Documentation Symfony :

Ajout d’un thème à la base de données :

D’abord on créer le contrôleur ThemeController

Avec la commande :

*php bin/console make:controller*

On le nomme ThemeController

Ensuite on créer le formulaire

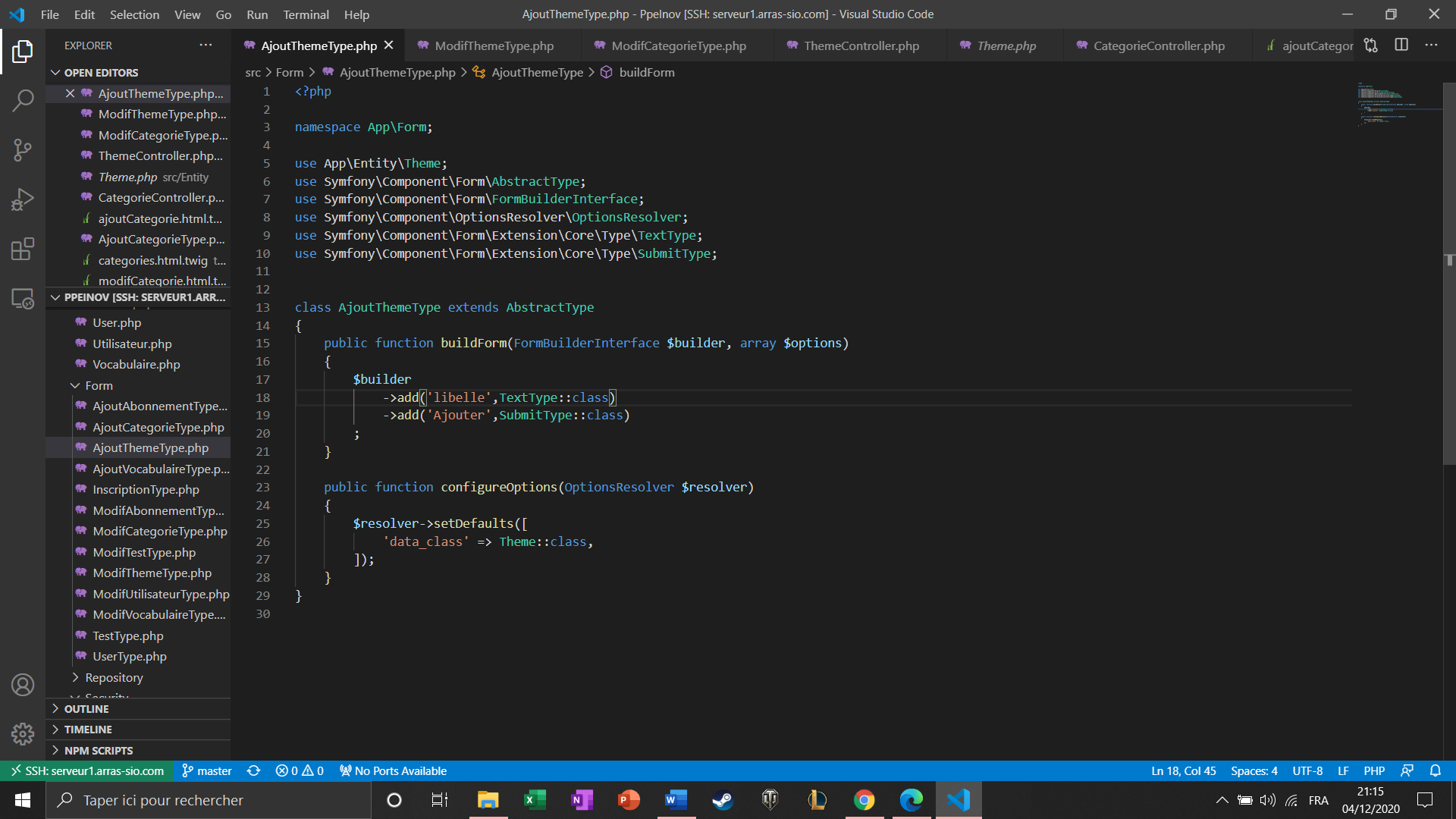
*php bin/console make:form*

On nomme le formulaire AjoutThemeType

Et on lui attribue l’entité Theme créer avec

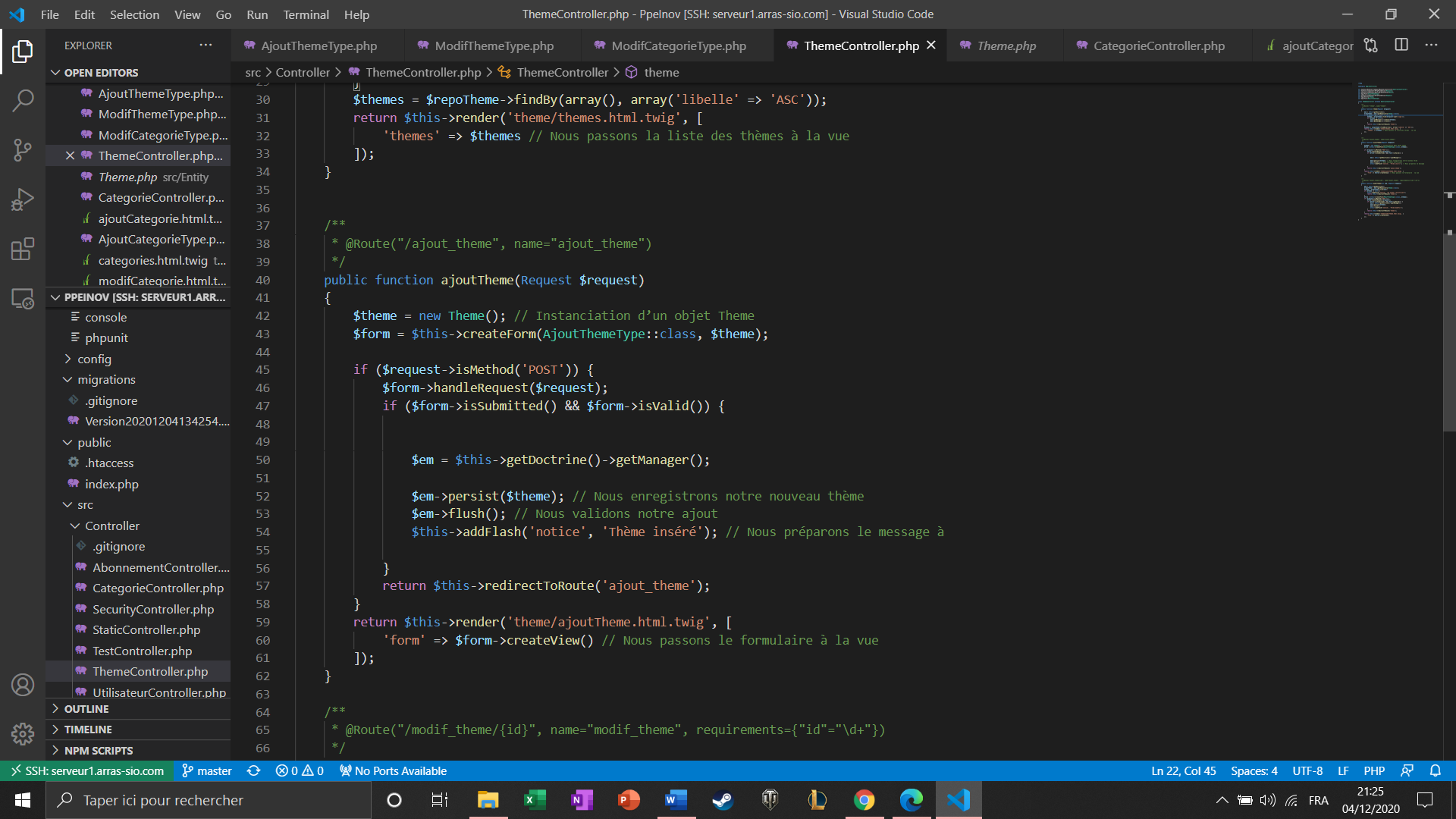
*php bin/console make:entity*

