|  |  |
| --- | --- |
|  | **ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ**  **«МОСКОВСКИЙ КОЛЛЕДЖ УПРАВЛЕНИЯ, ГОСТИНИЧНОГО БИЗНЕСА И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ «ЦАРИЦЫНО»** |

Отделение управления и информационных технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К курсовому проекту по разработке сайта видео хостинга любителей литературы «Litwatch»**

Выполнил студент:

Тихонов Иван Александрович

Номер группы: 20ИС-21

Специальность: 09.02.07

Руководитель курсовой работы:

Киселёва Светлана Владимировна

Дата сдачи работы: «26» апреля 2022г

|  |  |
| --- | --- |
| **Согласовано**  На заседании кафедры  информационных технологий  Протокол № \_\_\_\_\_  от «\_\_ »\_\_\_\_\_\_\_\_2021 г.  Заведующий кафедрой  \_\_\_\_\_\_\_\_\_\_\_ Е.Ф.Писчасова | **Утверждаю** Руководитель  учебно-методического отдела  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.  \_\_\_\_\_\_\_\_\_\_ Т.Н. Михайлова |

ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Студенту (ке) Тихонову Ивану Александровичу

Группы 20ИС-21 специальности 09.02.07 Информационные системы и программирование

Дата выдачи задания: « 21» марта  2022 г.

Срок сдачи проекта: « 28» апреля  2022 г.

**Тема курсового проекта:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Техническое задание**

**Исходные данные** Разработка информационной системы работы сайта литературного видео-хостинга "Litwatch"

**Задачи:** разработать информационную систему, включающую в себя базу данных, сайт, а также документацию к нему

Содержание пояснительной записки

|  |  |
| --- | --- |
| Раздел пояснительной записки | Объем, % |
| Введение | 1% |
| 1. Обзорно-постановочная часть:  * Анализ предметной области; * Обоснование необходимости использования веб-приложения для решения задачи; * Постановка цели и задач веб-приложения; * Анализ существующих разработок; * Обоснование проектных решений | 10% |
| 1. Раздел проектирования веб-приложения:  * Анализ содержания проекта; * Проектирование базы данных (определение множества сущностей, построение модели уровня сущностей, построение модели уровня ключей, построение полноатрибутной модели); * Проектирование информационной архитектуры сайта; * Разработка дизайн-макета сайта | 20% |
| 1. Раздел реализации приложения:  * Разработка интерфейса пользователя (создание семантической разметки по макету, позиционирование элементов на странице, верстка текстового содержания, верстка навигационных элементов); * Реализация логики работы веб-приложения (извлечение, ввод и изменение данных в бд, операторы и функции, интерфейс с php, администрирование и безопасность);   1. Реализация регистрации.   2. Реализация авторизации. | 67% |
| 1. Раздел описания и тестирования приложения |  |
| Заключение | 1% |
| Список использованной литературы | 1% |

Графическая часть:

Выносной лист 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выносной лист 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выносной лист 3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Презентация.

Дипломник: /                   /

Руководитель проекта                                         /                             /

Заведующий кафедрой:                              /

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Разраб.

Тихонов И. А.

Пров.

Реценз.

Киселёва С. В.

*Синицын*

Н. контр.

Утв.

Лит.

Листов

**ГБПОУ Колледж «Царицыно»**

Оглавление

[Введение 10](#_Toc101901260)

[Раздел 1. Обзорно-постановочная часть 11](#_Toc101901261)

[1. Анализ предметной области 11](#_Toc101901262)

[2. Обоснование необходимости и цели использования системы для решения задачи. 11](#_Toc101901263)

[3. Постановка цели и задач веб-приложения 12](#_Toc101901264)

[4. Анализ существующих разработок 12](#_Toc101901265)

[5. Обоснование проектных решений 12](#_Toc101901266)

[Раздел 2. Раздел проектирования веб-приложения 13](#_Toc101901267)

[1. Анализ содержания проекта 13](#_Toc101901268)

[2. Проектирование базы данных 13](#_Toc101901269)

[3. Проектирование информационной архитектуры сайта 14](#_Toc101901270)

[4. Разработка дизайн-макета сайта 14](#_Toc101901271)

[Раздел 3. Раздел реализации приложения 17](#_Toc101901272)

[1. Разработка интерфейса пользователя 17](#_Toc101901273)

[2. Логика работы веб-приложения 18](#_Toc101901274)

[2.1 Логика регистрации. 18](#_Toc101901275)

[2.2 Логика авторизации. 21](#_Toc101901276)

[Раздел 4. Раздел описания и тестирования приложения 22](#_Toc101901277)

[1. Реализация регистрации 22](#_Toc101901278)

[2. Реализация авторизации 25](#_Toc101901279)

[3. Реализация кабинета пользователя 26](#_Toc101901280)

[4. Реализация добавления видео 28](#_Toc101901281)

[5. Реализация видеоплеера 31](#_Toc101901282)

[5.1 Загрузка страницы 31](#_Toc101901283)

[5.2 Алгоритм оценки. 34](#_Toc101901284)

[5.3 Алгоритм добавления комментария 36](#_Toc101901285)

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное образовательное учреждение города Москвы «Московский колледж управления, гостиничного бизнеса

и информационных технологий «Царицыно»

(ГБПОУ Колледж «Царицыно»)

Отделение управления и информационных технологий

ОТЗЫВ НА КУРСОВОЙ ПРОЕКТ

Студент (Ф.И.О.): Тихонов Иван Александрович

  Курс: II Группа: 20ИС-21

  Специальность: 09.02.07 Информационные системы и программирование

**ПМ.**: ПМ.05 Проектирование и разработка информационных систем

***МДК.:*** МДК.05.02 Разработка кода информационных систем

**Тема:**Разработка информационной системы работы сайта литературного видео-хостинга "Litwatch"

***Критерии, при наличии хотя бы одного (n1-4) или двух (п. 5-6) из которых работа оценивается «неудовлетворительно»***

|  |  |  |
| --- | --- | --- |
| ***№***  ***пп/п*** | Наименование показателей | Отметка о несоответствии |
| 1 | Тема и (или) содержание работы не относится к предмету дисциплины |  |
| 2 | Работа перепечатана из Интернета или из других информационных источников |  |
| 3 | Не структурированный план курсовой работы |  |
| 4 | Объем работы менее 25 листов машинописного текста |  |
| 5 | В работе отсутствуют ссылки сноски на нормативные и другие источники |  |
| 6 | Оформление курсовой работы не соответствуют требованиям |  |

***Рейтинг работы (при неудовлетворительной оценке не заполняется)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***№***  ***п/п*** | ***Наименование показателей*** | ***Оценка*** | | |
| ***3***  ***«удовл»*** | ***4***  ***«хор»*** | ***5***  ***«отл»*** |
| 1 | Степень раскрытия темы |  |  |  |
| 2 | Полнота охвата научной литературы |  |  |  |
| 3 | Использование нормативных актов |  |  |  |
| 4 | Индивидуальный подход к написанию КР |  |  |  |
| 5 | Последовательность и логика изложения материала |  |  |  |
| 6 | Качество оформления, язык, стиль и грамматический уровень работы |  |  |  |
| 7 | Использование иллюстрационного материала (рисунки, таблицы, графики, диаграммы и т.п.) |  |  |  |

Дополнительные замечания\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Итоговая оценка (прописью)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

      Руководитель курсовой работы

  Киселева С.В.                                             \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                          Ф.И.О.                                                                                    /Подпись/                                                                   **«**28**»** апреля 2022г.

# Введение

Литературное искусство сопровождало человечество на протяжении всего его развития. В данный момент, к классической литературе причисляют не только произведения, написанные в XVI веке, как например «Ромео и Джульетта», но и произведения, написанные относительно недавно, вроде «Мастера и Маргариты».

Однако в последнее время спрос на литературу значительно упал. Спрос на художественную литературу за 10 лет с 2010 года упал на 25%. Читателей становиться всё меньше, т. к. населению слишком привыкло к динамическим социальным сетям, чтобы уследить за статичным текстом.

Тем не менее, именно современные средства могут помочь вывести литературу из «читательского кризиса». Если получится совместить литературную тематику и современные сети и привлечь внимание пользователей к такому продукту, то пользователи начнут больше интересоваться книгами.

Именно таким продуктом является сайт видео хостинг «Litwatch». Проект создаётся с целью привлечения внимания к литературе пользователей интернета.

# Раздел 1. Обзорно-постановочная часть

## Анализ предметной области

Как и было сказано выше, в качестве предметной области выступает фирма «Литнесс». Фирма занимается продажей книг на различную тематику на различных языках, которые покупаются или заказываются книги за границей, привозятся в Россию и продаются в книжных магазинах по всей стране. Главная задача фирмы-заработок, путем продажи книг, выявление интересов в определенном регионе и составление статистики по регионам. Для привлечения внимания существует рекламный отдел, который собирал информацию о спросе потребителей путем просмотра отзыва о книгах в интернете, а также опрашивал людей. В конце каждого месяца отдел составлял статистику по интересам, на основе которых закупались книги.

