



МИНПРОСВЕЩЕНИЯ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тульский государственный педагогический университет им. Л.Н. Толстого»  
(ТГПУ им. Л.Н. Толстого)

Кафедра информатики и информационных технологий

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРСКАЯ РАБОТА)**

на тему:

**РАЗРАБОТКА И ИСПОЛЬЗОВАНИЕ В УЧЕБНОМ ПРОЦЕССЕ  
ИНФОРМАЦИОННОЙ СИСТЕМЫ "КАТАЛОГ  
ВИДЕОФРАГМЕНТОВ"**

Выполнил

студент группы 121161

очной формы обучения

факультета математики, физики и  
информатики

Чекрыжов Вячеслав Витальевич

Тула – 2020

**Работа выполнена на факультете математики, физики и информатики**

**ФГБОУ ВО «ТГПУ им. Л. Н. Толстого»**

Руководитель выпускной квалификационной работы – Мартынюк Юлия Михайловна, кандидат педагогических наук, доцент кафедры информатики и информационных технологий

\_\_\_\_\_

*(Дата)*

\_\_\_\_\_

*(Подпись)*

Ю.М. Мартынюк

**Работа допущена к защите:**

Заведующий кафедрой информатики и информационных технологий

\_\_\_\_\_

*(Дата)*

\_\_\_\_\_

*(Подпись)*

Ю.И. Богатырева

Защита состоится «\_\_\_\_\_» \_\_\_\_\_ 2020 года в учебном корпусе №4 ТГПУ им. Л.Н. Толстого, в аудитории №\_\_\_\_\_ в \_\_\_\_\_ часов.

Декан факультета математики, физики и информатики

\_\_\_\_\_

*(Дата)*

\_\_\_\_\_

*(Подпись)*

И.Ю. Реброва

**Содержание**

ВВЕДЕНИЕ .....	4
Глава 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ «КАТАЛОГ ВИДЕОФРАГМЕНТОВ»... 7	
1.1. Основные понятия предметной области .....	7
1.2. Технология создания интернет-сервисов.....	13
1.3. Анализ существующих систем для использования видеофрагментов в учебном процессе .....	15
1.4. Обоснование выбора платформы для реализации информационной системы .....	20
Глава 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «КАТАЛОГ ВИДЕОФРАГМЕНТОВ».....	25
2.1. Первый этап – концептуальное проектирование.....	25
2.2. Второй этап – логическое проектирование .....	28
2.3. Третий этап – физическое проектирование .....	30
Глава 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ .....	35
3.1. Физическая структура программного продукта .....	35
3.2. Интерфейс информационной системы .....	39
3.3. Функционал системы .....	42
3.4. Тестирование системы.....	58
ЗАКЛЮЧЕНИЕ .....	60
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....	61
ПРИЛОЖЕНИЕ. ТЕКСТ ПРОГРАММЫ .....	63

## **ВВЕДЕНИЕ**

На сегодняшний день система образования развивается довольно стремительными темпами. Все изменения отображаются в образовательных стандартах, профессиональном стандарте педагога и других аналогичных документах. Информационные технологии бурно развиваются и все более активно входят в нашу жизнь. Для получения высоких результатов, нужно улучшать формы и способы обучения. Наглядные методы применяются уже на протяжении длительного времени и имеют свои преимущества. А в наше время они начинают превалировать в обучении.

В «Большом толковом словаре русского языка» под ред. С. А. Кузнецова говорится, что видеоматериал – это визуальное изображение, которое является исходным материалом для последующего воспроизведения. Согласно Европейской конвенции о защите культурного наследия в форме аудио-, видеопроизведений, под видеоматериалом подразумевают любой комплект движущихся изображений, записанных любым способом, как в сопровождении звука, так и без него, способный передать движение.

Сейчас процессы хранения, поиска и доступа к данным с помощью различного рода информационных систем становятся привычными и естественными не только в областях человеческой деятельности, тесно связанных с компьютерными технологиями, но и в традиционных сферах жизни общества.

Информационная система (ИС) – это совокупность программ, главными задачами которых является сбор, хранение информации, а также ее поиск и обработка. Информация, с которой работает ИС, может быть, как текстовой, так и графической. Сервисы и инструменты сети интернет добавляют дополнительные возможности для функционирования ИС и решения с их помощью большого круга задач.

Таким образом, *актуальность* данной выпускной квалификационной работы обусловлена тенденциями развития и совершенствования различных сфер образования, с одной стороны, и необходимостью проектирования и создания системы «Каталог видеофрагментов» в виде веб-сервиса, с другой.

*Цель* работы – разработать информационную систему «Каталог видеофрагментов» для использования в учебном процессе в виде веб-сервиса.

Достижение поставленной цели потребовало решения следующих основных задач:

- разработать модель и структуру информационной системы "Каталог видеофрагментов"; определить основные компоненты и способы их взаимодействия;
- рассмотреть и проанализировать существующие аналогичные решения; определить их сильные и слабые стороны;
- обосновать выбор платформы для реализации информационной системы;
- определить функционал информационной системы и ее основные структурные компоненты;
- реализовать информационную систему в виде веб-сервиса.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка используемой литературы и приложения.

В первой главе рассматриваются основные понятия предметной области и анализируются некоторые из существующих систем для использования видеофрагментов в учебном процессе. В главе описывается технология создания веб-сервисов и обосновывается необходимость разработки информационной системы с выбором средств для ее реализации.

Вторая глава описывает процесс создания базы данных и структуры разрабатываемой системы. Здесь также формируется система требований к ПО.

Третья глава содержит описание программной реализации информационной системы.

# Глава 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ «КАТАЛОГ ВИДЕОФРАГМЕНТОВ»

## 1.1. Основные понятия предметной области

*Информационная система (ИС)* — совокупность программ, предназначенная для сбора, хранения, поиска и обработки информации.

Свойства ИС:

- может подвергаться анализу;
- динамичная и развивающаяся система;
- система обработки информации, состоящая из компьютерных и коммуникационных устройств, реализованная на базе передовых технологий;
- выходная продукция - информация, на базе которой принимаются решения;
- участие человека в функционировании ИС зависит от сложности системы, типов и наборов данных, а также степени формализации решаемых задач.

Приложения информационной системы, рассчитанные на неподготовленного в области компьютерной техники конечного пользователя, должны обладать понятным и эргономичным интерфейсом, не вызывающим трудностей и отторжения при работе с ними.

Основные требования, предъявляемые к информационным системам:

1. Эффективность информационной системы – определяется сопоставлением всех связанных с рассматриваемыми мероприятиями затрат и получаемых при этом результатов.

2. Качество функционирования информационной системы – это уровень приспособленности системы к выполнению заявленных функций.

Основных свойства определения качества функционирования информационной системы:

- адекватность функционирования информационной системы;
- наличие технических возможностей информационной системы к взаимодействию, совершенствованию и развитию;
- надежность и своевременность представления информации и выполнения функциональных технологических операций;
- полнота, безошибочность, актуальность и конфиденциальность представляемой информации.

3. Надежность информационной системы определяется надежностью технических средств ее оснащения и ошибками исполнителей.

4. Безопасность информационной системы предполагает:

защиту информации, циркулирующей в данной системе;

защита пользователей информационной системы от вредного воздействия как информации, находящейся в этой системе, так и объектов данной системы;

защита информационной системы и ее объектов от несанкционированного изменения ее заданных параметров и режима эксплуатации.

Создание и использование в учебном процессе различного рода информационных систем имеет основной целью предоставление пользователям необходимой новой и актуальной информации, влияющей на уровень общей эрудиции и расширяющей кругозор. Конечными пользователями в сфере образования являются ученики, студенты и аспиранты.

Спецификой представляемой информации в информационных образовательных системах является, прежде всего, ее ориентированность на широкий круг пользователей. Поэтому язык предоставляемой информации должен быть понятен всем категориям обучаемых: от школьника до докторанта.

Информационные системы в образовании применяются в различных областях, таких как: использование различных информационных источников в учебном процессе, выставление оценок, ведение журнала, использование библиотек, дистанционное обучение.

Одним из источников информации, используемых в учебном процессе, являются видеотрегменты.

Традиционно под *видео* принято понимать электронную технологию формирования, записи, обработки, передачи, хранения и воспроизведения движущегося изображения, основанную на принципах телевидения, а также аудиовизуальное произведение, записанное на физическом носителе (диске, флеш-носителе, магнитной пленке и т. п.).

В образовательном процессе могут использоваться следующие видеоматериалы:

- фильмы (в том числе художественные, документальные, учебные (являющиеся частью учебно-методического комплекса));
- видеокорсы,
- авторские видеопрограммы (снятые на любительскую видеокамеру) и т. п.

У использования видеоматериалов на занятиях есть ряд преимуществ:

- возможность быстро донести информацию до обучающихся;
- постоянное совершенствование педагогом собственных знаний;

- возможность применить готовые материалы неограниченное количество раз;
- высокий уровень наглядности;
- возможность неоднократно просматривать материал и систематически его усваивать;
- устранение психологических барьеров в обучении за счет создания эффекта индивидуального контакта педагога с обучаемым;
- возможность просмотра материала с остановками, что позволяет держать под контролем процесс обучения;
- возможность проводить обучение в любое время и в любом месте;
- возможность публикации видеоматериалов в глобальной сети (в том числе в режиме субтитров).

При применении видеоматериалов в учебном процессе преследуются несколько целей: активизация работы учащихся, выработка у них интереса к изучаемому предмету, контроль усвоения темы, компактное хранение информации, наглядное представление материала.

Использовать видеоматериалы на занятиях можно в разные моменты:

- для закрепления полученных знаний (в конце занятия).
- для подведения итогов или обобщения материала (в любой момент занятия);
- для поиска необходимой информации в ходе изучения нового материала (в любой момент проведения занятия);
- для поддержания интереса на занятии при актуализации, постановке цели занятия или проблемной ситуации.

Основные условия, определяющие эффективность использования видео-контента:

- использование понятного и доступного видео-контента;

- дозированная подача видеоматериала;
- систематическое использование видео-контента;
- использование видео-контента как материала для аналитической самостоятельной работы;
- встраивание видео-контента в структуру занятия;
- выявление целесообразности использования видео-контента для поставленных учебных целей;
- предварительный выбор видеоматериала.

Для разработки ИС «Каталог видеофрагментов» необходимо решить следующие основные задачи:

– разработать ядро системы – базу данных, предназначенную для обработки и хранения информации;

*База данных (БД)* — это организованная структура, которая предназначена для хранения, обработки и изменения взаимосвязанной информации.

Взаимодействие с базой данных обеспечивается с помощью специального программного инструментария – *системы управления базами данных(СУБД)*. Это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации. Наиболее распространенными СУБД являются MySQL, Microsoft SQL Server, PostgreSQL, Oracle и т.п.

- разработать графический интерфейс пользователя;
- разработать функционал ИС, который включает в себя:
  - регистрацию пользователя двух видов (студент, преподаватель);
  - авторизацию пользователя;
  - создание, изменение, удаление курсов (каталогов) для преподавателей;

- добавление, удаление видеофрагментов в свой курс (каталог) для педагогов;
- возможность подписки на интересующий курс для пользователя

## 1.2. Технология создания интернет-сервисов

На сегодняшний день в интернете можно найти огромное количество сайтов разных тематик: информационные, рекламные, сервисы, социальные сети и т.д. Каждый из них выполняет свою роль. Каждая крупная организация обладает собственным сайтом, так как именно через интернет современные люди предпочитают искать и узнавать нужную им информацию. Работающий с веб-приложениями пользователь использует интернет-клиенты – браузеры. Поэтому необходимо рассмотреть инструменты и сервисы, применяемые в клиентских приложениях, и отдельно – инструментарий веб-серверов.

Технологии, применяемые в веб-клиентах

Скриптовые языки

Большое количество современных браузеров способны интерпретировать код на скриптовых языках, таких как VBScript и JavaScript. Код на данных языках внедряется в веб-страницу и интерпретируется браузером. Одним из примеров применения таких языков — проверка корректности данных, которые пользователь вводит в предоставленные ему поля HTML-формы, непосредственно в процессе ввода или после него, без обращения к веб-серверу.

Другими примерами применения скриптовых языков являются решения дизайнерских идей, например, кнопки, меняющие свой вид при наведении на них курсора, выпадающие списки, всплывающие окна и т.д.

Код, написанный с помощью скриптовых языков, не работает самостоятельно — он выполняется в адресном пространстве браузера. Кроме того, скриптовые языки содержат ограниченный набор средств (например, они не обладают средствами доступа к файловой системе).

## Технологии создания серверных частей веб-приложений

веб-сервер — это совокупность аппаратно-программных средств, которая принимает HTTP-запросы от клиентов (как правило, веб-браузеров) и выдает HTTP-ответы. Пример ответа веб-сервера — это содержание расположенных на нем файлов: документов, изображений, фильмов и т.д. Такие данные называют статическими: если пользователь запросит несколько раз информацию по одному и тому же адресу, веб-сервер выдаст то же самое содержимое.

Язык программирования веб-приложений PHP (Personal Home Pages) давно и прочно завоевал заслуженную популярность в среде разработчиков приложений. Основная идея программирования на этом языке заключается в создании html-страниц с внедренными в них фрагментами php-кода. В отличие от скриптовых языков, используемых в целях расширения функционала браузеров, фрагменты php-кода интерпретируются напрямую сервером, а не браузером. В следствие чего результат выполнения фрагментов замещает собственно фрагмент кода в том месте страницы, откуда он передается пользовательскому браузеру.

### 1.3. Анализ существующих систем для использования видеофрагментов в учебном процессе

Перед тем как приступить к разработке информационной системы необходимо проанализировать уже имеющиеся программы подобного рода с целью оценки их достоинств и недостатков, отражающих специфику предметной области и во избежание реализации программного продукта, уступающего им по возможностям.

1. «Смотри.Учись»- веб-сервис с различными видеокурсами (см. Рис.1). На сайте представлено большое количество курсов по разным категориям. Все курсы, в большинстве своем, платные.

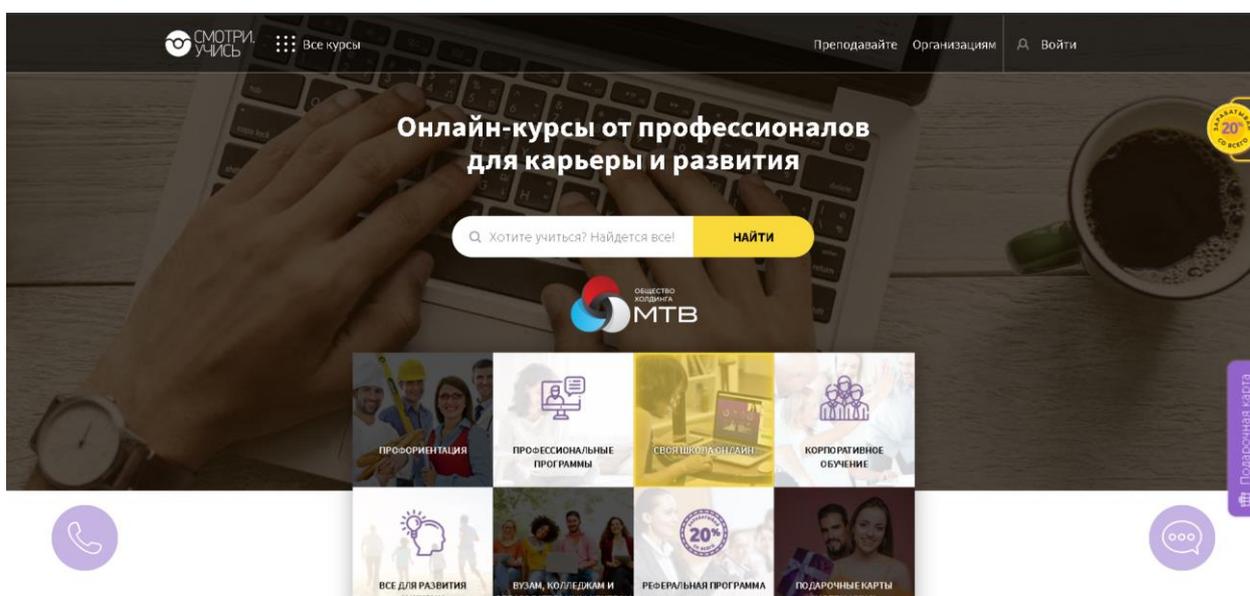


Рис. 1. Система «Смотри.Учись»

#### *Достоинства:*

- огромный выбор курсов
- авторизация через любую социальную сеть

#### *Недостатки:*

- мало бесплатных курсов
- неудобный дизайн

2. «InternetUrok»- веб-сервис с видеоуроками (см. Рис. 2). При приобретении абонемента, длительность которого месяц, открывается доступ к большинству уроков школьной программы.

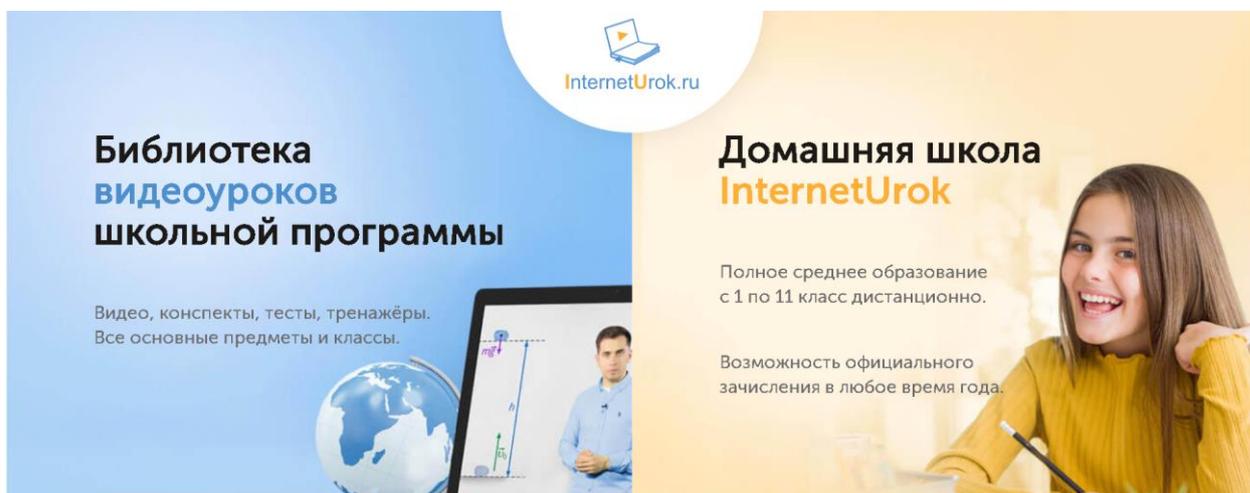


Рис. 2. Система «InternetUrok»

*Достоинства:*

- удобная навигация
- красивый дизайн

*Недостатки:*

- для доступа к урокам нужно оформлять подписку
- уроки содержат только школьную программу

3. «Яндекс.Учебник»- веб-сервис для обучения школьников 1-5 классов(см. Рис. 3).

