

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Анализ задания на выпускную квалификационную работу	6
1.1 Анализ существующих систем	6
1.1.1 Портал олимпиады школьников «Ломоносов».....	6
1.1.2 Сайт олимпиады НИЯУ МИФИ.....	7
1.1.3 Сайт олимпиады высшей школы экономики.....	8
1.2 Выбор средств разработки	9
1.2.1 Язык программирования PHP	9
1.2.2 Язык программирования JavaScript.....	10
1.2.3 Язык разметки HTML	11
1.2.4 Язык программирования CSS	12
1.2.5 База данных MySQL	13
1.3 Обоснование выбранных средств разработки.....	14
1.4 Техническое задание.....	15
1.4.1 Наименование системы	15
1.4.2 Назначение и цели создания системы.....	15
1.4.2.1 Назначение системы	15
1.4.2.2 Цель создания системы	16
1.4.3 Требования к системе	16
1.4.3.1 Требования к структуре и функциональной части системы	16
1.4.3.2 Требования к разделению доступа.....	16
1.4.3.3 Требования по эргономике и технической эстетике	17
1.4.4 Требования к видам обеспечения	17
1.4.4.1 Требования к лингвистическому обеспечению	17
1.4.4.2 Требования к программному обеспечению.....	17
1.5 Итоги анализа	18
2 Разработка архитектуры и основных технических решений	19
2.1 Общая архитектура системы.....	19
2.2 Описание структуры базы данных	21

2.3 Работа системы с СУБД	23
2.4 Разработка интерфейса печати документов	24
2.5 Анализ существующих способов печати документа.....	28
2.5.1 Печать HTML страницы	28
2.5.2 Работа с Word-шаблонами посредством XML.....	28
2.5.3 Генераторы отчетов	29
2.5.4 Библиотека odtPHP	29
2.6 Разработка функции печати документов	30
2.6.1 Разработка шаблонов odt.....	30
2.6.2 Функция генерации документов.....	33
2.7 Итоги разработки	37
3 Руководство пользователя	38
3.1 Руководство администратора по разворачиванию системы	38
3.1.1 Разворачивание базы данных и файлов на сервере.....	38
3.1.2 Шаблоны документов.....	40
3.2 Руководство пользователя по работе с системой	40
3.2.1 Вход в панель администратора.....	40
3.2.2 Печать документов.....	43
3.2.3 Демонстрация сформированных документов	44
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	48

ВВЕДЕНИЕ

Современный мир немало изменился за последние десятилетия – многое из того что есть сейчас еще недавно казалось просто фантастикой. В век информационных технологий вы уже вряд ли встретите человека, который не знаком с компьютером, смартфоном или другими распространенными повсеместно приспособлениями. Нужно что-то узнать – пожалуйста, интернет всегда под рукой. Необходимо узнать номер или адрес нужной вам организации – и здесь google поможет.

С 1995 года на базе ФИВТ, а позже и на базе ИКИТа начала проводиться олимпиада по АСМ программированию, и с каждым годом учет и регистрация участников, становится все более утомляющей, потому как все приходится делать вручную.

Именно поэтому актуальность выбранной выпускной работы не вызывает сомнений. Когда под рукой интернет, человеку гораздо проще зайти на сайт, узнать всю нужную информацию и удаленно зарегистрироваться, если в том возникнет необходимость, нежели терять время на путь до места проведения олимпиады и регистрироваться вручную.

Объектом выпускной работы являются интернет технологии и способы их взаимодействия с пользователями. Предмет ВКР заключается в разработке веб-сервера для удобной регистрации, учета и обновления информации для участников олимпиад.

Конечная цель данной выпускной работы – разработать действующий сайт, с помощью которого участники олимпиады по программированию будут иметь возможность подать заявку на участие в олимпиаде, или узнать последние новости по ее проведению. Также на сайте должна быть функция автоматического создания и просмотра различных документов: списки участников, бейджики и т.д.

В связи с указанной целью возникают следующие задачи:

- изучить материалы по данной области;

- рассмотреть наилучшие варианты разработки будущего сайта;
- создать базу данных для хранения информации;
- разработать структурную модель будущего сайта.

Создание такого сайта обеспечит автоматизацию во многих пунктах проведения олимпиады по программированию АСМ, что позволит значительно сэкономить время, как для участников олимпиад, так и для тех, кто эти олимпиады проводит.

1 Анализ задания на выпускную квалификационную работу

В данной выпускной работе необходимо разработать веб-сайт для участников олимпиад по АСМ программированию, который должен включать в себя следующие функции:

- Возможность публикации различных материалов по проведению олимпиады (новостей, объявлений);
- Возможность регистрации участников олимпиады;
- Внесение данных об участниках олимпиады в базу данных сайта;
- Возможность автоматически формировать документы по участникам для организационного комитета;

Олимпиады для студентов и школьников проводятся отдельно. Олимпиады могут быть командными (тренер + 3 участника), или индивидуальными.

1.1 Анализ существующих систем

Сейчас существует множество разнообразных сайтов для проведения различных олимпиад. Причем многие из них имеют совершенно разную структуру и способы регистрации участников. На некоторых сайтах такие функции и вовсе не реализованы. Ниже приведены примеры подобных сайтов:

- Портал олимпиады школьников «Ломоносов»;
- Сайт олимпиады «Абитуриент-студент» НИЯУ МИФИ;
- Сайт олимпиады Высшей школы экономики.
- Единая система регистрации на олимпиады «Мега-Талант»;

1.1.1 Портал олимпиады школьников «Ломоносов»

Сайт олимпиады «Ломоносов» предназначен для проведения различных олимпиад среди школьников. На этом сайте ученики школ со всей России

могут испытать свои знания по любым общеобразовательным предметам, начиная с английского и заканчивая математикой или литературой.

На сайте предусмотрена публикация новостей, просмотр различных документов и архивов олимпиад, а также регистрация участников. На рисунке 1 представлена главная страница сайта [2].

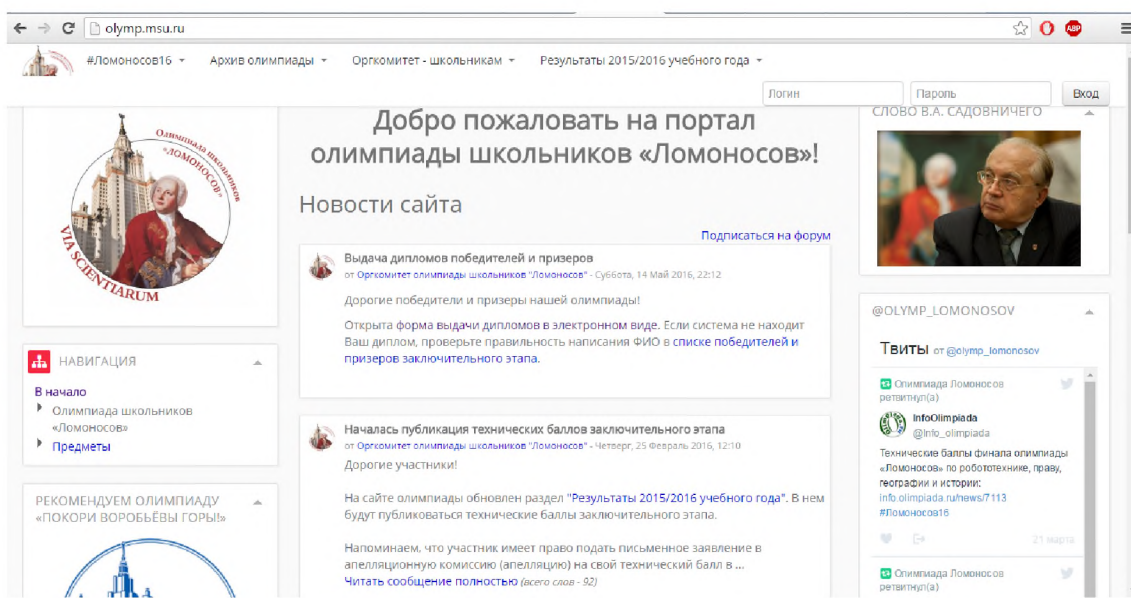


Рисунок 1 – Главная страница сайта

1.1.2 Сайт олимпиады НИЯУ МИФИ

Данный сайт принадлежит Национальному исследовательскому ядерному университету. Основной его целью является помощь при регистрации на олимпиады для абитуриентов, поступающих в этот университет.

На сайте имеется не так много функций, однако основной все же является регистрация участников. На рисунке 2 можно увидеть форму регистрации участников [3].

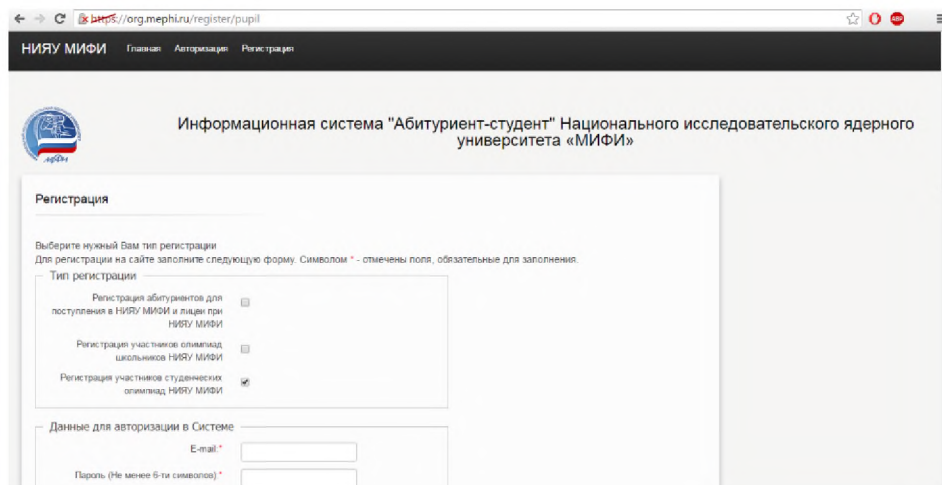


Рисунок 2 – Форма регистрации участников

1.1.3 Сайт олимпиады высшей школы экономики

На данном сайте Высшей школы экономики присутствует раздел выбора проводимых олимпиад с последующей регистрацией. Как и предыдущие сайты, он предназначен для удобного донесения информации до участников о проводимых олимпиадах и помощи оргкомитету в их проведении. На рисунке 3 изображена страница регистрации участников [4].

