МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по научно-исследовательской работе

Тема: Совершенствование системы программной поддержки учебного процесса на кафедре ФВиС на основе анализа результатов опытной эксплуатации и создание подсистемы хранения архивной информации.

Студент(ка) гр. 3303 _____ Крень А.В.

Руководители _____ Фомичева Т.Г.

_____ Кринкин К.В.

Санкт-Петербург
2018

ЗАДАНИЕ

НА НАУЧНО-ИССЛЕДОВАТЕЛЬСКУЮ РАБОТУ

Студент(ка) Крень А.В.

Группа 3303

Тема НИР: Совершенствование системы программной поддержки учебного процесса на кафедре ФВиС на основе анализа результатов опытной эксплуатации и создание подсистемы хранения архивной информации.

Задание на НИР:

1) Провести исследование предметной области;
2) Описать предполагаемые способы решения;
3) Создать прототип интерфейса приложения.

Сроки выполнения НИР: 13.09.2018 – 20.12.2018

Дата сдачи отчета: 20.12.2018

Дата защиты отчета: 20.12.2018

Студент(ка)	_____	Крень А.В.
Руководители	_____	Фомичева Т.Г.
	_____	Кринкин К.В.

АННОТАЦИЯ

Описывается автоматизированная информационная система или АИС кафедры Физического воспитания и спорта, которая представляет собой клиент-серверное приложение, разработанное на базе системы управления контентом (ContentManagementSystem или CMS) WordPress. Данная система предназначена для трех категорий пользователей: студент, преподаватель и администратор системы. Описывается результат эксплуатации программной системы организации учебного процесса на кафедре физического воспитания и спорта. Описывается предполагаемый способ решения, выбор средств разработки. Представлена архитектура приложения, а также описана реализация архивных баз данных.

SUMMARY

Describes an automated information system or AIS of the Department of Physical Education and Sport, which is a client-server application developed on the basis of the content management system (ContentManagementSystem or CMS) WordPress. This system is intended for three categories of users: student, teacher and system administrator. Describes the result of the operation of the software system of the organization of the educational process at the department of physical education and sport. Describes the intended solution method, the choice of development tools. The application architecture is presented, and the implementation of archive databases is described.

Оглавление

Введение 5
1. Обзор предметной области 6
1.1.Описание АИС ФВИС 6
1.2. Требования к системе работы с архивной информацией 11
1.3. Результат эксплуатации программной системы организации учебного процесса на кафедре физического воспитания и спорта 13
2. Описание предполагаемого решения 17
2.1. Выбор средств разработки 17
2.1.1. Knowage 17
2.1.2. PentahoBI + Saiku 18
2.1.3. Cubes + CubesViewer 18
2.1.4. Критерии сравнения 19
2.1.5. Результаты сравнения 20
2.2. Описание выбранных средств разработки 21
2.2.1. Cubes 21
2.2.2. CubesViewer 28
2.3. Архитектура приложения 30
2.4. Описание реализации архивных БД 31
2.5. Описание реализации архивной БД для просмотра по годам 31
3. Прототип интерфейса 32
Заключение 47
Список использованных источников 48

ВВЕДЕНИЕ

1. Проведение обзора предметной области;
2. Описание предполагаемого решения;
3. Реализация прототипа интерфейса приложения;
4. Изучение реализации архивных БД.

1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание АИС ФВИС

Автоматизированная информационная система или АИС кафедры Физического воспитания и спорта представляет собой клиент-серверное приложение, разработанное на базе системы управления контентом (ContentManagementSystem или CMS) WordPress.

Разработанная система предназначена в первую очередь, для администрации кафедры ФВиС, а также для для преподавателей и студентов 1 и 2 курса, обучающихся на кафедре ФВиС, то есть можно выделить 3 категории пользователей. А также у системы будет кафедральный администратор.

Администратору кафедры доступны следующие возможности:

- Редактирование расписания занятий по дисциплинам, обеспечиваемым кафедрой: заполнение, изменение, удаление;
- Работа со списками студентов: просмотр списков студентов по дисциплинам, просмотр общего списка студентов по факультетам, курсу, учебной группе, удаление студентов с дисциплины, перевод на другую дисциплину, редактирование данных студентов;
- Редактирование состава кафедры: изменение состава руководства кафедры, добавление преподавателей, удаление преподавателей, изменение данных о преподавателях, изменение дисциплины, которую обеспечивает преподаватель;
- Редактирование журналов посещаемости и сдачи нормативов студентами: изменение результатов сдачи нормативов, изменение результатов посещения студентами дисциплины;

- Редактирование состава дисциплин: добавление новой дисциплины, удаление и модификация дисциплины;
- Печать ведомостей, журналов и отчётов по дисциплинам, по учебным группам, сохранение в формате CSV (журнала посещаемости, журнала сдачи нормативов, итоговых оценок по дисциплине), печать шаблона журнала для использования на занятии, печать ведомостей по учебным группам деканатов факультетов. Сохранения этих документов в текстовом формате;
- Редактирование новостей кафедры: добавление, удаление корректировка;
- Установление контрольных точек: предоставления справки о прохождении медицинского обследования, промежуточных аттестаций по спортивным дисциплинам, выставления зачетов за семестр.

Диаграмма прецедентов для администратора системы изображена на рис. 1.1.

Рисунок 1.1 — Диаграмма прецедентов для администратора систем

Преподаватели кафедры имеют возможность:

- Просмотра расписания занятий;
- Просмотра списка преподавателей кафедры;
- Ведения журналов успеваемости и сдачи нормативов студентами: добавление записей о посещении студентом занятия, о сдаче им норматива, заполнение дат проведенных занятий в журнале;

- Выставления зачетов студентам в конце каждого семестра и баллов промежуточной аттестации;
- Добавления отметки о наличии у студента медицинской справки;
- Просмотр списков студентов;
- Просмотра списков студентов по дисциплинам, по факультетам, по учебным группам;
- Печати ведомостей, журналов и отчётов по дисциплинам, по учебным группам, сохранение в формате CSV (журнала посещаемости, журнала сдачи нормативов, итоговых оценок по дисциплине), печати шаблона журнала для использования на занятии, печати ведомостей по учебным группам для деканатов факультетов. Сохранения этих документов в текстовом формате;
- Редактирования нормативов: добавления новых нормативов, удаление и изменение нормативов;
- Просмотра новостей кафедры.

Диаграмма прецедентов для преподавателя изображена на рис. 1.2.

Рисунок 1.2 - Диаграмма прецедентов для преподавателя

Студент имеет возможность:

- Просмотра расписания занятий;

- Просмотра списка своей группы по дисциплине;
- Просмотра списка преподавателей кафедры;
- Просмотра информации о своей успеваемости;
- Выбора дисциплины;
- Просмотра новостей кафедры.

Диаграмма прецедентов для студента изображена на рис. 1.3.

Рисунок 1.3 – Диаграмма прецедентов для студента

Для хранения данных используется реляционная БД, в качестве СУБД используется MySQL. Архитектура приложения приведена на рисунке 1.4.

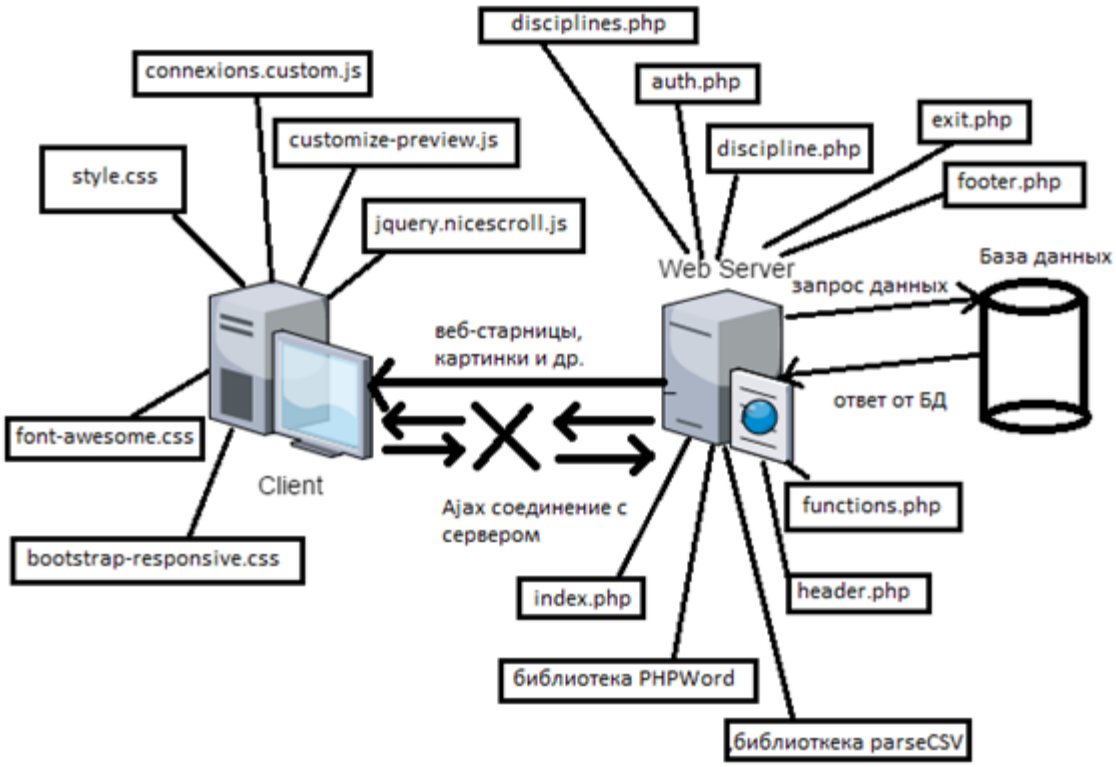


Рисунок 1.4 – Архитектура АИС ФВиС

1.2. Требования к системе работы с архивной информацией

Разрабатываемая система работы с архивной информацией тесно связана с АИС кафедры ФВиС, поэтому система не должна быть самостоятельной, т. е. должна быть частью АИС.

Пользоваться системой будет только администратор системы, а для других пользователей доступ к системе должен быть запрещен.

При анализе хранимых данных разработанной системы для кафедры ФВиС были выявлены данные, для которых необходимо обеспечить просмотр по годам:

- список групп и студентов групп;
- список нормативов по дисциплинам;
- список преподавателей;
- архивные документы: рабочие программы дисциплин и индивидуальные планы преподавателей.

- Также необходимо иметь возможность анализировать следующие данные по параметрам и получать обобщенные характеристики:

- Для дисциплин, необходимо знать количество часов, выделенных на сдачу нормативов, на практические занятия, самостоятельную работу, количество зачетных единиц, общее количество часов, предусмотрены ли курсовая работа или курсовой проект, а также какие формы итогового контроля предусмотрены: экзамен, зачет. Для этих данных должна быть предоставлена возможность анализа по годам, направлениям, учебным планам и дисциплинам.
- Для неаудиторной нагрузки преподавателя необходимо знать количество часов, выделенных для данного вида нагрузки этому преподавателю. Для

этих данных должна быть предоставлена возможность анализа по годам, преподавателям и видам неаудиторной нагрузки.

