


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н. П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра)
общепрофессиональных и специальных (информационно-коммуникационных)
дисциплин

УТВЕРЖДАЮ

Зам. декана по учебной
работе, преподаватель ВКК


В. А. Богатырская
(подпись)

«13» июня 2019 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ДИПЛОМНОМУ ПРОЕКТУ

на тему Разработка системы проведения многопользовательских он-лайн олимпиад

Автор проекта


(подпись)

27.05.19
(дата)

А. А. Рузманов

Специальность 09.02.03 Программирование в компьютерных системах

Обозначение дипломного проекта ДП-02069964-09.02.03-13-19

Руководитель проекта

преподаватель


(подпись)

03.06.19
(дата)

А. А. Прокин

Нормоконтролер

преподаватель ПКК


(подпись)

07.06.19
(дата)

М. Ф. Петянкин

Рецензент

начальник отдела разработки
и внедрения программного
обеспечения ООО «Престиж»


(подпись)

10.06.19
(дата)

Д. С. Бобров

Саранск
2019

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИМ. Н.П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра)
общепрофессиональных и специальных (информационно-коммуникационных)
дисциплин

УТВЕРЖДАЮ

Зам. декана по учебной
работе, преподаватель ВКК

В. А. Богатырская

(подпись)

« 28 » декабря 2018 г.

ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ

Студент Рузманов Андрей Александрович

1 Тема Разработка системы проведения многопользовательских он-лайн олимпиад

Утверждено приказом 1395-С от 25.03.19

2 Срок представления работы к защите 12.06.19

3 Исходные данные для научного исследования (проектирования) техническое задание на разработку проекта

Программный продукт должен быть реализован с интуитивно понятным интерфейсом пользователя и выполнять следующие функции:

- отображать основную информацию о проекте;
- предоставлять взаимодействие;
- отображать данные в понятном для пользователя виде

4 Содержание выпускной квалификационной работы

4.1 Изучение предметной области

4.1.1 Анализ предметной области

- 4.1.2 Анализ методов и средств проектирования
- 4.1.3 Сбор материалов для составления технического задания по теме дипломного проекта
- 4.1.4 Расчет показателей экономической эффективности программного продукта
- 4.2 Разработка проекта
 - 4.2.1 Проектирование и разработка интерфейса
 - 4.2.1.1 Проектирование диаграммы компонентов
 - 4.2.1.2 Проектирование диаграммы последовательности
 - 4.2.1.3 Проектирование диаграммы активности
 - 4.2.1.4 Проектирование пользовательского интерфейса в графическом редакторе «Krita»
 - 4.2.2 Внедрение пользовательского интерфейса
 - 4.2.2.1 Создание интерфейса пользователя на основе HTML-форм
 - 4.2.2.2 Внедрение CSS кода для HTML-форм
 - 4.2.2.3 Добавление JavaScript кода
 - 4.2.3 Проектирование серверной части и логики сервера
 - 4.2.4 Разработка и создание базы данных
 - 4.2.5 Реализация защиты данных в базе
 - 4.2.5.1 Защита базы данных на уровне операционной системы
 - 4.2.5.2 Защита базы данных на уровне СУБД
 - 4.2.5.3 Криптографическая защита данных
 - 4.2.5.4 Шифрование передаваемых данных
 - 4.2.5.5 Экранирование sql запросов
- 4.3 Проведение испытаний, отладка и внедрение программного продукта на предприятии
 - 4.3.1 Проведение функционального тестирования

4.3.2 Отладка и внедрение программного продукта на предприятии

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А (обязательное) Листинг HTML кода страниц и форм

ПРИЛОЖЕНИЕ Б (обязательное) Листинг кода серверной части

ПРИЛОЖЕНИЕ В (обязательное) Модуль шифрования данных

ПРИЛОЖЕНИЕ Г (обязательное) Чек-лист страницы elementaria.ru

ПРИЛОЖЕНИЕ Д (обязательное) Техническое задание

Руководитель работы


28.12.18
подпись, дата

А. А. Прокин

Задание принял к исполнению


28.12.18
подпись, дата

А. А. Рузманов

РЕФЕРАТ

Пояснительная записка к дипломному проекту содержит 50 страниц, 16 рисунков, 4 таблицы, 11 использованных источников, 5 приложений.

ОЛИМПИАДА, ПОЛЬЗОВАТЕЛЬ, РАЗРАБОТКА, ДИЗАЙН,
ПРОГРАММНЫЙ ПРОДУКТ, HTML, CSS, JS, SOCKET.IO.

Объектом разработки является проект по проведению многопользовательских он-лайн олимпиад, предназначенных для контроля, проверки и закрепления знаний.

Цель работы — разработка проекта для проведения многопользовательских он-лайн олимпиад.





Для создания проекта требуется решить следующие задачи:

- составление технического задания;
- составление диаграмм последовательности, компонентов и активности;
- проектирование интерфейса и клиентской части;
- проектирование серверной части;
- создание базы данных;
- реализация защиты данных в базе;
- внедрение.

В процессе работы прорабатывалась структура проекта, проводилось теоретическое исследование процесса его создания.

В результате проведенной работы создан полноценный проект на основе анализа предметной области и технического задания.

Степень внедрения — полная.

					ДП-02069964-09.02.03-13-19			
Имя	Лист	№ докум	Подпись	Дата	Разработка системы проведения многопользовательских он-лайн олимпиад	Лит	Лист	Листов
Разраб		Русланов		22.05		Д	5	68
Провер		Проким		07.06				
И.контр		Петанкин		08.06				
Утверд		Богатырская		13.06				
						ФДП и СПО, в/а, ПКЭС, 409		

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	8
1 Изучение предметной области	10
1.1 Анализ предметной области	10
1.2 Анализ методов и средств проектирования	11
1.3 Сбор материалов для составления технического задания по теме дипломного проекта	14
1.4 Расчет показателей экономической эффективности программного продукта	15
2 Разработка программного обеспечения на основе технического задания дипломного проекта	19
2.1 Проектирование и разработка интерфейса	19
2.1.1 Проектирование диаграммы компонентов	19
2.1.2 Проектирование диаграммы последовательности	21
2.1.3 Проектирование диаграммы активности	23
2.1.4 Проектирование пользовательского интерфейса в графическом редакторе «Krita»	24
2.2 Внедрение пользовательского интерфейса	25
2.2.1 Создание интерфейса пользователя на основе HTML-форм	25
2.2.2 Внедрение CSS кода для HTML-форм	26
2.2.3 Добавление JavaScript кода	28
2.3 Проектирование серверной части и логики сервера	28
2.4 Разработка и создание базы данных	32
2.5 Реализация защиты данных в базе	36
2.5.1 Защита базы данных на уровне операционной системы	36
2.5.2 Защита базы данных на уровне СУБД	39

2.5.3 Криптографическая защита данных	40
2.5.4 Шифрование передаваемых данных	41
2.5.5 Экранирование sql запросов	42
3 Проведение испытаний, отладка и внедрение программного продукта	45
3.1 Проведение функционального тестирования	45
3.2 Отладка и внедрение программного продукта на предприятии	45
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	49
ПРИЛОЖЕНИЕ А (обязательное) Листинг HTML кода страниц и форм	51
ПРИЛОЖЕНИЕ Б (обязательное) Листинг кода серверной части	54
ПРИЛОЖЕНИЕ В (обязательное) Модуль шифрования данных	57
ПРИЛОЖЕНИЕ Г (обязательное) Чек-лист страницы elementaria.ru	59
ПРИЛОЖЕНИЕ Д (обязательное) Техническое задание	61

ВВЕДЕНИЕ

Знание — это результат познавательной деятельности человека. А также самое дорогое богатство, чем может владеть человек. Благодаря новым знаниям человек развивается, вносит вклад в развитие будущего. Знания очень разнообразны, они не ограничены ни какими-либо науками, ни какими-либо сферами жизни. Знания человек получает непосредственно из окружающей его информации. Информация — это сведения в любой форме их представления.

В наше время люди много времени проводят в интернете. И зачастую время, потраченное в интернете, не способствует получению какой-либо полезной информации. Но существуют и множество развивающих проектов, направленных на получение, закрепление и усвоение знаний. Большинство из них развивающие порталы, порталы с различными статьями, словари и т.д. Отдельным этапом идут предметные студенческие олимпиады.

Предметная олимпиада — соревнования учащихся высшего, среднего общего, или профессионального образования, требующие от участников демонстрации знаний и навыков в области одной или нескольких изучаемых дисциплин. Международные студенческие олимпиады стали проводиться с 1994 года. Они включают в себя задачи из разных областей.

Олимпиады проводятся с целью:

- выявить наиболее талантливых учащихся в различных областях;
- дать возможность всем желающим учащимся проверить свои знания в различных научных областях в условиях соревнования;
- привлечения новых учащихся к научно-исследовательской работе.

Предметные олимпиады обычно проводятся в несколько этапов:

- школьный;
- городской;
- областной;

- российский;
- международный.

К участию в олимпиаде обычно заранее готовятся, что способствует закреплению учебного и дополнительного материала.

Предметные олимпиады предназначены для контроля и развития знаний человека, а также помогают в развитии личности. Участие в олимпиадах требует предварительную подготовку по отдельным предметам. Таким образом совершенствуются знания человека, пробуждается интерес к различным дисциплинам, предметам, наукам.

					<i>ДП-02069964-09.02.03-13-19</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		9

капитала. Целесообразность создания и функционирования веб-сайта должна подтверждаться расчетами экономической эффективности.

Экономическая эффективность веб-сайта в зависимости от вида оценки может быть выражена в форме абсолютной и сравнительной эффективности, выражением которой является соотношение эффекта к затратам. На основе результатов сравнительной экономической эффективности осуществляется выбор как экономически целесообразного варианта создания или развития веб-сайта в целом, так и обоснование отдельных решений по веб-сайту, по отдельным подсистемам, входящим в состав веб-сайта.

Основными источниками требуемых для расчета экономической эффективности веб-сайта исходных данных являются: данные проектной и отчетной документации; бухгалтерской и статистической отчетности административных территориальных органов, предприятий и организаций, обслуживающих веб-сайт, организаций-пользователей и др.

Для веб-сайтов факторы экономической эффективности определяются прежде всего сокращением прямых потерь ресурсов (материалов, рабочего времени, непроизводительных затрат) и потерь от недоиспользования основных производственных фондов. На этом уровне выделяют следующие основные сферы получения экономического эффекта от внедрения веб-сайта: на объекте управления — прибыль, получаемая за счет улучшения технико-эксплуатационных и экономических показателей деятельности; в управляющей системе — экономия в рамках автоматизации технологического процесса обработки информации.

Таким образом, факторы экономической эффективности веб-сайта предполагают более качественное, своевременное и комплексное решение задач, связанных с использованием материальных, трудовых и финансовых ресурсов организации.