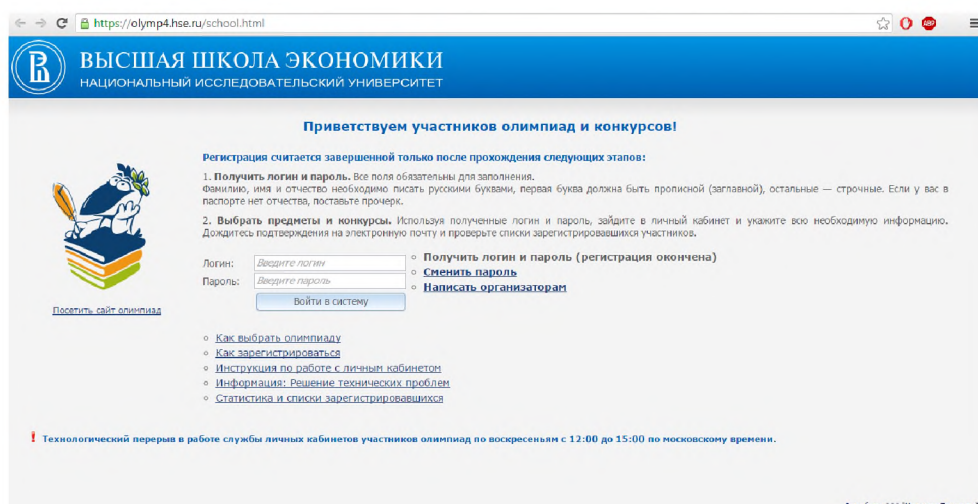


Рисунок 3 – Страница регистрации участников

1.2 Выбор средств разработки

Существует множество различных языков программирования и программных продуктов, которые используются для разработки, оптимизации или отладки сайтов. Многие из них известны любому, кто хоть раз занимался созданием сайтов, или хотя бы пробовал работать с ними. Другие программные продукты известны лишь узкому кругу пользователей. Тем не менее, все они служат для одной цели – разработать веб-сервер, в котором нуждается заказчик или пользователь для решения определенных типов задач.

Особняком среди них держатся такие языки как HTML, PHP и JavaScript. А причиной тому является то, что именно эти языки считаются основными при создании практически любого сайта в наши дни.

1.2.1 Язык программирования PHP

Аббревиатура PHP с английского расшифровывается как Hypertext Preprocessor, или на русском Препроцессор Гипертекста. PHP представляет из себя язык программирования общего назначения, с открытым исходным кодом, предназначенный для разработки и написания различных скриптов или сценариев, выполняемых на Web-сервере.

Среди достоинств этого языка на первом месте, несомненно, окажется такой пункт как простота в освоении. Для программиста уже знакомого с другими языками не составит большого труда писать простейшие PHP-скрипты уже после пары часов изучения языка. Вторым главным достоинством является то, что язык PHP разработан специально для web-программирования, а потому его код может внедряться непосредственно в html-код. Иными словами, программисту не нужно создавать отдельные файлы для своих сценариев, так как он может добавлять их напрямую в html. Код PHP можно определить по специальным тегам, которые позволяют переключаться между -PHP и -HTML

кодом. Выглядят они следующим образом: `<?php` – начальный тег, `?>` – конечный тег.

В отличие от JavaScript, скрипты PHP выполняются на сервере, после чего генерируют HTML-файл, посылаемый клиенту. Это очень удобно, потому что благодаря этому PHP не зависит от скорости компьютера, или браузера, который использует пользователь, и полностью выполняется на сервере. Порой пользователь даже не знает, получает ли он обычный файл или результат скрипта.

PHP включает в себя практически все операторы и функции стандартных библиотек C, что делает его еще более привлекательным для программистов. Во многом язык PHP также популярен, благодаря своей связи с базой данных MySQL. Именно поэтому многие пользователи выбирают PHP для создания своих сайтов [5, 6, 7].

1.2.2 Язык программирования JavaScript

Основная функция JavaScript это возможность делать web-страницы живыми. Другими словами, JavaScript это язык программирования, который позволяет писать скрипты, делающие из скучной, неподвижной страницы живую и интерактивную. Такие скрипты напрямую подключаются к HTML-файлу и выполняются сразу после загрузки страницы.

Java-скрипты представляют из себя простой текст и не требуют каких-либо дополнительных приготовлений. Их основная особенность в том, что они могут выполняться где угодно, лишь бы на выполняемом устройстве или программном обеспечении имелся интерпретатор. Интерпретатор анализирует поступающий исходный код программы, если нужно преобразует его, а затем выполняет. Современные интерпретаторы перед выполнением обычно преобразуют JavaScript в машинный код, что позволяет ему работать очень быстро.

Так как JavaScript был с самого начала ориентирован на браузеры, соответственно в них он может творить все что угодно, касаясь манипуляций со страницами, взаимодействием с пользователями и сервером. Он может менять стили элементов, скрывать или показывать элементы, обрабатывать клики и перемещения мыши, посылать запросы на сервер и выполнять еще много других полезных функций. Именно поэтому JavaScript является самой распространенной на сегодняшний день технологией разработки браузерных интерфейсов. Ведь она по умолчанию включена и поддерживается всеми известными браузерами [8].

1.2.3 Язык разметки HTML

Язык HTML относится скорее не к языкам программирования, а к языкам разметки. Данная аббревиатура расшифровывается как HyperText Markup Language, или «язык гипертекстовой разметки». Это значит, что он указывает браузеру, каким образом должны располагаться элементы на загруженной странице. Под элементами подразумеваются все размещаемые на странице объекты: тексты, рисунки, видео и так далее.

Для взаимодействия с элементами HTML использует специальные теги. Теги можно охарактеризовать как кирпичики, из которых состоит страница. Они нужны, например, для того чтобы сделать текст жирным, выделить его курсивом, сделать его заголовком или начать предложение с новой строки. Каждому такому действию соответствуют различные теги, причем каждый тег может иметь определенный набор атрибутов. Каждый тег начинается символом < и заканчивается символом >, например: <p>. Они делятся на парные и одиночные, и в закрывающем теге используется символ /. Парные теги используют для оформления некоторого участка текста. С помощью этих тегов можно указать начало и конец редактируемого участка.

Однако есть теги, не предназначенные для оформления фрагментов текста. К примеру, тег для вставки изображения. Он добавляет на страницу

одиначный объект, а потому ему не нужно заключать в себя какой-то текст. Именно такие теги называют одиночными. Примерами подобных тегов могут служить: `
`, `<hr>`, ``.

Таким образом, язык HTML является одним из главных средств при разработке веб-сайта [9, 10].

1.2.4 Язык программирования CSS

При создании веб-страниц используются два языка: HTML и CSS. В данной связке HTML берет на себя структуру и содержание, а CSS отвечает за оформление. Объединяя между собой HTML- и CSS- код, браузер формирует внешний вид страницы.

CSS, по сути, является языком для управления внешним видом HTML-документа. С его помощью можно задавать параметры отображения любого тега: ширину, высоту, отступы, цвет, размер шрифта, фон и так далее. Аббревиатура расшифровывается как Cascading Style Sheets, что в переводе означает «каскадные таблицы стилей», но обычно CSS называют просто «стилями». CSS состоит из селекторов и свойств. Селекторы отвечают за то, какие конкретные элементы или группы элементов будут обладать заданными для них свойствами.

Существует два способа задания стилей для тегов с использованием CSS. В первом случае можно прописать дополнительный атрибут `<style>` внутри тега в HTML-файле. В таком случае заданный стиль будет применен только к этому конкретному тегу. Пример `<p style="color: red">`.

Однако подобное применение стилей чревато большими затратами сил и времени, а также жертвует эстетической составляющей HTML-кода. А потому в этом случае будет лучше применить внешние файлы и селекторы. Селекторы это то, что помогает браузеру стилизовать определенные теги. Достаточно лишь прописать в отдельном файле нужные стили и подключить его к существующему HTML-файлу, и Web-страница заиграет новыми красками.

Когда браузер видит нужный селектор, он применяет стили из правила ко всем подходящим тегам. Например, ко всем абзацам или ко всем заголовкам первого уровня.

Основные преимущества CSS:

1. Чистый код
 - Этот код легко поддерживать
 - Он быстро загружается
2. Модульный код
 - Стили могут применяться сразу к нескольким страницам
 - Единообразный дизайн
3. Дизайн
 - Точность контроля (позиционирование, размер, поля и др.)
4. Разделение труда
 - Разработчик — разрабатывает, дизайнера — создает дизайн
5. Доступность
 - Пользователи могут переписывать стилевые таблицы автора [11].

1.2.5 База данных MySQL

MySQL – система управления базами данных. Одна из наиболее распространенных и популярных СУБД на сегодняшний день. Как можно догадаться по названию, система использует язык SQL – язык структурированных запросов. MySQL-сервер используется в совершенно различных системах от небольших сайтов, до крупных интернет-проектов.

Система отличается высокой скоростью работы, надежностью, гибкостью. Достигается это путем использования реляционных баз данных, в которых данные хранятся в отдельных таблицах. Таблицы можно связать между собой, или, при необходимости, объединить при выполнении запроса данные из нескольких таблиц. Работа с MySQL, обычно, не вызывает больших

затруднений. Поддержка сервера MySQL автоматически включается в поставку PHP.

