image de fond d'écran du jeu

[Choisir un fichier](#) Aucun fichier choisi

Theme :

Description :

[Création du packet](#)

## Annexe n°22

[Retour](#)

### Modification packet de cartes :

Nom du jeu de cartes :

Tristan Six

Image du packet de cartes (favorisé des images avec une resolution 360x360)

[Choisir un fichier](#) Aucun fichier choisi

Image de fond d'écran du jeu

[Choisir un fichier](#) Aucun fichier choisi

Theme :

Aventure

Description :

test

[Modification packet](#)

## Annexe n°23

### Mes questions

[Crée une carte](#)

Question N°1

Allan pourquoi je fais ça ?

[Modifier](#)

[Supprimer](#)

Question N°2

efsd

[Modifier](#)

[Supprimer](#)

Question N°3

dsqds

[Modifier](#)

[Supprimer](#)

Question N°4

sdfsdfsdfs

[Modifier](#)

[Supprimer](#)

Question N°5

zea

[Modifier](#)

[Supprimer](#)

## Annexe n°24

[Retour aux créations](#)

### Création d'une carte :

Question

Bonne réponse

Mauvaise réponse

[Ajouter une réponse](#)

[Création de la carte](#)

## Annexe n°25

[Retour aux cartes](#)

### Modification d'une carte :

Question

Bonne réponse

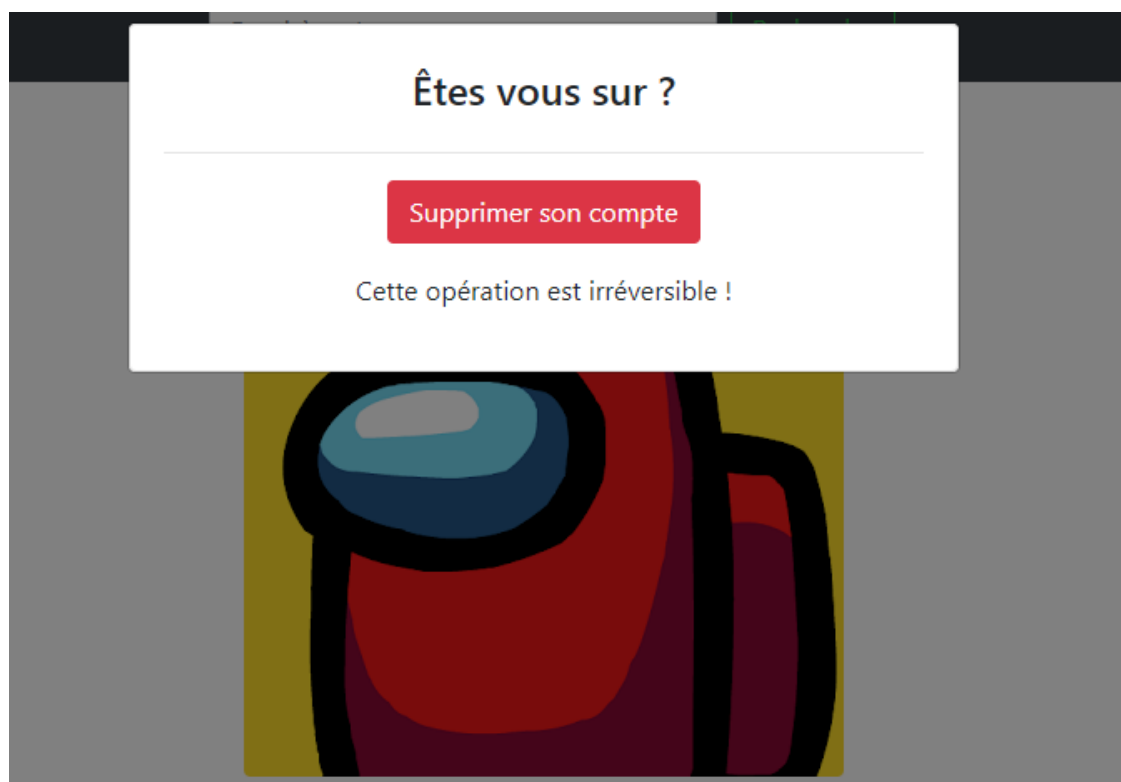
Mauvaise réponse

Mauvaise réponse supplémentaire

[Ajouter une question](#)

[Modification de la carte](#)

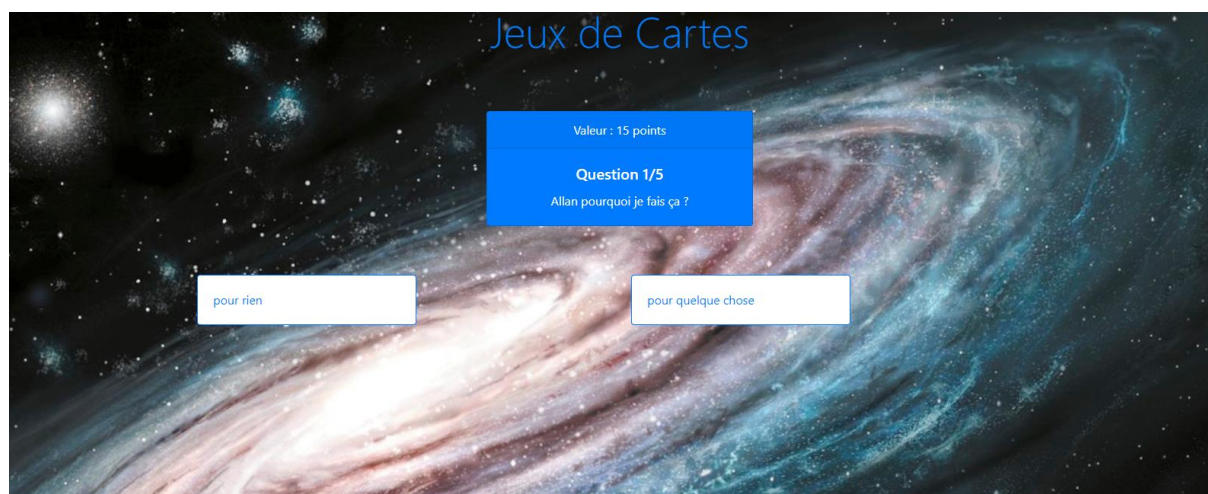
Annexe n°26



Annexe n°27



Annexe n°28



Annexe n°29

**Bravo vous avez gagnez : 45points**

Nous t'invitons à essayer tout les autres quiz de la page pour faire grimper ton score !