

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

БИРСКИЙ ФИЛИАЛ  
ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
КАФЕДРА ИНФОРМАТИКИ И ЭКОНОМИКИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ПО ПРОГРАММЕ БАКАЛАВРИАТА

САИТОВА ЕЛЕНА БИКЯНОВНА

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА СПОРТИВНЫХ  
ДОСТИЖЕНИЙ ШКОЛЬНИКОВ

Выполнил:  
Студентки 5 курса заочной формы обучения  
Направление подготовки 09.03.03 Прикладная  
информатика  
Направленность (профиль) Прикладная  
информатика в информационной сфере

Допущено к защите в ГЭК и проверено на  
объем заимствования:

Заведующий кафедрой  
\_\_\_\_\_ к.п.н., доцент \_\_\_\_\_  
(ученая степень, ученое звание)

Руководитель  
\_\_\_\_\_ к.ф-м.н., доцент \_\_\_\_\_  
(ученая степень, ученое звание)

\_\_\_\_\_ / \_Волкова Т.И.\_  
(подпись) (И.О. Фамилия)

\_\_\_\_\_ / \_Пушкарев Н.А.\_  
(подпись) (И.О. Фамилия)

« \_\_\_\_ » \_\_\_\_\_ 2017г.

## **Содержание**

<b>Введение</b>	<b>3</b>
<b>1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ</b>	
1.1. Постановка задачи	5
1.2. Анализ предпроектной ситуации	5
1.3 Обоснование выбранного метода	7
<b>2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ</b>	
2.1. Обоснование необходимости разработки информационной системы	9
2.2. Выбор программно-аппаратной платформы	10
2.3. Анализ входных данных	15
2.4. Выбор и обоснование базового варианта системы	15
2.5. Разбиение системы на модули и организация взаимодействия между ними	15
2.6. Проектирование пользовательского интерфейса	16
2.7. Проектирование выходных форм	19
<b>3. РАЗРАБОТКА КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ</b>	
3.1. Разработка модуля основных элементов программирования информационной системы	23
3.2. Разработка модуля баз данных	25
3.3 Разработка модуля обработки данных	26
3.4. Разработка руководства программиста	46
<b>ЗАКЛЮЧЕНИЕ</b>	<b>52</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>53</b>

## ВВЕДЕНИЕ

Актуальность выбранной темы заключается в том, что создание информационной системы учета спортивных достижений школьников следует рассматривать как одну из важнейших программ развития спортивного направления и не только, состоящую из цепочки взаимосвязанных проектов, результаты каждого из которых необходимы для реализации последующих.

Объектом исследования является информационная система учета спортивных достижений школьников.

Предметом исследования является использование и применение информационной системы, разработанная, с помощью средств программирования.

Целью нашего дипломного проекта является программный продукт, который является средством учета спортивных достижений школьников.

Для достижения цели были сформулированы следующие задачи:

- Дать понятие и определить сущность информационной системы для учета спортивных достижений школьников;
- Изучить и подобрать необходимую литературу по созданию информационной системы;
- Раскрыть общие сведения о языке программирования PHP и системе управления базами данных MySQL, описать язык гипертекстовой разметки HTML;
- Разработать информационную систему учета спортивных достижений школьников;
- Описать этапы создания информационной системы.

Были использованы следующие методы исследования: анализ научной литературы, сравнение, систематизация и моделирование.

Теоретическая значимость работы заключается в том, что раскрыты понятия информационной системы для оценки эффективности работы преподавателя. Также были получены знания о языке PHP, СУБД MySQL и языке гипертекстовой разметке HTML.

Практическая значимость исследования заключается в создании функционирующей информационной системы, которая позволяет автоматизировать учет спортивных достижений школьников.

Управление учебным процессом при различных информационных систем отличается от того, который используется в обычных школах и других учебных заведениях.

Для решения проблемы автоматизации управления учебным процессом был выбран метод разработки информационной системы для учета спортивных достижений школьников.

Клиент-серверная часть информационной системы учета спортивных достижений школьников представляет собой веб-приложение и использует в своей работе все современные технологии из области веб-программирования. Все информационные данные содержатся в реляционной базе данных. Для добавления, изменения и получения информации из базы данных используются скрипты, написанные на языке PHP. В результате их выполнения формируется HTML-страница, которая открывается в браузере. Эта страница представляет собой интерфейс для взаимодействия пользователя с программой.

В процессе создания программного комплекса использовалось следующее специализированное программное обеспечение: PHP, HTML, CSS, JavaScript, jQuery, MySQL

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. Постановка задачи

Управление учебным процессом при различных информационных системах отличается от того, который используется в обычных школах и других учебных заведениях. Основной сложностью является большое количество информации, которое необходимо обрабатывать преподавателям, при участии в различных соревнованиях и конкурсах. В условиях ограниченного количества обслуживающего персонала это является трудоемкой задачей, которая требует использования информационных технологий.

Проблема использования программного обеспечения для автоматизации управления учебным процессом назрела давно, так как существующие методы документооборота не предоставляет возможности оперативно решать поставленные задачи. Используя для работы только офисные пакеты, появляется множество избыточной информации, которую сложно поддерживать ее в актуальном состоянии, так как требуется вносить изменения во множество документов.

Внесение всей необходимой информации в базу данных позволит легко поддерживать ее в актуальном состоянии, а также избавит от проблемы избыточности и дублирования информации. С помощью программных средств из базы данных можно получать информацию в любом удобном виде, производить выборку только нужных данных по заданным параметрам. Ведь это в свою очередь, облегчает работу не только программиста, который занимается всей этой деятельностью, но облегчает просмотр всей базы в целом для пользователя. Использование программных средств также позволит безошибочно производить автоматизированные расчеты.

Все это можно реализовать в информационной системе, которая бы помогала в управлении учебным процессом в целом, а не только в конкретных направлениях.

## 1.2. Анализ предпроектной ситуации

Для достижения поставленной цели необходимо при учете спортивных достижений школьников использовать специализированное программное обеспечение. Существует несколько методов решения данной проблемы:

### 1. Использовать существующее программное обеспечение.

Плюсами данного метода является:

- Нет необходимости в разработке программы, что экономит время и средства на создание программы;
- Программой можно пользоваться сразу после приобретения.

Минусами данного метода является:

- Программа может не подойти под конкретные особенности учета спортивных достижений школьников.

### 2. Заказать разработку программы в фирмах по созданию программного обеспечения.

Плюсами данного метода является:

- Нет необходимости в поиске программиста, дизайнера, верстальщика;
- Программа будет разработана под конкретные особенности учета спортивных достижений школьников.

Минусами данного метода является:

- Высокая стоимость разработки, дальнейшая доработка также требует значительных финансовых вложений.

### 3. Разработать информационную систему учета спортивных достижений школьников.

Плюсами данного метода является:

- Невысокая стоимость разработки;
- Создание приложения с необходимым функционалом;
- Расширение функционала без дополнительных затрат.

Минусами данного метода является:

- Время на изучение средств программирования и на разработку.

Для решения проблемы автоматизации учета спортивных достижений школьников был выбран метод разработки информационной системы учета спортивных достижений школьников самостоятельно. Обоснованием выбора данного метода служат следующие факторы:

1. На рынке программного обеспечения нет подходящих под учет спортивных достижений школьников программных продуктов в школе;
2. Заказ на разработку программы в фирмах по созданию программного обеспечения является слишком дорогостоящим для бюджетной организации;
3. При самостоятельной разработке программного обеспечения будут учитываться все особенности учета спортивных достижений школьников;
4. Низкая стоимость разработки программного обеспечения.

В процессе проектирования информационной системы учета спортивных достижений школьников были поставлены задачи:

- создать приложение с возможностью модернизации и добавления модулей;
- создать качественный, удобный и интуитивно понятный интерфейс;
- создать модуль авторизации с разделением прав доступа на администратора и преподавателя;
- создать модули, содержащие информацию об учениках, классах, наградах, нормативах;
- создать средства для ввода преподавателем дополнительных учеников.

### 1.3 Обоснование выбранного метода

Для решения проблемы автоматизации управления учебным процессом был выбран метод разработки информационной системы для учета спортивных достижений школьников. Обоснованием выбора данного метода служат следующие факторы:

1. На рынке программного обеспечения нет подходящих под управление учебным процессом программных продуктов;
2. Заказ на разработку программы в фирмах по созданию программного обеспечения является слишком дорогостоящим для бюджетной организации;
3. При самостоятельной разработке программного обеспечения будут учитываться все особенности управления учебным процессом в дистанционном образовании;
4. Низкая стоимость разработки программного обеспечения.

Для реализации выбранного метода были использованы следующие средства разработки:

1. Язык программирования PHP, который специально разработан для создания веб-приложений и содержит все необходимые возможности для создания программного комплекса;
2. СУБД MySQL, так как является полностью бесплатной и идеально подходит для создания веб-приложений;
3. Язык разметки HTML, а также JavaScript, так как интерфейс приложения будет представлять собой веб-страницу.



## 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

### 2.1. Обоснование необходимости разработки информационной системы

В учреждении присутствует постоянная необходимость в организации возможности удалённого обмена данными между преподавателями и управленческим персоналом в процессе учета спортивных достижений школьников.

Для организации всей этой работы в учреждении используется бумажный документооборот, который уже не отвечает требованиям мобильности, отличается большими энергозатратами и потреблением финансовых ресурсов. Также со временем становится очень сложно хранить сформировавшиеся архивы.

Мы пришли к выводу, что в наше время актуальной встаёт проблема проведения документооборота учреждения.

Основной целью проведения данной оптимизации станет автоматизация учета спортивных достижений школьников посредством клиент-серверной архитектуры информационной системы

Проблема использования программного обеспечения для автоматизации учета спортивных достижений школьников назрела давно, так как существующие методы документооборота не предоставляют возможности оперативно решать поставленные задачи.

Внесение всей необходимой информации в базу данных позволит легко поддерживать ее в актуальном состоянии, а также избавит от проблемы избыточности и дублирования информации. С помощью программных средств из базы данных можно получать информацию в любом удобном виде, производить выборку только нужных данных по заданным параметрам. Использование программных средств также позволит безошибочно производить автоматизированные расчеты.

Все это можно реализовать в программном продукте, который бы помогал в учете спортивных достижений школьников.

## 2.2. Выбор программно-аппаратной платформы

Для выбора программно-аппаратной платформы мы руководствовались следующими понятиями и определениями:

- Программный комплекс;
- Веб-приложение;
- Веб-сервер;
- БД;
- СУБД;
- MVC.

Программный комплекс — это программа, состоящая из двух или более компонентов и (или) комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе другого комплекса. Как правило, компоненты программного комплекса работают с одной общей информационной базой данных и могут взаимодействовать между собой. Компоненты являются отдельными приложениями в составе комплекса, или функционируют в составе одного приложения.

Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами. Для работы с веб-приложением необходим только веб-браузер и сетевой доступ к веб-серверу приложения. Это очень удобно при работе с приложением, и позволяет работать практически с любого устройства, будь то персональный компьютер, ноутбук или планшет, и с любой операционной системой. Веб-приложение состоит из клиентской и серверной частей, тем самым реализуя технологию «клиент-сервер».

Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него.

Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует веб-страницу и отправляет её клиенту по сети с использованием протокола HTTP.

Само веб-приложение может выступать в качестве клиента других служб, например, базы данных или другого веб-приложения, расположенного на другом сервере. Ярким примером веб-приложения является система управления содержимым статей Википедии: множество её участников могут принимать участие в создании сетевой энциклопедии, используя для этого браузеры своих операционных систем (будь то Microsoft Windows, GNU/Linux или любая другая операционная система) и не загружая дополнительных исполняемых модулей для работы с базой данных статей.

Веб-сервер. Понятие "Веб-сервер" может относиться как к аппаратному, так и к программному обеспечению. С аппаратной точки зрения, Веб-сервер – это компьютер который хранит ресурсы сайта (HTML документы, CSS стили, JavaScript файлы и другое) и доставляет их на устройство конечного пользователя (веб-браузер и т.д.). Обычно подключен к сети Интернет и может быть доступен через, доменное имя.

БД (База данных) — набор сведений, хранящихся некоторым упорядоченным способом. Можно сравнить базу данных со шкафом, в котором хранятся документы. Иными словами, база данных - это хранилище данных. Сами по себе базы данных не представляли бы интереса, если бы не было систем управления базами данных (СУБД).

СУБД (Система Управления Базами Данных) — это совокупность языковых и программных средств, которая осуществляет доступ к данным, позволяет их создавать, менять и удалять, обеспечивает безопасность данных и т.д. В общем СУБД - это система, позволяющая создавать базы данных и манипулировать сведениями из них. А осуществляет этот доступ к данным СУБД посредством специального языка - SQL.

MVC (Model-View-Ccontroller, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский

интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.

Для создания клиент-серверных веб-приложений существует множество средств разработки, особенно выделяются PHP, Ruby и Python. Рассмотрим каждый из них.

## PHP

Это язык программирования, исполняемый на стороне веб-сервера, спроектированный в качестве инструмента создания динамических и интерактивных веб-сайтов. Этот язык является достаточно гибким и мощным, поэтому приобрёл большую популярность и используется в проектах любого масштаба: от простого блога до крупнейших веб-приложений.

Основные преимущества PHP:

- является свободным программным обеспечением, распространяемым под особой лицензией (PHP license);
- легок в освоении на всех этапах;
- поддерживается большим сообществом пользователей и разработчиков;
- имеет развитую поддержку баз данных;
- имеется огромное количество библиотек и расширений языка;
- может быть развёрнут почти на любом сервере;
- портирован под большое количество аппаратных платформ и операционных систем.

Недостатки PHP:

- не подходит для создания десктопных приложений или системных компонентов.

## Ruby

Ruby – динамический императивный объектно-ориентированный язык программирования. Ruby был создан под влиянием таких языков, как Perl,

Eiffel и Smalltalk. Он характеризуется динамической типизацией и автоматическим управлением памятью. Язык Ruby используется в веб-разработке в составе открытого веб-фреймворка Rails, чаще называемого Ruby on Rails (RoR).

Преимущества Ruby:

- открытая разработка;
- работает на многих платформах;
- предоставляет продвинутые методы манипуляции строками и текстом;
- легко интегрирует в свои программы высокопроизводительные серверы баз данных (DB2, MySQL, Oracle и Sybase);
- имеется простой программный интерфейс для создания многопоточных приложений;
- имеет продвинутые средства для работы с массивами;
- дополнительные возможности для обеспечения безопасности.

Недостатки Ruby:

- обучение языку выше начального уровня может оказаться непростым;
- информационных ресурсов, посвящённых Ruby, явно недостаточно;
- Ruby менее производителен по сравнению со многими другими языками, применяемыми в веб-разработке;
- Ruby относительно медленно разрабатывается и развивается.

Python

Python широко применяется как интерпретируемый язык для скриптов различного назначения. Как и Ruby, Python имеет целью приблизить синтаксис реальной программы, написанной на нём, к описывающему задачу псевдокоду, что позволяет программисту уменьшить объём программы.

Преимущества Python:

- открытая разработка;
- довольно прост в изучении, особенно на начальном этапе;

- множество полезных библиотек и расширений языка можно легко использовать в своих проектах благодаря предельно унифицированному механизму импорта и программным интерфейсам;
- механизмы модульности хорошо продуманы и могут быть легко использованы;
- абсолютно всё в Python является объектами в смысле ООП, но при этом объектный подход не навязывается программисту.

Недостатки Python:

- ограниченность средств для работы с базами данных;
- меньшая производительность Python по сравнению с основными PHP и Ruby, что создаёт этому языку репутацию медленного.

Рассмотрев и проанализировав все преимущества и недостатки языков PHP, Ruby и Python, приходим к выводу, что язык PHP лучше других подходит для выполнения поставленных задач, так как он легче других в освоении, позволяет создавать проекты любого масштаба и имеет удобные средства работы с базами данных.

Для разработки веб-приложения на PHP необходим веб-сервер со следующими требованиями:

Основные аппаратные требования:

- 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 2 гигагерц или выше;
- 2 гигабайта оперативной памяти (ОЗУ);
- 1 гигабайт свободного пространства на жестком диске;
- монитор с разрешением по ширине от 1280 пикселей.

Основные программные требования к серверу не ниже:

- операционная система Windows 7 и выше;
- версия Apache не ниже 2.2;
- версия интерпретатора PHP не ниже 5.1;
- версия СУБД MySQL не ниже 5.6.

### 2.3. Анализ входных данных

Для автоматизации учета спортивных достижений школьников необходимы следующие входные данные:

1. Список учеников школы;
2. Список класса с фамилией классного руководителя;
3. Наименование наград;
4. Список нормативов, сдаваемых школьниками.

### 2.4. Выбор и обоснование базового варианта системы

При проведении анализа предпроектной ситуации, нами было проведено сравнение методов решения поставленной проблемы. Исходя из факторов этого сравнения, для решения проблемы автоматизации учета спортивных достижений школьников, был выбран метод разработки информационной системы для учета спортивных достижений школьников.

Информационная система будет содержать информацию о школьниках, нормативах и наградах. Для этого будут созданы механизмы их создания, редактирования и удаления. Эта информация необходима создания электронной ведомости, которая будет вести автоматизированный подсчет спортивных достижений.

### 2.5. Разбиение системы на модули и организация взаимодействия между ними

Исходя из поставленных задач для формирования нашей информационной системы нами были разработаны следующие модули:

1. Модуль авторизации – служит для входа в информационную систему и разграничения прав доступа;
2. Модуль «Ученики» - добавление, редактирование и удаление учеников;
3. Модуль «Классы» - добавление, редактирование и удаление классов;
4. Модуль «Награды» - добавление, редактирование и удаление наград администратором;

5. Модуль «Нормативы» - добавление, редактирование и удаление нормативов администратором.

Взаимодействие модулей между собой представлено на рисунке 1

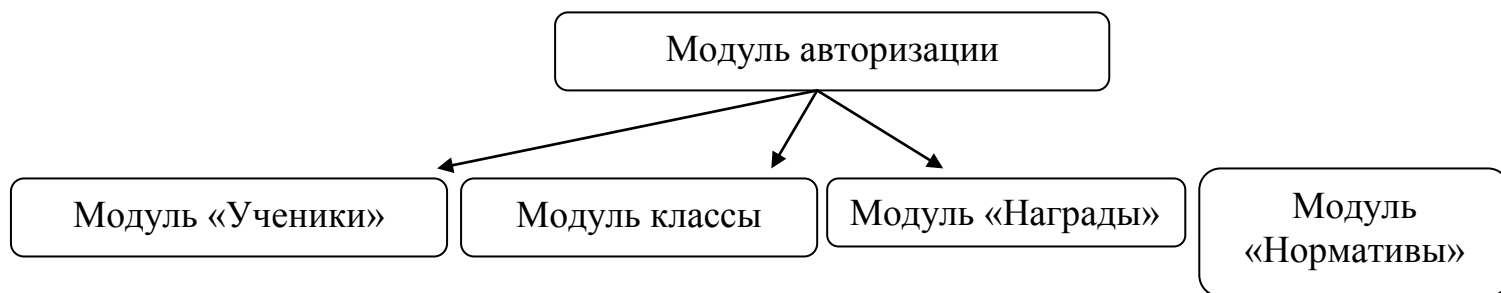


Рисунок 1. - Взаимодействие модулей между собой

## 2.6. Проектирование пользовательского интерфейса

При проектировании интерфейса информационной системы основное внимание мы уделили задаче, которая была поставлена перед нами в техническом задании:

- создать качественный, удобный и интуитивно понятный интерфейс.

Это было обусловлено тем, что любой преподаватель должен суметь работать в данной информационной системе, имея лишь начальный уровень знаний пользования персональным компьютером. Исходя из этого, было принято решение, что после создание информационной системы, будет сформировано:

- Руководство программиста – необходимое для сотрудников, которые будут заниматься непосредственно установкой и настройкой информационной системы;
- Руководство оператора – с описанием работы интерфейса информационной системы «Управления учебным процессом», необходимое для администратора и преподавателей, которые будут пользоваться этой системой.

Так как интерфейс приложения представляет собой веб-страницу, были использованы: язык разметки HTML и средство описание стилей CSS.



