|  |  |
| --- | --- |
|  | **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 | 2% |
| 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 |  |
| 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 |  |
| 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. |  |
| 1. Раздел описания и тестирования приложения |  |
| Заключение | 2% |
| Список использованной литературы | 1% |

Графическая часть:

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

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

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

Презентация.

Дипломник: /                   /

Руководитель проекта                                         /                             /

Заведующий кафедрой:                              /

Оглавление

Изм.

Лист

№ докум.

Подпись

Дата

Лист

Разраб.

Тихонов И. А.

Пров.

Реценз.

Киселёва С. В.

*Синицын*

Н. контр.

Утв.

Лит.

Листов

**ГБПОУ Колледж «Царицыно»**

[ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ 2](#_Toc101434366)

[Содержание пояснительной записки 3](#_Toc101434367)

[Графическая часть: 4](#_Toc101434368)

[ОТЗЫВ НА КУРСОВОЙ ПРОЕКТ 7](#_Toc101434369)

[Введение 10](#_Toc101434370)

[Раздел 1. Обзорно-постановочная часть 11](#_Toc101434371)

[1. Анализ предметной области 11](#_Toc101434372)

[2. Обоснование необходимости и цели использования системы для решения задачи. 11](#_Toc101434373)

[3. Постановка цели и задач веб-приложения 12](#_Toc101434374)

[4. Анализ существующих разработок 12](#_Toc101434375)

[5. Обоснование проектных решений 12](#_Toc101434376)

[Раздел 2. Раздел проектирования веб-приложения 13](#_Toc101434377)

[1. Анализ содержания проекта 13](#_Toc101434378)

[2. Проектирование базы данных 13](#_Toc101434379)

[3. Проектирование информационной архитектуры сайта 14](#_Toc101434380)

[4. Разработка дизайн-макета сайта 15](#_Toc101434381)

[Раздел 3 Раздел реализации приложения 17](#_Toc101434382)

[1. Разработка интерфейса пользователя 17](#_Toc101434383)

[2. Реализация логики работы веб-приложения 18](#_Toc101434384)

[2.1 Реализация регистрации. 18](#_Toc101434385)

[2.2 Реализация авторизации. 22](#_Toc101434386)

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

Предметная область данного проекта – фирма по продаже зарубежных книг «Литнесс». 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. Однако в наше время актуальна проблема снижения интереса к литературе, в частности к печатной. Данный проект был инициирован с целью решения этой проблемы.

# 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

Все эти сущности взаимосвязаны между собой (Неавторизованный пользователь не может загрузить видео или оставить комментарий, а комментарий нельзя оставить к несуществующему видео). На рисунке 1 представлен ранний концепт базы данных.

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

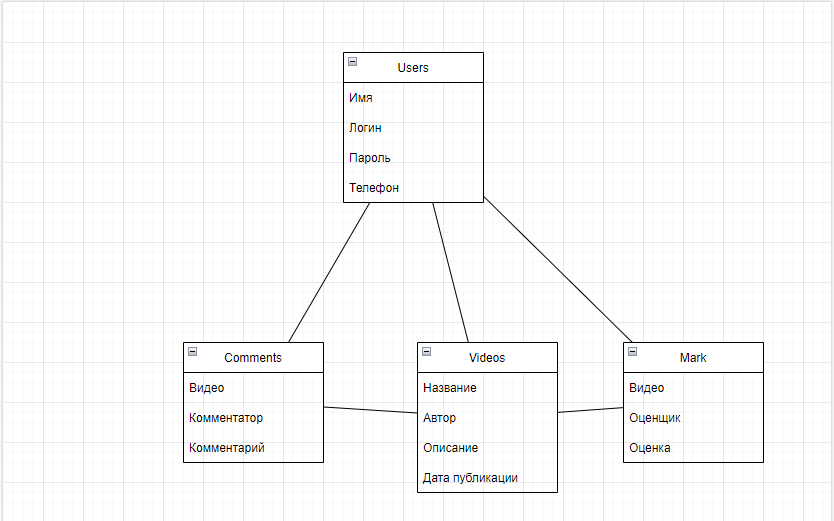


Рисунок 1. Ранний концепт БД.