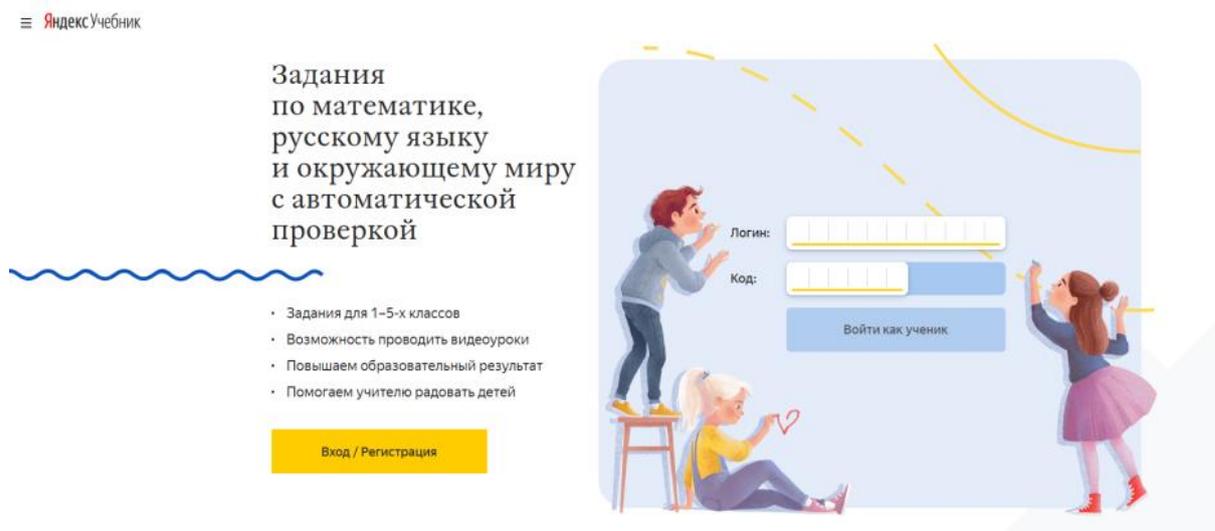


Рис. 3. Система «Яндекс.Учебник»

*Достоинства:*

- удобный поиск
- возможность регистрации в качестве родителя для наблюдения за успехами ребенка

*Недостатки:*

- уроки содержат только программу 1-5 классов

## 4. «Учи.Ру»- веб-сервис для интерактивного обучения (см. Рис. 4).

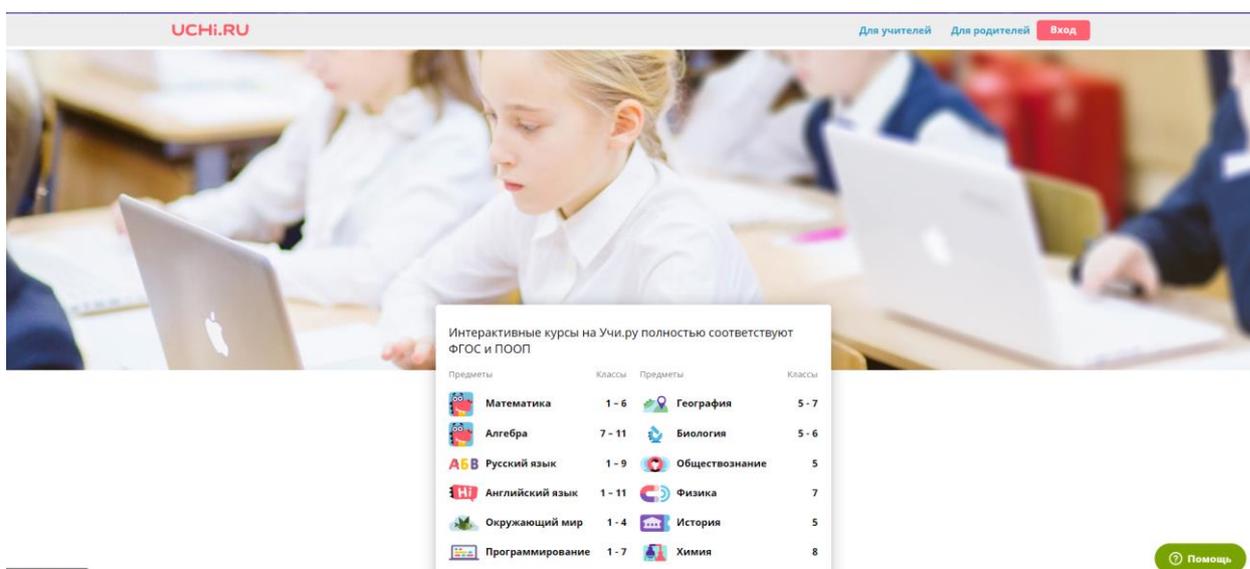


Рис. 4. Система «Учи.Ру»

*Достоинства:*

- Задания в игровой форме
- возможность регистрации в качестве родителя для наблюдения за успехами ребенка
- виртуальный класс для видеоуроков

*Недостатки:*

- уроки содержат только школьную программу

5. «Российская электронная школа»- интерактивные уроки по всему школьному курсу с 1 по 11 класс (см. Рис 5).

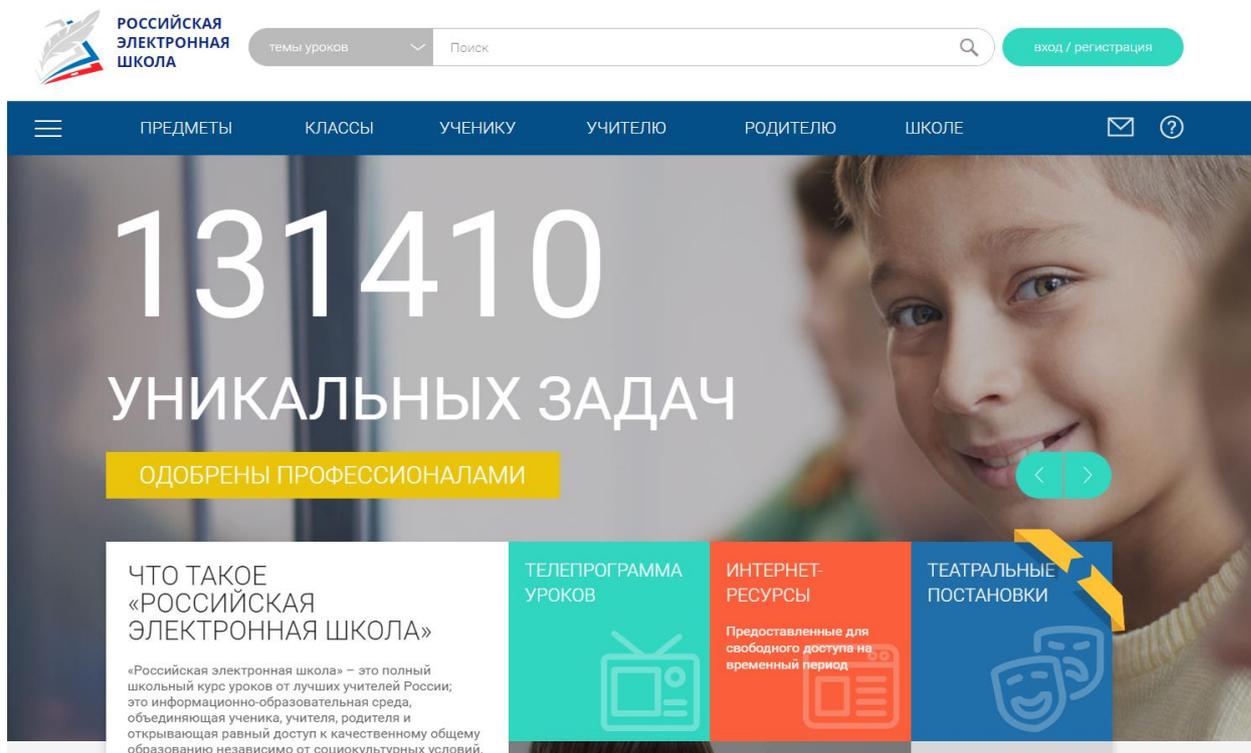


Рис. 5. Система «Российская электронная школа»

*Достоинства:*

- после каждого урока есть тренировочные и контрольные задания
- возможность регистрации в качестве родителя для наблюдения за успехами ребенка
- широкий выбор уроков

*Недостатки:*

- уроки содержат только школьную программу

#### **1.4. Обоснование выбора платформы для реализации информационной системы**

Было принято решение создать систему под названием «Смотри и Учись» для использования в учебном процессе в виде веб-сервиса, так как интернет дает широкие возможности для использования программы: установки дополнительного программного обеспечения не требуется и программа доступна с любого устройства, подключенного к сети.

Исходя из проектируемого функционала разрабатываемой информационной системы были определены следующие средства и технологии ее создания: язык веб-программирования PHP и система управления базами данных MySQL.

##### **Язык программирования PHP**

PHP – это интерпретируемый язык программирования общего назначения с открытым исходным кодом. Он был разработан специально для веб-разработок. Главное преимущество PHP-скриптов это возможность интегрироваться в обычные html-документы, сохраняя всю функциональность. Но для этого необходимо наличие на веб-сервере интерпретатора языка PHP.

Основной областью использования языка PHP считается написание скриптов, которые работают на стороне сервера. Кроме различных операций с содержимым ваших html-страниц, с помощью PHP можно форматировать HTTP-заголовки, управлять аутентификацией, а также перенаправлять пользователя на другие страницы. PHP предоставляет большие возможности по доступу к базам данных, поддерживает интеграцию с различными внешними библиотеками.

##### **Некоторые положительные стороны PHP:**

- весь код обрабатывается и исполняется на стороне сервера;

- поддерживает работу с множеством СУБД (MySQL, Oracle, PostgreSQL и т.д.);
- является программным обеспечением с открытым исходным кодом;
- работает на разных платформах (Windows, Linux, Unix);
- PHP очень прост для изучения;
- PHP очень эффективен, благодаря своему «движку», который является транслирующим интерпретатором, что позволяет с высокой скоростью обрабатывать сценарии;
- Бесплатное распространение
- поддержка различных веб-серверов, так как PHP не содержит кода, который ориентирован на какой-либо определенный сервер.

Таким образом, выбрав язык программирования PHP, мы получим огромные возможности по использованию, свободу выбора понравившегося веб-сервера и операционной системы.

### Язык дизайна CSS

CSS (англ. Cascading Style Sheets, каскадные таблицы стилей) — простой язык дизайна, который служит для упрощения процесса оформления веб-страниц.

CSS обрабатывает внешний вид веб-страницы. Воспользовавшись CSS, можно контролировать цвет текста, стиль шрифтов, размеры и расположение элементов страницы, фоновые изображения, макеты дизайна, варианты отображения на разных устройствах и размерах экрана.

### Преимущества CSS:

- CSS стоит написать один раз и использовать свою таблицу стилей на нескольких веб-страницах;
- Возможность написать стиль для любого HTML-элемента и применить его ко всем веб-страницам
- При внесении изменений все элементы на всех страницах обновятся автоматически;
- Имеет больший набор атрибутов, чем HTML;
- С помощью CSS можно просматривать сайты, находясь в оффлайн режиме, так как веб-приложения хранят CSS локально в кэше;

### СУБД MySQL

MySQL – свободная система управления базами данных. SQL – это сокращение от Structured Query Language (структурированный язык запросов). SQL создан для работы с реляционными базами данных, позволяющий пользователям работать с базами данных (просматривать, искать, изменять, добавлять и управлять данными).

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать таблицы типа MyISAM, которые поддерживают полнотекстовый поиск, или типа InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

MySQL портирована на большое количество платформ и имеет API для языков C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk и Tcl,

библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.

Основные положительные стороны MySQL:

- Многопоточность. Поддержка нескольких одновременных запросов;
- Интерфейс для языков PHP, Java, C, C++;
- Большое количество типов столбцов;
- Записи фиксированной и переменной длины;
- ODBC (Open-DataBase-Connectivity) драйвер в комплекте с исходником;
- Гибкая и безопасная система привилегий и паролей;
- Вся передача паролей шифрована;
- Поддержка ключевых полей и специальных полей в операторе;
- Основанная на потоках, быстрая система управления памятью;
- Утилита проверки и ремонта таблицы (myisamchk);
- Сервер выдает сообщения об ошибках и диагностику на разных языках;
- Все операции работы со строками не обращают внимания на регистр символов в обрабатываемых строках;
- Поддержка огромных объемов данных;
- Псевдонимы применимы к таблицам и к отдельным колонкам в таблиц;

- Все поля имеют значение по умолчанию, можно использовать на любом подмножестве полей;
- Легкость управления таблицами.

## Глава 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «КАТАЛОГ ВИДЕОФРАГМЕНТОВ»

### 2.1. Первый этап – концептуальное проектирование

Первым этапом процесса разработки базы данных является концептуальное проектирование. На данном этапе разрабатывается концептуальная модель данных, отражающая основные сущности, их свойства и взаимосвязи в рамках выбранной предметной области на основе данных, полученных от конечного пользователя. Обычно такие данные записываются в некоторых стандартизованных спецификациях требований. Концептуальное моделирование базы данных абсолютно не привязано к конкретным, выбранным ранее в качестве рабочих инструментов, СУБД и языков программирования. Обычно основной целью концептуального моделирования является определение удобных и доступных для конечного пользователя способов получения и обработки данных, которые необходимо сохранять и обрабатывать в создаваемой информационной системе средствами выбранной СУБД. Поэтому концептуальная модель строится как абстрактная формальная система на основе естественного языка.

В процессе разработки концептуальная модель подвергается постоянному мониторингу с целью проверки на соответствие требованиям конечного пользователя. Созданная концептуальная модель данных является источником информации и основой для построения логической модели данных.

К основным объектам концептуальной модели относятся *сущности* и *связи*.

Сущность (или таблица) – это множество различных экземпляров объектов, обладающих общими свойствами.

К сущностям выбранной предметной области относятся такие классы объектов как "Предметы", "Курсы" (см. Рис.6).

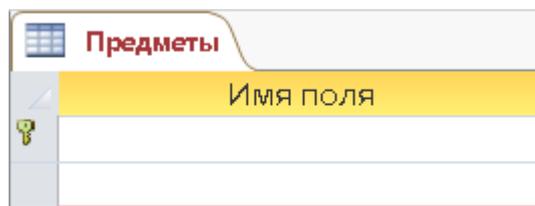


Рис. 6 Сущность

Атрибут сущности – это именованная характеристика, отражающая определенное свойство сущности. Атрибут именуется именем существительным, в именительном падеже, единственном числе. Допускается использование в качестве дополнения характеризующих прилагательных.

Например, атрибутами сущности "Предметы" могут быть "Код", "Предмет" и т.п. (см. Рис.7)

Атрибуты указываются в пределах прямоугольника, определяющего сущность.

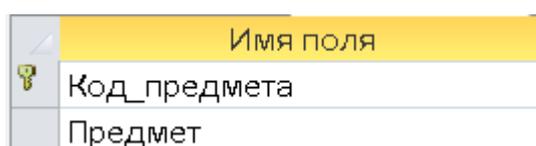
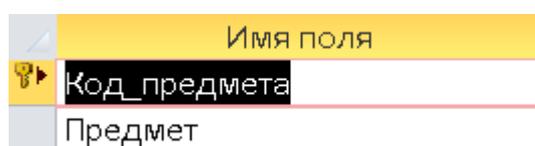


Рис. 7 Атрибуты сущности

Ключом сущности считается атрибут или набор атрибутов, имеющих уникальное значение для каждого экземпляра сущности (см. Рис 8).



## Рис. 8 Ключ сущности

Связью называется функциональная зависимость между сущностями. Благодаря связям становится возможным нахождение некоторой сущности по другим сущностям, связанным с ней.

Графически связь изображается в виде линии, связывающей две сущности или сущность с ней самой.

Принято различать следующие типы связей:

*один-к-одному*: один экземпляр одной сущности связан с одним экземпляром другой сущности;

*один-ко-многим*: один экземпляр одной сущности связан с несколькими экземплярами другой сущности;

*много-ко-многим*: каждый экземпляр одной сущности может быть связан с несколькими экземплярами другой сущности, и каждый экземпляр другой сущности может быть связан с несколькими экземплярами первой сущности.

## 2.2. Второй этап – логическое проектирование

На втором этапе разработки базы данных, как основного ядра информационной системы, в рамках логического проектирования создается логическая модель данных для исследуемой предметной области. Логическая модель уточняет и дополняет созданную ранее концептуальную модель.

Основными требованиями к данным, хранящимся и обрабатываемым в базе данных, являются требования их достоверности и непротиворечивости. Причем данные свойства должны не зависеть от срока работы системы, а также от производимых в процессе работы изменений, удалений и добавлений данных по отношению к изначальному состоянию базы данных.

Анализ выбранной предметной области позволил определить следующую логическую модель:

1. Студенты (Код Студента, Логин, Фамилия, Имя, Отчество, Пароль, Фото);
2. Преподаватели (Код Преподавателя, Логин, Фамилия, Имя, Отчество, Пароль);
3. Предметы (Код Предмета, Предмет);
4. Категории (Код Категории, Предмет, Категория);
5. Курсы (Код Курса, Предмет, Категория, Курс, Описание, Автор, Подписчики);
6. Видео (Код Видео, Название, Описание, Предмет, Категория, Курс, Автор, Дата, Файл);
7. Подписки (Код Подписки, Код Курса, Логин);

Каждой сущности соответствует отдельная таблица, создаваемая на основе логической модели. Первичным ключам сущностей ставятся в соответствие ключевые поля таблиц.

Логическая модель данных выбранной предметной области (см. Рис.9) является основой этапа физического проектирования и играет важную роль на этапе внедрения, эксплуатации и сопровождения готовой информационной системы. Правильно организованное сопровождение

информационной системы, поддерживаемой в актуальном и достоверном состоянии, позволяет отслеживать любые изменения вносимых данных и корректировать оказываемое ими влияние на уже имеющиеся в базе данные, а также на различного рода прикладные программы и инструменты.