Расчет себестоимости машинного часа эксплуатации вычислительной и оргтехники (ВиОТ):

					ДП-02069964-09.02.03-13-19	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		16

$$C_{\text{м.ч.}} = \frac{\sum_{i=1}^n Z_i}{F_{\text{п}} K_{\text{г}}}, \quad (1)$$

$$C_{\text{м.ч.}} = \frac{241500}{4380 * 0,95} = 58,03,$$

где $C_{\text{м.ч.}}$ – себестоимость машинного часа;

– Z_i — годовые затраты, связанные с эксплуатацией и обслуживанием ВиОТ

– $F_{\text{п}}$ — годовой полезный фонд времени работы единицы оборудования;

– $K_{\text{г}} = 0,95$ — коэффициент готовности.

$$F_{\text{п}} = F_{\text{н}} * (1 - \alpha_{\text{р}}), \quad (2)$$

где $F_{\text{н}}$ – номинальный годовой фонд рабочего времени в часах;

$\alpha_{\text{р}} = (0,05 \dots 0,2)$ — коэффициент, учитывающий время, затраченное на ремонт, настройку, обслуживание ВиОТ.

Расчет суммарных годовых затрат.

$$F_{\text{п}} = 4380(1 - 0,05) = 4161(\text{руб})$$

Для расчета годовых затрат, необходимо определить балансовую стоимость $C_{\text{БАЛ}}$ условного комплекта, необходимого для создания дипломного проекта.

Таблица 1 — Состав условного комплекта

№	Наименование	Кол-во шт.	Цена руб.
1	Ноутбук Lenovo G570	1	30000
	Итого:		30000

$$C_{\text{БАЛ}} = \sum_{i=1}^n C_i K_i + P_{\text{д}}, \quad (3)$$

$$C_{\text{бал}} = 30000 + 3000 = 33000.$$

где C_i — цена единицы условного комплекта;

– K_i — количество единиц условного комплекта;

– P_d — дополнительные расходы на доставку, установку, первоначальную наладку.

$$P_d = 0,1 \sum C_i K_i, \quad (4)$$

$$P_d = 0,1 * 30000 = 3000(\text{руб}).$$

Затраты на материалы:

$$Z_m = 0,02 * C_{\text{бал}}, \quad (5)$$

$$Z_m = 0,02 * 33000 = 660(\text{руб}).$$

Основная и дополнительная заработная плата персонала, обслуживающего ВиОТ:

$$Z_{z/p} = \frac{\sum O_i * 12 * (1 + K_d) + (1 + K_p)}{N_{\text{Обе.ПК}}}, \quad (6)$$

где O_i — оклад i -го работника, обслуживающего ВиОТ;

– $K_d = 0,1$ — коэффициент дополнительной заработной платы;

– $K_p = 0,15$ — районный коэффициент.

Начисления на заработную плату – $Z_{z/p}$.

$$Z_{z/p} = \frac{10000 * 2 * (1 + 0,1) + (1 + 0,15)}{1} = 45000(\text{руб})$$

Износ программных продуктов.

Условный комплект обладает следующими необходимыми программными продуктами, которые приведены в таблице 3.

Таблица 2 — Используемые программные средства

Наименование	Цена (руб.)
SublimeText3	5026
Виртуальный сервер	3000
Доменная зона	220
Ssl сертификат	1500
Итого	9746

Расходы на содержание и эксплуатацию оборудования состоят из:

– затрат на ремонт и специальное обслуживание

$$Z_{рвм} = 0,03 * C_{бал}, \quad (8)$$

$$Z_{рвм} = 0,03 * 33000 = 990(\text{руб});$$

– затрат на электрическую энергию

$$Z_{э} = \sum M * F_n * K_{г} * Ц_{квт/ч}, \quad (9)$$

$$Z_{э} = 0,25 * 4161 * 0,95 * 4,6 = 4545(\text{руб}),$$

где $\sum M$ — суммарная мощность;

– $K_{г}$ — коэффициент готовности;

– $Ц_{квт/ч}$ — стоимость кВт/ч.

Расходы на содержание и эксплуатацию оборудования

$$Z_{рэ} = Z_{рвм} + Z_{э}, \quad (10)$$

$$Z_{рэ} = 990 + 4545 = 5535(\text{руб}).$$

Прочие расходы.

Так как используется один тип техники и количество обслуживающего персонала не превышает трех человек, следовательно:

$$Z_{пр} = (0,4 \dots 0,8) * C_{бал}, \quad (11)$$

$$Z_{пр} = 0,4 * 33000 = 13200(\text{руб}).$$

Суммарные годовые затраты.

$$\sum Z_i = Z_m + Z_{з/п} + H_{з/п} + Z_{рэ} + Z_{пр}, \quad (12)$$

$$\sum Z_i = 660 + 45000 + 13590 + 5535 + 13200 + 9746 = 87641(\text{руб}).$$

2 Разработка программного обеспечения на основе технического задания дипломного проекта

2.1 Проектирование и разработка интерфейса

2.1.1 Проектирование диаграммы компонентов

Диаграмма компонентов описывает особенности физического представления системы. Она помогает разработчику системы определить зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

Для системы проведения on-line олимпиад была спроектирована диаграмма компонентов, представленная на рисунке 1.



Рисунок 1 — Диаграмма компонентов

Данная диаграмма показывает структуру и взаимодействие отдельных модулей в системе между собой. На диаграмме присутствует компонент «web-сайт» объединяющий собой элементы форм и страниц.

Страницы web-сайта:

- главная;
- страница олимпиады;
- рейтинг;
- FAQ;
- контакты;
- о нас.

Формы web-сайта:

- форма регистрации;
- форма авторизации.

2.1.2 Проектирование диаграммы последовательности

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности. Взаимодействующие объекты обмениваются между собой некоторой информацией. При этом информация принимает форму законченных сообщений. Другими словами, хотя сообщение и имеет информационное содержание, оно приобретает дополнительное свойство оказывать направленное влияние на своего получателя.

Для более точного понимания назначения форм и страниц разработаем

										Лист
										21
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19					

диаграмму последовательности, представленную на рисунке 2.

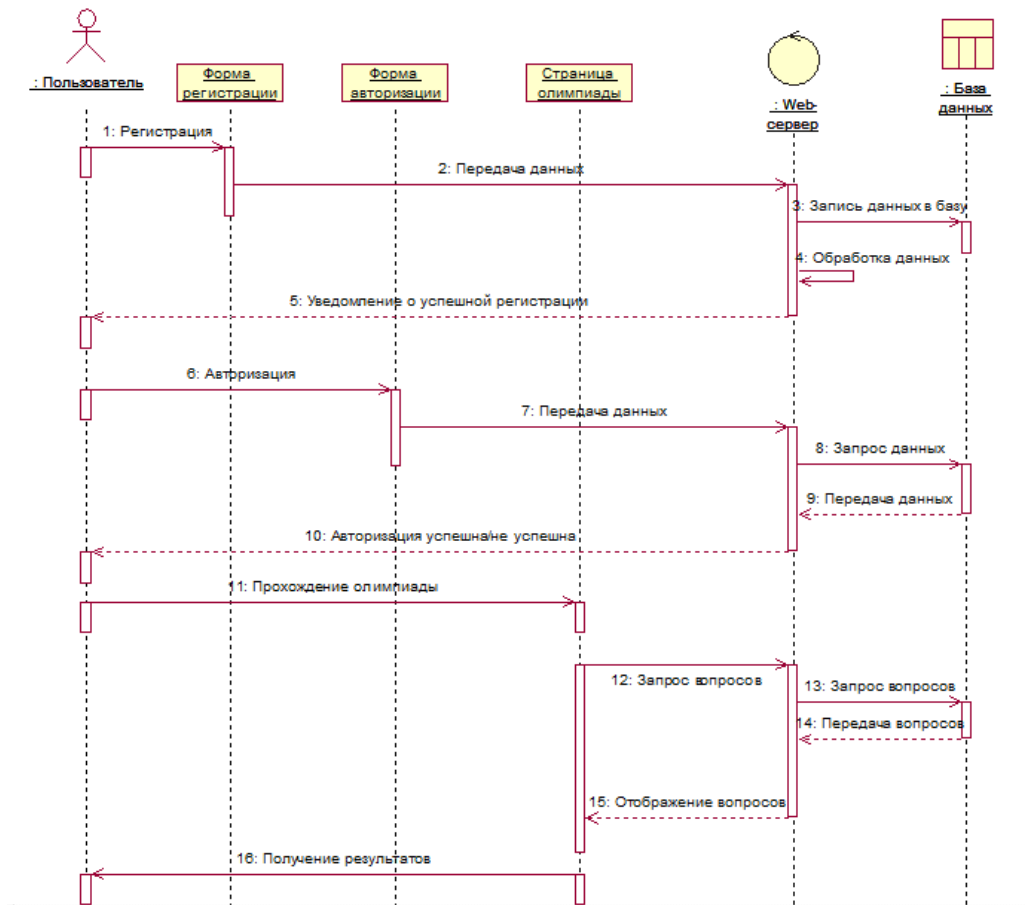


Рисунок 2 — Диаграмма последовательности

На разработанной диаграмме изображена последовательность действий и операций, происходящих в результате работы программы. Последовательность действий, выполняемых на клиенте:

- регистрация;
- авторизация;
- выполнение олимпиады;
- получение результата.

2.1.3 Проектирование диаграммы активности

Диаграмма активностей (видов деятельности) — один из доступных видов диаграмм. Она, как и диаграмма состояний, отражает динамические аспекты поведения системы. По существу эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

Спроектируем диаграмму активности для реализации взаимодействия между страницами и формами представленную на рисунке 3.

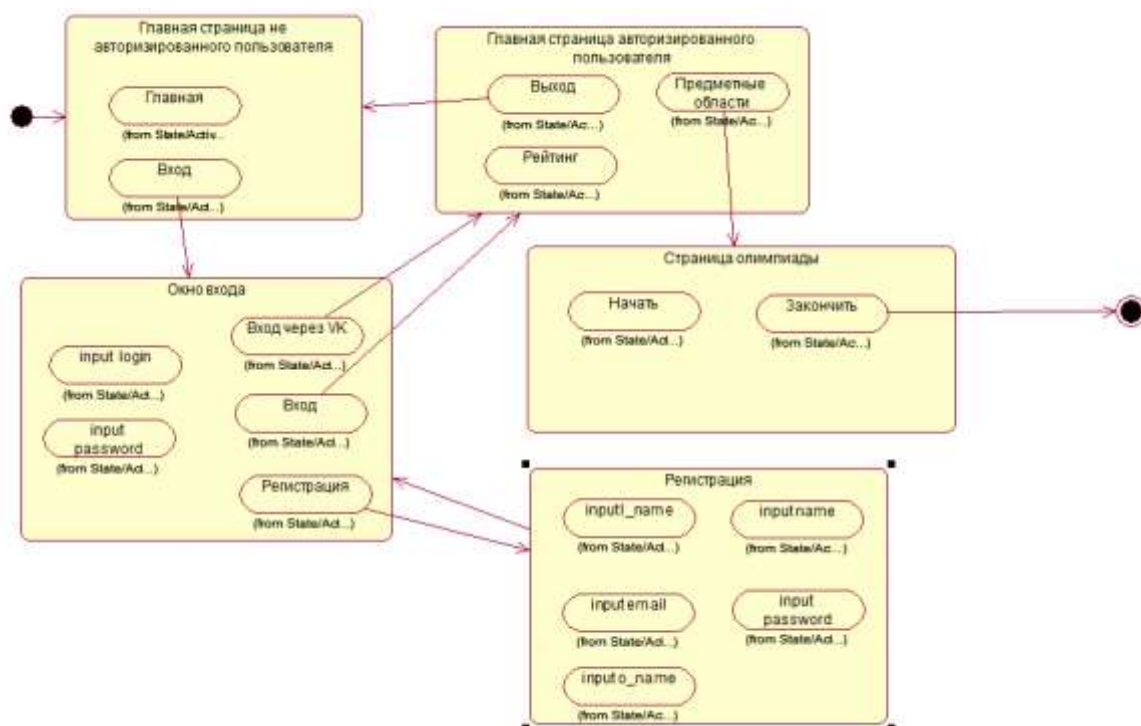


Рисунок 3 — Диаграмма активности

Данная диаграмма включает в себя страницы web-сайта и взаимодействие между ними с помощью кнопок и ссылок.