Необходимость длительного хранения информации довольно часто встречается в Web-программировании: подсчёт посетителей на сайте, хранение новостей на форуме, управление содержимым сайта и прочее.

Способы работы с файлами весьма трудоёмки: нужно заботиться о помещении в них информации, о её сортировке и извлечении. Притом не стоит забывать, что все это обычно происходит на сервере хост-провайдера, где с большой вероятностью установлен один из вариантов Unix, следовательно, нужно будет так же позаботиться о правах доступа к файлам и их размещении. Все это приводит к тому, что объём кода немало так возрастает, и совершить ошибку в программе становится очень просто.

Именно таких проблем помогают избежать базы данных. Они сами заботятся о безопасности информации, её сортировке, а также позволяют извлечь или поместить информацию при помощи одной лишь строчки. Коды с использованием баз данных получаются намного компактнее, и отлаживать их значительно легче. Также не стоит забывать и о скорости - выборка информации из базы данных получается куда как быстрее, выборки из файлов [12, 13].

1.3 Обоснование выбранных средств разработки

Такие средства разработки были выбраны не случайно. Каждое из них имеет определенный набор качеств и характеристик, которые наиболее подходят для данной работы.

К примеру, основным плюсом СУБД MySQL является ее свободное распространение и поддержка практически всеми известными платформами. Также немаловажными факторами служат ее быстрдействие и удобство администрирования. Поэтому наиболее подходящей системой и была выбрана именно MySQL.

Рассматривая такие программные продукты как HTML, CSS и JavaScript, становится понятным, что ни одна современная разработка веб-приложений не может обойтись без них. Потому тут все очевидно.

Что же касается выбора языка PHP, то это самый популярный язык в веб-программировании на сегодняшний день. Он достаточно прост и удобен в использовании, и что весьма важно поддерживает работу с MySQL. Ну и, конечно же, нельзя не упомянуть огромное количество библиотек и готовых решений, что сформировались за годы существования PHP.

Суммируя все это, становится понятна причина выбора именно этих средств реализации.

1.4 Техническое задание

1.4.1 Наименование системы

Полное наименование системы:

Веб-сервер для регистрации участников олимпиады АСМ по программированию. Автоматическая генерация документов.

Краткое наименование системы:

Веб-сайт для проведения олимпиады.

1.4.2 Назначение и цели создания системы

1.4.2.1 Назначение системы

Среди основных пунктов назначения системы:

- Помощь в донесении нужной информации пользователям путем публикации различных новостей на сайте;
- Помощь оргкомитету в проведении олимпиад путем автоматизации регистрации пользователей и занесением их в базу данных.

- Автоматизация заполнения необходимых документов путем вытягивания нужных данных из базы данных.

1.4.2.2. Цель создания системы

Основная цель разработки данной системы это облегчение взаимодействия между оргкомитетом олимпиады и ее участниками. Такая система позволит автоматизировать процесс регистрации участников и процесс заполнения нужных для олимпиады документов, что значительно сэкономит время и облегчит работу всем принимающим в этом участие.

1.4.3 Требования к системе

1.4.3.1 Требования к структуре и функциональной части системы

Функция печати документов должна быть реализована в административной части документа. У пользователя должна быть возможность выбора о том, какой документ подлежит печати. Документ должен автоматически генерироваться при печати, вытягивая нужные поля из базы данных. Загруженный документ также должен поддерживать форматы doc или odt, для возможности последующего редактирования документа членами оргкомитета.

1.4.3.2 Требования к разделению доступа

Возможность печати документов должен иметь только администратор. Рядовые пользователи не имеют возможности скачивать или просматривать документы.

1.4.3.3 Требования по эргономике и технической эстетике

Требования к внешнему оформлению

При разработке графических элементов интерфейса и цветового оформления учитывать удобство и эстетическую составляющую системы. Интерфейс должен быть приятен для восприятия, содержимое полей не должно выбиваться за рамки или прижиматься к краям экрана.

Требования к диалогу с пользователем

Интерфейс не должен вызывать у пользователя лишних вопросов, напротив должен быть интуитивно понятен.

1.4.4 Требования к видам обеспечения

1.4.4.1 Требования к лингвистическому обеспечению

Языки программирования

Разработка программного обеспечения должна быть написана с помощью языков высокого уровня.

Языки взаимодействия пользователей и системы

Основным языком взаимодействия должен быть русский язык. Основной интерфейс должен взаимодействовать с пользователем на русском языке.

1.4.4.2 Требования к программному обеспечению

Программное обеспечение должно представлять собой комплекс взаимосвязанных средств разработки, необходимых для нормального развертывания и функционирования системы.

1.5 Итоги анализа

Таким образом, задание было детально проанализировано. Изучены различные существующие подобные системы, выбраны наиболее подходящие средства разработки.

2 Разработка архитектуры и основных технических решений

2.1 Общая архитектура системы

Прежде всего, любая разработка веб-приложений подразумевает под собой наличие модели клиент-сервер, так как взаимодействие между пользователями и системой обычно происходит через браузер.

Рассмотрим данную модель подробнее. Архитектура клиент-сервер позволяет распределить нагрузку между пользователем и системой. Вся основная работа происходит на сервере, который ожидает запроса от клиента, чтобы обработать и выдать ответ с необходимой клиенту информацией. На рисунке 4 представлена структура данной модели.

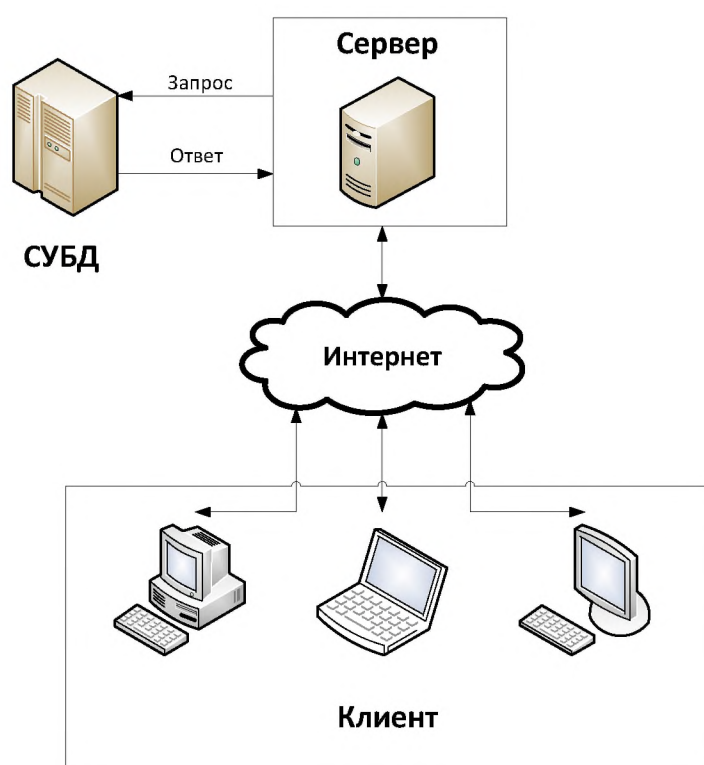


Рисунок 4 – Структурная схема обмена данными между клиентами и сервером

При разработке системы, в качестве сервера был использован виртуальный хостинг Beget. Именно на этом хостинге хранятся все данные и файлы, касающиеся разработанного сайта. И хотя данный хост является платным, он обеспечивает достаточно удобную работу с файлами сайта, имеет приятную в использовании панель управления и обладает встроенной базой данных. На рисунке 5 изображена главная страница хостинга.

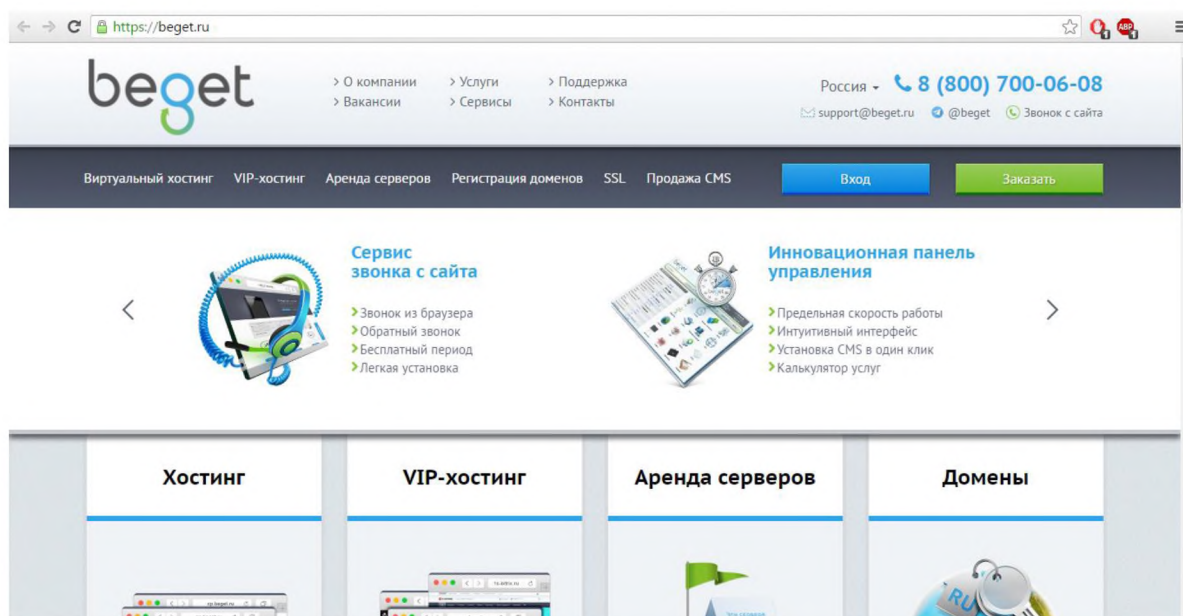


Рисунок 5 – Главная страница хостинга Beget

Работа структуры клиент-сервер, выражается в обмене данными между сервером Beget, на котором хранятся и обрабатываются все данные сайта, и пользователем-клиентом, который работает на данном сайте. Клиент обращается по соответствующему URL в адресной строке веб-браузера, и попадает на главную страницу сайта, где он может перейти в нужные ему разделы. Сервер обрабатывает каждый запрос клиента и выдает соответствующий ответ.