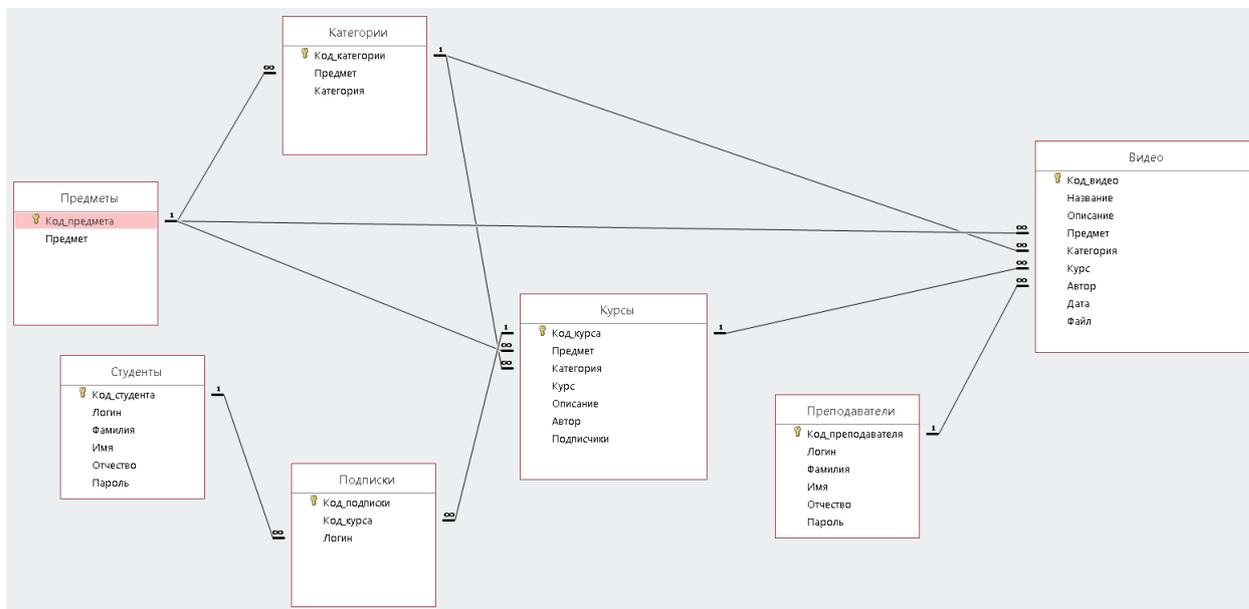


Рис. 9 Схема данных

### 2.3. Третий этап – физическое проектирование

На этапе физического проектирования разработчик информационной системы принимает решения о методах и способах ее реализации. Этот этап призван физически реализовать логическую модель, созданную на основе концептуальной модели. Физическое проектирование в рамках реляционной модели предполагает следующее:

- создаются реляционные таблицы и описываются их ограничения на основе данных, представленных в логической модели;
- определяется конкретная структура хранения данных и способы доступа к ним, обеспечивающие в совокупности максимальную производительность разрабатываемой системы;
- определяются и разрабатываются средства защиты создаваемой системы.

С точки зрения окончательного источника информации о данных, обрабатываемых системой, физическая модель содержит всю информацию о свойствах и поведении данных, а также описывает события, возникающие в системе. Обычно каждая модель представляет собой группу кодовых файлов, осуществляющих следующие действия:

- поиск и выбор данных из базы;
- вывод данных на экран или в файл;
- добавление данных в базу;
- редактирование данных;
- удаление данных из базы.

Наиболее приспособленной для применения в веб-среде является система управления базами данных MySQL в силу высокого уровня надежности и эффективности работы в условиях предоставления провайдерами небольшого количества ресурсов для исполнения приложений клиента на большинстве хостинг-площадок.

Представим через СУБД MySQL описанные на предыдущих этапах

проектирования системы таблицы выделенных сущностей и их атрибутов (см. Рис. 10-23):

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> categories		9	InnoDB	utf8mb4_unicode_ci	48 КИБ	-
<input type="checkbox"/> courses		3	InnoDB	utf8mb4_unicode_ci	64 КИБ	-
<input type="checkbox"/> student		2	InnoDB	utf8mb4_unicode_ci	16 КИБ	-
<input type="checkbox"/> subjects		8	InnoDB	utf8mb4_unicode_ci	32 КИБ	-
<input type="checkbox"/> subscribes		1	InnoDB	utf8mb4_unicode_ci	16 КИБ	-
<input type="checkbox"/> teacher		1	InnoDB	utf8mb4_unicode_ci	16 КИБ	-
<input type="checkbox"/> video		4	InnoDB	utf8mb4_unicode_ci	64 КИБ	-
<b>7 таблиц</b>	<b>Всего</b>	<b>28</b>	<b>InnoDB</b>	<b>utf8mb4_unicode_ci</b>	<b>256 КИБ</b>	<b>0 Байт</b>

Рис. 10 Структура базы данных «Каталог Видеофрагментов»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id_teacher	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/> 2	login	varchar(25)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 3	f	varchar(30)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 4	i	varchar(30)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 5	o	varchar(30)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 6	password	varchar(25)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 7	foto	varchar(40)	utf8mb4_unicode_ci		Нет	Нет		

Рис. 11 Структура таблицы «Преподаватели»

Индексы						
Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	
	PRIMARY	BTREE	Да	Нет	id_teacher	
	login	BTREE	Да	Нет	login	

Рис. 12 Индексы таблицы «Преподаватели»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id_student	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/> 2	login	varchar(25)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 3	f	varchar(30)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 4	i	varchar(30)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 5	o	varchar(30)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 6	password	varchar(25)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 7	foto	varchar(30)	utf8mb4_unicode_ci		Нет	Нет		

Рис. 13 Структура таблицы «Студенты»

Индексы					
Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
 	PRIMARY	BTREE	Да	Нет	id_student
 	login	BTREE	Да	Нет	login

Рис. 14 Индексы таблицы «Студенты»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id_subject 	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/> 2	subject 	varchar(50)	utf8mb4_unicode_ci		Нет	Нет		

Рис. 15 Структура таблицы «Предметы»

Индексы					
Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец
 	PRIMARY	BTREE	Да	Нет	id_subject
 	subject	BTREE	Да	Нет	subject

Рис. 16 Индексы таблицы «Предметы»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/> 1	id_category 	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/> 2	subject 	varchar(50)	utf8mb4_unicode_ci		Нет	Нет		
<input type="checkbox"/> 3	category 	varchar(50)	utf8mb4_unicode_ci		Нет	Нет		

Рис. 17 Структура таблицы «Категории»

Индексы									
Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
 	PRIMARY	BTREE	Да	Нет	id_category	8	A	Нет	
 	category	BTREE	Да	Нет	category	8	A	Нет	
 	subject	BTREE	Нет	Нет	subject	8	A	Нет	

Рис. 18 Индексы таблицы «Категории»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	<b>id_course</b>	int(11)		Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	<b>subject</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	3	<b>category</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	4	<b>course</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	5	<b>description</b>	varchar(100)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	6	<b>author</b>	varchar(25)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	7	<b>subscribes</b>	int(4)		Нет	Нет		

Рис. 19 Структура таблицы «Курсы»

Индексы			
Действие	Имя индекса	Тип	Уникальный
	<b>PRIMARY</b>	BTREE	Да
	<b>category</b>	BTREE	Нет
	<b>subject</b>	BTREE	Нет
	<b>course</b>	BTREE	Нет

Рис. 20 Индексы таблицы «Курсы»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	<b>id_subscribe</b>	int(11)		Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	<b>id_sub</b>	varchar(10)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	3	<b>login_subscriber</b>	varchar(25)	utf8mb4_unicode_ci	Нет	Нет		

Рис. 21 Структура таблицы «Подписки»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	<b>id_video</b>	int(11)		Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	<b>name</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	3	<b>description</b>	varchar(100)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	4	<b>subject</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	5	<b>category</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	6	<b>course</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	7	<b>author</b>	varchar(25)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	8	<b>date_v</b>	date		Нет	Нет		
<input type="checkbox"/>	9	<b>file</b>	varchar(50)	utf8mb4_unicode_ci	Нет	Нет		
<input type="checkbox"/>	10	<b>for_lent</b>	varchar(1)	utf8mb4_unicode_ci	Нет	Нет		

Рис. 22 Структура таблицы «Видео»

Индексы						
Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	
	PRIMARY	BTREE	Да	Нет	id_video	
	subject	BTREE	Нет	Нет	subject	
	category	BTREE	Нет	Нет	category	
	video_ibfk_3	BTREE	Нет	Нет	course	

Рис. 23 Индексы таблицы «Видео»

Схема базы данных представлена на Рисунке 24.

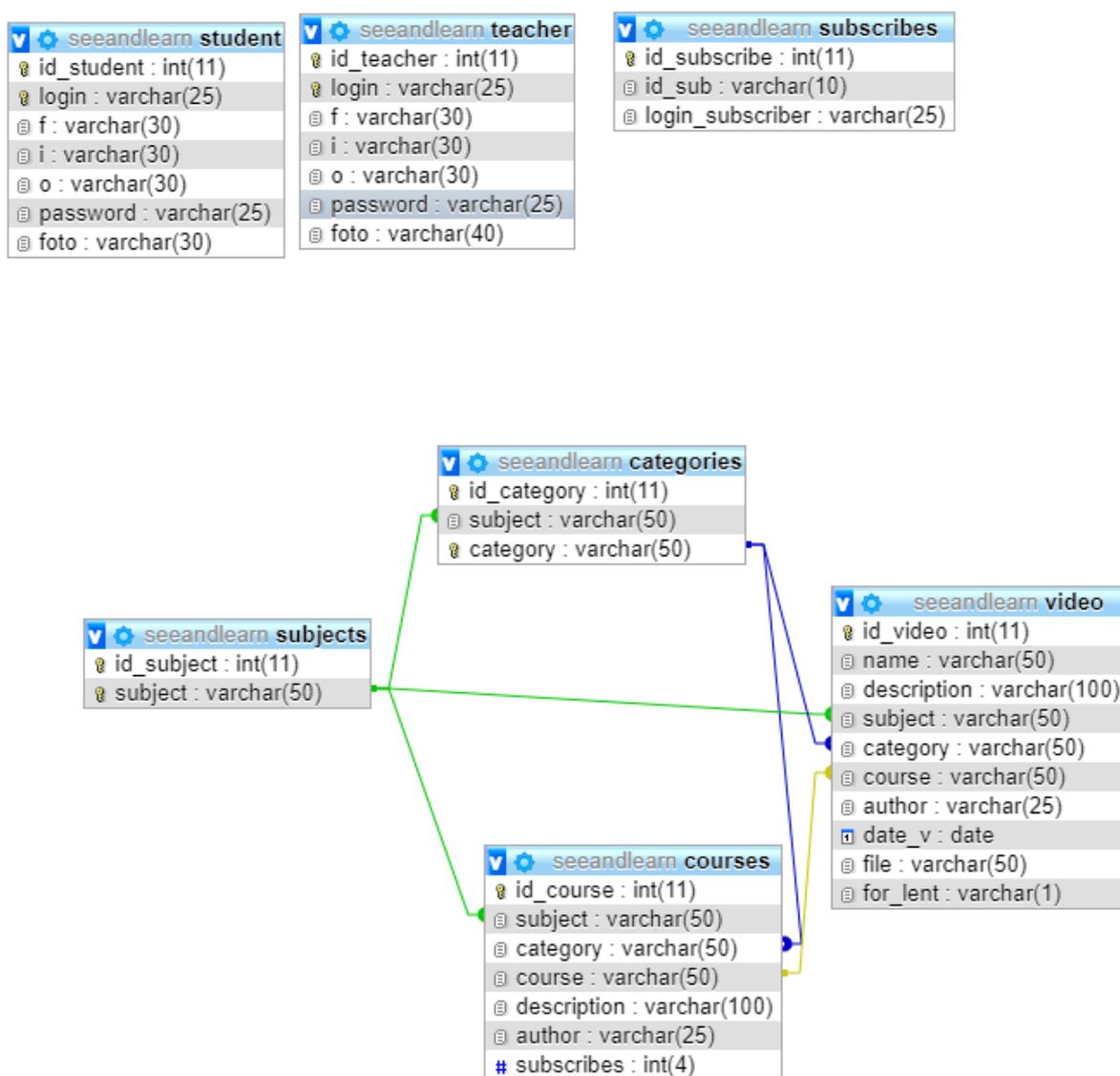


Рис. 24 Схема базы данных «Каталог видеофрагментов» в MySQL

## Глава 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

### 3.1. Физическая структура программного продукта

Опишем физическую структуру программного продукта «Каталог видеофрагментов», представляющую собой совокупность файлов и папок системы (см. Рис. 25). В корневой папке находятся: файлы основных разделов системы, главная страница, страницы авторизации, регистрации, курса, добавления файлов, поиска, параметры подключения к базе данных, таблица стилей. Файлы с фотографиями и видео хранятся в отдельных папках. Такая структура позволяет легко найти нужный файл.

Имя	Дата изменения	Тип	Размер
css	31.03.2020 14:05	Папка с файлами	
font	23.03.2020 17:26	Папка с файлами	
photo	03.05.2020 10:14	Папка с файлами	
video	14.05.2020 16:38	Папка с файлами	
addcourse	04.05.2020 12:23	Файл "PHP"	2 КБ
captcha	23.03.2020 17:26	Файл "PHP"	1 КБ
connectbase	15.03.2020 14:18	Файл "PHP"	1 КБ
course	14.05.2020 13:49	Файл "PHP"	3 КБ
delcourse	04.05.2020 12:23	Файл "PHP"	1 КБ
delvideo	14.05.2020 10:31	Файл "PHP"	1 КБ
desub	09.04.2020 17:07	Файл "PHP"	1 КБ
editcourse	14.05.2020 13:36	Файл "PHP"	4 КБ
editvideo	14.05.2020 13:44	Файл "PHP"	2 КБ
fileupload	14.05.2020 16:01	Файл "PHP"	2 КБ
header	14.05.2020 16:08	Файл "PHP"	5 КБ
login	09.04.2020 17:08	Файл "PHP"	3 КБ
logout	09.04.2020 17:08	Файл "PHP"	1 КБ
main	12.04.2020 16:29	Файл "PHP"	3 КБ
new_user	14.05.2020 13:26	Файл "PHP"	3 КБ
newcourse	14.05.2020 14:14	Файл "PHP"	2 КБ
photoupload	09.04.2020 17:08	Файл "PHP"	1 КБ
registration	09.04.2020 15:18	Файл "PHP"	3 КБ
request	14.05.2020 13:44	Файл "PHP"	13 КБ
savecourse	14.05.2020 10:49	Файл "PHP"	2 КБ
savevideo	14.05.2020 13:47	Файл "PHP"	2 КБ
search	09.04.2020 18:05	Файл "PHP"	1 КБ
subscribe	09.04.2020 17:07	Файл "PHP"	1 КБ
usercourses	09.04.2020 17:06	Файл "PHP"	1 КБ
video	04.05.2020 12:15	Файл "PHP"	2 КБ

Рис. 25 Физическая структура разрабатываемой системы

addcourse.php – страница для добавления курса;

captcha.php – файл, формирующий картинку для проверки пользователя, что он человек;

connectbase.php – файл содержащий данные о подключении к базе данных;

course.php – страница курса;

delcourse.php – страница для удаления курса;

delvideo.php – страница для удаления видео;

desub.php – файл, оформляющий отписку пользователя от курса;

editcourse.php – страница для изменения курса;

editvideo.php – страница для изменения видео;

fileupload.php – страница с загрузкой видео;

header.php – шапка сайта;

login.php – файл, организующий авторизацию пользователя;

logout.php – файл, позволяющий пользователю выйти и закончить сессию;

main.php – главная страница сайта;

new\_user.php – файл, добавляющий нового пользователя;

newcourse.php – файл, добавляющий новый курс;

photoupload.php – страница с загрузкой фото пользователя;

registration.php – страница регистрации;

request.php – файл, содержащий все функции с sql запросами для работы всего сайта с базой данных;

savecourse.php – файл, осуществляющий сохранение изменений в названии и описании курса;

savevideo.php – файл, осуществляющий сохранение изменений в названии и описании видео;

search.php – страница поиска;

subscribe.php – файл, оформляющий подписку пользователя от курса;

usercourses.php – страница с курсами пользователя;

video.php – страница воспроизведения видео.

Содержание отдельных папок представлено на Рис. 26-29:

Имя	Дата изменения	Тип	Размер
l	31.03.2020 14:05	Файл "PNG"	30 КБ
nice	09.04.2020 16:17	Файл "CSS"	7 КБ

Рис. 26 Содержание папки «css»

l.png – логотип сайта;

nice.css – таблица стилей сайта;

Имя	Дата изменения	Тип	Размер
Karate	06.06.2009 18:45	Файл шрифта Ttu...	21 КБ

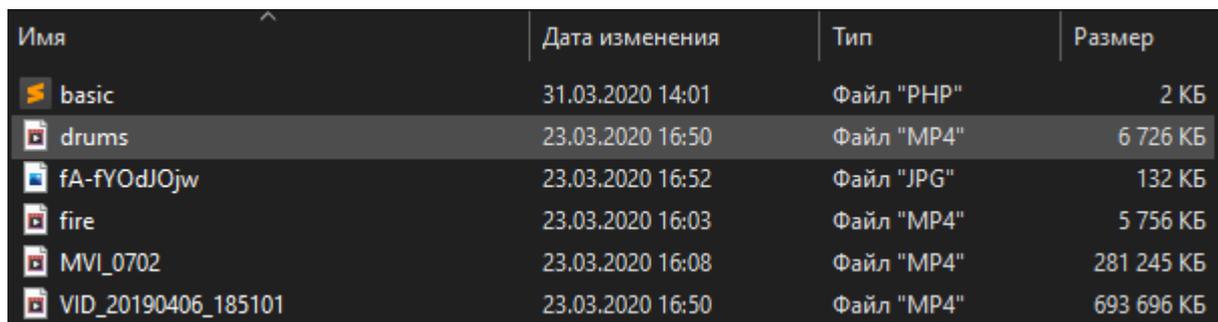
Рис. 27 Содержание папки «font»

karate.ttf – шрифт для картинки на проверку, что пользователь человек;

Имя	Дата изменения	Тип	Размер
C3eHYyjNm2l	26.03.2020 14:09	Файл "JPG"	1 792 КБ
DSC_0024	31.03.2020 11:54	Файл "JPG"	6 880 КБ
nofoto	26.03.2020 13:29	Файл "JPG"	26 КБ
photoload	31.03.2020 11:53	Файл "PHP"	2 КБ
yum	09.04.2020 16:43	Файл "JPG"	72 КБ

Рис. 28 Содержание папки «foto»

photoload.php – файл, осуществляющий загрузку фото пользователя на сервер;



Имя	Дата изменения	Тип	Размер
basic	31.03.2020 14:01	Файл "PHP"	2 КБ
drums	23.03.2020 16:50	Файл "MP4"	6 726 КБ
fA-fYOdJOjw	23.03.2020 16:52	Файл "JPG"	132 КБ
fire	23.03.2020 16:03	Файл "MP4"	5 756 КБ
MVI_0702	23.03.2020 16:08	Файл "MP4"	281 245 КБ
VID_20190406_185101	23.03.2020 16:50	Файл "MP4"	693 696 КБ

Рис. 29 Содержание папки «video»

basic.php – файл, на котором происходит загрузка видео пользователя на сервер.

### 3.2. Интерфейс информационной системы

Любая современная информационная система предполагает выполнение требования интерактивности. Общение с пользователем осуществляется посредством пользовательского интерфейса. К интерфейсу также предъявляется ряд требований, среди которых наиболее важными являются следующие:

- максимальное соответствие целям конечного пользователя;
- исключение решения посторонних задач, не относящихся напрямую или косвенно к выделенной предметной области;
- оптимальное соотношение функционала системы и удобства и эргономичности его использования;
- разделение функционала системы на категории во избежание путаницы направлений деятельности пользователей.

В данном проекте пользовательский интерфейс разработан в едином стиле. В верхней части окна находится шапка сайта, где расположена навигация, название системы и кнопку для авторизации в системе. Центральную часть занимает рабочая область, осуществляющая работу с данными.

