



CHALLENGE CENTRALE LYON

Fiche explicative du site de l'écran géant

VP Digit Web :
Ewen ORSAL (Guyrek) 2022

13 août 2021

Table des matières

1	Installer le projet gitlab	2
2	Lancer le site en local	3
3	Fonctionnement du site	5
3.1	Fonctionnement général	5
3.2	Format de la base de données	5
3.3	Le code PHP	6
4	Modifier le site	8
4.1	Ajouter/Modifier un visuel	8
4.2	Ajouter/Modifier un élément dans une page	8
4.3	Mettre ses changements en ligne	8

1 Installer le projet gitlab

Tout d'abord il faut installer XAMPP pour php 7.1, téléchargeable [ici](#) . C'est le logiciel qui va permettre de lancer le site dans le serveur local à votre ordinateur.

Ensuite aller sur le [gitlab](#) du projet, cliquer sur **Clone** en bleu et copier l'URL HTTPS :

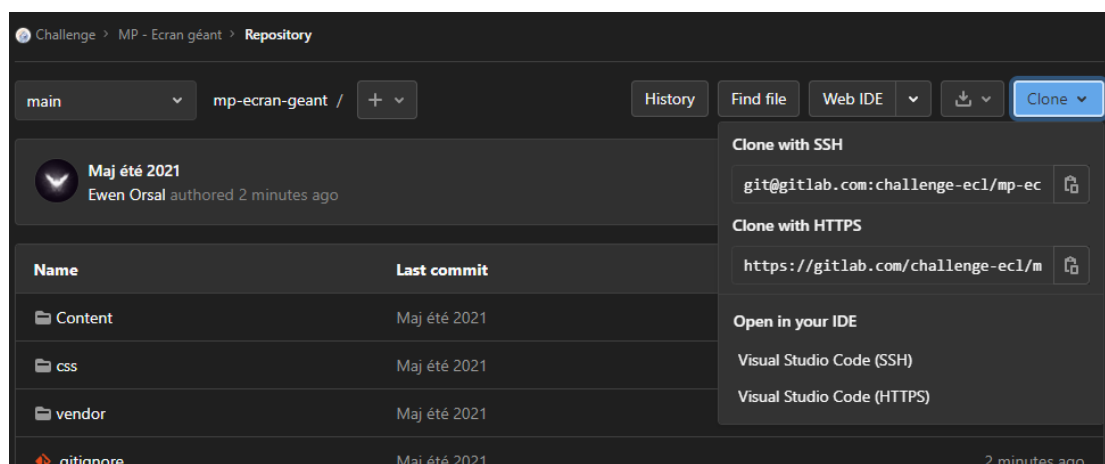


FIGURE 1 – Cloner le repository gitlab

Enfin on va créer le dossier du projet. Pour cela on ouvre l'invite de commande (Windows+R et rentrer **cmd**). Il faut maintenant naviguer jusqu'au dossier **xampp/htdocs** à l'aide de la commande `cd Chemin_du_dossier`.

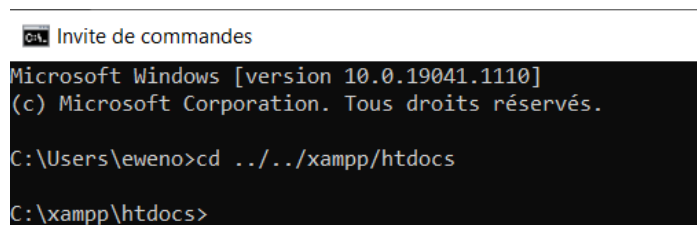


FIGURE 2 – Invite de commande

On va alors rentrer la commande `git clone URL_HTTPS Ecran` en remplaçant `URL_HTTPS` par l'URL copiée précédemment. Cela va créer un dossier **Ecran** qui sera relié au projet gitlab. Vous pourrez alors mettre en ligne vos modifications en un rien de temps.

2 Lancer le site en local

Rechercher **XAMPP Control** et le lancer, vous devriez être face à cette interface :