Для того чтобы иметь возможность печати документов, пользователь должен перейти в личный кабинет, и войти в него под именем администратора.

Администратору доступна «Панель администрирования», в которой он имеет возможность печатать необходимые документы, редактировать новости и видеть списки всех зарегистрировавшихся участников. На рисунке 6 показана схема получения документов администратором.



Рисунок 6 – Структурная схема получения клиентом документов

При нажатии администратором на кнопку печати документа, запрос посылается на сервер. После чего сервер производит обращение к собственной базе данных и «вытягивает» из нее нужные поля в соответствующий шаблон документа. Далее готовый документ посылается обратно пользователю.

2.2 Описание структуры базы данных

При формировании документа используется созданная база данных MySQL, в которой хранятся таблицы с данными зарегистрировавшихся участников олимпиады. На рисунке 7 представлена структура таблицы участников.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Первичный Ещё
2	fam	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
3	name	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
4	otch	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
5	fam_en	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
6	name_en	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
7	otch_en	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
8	date	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
9	uch_zav	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
10	email	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
11	number	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
12	tren_fam	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
13	tren_nam	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
14	tren_otch	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
15	tren2_fam	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
16	tren2_nam	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
17	tren2_otch	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
18	tip	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё

Рисунок 7 – Структура таблицы зарегистрированных участников

Всего в этой таблице имеется 22 идентификатора, в каждый из которых заносятся определенные данные о зарегистрировавшемся. Рассмотрим некоторые пункты более подробно:

- **id** - идентификатор участника;
- **fam** - фамилия;
- **name** - имя;
- **otch** - отчество;
- **fam_en** - фамилия на английском;
- **name_en** - имя на английском;
- **otch_en** - отчество на английском;
- **date** - дата рождения;
- **uch_zav** - учебное заведение;
- **email** - почта;
- **number** - номер телефона;
- **tren_fam** - фамилия тренера;

- **tren_nam** - имя тренера;
- **tren_otch** - отчество тренера;
- **tren2_fam** - фамилия второго тренера;
- **tren2_nam** - имя второго тренера;
- **tren2_otch** - отчество второго тренера;
- **tip** - статус участника: школьник или студент;
- **password** - пароль.

С помощью имен полей данной таблицы, имеется возможность напрямую обращаться к ним, когда необходимо получить соответствующую информацию из базы данных.

2.3 Работа системы с СУБД

Для того чтобы система успешно взаимодействовала с базой данных, нужно просто подключить ее в начале исполняемого файла. Происходит это достаточно просто:

```
{
  /**Подключение к БД
  $db = mysql_connect("localhost","c933322l_bd","stalker");
  /**Выбор таблицы в БД
  mysql_select_db("c933322l_bd",$db);
  /**Установка кодировки
  mysql_set_charset('utf8',$db);
  if (!mysql_connect(localhost, "c933322l_bd", "stalker")) {
    echo "Ошибка подключения к серверу MySQL";
    exit;
  }
```

После подключения к базе данных возможна дальнейшая работа, при которой можно напрямую обращаться к ее полям.

2.4 Разработка интерфейса печати документов

Для того чтобы напечатать определенный документ, пользователю необходимо разобраться как это сделать, а потому немаловажную роль играет разработка удобного и понятного интерфейса.

Функция печати документов реализована в административной панели. Для того чтобы выбрать, какой же собственно документ печатать был использован такой инструмент как jQuery Helium Modal. Это простой jQuery плагин, который помогает создать отзывчивые и стильные окошки для выбора каких-либо действий на вашей странице. На рисунке 8 представлен пример такого окошка [14].

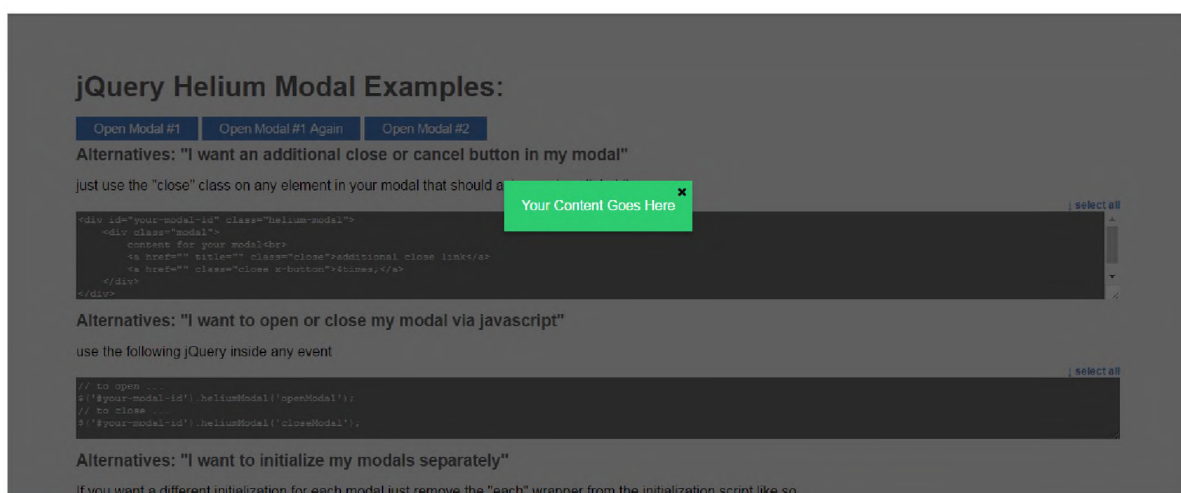


Рисунок 8 – Пример всплывающего окошка Helium Modal

Особенности данного модуля:

- Не требует установки ширины;
- Полная поддержка CSS стилей;
- Централизует окошко в зависимости от ширины текста;
- Настраиваемая анимация показать/скрыть;
- Срабатывает через ссылку или JavaScript;

Для настройки модуля были загружены и подключены готовые скрипты **helium.modal.js** и **helium.mutate.js**. После чего нужно было настроить контент в исполняемом файле админ-панели.

```
<div id="document-select-modal" class="helium-modal">
<div class="modal">
<a href="" class="close x-button">&times;</a>
<table>
<tr>
<td><a href="pechat.php?print=pupil"> Печать бейджиков для учеников</a></td>
<td><a href="pechat.php?print=trainer"> Печать бейджиков для тренеров</a></td>
</tr>
<tr>
<td><a href="pechat.php?print=register_list_students"> Печать списка
зарегистрировавшихся студентов</a></td>
<td><a href="pechat.php?print=register_list_pupils"> Печать списка
зарегистрировавшихся учеников</a></td>
</tr>
</table>
</div>
</div>
```

Так реализована функция выбора документа, который подлежит печати. Однако чтобы интерфейс не казался сырым, следует также прописать для данных тегов стили и скрипт, который будет воспроизводить анимацию.

```
<style type="text/css">
/* === Обязательные стили === */
.helium-modal{
    position: absolute;
    top:0;
    left:0;
    width:100%;
```

```

background-color: rgba(0,0,0,0.6);
z-index: 2000;
display: none;
}
.helium-modal .modal{
position: absolute;
left: -9999px;
}
/* === Настраиваемые стили === */
.helium-modal .modal{
background-color: #ffffff;
padding: 20px;
box-shadow: 0 4px 4px rgba(0,0,0,0.2);
}
.helium-modal .modal a.x-button{
position: absolute;
top: 3px;
right: 6px;
font-weight: bold;
font-family: Arial, Helvetica, sans-serif;
text-decoration: none;
font-size: 20px;
line-height: 20px;
color: #222;
}
/* === Стилизация таблицы === */
.helium-modal .modal table{
width: 100%;
text-align: center;
border-collapse: separate;
border-spacing: 7px 5px; /* Расстояние между границ */
}
.helium-modal .modal table td{
background-color: #0099CC;
color: white;
}

```

```

        text-align: center;
        text-decoration: none;
        margin: 0 5px;
    }
    .helium-modal .modal table td>a{
        display: block;
        padding: 14px 25px;
        color: white;
        text-align: center;
        text-decoration: none;
    }
    .helium-modal .modal table td:hover{
        background-color: #00688a;
    }
</style>

```

И скрипт анимации:

```

<script type="text/javascript" charset="utf-8">
    $(function() {
        $(".helium-modal").each( function(){
            $(this).heliumModal({
                vert: 50,          // Позиция модального окна 50% по высоте
                speed: 500,       // Скорость анимации
                easing: 'swing',  // Easing анимация модального окна
                trapFocus: false // Автофокус при открытии
            });
        });
    });
</script>

```

2.5 Анализ существующих способов печати документа

Для того чтобы реализовать на сайте функцию печати документов были рассмотрены несколько существующих способов того, как это можно сделать.

2.5.1 Печать страницы HTML

Первым и, пожалуй, самым простым способом является обычная печать HTML страницы. Данный способ представляет собой вывод предполагаемого содержимого документа на HTML-страницу с его последующей печатью.

Делается это достаточно просто и не требует каких-либо особых приемов. Однако такой способ имеет существенный недостаток – при таком варианте печати документа, организационная комиссия олимпиады не сможет редактировать содержимое документа, если у них возникнет с том необходимость.

2.5.2 Работа с Word-шаблонами посредством XML

Для данного способа создается шаблон Word документа, с полностью заполненным оформлением, с автоматически формируемыми полями и колонтитулами. Позже в этот шаблон подставляются в нужные места необходимые данные.

