**Une image contenant texte, clipart

Description générée automatiquementConcepteur Développeur d’Applications**

**Evaluation Rattrapage**

**PHP/HTML/CSS/JS : Application Web**

**Durée : 2 h**

**Documents autorisés : Oui**

**Internet autorisé : Oui**

**Consignes :**

Ce document est à compléter et à mettre à la racine de votre espace réseau dans le dossier **EVAL\_PHP\_Rattr\_[Prénom]**

L’évaluation porte sur **vos connaissances en matière de développement web**.

* Les réponses doivent être précises. Mettre des extraits courts de code (uniquement ceux qui répondent à la question), préciser le fichier et la méthode s’il y a lieu.

**Sujet :**

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

## La page connexion au catalogue

Si la connexion s’est bien dérouler, alors l’utilisateur est connecté et est automatiquement rediriger vers la page catalogue

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

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

## La liste des articles au formulaire de modification

Grace a un élément d’ancre dans lequel on précisera le mode et l’id de l’objet

Ex :

<a href="index.php?page=FormArticles&mode=Afficher&id='.$unObjet->getIdArticle().'">

## Comment s’effectue le retour

Le retour s’effectue via un bouton, qui renvoie vers la page dont le chemin a été renseigné dans le Href

EX : <div><a href="index.php?page=ListeArticles"><button type="button"><i class="fas fa-sign-out-alt fa-rotate-180"></i></button></a></div>

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

Grace a la fonction toString qui se trouve dans la classe TypeArticle.Class.php et qui renvoie le libelleTypeArticle, on fait un findbyId Afin de récupérer l’objet que l’on veut, puis on fait un toString de cet Objet

public function toString()

    {

        return $this->getLibelleTypeArticle();

    }

PUIS

TypesArticlesManager::findById($unObjet->getIdTypeArticle())->ToString()

# Dans le formulaire article

## Comment faire apparaitre l’image ?

Grace à la fonction ChoixImage dans FromArticles.js qui récupère le chemin de l’image sélectionner dans l’input de type file et l’affiche

 fichier = e.target.files[0];

img.file = fichier;

reader = new FileReader();

reader.addEventListener("load", chargeImage);

reader.readAsDataURL(img.file);

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

En utilisant la méthode creerSelect qui se trouve dans Outils.php

Ex :

creerSelect($elm>getIdTypeArticle(),"TypesArticles",["LibelleTypeArticle"],$disabled)

## Quels fichiers, quelles méthodes entre en jeu ?

La méthode qui entre en jeu est creeSelect qui se trouve dans Outils.php

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

modif = document.createElement("input");

modif.name = "modifImage";

modif.type = "hidden";

inputFile.parentNode.appendChild(modif);

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

Cela veut dire qu’on accepte que l’élément passé en paramètre soit null ou de type int

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

Utilisation de la méthode texte

Ex :

texte("Bonjour") et qui renvoie soit Bonjour soit Hello selon la langue sélectionner

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

La fichier DAO.class.php permet la récupération, l’ajout, la modification et la suppression des données stockées en base.

EX :

DAO::select(LignesPaniers::getAttributes(), "LignesPaniers", ["IdLignePanier" => $id])[0] ;

return DAO::add($obj) ;

return DAO::update($obj) ;

return DAO::delete($obj) ;

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

Pour forcer le role et le mot de passe il faut intervenir dans ActionUtilisateurs

$elm->setRole(1);

$elm->setMotDePasse(crypte($elm->getNom().$elm->getPrenom()));

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

On vérifie d’abord si le client possède déjà un panier (sinon il doit en créer un), en récupérant via le getList le panier pour lequel l’id Client est égale a l’id Utilisateur de l’Utilisateur actuellement connecté, Puis on récupère via le getList les lignes de son panier pour lequel l’id panier est égale à l’id Panier associé au client actuellement connecté

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

if (!$cmd)

    echo '<div class="gridListe center">Pas de panier en cours</div>';

else {

    $objets = LignesPaniersManager::getList(null, ["idPanier" => $cmd[0]->getIdPanier()]);

# 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 ?

En ajoutant ces 2 propriétés au footer dans le CSS :

position: fixed;

bottom:0

# Présentation sous forme de Grid :

## Où intervenir, comment le paramétrer ?

Il faut intervenir dans le CSS , on le paramètre grâce à la propriété :

grid-template-columns : ;

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

grid-column:1/3;

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

La gestion d’erreur se fait dans le fichier index.php , si le chemin de la page correspond a une route prédéfini alors celui-ci est afficher sinon l’utilisateur est renvoyé vers une page d’erreur

if (isset($routes[$page])) {

        AfficherPage($routes[$page]);

    } else {

        AfficherPage($routes["404"]);}