Рассмотрим основные шаблоны проектирования интерфейса информационной системы:

1. Общий шаблон для всех страниц, это позволит создать окна приложения с единой концепцией размещения объектов (рисунок 2)

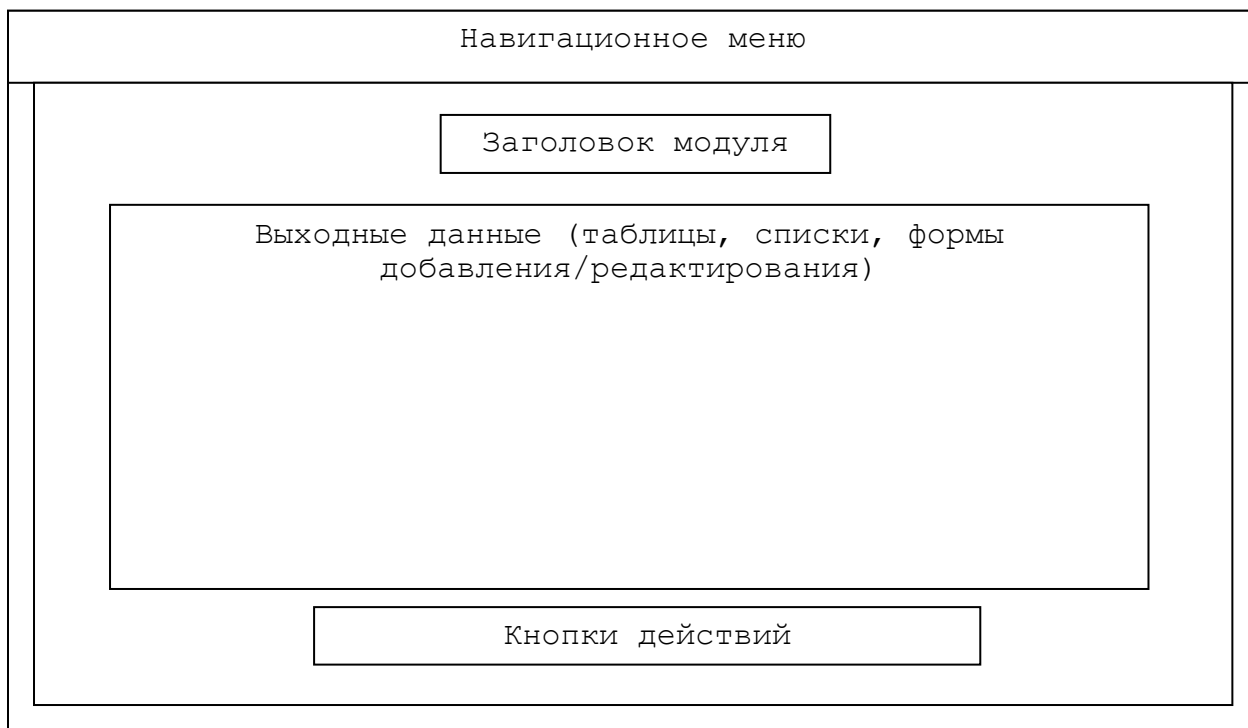


Рисунок 2 – Общий шаблон для всех страниц

2. Навигационное меню для администратора (рисунок 3)

Главная	Ученики	Классы	Награды	Нормативы	Администратор
---------	---------	--------	---------	-----------	---------------

Рисунок 3 – Навигационное меню администратора

3. Форма авторизации (рисунок 4)

The diagram illustrates the layout of an authorization form. It is contained within a rectangular frame with a header bar at the top. The form elements are arranged vertically in the center:

- A rectangular box labeled "Авторизация" (Authorization) at the top.
- A rectangular box labeled "Поле для ввода логина" (Login input field) below it.
- A rectangular box labeled "Поле для ввода пароля" (Password input field) below the login field.
- A rectangular box labeled "Кнопка входа" (Login button) at the bottom.

Рисунок 4 – Форма авторизации

#### 4. Форма добавления преподавателя (рисунок 2.6.5)

Главная	Ученики	Класс	Нормативы	Награды	Администратор
---------	---------	-------	-----------	---------	---------------

Добавление ученика

Поле для ввода фамилии ученика

Поле для ввода имени ученика

Поле для ввода отчества ученика

Выпадающий список выбора класса

Выпадающий список выбора пола

Выпадающий список даты рождения

Кнопка Добавить

Кнопка Отмена

Рисунок 5 – Форма добавления ученика

Остальные формы имеют схожий шаблон размещения информации и управляющих элементов.

## 2.7. Проектирование выходных форм

Исходя из спроектированного интерфейса и основных требований к информационной системе нами был разработаны следующие выходные формы:

1. Форма авторизации (рисунок 6). Связанная таблица БД – users

МБОУ СОШ №2 г. Туймазы

Вход для администратора

Спортивные достижения учащихся

Вход для администратора

Логин

Пароль

Войти

Рисунок 6 – Форма авторизации

2. Форма добавления ученика (рисунок 7). Связанная таблица БД – pupils

Добавление ученика

Фамилия

Имя

Отчество

Класс

Пол

Дата рождения

5А

мужской

1

январь

2010

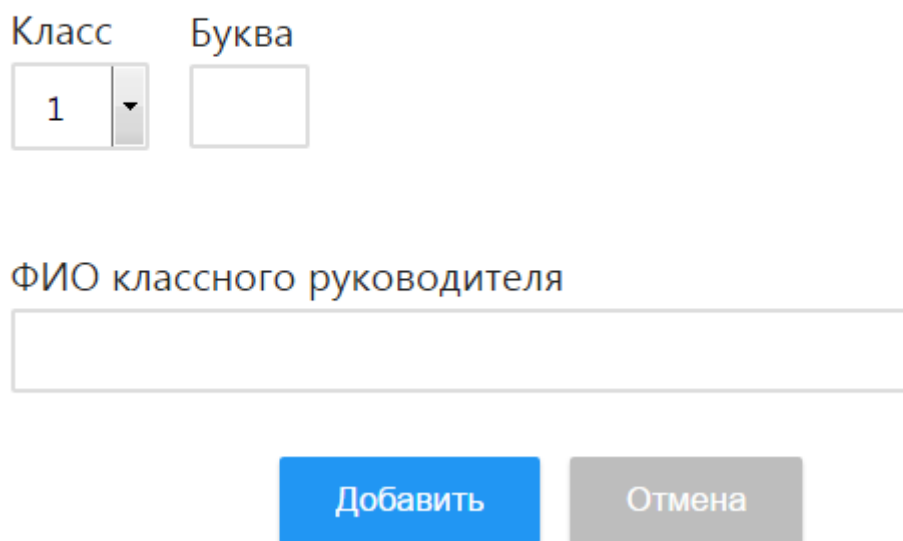
Добавить

Отмена

Рисунок 7 – Форма добавления ученика

3. Форма добавления класса (рисунок 8). Связанная таблица БД – classes

## Добавление класса



Класс Буква

1

ФИО классного руководителя

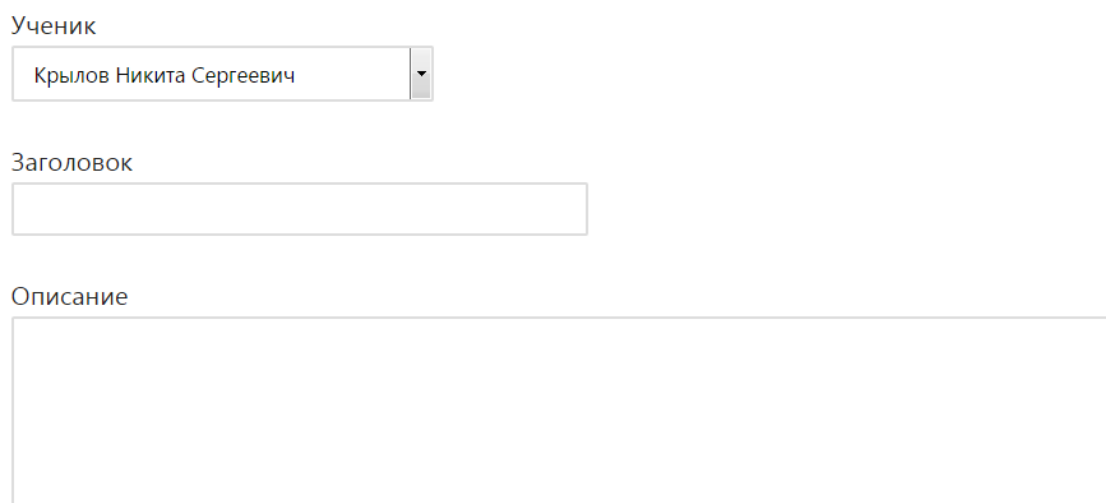
Добавить Отмена

The form consists of two input fields at the top: 'Класс' (Class) with a dropdown menu showing '1' and 'Буква' (Letter) with an empty text box. Below these is a single-line text input field for 'ФИО классного руководителя' (Class teacher's full name). At the bottom are two buttons: a blue 'Добавить' (Add) button and a grey 'Отмена' (Cancel) button.

Рисунок 8 – Форма добавления класса

4. Форма создания награды учеников(рисунок 9). Связанная таблица БД – pupil\_rewards

## Добавление спортивного достижения



Ученик

Крылов Никита Сергеевич

Заголовок

Описание

The form has four main sections. The first is 'Ученик' (Student) with a dropdown menu showing 'Крылов Никита Сергеевич'. The second is 'Заголовок' (Title) with a single-line text input field. The third is 'Описание' (Description) with a large multi-line text area. On the right side of the page, there is a vertical scrollbar.

Рисунок 9 – Форма создания наград

5. Форма создания нормативов (рисунок 10). Связанная таблица БД – pupil\_specs

## Добавление норматива

Название норматива

Добавить

Отмена

Рисунок 10 – Добавление формы Нормативы

### 3. РАЗРАБОТКА КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ

#### 3.1. Разработка модуля основных элементов программирования информационной системы

Клиент-серверная часть информационной системы учета спортивных достижений школьников представляет собой веб-приложение и использует в своей работе все современные технологии из области веб-программирования. Все информационные данные содержатся в реляционной базе данных. Для добавления, изменения и получения информации из базы данных используются скрипты, написанные на языке PHP. В результате их выполнения формируется HTML-страница, которая открывается в браузере. Эта страница представляет собой интерфейс для взаимодействия пользователя с программой.

В процессе создания программного комплекса использовалось следующее специализированное программное обеспечение:

PHP — язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере. Аббревиатура PHP означает “Hypertext Preprocessor (Препроцессор Гипертекста)”. Синтаксис языка берет начало из C, Java и Perl. PHP достаточно прост для изучения. Преимуществом PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц. Подробнее о преимуществах PHP можно узнать здесь. Важным преимуществом языка PHP перед такими языками, как языков Perl и C заключается в возможности создания HTML документов с внедренными командами PHP. Подробнее об этой возможности смотрите здесь. Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера. Вы даже можете сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

HTML — это язык разметки документов, используемый при создании веб-страниц. Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме.

CSS — это набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид. Возможность работы со стилями издавна включают в развитые издательские системы и текстовые редакторы, тем самым позволяя одним нажатием кнопки придать тексту заданный, заранее установленный вид. Теперь это доступно и создателям сайта, когда цвет, размеры текста и другие параметры хранятся в определенном месте и легко «прикручиваются» к любому тегу. Еще одним преимуществом стилей является то, что они предлагают намного больше возможностей для форматирования, чем обычный HTML.

JavaScript — прототипно-ориентированный сценарный язык программирования. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

MySQL — свободная реляционная система управления базами данных. MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.



Веб-браузер — прикладное программное обеспечение для просмотра веб-страниц; содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач. В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов.

ХАМРР — кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

Информационная модель базы данных, созданная в СУБД MySQL.