Предметная область проекта «Литвотч» обладает рядом особенностей:

* Литературная направленность
* Работа с поставками за пределами России
* Работа со статистическими данными

## Обоснование необходимости и цели использования системы для решения задачи.

Проект разрабатывается для привлечения потенциальных покупателей к сети магазинов «Литнесс» и к литературе в целом, а также выявления их интересов.

Эта информационная система разрабатывается для дополнения рекламного отдела фирмы, которая имела следующие недостатки:

* Низкий спрос на книжную продукцию в определенных областях.
* Неточная статистика спроса потребителей, влияющая на продажи.
* Низкая оперативность отдела, снижающая качество информации.
* Сложность сбора информации, влияющая на статистику

Также этот отдел использовал для отчётов необработанные результаты опросов, что является неудобным способом передачи информации между отделами.

## Постановка цели и задач веб-приложения

Целью создания системы может служить:

* Привлечение внимания к фирме «Литнесс»
* Повышение качества и точности информации.
* Упрощение сбора информации.

Назначением системы может являться:

* Сбор информации о спросе на книжную продукцию
* Составление статистики на основе полученных данных и предоставления статистики администратору.

## Анализ существующих разработок

Одним из немногих приложение подобного рода является видео хостинг «You-Tube». Данный продукт позволяет пользователям:

* регистрировать аккаунты
* создавать и публиковать видео
* Оценивать и комментировать уже имеющиеся видео

## Обоснование проектных решений

Для создания системы используются следующие средства:

* Локальный сервер «OpenServer»
* СУБД «phpmyadmin»
* Среда разработки «Visual Studio Code»

Данные средства являются наиболее доступными, а также предоставляют достаточно широкий набор инструментов для создания базы данных (далее БД) и веб-приложения, которые необходимы для создания и реализации системы.

# Раздел 2. Раздел проектирования веб-приложения

## Анализ содержания проекта

Проект должен включать в себя:

* Базу данных, хранящую информацию о пользователях и их действиях
* Полностью стилизованный сайт, дающий возможность регистрироваться новым и авторизоваться зарегистрированным пользователям.
* Документация к проекту

## Проектирование базы данных

База данных должна включать в себя следующие сущности:

* Пользователь
* Видео
* Комментарий

Таким образом, получается следующая матрица связей:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | E1/Users | E2/Videos | E3/Comments | E4/Mark |
| E1/Users | X | R1 | R2 | R3 |
| E2/Videos | R1 | X | R4 | R5 |
| E3/Comments | R2 | R4 | X |  |
| E4/Mark | R3 | R5 |  | X |

Таблица 1. Матрица связей.

Все эти сущности взаимосвязаны между собой (Неавторизованный пользователь не может загрузить видео или оставить комментарий, а комментарий нельзя оставить к несуществующему видео).

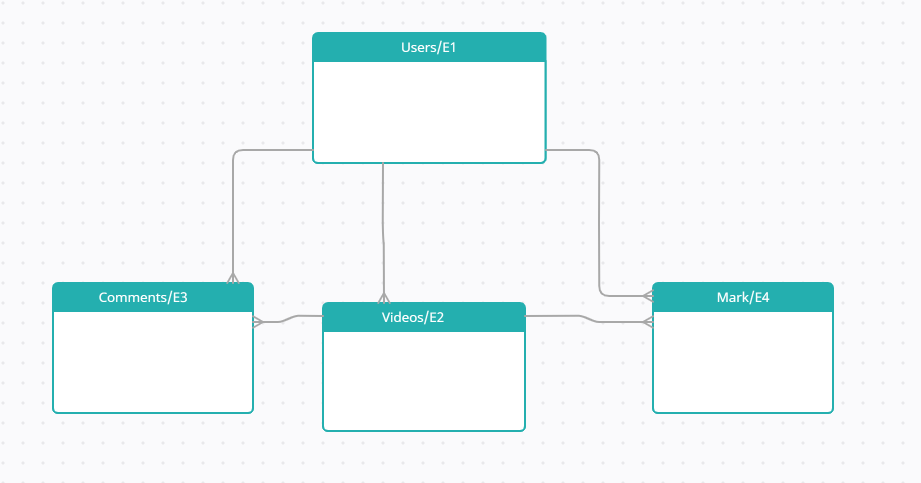


Рисунок 1. Диаграмма уровня сущностей.

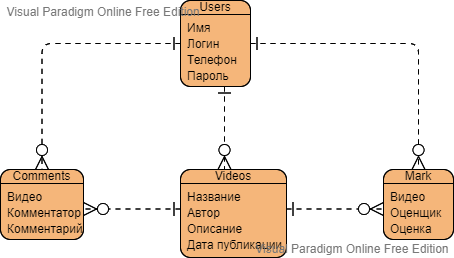


Рисунок 2. Диаграмма уровня атрибутов.

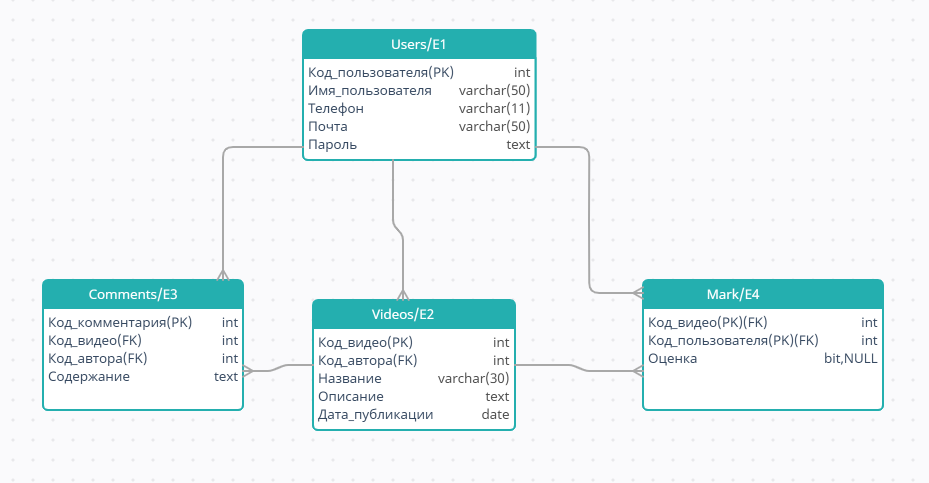


Рисунок 3. полно атрибутная диаграмма

После построения моделей БД, можно приступать к её созданию. Для этого воспользуемся СУБД «PhpMyAdmin».

## Проектирование информационной архитектуры сайта

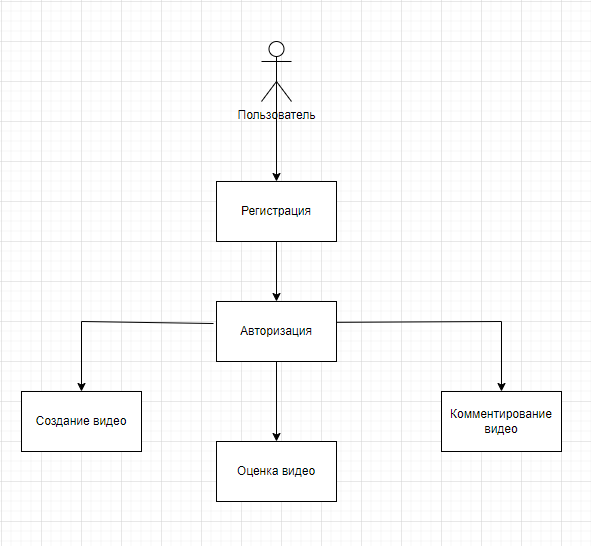
Пользователь должен иметь самые необходимые функции для взаимодействия с системой. На рисунке 2 представлены все возможности пользователя.

Рисунок 4. Возможности пользователя

## Разработка дизайн-макета сайта

Для разработки дизайн-макета подойдёт сервис «Figma», позволяющий без особых проблем создать макет любой сложности и экспортировать его. На рисунке 3 представлен макет данного проекта.

