Randomizer Smite Build

Dossier projet pour le titre professionnel développeur Web / Web Mobile

S.O.M.M.A.I.R.E

[Introduction 3](#_Toc74686629)

[Résumé du dossier 3](#_Toc74686630)

[Cahier des charges 4](#_Toc74686631)

[Présentation du jeu 4](#_Toc74686632)

[Présentation du projet 4](#_Toc74686633)

[Les cibles 4](#_Toc74686634)

[Le contenu 4](#_Toc74686635)

[Enjeux 4](#_Toc74686636)

[Inventaire technique 5](#_Toc74686637)

[Web mobile 5](#_Toc74686638)

[Objectif à atteindre 6](#_Toc74686639)

[Critères 6](#_Toc74686640)

[Présentation de l’application 7](#_Toc74686641)

[Objectif du projet 7](#_Toc74686642)

[Fonctionnalités attendues 7](#_Toc74686643)

[Spécifications fonctionnelles 7](#_Toc74686644)

[Maquette, conception de l’interface 8](#_Toc74686645)

[Modèle de données 9](#_Toc74686646)

[Tests 9](#_Toc74686647)

# 

# Introduction

## Résumé du dossier

Smite est un jeu vidéo de type MOBA (*Multiplayer Online Battle Arena*) développé par Hi-Rez Studios. Le développement de ce jeu a commencé en 2011 et il est sorti officiellement sur PC, en 2014.

Le projet que je vais vous présenter a pour but la création d’une application web, qui va choisir pour le joueur aléatoirement un personnage de l’univers de Smite ainsi que les objets qu’il va devoir acheter au cours de la partie.

Cette application personnelle pourra possiblement s’étendre à un plus large public.

Premièrement, j’ai dû établir les spécifications fonctionnelles de l’application.

En second lieu, j’ai défini les spécifications techniques, ce après quoi il a fallu créer la base de données de l’application, avec toutes les liaisons nécessaires à son bon fonctionnement.

De plus il a fallu créer la maquette, choisir le design et les couleurs du site.

Ainsi, j’ai effectué le développement du site en commençant par la partie front, tout en respectant le mockup que j’avais mis en place.

Ce dossier comprend donc : L’expression des besoins, les spécifications fonctionnelles, les spécifications techniques, les documents liés à la création du site (mockup, arborescence, documentation…) et enfin, un extrait d’une documentation anglophone, que j’ai traduit moi-même, afin de l’utiliser pour la création de mon application.

# Définition du projet

## Présentation du jeu

Smite est un jeu vidéo d'arène de bataille en ligne multijoueur en vue à la troisième personne où les joueurs incarnent des divinités. Développé et édité par Hi-Rez Studios, le jeu est sorti le 25 mars 2014. Ce jeu servira de support pour ce projet.

## Présentation du projet

Ce projet est mis en place afin de créer des challenges sur le jeu vidéo « SMITE ». Cette application choisis aléatoirement le personnage à jouer ainsi que les objets qu’il devra acheter au cours d’une partie.

### Les cibles

Ce site s’adresse aux joueurs ayant un minimum de connaissance sur le jeu, sur les dieux mais aussi sur les objets.

### Le contenu

Tout le contenu de ce projet est libre d’utilisation.

## Enjeux

1. Mise en avant du dieu (choisit aléatoirement à partir d’une base de données) que le joueur devra jouer.

2. Mise en avant des objets (choisit aléatoirement à partir d’une base de données) que le joueur devra acheter au cours de sa partie.