### 3.2. Разработка модуля баз данных

Для информационной системы учета спортивных достижений школьников нами были разработаны базы данных со следующим содержанием:

Таблица 1 - **classes** - классы

№	Название поля	Тип поля	Описание
1	id	INT unsigned	Идентификатор класса
2	class_num	INT unsigned	Номер класса
3	class_litera	VARCHAR(1)	Буква класса
4	class_teacher	VARCHAR(150)	ФИО классного руководителя

Таблица 2 - **pupils** - ученики

№	Название поля	Тип поля	Описание
1	id	INT unsigned	Идентификатор ученика
2	class_id	INT unsigned	Идентификатор класса
3	f	VARCHAR(100)	Фамилия
4	i	VARCHAR(100)	Имя
5	o	VARCHAR(100)	Отчество
6	gender	VARCHAR(1)	Пол
7	birthdate	DATETIME	Дата рождения

Таблица 3 - **pupil\_rewards** – награды учеников

№	Название	Тип поля	Описание
---	----------	----------	----------

	поля		
1	id	INT unsigned	Идентификатор награды
2	pupil_id	INT unsigned	Идентификатор ученика
3	title	VARCHAR(100)	Название награды
4	description	TEXT	Описание награды
5	date	DATETIME	Дата
6	photo	VARCHAR(255)	Фотография

Таблица 4- **specs - названия нормативов**

№	Название поля	Тип поля	Описание
1	id	INT unsigned	Идентификатор названия норматива
2	spec_name	VARCHAR(255)	Название норматива

Таблица 5 - **pupil\_specs – нормативы учеников**

№	Название поля	Тип поля	Описание
1	id	INT unsigned	Идентификатор норматива
2	spec_id	INT unsigned	Идентификатор названия норматива
3	pupil_id	INT unsigned	Идентификатор ученика
4	spec_value	VARCHAR(100)	Значение норматива

Таблица 6 - **users – пользователи**

№	Название поля	Тип поля	Описание
1	id	INT unsigned	Идентификатор пользователя
2	user_login	VARCHAR(32)	Логин
3	user_password	VARCHAR(32)	Пароль
4	user_hash	VARCHAR(32)	Идентификатор авторизации
5	school_name	VARCHAR(255)	Название школы

Все таблицы созданы с помощью языка структурированных запросов (SQL) в системе управления базами данных MySQL.

### 3.3 Разработка модуля обработки данных

Для связи таблиц между собой используются специальные запросы, данные из которых поступают в обработчик, работающий на PHP. Скрипты обработчика формируют интерфейс программы в виде HTML страницы, которая содержит необходимую информацию и средства управления.

Исполняемый скрипт PHP - index.php

```

<?php
//application initialize
//defines
define('_A',1);

if (file_exists(dirname(__FILE__) . '/config.php')) {
    include_once dirname(__FILE__) . '/config.php';
}
else die('Configuration file not found!');

date_default_timezone_set ("Europe/Moscow");
header('Content-Type: text/html; charset=UTF-8');
error_reporting(E_ALL & ~E_NOTICE);

define('COMPATH','application/components/');
if ($_SERVER['REQUEST_METHOD']=='GET'){
    define('METHOD','GET');
}
else {
    define('METHOD','POST');
}
$sarr = parse_url($_SERVER['REQUEST_URI']);
define('BASEURL',$sarr['path']);

//variables
$modulesList = array(
    'administrator',
    'teacher',
    'users'
);
$mainList = array(
    2 => 'teacher',
    101 => 'administrator'
);

//includes
require_once 'application/functions.php';
require_once 'application/db.php';
require_once 'application/app.php';

//db connection
$db = new Db();

//route
$component = 'main';
if ($com = F::getVar('com')) {
    if (in_array($com,$modulesList)){
        $component = $com;
    }
    else {
        F::redirect(BASEURL);
    }
}

```

```

//login check
$user = checkLogin($db,$component);
$id = $user['id'];
$type = $user['user_type'];
if ($component == 'main') {$component = $mainList[$userType];}

//access check

//component create
$comFile = COMPATH.'com_'. $component.'/component.php';
require_once $comFile;
$com_class = new Component($component,$db,$user);

function checkLogin($db,$component){
    $result = false;
    $view = F::getVar('view');

    if ((($_COOKIE['id'])and($_COOKIE['hash'])) ) {
        if ( ( is_numeric($_COOKIE['id'])) and (preg_match("/^[0-9a-zA-Z]+$/",$_COOKIE['hash'])) ){
            $query = array(
                'text' => "SELECT *
                                FROM users
                                WHERE id=:id
                                AND user_hash=:hash
                                LIMIT 1",
                'data' => array (
                    'id' => $_COOKIE['id'],
                    'hash'=> $_COOKIE['hash']
                ),
                'limit1' => true,'select' => true
            );
            if ($row = $db->queryPrepared($query)) {
                $result = true;
            }
        }
    }
    if (($component === 'users') and (( $view === 'login') or ( $view === 'post_login'))){
        if ($result) {
            F::home();
        }
    }
    elseif (!$result) {
        F::redirect('?com=users&view=login');
        exit;
    }
    else {
        return $row;
    }
}
echo '<!--'.$component.'-->';
?>

```

## Исполняемый скрипт PHP - app.php

```
<?php
defined('_A') or die;

class appComponent{
    public $userTypes = array(
        101=>'Администратор',
        1=>'Ученик',

    );
    public $monthNames = array(
        1 => 'январь',2 => 'февраль',3 => 'март',4 => 'апрель',5 => 'май',6 => 'июнь',7 =>
'июль',8 => 'август',9 => 'сентябрь',10 => 'октябрь',11 => 'ноябрь',12 => 'декабрь');
    public $monthNames2 = array(
        1 => 'января',2 => 'февраля',3 => 'марта',4 => 'апреля',5 => 'мая',6 => 'июня',7
=> 'июля',8 => 'августа',9 => 'сентября',10 => 'октября',11 => 'ноября',12 => 'декабря');

    public function __construct($com,$db,$user){
        $this->com = $com;
        $this->getView();
        $this->Db = $db;
        $this->user = $user;
        $this->userId = $user['id'];
        $this->userType = $user['user_type'];
        if (METHOD === 'GET') {
            $this->template();
        }
        else {
            $this->action();
        }
    }

    function getView(){
        $view = 'index';
        if ($v = F::getVar('view')) {
            if ($this->viewList[$v]){
                $view = $v;
            }
            else {
                F::home();
            }
        }
        $this->view = $view;
    }

    public function template(){
        include 'template/template.php';
    }

    public function getContent(){
        $this->action();
    }
}
```

```

public function action(){
    $action = 'action_'. $this->view;
    $this->$action();
}
public function action_index(){ }
public function checkAccess(){ }

function messagesClear(){
    if (isset($_COOKIE['wrongmsg'])){
        foreach($_COOKIE['wrongmsg'] as $key => $value){
            setcookie('wrongmsg['.$key.']", ');
        }
    }
    if (isset($_COOKIE['infomsg'])){
        foreach($_COOKIE['infomsg'] as $key => $value){
            setcookie('infomsg['.$key.']", ');
        }
    }
}

}

```

Исполняемый скрипт PHP –  
 application\components\com\_administrator\component.php

```

<?php
defined('_A') or die;
//administrator
class Component extends appComponent{
    public $viewList = array(
        'filials' => 1,
        'filial_add' => 1,
        'post_filial_add' => 1,
        'filial_edit' => 1,
        'post_filial_edit' => 1,
        'filial_delete' => 1,
        'chairs' => 1,
        'chair_add' => 1,
        'post_chair_add' => 1,
        'chair_edit' => 1,
        'post_chair_edit' => 1,
        'chair_delete' => 1,
        'teacher' => 1,
        'teachers' => 1,
        'teacher_add' => 1,
        'post_teacher_add' => 1,
        'teacher_edit' => 1,
        'post_teacher_edit' => 1,
        'teacher_delete' => 1,
        'teacher_password_change' => 1,
        'post_teacher_password_change' => 1
    );

    public function getContent(){

```

```

        if ($this->userType == 101) {
            $this->action();
        }
        else {F::redirect(BASEURL.'?no_access');}
    }

    public function action_index(){
        include(COMPATH.'com_administrator/main.php');
    }

/*-----filials-----*/

    public function action_filials(){
        $query = "SELECT * FROM filials";
        $filials = $this->Db->query($query);
        include(COMPATH.'com_administrator/filials.php');
    }

    public function action_filial_add(){
        include(COMPATH.'com_administrator/filial_add_form.php');
    }

    public function action_post_filial_add(){
        $filialTitle = F::getVar('filial_title');
        if ($filialTitle) {
            $query = array(
                'text' => "INSERT INTO `filials`
                        (filial_title)
                        VALUES
                        (:filial_title)",
                'data' => array (
                    'filial_title' => $filialTitle
                )
            );
            $this->Db->queryPrepared($query);
            F::redirect('?com=administrator&view=filials');
        }
        else{
            F::redirect('?com=administrator&view=filial_add');
        }
    }

    public function action_filial_edit(){
        if ( ($this->filialEditId = F::getVar('id')) and (is_numeric($this->filialEditId)) ) {
            $query = array(
                'text' => "SELECT filial_title
                        FROM filials
                        WHERE id = :id
                        LIMIT 1",
                'data' => array (
                    'id' => $this->filialEditId

```

```

        ),
        'limit1' => true, 'select' => true
    );
    if ($row = $this->Db->queryPrepared($query)){
        $this->filialEditTitle = $row['filial_title'];
        include(COMPATH.'com_administrator/filial_edit_form.php');
    }
    else {
        F::redirect('?com=administrator&view=filials');
    }
}
else {
    F::redirect('?com=administrator&view=filials');
}
}

```

```

public function action_post_filial_edit(){
    $filialId = F::getVar('id');
    $filialTitle = F::getVar('filial_title');
    if ($filialTitle || $filialId) {
        $query = array(
            'text' => "UPDATE filials
                        SET filial_title = :filial_title
                        WHERE id = :id
                        LIMIT 1",
            'data' => array (
                'id' => $filialId,
                'filial_title' => $filialTitle
            )
        );
        $this->Db->queryPrepared($query);
        F::redirect('?com=administrator&view=filials');
    }
    else{
        F::redirect('?com=administrator&view=filial_edit&id='.$filialId);
    }
}

```

```

public function action_filial_delete(){
    $filialId = F::getVar('id');
    if (!is_numeric($filialId)) {
        F::redirect('?com=administrator&view=filials');
    }
    $query = array(
        'text' => "SELECT COUNT(*)
                    FROM users
                    WHERE teacher_filial = :id",
        'data' => array (
            'id' => $filialId
        ),
        'limit1' => true, 'select' => true
    );
    $rows = $this->Db->queryPrepared($query);
    $count = $rows['COUNT(*)'];
}