#### Эргономичность интерфейса системы

Веб-сервис «Смотри и Учись», предназначенный для использования видеофрагментов в учебном процессе, имеет понятный и удобный интерфейс, рассчитанный на работу с неподготовленными пользователями, имеющими минимальные навыки работы с информационными технологиями.

Основными критериями эргономичности информационных систем, выполненных в форме веб-разработок, могут служить следующие:

1. Естественность (интуитивность). У пользователя не возникает психологического барьера и ощущения неудобства при работе с системой.

2. Неизбыточность информации. Для работы с системой пользователю требуется вводить минимум информации (регистрация, авторизация, обратная связь). Данное требование конкретизируется необходимостью защиты персональных данных.

3. Непротиворечивость. Приёмы работы со схожими элементами интерфейса и структурами данных должны быть реализованы одинаково и в соответствии с установленными нормами (например, использование клавиш Tab или Ctrl+Tab перемещает курсор в следующее или предыдущее поле/элемент).

4. Гибкость. Интерфейс подстраивается под уровень квалификации пользователя: для неопытного пользователя организуется как иерархическая структура меню, для профессионала – как команды, параметры и комбинации нажатий клавиш.

#### Меню системы

Меню системы организуется понятно и легко в использовании. Опции меню упорядочиваются по смыслу решаемых задач и предоставляемых возможностей. При разработке меню следует придерживаться определенных правил, например:

- 1) структура опций меню соответствует структуре решаемой задачи;
- 2) порядок опций отражает наиболее эффективную последовательность шагов для достижения поставленной цели;

2) выбор опций для перемещения по программе обеспечивается как при помощи мыши, сенсорных устройств, так и при помощи клавиатуры.

Ниже показано окно информационной системы «Смотри и Учись» для использования видеофрагментов в учебном процессе (см.Рис.30).

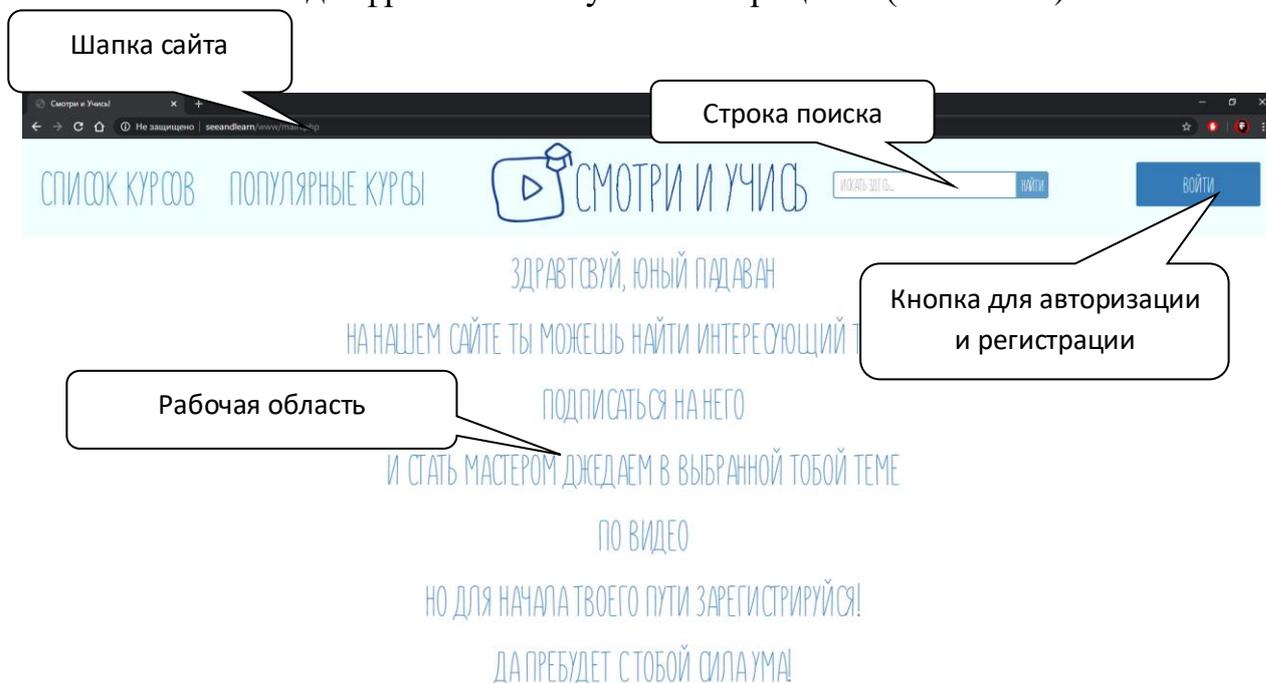


Рис.30 Вид системы с описанием элементов пользовательского интерфейса

### 3.3. Функционал системы

При входе на сайт для незарегистрированного пользователя на главной странице выводится сообщение (см. Рис.31):

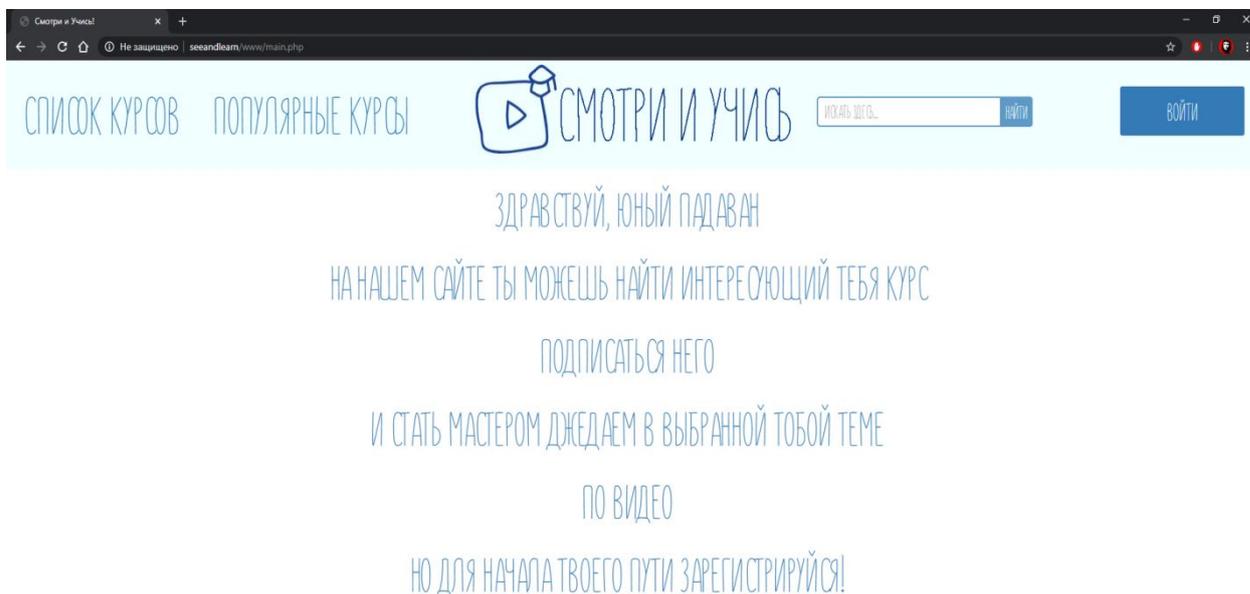


Рис.31. Главная страница для незарегистрированного пользователя

При нажатии на кнопку «Войти» на шапке сайта, открывается окошко «Авторизации», в котором присутствуют поля для ввода «Логина» и «Пароля», а также кнопки «Войти», «Регистрация» и «Заккрыть» (см. Рис.32).

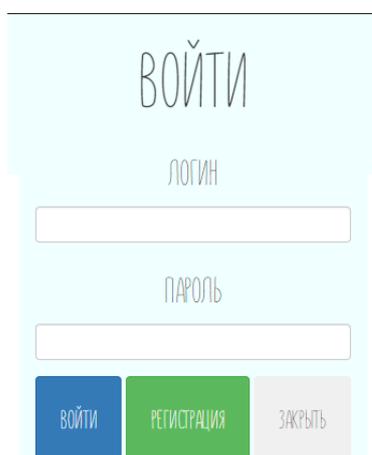


Рис.32 Окно авторизации

При успешной «Авторизации», сайт затемняется и в центре экрана появляется окошко с сообщением «Вы успешно вошли!» и кнопкой «Заккрыть окно» (см. Рис.33)

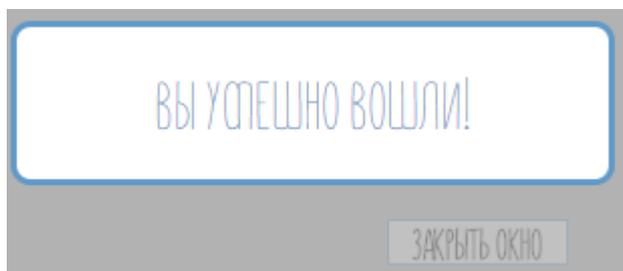


Рис.33 Окно с сообщением

При нажатии на кнопку «Регистрация» в окне авторизации, происходит переход на страницу регистрации, где пользователю предложено ввести логин, пароль и ФИО, выбрать кем является пользователь, студентом или преподавателем, а также пройти проверку является ли пользователь человеком или нет, введя цифры с генерируемой картинки (см. Рис.34).

A screenshot of a web browser showing the registration page. The browser's address bar shows "seeandlearn/www/registration.php". The page has a light blue header with navigation links: "СПИСОК КУРСОВ", "ПОПУЛЯРНЫЕ КУРСЫ", a play button icon with the text "СМОТРИ И УЧИСЬ", a search bar with "ИЗДАТЬ ЗАЯВ...", and a "ВОЙТИ" button. The main content area is titled "РЕГИСТРАЦИЯ" and contains several input fields: "ВАШ ЛОГИН:", "ВАШ ПАРОЛЬ:", "ВАША ФАМИЛИЯ:", "ВАШЕ ИМЯ:", "ВАШЕ ОТЧЕСТВО:". Below these is a "ВЫБЕРИТЕ ТИП:" section with radio buttons for "СТУДЕНТ" and "ПРЕПОДАВАТЕЛЬ". At the bottom, there is a "ВВЕДИТЕ КОД С КАРТИНКИ:" section with a CAPTCHA image showing the number "7604" and a "ЗАРЕГИСТРИРОВАТЬСЯ" button.

Рис.34 Страница регистрации

Если пользователь оставит какое-либо поле пустым, будет выведено сообщение «Заполните все поля!» (см. Рис.35)



Рис.35 Окошко с сообщением

Если пользователь неправильно введет данные с картинки, будет выведено «Неправильно введена картинка!» (см. Рис.36)

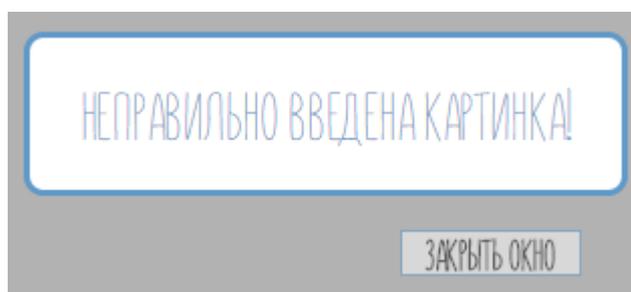


Рис.36 Окошко с сообщением

После успешного завершения регистрации будет выведено сообщение «Вы успешно зарегистрировались!» и пользователь будет добавлен в базу данных. (см. Рис.37)



Рис.37 Окошко с сообщением

Как только пользователь авторизуется, кнопка «Войти» превращается в кнопку «Выйти», а над кнопкой написан логин пользователя; также в шапке сайта выводится фотография пользователя, при наведении на которую курсором мыши, появляются кнопки «Загрузить фотографию» и «Мои курсы» (см. Рис.38)

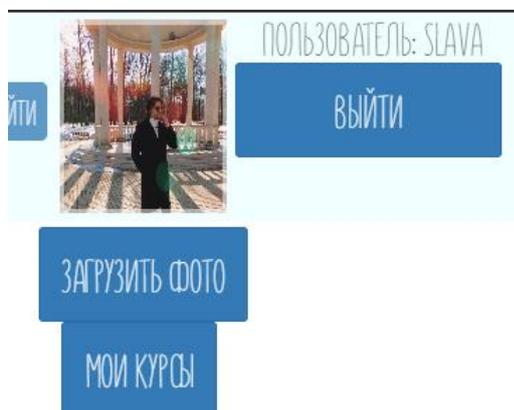


Рис.38 Правый угол шапки сайта после авторизации пользователя

В рабочей области сайта после авторизации появляются слова приветствия и лента подписок пользователя на курсы (см. Рис.39)

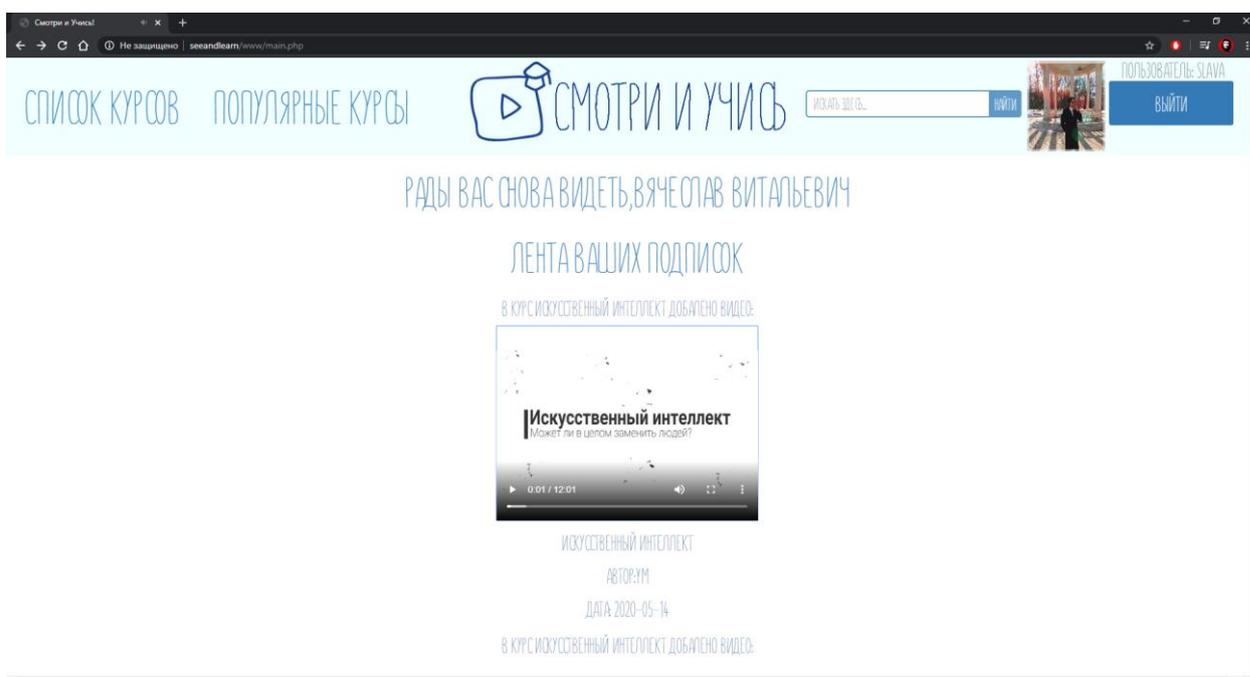


Рис.39 Лента подписок

При нажатии на кнопку «Загрузить фото», пользователю предлагается выбрать файл и загрузить его, тем самым, обновив свою фотографию на сайте. (см. Рис.40)

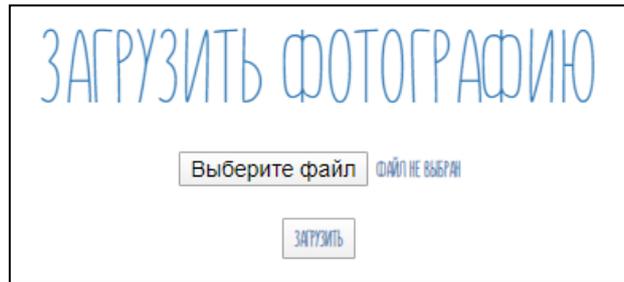


Рис.40 Лента подписок

При нажатии на кнопку «Мои курсы», пользователя переносит на страницу со списком курсов, на которые пользователь подписан. (см. Рис.41)

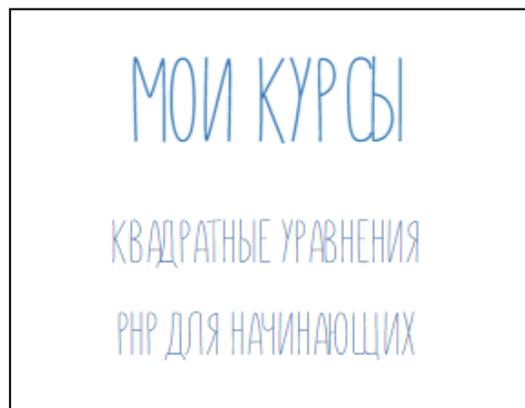


Рис.41 Лента подписок

При наведении курсора мыши на «Список курсов» всплывает многоуровневый список предметов с такими же всплывающими списками внутри себя (категории и курсы) (см. Рис.42).