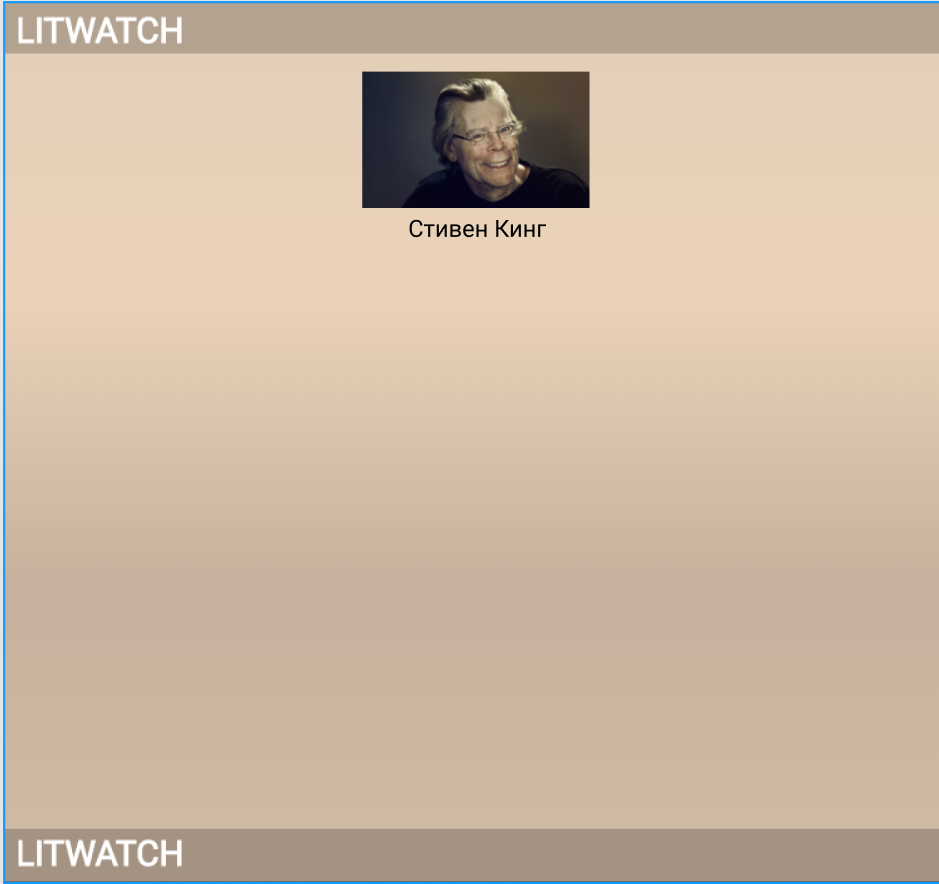


Рисунок 5. Макет-дизайн главной страницы

# Раздел 3. Раздел реализации приложения

## **Разработка интерфейса пользователя**

Для начала необходимо определиться с общей структурой страниц. Наиболее подходящей является следующая структура:

<header> - Шапка страницы, идентична для всех страниц

<main> - Главная часть страницы, основное содержимое

<footer> - Подвал страницы, идентичен для всех страниц

После необходимо определиться со стилизацией сайта. Для удобства вынесем стили в отдельный файл «style.css». Основной шрифт сайта – Roboto, без засечек, 16px. На заднем фоне - градиент цветов следующих кодировок: #e2ceb8, #ebd3b9, #c5b19d, #d1bda6. Далее определимся с размещением основных составляющих сайта. Для этого стоит использовать grid, чтобы в процентном соотношении установить их размеры относительно всей страницы. И, наконец, панель навигации. Её нужно вставить в шапку сайта и, используя flex-контейнер, растянуть по горизонтали с одинаковыми пробелами вокруг flex-элементов, предварительно отредактировав ссылки a. Итоговая разметка представлена на рисунках 4-6.

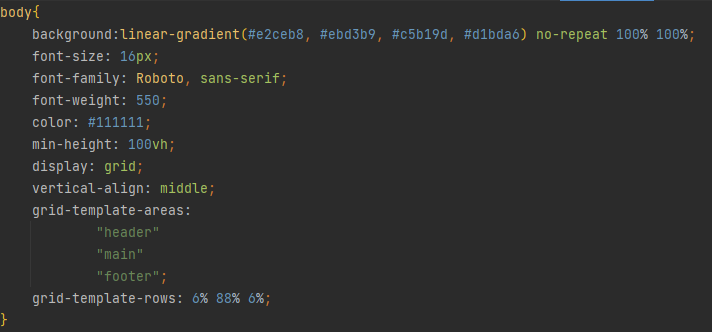


Рисунок 6. Стилизация тела страницы

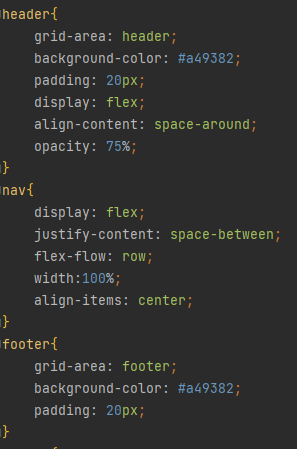


Рисунок 7. Стилизация шапки, панели навигации и подвала

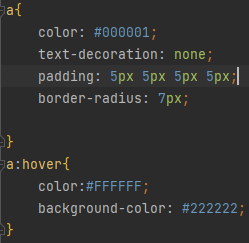


Рисунок 8. Стилизация ссылок

## Логика работы веб-приложения

### 2.1 Логика регистрации.

После стилизации страниц, можно приступать к созданию формы регистрации и авторизации. Для регистрации нового пользователя, понадобиться ввести следующие данные:

* Имя Пользователя
* Телефон
* Почта
* Пароль

При заполнении всех данных (Пользователь не может отправить данные, не заполнив все поля) и нажатии кнопки «Регистрация» пользователь должен быть автоматически зарегистрирован и авторизирован. Алгоритм выглядит следующим образом:

* Отправка введённых данных на проверку, отмена в случае ошибки и вывод сообщения
* Запись данных в БД
* Сохранение имени текущего пользователя в сессии
* Возращение на главную страницу

Перед отправкой данных сайт должен их проверить. Параметры проверки таковы:

* Имя пользователя не занято
* Телефон введён корректно
* Почта пользователя не занята
* Введённые пароли совпадают

В соответствии с каждым пунктом на языке php реализована проверка:

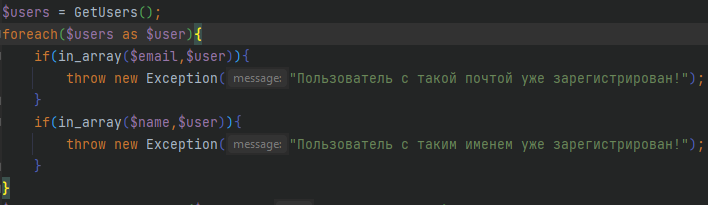


Рисунок 13. Проверка почти и имени.

Представленный алгоритм сравнивает почту и имя нового пользователя со старыми пользователями. Если находится совпадение, срабатывает исключение, которое будет обработано позже.

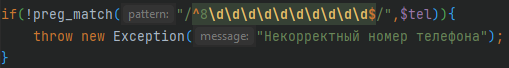


Рисунок 14. Проверка телефона.

На рисунке 12 производится проверка телефона на соответствие шаблону. Если телефон не проходит проверку, срабатывает исключение.

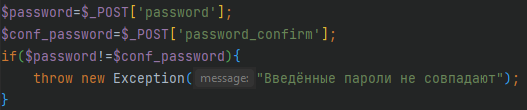


Рисунок 15. Проверка паролей

При проверке паролей просто сравниваются два поля – пароль и подтверждение пароля. Если поля не совпадают, срабатывает исключение.

После всех проверок, можно отправить данные в БД, в таблицу Пользователи, а также запустить сессию с именем пользователя.

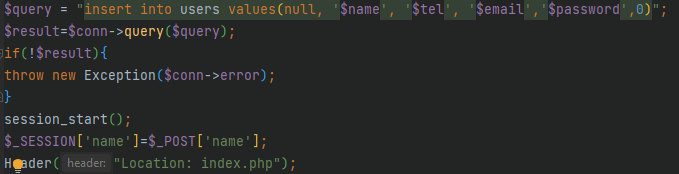


Рисунок 16. Отправка данных и начало сессии

Однако, если в процессе обработки возникли проблемы, в конце кода находится обработчик исключений, который получает сообщение об ошибке и перенаправляет пользователя на специальную страницу «Error.php», выводящую ошибку пользователю.

