

СОДЕРЖАНИЕ

| | |
|--|----|
| Введение..... | 5 |
| 1 Обоснование выбора методов реализации проекта..... | 7 |
| 1.1 Выбор методологии проектирования..... | 7 |
| 1.2 Выбор языка и среды разработки..... | 8 |
| 1.3 Выбор СУБД..... | 12 |
| 1.4 Выбор методов реализации пользовательского интерфейса..... | 13 |
| 2 Практическая реализация проекта | 15 |
| 2.1 Проектирование WEB-приложения | 15 |
| 2.1.1 Предмет разработки..... | 15 |
| 2.1.2 Требования к графическому дизайну сайта | 15 |
| 2.1.3 Функциональные требования | 16 |
| 2.1.4 Требования к структуре сайта | 20 |
| 2.1.5 Требования к системе управления сайтом | 20 |
| 2.1.5.1 Требования к административной части | 20 |
| 2.1.5.2 Требования к управления разделами сайта..... | 21 |
| 2.1.5.3 Управление наполнением сайта | 22 |
| 2.1.5.4 Требования к разделению доступа | 23 |
| 2.1.6 Требования к видам обеспечения..... | 23 |
| 2.1.6.1 Требования к информационному обеспечению..... | 23 |
| 2.1.6.2 Требования к языкам программирования..... | 26 |
| 2.1.6.3 Требования к организации гиперссылок | 26 |
| 2.1.7. Требования к программному обеспечению..... | 27 |
| 2.1.7.1 Требования к программному обеспечению серверной части..... | 27 |

| | | | | | | | | |
|-------------|-------------|-----------------|----------------|-------------|--|--------------|-------------|---------------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | | | |
| <i>Изм.</i> | <i>Лист</i> | <i>№ докум.</i> | <i>Подпись</i> | <i>Дата</i> | Разработка WEB-приложения тестирования по английскому языку для ГБПОУ РО «РКРИПТ». Пояснительная записка | <i>Лит.</i> | <i>Лист</i> | <i>Листов</i> |
| Разраб. | | Калантарян А.А | | | | | | |
| Провер. | | Галкина Н.Г. | | | | | 2 | 60 |
| Реценз. | | Петрова М.Е. | | | | ПО-42 | | |
| Н. Контр. | | Горбачук М.А. | | | | | | |
| Утверд. | | Будасова С.А. | | | | | | |

| | |
|---|----|
| 2.1.7.2 Требования к клиентскому программному обеспечению | 27 |
| 2.1.8 Требования к техническому обеспечению | 27 |
| 2.1.9 Требования к персоналу | 28 |
| 2.2 Реализация программного продукта | 28 |
| 2.2.1 Среда разработки | 28 |
| 2.2.2 Языки программирования | 30 |
| 2.2.3 Методы, использованные при разработке WEB-приложения..... | 31 |
| 2.2.4 Методы отладки | 35 |
| 2.3 Тестирование WEB-приложения..... | 37 |
| 2.4 Защита данных и WEB-приложения..... | 39 |
| 2.5 Описание порядка работы с программным продуктом | 41 |
| 2.5.1 Общие сведения | 41 |
| 2.5.2 Порядок запуска WEB-приложения..... | 42 |
| 2.5.3 Работа с WEB-приложением..... | 44 |
| 2.5.4 Порядок запуска административной части WEB-приложения..... | 51 |
| 2.6 Охрана труда пользователей программного продукта..... | 59 |
| 2.6.1 Общие положения | 59 |
| 2.6.2 Меры безопасности для предотвращения вредного или опасного фактора на рабочем месте | 59 |
| 2.6.3 Способы защиты от опасных и вредных факторов | 61 |
| 2.6.4 Выводы по разделу «Охрана труда» | 62 |
| 3 Экономическая часть | 63 |
| 3.1 Определение трудоемкости разработки проекта | 63 |
| 3.2 Определение величины затрат на создание | 64 |
| 3.3 Определение себестоимости разработанного проекта..... | 68 |
| Заключение | 69 |
| Список использованных источников | 71 |
| Приложение А (Обязательное) Диаграмма вариантов истользования.. | 74 |
| Приложение Б (Обязательное) Диаграмма взаимоействия | 75 |

| | |
|--|----|
| Приложение В (Обязательное) ER-диаграмма..... | 76 |
| Приложение Г (Обязательное) Диаграмма компонентов | 77 |
| Приложение Д (Обязательное) Схема пользовательского интерфейса | 78 |
| Приложение Е (Обязательное) Диаграмма развертывания | 79 |

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 4 |

ВВЕДЕНИЕ

Для написания дипломного проекта была выбрана тема: «Разработка WEB-приложения тестирования по английскому языку для ГБПОУ РО «РКРИПТ».

Выбранная тема является актуальной так, как проблема контроля за уровнем знаний, умений и навыков обучающихся – вечная проблема образования, и в разные времена предлагалось решать её по-разному. Преобладали то одни формы контроля, то другие, создавались контролирующие системы.

Тест может служить методом педагогического измерения на любом этапе обучения. Тестовые задания эффективны и для самопроверки знаний. Использование тестов позволяет сэкономить массу времени, проверить широкий спектр знаний учащихся по предмету, уровень сформированности некоторых навыков и умений.

WEB-приложения имеют ряд преимуществ перед традиционными программами, такие как удобный доступ через интернет:

- WEB-приложения не нужно устанавливать на каждом устройстве, на котором они будут использоваться. Это означает, что пользователи могут получить доступ к приложению из любого места, где есть доступ к интернету, используя любое устройство, включая компьютеры, планшеты и смартфоны;

- WEB-приложения могут быть лучше защищены от вирусов и злонамеренного программного обеспечения, так как пользователи не загружают приложение на свои устройства.

WEB-приложение для тестирования по английскому языку предполагает:

- объективность оценок;
- возможность тиражирования тестов;
- возможность самоконтроля;
- обратная связь с обучающимся;
- возможность индивидуального выбора времени и места тестирования.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 5 |

Обучающиеся могут проходить многократно короткие тесты, проводимые в течение всего срока обучения. Данные, полученные в результате этих оценок, могут быть соотнесены с необходимыми стандартами, для того чтобы обучающиеся могли быть измерены по этим стандартам.

Для реализации поставленной цели необходимо выполнить следующие шаги:

- проанализировать предметную область;
- разработать схему пользовательского интерфейса;
- разработать структуру базы данных (БД);
- спроектировать пользовательский интерфейс;
- спроектировать базу данных;
- реализовать WEB-приложение;
- заполнить приложение тестовыми данными;
- отладить и протестировать WEB-приложение;
- составить программную документацию.

Результатом выполнения работы будет WEB-приложение тестирование по английскому языку, позволяющий студентам ГБПОУ РО «РКРИПТ» проходить тесты по английскому языку, а преподавателям - отслеживать результаты прохождения тестов студентами.

Программный продукт может быть использован в ГБПОУ РО «РКРИПТ» или же другими образовательными учреждениями.

Пояснительная записка содержит: 80 страницы, 42 рисунков, 9 таблиц, 6 приложений.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 6 |

1 Обоснование выбора методов реализации проекта

1.1 Выбор методологии проектирования

Для выполнения дипломного проекта будут рассмотрены две методологии проектирования: Agile и Waterfall. Обе эти методологии являются оптимальным выбором для достижения поставленных целей в ходе разработки WEB-приложения [11].

Waterfall – это четко запланированный и детализированный подход, где исполнитель придерживается плану. Agile – прямая противоположность, которая предполагает гибкость разработки с возможностью внесения изменений на каждом этапе проекта.