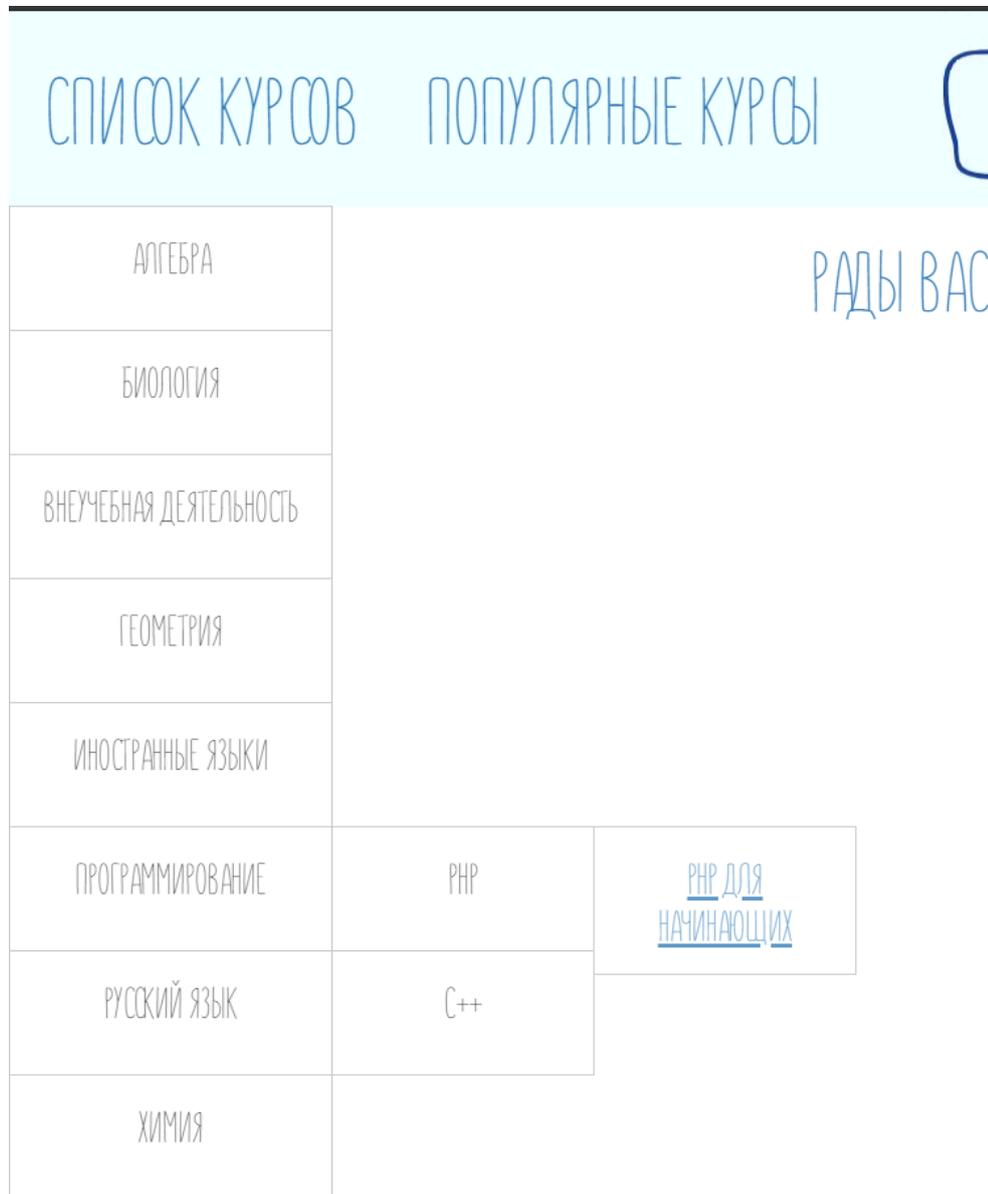


Рис.42 Многоуровневый список курсов

При наведении курсора мыши на «Популярные курсы» выводится список курсов отсортированных по количеству подписчиков по убыванию (см. Рис.43)



Рис.43 Список популярных курсов

При нажатии на название курса происходит переход на его страницу, на котором выводится его название, описание, количество подписчиков, всех видео, присутствующих в этом курсе, а также кнопка «Подписаться» (см. Рис.44)

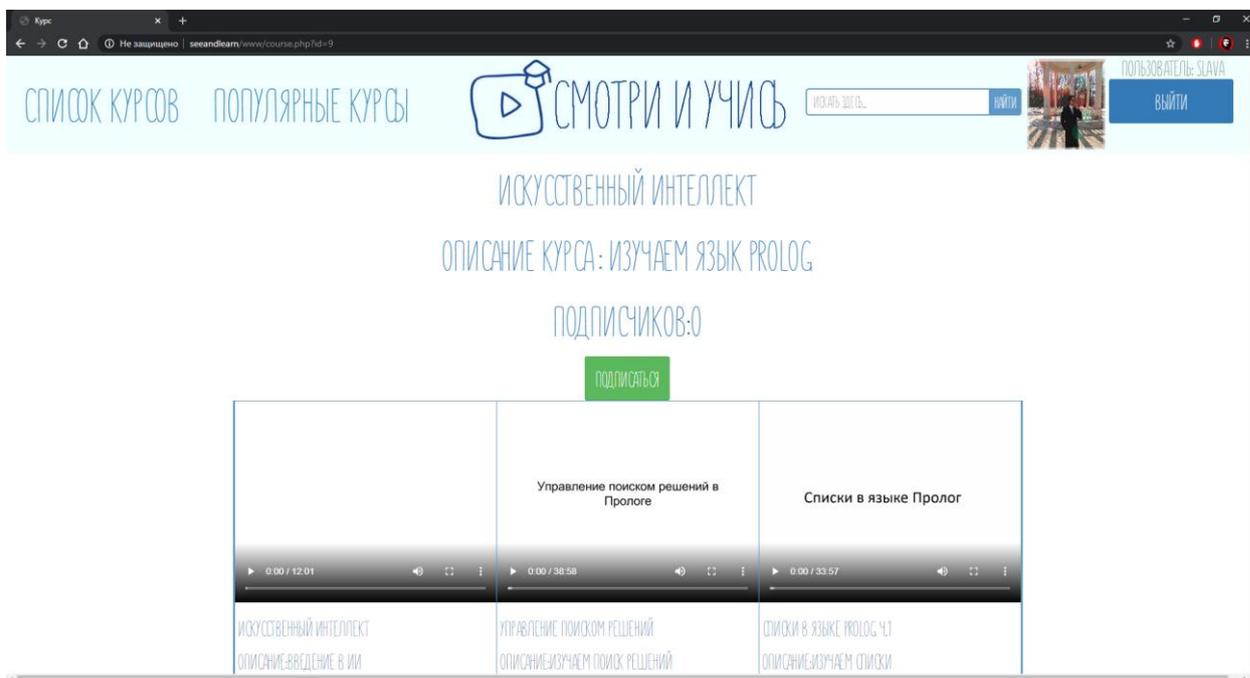


Рис.44 Страница курса

При нажатии на кнопку «Подписаться» курс добавляется в курсы пользователя, появляется окно с сообщением «Вы успешно подписались!» (см. Рис.45), и кнопка «Подписаться» заменяется кнопкой с функцией «Отписаться» (см. Рис.46), принцип действия которой прямо противоположен

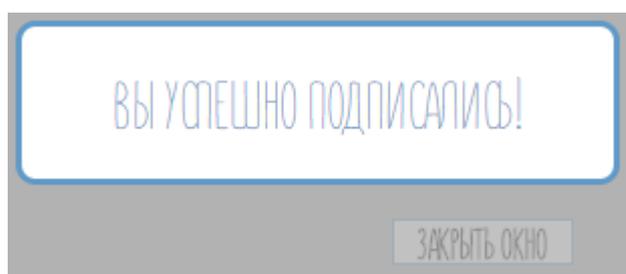


Рис.45 Окно с сообщением

ПОДПИСЧИКОВ:1

ОТПИСАТЬСЯ

Рис.46 Результат действия кнопки «Подписаться»

При нажатии на любое из предложенных видео открывается другая страница с названием видео, его описанием и автором, и, самое главное, с большим размером плеера (см. Рис.47)

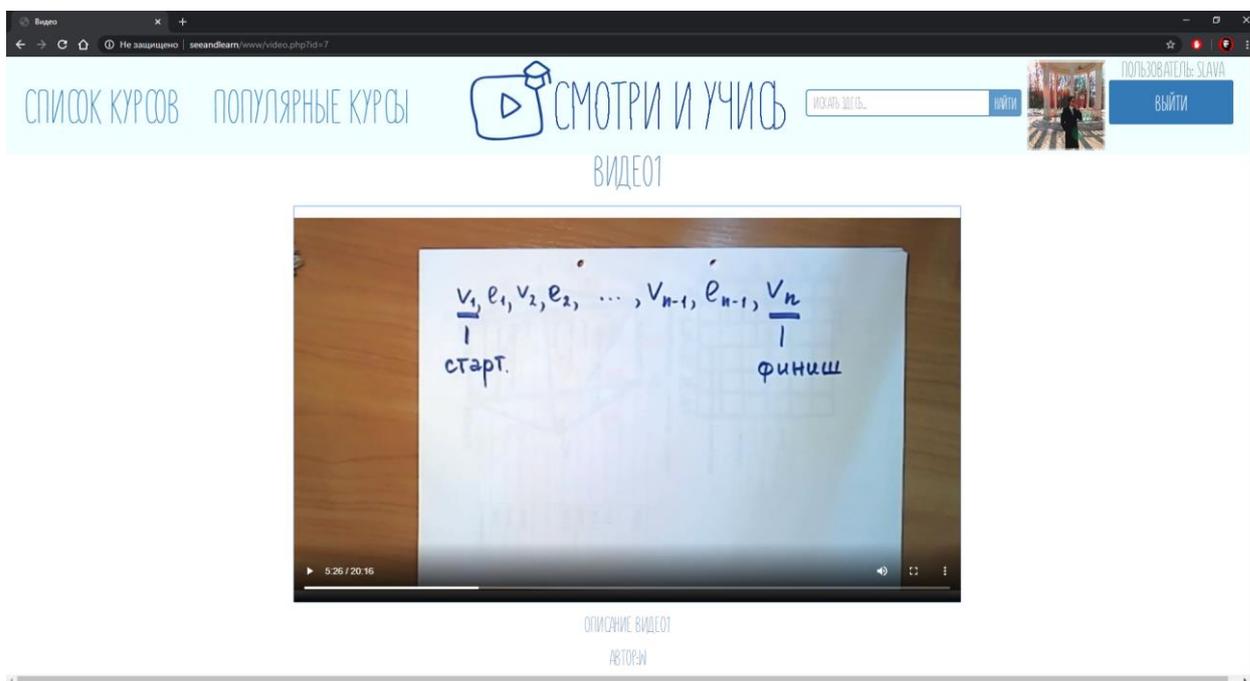


Рис.47 Страница с видео

Если пользователь зарегистрирован, как преподаватель, то на главной странице помимо ленты подписок, появляется кнопка «Создать курс» (см. Рис.48)

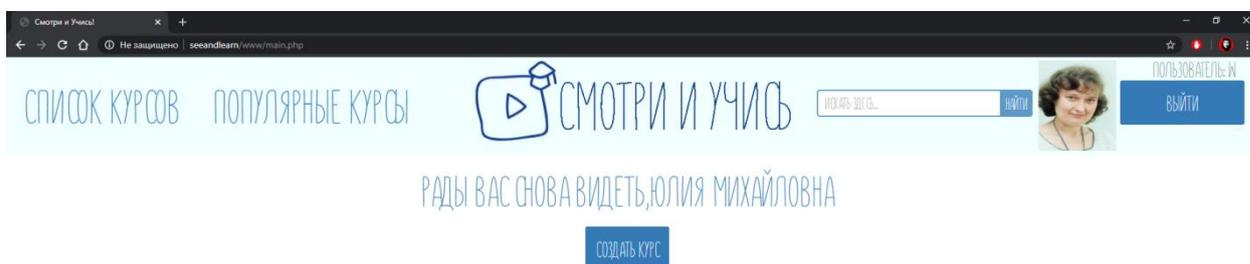


Рис.48 Главная страница для пользователя-преподавателя

При нажатии на кнопку «Создать курс», пользователя переносит на страницу с полями, куда нужно ввести данные создаваемого курса (выбрать предмет, категорию, заполнить название и описание) (см. Рис.49)

### СОЗДАНИЕ КУРСА

ВЫБЕРЕТЕ ПРЕДМЕТ:

ВЫБЕРЕТЕ КАТЕГОРИЮ:

НАЗВАНИЕ КУРСА:

ОПИСАНИЕ КУРСА:

Рис.49 Страница создания курса

На странице курса, если пользователь является преподавателем, а также автором данного курса, то появляются кнопки «Добавить файл», «Изменить» и «Удалить курс» (см. Рис.50)

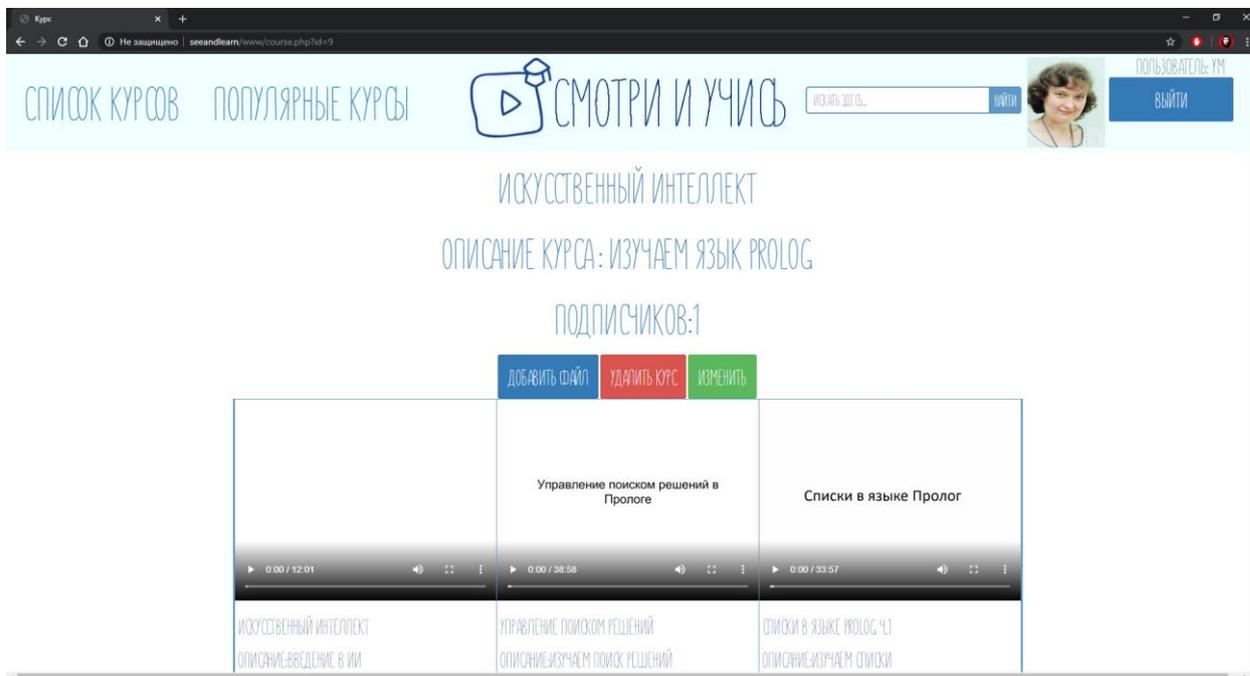


Рис.50 Страница курса для его автора

При нажатии на кнопку «Добавить файл», происходит переход на другую страницу, где пользователю перед тем, как загрузить видео, предлагается воспользоваться полезными онлайн-сервисами, если ему требуется отредактировать свое видео, если это не требуется, то нужно заполнить название и описание загружаемого видео. Загрузить видео можно двумя способами либо выбрать файл с диска, либо вставить ссылку на нужное видео (см. Рис.51)

ЗАГРУЗИТЬ ВИДЕО

ЕСЛИ ВАМ НУЖНО ОТРЕДАКТИРОВАТЬ ВИДЕО,  
ТО ПРЕДЛАГАЕМ ВАМ ВОСПОЛЬЗОВАТЬСЯ  
ЭТИМИ ОНЛАЙН-СЕРВИСАМИ:

ОНЛАЙН ОБРЕЗКА ВИДЕО  
ОНЛАЙН РЕДАКТИРОВАНИЕ ВИДЕО

ЧТОБЫ ЗАГРУЗИТЬ ВИДЕО, ВЫБЕРИТЕ ФАЙЛ

Выберите файл ФАЙЛ НЕ ВЫБРАН

ИЛИ

ВВЕДИТЕ ССЫЛКУ

ВВЕДИТЕ НАЗВАНИЕ

ОПИСАНИЕ ВИДЕО:

Рис.51 Загрузка видео

При нажатии на кнопку «Удалить курс» пользователю выведется сообщение о том, что курс успешно удален (см. Рис 52).

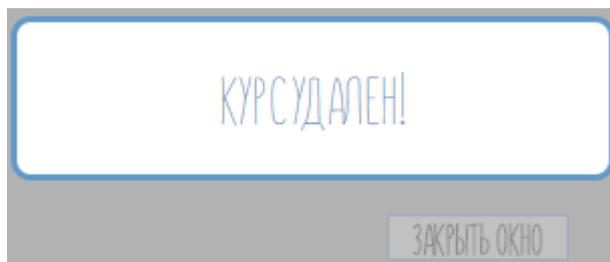


Рис.52 Сообщение об удалении курса

При нажатии на кнопку «Изменить» открывается страница с информацией данного курса, которую можно изменить, под кнопкой «Сохранить» располагается список видео, принадлежащих этому курсу, каждый из которых можно также удалить или изменить, нажав на соответствующие кнопки. (см. Рис 53)

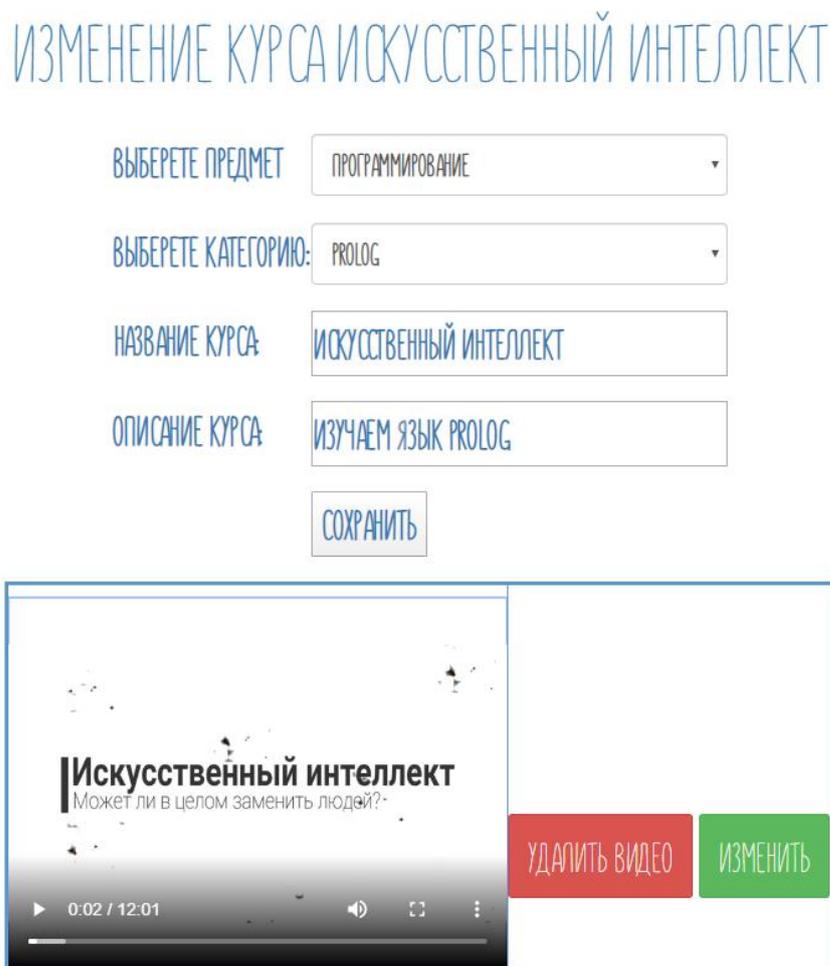


Рис.53 Изменение данных курса

На странице с видео, если пользователь является его автором, появляется возможность изменить название и описание, а также удалить это видео (см. Рис.54)

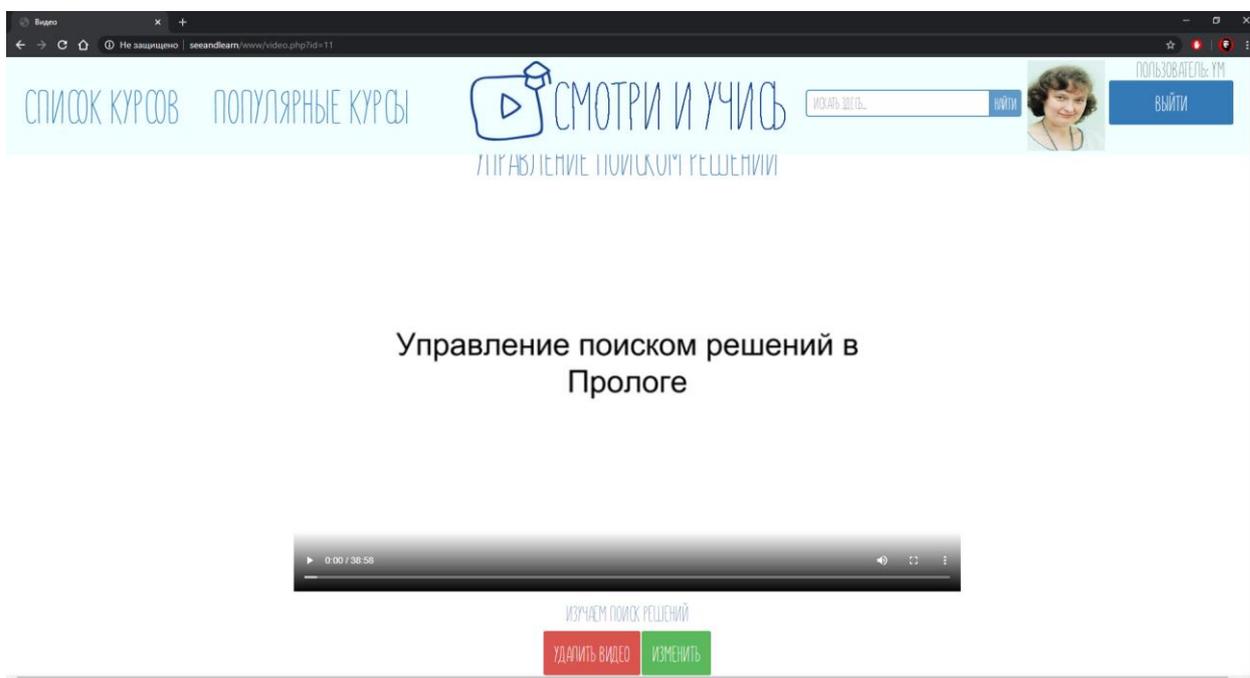


Рис.54 Страница с видео для его автора

При нажатии на кнопку «Удалить видео» будет выведено сообщение о том, что видео удалено (см. Рис 55).