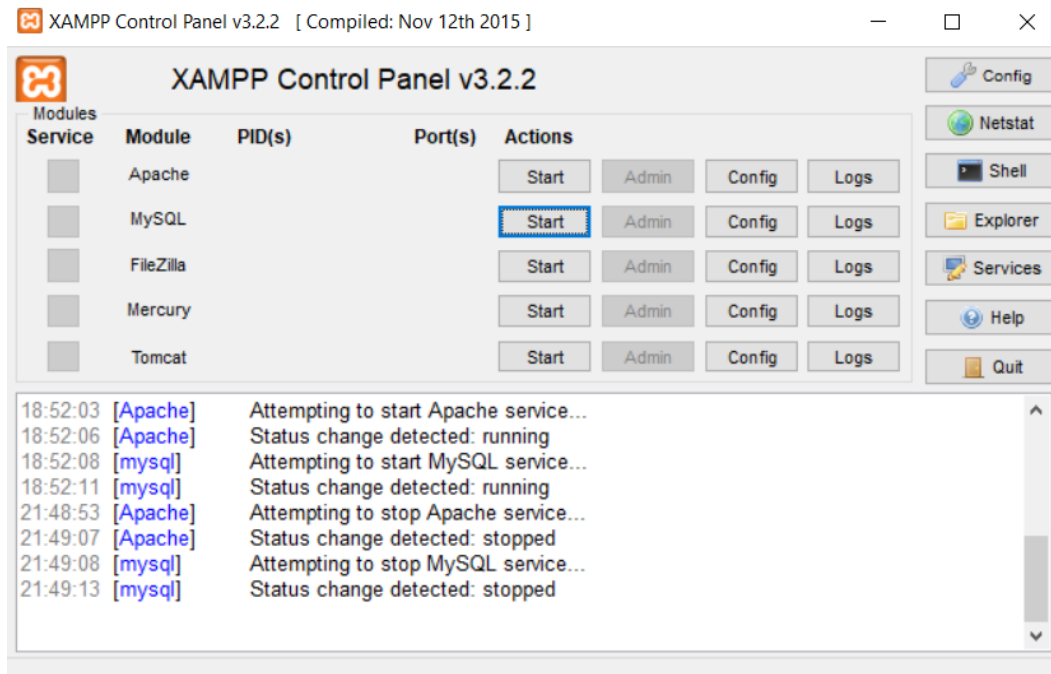


FIGURE 3 – Interface de XAMPP

Les deux modules qui seront utilisés pour le site seront **Apache** pour visualiser le site, et **MySQL** pour communiquer avec la base de données. Appuyer sur **Start** sur ces deux modules afin de les lancer. Une fois cela fait les cases **Admin**, jusque là grisées, deviennent cliquables. Appuyer sur ces deux boutons pour visualiser le site dans votre navigateur. Deux pages vont s'ouvrir :

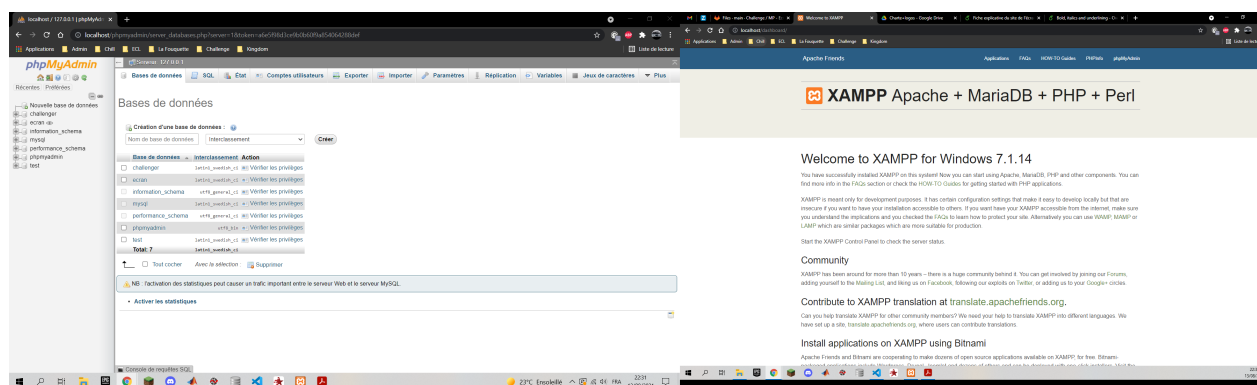


FIGURE 4 – Pages MySQL et Dashboard

La page MySQL correspond à la base de données du serveur local et la page dashboard sera la future page ou vous pourrez visualiser le site.

D'abord il faut créer la base de données sur le localhost. Pour cela, se rendre sur la page MySQL. Cliquer sur **Nouvelle base de données**, rentrer le nom **Ecran** puis appuyer sur **Créer**.

Une fois cela fait, on va la remplir automatiquement à l'aide du fichier **ecran.sql** dans le dossier du projet. Pour cela, se rendre dans l'onglet **Importer**, sélectionner le fichier et cliquer sur **Exécuter**. L'opération prendra alors un peu de temps car la base de données est assez conséquente.