Code du Formulaire Theme



On utilise TextType et SubmitType pour afficher un champ de texte ainsi qu’un bouton de validation. Ne pas oublier de les importer)

Code du contrôleur :



On créer une fonction ajoutTheme dans laquelle on instancie l’entité Theme (ligne 42).

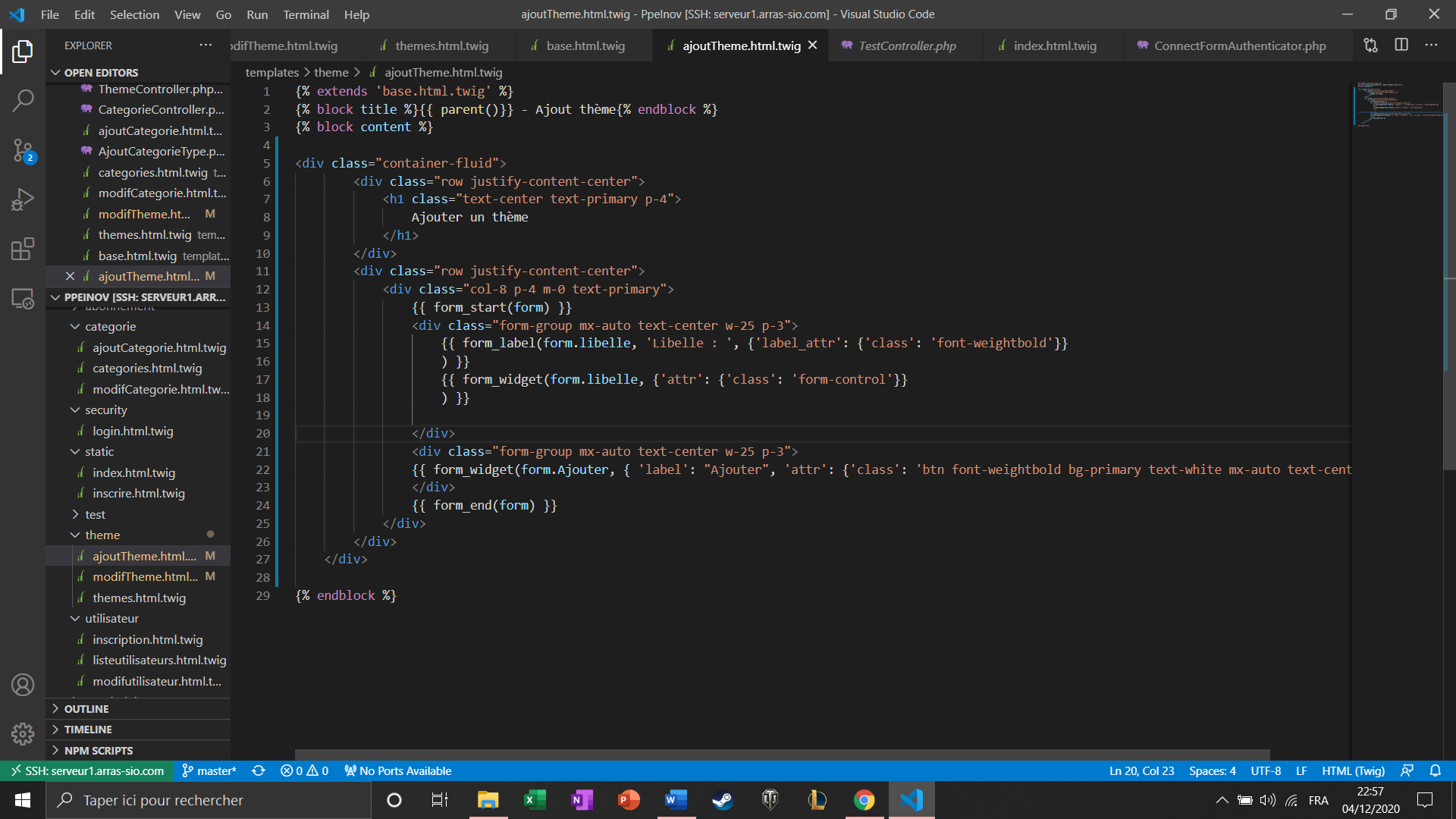
On créer le formulaire d’ajout (ligne 43).

Ajout à la BDD (ligne 45-53).

addFlash permet juste d’envoyer un message.