Документ Word, а точнее его .docx формат представляет собой ZIP-архив, состоящий из XML-файлов и прочих элементов. При распаковке такого документа, можно просмотреть его содержимое, где основным файлом, в котором хранится все содержимое, является document.xml.

Суть подобного способа заключается в программном изменении содержимого document.xml, упаковке его в нужное место шаблона Word и отправки документа пользователю. Этот способ использует ZIP-расширение, которое помогает работать с шаблонами.

Среди минусов такого способа – неудобная работа с содержимым XML-документов [15].

2.5.3 Генераторы отчетов

Еще один популярный способ создания документов на основе БД – генераторы отчетов, которые встроены в некоторые средства разработки. Наиболее известными представителями таких генераторов считаются QuickReport, из комплекта Delphi и C++ Builder, и генератор отчетов, встроенный в Microsoft Access 2000.

Однако такие генераторы так же имеют ряд существенных недостатков. Дело в том, что подобные инструменты, зачастую, обладают гораздо меньшим функционалом, нежели специализированные средства генерации отчетов. Например, они могут не поддерживать некоторые механизмы доступа к данным, или обладают не достаточно точными инструментами для позиционирования элементов отчета. Основная причина этого в том, что, как правило, генераторы отчетов используются, когда от печатного документа, генерируемого приложением или СУБД, требуется простота и скорость его получения, а не высокая точность печати или сложность алгоритма печати содержимого документа [16].

2.5.4 Библиотека odtRHP

Для того чтобы реализовать функцию печати документов было решено использовать одну из стандартных библиотек RHP. Библиотека odtRHP является стандартным ориентированным объектом RHP. Она позволяет автоматически генерировать текстовые документы OpenOffice на основе имеющихся шаблонов.

Данная библиотека очень проста в использовании. Несмотря на небольшое количество кода, появляется возможность создавать простые

документы, просто заменяя теги в шаблоне документа. Кроме того, можно воспользоваться расширенными возможностями для генерации сложных документов путем повторения части документа или строки массива.

Другими словами, odtPHP позволяет заменить специальные теги в вашем документе на какой-то определенный текст или изображение. На рисунке 9 приведен пример работы библиотеки [17].

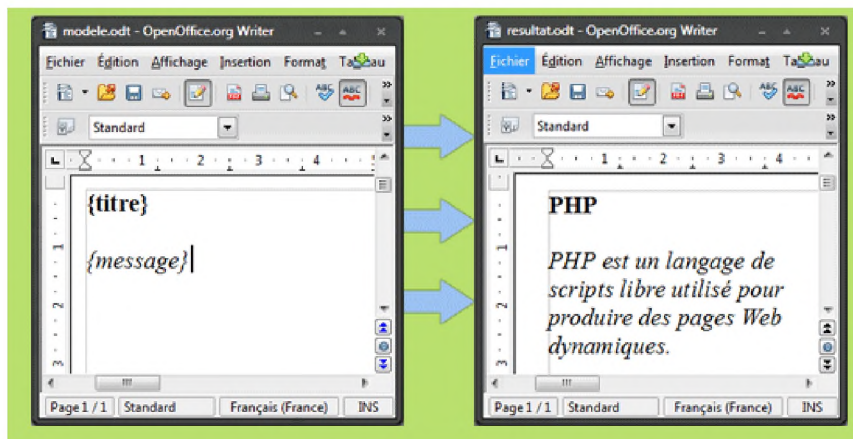


Рисунок 9 – Пример работы odtPHP

2.6 Разработка функции печати документов

2.6.1 Разработка шаблонов odt

Для того чтобы генерировались нужные документы, необходимо создать для них соответствующие шаблоны. Были созданы несколько шаблонов таких документов, которые хранятся на сервере в папке /doc.

Шаблон печати бейджей для участников олимпиады называется **template.odt**, и представляет собой образец будущего бейджика с соответствующими тегами, на местах куда должны «вытягиваться» определенные поля из базы данных. Также в шаблоне присутствуют обязательные поля `[--! BEGIN articles --]` и `[--! END articles --]`, необходимые для того чтобы определить начало и конец образца повторяемого сегмента.

Подобным же образом генерируются и остальные шаблоны с той лишь разницей, что каждому нужно задавать определенные параметры. Шаблоны документов печати бейджей для тренеров и списки зарегистрировавшихся школьников и студентов называются соответственно: **template_trainer.odt**, **template_pupils.odt**, **template_students.odt**. На рисунках 10-13 изображены шаблоны имеющихся документов.



Рисунок 10 – Шаблон бейджика участника



Рисунок 11 – Шаблон бейджика тренера

Полуфинальные соревнования Всероссийской командной олимпиады школьников по информатике и программированию
Восточно-Сибирского региона
23 по 25 октября 2016 г.

№ д/п	Учебное заведение	Название	Login	Pass	Аудитория	Ф.И.О. участника
[!-- BEGI N articl es-- {nu mber }	{uch_zav}	Команда {number}	Команда {number}	{pass }		{fio}
						{fio1}
						{fio2}
[!-- END articl es--]						

Рисунок 12 – Шаблон списка зарегистрировавшихся учеников

ЧЕТВЕРТЬФИНАЛЬНЫЕ СОРЕВНОВАНИЯ ЧЕМПИОНАТА МИРА ПО ПРОГРАММИРОВАНИЮ
среди сборных команд высших учебных заведений
ВОСТОЧНО-СИБИРСКОГО РЕГИОНА РОССИИ
23 по 25 октября 2016 г

№ д/п	Вуз	Название команды	Ф.И.О. участника	Курс	Группа	Ф.И.О. тренера
[!-- BE GI N arti cle s-- {n um ber }	{vuz}	Команда {number}	{fio}	?	?	{trainer}
			{fio1}	?	?	{trainer1}
			{fio2}	?	?	{trainer2}
[!-- EN D arti cle s--]						

Рисунок 13 – Шаблон списка зарегистрировавшихся студентов

2.6.2 Функция генерации документов

В первую очередь нужно было скачать библиотеку odtPHP. Она была размещена на сервере в папке /doc вместе с шаблонами. В папке с библиотекой имеются все файлы необходимые для ее нормального функционирования. Главный файл печати документов **pechat.php** отвечает уже непосредственно за автоматическое заполнение и генерацию нужного документа.

Для начала в этом файле необходимо подключить данную библиотеку и установить соединение с базой данных:

```
if (!empty($_GET)) {
    include('doc/odtphp-1.0.1/library/odf.php');
    $listeArticles = array(); /* Объявление массива результатов
    $db = mysql_connect("localhost","c933322l_bd","stalker");
    mysql_select_db("c933322l_bd",$db);
    mysql_set_charset('utf8',$db);
    if (!mysql_connect(localhost, "c933322l_bd","stalker")) {
        echo "Ошибка подключения к серверу MySQL";
        exit;
    }
}
```

Далее идет раздел печати бейджей. Выбор печати происходит между участниками и их тренерами, после чего выбирается соответствующий шаблон.

```
if ($_GET['print'] == 'pupil' || $_GET['print'] == 'trainer') {
    switch ($_GET['print']) {
        case 'pupil':
            $odf = new odf("doc/template.odt");
            $result = mysql_query("SELECT fam, name, otch, uch_zav FROM uch ");
            $texte = 'УЧАСТНИК';
        case 'trainer':
            $odf = new odf("doc/template_trainer.odt");
            $result = mysql_query("SELECT tren_fam, tren_nam, tren_otch, uch_zav
FROM uch");
            $texte = 'ТРЕНЕР';
    }
}
```

Следующим шагом работает цикл, который пробегается по данным, полученным из базы данных, и добавляет новые элементы в *listeArticles* в необходимом для odtPHP формате для участников:

```
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {  
    $listeArticles[] = array('titre' => "{$row['fam']} \n {$row['name']} {$row['otch']}",  
        'texte' => $texte, 'vuz' => $row['uch_zav']);  
}
```

Для тренеров:

```
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {  
    $listeArticles[] = array('titre' => "{$row['tren_fam']} \n {$row['tren_nam']}  
{$row['tren_otch']}",  
        'texte' => $texte, 'vuz' => $row['uch_zav']);  
}
```

Далее определяется название сегмента BEGIN-END в шаблоне и происходит циклическая обработка этого сегмента.

```
$article = $odf->setSegment('articles');  
    foreach ($listeArticles AS $element) {  
        $article->titreArticle($element['titre']);  
        $article->texteArticle($element['texte']);  
        $article->vuzArticle($element['vuz']);  
        $article->merge();  
    }
```

Затем результаты добавляются в генерируемый документ, и он отдается на скачивание.

```
$odf->mergeSegment($article);  
$odf->exportAsAttachedFile();
```

Немного иначе происходит генерация списков зарегистрировавшихся участников. Сначала так же происходит выборка между списком школьников и студентов, и подгружается соответствующий шаблон.

```
if ($_GET['print'] == 'register_list_pupils' || $_GET['print'] == 'register_list_students'){  
    switch ($_GET['print']){  
        case 'register_list_students':
```

```

    $odf = new odf("doc/template_students.odt");
    $query = sprintf("SELECT fam, name, otch, tren_fam, tren_nam, tren_otch,
uch_zav, tip FROM uch WHERE tip = '%s' OR tip = '%s'",
    mysql_real_escape_string("Смудеум"),
    mysql_real_escape_string("Смудеум ИКИТ"));

```

Далее запрашиваются данные из базы данных по предыдущему запросу \$query и считается количество строк пришедших из нее.

```

$result = mysql_query($query);
$num_rows = mysql_num_rows($result);
$rows = [];

