

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**   
**«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»**

Факультет Физико-технический

Кафедра Компьютерных технологий (КТ)

|  |  |  |
| --- | --- | --- |
|  | | |
| Зав. кафедрой | | КТ |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | Т. В. Ермоленко |
| (подпись) | |  |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе бакалавра 3 курса

на тему:

|  |
| --- |
| РАЗРАБОТКА ДИНАМИЧЕСКОГО САЙТА ДЛЯ УЧЕТА МЕТОДИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ КАФЕДРЫ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ |
|  |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Автор работы |  |  | Д.И. Фурсов |

подпись

|  |  |  |
| --- | --- | --- |
| Направление | 09.03.01 | Информатика и вычислительная техника |

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преподаватель В.Н. Котенко

подпись

Консультанты по разделам:

Сетевые технологии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преподаватель А.Е. Гукай

подпись

Нормоконтроль \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. лаборант В.Г. Медведева

подпись

Курсовая работа защищена \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата итоговая оценка комиссия

Подписи членов комиссии: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

Донецк  
2019

ГОУ ВПО «Донецкий национальный университет»

кафедра Компьютерных технологий

Утверждаю

Зав. кафедрой КТ

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

подпись

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

дата

**ЗАДАНИЕ**

на курсовую работу студента 3 курса Фурсова Д.В.

*Тема курсовой работы*: Разработка динамического сайта для учета методической деятельности кафедры компьютерных технологий

*Краткая постановка задачи*: 1. Изучить и проанализировать предметную область методической деятельности кафедры 2. Описать бизнес-процессы системы. 3. Ознакомиться с программными продуктами для создания динамических сайтов, выполнить установку Web-сервера Apache, реляционной системы управления базами данных MySQL, языка разработки серверных скриптов PHP, веб-фреймворка Laravel. 4. Разработать техническое задание на создание динамического сайта. 5. Разработать проект программного обеспечения системы: спроектировать концептуальную, логическую и физическую модели базы данных, создать базу данных для хранения информации о методических изданиях кафедры компьютерных технологий, преподавателях, дисциплин, видах изданий. 6. Разработать средствами языка программирования серверных приложений PHP, веб-фреймворка Laravel, набора инструментов Bootstrap 4, который позволяет выполнять учет методических изданий кафедры компьютерных технологий. 8. Протестировать программное обеспечение; 9. Оформить отчёт.

*Исходные данные*: 1. Рекомендации для учета методической деятельности кафедры. 2. Документация по веб-фреймворку Laravel. 3. Документация по набору инструментов Bootstrap 4. 4. Документация по системе управления базами данных MySQL.

*Ожидаемые результаты*: Разработка динамического сайта для учета методической деятельности кафедры компьютерных технологий.

*Календарный план работы:*

|  |  |  |
| --- | --- | --- |
| Даты консультаций | Этапы выполнения работы | Отметки о выполнении |
| 04.02.2019 | Постановка задачи и обсуждение литературы | Выполнено |
| 08.02.2019 | Предварительное утверждение содержания отчёта | выполнено |
| 15.02.2019 | Утверждение проекта, алгоритмов, методов, технологий | выполнено |
| 13.03.2019 | Ход реализации проекта | выполнено |
| 21.03.2019 | Обсуждения организации тестирования программы | выполнено |
| 01.02.2019 | Демонстрация программного продукта руководителю | выполнено |
| 01.02.2019 | Оформление отчёта | выполнено |
| 01.02.2019 | Предоставление отчёта руководителю | выполнено |

Дата выдачи задания 04.02.2019 года

Студент Д.В. Фурсов

Руководитель В.Н. Котенко

**АННОТАЦИЯ**

Отчёт о курсовой работе: 47 с., 17 рис., 2 приложения,   
10 источников.

Объект исследования – методической деятельность кафедры компьютерных технологий.

Предмет исследования – автоматизация учета методической деятельности кафедры компьютерных технологий.

Цель работы – разработать динамический сайт для автоматизации учета методической деятельности кафедры компьютерных технологий.

Метод исследования – анализ возможностей языка PHP и фреймвёрка Laravel для создания и обработки баз данных и разработки серверных приложений.

В курсовой работе было разработать динамический сайт для автоматизации учета методической деятельности кафедры компьютерных технологий.

Динамический сайт для автоматизации учета методической деятельности кафедры компьютерных технологий может использоваться

Дальнейшее развитие системы связано с добавлением веб-фреймворка Vue.JS, расширением базы для возможности учета методических изданий физико-технического факультета, расширением круга пользователей, включая студентов, возможностью онлайн просмотра и скачивания методических изданий.

УЧЕТ МЕТОДИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ КАФЕДРЫ, ДИНАМИЧЕСКИЙ САЙТ, APACHE, MYSQL, PHP, LARAVEL.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 7](#_Toc6684038)

[1 Анализ предметной области 8](#_Toc6684039)

[1.1 Состояние вопроса 8](#_Toc6684040)

[1.2 Моделирование существующих бизнес-процессов 8](#_Toc6684041)

[1.3 Актуальность и цель работы 9](#_Toc6684042)

[2 Техническое задание 11](#_Toc6684043)

[2.1 Описание области применения и исходных данных приложения 11](#_Toc6684044)

[2.2 Требования к пользовательским интерфейсам 11](#_Toc6684045)

[2.3 Требования к аппаратным и программным интерфейсам 11](#_Toc6684046)

[2.4 Требования к пользователям продукта 12](#_Toc6684047)

[2.5 Функции продукта 12](#_Toc6684048)

[2.6 Ограничения 13](#_Toc6684049)

[3 Обоснование выбора инструментальных средств для реализации поставленной задачи 13](#_Toc6684050)

[3.1 Язык программирования серверных скриптов PHP 13](#_Toc6684051)

[3.2 Реляционная система управления базами данных MYSQL 16](#_Toc6684052)

[3.3 WEB-сервер APACHE 17](#_Toc6684053)

[3.4 Laravel 5 19](#_Toc6684054)

[3.5 Bootstrap 4 21](#_Toc6684055)

[3.6 Архитектура «клиент-сервер» 21](#_Toc6684056)

[4 Разработка базы данных для динамического сайта для учета методической деятельности кафедры компьютерных технологий 23](#_Toc6684057)

[4.1 Концептуальное проектирование БД 23](#_Toc6684058)

[4.2 Логическое проектирование БД 24](#_Toc6684059)

[4.3 Физическое проектирование БД 25](#_Toc6684060)

[5 Динамический сайт для учета методической деятельности кафедры компьютерных технологий 28](#_Toc6684061)

[5.1 Входные и выходные данные 28](#_Toc6684062)

[5.2 Структура динамического сайта 28](#_Toc6684063)

[5.3 Описание алгоритма работы и SQL-запросов скриптов динамического сайта 30](#_Toc6684064)

[6 Тестирование программного продукта 36](#_Toc6684065)

[6.1 Аппаратные, системные и программные требования 36](#_Toc6684066)

[6.2 Описание установки динамического сайта. 36](#_Toc6684067)

[6.2.1 Установка, настройка и запуск Open Server 36](#_Toc6684068)

[6.2.2 Установка и настройка Composer и Laravel 37](#_Toc6684069)

[6.2.3 Установка, настройка и запуск сервера баз данных MySQL 37](#_Toc6684070)

[6.2.4 Установка и запуск динамического сайта 38](#_Toc6684071)

[6.3 Описание контрольных примеров 38](#_Toc6684072)

[ЗАКЛЮЧЕНИЕ 39](#_Toc6684073)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 40](#_Toc6684074)

[ПРИЛОЖЕНИЕ А 42](#_Toc6684075)

[ПРИЛОЖЕНИЕ Б 44](#_Toc6684076)

# ВВЕДЕНИЕ

В последнее десятилетие благодаря бурному развитию Интернета в программировании выделяют отдельное направление — Web-программирование. Поначалу оно не могло и сравниться по своей сложности с другими областями программистского ремесла, не "дотягиваясь" не только до системного, но даже и до прикладного уровня. В наши дни, однако, роль Web-программирования в структуре глобальной сети возрастает, соответственно увеличивается и средняя оценка сложности сценариев. Многие системы (например, поисковые) по объему кода приближаются к размеру исходных кодов серьезных прикладных программ. Доля же статических страниц в Web постоянно падает; на смену им приходят динамические страницы, сгенерированные автоматически тем или иным сценарием. [1]

Методическая работа является важнейшей составной частью деятельности руководства и Ученого совета вуза, факультетов, кафедр, центров и лабораторий, всего профессорско-преподавательского, научного и учебно-вспомогательного состава вуза. Методическая работа в вузе осуществляется с целью качественной разработки образовательных программ, обеспечения эффективной организации учебно-воспитательного процесса, качественного проведения всех видов учебных занятий, повышения профессионального уровня руководящего, профессорско-преподавательского, научного и учебно-вспомогательного состава вуза. [2]

В данной курсовой работе будет разработан динамический сайт для автоматизации учета методической деятельности кафедры компьютерных технологий.

# 1 Анализ предметной области

## 1.1 Состояние вопроса

Результаты учебно-методической деятельности университета являются важнейшей составляющей его успешности и конкурентоспособности на рынке образования. Над проблемой повышения результативности и эффективности деятельностей, приведенных выше, в последние годы работают не только различные государственные инстанции, но сотрудники различных вузов, в том числе и ДОННУ.

