# Quels sont les points techniques qui permettent de passer de :

## La page connexion au catalogue

Dans action connexion si la connexion de l’utilisateur est réussi on est envoyé vers le catalogue sinon( si le mot de passe et incorrect ou le mail ) elle nous envoie un message erreur

  if ($uti[0]->getMotDePasse() == crypte($\_POST['motDePasse'])) {

        echo 'connection reussie';

        $\_SESSION['utilisateur'] = $uti[0];

        header("location:index.php?page=Catalogue")

## La liste des articles au formulaire de modification

echo '<div class="caseListe"> <a href="index.php?page=FormArticles&mode=Modifier&id='.$unObjet->getIdArticle().'"><i class="fas fa-pen"></i></a></div>';

## Comment s’effectue le retour

# La liste des articles me fournit un IdTypeArticle, comment récupérer un texte ?

Avec le Tostring qui transforme l’objet en chaine de caractère ..grâce a cette méthode cela permet de données une description satisfaisante

# Dans le formulaire article

## Comment faire apparaitre l’image ?

echo '<label for=Photos class="caseForm labelForm">'.texte("Photos").'</label>';

echo '<div class="caseForm donneeForm"><img src="./IMG/'.$elm->getPhotos().'" /></div>';

echo '<div class="caseForm">

                <input type="hidden" name="AnciennePhotos" value="'.$elm->getPhotos().'">

                <input type="file" name="Photos" '.$disabled.'/></div>';

echo '<div class="caseForm checkForm"><i class="fas fa-check-circle"></i></div>';

## Comment transformer l’input TypeArticle pour avoir une liste déroulante ?

Avec un select et la fonction créer select

echo '<div class="caseForm donneeForm">'.creerSelect($elm->getIdTypeArticle(),"TypesArticles",["LibelleTypeArticle"],$disabled).'</div>';

## Quels fichiers, quelles méthodes entre en jeu ?

Le fichier outils.php il y a fonction créer select qui permet de créer le select

function creerSelect(?int $valeur, string $table, array $nomColonnes, ?string $attributs = "", array $condition = null, string $orderBy = null)

{

        $nomId= $table::getAttributes()[0];

        $select = '<select id="' . $nomId . '" name="' . $nomId . '"' . $attributs . '>';

        $methode = $table . 'Manager';

        $libelle= $nomColonnes;

        array\_push($nomColonnes, $nomId);

        $liste = $methode::getList($nomColonnes, $condition, $orderBy);

        if ($valeur == null) {

                $select .= '<option value="" SELECTED>'.texte("inputDefault").'</option>';

        } else {

                $select .= '<option value="">'.texte("inputDefault").'</option>';

        }

        foreach ($liste as $elt) {

                $content = "";

                foreach ($libelle as $value) {

                        $content .= appelGet($elt, $value) . " ";

                }

                if ($valeur == appelGet($elt, $nomId)) {

                        $select .= '<option value="' . appelGet($elt, $nomId) . '" SELECTED>' . $content . '</option>';

                } else {

                        $select .= '<option value="' . appelGet($elt, $nomId) . '">' . $content . '</option>';

                }

            }

        $select .= "</select>";

        return $select;

}

## Quels sont les modifications à apporter pour permettre le changement d’image ?

Dans actionArticle.php dans le switch il y a la modification de l’image …

L’ancienne photos est supprimer on charge la nouvelle image avec la fonction charger Image

celle ci lui définit un id unique (elle commence par JPG et elle terminera par .JPG)

dans la parti js celle si gerer le chargement de la photo sur le site avec la fonction choix image

case "Modifier": {

        if (isset($\_POST["modifImage"]))

        {   /\* suppression de l'ancienne image\*/

            unlink("IMG/" . $\_POST['AnciennePhotos']);

            /\* on charge la nouvelle image \*/

            $elm->setPhotos(chargerImage());

        }

        else $elm->setPhotos($\_POST['AnciennePhotos']);

        $elm = ArticlesManager::update($elm);

        break;

dans la parti js celle si gerer le chargement de la photo sur le site avec la fonction choix image

function choixImage(e) {

    if (inputFile.value.length > 0) {

        imageType = /^image\//;

        fichier = e.target.files[0];

        if (!imageType.test(fichier.type)) {

            alert("veuillez sélectionner une image");

            e.target.value = "";

# Dans les classes, au niveau des setters, que veut dire ?int ?

Cela veut dire que la valeur attendu est un entier (chiffre) celui-ci est en concordance avec la base de donnees..

public function setIdArticle(?int $IdArticle)

    {

        $this->\_IdArticle=$IdArticle;

# Traduction : quels sont les points techniques qui entrent en jeu ? (A tous les niveaux)

Outils.php

function texte($codeTexte)