On n’oublie pas d’importer entité, formulaire et Request

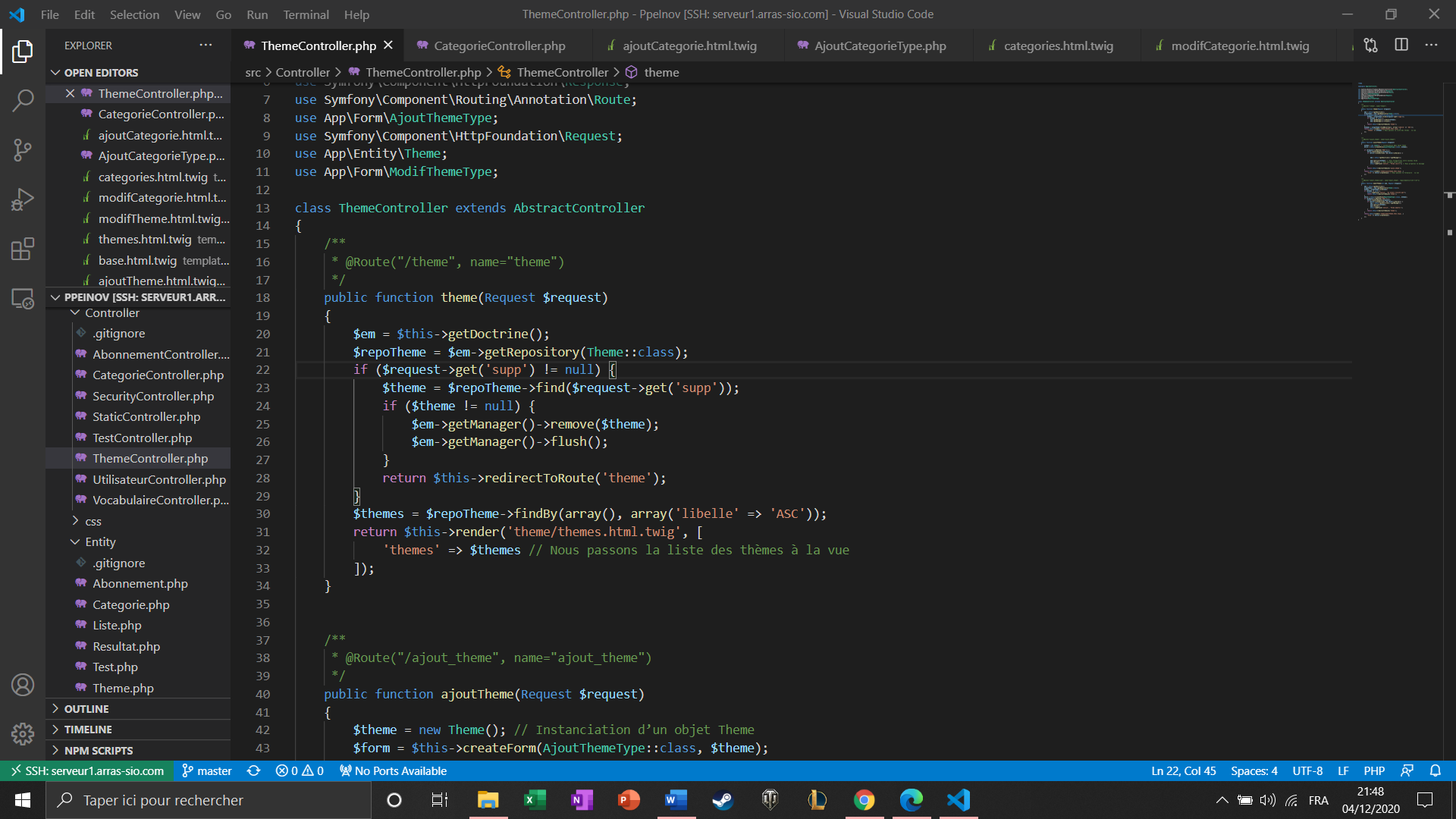
Code de la page ajoutTheme :



On ajoute simplement le formulaire dans la page.

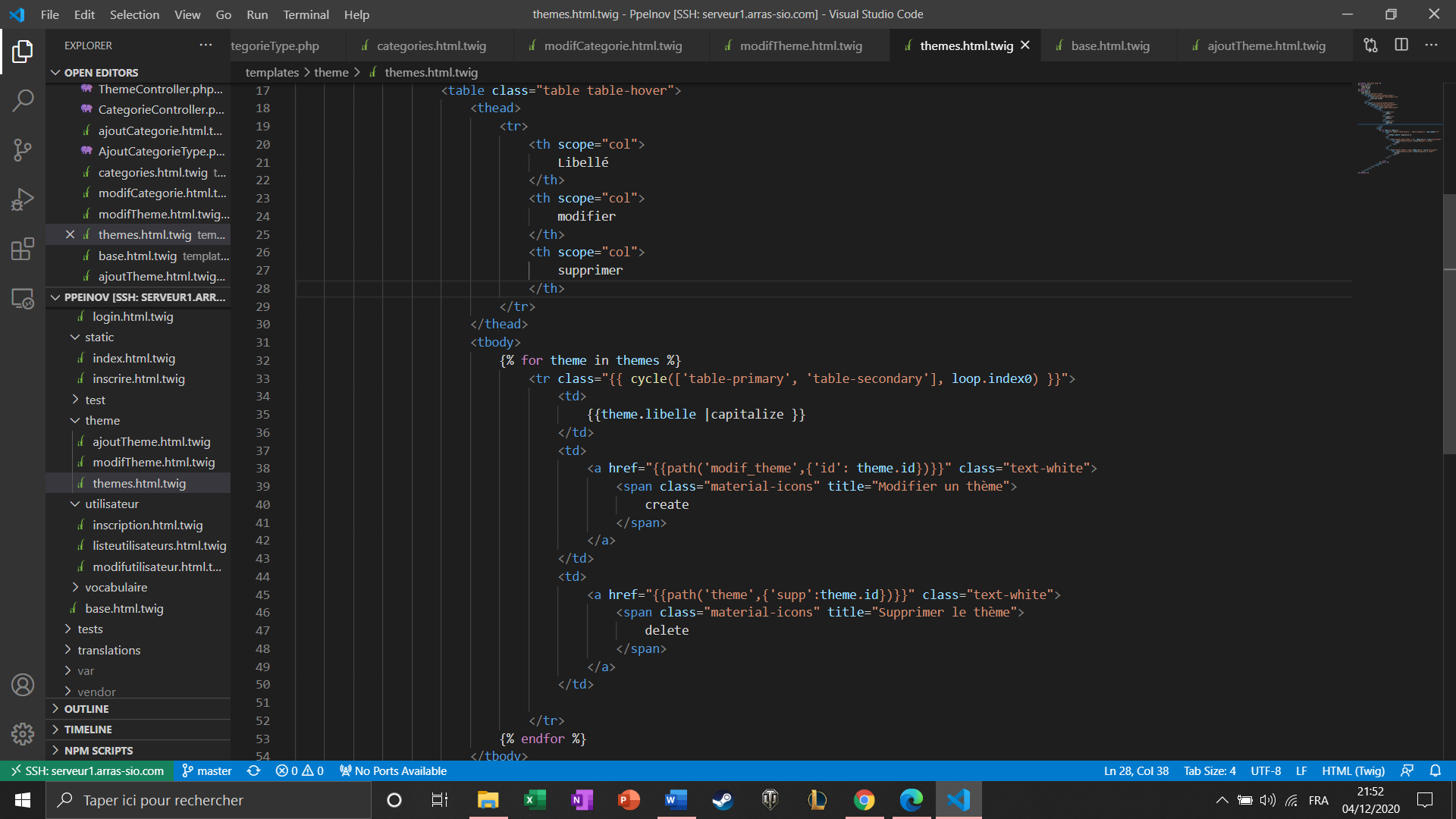
Affichage des Thèmes :

