Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г. Короленко»

**ОТЧЕТ**

**по учебной практике**

**ПМ 01 Разработка программных модулей программного обеспечения для компьютерных систем**

**Специальность 09.02.03 Программирование в компьютерных системах**

|  |
| --- |
| Выполнил(а): |
| студент(ка) ОП СПО группа 33 |
|  |
|  |
| Руководитель практики: ст. преподаватель кафедры математики и информатики Касаткин К.А. |

г. Глазов 2018

**Спецификации**

* Название приложения: Справочник владельца видеотеки.
* Создание базы данных «Справочник владельца видеотеки»
* Создание таблицы для базы данных «Справочник владельца видеотеки» основной информации с полями: название фильма, студия, жанр, режиссер, год выпуска, главная роль, оценка фильма, наличие.
* Создание таблицы для базы данных «Справочник владельца видеотеки» режиссер с полями: имя и фамилия, дата рождения.
* Создание таблицы для базы данных «Справочник владельца видеотеки» жанр: жанр.
* Создание таблицы для базы данных «Справочник владельца видеотеки» оценка фильма: оценка, код оценки на сайте «Кинопоиск».
* Для работы с базой данных нужно установить программное обеспечение MySQL (версия 5.1 или выше).

Исходный код программы

Файл: index.php

Место расположения: ../movie/api/

Назначение:

Выполняет подключение таблиц БД

<?php

ob\_start();

function \_\_autoload($class\_name)

{

require\_once dirname(\_\_FILE\_\_) . DIRECTORY\_SEPARATOR . 'classes' . DIRECTORY\_SEPARATOR . strtolower($class\_name) . '.class.php';

}

$db = new DB(Config::$dbHost, Config::$dbUser, Config::$dbPass, Config::$dbBase);

$request = $\_SERVER['REQUEST\_URI'];

$request = urldecode(trim($request, '/'));

$query = explode('?', $request);

$query = $query[1];

$queryData = explode('&', $request);

$query = array();

foreach ($queryData as $item) {

$item = explode('=', $item);

$query[$item[0]] = $item[1];

}

$request = explode('/', $request);

function \_view($clients) {

echo '<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>';

echo "<table class=\"table table-hover\">";

echo "<tr>";

foreach ($clients[0] as $key => $value) {

echo "<td>";

echo $key;

echo "</td>";

}

echo "</tr>";

foreach ($clients as $client) {

echo "<tr>";

foreach ($client as $value) {

echo "<td>";

echo $value;

echo "</td>";

}

echo "</tr>";

}

echo "</table>";

}

$data = array();

switch ($request[2]) {

case 'movies':

switch ($request[3]) {

case 'all':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `movies`");

break;

case 'get':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `movies` WHERE `ID` = " . (int)$db->escape($request[4]));

$data['data'] = $data['data'][0];

break;

case 'view':

$clients = $db->sql("SELECT \* FROM `movies`");

\_view($clients);

exit;

}

break;

case 'directorr':

switch ($request[3]) {

case 'all':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `directorr`");

break;

case 'get':

$data['type'] = 'success';

$d = $db->escape($request[4]);

$data['data'] = $db->sql("SELECT \* FROM `directorr` WHERE `directors` = '$d'");

$data['data'] = $data['data'][0];

break;

case 'view':

$clients = $db->sql("SELECT \* FROM `directorr`");

\_view($clients);

exit;

}

break;

case 'genress':

switch ($request[3]) {

case 'all':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `genress`");

break;

case 'get':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `movies` WHERE `genres` = " . $db->escape($request[4]));

$data['data'] = $data['data'][0];

break;

case 'view':

$clients = $db->sql("SELECT \* FROM `movies`");

\_view($clients);

exit;

}

break;

case 'studioo':

switch ($request[3]) {

case 'all':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `studioo`");

break;

case 'get':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `movies` WHERE `studios` = " . $db->escape($request[4]));

$data['data'] = $data['data'][0];

break;

case 'view':

$clients = $db->sql("SELECT \* FROM `studio`");

\_view($clients);

exit;

}

break;

case 'resultt':

switch ($request[3]) {

case 'all':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `resultt`");

break;

case 'get':

$data['type'] = 'success';

$data['data'] = $db->sql("SELECT \* FROM `resultt` WHERE `ID` = " . $db->escape($request[4]));

$data['data'] = $data['data'][0];

break;

case 'view':

$clients = $db->sql("SELECT \* FROM `Result`");

\_view($clients);

exit;

}

break;

default:

$data['type'] = 'error';

$data['desc'] = 'Bad Request';

}

header('Content-type: application/json');

echo json\_encode($data);

ob\_get\_contents();**Описание программы**

Веб-сайт «Справочник владельца видеотеки» может отображать: название, жанр, год, режиссера, главную роль, оценку, и наличие. Выводить информацию о конкретном выбранном фильме, режиссере, главных актерах и оценки фильма.

