Міністерство освіти і науки України  
Івано-Франківський національний технічний університет нафти і газу

Кафедра програмного   
забезпечення автоматизованих систем

Лабораторна робота №3

Розроблення адміністративної частини сайту (backend на PHP)

Виконав: ст. гр. ПІ-13-1

Колесник Андрій

Перевірив:   
Броновський І.В.

м. Івано-Франківськ

2016

**Мета:** Навчитися реалізовувати backend веб-сайтів за допомогою мови PHP.

**Завдання**: Створити адміністративну частину сайту з можливістю реєстрації користувачів в системі. Додати можливість додавання товарів (книжок, людей, згідно вашої теми). Додати можливість додавання, видалення, редагування. Організувати авторизацію на сайті за допомогою логіна і паролю.

**Виконання завдання:**

Для реалізації системи було обрано підхід на основі шаблону MVC.

Для реалізації MVC було розроблено набір класів, за допомогою яких можна реалізувати Контролери, Моделі та Представлення, а також здійснювати основні операції з базою даних без написання SQL-коду.

1) Клас Controller:

Даний клас являється батьківським для всіх контролерів, які будуть створені в системі.

Параметр view являє собою об’єкт класу View, за допомогою якого можна буде рендерити представлення.

Параметр params використовується для спрощення доступу до параметрів, які передаються разом із GET та POST-запитами на сервер.

<?php

class Controller {

public $view;

public $params;

function \_\_construct() {

// Gets parameters from URI and POST and stores them in the params array

$parts = parse\_url($\_SERVER['REQUEST\_URI'], PHP\_URL\_QUERY);

parse\_str($parts, $this->params);

if (count($\_POST) > 0) {

$this->params = array\_merge ($this->params, $\_POST);

}

$this->view = new View();

}

// Redirects user to page404

public function to404 () {

$this->view->generate('404.php');

die();

}

}

2) Клас View:

Даний клас використовується для рендеру сторінки в шаблоні або без нього.

<?php

class View

{

function generate($content\_view, $template\_view = null, $params = null)

{

if ($template\_view) {

include 'application/views/'.$template\_view;

} else {

include 'application/views/'.$content\_view;

}

}

}

3) Клас Model

Нащадки цього класу відображають сутності, які зберігаються в базі даних. Дозволяють здійснювати зв’язок із БД.

<?php

class Model {

protected static $table\_name = "";

const DBHOST = 'ec2-54-228-213-36.eu-west-1.compute.amazonaws.com';

const DBPORT = '5432';

const DBNAME = 'dcoqh2c16k4jpa';

const DBUSER = 'ukmubttubjowpc';

const DBPASSWORD = 'LxnVhxay4lMyTQF7GCLa7ZyCIr';

public static function connect () {

$connection = pg\_pconnect("host=".Model::DBHOST." port=".Model::DBPORT." dbname=".Model::DBNAME." user=".Model::DBUSER." password=".Model::DBPASSWORD);

if (!$connection) {

echo "\n\nFAILED TO CONNECT\n\n";

die("Connection failed: " . $conn->connect\_error);

}

return $connection;

}

public static function executeQuery ($query) {

$connection = Model::connect();

$result = pg\_query($connection, $query);

$result\_arr = array();

while ($row = pg\_fetch\_assoc ($result)) {

array\_push($result\_arr, $row);

}

pg\_close($connection);

return $result\_arr;

}

public static function find ($id) {

$sql = "SELECT \* FROM ".static::$table\_name." WHERE id=".$id." LIMIT 1";

$res = static::executeQuery($sql);

if (count($res) == 1) {

$res = $res[0];

}

return $res;

}

public static function where ($query\_str) {

$query = new Query(static::$table\_name);

$query->where($query\_str);

return $query;

}

public static function order\_by ($query\_str) {

$query = new Query(static::$table\_name);

$query->order\_by($query\_str);

return $query;

}

public static function exists ($query\_str) {

$query = new Query(static::$table\_name);

$result = $query->where($query\_str)->limit(1)->take();

return (count($result) > 0);

}

public static function insert (array $data) {

$connection = Model::connect();

return pg\_insert($connection, static::$table\_name, $data);

}

public static function update (array $data, array $where) {

$connection = Model::connect();

return pg\_update($connection, static::$table\_name, $data, $where);

}

public static function delete ($id) {

$connection = Model::connect();

return pg\_delete($connection, static::$table\_name, array("id" => $id));

}

}

4) Клас Query

Даний клас використовується для формування динамічних запитів до бази даних

<?php

class Query {

private $where;

private $limit;

private $order\_by;

private $model;

function \_\_construct ($model) {

$this->where = array();

$this->order\_by = array();

$this->model = $model;

$this->limit = 0;

}

public function where ($query) {

array\_push($this->where, (string)$query);

return $this;

}

public function order\_by ($query) {

array\_push($this->order\_by, (string)$query);

return $this;

}

public function limit ($limit) {

$this->limit = (string)$limit;

return $this;

}

public function take\_one () {

$this->limit = 1;

return $this->take();

}

public function take () {

// Default query beginning

$sql = "SELECT \* FROM ".$this->model;