## Проектирование информационной архитектуры сайта

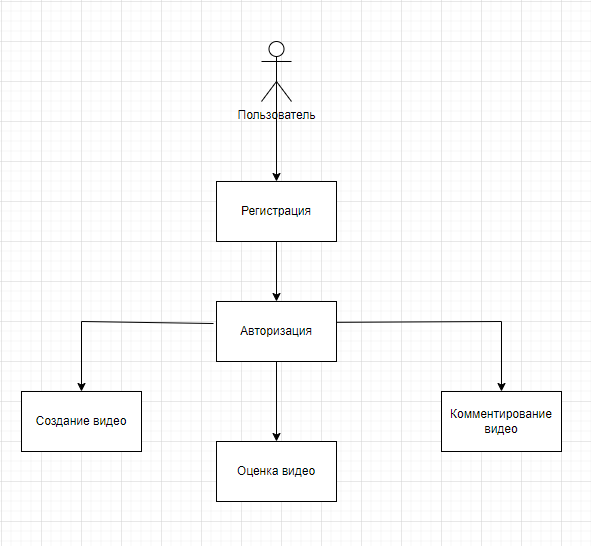
Пользователь должен иметь самые необходимые функции для взаимодействия с системой. На рисунке 2 представлены все возможности пользователя.

Рисунок 2. Функции 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.

