**Situation 2 : Athéna.**

**Présentation**

La société Athéna effectue un appel d’offre auprès des différentes Ssii et web agency. Celui-ci porte sur la création d’un CMS (Contrat Management System) sur mesure

Notre société devra répondre à cet appel d’offre et proposer une solution complète et maquetté à la société Athéna

Une fois à la réponse l’appel d’offre validé par le client, chaque prestataire devra réaliser le CMS et le présenter au client en s’appuyant au dossier technique complet.

Pour ce projet, notre équipe composée de Thanussan et Ulysse, représentant une société de service, avec un chef de projet identifié qui coordonnera l’activité de son équipe et définira les axe de travail. Seul ce chef de projet pourra communique avec les formateurs qui représenteront la société ATHENA afin de pouvoir demander des compléments d’information nécessaire au projet

Un espace de stockage centralisé sera mis à disposition pour les équipes. La sécurisation de l’accès aux données est à leurs charges.

**Expression du besoin**

La société Athéna est une société spécialisée dans le développement de sites Internet et Intranet. Son activité dépend actuellement fortement des systèmes CMS existants, qui ne correspondent que partiellement aux besoins des clients.

Cela entraine un temps de développement plus important, certains modules proposés par les CMS classiques devant être désactivés, remodelés voir même redéveloppés.

L’appel d’offre porte sur le développement d’un outil de gestion de contenu répondant aux besoins suivants. La liste exprime les fonctionnalités minimales devant apparaitre, chaque prestataire est libre d’y ajouter des fonctionnalités additionnelles.

* L’objectif de ce CMS est de déployer rapidement des sites de ventes de produit particuliers
* Les usagers doivent s’identifier par un couple login/mot de passe pour bénéficier des fonctionnalités d’achat et de vente. Un utilisateur non connecté ne peut que parcourir la liste des articles disponible
* Les articles proposés à la vente sont stocké dans une base de données. Les usagers authentifiés peuvent ajouter de nouveaux articles à cette base de donné.
* A chaque article correspond une fiche contenant les informations détaillées de celui-ci.
* Lorsqu’un article est ajouté à la base de donné, il est directement disponible à la vente.

Les administrateurs vérifient la fiche descriptive et la valide. Une fois cette fiche validée, les informations renseignées ne sont plus modifiables.

* Lorsqu’un usager désire mettre un article en vente, il le sélectionne dans la base de données et indique qu’il possédé un exemplaire à vendre ainsi que son prix de vente.
* Un usager acheteur doit se voir la liste des vendeurs pour un article et les prix demandés par chacun
* Une fonction de recherche doit permette aux usagers de localiser un article par son nom ou par des tags associés.
* Un système de back-office doit permettre aux sociétés clientes la gestion des comptes utilisateurs, des articles présents en base de données et de catégories d’articles.
* Un système de panier doit permettre aux usagers d’acheter plusieurs articles simultanément
* Un système de contact doit permettre l’envoi de mails de la part des usagers aux administrateurs du site.
* L’application doit être accessible aussi bien sur pc que sur mobile

Ce système étant destiné à des clients travaillant sur des secteurs d’activité distincts, votre solution devra présenter une modularité suffisante pour s’adapter à n’importe quel type de contenu.

**Réponse au besoin**

* L’objectif de ce CMS est de déployer rapidement des sites de ventes de produit particuliers

Nous avons décidé de faire un site de commerces, ou les utilisateurs peuvent ajouter, acheter ou vendre le produit avec un système de panier. Cependant les fonctionnalités de livraison, d’achat avec carte bancaire n’a pas été mise en place pour que le site soit le plus simple mais permette la vente et l’achat de biens. Les utilisateurs savent qui est le vendeur éventuel

* Les usagers doivent s’identifier par un couple login/mot de passe pour bénéficier des fonctionnalités d’achat et de vente. Un utilisateur non connecté ne peut que parcourir la liste des articles disponible

Le site est décompensé en trois stades différents. Le visiteur pourra qu’accéder au page Accueil, connexion et inscription avec la demande du login et du mot de passe. Le visiteur pourra voir les produits en vente sur la page accueil. On stocke le couple login et mot de passe dans la base de donné.