```



```

        if ($count){

F::redirect('?com=administrator&view=filials&id='.$filialId.'&no_delete='.$count);
        }else{
            $query = array(
                'text' => "DELETE FROM filials
                                WHERE id = :id
                                LIMIT 1",
                'data' => array (
                    'id' => $filialId
                )
            );
            $this->Db->queryPrepared($query);
            F::redirect('?com=administrator&view=filials');
        }

    }

/*-----chairs-----*/

    public function action_chairs(){
        $query = "SELECT * FROM chairs";
        $chairs = $this->Db->query($query);
        include(COMPATH.'com_administrator/chairs.php');
    }

    public function action_chair_add(){
        include(COMPATH.'com_administrator/chair_add_form.php');
    }

    public function action_post_chair_add(){
        $chairTitle = F::getVar('chair_title');
        if ($chairTitle) {
            $query = array(
                'text' => "INSERT INTO `chairs`
                                (chair_title)
                                VALUES
                                (:chair_title)",
                'data' => array (
                    'chair_title' => $chairTitle
                )
            );
            $this->Db->queryPrepared($query);
            F::redirect('?com=administrator&view=chairs');
        }
        else{
            F::redirect('?com=administrator&view=chair_add');
        }
    }

    public function action_chair_edit(){

```

```

if ( ($this->chairEditId = F::getVar('id')) and (is_numeric($this->chairEditId)) ) {
    $query = array(
        'text' => "SELECT chair_title
                    FROM chairs
                    WHERE id = :id
                    LIMIT 1",
        'data' => array (
            'id' => $this->chairEditId
        ),
        'limit1' => true, 'select' => true
    );
    if ($row = $this->Db->queryPrepared($query)){
        $this->chairEditTitle = $row['chair_title'];
        include(COMPATH.'com_administrator/chair_edit_form.php');
    }
    else {
        F::redirect('?com=administrator&view=chairs');
    }
}
else {
    F::redirect('?com=administrator&view=chairs');
}
}

public function action_post_chair_edit(){
    $chairId = F::getVar('id');
    $chairTitle = F::getVar('chair_title');
    if ($chairTitle || $chairId) {
        $query = array(
            'text' => "UPDATE chairs
                        SET chair_title = :chair_title
                        WHERE id = :id
                        LIMIT 1",
            'data' => array (
                'id' => $chairId,
                'chair_title' => $chairTitle
            )
        );
        $this->Db->queryPrepared($query);
        F::redirect('?com=administrator&view=chairs');
    }
    else{
        F::redirect('?com=administrator&view=chair_edit&id='.$chairId);
    }
}

public function action_chair_delete(){
    $chairId = F::getVar('id');
    if (is_numeric($chairId)) {
        F::redirect('?com=administrator&view=chairs');
    }
    $query = array(
        'text' => "SELECT COUNT(*)
                    FROM users

```

```

WHERE teacher_chair = :id",
        'data' => array (
            'id' => $chairId
        ),
        'limit1' => true, 'select' => true
    );
    $rows = $this->Db->queryPrepared($query);
    $count = $rows['COUNT(*)'];
    if ($count){
        F::redirect('?com=administrator&view=chairs&id='.$filialId.'&no_delete='.$count);
    }else{
        $query = array(
            'text' => "DELETE FROM chairs
                        WHERE id = :id
                        LIMIT 1",
            'data' => array (
                'id' => $chairId
            )
        );
        $this->Db->queryPrepared($query);
        F::redirect('?com=administrator&view=chairs');
    }
}

}

/*-----teachers-----*/

public function action_teachers(){
    /*$query = "SELECT t.id AS
teacher_id,user_name,user_login,teacher_filial,teacher_chair,teacher_post,teacher_rate
            filial_title,chair_title
            FROM users t
            LEFT JOIN filials f
            ON t.teacher_filial = f.id
            LEFT JOIN chairs c
            ON t.teacher_chair = c.id
            WHERE t.user_type = 2
            ";*/
    $query = "SELECT id,user_name,user_login
            FROM users
            WHERE user_type = 2
            ORDER BY user_name
            ";
    $teachers = $this->Db->query($query);
    include(COMPATH.'com_administrator/teachers.php');
}

public function action_teacher(){
    $teacherId = F::getVar('id');
    $query = array(

```

```

        'text' => "SELECT t.id AS
teacher_id,user_name,user_login,teacher_filial,teacher_chair,

teacher_post,teacher_rate,teacher_post_sub,teacher_rate_sub,teacher_zk,filial_title,chair_title
FROM users t
LEFT JOIN filials f
ON t.teacher_filial = f.id
LEFT JOIN chairs c
ON t.teacher_chair = c.id
WHERE t.id = :id
LIMIT 1",

        'data' => array (
            'id' => $teacherId
        ),
        'limit1' => true, 'select' => true
    );
    $teacher = $this->Db->queryPrepared($query);
    include(COMPATH.'com_administrator/teacher.php');
}

public function action_teacher_add(){
    $query = "SELECT * FROM filials";
    $filials = $this->Db->query($query);
    $query = "SELECT * FROM chairs";
    $chairs = $this->Db->query($query);
    include(COMPATH.'com_administrator/teacher_add_form.php');
}

public function action_post_teacher_add(){
    $user_name = F::getVar('user_name');
    $user_login = F::getVar('user_login');
    $user_password = F::getVar('user_password');
    $teacher_filial = F::getVar('teacher_filial');
    $teacher_chair = F::getVar('teacher_chair');
    $teacher_post = F::getVar('teacher_post');
    $teacher_rate = F::getVar('teacher_rate');
    $teacher_post_sub = F::getVar('teacher_post_sub');
    $teacher_rate_sub = F::getVar('teacher_rate_sub');
    $teacher_zk = F::getVar('teacher_zk');

    if ($user_name || $user_login || $user_password) {
        $user_password = md5($user_password);
        $query = array(
            'text' => "INSERT INTO users

            (user_type,user_name,user_login,user_password,teacher_filial,teacher_chair,teacher_post,teacher_rate,teacher_post_sub,teacher_rate_sub,teacher_zk)
            VALUES

            (2,:user_name,:user_login,:user_password,:teacher_filial,:teacher_chair,:teacher_post,:teacher_rate,:teacher_post_sub,:teacher_rate_sub,:teacher_zk)",
            'data' => array (
                'user_name' => $user_name,

```

```

        'user_login' => $user_login,
        'user_password' => $user_password,
        'teacher_filial' => $teacher_filial,
        'teacher_chair' => $teacher_chair,
        'teacher_post' => $teacher_post,
        'teacher_rate' => $teacher_rate,
        'teacher_post_sub' => $teacher_post_sub,
        'teacher_rate_sub' => $teacher_rate_sub,
        'teacher_zk' => $teacher_zk
    )
);
$this->Db->queryPrepared($query);
$query = array(
    'text' => "SELECT id
                FROM users
                WHERE user_name = :user_name
                LIMIT 1",
    'data' => array (
        'user_name' => $user_name
    ),
    'limit1' => true, 'select' => true
);
$user = $this->Db->queryPrepared($query);
F::redirect('?com=administrator&view=teacher&id='.$user['id']);
}
else{
    F::redirect('?com=administrator&view=teacher_add');
}

}

public function action_teacher_edit(){
    if ( ($this->teacherEditId = F::getVar('id')) and (is_numeric($this->teacherEditId))
) {
        $query = array(
            'text' => "SELECT *
                        FROM users
                        WHERE id = :id
                        AND user_type = 2
                        LIMIT 1",
            'data' => array (
                'id' => $this->teacherEditId
            ),
            'limit1' => true, 'select' => true
        );
        if ($teacher = $this->Db->queryPrepared($query)){
            $query = "SELECT * FROM filials";
            $filials = $this->Db->query($query);
            $query = "SELECT * FROM chairs";
            $chairs = $this->Db->query($query);
            include(COMPATH.'com_administrator/teacher_edit_form.php');
        }
        else {
            F::redirect('?com=administrator&view=teachers');
        }
    }
}

```

```

    }
}
else {
    F::redirect('?com=administrator&view=teachers');
}
}

public function action_post_teacher_edit(){
    print_r($_POST);
    $user_id = F::getVar('id');
    $user_name = F::getVar('user_name');
    $user_login = F::getVar('user_login');
    $teacher_filial = F::getVar('teacher_filial');
    $teacher_chair = F::getVar('teacher_chair');
    $teacher_post = F::getVar('teacher_post');
    $teacher_rate = F::getVar('teacher_rate');
    $teacher_post_sub = F::getVar('teacher_post_sub');
    $teacher_rate_sub = F::getVar('teacher_rate_sub');
    $teacher_zk = F::getVar('teacher_zk');

    if ($user_id || $user_name || $user_login) {
        $query = array(
            'text' => "UPDATE `users` SET
                user_name = :user_name,
                user_login = :user_login,
                teacher_filial = :teacher_filial,
                teacher_chair = :teacher_chair,
                teacher_post = :teacher_post,
                teacher_rate = :teacher_rate,
                teacher_post_sub = :teacher_post_sub,
                teacher_rate_sub = :teacher_rate_sub,
                teacher_zk = :teacher_zk
            WHERE id=:id AND user_type = 2",
            'data' => array (
                'id' => $user_id,
                'user_name' => $user_name,
                'user_login' => $user_login,
                'teacher_filial' => $teacher_filial,
                'teacher_chair' => $teacher_chair,
                'teacher_post' => $teacher_post,
                'teacher_rate' => $teacher_rate,
                'teacher_post_sub' => $teacher_post_sub,
                'teacher_rate_sub' => $teacher_rate_sub,
                'teacher_zk' => $teacher_zk
            )
        );
        $this->Db->queryPrepared($query);
        F::redirect('?com=administrator&view=teacher&id='.$user_id);
    }
    else{
        F::redirect('?com=administrator&view=teachers');
    }
}
}

```

```

public function action_teacher_delete(){
    $teacherId = F::getVar('id');
    if($teacherId){
        $query = array(
            'text' => "DELETE FROM users
                                WHERE id = :id
                                AND user_type = 2
                                LIMIT 1",
            'data' => array (
                'id' => $teacherId
            )
        );
        $this->Db->queryPrepared($query);
        F::redirect('?com=administrator&view=teachers');
    }
}

public function action_teacher_password_change(){
    if ( ($this->teacherId = F::getVar('id')) and (is_numeric($this->teacherId)) ) {
        $query = array(
            'text' => "SELECT user_name
                                FROM users
                                WHERE id = :id
                                AND user_type = 2
                                LIMIT 1",
            'data' => array (
                'id' => $this->teacherId
            ),
            'limit1' => true, 'select' => true
        );
        if ($teacher = $this->Db->queryPrepared($query)){

include(COMPATH.'com_administrator/teacher_changepassword_form.php');
        }
        else {
            F::redirect('?com=administrator&view=teachers');
        }
    }
    else {
        F::redirect('?com=administrator&view=teachers');
    }
}

public function action_post_teacher_password_change(){
    print_r($_POST);
    $user_id = F::getVar('id');
    $new_password = F::getVar('new_password');

    if ($user_id || $new_password){
        $new_password = md5($new_password);
        $query = array(
            'text' => "UPDATE users
                                SET user_password=:new_password