On ajoute une fonction dans notre contrôleur themeController



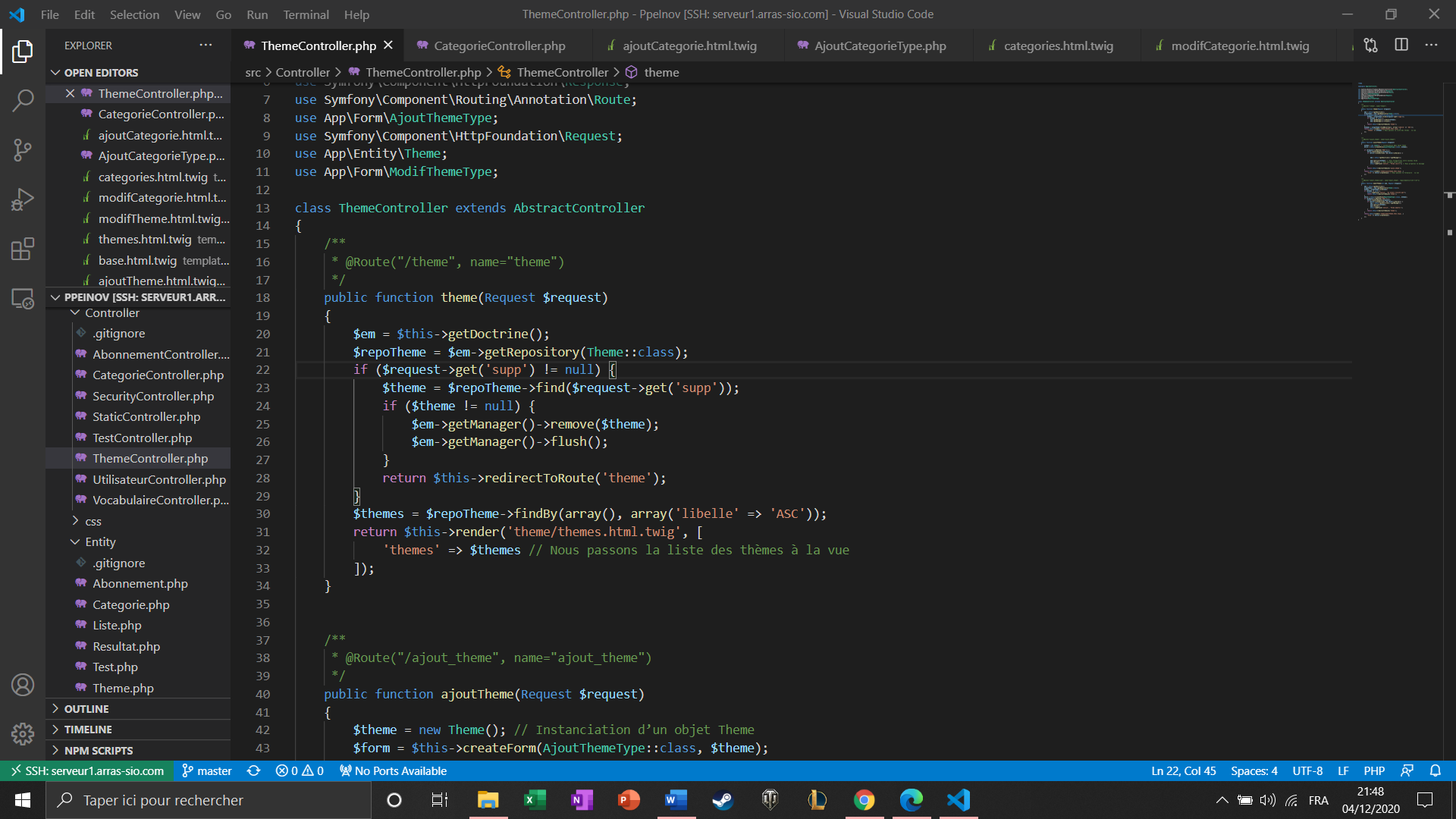
Ce contrôleur contient la suppression (ligne 22) mais on récupère et envoie les données à la vue (ligne 21 et ligne 31-32)

Page d’affichage :



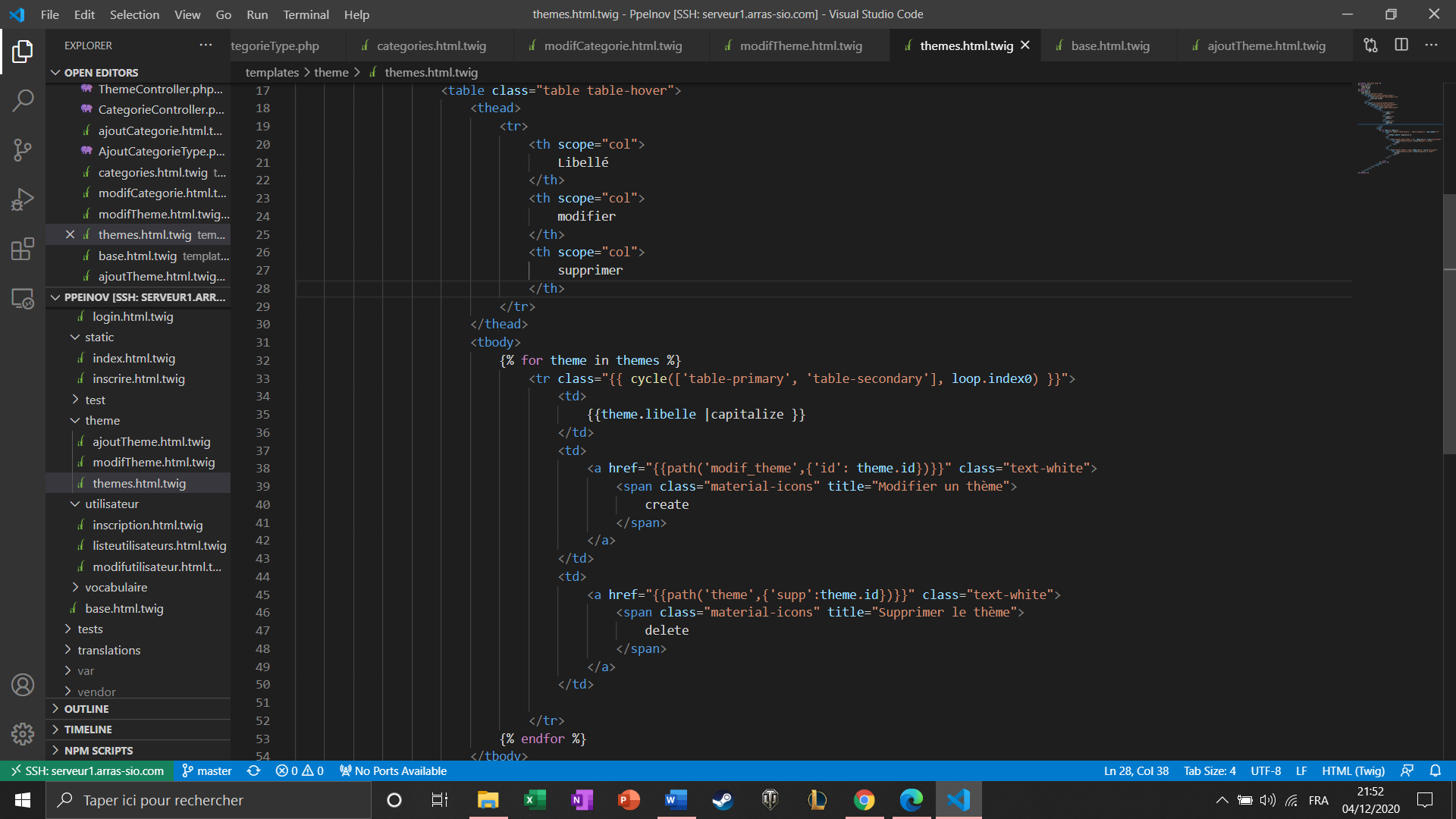
On créer un tableau dans lequel on créer une boucle pour afficher les différentes valeurs (ligne 33)

La suppression d’un thème :



On ajoute la requête vue précédemment qui permet de supprimer le thème via son id (Méthode Get) (ligne 22-26).