```

Все это нужно для формирования команд по три человека из одного учебного заведения. Поэтому далее создается функцию, формирующую только одну строку в документе:

```

function get_row($array, $keys, $i){
    if (isset($i) && $keys) {
        if (@$array[$i]) {
            return array(
                $keys[0] => "{$array[$i]['fam']} {$array[$i]['name']} {$array[$i]['otch']}",
                $keys[1] => "{$array[$i]['tren_fam']} {$array[$i]['tren_nam']}
{$array[$i]['tren_otch']}",
                $keys[2] => "{$array[$i]['uch_zav']}"
            );
        } else {
            return array(
                $keys[0] => " ",
                $keys[1] => " ";
            );
        }
    }
}

```

После этого происходит преобразование данных из БД в массив rows.

```

while ($row = mysql_fetch_assoc($result)){
    $rows[] = $row;
}

```

Затем данные сортируются по учебному заведению, для того чтобы в команду попадали только студенты из одного образовательного учреждения.

```

$sortedData = array();

```

```

    $listeArticles = array();
    foreach($rows as $row) {
        $sortedData[$row['uch_zav']] = $row;
    }

```

Теперь нужно еще раз пройтись по отсортированным данным и сделать так, чтобы в команде оказалось по три студента от одного учреждения. Если же второго и третьего студента не окажется, то на их месте запишется пустота. Результаты записываются в *listeArticles*.

```

    foreach ($sortedData as $item) {
        $length = count($item);
        for ($i = 0; $i <= $length; $i = $i+3){
            $listeArticles[] = array_merge(
                get_row($item, array('fio', 'trainer', 'vuz'), $i),
                get_row($item, array('fio1', 'trainer1', 'vuz'), $i+1),
                get_row($item, array('fio2', 'trainer2', 'vuz'), $i+2)
            );
        }
    }

```

Далее снова идет работа с сегментом образца в шаблоне, куда подставляются элементы из *listeArticles*.

```

    $article = $odf->setSegment('articles');
    $number = 0; /* Счетчик команд
    foreach($listeArticles AS $element) {
        $number++;
        $article->setVars('number', $number);
        $article->setVars('fio', $element['fio'], true, 'UTF-8');
        $article->setVars('trainer', $element['trainer'], true, 'UTF-8');
        $article->setVars('fio1', $element['fio1'], true, 'UTF-8');
        $article->setVars('trainer1', $element['trainer1'], true, 'UTF-8');
        $article->setVars('fio2', $element['fio2'], true, 'UTF-8');
        $article->setVars('trainer2', $element['trainer2'], true, 'UTF-8');
        $article->setVars('vuz', $element['vuz'], true, 'UTF-8');
        $article->merge(); }

```

Остается только сгенерировать нужный документ и отдать его для печати:

```
$odf->mergeSegment($article);
```

```
$odf->exportAsAttachedFile();
```

Вывод второго списка зарегистрированных участников происходит похожим образом.

2.7 Итоги разработки

В результате разработки системы получилось создать рабочую функцию печати документов на сайте. Данная система позволит организаторам олимпиад затрачивать меньшее количество усилий и времени на заполнение необходимых документов, касающихся проведения олимпиады.

3 Руководство пользователя

3.1 Руководство администратора по развертыванию системы

3.1.1 Развертывание базы данных и файлов на сервере

Для того чтобы система правильно функционировала на новом сервере, нужно обеспечить ей корректное соединение с базой данных. Для этого нужно либо изменить все названия БД в файлах, что является весьма утомительной задачей, либо, что гораздо проще, можно создать базу данных с уже имеющимся названием **c933322l_bd**. Для этого необходимо зайти в веб-интерфейс администрирования **phpmyadmin** и создать там базу данных со схожим названием (рисунок 14).

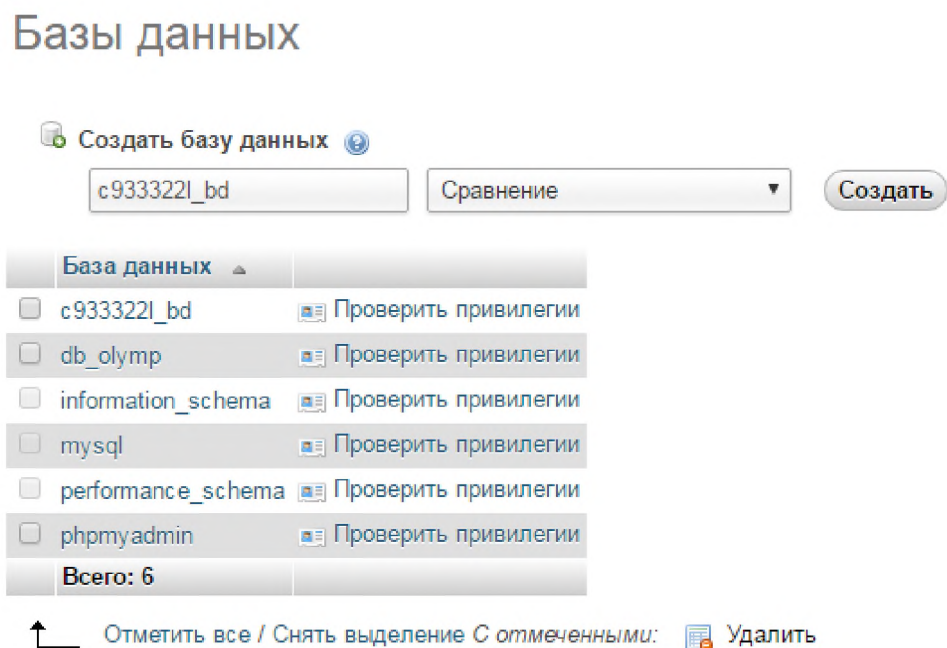


Рисунок 14 – Создание базы данных

Далее происходит импорт файла с имеющейся базой данных **c9333221_bd.sql**, где содержатся все таблицы с данными веб-сервера (рисунок 15).

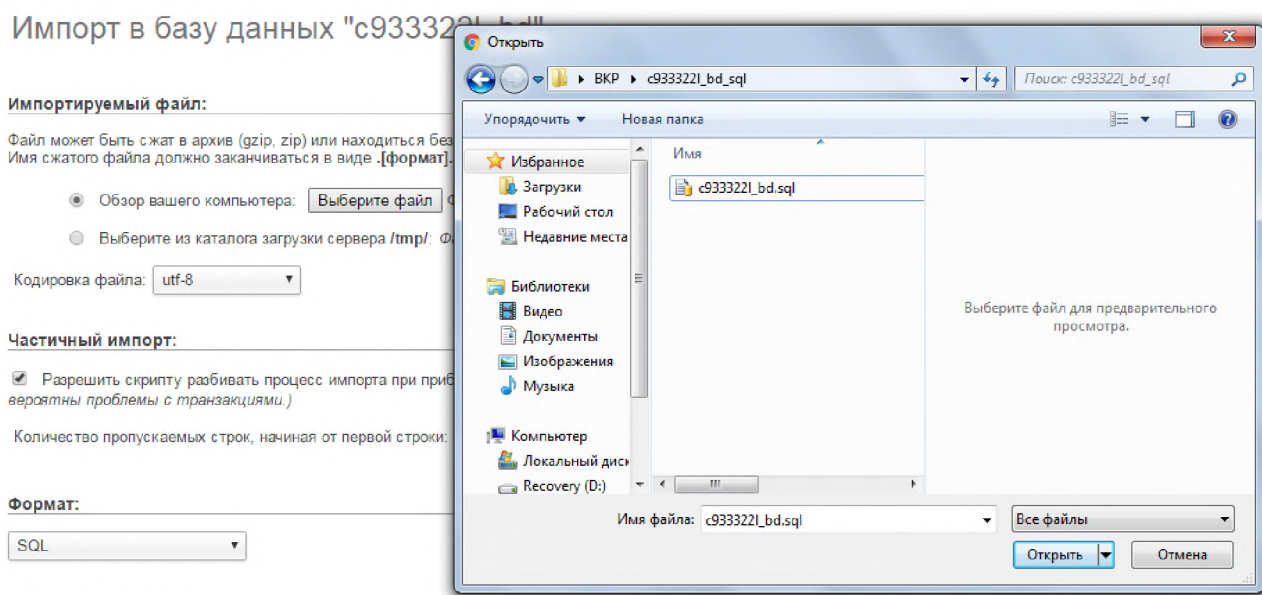


Рисунок 15 – Импорт базы данных

После развертывания базы данных, необходимо перенести на сервер нужные для работы системы файлы, среди которых в функции печати документов принимают участие:

- **adminka.php** – файл панели администратора;
- **pechat.php** – файл генерации и печати документов;
- библиотека **odtPHP**;
- **template.odt** – шаблон бейджей для участников;
- **template_trainer.odt** – шаблон бейджей для тренеров;
- **template_pupils.odt** – шаблон списка учеников;
- **template_students.odt** – шаблон списка студентов.

3.1.2 Шаблоны документов

Одним из основных инструментов печати являются редактируемые шаблоны документов. Шаблоны нужные для печати находятся в папке **doc/**. Эти шаблоны можно менять и редактировать, если в том возникает необходимость. На рисунке 16 представлено содержимое папки **doc/**.

Имя	Дата изменения	Тип	Размер
odtphp-1.0.1	13.06.2016 16:21	Папка с файлами	
~\$mplate.odt	29.05.2016 11:08	Текстовый докум...	1 КБ
template.odt	29.05.2016 11:08	Текстовый докум...	13 КБ
template_copy.odt	29.05.2016 15:53	Текстовый докум...	13 КБ
template_pupils.odt	29.05.2016 11:08	Текстовый докум...	8 КБ
template_students.odt	29.05.2016 11:08	Текстовый докум...	8 КБ
template_trainer.odt	29.05.2016 11:08	Текстовый докум...	15 КБ
template_trainer_copy.odt	13.06.2016 18:05	Текстовый докум...	15 КБ

Рисунок 16 – Содержимое папки doc

3.2 Руководство пользователя по работе с системой

3.2.1 Вход в панель администратора

Доступ к функции печати документов предоставляет панель администратора, поэтому для того чтобы начать работу необходимо авторизоваться под логином администратора. Для этого нужно перейти в админ-панель и в полях email и пароль ввести учетные данные (рисунок 17).