## Inventaire technique

**HTML 5 :** Utilisé pour la version statique du site web

**CSS 3 :** Pour l’habillage du site

**Microsoft Office Word :** Pour la rédaction du dossier projet

**PHP 7.4.9 :** Utilisé pour rendre le site dynamique et pour la création des fonctions

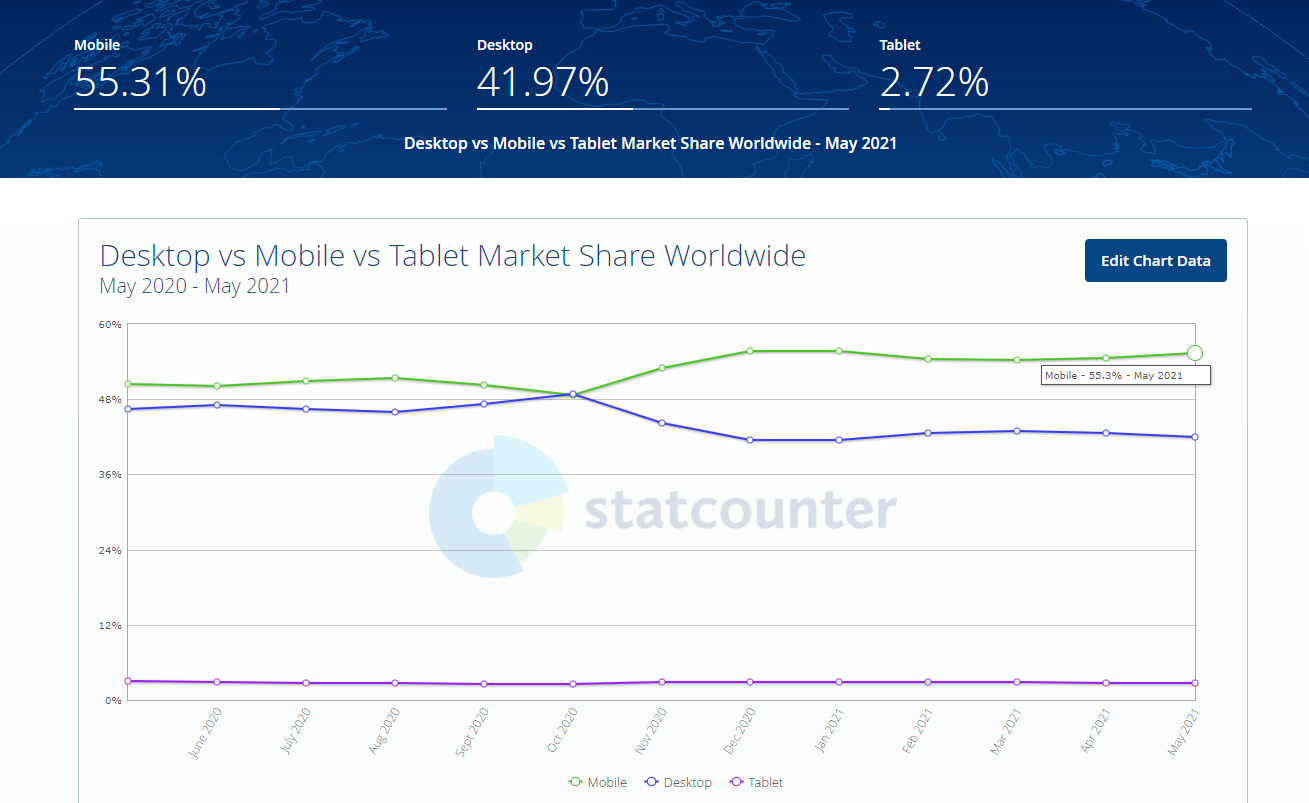
**Google Chrome et Opera GX :** Pour les différents test et l’optimisation du site

