**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

Dans l’action connexion, si on rentre dans les conditions du IF, on est redirigé vers la page catalogue comme nous l’indique (ActionConnexion/ligne 8) :

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

## La liste des articles au formulaire de modification

Dans ListeArticles.php, le php permet d’intégrer du html, on y intègre certains lien dont un qui amène au formulaire de modification (ListeArticles/ligne 42) : href="index.php?page=FormArticles&mode=Modifier&id='.$unObjet->getIdArticle().'"

## Comment s’effectue le retour

Il s’effectue grâce à un bouton qui ramène à la page accueil come par exemple (ListeArticles/ligne 49) :

<a href="index.php?page=Accueil"><button><i class="fas fa-sign-out-alt fa-rotate-180"></i></button></a>

On ajoute une icone via fontawsome pour l’esthétique.

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

Grace à la methode ToString (ListeArticles/ligne 39) :

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

# Dans le formulaire article

## Comment faire apparaitre l’image ?

On intègre une balise <img> avec le PHP dont la source sera complétée par la méthode GetPhoto() qui ramènera le chemin d’où est stoker l’image (FormArticles/ligne 46) :

<img src="./IMG/'.$elm->getPhotos().'" />

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

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

## Quels fichiers, quelles méthodes entre en jeu ?

Grace à la méthode creerselect() qui se trouve dans Outils.php :

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

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

Dans ActionArticles.php la methode chargerImage() qui recupere un string qui est le chemin de notre image, on assigne cette methode au setPhoto() lorsque le formulaire est en mode ‘’Modifier‘’ (ActionArticles/ligne 30) :

*function* chargerImage()

{

    if (**is\_uploaded\_file**($\_FILES["Photos"]["tmp\_name"]))

    {

        $leNom = **uniqid**('jpg\_') . '.jpg';

**move\_uploaded\_file**($\_FILES['Photos']['tmp\_name'], 'IMG/' . $leNom);

    }

    return $leNom;

}

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

Cela signifie que le setter attend soit un Int soit un Null, et s’il ne reçoit rien il sera de type Null.

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

Au niveau de la base de donnée, une table Texte a été ajouté, celle-ci comporte toutes les traductions suivant le code langue.

Au niveau du PHP, une fonction texte() a été ajouté dans Outils.php, celle-ci retourne le code langue qu’il faudra appliqué sur la page.

Cette fonction est utilisée lors de l’intégration Html, elle est appliqué au texte affiché sur la page web, comme les labels, les titres etc…

Comme par example (ListeArticles/ligne 22) :

<div class="caseListe gridLabel">'.texte("Prix").'</div>

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

Il sert à simplifier la rédaction des Managers, car il regroupe toutes les fonctions nécessaires à l’élaboration du manager (ArticlesManager/ligne 8) :

        return *DAO*::add($obj);

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

(ActionUtilisateur/ligne 3) :

$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 ?

Il faut récupérer le panier de l’utilisateur connecté, pour cela il faut (ListeLignesPaniers/ligne 9) :

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

Pour recuperer les lignes du panier en question on écrit :

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

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

Il faut faire attention à ce que la lignePaniers puisse être de type Null en base de donnée, et que l’on puisse rajouter une ligne vide en base de donnée (LignesPaniers.Class/ligne 21) :

public *function* setIdLignePanier(?int $IdLignePanier)

On fait un if qui vérifie si le panier de l’utilisateur est vide au pas, au niveau de ListeLignesPaniers.php (ListeLignesPaniers/ligne 11) :

if (!$cmd)

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

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

Grace au CSS, on le fixe et on le met tout en bas (style.css/ligne 15)

footer{

*position*: fixed;

*bottom*:0;

# Présentation sous forme de Grid :

## Où intervenir, comment le paramétrer ?

Au niveau du php, on intègre une div avec la classe ‘’ grid-col-8 ‘’ par exemple celle-ci permettra la création d’une grille à 8 colonnes (ListeArticles/ligne 12) :

**echo** '<div class="grid-col-8 gridListe">';

Au niveau du css, on ajoute autant de classe que l’on veut suivant la grille que l’on aimerait utilisé, pour le parametrage des grilles on utilise *grid-template-columns* pour le nombre de colonnes desirer et *grid-column* si on desire avoir une ligne qui prend plusieurs colonnes par example

grids.css/ligne 65) :

.grid-columns-span-8 {

*grid-column*:1/9;

} (Pour obtenir un élément qui prend 8 colonnes d’une grille)

(grids.css/ligne 58) :

.grid-col-8 {

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

}

J’ai enlevé certains morceaux essentiels à la création d’une grille, mais ici je montre le paramétrage d’un grille de 8 colonnes.

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

(grids.css/ligne 44) :

.grid-columns-span-2 {

*grid-column*:1/3;

}

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

Il faut créer les routes au niveau de l’index.php, celles-ci permettent de rediriger l’utilisateur vers la page adéquat à son erreur.

Exemple (index.php/ligne 62) :

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

Pour la detection de l’erreur Il faut créer une page qui récupère Erreur.php qui regarde dans l’url le code erreur de la session et qui retourne la page correspondant à l’erreur et si il n’y en a pas retourne la page de connexion.