# Objectifs

1. Inscription d’un utilisateur
2. Administration CRUD de la table snows

# PARTIE 1 Inscription d’un nouvel utilisateur

Créez la vue register.php :

<?php  
 $title ='Rent A Snow – Register';  
  
ob\_start();  
?>  
 <h1>S'inscrire</h1>  
 <?php if (@$\_GET['registerError'] == true) :?>  
 <h5><span style="color:red">Inscription refusée</span></h5>  
 <?php endif ?>  
 <article>  
 <form class='form' method='POST' action="index.php?action=register">  
  
 <p>Pour vous inscrire chez Snows, nous vous prions de renseigner les champs suivants:</p>  
 <div class="container">  
 <label for="userEmail"><b>Username</b></label>  
 <input type="email" placeholder="Enter email address" name="inputUserEmailAddress" required>  
  
 <label for="userPsw"><b>Password</b></label>  
 <input type="password" placeholder="Enter Password" name="inputUserPsw" required>  
  
 <label for="psw-repeat"><b>Repeat Password</b></label>  
 <input type="password" placeholder="Repeat Password" name="inputUserPswRepeat" required>  
  
 <p>En soumettant votre demande de compte, vous validez les conditions générales d'utilisation.<a href="https://termsfeed.com/blog/privacy-policies-vs-terms-conditions/">CGU et vie privée</a>.</p>  
 <button type="submit" class="registerbtn">Register</button>  
 </div>  
 </form>  
 <div class="container signin">  
 <p>Déjà membre ? <a href="index.php?action=login">Login</a>.</p>  
 </div>  
 </article>  
<?php   
 $content = ob\_get\_clean();  
 require 'gabarit.php';  
?>

Dans index.php, Ajoutez la route register :

case 'register' :  
 register($\_POST);  
 break;

Dans controller.php ajoutez la fonction suivante :

function register($registerRequest){  
 //variable set  
 if (isset($registerRequest['inputUserEmailAddress']) && isset($registerRequest['inputUserPsw']) && isset($registerRequest['inputUserPswRepeat'])) {  
  
 //extract register parameters  
 $userEmailAddress = $registerRequest['inputUserEmailAddress'];  
 $userPsw = $registerRequest['inputUserPsw'];  
 $userPswRepeat = $registerRequest['inputUserPswRepeat'];  
  
 if ($userPsw == $userPswRepeat){  
 require\_once "model/usersManager.php";  
 if (registerNewAccount($userEmailAddress, $userPsw)){  
 createSession($userEmailAddress);  
 $\_GET['registerError'] = false;  
 $\_GET['action'] = "home";  
 require "view/home.php";  
 }  
 }else{  
 $\_GET['registerError'] = true;  
 $\_GET['action'] = "register";  
 require "view/register.php";  
 }  
 }else{  
 $\_GET['action'] = "register";  
 require "view/register.php";  
 }  
}

Ajoutez la fonction suivante à usersManager.php :

*/\*\*  
 \* This function is designed to register a new account  
 \** ***@param*** *$userEmailAddress  
 \** ***@param*** *$userPsw  
 \** ***@return*** *bool|null  
 \*/*function registerNewAccount($userEmailAddress, $userPsw){  
 $result = false;  
  
 $strSeparator = '\'';  
  
 $userHashPsw = password\_hash($userPsw, *PASSWORD\_DEFAULT*);  
  
 $registerQuery = 'INSERT INTO users (`userEmailAddress`, `userHashPsw`) VALUES (' .$strSeparator . $userEmailAddress .$strSeparator . ','.$strSeparator . $userHashPsw .$strSeparator. ')';  
  
 require\_once 'model/dbConnector.php';  
 $queryResult = executeQuery($registerQuery);  
 if($queryResult){  
 $result = $queryResult;  
 }  
 return $result;  
}

Ajoutez la fonction suivante à dbConnector.php

*/\*\*  
 \* This function is designed to insert value in database  
 \** ***@param*** *$query  
 \** ***@return*** *bool|null : $statement->execute() returne true is the insert was successful  
 \*/*function executeQuery($query){  
 $queryResult = null;  
  
 $dbConnexion = openDBConnexion();//open database connexion  
 if ($dbConnexion != null)  
 {  
 $statement = $dbConnexion->prepare($query);//prepare query  
 $queryResult = $statement->execute();//execute query  
 }  
 $dbConnexion = null;//close database connexion  
 return $queryResult;  
}

Quelle différence notez-vous par rapport à la fonction executeQuerySelect ?

# PARTIE 2 Administration de la table snows

Nous allons créer les pages CRUD (Create Read Update Delete) pour la table snows. La partie Read existe déjà, c’est la vue snows sous forme de tableau accessible par les vendeurs uniquement. (Ici la méthode qui charge snowsSeller.php pour les vendeurs.

