**Category**

**Source Code and Results:**

By Robera Worku and Belayneh Mathewos

1.itemClass

This takes a posted item details and inserts onto database. Retrieves from a database and

<?php

class itemClass{

public $category;

//a method to post the items

public function postItemMethod(){ //to get an item details from posting page.

$category=new category();

global $pdo;////global variable db to store the pdo object created on the db connector

$postFormSubmitted=isset($\_POST['post']);//checks if the form has been submitted

if($postFormSubmitted){

//assign values to the variables to be posted on to the db

$itemName=$\_POST['itemName'];

$itemImage=$\_POST['itemImage'];

$description=$\_POST['description'];

$availability=$\_POST['availability'];

$quantity=$\_POST['quantity'];

$price=$\_POST['price'];

if(isset($\_POST['new\_category'])){

Category::addCategory($\_POST['new\_category']);

$cat = $pdo->lastInsertId();

}

else

$cat = $\_POST['category'];

$catName=Category::getCatName($cat);

$userId=$\_SESSION['user']->getUserId();

// a query to be executed with place holders in it....for secuirity purposes..

$sql="INSERT INTO items(item\_name,description,post\_date,contract\_period,quantity,price,category, uploader\_id) VALUES

(:itemName,:itemDescription,CURRENT\_TIMESTAMP,:contractPeriod,:quantity,:price,:cat, :u)";

try{

$statement=$pdo->prepare($sql);//prepares the query to be executed and stores it on the variable statement

//binding values to the place holders in the query statement

$statement->bindValue(':itemName',$itemName);

//$statement->bindValue(':itemImage',$itemImage);

$statement->bindValue(':itemDescription',$description);

$statement->bindValue(':contractPeriod',$availability);

$statement->bindValue(':quantity',$quantity);

$statement->bindValue(':price',$price);

$statement->bindValue(':cat',$catName);

$statement->bindValue(':u', $userId);

//$statement->bindValue(':userId',$userId);

$statement->execute();

$id = $pdo->lastInsertId();

header('Location: '.getRootPath().'detail\_page.html.php?id'.$id, true, 303);

}

catch(Exception $e){

$error=$e->getMessage();

echo 'Couldnt insert item......sql error, <br>'.$error;//display the error occured

}

}

}

public static function fetchItems($categoryName){//To retreive an items from database.

global $pdo;

try{

$sql="SELECT \* FROM items WHERE Category=:categoryName";

$statement=$pdo->prepare($sql);

$statement->bindValue(':categoryName',$categoryName);

$statement->execute();

}

catch(Exception $e){

echo 'sql error'. '<br>'.$e->getMessage();

}

return $statement;

}

public static function displayItems($name){

$categoryName=$name;

$items=self::fetchItems($categoryName);

$display="<ul>";

while($row=$items->fetch()){

$display.="<form action='' method='post'>

<li>

<img src=''/>

<p><a href='".getRootPath()."php/pages/item/item\_info.html.php?cid=".Category::getCatId($name)."&id=".$row['item\_id']."'>".$row['item\_name']."</a></p>

<a href='".getRootPath()."php/payment/add\_to\_cart.php?id=".$row['item\_id']."&cid=".Category::getCatId($name)."' method='post'>

<input class='button rounded primary' value='Add To Cart' style='width:120px'>

</a>

</li>

</form>

";

}

return $display;

}

}

**2.Category Class**

Checks if category name exists in a category list and if not it adds to category list in database.

<?php

require\_once $\_SERVER['DOCUMENT\_ROOT'].'/shola/php/includes/helpers\_inc.php';

require\_once $\_SERVER['DOCUMENT\_ROOT'].'/shola/php/includes/db\_inc.php';

class Category{

public static function categoryExists($categoryName){ //checks if category is already exists in category list

global $pdo;

$sql="SELECT count(\*) FROM category WHERE category\_name=:cn";

$statement=$pdo->prepare($sql);

$statement->execute(array(":cn"=>strtolower($categoryName)));

return $statement->fetch()[0]>=1;

}

public static function addCategory($categoryName){ )//it checks and adds into category list

global $pdo;

if(self::categoryExists(strtolower($categoryName))){

return false;

}

else{

try{

$sql="INSERT INTO category(category\_name) VALUES('$categoryName')";//it adds into category list

$statement=$pdo->prepare($sql);

$statement->execute();

return true;

}

catch(Exception $e){

return false;

}

}

}

public static function getCategoryList(){

global $pdo;

$sql = "SELECT category\_name, id from category ORDER BY category\_name";

$stm = $pdo->prepare($sql);

$stm->execute();

$result = $stm->fetchAll(PDO::FETCH\_ASSOC);

return $result;

}

public static function getCategoryCount(){

global $pdo;

$sql = "SELECT count(\*) FROM category";

$stm = $pdo->prepare($sql);

$stm->execute();

return $stm->fetch()['count(\*)'];

}

public static function getCatName($cat\_id){

global $pdo;

$sql = "SELECT category\_name from category WHERE id = :i";

$stm = $pdo->prepare($sql);

$stm->execute(array(":i"=>$cat\_id));

return $stm->fetch()['category\_name'];

}

public static function getCatId($cat\_name){

global $pdo;

$sql = "SELECT id from category WHERE category\_name = :i";

$stm = $pdo->prepare($sql);

$stm->execute(array(":i"=>$cat\_name));

return $stm->fetch()['id'];

}

public static function getItemsCategory($item\_id){

global $pdo;

$sql = "SELECT category from items WHERE item\_id = :i";

$stm = $pdo->prepare($sql);

$stm->execute(array(":i"=>$item\_id));

return $stm->fetch()['category'];

}

}

?>

**3.Form to upload an item**

Form for a user to upload his item.