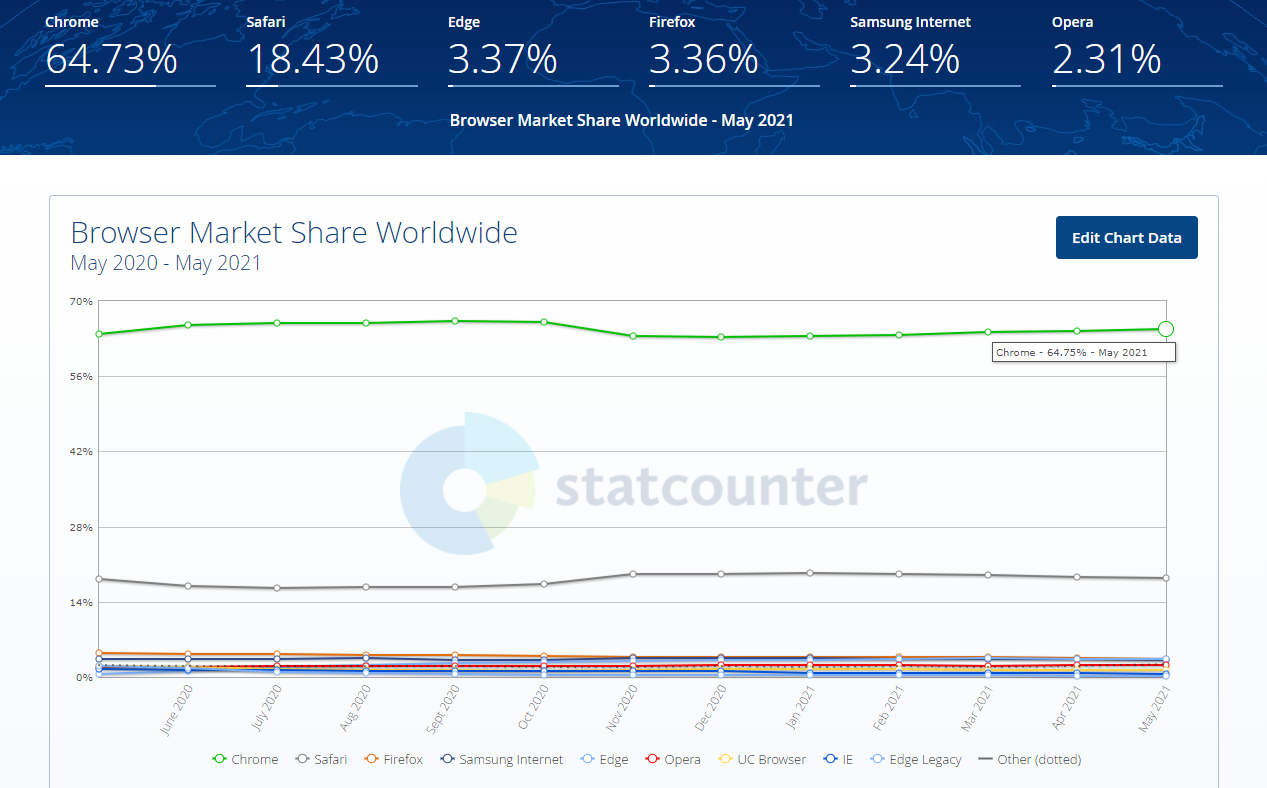
**Visual Studio Code :** L’éditeur de texte pour le développement du site

**Trello :** Pour l’organisation du projet

## Web mobile

Le site doit être adaptable sur différents appareils et navigateurs. Car en mai 2021, 64.75% des internautes utilisent Google Chrome pour naviguer sur le web. De plus, 55.3% des utilisateur utilisent un appareil mobile.





# Objectif à atteindre

Concevoir et réaliser une fonction regroupant des données récupérées via une API et une base de données pour être réutiliser pour de futures fonctions.

## Critères

Utilisation du langage Python

# Présentation de l’application

## Objectif du projet

Création d’un lecteur de musique grâce a des API de service de musique (Spotify pour commencer) avec un traitement des données (comme nos playlists) en base de données.

L’application étant un blog, actuellement, la page d’accueil présente les derniers articles postés, avec la possibilité de cliquer pour lire l’article au complet ainsi que de le commenter.

Pour accéder à l’accueil, l’utilisateur doit se connecter sur son compte, s’il n’en possède pas, un lien est présent pour le créer.

## Fonctionnalités attendues

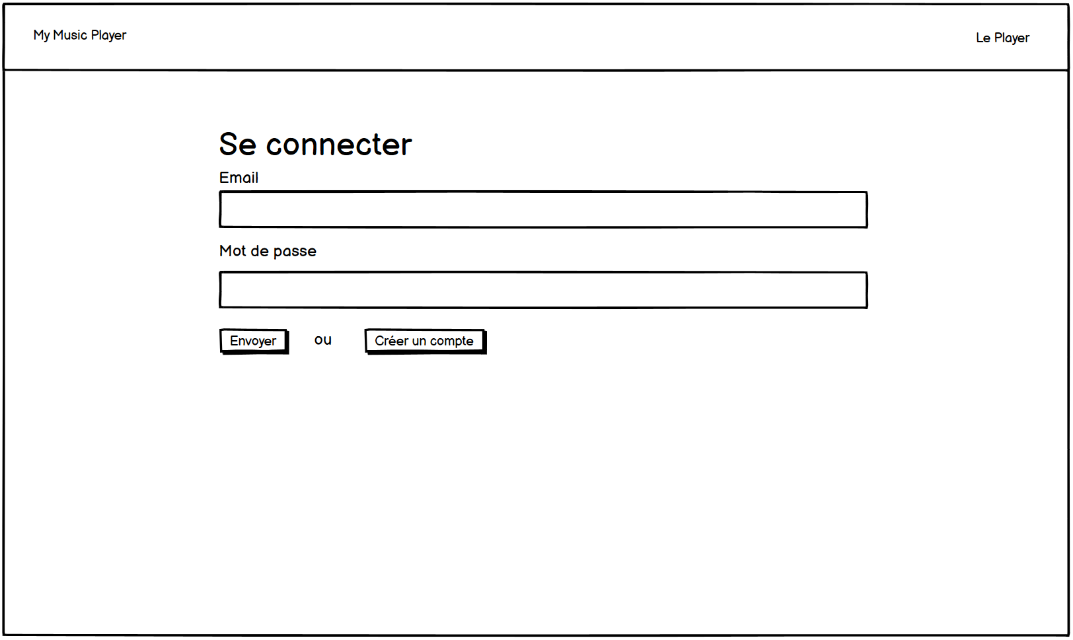
Possibilité d’insérer, de mettre à jour, de supprimer ou de lire les données en base de données comme les comptes.