{

    $retour=TextesManager::findByCodes($\_SESSION['lang'], $codeTexte);;

    if ($retour==false) return $codeTexte;

    return $retour;

}

testesManager.Class.php permet de voir avec la fonction checkIfLangExist

public static function checkIfLangExist($codeLangue)

    {

        $db=DbConnect::getDb();

        $q=$db->prepare("SHOW COLUMNS FROM textes LIKE :codeLangue");

        $q->bindValue(":codeLangue", $codeLangue, PDO::PARAM\_STR);

        $q->execute();

        $results = $q->fetch(PDO::FETCH\_ASSOC);

        return ($results != false); // renvoi vrai si la requete retourne des données, false sinon

    }

}

# Dans quel(s) contexte(s) sert le fichier DAO.Class.php

Le DAO.class .php a accès avec la base de données

Le manager envoi les données à la dao qui elle va communiquer avec la base de données (qui a accès a la base de données d’envoyer l’objet vers la base de données

Il regroupe toute les fonctions et permet de faire des requêtes via la base de données

# Utilisateurs : A quel endroit intervenir pour forcer le rôle et le mot de passe ?

Dans les routes en mettant 0 qui permet

"Default" => ["PHP/VIEW/FORM/", "FormConnexion", "Connexion", 0, false],

    "Inscription" => ["PHP/VIEW/FORM/", "FormInscription", "Inscription", 0, false],

    "Accueil" => ["PHP/VIEW/GENERAL/", "Accueil", "Accueil", 0, false],

    "Erreur" => ["PHP/VIEW/GENERAL/", "Erreur", "titreErreur", 0, false],

    "ActionConnexion" => ["PHP/CONTROLLER/ACTION/", "ActionConnexion", "Action de la connexion", 0, false],

    "ActionInscription" => ["PHP/CONTROLLER/ACTION/", "ActionInscription", "Action de l'inscription", 0, false],

    "ActionDeconnexion" => ["PHP/CONTROLLER/ACTION/", "ActionDeconnexion", "Action de deconnexion", 0, false],

    "ListeMailAPI" => ["PHP/MODEL/API/", "ListeMailAPI", "ListeMailAPI", 0, true],

    "Catalogue" => ["PHP/VIEW/LISTE/", "ListeCatalogue", "Catalogue", 1, false],

    "FormArticlesClients" => ["PHP/VIEW/FORM/", "FormArticlesClients", "Formulaire Articles", 1, false],

    "ListeArticles" => ["PHP/VIEW/LISTE/", "ListeArticles", "Liste Articles", 2, false],

    "FormArticles" => ["PHP/VIEW/FORM/", "FormArticles", "Formulaire Articles", 2, false],

    "ActionArticles" => ["PHP/CONTROLLER/ACTION/", "ActionArticles", "Action Articles", 2, false],

    "ListePaniers" => ["PHP/VIEW/LISTE/", "ListePaniers", "Liste Paniers", 2, false],

    "FormPaniers" => ["PHP/VIEW/FORM/", "FormPaniers", "Formulaire Paniers", 2, false],

    "ActionPaniers" => ["PHP/CONTROLLER/ACTION/", "ActionPaniers", "Action Paniers", 2, false],

    "ListeLignesPaniers" => ["PHP/VIEW/LISTE/", "ListeLignesPaniers", "Liste Lignes de Paniers", 1, false],

    "FormLignesPaniers" => ["PHP/VIEW/FORM/", "FormLignesPaniers", "Formulaire Lignes de Paniers", 1, false],

    "ActionLignesPaniers" => ["PHP/CONTROLLER/ACTION/", "ActionLignesPaniers", "Action LignesPaniers", 1, false],

    "ListeTypesArticles" => ["PHP/VIEW/LISTE/", "ListeTypesArticles", "Liste Types d'Articles", 2, false],

    "FormTypesArticles" => ["PHP/VIEW/FORM/", "FormTypesArticles", "Formulaire Types d'Articles", 2, false],

    "ActionTypesArticles" => ["PHP/CONTROLLER/ACTION/", "ActionTypesArticles", "Action TypesArticles", 2, false],

    "ListeUtilisateurs" => ["PHP/VIEW/LISTE/", "ListeUtilisateurs", "Liste Clients", 2, false],

    "FormUtilisateurs" => ["PHP/VIEW/FORM/", "FormUtilisateurs", "Formulaire Clients", 2, false],

    "ActionUtilisateurs" => ["PHP/CONTROLLER/ACTION/", "ActionUtilisateurs", "Action Utilisateurs", 2, false],

    "404" => ["PHP/VIEW/GENERAL/", "Erreur", "404", 0, false],

# Panier Utilisateur : Quelles sont les actions à mener pour récupérer les lignes du panier du client ?

actionsLignesPaniers.php

<?php

$elm = new LignesPaniers($\_POST);