- Также необходимо знать количество студентов, которые были освобождены по состоянию здоровья (эти данные необходимы для центров, которые собирают общую статистику по учебным заведениям).

Так как объем хранимых данных сравнительно небольшой, то по быстродействию системы требований нет.

1.3. Результат эксплуатации программной системы организации учебного процесса на кафедре физического воспитания и спорта

В результате эксплуатации программной системы организации учебного процесса на кафедре физического воспитания и спорта было решено переписать данную систему с CMS Wordpress на Framework Laravel. Для Front части было решен использовать Vue.js.

Причины, из-за которых было решено переписать данную систему:

1. Упростить структуру БД;
2. Увеличить производительность проекта;
3. Использование миграций;
4. Избавление от жестко прописанного домена;
5. Улучшение структуры кода;
6. Добавление простоты тестирования проекта;
7. Увеличение скорости работы системы;
8. Использование более современных технологий.

- В результате сравнения CMS Wordpress и Framework Laravel была получена следующая таблица 1.1:

Различия	CMS WordPress	Framework Laravel
1. Структура БД	Больше таблиц за счет внутренних структур	Проще для восприятия
2. Производительность БД	Не позволяет проектировать БД. От гибкости БД напрямую зависит ее производительность.	С нуля происходит проектирование базы и появляется возможность распределить данные и их связи самым оптимальным способом, можно создавать индексы, триггеры и тд.
3. ORM	ORM нет	Есть решение Eloquent, реализующее ORM по принципу Active Record.
4. Миграции	Миграций нет	Миграции реализованы стандартно
5. Хардкод домена	В БД во многих местах жестко прописан домен. При переносе базы нужно всегда делать поиск с заменой.	Нет
6. Структура кода	Вся логика лежит либо в functions.php, либо прямо в шаблонах страниц	Заранее установлена логика MVC, основные принципы ООП и кое-где логика авторов фреймворка

7. Роутинг	Самый часто используемый способ – создать новый url – создать страницу, а анна нее навесить шаблон, в котором разместить всю логику.	Все маршруты и url для веб-интерфейса описаны в файле routes/web.php
8. Внутренние зависимости	Нет никаких механизмов для разрешения зависимостей	Мощный механизм разрешения зависимостей и гибкое управление ими
9. Пакетный менеджер (внешние зависимости)	Можно установить composer, используется менеджер плагинов	Используется Composer
10.События	Есть Hookies	Есть систем объектных событий и listeners
11.Административная панель	Понятна для использования	Существует платная версия (100$)
12.Автоматические тесты	Существуют, но используются только разработчиками плагинов	Отличная поддержка автоматических тестов PHPUnit на различных уровнях
13.Очереди	Нет очередей. Реализуются руками	Единственный php-фреймворк, в котором очереди для Redis реализованы полностью из коробки.

14.Обработка ошибок	Есть отдельный класс WP_error, который нельзя обработать как Exception	Используются стандартные для PHP реализации класс Exception (исключение)
15.КЭШ	Необходимо устанавливать дополнительные плагины для ускорения	Имеет встроенные механизмы кэширования, из короботы работающие с Redis и Memcached
16.Планировщики	Использует wp_cron	Использует обычный серверный Cron

2. ОПИСАНИЕ ПРЕДПОЛАГАЕМОГО РЕШЕНИЯ

2.1. Выбор средств разработки

Для реализации многомерных кубов данных, помимо создания БД с особой архитектурой, необходимо использование OLAP-сервера. В серверных OLAP-средствах вычисления и хранение агрегатных данных выполняются на сервере, в то время как клиентское приложение получает только результаты запросов к многомерным кубам, которые хранятся на сервере, например, для дальнейшей их визуализации. Некоторые OLAP-серверы поддерживают хранение данных только в реляционных базах, некоторые - только в многомерных. Многие современные OLAP-серверы поддерживают все три способа хранения данных: MOLAP, ROLAP и HOLAP.

В связи с тем, что кафедра ФВиС является сторонником открытого ПО, необходимо полностью бесплатное и открытое решение, поэтому далее рассматриваться будут только open-source решения.

2.1.1. Knowage

Knowage - это пакет для бизнес-аналитики с открытым исходным кодом, который объединяет традиционные данные и bigdata источники в ценную и содержательную информацию. Помимо коммерческой лицензии имеется также community версия с полным набором аналитических возможностей.

Knowage представляет из себя веб-приложение, написанное на Java (J2EE) с различными правами доступа. Имеется OLAP движок с поддержкой стандартных операций (свертки, детализирования, срезов), возможность строить различные графики с сохранением. Поддержка вычисляемых полей, временных рядов имеется только в коммерческой версии. Помимо работы с многомерными данными предоставляется еще обширный функционал, который не требуется в данной работе и может вызвать сложность для понимания пользователем.

Пакет предоставляется на английском языке, при этом большая часть работы пользователя связана с перемещением по интерфейсу, что также приводит к усложнению понимания. Не имеет поддержки встраивания в другое веб-приложение.

2.1.2. PentahoBI + Saiku

PentahoBISuite (PentahoBusinessIntelligenceSuite) – свободное ПО для бизнес-анализа на программной платформе Java, разрабатываемое компанией Pentaho. В состав Pentaho входит OLAP-сервер Mondrian, который поддерживает все операции OLAP. Pentaho позволяет преобразовать схему Mondrian (описание схемы куба в формате XML) в многомерный куб. Компания Pentaho предоставляет также дополнительные инструменты для преобразования таблицы фактов в схему «звезда» и, далее, в схему Mondrian.

Для работы с построенными кубами предоставляется дополнительный плагин Saiku Analytics. Saiku предоставляет множество инструментов для построения запросов, графиков, таблиц, выполнения OLAP-операций. Имеет английский интерфейс, хотя и довольно понятный в целом (имеется много иконок вместо надписей, что позволяет понять назначение элементов). Поддержку встраивания в другое веб-приложение имеет только коммерческое решение со своим собственным сервером.

2.1.3. Cubes + CubesViewer

Cubes – полностью открытый, легковесный фреймворк для Python и набор инструментов для OLAP, многомерного анализа и просмотра агрегированных данных. Входит в состав DataBrewery – набора Python фреймворков и инструментов для обработки и анализа данных.

Cubes по умолчанию использует технологию ROLAP, поддерживает все основные операции OLAP, включая арифметические операции над мерами и временные ряды. Сервер работает по протоколу HTTP.

В качестве клиента предоставляется веб-инструмент CubesViewer, который может использоваться для выполнения операций над кубами, построения различных графиков и фильтрации данных. CubesViewer имеет также англоязычный интерфейс, хотя большинство действий иллюстрированы иконками.

CubesViewer предоставляет возможность полного встраивания в другие веб-приложения, при этом позволяет разместить на одной странице несколько представлений (графическое представление куба с интерфейсом для выполнения операций), что может быть полезно, если нет желания переключатся между разными экранами/страницами.

2.1.4. Критерии сравнения

Для того, чтобы выбрать подходящие инструменты, были выделены следующие критерии сравнения:

Поддержка ROLAP. В связи с выбранным способом реализации многомерных кубов необходимо, чтобы OLAP-сервер поддерживал возможность работать с данными, хранящимися в реляционном виде;

Наличие удобного пользовательского интерфейса. У пользователя не должно возникать затруднений при работе с многомерными данными, при этом выполнение операций не должно состоять из большого количества действий;

Поддержка OLAP-операций. OLAP-сервер должен поддерживать следующие операции: свертка/детализация, вращение, срезы по измерениям,

поддержка иерархий, арифметические операции над мерами, фильтрация данных.

Возможность встраивания в АИС кафедры ФВиС. Одним из требований к системе работы с архивной информации является именно возможность встраивания, т.е. система не должна быть отдельным приложением.

Простота использования в разработке. Оценивается простота установки и настройки сервера, простота формирования моделей кубов из реляционных данных и простота настройки пользовательского интерфейса.

2.1.5. Результаты сравнения

Результаты сравнения по выделенным в разделе 1.4 критериям приведены в таблице 1.1. Использовалась градация оценок от 0 (полное отсутствие выполнения критерия) до 5 (критерий полностью выполняется).

Таблица 2.1 – сравнение средств разработки

Критерий	Knowage	Pentaho + Saiku	Cubes + CubesViewer
Поддержка ROLAP	5	5	5
Наличие удобного пользовательского интерфейса	3	5	5
Поддержка OLAP-операций	4	5	5
Возможность встраивания в АИС кафедры	0	0	5
Простота использования в разработке	3	4	5
Итого	15	19	25

По результатам анализа был выбран союз Cubes + CubesViewer, т.к. по сумме критериев он набрал максимальную оценку.

2.2. Описание выбранных средств разработки

2.2.1. Cubes

Цель данного фреймворка – предоставить аналитику или любому другому конечному пользователю понятный и естественный способ отчетности, используя концепцию кубов – многомерных объектов данных.

Cubes обладает следующими особенностями:

- логическое представление данных – как видят данные аналитики, а не как они хранятся физически;
- OLAP и агрегированный просмотр (используется ROLAP);
- иерархические измерения (атрибуты, которые имеют иерархические зависимости, такие как категория-подкатегория, страна-регион);
- наличие множества иерархий в измерении;
- локализуемые метаданные и данные;
- аутентификация и авторизация кубов и их данных;
- подключаемые хранилища данных – подключение других, похожих на кубы (многомерных) источников данных.

Для реализации поставленной задачи будут использоваться все перечисленные свойства, за исключением аутентификации.

Cubes разделяют на следующие части:

- Workspace (рабочее пространство) – аналитическое рабочее пространство, которое управляет кубами, публичными измерениями, хранилищами данных (stores), провайдерами модели (modelproviders) и метаданными модели, а также предоставляет обозреватели агрегаций и обслуживает соединения с базами данных. Хранилище данных представляет собой

источник данных, например, таблицу в БД. Провайдер модели создает модели кубов, измерения и другие аналитические объекты;

- Model (модель) – представляет собой описание данных (метаданные): кубы, измерения, иерархии, атрибуты, метки, локализацию;
- Browser (обозреватель) – просмотр агрегаций, slicing-and-dicing, drill-down;
- Backend (серверная часть) – реализация агрегаций и вспомогательные функции. Cubes использует встроенную ROLAP серверную часть, которая использует БД SQL с помощью SQLAlchemy (набор SQL и ORM (Object-RelationalMapping, объектно-реляционное отображение) инструментов для Python);
- Server (сервер) - WSGI HTTP сервер для Cubes. WSGI (WebServerGatewayInterface) – стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и веб-сервером. Сервер предоставляет jsonAPI для большей части функционала фреймворка;
- Formatters (форматеры) – форматеры данных;
- slicer – CommandLineTool (инструмент командной строки) – облегчает управление сервером посредством ввода команд (например, таких команд, как запуск сервера, валидация моделей).

Чтобы Cubes мог работать с кубами, необходимо описать их модель. Логическая модель описывает данные с точки зрения анализа пользователя: как данные будут измерены, как агрегированы и как будут выводиться в отчет. Модель не зависит от физической реализации данных, что позволяет проще сфокусироваться на сути самих данных.