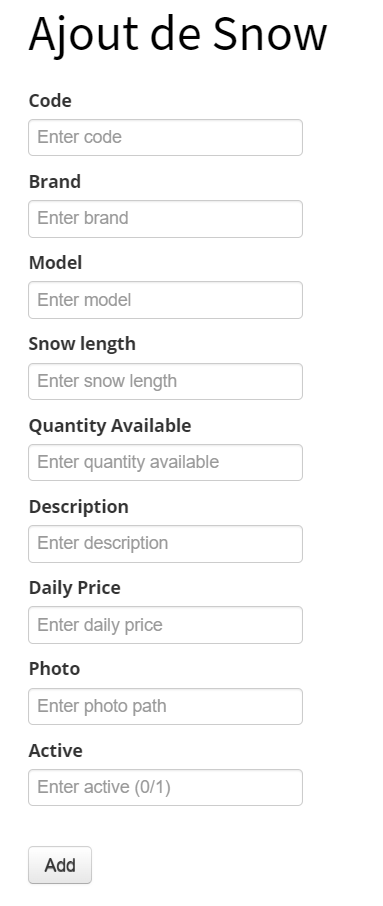
function displaySnows(){  
 if (isset($\_POST['resetCart'])) {  
 unset($\_SESSION['cart']);  
 }  
  
 require\_once "model/snowsManager.php";  
 $snowsResults = getSnows();  
  
 $\_GET['action'] = "displaySnows";  
 if (isset($\_SESSION['userType']))  
 {  
 switch ($\_SESSION['userType']) {  
 case 0://this is a customer  
 require "view/snows.php";  
 break;  
 case 1://this a seller  
 require "view/snowsSeller.php";  
 break;  
 default:  
 require "view/snows.php";  
 break;  
 }  
 }else{  
 require "view/snows.php";  
 }  
}

Nous ajouterons les liens Ajout, Supprimer, Modifier (ici icones fontawesome)



# CREATE

Dans un premier temps vous pouvez créer la vue addSnow.php et les fonctions pour l’ajout en vous basant sur la Partie 1 :



# DELETE

Le lien de chaque snow à supprimer va varier. On fait passer dynamiquement le code du snow en querystring :

http://localhost:81/index.php?action=deleteSnow&code=B101

http://localhost:81/index.php?action=deleteSnow&code=B126

…

Ajoutez ce lien dans la vue de gestion des snows.

Ajoutez la route deleteSnow

Ajoutez la fonction suivante au controleur :

function deleteSnow($snow\_code){  
 require\_once "model/snowsManager.php";  
 if (deleteASnow($snow\_code)){  
 $\_GET['action'] = "deleteSuccess";  
 $snowsResults = getSnows();  
 require "view/snowsSeller.php";  
 }else{  
 $\_GET['addSnowError'] = true;  
 require "view/snowsSeller.php";  
 }  
}

Dans snowsManager.php créez la fonction deleteASnow qui contiendra la requête :

DELETE FROM snows WHERE (`code`) LIKE ( !? à vous de compléter !? )';

# MODIFIER : EDIT ET UPDATE

La partie modifier se décompose en fait en 2 parties :

Edit : Affiche le formulaire (même qu’ajout) avec les valeurs du snow sélectionné (value).

Update : Récupère les valeurs du form et modifie l’enregistrement de la DB (UPDATE)

EDIT :

Sur la page de gestion des snows le lien de chaque snow à editer va varier. On fait passer dynamiquement le code du snow en querystring :

http://localhost:81/index.php?action=editSnow&code=B101

http://localhost:81/index.php?action=editSnow&code=B126

…

Ajoutez la route editSnow

Ajoutez au controleur une fonction *editSnow* qui chargera la vue editSnow.php  :

function editSnow($snow\_code){  
 require\_once "model/snowsManager.php";  
 $snowsResults= getASnow($snow\_code);  
 require "view/editSnow.php";  
  
}

Modifiez si nécessaire la fonction *getASnow* qui devra aussi séléctionner le champ *active*

$snowQuery = 'SELECT code, brand, model, snowLength, dailyPrice, qtyAvailable, description, photo, active FROM snows WHERE code=' . $strSeparator . $snow\_code . $strSeparator ;

Créez la vue editSnow.php en copiant addSnow.php, affichez les variables du snow sélectionné dans l’atribut value du champ de formulaire correspondant. Pour ceci utilisez le même système que dans la vue aSnow.php

Le formulaire sera finalement envoyé a la route : index.php?action=updateSnow

UPDATE :

Ajoutez la fonction *updateSnow* au controleur. Elle sera similaire à la fonction *addSnow*.

Modifiez le nom de la fonction du modèle appelée (updateASnow au lieu addNewSnow)

if (updateASnow($code, $brand, $model, $snowLength, $qtyAvailable, $description, $dailyPrice, $photo, $active)){…}

Dans le modele *snowsManager.php* ajoutez la fonction *updateASnow*

Cette fonction est similaire à la fonction addNewSnow, il faudra principalement modifier la requête SQL pour faire un UPDATE :

$updateSnowQuery = 'UPDATE snows SET code= ' . $strSeparator . $code.$strSeparator  
 . ', brand=' . $strSeparator . $brand . $strSeparator  
 . ', model=' . $strSeparator. $model .$strSeparator  
 . ', snowLength=' . $strSeparator. $snowLength .$strSeparator  
 . ', qtyAvailable=' . $strSeparator. $qtyAvailable .$strSeparator  
 . ', description=' . $strSeparator. $description .$strSeparator  
 . ', dailyPrice=' . $strSeparator. $dailyPrice .$strSeparator  
 . ', photo=' . $strSeparator. $photo .$strSeparator  
 . ', active=' . $strSeparator. $active .$strSeparator  
 .' WHERE code LIKE '.$strSeparator.$code.$strSeparator;