2.1.4 Разработка пользовательского интерфейса

Для проектирования пользовательского интерфейса будем использовать бесплатный растровый графический редактор «Krita». Среди преимуществ данного графического редактора можно выделить:

- полноценные инструменты для работы с покадровой анимацией;
- широкий выбор кистей (в том числе смешивающие, фильтрующие, эффектные, спрей, кисти для заполнения объемов);
- большое количество режимов наложения;
- управление динамикой кистей с помощью графического планшета;
- имитацию бумаги и пастели;
- поворот и зеркалирование холста;
- псевдо-бесконечный холст;
- поддержку горячих клавиш Photoshop и SAI.

В результате после анализа диаграмм, предметной области и выбора инструментальных средств был спроектирован пользовательский интерфейс приложения, представленный на рисунке 4.

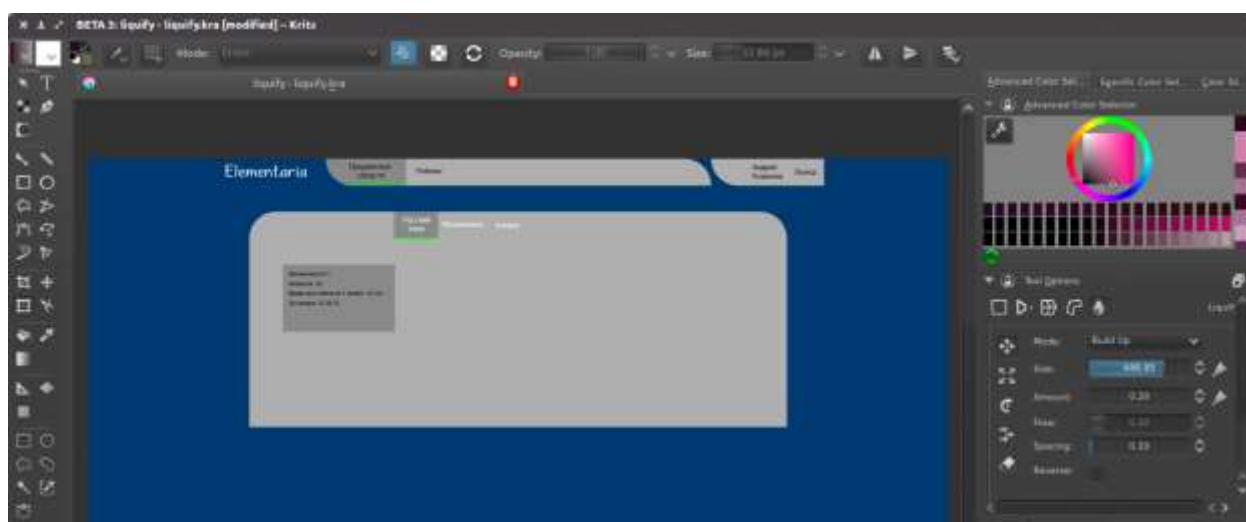


Рисунок 4 — Разработанный пользовательский интерфейс

При разработке пользовательского интерфейса учтены все объекты, разработанные в диаграмме компонентов. В качестве основных цветов были выбраны синий (с кодом #003d71) для фона и серый (с кодом #adadad) для отображения форм и блоков. Данные цвета сочетаются друг с другом образовывая приятную цветовую палитру удобную для пользователя. Проанализировав разработанную диаграмму последовательности, навигационный блок был размещен вверху сайта и всегда виден пользователю, что позволяет сократить количество действий, требуемых для достижения результата.

В результате разработан пользовательский интерфейс, удовлетворяющий поставленным требованиям и функционалу. А также из-за того, что данный пользовательский интерфейс проектировался на основе диаграмм компонентов и последовательности, удовлетворяет требованиям UX и UI дизайна.

2.2 Внедрение пользовательского интерфейса

2.2.1 Создание интерфейса пользователя на основе HTML-форм

Так как проект представляет собой web-сайт, пользовательский интерфейс реализуется с помощью стека технологий: HTML, CSS, JS, а также с использованием шаблонизатора «*twig*». Для внедрения пользовательского интерфейса воспользуемся бесплатным текстовым редактором «*Sublime Text 3*», внешний вид которого с кодом главного исполняемого файла изображен на рисунке 5 [10].

						ДП-02069964-09.02.03-13-19	Лист 25
Изм.	Лист	№ докум.	Подпись	Дата			

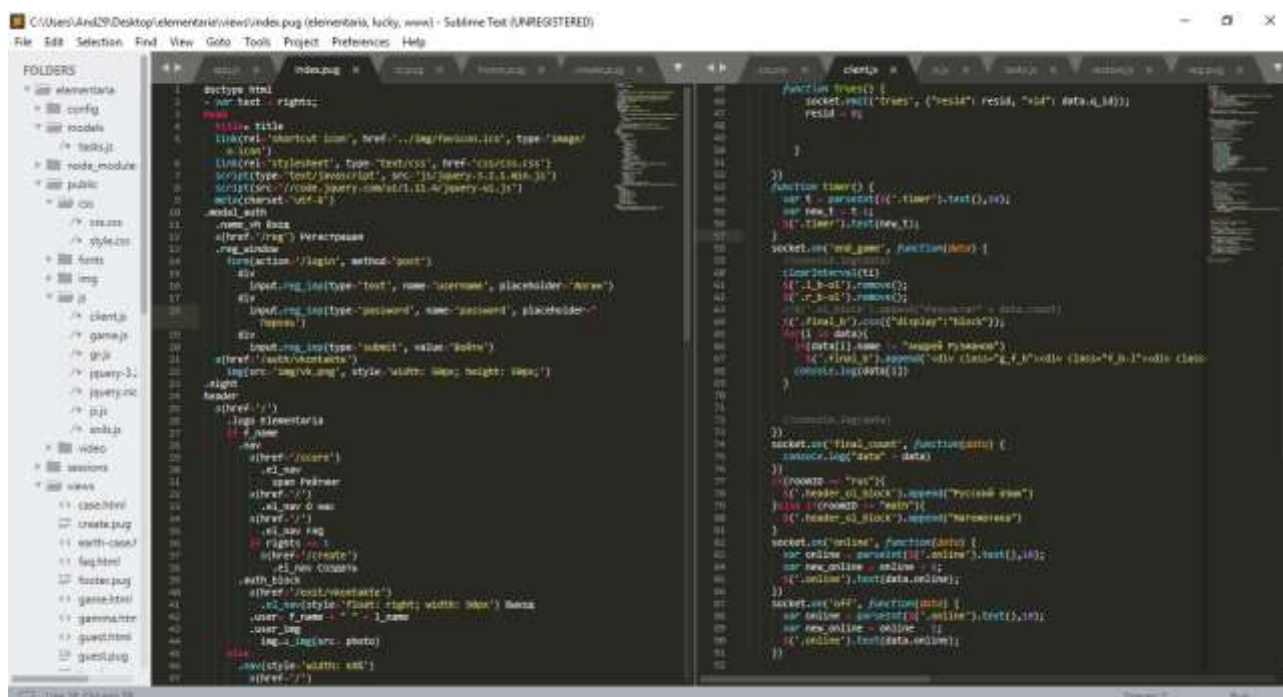


Рисунок 5 — Графический интерфейс «Sublime Text 3»

После реализации пользовательского интерфейса необходимо внедрить его в проект. Так как проект представляет собой web-сайт, то пользовательский интерфейс будет представлять собой HTML формы измененные шаблонизатором «pug» [2].

В результате, на данном этапе сконструирован HTML каркас страниц и форм программного продукта. Листинг HTML форм указан в приложение А.

2.2.2 Внедрение CSS кода для HTML-форм

Разработанных HTML форм недостаточно для полноценного и удобного функционирования системы, поэтому требуется добавить к разработанному HTML каркасу каскадные таблицы стилей (CSS).

После написания CSS кода, внедрение графического интерфейса почти закончено, пользовательский интерфейс изображен на рисунке 6.

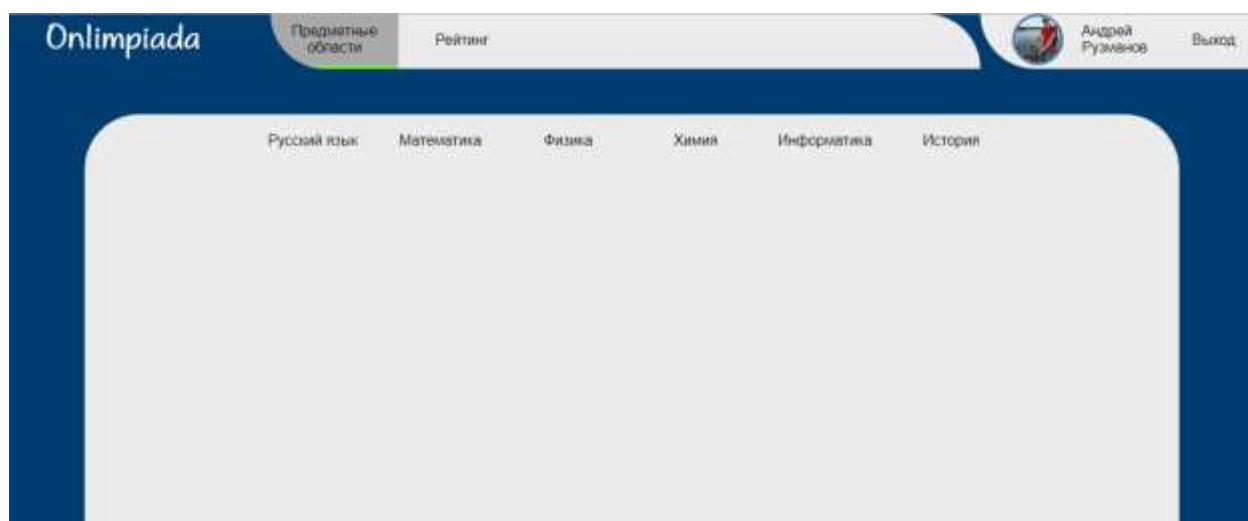


Рисунок 6 — Пользовательский интерфейс

В процессе оформления пользовательского интерфейса использованы следующие ключевые элементы:

- основной шрифт — «Arial»;
- шрифт логотипа — «BauhausITC»;
- лаконичное закругление блоков в 60px для придания интерфейсу целостности;
- минималистичное оформление кнопок;
- практичное расположение всех элементов на странице.

В результате к уже готовым HTML-формам были добавлены каскадные таблицы стилей, и интерфейс стал удобным для пользователя.