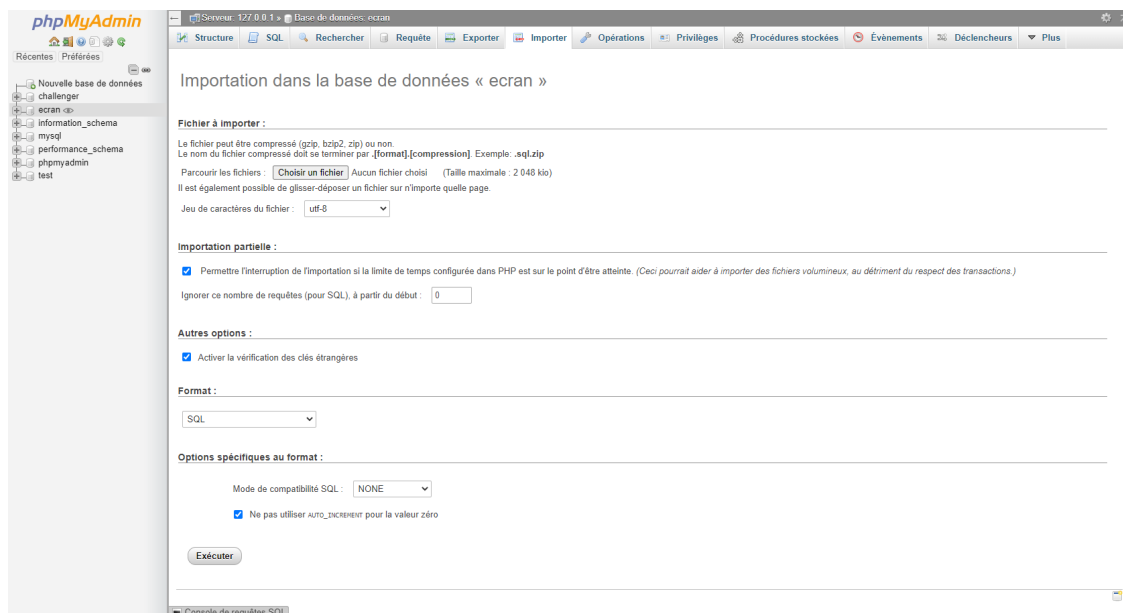


FIGURE 5 – Import de la base de données

Lorsque l'import est terminé, se rendre sur la page **Dashboard**. Afin de visualiser le site en ligne, il suffit de remplacer **dashboard** par **Ecran** dans la barre d'URL en haut.

3 Fonctionnement du site

3.1 Fonctionnement général

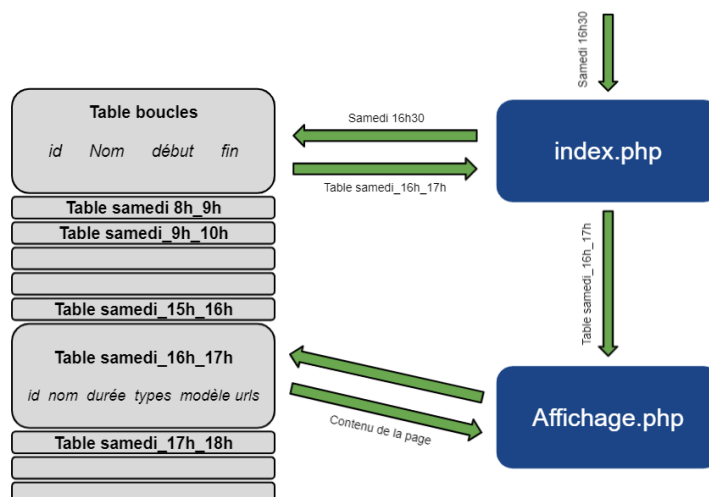


FIGURE 6 – Fonctionnement du site

Le site s'organise en boucles par tranches horaires. En effet, une table contient le contenu à afficher le samedi entre 6h et 7h, une autre le dimanche entre 13h et 14h etc... Le fichier **index.php** détermine l'heure actuelle, et effectue une requête auprès de la table **boucles** en fonction de l'heure. La table renvoie alors la table correspondant à la tranche horaire, qui est transmise au fichier **affichage.php**. Comme son nom l'indique, c'est ce fichier qui est responsable de l'affichage des pages. Il exécute une requête auprès de la table renseignée par **index.php** afin d'en obtenir le texte, les images etc...