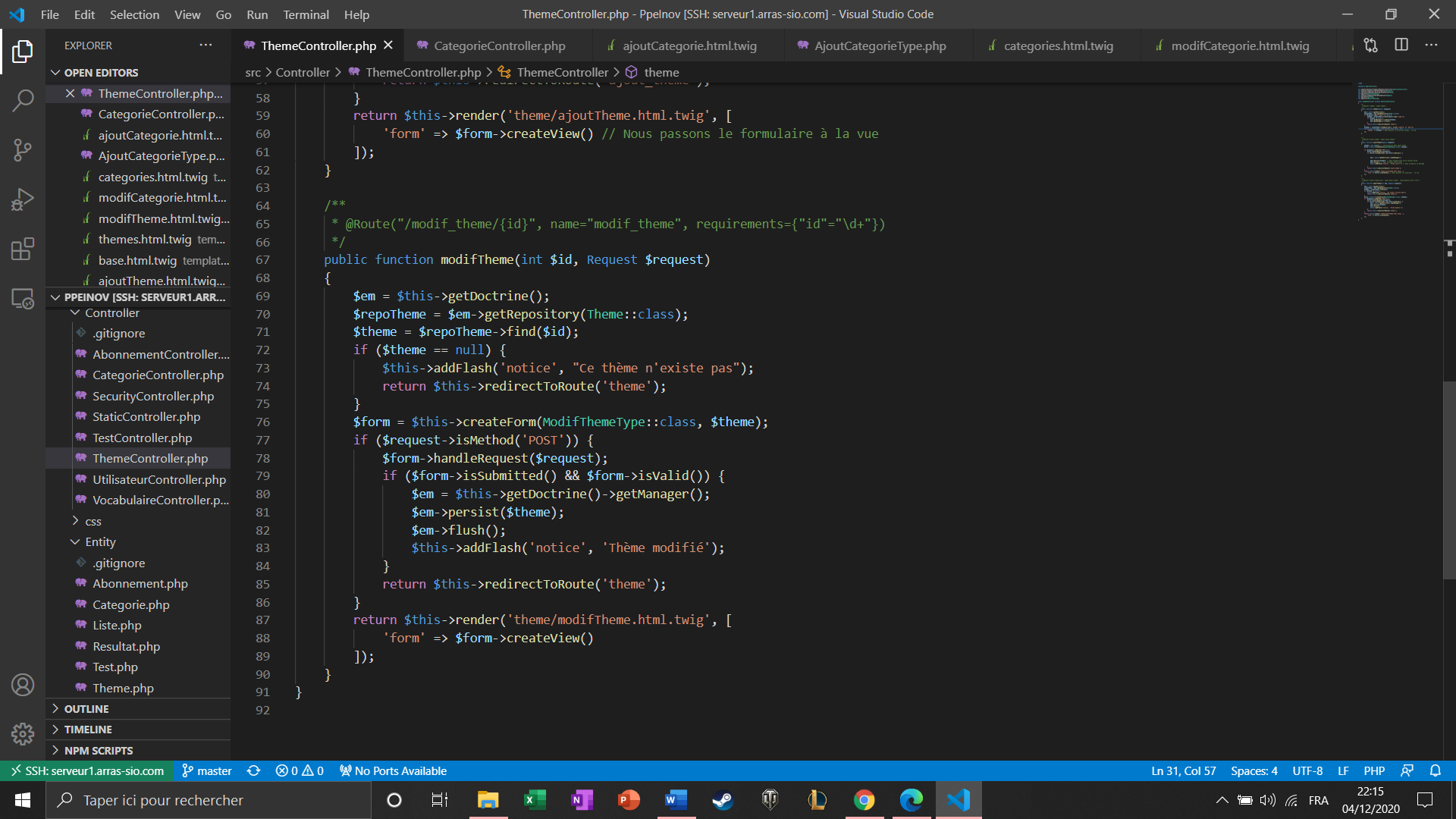
On ajoute un bouton dans notre liste de thème :



Celle est toujours dans la boucle For précédente, elle permet de l’envoie l’id du thème en Get (via l’url)

Modifier un Thème :

Création d’un nouveau formulaire que l’on nommera modifTheme, on ajoute aussi une fonction dans le contrôleur :