Agile – система, основанная на принципе «гибкого» управления проектами. Сюда относят методики Scrum, FDD, Kanban, Экстремальное программирование (XP), Lean и т.д. Ключевая особенность такого подхода - создание проекта в несколько циклов (итераций), в конце каждого виден конкретный результат, который позволяет понять, по какому пути двигаться дальше [13].

Главные принципы Agile:

- внести необходимые изменения можно в любом из циклов разработки;
- лучший способ получения обратной связи с заказчиком и коллегами – личное общение;
- создаваемый продукт обновляется в конце каждого цикла или один раз в несколько месяцев;
- готовность к изменениям в процессе разработки важнее, чем беспрекословное следование изначальному плану;

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 7 |

– Waterfall (с англ. – «водопад») – предполагает последовательный переход к каждому этапу разработки и невозможностью вернуться на шаг назад. Внести какие-либо изменения будет возможно только после релиза проекта [14].

Выделяют следующие стадии разработки в Waterfall:

- анализ системных и программных требований, которые закреплены в документе (Word или PDF);
- планирование всех этапов разработки. Важный пункт, так как вся последующая работа будет четко следовать составленному плану;
- проектирование. Разрабатывается внутренняя архитектура проекта, его внешний вид, структура, рассматриваются варианты реализации;
- реализация дизайна, верстки, программного продукта;
- интеграция. Проводятся необходимые работы по обмену данными и пишется код программы;
- тестирование. Готовый продукт проверяется на наличие программных ошибок, также выявляются недочеты функционала. После этого идет исправление нужных багов;
- выпуск продукта. Релиз готового проекта и окончание разработки. Возможна также работа по адаптации проекта к иным видам систем [9].

Для достижения успешного результата в разработке WEB-приложения была выбрана Agile, так как необходимые изменения можно внести в любом из циклов разработки.

1.2 Выбор языка и среды разработки

Перед началом создания любого сайта необходимо выбрать инструменты разработки. И чтобы не писать сайт с нуля, что может занять очень много времени,

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 8 |

разработчики часто используют CMS. Это помогает сэкономить время и упростить процесс.

CMS («Content Management System» – перев. система управления контентом) – инструмент для быстрой разработки, с готовым функционалом. Но он ставит определенные рамки, выйти за которые невозможно или возможно лишь потратив кучу времени, и таким странным способом, что можно навредить проекту. Это могут быть уязвимости, излишний расход ресурсов и т.д [15].

Ручное сайтостроение сопряжено с многими трудностями. Во-первых, они заключаются в необходимости изучения языка разметки и программирования. Это достаточно просто выучить и использовать, хотя все структуры ресурса нужно продумывать изначально.

Посредством кода HTML есть возможность создать тот дизайн, который изначально задумывался. Не всегда это можно выполнить при помощи CMS, для которых дизайн не всегда просто реализовать. Готовые дизайны либо являются дефектными, либо не полностью устраивают веб-мастера. По этой причине разработка HTML сайта выглядит намного более привлекательной, особенно для новичков сферы веб-разработок.

Высокая практичность HTML обеспечивается тем, что на сайте можно самостоятельно продумать необходимый функционал, если подключить к разработке свои знания PHP. CMS имеет спектр возможностей, которые оптимально подходят для новичков, однако пользоваться ими без знания разметки HTML практически невозможно. При этом важны также и мета-теги, за счет которых поисковые машины будут получать информацию о ресурсе. На HTML из написать сложнее, нежели заполнить специальные формы в CMS.

Для дипломного проекта будут выбраны три языка программирования: JavaScript, PHP и SQL. Также язык разметки HTML и язык стилей CSS.

Основной язык разметки, который будет использоваться в веб приложении, это HTML (HyperText Markup Language). HTML позволяет структурировать содержимое веб-страницы и определить ее элементы, такие как заголовки,

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| | | | | | | 9 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

параграфы, ссылки, изображения и многое другое. В дипломном проекте HTML будет использоваться для создания пользовательского интерфейса приложения, размещения текстовых блоков, кнопок и форм.

Для стилизации и внешнего оформления веб-сайта будет использоваться CSS (Cascading Style Sheets). CSS определяет внешний вид элементов HTML, таких как цвет фона, шрифт, размеры, положение и другие атрибуты. В дипломном проекте CSS будет использован, чтобы придать приложению согласованный и привлекательный вид. CSS также позволяет создавать адаптивный дизайн, чтобы приложение корректно отображалось на разных устройствах и экранах.

Также будет использоваться инспектор браузера, такой как Chrome DevTools, для отладки и тестирования стилей CSS на живой веб-странице.

В целом, HTML и CSS будут важными средствами разработки WEB-приложения. HTML определяет структуру и содержимое страницы, а CSS обеспечивает ее стилизацию и внешний вид.

JavaScript будет использоваться для асинхронных запросов с помощью AJAX. Весь остальной функционал, такой как: аутентификация через сессии, проверка на регистрацию, создание административной панели для администратора и многое другое, будут реализованы на PHP.

Один из основных факторов, по которым был выбран PHP, была его простота в использовании. Большинство сайтов сделаны на PHP, PHP предоставляет мощные инструменты для создания веб-сайтов и веб-приложений. Кроме того, PHP легко работает с базами данных, такими как MySQL, которая будет использована для хранения данных.

SQL будет выбран потому, что он является стандартным языком запросов к базам данных. Он будет использован для создания таблиц и запросов к ним. Это позволит хранить данные в базе данных и быстро извлекать их при необходимости.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| | | | | | | 10 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

В процессе разработки будут использоваться различные инструменты и редакторы кода, чтобы облегчить работу с HTML, CSS, JS и PHP. Одним из таких инструментов будет Visual Studio Code, который предоставляет подсветку синтаксиса, автодополнение и другие полезные функции для удобной разработки кода.

Visual Studio Code. Это легкий и удобный в использовании текстовый редактор, который обеспечит быструю и продуктивную разработку кода. Visual Studio Code также обеспечивает широкий спектр расширений и плагинов, которые помогают автоматизировать задачи и ускорять процесс разработки.

Кроме выбора языков программирования, будет выбран OpenServer в качестве среды для разработки и тестирования веб-сайта.

OpenServer — это бесплатная и удобная среда для локальной разработки веб-сайта. Он предоставляет полный стек технологий для веб-разработки, включая в себя сервер Apache, языки программирования PHP и JavaScript, а также базы данных MySQL и phpMyAdmin для управления базами данных. Это позволит быстро настроить локальный сервер и начать разработку приложения без необходимости покупать и настраивать сторонние ресурсы. Кроме того, OpenServer имеет удобный интерфейс, что сделает работу с приложением еще более простой и удобной.

Выбранные языки и среда разработки предоставляют все необходимые инструменты для создания WEB-приложения и ими будет комфортно пользоваться. Кроме того, они имеют широкое сообщество разработчиков и обширную документацию, что позволит быстро находить решения для решения возникающих проблем.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 11 |

1.3 Выбор СУБД

СУБД (система управления базами данных) — это программное обеспечение, которое используется для создания и управления базами данных (БД). СУБД позволяет хранить, организовывать, обрабатывать и извлекать данные, а также обеспечивает безопасность и целостность информации [16].

Существует множество различных СУБД, каждая из которых имеет свои особенности и предназначена для разных целей. Некоторые из других популярных СУБД, которые рассматривались до выбора MySQL, могут включать:

- Oracle Database: мощная и распространенная коммерческая СУБД, используемая в широком спектре предприятий.
- Microsoft SQL Server: коммерческая СУБД, разработанная для работы с платформой Microsoft Windows.
- PostgreSQL: бесплатная и открытая СУБД с акцентом на расширяемость, надежность и соответствие стандартам.
- MongoDB: открытая и гибкая NoSQL СУБД, предназначенная для хранения и обработки структурированных, полуструктурированных и неструктурированных данных.
- SQLite: встроенная СУБД, которая является частью многих операционных систем и программных продуктов, известная своей простотой и компактностью.