Уровень целостного образовательного процесса в вузе в большей мере зависит от его организации, профориентационной работы, разработанных учебных планов, контроля и оценки качества образования и т.п. Особое внимание уделяется квалификационным требованиям к сотрудникам, в которых выделены показатели учебно-методической работы. В основном эти виды деятельности относятся к преподавателям вуза и направлены на повышение уровня их научно-исследовательской деятельности, воспитания и квалификации будущих специалистов.

## 1.2 Моделирование существующих бизнес-процессов

Контекстная диаграмма, описывающая функционирование учета методической деятельности кафедры компьютерных технологий приведена на рисунке 1.1.

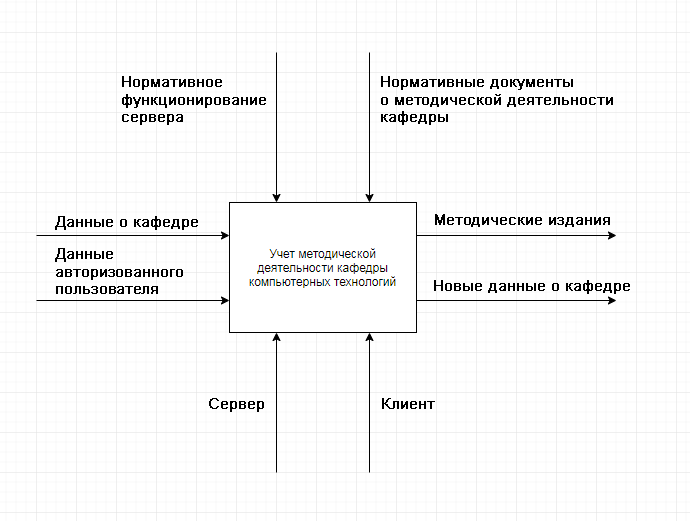


Рисунок 1.1 – Контекстная диаграмма

## 1.3 Актуальность и цель работы

Модернизация системы образования рассматривается сегодня как потенциал инновационного развития общества, залог будущего благополучия граждан и безопасности страны. И это закономерно, так как именно изменения в образовании во многом обеспечивают интенсивное движение экономических, социальных и политических процессов в современном мире.

Преподаватель — ключевая фигура реформирования образования. В стремительно меняющемся открытом мире человеку предъявляются жесткие требования — это высокое качество образования, коммуникабельность, целеустремленность, креативность. Обучить человека современному мышлению, способного самореализоваться в жизни, могут только преподаватели, обладающие высоким профессионализмом.

Не секрет, что образовательные результаты студентов напрямую зависят от качества преподавания. Данный постулат объясняет актуальность эффективного механизма профессионального развития преподавателей, каким является методическая работа, и требует её развития и совершенствования.

Поскольку методическая работа может существенно влиять на качество обучения, на конечные результаты работы образовательного учреждения, можно рассматривать ее как важный фактор управления образовательным процессом. [3]

Ситуация, описанная выше, обосновывает актуальность методической работы. Однако, имея большое количество видов методических изданий, преподавателей и дисциплин на кафедре, учёт методических изданий становится всё сложнее и запутаннее. Данная курсовая работа будет направлена на устранение этой проблемы.

# 2 Техническое задание

## 2.1 Описание области применения и исходных данных приложения

Данное приложение предназначено для преподавателя, ведущего учет методической деятельности кафедры компьютерных технологий.

## 2.2 Требования к пользовательским интерфейсам

Пользовательский интерфейс должен предполагать черно-белые оттенки, анимацию для выделения элементов на сайте, навигационное меню с возможностью просмотра всех изданий, добавления новых изданий, взаимодействия с данными о кафедре, входом и выходом из аккаунта, подвал с информацией о кафедре.

Перед входом на главную страницу отправить пользователя на страницу авторизации.

На главной странице вывести таблицу с информацией о методических изданиях кафедры компьютерных технологии, кнопки для редактирования и удаления записей, фильтры для поиска нужных изданий по авторам, дисциплинам и году издания.

Все возможные данные, которые предоставляет база данных пользователю, отобразить пользователю в виде выпадающего списка с возможностью поиска и выбора нескольких вариантов.

## 2.3 Требования к аппаратным и программным интерфейсам

Для установки и работы программы необходимо иметь вычислительную систему следующей базовой конфигурации:

* Windows (32-bit или 64-bit): Windows 8 / Windows 7 / Windows Server 2008 / Windows Vista / Windows XP SP3;
* процессор: 2.0 ГГц;
* оперативная память: 512 Mb;
* HDD: 1 Гб.

Необходимо обеспечить программное взаимодействие системы с:

* набором библиотек Microsoft Visual C++ 2005-2008-2010 Redistributable Package x86;
* PHP и Composer;
* веб-фреймворк Laravel;
* библиотекой  jQuery и набором инструментов Bootstrap 4;
* Web-сервером Apache;
* сервером управления базами данных (БД) MySQL.

## 2.4 Требования к пользователям продукта

Пользователь должен обладать базовыми навыками владения персональным компьютером, а также владеть навыками работы с Open Server.

## 2.5 Функции продукта

Основные функции:

1. Авторизация пользователей;
2. Просмотр списка методических изданий;
3. Добавление нового издания;
4. Редактирование существующего издания;
5. Удаление издания;
6. Фильтрация списка методических изданий по авторам, дисциплинам и году издания;
7. Добавление авторов, дисциплин и видов изданий;
8. Редактирование существующих авторов, дисциплин и видов изданий;
9. Удаление авторов, дисциплин и видов изданий.

## 2.6 Ограничения

1. Динамический сайт не будет работать без заранее установленного проекта Laravel.
2. После каждого действия на сайте, страница будет полностью перезагружаться.

# 3 Обоснование выбора инструментальных средств для реализации поставленной задачи

## 3.1 Язык программирования серверных скриптов PHP

PHP (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) - это распространенный язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML.

PHP отличается от JavaScript тем, что PHP-скрипты выполняются на сервере и генерируют HTML, который посылается клиенту. Если бы у вас на сервере был размещен скрипт, подобный вышеприведенному, клиент получил бы только результат его выполнения, но не смог бы выяснить, какой именно код его произвел. Вы даже можете настроить свой сервер таким образом, чтобы обычные HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта. Хотя PHP, главным образом, предназначен для работы в среде веб-серверов, область его применения не ограничивается только этим.

Главная область применения PHP - написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять все то, что выполняет любая другая программа CGI, например, обрабатывать данные форм, генерировать динамические страницы или отсылать и принимать cookies. Но PHP способен выполнять намного больше.

Существуют три основных области применения PHP.

Создание скриптов для выполнения на стороне сервера. PHP традиционно и наиболее широко используется именно таким образом. Для этого вам будут необходимы три вещи. Интерпретатор PHP (в виде программы CGI или серверного модуля), веб-сервер и браузер. Для того чтобы можно было просматривать результаты выполнения PHP-скриптов в браузере, нужен работающий веб-сервер и установленный PHP. Просмотреть вывод PHP-программы можно в браузере, получив PHP-страницу, сгенерированную сервером. В случае, если вы просто экспериментируете, вы вполне можете использовать свой домашний компьютер вместо сервера.

Создание скриптов для выполнения в командной строке. Вы можете создать PHP-скрипт, способный запускаться без сервера или браузера. Все, что вам потребуется - парсер PHP. Такой способ использования PHP идеально подходит для скриптов, которые должны выполняться регулярно, например, с помощью cron (на платформах \*nix или Linux) или с помощью планировщика задач (Task Scheduler) на платформах Windows. Эти скрипты также могут быть использованы в задачах простой обработки текстов.

Создание оконных приложений, выполняющихся на стороне клиента. Возможно, PHP является не самым лучшим языком для создания подобных приложений, но, если вы очень хорошо знаете PHP и хотели бы использовать некоторые его возможности в своих клиентских приложениях, вы можете использовать PHP-GTK для создания таких приложений. Подобным образом вы можете создавать и кросс-платформенные приложения. PHP-GTK является расширением PHP и не поставляется вместе с основным дистрибутивом PHP.

PHP [доступен](https://www.php.net/manual/ru/install.php) для большинства операционных систем, включая Linux, многие модификации Unix (такие как HP-UX, Solaris и OpenBSD), Microsoft Windows, macOS, RISC OS и многие другие. Также в PHP включена поддержка большинства современных веб-серверов, таких как Apache, IIS и многих других. В принципе, подойдет любой веб-сервер, способный использовать бинарный файл FastCGI PHP, например, lighttpd или nginx. PHP может работать в качестве модуля или функционировать в качестве процессора CGI.

Таким образом, выбирая PHP, вы получаете свободу выбора операционной системы и веб-сервера. Более того, у вас появляется выбор между использованием процедурного или объектно-ориентированного программирования (ООП) или же их сочетания.

PHP способен генерировать не только HTML. Доступно формирование изображений, файлов PDF и даже роликов Flash (с использованием libswf и Ming), создаваемых “на лету“. PHP также способен генерировать любые текстовые данные, такие, как XHTML и другие XML-файлы. PHP может осуществлять автоматическую генерацию таких файлов и сохранять их в файловой системе вашего сервера вместо того, чтобы отдавать клиенту, организуя, таким образом, серверный кэш для вашего динамического контента.