Логическая модель позволяет пользователям:

- видеть данные с точки зрения бизнеса;
- скрыть физическую структуру данных;

- указать иерархии атрибутов;
- предоставить более описательные названия атрибутов для отображения в приложении или отчетах;
- прозрачную локализацию метаданных и данных.

Логическая модель описывается с использованием словаря метаданных моделей. В ней содержится описание логических объектов, физического хранилища и другая дополнительная информация. Состав модели приведен на рисунке 2.1.

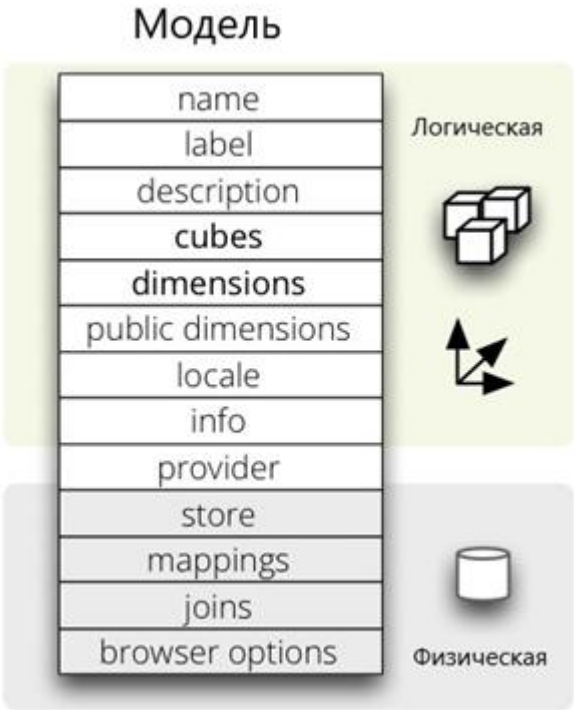


Рис. 2.1. – Описание модели

Логическая часть описания модели состоит из:

- name – имя модели;
- label – удобное для восприятие человеком имя модели (необязательно);
- description – удобное для восприятие человеком описание модели (необязательно);

- locale – локаль, в которой написаны метаданные модели (необязательно, используется для локализируемых моделей);
- cubes – список метаданных кубов (рассматриваются ниже);
- dimensions – список метаданных измерений (рассматриваются ниже).

Физическая часть описания модели состоит из:

- store – имя хранилища данных, где хранятся кубы модели, по умолчанию default;
- mappings – словарь отображений логических наименований в физические, данный словарь наследуется всеми кубами в модели;
- joins – список словарей для спецификации операций JOIN;
- browser_options – опции, передающиеся в обозреватель.

Если в описании модели указаны mappings и/или joins, то они наследуются всеми кубами модели. Mappings объединяются таким образом, что mappings куба заменяются существующими mappings модели с тем же именем. Joins объединяются или сливаются по их имени.

Модель может быть представлена либо как файл в формате JSON, либо как каталог с файлами JSON. Спецификация модели в одном файле – это просто словарь со свойствами модели. Если модель описана в нескольких файлах, то каталог должен иметь следующее содержание:

- model.json – основные метаданные модели (то же, что и однофайловая модель);
- dim_*.json – файл с метаданными измерения (словарь одного измерения);
- cube_*.json – метаданные куба (словарь одного куба).

Описание кубов хранится в словаре с ключом cubes в словаре описания модели или в файлах JSON с префиксом cube_.

Описание куба состоит из:

- базовых элементов:
  - name * – имя куба, уникальный идентификатор;
  - label – удобное для восприятия человеком имя куба (используется для отображения в приложении);
  - description – удобное для восприятия человеком описание куба;
  - info – пользовательская информация, например, форматирование. Не используется фреймворком Cubes;
  - dimensions * – список имен измерений или ссылок на измерения;
  - measures – список мер куба;
  - aggregates – список агрегируемых мер;
  - details – список деталей фактов – атрибуты, которые не относятся к агрегации, но полезны при отображении фактов;
- физических:
  - joins – спецификация физических соединений таблицы;
  - mappings – отображение логических атрибутов в физические;
  - key – имя столбца с ключом в таблице фактов (по умолчанию id);
  - fact – имя таблицы фактов, если не указано, то используется имя куба;
- продвинутых:
  - browser_options – специфичные опции обозревателя;
  - store – имя хранилища, в котором хранится куб.

Поля, отмеченные звездочкой, обязательны.

Меры, которые являются числовыми характеристиками факта, агрегируются в агрегаты меры, пример расчета агрегата путем суммирования представлен на рисунке 2.

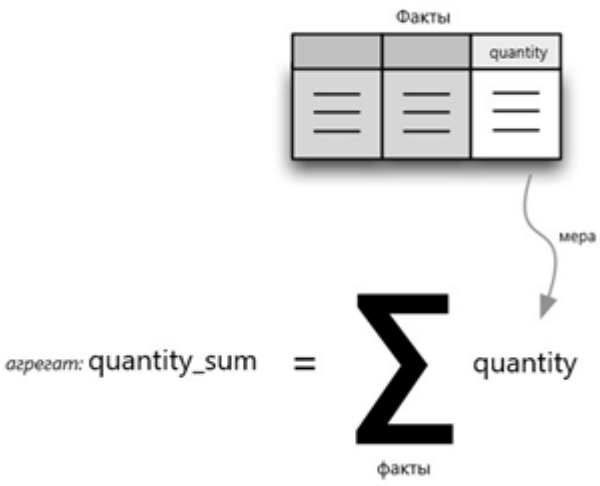


Рисунок 2 – Пример агрегата меры quantity (количество)

Меры описываются с помощью следующих элементов:

- name * – идентификатор меры;
- label – удобное для восприятия человеком имя;
- info – дополнительная пользовательская информация;
- aggregates – список агрегирующих функций, которые будут использоваться для данной меры. Это свойство используется для автоматической генерации агрегатов, однако рекомендуется описывать агрегаты явно;
- window_size – количество элементов в окне для оконных функций, таких как скользящее среднее (по умолчанию 1).

Агрегаты мер описываются с помощью:

- name – идентификатор агрегата (например, quantity_sum);
- label – удобное для восприятия человеком имя;
- measure – мера, для которой производятся вычисления (может быть пустым);

- function – имя агрегирующей функции, которая будет применяться к мере (например, sum, min, max). Функции не обязательно должны применяться к мере, есть функции, для которых не нужно указание меры, например, подсчет количества строк (count);
- window_size – количество элементов в окне для оконных функций;
- info – дополнительная пользовательская информация.

Описание измерений хранится в словаре модели под ключом dimensions. Описание каждого измерения содержит следующие поля:

- базовые:
    - name * – имя измерения (уникальный идентификатор);
    - label – удобное для восприятия человеком имя;
    - description – удобное для восприятия человеком описание измерения;
    - info – пользовательская информация;
    - levels – список описаний уровней;
    - hierarchies – список иерархий измерения;
    - default_hierarchy_name – имя иерархии, используемой по умолчанию;
- продвинутые:
    - cardinality – мощность измерения;
    - role – роль измерения;
    - category – логическая категория;
    - template – имя измерения, которое будет использоваться в качестве шаблона.

При использовании нескольких измерений с одинаковой или похожей структурой Cubes позволяет определить шаблон измерения, который затем может использоваться в качестве основы для других измерений.

Уровни иерархии измерения описываются с использованием следующих полей:

- name * – имя уровня (уникальный идентификатор);
- label – удобное для восприятия человеком имя;
- attributes – список других дополнительных атрибутов, которые относятся к уровню;
- key – ключевое поле уровня (например, имя месяца для уровня «месяц»);
- label_attribute – имя атрибута, содержащего надпись, которая будет отображена;
- order_attribute – имя атрибута, который будет использоваться для сортировки;
- cardinality – символическое приближение количества членов уровня;
- role – роль уровня;
- info – пользовательская информация.

Чтобы описать иерархию измерения используются следующие поля:

- name * – имя иерархии (уникальный идентификатор);
- label – удобное для восприятия человеком имя;
- levels – упорядоченный список имен уровней от наименее детализированного до наиболее детализированного (например, от года до дня).

2.2.2. CubesViewer

CubesViewer – это визуальное, отзывчивое (легко подстраиваемое под разные размеры, «резиновое») HTML5 приложение и библиотека для исследования и визуализации разных типов наборов данных. CubesViewer может использоваться для исследования и аудита данных, генерации отчетов,

проектирования и встраивания диаграмм, а также в качестве простого приложения для аналитики всей компании.

Особенности CubesViewer:

- OLAP обозреватель – возможность исследовать данные по измерениям, фильтровать и применять drill-down, чтобы обнаружить новые факты и тренды.
- Диаграммы – доступен ряд опций визуализации данных, есть возможность переключаться между представлениями просто с помощью клика.
- Ряды – возможность определять и использовать ряды данных для обнаружения и выделения функций.
- Экспорт данных и диаграмм.
- Отзывчивый интерфейс.

Операции отмены, повторения и клонирования – CubesViewer позволяет легко пробовать разные вариации представлений.

Встраивание представлений – представления могут быть легко встроены в другие веб-сайты.

CubesViewer Server – опциональная многопользовательская серверная часть для хранения и совместного просмотра представлений.

2.3. Архитектура приложения

Система АИС ФВиС разрабатывалась с использованием CMS Wordpress . Wordpress написан на языке PHP и использует в качестве хранилища данных реляционную БД. Но после эксплуатации системы было решено переписать данную систему на framevork Laravel. Сравнительная таблица CMS Wordpress и framework Laravel представлена в описании предметной области.

Подсистема для работы с архивной информацией будет разрабатываться в виде модуля archive. Архитектура приложения приведена на рисунке 2.3.

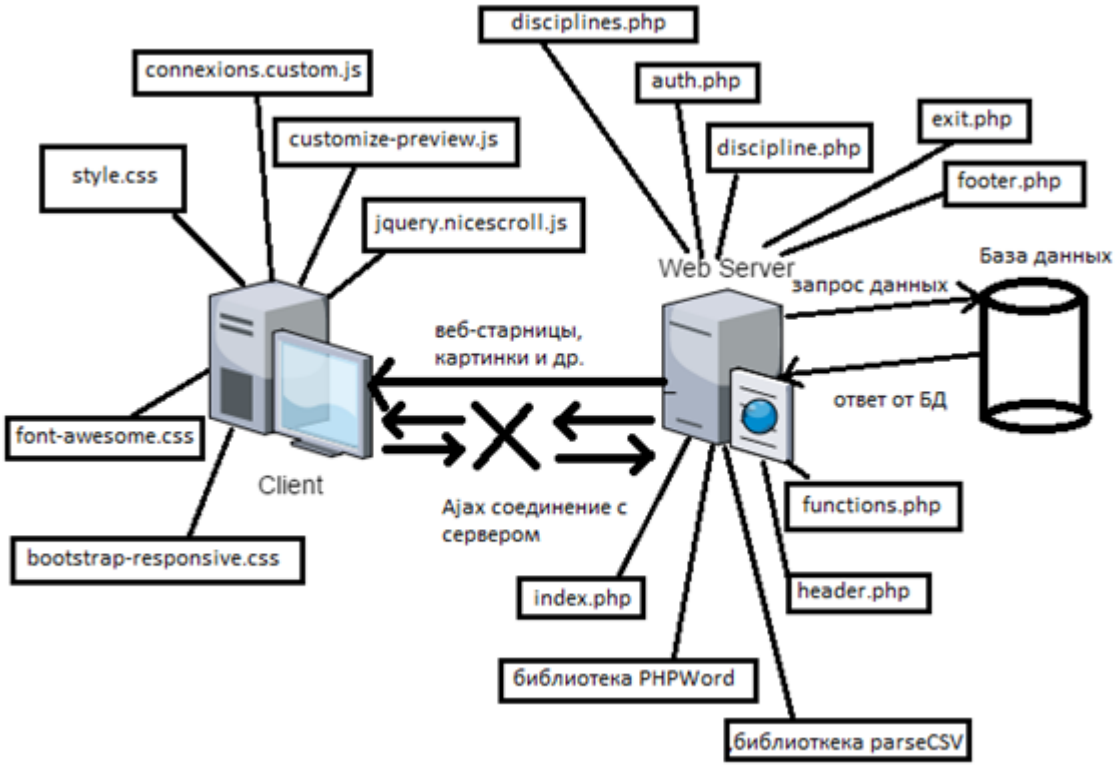


Рисунок 2.3 – Архитектура АИС ФВиС

2.4. Описание реализации архивных БД

При разработке БД будет использоваться инструмент PhpMyAdmin, представляющий из себя веб-приложение и позволяющий осуществлять администрирование сервера БД через браузер, выполнять SQL запросы и просматривать содержимое и структуру таблиц и баз данных.