Email

rock-al@mail.ru

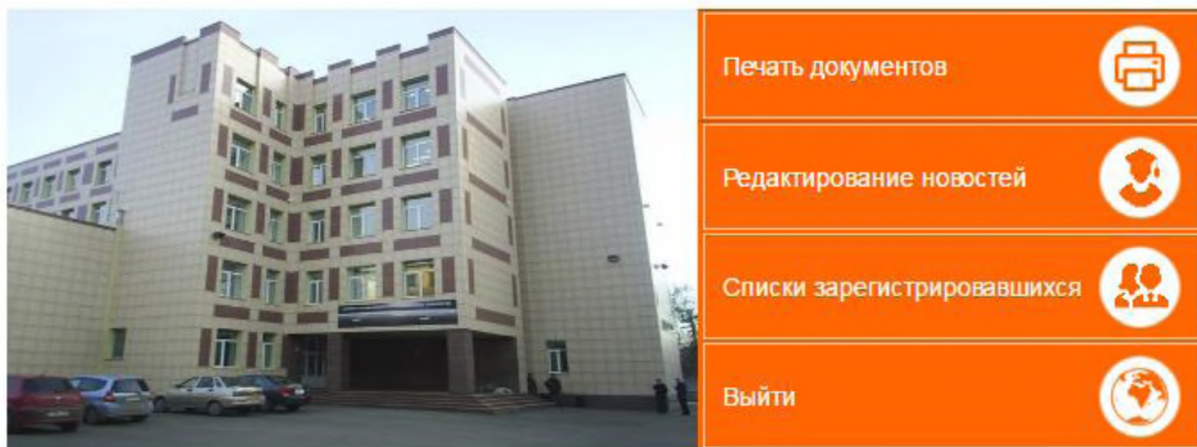
Пароль

.....|

Войти

Рисунок 17 – Авторизация пользователя под администратором

После авторизации пользователю становится доступна админ-панель, где находится функция печати документов (рисунок 18).



Вход в админ панель

Здравствуйте, Админ!

Рисунок 18 – Панель администратора

3.2.2 Печать документов

При нажатии на «Печать документов» в админ-панели появляется окошко, где можно выбрать необходимый документ (рисунок 19).

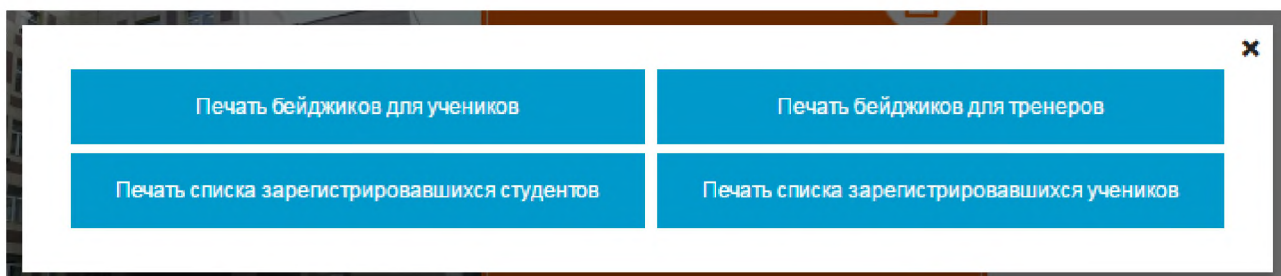


Рисунок 19 – Окно выбора документа

После выбора соответствующего варианта, происходит скачивание сформированного документа (рисунок 20).

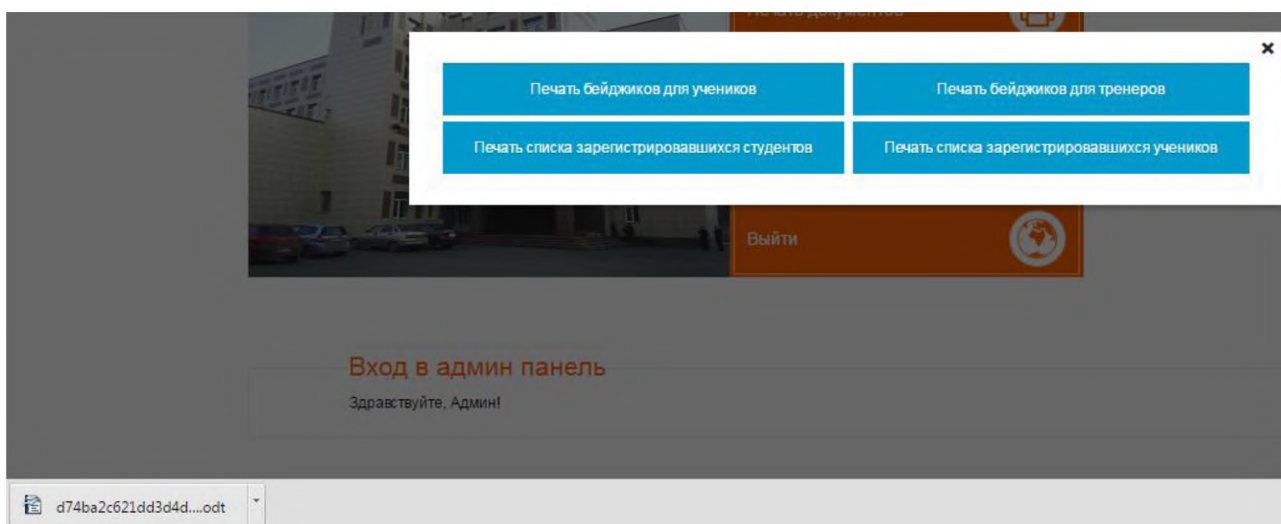


Рисунок 20 – Скачивание документа

3.2.3 Демонстрация сформированных документов

Документ с бейджками участников представлен на рисунке 21.

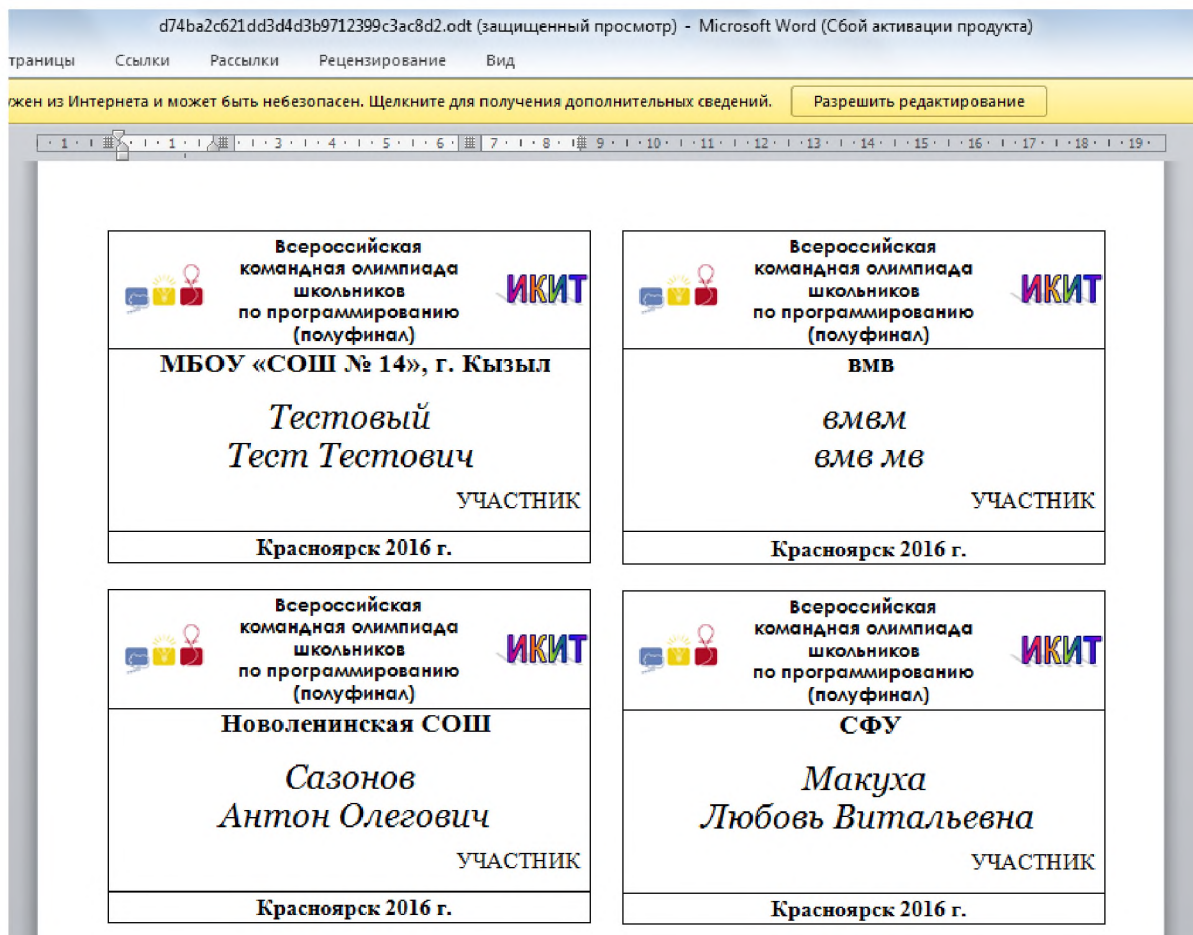


Рисунок 21 – Документ с бейджиками участников

Документ с бейджками тренеров показан на рисунке 22.