Одним из значительных преимуществ PHP является поддержка [широкого круга баз данных](https://www.php.net/manual/ru/refs.database.php). Создать скрипт, использующий базы данных, - невероятно просто. Можно воспользоваться расширением, специфичным для отдельной базы данных (таким как [mysql](https://www.php.net/manual/ru/book.mysqli.php)) или использовать уровень абстракции от базы данных, такой как [PDO](https://www.php.net/manual/ru/book.pdo.php), или подсоединиться к любой базе данных, поддерживающей Открытый Стандарт Соединения Баз Данных (ODBC), с помощью одноименного расширения [ODBC](https://www.php.net/manual/ru/book.uodbc.php). Для других баз данных, таких как CouchDB, можно воспользоваться [cURL](https://www.php.net/manual/ru/book.curl.php) или [сокетами](https://www.php.net/manual/ru/book.sockets.php).

PHP также поддерживает взаимодействие с другими сервисами через такие протоколы, как LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) и многих других. Кроме того, вы получаете возможность работать с сетевыми сокетами напрямую. PHP поддерживает стандарт обмена сложными структурами данных WDDX практически между всеми языками веб-программирования. Обращая внимание на взаимодействие между различными языками, следует упомянуть о поддержке объектов Java и возможности их использования в качестве объектов PHP.

PHP имеет много возможностей по [обработке текста](https://www.php.net/manual/ru/refs.basic.text.php), включая регулярные выражения Perl ([PCRE](https://www.php.net/manual/ru/book.pcre.php)) и много других расширений и инструментов [для обработки и доступа к XML-документам](https://www.php.net/manual/ru/refs.xml.php). В PHP обработка XML-документов стандартизирована и происходит на базе мощной библиотеки [libxml2](https://www.php.net/manual/ru/book.libxml.php), расширив возможности обработки XML добавлением новых расширений [SimpleXML](https://www.php.net/manual/ru/book.simplexml.php), [XMLReader](https://www.php.net/manual/ru/book.xmlreader.php) и [XMLWriter](https://www.php.net/manual/ru/book.xmlwriter.php). [4]

## 3.2 Реляционная система управления базами данных MYSQL

Если имеется небольшой статический проект, в котором содержится всего несколько html страниц, то применение базы данных (БД) вовсе и не нужно. Но это редкие случаи, когда создаваемые web-мастерами проекты остаются практически незаполненными. Как правило, сайты продолжают наполняться контентом, они становятся более загруженными. Такие проекты уже являются динамичными, и без базы данных вести их очень сложно.

Храня гигабайты информации, распределенной по сотням файлов, вам придется тратить уйму времени при выдаче необходимых строк в процессе функционирования сервера. Чтобы избежать этого, нужны БД, занимающиеся группировкой и упорядочиванием информации. Код для базы данных значительно проще, чем код, предназначенный с целью применения файлов. При этом запрос обрабатывается куда быстрее.

В БД все данные представлены таблицей с комментариями, информацией об объектах и т.п. Стоит отметить, что БД постоянно меняется, дополняется новыми данными, исправляется та информация, которая уже есть в ней. И чтобы не возникало трудностей в процессе администрирования, добавления и изменения информации, были придуманы специальные системы управления БД. Одной из них является MySQL, которая в своей сфере пользуется наибольшим спросом.

База данных сайта MySQL – это система, предназначенная для хранения и обработки информации. Комплекс таблиц, взаимосвязанных между собой, для доступа к которым применяется система управления базами данных (СУБД) MySQL. По сути, MySQL – это специальная программа с открытым кодом, которая используется на сервере SQL. Данная программа не способна обрабатывать большое количество информации, однако она идеальна для небольших и крупных веб-ресурсов.

Одними из главных плюсов MySQL, благодаря которым программа нравится многим веб-мастерам, является высокая скорость обработки информации и функциональность СУБД. Она доступна любому пользователю абсолютно бесплатно, что тоже склоняет чашу весов в сторону MySQL. Большинство современных хостингов используют именно MySQL.

Веб-мастер получает возможность совершенствовать код и вносить в него собственные изменения, что делает программу гибкой. MySQL является лишь одним из многих программных обеспечений, работающих с SQL. [5]

## 3.3 WEB-сервер APACHE

Apache – это программное обеспечение с открытым исходным кодом, веб-сервер, [который обеспечивает работу около 46% сайтов](https://w3techs.com/technologies/details/ws-apache/all/all) по всему миру. Официальное название – [Apache HTTP Сервер](https://httpd.apache.org/" \t "_blank), поддерживается и развивается компанией Apache Software Foundation.

Веб-сервер позволяет владельцам сайтов обслуживать их контент в интернете, о чём понятно с самого название “веб-сервер”. Apache один из самый старых и надёжный веб-серверов с первой версией выпуска более 20 лет назад в 1995 году.

Когда кто-либо хочет посетить сайт, он вводит доменное имя в адресную строку своего браузера. Затем веб-сервер доставляет запрашиваемые файлы как виртуальный посыльный.

Хоть Apache и называется веб-сервер, но в реальном положении вещей он является не сервером, а программой, которая запускается на сервере. Его задача установить соединение между сервером и браузером посетителей (Firefox, Google Chrome, Safari и др.) при доставке файлов туда и обратно между ними (клиент-серверная структура). Apache – это кроссплатформенное программное обеспечение, что значит оно хорошо работает как на Unix, так и на Windows серверах.

Когда посетитель хочет загрузить страницу вашего сайта, например, домашнюю страницу или страницу “О нас”, его браузер отправляет запрос на ваш сервер и Apache возвращает ответ со всеми запрошенными файлами (текст, изображение и так далее). Сервер и клиент взаимодействуют по протоколу HTTP и Apache ответственен за гладкое и безопасное соединение между двумя машинами.

Apache хорошо и удобно настраиваемый поскольку имеет модульную структуру. Модули позволяют администраторам сервера включать или выключать дополнительную функциональность. У Apache есть [модули](https://httpd.apache.org/docs/2.4/mod/) безопасности, кэширования, редактирования URL, аутентификации по средством пароля и другие.

Веб-сервер Apache может быть прекрасным выбором для запуска вашего сайта на стабильной и универсальной платформе. Но у него есть и свои недостатки, на которые стоит обратить внимание.

Преимущества:

1. Бесплатный даже для использования в коммерческих целях.
2. Надёжный, стабильное программное обеспечение.
3. Часто обновляемый, регулярные патчи безопасности.
4. Гибкий благодаря своей модульной структуре.
5. Легко настраиваемый, дружелюбный для начинающих.
6. Кроссплатформенный (работает одинаково хорошо на Unix и на Windows серверах).
7. Работает для сайтов на WordPress сразу после установки без дополнительных настроек.
8. Большое сообщество и легко доступная поддержка в случае любой проблемы.

Недостатки:

1. Проблемы производительности на сайтах с большим трафиком.
2. Слишком много параметров конфигруации могут привести к уязвимостям в безопасности. [6]

## 3.4 Laravel 5

Laravel – PHP-фреймворк, быстро набирающий популярность среди веб-разработчиков и помогающий обеспечить максимально удобное [создание сайтов](http://it-devgroup.com/) различного уровня сложности с изящным и четко структурированным синтаксисом и архитектурой. Несмотря на свой юный возраст (первый выпуск осуществлен в 2011 году), этот фреймворк считается одним из лучших, имеет большое сплоченное сообщество и продолжает успешно развиваться.

Разработка на Laravel 5 прекрасно подходит для создания:

1. [CRM](https://it-devgroup.com/uslugi/razrabotka-crm/) и [ERP систем](https://it-devgroup.com/uslugi/razrabotka-erp/);
2. [Корпоративных порталов](https://it-devgroup.com/uslugi/razrabotka-portala/);
3. [Интернет-магазинов](https://it-devgroup.com/uslugi/razrabotka-internet-magazina/);
4. [Систем онлайн-бронирования](https://it-devgroup.com/uslugi/razrabotka-buking-sistem/).

Laravel является достаточно гибким фреймворком и позволяет решать нестандартные задачи, структурировать веб-сайт в соответствии с существующей логикой и поставленными целями.

Также к достоинствам Laravel можно отнести:

1. Обширный функционал. Cоздание сайтов любого уровня возможно благодаря огромной функциональности. Используя этот фреймворк, можно реализовать проекты, предоставляющие возможность интеграции необходимого функционала в соответствии с индивидуальными требованиями и особенностями конкретного бизнеса.
2. Возможность создать гибкую админ панель. Можно реализовать наиболее удобный вариант управления ресурсом, создавая индивидуальную панель администратора под задачи конкретного веб-проекта.
3. Безопасность баз данных. Возможность получить несанкционированный доступ к базе данных, созданной с использованием Laravel, крайне сложно. Высокий уровень безопасности гарантирует надежную защиту от SQL- injection, атак типа XSS, CSRF.
4. Регулярные релизы. Исходный код изменяется с учетом нововведений в PHP и потребностей программистов. Свежие обновления помогают устранить ранее существовавшие проблемы и сделать фреймворк еще более удобным.
5. Популярность и активное сообщество. Наличие большого сообщества открывает простор для динамичной коммуникации, обмена личным опытом и мнениями, решения всевозможных вопросов связанных с проектированием и поддержкой интернет-ресурсов.
6. Масштабируемость. Разработка на Laravel 5 предполагает возможность расширения функционала (интеграции дополнительных модулей) без существенных затрат на изменение текущей системы, а также риска возникновения нежелательных потерь для веб-ресурса.

Недостатки Laravel:

1. Реализация веб-проектов со специфическим функционалом с нуля отнимает значительно больше времени в сравнении с созданием шаблонных вариантов.
2. Документация достаточно сложная, поэтому новичку будет крайне сложно создать полноценный веб-сайт. [7]

## 3.5 Bootstrap 4

Bootstrap – это WEB-фреймворк, содержащий готовые CSS, HTML и JavaScript компоненты, который изначально создавался для внутреннего использования компанией «Twitter» с рабочим названием «TwitterBlueprint», но в итоге был опубликован в открытый доступ и стал хорошим набором инструментов для front-end разработки под названием «Bootstrap».

