# Compte rendu du projet HTML/CSS + DB

Table des matières

[Compte rendu du projet HTML/CSS + DB 1](#_Toc153033508)

[I. Membres de l’équipe 1](#_Toc153033509)

[II. Réponses aux question du TP 2](#_Toc153033510)

[Question 1 2](#_Toc153033511)

[Question 2 et 3 2](#_Toc153033512)

[Question 4 3](#_Toc153033513)

[Question 5 3](#_Toc153033514)

[Question 6 4](#_Toc153033515)

[III. Outils et technologies utilisés 4](#_Toc153033516)

[IV. Read me 4](#_Toc153033517)

[V. Fonctionnement du site 5](#_Toc153033518)

[a. Description du fichier index.php 5](#_Toc153033519)

[b. Description du fichier functions.php 5](#_Toc153033520)

[c. Le fichier config.php 5](#_Toc153033521)

[d. Le fichier style.css 6](#_Toc153033522)

[VI. Choix des outils 6](#_Toc153033523)

[a. Éditeur de texte 6](#_Toc153033524)

[b. Langages des programmation 6](#_Toc153033525)

[c. Serveur WAMP 6](#_Toc153033526)

[d. Choix des navigateurs web et github 7](#_Toc153033527)

[e. Docker 7](#_Toc153033528)

# Membres de l’équipe

Léo-Paul FALQ

Ivan-Corneille MAGAGI

# Réponses aux question du TP

## Question 1

L’image décrit le processus d'interaction entre un navigateur web, un serveur web utilisant PHP, et une base de données MySQL. Voici les différentes étapes impliquées :

1. Requête de la page : tout commence lorsque le navigateur web d'un utilisateur fait une requête pour une page en envoyant cette demande au serveur web via Internet.
2. Traitement par le Serveur Web : le serveur web reçoit la requête et détermine qu'un script PHP doit être exécuté pour répondre à cette requête.
3. Exécution : le serveur exécute le script PHP correspondant à la page demandée. PHP est souvent utilisé pour générer du contenu dynamique.
4. Interaction avec MySQL : Le script PHP a besoin de récupérer des informations stockées dans une base de données MySQL pour construire la page web. PHP envoie une requête à MySQL.
5. Réponse de MySQL : MySQL traite la requête du script PHP, accède aux données demandées, et envoie ces informations au script PHP.
6. Génération du contenu HTML : PHP intègre les données récupérées dans le contenu HTML de la page web. Ce processus crée le contenu dynamique spécifique à la requête de l'utilisateur ou à la session en cours.
7. Envoi du HTML au navigateur : Une fois que le script PHP a terminé l'exécution et a généré la sortie HTML complète, le serveur web envoie cette sortie au navigateur web de l'utilisateur via Internet.
8. Affichage de la Page Web : Enfin, le navigateur web reçoit le HTML et l'affiche, permettant à l'utilisateur de voir la page web complète avec le contenu dynamique généré.

## Question 2 et 3

1. Navigateur web : (google chrome, firefox, safari…) envoie une demande de page au serveur web et affiche la page web qui lui est renvoyée.
2. Serveur Web : reçoit la demande du navigateur et fait appel à un script PHP pour générer la page web dynamique. Il sert également de pont entre le navigateur et la base de données. Exemples de serveurs web : Apache HTTP Server (Apache) et Nginx.
3. Scripts php: résident sur le serveur web et sont exécutés en réponse à une demande de page. Ils sont responsables de la création de contenu dynamique, souvent en interrogeant une base de données et en intégrant les résultats dans le HTML qui sera envoyé au navigateur.
4. Base de données MySQL :MySQL est un système de gestion de base de données qui stocke les informations nécessaires au site web. Les scripts PHP font des requêtes à MySQL pour récupérer, mettre à jour, insérer ou supprimer des données.
5. Internet : est le réseau qui connecte le navigateur web de l'utilisateur avec le serveur web, permettant le transfert de demandes et de réponses entre eux.

Il existe plusieurs alternatives au script PHP pour générer des pages web dynamiques, chacune avec ses propres langages de programmation, cadres (frameworks) et bibliothèques. Quelques-unes des plus courantes sont :Node.js, python, Django, Flask, Pyrmid, Ruby, Java, ASP.NET, Goang..