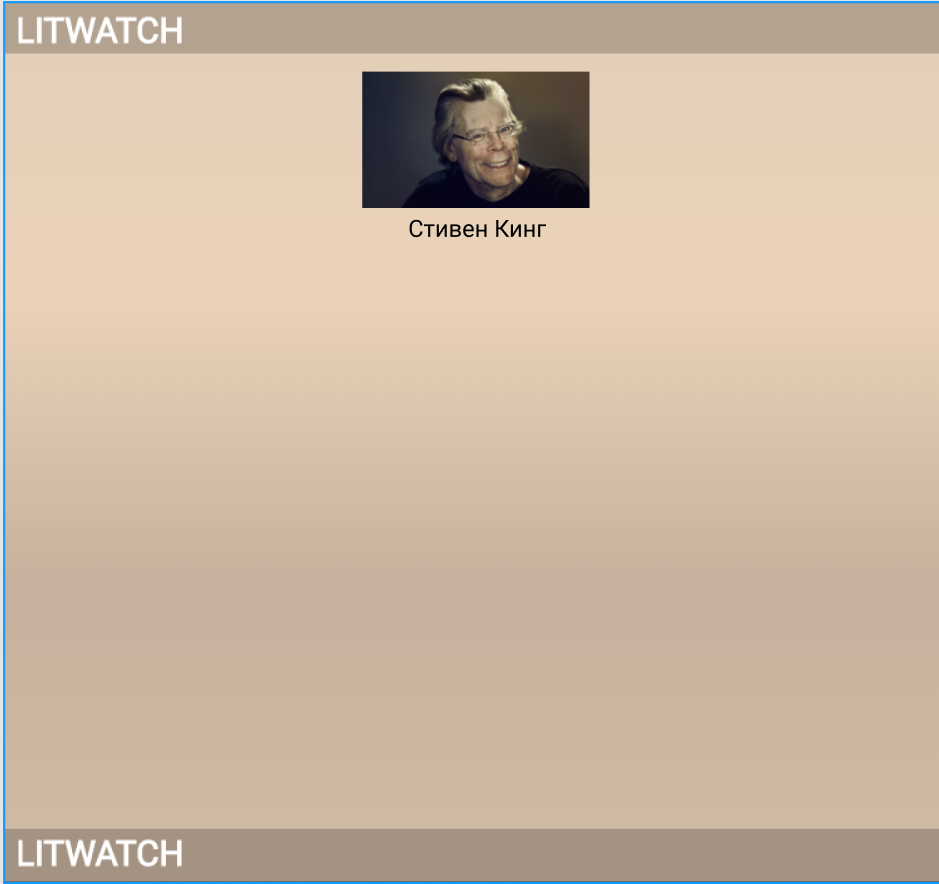


Рисунок 3. Макет-дизайн главной страницы

# 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.

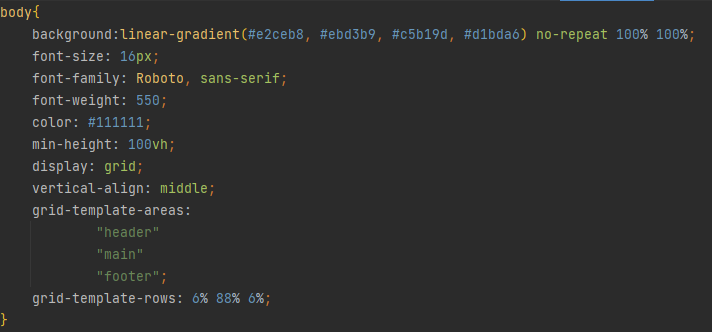


Рисунок 4. Стилизация тела страницы

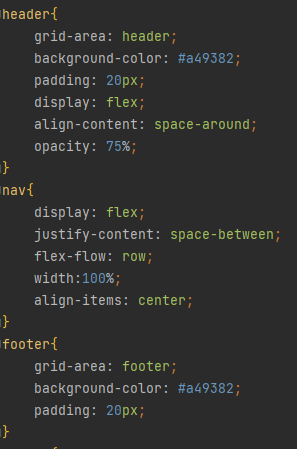


Рисунок 5. Стилизация шапки, панели навигации и подвала

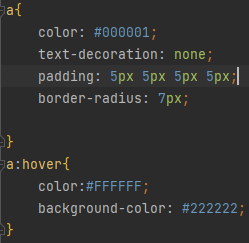


Рисунок 6. Стилизация ссылок

## 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

На рисунке 7 представлена итоговая форма:



Рисунок 7. Форма регистрации.

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

Алгоритм регистрации находится в файле с регистрацией и активируется, когда отправляются данные. При отправке данных не стоит забывать о безопасности данных. Поэтому при создании нового пользователя необходимо произвести хеширование пароля. После записи, записываем имя пользователя в сессию и перенаправляем его на главную страницу. На рисунке 8 предоставлен полный код добавления нового пользователя. Проверка данных будет разобрана позже.