Преимущества фреймворка Bootstrap:

1. Высокая скорость разработки макетов страниц сайта. Bootstrap содержит огромный набор готовых решений и элементов.
2. Кроссбраузерность и адаптивность сайта. Все элементы фреймворка адаптивны под все устройства и корректно отображаются во всех современных браузерах.
3. Легкость в использовании. Даже человек, имеющий базовые знания о HTML и CSS, может свободно создавать web-страницы с использованием фреймворка.
4. Простота в обучении. У Bootstrap очень хорошая документация с большим количеством примеров готового кода.

О качестве фреймворка говорит то, что множество тем оформления для подавляющего большинства популярных CMS, таких как WordPress, Joomla и тому подобных, разработаны с применением Bootstrap. [10]

## 3.6 Архитектура «клиент-сервер»

Архитектура клиент-сервер определяет лишь общие принципы взаимодействия между компьютерами, детали взаимодействия определяют различные протоколы. Данная концепция нам говорит, что нужно разделять машины в сети на клиентские, которым что-то нужно и на серверные, которые отправляют то, что нужно. При этом взаимодействие всегда начинает клиент, а правила, по которым происходит взаимодействие описывает протокол.

Преимуществом модели взаимодействия клиент-сервер является то, что программный код клиентского приложения и серверного разделен. Если мы говорим про локальные компьютерные сети, то к преимуществам архитектуры клиент-сервер можно отнести пониженные требования к машинам клиентов, так как большая часть вычислительных операций будет производиться на сервере, а также архитектура клиент-сервер довольно гибкая и позволяет администратору сделать локальную сеть более защищенной.

К недостаткам модели взаимодействия клиент-сервер можно отнести то, что стоимость серверного оборудования значительно выше клиентского. Сервер должен обслуживать специально обученный и подготовленный человек. Если в локальной сети отключается сервер, то и клиенты не смогут работать.

В качестве заключения явно стоит акцентировать внимание на том, что архитектура клиент-сервер не делит машины на “только клиент” или “только сервер”, а скорее позволяет распределить нагрузку и разделить функционал между клиентской частью и серверной. [8]

# 4 Разработка базы данных для динамического сайта для учета методической деятельности кафедры компьютерных технологий

## 4.1 Концептуальное проектирование БД

Концептуальное проектирование – построение семантической модели предметной области (информационной модели самого высокого уровня абстракции). Концептуальная модель базы данных включает в себя:

1. Объекты, обозначающие вещи, обозначаются в виде прямоугольника;
2. Атрибуты, характеризующие объект – в виде овала;
3. Связи между объектами – ромбами.

В ходе изучения предметной области выделены следующие сущности, представляющие информационное значение для задачи:

1. Методические издания – информация об изданиях кафедры компьютерных технологий;
2. Дисциплина – информация об имеющихся дисциплинах на кафедре компьютерных технологий;
3. Преподаватели – информация о преподавателях кафедры компьютерных технологий;
4. Вид издания – информация о видах методических изданий;
5. Обложка – информация о типах обложек, которые будут использованы в методических изданиях;
6. Формат бумаги – информация о формате бумаги, которую будут использовать для методических изданий;
7. Месяц издания – информация о месяцах, в которых будут изданы методические издания.

Концептуальное представление базы данных приведено на рисунке 4.1.

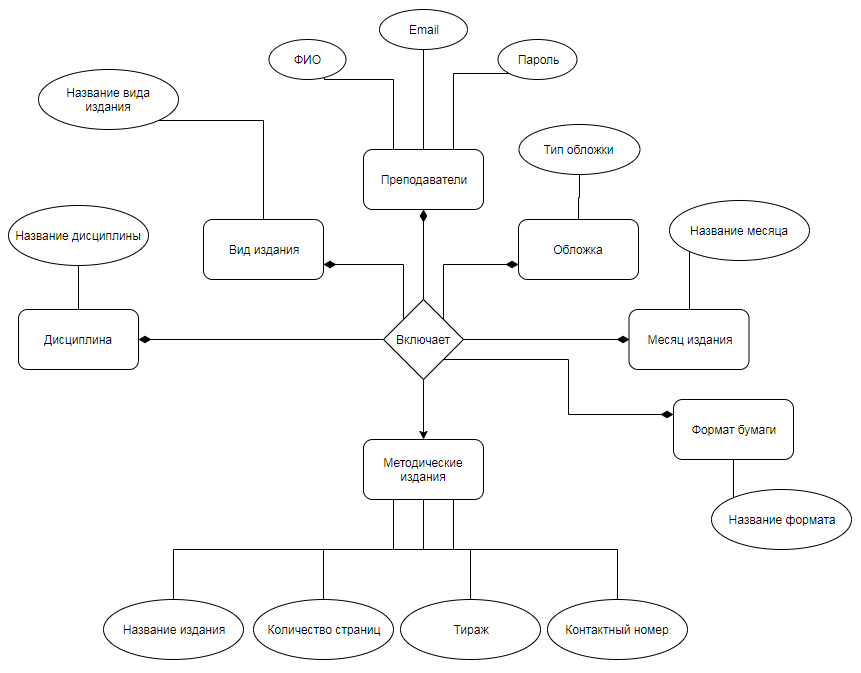


Рисунок 4.1 – Концептуальная модель базы данных

## 4.2 Логическое проектирование БД

Второй этап проектирования базы данных называется логическим проектированием базы данных. Его цель состоит в создании логической модели данных для исследуемой части предприятия. Концептуальная модель данных, созданная на предыдущем этапе, уточняется и преобразуется в логическую модель данных. Логическая модель данных учитывает особенности выбранной модели организации данных в целевой СУБД (например, реляционная модель).

Реляционная модель характеризуется своей простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для манипулирования данными.

В реляционных моделях данных объекты и взаимосвязи между ними представляются с помощью таблиц. Каждая таблица представляет один объект и состоит из строк и столбцов. [9]

Так как для разрабатываемого программного продукта была выбрана реляционная модель данных MySQL.

Логическое представление базы данных приведено на рисунке 4.2.

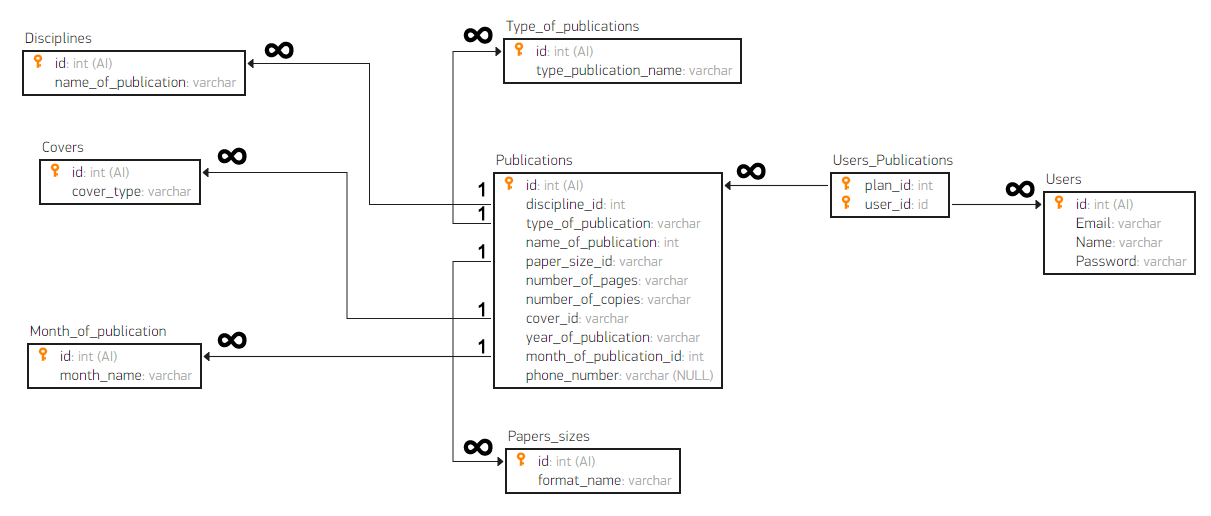


Рисунок 4.2 – Логическая модель базы данных

## 4.3 Физическое проектирование БД

Физическое проектирование – создание схемы базы данных для конкретной СУБД.

Физическая модель данных описывает то, как данные хранятся в компьютере, представляя информацию о структуре записей, их упорядоченности и существующих путях доступа.

В физической модели описываются типы, идентификаторы и разрядность полей. Физическая модель данных отражает физическое размещение данных на машинных носителях, а именно какой файл, какие объекты, с какими атрибутами содержит и каковы типы этих атрибутов.

Физическое представление таблиц базы данных приведено на рисунках 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4,10.