В конечном итоге, для разработки WEB-приложения было принято решение использовать MySQL. Этот выбор обусловлен широким использованием MySQL в IT-индустрии, наличием хорошей поддержки и документации. MySQL также известна своей высокой производительностью и способностью обрабатывать большое количество пользователей и баз данных, что делает ее подходящим выбором для хранения и управления данными приложения.

MySQL — это бесплатная и открытая СУБД, которая имеет высокую производительность и поддерживает большое количество пользователей и баз данных. Благодаря этим преимуществам, она будет выбрана для хранения и управления данными приложения.

1.4 Выбор методов реализации пользовательского интерфейса

При выборе методов реализации пользовательского интерфейса для дипломного проекта, будут приняты ряд решений, которые включают в себя использование CSS, CSS-фреймворк Bootstrap (версии 5.3.0), шрифты с сервиса Google Fonts и пиктограммы из набора Kit fontawesome, а также создание макета с использованием инструмента Figma.

CSS будет выбран для стилизации и внешнего оформления веб-сайта. Он будет использоваться для определения цветовой схемы, шрифтов, размеров элементов, расположения и других атрибутов, чтобы создать согласованный и привлекательный интерфейс.

Для ускорения разработки UI будет использоваться CSS-фреймворк Bootstrap. Bootstrap предоставляет готовые стили и компоненты, которые можно легко интегрировать в приложение. В нём есть классы для быстрого создания сетки, кнопок, форм, навигации и других элементов интерфейса.

Чтобы добавить разнообразие и привлекательность в текстовые элементы интерфейса, будет использоваться сервис Google Fonts. Google Fonts предоставляет широкий выбор бесплатных шрифтов, которые можно легко интегрировать в веб-сайте.

Для добавления иконок в интерфейс будет использоваться Kit fontawesome. Kit fontawesome предоставляет библиотеку иконок, которые можно использовать в приложении. Он имеет все необходимые иконки, которые добавляются в HTML-

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 13 |

разметку, а также для них стили, чтобы улучшить визуальное представление элементов интерфейса.

Для создания макета веб приложения и дизайна интерфейса будет использоваться инструмент Figma.

Figma позволяет создавать векторные графики, макеты, прототипы веб-сайта. В дипломном проекте он будет использоваться для разработки визуальной концепции, размещения элементов интерфейса на странице и создания прототипа пользовательского опыта.

Все эти выбранные методы и инструменты совместно помогут создать эстетически привлекательный и функциональный пользовательский интерфейс для WEB-приложения тестирования по английскому языку.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 14 |

2 Практическая реализация проекта

2.1 Проектирование WEB-приложения

2.1.1 Предмет разработки

Предметом разработки является WEB-приложение тестирования по английскому языку для ГБПОУ РО «РКРИПТ», с системой динамического управления наполнением на базе WEB-интерфейса.

Назначение WEB-приложения:

- предоставление возможности прохождения тестирования студентам по английскому языку;
- предоставление студентам информации о пройденных тестах по английскому языку;
- обеспечение студентам возможности выгрузки своих результатов тестирования;
- обеспечение преподавателям возможности просмотра результатов тестирования студентов и выгрузки их.

Цель создания WEB-приложения: предоставление проведение тестирования по английскому языку для студентов ГБПОУ РО «РКРИПТ».

2.1.2 Требования к графическому дизайну сайта

При разработке WEB-приложения должны быть использованы преимущественно светлые стили.

Основные разделы сайта должны быть доступны с первой страницы.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 15 |

В дизайне сайта не должны присутствовать:

- мелькающие баннеры;
- много сливающегося текста;
- отсутствие логотипа;
- неудобное и нелогичное расположение элементов;
- избыточная анимация;
- проблемы с формами регистрации.

2.1.3 Функциональные требования

Главная страница для студента (рисунок 1) должна содержать графическую часть, навигационное меню сайта, поиск, а также контентную область для того, чтобы пользователь с первой страницы мог получить вводную информацию о тестах, а также ознакомиться с последними новыми тестами.

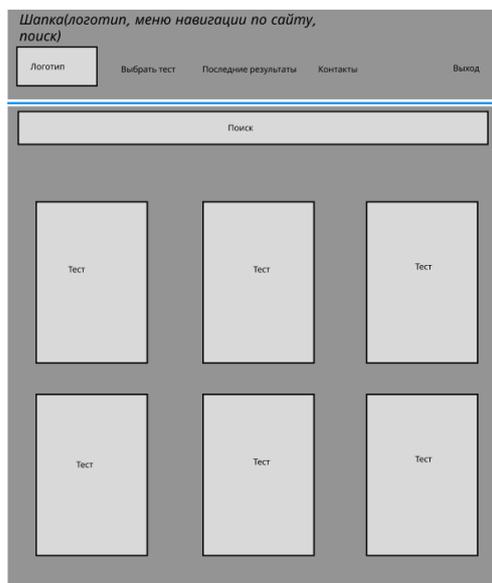


Рисунок 1 – Пример размещения элементов главной страницы

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

Контентная область первой страницы должна делиться на следующие разделы:

- вверху страницы отображаются облегченная навигационная панель, которая обеспечивает переход к основным пунктам меню сайта (выбрать тест, последние результаты, контакты, выйти из аккаунта.);
- поле поиска – предназначено для выполнения полнотекстового поиска по сайту;
- вступительные карточки о последних тестах со ссылкой, ведущей на сам тест;
- краткая контактная информация - техподдержка, почта и вк.

Графическая оболочка внутренних страниц (рисунок 2-5) должна делиться на следующие разделы:

- шапка;
- поле для отображения контента выбранной страницы сайта;
- внизу страницы - краткая контактная информация (телефон, e-mail колледжа, телеграмм и вк).

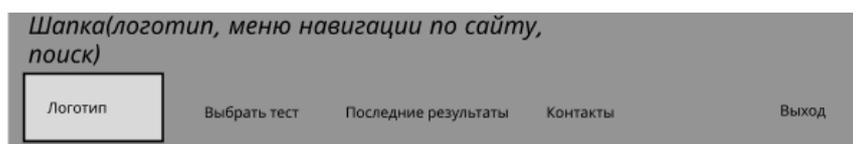


Рисунок 2 – Пример размещения внутренних элементов страниц (шапка)

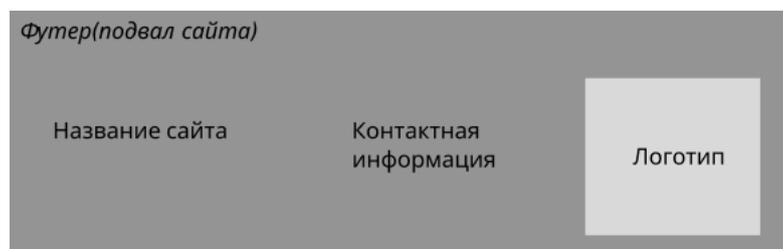


Рисунок 3 – Пример размещения внутренних элементов страниц (подвал)