## Spécifications fonctionnelles

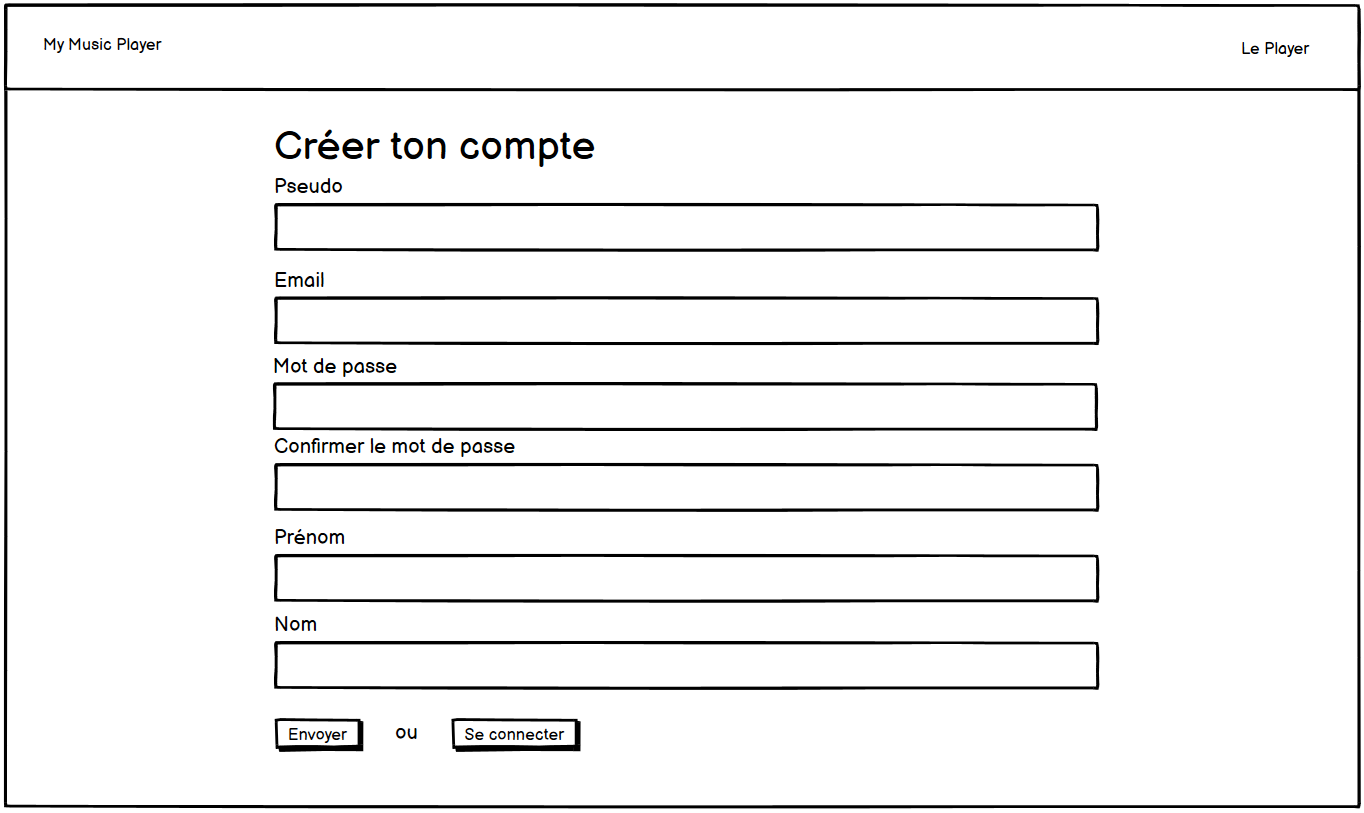
Gestion complète des catégories stockées en base de données :

