1. Create an Add movie page

<?php

class Moviesform\_Controller extends Base\_Controller

{

public function action\_index()

{

$record = new Movies();

$record->title = $\_POST['title'];

$record->year = $\_POST['year'];

$record->save();

$array = array('success' => 'true', 'msg' => 'Record added successfully');

$json = json\_encode($array);

return $json;

}

}

**<?php**

class Movies

{

static function getMovie($id)

{

$sql = 'SELECT mov\_id,mov\_title,mov\_description\_short,mov\_description\_long,mov\_director,mov\_year,mov\_image FROM movie WHERE mov\_id=:id';

$sth = dbConnection()->prepare($sql);

$sth->execute(array(':id' => $id));

$fetch = $sth->fetch();

$movie = new Movie($fetch['mov\_id'], $fetch['mov\_title'], $fetch['mov\_description\_short'], $fetch['mov\_description\_long'], $fetch['mov\_director'], $fetch['mov\_year'], $fetch['mov\_image']);

return $movie;

}

static function getMovies()

{

$sql = 'SELECT mov\_id FROM movie;';

$sth = dbConnection()->prepare($sql);

$sth->execute();

$fetch = $sth->fetchAll();

$i = 0;

$movies = array();

foreach ($fetch as $row) {

$movies[$i] = Movies::getMovie($row['mov\_id']);

$i++;

}

return $movies;

}

static function editMovie($m)

{

if ($m->title != '') {

$sql = 'UPDATE Movie SET mov\_title=:title, mov\_description\_short=:short\_description, mov\_description\_long=:long\_description, mov\_director=:director, mov\_year=:year, mov\_image=:image WHERE mov\_id=:id';

$sth = dbConnection()->prepare($sql);

$sth->execute(array(':id' => $m->id, ':title' => $m->title, ':short\_description' => $m->short\_description, ':long\_description' => $m->long\_description, ':director' => $m->director, ':year' => $m->year, ':image' => $m->image));

$fetch = $sth->fetch();

return $fetch;

} else {

return false;

}

}

static function insertMovie($m)

{

if ($m->title != '') {

$sql = 'INSERT INTO Movie(mov\_title,mov\_description\_short,mov\_description\_long,mov\_director,mov\_year,mov\_image) VALUES (:title,:short\_description,:long\_description,:director,:year,:image)';

$sth = dbConnection()->prepare($sql);

$sth->execute(array(':title' => $m->title, ':short\_description' => $m->short\_description, ':long\_description' => $m->long\_description, ':director' => $m->director, ':year' => $m->year, ':image' => $m->image));

$fetch = $sth->fetch();

return $fetch;

} else {

return false;

}

}

static function deleteMovie($id)

{

$sql = 'DELETE FROM Movie WHERE mov\_id=:id;';

$sth = dbConnection()->prepare($sql);

$sth->execute(array(':id' => $id));

$fetch = $sth->fetch();

return $fetch;

}

static function insertImage($image)

{

if ($image["name"] == "") {

return true;

}

if ($image['error'] > 0) {

throw new Exception("Erreur lors du transfert.");

}

if ($image["type"] > 20000000) {

throw new Exception("Le fichier est trop volumieux.");

}

$extensions\_valides = array('jpg', 'jpeg', 'gif', 'png');

$extension\_upload = strtolower(substr(strrchr($image['name'], '.'), 1));

if (!in\_array($extension\_upload, $extensions\_valides)) {

throw new Exception("Le type du fichier choisit est incorrect.");

}

$nom = $image['name'];

$nom = "images/" . $nom;

$resultat = move\_uploaded\_file($image['tmp\_name'], $nom);

return $resultat;

}

}