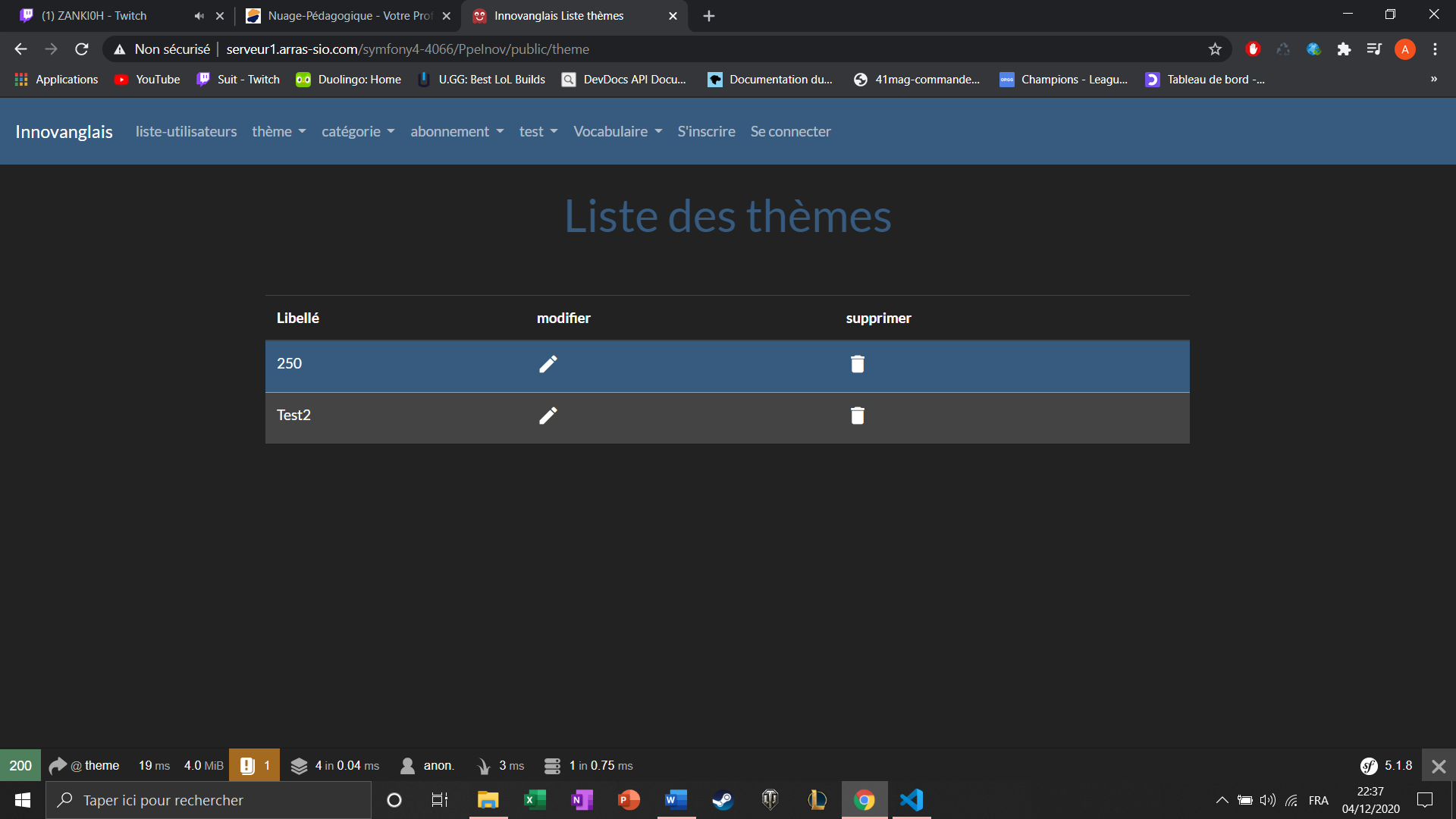


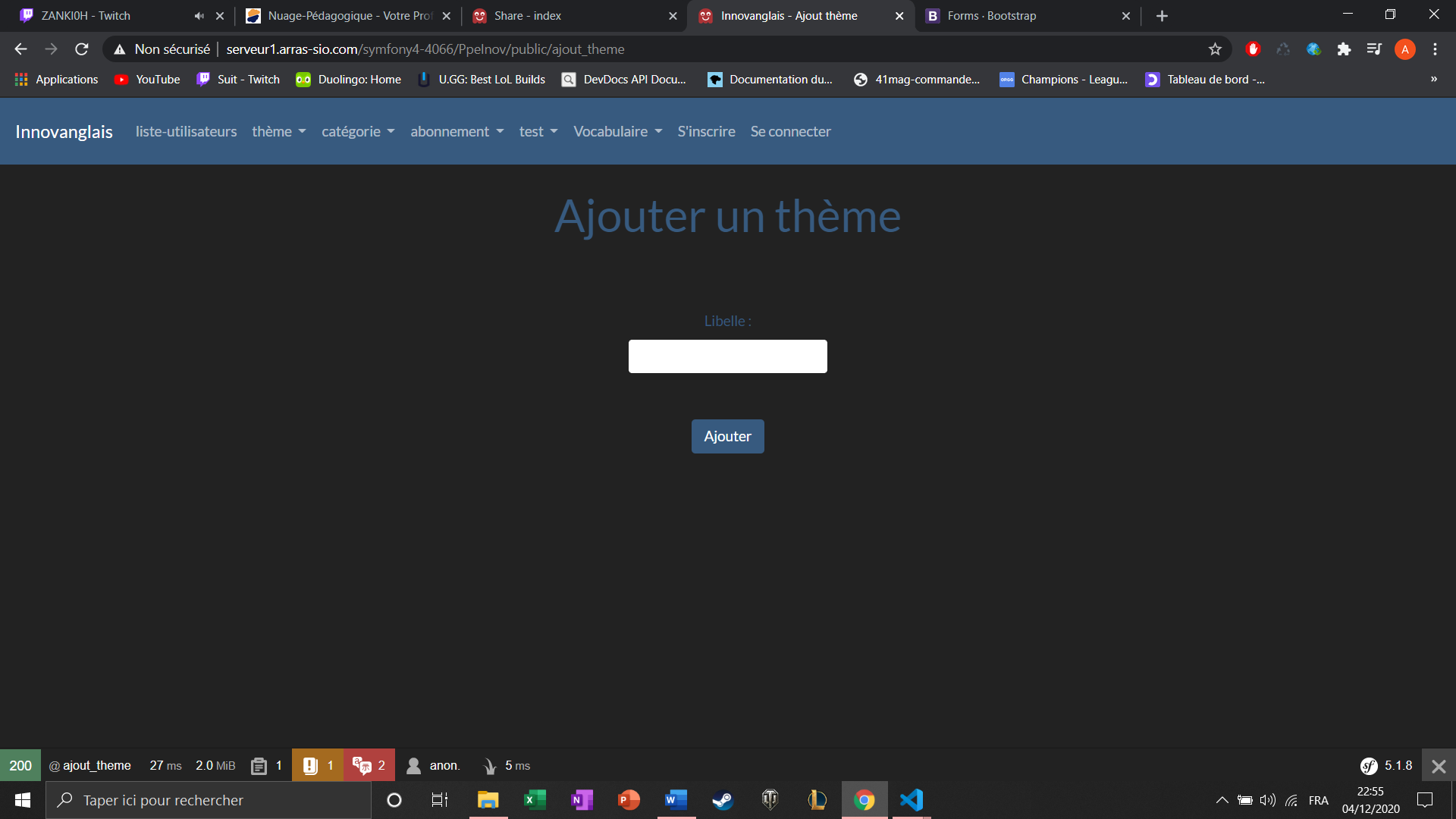
Si l’id donné est faux ou nulle, envoie un message (ligne 72-74).

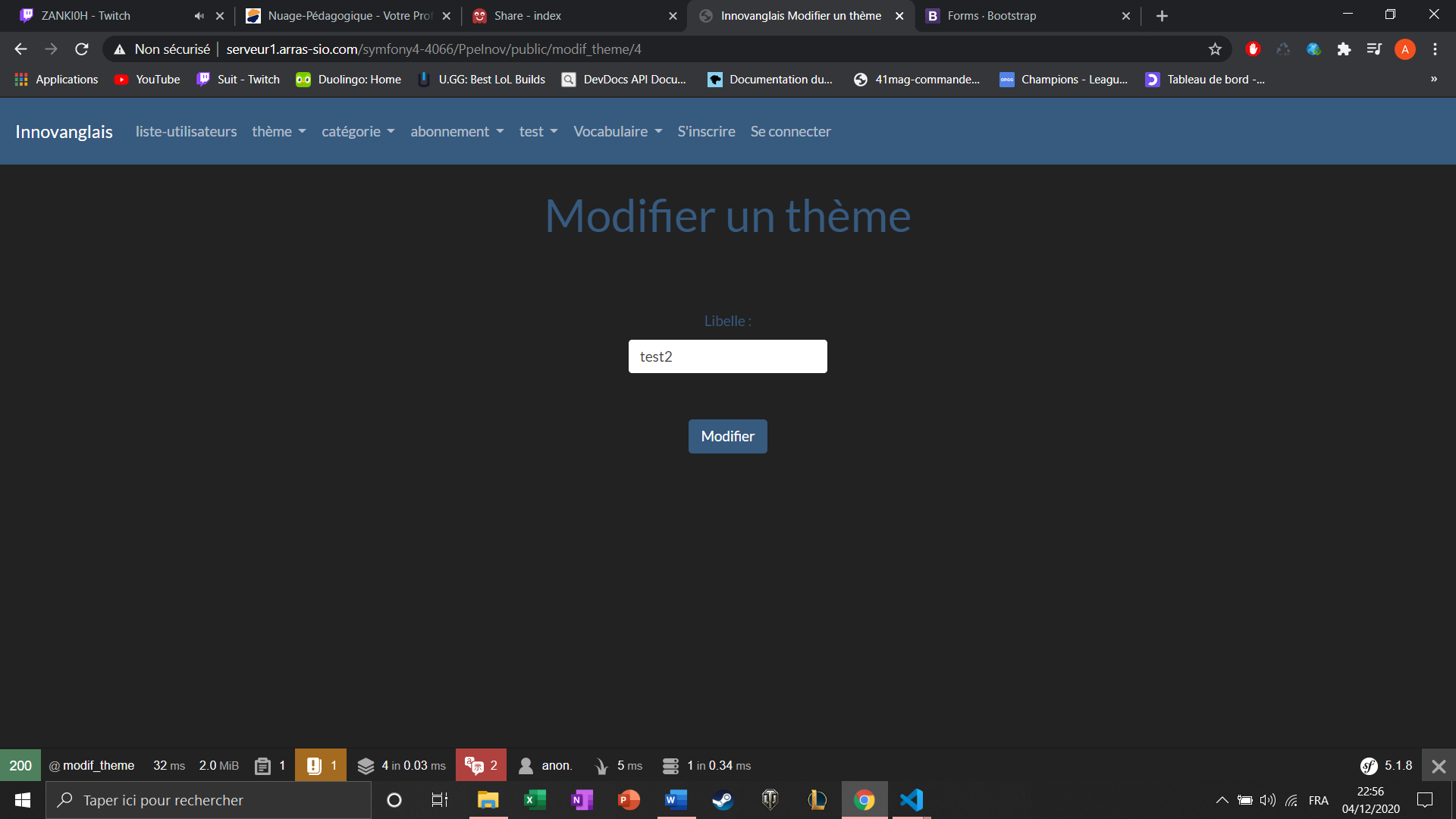
Après les modifications on renvoie sur à la liste des thèmes (ligne 85)