2.2.3 Добавление JavaScript кода

Для полноценного функционирования пользовательского интерфейса, он должен быстро и просто (для пользователя) выполнять поставленные задачи, поэтому с помощью языка программирования «JavaScript» реализован процесс передачи данных между клиентом и сервером [2].

Также в разработанном пользовательском интерфейсе присутствует модульное окно для авторизации, поэтому воспользуемся JavaScript для реализации анимации модульного окна, представленного на рисунке 7.



Рисунок 7 — Модульное окно авторизации

2.3 Проектирование серверной части и логики сервера

Сервер приложения — это главный файл программы, который обрабатывает входящие данные, реализует логику приложения и выполняет различные функции [7].

В данном случае сервер будет написан на серверной платформе Node.js с использованием языка программирования JavaScript. Он прост в изучении и написании кода, а также на нем можно реализовать все необходимые задачи. Node.js является модульной системой, поэтому понадобятся определенные модули для решения различных задач. Модуль — это функциональный блок какой-либо программы. Установка модулей в Node.js происходит через команду «npm install».

Для решения поставленных задач требуется реализовать следующие модули:

- регистрация и авторизация пользователей;
- роутинг;
- модуль подключения и работы с базой данных;
- обработчик входных и выходных данных.

Для реализации серверной части необходимы следующие модули:

– express. Это — web-фреймворк на Node.js, который применяется для создания web-сервисов. Для создания сервера используем подключение модуля объявляя переменную и присваивая ей модуль «express» «var express = require('express');». Далее передаем в переменную функцию модуля «var app = express();». Теперь можно запустить сервер «app.listen(3000, 'localhost');». Порт был выбран 3000, так как он находится в диапазоне незарезервированных портов и доступен для использования.

- consolidate. Модуль необходимый для реализации роутинга на сайте.

Чтобы сервер видел нужные файлы для загрузки их на страницу, необходимо указать путь к директории с этими файлами. Для указания директории с web-страницами пользуемся модулем «app.set('views', __dirname + '/views');». Заодно указывается путь к статичным файлам «app.use(express.static(__dirname + '/public'));».

– andlebars. Handlebars — это шаблонизатор, используемый для подгрузки web-страниц, передачи и отображения данных на клиенте. Объявление

«`app.engine('html', templating.handlebars); app.set('view engine','html');`». Теперь сервер сможет подгружать web-страницы и передавать в них данные. Для отображения данных требуется передать переменную с помощью шаблонизатора на клиент. Вывод переменной осуществляется с помощью конструкции «`{{переменная}}`». Также есть возможность работать с циклами и условиями.

– `passport-local`. Данный модуль позволяет реализовать локальную авторизацию и регистрацию. Для подключения модуля необходимо объявить переменную и присвоить ей путь к модулю и обозначить стратегию «`var LocalStrategy = require('passport-local').Strategy;`». Теперь для использования функции авторизации или регистрации будет использоваться функция модуля «`passport.use(new LocalStrategy(function(username, password, done) {...}))`» для приема данных с клиента.

– `passport-vkontakte`. Модуль позволяющий авторизоваться на сайте через социальную сеть «ВКонтакте». Подключение модуля с объявлением стратегии «`const VKontakteStrategy = require('passport-vkontakte').Strategy;`». По аналогии с предыдущим модулем используем стандартную функцию и задаем стратегию «`passport.use(new VKontakteStrategy({...}))`».

– `express-session`. Позволяет работать с сессиями. Сессия — это данные, хранящиеся на клиенте, которые позволяют ему взаимодействовать с сервером. Для создания сессии необходимо подключить модуль «`var session = require('express-session');`». Далее используем функцию нашего приложения `app.use(session({...}))` и назначаем сессию, время и параметры. Эта функция будет автоматически принимать параметры, переданные из функций авторизации.

– `body-parser`. Данный модуль нужен для преобразования данных, приходящих с клиента в json формат, для более удобной обработки данных. JSON — формат, основанный на JavaScript представляющий собой ассоциативный массив данных.

					<i>Лист</i>
					<i>30</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>ДП-02069964-09.02.03-13-19</i>

Для работы с базой данных требуется использовать модуль «mysql». Подключение «var mysql = require('mysql');», затем используя функцию модуля, создаем пул соединений с базой данных, передавая в нее необходимые данные «var pool = mysql.createPool({...})». Для оптимизации кода создадим массив объектов «Tasks», в котором будут храниться функции для обращения к базе данных. Теперь вынесем код в отдельный файл под названием «tasks.js» и экспортируем массив «Tasks», после этого в главном файле сервера подключим модуль «var tasks = require('./models/tasks');».

Обработку входных и выходных данных будет осуществлять «express» сервер. Используя переменную «app», которая хранит функцию сервера, и метод запроса сервер может принимать и передавать данные на клиент и обратно «app.get('/', function(req, res){...})», параметр «req» отвечает за входящие данные на сервер, а «res» за отправку данных клиенту. Так, например, для того чтобы отправить клиенту web-страницу, необходимо воспользоваться функцией «render» и передать название файла «res.render('login')».

Используя все эти модули, можно реализовать удобную и результативную серверную логику с возможностью усовершенствования ее в дальнейшем.

Для реализации процесса многопользовательского одновременного прохождения олимпиады требуется реализовать возможность удобного и быстрого взаимодействия пользователей с сервером и друг с другом. Для этой задачи идеально подойдет «Socket.io», основанный на работе протокола «Web socket». Веб-сокеты (Web Socket) — это протокол, который создает интерактивное соединение между клиентом и сервером для обмена данными в режиме реального времени. В отличие от HTTP протокола, который построен на модели «запрос – ответ», веб-сокеты позволяют установить соединение между клиентом и сервером и сохранять его пока это необходимо, что позволяет

значительно ускорить обмен данными между пользователями и сервером [5].

Для начала использования основного модуля приложения и установления первоначального соединения необходимо подключить модуль «`var io = require('socket.io').listen(app.listen(3000, 'localhost'));`», затем установить первоначальное соединение с веб-сокетом «`io.on('connection', function(client) { ... })`». В данной безымянной функции и будут происходить все взаимодействия между клиентами и сервером. Принимаемый параметр «client» содержит передаваемые параметры приходящие с клиента на сервер.

Общение клиентов с сервером происходит посредством вызываемых функций:

- `socket.on()` — отправляет данные всем кто подключен к сокету;
- `socket.join('some room')` — подключение к комнате;
- `socket.broadcast.to(id).emit('my message', msg)` — отправка сообщения в определённую комнату.

Благодаря внедрению данной технологии улучшилась передача данных и оптимизировалась работа системы в целом. Листинг кода серверной части указан в приложении Б.

2.4 Разработка и создание базы данных

База данных необходима для хранения данных о пользователях, вопросах и истории проведения олимпиад. Данные не обладают большим размером, но их будет храниться много. Для этой задачи подойдет реляционная система управления базами данных MySQL. Она подходит для малых и средних приложений.

При создании модели данных используется СУБД MySQL. Для администрирования базы данных PHP MyAdmin. MySQL является одной из самых популярных баз данных в мире. Работа с базой данных MySQL осуществляется с помощью SQL запросов. Это позволяет осуществлять такие операции, как добавление данных в базу, удаление, извлечение и редактирование данных. Система управления базами данных MySQL может поддерживать огромное количество таблиц и типов данных.

При создании базы данных потребуется реализовать минимум 3 таблицы. В процессе администрирования проекта будут добавляться новые таблицы.

Таблица users, представленная на рисунке 8, содержит поля: id, uid, login, name, l_name, password.

id	uid	f_name	l_name	sex	bdate	city	rights
1	31724996	Андрей	Рузманов	2	29.6.1999	Саранск	1
2	493166387	Андрей	Иванов	2	29.6.1999		0
3	18276949	Александр	Прокин	2	11.12.1990	Саранск	0
4	264972828	Андрей	Корневалов	2	26.1.1998	Москва	0
5	356746742	Александр	Ерушев	2			0
6	147628180	Павел	Плеханов	2	19.8.1997	Саранск	0

Рисунок 8 — Таблица пользователей

id — уникальное поле, с автоинкрементном, имеющее тип данных «int» и длину поля 255 символов, служит для идентификации пользователя.

uid — уникальное поле, содержащее уникальный идентификатор пользователя из социальной сети «Вконтакте», имеющее тип данных «int» и длину поля 255 символов, также служит для идентификации пользователя.

login — поле, содержащее логин пользователя для авторизации на сайте, имеющее тип данных «varchar» и длину поля 255 символов, служит для возможности авторизации на сайте.

name — поле, содержащее имя пользователя, имеющее тип данных «varchar» и длину поля 255 символов, служит для идентификации авторизированных пользователей на сайте.

l_name — поле, содержащее фамилию пользователя, имеющее тип данных «varchar» и длину поля 255 символов, также служит для идентификации авторизированных пользователей на сайте.

password — уникальное поле, содержащее пароль пользователя, имеющее тип данных «varchar» и длину поля 255 символов, служит для авторизации пользователей на сайте.

Таблица questions, представленная на рисунке 9, содержит поля: q_id, question, res1, res2, res3, trues.

q_id	question	res1	res2	res3	trues	type
1	0 Укажите слово, в котором три мягких согласных звук...	1) жизнь	2) семья	3) брошюра	2	rus
1	1 Укажите слово, в котором количество звуков указано...	1) бьются — 5 зв.	2) юноша — 5 зв.	3) грустно — 7 зв.	1	rus
2	2 Отметьте, в каком слове звуков больше, чем букв:	1) семья	2) повесть	3) проезд	3	rus
3	3 Определите, в каком слове все согласные звуки твёр...	1) живёт	2) жила	3) час	2	rus
4	4 Определите, в каком слове все согласные звуки твёр...	1) автосалон	2) стоять	3) хищник	1	rus
5	5 Определите, в каком слове звуков больше, чем букв:	1) карьера	2) если	3) любой	2	rus
6	6 Определите, в каком слове все согласные звуки твёр...	1) молод	2) борщ	3) мощный	1	rus
7	7 Отметьте правильное высказывание:	1) Фонетика – раздел языкознания, изучающий звуки.	2) Фонетика – раздел языкознания, изучающий произн...	3) Фонетика – раздел языкознания, изучающий правоп...	1	rus
8	8 Отметьте сложное предложение, в котором вы постави...	1) Добавь радость к заботам.	2) Жить с женой – дело нелёкое но жить без неё во...	3) Живи помня как коротка жизнь.	3	rus

Рисунок 9 — Таблица вопросов по русскому языку

q_id — уникальное поле с автоинкрементом, имеющее тип данных «int» и длину поля 255 символов, служит для идентификации вопроса в базе данных, а также для удобного использования на сервере.

question – текстовое поле, содержащее вопрос по определенному предмету и имеющее тип данных «varchar» и длину поля 255 символов, служит для процесса олимпиады.