* + Insertion de données
  + Mise à jour de données
  + Suppression de données
  + Affichage des données
  + Recherche des données

# Maquette, conception de l’interface

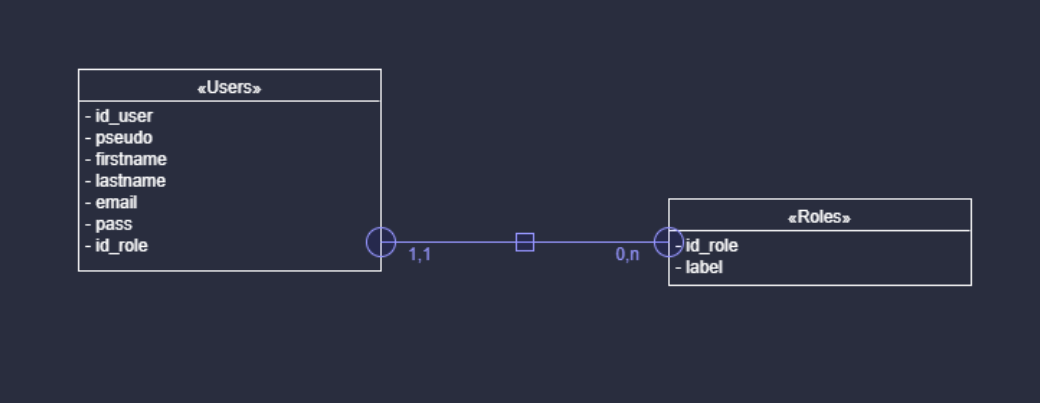
Page pour se connecter

Page pour créer son compte



Page de gestion des comptes par les administrateurs

# Modèle de données



La partie gestion de compte comporte 2 tables :

* La table « **users** » pour les données utilisateurs
* La table « **roles** » pour assigner des rôles aux utilisateurs

Les deux tables sont reliées via une clé primaire id\_role (dans la table roles) à une clé étrangère id\_role (dans la table users).

Un utilisateur a toujours un rôle, et seulement un rôle (cardinalité 1, 1) or un rôle peut ne pas avoir d’utilisateur ou en avoir plusieurs.

Par défaut, un nouvel utilisateur aura le rôle « **User** ». Son rôle pourra être modifié par la suite.

# Tests

Seul un administrateur connecté peut gérer les comptes utilisateurs.

|  |  |  |
| --- | --- | --- |
| Module | Tests | Résultat du test |
| Afficher les comptes | On vérifie qu’il y a bien des comptes dans la base de données si oui on les affiche dans un tableau | On affiche toutes les informations de tous les comptes présents en base de données (à l’exception du mot de passe) |
| Ajouter un compte | On a un formulaire pour créer un compte | * Si le formulaire est vide, on affiche un message d’erreur demandant à l’utilisateur de remplir le champ. * Si le pseudo ou l’adresse email existe déjà en base de données, alors un message d’erreur apparait. * Si les deux champs pour le mot de passe ne corresponds pas, alors un message d’erreur apparait. * Sinon, le compte est créé et stocké en base de données, et l’utilisateur est redirigé vers la page de connexion. |
| Modifier un compte | On clique sur le bouton pour éditer un compte, on est redirigé sur un formulaire, remplie par défaut par les infos du compte sélectionné (seul le mot de passe ne peut pas être modifié pour le moment) | * Si le formulaire est vide, on affiche un message d’erreur demandant à l’utilisateur de remplir le champ. * Si le pseudo ou l’adresse email existe déjà en base de données (sauf si c’est l’email et le pseudo du compte en cours de modification), alors un message d’erreur apparait. * Sinon les informations sont modifiées en base de données et l’administrateur est redirigé vers la page de gestion des comptes. |
| Supprimer un compte | On clique sur le bouton pour supprimer un compte |  |