La requête SQL utilisée sont :

* Sur la page connexion.php : SELECT \* FROM utilisateur WHERE login = : login AND motdepasse = : motdepasse
* Sur la page inscription : "INSERT INTO utilisateur (nom, prenom, mail, login, motdepasse, groupe, rue, codepostal, ville) VALUES ('".$nom." ', '".$prenom."', '".$email."', '".$email."', '".$password."', 4 , '".$rue."', '".$codepostal."', '".$ville."')"

Il est à noter que sur les pages du site, pour éviter que le visiteur accède au page avec la barre de navigation, on a écrit :

$login = $\_SESSION['login'] ;

$motdepasse = $\_SESSION['motdepasse'] ;

If(($login=='') && ($motdepasse=='') )

{ header('Location: deconnexion.php'); }

Avec deconnexion.php qui déconnectera automatiquement

Voir inscription.php ; connection.php  pour le code lié à la fonctionnalité login/motdepasse

La tabler utilisateur est

|  |
| --- |
| utilisateur |
| Clé primaire : id(int) |
| Nom (varchar)  Prenom(varchar)  mail(varchar)  login(varchar)  motdepasse (varchar)  groupe (int)  rue (varchar)  codepostal(int)  ville (varchar) |

* Les articles proposés à la vente sont stocké dans une base de données. Les usagers authentifiés peuvent ajouter de nouveaux articles à cette base de donné.

Pour stocker les annonces, les produit différents, on créer une base de donné produit.

Voici la table produit :

|  |
| --- |
| Produit |
| Clé primaire : id\_produit (int) |
| Titre (varchar)  Image(varchar)  Prix (int)  Fiche(varchar)  Stock (varchar)  Validation (bool)  Id\_vendeur (int)  Id\_catégorie(int)  Date (date)  Tag (varchar) |

Pour l’ajout d’un article à la base de donné la requête SQL sera : "INSERT INTO `athena`.`produit` (`id\_produit`, `titre`, `image`, `prix`, `fiche`, `stock`, `validation`, `id\_vendeur`, `id\_categorie`, `date`, `tag`) VALUES (NULL,'".$nom."', '".$prix."', '' ,'".$fiche."', " '1', '0', '".$id."', '', NOW() , '".$tag."'); "

La page qui traite de l’ajout du produit est vendre.php

* A chaque article correspond une fiche contenant les informations détaillées de celui-ci.

L’affichage de la table produit en annonce se fait avec la requête SQL : "SELECT `produit`.`titre` , `produit`. `date`, `produit`.`fiche`, `produit`.`prix`, `utilisateur`.`nom` FROM `produit`, `utilisateur` WHERE `utilisateur`.`id` = `produit`.`id\_vendeur` Order By `date` desc limit 10 "

Voir affichep.php

* Lorsqu’un article est ajouté à la base de donné, il est directement disponible à la vente.

Les administrateurs vérifient la fiche descriptive et la valide. Une fois cette fiche validée, les informations renseignées ne sont plus modifiables

Pour la validation des fiches on utilisera la fonction

"UPDATE `athena`.`produit` SET `validation` = '1' WHERE `produit`.`id\_produit` = '".$id."';";

Le booléen validation permet de ne pas voir les fiche validée, et donc de ne sont pas modifiable après.

Voir la page : validerf.php

* Lorsqu’un usager désire mettre un article en vente, il le sélectionne dans la base de données et indique qu’il possédé un exemplaire à vendre ainsi que son prix de vente.

Même si on n’a pas mis de sélection sur la base de donné, l’utilisateur peut ajouter un produit et vendre son bien avec la page ajouter un produit.

* Un usager acheteur doit se voir la liste des vendeurs pour un article et les prix demandés par chacun

L’affichage actuel des annonces des produits permet l’affichage du vendeur et du prix qu’il demande

Voir : affichep.php

* Une fonction de recherche doit permette aux usagers de localiser un article par son nom ou par des tags associés.

On a fait une page Recherche sur le site qui prend en compte le nom, la description du produit mais aussi des tags éventuels ajoutés.

La commande SQL est : "SELECT \* FROM produit where ('nom' LIKE '$recherchec' ) OR (`fiche` LIKE '%$recherchec%' )" OR (`tag` LIKE '%$recherchec%' )" ORDER BY 'nom' ;")

Voir rechercher.php

* Un système de back-office doit permettre aux sociétés clientes la gestion des comptes utilisateurs, des articles présents en base de données et de catégories d’articles.