<body>

<div class ="bg-white" >

<form id="cat\_form" action="<?php echo $\_SERVER['PHP\_SELF'];?>" method="post">//prepares the form for item details will be posted

<div>

<label for="itemName">Item Name: </label>

<input type="text" name="itemName" id="itemName" required/>

</div>

<div>

<label for="itemImage">image: </label>

<input type="text" name="itemImage" id="itemImage" required/>

</div>

<div>

<label for="description">Description: </label>

<input type="text" name="description" id="description" required/>

</div>

<div>

<label for="availability">How many days would you like your item be available: </label>

<input type="text" name="availability" id="availability" required/>

</div>

<div>

<label for="quantity">Quantity: </label>

<input type="text" name="quantity" id="quantity" required/>

</div>

<div>

<label for="price">Price: </label>

<input type="text" name="price" id="price" required/>

</div>

<div id="cat\_inp\_div">

<select id="select\_cat" name="category" required>

<?php

foreach($category\_list as $c)

echo '<option value="'.$c['id'].'">'.$c['category\_name']."</option>";

?>

</select>

<button type="button" id="gen\_but" onclick="newCat()">Other</button>

</div>

<div>

<input type="checkbox" name="chatbotset" id="chatbotset" />Do you want to create a chat bot for this product?

</div>

<div>

<input type="submit" value="Post" name="post"/>

</div>

</form>

<a href="<?php echo getRootPath()?>"><button>Cancel</button></a>

</div >

<script>

function newCat(){

var form\_in = document.getElementById("cat\_inp\_div");

var inp = document.createElement("input");

inp.type="text"; inp.id="inp\_category"; inp.placeholder="Enter Category";inp.name="new\_category";

form\_in.appendChild(inp);

var sel\_form = document.getElementById("select\_cat");

sel\_form.style = "display: none";

var but = document.getElementById("gen\_but");

but.style = "display: none";

console.log(but);

}

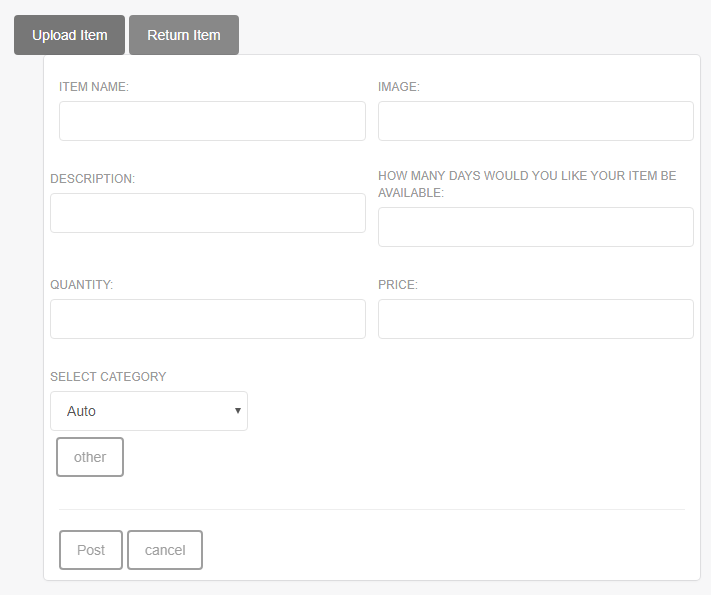
</script>

<?php

require\_once $\_SERVER['DOCUMENT\_ROOT'].'shola/php/pages/footer.html.php';

?>

The form looks like the following.



**4. display an items**

This retrieves from database and then display an items

<?php

require\_once "../includes/helpers\_inc.php";

include\_once ROOT\_DIR.'php/classes/item.class.php';

$retrievedItems=new ItemClass();

$items=$retrievedItems->fetchItems($categoryName);//calls a method in itemClass to fetch an items from a database.

$display="<ul>";

while($row=$items->fetchObject()){

$display.="<form action='' method='post'>

<li>

<img src='$row->ItemImage'/>

<p><a href='someurl'>$row->Itemname</a></p>

<input type='submit' value='Buy Now' name='buy' />

</li>

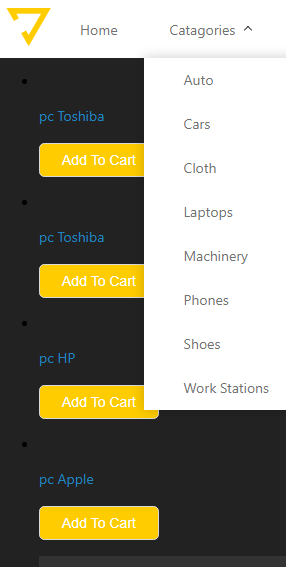
</form>

";

}

return $display;

Result of displaying an items.



**5.**