// WHERE statement

if (count($this->where) > 0) {

$sql .= " WHERE";

for ($i = 0; $i < count($this->where); $i++) {

$sql .= " ".$this->where[$i];

if ($i != count($this->where) - 1) {

$sql .= " AND ";

}

}

}

// ORDER BY statement

if (count($this->order\_by) > 0) {

$sql .= " ORDER BY";

for ($i = 0; $i < count($this->order\_by); $i++) {

$sql .= " ".$this->order\_by[$i];

if ($i != count($this->order\_by) - 1) {

$sql .= ',';

}

}

}

// LIMIT statement

if ($this->limit > 0) {

$sql .= " LIMIT ".$this->limit;

}

#echo $sql;

// Getting the query result from db

$result = Model::executeQuery($sql);

// If user asked for just one row

if ($this->limit == 1) {

count($result) == 1 ? $result = $result[0] : $result = null;

}

return $result;

}

}

Для реалізації непрямого доступу до веб-сторінок було створено систему маршрутизації, яка дозволяє інкапсулювати реальну структуру веб-додатку та запити, за якими можна отримати доступ до веб-сторінок.

4) Клас Route.php

Метод start даного класу викликаєтсья коли сервер отримує запит. Він перевіряє правильність отриманого запиту та викликає action, який відповідає отриманому запиту.

<?php

class Route {

static function redirect($url, array $params = null, $statusCode = 303) {

$url = 'Location: ' . $url;

if ($params && count($params) > 0) {

$url .= '?';

foreach ($params as $key => $value) {

$url .= $key.'='.$value.'&';

}

$url = rtrim($url, '&');

}

header($url, true, $statusCode);

die();

}

static function include\_models () {

include "application/core/query.php";

foreach (glob("application/models/\*.php") as $filename) {

include $filename;

}

}

static function login\_user () {

session\_start();

$\_SESSION["user"] = User::get\_current\_user();

}

static function start() {

include "application/routes.php";

// Default controller and action

$controller\_name = 'Main';

$action\_name = 'index';

$url = strtok($\_SERVER["REQUEST\_URI"],'?');

if (array\_key\_exists($url, $routes)) {

$url = $routes[$url];

}

$route = explode('/', $url);

// Controller name

if (!empty($route[1])) {

$controller\_name = $route[1];

}

// Action name

if (!empty($route[2])) {

$action\_name = $route[2];

}

$controller\_name = $controller\_name.'\_Controller';

$controller\_file = strtolower($controller\_name).'.php';

$controller\_path = "application/controllers/".$controller\_file;

if(file\_exists($controller\_path)) {

include $controller\_path;

Route::include\_models();

Route::login\_user();

} else {

Route::ErrorPage404();

return;

}

$controller = new $controller\_name;

$action = $action\_name;

if(method\_exists($controller, $action)) {

$controller->$action();

} else {

Route::ErrorPage404();

}

}

static function ErrorPage404() {

include "application/controllers/err404\_Controller.php";

$err = "Err404\_Controller";

$errController = new $err;

$errController->view->generate('404.php');

}

}

4.1) Вміст файлу Routes.php:

Даний файлі оголошено асоціативний масив, в якому ключами є підтримувані системою запити від користувача, а значеннями – адреса дії(action), на яку потрібно переадресувати користувача.

<?php

$routes = array(

'/' => '/main/index',

'/index' => '/main/index',

'/login' => '/login',

'/login\_confirm' => '/login/confirm\_login',

'/logout' => '/login/logout',

'/signup' => '/registration/register',

'/edit\_user' => '/users/edit',

'/signup\_confirm' => '/registration/confirm\_registration',

'/new\_institution' => '/institutions/new\_institution',

'/edit\_institution' => '/institutions/edit',

'/delete\_institution' => '/institutions/delete',

'/add\_institution' => '/institutions/add\_institution',

'/user' => '/users/user',

'/institution' => '/institutions/institution',

'/add\_post' => '/posts/add\_post',

'/add\_rating' => '/rating/add\_rating',

'/remove\_post' => '/posts/remove\_post',

'/remove\_rating' => '/rating/remove\_rating',

'/search' => '/institutions/search',

# API

'/api/user' => '/api/user\_data',

'/api/institution' => '/api/institution\_data',

'/api/user/search' => '/api/find\_users',

'/api/institution/search' => '/api/find\_institutions',

'/api/institution/update' => '/api/update\_institution',

'/api/user/update' => '/api/update\_user',

'/api/institution/new' => '/api/new\_institution',

'/api/user/new' => '/api/signup',

);

5) Контролер Login

<?php

class Login\_Controller extends Controller {

function login() {

$this->view->generate('login.php', 'template.php');

}

function index() {

$this->login();

}

function confirm\_login() {

$user = User::where("email='".$this->params["email"]."' AND password='".$this->params["password"]."'")->take\_one();

if ($user) {

$\_SESSION["email"] = $this->params["email"];

$\_SESSION["password"] = $this->params["password"];

Route::redirect("/");

} else {

Route::redirect("/login");

}

}

function logout () {

$\_SESSION["user"] = null;

$\_SESSION["email"] = null;

$\_SESSION["password"] = null;

Route::redirect("login");

}

}

5) Контролер Login

Даний контролер реалізує функціонал авторизації користувачів у системі.

