Projet web

Présentation générale

CIR2 Brest/Caen/Nantes/Rennes - 2023

ayoub.karine@isen-ouest.yncrea.fr

benoit.lardeux@isen-ouest.yncrea.fr

thibault.napoleon@isen-ouest.yncrea.fr

christophe.vignaud@isen-ouest.yncrea.fr

christophe.le-reste@isen-ouest.yncrea.fr

Sujet

Objectif

Concevoir et développer une application de lecture de musique

Cette application devra permettre de rechercher des artistes, albums et morceaux pour les lire. Elle devra aussi permettre de gérer les profils utilisateurs avec leurs listes de lecture et indiquer les derniers morceaux joués.

Fonctionnement interne :

- **Partie *front-end*** : recherche, profils utilisateurs, listes de lecture, derniers morceaux joués, lecture des morceaux
- **Partie *API*** : partie permettant de fournir les informations au front (*API REST*)
- **Communication *front-end* et *API*** : *AJAX*

Processus de développement

Analyse :

- *MCD*
- Maquette du site web (*FIGMA, MockFlow…*)
- Requêtes *REST* associées aux pages présentées

Conception et développement :

- *HTML*
- *CSS*
- *JavaScript*
- *PHP*
- *SQL*

Test

Recette fonctionnelle et mise en place dans le serveur

Cahier des charges

Fonctionnalités

6 fonctionnalités principales sont attendues :

1. Authentification
2. Accueil
3. Résultats d'une recherche
4. Détails pour un artiste ou un album
5. Gestion des listes de lecture
6. Compte utilisateur

Attention

Les fonctionnalités additionnelles que vous développerez seront évaluées uniquement si les 6 fonctionnalités principales sont en place.

Cahier des charges

Fonctionnalités - détails

Fonctionnalité 1 : Authentification (pas de *OAuth2*)

- Si le compte existe :
  - E-mail
  - Mot de passe
  - Bouton se connecter
- Sinon bouton créer un compte :
  - Prénom
  - Nom
  - Age
  - Email
  - Mot de passe
  - Vérification du mot de passe
  - Bouton s'inscrire

Bonus :

- Gérer les mots de passe oubliés
- Utilisez les sessions
- Captcha

Cahier des charges

Fonctionnalités - détails

Fonctionnalité 2 : Accueil

- Champ de recherche avec filtre
  - Filtrage des artistes
  - Filtrage des albums
  - Filtrage des morceaux
- Derniers morceaux écoutés
  - Affichage des 10 derniers morceaux écoutés par l'utilisateur
- Liste de lectures disponibles
  - Affichage des listes de lecture de l'utilisateur
- Liste des morceaux favoris

Bonus :

- Tri des résultats de recherche par date
- Tri des résultats de recherche par affinité de style
- Voir 10 morceaux supplémentaires pour les derniers morceaux joués

Cahier des charges

Fonctionnalités - détails

Fonctionnalité 3 : Résultats d'une recherche

- Affichage des résultats :
  - Liste des artistes correspondants
  - Liste des albums correspondants
  - Liste des morceaux correspondants
- Les éléments doivent être cliquable pour permettre de voir le détail

Bonus :

- Pagination
- Classement par préférence

Cahier des charges

Fonctionnalités - détails

Fonctionnalité 4 : Détails pour un artiste, un album, un morceau

- Pour un artiste :
  - Nom
  - Type (groupe, chanteur…)
  - Liste de ses albums
- Pour un album :
  - Artiste
  - Titre
  - Date de parution
  - Image
  - Style musical (rap, pop…)
- Pour un morceau :
  - Artiste
  - Album
  - Titre
  - Durée

Bonus :

- Éléments cliquables pour relancer une recherche

Cahier des charges

Fonctionnalités - détails

Fonctionnalité 5 : Gestion des listes de lecture

- Visualisation des listes
- Création de nouvelles listes

  → avec date de création
- Ajout de morceaux dans une liste à partir d'une page de détails d'un album

  → avec date d'ajout
- Ajout en favoris d'un morceau d'une liste de lecture
- Suppression :
  - d'un morceau dans une liste
  - d'une liste

Bonus :

- Gestion de liste par style musical
- Classement des listes

Cahier des charges

Fonctionnalités - détails

Fonctionnalité 6 : Compte utilisateur

- Gestion du compte :
  - Email
  - Nom
  - Prénom
  - Age
  - Mot de passe

Bonus :

- Photo de profil
- Nettoyage des dernières écoutes

Cahier des charges

Fonctionnalités

Interactions client – serveur, la partie *API* doit permettre de :