```

```

WHERE id=:id",
    'data' => array (
        'new_password' => $new_password,
        'id' => $user_id
    )
);
$this->Db->queryPrepared($query);
F::redirect('?com=administrator&view=teacher&id='.$user_id);
}
else {
    //F::redirect('?com=administrator&view=teachers');
}
}
}

```

Исполняемый скрипт PHP – application\components\com\_teacher\component.php

```

<?php
defined('_A') or die;
//tests

class Component extends appComponent{
    public $viewList = array(
        'post_edit' => 1,
    );

    public function action_index(){
        $posts = array(
            1 => "Ассистент",
            2 => "Старший преподаватель",
            3 => "Доцент",
            4 => "Профессор"
        );
        $zk = array(
            1 => "Заведующий выпускающей кафедрой",
            2 => "Заведующий невыпускающей кафедрой"
        );
        $indexesFiles = array(
            1 => "assistant",
            2 => "senior_lecturer",
            3 => "docent",
            4 => "professor"
        );
        $umr = "";
        $nir = "";
        $pvor = "";
        $iya = "";
        $pb = "";
        $summ_p = "";
        $total_p = "";
    }
}

```



```

$zv = "";
$total_zv = "";
$zn = "";
$total_zn = "";
$indexesFileName = $indexesFiles[$this->user['teacher_post']].'_form.php';
if($this->user['teacher_indexes']){
    $teacher_indexes = json_decode($this->user['teacher_indexes']);
    $sumr = $teacher_indexes->umr3 + $teacher_indexes->umr4 +
$teacher_indexes->umr5 + $teacher_indexes->umr6 + $teacher_indexes->umr7 +
$teacher_indexes->umr8;
    $nir = $teacher_indexes->nir2 + $teacher_indexes->nir3 +
$teacher_indexes->nir4 + $teacher_indexes->nir5 + $teacher_indexes->nir6 + $teacher_indexes-
>nir7 + $teacher_indexes->nir8 + $teacher_indexes->nir9 + $teacher_indexes->nir10 +
$teacher_indexes->nir11 + $teacher_indexes->nir12;
    $pvor = $teacher_indexes->pvor3 + $teacher_indexes->pvor4 +
$teacher_indexes->pvor5;
    $iya = $teacher_indexes->iya2 + $teacher_indexes->iya3 +
$teacher_indexes->iya4 + $teacher_indexes->iya5 + $teacher_indexes->iya6 + $teacher_indexes-
>iya7 + $teacher_indexes->iya8 + $teacher_indexes->iya9 + $teacher_indexes->iya10 +
$teacher_indexes->iya11 + $teacher_indexes->iya12;
    $pb = $teacher_indexes->pb1 + $teacher_indexes->pb2;
    $summ_p = $sumr + $nir + $pvor + $iya + $pb;
    $total_p = $summ_p / ( $this->user['teacher_rate'] + $this-
>user['teacher_rate_sub'] );

    $zv = $teacher_indexes->zv3 + $teacher_indexes->zv4 + $teacher_indexes-
>zv5 + $teacher_indexes->zv6 + $teacher_indexes->zv7 + $teacher_indexes->zv8 +
$teacher_indexes->zv9 + $teacher_indexes->zv10 + $teacher_indexes->zv11 + $teacher_indexes-
>zv12 + $teacher_indexes->zv13 + $teacher_indexes->zv14 + $teacher_indexes->zv15;
    $total_zv = $zv / ( $this->user['teacher_rate'] + $this-
>user['teacher_rate_sub'] );

    $zn = $teacher_indexes->zn3 + $teacher_indexes->zn4 + $teacher_indexes-
>zn5 + $teacher_indexes->zn6 + $teacher_indexes->zn7 + $teacher_indexes->zn8;
    $total_zn = $zn / ( $this->user['teacher_rate'] + $this-
>user['teacher_rate_sub'] );

    $total = $total_p + $total_zv + $total_zn;
}

$query = array(
    'text' => "SELECT *
                                FROM filials
                                WHERE id = :id
                                LIMIT 1",
    'data' => array (
        'id' => $this->user['teacher_filial']
    ),
    'limit1' => true, 'select' => true
);
$filial = $this->Db->queryPrepared($query);
$query = array(
    'text' => "SELECT *
                                FROM chairs

```

```

                WHERE id = :id
                LIMIT 1",
            'data' => array (
                'id' => $this->user['teacher_chair']
            ),
            'limit1' => true, 'select' => true
        );
        $chair = $this->Db->queryPrepared($query);
        include(COMPATH.'com_teacher/'.$indexesFileName);
    }

    public function action_post_edit(){
        $indexes = F::getVar('indexes');
        $indexes = json_encode($indexes);
        if ($indexes) {
            print_r($indexes);
            $query = array(
                'text' => "UPDATE users
                                SET teacher_indexes = :teacher_indexes
                                WHERE id = :id
                                LIMIT 1",
                'data' => array (
                    'id' => $this->userId,
                    'teacher_indexes' => $indexes
                )
            );
            $this->Db->queryPrepared($query);
        }
        F::home();
    }
}
}

```

Исполняемый скрипт PHP – application\components\com\_user\component.php

```

<?php
defined('_A') or die;
//user
class Component extends appComponent{
    public $viewList = array(
        'login' => 1,
        'post_login' => 1,
        'logout' => 1,
        'profile' => 1,
        'change_password' => 1,
        'post_change_password' => 1
    );

    public function action_login(){
        $this->loginFormLogin = $_COOKIE['login'];
        include(COMPATH.'com_users/login_form.php');
    }
}

```

```

public function action_post_login(){
    setcookie('login','');
    $login = F::getVar('userlogin');
    $pass = F::getVar('userpassword');
    $result = true;
    if (F::getVar('submit')){
        if (!$login or (!preg_match("/^[0-9a-zA-Z_-]+$/",$login))) {
            F::setMsg('LoginWrong',2);
            $result = false;
        }
        if (!$pass or (!preg_match("/^[0-9a-zA-Z_-]+$/",$pass))) {
            F::setMsg('LoginWrong',2);
            $result = false;
        }
    }
    if ($result) {
        $password = md5($pass);
        $query = array(
            'text' => "SELECT id
                                FROM users
                                WHERE user_login = :login
                                AND user_password = :password
                                LIMIT 1",
            'data' => array (
                'login' => $login,
                'password'=> $password
            ),
            'limit1' => true, 'select' => true
        );
        if ($row = $this->Db->queryPrepared($query)) {
            $hash = md5(F::getHashCode(10));
            $id = $row['id'];
            echo '<br>id='.$id;
            $query = array(
                'text' => "UPDATE users
                                SET user_hash=:hash
                                WHERE id=:id",
                'data' => array (
                    'hash' => $hash,
                    'id' => $id
                )
            );
            setcookie('id',$id);
            setcookie('hash',$hash);
            $this->Db->queryPrepared($query);
            F::redirect(BASEURL);
            exit;
        }
        else {
            F::setMsg('LoginWrong',2);
            $result = false;
        }
    }
}

```

```

    }
    if (!$result) {
        setcookie('login',$login);
        F::redirect('?com=users&view=login');
    }
}

public function action_logout(){
    setcookie('id','');
    setcookie('hash','');
    echo 'logout';
    F::redirect('?com=users&view=login');
}

public function action_profile(){
    include(COMPATH.'com_users/profile_page.php');
}

public function action_change_password(){
    include(COMPATH.'com_users/changepassword_form.php');
}

public function action_post_change_password(){
    if (F::getVar('submit')){
        $result = true;
        $currentpassword = F::getVar('currentuserpassword');
        $newpassword = F::getVar('newuserpassword');
        $confirmpassword = F::getVar('confirmuserpassword');
        if (!$currentpassword or (!preg_match("/^[0-9a-zA-Z_-
]+$/", $currentpassword))) {
            F::setMsg('CurrentPasswordWrong',2);
            $result = false;
        }
        if (!$newpassword or (!preg_match("/^[0-9a-zA-Z_-]+$/", $newpassword)))
        {
            F::setMsg('NewPasswordWrong',2);
            $result = false;
        }
        if (!$confirmpassword or (!preg_match("/^[0-9a-zA-Z_-
]+$/", $confirmpassword))) {
            F::setMsg('ConfirmPasswordWrong',2);
            $result = false;
        }
        if ($result and ($newpassword !== $confirmpassword)){
            F::setMsg('PasswordsWrong',2);
            $result = false;
        }
        if ($result and ($newpassword === $currentpassword)){
            F::setMsg('CNPasswordWrong',2);
            $result = false;
        }

        if ($result){
            $currentpassword = md5($currentpassword);

```

```

$newpassword = md5($newpassword);
$query = array(
    'text' => "SELECT id, user_password
                FROM users
                WHERE id = :id
                AND user_password = :password
                LIMIT 1",

    'data' => array (
        'id' => $this->userId,
        'password'=> $currentpassword
    ),
    'limit1' => true, 'select' => true
);
if ($row = $this->Db->queryPrepared($query)) {
    $query = array(
        'text' => "UPDATE users
                    SET user_password=:password
                    WHERE id=:id",

        'data' => array (
            'password' => $newpassword,
            'id' => $this->userId
        )
    );
    $this->Db->queryPrepared($query);
    F::setMsg('PasswordChanged');
    F::redirect(BASEURL);
}
else {
    F::setMsg('CurrentPasswordWrong',2);
    F::redirect('?com=users&view=change_password');
}
else {F::redirect('?com=users&view=change_password');}
}
else {
    F::redirect(BASEURL);
}
}
}

```

### 3.4. Разработка руководства программиста

Установка веб-сервера на локальный компьютер под управлением операционной системой Microsoft Windows.

Для установки веб-сервера запустите исполняемый файл на компакт диске: `xampp-win32-5.6.20-0-VC11-installer.exe`. После запуска появится предупреждение о том, что на процесс установки может повлиять установленный в системе антивирус. Нажимаем кнопку «ОК» (рисунок 11).

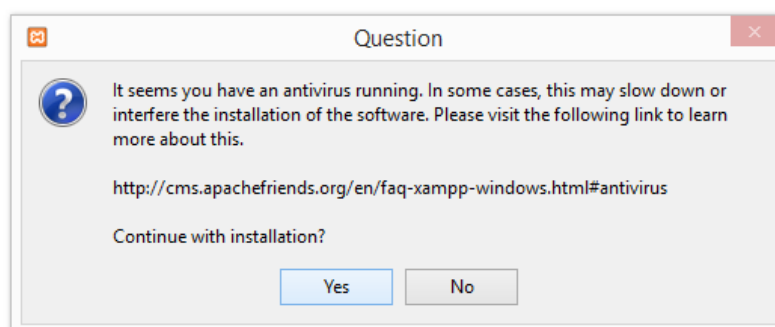


Рисунок 11 – Предупреждение перед установкой

Появляется приветствие программы установки – нажимаем «Next» (рисунок 12).