/\* On cherche si une Panier est en cours, sinon on la crée \*/

$cmd = PaniersManager::getList(null, ["idClient" => $\_SESSION['utilisateur']->getIdUtilisateur()]);

if (!$cmd) {

    //var\_dump($\_SESSION['utilisateur']->getIdUtilisateur());

    PaniersManager::add(new Paniers(["idClient" => $\_SESSION['utilisateur']->getIdUtilisateur()]));

    $cmd = PaniersManager::getList(null, ["idClient" => $\_SESSION['utilisateur']->getIdUtilisateur()]);

    //var\_dump($cmd);

}

$elm->setIdPanier($cmd[0]->getIdPanier());

switch ($\_GET['mode']) {

    case "Ajouter": {

            $elm->setIdArticle($\_GET["id"]);

            $elm->setQuantite(1);

            $elm = LignesPaniersManager::add($elm);

            break;

        }

    case "Modifier": {

            $elm = LignesPaniersManager::update($elm);

            break;

        }

    case "Supprimer": {

            $elm = LignesPaniersManager::delete($elm);

            break;

        }

}

header("location:index.php?page=ListeLignesPaniers");

listesLignesPagniers.php :

affichage des enregistrements de la base de données

// Affichage des ennregistrements de la base de données

    foreach ($objets as $unObjet) {

        echo '<div class="caseListe donneeListe">' . ArticlesManager::findById($unObjet->getIdArticle())->toString() . '</div>';

        echo '<div class="caseListe donneeListe">' . $unObjet->getQuantite() . '</div>';

        echo '<div class="caseListe"> <a href="index.php?page=FormLignesPaniers&mode=Modifier&id=' . $unObjet->getIdLignePanier() . '"><i class="fas fa-pen"></i></a></div>';

        echo '<div class="caseListe"> <a href="index.php?page=FormLignesPaniers&mode=Supprimer&id=' . $unObjet->getIdLignePanier() . '"><i class="fas fa-trash-alt"></i></a></div>';

    }

# Comment gérer les listes vides ? A quel(s) niveau(x) ?

# Comment fixer le footer en bas de la page et permettre de voir toute la page ?

footer{

    background-color: var(--color5);

    color: var(--color1);

    font-size: 0.8em;

    position: fixed;

    bottom:0;

    width: 100%;

# Présentation sous forme de Grid :

## Où intervenir, comment le paramétrer ?

Dans le grids.css

On peut intervenir pour changer le nombre de colonnesmais une 8 colonnes par exemples

.grid-columns-span-7 {

    grid-column:1/8;

}

.grid-col-5 {

    display : grid;

    /\* border : 0.5px solid #ff0080; \*/

    grid-template-columns : 2fr 2fr 0.5fr 0.5fr 0.5fr;

    grid-gap: 1em;

}

.grid-columns-span-5 {

    grid-column:1/6;

}

.grid-col-4 {

    display : grid;

    /\* border : 0.5px solid #ff0080; \*/

    grid-template-columns : 2fr 0.5fr 0.5fr 0.5fr;

    grid-gap: 1em;

}

## Comment obtenir une case qui fusionne 2 cases adjacentes }

.grid-columns-span-6 {

    grid-column:1/7;

permet le fusionner la grid en lui ajoutant un grid column

# Gestion des erreurs : que faut il faire pour mettre en place une gestion d’erreur ? A quel endroit ?

La pdo

<?php

class TextesManager

{

    public static function findByCodes($codeLangue,$codeTexte)

    {

        $retour = DAO::select([$codeLangue],"textes",["codeTexte" => $codeTexte],null,null,true);

        if ($retour==false) return false;

         return $retour[0][$codeLangue];

    }

    public static function checkIfLangExist($codeLangue)

    {

        $db=DbConnect::getDb();

        $q=$db->prepare("SHOW COLUMNS FROM textes LIKE :codeLangue");

        $q->bindValue(":codeLangue", $codeLangue, PDO::PARAM\_STR);

        $q->execute();

        $results = $q->fetch(PDO::FETCH\_ASSOC);

        return ($results != false); // renvoi vrai si la requete retourne des données, false sinon

    }

}

<?php

class DbConnect{

    private static $db;

    public static function getDb()

    {

        return DbConnect::$db;

    }

    public static function init()

    {

        try {

            self::$db= new PDO ( 'mysql:host='. Parametres::getHost().';port='. Parametres::getPort() .';dbname='. Parametres::getDbname().';charset=utf8', Parametres::getLogin(), Parametres::getPwd());

        }

        catch (Exception $e)

        {

            die('Erreur :'. $e->getMessage());

        }

    }

}