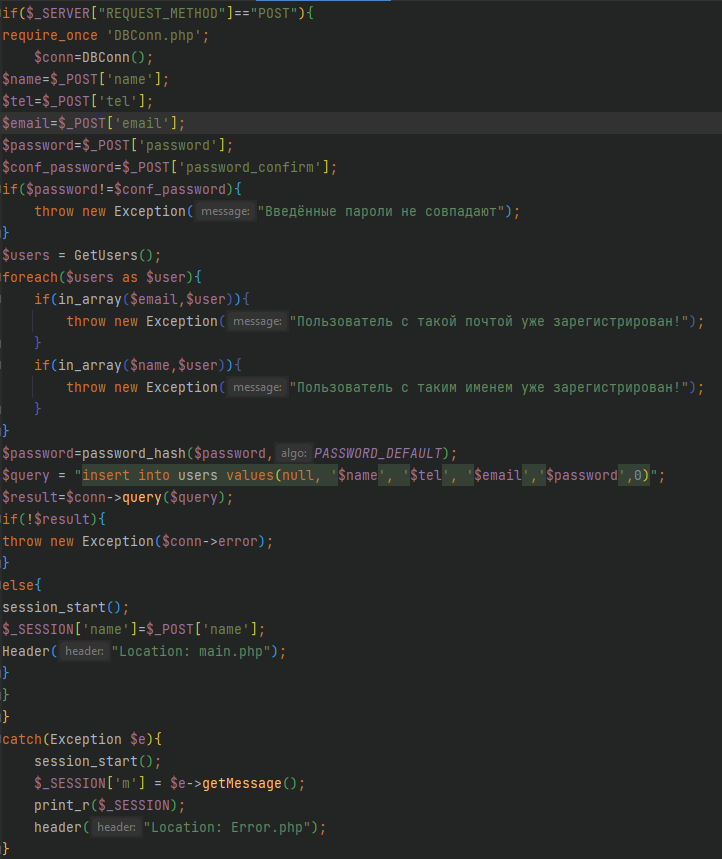


Рисунок 8. Регистрация нового пользователя

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 всех пользователей

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

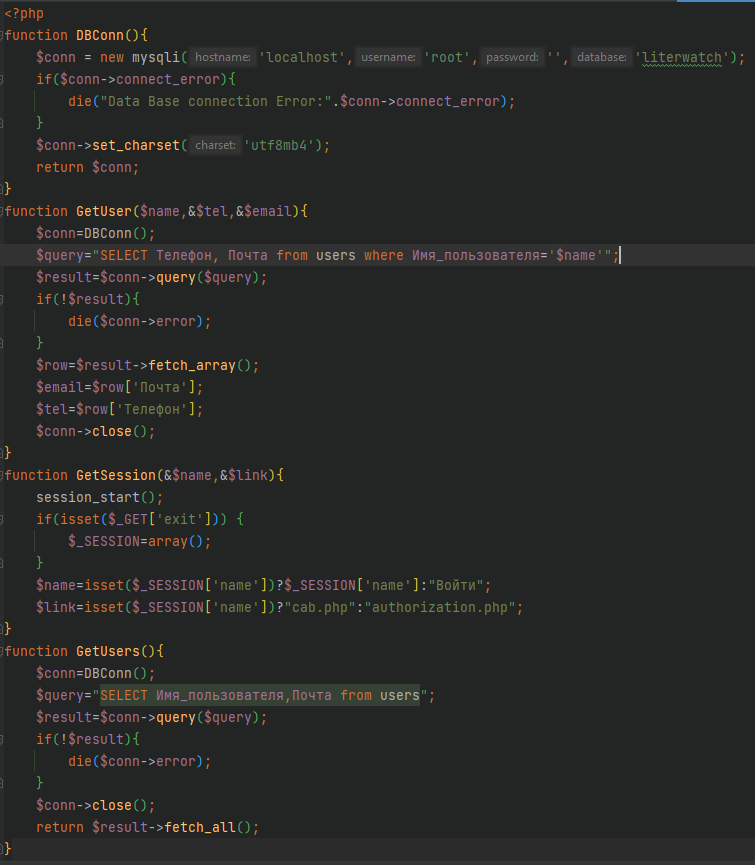


Рисунок 9. DBConn.php

### 2.2 Логика авторизации.

Если пользователь хочет авторизоваться, он попадает на форму авторизации, представленной на рисунке 10.

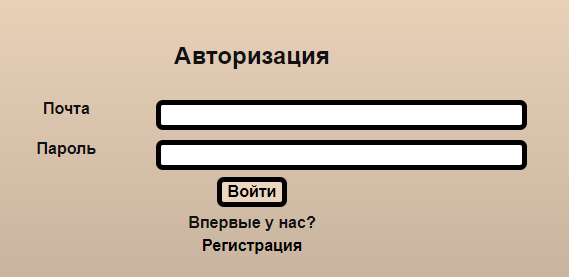


Рисунок 10. Форма авторизации.

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.

# 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

Как было сказано выше, 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:

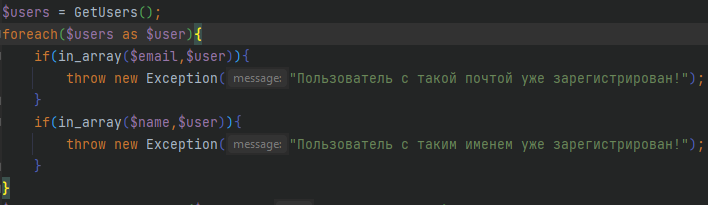


Рисунок 11. Проверка почти и имени.

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

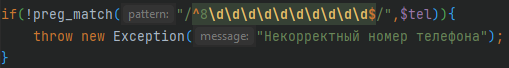


Рисунок 12. Проверка телефона.

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

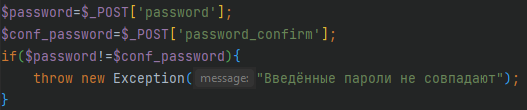


Рисунок 13. Проверка паролей

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.

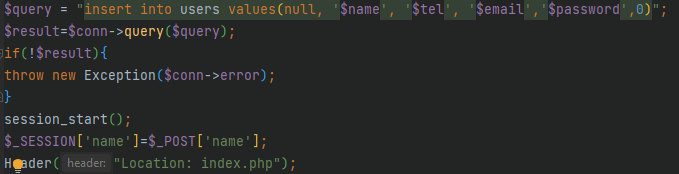


Рисунок 14. Отправка данных и начало сессии

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.