Рисунок 4.3 – Физическая модель таблицы «Publications»

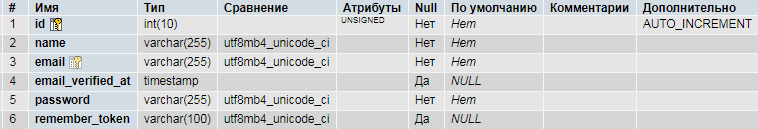


Рисунок 4.4 – Физическая модель таблицы «Users»



Рисунок 4.5 – Физическая модель таблицы «Users\_Publications»



Рисунок 4.6 – Физическая модель таблицы «Type\_of\_publication»



Рисунок 4.7 – Физическая модель таблицы «Papers\_sizes»



Рисунок 4.8 – Физическая модель таблицы «Month\_of\_publications»



Рисунок 4.9 – Физическая модель таблицы «Disciplines»



Рисунок 4.10 – Физическая модель таблицы «Covers»

# 5 Динамический сайт для учета методической деятельности кафедры компьютерных технологий

## 5.1 Входные и выходные данные

Входные данные динамического сайта:

1. Список дисциплин кафедры компьютерных технологий;
2. Список преподавателей кафедры компьютерных технологий;
3. Список видов методических изданий;
4. Формат бумаги и тип обложки;
5. Количество страниц;
6. Тираж;
7. Год и месяц выпуска;
8. Контактный телефон преподавателя.

Результатом работы приложения является список методических изданий кафедры компьютерных технологий.

## 5.2 Структура динамического сайта

Структура динамического сайта приведена на рисунке 5.1.

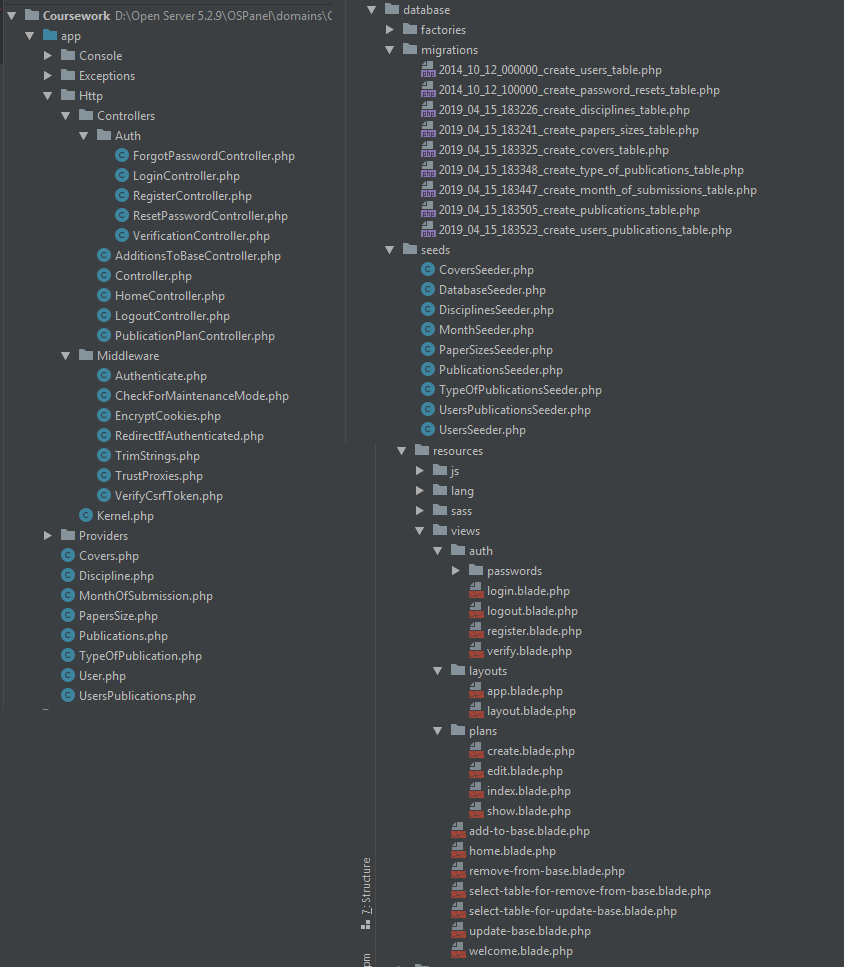


Рисунок 5.1 – Структура динамического сайта

Связь скриптов показана на рисунке 5.2.

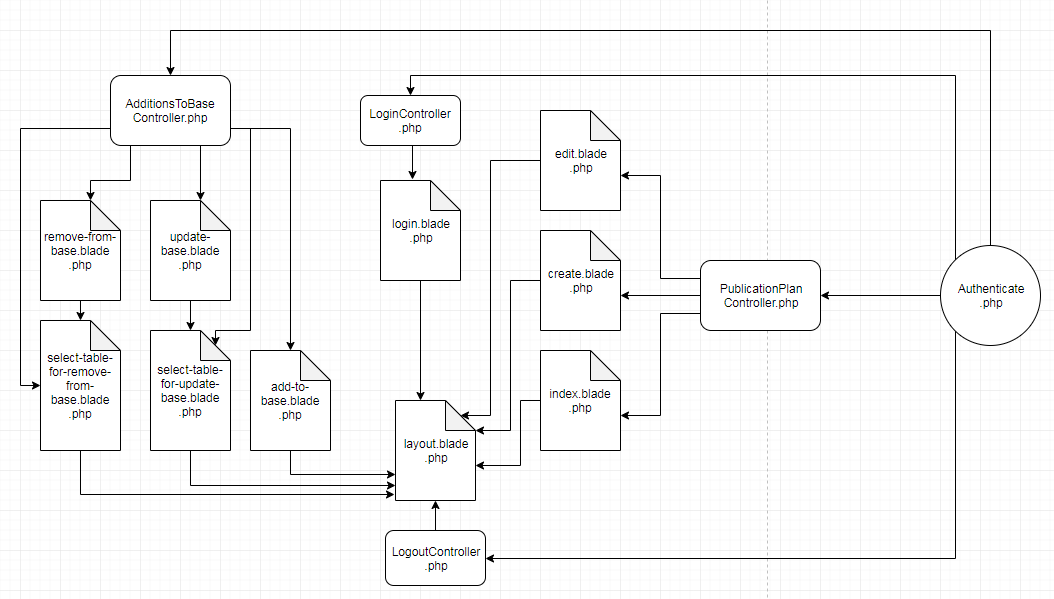


Рисунок 5.2 – Связь скриптов динамического сайта

## 5.3 Описание алгоритма работы и SQL-запросов скриптов динамического сайта

За авторизацию пользователя отвечает автоматически сгенерированный фреймворком Laravel контроллер LoginController.php и посредник Authenticate.php. Также, стоит упомянуть, что через посредник Authenticate.php проходят все запросы. Таким образом, неавторизованные пользователи не смогут получить доступ к страницам сайта.

В проекте существует ресурсный контроллер PublicationPlanController.php, который отвечает за все действия, совершаемые с методическими изданиями. Рассмотрим, что делают некоторые функции этого контроллера:

1. index(Request $request)

Эта функция формирует запрос для методических изданий:

*// get all the plans with foreign key*

*$users = DB::table('publications')*

*->join('disciplines', 'publications.discipline\_id', '=', 'disciplines.id')*

*->join('type\_of\_publication', 'publications.type\_publication\_id', '=', 'type\_of\_publication.id')*

*->join('papers\_sizes', 'publications.paper\_size\_id', '=', 'papers\_sizes.id')*

*->join('covers', 'publications.cover\_id', '=', 'covers.id')*

*->join('month\_of\_submissions', 'publications.month\_of\_submission\_id', '=', 'month\_of\_submissions.id')*

*->select('publications.id', 'publications.discipline\_id', 'disciplines.name\_of\_discipline', 'type\_of\_publication.type\_publication\_name', 'publications.name\_of\_publication',*

*'papers\_sizes.format\_name' , 'number\_of\_pages', 'number\_of\_copies','covers.cover\_type', 'publications.year\_of\_publication',*

*'month\_of\_submissions.month\_name', 'phone\_number')*

*->get();*

Дисциплин и авторов для фильтрации:

*$disciplines\_table = Discipline::all();*

*$autors\_table = User::all();*

И возвращает вместе с шаблоном:

*return view('plans.index')->with([*

*'plans' => $users,*

*'users' => $users\_in\_plans,*

*'disciplines' => $disciplines\_table,*

*'autors' => $autors\_table,*

*'select\_year' => $request->input('select\_year'),*

*'select\_discipline' => $request->input('select\_discipline'),*

*'select\_author' => $select\_author,*

*]);*

1. store(Request $request)

В данном проекте, в функциях, где создаются или обновляются данные проводятся проверка данных подобно этой:

*// validate*

*$request->validate([*

*'discipline\_id' => 'required|numeric',*

*'type\_publication\_id' => 'required|numeric',*

*'name\_of\_publication' => 'required',*

*'author\_id' => "required|array|min:1",*

*'paper\_size\_id' => 'required|numeric',*

*'number\_of\_pages' => 'required|numeric',*

*'number\_of\_copies' => 'required|numeric',*

*'cover\_id' => 'required',*

*'month\_of\_submission\_id' => 'required|numeric',*

*'year\_of\_publication' => 'required|numeric',*

*]);*

Таким образом создается новое издание:

*$plan = new Publications();*

*$plan->discipline\_id = $request->input('discipline\_id');*

*$plan->name\_of\_publication = $request->input('name\_of\_publication');*

*……*

*$plan->save();*

Так, как связь “издания-авторы” относится к связи “многие-ко-многим”, после сохранения нового издания необходимо связать новое издание с авторами в таблице Users\_Publications:

*$newid = Publications::latest()->first()->id;*

*$array\_users = $request->input('author\_id.\*');*

*if(count($array\_users) > 0) {*

*for ($i=0; $i<count($array\_users); $i++) {*

*$plan\_user = new UsersPublications();*

*$plan\_user->plan\_id = $newid;*

*$plan\_user->user\_id = $array\_users[$i];*

*$plan\_user->save();*

*}*

*}*

1. Destroy ($id)

Для начала необходимо удалить внешние ключи в таблице Users\_Publications, а после удалить и само издание:

*// delete foreign key*

*$deletedRows = UsersPublications::where('plan\_id', '=', $id)->delete();*

*// delete*

*$plan = Publications::find($id);*

*$plan->delete();*

В проекте также существует контроллер AdditionToBaseController.php, который отвечает за все действия, совершаемые с данными о кафедре.