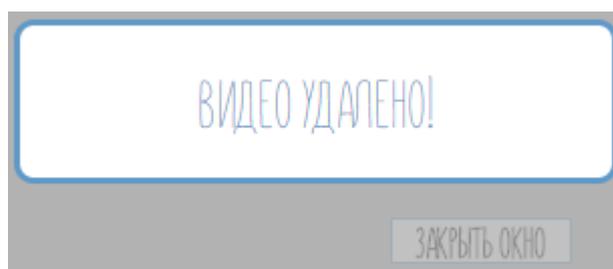


Рис.55 Сообщение об удалении видео

При нажатии на кнопку «Изменить» пользователю предлагается ввести новые данные видео, а именно название и описание, и нажать на кнопку «Сохранить» (см. Рис 56).

## ИЗМЕНЕНИЕ ИНФОРМАЦИИ ВИДЕО ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

НАЗВАНИЕ ВИДЕО:

ОПИСАНИЕ ВИДЕО:

Рис.56 Сообщение об удалении видео

В шапке сайта есть строка поиска, в которую пользователь может ввести то, что ему нужно найти (см. Рис.57)

Рис.57 Строка поиска

При нажатии на кнопку «Найти» происходит переход на страницу с результатами поиска (см. Рис.58)

ВОТ, ЧТО УДАЛОСЬ НАЙТИ:

PHP ДЛЯ НАЧИНАЮЩИХ

C++ ДЛЯ НАЧИНАЮЩИХ

Рис.58 Результаты поиска

### **3.4. Тестирование системы**

На этапе тестирования системы выявляются ошибки, неточности и недостатки системы, препятствующие ее нормальной работе. Выявленные ошибки подлежат исправлению, а недостатки – устранению.

Тестирование относится к важным инструментам анализа системы. Вместе с тем, человеческий фактор, присутствующий на каждом из этапов проектирования и внедрения информационной системы, способен отрицательно влиять на процесс полного устранения ошибок и неточностей программного обеспечения. Конечная цель тестирования программного продукта, проводимого в любом виде, определяется как обеспечение его качества и надежности. При этом, учитываются все наиболее значимые факторы и составляющие конкретного программного продукта.

Тестирование информационной системы «Каталог видеофрагментов» было успешно проведено на базе ЦО №44 в процессе внедрения системы в организацию учебного процесса. Справка о внедрении прилагается (см. Рис. 59).

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ  
 АДМИНИСТРАЦИИ  
 ГОРОДА ТУЛЫ  
 муниципальное бюджетное  
 общеобразовательное учреждение -  
 «центр образования №44  
 имени Маршала  
 Советского Союза Г.К. Жукова»  
 300036, г.Тула, ул.Одоевское шоссе, 106  
 Тел./факс: 39-25-27  
 06.05.2020г. № 224

ФГБОУ ВО  
 «Тульский государственный  
 педагогический университет  
 им. Л. Н. Толстого»  
 Факультет математики физики  
 и информатики

### О внедрении результатов работы

#### СПРАВКА

Выдана студенту 4 курса очной формы обучения факультета математики, физики и информатики ФГБОУ ВО «ТГПУ им. Л. Н. Толстого» Чекрыжову Вячеславу Витальевичу в том, что в МБОУ ЦО № 44 (корпус 1) г. Тулы в 2019-2020 учебном году были внедрены в работу школы материалы выпускной квалификационной работы на тему «Разработка и использование информационной системы «Каталог видеофрагментов» для использования в учебном процессе».

Директор МБОУ ЦО №44



Рис.59 Справка о внедрении информационной системы

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения выпускной квалификационной работы был проведен анализ предметной области с целью выделения и изучения основных понятий, таких, как: информационная система, видео, база данных и т.п.

Был проведен анализ существующих информационных систем, предназначенных для использования видеофрагментов в учебном процессе; обоснован выбор и описана специфика технологий, отобранных для разработки и реализации информационной системы.

Также была разработана информационная система «Смотри и Учись», предназначенная для использования видеофрагментов в учебном процессе, описаны ее структура, интерфейс и функциональные возможности.

Таким образом, задачи, которые ставились на первоначальном этапе выполнения работы, успешно решены, и цель достигнута.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Петров В. Н. Информационные системы./ В. Н. Петров. – СПб: Питер, 2002.
2. Избачков Ю. С. Информационные системы: Учебник для вузов / Ю. С. Избачков, В. Н. Петров. – 2-е изд. – СПб.: Питер, 2006. – 656 с.
3. Горюнов В. С. Информационные системы в образовании / В. С. Горюнов. —/ Молодой ученый. — 2010. — № 5 (16). — Т. 2. — С. 159-161. — URL: <https://moluch.ru/archive/16/1540/>.
4. Берлёв С. В. Особенности применения видеоматериалов и учебных презентаций в преподавании технических дисциплин / С. В. Берлёв // Проблемы и перспективы развития образования: материалы Международной научной конференции: в 2 томах. – Пермь: Меркурий, 2011. Т. 2. С. 184–186.
5. Кит Андердал, Цифровое видео для "чайников, 4-е издание, 2006.
6. Грабер М. Справочное руководство по SQL./ М. Грабер. – М:ЛОРИ, 1997.
7. Мещеряков Е. В. Публикация баз данных в интернете./ Е. В. Мещеряков, А. Д. Хомоненко. – СПб: БХВ-Петербург, 2001.
8. Ахо А. Структуры данных и алгоритмы: Пер. с англ.: Учеб. Пособие / А. Ахо, Д. Хопкрофт, Д. Ульман. – М: Вильямс, 2000. – 384 с.
9. Фролов А. В. Базы данных в интернете./ А. В. Фролов, Г. В. Фролов – М: Русская Редакция, 2000.
10. Баженова И. Ю. Основы проектирования приложений баз данных: Учеб. пособие / И. Ю. Баженова. – М: Бином. Лаборатория знаний Интуит, 2009. – 325 с.
11. Мальцев М. Г. Базы данных: Учебник / М. Г. Мальцев, А. Д. Хомоненко, В. М. Цыганков. – 6-е изд. – М: Корона Принт, 2011. – 736 с.

12. Новиков Б. А. Настройка приложений баз данных / Б. А. Новиков, Г. Р. Домбровская. – СПб.: БХВ-Петербург, 2006. – 240 с.
13. Томсон Л. Разработка веб-приложений на PHP и MySQL: Пер. с англ. / Л. Томсон. – М: Вильямс, 2012.
14. Дронов В. А. PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных веб-сайтов / В. А. Дронов. – СПб: БХВ-Петербург, 2005. – 448 с.
15. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика./ Т. Коннолли, К. Бегг. 3-е издание.: Пер. с англ. – М: Вильямс, 2008. – 1440 с.
16. Справочник по CSS. // URL <http://htmlbook.ru/css>
17. Справочник по HTML. // URL <http://htmlbook.ru/html>
18. Справочник по PHP. // URL <https://www.php.net/manual/ru/langref.php>
19. Справочник по SQL. // URL <http://unetway.com/tutorials/sql/>

## ПРИЛОЖЕНИЕ. ТЕКСТ ПРОГРАММЫ

### *request.php*

```

<?php
function addUser ($db, $l, $pass){
    $sql= "INSERT INTO users (login,password)
VALUES (:login, :password)
    ";
    $stmt=$db->prepare($sql);
    $stmt->bindValue(':login', $l, PDO::PARAM_STR);
    $stmt->bindValue(':password', $pass, PDO::PARAM_STR);
    $stmt->execute();
}
function addStudent ($db, $l, $pass, $f, $i, $o, $foto){
    $sql= "INSERT INTO student (login,f,i,o,password, foto)
VALUES (:login, :f, :i, :o, :password, :foto)
    ";
    $stmt=$db->prepare($sql);
    $stmt->bindValue(':login', $l, PDO::PARAM_STR);
    $stmt->bindValue(':f', $f, PDO::PARAM_STR);
    $stmt->bindValue(':i', $i, PDO::PARAM_STR);
    $stmt->bindValue(':o', $o, PDO::PARAM_STR);
    $stmt->bindValue(':password', $pass, PDO::PARAM_STR);
    $stmt->bindValue(':foto', $foto, PDO::PARAM_STR);
    $stmt->execute();
}
function addTeacher($db, $l, $pass, $f, $i, $o, $foto){
    $sql= "INSERT INTO teacher (login,f,i,o,password, foto)
VALUES (:login, :f, :i, :o, :password, :foto)
    ";
    $stmt=$db->prepare($sql);
    $stmt->bindValue(':login', $l, PDO::PARAM_STR);
    $stmt->bindValue(':f', $f, PDO::PARAM_STR);
    $stmt->bindValue(':i', $i, PDO::PARAM_STR);
    $stmt->bindValue(':o', $o, PDO::PARAM_STR);
    $stmt->bindValue(':password', $pass, PDO::PARAM_STR);
    $stmt->bindValue(':foto', $foto, PDO::PARAM_STR);
    $stmt->execute();
}
function addCourse ($db, $sub, $cat, $name, $des, $aut,
$subscr){
    $sql= "INSERT INTO courses
(subject,category,course,description,author, subscribes)
VALUES (:subject, :category, :course, :description, :author,
:subscribes)
    ";
    $stmt=$db->prepare($sql);
    $stmt->bindValue(':subject', $sub, PDO::PARAM_STR);
    $stmt->bindValue(':category', $cat, PDO::PARAM_STR);
    $stmt->bindValue(':course', $name, PDO::PARAM_STR);
    $stmt->bindValue(':description', $des, PDO::PARAM_STR);
    $stmt->bindValue(':author', $aut, PDO::PARAM_STR);
}

```

```

        $stmt->bindValue(':subscribes',$subscr,PDO::PARAM_STR);
        $stmt->execute();
    }
function addVideo ($db, $name,$des, $sub, $cat,$cour,$dir,
$aut){
    $sql= "INSERT INTO video (name,
description,subject,category,course,author,file)
VALUES (:name,:description,:subject,:category,:course,:author,:fi
le)
    ";
    $stmt=$db->prepare($sql);
    $stmt->bindValue(':name',$name,PDO::PARAM_STR);
    $stmt->bindValue(':description',$des,PDO::PARAM_STR);
    $stmt->bindValue(':subject',$sub,PDO::PARAM_STR);
    $stmt->bindValue(':category',$cat,PDO::PARAM_STR);
    $stmt->bindValue(':course',$cour,PDO::PARAM_STR);
    $stmt->bindValue(':author',$aut,PDO::PARAM_STR);
    $stmt->bindValue(':file',$dir,PDO::PARAM_STR);
    $stmt->execute();
}
function isStudent($db, $user, $password){
    $sql= "SELECT login, password FROM student
        WHERE login=:login AND password=:password
    ";
    $stmt=$db->prepare($sql);
    $stmt->bindValue(':login',$user,PDO::PARAM_STR);
    $stmt->bindValue(':password',$password,PDO::PARAM_STR);
    $stmt->execute();
    $row= $stmt->fetch(PDO::FETCH_ASSOC);
    if(!empty($row)){
        return true;
    } else {
        return false;
    }
}
function getStudent ($db,$log) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM student
    WHERE login='$log'
    ";
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    $row=$stmt->fetch(PDO::FETCH_ASSOC);
    $result=$row;
    return $result;
}
function getTeacher ($db,$log) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM teacher
    WHERE login='$log'
    ";
    $stmt= $db->prepare($sql);

```

```

$res=$stmt->execute();
$row=$stmt->fetch(PDO::FETCH_ASSOC);
$result=$row;
return $result;
}
function getStudentFotoByLogin ($db,$login) {
$db->query( "SET CHARSET utf8" );
$sql = "SELECT foto FROM student
WHERE login='$login'
";
$stmt= $db->prepare($sql);
$res=$stmt->execute();
$row=$stmt->fetch(PDO::FETCH_ASSOC);
$result=$row;
return $result;
}
function getTeacherFotoByLogin ($db,$login) {
$db->query( "SET CHARSET utf8" );
$sql = "SELECT foto FROM teacher
WHERE login='$login'
";
$stmt= $db->prepare($sql);
$res=$stmt->execute();
$row=$stmt->fetch(PDO::FETCH_ASSOC);
$result=$row;
return $result;
}
function isTeacher($db, $user, $password){
$sql= "SELECT login, password FROM teacher
WHERE login=:login AND password=:password
";
$stmt=$db->prepare($sql);
$stmt->bindValue(':login', $user, PDO::PARAM_STR);
$stmt->bindValue(':password', $password, PDO::PARAM_STR);
$stmt->execute();
$row= $stmt->fetch(PDO::FETCH_ASSOC);
if(!empty($row)) {
return true;
} else {
return false;
}
}
function getAllCourse ($db){
$db->query( "SET CHARSET utf8" );
$sql= "SELECT * FROM courses
";
$result=array();
$stmt= $db->prepare($sql);
$res=$stmt->execute();
while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
result[$row['id_course']]=$row;
}
}

```

```

        return $result;
    }
function getPopularCourse ($db){
    $db->query( "SET CHARSET utf8" );
    $sql= "SELECT * FROM courses ORDER BY subscribes DESC
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_course']]=$row;
    }
    return $result;
}

function Search ($db, $s){
    $db->query( "SET CHARSET utf8" );
    $sql= "SELECT * FROM courses
    WHERE course LIKE '%$s%'
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_course']]=$row;
    }
    return $result;
}

function getAllSubject ($db){
    $db->query( "SET CHARSET utf8" );
    $sql= "SELECT * FROM subjects
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_subject']]=$row;
    }
    return $result;
}

function getAllCategory ($db){
    $db->query( "SET CHARSET utf8" );
    $sql= "SELECT * FROM categories
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_category']]=$row;
    }
    return $result;
}

```

```

function getCourseById ($db,$id) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM courses
    WHERE id_course=:id_course
    ";
    $stmt = $db->prepare($sql);
    $stmt->bindValue ('id_course',$id, PDO::PARAM_INT);
    $stmt->execute();
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
    return $row;
}
function getCourseByVideo ($db,$vid) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM courses
    WHERE course=:vid_course
    ";
    $stmt = $db->prepare($sql);
    $stmt->bindValue ('vid_course',$vid, PDO::PARAM_INT);
    $stmt->execute();
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
    return $row;
}
function getVideoByCourseSort ($db) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM video
    WHERE for_lent='+' ORDER BY date_v DESC
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_video']]=$row;
    }
    return $result;
}
function getVideoById ($db,$id) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM video
    WHERE id_video=:id_video
    ";
    $stmt = $db->prepare($sql);
    $stmt->bindValue ('id_video',$id, PDO::PARAM_INT);
    $stmt->execute();
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
    return $row;
}
function getVideoByCourse ($db,$cour) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM video
    WHERE course='$cour'
    ";
    $result=array();

```

```

$stmt= $db->prepare($sql);
$res=$stmt->execute();
while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
    $result[$row['id_video']]=$row;
}
return $result;
}
function getCategoryBySubject ($db,$sub) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM categories
    WHERE subject='$sub'
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_category']]=$row;
    }
    return $result;
}
function getCourseByCategory ($db,$cat) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT * FROM courses
    WHERE category='$cat'
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_course']]=$row;
    }
    return $result;
}
function NewSubscribe ($db, $idc, $ls){
    $sql= "INSERT INTO subscribes (id_sub,login_subscriber)
    VALUES (:id_sub,:login_subscriber)
    ";
    $stmt=$db->prepare($sql);
    $stmt->bindValue(':id_sub',$idc,PDO::PARAM_STR);
    $stmt->bindValue(':login_subscriber',$ls,PDO::PARAM_STR);
    $stmt->execute();
}
function PlusSubscribe ($db, $idc){
    $sql= "UPDATE courses SET subscribes=subscribes+1
    WHERE id_course='$idc'
    ";
    $stmt=$db->prepare($sql);
    $stmt->execute();
}
function getPlusForLent ($db, $c){
    $sql= "UPDATE video SET for_lent='+
    WHERE course='$c'

```

```

";
$stmt=$db->prepare($sql);
$stmt->execute();
}
function getClearForLent ($db){
$sql= "UPDATE video SET for_lent=' '
";
$stmt=$db->prepare($sql);
$stmt->execute();
}
function addFotoStudent ($db, $login, $dir){
$sql= "UPDATE student SET foto='$dir'
WHERE login='$login'
";
$stmt=$db->prepare($sql);
$stmt->execute();
}
function addFotoTeacher ($db, $login, $dir){
$sql= "UPDATE teacher SET foto='$dir'
WHERE login='$login'
";
$stmt=$db->prepare($sql);
$stmt->execute();
}
function MinusSubscribe ($db, $idc){
$sql= "UPDATE courses SET subscribes=subscribes-1
WHERE id_course='$idc'
";
$stmt=$db->prepare($sql);
$stmt->execute();
}
function getIdCourseBySubscriber ($db,$user) {
$db->query( "SET CHARSET utf8" );
$sql = "SELECT id_sub FROM subscribes
WHERE login_subscriber='$user'
";
$result=array();
$stmt= $db->prepare($sql);
$res=$stmt->execute();
while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
$result[$row['id_sub']]=$row;
}
return $result;
}
function isUser($db, $user, $password){
$sql= "SELECT student.login, student.password,
teacher.login, teacher.password FROM student,teacher
WHERE student.login=:login AND student.password=:password
OR teacher.login=:login AND teacher.password=:password
$stmt=$db->prepare($sql);
$stmt->bindValue(':login', $user, PDO::PARAM_STR);
$stmt->bindValue(':password', $password, PDO::PARAM_STR);