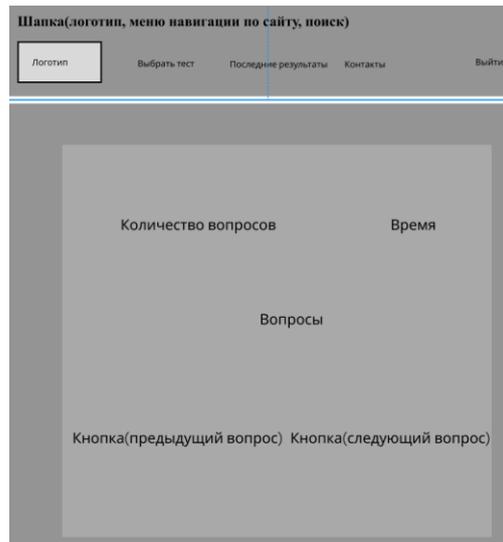


Рисунок 4 – Пример размещения внутренних элементов страниц (тест 1)

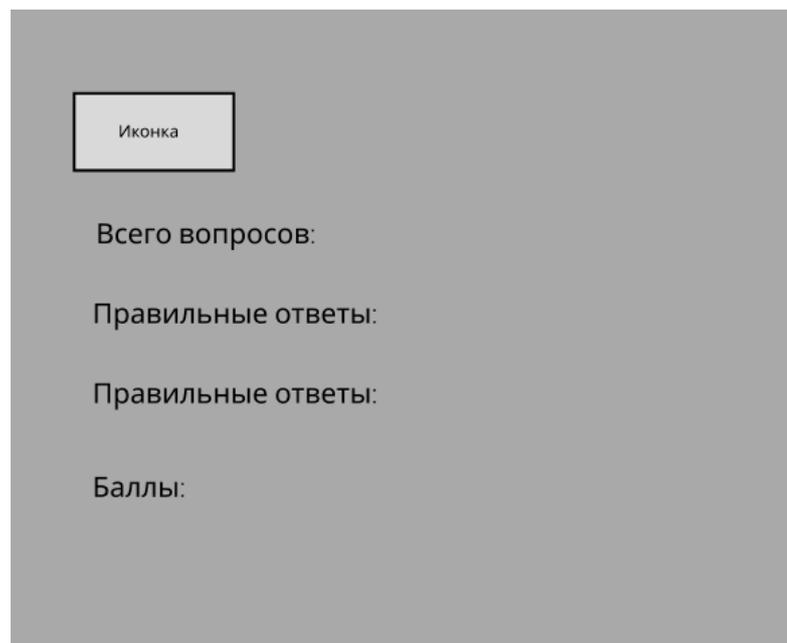


Рисунок 5 – Пример размещения внутренних элементов страниц (результат тестирования)

Главная страница для учителя (рисунок 6) должна содержать графическую часть, навигационное меню сайта, а также контентную область для того, чтобы

учитель с первой страницы мог получить вводную информацию о студентах, которые прошли тестирование.

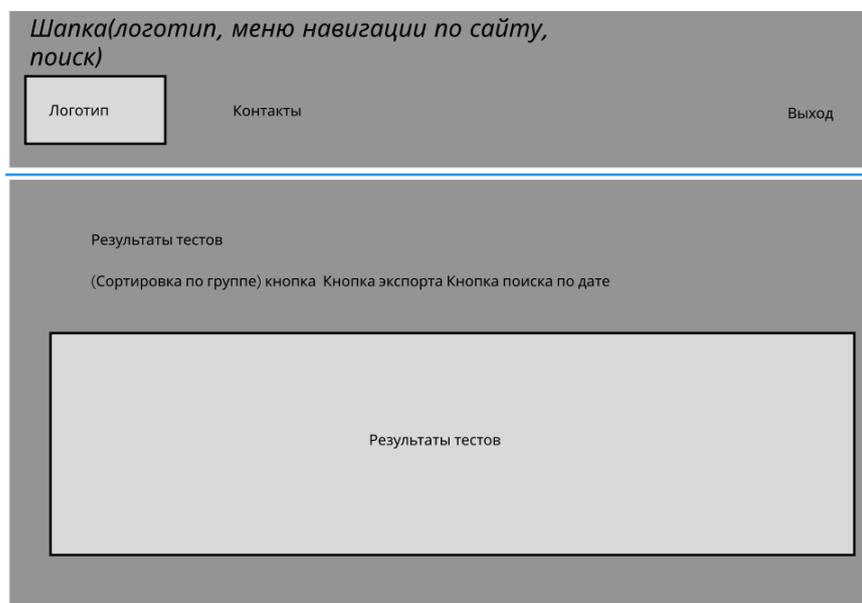


Рисунок 6 – Пример размещения внутренних элементов главной страницы для учителя (результат тестирования)

Контентная область главной и единственной страницы должна делиться на следующие разделы (рисунок 6):

- вверху страницы отображаются облегченная навигационная панель, которая обеспечивает переход к основным пунктам меню сайта (контактной информация и кнопка выхода.);
- основной контент (результаты тестов, экспорт и сортировка студентов);
- подвал.

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

2.1.4 Требования к структуре сайта

Все названия разделов сайта, приведенные ниже, являются условными и могут корректироваться по согласованию с Заказчиком в ходе проектирования.

Сайт должен быть оптимизирован для просмотра на планшетах, компьютерах и для основных типов разрешений.

Элементы управления должны быть сгруппированы однотипно – горизонтально либо вертикально – на всех страницах.

На каждой странице должен отображаться логотип колледжа и контактная информация.

Интерфейс подключаемых модулей должен быть выполнен в едином стиле с интерфейсом ядра системы и должен обеспечивать возможность прозрачного перемещения администратора между модулями системы и использование одинаковых процедур управления и навигационных элементов для выполнения однотипных операций.

2.1.5 Требования к системе управления сайтом

2.1.5.1 Требования к административной части

Для получения доступа к административной части сайта необходимо указать определенный адрес в строке браузера и пройти авторизацию.

Страница административной части (рисунок 7) должна содержать следующие пункты меню:

- Добавление темы теста;
- Добавление вопросов для темы;

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 20 |

- Добавление учетной записи учителя;
- Результаты тестирования;
- Список пользователя;
- Кнопка выхода.

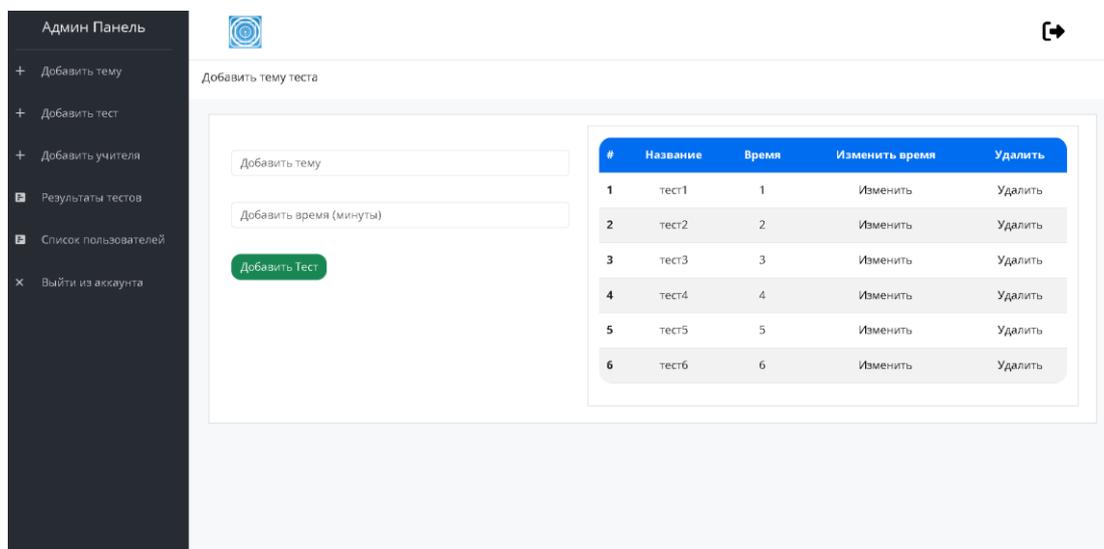


Рисунок 7 – Макет формы главной страницы административной части сайта

2.1.5.2 Требования к управления разделами сайта

Для управления разделами сайта должны быть предусмотрены следующие функции:

- администратор может войти в систему;
- администратор может создавать новые темы тестов, а также удалять их;
- администратор может установить время проведения экзамена и изменить это время;

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

- администратор может добавить вопросы для темы, удалить вопросы и изменить их;
- администратор может видеть результаты тестирования всех студентов и выгружать их в формат xls или csv;
- администратор может видеть список зарегистрированных студентов, делать поиск по определённой группе, удалять их и выгружать их в формат xls или csv;
- администратор может добавить учетную запись для учителя, а также удалить её;
- администратор может выйти из системы.