Работает он схожим образом как и контроллером описанным выше, за исключением того, что некоторые функции сначала выдают шаблон, для того чтобы пользователь выбрал интересующую его таблицу:

*$select\_table = $request->input('table');*

*$collection = array();*

*switch ($select\_table) {*

*case 'discipline':*

*$table = Discipline::all();*

*$number\_table = 1;*

*foreach ($table as $value) {*

*$collection[$value->id] = $value->name\_of\_discipline;*

*}*

*break;*

*case 'type\_publication':*

*$table = TypeOfPublication::all();*

*$number\_table = 2;*

*foreach ($table as $value) {*

*$collection[$value->id] = $value->type\_publication\_name;*

*}*

*break;*

*case 'name':*

*$table = User::all();*

*$number\_table = 3;*

*foreach ($table as $value) {*

*$collection[$value->id] = $value->name;*

*}*

*break;*

*default:*

*Session::flash('message', 'Что-то пошло не так. Таблицы не существует!');*

*return redirect('/select-table-for-remove-from-base');*

*break;*

*}*

*return view("remove-from-base")->with([*

*'values' => $collection,*

*'select\_table' => $number\_table,*

*]);*

А затем, в зависимости от выбранной таблицы, взаимодействует с данными:

*public function destroy(Request $request)*

*{*

*$elements = $request->input('elements.\*');*

*if (count($elements) > 0) {*

*for ($i = 0; $i < count($elements); $i++) {*

*$element = substr($elements[$i], 0, 1);*

*switch ($element) {*

*case 1:*

*$id = substr($elements[$i], 1);*

*$plans = Publications::all()->where('discipline\_id', '=', $id);*

*foreach ($plans as $key => $value) {*

*$deletedRows = Users\_Publications::where('plan\_id', '=', $value->id)->delete();*

*$value->delete();*

*}*

*$discipline = Discipline::find($id);*

*$name\_element = $discipline->name\_of\_discipline;*

*$discipline->delete();*

*break;*

…….

Стоит упомянуть, что эти данные связаны внешним ключом, поэтому, если вы удалите определенную дисциплину, то вместе с ней вы удалите все методические издания, связанные с этой дисциплиной.

# 6 Тестирование программного продукта

## 6.1 Аппаратные, системные и программные требования

Для функционирования реализованного динамического сайта необходимы следующие аппаратные ресурсы:

* Windows (32-bit или 64-bit): Windows 8 / Windows 7 / Windows Server 2008 / Windows Vista / Windows XP SP3;
* процессор: 2.0 ГГц;
* оперативная память: 512 Mb;
* HDD: 1 Гб.

Для работы динамического сайта и взаимодействия его с созданной базой данных необходимо следующее программное обеспечение:

* набором библиотек Microsoft Visual C++ 2005-2008-2010 Redistributable Package x86;
* PHP и Composer;
* веб-фреймворк Laravel;
* библиотекой  jQuery и набором инструментов Bootstrap 4;
* Web-сервером Apache;
* сервером управления базами данных (БД) MySQL.

## 6.2 Описание установки динамического сайта.

### 6.2.1 Установка, настройка и запуск Open Server

Для работы динамического сайта, прежде всего нужны Apache, MySQL и PHP. Всё это можно получить из локального веб-сервера Open Server.

Для установки Open Server необходимо скачать установщик с официального сайта https://ospanel.io. Следовать инструкции во время установки.

После установки необходимо настроить модули HTTP, PHP и MySQL. Для этого необходимо перейти в настройки Open Server, открыть вкладку “Модули” и выбрать:

* HTTP ­­– Apache-PHP-7.2;
* PHP – PHP-7.2;
* MySQL/MariaDB – MySQL-5.7.

### 6.2.2 Установка и настройка Composer и Laravel

Для управления зависимостями Laravel, необходимо установить Composer, а затем Laravel.

Для начала, необходимо скачать и установить Composer. Во время установки можно взять PHP из Open Server из директории “../ospanel/modules/php/PHP-7.2”.

Для создания проекта Laravel необходимо открыть консоль и вписать команду «composer create-project --prefer-dist laravel/laravel <имя\_проекта>».

### 6.2.3 Установка, настройка и запуск сервера баз данных MySQL

В PhpMyAdmin, установленном в Open Server, создать базу данных.

Далее, в папке с проектом создайте файл “.env” в корневой папке проекта, если он отсутствует, и заполните его соответствующей информацией, такой как информация о его базе данных. В качестве примера откройте файл .env.example.

После того как файл создан, нужно заполнить его исходными таблицами и данными о кафедре. Для этого в консоли нужно перейти в директорию с проектом и прописать команду “php artisan migrate:fresh –seed”

### 6.2.4 Установка и запуск динамического сайта

После установки проекта Laravel настроим Open Server так, чтобы проект запускался по его прямому доменному имени, без дополнительного указания в URL папки /public. В настройках Open Server переходим на вкладку "Домены". В списке "Управление доменами" выбираем "Ручное + Автопоиск". В поле "Имя домена" прописываем свой домен, в поле "Папка домена" выбираем каталог проекта и обязательно в конце указываем папку \public. Нажимаем кнопки "Добавить" и "Сохранить".

После проведения выше перечисленных действий, если все сделано верно, то при наборе в адресной строке только что добавленного доменного имени вы увидите страницу авторизации.

## 6.3 Описание контрольных примеров

После запуска в окне браузера отображается страница авторизации (рисунок A.1).

После авторизации выводится страница с методическими изданиями кафедры (рисунок A2). На этой странице возможно отредактировать существующие издания, удалить их, а также отфильтровать по трем категориям “Год выпуска”, “Дисциплины”, “Авторы” с помощью выпадающих списков. Выпадающие списки поддерживают поиск по введенным символам.

При нажатии на пункт в меню “Новое издание”, находящуюся в верхней части сайта, выводится форма для заполнения информации о новом методическом издании (рисунок A3).

После нажатия на выпадающее меню “Взаимодействие с базой”, можно выбрать необходимое действие “Добавить в базу”, “Обновить в базе”, ”Удалить из базы”. Следуя инструкциям, вы сможете взаимодействовать с элементами в базе данных, которые в дальнейшем появятся в выпадающих списках (рисунок A4). Однако если вы удалите какой-либо элемент, то вы удалите и все связанные с ним методические указания.

# ЗАКЛЮЧЕНИЕ

В результате, разработан динамический сайт для учета методической деятельности кафедры компьютерных технологий.

Разработанный динамический сайт позволяет выполнять учет методических изданий кафедры компьютерных технологий, а именно добавлять, редактировать, удалять методические издания, фильтровать по категориям, добавлять, редактировать и удалять данные о кафедре компьютерных технологий.

Динамический сайт для учета методической деятельности кафедры компьютерных технологий может использоваться преподавателем, который ведет учет методических изданий кафедры компьютерных технологий.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Котеров Д. В. PHP 5. 2 изд. – БХВ-Петербург, 2012. – 1062 с.
2. Коробко А. И. Методическая работа в вузе: цели, задачи, пути и формы ее ведения //Вестник Московского государственного лингвистического университета. – 2012. – №. 16 (649).
3. Актуальность методической работы в общеобразовательном учреждении // Научный журнал “Молодой ученый”

<URL:https://moluch.ru/conf/ped/archive/309/14628/>

(дата обращения: 10.04.2019)

1. Что такое PHP? // Официальный сайт PHP

<URL:https://www.php.net/manual/ru/intro-whatis.php>

(дата обращения: 11.04.2019)

1. База данных сайта MySQL: что это такое, преимущества перед другими СУБД // Вебмастерье.

URL: https://webmasterie.ru/razrabotka/hosting/baza-dannih-mysql

(дата обращения: 09.04.2019)

1. Что Такое Apache? Подробный Обзор Веб-Сервера Apache // Руководства Hostinger.

