BAGGIO	Site Web Vitrine	BTS
Projet	**Mini-projet en équipe**	**SNIR**

Mode projet

Pour cette activité, nous serons en mode Projet. Cela signifie que :

- Le temps est compté ! N'hésitez pas à travailler hors des moments prévus en cours.
- Des documents sont attendus : diagramme de Gantt, cahier de tests…
- La constitution de l'équipe est fixe. En cas d'absence d'un membre à la présentation ou au rendu final, les autres membres doivent pouvoir assurer la partie « manquante ».
- L'enseignant ne sera disponible qu'en tant que « client » ou sur des conditions très exceptionnelles.

Projet et librairie

Trois projets sont disponibles ; l'un d'entre eux sera attribué à votre équipe.

Chaque projet utilisera deux librairies : une générique « NoSql.php » que vous n'aurez ni besoin de connaître, ni modifier, ainsi qu'une autre librairie qui dépend de votre projet.

Vous allez devoir utiliser directement cette dernière librairie, c'est-à-dire appeler certaines de ses fonctions selon vos besoins. De même, la documentation de ces librairies, partiellement fournie dans les documents de votre projet, est détaillée dans les commentaires du code. Enfin, vous pourriez, selon votre projet, avoir à modifier certains éléments mineurs de cette librairie.

Authentification basique

Quand une authentification basique est demandée, cela signifie une authentification basique par htaccess ou similaire :
https://developer.mozilla.org/fr/docs/Web/HTTP/Authentication#restreindre_lacc%C3%A8s_avec_apache_et_lauthentification_basique

Vous pourrez réunir les pages d'administration dans un sous-dossier, afin de simplifier votre application.

Attendus

Sont attendus :

- Les documents projets, c'est-à-dire :
  - Le diagramme de Gantt
  - Le cahier de recettage
  - Le cahier de tests
- La livraison du projet, soit :
  - L'installation du serveur web sur un appareil Raspberry pour la démo
  - Le développement du site vitrine et son déploiement sur le serveur web
  - La livraison des accès à l'authentification
  - L'accès au dépôt github privé du projet finalisé

Timeline

W48	Fin W48	Fin W49	Fin W50
Diagramme de Gantt Cahier de recettage	Première présentation du style au client	Au moins une réunion client effectuée	Cahier de tests Livraison finale

Utilisation des raspberry

Lorsque vous souhaitez mettre en place un serveur web sur Raspberry, vous aurez besoin :

- D'initialiser le Raspberry, donc :
  - De le brancher correctement
  - De le démarrer en mode BIOS (en maintenant la touche MàJ enfoncée au démarrage)
  - De lancer l'installation de Raspberry Pi OS (installation « normale »)
- D'installer les packages logiciels
  - De mettre à jour la liste des paquets via apt update
  - De mettre à jour les paquets via apt upgrade
  - D'installer les paquets apache2 , php-common , php-cli , libapache2-mod-php
- De déployer votre applicatif
  - En plaçant vos fichiers dans le dossier /var/www/html
  - En testant votre application, sur votre Raspberry, dans votre navigateur

Commandes Linux pratiques

sudo

Préfixez n'importe quelle commande avec « sudo » pour lancer la commande en mode Administrateur.

apt update

Met à jour la liste des packages

apt upgrade

Met à jour les packages

apt install <x>

Installe le package "<x>"

cd <dir>

Permet de se déplacer dans le répertoire « <dir> »

pwd

Affiche le répertoire courant

ls

Liste les fichiers et répertoires du répertoire courant

cp <f> <t>

Copie le fichier ou le répertoire « <f> » vers la cible « <t> »

Equipes

	A	B	C	D	E	G	H	J	K	L
Chef	Killian M	Rayane	Tom	Augustin	Théo	Enzo	Nihel	Vianney	Larry	Hasseb
Tech1	Lucas H	Lucas M	Mohamed B	Mehdi	Mohamed E	Hicham	Naoufel	Florian	Aimane	Danil
Tech2	Joel		Riyad	Wassim	Killian V		Kevin	Mohamed C	Calvin	