2.1.5.3 Управление наполнением сайта

Для управления наполнением сайта должны быть предусмотрены следующие блоки (рисунок 8-9):

- название теста;
- тема теста;
- время теста;
- вопросы теста.

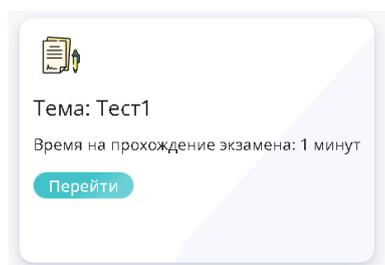


Рисунок 8 – Поля элемента контента

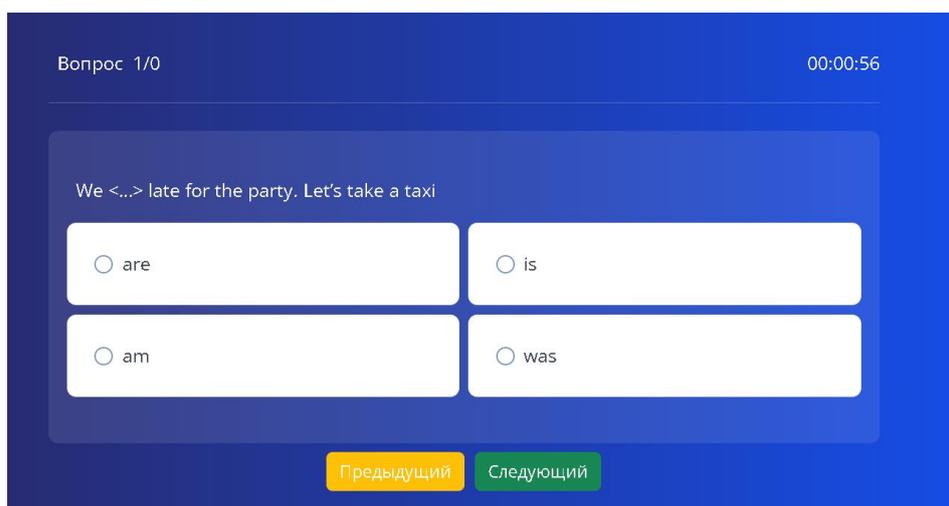


Рисунок 9 – Поля элемента контента (2)

2.1.5.4 Требования к разделению доступа

Все опубликованные разделы сайта должны открываться для доступа на чтение после аутентификации студента, преподавателя и администратора. До аутентификации студент, преподаватель и администратор не смогут зайти на другие разделы. После аутентификации разделы приложения будут доступны.

2.1.6 Требования к видам обеспечения

2.1.6.1 Требования к информационному обеспечению

Все данные сайта должны храниться в структурированном виде под управлением реляционной СУБД MySQL. Исключения составляют файлы данных, предназначенные для просмотра и скачивания (изображения, видео,

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 23 |

документы и т.п.). Такие файлы сохраняются в файловой системе, а в БД размещаются ссылки на них.

В таблице 1 представлена структура таблицы «admin_login». Она предназначена для авторизации администратора

Таблица 1 – admin_login

| Название поля | Тип поля | Описание поля |
|---------------|----------|----------------------------|
| Код админа | Int | Поле идентификации |
| Почта | Char | Почта администратора сайта |
| Пароль | Char | Пароль для авторизации |

В таблице 2 представлена структура таблицы «exam_category». Она предназначена для добавления администратором темы теста.

Таблица 2 - exam_category

| Название поля | Тип поля | Описание поля |
|--------------------|----------|------------------------------|
| Код категории | Int | Поле идентификации |
| Название категории | Char | Название категории экзамена |
| Время экзамена | Date | Время на проведение экзамена |

В таблице 3 представлена структура таблицы «exam_results». Она предназначена для вывода результат тестирования студентов, а также для её экспорта.

Таблица 3 - exam_results

| Название поля | Тип поля | Описание поля |
|--------------------|----------|-------------------------------|
| Код результата | Int | Поле идентификации |
| Код пользователя | Int | Номер пользователя |
| Код категории | Int | Номер категории |
| Код теста | Int | Номер теста |
| Никнейм | Char | Никнейм пользователя |
| Название категории | Char | Название категории экзамена |
| Всего вопросов | Char | Количество всех вопросов |
| Правильные ответы | Char | Количество правильных ответов |

| Название поля | Тип поля | Описание поля |
|---------------------|----------|---|
| Неправильные ответы | Char | Количество неправильных ответов |
| Баллы | Char | Количество баллов за экзамен |
| Время экзамена | Date | Время на проведение экзамена |
| Группа | Char | Группа зарегистрированного пользователя |

В таблице 4 представлена структура таблицы «questions». Она предназначена для добавления вопросов темы администратором сайта.

Таблица 4 - questions

| Название поля | Тип поля | Описание поля |
|--------------------|----------|-----------------------------|
| Код теста | Int | Поле идентификации |
| Код категории | Int | Номер категории |
| Номер теста | Char | Номер вопроса |
| Вопрос теста | Char | Вопрос теста |
| Вариант ответа 1 | Char | Вариант ответа на тест 1 |
| Вариант ответа 2 | Char | Вариант ответа на тест 2 |
| Вариант ответа 3 | Char | Вариант ответа на тест 3 |
| Вариант ответа 4 | Char | Вариант ответа на тест 4 |
| Правильный ответ | Char | Правильный ответ на тест |
| Название категории | Char | Название категории экзамена |

В таблице 5 представлена структура таблицы «registration». Она предназначена для регистрации пользователя.

Таблица 5 - registration

| Название поля | Тип поля | Описание поля |
|------------------|----------|----------------------|
| Код пользователя | Int | Поле идентификации |
| Имя | Char | Имя пользователя |
| Фамилия | Char | Фамилия пользователя |
| Никнейм | Char | Никнейм пользователя |
| Пароль | Char | Пароль пользователя |
| Почта | Char | Почта пользователя |
| Группа | Char | Группа пользователя |

В таблице 6 представлена структура таблицы «teacher». Она предназначена для добавления учетной записи учителя администратором.

Таблица 6 - teacher

| Название поля | Тип поля | Описание поля |
|---------------|----------|--------------------|
| Код учителя | Int | Поле идентификации |
| Никнейм | Char | Никнейм учителя |
| Пароль | Char | Пароль учителя |

2.1.6.2 Требования к языкам программирования

Приложение реализовано с помощью HTML, CSS, PHP, JS.

Для реализации шаблонов должны использоваться языки HTML 4.0 и CSS. Исходный код должен разрабатываться в соответствии со стандартами W3C (HTML 4.0).

Для реализации интерактивных элементов клиентской части должен использоваться язык JavaScript.

Для реализации динамических страниц должен использоваться язык PHP.