Две базы данных: fvis_archive – архивная БД для просмотра по годам, и fvisdb_cubes – архивная БД для многомерного анализа. В качестве первичного ключа в таблицах использовались числовые генерируемые СУБД идентификаторы.

Описание реализации архивной БД для просмотра по годам

Для реализации данной БД будет написан скрипт, который будет создавать таблицы системы в исходной БД. Этот скрипт будет модифицирован путем добавления в таблицы полей year (год) и fvis_id (ID в исходной БД). Также удаляться внешние ключи, т.к. в архивную БД новые данные добавляться не будут (только во время сброса данных в архив), также архивные данные не будут удаляться и изменяться, т.е. не будет требоваться сохранения ссылочной целостности. К тому же, при переносе данных в архив с внешними ключами требуются дополнительные трудозатраты: нужно вставлять данные в определенном порядке, а вставлять записи в таблицы, которые зависят друг от друга (например, у студента есть группа, а у группы есть факультет), придется в несколько «заходов».

3. ПРОТОТИП ИНТЕРФЕЙСА

Для того чтобы начать работу с системой необходимо зайти на главную страницу и ввести в форме авторизации логин и пароль, выданные администратором системы.

Форма авторизации в системе показала на рис. 3.1.

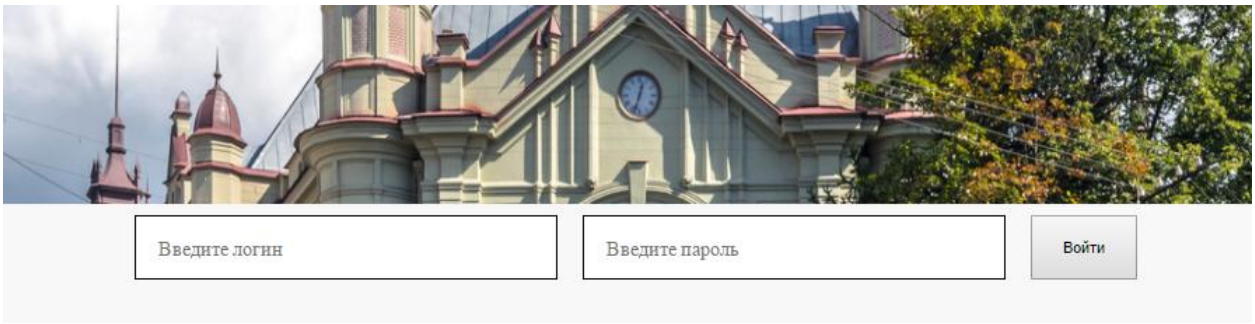


Рисунок 3.1. – Форма авторизации пользователя

Так как администратор обладает всеми возможностями пользователей, то необходимо рассмотреть все действия, которые он может сделать.

Редактирование состава дисциплин: добавление новой дисциплины, удаление и модификация дисциплины;

Редактирование состава кафедры: изменение состава руководства кафедры, добавление преподавателей, удаление преподавателей, изменение данных о преподавателях, изменение дисциплины, которую обеспечивает преподаватель;

Редактирование расписания занятий по дисциплинам, обеспечиваемым кафедрой: заполнение, изменение, удаление;

Работа со списками студентов: удаление студентов с дисциплины, перевод на другую дисциплину, редактирование данных студентов;

Редактирование журналов посещаемости и сдачи нормативов студентами: изменение результатов сдачи нормативов, изменение результатов посещения студентами дисциплины;

Редактирование новостей кафедры: добавление, удаление корректировка;

Установление контрольных точек: предоставления справки о прохождении медицинского обследования, промежуточных аттестаций по спортивным дисциплинам, выставления зачетов за семестр.

Покажем каждое действие более подробно.

Добавление новой дисциплины

Для того, чтобы добавить новую дисциплину необходимо перейти в меню, нажав на кнопку меню (рис. 3.2.), а затем выбрать в меню показанного на рис. В3 и нажать на “Редактирование списка дисциплин”.

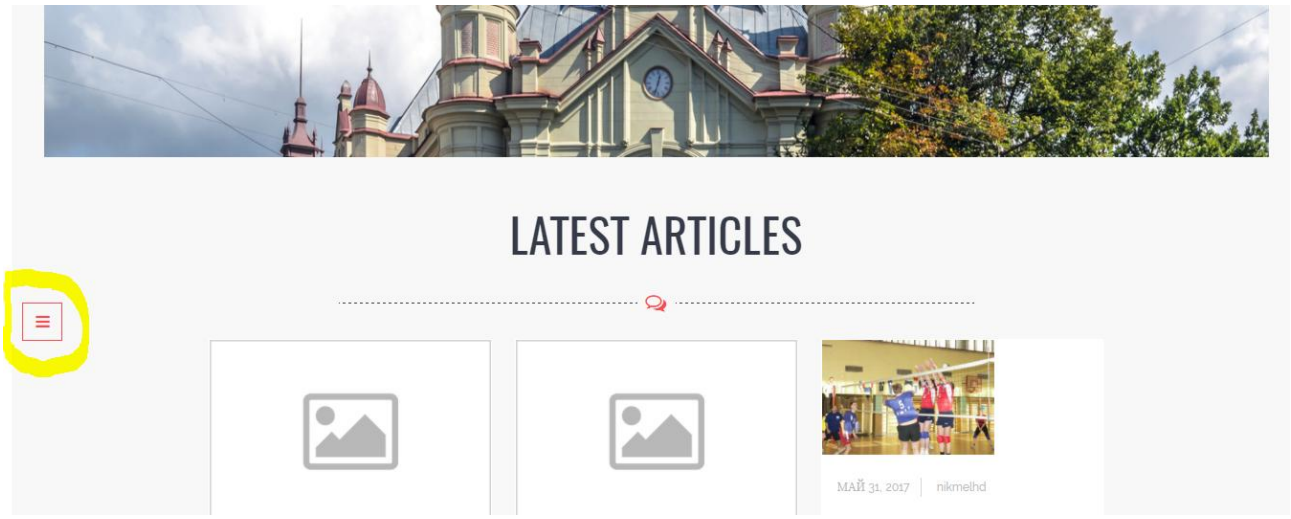


Рисунок 3.2. – Кнопка для показа меню

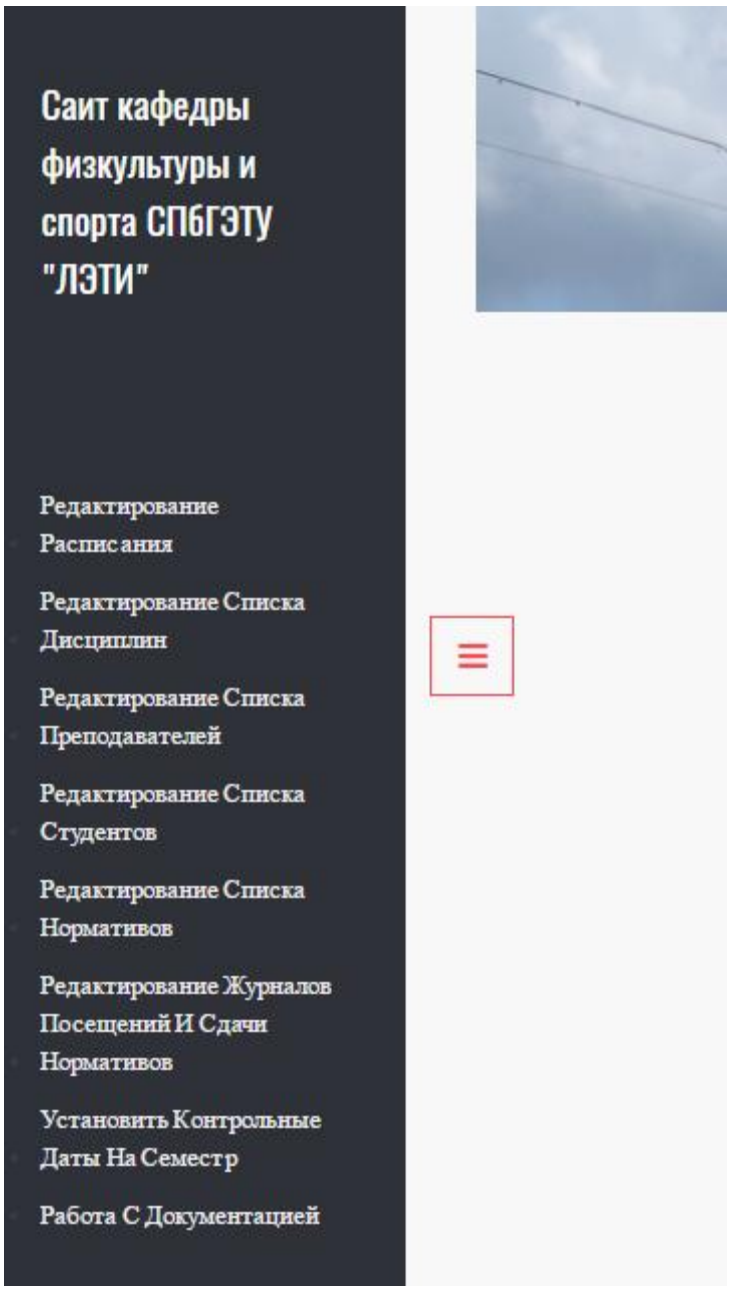


Рисунок 3.3. – Меню администратора системы

После этого откроется новое окно со списком дисциплин на кафедре (см. рис. 3.4.).