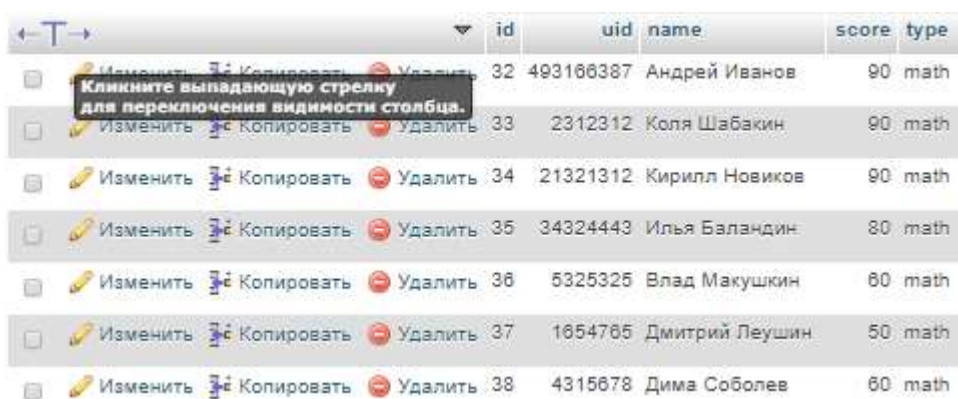
res1 — текстовое поле, содержащее первый ответ на вопрос по определенному предмету, имеющее тип данных «varchar» и длину поля 255 символов, служит для процесса олимпиады.

res2 — текстовое поле, содержащее второй ответ на вопрос по определенному предмету, имеющее тип данных «varchar» и длину поля 255 символов, служит для процесса олимпиады.

res3 — текстовое поле, содержащее третий ответ на вопрос по определенному предмету, имеющее тип данных «varchar» и длину поля 255 символов, служит для процесса олимпиады.

trues — поле, содержащее номер правильного ответа на вопрос пользователя, имеющее тип данных «int» и длину поля 255 символов, служит для процесса олимпиады.

Таблица memory, представленная на рисунке 10, содержит поля: id, uid, name, score, type.



	id	uid	name	score	type
<input type="checkbox"/>	32	493166387	Андрей Иванов	90	math
<input type="checkbox"/>	33	2312312	Коля Шабакин	90	math
<input type="checkbox"/>	34	21321312	Кирилл Новиков	90	math
<input type="checkbox"/>	35	34324443	Илья Баландин	80	math
<input type="checkbox"/>	36	5325325	Влад Макушкин	60	math
<input type="checkbox"/>	37	1654785	Дмитрий Леушин	50	math
<input type="checkbox"/>	38	4315678	Дима Соболев	60	math

Рисунок 10 — Таблица истории проведения олимпиад

`id` — уникальное поле, с автоинкрементном, имеющее тип данных «`int`» и длину поля 255 символов, служит для структурирования результатов олимпиады.

`uid` — уникальное поле, имеющее тип данных «`int`» и длину поля 255 символов, служит для хранения идентификатора пользователя и идентификации.

`name` — текстовое поле, содержащее имя и фамилию пользователя, имеющее тип данных «`varchar`» и длину поля 255 символов, служит для вывода результатов.

`score` — числовое поле, содержащее количество баллов пользователя по конкретному предмету, имеющее тип данных «`int`» и длину поля 255 символов, служит для хранения истории.

`type` — поле, содержащее название олимпиады, имеющее тип данных «`varchar`» и длину поля 255 символов, служит для идентификации предмета.

Подключение к базе данных осуществляется с помощью модуля написанного на языке JavaScript.

2.5 Реализация защиты данных в базе

2.5.1 Защита базы данных на уровне операционной системы

Ограничение доступа к серверу является одним из самых действенных способов защиты данных. Не имея доступа к серверу, злоумышленники не смогут каким-либо способом нарушить работу сервера путем кражи или внесения вредоносных файлов.

В качестве контроля доступа к серверу будут использоваться стандартные функции Linux систем. Для контроля доступа и успешных входов в систему

используется файл «wtmp», который содержит данные о успешных подключениях к серверу.

Для того чтобы убедиться, что удаленных несанкционированных подключений к серверу не было, осуществим вывод логов успешных входов в систему с помощью команды «last -f /var/log/wtmp». В результате работы данной функции сгенерировался список успешных входов в систему, часть которых представлена в таблице 3.

Таблица 3 — Список успешных подключений к серверу

root	pts/0	95-83-61-225.sar	Sun	Mar	17	(03:14)
admin	Ftdp3042	95-83-61-225	Sun	Mar	12	(00:00)
root	pts/0	95-83-61-225.sar	Sun	Mar	12	(01:52)
root	pts/1	95-83-61-225.sar	Sun	Mar	10	(00:02)
reboot	System boot	95-83-61-225.sar	Sun	Mar	10	(00:02)

Для отслеживания неудачных подключений откроем файл «btmр» командой last -f /var/log/btmp. Данный файл содержит информацию о неудачных подключениях к серверу, который включает в себя логины, вводимые при попытке входа, ip-адреса и дату. После выполнения данной команды обнаруживаем большое количество попыток входа с разными логинами. Это означает, что злоумышленники уже пытались подобрать логин и пароль для авторизации на сервере путем перебора простейших логинов и паролей. В таблице 4 видно, что с одного ip-адреса «88.214.173.10» осуществлялось множество неудачных входов в систему под разными логинами, это означает, что злоумышленники пользовались программным обеспечением для перебора логинов и паролей. После чего было предпринято решение сделать резервную копию проекта с базой данных и переустановить операционную систему и сменить все стандартные порты web-сервера.

Таблица 4 — Список неудачных подключений к серверу

man	ssh:notty	127.247.194.58	Wed	Mar	6	(00:00)
agencia	ssh:notty	88.214.173.10	Wed	Mar	6	(00:00)
agencia	ssh:notty	88.214.173.10	Wed	Mar	6	(00:00)
tom	ssh:notty	59.145.221.103	Wed	Mar	6	(00:00)
admin	ssh:notty	88.214.173.10	Wed	Mar	6	(00:00)
root	ssh:notty	118.45.130.250	Wed	Mar	6	(00:00)

Для отслеживания установленных пакетов на сервер будем отслеживать логи в файле yum.log. Данная процедура необходима для отслеживания установленных файлов и пакетов на сервер, что позволяет отслеживать внесение неизвестного или вредоносного программного обеспечения на сервер. Листинг установленных пакетов представлен на рисунке 11.

Правильное использование паролей является важным методом защиты баз данных и не только. Для безопасной работы системы требуется постоянное поддержание паролей в актуальном состоянии. Частая смена пароля является недостатком. Режим, когда смена пароля требуется каждые 2-3 недели не эффективен, так как высока вероятность того, что пользователи будут использовать максимально легкие пароли для запоминания, тем самым снижая безопасность системы [4].

```

yum.log
1 Mar 09 20:31:31 Installed: 2:nodejs-10.15.3-1nodesource.x86_64
2 Mar 09 20:32:04 Updated: glibc-common-2.17-260.el7_6.3.x86_64
3 Mar 09 20:32:06 Updated: glibc-2.17-260.el7_6.3.x86_64
4 Mar 09 20:32:07 Installed: mpfr-3.1.1-4.el7.x86_64
5 Mar 09 20:32:07 Installed: libmpc-1.0.1-3.el7.x86_64
6 Mar 09 20:32:08 Installed: cpp-4.8.5-36.el7.x86_64
7 Mar 09 20:32:09 Installed: libstdc++-devel-4.8.5-36.el7.x86_64
8 Mar 09 20:32:11 Installed: kernel-headers-3.10.0-957.5.1.el7.x86_64
9 Mar 09 20:32:12 Installed: glibc-headers-2.17-260.el7_6.3.x86_64
10 Mar 09 20:32:12 Installed: glibc-devel-2.17-260.el7_6.3.x86_64
11 Mar 09 20:32:15 Installed: gcc-4.8.5-36.el7.x86_64
12 Mar 09 20:32:16 Installed: gcc-c++-4.8.5-36.el7.x86_64
    
```

Рисунок 11 — Листинг установленных пакетов

2.5.2 Защита базы данных на уровне СУБД

Контролировать права пользователей необходимо для стабильной работы системы и ее защиты. Важно, чтобы пользователь не имел доступа к данным, не предназначенным для него.

Изначально в СУБД нет ни одного пользователя, кроме администратора. Для полноценного функционирования системы одного администратора недостаточно. Создадим пользователя с именем «user_1» с помощью команды «CREATE USER 'user_1'@'localhost' IDENTIFIED BY 'user1user'» и установим ему доступ для просмотра и редактирования значений в таблицах связанных с вопросами олимпиады с помощью команд «GRANT UPDATE ON db.mathquestins TO 'user1'@'localhost'» и «GRANT SELECT ON db.mathquestins TO 'user1'@'localhost'».

В данном случае пользователю будут необходимы только права на просмотр таблиц и изменения их значений. Тем самым исключается возможность изменения структуры таблиц, их удаление и добавление новых полей у пользователя. В результате, при попытке удаления таблицы или других действий, не разрешенных пользователю, у него будет возникать ошибка с информацией о недостаточности привилегий, показанная на рисунке 12.

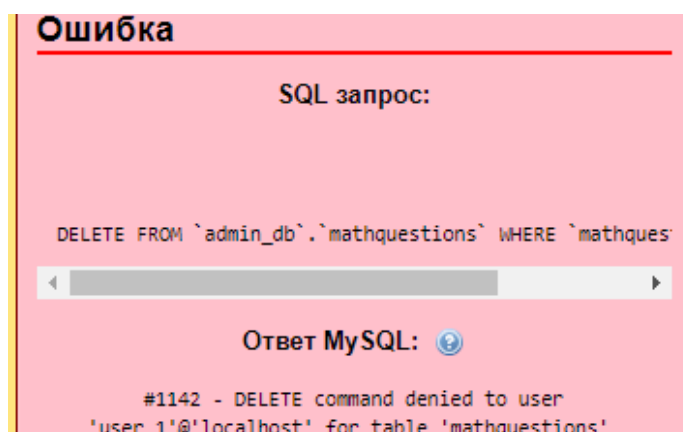


Рисунок 12 — Ошибка из-за отсутствия привилегий

Резервное копирование имеет огромную роль на протяжении всего жизненного цикла проекта. Ситуаций, из-за которых может понадобиться резервная копия, достаточно много от банальной неосторожности до физического уничтожения сервера.

Сделать резервную копию базы можно с помощью команды «mysqldump». Данный метод позволяет скопировать базу данных со всей структурой, данными и настройками в формате sql или в архиве. Данный метод не особо результативный, так как требует постоянного ручного резервного копирования. В бесплатной панели управления сервером VestaCP реализована функция ежедневного резервного копирования. В результате на сервере ежедневно сохраняется архив с резервной копией web-сервера и СУБД со всей структурой, и данными.

Но после значительных изменений необходимо сохранять резервную копию вручную на облачное хранилище или какой-либо другой носитель.

2.5.3 Криптографическая защита данных

После того, как определились, какие виды шифрования бывают и зачем нужно шифрование, проанализируем данные из базы на предмет защиты. В базе данных «db» имеется таблица пользователей «users», таблицы с вопросами по различным предметам «questions» и таблица с результатами «score». Приходим к выводу: самая конфиденциальная информация — это пароли пользователей в таблице «users», в которой пароли хранятся в открытом виде, что не допустимо. Заполучив доступ к базе данных, злоумышленники будут иметь доступ к аккаунтам пользователей, но так как некоторые пользователи устанавливают один и тот же пароль на почту и учетные записи, злоумышленники могут заполучить доступ и к другим

									Лист
									40
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19				

аккаунтам пользователей. Тем самым нарушается конфиденциальность хранящейся информации.