<?php

class Login\_Controller extends Controller {

function login() {

$this->view->generate('login.php', 'template.php');

}

function index() {

$this->login();

}

function confirm\_login() {

$user = User::where("email='".$this->params["email"]."' AND password='".$this->params["password"]."'")->take\_one();

if ($user) {

$\_SESSION["email"] = $this->params["email"];

$\_SESSION["password"] = $this->params["password"];

Route::redirect("/");

} else {

Route::redirect("/login");

}

}

function logout () {

$\_SESSION["user"] = null;

$\_SESSION["email"] = null;

$\_SESSION["password"] = null;

Route::redirect("login");

}

}

6) Контролер Registration

Даний контролер реалізує функціонал реєстрації користувачів у системі та зміни даних уже існуючих користувачів.

<?php

class Registration\_Controller extends Controller {

function index () {

$this->register();

}

function register () {

$this->view->generate('signup.php', 'template.php');

}

function confirm\_registration () {

$res = null;

if ($this->params["id"]) {

$updated\_user = User::find($this->params["id"]);

if (!isset($updated\_user)) {

die();

return;

}

if ($this->params["permissions"] && $\_SESSION["user"]["permissions"] == 'admin'

&& $updated\_user["id"] != $\_SESSION["user"]["id"]) {

$this->params["user"]["permissions"] = $this->params["permissions"];

}

$res = User::update($this->params["user"], array("id"=>$this->params["id"]));

} else if (!User::exists("email='".$this->params["user"]["email"]."'")) {

if ($this->params["user"]["contact\_phone"][0] != '+') {

$this->params["user"]["contact\_phone"] = '+'.$this->params["user"]["contact\_phone"];

}

$res = User::insert($this->params["user"]);

} else {

echo json\_encode(array('status'=>'error', 'error'=>'Користувач з такою електронною адресою вже існує'));

}

if (isset($res)) {

echo json\_encode(array('status'=>'success', 'redirect\_to'=>'/login'));

} else {

echo json\_encode(array('status'=>'error', 'error'=>pg\_last\_error()));

}

}

}

7) Контролер Institutions

Даний контролер реалізує функціонал додавання закладів, редагування та видалення вже існуючих закладів та пошуку.

<?php

class Institutions\_Controller extends Controller {

function index () {

$this->institution();

}

function institution () {

if (!array\_key\_exists("id", $this->params)) {

$this->to404();

}

try {

$institution = @Institution::find($this->params["id"]);

if (!$institution)

throw new Exception("Невірне ID", 1);

$this->params["institution"] = $institution;

$this->view->generate('institution.php', 'template.php', $this->params);

} catch (Exception $e) {

$this->to404();

}

}

function new\_institution () {

$this->view->generate('new\_institution.php', 'template.php');

}

function edit () {

$this->params["edit"] = true;

$this->params["institution"] = Institution::find($this->params["id"]);

$this->view->generate('new\_institution.php', 'template.php', $this->params);

}

function search () {

$this->params["query"] = mb\_strtolower($this->params["query"]);

$result = Institution::where("LOWER(name) LIKE '%".$this->params["query"]."%'")->take();

if ($result) {

$res ='';

foreach ($result as $key => $value) {

$res .= Institution::to\_html($value);

}

echo json\_encode(array('status'=>'success', 'result'=>$res));

} else {

echo json\_encode(array('status'=>'error', 'error'=>pg\_last\_error()));

}

}

function delete () {

$id = $this->params["id"];

$institution = Institution::find($id);

if ($id == $institution["account\_id"] || $\_SESSION["user"]["permissions"] == 'admin') {

Institution::delete($id);

}

Route::redirect("user?id=".$\_SESSION["user"]["id"]);

}

function add\_institution () {

if (array\_key\_exists("id", $this->params)) {

$res = Institution::update($this->params["institution"], array("id"=>$this->params["id"]));

if ($res) {

echo json\_encode(array('status'=>'success', 'redirect\_to'=>'institution?id='.$this->params["id"]));

} else {

echo json\_encode(array('status'=>'error', 'error'=>pg\_last\_error()));

}

} else if (!Institution::exists("name='".$this->params["institution"]["name"]."' AND city='".$this->params["institution"]["city"]."' AND address='".$this->params["institution"]["address"]."'")) {

$this->params["institution"]["account\_id"] = $\_SESSION["user"]["id"];

$res = Institution::insert($this->params["institution"]);

if ($res) {

echo json\_encode(array('status'=>'success', 'redirect\_to'=>'user?id='.$\_SESSION["user"]["id"]));

} else {

echo json\_encode(array('status'=>'error', 'error'=>pg\_last\_error()));

}

} else {

echo json\_encode(array('status'=>'error', 'error'=>'Такий заклад вже існує'));

}

}

}

**Висновок:** На цій лабораторній роботі було реалізовано backend сайту за допомогою мови PHP та принципів MVC.