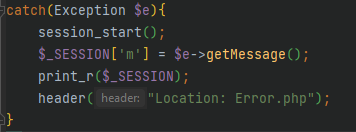


Рисунок 15. Обработка исключений

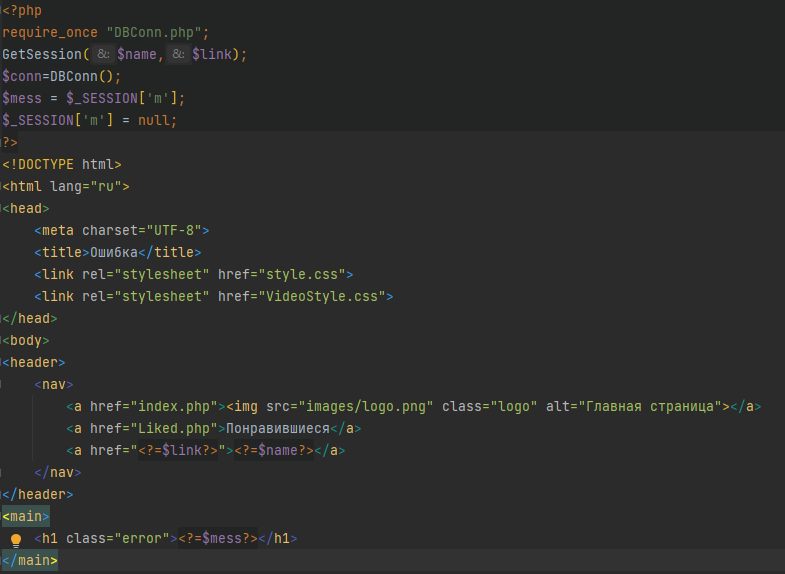


Рисунок 16. Error.php



Рисунок 17. Ошибка.

Все дальнейшие исключения будут работать идентичным образом.

## Реализация авторизации

Если пользователь зарегистрировался ранее, он может пройти авторизацию. 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.

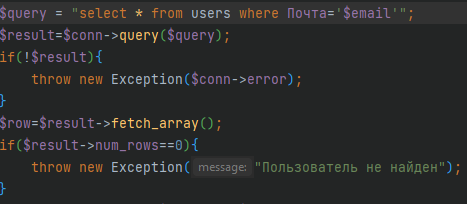


Рисунок 18. Поиск пользователя

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

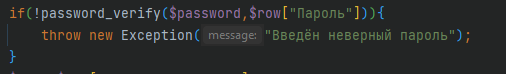


Рисунок 19. Проверка пароля

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.

## 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:

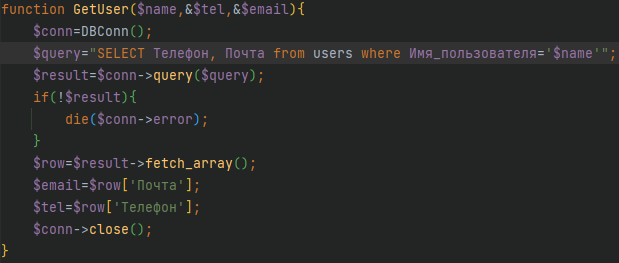


Рисунок 20. GetUser.

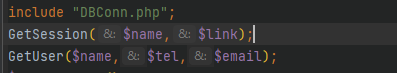


Рисунок 21. Вызов функции в cab

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.

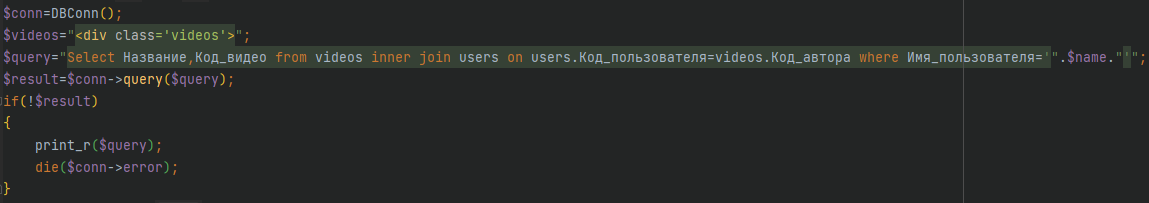


Рисунок 22. Запрос на получение видео пользователя.