2.1.6.3 Требования к организации гиперссылок

Все ссылки в приложении должны быть относительными (за исключением внешних).

2.1.7. Требования к программному обеспечению

2.1.7.1 Требования к программному обеспечению серверной части

Для функционирования приложения необходимо следующее программное обеспечение:

- Операционная система – Windows 10 и Windows Server 2022;
- Веб-сервер – Apache версии не ниже 2.4.29;
- СУБД – MySQL версии не ниже 8.0.22.

2.1.7.2 Требования к клиентскому программному обеспечению

WEB-приложение должно быть доступно для полнофункционального просмотра с помощью следующих браузеров:

- Opera 11.0 и выше;
- Google Chrome 107.0.5304 и выше;
- Яндекс.Браузер 22.1.0.2510 и выше;
- Mozilla Firefox 120 и выше.

2.1.8 Требования к техническому обеспечению

Для функционирования сайта необходимо следующее техническое обеспечение со следующими минимальными характеристиками:

- процессор – Intel Core i3 10 поколения;

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 27 |

- оперативная память – 4 Gb RAM;
- жесткий диск - 120 Gb HDD.

2.1.9 Требования к персоналу

Для эксплуатации WEB-приложения от администратора не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб-браузером.

2.2 Реализация программного продукта

2.2.1 Среда разработки

В ходе разработки WEB-приложения был использован VS Code в качестве основной среды разработки [20]. Вот некоторые ключевые плагины, которые были использованы:

- расширение терминала (Terminal);
- расширение Git;
- расширение prn;
- расширение Gulp;
- Emmet.

Встроенное расширение терминала позволяет взаимодействовать с командной строкой или терминалом прямо из VS Code. Это удобно для

выполнения различных команд, управления версиями или запуска скриптов без необходимости переключаться на внешнюю консоль.

Встроенная поддержка Git в VS Code обеспечивает возможность работать с системой контроля версий Git без необходимости использовать внешние инструменты. Оно позволяет выполнять операции коммита, слияния, ветвления, отслеживания изменений и другие действия, упрощая работу с репозиторием Git.

Расширение npm в VS Code предоставляет функциональность для работы с пакетным менеджером npm (Node Package Manager). Оно позволяет управлять зависимостями проекта, устанавливать, обновлять и удалять пакеты, а также выполнять другие операции, связанные с разработкой на основе Node.js.

Gulp является сборщиком проектов и в проекте использовался для автоматизации различных задач разработки. С помощью соответствующего расширения в VS Code можно управлять и запускать задачи Gulp, такие как компиляция, сжатие, оптимизация файлов, обновление файлов в реальном времени при изменении CSS и JavaScript, что значительно упрощает процесс разработки.

Emmet является плагином, облегчающий и ускоряющий процесс написания HTML и CSS кода. Встроенная поддержка Emmet в VS Code позволяет использовать сокращения и сниппеты для быстрого и эффективного написания разметки и стилей.

VS Code обладает встроенной отладочной поддержкой, что позволяет разработчикам выполнять отладку своего кода непосредственно в редакторе. Можно устанавливать точки останова, выполнять шаги отладки, просматривать значения переменных и выполнять другие отладочные операции, что значительно упрощает обнаружение и исправление ошибок.

В процессе разработки проекта в среде разработки VS Code были использованы следующие полезные функции:

- интегрированная отладка (Integrated Debugging);
- интеграция с системой контроля версий (Version Control Integration);

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 29 |

- автодополнение кода (Intellisense);
- расширяемость (Extensibility);
- интегрированная поддержка терминала (Integrated Terminal).

VS Code предлагает широкий выбор инструментов и плагинов, которые использованы в проекте, что делает его удобным инструментом для разработки.

2.2.2 Языки программирования

Язык программирования — это формальный набор инструкций и правил, которые используются для написания программ компьютера. Они определяют способы описания алгоритмов и структуру данных, позволяют управлять выполнением программы и взаимодействовать с компьютерными ресурсами.

Для дипломного проекта были использованы следующие три языка программирования:

- PHP;
- JavaScript;
- SQL.

PHP (Hypertext Preprocessor) был использован для разработки основной функциональности веб-приложения. Этот язык программирования широко используется для создания динамических веб-сайтов и веб-приложений. В проекте PHP был применен для создания административной панели, реализации механизма аутентификации, управления сессиями, удаление, добавление и изменение тестов, добавление учётной записи для учителей, а также её удаление, выгрузка результатов тестирования, выгрузка списка зарегистрированных пользователей, и обработки данных на серверной стороне.

JavaScript является клиентским языком программирования, который был использован для реализации анимаций и обеспечения асинхронного поведения веб-приложения. С помощью JavaScript были созданы кастомные анимации, которые улучшают пользовательский опыт и делают интерфейс более привлекательным. Также JavaScript использовался для выполнения асинхронных запросов к серверу без перезагрузки страницы, что позволяет пользователям взаимодействовать с приложением без задержек.

SQL (Structured Query Language) использовался в рамках среды разработки Open Server для работы с базой данных MySQL. SQL предоставляет мощные возможности для создания, изменения и извлечения данных из базы данных. В проекте SQL был использован для хранения и управления информацией о тестах, результатов студентов и других данных, необходимых для функционирования приложения.

2.2.3 Методы, использованные при разработке WEB-приложения

В ходе разработки WEB-приложения был использован метод управления сессиями на основе сессионных файлов.

Код, который представлен на скриншоте, используется для управления сессиями пользователей на веб-сайте и обеспечивает некоторую защиту и безопасность.

Первая строка `session_start()` запускает сессию или возобновляет текущую сессию, если она уже существует. Сессия — это механизм, который позволяет хранить и передавать данные между страницами веб-сайта для конкретного пользователя (рисунок 10).

Для обеспечения безопасности, код проверяет время бездействия пользователя. Если прошло время, превышающее время жизни сессии, пользователь считается неактивным, и снова происходит перенаправление на страницу логина.

Чтобы отслеживать активность пользователя, код обновляет время последней активности (`$_SESSION["last_activity"] = time();`) при каждом действии пользователя. Это позволяет сравнивать время последней активности с текущим временем и определять, прошло ли достаточно времени бездействия для истечения срока сессии.

Чтобы продлить время жизни сессии при каждом действии пользователя, код проверяет существование куки сессии на стороне клиента (`isset($_COOKIE[session_name()])`). Если куки сессии существуют, то устанавливается новое время жизни для куки, чтобы оно соответствовало времени жизни сессии на сервере.

Для дополнительной защиты от атак перехвата сессии, код обновляет идентификатор сессии (`session_regenerate_id(true)`) каждые 5 минут (`isset($_SESSION["last_regenerate"]) && (time() - $_SESSION["last_regenerate"] > 3000)`). Это создает новый уникальный идентификатор сессии и старый идентификатор становится недействительным.

Обновление идентификатора сессии помогает предотвратить возможность злоумышленникам использовать старый идентификатор для несанкционированного доступа.

Если переменная "last_regenerate" не существует в сессии, она устанавливается в текущее время. Это гарантирует, что идентификатор сессии будет обновлен при первой активности пользователя после создания сессии.

Чтобы отслеживать активность пользователя, код обновляет время последней активности (`$_SESSION["last_activity"] = time();`) при каждом действии пользователя. Это позволяет сравнивать время последней активности с текущим

временем и определять, прошло ли достаточно времени бездействия для истечения срока сессии.

Чтобы продлить время жизни сессии при каждом действии пользователя, код проверяет существование куки сессии на стороне клиента (if (isset(\$_COOKIE[session_name()]))). Если куки сессии существуют, то устанавливается новое время жизни для куки, чтобы оно соответствовало времени жизни сессии на сервере.