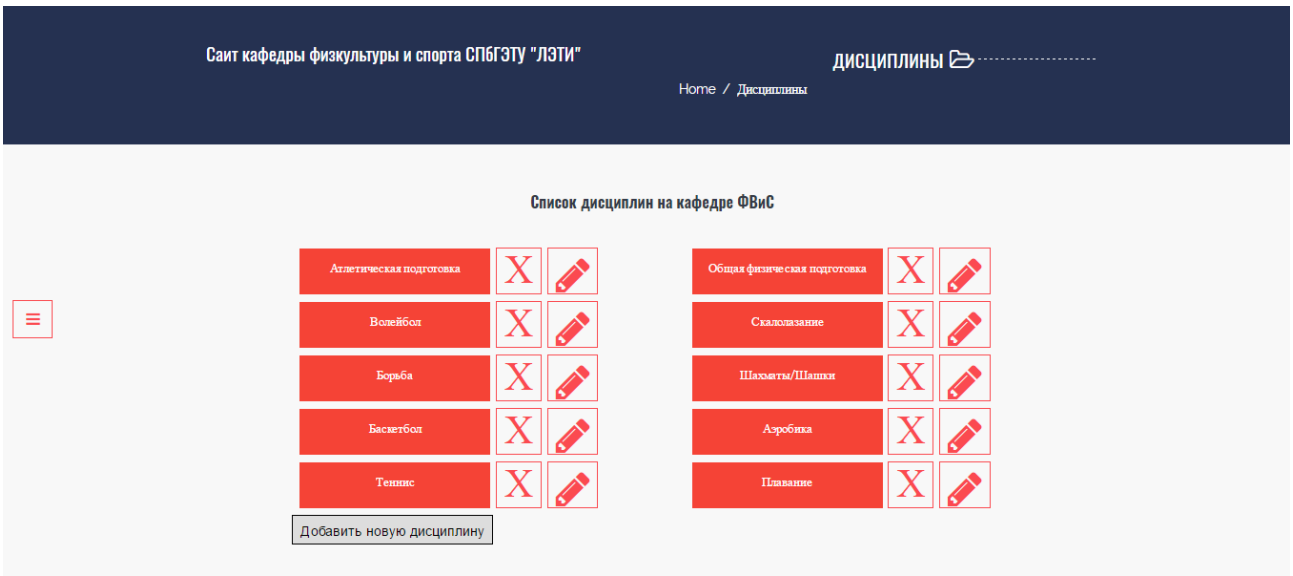


Рисунок 3.4. – Окно редактирования списка дисциплин на кафедре

Для того, чтобы добавить новую дисциплину нужно нажать на кнопку “Добавить новую дисциплину”, после чего откроется форма добавления новой дисциплины на кафедру ФВиС (см. рис. 3.5.).

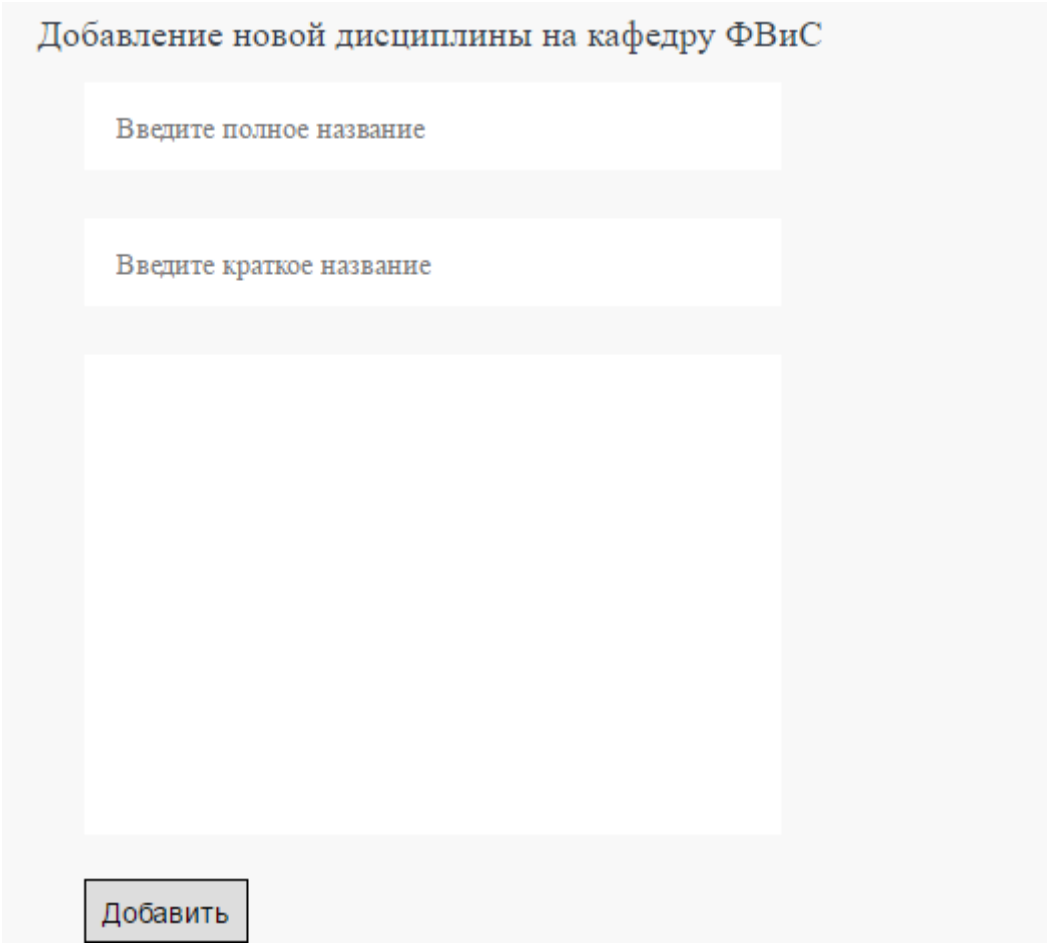


Рисунок 3.5. – Форма добавления новой дисциплины

После того как будут заполнены все поля формы и нажатия кнопки “Добавить” в списке дисциплин появится новая дисциплина.

Удаление дисциплины

Для того, чтобы удалить дисциплину необходимо на странице “Список дисциплин кафедры ФВиС” нажать на кнопку “крестик” возле той дисциплины, которую необходимо удалить. После этого дисциплина удалиться из списка дисциплин кафедры.

Модификация дисциплины

Модификация заключается в том, что пользователь может изменить название дисциплины или её описание. Для того, чтобы изменить дисциплину, необходимо нажать на кнопку редактирования около той дисциплины, которую необходимо изменить, после чего откроется форма изображенная на рис. 3.5.

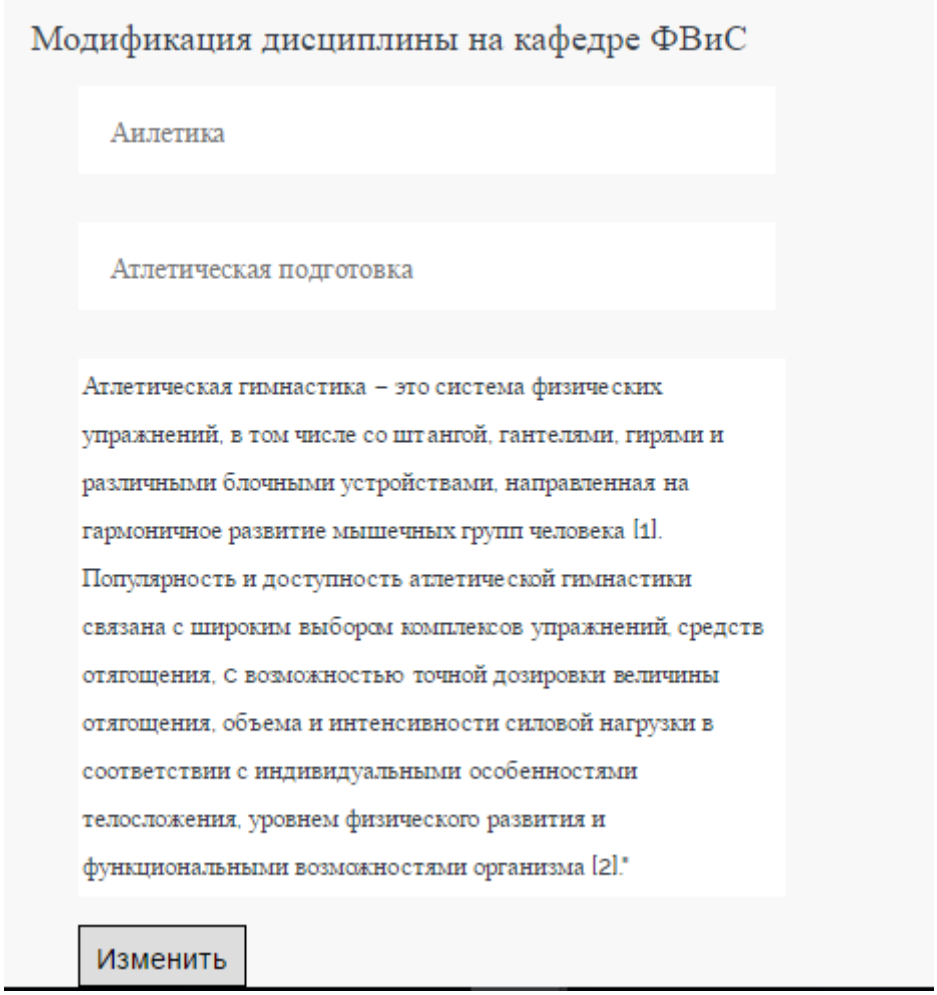
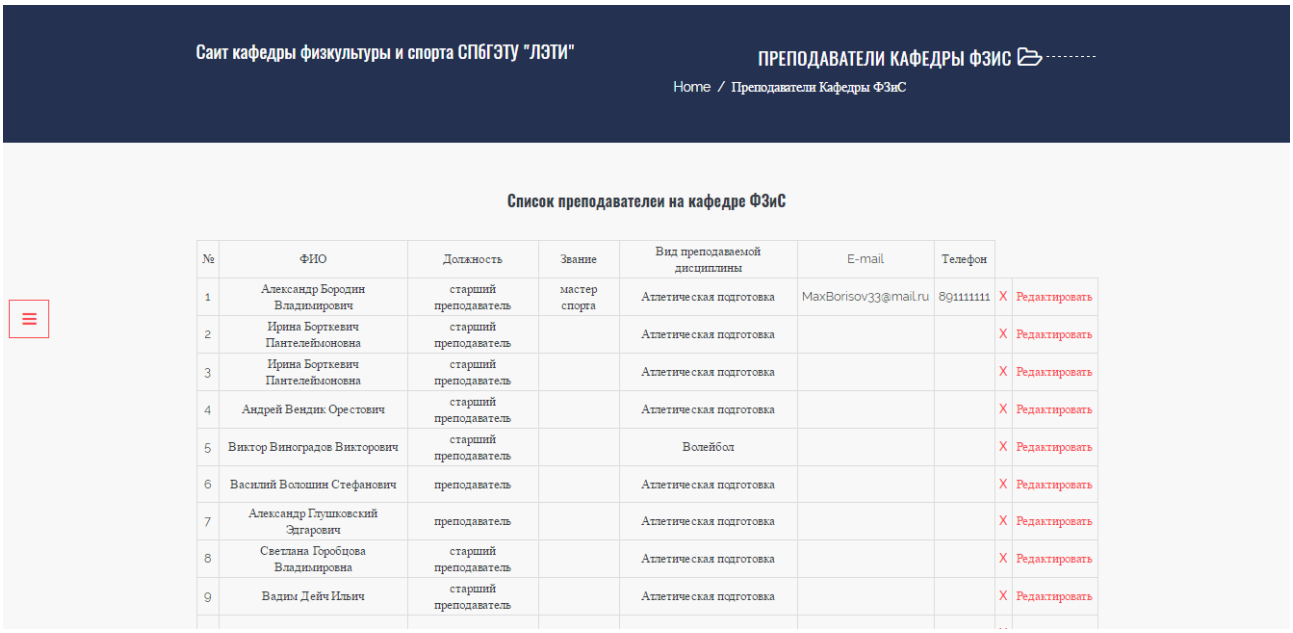


Рисунок 3.5. – Форма модификации дисциплины

После изменения информации в полях, которые необходимо было изменить, пользователю необходимо нажать кнопку “Изменить”.