3.2 Format de la base de données


















+ Options								
<div><div></div><div></div><div></div></div>				id	nom	heure_debut	heure_fin	nomboucle
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	1	samedi_1	2021-03-26 05:00:00	2022-03-26 06:00:00	samedi_5h_6h
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2	samedi_2	2022-03-26 06:00:00	2022-03-26 07:00:00	samedi_6h_7h
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	3	samedi_3	2022-03-26 07:00:00	2022-03-26 08:00:00	samedi_7h_8h
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	4	samedi_4	2022-03-26 08:00:00	2022-03-26 09:00:00	samedi_8h_9h

FIGURE 7 – Format de la table **boucles**


+ Options


				id	nom	duree	types	modele	urls	titre
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	1	page1	01:00:30	image, video	1	wave.png, rocketLeague.mp4	
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2	streamtest	01:00:20	image, stream, image	2	building1.jpg, https://www.youtube.com/embed/5qap5...	




☐ Tout cocher

Avec la sélection :

 Éditer

 Copier

 Supprimer


 Exporter

FIGURE 8 – Format des tables de contenu

Les tables du type **samedi_16h_17h** comportent plusieurs champs :

- **id** est l'id auto-incrémenté de la page affichée
- **nom** est un identifiant pour nous aider à nous repérer
- **duree** est la durée d'affichage de la page avant de passer à la suivante
- **types** est la liste des types de contenu affiché sur la page
- **modele** correspond à l'agencement de la page choisi
- **urls** est le chemin d'accès du contenu à afficher. L'ordre doit être le même que pour **types**.

3.3 Le code PHP

```
//on récupère les infos de la page que l'on veut générer
$reponse = $bdd->query('SELECT id, nom, TIME_TO_SEC(duree) as secondes, types, modele, urls, titre
FROM '.$table.' WHERE id = '.$num);
$donnees = $reponse->fetch();
```

FIGURE 9 – Code de requête vers la DB

La variable **donnees** contient tous les données nécessaires à l'affichage de la page. C'est un array dont les clés sont les champs de la table, et dont les valeurs sont celles correspondantes dans la table.

```
<!-- Lien vers la page css de style-->
<link rel="stylesheet" href="css/default_page.css" />
<link rel="stylesheet" href="css/visuel".$donnees['modele'].".css" ?> />
```

FIGURE 10 – Code de lien vers les fichiers css

La première stylesheet contient les informations de la page par défaut. Cela comprend l'image de fond, le logo, l'horloge ainsi que la bande défilante en bas. La variable **donnees['modele']** renvoie vers le numéro du visuel que l'on désire pour l'agencement de la page.

```
<!-- Corps de la page -->
<div id="content">
    <?php for($i=0; $i<count($types); $i++) {
        if ($types[$i] == 'image') { ?> <!--IMAGE-->
            <img id = <?php echo "Image".$currentImage ?> src = <?php echo "Content/Images/".$urls[$i] ?> />
            <?php $currentImage++?>

        <?php } elseif ($types[$i] == 'video') { ?> <!--VIDEO-->
            <video id = "vid" autoplay controls muted poster = "css/Images/NLambiance.png">
            <source src = <?php echo "Content/Videos/".$urls[$i] ?> ></video>

        <?php } elseif ($types[$i] == 'stream') { ?> <!--STREAM-->
            <iframe id = 'stream' src="<?php echo $urls[$i] ?>" allow="autoplay" allowfullscreen></iframe>
```

FIGURE 11 – Code d'analyse du contenu

Sur la figure 11, le code parcourt tous les éléments à afficher, détecte leur type et crée un objet html avec les paramètres correspondants à ceux donnés par la base de données.

```
<!-- Pied de page -->
<div id="footer" class="bande"> <!--Bande défilante-->
    <marquee loop="-1" truespeed SCROLLDELAY="30">
        <?php $url_2 = './info.txt';
            $raw_2 = file_get_contents($url_2,FILE_USE_INCLUDE_PATH);
            echo $raw_2; ?>
    </marquee>
</div>
<!-- Fin du pied de page -->
```

FIGURE 12 – Code de la bande défilante

Sur la figure 12, le code récupère le contenu du fichier **info.txt** et les affiche dans le bandeau défilant. A terme, il serait bien que les messages soient envoyés par les participants via l'application, et lu par le code depuis le serveur.

```
<!-- Automatisation du bouclage -->
<?php
    $num = $num + 1 ;// On passe au numéro de page suivante dans la boucle à afficher-->
    $t = $bdd->query("SELECT COUNT(*) as taille_table FROM {$table} ");
    $taille = $t -> fetch(); //On récupère la taille de la boucle en cours (ie le nombre de pages qu'elle contient)

    if ($num <= $taille['taille_table']) //Cas ou il reste des pages dans la boucle
    {
        $url = 'affichage.php?id='.$num.'&tablename='.$table;
    }
    else //Cas ou la boucle est finie et on retourne à l'index
    {
        $url = 'index.php';
    }
?>
```

FIGURE 13 – Code du bouclage

Après avoir parcouru toutes les pages repertoriées dans la table, le code détermine renvoi vers **index.php** qui redetermine l'heure, renvoie vers la bonne table etc...