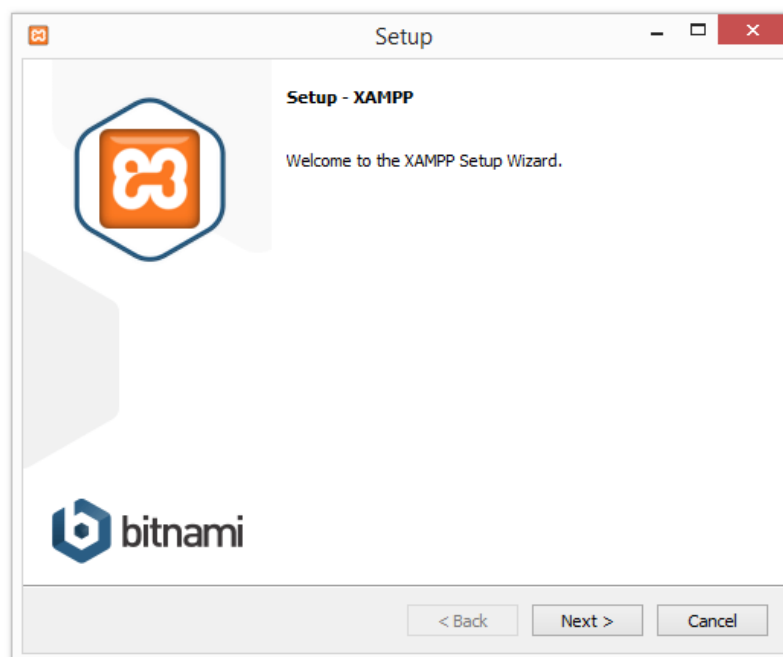


Рисунок 12 – Приветствие программы установки

В следующем окне выбора устанавливаемых модулей веб сервера (рисунок 13) достаточно оставить галочки напротив следующих модулей: Apache, MySQL, PHP и PHPMyAdmin и нажать кнопку «Next».

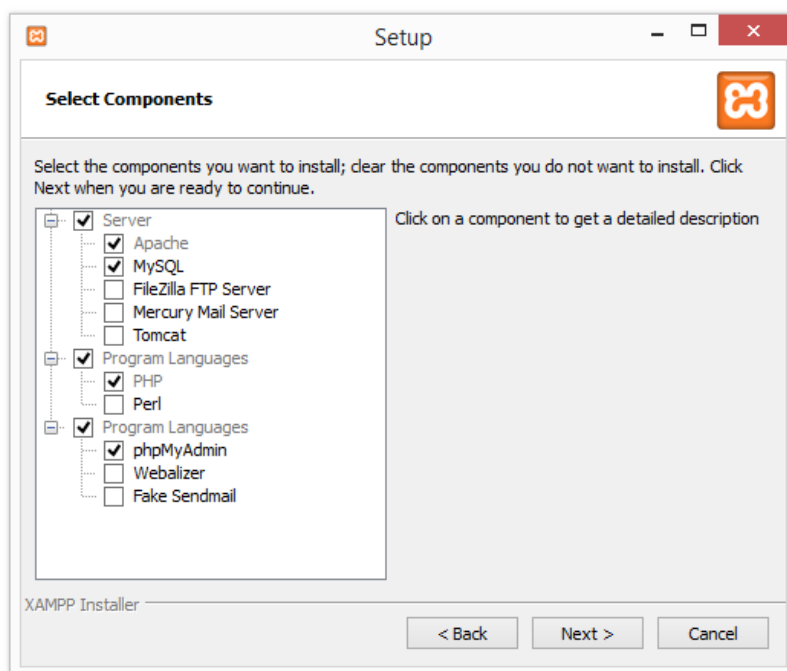


Рисунок 13 – Окно выбора устанавливаемых модулей

Во всех следующих окнах установки необходимо нажимать «Next» и дожидаться окончания установки. Веб-сервер установлен. Теперь нужно его запустить через ярлык с названием «XAMPP», который появится на рабочем столе. Откроется окно управления модулями веб-сервера (рисунок 14). Необходимо запустить модули Apache и MySQL нажатием на кнопки «Start» напротив каждого из модулей.

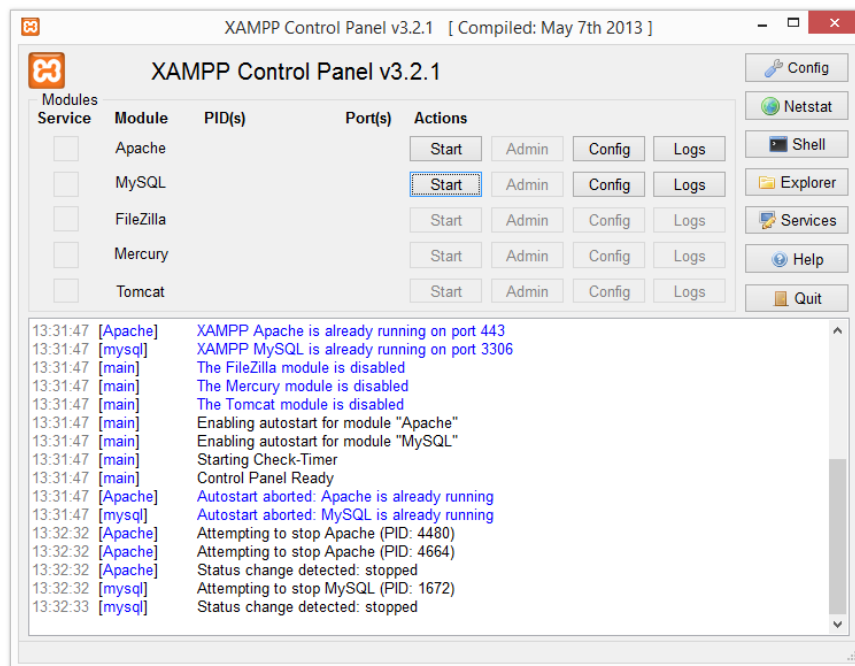


Рисунок 14 – Окно управления модулями веб-сервера

После этого нужно создать базу данных для программного комплекса.

Напротив модуля MySQL нажимаем кнопку «Admin». Откроется средство администрирования PHPMyAdmin (рисунок 15).

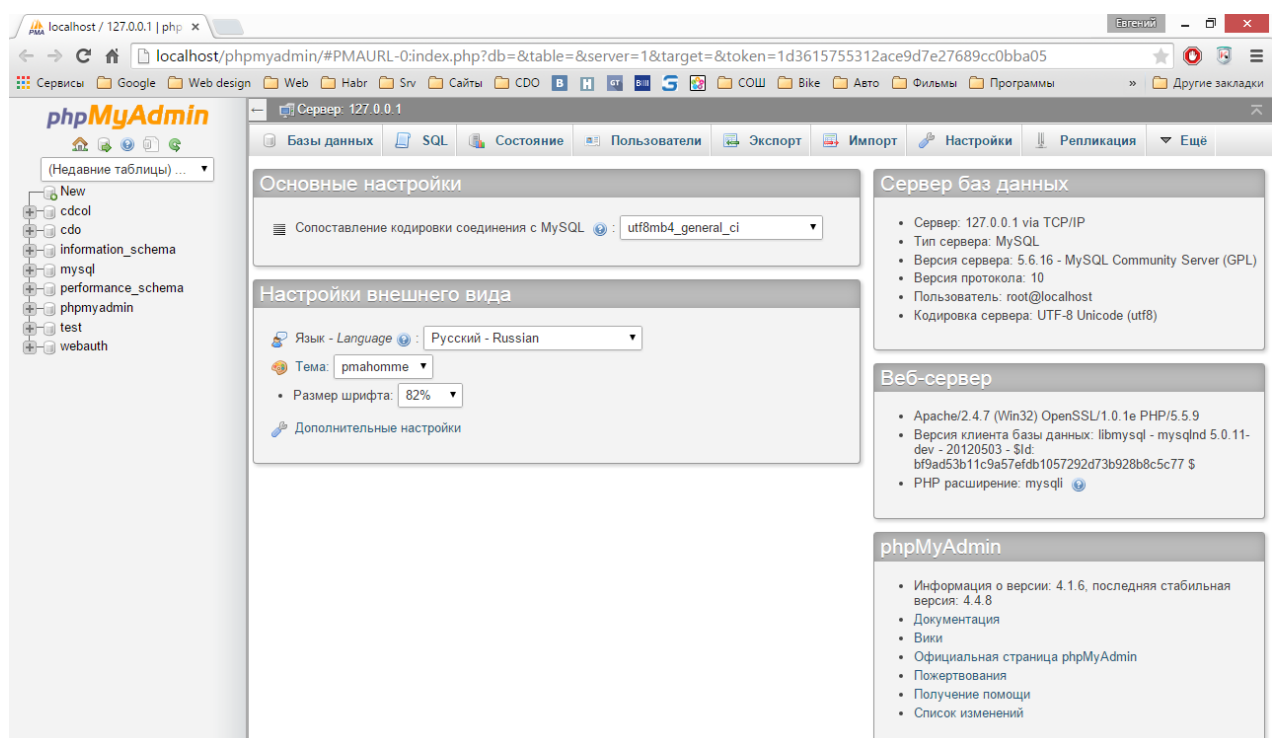


Рисунок 15 – Окно PHPMyAdmin



Для создания новой базы данных нажимаем ссылку «New» над списком существующих баз данных слева. В открывшемся окне создания базы данных (рисунок 16) необходимо ввести название базы данных, выбрать тип сравнения utf8\_general\_ci и нажать кнопку «ОК».