```

```

$stmt->execute();
$row= $stmt->fetch(PDO::FETCH_ASSOC);
if(!empty($row)){
    return true;
} else {
    return false;
}
}
function SubscribeOrNot ($db,$id,$user) {
    $db->query( "SET CHARSET utf8" );
    $sql = "SELECT id_subscribe FROM subscribes
    WHERE login_subscriber='$user' AND id_sub='$id'
    ";
    $result=array();
    $stmt= $db->prepare($sql);
    $res=$stmt->execute();
    while ($row=$stmt->fetch(PDO::FETCH_ASSOC)) {
        $result[$row['id_sub']]=$row;
    }
    if (!empty($result)){
        return true;
    } else {
        return false;
    }
}
function DelSubscribe ($db, $id, $user){
    $sql= "DELETE FROM subscribes WHERE id_sub='$id' AND
    login_subscriber='$user'
    ";
    $stmt=$db->prepare($sql);
    $stmt->execute();
}

```

### **header.php**

```

<?session_start();
?>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.
min.js"></script>
<script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap
.min.js"></script>
<script>
function openForm() {
    document.getElementById("myForm").style.display = "block"; }
function closeForm() {
    document.getElementById("myForm").style.display = "none";
} </script>
<?php include 'connectbase.php';
include 'request.php';
$subjects= getAllSubject($db);

```

```

    $populars=getPopularCourse($db);
    /*$categories= getAllCategory($db);
    $courses= getAllCourse($db);*//////////
    ?>
<nav class="navbar navbar-fixed-top navbar-dark " >
    <ul class="nav navbar-nav">
        <li><a href="" class="active"><h1>Список
курсов</h1><span class="fa fa-angle-down"></span></a>
        <ul class="submenu">
            <?php foreach ($subjects as $subject) { ?>
            <li><a href=""><h2><? echo $subject ['subject'];
?></h2><span class="fa fa-angle-down"></span></a>
            <ul class="submenu">
                <?php
                $sub=$subject ['subject'];
                $categories=getCategoryBySubject($db,$sub);
                foreach ($categories as $category) {
                    ?>
                    <li><a href=""><h2><? echo $category ['category'];
?></h2><span class="fa fa-angle-down"></span></a>
                    <ul class="submenu">
                        <?php
                        $cat=$category ['category'];
                        $courses=getCourseByCategory($db,$cat);
                        foreach ($courses as $course) {
                            ?>
                            <li><a href="course.php?id=<?php echo
$course['id_course'];?>"><h2><? echo $course ['course'];
?></h2></a> </li>
                            <?> ?>
                        </ul>
                    </li>
                    <?> ?>
                </ul>
            </li>
            <?> }?>
        </ul>
    </li>
        <li><a href="" class="active"><h1>Популярные
курсы</h1><span class="fa fa-angle-down"></span></a>
        <ul class="submenu">
            <?php
            foreach ($populars as $popular) {
                ?>
                <li><a href="course.php?id=<?php echo
$popular['id_course'];?>"><h2><? echo $popular ['course']; ?>
<p>Кол-во подписчиков: <? echo $popular
['subscribes'];?></h2></a> </li>
                <?> ?>
            </ul>
        </li>
    </ul>

```

```

    <div class="home">
    <a href="main.php"></a>
</div>
<div class="d1">
    <form id="lol" action="search.php" method="post">
    <input name="search" type="text" placeholder="Искать
здесь...">
    <button type="submit"></button>
    </form>
</div>
    <?php if(!empty($_SESSION)){
        if ($_SESSION['type']=='stud'){

$user=getStudentFotoByLogin($db,$_SESSION['user']);
        else {

$user=getTeacherFotoByLogin($db,$_SESSION['user']);
        }
    ?>
        <div class="foto">
<span class="fa fa-angle-down">
    <div class="descr">
<button class="btn btn-primary btn-sm"
onclick="window.location.href='photoupload.php'"><h2>Загрузить
фото</h2></button>
<button class="btn btn-primary btn-sm"
onclick="window.location.href='usercourses.php'"><h2>Мои
курсы</h2></button>
<!-- <button class="btn btn-primary btn-sm"
onclick="window.location.href='logout.php#zatemnenie'"><h2>ВЫЙТИ
</h2></button> -->
</div>
</div>
                <div class="log">
                <h2>Пользователь: <?php echo
$_SESSION['user']; ?></h2></div>
                <button class="btn btn-primary btn-sm"
id="open-button"
onclick="window.location.href='logout.php#zatemnenie'"><h2>ВЫЙТИ
</h2></button>
                <?php } else { ?>

    <button class="btn btn-primary btn-sm" id="open-button"
onclick="openForm()"><h2>ВОЙТИ</h2></button>
    <div class="form-popup" id="myForm">
        <div class="container-fluid">
    <form action="login.php#zatemnenie" method="POST" role="form">
    <h1>Войти</h1>
    <div class="form-group">
        <label for=""><h2>Логин</h2></label>
        <input type="text" class="form-control" id="login"
name="login" >

```

```

</div>

<div class="form-group">
  <label for=""><h2>Пароль</h2></label>
  <input type="password" class="form-control" id="password"
name="password" >
</div>
  <button type="submit" class="btn btn-
primary"><h3>Войти</h3></button> <a class="btn btn-success"
href="registration.php"><h3>Регистрация</h3></a>
  <button type="button" class="btn cancel"
onclick="closeForm()"><h3>ЗАКРЫТЬ</h3></button>
</form>

</div>
      <?php } ?>
    </div>
</nav>

```

### **main.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Смотри и Учись!</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
  </div>
<div id="cen">
<? if(!empty($_SESSION)){
  if($_SESSION['type']=='stud'){
    $user=getStudent($db,$_SESSION['user']);}
  else {$user=getTeacher($db,$_SESSION['user']);}
  ?>
  <h1>Рады вас снова видеть,<? echo $user['i'];?> <?echo
$user['o']; ?></h1>
  <? if ($_SESSION['type']=='teach') { ?>
<button class="btn btn-primary btn-sm"
onclick="window.location.href='newcourse.php'"><h2>Создать
курс</h2></button>
  <? } ?>
  <?

```

```

        $id_c=getIdCourseBySubscriber($db,$_SESSION['user']);
        foreach ($id_c as $id) {

            $courses=getCourseById($db,$id['id_sub']);
            getPlusForLent($db,$courses['course']);
        }

        $videos=getVideoByCourseSort($db);
        getClearForLent($db);
        if (!empty($videos)){
            echo "<h1>Лента ваших подписок</h1>";
            foreach ($videos as $video) {
                $course=getCourseByVideo($db,$video['course']);
            }
        }
        <p>
        <h2>В курс <a href="course.php?id=<?php echo
        $course['id_course'];?>"><? echo $course['course']?></a>
        добавлено видео:</h2>
            <video width="400" height="300"
            controls="controls"
            onclick="window.location.href='video.php?id=<?php echo
            $video['id_video'];?>' ">
            <source src="<?=$video['file']?>" >
        </video>
        <h2><? echo $video['name']?></h2>
        <h2>Автор:<? echo $video['author']?></h2>
        <h2>Дата: <? echo $video['date_v']?></h2></td>
            <?
            }
        }
    }
    else { ?>
        <h1>Здравствуй, юный падаван</h1>
        <h1>На нашем сайте ты можешь найти интересующий тебя
        курс</h1>
        <h1>подписаться на него</h1>
        <h1>и стать мастером джедаем в выбранной тобой теме</h1>
        <h1>по видео</h1>
        <h1>но для начала твоего пути зарегистрируйся!</h1>
        <h1>Да пребудет с тобой сила ума!</h1>
        <? }
        ?>
        </center>
    </div>
</body>
</html>

```

### **newcourse.php**

```

<?php session_start(); ?>
<!DOCTYPE html>
<html>
<head>

```

```

        <meta charset="utf-8">
        <title>Новый курс</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">

</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
    </div>
<div id="cen">
    <center>
    <h1>Создание курса</h1>
    <form action="addcourse.php#zatemnenie" method="post">
    <table>
        <tr>
<p>
    <td><label>Выберете предмет <br></label></td>
    <td><select name="sub" class="form-control" id="sub">
        <?php
            $subjects=getAllSubject($db);
            foreach ($subjects as $subject) {
                echo "<option value=".".$subject
['id_subject'].>".$subject['subject'].></option>";
            }
        <?>
        </select>
    </td>
    </p>
</tr>
<tr>
<p>
    <td><label>Выберете категорию:<br></label></td>
    <td><select name="cat" class="form-control" id="cat">
        <?php
            $categories=getAllCategory($db);
            foreach ($categories as $category) {
                echo "<option value=".".$category
['id_category'].>".$category['category'].></option>";
            }
        <?>
        </select></td>
    </p>
</tr>
<tr>
<p>
    <td><label>Название курса:<br></label></td>

```

```

        <td><input name="nam" type="text" size="40"
maxlength="15"></td>
    </p>
</tr>
<tr>
<p>
    <td><label>Описание курса:<br></label></td>
    <td><input name="des" type="text" size="40"
maxlength="15"></td>
    </p>
</tr>
<tr>
<p>
        <tr>
            <td></td>
            <td><p>
                <input type="submit" name="submit" value="Добавить">
            </p>
        </td>
    </tr>
</table>
        </form>
    </center>
</div>
</body>
</html>

```

### **usercourses.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Мои курсы</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<div id="cen">
    <h1>Мои курсы</h1>
    <?
        $id_c=getIdCourseBySubscriber($db,$_SESSION['user']);
        foreach ($id_c as $id) {

```

```

        $courses=getCourseById($db,$id['id_sub']);
?>
        <a href="course.php?id=?php echo
$courses['id_course'];?>"><h2><? echo $courses ['course'];
?></h2></a>
        <?
                }
?>
</div>
</body>
</html>

```

### **nice.css**

```

body, html {
    width: 1920px;
    height: 967px;
margin:0px;
padding:0px;
text-align:center;
}

#zatemnienie {
    background: rgba(102, 102, 102, 0.5);
    width: 100%;
    height: 100%;
    position: absolute;
    top: 0;
    left: 0;
    display: none;
}
#okno {
    width: 300px;
    height: 82px;
    text-align: center;
    padding: 10px;
    border: 3px solid #5f99c8;
    border-radius: 10px;
    color: #5f99c8;
    position: absolute;
    top: 0;
    right: 0;
    bottom: 0;
    left: 0;
    margin: auto;
    background: #fff;
}
#zatemnienie:target {display: block;}
.close {
    display: inline-block;
    border: 1px solid #5f99c8;
    color: #5f99c8;
}

```

```
padding: 0 12px;
margin: 10px;
text-decoration: none;
background: #ffffff;
font-size: 28px;
cursor:pointer;
}
.close:hover {background: #ffffff;}

#list{
margin:0 auto;
text-align:left;
width:800px;
}

#header{
height: 150px;
background-color:#F0FFFF;
background-image:url(l.png);
background-position: center;
background-repeat: no-repeat;
padding: 0px 10px;
}

#header h1 {
margin: 10px;
padding: 10px 0;
font-family: Monly Lite Light;
font-size: 72px;
}

#header h2 {
margin: 5px;
padding: 5px 0;
font-size: 33px;
font-family: Monly Lite Light;
}

.dropdown-menu {
margin: 5px;
padding: 5px 0;
font-size: 33px;
font-family: Monly Lite Light;
}

#header h3 {
margin: 10px;
padding: 10px 0;
font-family: Monly Lite Light;
```

```
}

#open-button {
  position: fixed;
  top: 36px;
  right: 26px;
  width: 190px;
  .. }

.form-popup {
  background-color: #F0FFFF;
  display: none;
  position: fixed;
  top: 0;
  right: 14px;
  height: 420px;
}

.form-container {
  max-width: 320px;
  ..
}

.form-container input[type=text], .form-container
input[type=password] {
  width: 100%;
  ..
}

.form-container input[type=text]:focus, .form-container
input[type=password]:focus {
  background-color: #ddd;
  ..
}

.form-container .btn {
  background-color: #1976D2;
  ..
  opacity: 0.7;
}

.form-container .cancel {
  background-color: #448AFF;
}

.form-container .btn:hover, .open-button:hover {
  opacity: 1;
}

#lol{
```

```
position: fixed;
top: 50px;
  right: 350px;
width: 330px;
margin: 0 auto;
}

.log h2{
  position: fixed;
  top: -10px;
  right: 34px;
}

.home a {
position: fixed;
  top: 5px;
  right: 788px;
  width: 497px;
  height: 128px;
  border: none;
  background: #5f99c800;
  cursor: pointer;
}

.foto img {
  position: fixed;
  top: 5px;
  right: 222px;
  height: 139px;
  width: 119px;
}

.descr{
  display:none;
  margin-left:-350px;
  padding:10px;
  margin-top:17px;
  background:#5f99c800;
  height:150px;
/*   -moz-box-shadow:0 5px 5px rgba(0,0,0,0.3);
   -webkit-box-shadow:0 5px 5px rgba(0,0,0,0.3);
   box-shadow:0 5px 5px rgba(0,0,0,0.3);*/
}

.foto:hover .descr{
  display:block;
  position:fixed;
  top:127px;
  right: 203px;
  width:163px;
```

```

}

.dl input {
  width: 330px;
  height: 42px;
  padding-left: 10px;
  border: 2px solid #5f99c8;
  border-radius: 5px;
  outline: none;
  font-family: Monly Lite Light;
  font-size: 25px;
  background: #ffffff;
  color: #5f99c8;
}

.dl button {
  position: absolute;
  top: 0;
  right: 0px;
  width: 50px;
  height: 42px;
  border: none;
  background: #5f99c8;
  border-radius: 0 5px 5px 0;
  cursor: pointer;
}

.dl button:before {
  content: "Найти";
  font-family: Monly Lite Light;
  font-size: 25px;
  color: #ffffff;
}

#left{
background-color:#F0FFFF;
width:250px;
height: 815px;
float:left;
}

#right{
padding-left: 5px;
width:250px;
height: 150px;
background-color:#F0FFFF;
float:right;
font-family: Monly Lite Light;
}

#cen {
width: 1900px;
height: 600px;
font-family: Monly Lite Light;
}

```

```
    color: #337ab7;
background-color: #FFFFFF;
margin:0px;
margin-left:0px;
margin-right:0px;

}

#cen td {
margin: 10px;
padding: 10px 0;
font-family: Monly Lite Light;
font-size: 33px;
}

#cen h1 {
margin: 10px;
padding: 10px 0;
font-family: Monly Lite Light;
font-size: 65px;
}

#cen h2 {
margin: 5px;
padding: 5px 0;
font-size: 33px;
font-family: Monly Lite Light;

}

#cen select{
    height: 50px;
    font-weight: 900;
    font-size:25px;
}
#cen input{
    font-weight: 900;
}
#clear{
height:0;
font-size:1px;
line-height:0px;
clear:both;
}

#footer{
    height: 219px;
background-color:#F0FFFF;
}

nav {background-color:#F0FFFF;
background-image:url(1.png);
```

```

background-position: center;
background-repeat: no-repeat;}
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav ul:after {
  content: "";
  display: table;
  clear: both;
}
nav a {
  text-decoration: none;
  display: block;
  transition: .3s linear;
}
.topmenu > li {
  float: left;
  position: relative;
  border-left: 1px solid black;
}
.topmenu > li:first-child {border-left: 0;}
.topmenu > li > a {
  padding: 20px 30px;
  font-size: 12px;
  text-transform: uppercase;
  color: #FEFDFD;
  letter-spacing: 2px;
}
.topmenu > li > a.active,
.submenu a:hover {color: #5f99c8;}
.topmenu .fa,
.submenu .fa {
  margin-left: 5px;
  color: inherit;
}
.submenu {
  position: absolute;
  z-index: 5;
  min-width: 200px;
  background: white;
  border-top: 1px solid #CBCBCC;
  border-left: 1px solid #CBCBCC;
  border-right: 1px solid #CBCBCC;
  visibility: hidden;
  opacity: 0;
  transform-origin: 0% 0%;
  transform: rotateX(-90deg);
  transition: .3s linear;
}
.submenu li {position: relative;}

```

```
.submenu li a {
  color: #282828;
  padding: 10px 20px;
  font-size: 13px;
  border-bottom: 1px solid #CBCBCC;
}
.submenu .submenu {
  position: absolute;
  left: 100%;
  top: -1px;
  transition: .3s linear;
}
nav li:hover > .submenu {
  transform: rotateX(0deg);
  visibility: visible;
  opacity: 1;
}
```

### **course.php**

```
<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Курс</title>
  <link rel="stylesheet" type="text/css" href="css/nice.css">
  <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<div id="cen">
  <center>
<?php
  $id= $_GET['id'];
  if ($id){
    $course= getCourseById($db, $id);
  } else {
    echo "<h1>Ошибка</h1>";
    exit ();
  }
?>
<h1><? echo $course['course']; ?></h1>
<h1>Описание курса</h1>
<h1> Подписчиков:<? echo $course['subscribes']; ?></h1>
```

```

        <? if
    (($_SESSION['type']=='teach') && ($_SESSION['user']==$course['author'])) { ?>
    <button class="btn btn-primary btn-sm"
    onclick="window.location.href='fileupload.php?id=?php echo
    $course['id_course'];?>' "><h2>добавить файл</h2></button>
    <button class="btn btn-danger btn-sm"
    onclick="window.location.href='delcourse.php?id=?php echo
    $course['id_course'];?>' "><h2>Удалить курс</h2></button>
    <button class="btn btn-success btn-sm"
    onclick="window.location.href='editcourse.php?id=?php echo
    $course['id_course'];?>' "><h2>Изменить</h2></button>
    <? }

    else {
        if (SubscribeOrNot($db,$id,$_SESSION['user'])) {
            ?>
            <button class="btn btn-danger btn-sm"
            onclick="window.location.href='desub.php?id=?php echo
            $course['id_course'];?>#zatemnenie'"><h2>Отписаться</h2></button
            >

            <? } else { ?>
                <button class="btn btn-success btn-sm"
                onclick="window.location.href='subscribe.php?id=?php echo
                $course['id_course'];?>#zatemnenie'"><h2>Подписаться</h2></button
                n>

                <?php }
            }
            $videos= getVideoByCourse($db, $course['course']);
            ?> <table border="3" bordercolor="#5f99c8"><tr> <?
            foreach ($videos as $video) {
                ?>
                <td>
                    <video width="400" height="300" controls="controls"
                    onclick="window.location.href='video.php?id=?php echo
                    $video['id_video'];?>' ">
                    <source src="<?=$video['file']?>" >
                    </video>
                    <h2><? echo $video['name']?></h2>
                    <h2>Автор:<? echo $video['author']?></h2></td>
                    <? } ?>
                </tr></table>
            </center>
        </div>
    </body>
</html>