Le formulaire est sur la page modifTheme et dans la liste des thèmes.

Rendu :







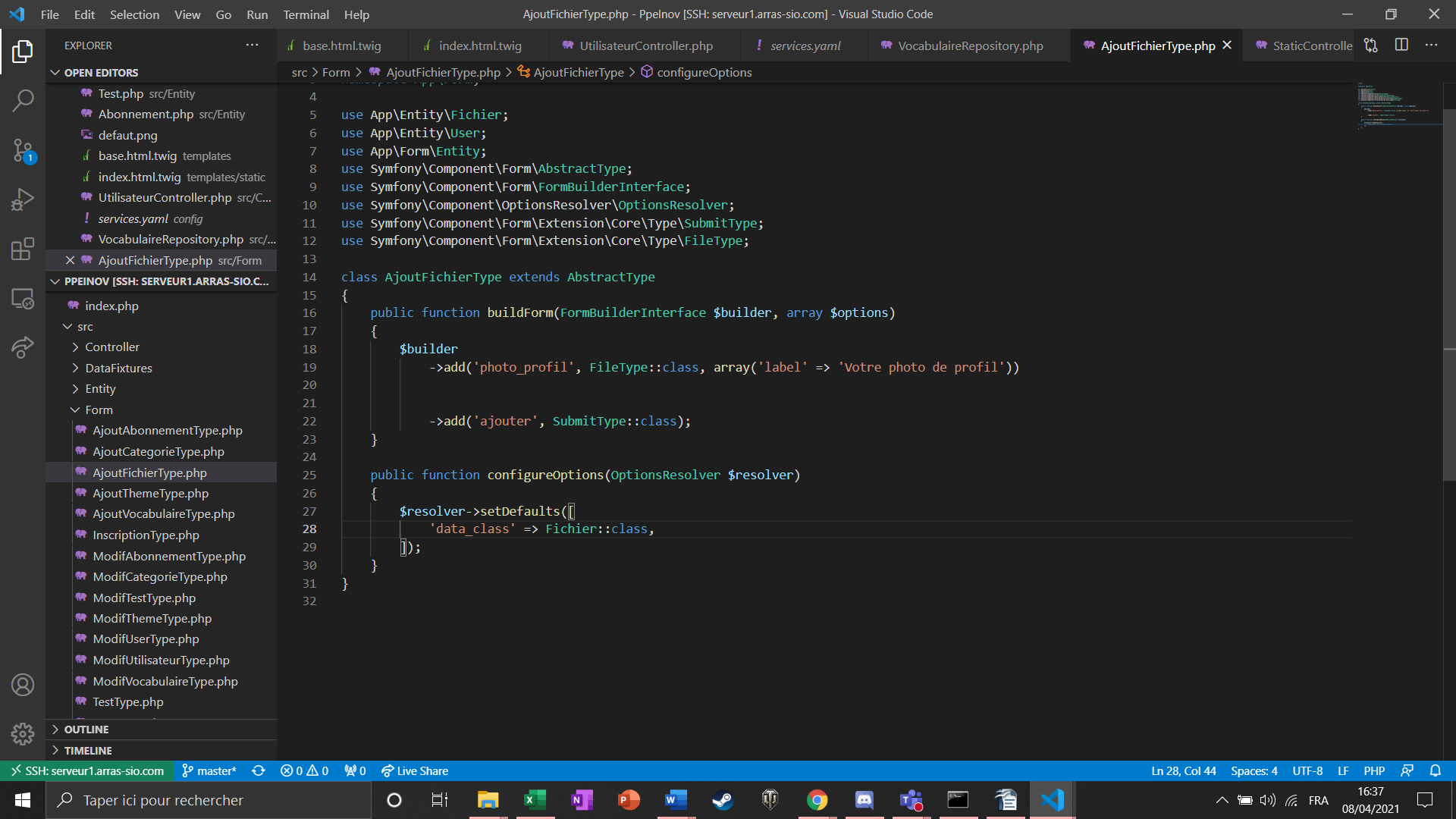
Je fais la même chose pour Categorie…

Création d’un profil :