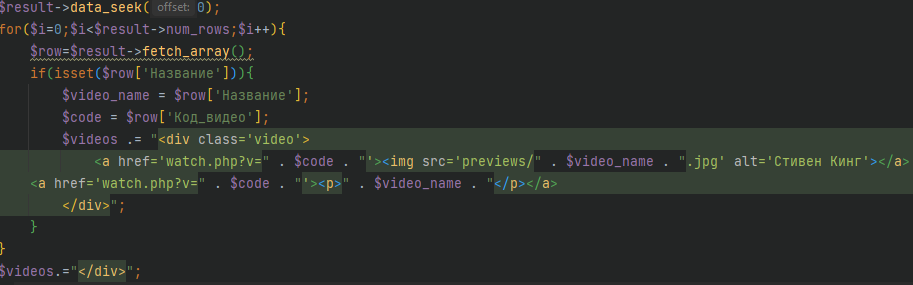


Рисунок 23. Цикл вывода списка видео

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

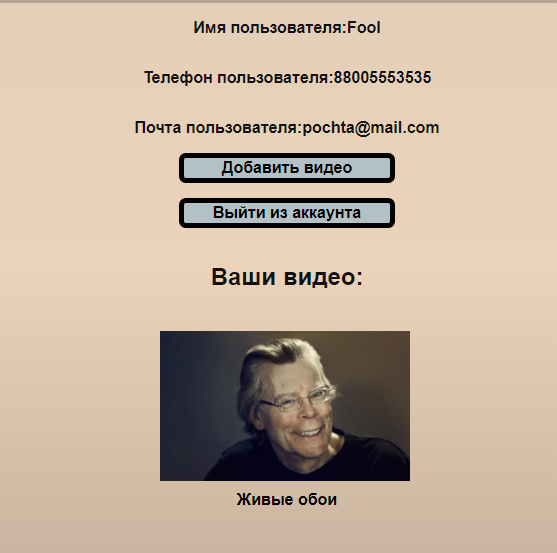


Рисунок 24. Страница кабинета.

## 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.

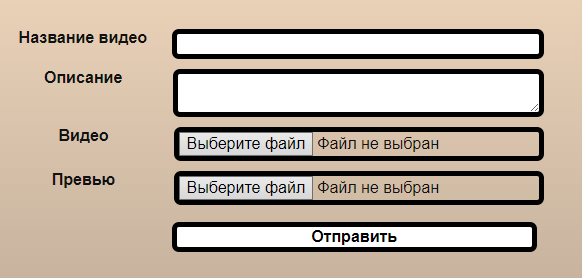


Рисунок 25. Форма добавления видео

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.

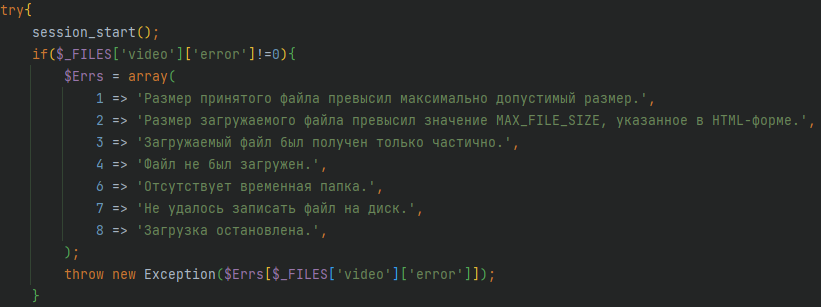


Рисунок 26. Обработка файлов

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 27. Сохранение и запись нового видео.

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

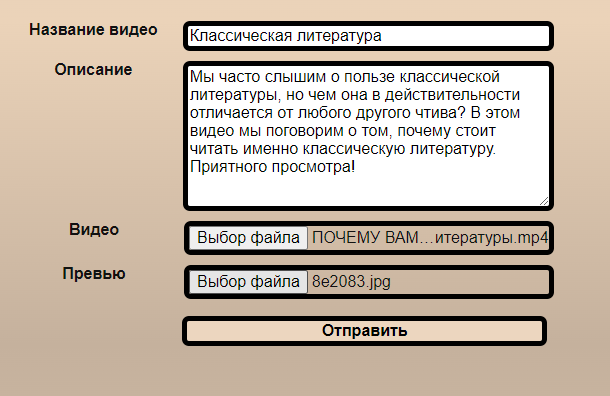


Рисунок 28. Заполненная форма.