Для хэширования пароля будем использовать «соль» с генерацией уникальных ключей. «Соль» представляет собой алгоритм, который генерирует уникальный ключ из заданных параметров путем многократного хеширования. Далее из нашего ключа и пароля пользователя, используя алгоритм «sha256» создаем уникальный хэш. В результате вместо обычного пароля в базе данных хранится уникальный хэш. Теперь, из-за «соли», исключается появление одинаковых хэшей у людей с идентичными паролями.

В результате проделанных действий получаем в таблице «users» в поле «password» вместо пароля, в открытом виде, уникальный хэш, представленный на рисунке 13, на расшифровку которого у злоумышленников уйдет очень много времени, и даже если они смогут расшифровать какую-либо запись и узнать ключ, с помощью него невозможно будет получить доступ к другим записям. Листинг модуля шифрования данных представлен в приложении В.

id	uid	login	name	l_name	password
30	0	And290699@gmail.com	Андрей	Рузманов	6679ea8918ee3e316ee54796e3268b5e3eb5fc9f3443d8abc6...

Рисунок 13 — Запись пользователя в таблице «users»

2.5.4 Шифрование передаваемых данных

Удобным методом шифрования передаваемых данных будет ssl сертификат. Ssl — это криптографический протокол который использует асимметричную криптографию для аутентификации ключей обмена,

симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений [4].

Проект расположен на виртуальном сервере и приобретен домен с бесплатным ssl сертификатом на год. В качестве прокси сервера установлен Nginx, поэтому результативней будет ssl сертификат устанавливать на него. После установки ssl сертификата на сайте в строке поиска появился протокол «https» и возможность посмотреть сертификат (рисунок 14) [2].

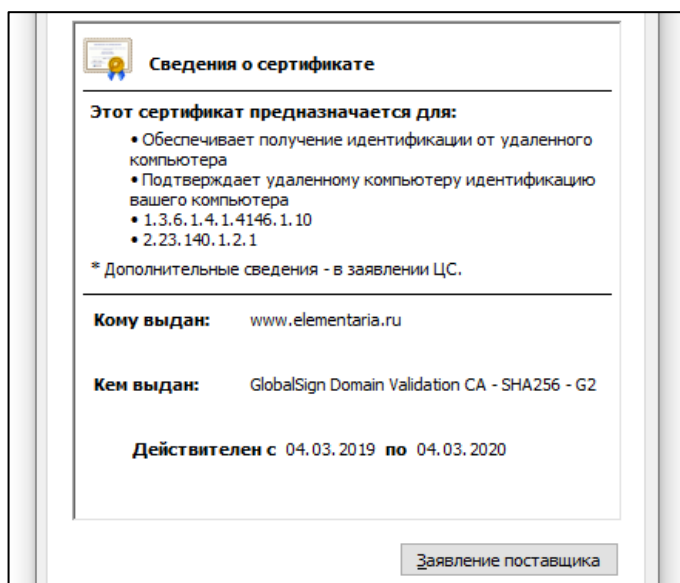


Рисунок 14 – Ssl сертификат

В результате подключенный ssl сертификат будет шифровать передаваемые данные.

2.5.5 Экранирование sql запросов

Sql инъекция является одним из самых распространенных способов взлома программ и сайтов работающих с базами данных, основанный на внедрении

в запрос SQL-кода.

Основные способы отправки sql кода на сервер — это формы, используемые на сайте, а также поисковая строка. Для отправки sql запроса через форму используется конкатенация. К строке, которая завершает запрос, прибавляется вредоносный код, производящий какие-либо действия с базой данных. В итоге злоумышленники могут завладеть, исказить или даже стереть информацию в базе данных [8].

Для защиты от таких атак используют разные методы, но наиболее действенный и максимально простой — экранирование передаваемых данных. Данный метод позволяет экранировать данные, в результате чего если не обрабатывать входные данные, они будут записываться в базу данных как значения, тем самым исключая какие-либо изменения в базе данных. Все возможные спецсимволы не влияют на запрос [3].

Теперь перейдем непосредственно к модулю, содержащему функции обращения к базе данных. Запрос к базе данных генерируется из передаваемых параметров, показанных на рисунке 15. Данный способ не является защищённым, так как в запрос можно интегрировать еще один запрос или изменить этот, поэтому необходимо внедрить защиту для передаваемых запросов.

```
connection.query("INSERT INTO `admin_db`.`users` (`id`, `uid`, `login`, `name`, `l_name`, `password`)  
VALUES (NULL, " + uid + ", '" + name + "', '" + name + "', '" + l_name + "', '');" , callback);
```

Рисунок 15 — Запрос к базе данных

Для защиты запросов будем экранировать передаваемые данные, используя конструкцию библиотеки mysql «INSERT INTO SET» с параметром callback, который содержит набор необходимых данных, передаваемых в функцию.

После всех манипуляций с кодом запрос стал значительно короче и больше не уязвим для sql-инъекций. Демонстрация нового sql-запроса показана на рисунке 16.

```
connection.query('INSERT INTO `users` SET ?;', callback)|
```

Рисунок 16 — Экранированный sql запрос

Как видно на рисунке, запрос стал выглядеть короче и не содержит в себе ничего лишнего. С помощью символа «?» экранируются данные из массива объектов «callback» и подставляются в запрос. Данный метод позволяет передавать данные в запросе как значения, а не целой строкой.

В результате проведенного теста данный метод полностью экранирует строку и не обращает внимания на спецсимволы, записывая данные в базу обычной строкой и не выполняя код, переданный в этой строке.

									Лист
									44
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19				

3 Проведение испытаний, отладка и внедрение программного продукта

3.1 Проведение функционального тестирования

Функциональное тестирование — это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования описывают сервисы, предоставляемые программной системой, ее поведение в определенных ситуациях, реакцию на те или иные входные данные и действия, которые система позволит выполнять пользователям. Иногда сюда добавляются сведения о том, чего система делать не должна [11].

Поэтому был составлен чек-лист для функционального тестирования, листинг которого предоставлен в приложение Г.

В результате составления чек-листа и проведения функционального тестирования не выявлено критических ошибок и реализована вся функциональность программного продукта.

3.2 Отладка и внедрение программного продукта

Для внедрения программного продукта требуется подобрать хостинг с подходящим виртуальным сервером и достаточным количеством ресурсов для внедрения приложения написанного на «node.js». Для внедрения программного продукта выбран виртуальный сервер на хостинге «ihor». Также для полноценного функционирования был приобретен домен и ssl сертификат. При

						Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19	

внедрении программного продукта были использованы стандартные методы «прт». Таким образом, при переносе файлов нет необходимости переносить все зависимости проекта, достаточно перенести структуру проекта с файлом конфигурации и скачать все необходимые модули, что сэкономит много времени. После внедрения и настройки сервера на виртуальном сервере, проект стал общедоступным и готов для эксплуатации.

					<i>ДП-02069964-09.02.03-13-19</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		46

ЗАКЛЮЧЕНИЕ

В результате проделанной работы создан полноценный проект по проведению многопользовательских он-лайн олимпиад, предназначенных для контроля, проверки и закрепления знаний. Данный проект является общедоступным тренажером для беспристрастной проверки знаний.

В процессе разработки решены следующие задачи:

- составление технического задания;
- составление диаграмм последовательности, компонентов и активности;
- проектирование интерфейса и клиентской части;
- проектирование серверной части;
- создание базы данных;
- реализация защиты данных в базе;
- внедрение.

Система разрабатывалась с целью оптимизировать процесс прохождения олимпиад и сделать его общедоступным. А также популяризировать данное направление среди студентов и не только.

Данный проект располагается на виртуальном сервере и доступен всем по ссылке: <https://elementaria.ru>. А также разработанный проект включает в себя основные элементы полноценного проекта:

- пользовательский интерфейс
- функционал;
- защищенность;
- доступность.

Разработанный проект имеет множество плюсов по отношению к стандартному типу проведения олимпиад, а именно:

- сокращенно время на подготовку и проведение олимпиад;
- минимизированы финансовые и трудовые затраты;

					Лист
					47
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19

- минимизировано количество персонала;
- отсутствует потребность в помещениях;
- моментальный результат;
- возможность модернизации;
- адаптируемость под разные олимпиады;
- сохранен в полном объеме основной принцип проведения олимпиад, а именно, одновременное и централизованное участие.

Система проведения онлайн олимпиад дает возможность пользователю участвовать в олимпиаде без чьей-либо помощи, принимать участие по любым доступным предметам, что позволяет оценивать знания, умения и навыки в различных областях.

					<i>ДП-02069964-09.02.03-13-19</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		48

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Баканов А. С. Проектирование пользовательского интерфейса: эргономический подход [Электронный ресурс] : монография / А. С. Баканов, А. А. Обознов. — Электрон. дан. — Москва : 2009. — 184 с. — Режим доступа: <https://e.lanbook.com/book/108843>. — Загл. с экрана.

2 Заяц А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js [Электронный ресурс] : учебное пособие / А. М. Заяц, Н. П. Васильев. — Электрон. дан. — Санкт-Петербург : Лань, 2019. — 120 с. — Режим доступа: <https://e.lanbook.com/book/115516>. — Загл. с экрана.

3 Александров Д. В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы [Электронный ресурс] : учебное пособие / Д. В. Александров. — Электрон. дан. — Москва : Финансы и статистика, 2011. — 224 с. — Режим доступа: <https://e.lanbook.com/book/5306>. — Загл. с экрана.

4 Введение в защиту информации от внутренних ИТ-угроз [Электронный ресурс] : учебное пособие. — Электрон. дан. — Москва, 2016. — 39 с. — Режим доступа: <https://e.lanbook.com/book/100720>. — Загл. с экрана.

5 Рузманов А. А. Прокин А. А. Использование веб-сокетов в интернет приложениях [Электронный ресурс]: Электронное периодическое издание — Эл № ФС77-66730 — Режим доступа: <http://e-scio.ru/wp-content/uploads/2019/04/Рузманов-А.-А.-Прокин-А.-А..pdf> — Загл. с экрана.

6 Зудилова Т. В. Web-программирование JavaScript [Электронный ресурс]: учеб.-метод. пособие / Т.В. Зудилова, М.Л. Буркова. — Электрон. дан. — Санкт-Петербург: НИУ ИТМО, 2012. — 68 с. — Режим доступа: <https://e.lanbook.com/book/43561>. — Загл. с экрана.

7 Хэррон Д. Node.js. Разработка серверных веб-приложений в JavaScript [Электронный ресурс] / Д. Хэррон ; пер. с англ. Слинкина А. А. — Электрон. дан. — Москва: ДМК Пресс, 2012. — 144 с. — Режим доступа: <https://e.lanbook.com/book/50571>. — Загл. с экрана.

8 Ульман, Л. MySQL [Электронный ресурс] / Л. Ульман. — Электрон. дан. — Москва: ДМК Пресс, 2008. — 352 с. — Режим доступа: <https://e.lanbook.com/book/1241>. — Загл. с экрана.