URL: https://www.hostinger.com.ua/rukovodstva/web-server-apache/

(дата обращения: 09.04.2019)

1. Обзор Laravel 5. Основные преимущества и недостатки // IT DEV GROUP

URL: https://it-devgroup.com/blog/blog-razrabotka/obzor-laravel-5.-osnovnyie-preimushhestva-i-nedostatki/

(дата обращения: 12.04.2019)

1. О модели взаимодействия клиент-сервер // IT-блог о веб-технологиях, серверах, протоколах

URL: https://zametkinapolyah.ru/servera-i-protokoly/o-modeli-vzaimodejstviya-klient-server-prostymi-slovami-arxitektura-klient-server-s-primerami.html

(дата обращения: 12.04.2019)

1. Карпова Т.С. Базы данных: модели, разработка, реализация / Т.С. Карпова. ­– СПб: Питер, 2001. –304 с.
2. Фреймворк Bootstrap – что это такое? // Блог о разработке и продвижении сайта

URL: https://zyubin.ru/frameworks/bootstrap/frejmvork-bootstrap-chto-eto-takoe.html

(дата обращения: 13.04.2019)

# ПРИЛОЖЕНИЕ А

**Экранные формы**

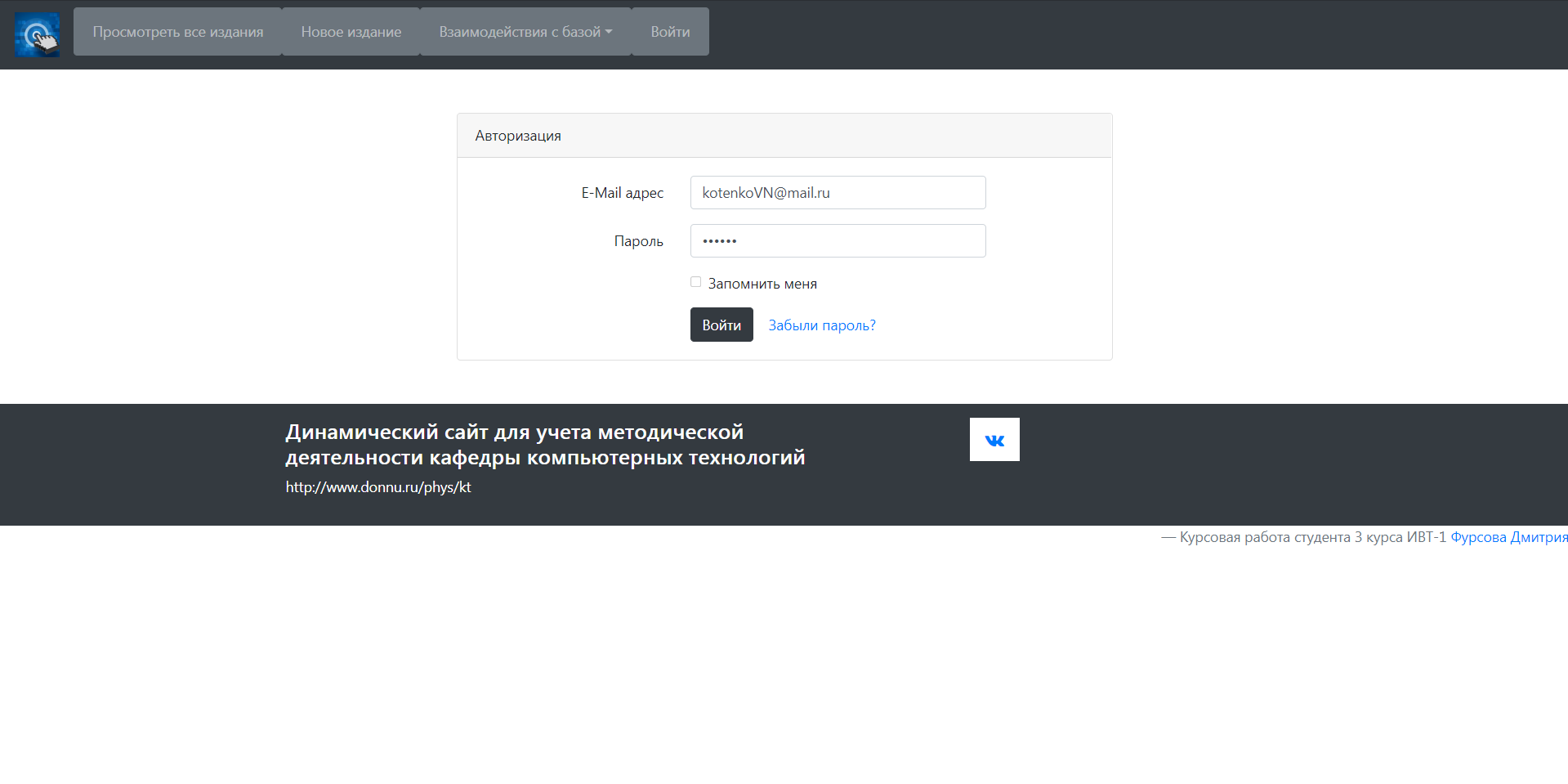


Рисунок А.1 – Страница с формой для добавления нового материала преподавателя

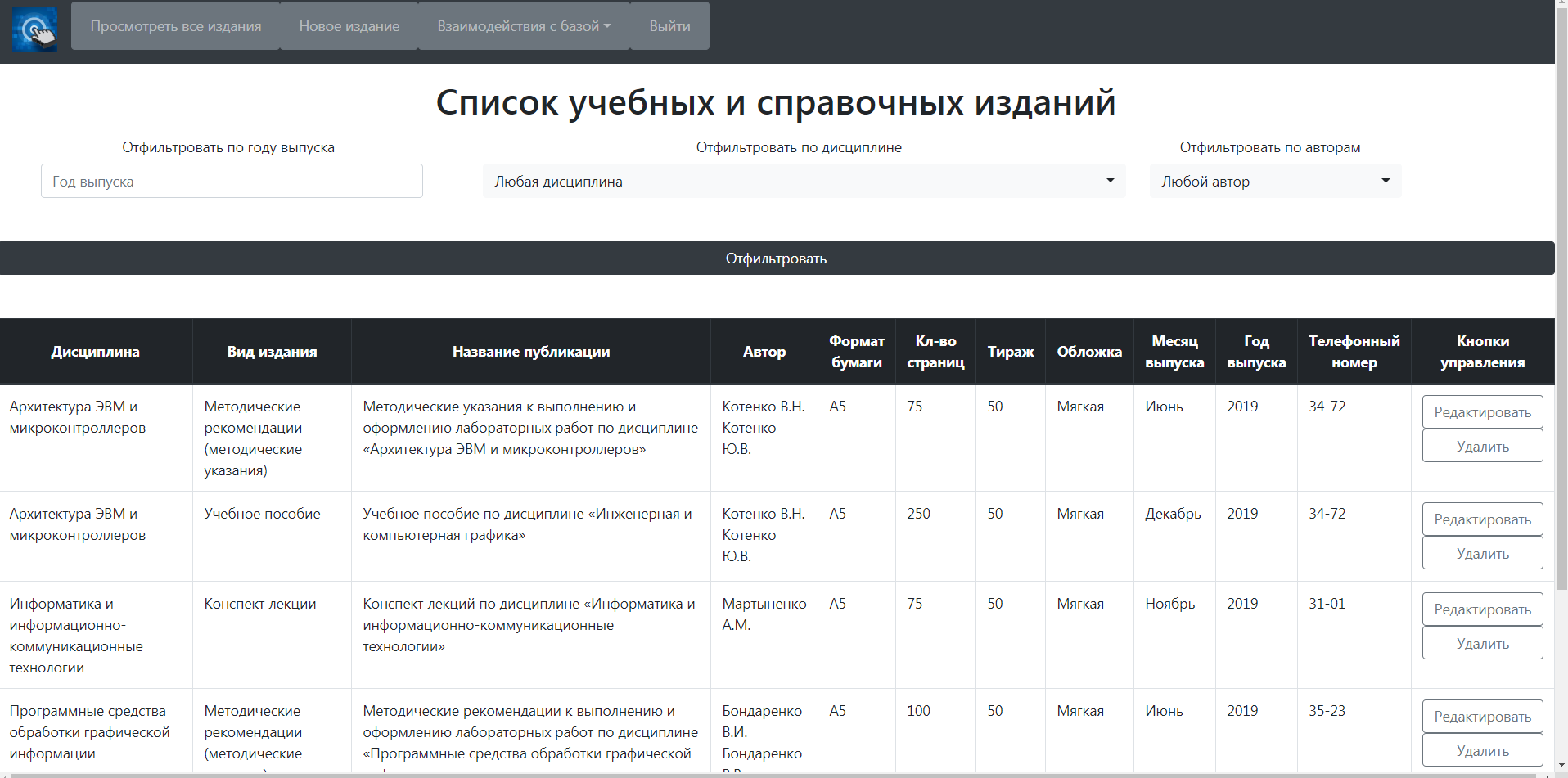


Рисунок А.2 – Страница с формой для добавления нового материала преподавателя

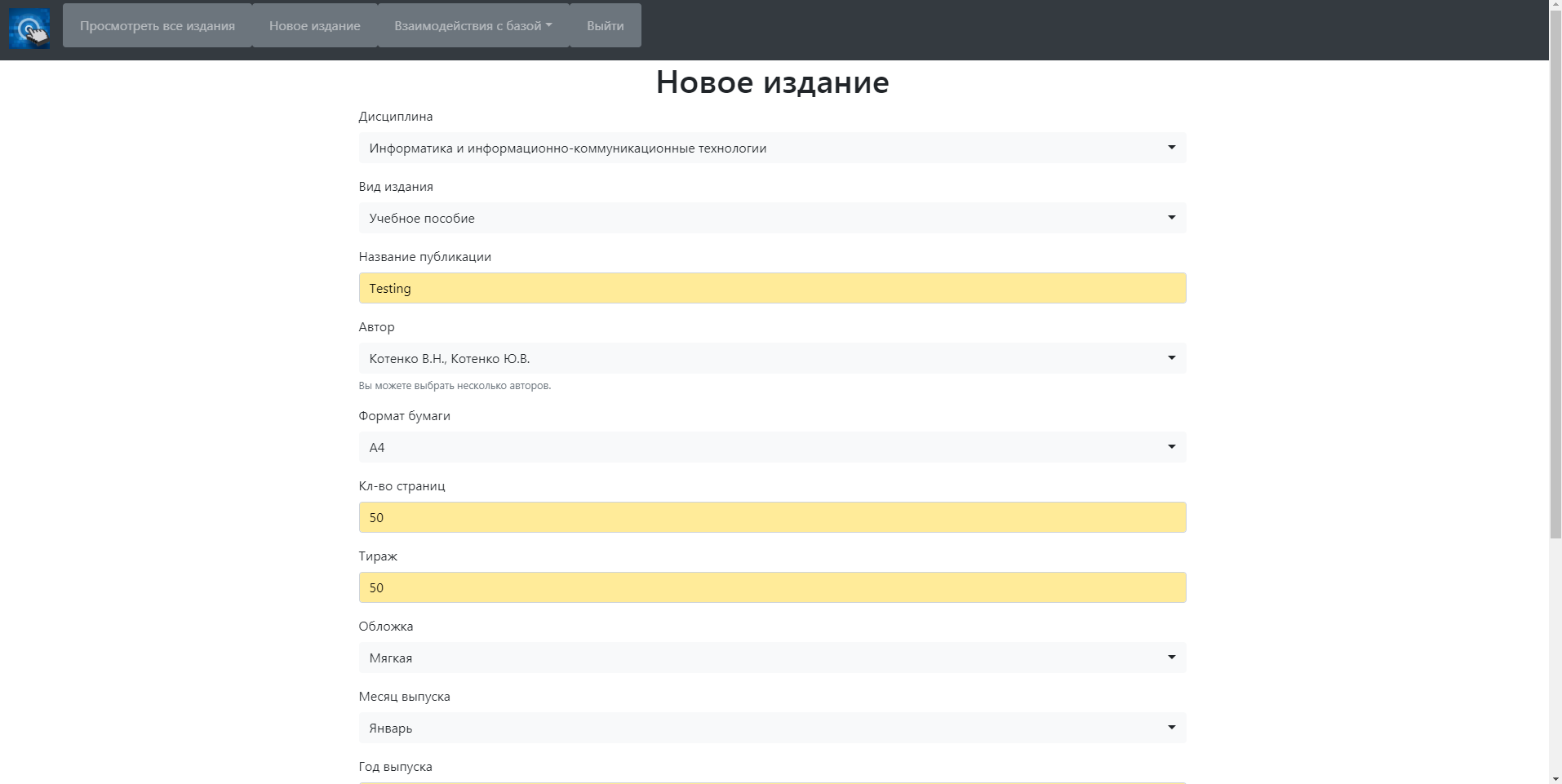


Рисунок А.3 – Страница с формой для добавления нового материала преподавателя

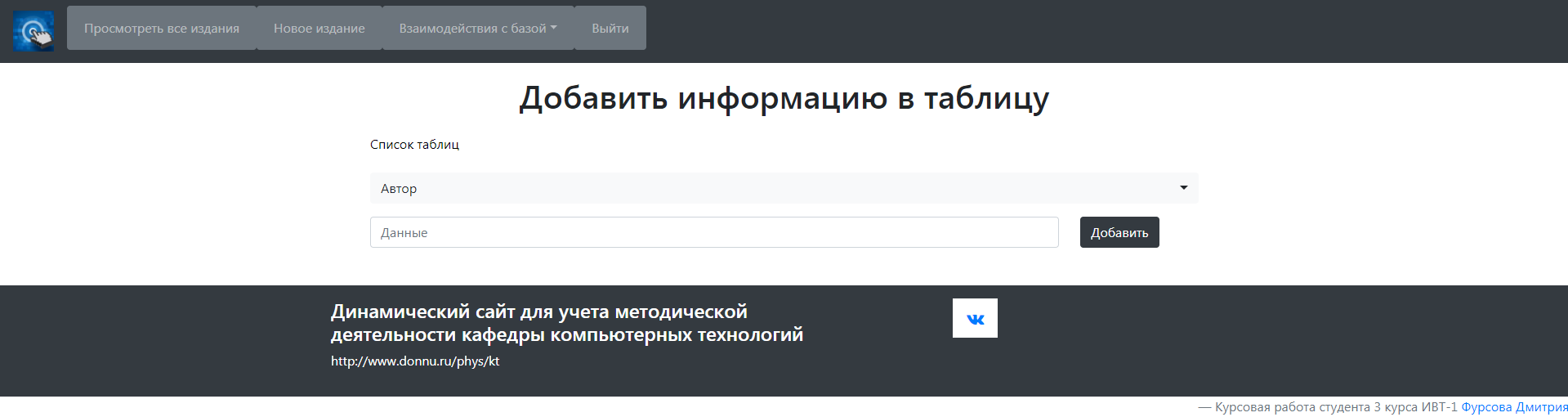


Рисунок А.4 – Страница с формой для добавления нового материала преподавателя

# ПРИЛОЖЕНИЕ Б

**Фрагменты листинга**

Листинг Б.1 **–** файл маршрутов web.php

<?php

Route::get('/', function () {

return redirect('/plans');

});

Auth::routes();

Route::resource('plans', 'PublicationPlanController');

Route::get('add-to-base', 'AdditionsToBaseController@create');

Route::post('add-to-base', 'AdditionsToBaseController@store');

Route::get('select-table-for-remove-from-base', 'AdditionsToBaseController@remove');

Route::post('select-table-for-remove-from-base', 'AdditionsToBaseController@select\_table\_remove');

Route::post('remove-from-base', 'AdditionsToBaseController@destroy');

Route::get('select-table-for-update-base', 'AdditionsToBaseController@change');

Route::post('select-table-for-update-base', 'AdditionsToBaseController@select\_table\_for\_update');

Route::post('update-base', 'AdditionsToBaseController@update');

Route::get('logout', 'LogoutController@logout');

Листинг Б.2 **–** PublicationPlanController.php. Редактирование существующего издания

public function edit($id)

{

// get the plan

$plan = Publications::find($id); //selected plan

$papers\_size = PapersSize::all();

$months = MonthOfSubmission::all();

$cover = Covers::all();

$type\_publication = TypeOfPublication::all();

$disciplines = Discipline::all();

$users = User::all();

$plan\_users = DB::table('users\_publications')->where('plan\_id', '=', $id)->get();

$selected\_users = collect();

$unselected\_users = collect();

for ($i = 0; $i < count($users); $i++)

{

for ($j = 0; $j < count($plan\_users); $j++)

{

if($users[$i]->id == $plan\_users[$j]->user\_id)

{

$selected\_users->push($users->slice($i,1));

}

}

}

for ($i = 0; $i < count($users); $i++)

{

for ($j = 0, $selected = false; $j < count($plan\_users); $j++)

{

if($users[$i]->id == $plan\_users[$j]->user\_id)

{

$selected = true;