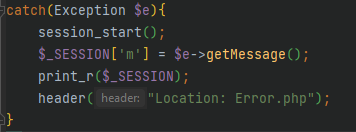


Рисунок 17. Обработка исключений

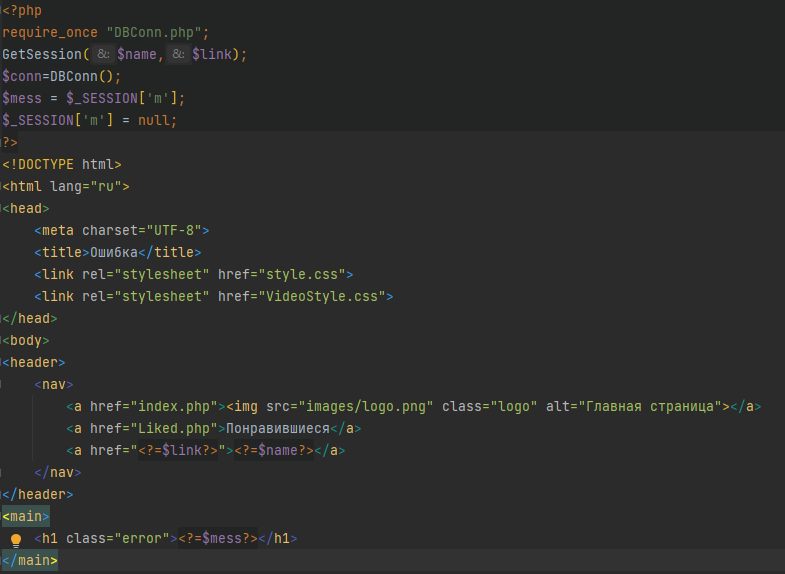


Рисунок 18. Error.php

Для упрощения доступа к БД создан файл “DBConn.php”, который содержит в себе функции, отвечающие за:

* Создание соединения с БД
* Получение почты и телефона пользователя
* Обработки сессии
* Получение имени и почты всех пользователей

На рисунке 9 представлен полный файл “DBConn.php”.

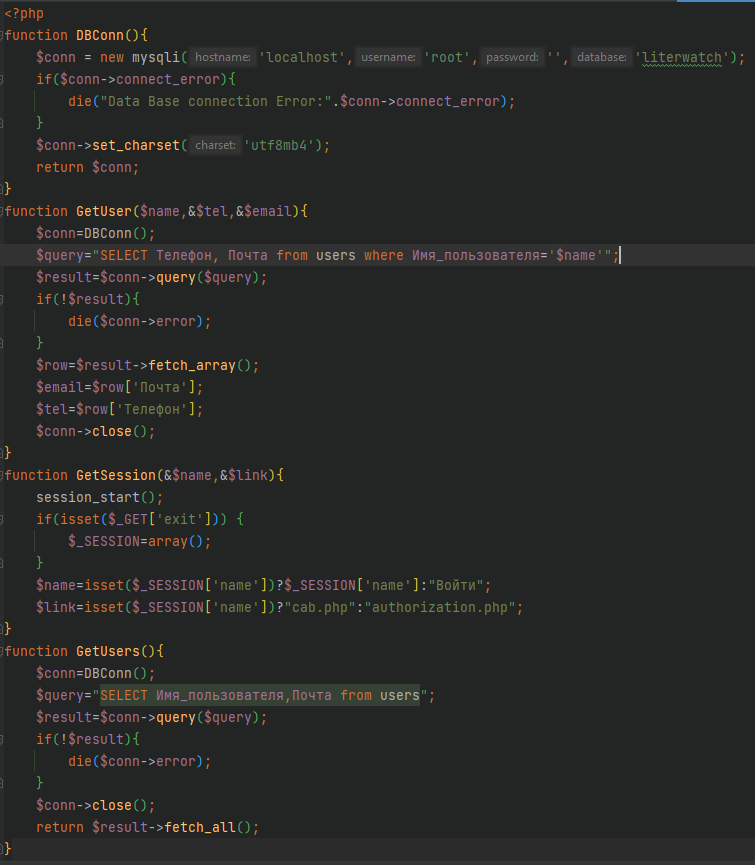


Рисунок 11. DBConn.php

### 2.2 Логика авторизации.

Если пользователь хочет авторизоваться, он попадает на форму авторизации. Пользователь вводит данные и, если пользователь с такой почтой есть в БД и его пароль введен корректно, попадаем на главную страницу, где вместо ссылки на эту форму появляется имя пользователя и, если нажать на него, пользователь попадёт в кабинет.

После введения данных, страница проверяет следующее:

* Наличие пользователя в БД
* Соответствие пароля

Сначала следует запрос, который ищет пользователя по почте в БД. Если он его не находит, пользователю выводится соответствующая ошибка.

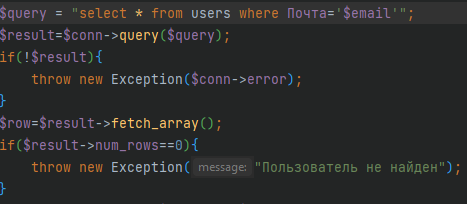


Рисунок 20. Поиск пользователя

Если пользователь найден (несколько пользователей быть не может, поскольку при регистрации стоит проверка на уникальность почты и имени), идёт проверка пароля.

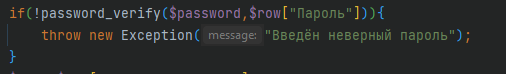


Рисунок 21. Проверка пароля

Т.к. в БД хранится не пароль, а его соль, необходимо воспользоваться функцией password\_verify, чтобы произвести проверку пароля. После проверок пользователь авторизуется, т.е. его имя сохраняется в сессии, и он переходит на главную страницу.

# Раздел 4. Раздел описания и тестирования приложения

Конец предыдущего раздела поверхностно описывал работу регистрации и авторизации сайта.

В этом разделе описание функционала сайта будет подкреплено PHP-кодом.

## Реализация регистрации



Рисунок 19. Ошибка.

Все дальнейшие исключения будут работать идентичным образом.

## Реализация авторизации

## Реализация кабинета пользователя

Кабинет предоставляет пользователю:

* Имя, почту, телефон текущего пользователя.
* Возможность добавить новое видео
* Возможность выйти из аккаунта
* Список всех его видео

При загрузке создаётся соединение с БД и текущей сессией. После запускается функция «GetUser». Данная функция была описана в файле DBConn. Она запрашивает почту и телефон текущего пользователя и записывает их в соответствующие переменные:

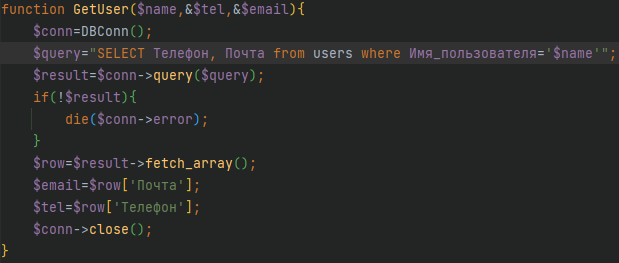


Рисунок 22. GetUser.

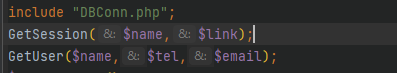


Рисунок 23. Вызов функции в cab

Далее страница запрашивает название и код всех видео текущего пользователя. После получения данных следует цикл, который записывает в переменную videos все видео пользователя по следующему принципу:

* Контейнер div, хранящий все видео
* Контейнер div, хранящий одно видео
* Ссылка a, содержащая превью видео и его название

Ссылка на видео формируется из файла watch.php, который будет рассмотрен ниже и переменной v, которая хранит в себе код видео.

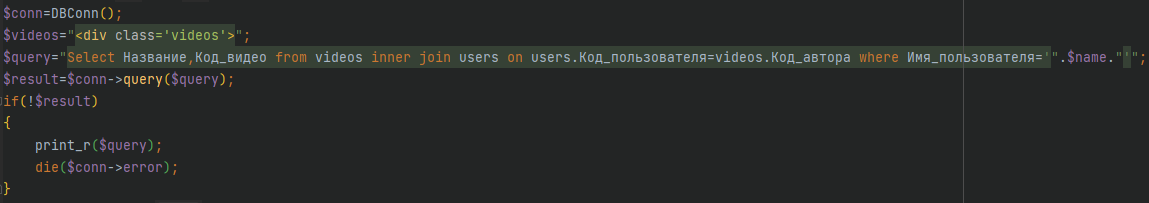


Рисунок 24. Запрос на получение видео пользователя.

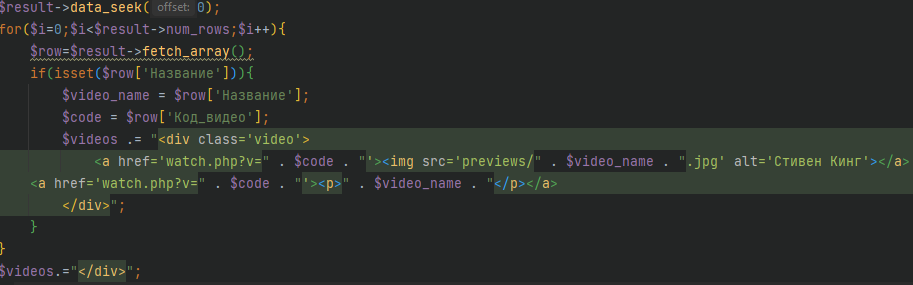


Рисунок 25. Цикл вывода списка видео

После того, как контейнер будет готов, страница выведет всю информацию о пользователе, кнопку выхода, кнопку записи видео и видео пользователя.