Для дополнительной защиты от атак перехвата сессии, код обновляет идентификатор сессии (session_regenerate_id(true)) каждые 5 минут (if (isset(\$_SESSION["last_regenerate"]) && (time() - \$_SESSION["last_regenerate"] > 3000))). Это создает новый уникальный идентификатор сессии и старый идентификатор становится недействительным.

Обновление идентификатора сессии помогает предотвратить возможность злоумышленникам использовать старый идентификатор для несанкционированного доступа.

В итоге, этот код обеспечивает следующую функциональность:

- проверка авторизации пользователя и перенаправление на страницу логина, если пользователь не авторизован;
- контроль времени бездействия пользователя и перенаправление на страницу логина при превышении времени жизни сессии;
- обновление времени последней активности пользователя при каждом действии;
- продление времени жизни сессии при каждом действии пользователя.
- обновление идентификатора сессии каждые 5 минут для предотвращения атак перехвата сессии;
- обеспечение безопасного и надежного управления сессиями пользователей на веб-сайте.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 34 |

2.2.4 Методы отладки

Отладка — это процесс поиска, идентификации и исправления ошибок в программном коде. Она позволяет выявить проблемные места в приложении и устранить возникшие ошибки. Во время отладки можно использовать различные методы и инструменты для анализа кода и его выполнения [17].

При разработке дипломного проекта тестирования по английскому языку для ГБПОУ РО «РКРИПТ» использовались различные методы отладки для проверки кода и обнаружения ошибок:

- вывод на экран (`print/echo`), использовались две команды `print` и `echo` для вывода значений переменных, сообщений об ошибках или другой отладочной информации непосредственно на странице;
- журналирование (`logging`), с помощью неё можно настроить запись отладочной информации в файл журнала. Для этого использовалась функция `error_log()`;
- отладчик (`debugger`): PHP также предоставляет набор инструментов для отладки, таких как Xdebug. Они позволяют установить точки останова, отслеживать выполнение кода шаг за шагом, анализировать значения переменных и многое другое;
- проверка ошибок (`error checking`). PHP имеет множество встроенных функций для проверки наличия ошибок в коде. Некоторые из них полезны при отладке, например, `error_reporting()`, `ini_set('display_errors', 1)`, `ini_set('log_errors', 1)`;
- `var_dump()` и `print_r()`. Эти функции помогают выводить содержимое переменных для анализа. Они особенно полезны при отладке сложных структур данных, таких как массивы или объекты;

– `die()` и `exit()`. Эти функции необходимы, чтобы остановить выполнение скрипта в определенной точке и вывести отладочное сообщение. Это полезно для проверки определенной части кода или убедиться, что код до этого места выполняется корректно;

– `error_reporting()`. Эта функция позволяет настроить уровень отображаемых ошибок и предупреждений во время выполнения скрипта. Она использовалась для проверки ошибок, которые не отображаются по умолчанию, и получения более подробной информации о проблеме.

В Visual Studio Code (VS Code) доступны мощные возможности отладки, которые помогают идентифицировать и исправлять ошибки в вашем коде. Вот некоторые из них, которые использовались:

– установка точек останова (breakpoints), можно установить точки останова в коде, указав конкретные строки, где необходимо было приостановить выполнение программы. Когда код достигает точки останова, выполнение приостанавливается, можно анализировать состояние переменных и выполнять другие отладочные операции;

– обнаружение ошибок времени выполнения: VS Code может автоматически обнаруживать ошибки времени выполнения, такие как исключения, и предлагать соответствующие действия. Это помогает быстро идентифицировать и исправлять проблемы, которые возникают во время выполнения кода;

– логирование и вывод сообщений: В VS Code можно использовать консоль для вывода сообщений, ошибок и отладочной информации. Это позволяет визуализировать промежуточные результаты, сообщения об ошибках или другую отладочную информацию прямо в среде разработки [18].

2.3 Тестирование WEB-приложения

Тестирование WEB-приложений – это наблюдение за функционированием программы в искусственно созданных обстоятельствах, которое позволяет определить ее соответствие поставленным требованиям, например: производительность WEB-приложения, его степень защиты и комфорт использования для рядового пользователя [10].

Задачи тестирования WEB-приложений:

- нахождение ошибок в работе программы и эффективное их устранение;
- проверка приложения на соответствие конкретным требованиям;
- проверка качества работы создателей приложения;
- обработка информации, которая станет основой для принятия последующих действий.

Информация, полученная в процессе тестирования, позволяет выстроить последующую стратегию развития приложения. Важно понимать, что тестирование WEB-приложений является важным этапом, так как даже самые успешные программисты могут допустить ошибки, которые в последствии могут стать причиной уязвимости WEB-приложения и утечки информации его пользователей.

Выделяются такие виды тестирования программ, как:

- тестирование функционала;
- оценка удобства использования веб-приложения;
- оценка интерфейса;
- оценка производительности;
- оценка безопасности.

Клиент является первым элементом архитектуры, где под клиентом понимается браузер, и здесь сразу возникает вопрос тестирования кроссбраузерности.

Кроссбраузерность — это тестирование правильности работы WEB-приложения в различных браузерах и на различных операционных системах.

Кроссбраузерное тестирование проводит проверку по таким пунктам:

- функционал, который реализуется на «стороне» клиента;
- проверка правильности отображения графических элементов WEB-приложения;
- интерактивность веб-приложения.

Методика тестирования — это систематический и организованный подход к выполнению тестирования, включающий набор шагов, процедур, инструментов и методов, которые помогают достичь поставленных целей тестирования [21].

В таблице 7 представлена методика тестирования.

Таблица 7 – Методика тестирования

| № | Действие | Результат |
|---|--|---|
| 1 | Сценарий «Запуск сайта» | |
| | Запустить сайт по ссылке < https://online-quiz.ru/ > | Выдается главная страница сайта |
| 2 | Сценарий «Просмотр информации» | |
| | Нажать на кнопку «Регистрация». | Открывается страница с формой регистрации. |
| | Нажать на кнопку «Авторизация». | Открывается страница с формой авторизации, после авторизации происходит перенаправление на главную страницу сайта и открывается доступ к переходу другим страницам сайта. |
| | Нажать на кнопку «Перейти к тестированию». | Открывается страница теста, где необходимо ответить на все вопросы, после произойдет |

| № | Действие | Результат |
|---|--|---|
| | | перенаправление на страницу результат тестирования. |
| | Нажать на раздел «Последние результаты». | Откроется страница со всеми результатами тестирования за всё время. |
| | Нажать на раздел «Контакты». | Открывается страница контактной информации. |
| 3 | Сценарий «Экспорт расписания» | |
| | На странице с последние результаты нажать на кнопку «Экспорт в xls или csv». | Запускается загрузка файла xls или csv с результатами тестирования. |
| 4 | Сценарий «Поиск» | |
| | Ввод в пустое поле «Поиск». | После ввода происходит поиск тестов по темам |
| 5 | Сценарий «Проверка ссылок» | |
| | Перейти по всем ссылкам на сайте | Все ссылки на сайте работают корректно и ведут на соответствующие страницы |
| 6 | Сценарий «Выход» | |
| | Нажать на раздел «Выход» | Откроется кнопка выхода, после её нажатия закроется сессия пользователя, произойдёт перенаправление на страницу авторизации и закроется доступ ко всем остальным страницам, кроме регистрации пока пользователь не авторизуется |

2.4 Защита данных и WEB-приложения

Защита данных — это процесс обеспечения конфиденциальности, целостности и доступности данных от несанкционированного доступа, изменений или уничтожения. Она включает в себя применение различных мер и методов для

защиты данных на всех этапах их жизненного цикла - от сбора и хранения до передачи и уничтожения [12].