}

}

if($selected == false)

{

$unselected\_users->push($users->slice($i,1));

}

}

return view('plans.edit',[

'plan' => $plan,

'papers\_size' => $papers\_size,

'months' => $months,

'cover' => $cover,

'type\_publication' => $type\_publication,

'disciplines' => $disciplines,

'users' => $users,

'plan\_users' => $plan\_users,

'selected\_users' => $selected\_users,

'unselected\_users' => $unselected\_users,

]);

}

Листинг Б.3 **–** AdditionsToBaseController.php. Удаление данных о кафедре

public function remove()

{

return view("select-table-for-remove-from-base");

}

public function select\_table\_remove(Request $request)

{

$select\_table = $request->input('table');

$collection = array();

switch ($select\_table) {

case 'discipline':

$table = Discipline::all();

$number\_table = 1;

foreach ($table as $value) {

$collection[$value->id] = $value->name\_of\_discipline;

}

break;

case 'type\_publication':

$table = TypeOfPublication::all();

$number\_table = 2;

foreach ($table as $value) {

$collection[$value->id] = $value->type\_publication\_name;

}

break;

case 'name':

$table = User::all();

$number\_table = 3;

foreach ($table as $value) {

$collection[$value->id] = $value->name;

}

break;

default:

Session::flash('message', 'Что-то пошло не так. Таблицы не существует!');

return redirect('/select-table-for-remove-from-base');

break;

}

return view("remove-from-base")->with([

'values' => $collection,

'select\_table' => $number\_table,

]);

}

public function destroy(Request $request)

{

$elements = $request->input('elements.\*');

if (count($elements) > 0) {

for ($i = 0; $i < count($elements); $i++) {

$element = substr($elements[$i], 0, 1);

switch ($element) {

case 1:

$id = substr($elements[$i], 1);

$plans = Publications::all()->where('discipline\_id', '=', $id);

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

$deletedRows = Users\_Publications::where('plan\_id', '=', $value->id)->delete();

$value->delete();

}

$discipline = Discipline::find($id);

$name\_element = $discipline->name\_of\_discipline;

$discipline->delete();

break;

case 2:

$id = substr($elements[$i], 1);

$plans = Publications::all()->where('type\_publication\_id', '=', $id);

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

$deletedRows = Users\_Publications::where('plan\_id', '=', $value->id)->delete();

$value->delete();

}

$type = TypeOfPublication::find($id);

$name\_element = $type->type\_publication\_name;

$type->delete();

break;

case 3:

$id = substr($elements[$i], 1);

$user\_publications = Users\_Publications::all()->where('user\_id', '=', $id);

$deletedRows = Users\_Publications::where('user\_id', '=', $id)->delete();

foreach ($user\_publications as $key => $value) {

$count\_publication = Users\_Publications::all()->where('plan\_id', '=', $value->plan\_id)->count();

if ($count\_publication == 0) {

$deletedRows = Publications::where('id', '=', $value->plan\_id)->delete();

}

}

$autor = User::find($id);

$name\_element = $autor->name;

$autor->delete();

break;

}

}

Session::flash('message', 'Данные об ' . $name\_element . ' и все связанные с ним(-и) данные были удалены!');

return redirect('/select-table-for-remove-from-base');

}

}