On a créé un page admin.php, caché sur le site pour que l’administrateur accède, avec un login/mot de passe pour aller sur le tableau de bord et notamment sur la gestion des comptes

La commande SQL : SELECT \* FROM utilisateur WHERE login = :login AND motdepasse = :motdepasse AND groupe ='3'

**Pour la gestion des comptes,**

On a créé la page gcompte.php qui intègre lui-même différent page

Pour afficher les comptes on utilise afficheu.php

Commande SQL : "SELECT \* FROM utilisateur" ;

Pour modifier les comptes on utilise modifu.php

Commande SQL : "UPDATE athena.utilisateur SET nom ='".$nom."', prenom ='".$prenom."', groupe = '".$groupe."', mail = '".$email."', login ='".$email."' , motdepasse = '".$password."', rue= '".$rue."', codepostal = '".$codepostal."', ville = '".$ville."' WHERE `utilisateur`.`id` = '".$id."' ;"

Pour ajouter les comptes on utilise ajoutu.php :

"INSERT INTO utilisateur(nom, prenom, groupe, mail, login, motdepasse, rue, codepostal, ville) VALUES ('".$nom."' ,'".$prenom."' ,'".$groupe."' ,'".$email."' , '".$email."', '".$password."', '".$rue."', '".$codepostal."', '".$ville."')"

L’utilisateur peut gérer son compte sur la page compte.php avec les pages afficheut.php qui affiche son compte et modifuco.php pour qu’il puise modifier celui-ci

Pour afficher la commande SQL : "SELECT \* FROM utilisateur where `login` = '".$login."' AND `motdepasse` = '". $motdepasse."‘;"

Pour modifier la commande SQL : "SELECT \* FROM utilisateur where `login` = '".$login."' AND `motdepasse` = '". $motdepasse."‘;"

Voir admin.php pour l’accès au tableau de bord ; Voir gcompte.php, afficheu.php , ajoutu.php, modifu.php pour la gestion des comptes en tant qu’admin . pour l’utilisateur la gestion de son compte se fera avec :compte.php , afficheut.php et modifuco.php

**Des articles présents en base de données**

**Thanussan**

Voir carticle.php

**Catégories d’articles.**

La page du tableau de bord qui regroupe les fichiers traitant des catégories d’article est mcategorie.php.

Pour afficher les catégories on utilise la page affichec.php

Commande sql: "SELECT \* FROM `categorie` "

Pour ajouter une catégorie on utilise la page ajoutc.php

Commande SQL : "INSERT INTO categorie (libelle\_categorie) VALUES('".$libelle."')"

Table Catégorie

|  |
| --- |
| Catégorie |
| Clé primaire : id\_categorie (int) |
| Libelle\_categorie (varchar) |

Voir mcategroie.php, affichec.php et ajoutc.php ;

* Un système de panier doit permettre aux usagers d’acheter plusieurs articles simultanément

On a fait un système de panier qui permet d’ajouter un produit à la fois. Pour des besoins de temps et de fonctionnalité on n’a pas gérer la possibilité d’acheter plusieurs articles en simultanée.

Le formulaire actuel permet juste l’ajout d’un panier

La page du panier est : panier.php

Pour l’ajout du panier on utiliser le fichier : ajoutpanier.php .

Commande SQL : " INSERT INTO `athena`.`commande` (`id\_commande`, `id\_utilisateur`, `id\_produit`, `stock`, `prix`, `nom`)"

. "VALUES (NULL, '".$idu."', '".$idp."', '".$stock."', '".$prixt."', '".$nom."'); "

Pour la consultation du panier : consulpanier.php .

Commande sql : "SELECT `prix`,`stock`,`id\_produit`,`nom` FROM `commande` WHERE 1 ";

Table panier

|  |
| --- |
| Panier |
| Clé primaire : id\_categorie (int) |
| Libelle\_categorie (varchar) |

A voir :panier.php ; ajoutpanier.php ; consulpanier.php .

* Un système de contact doit permettre l’envoi de mails de la part des usagers aux administrateurs du site.

On utilise la fonction mail. La page pour envoyer un mail est contact.php

Fonction : mail($destinataire, $sujet, $message);

* L’application doit être accessible aussi bien sur pc que sur mobile

Le site est responsive. Pour cela on a utilisé bootstrap.

Pour intégrer bootstrap sur toutes les pages on a fait :

<link rel="stylesheet" href="bootstrap-3.3.6-dist/css/bootstrap.min.css">

<link rel="stylesheet" href="css/bootstrap.css">