## Question 4

Une base de données est nécessaire pour un site web lorsqu’on a besoin de stocker, organiser, et gérer des données de manière dynamique (achat, vente, publication...). Pour des sites qui ont des contenus fréquemment mis à jour, comme les articles de blog, les actualités, ou les listes de produits, une base de données permet de gérer ces contenus sans avoir à modifier le code HTML manuellement.

On aura aussi besoin d’une base de données lorsqu'un site offre des fonctionnalités telles que l'inscription des utilisateurs, la connexion, les profils personnalisés, ou le stockage des préférences.

## Question 5

Pour établir une connexion à une base de données MySQL en PHP, on utilise généralement les extensions MySQLi ("i" pour "improved") ou PDO (PHP Data Objects). Voici un exemple de code utilisant MySQLi pour se connecter à une base de données MySQL.

Exemple:

function connectToDatabase($servername, $username, $password, $dbname) {

// Tente de se connecter à la base de données

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// Vérifie la connexion

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

return $conn;

}

Pour utiliser la fonction:

$servername = "localhost";

$username = "votre\_nom\_utilisateur";

$password = "votre\_mot\_de\_passe";

$dbname = "nom\_de\_votre\_base\_de\_donnees";

// Utilisation de la fonction pour établir la connexion

$conn = connectToDatabase($servername, $username, $password, $dbname);

### Question 6

SELECT column1, column2, column3

FROM tableName

WHERE conditionColumn = 'value';

# Outils et technologies utilisés

Dans notre projet, nous avons utilisés les outils et technologies suivants :

* Editeur de code : Visual Studio Code
* Langages de programmation : Html, CSS, PHP et SQL.
* Serveur de base de données : Wamp.
* Navigateurs web de test : Microsoft Edge et Google Chrome.
* Outils de gestion de versions : Github.
* Conteneur : Docker.

# Read me

Pour exécuter le projet en local, il faut:

* Télécharger le serveur web local Wamp (lien disponible [ici](https://www.wampserver.com/en/download-wampserver-64bits/)).
* Installer Wamp.
* Lancer le serveur Wamp
* Copier/cloner le dépôt (dossier) de notre projet (sous le nom “html-css-bdd”)dans le dossier “C:\wamp64\www”.

Pour ce faire, réaliser les manipulations suivantes:

* Ouvrez l’invite de commandes ou PowerShell en appuyant sur la touche Windows + R, en tapant "cmd" ou "powershell" dans la boîte de dialogue, puis en appuyant sur Enter.
* Exécuter à tour de rôles les lignes de commandes suivantes:
* Pour s’assurer d’être dans le bon dic : “C:”
* Pour accéder dans le bon répertoire: “cd C:\wamp64\www”
* Pour cloner le projet : “git clone <https://github.com/falqlp/html-css-bdd.git>”
* Pour accéder au répertoire cloné : “cd html-css-bdd”
* Si le serveur de base de données MariaDB n’est pas installé sur votre local, vous pouvez installer docker est exécuter le fichier docker-compose avec la ligne de commande “docker compose up --build”. Il faut pour cela que le port 3306 soit disponible.
* Exécuter le script de création de la base de données disponible dans le fichier “objects.txt“
* Lancer dans le navigateur de votre choix le lien “localhost/html-css-bdd“

# Fonctionnement du site

Nous travaillons avec une base de données des ressources humaines.

## Description du fichier index.php

Le fichier index.php est la page d'accueil du site web.Le script PHP est utilisé pour intégrer des fonctionnalités dynamiques. Il commence par inclure le fichier functions.php, qui fournit des fonctions utilisées pour interagir avec la base de données ou effectuer d'autres opérations communes. Le fichier index.php inclut également un lien vers un fichier CSS pour le style et utilise des polices de Google Fonts.

Gestion des onglets : le fichier gère les onglets de la page. Il vérifie si un onglet est sélectionné via les paramètres GET de l'URL. Ensuite, il récupère les informations des départements (de la base de données) et affiche les noms des départements comme onglets.