9 Алексеев А. П. Введение в Web-дизайн [Электронный ресурс] : учебное пособие / А. П. Алексеев. — Электрон. дан. — Москва : СОЛОН-Пресс, 2008. — 192 с. — Режим доступа: <https://e.lanbook.com/book/13768>. — Загл. с экрана.

10 Заяц А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js [Электронный ресурс] : учебное пособие / А. М. Заяц, Н. П. Васильев. — Электрон. дан. — Санкт-Петербург : Лань, 2019. — 120 с. — Режим доступа: <https://e.lanbook.com/book/115516>. — Загл. с экрана.

11 Котляров В. П. Основы тестирования программного обеспечения [Электронный ресурс] : учебное пособие / В.П. Котляров. — Электрон. дан. — Москва, 2016. — 248 с. — Режим доступа: <https://e.lanbook.com/book/100352>. — Загл. с экрана.

											Лист
											50
Изм.	Лист	№ докум.	Подпись	Дата							

ДП-02069964-09.02.03-13-19

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг HTML кода страниц и форм

```
doctype html
- var text = rights;
head
  title= title
  link(rel='shortcut icon', href='../img/favicon.ico', type='image/x-icon')
  link(rel='stylesheet', type='text/css', href='css/css.css')
  script(type='text/javascript', src='js/jquery-3.2.1.min.js')
  script(src='//code.jquery.com/ui/1.11.4/jquery-ui.js')
  meta(charset='utf-8')
.modal_auth
  .name_vh Вход
  a(href='/reg') Регистрация
  .reg_window
    form(action='/login', method='post')
      div
        input.reg_inp(type='text', name='username', placeholder='Логин')
      div
        input.reg_inp(type='password', name='password', placeholder='Пароль')
      div
        input.reg_inp(type='submit', value='Войти')
  a(href='/auth/vkontakte')
    img(src='img/vk.png', style='width: 50px; height: 50px;')
.night
header
  a(href='/')
  .logo Elementaria
  if f_name
    .nav
      a(href='/score')
        .el_nav
          span Рейтинг
      a(href='/')
        .el_nav 0 нас
      a(href='/')
        .el_nav FAQ
```

										Лист
										51
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19					

Продолжение ПРИЛОЖЕНИЯ А

```
    if rights == 1
      a(href='/create')
        .el_nav Создать
    .auth_block
      a(href='/exit/vkontakte')
        .el_nav(style='float: right; width: 50px') Выход
      .user= f_name + " " + l_name
      .user_img
        img.u_img(src= photo)
  else
    .nav(style='width: 69%')
      a(href='/')
        .el_nav 0 нас
      a(href='/')
        .el_nav FAQ
    .auth_block(style='width: 10%')
      #auth_but.el_nav(style='width: 100%; padding-left: 0;') Войти
#wrapper
  case page
    when "home"
      include home.pug
    when "guest"
      include guest.pug
    when "ol"
      include ol.pug
    when "create"
      include create.pug
  footer
    .footer-copi © elementaria.ru 200000000 д.н.э - 2019
    script(type='text/javascript', src='js/js.js')

    </div>
    {{/if}}
    {{#unless username}}
    <div class="nav" style="width: 69%">
      <a href="/"><div class="el_nav">0
нас</div></a>
```

									Лист
									52
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19				

Окончание ПРИЛОЖЕНИЯ А

```
<a href="/"><div
class="el_nav">FAQ</div></a>
<a href="/"><div
class="el_nav">Контакты</div></a>
</div>
<div class="auth_block" style="width: 10%">
  <!--<a href="/auth/vkontakte"><div
class="el_nav">Войти через ВК</div></a-->
  <div id="auth_but" class="el_nav"
style="width: 100%; padding-left: 0;">Войти</div>
  <!--<a href="/login"><div class="el_nav"
style="width: 100%; padding-left: 0;">Войти</div></a-->
  <!--<a href="/reg"><div
class="el_nav">Регистрация</div></a-->
</div>
{{/unless}}

//листинг кода регистрации
<div class="content">
  <div class="reg_window">
    <input class="reg_inp" type="text"
id="email" placeholder="E-mail" >
    <input class="reg_inp" type="text"
id="username" placeholder="Имя">
    <input class="reg_inp" type="text"
id="lastname" placeholder="Фамилия" >
    <input class="reg_inp" type="text"
id="password" placeholder="Пароль" >
    <input class="reg_inp" type="text"
id="password2" placeholder="Повторите пароль" >
    <!--<input type="password" id="password">-
->
    <div id="reg_but"
class="reg_inp">Зарегистрироваться</div>
```

						Лист
					ДП-02069964-09.02.03-13-19	53
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг кода серверной части

```
var tasks = require('./models/tasks');
const fs = require('fs');
const express = require('express');
var app = express();
var server = require('http').createServer(app);
var io = require('socket.io').listen(app.listen(3000, 'localhost'));
/*app.listen(3000, 'localhost')*/
/*server.listen(3000);*/

var color = require('colors');

var templating = require('consolidate');

var urlutils = require('url');
var session = require('express-session');
const passport = require('passport');
const FileStore = require('session-file-store');
var LocalStrategy = require('passport-local').Strategy;

const VKontakteStrategy = require('passport-vkontakte').Strategy;
var bodyParser = require('body-parser');
var cookieParser = require('cookie-parser');
app.use(bodyParser.urlencoded({extended: true}));
app.use(bodyParser.json());
app.use(cookieParser());
app.use(session({secret: 'qwertyui', cookie: {maxAge: 10000 * 8640}, resave: true,
saveUninitialized: true}));

app.use(passport.initialize());
app.use(passport.session()); // persistent login sessions
//шаблонизатор&роутинг
app.engine('html', templating.handlebars);
app.set('view engine', 'html');
app.set('views', __dirname + '/views');
app.use(express.static(__dirname + '/public'));
app.use('/room/', express.static(__dirname + '/public'));
passport.use(new LocalStrategy(
  function(username, password, done) {
    tasks.list(function(err, req){
      var secr = username + password;
      crypto.pbkdf2(secr, 'salt', *, *, *, (err, key)
=> {
        if (err) throw err;
        var key = key.toString(*); //
        var sumpas = * + *;
        var h = crypto.createHmac(*, sumpas)
          .update('*')
          .digest(*);
        console.log(h)
        for(i in req){
```

									Лист
									54
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19				

Продолжение ПРИЛОЖЕНИЯ Б

```

    if(req[i].login == username
        //console.log(req[i])
        return done(null, {
            username: req[i].name
        });
    }
}
    })
});

app.get('/', function(req,res) {
    //console.log(session.Cookie())
    //req.session.touch()

    res.render('home',function(err, html) {
        if(err)
            throw res.redirect('/guest');
        res.render('index',{
            username: req.user.username,
            photo: req.user.photoUrl,
            score: tasks,
            content: html,
            title : title
        })
        res.cookie('username', req.user.username, {
            maxAge: 900000, httpOnly: true });
    })
    //res.cookie('username', req.user.username, {
    maxAge: 900000, httpOnly: true });
})
io.on('connection', function(client) {
    console.log("+")

    console.log(client.request)
    client.on('disconnect', function(data) {
        client.leave(client.roomID);
        console.log("-")
        users.splice(users.indexOf(client.user), 1)
        //users = [];
    });

    client.on('login', function(data) {
        client.roomID = data.roomID;
        client.user = data.user;
        client.count = count;

        client.join(data.roomID);
        //console.log(client.user)
        //console.log(users[0])
        users.push({name: data.user})
        console.log(users)
        //count['user'] = user;
    });
});

```

									Лист
									55
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19				

Окончание ПРИЛОЖЕНИЯ Б

```
//client.emit('output',{ "data": "Привет мир"})
})
client.on('messege', function(data) {
  var a = setInterval(send, timer);
  var c = 0;

  function send() {

    tasks.question(function(err, task) {
      if(c == task.length){

        clearInterval(a)
        setTimeout(end_game,
timer+1000)

        }
        function end_game() {
          count = String(client.count);
          console.log(count)

          client.broadcast.to(client.roomID).emit('end_game
', {"count": count});

          }

          client.broadcast.to(client.roomID).emit('output',
task[c-1]);

          })
          c++;

          }

          })
          client.on('trues', function(data) {
            tasks.question(function(err, task) {
              //console.log(task)

              if(task[data.vid].trues == data.resid){
                client.count +=10;
                console.log(task[data.vid].trues + "
= " + data.resid + " | " + client.count)
              }else{
                console.log(task[data.vid].trues + "
!= " + data.resid + " | " + client.count)
              }

              //client.broadcast.to(client.roomID).emit('end_ga
me');
```

Изм.	Лист	№ докум.	Подпись	Дата

ДП-02069964-09.02.03-13-19

Лист

56

ПРИЛОЖЕНИЕ В

(обязательное)

Модуль шифрования данных

```
passport.use(new LocalStrategy(
  function(username, password, done) {
    tasks.list(function(err, req){
      console.log(req)
      var secr = username + password;
      crypto.pbkdf2(secr, 'salt', 5, 256, 'sha256',
(err, key) => {
          if (err) throw err;
          var key = key.toString('hex'); //
'c5e478d...1469e50'
          var sumpas = password + key;
          var h = crypto.createHmac('sha256',
sumpas)
              .update('I love cupcakes')
              .digest('hex');
          console.log(h)
          for(i in req){
              //console.log(i)
              if(req[i].login == username
&& req[i].password == h){
                  return done(null, {
                      username: req[i].name
                  });
              }
          }
      })
    });
  });
```

Изм.	Лист	№ докум.	Подпись	Дата

ДП-02069964-09.02.03-13-19

Лист

57

Окончание ПРИЛОЖЕНИЯ В

```
app.post("/reg", function (req, res) {
    //console.log(req.body)
    var secr = req.body.email + req.body.password;
    //console.log(secr)
    crypto.pbkdf2(secr, 'salt', 5, 256, 'sha256',
(err, key) => {
        if (err) throw err;
        var key = key.toString('hex'); //
'c5e478d...1469e50'
        var sumpas = req.body.password + key;
        var h = crypto.createHmac('sha256', sumpas)
            .update('I love cupcakes')
            .digest('hex');
        console.log(h)
        var masadd = {
            login: req.body.email,
            name: req.body.username,
            l_name: req.body.lastname,
            password: h
        }
        tasks.addlocal(masadd);
    });
    res.redirect('/')
    //res.send(a+ ' '+b);
});
```

ПРИЛОЖЕНИЕ Г

(обязательное)

Чек-лист страницы elementaria.ru

Таблица Г.1 — Чек-лист elementaria.ru

elementaria.ru		
Название	Статус	Приложения
Наличие логотипа "elementaria" слева в углу при нажатии на который осуществляется переход по ссылке указанной в приложении	Pass	https://elementaria.ru/
Ссылка "О нас" в навигации ведущая на страницу указанную в приложении	Pass	https://elementaria.ru/info
Ссылка "FAQ" в навигации ведущая на страницу указанную в приложении	Pass	https://elementaria.ru/faq
Ссылка "Контакты" в навигации ведущая на страницу указанную в приложении	Pass	https://elementaria.ru/contacts
Кнопка "Вход" в правом верхнем углу при нажатии на которую вызывается выпадающая форма входа	Pass	

Изм.	Лист	№ докум.	Подпись	Дата

ДП-02069964-09.02.03-13-19

Лист

59

Окончание ПРИЛОЖЕНИЯ Г

Окончание таблицы Г.1

Модальное окно		
Заголовок "Вход"	Pass	
Форма ввода логина	Pass	
Форма ввода пароля	Pass	
Кнопка "Зарегистрироваться" в правом верхнем углу	Pass	
Кнопка "Войти"	Pass	
Иконка входа через социальную сеть "Вконтакте"	Pass	

ПРИЛОЖЕНИЕ Д
(обязательное)
Техническое задание

Термины и определения

1 общие сведения;

1.1. Наименование системы

Полное наименование: система проведения on-line олимпиад.