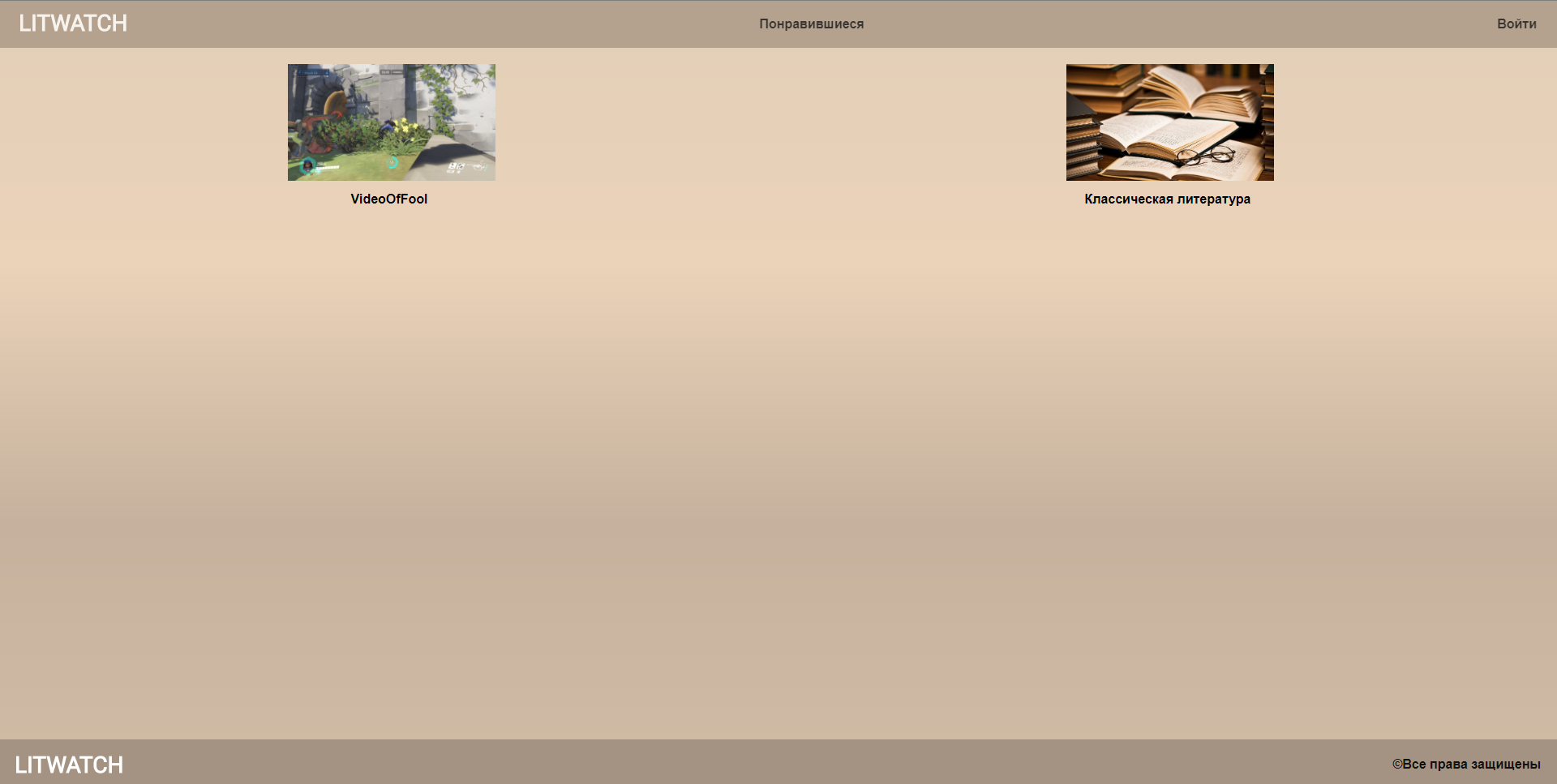


Рисунок 29. Новое видео на главной странице.