- Répondre aux requêtes du *front* (client) en respectant l'architecture *REST* :
  - Récupération de données: *GET*
  - Ajout de données : *POST*
  - Modification de données : *PUT*
  - Suppression de données : *DELETE*
- Interroger la base de données
- Renvoyer des informations au format *JSON*

Base de données et *back* :

- Toutes les informations doivent être stockées dans une base de données relationnelle modélisée en utilisant les règles de normalisation classique
- Utilisation de *PDO* en *PHP* pour les interactions avec la base de données
- L'utilisation de classes *PHP* est fortement recommandée

Cahier des charges

Technologies à utiliser

Partie *front-end* :

- Le front doit être développé uniquement avec les technologies :

- Les bibliothèques suivantes, et uniquement celles-ci, peuvent être utilisées :

- Pour l'authentification :

  → Utilisation de *PHP* possible

Cahier des charges

Technologies à utiliser

Partie *API* :

- Exclusivement en *PHP* 
- *SCBD* de type *PostgreSQL* ou *MariaDB*  
- Communications avec le client avec *AJAX* 
- Échange de données avec le client en *JSON* 

Déploiement :

Configuration obligatoire de la machine virtuelle (VM) :

- *Apache 2*
- *PHP 7.4*
- *Postgresql 11 ou MariaDB 10*

Configuration de la VM :

Mettre à jour la VM :

› `sudo apt update`
› `sudo apt upgrade`

Les paquets à installer sont (avec `sudo apt install`) :

› `apache2`
› `mariadb-server`
› `php`
› `php-mysql`
› `phpmyadmin`

Cahier des charges

Contraintes

Mise en page :

Il faut que la mise en page du *front* prenne en compte un en-tête et un pied de page commun à toutes les pages.

Accès :

L'accès au site web se fera via un *virtualhost* comme suit : `http://prj-web-cir2-grp-xx` ou `xx` est le numéro du groupe

L'application doit être :

- Responsive
- Évolutive : possibilité de rajouter de nouvelles fonctionnalités
- Ergonomique : facilité d'utilisation, homogénéité des informations, respect des normes utilisées pour le Web

Le code doit être :

- Correctement architecturé
- Réexploitable : code lisible, code bien commenté, noms de variables/classes/fonctions explicites
- Séparé en plusieurs fichiers (par fonctionnalités)

Cahier des charges

Fonctionnalités bonus

De nombreuses autres fonctionnalités pourront être ajoutées, exemple :

- Ajout d'un script *Python* d'insertion de données (fortement recommandé)
- Lecture automatique de la suite d'un album
- Proposition de liste de lecture par style musical
- Proposition automatique de morceaux proches de celui en cours de lecture
- Partage de liste de lecture entre personnes
- Liste d'amis
- Interface administrateur pour l'ajout d'artistes, albums ou morceaux
- Ajout d'un chat
- Document explicatif de l'utilisation du site

Cahier des charges

Compétences techniques

Utilisées et consolidées :

- Programmation Web :
  - *Front* : *HTML/CSS/Bootstrap*
  - *API* : *PHP*
  - Communication client/serveur : *AJAX/JSON*
- Base de données :
  - Modélisation: *JMerise*
  - Requêtes *SQL* : *MySQL* ou *PostgreSQL*
- Serveur Web :
  - Déploiement et gestion d'un serveur Web

Apprises :

- Gestion de projet
- Maquette d'un site web
- Développement d'une application web complète

Déroulement

Calendrier

Organisation du projet au jour le jour :

Semaine 1 (26h)				Semaine 2 (19h)		
Mardi (3h30)	Mercredi (7h)	Jeudi (7h)	Vendredi (7h)	Lundi (7h)	Mardi (7h)	Mercredi (3h30)
	• Maquette • Modèle *BDD* • Interfaces *JS/PHP*	• **Vérification des machines virtuelles par les encadrants** • Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	• Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	• Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	• Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	**Recette fonctionnelle** ***Dépôt du rendu final avant 12h sur Moodle***
• Présentation du sujet, des attendus et des recettes • Préparation de la machine virtuelle • Maquette • Modèle *BDD* • Interfaces *JS/PHP*	**Recette MCD + Maquette** ***Dépôt du MCD et de la maquette avant 18h sur Moodle***	• Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	• Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	• Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	• Développement de l'application web coté client (JS) • Développement de l'application web coté serveur (PHP)	Soutenances et démonstrations à partir de 13h (Groupes caennais en premier)

Déroulement

Organisation

Travail en trinôme :

- Chaque étudiant dans le binôme connaît l'ensemble du projet : utiliser un gestionnaire de code (*Git* par exemple)
- Attention à bien se répartir le travail
- Ajoutez une page dans la présentation qui répond à la question : Qui a fait quoi ?