Добавление преподавателей

Для того, чтобы добавить нового преподавателя необходимо открыть меню (см. рис. 3.3.) и выбрать “Редактирование списка преподавателей”, после чего откроется окно, изображенное на рис. 3.6.



№	ФИО	Должность	Звание	Вид преподаваемой дисциплины	E-mail	Телефон		
1	Александр Бородин Владимирович	старший преподаватель	мастер спорта	Атлетическая подготовка	MaxBorisov33@mail.ru	891111111	X	Редактировать
2	Ирина Борткевич Пантелеймоновна	старший преподаватель		Атлетическая подготовка			X	Редактировать
3	Ирина Борткевич Пантелеймоновна	старший преподаватель		Атлетическая подготовка			X	Редактировать
4	Андрей Вендик Орестович	старший преподаватель		Атлетическая подготовка			X	Редактировать
5	Виктор Виноградов Викторович	старший преподаватель		Волейбол			X	Редактировать
6	Василий Волошин Стефанович	преподаватель		Атлетическая подготовка			X	Редактировать
7	Александр Глушковский Эдгарович	преподаватель		Атлетическая подготовка			X	Редактировать
8	Светлана Горобцова Владимировна	старший преподаватель		Атлетическая подготовка			X	Редактировать
9	Вадим Дейч Ильич	старший преподаватель		Атлетическая подготовка			X	Редактировать

Рисунок 3.6. – Список преподавателей кафедры ФВиС

Для того, чтобы добавить нового преподавателя необходимо нажать “Добавить нового преподавателя внизу страницы” (см.рис. 3.7.).

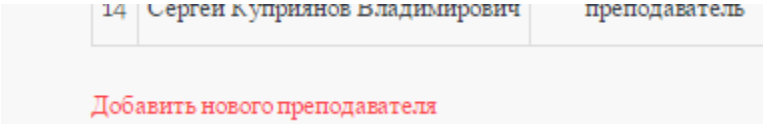


Рисунок 3.7. – Кнопка добавления нового преподавателя

После того, как администратор нажмёт эту кнопку, откроется новое окно с формой добавления нового преподавателя (см. рис. 3.8).

Логин	
Пароль	3YL69Y7G
Фамилия	
Имя	
Отчество	
Должность	
Звание	
Дисциплина	Атлетика ▼
email	
phone	

Добавить

Рисунок 3.8. – Форма добавления нового преподавателя

Заполнив все поля, администратор должен нажать кнопку “Добавить”, чтобы преподаватель появился в системе.

Удаление преподавателя

Для того, чтобы удалить преподавателя необходимо в списке преподавателей кафедры необходимо выбрать того преподавателя, которого нужно удалить и нажать на кнопку “удалить” рядом с этим преподавателем.

Редактирование преподавателя

Для того, чтобы изменить информацию о преподавателе, необходимо нажать на кнопку “Редактировать” рядом с выбранным преподавателем и после чего, откроется новое окно, изображенное на рис. 3.9.

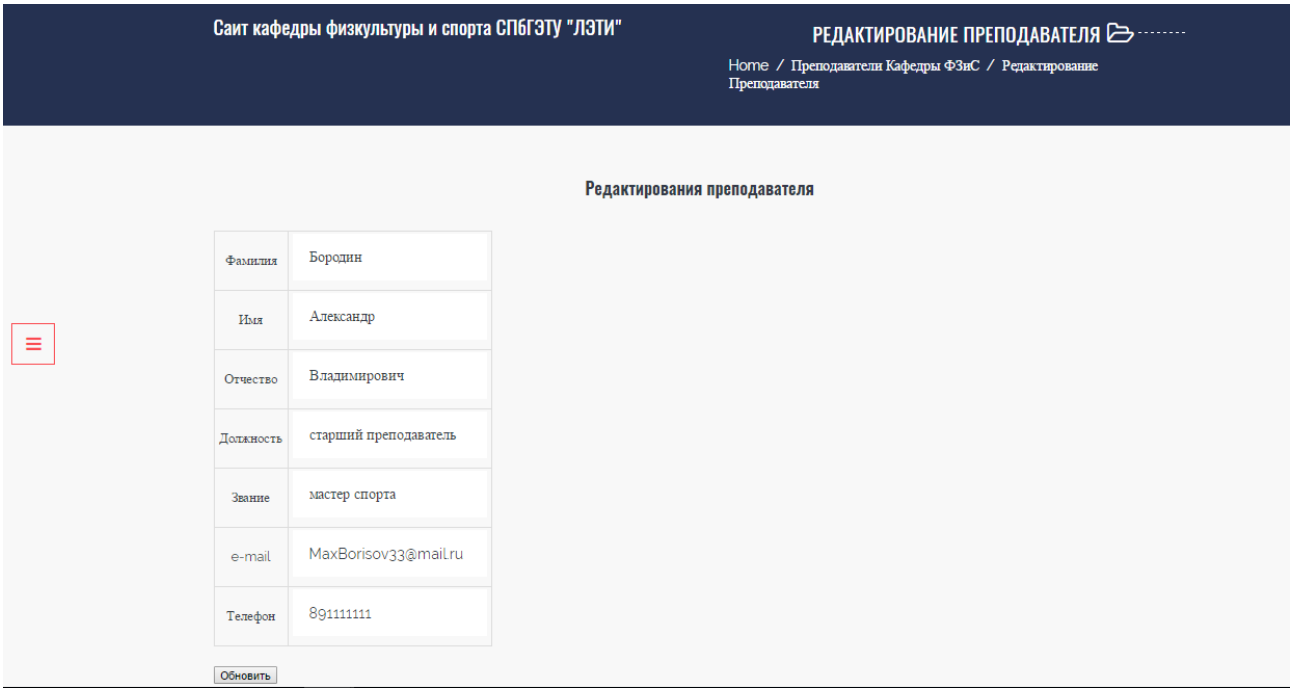


Рисунок 3.9. – Форма редактирования информации о преподавателе

После изменения данных необходимо нажать на кнопку “Обновить”, после чего данные о преподавателе изменяться.

Редактирование расписания занятий

Для того, чтобы редактировать расписание необходимо выбрать в меню (рис. 3.3.) “Редактирование расписания”, после чего откроется окно, изображенное на рис. 3.10.

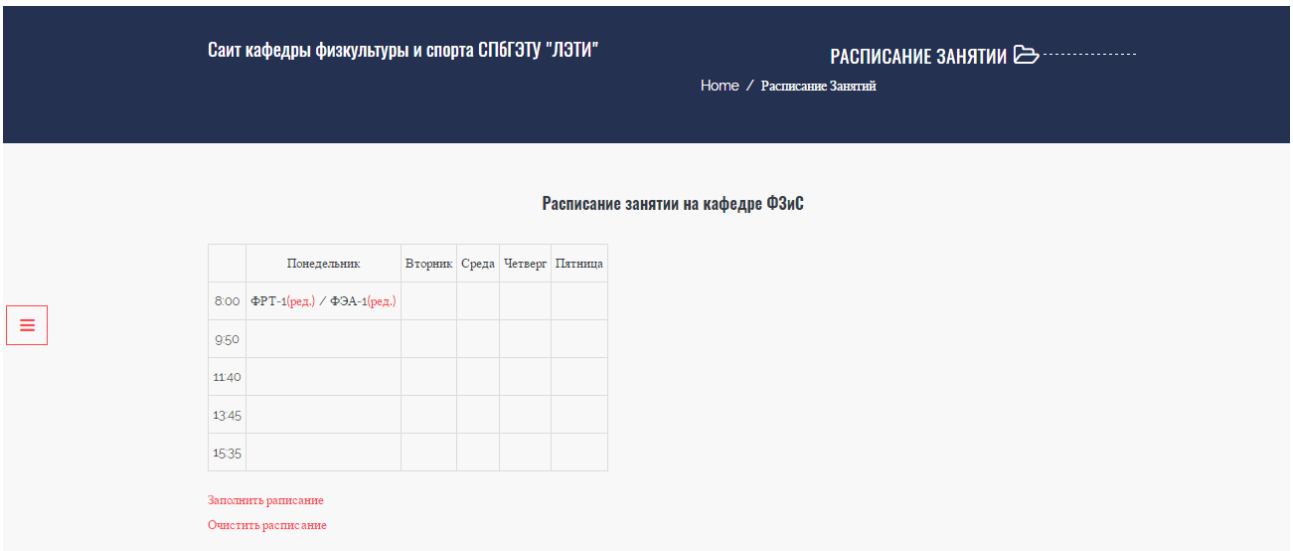


Рисунок 3.10. – Окно расписания занятий

Редактирование заключается в том, что администратор может изменить день недели и время занятия факультета, нажав на кнопку “(ред.)” рядом с той группой, время занятия которой он хочет изменить, после чего появится панель (рис. 3.11.), на которой есть возможность изменить время занятия.

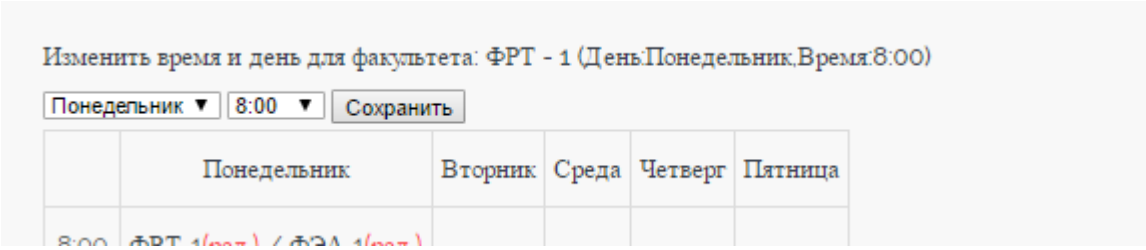


Рисунок 3.11.- Панели для изменения дня недели и времени занятий факультета

После нажатия кнопки “Сохранить”, факультет изменит своё расположение в расписании занятий.

Заполнение расписания

Для того чтобы заполнить расписание нужно нажать кнопку, расположенную под расписанием “Заполнить расписание”, после чего появится панель (рис. 3.12).

