Tout d’abord, on va commencer par la création du sous-dossier “dev web” pour notre projet sous le dossier \wamp\www\ pour la création de tous les pages php qu’on va utiliser.

**- “index.php” : Affichage de données.**

Cette page contient le Front-end de ce projet. Elle est basée sur HTML, CSS et BOOTSTARP tags .

Elle affiche tous les données qu’ils sont ajoutées et modifiées par l’utilisateur dans un tableau de 2 colonnes avec le tag “<table>”.

Deuxièmement, nous devons construire notre base de données. Et pour l’accès au SGBD on doit accéder à PhpMyAdmin . Puis on Crée une base de données et on la nomme “crud” qu’il va nous aider pour le stockage des données traitées. Et on lui ajoute un table de 2 colonnes (‘id’,’task’).

Nous devons connecter notre projet à la base de données par la page PHP “db.php”.

**-”db.php” rôle : connexion à la base de donnée.**

Nous stockons les informations de connexion dans un objet qu’on appelle ici $db. Cet objet représente notre connexion en soi. Et Pour se connecter, nous instancions la classe prédéfinie mysqli en passant au constructeur les informations suivantes : nom du serveur auquel on doit se connecter (localhost), nom d’utilisateur(root) et mot de passe (‘’/vide) .

Ensuite, nous devons tester la connexion si elle a bien été établie car dans le cas où celle-ci échoue on voudra renvoyer un message d’erreur

Et pour terminer, on doit inclure “db.php” dans “index.php” page avec l’instruction (include"db.php").

Maintenant, On est capable d’afficher les données dans le tableau.

Nous pouvons exécuter tout les données en appelant les fonctions suivantes :

* $sql="SELECT\*FROMtasksLIMIT ".$start.",".$perPage." ";

Stockage de la requête SQL dans la variable *$sql*. La requête signifie : Afficher tous les champs de la table tasks.

* $rows=$db->query($sql);

Exécution de la requête, puis stockage du résultat dans l'objet *$rows*.

* {while($row=$rows->fetch\_assoc()):}

Stocke la première ligne de la table *tasks* (le premier task) dans la variable de type tableau *$row*. La boucle *while* permet d’extraire tout le contenu de la table tasks, ligne après ligne.

**-”add.php”:pour ajouter une tâche.**

Premièrement, On inclus la page PHP “db.php” à “add.php”

Quand un utilisateur clique sur le bouton de soumission, les données des formulaires ont été envoyées à la page "add.php". La page "add.php" se relie à la base de données, et recherche les valeurs du formulaire avec les variables $\_POST.

on insère les données du formulaire dans la table par la requête suivante:

"INSERTINTO tasks (name) VALUES ('$name')"

Puis, la fonction de mysql\_query () exécute l'insertion dans le code, et un nouveau “task” sera ajouté à la table de "tasks".

Finalement, On teste si on a ajouté un champ . Si oui il va nous redirectionner à la page “index.php” sinon il affiche un erreur.

**-”delete.php”: Suppression des données ou tâches.**

Dans le fichier index, on a créé un bouton, « delete ». Ce bouton renvoie vers notre fichier delete.php :

* On a inclus le fichier de la connexion à la base de données « db.php »
* Si on a un id envoyé dans l’url ($\_GET), on le récupère par le lien crée en index.php«href="update.php?id=<?phpecho $row['id'] ; ?> »
* Pour supprimer les données, il faut exécuter la requête de type **DELETE**.

$sql="DELETEFROMtasksWHERE id = '$id'" ;

* Dans la variable $val on a récupéré les données et on a exécuté la requête par la fonction :

$val=$db->query($sql);

* En fin, si $val est vrai on va retourner à la page index.php

**-”update.php”: Mettre à jour les données.**

Dans le fichier index, on a créé un bouton, « Edit ». Ce bouton renvoie les données vers notre fichier Update.php :

* On ajoute un petit formulaire ou on va faire la mise à jour de la task et un bouton qui va nous renvoyer à la page index.php après la mise à jour.
* On a inclus le fichier PHP de connexion db.php
* On initialise une variable $id,
* Si on a bien un id envoyé dans l’url ($\_GET), on récupère le lien crée en index.php
* Pour modifier des données, il faut exécuter une requête de type **UPDATE**.

$sql="SELECT\*FROMtasksWHERE id=$id" ;

* Dans la variable $rows on a récupéré les données et exécuter la requête par la fonction :

$rows=$db->query($sql);

* Dans la variable $row on a récupéré le résultat de $rows sous forme d’un tableau associatif.
* Alors finalement, Si on clique sur update :

-on a mis la tache dans la variable $task

-on a créé la requête update

-on a récupéré les données et exécuté la requête

-et il va nous rediriger vers index.php.