Projet : Aux bonnes recettes

Demande client de base

Ce site web, nommé « Aux bonnes recettes », permettra plusieurs choses :

- Une page principale, contenant un titre centré « Aux bonnes recettes », ainsi que le texte « Trouvez plein de recettes sur cet incroyable site qui n'est en fait qu'un prototype réalisé par les bg de BTS SNIR ! »
- Depuis n'importe quelle page, on doit pouvoir rechercher une ou plusieurs recettes par titre ou par ingrédient, dans un formulaire discret.
- Lors de la recherche, on arrive sur une page à part affichant la liste des résultats de la recherche. On peut cliquer sur chaque résultat pour afficher le détail de la recette choisie.
- Une recette est constituée d'un titre (son nom), d'une liste d'ingrédients (un texte multilignes), de consignes de préparation (un texte multilignes) et d'un temps de préparation.
- Les pages d'administration ne doivent être disponibles que pour l'auteur du site : on souhaite mettre en place une authentification basique, dont vous nous communiquerez les accès.
- Depuis les pages d'administration, on doit pouvoir sauvegarder les détails d'une recette, pour en créer, en modifier ou en supprimer.
- Le public doit pouvoir, sur une recette, la commenter et attribuer une note, une seule fois par recette. Un filtrage par IP sera mis en place.
- Sur la page d'une recette, on doit donc afficher, dans l'ordre : son titre, son temps de préparation, ses ingrédients, ses consignes de préparation, la note moyenne, et la liste des commentaires et notes donnés par le public.

Style

Le style doit être simple et épuré. Nous avons besoin, sur la page d'accueil, d'une image simple, donnant une impression de cuisine conviviale et sympathique.

Les couleurs doivent être dans la même teinte, sur laquelle nous nous mettrons d'accord durant la réunion de cadrage.

Consignes

L'objectif est de respecter les demandes client. Il vous est possible de planifier deux réunions client : à vous de définir quand et sur quelles modalités. Ces réunions clients peuvent vous permettre de mieux cadrer le produit attendu : elles sont nécessaires. Elles seront limitées à 30 minutes, aussi assurez-vous bien de préparer vos questions.

Deux fichiers de librairies vous faciliteront le travail pour la persistance des données :

- librecipes.php est une librairie qui contient de nombreuses fonctions dont certaines sont détaillées dans la documentation fournie.
- NoSql.php est une librairie technique dont dépend librecipes.php. Vous n'aurez pas besoin d'inclure NoSql.php, mais ce fichier doit être placé dans le même dossier que librecipes.php

Documentation librairie « librecipes »

createRecipe()

Créer une recette et renvoie l'identifiant de la recette créée.

Argument	Type	Description
title	string	Nom (ou titre) de la recette

deleteAllRecipes()

Supprime toutes les recettes et avis associées.

deleteRecipe()

Supprime une recette et les avis associés.

Argument	Type	Description
id	string	Identifiant de la recette

getRecipe()

Revoie une recette en fonction de son identifiant.

Argument	Type	Description
id	string	Identifiant de la recette

hasAlreadyRated()

Renvoie vrai si l'utilisateur spécifié a déjà envoyé un avis sur la recette spécifiée. Sinon, renvoie faux.

Argument	Type	Description
recipeId	string	Identifiant de la recette
userIp	string	Adresse IP de l'utilisateur

rateRecipe()

Ajoute un avis sur la recette.

Argument	Type	Description
recipeId	string	Identifiant de la recette
userIp	string	Adresse IP de l'utilisateur
note	int	Note attribuée
comment	string	Commentaire / Avis

savePreptime()

Sauvegarde le temps de préparation associée à une recette.

Argument	Type	Description
id	string	Identifiant de la recette
time	int	Le temps, en minutes

searchRecipesByIngredient()

Recherche et renvoie une liste de recette en fonction d'ingrédient(s).