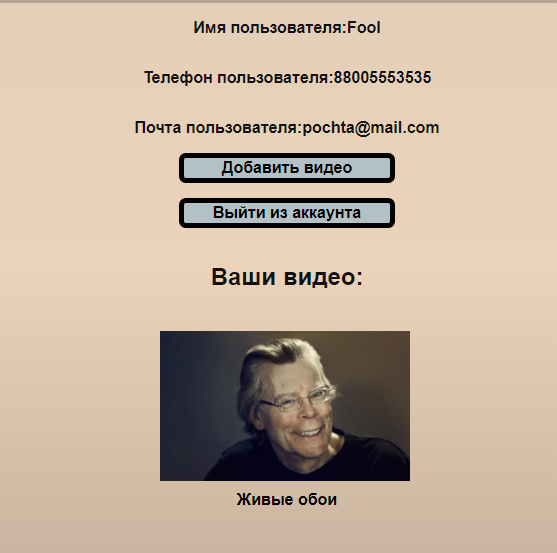


Рисунок 26. Страница кабинета.

## Реализация добавления видео

Перед тем, как рассматривать работу видеоплеера, необходимо разобрать алгоритм добавления видео на страницу. Итак, после нажатия на кнопку «добавить видео», пользователь видит форму, представленную на рисунке 25.

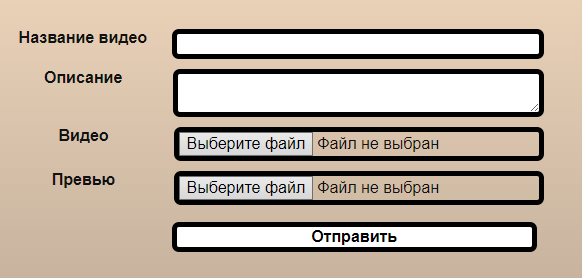


Рисунок 27. Форма добавления видео

После ввода всех данных, выбора видео и превью к нему, активируется скрипт, который проверяет данные и, в случае успеха, отправляет данные в БД. В первую очередь проверяются файлы. В случае ошибки, срабатывает исключение и загружается страницу ошибки и выводит сообщение.

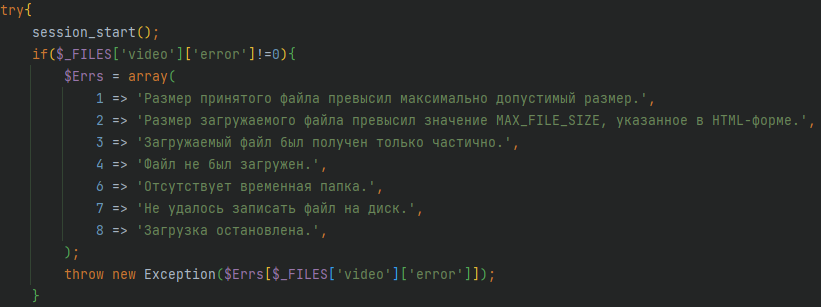


Рисунок 28. Обработка файлов

В массиве Errs хранятся коды ошибок и сообщения в соответствии с каждым кодом. После обработки файлов следует отправление нового видео в БД. Видео хранится в базе по названию, а хранится в папку videos, когда как превью – в папку previews.



Рисунок 29. Сохранение и запись нового видео.

После заполнения, видео попадает в БД и на главную страницу.

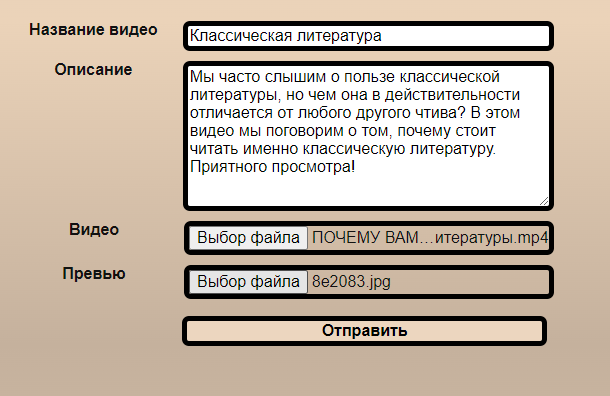


Рисунок 30. Заполненная форма.

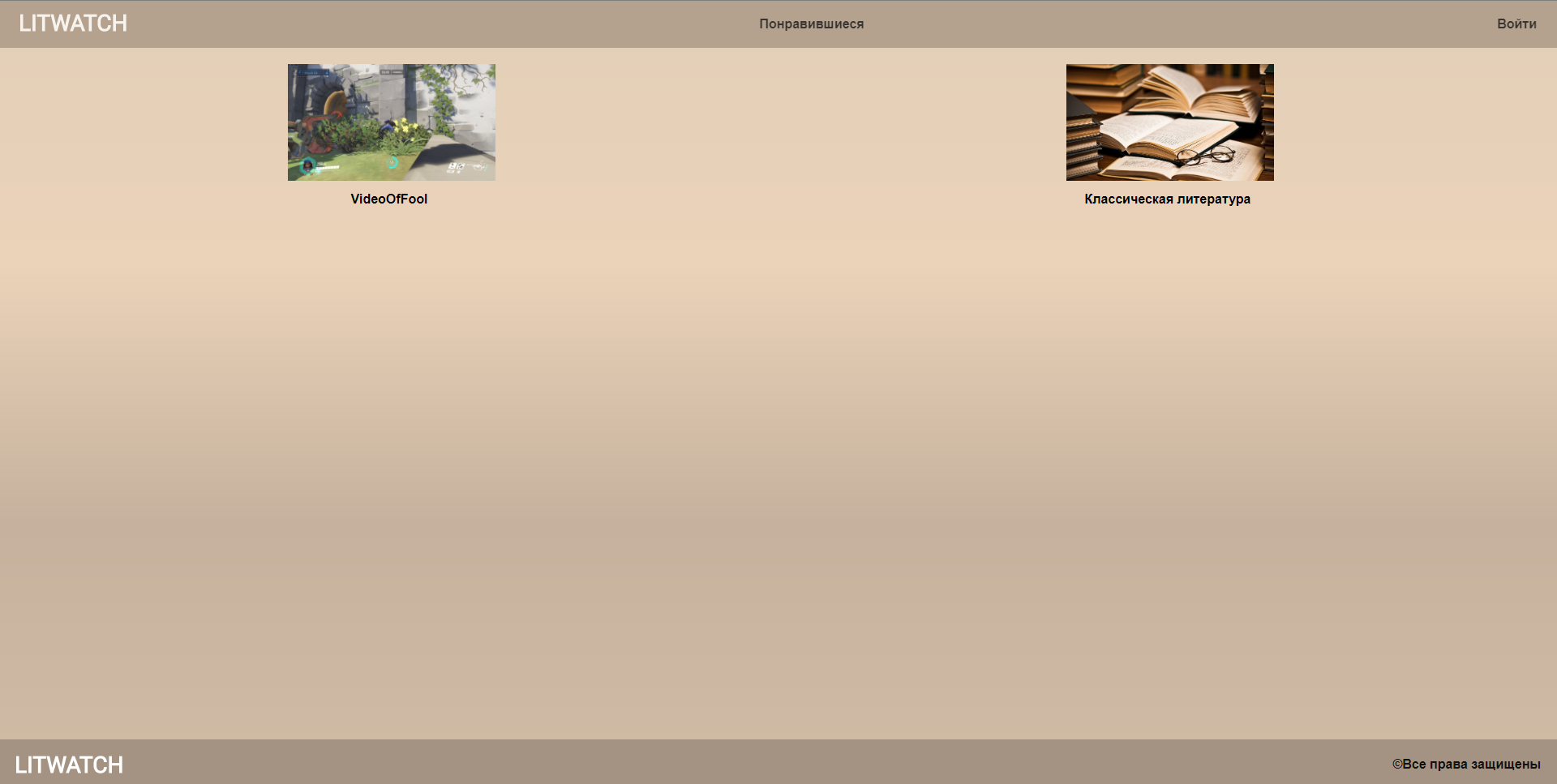


Рисунок 31. Новое видео на главной странице.

## Реализация видеоплеера

### 5.1 Загрузка страницы

Когда видео загружено, можно перейти к его просмотру. Пользователь нажимает на ссылку и попадает на страницу с видео.