## Реализация видеоплеера

### 5.1 Загрузка страницы

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

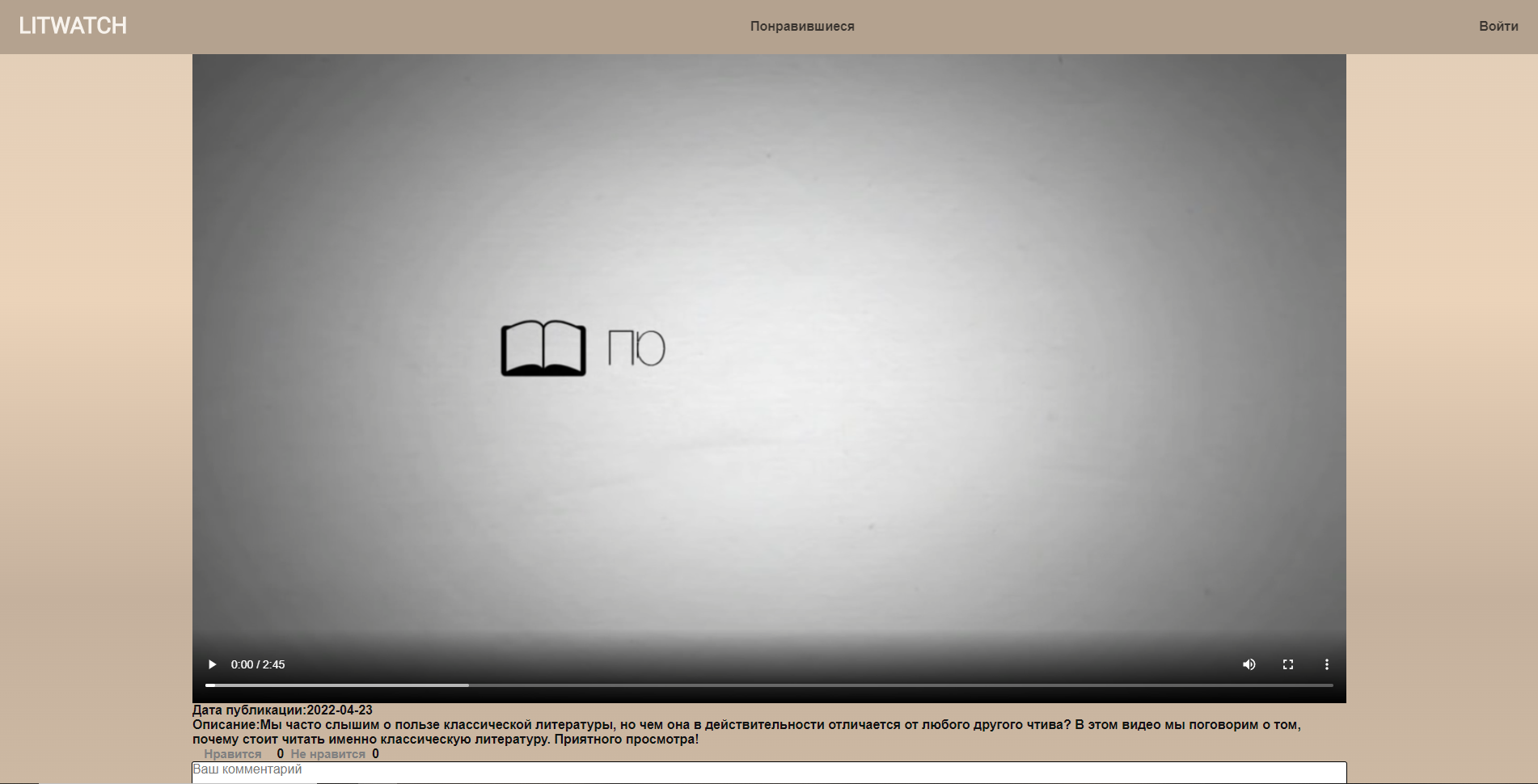


Рисунок 30. Страница видео.

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:



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.

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.

Также нужно вывести общее количество лайков и дизлайков.

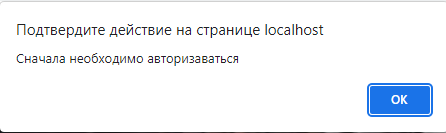
И, наконец, вывести запросить комментарии.

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

После, вместе с датой публикации, выводится описание.

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.

После этого следует форма добавления комментария. Неавторизованный пользователь не может оставлять комментарии и оценивать видео.



Сообщение о необходимости авторизации

После формы следуют оставленные ранее 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). Как только пользователь нажимает на кнопку, срабатывает JS функция SendMark. В зависимости от нажатой кнопки алгоритм:

* Прибавляет 1 к нажатой оценке
* Отнимает 1 от противоположной оценки (если та нажата)

В случае, если