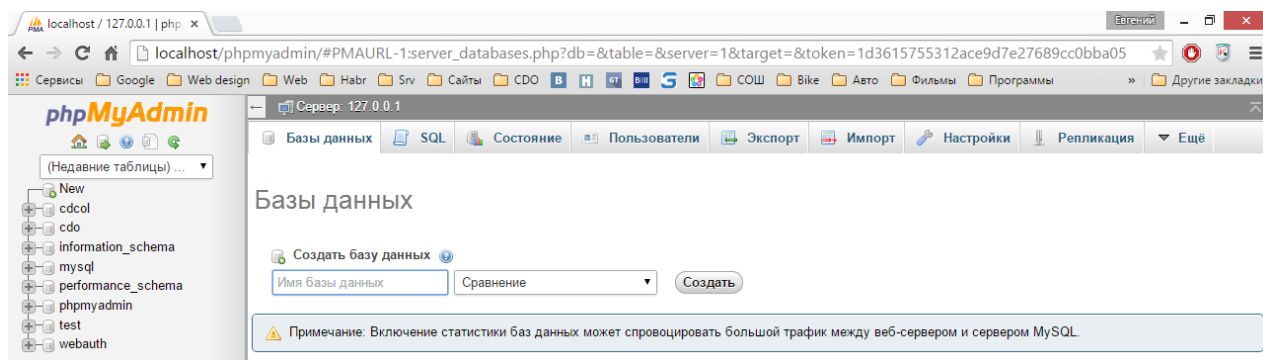
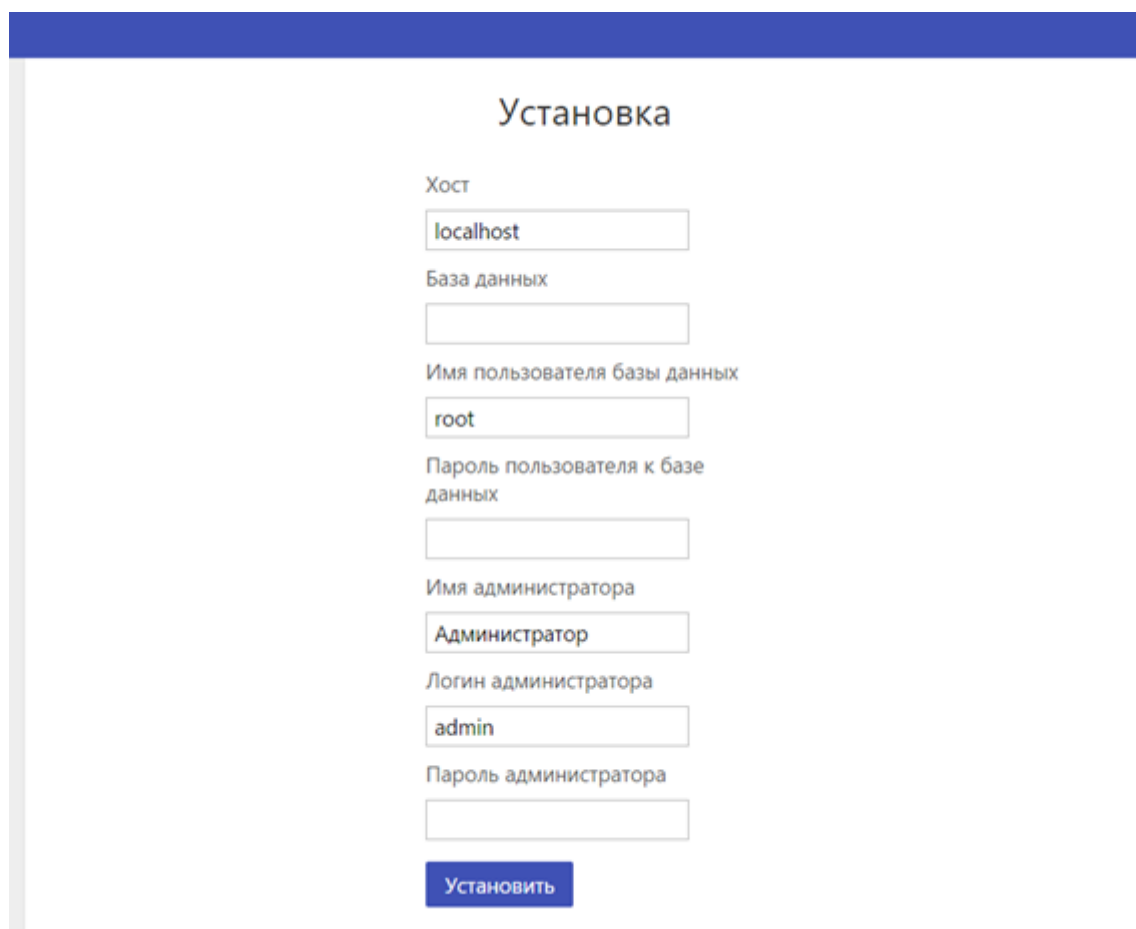


Рисунок 16 – Окно создания базы данных

Далее открываем через Проводник папку C:\xampp\htdocs и копируем в нее папку с компакт-диска «diplom». Для создания таблиц в базе данных запускаем веб-браузер и вводим адрес: <http://localhost/diplom/install/>. Откроется страница установки (рисунок 17).



Установка

Хост

База данных

Имя пользователя базы данных

Пароль пользователя к базе данных

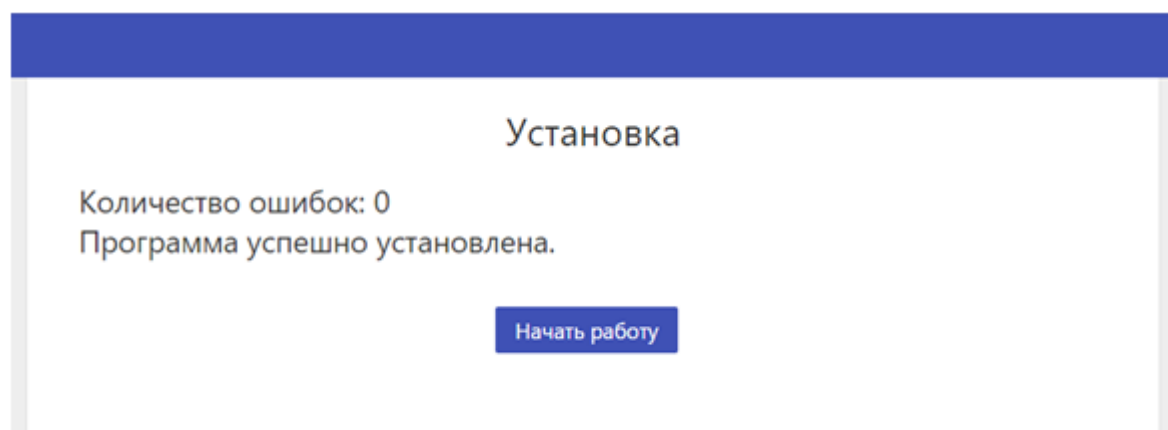
Имя администратора

Логин администратора

Пароль администратора

Рисунок 17 – Страница установки

В поле «База данных» введите имя созданной ранее базы данных. В поле «Пароль администратора» задайте пароль для администратора информационной системы. После этого необходимо нажать кнопку «Установить». После установки появится страница с сообщением об успешной установке (рисунок 18), на которой нажимаем кнопку «Начать работу». В результате откроет страница авторизации (<http://localhost/diplom/>) (рисунок 19). Программа готова к работе.



Установка

Количество ошибок: 0  
Программа успешно установлена.

Рисунок 18 – Завершение установки

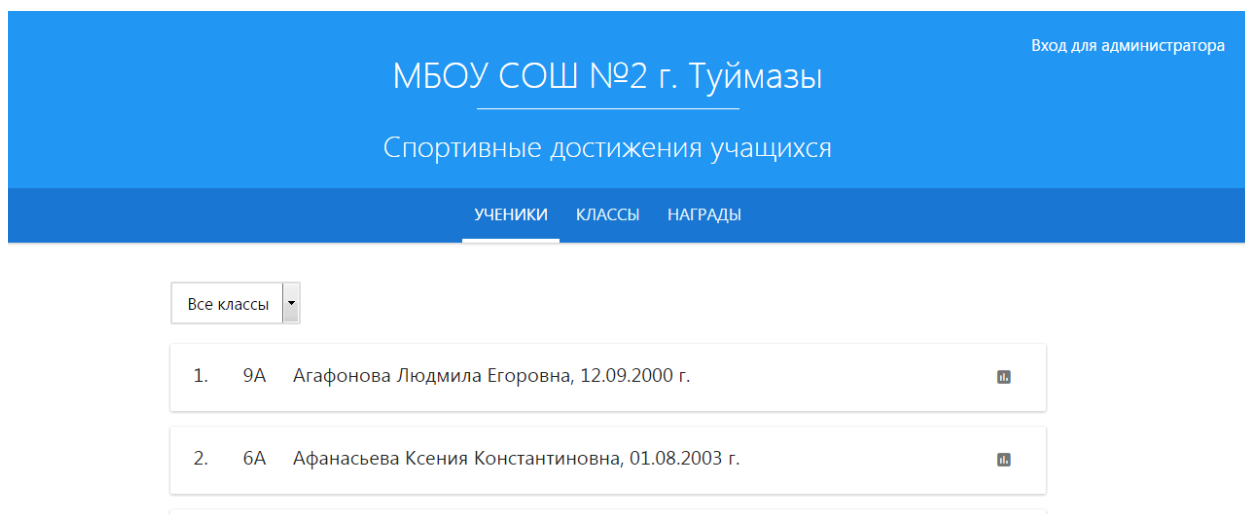


Рисунок 19 – Главная страница

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была разработана автоматизированная информационная система учета спортивных достижений школьников. Приложение является полностью готовым к работе.

В первом разделе выпускной квалификационной работы была поставлена проблема, а также выявлены различные способы ее решения. Из представленных методов решения был выбран наиболее подходящий метод.

Во втором разделе выпускной квалификационной работы была разработана автоматизированная информационная система учета спортивных достижений школьников, а также были составлены инструкции пользователя и администратора программы.

Цель выпускной квалификационной работы, разработка автоматизированной информационной системы учета спортивных достижений школьников, выполнена.

Разработанная информационная система имеет большую практическую ценность для автоматизации учета спортивных достижений школьников.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) PHP: Правильный Путь [Электронный ресурс]. – URL: <http://getjump.me/ru-php-the-right-way/>
- 2) Адам Фримен. jQuery для профессионалов. – М.: Вильямс, 2013.–960 с.
- 3) Бретт Маклафлин. PHP5 на практике. – Спб.:ПИТЕР, 2016. – 398 с.
- 4) Джейсон Ленгсторф. PHP и jQuery для профессионалов. – М.: Вильямс: Изд-во ИГЭА, 2011. – 362 с.
- 5) Колисниченко Д. К. Профессиональное программирование на PHP. – СПб.:БХВ-Петербург, 2013.–416 с.
- 6) Колисниченко Д. К. PHP и MySQL. Разработка веб-приложений. – М.: ДМК Пресс, 2014. – 323 с.
- 7) Кузнецов М. PHP 5/6. – СПб.:БХВ-Петербург, 2015.–870 с.
- 8) Люк Веллинг, Лаура Томсон. Разработка веб-приложений с помощью PHP и MySQL. – М.:Вильямс, 2013. – 848 с.
- 9) Основы работы с событиями [Электронный ресурс]/Современный учебник Javascript. – URL: <http://learn.javascript.ru/events-and-interfaces>
- 10) Руководство по PHP [Электронный ресурс]. – URL: <http://php.net/manual/ru/>
- 11) Руководство по PHP [Электронный ресурс]. – URL: <https://secure.php.net/manual/ru/index.php>
- 12) Справочник по CSS[Электронный ресурс]. – URL: <http://htmlbook.ru/css>
- 13) Справочное руководство по MySQL[Электронный ресурс]. – URL: <http://www.mysql.ru/docs/man/>
- 14) Учебник по PHP. Самоучитель, справочник PHP [Электронный ресурс]. – URL: [http://www.softtime.ru/bookphp/gl1\\_1.php](http://www.softtime.ru/bookphp/gl1_1.php)

15) Фленов М. PHP и MySQL Исчерпывающее руководство. – М.: Вильямс, 2011.–565 с.