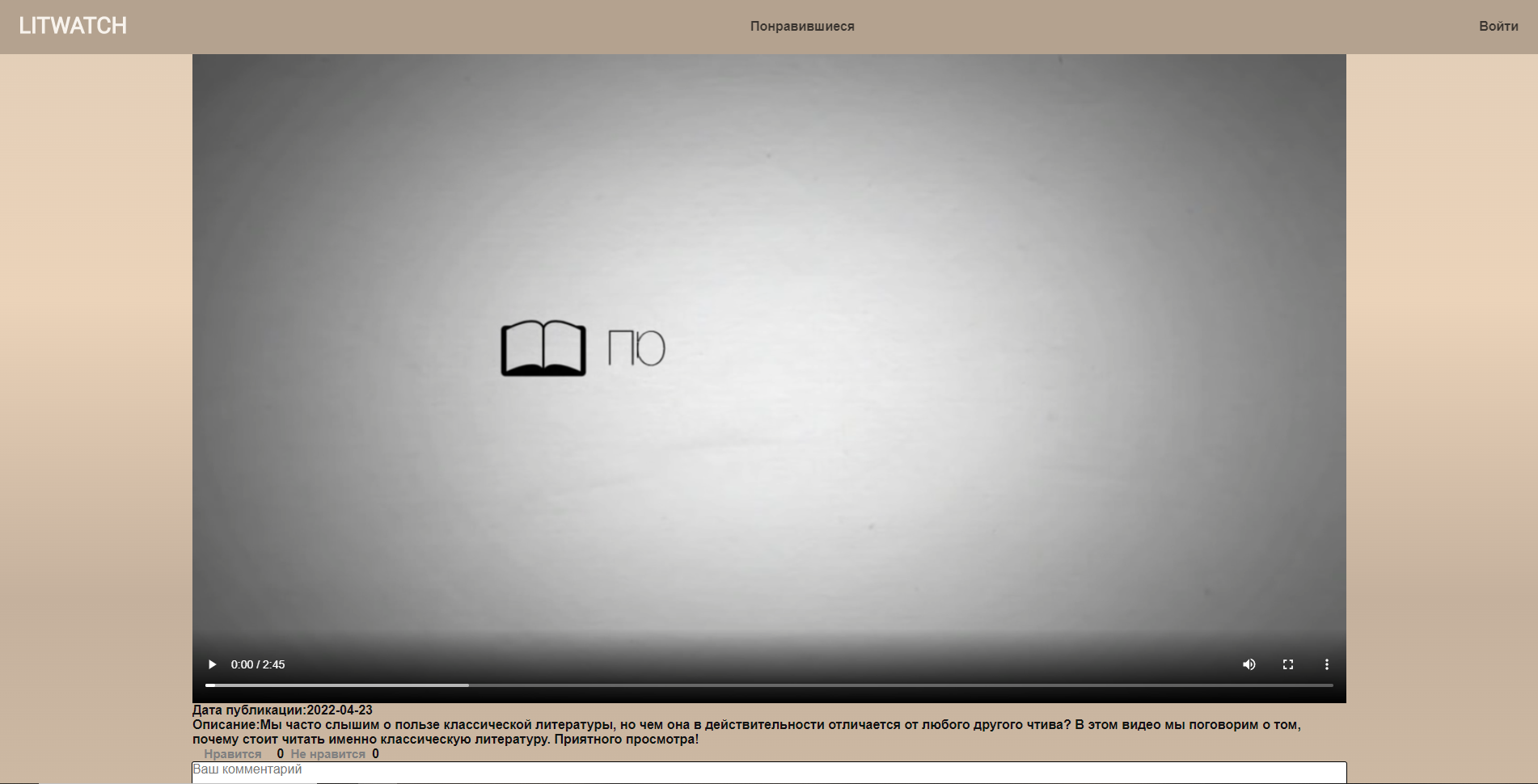


Рисунок 32. Страница видео.

Изначально страница не содержит в себе ничего, однако перед её загрузкой запускается алгоритм, который запрашивает из БД:

* Информацию о видео, которое выбрал пользователь
* Статистику лайков и дизлайков
* Информацию о комментариях

Информация о видео содержится в БД, чтобы её получить в ссылке, с помощью метода GET передаётся код видео:



Рисунок 33. Ссылка на видео

Такая ссылка создаётся на любой странице, где видео доступно для просмотра. Как только загружается страница, генерируется запрос, который по коду видео получает все данные о нём.

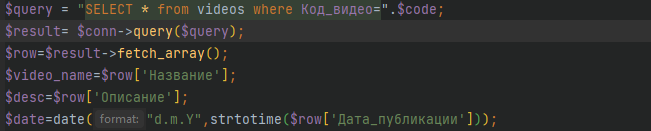


Рисунок 34. Получение данных о видео

После получения данных о видео, необходимо загрузить статистику лайков и дизлайков. Сначала нужно загрузить данные из статистики для текущего пользователя. Это нужно, чтобы понять, оценивал ли пользователь видео до нынешнего посещения страницы и, в соответствии с результатом, загрузить статистику.

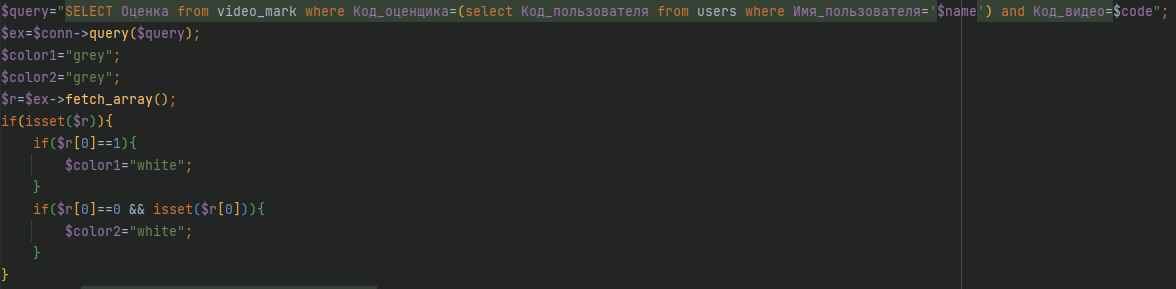


Рисунок 35. Получение для пользователя и выбор цвета кнопок.

Также нужно вывести общее количество лайков и дизлайков.

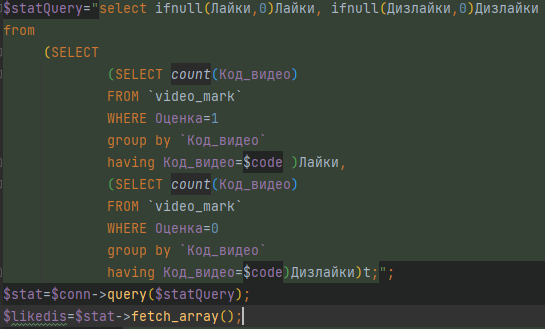


Рисунок 36. Получение общей статистики

И, наконец, вывести запросить комментарии.

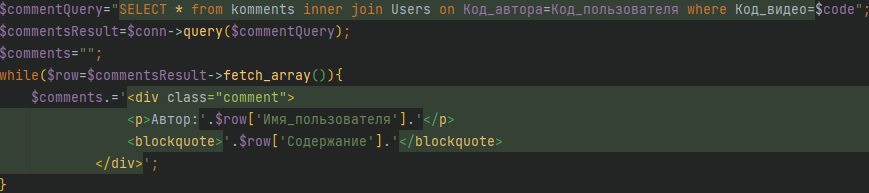


Рисунок 37. Получение комментариев

После получения данных, необходимо их вывести. В видеоплеер, загруженный на странице, вставляется ссылка на видео по названию, которое также является именем видео файла.



Рисунок 38. Видео плеер со ссылкой на видео файл.

После, вместе с датой публикации, выводится описание.



Рисунок 39. Дата публикации и описание видео.

Далее следуют кнопки оценок. Важно понимать, если пользователь уже оценил видео, то соответствующая кнопка будет подсвечена. Рядом с каждой кнопкой отображается текущее количество оценок соответственно. Алгоритм оценки будет рассмотрен в следующем разделе.



Рисунок 40. Кнопки лайка и дизлайка и их количество

После этого следует форма добавления комментария.

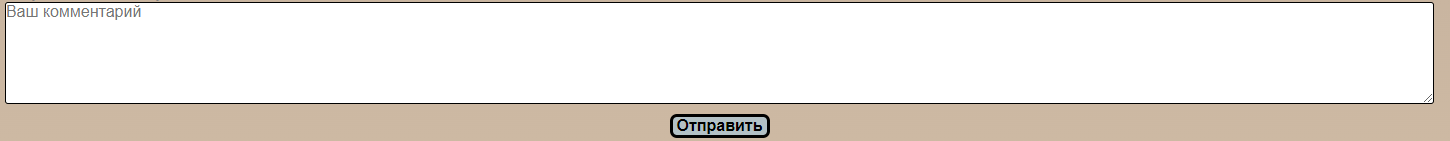


Рисунок 41. Форма комментария

Неавторизованный пользователь не может оставлять комментарии и оценивать видео.

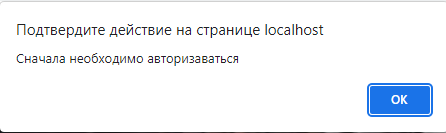


Рисунок 42. Сообщение о необходимости авторизации

После формы следуют оставленные ранее комментарии.