4 Modifier le site

De manière générale, si vous programmez dans un langage autre que Python, je vous suggère d'utiliser **Visual Studio Code**, téléchargeable [ici](#). Il a une bonne compatibilité avec git, c'est pourquoi je vais expliquer les démarches sur cet éditeur.

4.1 Ajouter/Modifier un visuel

Un visuel correspond à la manière d'agencer les éléments sur la page, donc à un fichier **.css**. Pour créer un visuel il faut donc créer un nouveau fichier **visuel.css** en ajoutant le numéro du visuel correspondant. On peut alors faire appelle à ce visuel en mettant son numéro dans le champ **modele** dans la base de données.

4.2 Ajouter/Modifier un élément dans une page

Pour ajouter un élément à une page, il faut tout d'abord placer le contenu dans le dossier **Ecran/Content** puis dans le dossier correspondant au type d'élément. Ensuite, il suffit de rajouter son type et son url aux listes présentes dans la base de données et le tour est joué.

4.3 Mettre ses changements en ligne

Une fois les changements validés et les erreurs corrigées, il faut partager son travail en mettant à jour le repository gitlab. Je vais détailler la démarche sur **VSCoDe**.

D'abord, ouvrez une invite de commande et naviguer vers le dossier **xampp/htdocs/Ecran**. Une fois cela fait rentrez la commande `git status`. Cela vous montrera sur quelle branche du repository vous vous trouvez et si vous êtes à jour avec cette branche. Vérifiez bien que vous êtes sur la branche **MP** et non **main**. Si vous êtes sur **main** rentrez la commande `git checkout MP` pour vous déplacer sur la branche **MP**. Une fois cela fait, se rendre sur l'onglet **Source Control** dans **VSCoDe**. Vous pouvez voir tous les changements que vous avez effectué par rapport à la version actuellement sur le repository git.

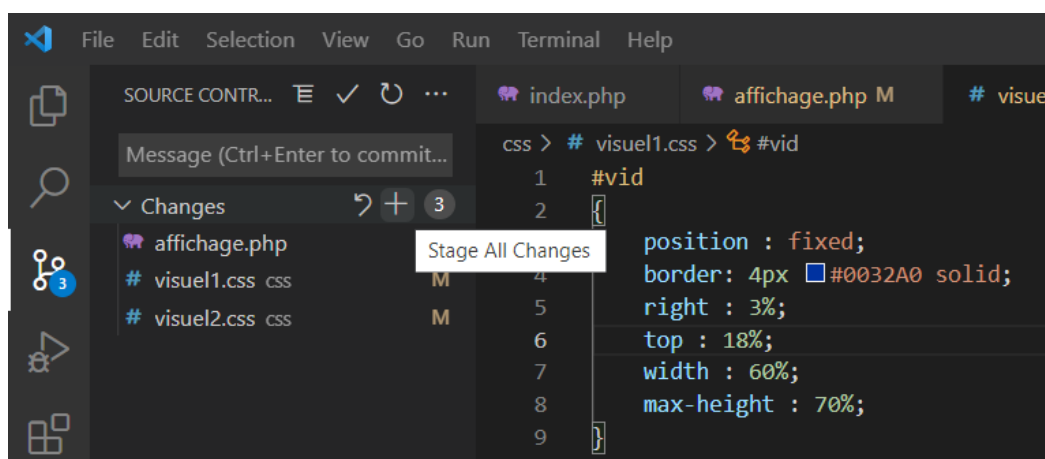


FIGURE 14 – Onglet Source Control de VSCoDe

En passant la souris près de **Changes** un + apparait. En cliquant dessus vous "validez" vos changements et les préparez à être commit. Ensuite, dans la barre de texte au dessus rentrez une description en quelques mots des changements que vous avez apporté (ex *Added : visuel3.css*) et cliquer sur l'icone **Valider** : vos changements viennent d'être commit.

Pour terminer la procédure, retourner sur l'invite de commande, et exécuter la commande *git push*. Voilà, vous avez mis à jour la branche **MP**. Le VP Digit Web sera chargera alors de fusionner les branches **main** et **MP** après s'être assuré qu'aucune erreur n'a été laissée.