Argument	Type	Description
ingredient	string	Ingrédient à rechercher

Projet : Squid Movies

Demande client de base

Ce site web, nommé « Squid Movies », permettra plusieurs choses :

- Une page principale, contenant un titre centré « Squid Movies », ainsi que le texte « Trouvez plein de films et séries sur cet incroyable site qui n'est en fait qu'un prototype réalisé par les bg de BTS SNIR ! »
- Depuis n'importe quelle page, on doit pouvoir rechercher un ou plusieurs films ou séries par titre ou par genre, dans un formulaire discret.
- Lors de la recherche, on arrive sur une page à part affichant la liste des résultats de la recherche. On peut cliquer sur chaque résultat pour afficher le détail de la série ou du film choisi.
- On considère chaque saison d'une série comme une série à part entière pour simplifier le tout.
- Pour chaque film, on enregistre son titre, ses genres, son temps, son « pitch » (la base du scénario), un commentaire personnel de l'auteur, et les plateformes sur lequel il est disponible.
- Pour chaque série, on enregistre son titre (avec le numéro de saison inclus), ses genres, son nombre d'épisodes (pour la saison), son « pitch » (la base du scénario), un commentaire personnel de l'auteur, et les plateformes sur lequel il est disponible.
- Les pages d'administration ne doivent être disponibles que pour l'auteur du site : on souhaite mettre en place une authentification basique, dont vous nous communiquerez les accès.
- Depuis les pages d'administration, on doit pouvoir sauvegarder les détails d'un film ou d'une série, pour en créer, en modifier ou en supprimer.
- Le public doit pouvoir, sur une série ou un film, le commenter et attribuer une note, une seule fois par film ou série. Un filtrage par IP sera mis en place.
- Sur la page d'une série ou d'un film, on doit donc afficher toutes ses informations, suivi de la note moyenne, et la liste des commentaires et notes donnés par le public.

Style

Le style doit être simple, épuré et plutôt sombre. Nous avons besoin, sur la page d'accueil, d'une image simple, donnant une impression de sérieux, de racé, à la manière d'un film sur la prohibition, tout en restant correct politiquement. Les couleurs doivent être dans une même teinte sombre.

Consignes

L'objectif est de respecter les demandes client. Il vous est possible de planifier deux réunions client : à vous de définir quand et sur quelles modalités. Ces réunions clients peuvent vous permettre de mieux cadrer le produit attendu : elles sont nécessaires. Elles seront limitées à 30 minutes, aussi assurez-vous bien de préparer vos questions.

Deux fichiers de librairies vous faciliteront le travail pour la persistance des données :

- libmovies.php est une librairie qui contient de nombreuses fonctions dont certaines sont détaillées dans la documentation fournie.
- NoSql.php est une librairie technique dont dépend libmovies.php. Vous n'aurez pas besoin d'inclure NoSql.php, mais ce fichier doit être placé dans le même dossier que libmovies.php

Documentation librairie « libmovies »

createMovie()

Créer un film et renvoie l'identifiant du film créé.

Argument	Type	Description
title	string	Nom (ou titre) du film

createShow()

Créer une série et renvoie l'identifiant de la série créée.

Argument	Type	Description
title	string	Nom (ou titre) de la série

deleteMovieOrShow()

Supprime une série ou un film, ainsi que les avis associés.

Argument	Type	Description
id	string	Identifiant de la série ou du film

hasAlreadyRated()

Renvoie vrai si l'utilisateur spécifié a déjà envoyé un avis sur la série ou le film spécifié. Sinon, renvoie faux.

Argument	Type	Description
movieId	string	Identifiant de la série ou du film
userIp	string	Adresse IP de l'utilisateur

rateMovieOrShow()

Ajoute un avis sur la série ou le film spécifié.

Argument	Type	Description
movieId	string	Identifiant de la série ou du film
userIp	string	Adresse IP de l'utilisateur
note	int	Note attribuée
comment	string	Commentaire / Avis

saveWhereToWatch()

Sauvegarde la liste des plateformes où cette série ou ce film est disponible.

Argument	Type	Description
id	string	Identifiant de la série ou du film
wtw	array	La liste des plateformes : utilisez les constantes WTW_

searchMoviesOrShowsByGenre()

Recherche et renvoie une liste de séries et films en fonction d'un genre.