Рисунок 43. Текущие комментарии.

### 5.2 Алгоритм оценки.

Оценка должна обновляться в реальном времени, поэтому помимо php, который необходим для добавления оценки в БД, понадобится JavaScript (далее JS). JS также отвечает за изменение цвет кнопок при загрузке.



Рисунок 44. Изменение цвета кнопок и получение текущего количества лайков и дизлайков.

Как только пользователь нажимает на кнопку, срабатывает JS функция SendMark. Именно она оповещает пользователя, что он не авторизован.

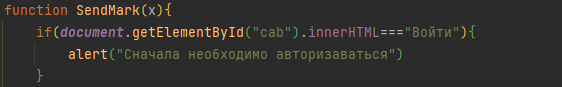


Рисунок 45. Начало функции, проверяющее авторизацию

В зависимости от нажатой кнопки алгоритм:

* Прибавляет 1 к нажатой оценке:

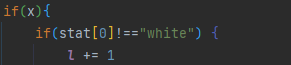


Рисунок 46. Прибавление лайка.

* Отнимает 1 от противоположной оценки (если та нажата)



Рисунок 47. Вычитание дизлайка

После этого меняет цвет кнопок и значения оценок



Рисунок 48. Изменение цвета кнопок.

В случае, если пользователь хочет снять оценку, функция просто отнимает 1 от нажатой оценки.

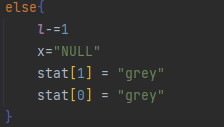


Рисунок 49. Снятие оценки

Аналогично и с дизлайком. Таким образом, меняется значение оценки.

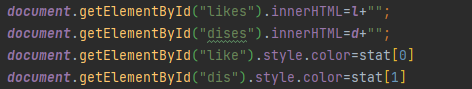


Рисунок 50. Применение изменений

Однако оно изменилось только для пользователя. Если оставить всё как есть, при обновлении страницы все изменения пропадут. Чтобы добавить изменения в БД, нужно послать запрос. Сделать это можно с помощью php. Чтобы передать данные из JS в php, используется функция ajax библиотеки jQuery, которую можно скачать на официальном сайте. Подключаем jQuery к файлу и используем ajax.



Рисунок 51. Подключение jQuery

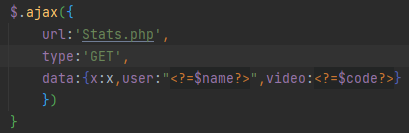


Рисунок 52. Применение ajax

С помощью данного метода, SendMark отправляет в файл Stats.php с помощью метода GET следующие данные:

* Оценка (1-лайк 0-дизлайк, NULL-снятие оценки)
* Имя текущего пользователя
* Код текущего видео

После этого активируется файл Stats.php. Пользователь этого не видит, для него страница не изменилась. Однако на фоне, скрипт формирует запрос, который либо создаёт новую строку данных в таблице video\_mark и вставляет туда поставленную оценку, либо, если запись с таким пользователем у этого видео есть, меняет её.

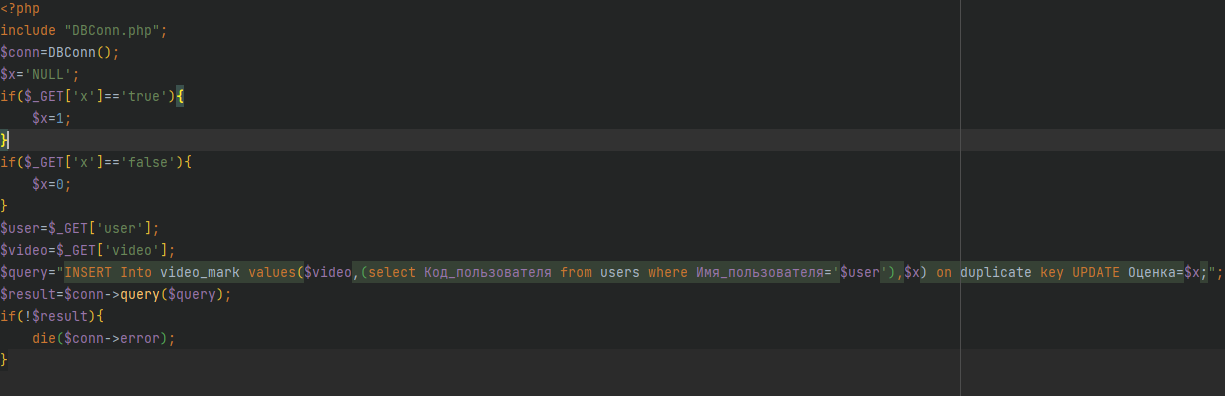


Рисунок 53. Запись оценки пользователя.

Запись сделана, поэтому, когда пользователь обновит страницу, данные останутся прежними.



Рисунок 54. Оценка видео пользователем

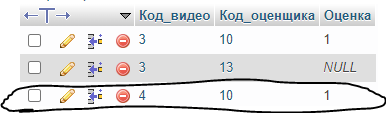


Рисунок 55. Соответствующая запись в БД

### 5.3 Алгоритм добавления комментария

Если пользователь захочет добавить комментарий, он может написать его в соответствующее поле и нажать «Отправить». Если поле осталось пустым, пользователю выведется сообщение.

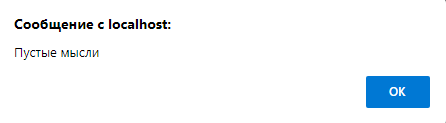


Рисунок 56. Оповещение пользователя о незаполненном поле комментария

Если всё в порядке, запускается функция ajax.

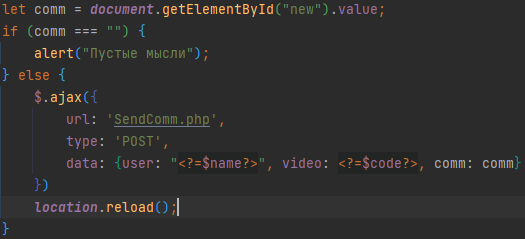


Рисунок 67. Запуск скрипта для записи комментария

В скрипт передаются:

* Имя пользователя
* Код видео
* Содержание комментария

Сам php-скрипт просто записывает новую строку в таблицу comments.

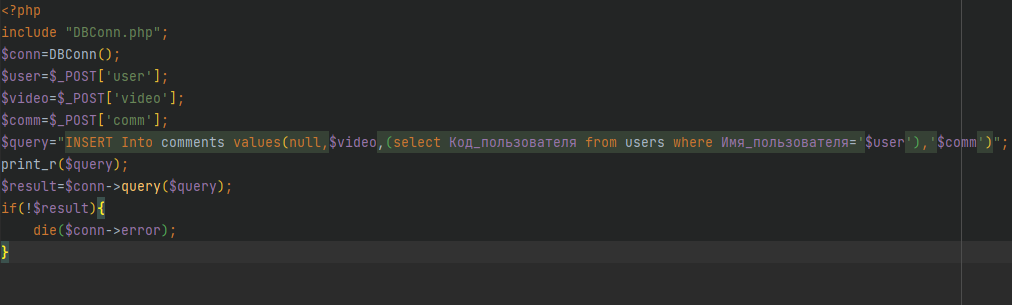


Рисунок 58. Запись нового комментария

После этого страница перезагружается и появляется новый комментарий.



Рисунок 59. Написанный комментарий



Рисунок 60. После нажатия на кнопку



Рисунок 61. Запись в БД

## Тестирование

После завершения написания кода, можно приступать к тестированию.