D’abord on créer un dossier qui va contenir toutes les images puis on ajoute

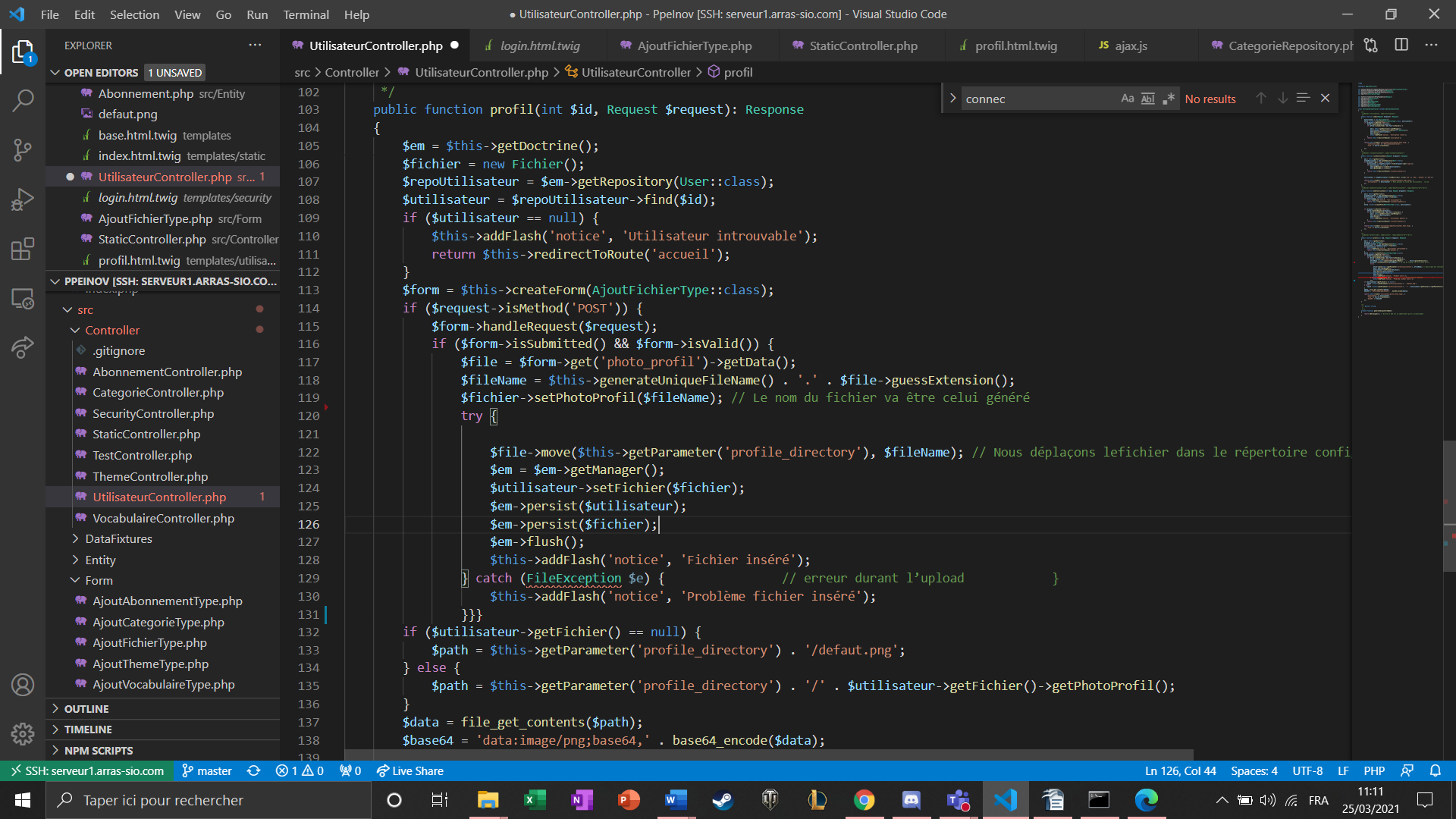
profile\_directory: '%kernel.project\_dir%/liendossier’

Dans le fichier config/service.yaml

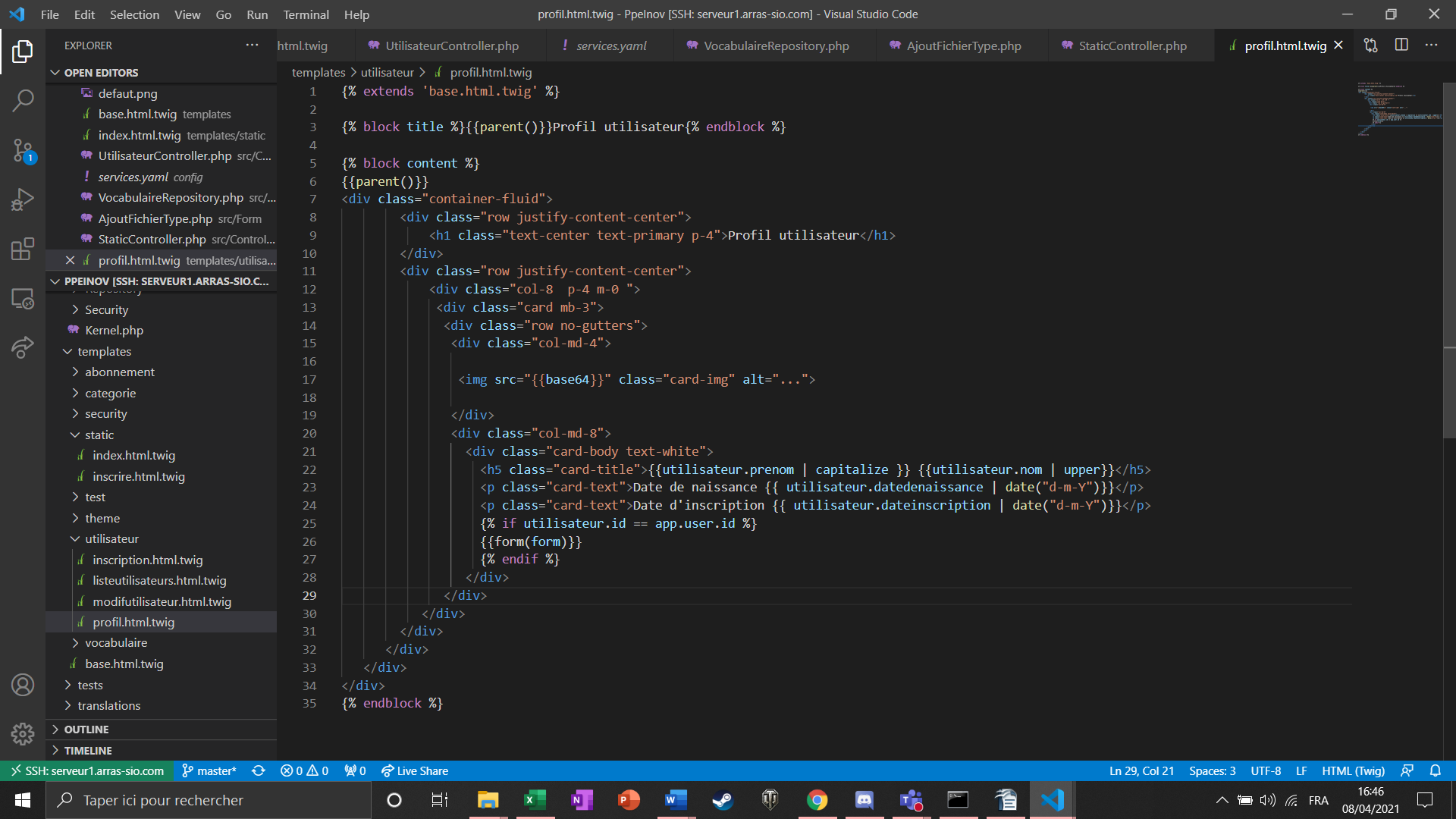


On donne la classe fileType dans le formulaire

Contrôleur :



On récupère les informations puis on les enregistre dans la base de données tandis que fichier sera envoyé dans le dossier précédemment en utilisant Profile\_directory. Si le fichier est null on donne une image de base déjà enregistrée.



Affichage basique pour un profil