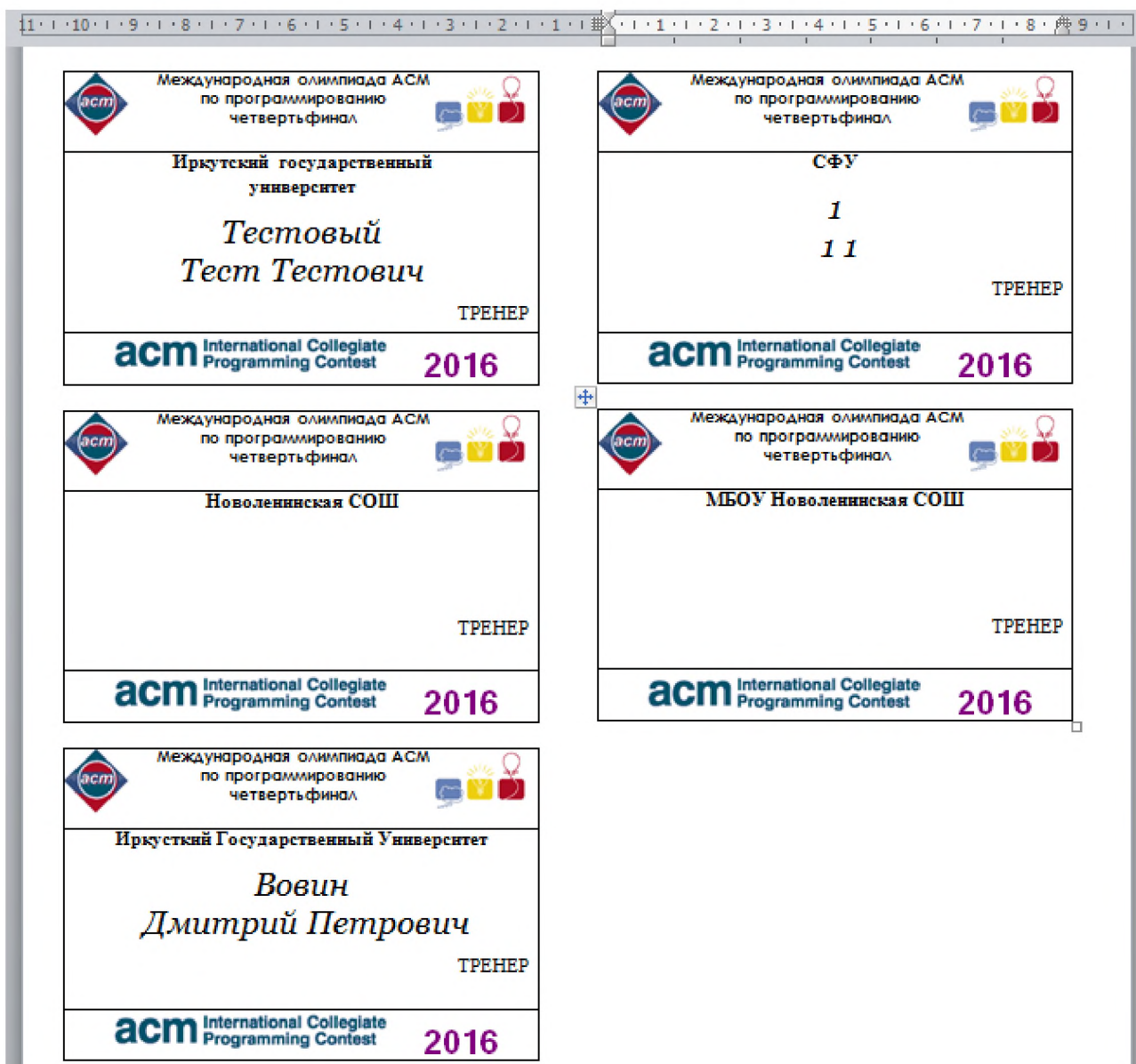


Рисунок 22 – Документ с бейджками тренеров

Список зарегистрировавшихся студентов представлен на рисунке 23.

ЧЕТВЕРТЬФИНАЛЬНЫЕ СОРЕВНОВАНИЯ ЧЕМПИОНАТА МИРА ПО ПРОГРАММИРОВАНИЮ
среди сборных команд высших учебных заведений
ВОСТОЧНО-СИБИРСКОГО РЕГИОНА РОССИИ
23 по 25 октября 2015 г

№ п/п	Вуз	Название команды	Ф.И.О. участника	Курс	Группа	Ф.И.О. тренера
1	Новоленинская СОШ	Команда 1	Сазонов Антон Олегович	?	?	
				?	?	
				?	?	
2	МБОУ Новоленинская СОШ	Команда 2	Слепцов CrazyExtrime dfgdfg	?	?	
				?	?	
				?	?	

Рисунок 23 – Список зарегистрированных студентов

Списка зарегистрированных школьников не представлено, так как на данный момент таблица пустая.

ЗАКЛЮЧЕНИЕ

На сегодняшний день автоматизация различных процессов становится все более и более актуальной проблемой. Уставая выполнять тяжелую и нудную повседневную работу, человек старается любыми способами облегчить свои нагрузки. Люди, занимающиеся организацией олимпиад, не являются исключением. Разработка сайта для проведения олимпиады по программированию с функцией автоматической генерации документов, лишний раз подтверждает эту теорию.

Данная система позволит оргкомитету значительно снизить затраты времени и сил на проведении олимпиады, а также в разы ускорит коммуникацию с ее участниками.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТО 4.2-07-2014 – Стандарт организации, система менеджмента качества, общие требования к построению, изложению и оформлению документов учебной деятельности. Дата введ. 30.12.2013 г., приказ №1520 (взамен СТО 4.2-07-2012).
2. Портал олимпиады школьников «Ломоносов» [Электронный ресурс]: Режим доступа: <http://olymp.msu.ru/>
3. Информационная система «Абитуриент-студент» [Электронный ресурс]: НИЯУ МИФИ – Режим доступа: <https://org.mephi.ru/>
4. Высшая школа экономики [Электронный ресурс]: Регистрация – Режим доступа: <https://olymp3.hse.ru/school.html>
5. Основы PHP [Электронный ресурс]: Синтаксис PHP – Режим доступа: <http://www.php.su/learnphp/?syntax>
6. Руководство по PHP [Электронный ресурс]: Что такое PHP? – Режим доступа: <http://php.net/manual/ru/intro-what-is.php>
7. Интернет программирование [Электронный ресурс]: Язык программирования PHP – Режим доступа: <http://www.mirsite.ru/php.htm>
8. Современный учебник JavaScript [Электронный ресурс]: Язык JavaScript – Режим доступа: <https://learn.javascript.ru/>
9. Курсы создания сайта [Электронный ресурс]: Основы HTML – Режим доступа: <http://www.site-do.ru/html/html1.php>
10. Интерактивные онлайн-курсы [Электронный ресурс]: HTML academy – Режим доступа: <https://htmlacademy.ru>
11. Руководство CSS для начинающих [Электронный ресурс]: Режим доступа: <http://technologyweb.org/>
12. Интернет программирование [Электронный ресурс]: Реляционные базы данных MySQL – Режим доступа: <http://www.mirsite.ru/mysql.htm>

13. PHP и MySQL совместная работа [Электронный ресурс]: Что такое MySQL. Взаимодействие с PHP – Режим доступа: http://www.softtime.ru/bookphp/gl12_1.php
14. jQueryScript.net [Электронный ресурс]: Плагин jQuery Helium Modal – Режим доступа: <http://www.jqueryscript.net/demo/Responsive-Customizable-jQuery-Modal-Window-Plugin-Helium-Modal/>
15. Habrahabr.ru [Электронный ресурс]: Создание печатной формы Microsoft Word с помощью PHP – Режим доступа: <https://habrahabr.ru/post/204484/>
16. Введение в базы данных [Электронный ресурс]: Генераторы отчетов – Режим доступа: <http://www.olap.ru/basic/introbd10.asp>
17. Библиотека odtPHP [Электронный ресурс]: Режим доступа: <https://web.archive.org/web/20120531095719/http://www.odtphp.com/index.php?i=home>

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
 А. И Легалов
подпись инициалы, фамилия

« ____ » _____ 2016 г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме _____ бакалаврской работы _____

Студенту Бандееву Олегу Евгеньевичу

фамилия, имя, отчество

Группа КИ12-09Б Направление (специальность) 09.03.01

номер

код

Высокопроизводительные вычислительные системы на базе больших ЭВМ

наименование

Тема выпускной квалификационной работы

Разработка веб-сервера для регистрации участников олимпиады ACM по программированию. Автоматическая генерация документов

Утверждена приказом по университету № _____ от _____

Руководитель ВКР Л. В. Макуха, старший преподаватель кафедры ВТ

инициалы, фамилия, должность, учёное звание и место работы

Исходные данные для ВКР документы необходимые при проведении олимпиады

Перечень разделов ВКР анализ задания на выпускную квалификационную работу, разработка архитектуры и основных технических решений, руководство пользователя

Перечень графического материала структурные схемы Visio, презентация в формате PowerPoint, демонстрационное видео

Руководитель ВКР

подпись

Л.В. Макуха

инициалы, фамилия

Задание принял к исполнению

подпись

О.Е. Бандеев

инициалы, фамилия

« ____ » _____ 2016 г.

РЕФЕРАТ

Тема данной выпускной квалификационной работы «Разработка веб-сервера для регистрации участников олимпиады АСМ. Автоматическая генерация документов». Пояснительная записка содержит 48 страниц текстового документа, 23 иллюстрации, 17 использованных источников.

Объект работы – автоматически генерируемые документы на сайте.

Задачи:

- анализ существующих способов генерации документов;
- оценка наилучшего варианта генерации;
- разработка интерфейса печати документов на сайте;
- реализация функции печати документов на сайте.

В результате работы должна быть реализована функция печати документов с содержимым заполняемым автоматически.

В итоге на сайте имеется возможность скачивать документы с бейджиками зарегистрированных участников, а также документы со списками зарегистрировавшихся.