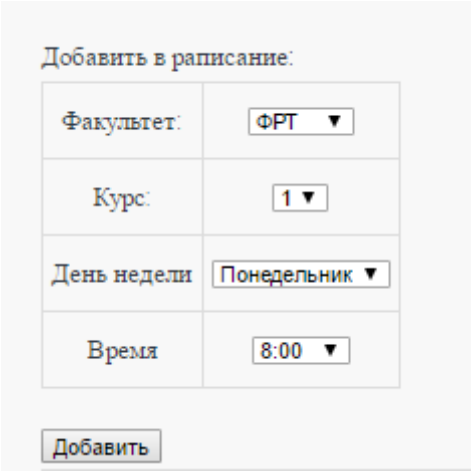


Рисунок 3.12. – Панель добавления группы в расписание

После нажатия кнопки “Добавить”, группа появится в расписании занятий.

Очищение расписания

Для того, чтобы очистить расписание занятий необходимо нажать кнопку “Очистить расписание”, после чего расписание станет пустым.

Удаление студентов

Выбрав в меню (рис. 3.3.) пункт “Редактирование списка студентов” в окне отобразит список всех студентов на кафедре ФВиС (рис. 3.13.).

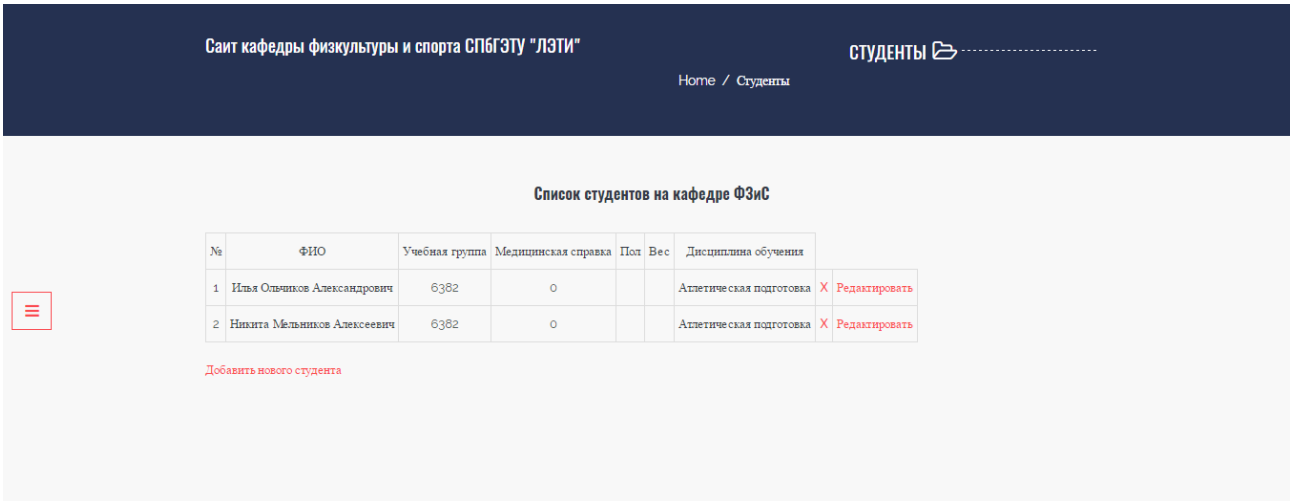


Рисунок 3.13. – Список студентов на кафедре ФВиС

Для того чтобы удалить студента необходимо нажать кнопку ”удалить”, после чего студент будет удалён.

Перевод студента на другую дисциплину

Для того чтобы перевести студента с одной дисциплины на другую необходимо перейти в режим редактирования студента, нажав на кнопку “Редактировать”, после чего откроется окно редактирования студента (см. рис. 3.14.).

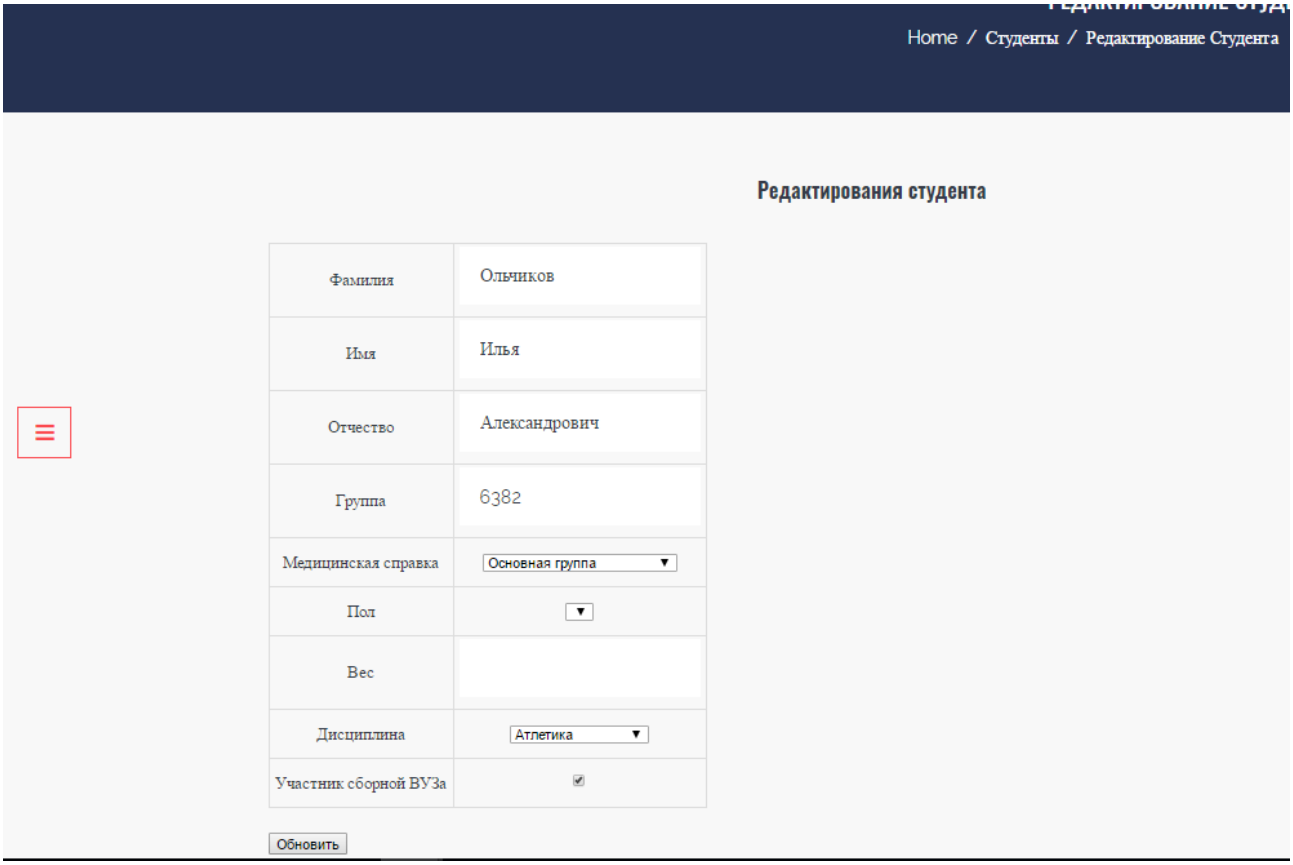


Рисунок 3.14. – Окно редактирования студента

Необходимо выбрать ту дисциплину, на которую студента переводим и нажать кнопку “Обновить”.

Редактирование студента

Редактирование данных студента производиться в том же окне, что и изменение дисциплины (см. рис. 3.14.).

Добавление студента

Для того, чтобы добавить студента необходимо в списке студентов нажать на кнопку “Добавить нового студента”, после чего откроется новое окно, изображенное на рис. 3.15.

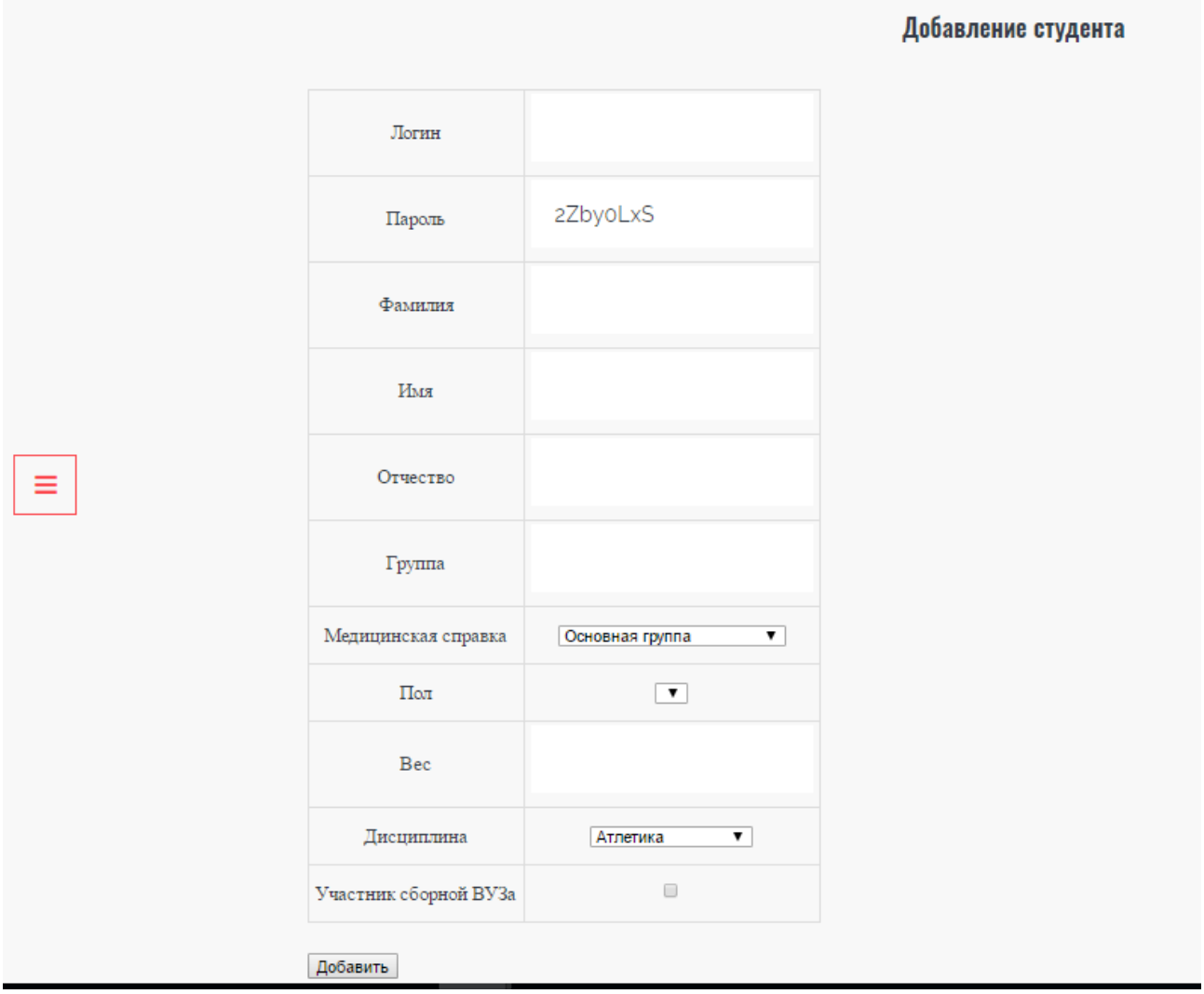


Рисунок 3.15. – Окно добавления студента в систему

После того, как все поля будут заполнены, необходимо нажать кнопку “Добавить”, после чего студент появится в системе.