Affichage dynamique : pour chaque département, il crée un lien, en mettant en évidence l'onglet sélectionné. Cela suggère une navigation basée sur les onglets pour afficher les informations spécifiques à chaque département.

## Description du fichier functions.php

Il contient un ensemble de fonctions PHP utilisées pour interagir avec la base de données. Les différentes fonctions réalisent en réalité des requêtes sur notre base de données.Nous leur avons donné les noms explicites pour faciliter la compréhension du code.

## Le fichier config.php

Il contient la configuration de la connexion à la base de données pour notre site. Cette configuration est un élément clé du site, permettant toutes les interactions de base de données nécessaires pour les fonctionnalités du site. En voici les éléments clés:

* création de la Connexion à la Base de Données : le fichier utilise mysqli pour établir une connexion à la base de données. mysqli est une extension PHP qui fournit une manière de se connecter à des bases de données MySQL.
* paramètres de connexion : la connexion est établie avec la base de données sur l'hôte localhost en utilisant le nom d'utilisateur “user” et le mot de passe “test”.
* gestion des erreurs de connexion : le script inclut une vérification pour s'assurer que la connexion à la base de données est établie. S'il y a un échec de connexion, un message d'erreur est affiché.

## Le fichier style.css

Ce dernier est appelé dans le fichier index.php. Il permet de faire une mise en forme de l’affichage de notre page web.

# Choix des outils

De manière générale, nous avons choisi les outils avec lesquels nous sommes déjà familiers. Cela nous devait nous permettre de réaliser un gain de temps et de devenir productifs assez rapidement.

## Éditeur de texte

Nous avons utilisé Visual Studio Code par habitude et donc par facilité d’usage, facilitant ainsi une meilleure productivité. Les différentes extensions de l’outil nous ont permis de faire une bonne gestion de version.

Cet outil peut être remplacé par d’autres éditeurs de codes.

## Langages des programmation

HTML/CSS : utilisés pour la structure de base et le style de la page. C'est standard pour tout développement web. Google Fonts fournit des polices personnalisées pour améliorer l'aspect visuel du site.

PHP : permet la création de contenu web dynamique et pour le scripting côté serveur. Il est utilisé pour la gestion des onglets et l'interaction avec la base de données, notamment pour récupérer et traiter les données. C'est une approche standard pour les applications web dynamiques.

SQL: utilisé pour interroger la base de données. Les requêtes SQL sont essentielles pour récupérer, insérer, mettre à jour et supprimer des données dans la base de données.

## Serveur WAMP

Wamp (Windows, Apache, MySQL, PHP/Perl/Python) a été choisi parce qu’il offre une installation simple et rapide, intégrant tous les composants nécessaires pour mettre en place un serveur web local sous Windows avec juste quelques clics. Il permet de développer et tester des applications web en PHP et d'utiliser MySQL pour la gestion des bases de données, le tout sur une machine locale, sans nécessiter une connexion internet ou un hébergement externe. WAMP est un logiciel gratuit. En travaillant localement,nous pouvons développer et tester en toute sécurité sans risquer d'affecter un environnement de serveur réel ou de divulguer des informations sensibles.

Nous avons un contrôle total sur l’environnement serveur avec une possibilité de personnalisation. WAMP bénéficie d'une grande communauté d'utilisateurs. Il y a donc beaucoup de documentation, de forums et de ressources d'apprentissage disponibles

Enfin, les projets développés sur WAMP peuvent être facilement transférés vers la plupart des hébergements web standards, car ils utilisent généralement des configurations Apache, MySQL et PHP similaires.

## Choix des navigateurs web et github

Google chrome et Microsoft edge ont servi pour tester nos développements parce que c’est l’un des navigateurs les plus utilisés dans notre entourage. Edge est bien souvent configuré comme navigateur par défaut sur des ordinateurs Windows. Tout de même, d’autres navigateurs auraient bien pu servir comme test.

Github est une ressource que nous maitrisons relativement bien et qui nous a permis de réaliser un travail collaboratif à distance. Il est gratuit et une documentation est disponible pour une prise en main efficace.

## Docker

Docker nous a permis d’utiliser la base de données sans devoir installer localement le serveur MySql. Il est gratuit et nous a permis de prendre en main les environnements de DevOps.