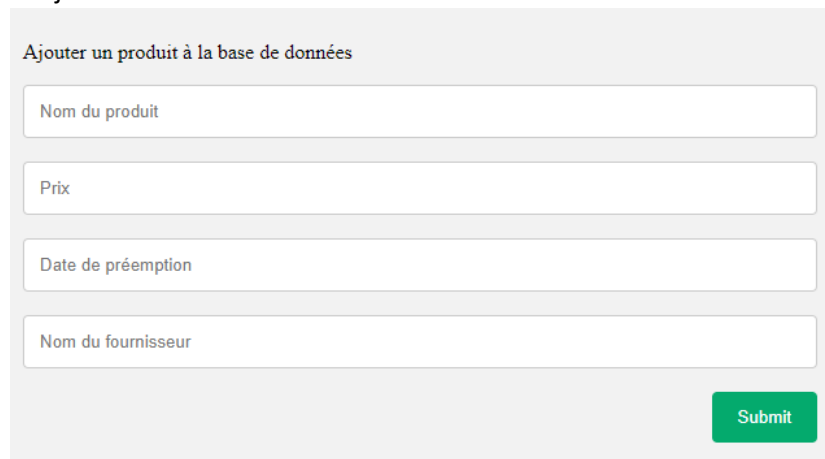


Application de gestion d'achats en PHP

Opérations CRUD

L'objectif de cet exercice est de créer une petite application de gestion d'achats en php en exploitant le paradigme de la Programmation Orientée Objet. L'application propose quelques fonctionnalités (opérations CRUD). Les opérations CRUD expriment les 4 opérations élémentaires (Create, Read, Update et Delete).

- ❑ Créer un dossier pour ce nouveau projet nommé «gestionachats» sur votre serveur web Apache (www pour Wampserver ou htdocs pour Xamp).
- ❑ A l'intérieur de ce dossier, créer le script php «index.php» (utiliser le modèle du TP1), qui a pour objectif d'afficher le formulaire suivant:



Ajouter un produit à la base de données

Nom du produit

Prix

Date de préemption

Nom du fournisseur

Submit

- ❑ Tester le script PHP en invoquant l'url <http://localhost/gestionachats/index.php> ou tout simplement <http://localhost/gestionachats/>
- ❑ Une fois l'utilisateur appuie sur le bouton «Submit», les données seront envoyées via la méthode http post à la même page. C'est-à-dire le traitement de ces données se fera dans le même script php.
- ❑ Créer une classe appelée *Produit* ayant les propriétés protégées suivantes (idProduit, nomProduit, prixProduit, datePreemption, NomFournisseur).
- ❑ Ajouter un constructeur à cette classe permettant de créer des objets de type *Produit*.
- ❑ Ajouter une méthode *AfficherProduit()* à la classe *Produit* permettant d'afficher les informations d'un objet de type *Produit* grâce à l'instruction *echo* ou *print*.
- ❑ Ajouter une méthode *ModifierProduit(\$idProduit, \$nomProduit, \$prixProduit, \$datePreemption, \$nomFournisseur)* à la classe *Produit* permettant de modifier les informations d'un produit selon les données fournies en paramètres à cette fonction.
- ❑ Créer une nouvelle classe appelée *ProduitSolde*. Cette classe représente un produit soldé et hérite de la classe *Produit* tout en ayant une propriété privée appelée *remise*.

- ❏ Redéfinir la méthode `afficherProduit()` pour afficher le produit avec le pourcentage de remise. Par exemple, et en plus des données du produit, la méthode affiche *le pourcentage de remise du Produit X est de 20%*.
- ❏ Redéfinir la méthode `ModifierProduit()` pour modifier les informations d'un produit soldé.
- ❏ Ajouter une méthode appelée `calculerRemise()` qui calcule et affiche le prix du produit en appliquant une remise
- ❏ Ajouter une méthode appelée `appliquerSolde()` qui vérifie si la date de préemption d'un produit s'approche en comparant cette date avec la date d'aujourd'hui. Cette méthode applique une remise X sur un produit dont la date de préemption est inférieure à 6mois.
- ❏ Modifier le traitement associé au bouton «submit» du formulaire pour instancier la classe *Produit* (créer un objet de type *Produit*) à chaque fois l'utilisateur remplit le formulaire et appuie sur le bouton de validation.
- ❏ Ajouter au script appelé «index.php», le code permettant d'enregistrer l'objet produit dans un fichier appelé «produits.txt». Exécuter le script et vérifier que les données sont enregistrées dans le fichier.
- ❏ Ajouter une nouvelle méthode appelée `afficherProduit($file)` qui prend en paramètre un fichier et affiche, sur la même page que le premier formulaire html, la liste des produits (objets) ajoutés au tableau sous la forme suivante. En ajoutant près de chaque produit deux icônes (icône modifier & icône supprimer).

Q

Chercher un produit ...

Produit ▼	Prix	Fournisseur		
PC BUREAU ASUS AIO	12 990 DH	Alami		
PC PORTABLE ACER ASPIRE	5 390 DH	Ahmed		
PC PORTABLE ASUS E210MA	3 690 DH	Alami		
PC PORTABLE LENOVO Legion5	13 999 DH	Alami		

- ❏ Vous pouvez aller plus loin en implémentant le code associé aux deux icônes modifier et supprimer affichés à côté de chaque produit.