Argument	Type	Description
genre	string	Genre à rechercher : utilisez les constantes GENRE_

Projet : Pay respect

Demande client de base

Ce site web, nommé « Pay respect », permettra plusieurs choses :

- Une page principale, contenant un titre centré « Pay respect », ainsi que le texte « Trouvez plein de jeux sur cet incroyable site qui n'est en fait qu'un prototype réalisé par les bg de BTS SNIR ! »
- Depuis n'importe quelle page, on doit pouvoir rechercher un ou plusieurs jeux par titre ou par genre, dans un formulaire discret.
- Sur une page dédiée, une recherche complexe permettra la recherche par titre, genre et plateforme.
- Lors de la recherche, on arrive sur une page à part affichant la liste des résultats de la recherche. On peut cliquer sur chaque résultat pour afficher le détail du jeu choisi.
- Un jeu est constitué d'un titre (son nom), d'une liste de genres (un tableau de GENRE_) et d'une liste de plateformes (un tableau de PLATFORM_).
- Les pages d'administration ne doivent être disponibles que pour l'auteur du site : on souhaite mettre en place une authentification basique, dont vous nous communiquerez les accès.
- Depuis les pages d'administration, on doit pouvoir sauvegarder les détails d'un jeu, pour en créer, en modifier ou en supprimer.
- Le public doit pouvoir, sur un jeu, le commenter et attribuer une note, une seule fois par jeu. Un filtrage par IP sera mis en place.
- Sur la page d'un jeu, on doit donc afficher, dans l'ordre : son titre, ses genres, les plateformes où il est disponible, la note moyenne, et la liste des commentaires et notes donnés par le public.

Style

Le style doit être cohérent et dynamique. Nous avons besoin, sur la page d'accueil, de deux à trois images, donnant une impression de dynamisme tout en présentant différents styles de jeu.

Les couleurs doivent être dans la même teinte, sur laquelle nous nous mettrons d'accord durant la réunion de cadrage.

Consignes

L'objectif est de respecter les demandes client. Il vous est possible de planifier deux réunions client : à vous de définir quand et sur quelles modalités. Ces réunions clients peuvent vous permettre de mieux cadrer le produit attendu : elles sont nécessaires. Elles seront limitées à 30 minutes, aussi assurez-vous bien de préparer vos questions.

Deux fichiers de librairies vous faciliteront le travail pour la persistance des données :

- libgames.php est une librairie qui contient de nombreuses fonctions dont certaines sont détaillées dans la documentation fournie.
- NoSql.php est une librairie technique dont dépend libgames.php. Vous n'aurez pas besoin d'inclure NoSql.php, mais ce fichier doit être placé dans le même dossier que libgames.php

Documentation librairie « libgames »

createGame()

Créer un jeu et renvoie l'identifiant du jeu créé.

Argument	Type	Description
title	string	Nom (ou titre) du jeu

deleteAllGames()

Supprime tous les jeux et avis associées.

getGame()

Revoie un jeu en fonction de son identifiant.

Argument	Type	Description
id	string	Identifiant du jeu

hasAlreadyRated()

Renvoie vrai si l'utilisateur spécifié a déjà envoyé un avis sur le jeu spécifié. Sinon, renvoie faux.

Argument	Type	Description
gameId	string	Identifiant du jeu
userIp	string	Adresse IP de l'utilisateur

rateGame()

Ajoute un avis sur le jeu.

Argument	Type	Description
gameId	string	Identifiant du jeu
userIp	string	Adresse IP de l'utilisateur
note	int	Note attribuée
comment	string	Commentaire / Avis

saveGenres()

Sauvegarde la liste des genres du jeu.

Argument	Type	Description
id	string	Identifiant du jeu
genre	array	La liste des genres : utilisez les constantes GENRE_

searchGames()

Recherche un jeu en fonction de son titre, d'un genre et/ou d'une plateforme.

Argument	Type	Description
title	string	Titre du jeu – null pour ne pas utiliser ce critère
genre	string	Genre (utilisez une constante GENRE_) – null pour ne pas utiliser ce critère
platform	string	Plateforme (utilisez une constante PLATFORM_) – null pour ne pas utiliser ce critère