Ressources externes :

- Tous les documents sont autorisés
- Attention à utiliser avec une grande précaution tout document extérieur : site de vulgarisation, forum, code d'autrui

Documentation du projet :

- Au fur et à mesure
- Standardisée
- Livraison de code ou de document :
  - Ne pas attendre la dernière minute pour poster un livrable
  - Préparer des livrables intermédiaires (surtout pour les sources)
  - Sauvegarder régulièrement vos données
  - Mise en production :
    - Vérifier votre configuration
    - Faire régulièrement des mises en production pour vérifier le fonctionnement

Livrables

Format de l'archive :

Archive *ZIP*, *TGZ*, *7ZIP*, pas de *RAR :* `projetweb_groupeX.zip` (remplacer X par votre numéro de binôme)

Le rendu intermédiaire doit contenir :

- Un fichier *PDF* contenant, soit votre maquette en *PDF*, soit un lien cliquable qui dirige vers la maquette conçue (vérifiez si vous pouvez l'ouvrir via une navigation privée de votre navigateur)
- Les schémas de la base de données : *MCD* et *MPD*
- La répartition des tâches dans le binôme

Le rendu final doit contenir :

- Les éléments du rendu intermédiaire
- Un fichier README (format *markdown*) détaillant l'installation de votre application
- Les scripts *SQL* de création des tables et d'insertion du jeu de test
- L'intégralité de votre code commenté avec vos ressources (images, icônes…)

Attention

Les livrables seront à poster sur l'intranet. Tout retard sera sanctionné (l'heure du réseau faisant foi).
Les fichiers au mauvais format ou avec un mauvais nommage seront pénalisés.

Notations

Barème indicatif :

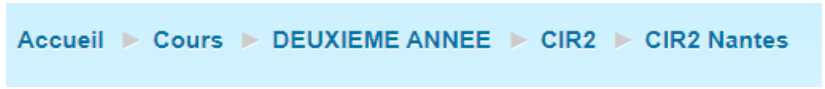
- Présentation *MCD* et maquette (rendu intermédiaire) : 20%
- Recette fonctionnelle : 40%
- Évaluation du code (rendu final) : 40%

Remarques :

- Malus possible sur l'un des membres du groupe si l'investissement est jugé trop faible
- Possibilité d'être interrogé durant le projet de façon individuelle
- Plagiat sévèrement sanctionné pour TOUS les membres du/des groupe(s)

Ressources et groupe

Chemin du projet sur Moodle :



Groupes:

	user1		user2		user3		Adresse IP
Groupe1	BUTERY	Florent	BRUN GRUNBERGER	Zelman	BERTIN	Antoine	10.10.51.71
Groupe2	PENNEC	Elouan	CHAUVEAU	Maxime	FANJUL	Esteban	10.10.51.72
Groupe3	FOUCHE	Clément	ANQUETIL	Augustin	KEBCI	Paul	10.10.51.73
Groupe4	PORODO	Théo	STEPHANT	Matthieu	BROCHOIRE	Paul	10.10.51.74
Groupe5	FONTAINE	Paul	COLLOBERT	Ryan	JOIN	Gauthier	10.10.51.75
Groupe6	GUICHET	Kilian	AZEAU BIGORGNE	Julia	CROGUENNOC	Romain	10.10.51.76
Groupe7	BELLENGER	Maxime	FOSSE	Raphael	LE PAN	Ethan	10.10.51.77
Groupe8	RENA	Dorian	YAVUZ	Emir	BEUNON	Nicolas	10.10.51.78
Groupe9	CUEFF	Allan	CAVARO	Alexandre	MEUNIER	Mathis	10.10.51.79
Groupe10	SOPRANSI	Maëlle	LE BARS	Nicolas	FRANQUET	Quentin	10.10.51.80
Groupe11	LE BOULCH	Antoine	LE GOFF	Quentin	PAITIER	Mathias	10.10.51.81
Groupe12	PERIDY	Cassie	PAJDAK	Antoine	SIMON	Nathan	10.10.51.82
Groupe13	LELIEVRE	Tom	SERMON-THUILLIER	Goustan	TAILLEBOIS	Nicolas	10.10.51.83
Groupe14	VARAKINE	Dimitri	RIOU	Margot	ROCHER	Vincent	10.10.51.84

Exemple de connexion ssh :

Pour user 1 :

ssh user1@10.10.51.71

MDP : user1

MDP à changer après la première connexion

Des questions ?