1.2. Наименование организаций — Заказчика и Разработчика

1.2.1. Заказчик

Заказчик: ФГБОУ ВО «МГУ им. Н.П. Огарева» ФДП и СПО

1.2.2. Разработчик

Разработчик: Рузманов Андрей Александрович

2 назначение и цели создания (развития) системы;

2.1. Назначение системы

«Система проведения on-line олимпиад» предназначена для удаленного проведения олимпиад среди школьников и студентов. Основным назначением «системы проведения on-line олимпиад» является автоматизация процесса проведения олимпиад.

2.2. Цели создания системы

Система проведения on-line олимпиад создается для:

Оптимизации проведения олимпиад

Возможностью участвовать в олимпиаде всем (даже людям с ограниченными возможностями)

Уменьшение трудозатрат и время затрат на проведение олимпиад

3 характеристика объектов автоматизации;

Предметные олимпиады

Олимпиады проф мастерства

					<i>ДП-02069964-09.02.03-13-19</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		61

Продолжение ПРИЛОЖЕНИЯ Д

4 требования к системе;

4.1. Требования к системе в целом

Система проведения онлайн олимпиад должна быть централизована и располагаться на виртуальном сервере.

4.1.1. Требования к структуре и функционированию системы

Система проведения on-line олимпиад должна представлять собой веб-сайт.

Для написания проекта требуется использовать стек технологий HTML, CSS, JS MYSQL.

Сервер приложения должен быть написан на языке программирования JavaScript с использованием серверной платформы Node.js.

4.1.2 Требования к дизайну сайта

Цветовая схема сайта должна быть выполнена в сине-серых цветах

Элементы сайта должны быть расположены в быстро доступных местах для удобства пользования

Все страницы сайта должны быть представлено в одном дизайне.

Стилистической особенностью проекта является закругленные края блоков с двух противоположенных углов

4.1.3 Список страниц и их содержание

Хедер (Для неавторизованного пользователя содержит логотип, кнопки «О нас», «FAQ», «Контакты» и с правой стороны кнопку «Войти». Для авторизованных пользователей содержит логотип, кнопки «О нас», «FAQ», «Контакты» «Предметные области» «Рейтинг» с правой стороны блок с информацией о клиенте содержащий аватар пользователя, имя и фамилию пользователя, и кнопку «Выход»)

Футер (Содержит копирайт, контактные данные, обратная связь)

										Лист
										62
Изм.	Лист	№ докум.	Подпись	Дата						

ДП-02069964-09.02.03-13-19

Продолжение ПРИЛОЖЕНИЯ Д

Главная страница (Для неавторизованного пользователя содержит хедер с кнопками «О нас», «FAQ», «Контакты» и с правой стороны кнопку «Войти»)

О нас (Содержит информацию о проекте включая хедер и футер)

FAQ (Список популярных вопросов включая хедер и футер)

Контакты (Содержит все контакты для связи)

Предметные области (Содержит список предметных областей по которым можно пройти олимпиаду (Русский язык, Математика, Физика, химия...)).

Рейтинг (содержит таблицы лучших пользователей которые классифицируются по предметным областям).

Страницу участия в олимпиаде(содержит блок с появляющимися вопросами)

Страница результата олимпиады (содержит количество баллов)

4.1.4 Структура серверной части

Основной модуль для рендеринга страниц

Модуль запросов к базе данных

Отдельные модули (по обстоятельствам)

4.1.5 Реализация защиты проекта и базы данных

Реализовать защиту от взлома на всех уровнях безопасности, на стороне сервера, стороне базы данных.

Реализовать защиту данных на стороне сервера, стороне приложения, и базы данных.

Реализовать шифрование передаваемых данных с помощью ssl сертификата

5 состав и содержание работ по созданию системы;

Разработка дизайна

Проектирование и создание серверной части

										Лист
										63
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19					

Продолжение ПРИЛОЖЕНИЯ Д

Проектирование и создание базы данных

6 порядок контроля и приемки системы;

Приемка работ на соответствие их объема и качества требованиям, установленным в контракте, осуществляется Заказчиком по окончании выполнения работ по контракту.

Виды, состав, объём и методы испытаний систем должны соответствовать требованиям ГОСТ 34.603-92 «Виды испытаний автоматизированных систем».

При проведении всех видов испытаний систем должен использоваться документ Программа и методика испытаний.

Приемка работ осуществляется приемочной комиссией. Состав приемочной комиссии, место и время проведения испытаний определяется Заказчиком. В состав комиссии по проведению испытаний должны входить представители Заказчика и Подрядчика. Заказчик может привлечь для приемки работ сторонние организации.

Комиссии по приемке предъявляются результаты всех выполненных работ. Виды, состав, объем, и методы испытаний систем должны соответствовать требованиям ГОСТ 34.603-92 и РД 50.34.698-90. В рамках проводимой экспертизы документов комиссией оцениваются:

- полнота;
- качество;
- непротиворечивость.

В случае выявления недостатков и замечаний к документации, Заказчик вправе потребовать от Подрядчика их полного устранения в согласованные сторонами сроки. Сторонами оформляется протокол замечаний, где описываются все обнаруженные недостатки, замечания к документации, устанавливаются сроки их устранения. Подрядчик должен представить

					ДП-02069964-09.02.03-13-19	Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ Д

обновленные отчетные материалы и документы в установленный протоколом срок.

Подрядчик вправе передавать Заказчику проекты документов для их изучения и корректировки в процессе разработки. Работы считаются выполненными в случае устранения всех замечаний, утверждения проектной документации Заказчиком и подписания сторонами Акта сдачи-приемки всех выполненных работ по контракту.

Сроки проведения испытаний определяются Разделом 7 настоящих Технических требований.

После завершения выполнения работ, предусмотренных контрактом, Подрядчик в течение 5 (пяти) рабочих дней, письменно уведомляет Заказчика о факте завершения выполнения работ и направляет в адрес Заказчика акт сдачи-приемки выполненных работ в 2 (двух) экземплярах, счет, счет фактуру.

Не позднее 5 (пяти) рабочих дней после получения от Подрядчика документов, указанных в п. 4.2. контракта, Заказчик рассматривает результаты и осуществляет приемку выполненных работ по контракту на предмет соответствия их объема и качества требованиям, изложенным в контракте.

Заказчик в течение 5 (пяти) рабочих дней со дня получения от Подрядчика акта сдачи-приемки выполненных работ рассматривает указанный акт и направляет Подрядчику один экземпляр подписанного акта приемки выполненных работ либо мотивированный отказ от приемки выполненных работ, в котором должны быть указаны выявленные Заказчиком недостатки. Заказчик вправе предоставить Подрядчику срок для устранения таких недостатков. Мотивированный отказ направляется в порядке, предусмотренном п. 11.1. контракта.

В сроки, указанные Заказчиком в мотивированном отказе от приемки выполненных работ, Подрядчик обязан за свой счет и своими силами, устранить

									Лист
									65
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19				

Продолжение ПРИЛОЖЕНИЯ Д

обнаруженные недостатки. В этом случае акт выполненных работ Заказчик подписывает в течение 10 (десяти) рабочих дней после устранения Подрядчиком указанных недостатков.

Если Подрядчик в установленный срок не устранит недостатки, Заказчик вправе отказаться от исполнения контракта и предъявить Подрядчику требование о возмещении понесенных убытков.

После устранения выявленных несоответствий или недостатков, работы принимаются повторно в порядке, установленном настоящим разделом.

Претензии по скрытым дефектам могут быть заявлены Заказчиком в течение всего гарантийного срока.

7 требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;

7.1 Проведение предварительных испытаний

На этапе проведения предварительных испытаний приказом создается комиссия для проведения предварительных испытаний.

Предварительные испытания проводятся в соответствии с документом Программа и методика испытаний, которая разрабатывается Подрядчиком и согласовывается с Заказчиком. Общий срок согласования документации не должен превышать 15 дней.

Испытания проводятся в течение 5 дней.

По результатам проведения испытаний Заказчиком совместно с Подрядчиком составляется протокол проведения испытаний. Общий срок согласования протокола не должен превышать 15 календарных дней.

Критичные замечания и ошибки, выявленные в ходе предварительных испытаний, устраняются в течение 15 дней с даты завершения приемочных испытаний.

											Лист
											66
Изм.	Лист	№ докум.	Подпись	Дата	ДП-02069964-09.02.03-13-19						

Продолжение ПРИЛОЖЕНИЯ Д

После устранения критических замечаний, работы считаются выполненными, Подрядчиком направляется Заказчику протокол проведения испытаний, акт выполненных работ.

7.2 Опытная эксплуатация

В ходе проведения опытной эксплуатации ведется рабочий журнал учета, продолжительности испытаний, ошибок и результатов их устранения.

В рамках данного этапа необходимо добиться устойчивого функционирования программного обеспечения, удовлетворяющего всем требованиям Технических требований.

7.3 Проведение приемо-сдаточных испытаний

В ходе проведения приемо-сдаточных испытаний комиссия согласно Программе и методике испытаний принимает решение о готовности программного обеспечения к вводу в промышленную эксплуатацию. Для этого должны быть исправлены все неисправности и замечания, выявленные на предыдущих этапах работ.

По итогам приемо-сдаточных испытаний формируется протокол приемо-сдаточных испытаний.

7.4 Ввод в промышленную эксплуатацию

Результатом успешного выполнения всех предшествующих этапов работ является подписание Технического акта (протокола) ввода информационных систем в соответствии с настоящими Техническими требованиями в промышленную эксплуатацию и Акта сдачи-приемки всех выполненных работ по контракту.

8 требования к документированию

Техническая и эксплуатационная документацию должна быть разработана с учетом требований комплекса государственных стандартов «Информационная технология».

					<i>ДП-02069964-09.02.03-13-19</i>	<i>Лист</i>
						67
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Окончание ПРИЛОЖЕНИЯ Д

Комплекс стандартов на автоматизированные системы»:

- ГОСТ 34.601-90 «Автоматизированные системы. Стадии создания»;
- ГОСТ 34.003-90 «Автоматизированные системы. Термины и определения»;
- ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»;
- ГОСТ 34.201-89 «Виды, комплектность и обозначение документов при создании автоматизированных систем»;
- ГОСТ 34.603-92 «Виды испытаний автоматизированных систем»;
- ГОСТ 19.301-79 «Программа и методика испытаний. Требования к содержанию и оформлению»;
- РД 50-34.698-90 «Автоматизированные системы. Требования к содержанию документов».

Возможно уточнение состава и объединение разрабатываемых документов по согласованию с Заказчиком.

					<i>ДП-02069964-09.02.03-13-19</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		68