**Формуляр**

Операционная система:

* Windows® XP/Vista/7/ 8/10 32/64-bit

Системные требования

* Процессор: AMD Sempron 2200
* Память: минимальная ОЗУ системы — 100 MB
* Жесткий диск: 2100 об./мин., с 200 MB свободного места

**Описание применения**

Веб-сайт Справочник владельца видеотеки применяется в сфере хранения информации о фильмах. Будет хорошим дополнениям любому пользователю.

Классы решаемых задач: веб-сайт решает проблему храненияинформации о большом количестве фильмов.

Руководство системного программиста

Использовать операционная систему: Windows® Vista/7/ 8/10 64-bit с последним пакетом обновлений. Установить XAMPP. Подключить базу данных movie к MySQL Server с помощью HeidiSQL или MySQL Command. Запустить программу Справочник владельца видеотеки. Программа может запускаться при наличие программного обеспечения XAMPP.

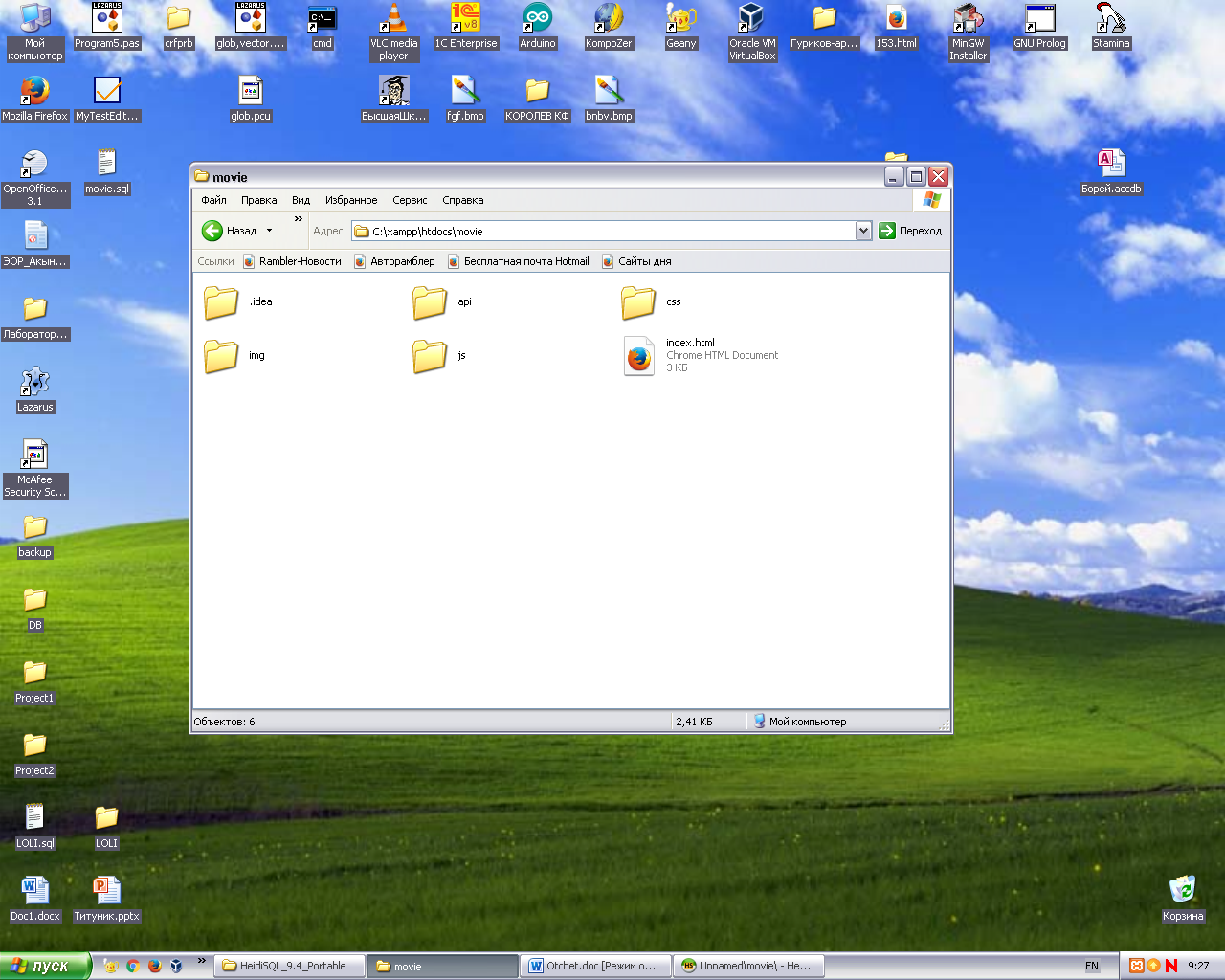


Рис.1. Файлы дистрибутива веб-сайта Записная книжка.