```

### ***videp.php***

```

<?session_start();
?>
<!DOCTYPE html>
<html>

```

```

<head>
    <meta charset="utf-8">
    <title>Видео</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<div id="cen">
    <center>
<?php
    $id= $_GET['id'];
    if ($id){
        $video= getVideoById($db, $id);
    } else {
        echo "<h1>Ошибка</h1>";
        exit ();
    }
    ?>
    <h1><? echo $video['name']; ?></h1>
        <video width="1020" height="610"
controls="controls">
    <source src="<?=$video['file']?>" >
</video>
<h2><? echo $video['description']?></h2>
<h2>Автор:<? echo $video['author']?></h2></td>
    <? if
(($_SESSION['type']=='teach')&&($_SESSION['user']==$video['autho
r'])) { ?>
<button class="btn btn-danger btn-sm"
onclick="window.location.href='delvideo.php?id=<?php echo
$course['id_course'];?>'"><h2>Удалить видео</h2></button>
<button class="btn btn-success btn-sm"
onclick="window.location.href='editvideo.php?id=<?php echo
$course['id_course'];?>'"><h2>Изменить</h2></button>
    <? } ?>
    </center>
</div>
</body>
</html>

```

### ***addcourse.php***

```

<?session_start();
?>
<!DOCTYPE html>

```

```

<html>
<head>
  <meta charset="utf-8">
  <title>Добавление курса</title>
  <link rel="stylesheet" type="text/css" href="css/nice.css">
  <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">

</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
<?
  if(isset($_POST['sub']) && $_POST['cat']&& $_POST['nam']&&
$_POST['des'] != ''){
    $s=$_POST['sub'];
    $c=$_POST['cat'];
    $n=$_POST['nam'];
    $d=$_POST['des'];
    $a=$_SESSION['user'];
    $subscr=0;
    addCourse($db, $s, $c,$n,$d,$a, $subscr);
  ?>
  <div id="cen">
<div id="zatemnenie">
  <div id="okno">
    <h2>Курс добавлен!</h2> <br>
    <a href="main.php" class="close">Закрыть окно</a>
  </div>
</div>
<?php

  }
  else { ?> <div id="cen">
<div id="zatemnenie">
  <div id="okno">
    <h2>Заполните все поля!</h2> <br>
    <a href="newcourse.php" class="close">Закрыть окно</a>
  </div>
</div> <?>
?>
</div>
</div>
</body>
</html>

```

**subscribe.php**

```
<?session_start();
```

```

?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Подписка</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
<?
    $id= $_GET['id'];
    if ($id){
        } else {
            echo "<h1>Ошибка</h1>";
            exit ();
        }
    }
NewSubscribe($db, $id, $_SESSION['user']);
PlusSubscribe($db, $id);
?>
</div>
<div id="cen">
    <div id="zatemnenie">
        <div id="okno">
            <h2>Вы успешно подписались!</h2> <br>
            <a href="course.php?id=<?php echo $id?>"
class="close">Закреть окно</a>
        </div>
    </div>
</div>
</body>
</html>

```

### ***desub.php***

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Отписка</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">

```

```

</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
<?
    $id= $_GET['id'];
        if ($id){

                } else {
                    echo "<h1>Ошибка</h1>";
                    exit ();
                }
    DelSubscribe($db, $id, $_SESSION['user']);
    MinusSubscribe($db, $id);
?>
</div>
<div id="cen">
    <div id="zatemnenie">
        <div id="okno">
            <h2>Вы успешно отписались!</h2> <br>
            <a href="course.php?id=<?php echo $id?>"
class="close">Закрыть окно</a>
        </div>
    </div>
</div>
</body>
</html>

```

### **fileupload.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Загрузка видео</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<div id="cen">
<center>

```

```

<? $id= $_GET['id']; ?>
    <h1>Загрузить видео</h1>
<form method="post" action="video/basic.php?id=?php echo
$id;?>" enctype="multipart/form-data">
    <table>
        <tr><td></td><td>ИЛИ</td></tr>
        <tr>
            <td><label for="inputfile"><h2>Введите
ссылку</h2></label></td>
            <td><input name="n" type="text" size="40" maxlength="30"></td>
        </tr>
        <tr>
            <td><label for="inputfile"><h2>Введите
название</h2></label></td>
            <td><input name="name" type="text" size="40"
maxlength="30"></td>
        </tr>
        <tr>
            <td><label for="inputfile"><h2>Описание видео:</h2></label></td>
            <td><input name="des" type="text" size="40"
maxlength="100"></td>
        </tr>
        <tr>
            <td><label for="inputfile"><h2>Выберите файл</h2></label>
            <input type="file" id="inputfile" name="inputfile"></br>
        </tr>
        <tr><td></td><td>
            <input type="submit" value="Загрузить"></td></tr>
    </table>
</center>
</div>
</body>
</html>

```

### **search.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Поиск купца</title>
    <link rel="stylesheet" type="text/css" href="css/nice.css">
    <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >

```

```

<?php include ('header.php'); ?>
        </div>
<div id="cen">
    <h1>Вот, что удалось найти:</h1>
    <?
    $s=$_POST['search'];
    $courses=Search($db,$s);
    if (!empty($courses)){
    foreach ($courses as $course) {
        ?>
        <a href="course.php?id=<?php echo
    $course['id_course'];?>"><h2><? echo $course ['course'];
    ?></h2></a>
    <? } }
    else echo "<h2>Ничего не найдено!</h2>"; ?>

</center>
</div>
</body>
</html>

```

### **photoload.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Загрузка фото</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
        </div>
<div id="cen">
<center>
    <h1>Загрузить фотографию</h1>
<form method="post" action="photo\photoload.php"
enctype="multipart/form-data">
<input type="file" id="inputfile" name="inputfile"></br>
<input type="submit" value="Загрузить">
</center>
</div>
</body>
</html>

```

**photoload.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
<?include
'C:\Users\1\Desktop\OSPanel\domains\SeeAndLearn\www\connectbase.
php';
    include
'C:\Users\1\Desktop\OSPanel\domains\SeeAndLearn\www\request.php'
;
if(isset($_FILES) && $_FILES['inputfile']['error'] == 0){
var_dump($_FILES['inputfile']['name']); // Проверяем, загрузил ли
пользователь файл
$destination_dir = dirname(__FILE__)
.'\..\'.$_FILES['inputfile']['name']; // Директория для размещения
файла
move_uploaded_file($_FILES['inputfile']['tmp_name'],
$destination_dir ); // Перемещаем файл в желаемую директорию
echo 'File Uploaded'; // Оповещаем пользователя об успешной
загрузке файла
$dir='photo'.'/'.$_FILES['inputfile']['name'];
if ($_SESSION['type']=='stud'){
addFotoStudent($db,$_SESSION['user'],$dir);}
else {
addFotoTeacher($db,$_SESSION['user'],$dir);}
}
else{
echo 'No File Uploaded'; // Оповещаем пользователя о том, что
файл не был загружен
}
?>
</body>
</html>

```

**basic.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
<?include
'C:\Users\1\Desktop\OSPanel\domains\SeeAndLearn\www\connectbase.
php';

```

```

include
'C:\Users\1\Desktop\OSPanel\domains\SeeAndLearn\www\request.php'
;
$id= $_GET['id'];
if ($id){
    $course= getCourseById($db, $id);
} else {
    echo "<h1>Ошибка</h1>";
    exit ();
}
if((isset($_FILES) && $_FILES['inputfile']['error'] == 0) &&
($_POST['name'] && $_POST['des'] != '')){
var_dump($_FILES['inputfile']['name']); // Проверяем, загрузил ли
пользователь файл
$destination_dir = dirname(__FILE__)
.'\\'.$_FILES['inputfile']['name']; // Директория для размещения
файла
move_uploaded_file($_FILES['inputfile']['tmp_name'],
$destination_dir ); // Перемещаем файл в желаемую директорию
echo 'File Uploaded'; // Оповещаем пользователя об успешной
загрузке файла
$author=$_SESSION['user'];
$name=$_POST['name'];
$des=$_POST['des'];
$sub=$course['subject'];
$cat=$course['category'];
$cour=$course['course'];
$dir='video'.'/'.$_FILES['inputfile']['name'];
addVideo($db,$name, $des, $sub, $cat,$cour,$dir,$author);
var_dump($name);
var_dump($des);
}
else{
echo 'No File Uploaded'; // Оповещаем пользователя о том, что
файл не был загружен
}
?>
</body>
</html>

```

### **login.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
<link rel="stylesheet" type="text/css" href="css/nice.css">

```

```

<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">

</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<div id="cen">
<?
if(!empty($_POST)){
    if ($_POST['login']!='' && $_POST['password']!=''){
        $login=trim(strip_tags($_POST['login']));
        $password=trim(strip_tags($_POST['password']));
        if (isStudent($db,$login,$password)){
            $_SESSION ['user']=$login;
            $_SESSION ['type']='stud';
            /* echo "<h1>Вы успешно вошли</h1> ";*/ ?>
            <div id="zatemnenie">
<div id="okno">
                <h2>Вы успешно вошли!</h2> <br>
                <a href="main.php" class="close">Закрыть окно</a>
            </div>
        </div>
            <?
            /*echo
'<script>location.replace("main.php");</script>';*/
                }
                else {
                    if (isTeacher($db,$login,$password)){
                        $_SESSION ['user']=$login;
                        $_SESSION ['type']='teach';
                        /* echo "<h1>Вы успешно вошли</h1> ";*/ ?>
                        <div id="zatemnenie">
<div id="okno">
                            <h2>Вы успешно вошли!</h2> <br>
                            <a href="main.php" class="close">Закрыть окно</a>
                        </div>
                    </div>
                </div>
                    <?
                    }
                    else{
                        ?>
                        <div id="zatemnenie">
<div id="okno">
                            <h2>Пользователь не найден!</h2> <br>
                            <a href="main.php" class="close">Закрыть окно</a>
                        </div>
                    </div>
                </div>
            </div>
        }
    }
}

```

```

        <?
    }

    }
    else {
        ?>
        <div id="zatemnenie">
        <div id="okno">
            <h2>Заполните все поля!</h2> <br>
            <a href="main.php" class="close">Заккрыть окно</a>
        </div>
        </div>
        <?
    }
}

?>
</div>
</body>
</html>

```

### **logout.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
    <link rel="stylesheet" type="text/css" href="css/nice.css">
    <link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
<?
session_destroy();
?>
</div>
<div id="cen">
    <div id="zatemnenie">
        <div id="okno">
            <h2>Вы успешно вышли из аккаунта!</h2> <br>
            <a href="main.php" class="close">Заккрыть окно</a>
        </div>
    </div>
</div>

```

```
</div>
</body>
</html>
```

### **registration.php**

```
<?php session_start(); ?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Регистрация</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
    </head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<div id="cen">
    <center>
        <h1>Регистрация</h1>

        <form action="new_user.php#zatemnenie" method="post">
            <table>
                <tr>
<p>
            <td><label>Ваш логин: <br></label></td>
            <td><input name="login" type="text" size="40"
maxlength="15"></td>
        </p>
    </tr>
<tr>
<p>
            <td><label>Ваш пароль:<br></label></td>
            <td><input name="password" type="password" size="40"
maxlength="15"></td>
        </p>
    </tr>
<tr>
<p>
            <td><label>Ваша фамилия:<br></label></td>
            <td><input name="f" type="text" size="40"
maxlength="15"></td>
        </p>
    </tr>
<tr>
<p>
```

```

        <td><label>Ваше имя:<br></label></td>
        <td><input name="i" type="text" size="40"
maxlength="15"></td>
    </p>
</tr>
<tr>
<p>
        <td><label>Ваше отчество:<br></label></td>
        <td><input name="o" type="text" size="40"
maxlength="15"></td>
    </p>
</tr>
<tr>
        <td><label>Выберите тип:</label></td>
        <td> <input type="radio" name="typereg" value="Stud">
Студент <br>
<input type="radio" name="typereg" value="Teach">
Преподаватель</td>
</tr>
<tr>
        <td><label>Введите код с картинки:
</label></td>
        <td> <img src='captcha.php'
id='captcha-image'> </td> </tr>
<tr>
        <td></td>
        <td><input type="text"
name="captcha" required /></td>
</tr>
<tr>
        <td></td>
        <td><p>
<input type="submit" name="submit"
value="Зарегистрироваться">
</p>
</td>
</tr>
</table>
</form>
</center>
</div>
</body>
</html>

```

### ***captcha.php***

```

<?session_start();
    $randomnr = rand(1000, 9999);
    $_SESSION["rand_captcha"] =$randomnr;
    $im = imagecreatetruecolor(100, 38);
    $white = imagecolorallocate($im, 255, 255, 255);
    $grey = imagecolorallocate($im, 150, 150, 150);
    $black = imagecolorallocate($im, 0, 0, 0);

```

```

    imagefilledrectangle($im, 0, 0, 200, 35, $black);
    //path to font - this is just an example you can use any
font you like:
        $font = dirname(__FILE__).'/font/Karate.ttf';
    imagettftext($im, 20, 4, 22, 30, $grey, $font, $randomnr);
    imagettftext($im, 20, 4, 15, 32, $white, $font, $randomnr);
        //prevent caching on client side:
    header("Expires: Wed, 1 Jan 1997 00:00:00 GMT");
    header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "
GMT");
    header("Cache-Control: no-store, no-cache, must-
revalidate");
    header("Cache-Control: post-check=0, pre-check=0", false);
    header("Pragma: no-cache");
    header ("Content-type: image/gif");
    imagegif($im);
    imagedestroy($im);
?>

```

### **connectbase.php**

```

<?php
$user = "root";
$password = "";
try {
    $db=new PDO
("mysql:host=localhost;dbname=seeandlearn",$user,$password);
} catch (Exception $e) {
    echo $e->getMessage();
}

```

### **new\_user.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
<?
    if(isset($_POST['login']) && $_POST['password']&&
$_POST['f']&& $_POST['i']&& $_POST['o']&&($_POST['typereg']) !=
'') {

```

```

    $captcha = trim($_POST["captcha"]);
    if($_SESSION["rand_captcha"] != $captcha){
        session_destroy();
        ?><div id="cen">
<div id="zatemnenie">
    <div id="okno">
        <h2>Неправильно введена картинка!</h2> <br>
        <a href="registration.php" class="close">Закреть
окно</a>
    </div>
</div> <?php
                                } else{
                                    if ($_POST['typereg']=='Stud'){
                                        $l=$_POST['login'];
                                        $pass=$_POST['password'];
                                        $f=$_POST['f'];
                                        $i=$_POST['i'];
                                        $o=$_POST['o'];
                                        $foto='photo/nofoto.jpg';
                                        addStudent($db, $l, $pass,$f,$i,$o, $foto);
                                        session_destroy();
                                    }
                                else {
                                    $l=$_POST['login'];
                                    $pass=$_POST['password'];
                                    $f=$_POST['f'];
                                    $i=$_POST['i'];
                                    $o=$_POST['o'];
                                    $foto='photo/nofoto.jpg';
                                    addTeacher($db, $l, $pass,$f,$i,$o, $foto);
                                    session_destroy();
                                }
                                ?>

                                <div id="cen">
<div id="zatemnenie">
    <div id="okno">
        <h2>Вы успешно зарегистрировались!</h2> <br>
        <a href="main.php" class="close">Закреть окно</a>
    </div>
</div>
<?php
    }
    }
    else { session_destroy(); ?> <div id="cen">
<div id="zatemnenie">
    <div id="okno">
        <h2>Заполните все поля!</h2> <br>
        <a href="registration.php" class="close">Закреть
окно</a>
    </div>
</div> <?>

```

```
?>
</div>
</div>
</body>
</html>
```

### ***delcourse.php***

```
<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Добавление курса</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<?
  $id= $_GET['id'];
  if ($id){
    DelCourse($db, $id);
  } else {
    echo "<h1>Ошибка</h1>";
    exit ();
  }
?>
  <div id="cen">
<div id="zatemnenie">
  <div id="okno">
    <h2>Курс удален!</h2> <br>
    <a href="main.php" class="close">Заккрыть окно</a>
  </div>
</div>
</div>
<div id="clear">
  </div>
</body>
</html>
```

### ***delvideo.php***

```
<?session_start();
?>
<!DOCTYPE html>
<html>
```

```

<head>
  <meta charset="utf-8">
  <title>Добавление курса</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
</div>
<?
  $id= $_GET['id'];
  if ($id){
    $video= getVideoById($db, $id);
    unlink($video['file']);
    DelVideo($db, $id);
  } else {
    echo "<h1>Ошибка</h1>";
    exit ();
  }
  ?>
  <div id="cen">
<div id="zatemnenie">
  <div id="okno">
    <h2>Видео удалено!</h2> <br>
    <a href="main.php" class="close">Закрыть окно</a>
  </div>
</div>
  </div>
<div id="clear">
  </div>
</body>
</html>

```

### **savecourse.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Сохранение курса</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>
<body>

```

```

<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
<?
    if(isset($_POST['sub'])    &&    $_POST['cat']&&    $_POST['nam']&&
$_POST['des'] != ''){
        $id= $_GET['id'];
        $s=$_POST['sub'];
        $c=$_POST['cat'];
        $n=$_POST['nam'];
        $d=$_POST['des'];
        UpCourse($db,$id, $s, $c,$n,$d);
        ?>
        <div id="cen">
<div id="zatemnenie">
<div id="okno">
<h2>Курс изменен!</h2> <br>
<a href="main.php" class="close">Закрыть окно</a>
</div>
</div>
<?php
        }
        else { ?> <div id="cen">
<div id="zatemnenie">
<div id="okno">
<h2>Заполните все поля!</h2> <br>
<a href="editcourse.php?id=?php echo $id?>"
class="close">Закрыть окно</a>
</div>
</div> <?>
?>
</div>
</div>
<div id="clear">
</div>
</body>
</html>

```

### **savevideo.php**

```

<?session_start();
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Сохранение видео</title>
<link rel="stylesheet" type="text/css" href="css/nice.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.mi
n.css" rel="stylesheet">
</head>

```

```

<body>
<div id="list">
</div>
<div id="header" >
<?php include ('header.php'); ?>
<?
    if(isset($_POST['nam']) && $_POST['des'] != '') {
        $id= $_GET['id'];
        $n=$_POST['nam'];
        $d=$_POST['des'];
        UpVideo($db,$id,$n,$d);
        ?>
        <div id="cen">
<div id="zatemnenie">
<div id="okno">
<h2>Информация о видео изменена!</h2> <br>
<a href="main.php" class="close">Закреть окно</a>
</div>
</div>
<?php
        }
        else { ?> <div id="cen">
<div id="zatemnenie">
<div id="okno">
<h2>Заполните все поля!</h2> <br>
<a href="editvideo.php?id=<?php echo $id?>"
class="close">Закреть окно</a>
</div>
</div> <?>
?>
</div>
</div>
<div id="clear">
</div>
</body>
</html>

```