|  |  |
| --- | --- |
| Номер | Тест кейс №1 |
| Название | Регистрация нового пользователя с корректными данными |
| Подготовка к тесту | - |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Нажать на кнопку «Регистрация»   4. Ввести в поле «Имя пользователя» корректное имя «New\_Fool»   5. Ввести в поле «Телефон» корректный телефон «89459163559»   6. Ввести в поле «Почта» корректную почту «email@mail.ru»   7. Ввести в поле «Пароль» корректный пароль «F1olO»   8. Ввести в поле «Подтверждение пароля» корректный пароль «F1olO» |
| Ожидаемый результат | Пользователь занесён в БД, перенаправлен на главную страницу и авторизирован (вместо кнопки «Войти» появляется имя пользователя и ссылка на кабинет). Пароль хеширован. |
| Фактический результат | Пользователь занесён в БД, перенаправлен на главную страницу и авторизирован (вместо кнопки «Войти» появляется имя пользователя и ссылка на кабинет). Пароль хеширован. |

|  |  |
| --- | --- |
| Номер | Тест кейс №2 |
| Название | Регистрация нового пользователя с недоступным именем. |
| Подготовка к тесту | Зарегистрировать пользователя с именем «New\_Fool» |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Нажать на кнопку «Регистрация»   4. Ввести в поле «Имя пользователя» недоступное имя «New\_Fool»   5. Ввести в поле «Телефон» корректный телефон «89459163559»   6. Ввести в поле «Почта» корректную почту «email1@mail.ru»   7. Ввести в поле «Пароль» корректный пароль «F1olO»   8. Ввести в поле «Подтверждение пароля» корректный пароль «F1olO» |
| Ожидаемый результат | Появляется ошибка «Пользователь с таким именем уже зарегистрирован!» |
| Фактический результат | Появляется ошибка «Пользователь с таким именем уже зарегистрирован!» |

|  |  |
| --- | --- |
| Номер | Тест кейс №3 |
| Название | Регистрация нового пользователя с недоступной почтой. |
| Подготовка к тесту | Зарегистрировать пользователя с почтой «email@mail.ru» |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Нажать на кнопку «Регистрация»   4. Ввести в поле «Имя пользователя» корректное имя «New\_Fool1»   5. Ввести в поле «Телефон» корректный телефон «89459163559»   6. Ввести в поле «Почта» недоступную почту «email@mail.ru»   7. Ввести в поле «Пароль» корректный пароль «F1olO»   8. Ввести в поле «Подтверждение пароля» корректный пароль «F1olO» |
| Ожидаемый результат | Появляется ошибка «Пользователь с такой почтой уже зарегистрирован!» |
| Фактический результат | Появляется ошибка «Пользователь с такой почтой уже зарегистрирован!» |

|  |  |
| --- | --- |
| Номер | Тест кейс №4 |
| Название | Регистрация нового пользователя с некорректным телефоном(только цифры) |
| Подготовка к тесту | - |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Нажать на кнопку «Регистрация»   4. Ввести в поле «Имя пользователя» корректное имя «New\_Fool1»   5. Ввести в поле «Телефон» некорректный телефон «99459163559»   6. Ввести в поле «Почта» корректную почту «email@mail.ru»   7. Ввести в поле «Пароль» корректный пароль «F1olO»   8. Ввести в поле «Подтверждение пароля» корректный пароль «F1olO» |
| Ожидаемый результат | Появляется ошибка «Некорректный номер телефона» |
| Фактический результат | Появляется ошибка «Некорректный номер телефона» |

|  |  |
| --- | --- |
| Номер | Тест кейс №5 |
| Название | Регистрация нового пользователя при несоответствии паролей |
| Подготовка к тесту | - |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Нажать на кнопку «Регистрация»   4. Ввести в поле «Имя пользователя» корректное имя «New\_Fool1»   5. Ввести в поле «Телефон» корректный телефон «89459163559»   6. Ввести в поле «Почта» корректную почту «email@mail.ru»   7. Ввести в поле «Пароль» корректный пароль «F1olO»   8. Ввести в поле «Подтверждение пароля» некорректный пароль «F1olo» |
| Ожидаемый результат | Появляется ошибка «Введённые пароли не совпадают» |
| Фактический результат | Появляется ошибка «Введённые пароли не совпадают» |

|  |  |
| --- | --- |
| Номер | Тест кейс №6 |
| Название | Авторизация пользователя с корректными данными |
| Подготовка к тесту | Зарегистрировать пользователя(имя-New\_Fool, пароль-F1olO) |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Ввести в поле «Логин» корректный логин «email@mail.ru»   4. Ввести в поле «Пароль» корректный пароль «F1olO» |
| Ожидаемый результат | Пользователь авторизован и перенаправлен на главную страницу |
| Фактический результат | Пользователь авторизован и перенаправлен на главную страницу |

|  |  |
| --- | --- |
| Номер | Тест кейс №7 |
| Название | Авторизация несуществующего пользователя |
| Подготовка к тесту | Удалить пользователя(имя-New\_Fool) |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Ввести в поле «Логин» корректный логин «email@mail.ru»   4. Ввести в поле «Пароль» корректный пароль «F1olO» |
| Ожидаемый результат | Появляется ошибка «Пользователь не найден» |
| Фактический результат | Появляется ошибка «Пользователь не найден» |

|  |  |
| --- | --- |
| Номер | Тест кейс №7 |
| Название | Авторизация пользователя с некорректным паролем |
| Подготовка к тесту | Зарегистрировать пользователя(имя-New\_Fool, пароль-F1olO) |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Ввести в поле «Логин» корректный логин «email@mail.ru»   4. Ввести в поле «Пароль» некорректный пароль «F1olo» |
| Ожидаемый результат | Появляется ошибка «Введён неверный пароль» |
| Фактический результат | Появляется ошибка «Введён неверный пароль» |

|  |  |
| --- | --- |
| Номер | Тест кейс №8 |
| Название | Добавление видео с корректными данными |
| Подготовка к тесту | Зарегистрировать пользователя(имя-New\_Fool, пароль-F1olO) |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Ввести в поле «Логин» корректный логин «email@mail.ru»   4. Ввести в поле «Пароль» корректный пароль «F1olO»   5. Нажать кнопку «New\_Fool»   6. Нажать кнопку «Добавить видео»   7. Ввести в поле «Название видео» значение «test»   8. Ввести в поле «Описание» значение «Тест»   9. Выбрать в поле видео файл «test1.mp4»   10. Выбрать в поле превью файл «test1.jpg»   11. Нажать «Отправить» |
| Ожидаемый результат | У пользователя появилось новое видео, отображаемое в виде превью и названия. В папке videos появляется в видео, выбранное пользователем, переименованное в название. Аналогично с превью в папке previews. В БД появляется запись |
| Фактический результат | У пользователя появилось новое видео, отображаемое в виде превью и названия. В папке videos появляется в видео, выбранное пользователем, переименованное в название. Аналогично с превью в папке previews. В БД появляется запись |

|  |  |
| --- | --- |
| Номер | Тест кейс №8 |
| Название | Добавление видео с корректными данными |
| Подготовка к тесту | Зарегистрировать пользователя(имя-New\_Fool, пароль-F1olO), загрузить на компьютер файлы test1.mp4 и test1.jpg весом менее 50МБ каждый |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Ввести в поле «Логин» корректный логин «email@mail.ru»   4. Ввести в поле «Пароль» корректный пароль «F1olO»   5. Нажать кнопку «New\_Fool»   6. Нажать кнопку «Добавить видео»   7. Ввести в поле «Название видео» значение «test»   8. Ввести в поле «Описание» значение «Тест»   9. Выбрать в поле видео файл «test1.mp4»   10. Выбрать в поле превью файл «test1.jpg»   11. Нажать «Отправить» |
| Ожидаемый результат | У пользователя появилось новое видео, отображаемое в виде превью и названия. В папке videos появляется в видео, выбранное пользователем, переименованное в название. Аналогично с превью в папке previews. В БД появляется запись |
| Фактический результат | У пользователя появилось новое видео, отображаемое в виде превью и названия. В папке videos появляется в видео, выбранное пользователем, переименованное в название. Аналогично с превью в папке previews. В БД появляется запись |

|  |  |
| --- | --- |
| Номер | Тест кейс №8 |
| Название | Добавление видео с корректными данными |
| Подготовка к тесту | Зарегистрировать пользователя(имя-New\_Fool, пароль-F1olO), загрузить на компьютер файлы test1.mp4 и test1.jpg весом более 50МБ |
| Шаги тестовой ситуации | * 1. Зайти на сайт http://localhost/litwatch   2. Нажать на кнопку «Войти»   3. Ввести в поле «Логин» корректный логин «email@mail.ru»   4. Ввести в поле «Пароль» корректный пароль «F1olO»   5. Нажать кнопку «New\_Fool»   6. Нажать кнопку «Добавить видео»   7. Ввести в поле «Название видео» значение «test»   8. Ввести в поле «Описание» значение «Тест»   9. Выбрать в поле видео файл «test1.mp4»   10. Выбрать в поле превью файл «test1.jpg»   11. Нажать «Отправить» |
| Ожидаемый результат | Появляется ошибка «Размер принятого файла превысил максимально допустимый размер.» |
| Фактический результат | Появляется ошибка «Размер принятого файла превысил максимально допустимый размер.» |

Данные тест-кейсы покрывают 88% всего проекта.

# Заключение

В результате проделанной работы было создано ПО, которое имеет:

* Возможность добавления новых пользователей в систему
* Возможность добавления видео на сайт
* Возможность оценки видео
* Возможность комментирования видео

Данные преимущества позволят привлечь большое количество новых пользователей на сайт, что позволит в дальнейшем расширить сайт. Таким образом, сайт сможет эффективнее выполнять свою основную задачу: привлечение внимание к литературе и, в частности, к продукции кампании «Литнесс».

Данный проект можно также использовать и за пределами основной цели. Например, смена тематики или отклонение то конкретной темы, что привлечёт большое количество пользователей. Таким образом, проект может быть модернизирован.