Для того чтобы начать редактирование журналов посещаемости и сдачи нормативов студентами необходимо в меню (см. рис. 3.3.) выбрать “Редактирование журнала”, после чего откроется окно журнала (см. рис. 3.16.).

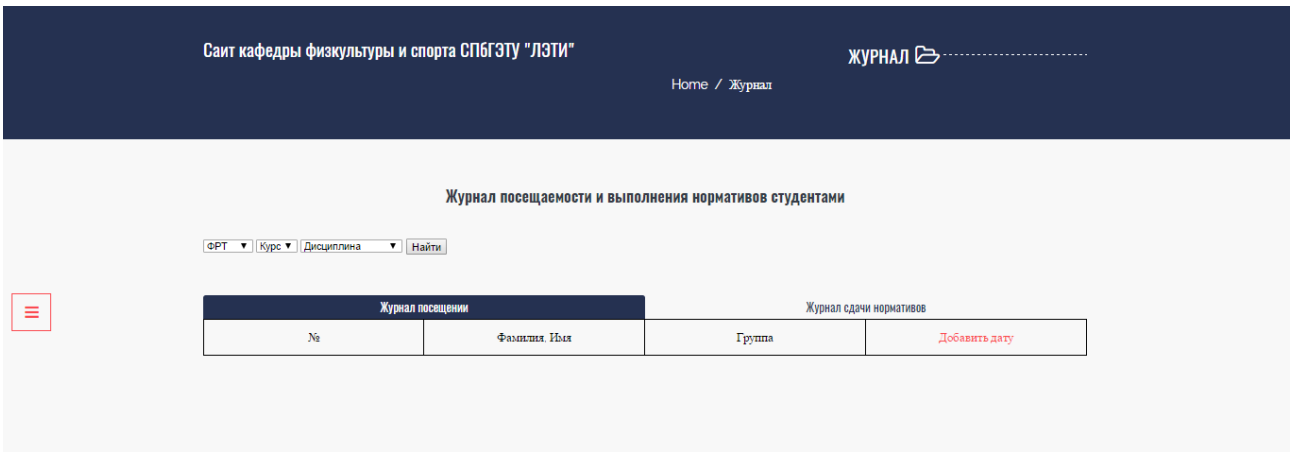


Рисунок 3.16. – Вид журнала

Изменение результатов посещения занятий

Первым делом необходимо выбрать факультет, курс и дисциплину, то есть группу, в которой будет производиться редактирование. После этого необходимо открыть вкладку “Журнал сдачи нормативов”. После чего откроется окно, изображенное на рис. 3.17.

Рисунок 3.17. – Журнал посещений занятий студентами

Далее необходимо нажать кнопку “Добавить дату”, после чего появится новая колонка в таблице (см. рис. 3.18.), в заголовке которой необходимо выбрать дату занятия, а в строках студентов выставить балл за посещение занятия студентом (от 0 до 2 баллов).

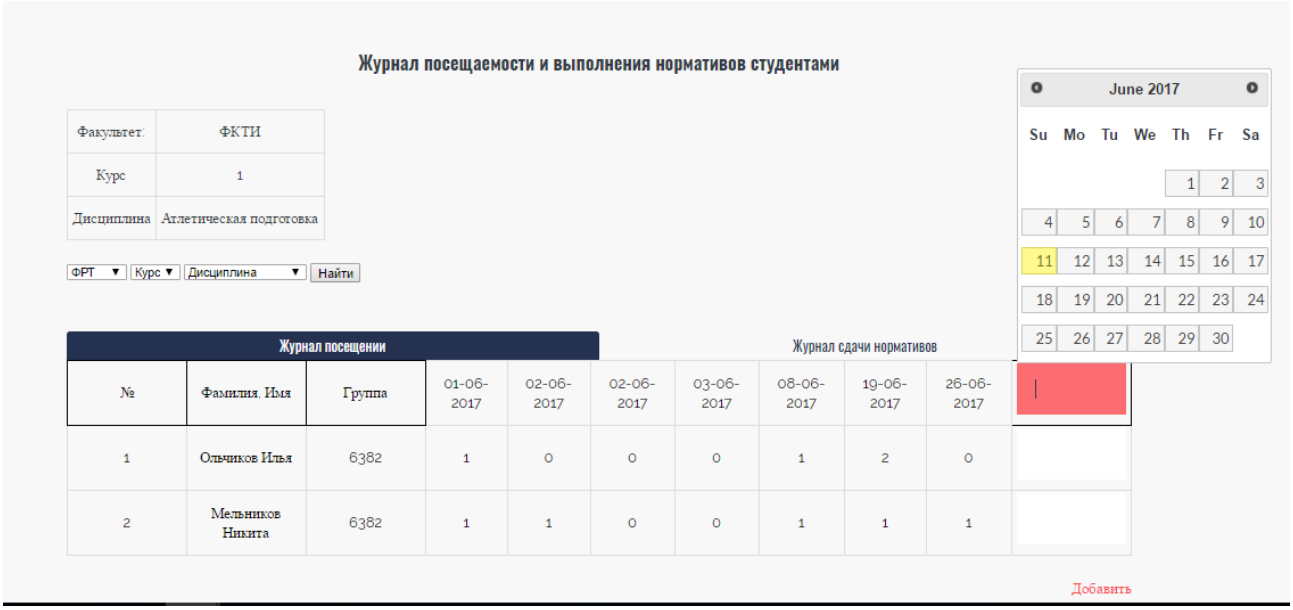


Рисунок 3.18. – Заполнение журнала посещения

Для того, чтобы изменить балл за посещенное студентом занятие необходимо дважды щелкнуть по той ячейке, которую необходимо изменить. А в конце добавления нажать кнопку “Добавить”.

Изменение результатов сдачи нормативов

Изменение результатов сдачи нормативов производится так же как и при изменении посещаемости занятий, только во вкладке “Журнал сдачи нормативов”. Вид журнала нормативов см. в приложении В19.

Рисунок 3.19. – Вид журнала сдачи нормативов.

Редактирование новостей кафедры: добавление, удаление корректировка;

Для того чтобы добавить новость, удалить или откорректировать новость необходимо перейти в административную панель WordPress. После чего необходимо выбрать вкладку “Записи”. А затем “Добавить запись” (см.рис. 3.20.).

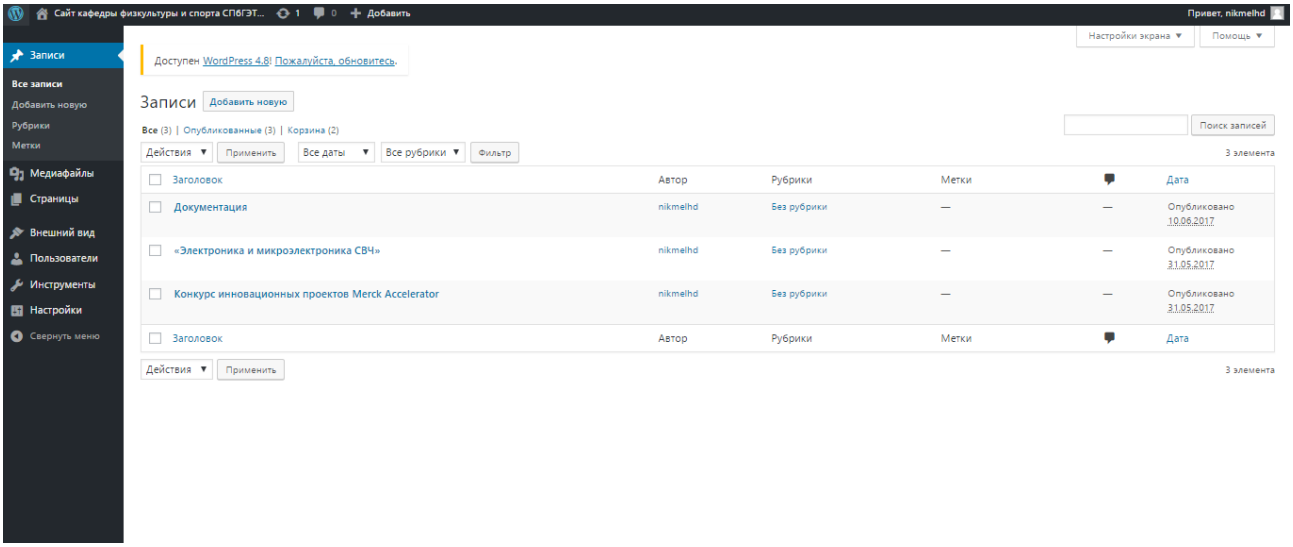


Рисунок 3.20. – Добавление записей в административной панели WordPress

Установка контрольных дат

Для того, чтобы перейти во вкладку установки контрольных дат, необходимо в меню (см. рис. 3.3.) выбрать вкладку “Редактирование контрольных дат”. На данной странице необходимо заполнить все поля. После чего нажать кнопку “Обновить”.

ЗАКЛЮЧЕНИЕ

Поставленные задачи, которые были решены:

1. Проведен обзор предметной области;
2. Описано предполагаемое решение;
3. Реализован прототип интерфейса приложения;
4. Изучены способы реализации архивных БД.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Новикова Т. В., Колтеева Т. В. Преимущества и недостатки системы электронного документооборота // Молодой ученый. – 2009. – №7. – С. 98-102. [Электронный ресурс]. – URL https://moluch.ru/archive/7/504/ (дата обращения: 20.10.2018).

2. Хранилище данных // Systems Engineering Thinking Wiki – 2016. – [Электронный ресурс]. URL: http://sewiki.ru/%D0%A5%D1%80%D0%B0%D0%BD%D0%B8%D0%BB%D0%B8%D1%89%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85 (дата обращения: 25.10.2018).

3. Eric Thomsen. OLAP Solutions. Building Multidimensional Information Systems. Second Edition. John Wiley & Sons, Inc., 2002. 688 с.

4. Matteo Golfarelli. Data Warehouse Design: Modern Principles and Methodologies. McGraw Hill Professional, 2009. 480 с.

5. Ann Weinberger. The Power of Hybrid OLAP in a Multidimensional World // SUGI Conference: 25, April 9-12, 2000 / Indiana Covention Center, Indiana. Paper 133.

6. Н.Чемеркина. OLAP-клиент - OLAP-сервер. Компромисс выбора // RM-Magazin, 2002, вып. №2.

7. Cubes – OLAP Framework // Cubes – 2016. – [Электронный ресурс]. – URL: http://cubes.readthedocs.io/en/v1.1/index.html (дата обращения: 25.05.2018).

8. CubesViewer - OLAP Visual Viewer and Explore Tool – 2016. – [Электронный ресурс]. – URL: https://github.com/jjmontesl/cubesviewer/blob/master/doc/guide/index.md (дата обращения: 23.11.2018).