**Руководство программиста**

В случае замены базы данных (БД) нужно скачать исходный код с репозитория (https://github.com/HIKAMURACHI/DataBase). В запросах к базе данных нужно поменять поля, при этом сохраняя количество полей в запросе такое же, как количеством полей в базе данных. Можно в базу данных movie добавлять новые таблицы и поля. Уже в созданные таблицы можно добавлять новые поля. Модифицировать базу данных можно с помощью программы HeidiSQL.

**Руководство оператора**

Для добавления записи надо перейти в HeidiSQL и нажать добавить строку, после заполнить все поля.

**Описание языка**

Для создания программного продукта Записная книжка использовался язык программирования PHP и JavaScript.

**Руководство по техническому обслуживанию**

Единственное, что нужно выяснить это минимальные требования к компьютеру они были указаны в формуляре. Для тестирования компьютера можно использовать программу (AIDA), если ваш компьютер удовлетворяет минимальным требованиям, то можно использовать веб-сайт.

**Программа и методика испытания**

Тест план

Проверить все функции программы Справочник владельца видеотеки на корректность работы:

1. Просмотр записи

Чек лист: Добавление пользователя

1. Проверка корректности отображения данных в поле Название.
2. Проверка корректности отображения данных в поле Студия.
3. Проверка корректности отображения данных в поле Жанр.
4. Проверка корректности отображения данных в поле Год.
5. Проверка корректности отображения данных в поле Режиссер.
6. Проверка корректности отображения данных в поле Главная роль.
7. Проверка корректности отображения данных в поле Краткое содержание.
8. Проверка корректности отображения данных в поле Оценка.
9. Проверка корректности отображения данных в поле Наличие.

Тест case:Добавление пользователя

**Условия воспроизведения:**

1. **Установить XAMPP(MySQL)**
2. Запустить MySQL server
3. Запустить HeidiSQL

**Как воспроизвести:**

1. Нажать на кнопку Добавить строку.
2. Проверка корректности ввода данных в поле Название.
3. Проверка корректности ввода данных в поле Студия.
4. Проверка корректности ввода данных в поле Жанр.
5. Проверка корректности ввода данных в поле Год.
6. Проверка корректности ввода данных в поле Режиссер.
7. Проверка корректности ввода данных в поле Главная роль.
8. Проверка корректности ввода данных в поле Краткое содержание.
9. Проверка корректности ввода данных в поле Оценка.
10. Проверка корректности ввода данных в поле Наличие.

**Фактический результат:**

**Выдаётся сообщение об ошибке**

**Ожидаемый результат:**  
При некорректном вводе данных в поле выводилось уведомление об этом.

**Пояснительная записка**

**Введение**

**Название программы:** Справочник владельца видеотеки

**Назначение и область применения**

Программа Справочник владельца видеотеки применяется в сфере хранение информации о фильмах. Программа Справочник владельца видеотеки будет хорошим дополнение для любого пользователя интернета.

**Классы решаем задач: программа** Справочник владельца видеотеки **решает проблему поиска информации о имеющихся фильмах.**

**Технические характеристики**

1. Задачи: создать функционал для работы программы Справочник владельца видеотеки, создать базу данных, написать пользовательское приложение.
2. Алгоритмы: алгоритмы в данной программе не применялись, потому что все алгоритмы были реализованы разработчиками MySQL.
3. Описание и обоснованный выбор организации входных и выходных данных: выходные и входные данные хранятся в 5 таблицах базы данных movie.
4. Описание и обоснованный выбор технических и программных средств: PHP был выбран, потому что там есть компоненты для работы с базой данных, MySQL был выбран, потому что он свободно распространяемый и имеет широкий спектр запросов, MySQL workbench был выбран, потому что он свободно распространяемый и мало занимает места на диске и имеет понятный интерфейс.

Технико-экономические показатели.

Программа Справочник владельца видеотеки имеет преимущество перед ручным справочником владельца видеотеки тем, что действия автоматизированы, поиск данных станет легче, возможность формирования электронных документов.