Ключевые аспекты защиты данных включают:

- конфиденциальность, защита данных от несанкционированного доступа. Это включает использование шифрования данных, контроля доступа и механизмов аутентификации;

- целостность, предотвращение несанкционированных изменений данных. Используются методы хеширования и контроля целостности данных для обнаружения и предотвращения несанкционированных изменений;

- доступность, обеспечение доступности данных для авторизованных пользователей. Меры доступности включают создание резервных копий данных, репликацию и обеспечение надежности инфраструктуры;

- аудит и мониторинг, отслеживание активности пользователей и системы для обнаружения потенциальных угроз или нарушений безопасности. Это включает использование систем журналирования, мониторинга сетевой активности и анализа данных.

WEB-приложение — это программное обеспечение, которое разрабатывается для выполнения определенных функций через Интернет с помощью веб-браузера. WEB-приложения обычно состоят из клиентской (Front-End) и серверной (Back-End) частей и включают в себя интерактивные веб-страницы, базы данных, обработку данных и взаимодействие с пользователями.

Важные аспекты WEB-приложений включают:

- аутентификация и авторизация: Обеспечение безопасного доступа к WEB-приложению только авторизованным пользователям. Это включает использование паролей, механизмов аутентификации, управление правами доступа и контроль идентификации пользователей.

- защита от взлома: Защита от типичных атак, таких как инъекции SQL, межсайтовый скриптинг (XSS), подделка межсайтовых запросов (CSRF) и других.

Для этого используются механизмы фильтрации ввода данных, проверка безопасности кода и обновление системы;

– шифрование данных: Защита конфиденциальности данных, передаваемых между клиентом и сервером. Шифрование SSL/TLS обеспечивает защищенное соединение и предотвращает перехват данных;

– управление сеансами: Эффективное управление сеансами пользователей, включая создание уникальных идентификаторов сеансов, их безопасное хранение и контроль времени сеанса;

– обновление и патчи: Регулярное обновление WEB-приложения и его компонентов для устранения уязвимостей безопасности и применения исправлений.

2.5 Описание порядка работы с программным продуктом

2.5.1 Общие сведения

Основная цель данного WEB-приложения - предоставить студентам возможность проходить многократно короткие тесты по предмету «Английский язык», проводимые в течение всего срока обучения.

Данные, полученные в результате этих тестов, могут быть соотнесены с необходимыми стандартами, для того чтобы обучающиеся могли быть измерены по этим стандартам.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 41 |

2.5.2 Порядок запуска WEB-приложения

Открытие WEB-приложения для студента происходит через переход по ссылке < <http://online-quiz/> >. Открывается форма с авторизацией, если студент не зарегистрирован, то при нажатии на кнопку регистрация открывается форма регистрацией, после успешной проверки регистрации студент может авторизоваться (рисунок 11-12).

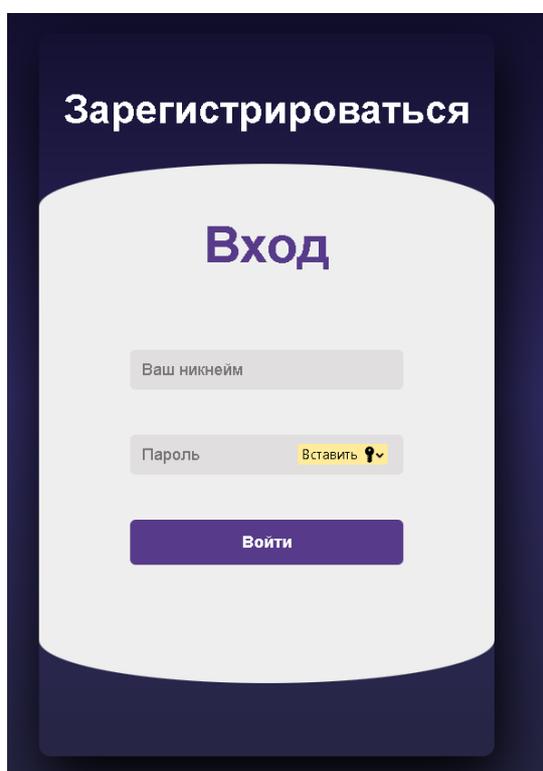


Рисунок 11 – Страница авторизации для студента

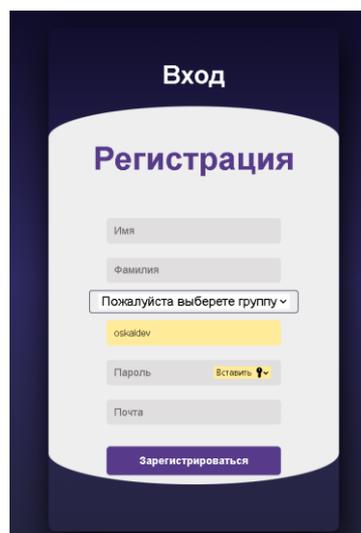


Рисунок 12 – Страница регистрации для студента

Открытие WEB-приложения для учителя происходит через переход по ссылке <https://online-quiz.ru/teacher/login>. Открывается форма с авторизацией, саму учетную запись учителя добавляет администратор сайт (рисунок 13-14).

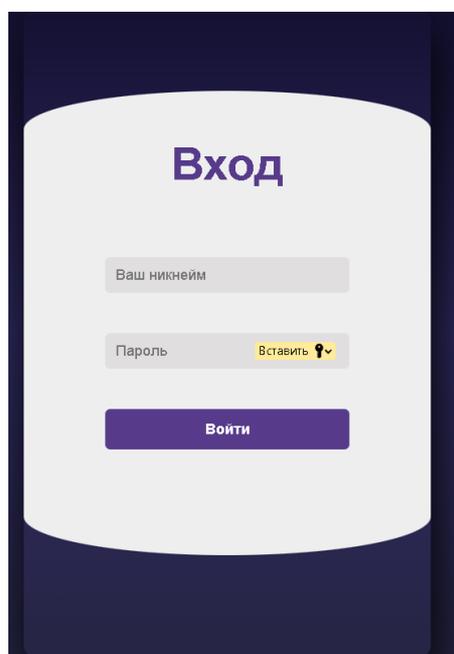


Рисунок 13 – Страница регистрации для учителя

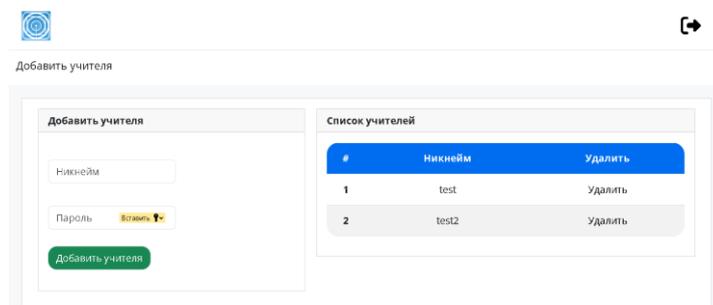


Рисунок 14 – Страница добавления учетной записи для учителя

2.5.3 Работа с WEB-приложением

После авторизации и перейдя на главную страницу, студент знакомится с шапкой сайта (рисунок 15). На шапке располагается иконка колледжа, навигационное меню состоящая из страниц (выбрать тест, последние результаты, контакты и кнопка выхода из учётной записи), а чуть ниже располагается поиск по тестам.



Рисунок 15 – Шапка сайта

После прокрутки вниз, студент видит главный блок, в котором расположены тесты по темам с краткой информацией, тема теста и его время прохождения (рисунок 16).

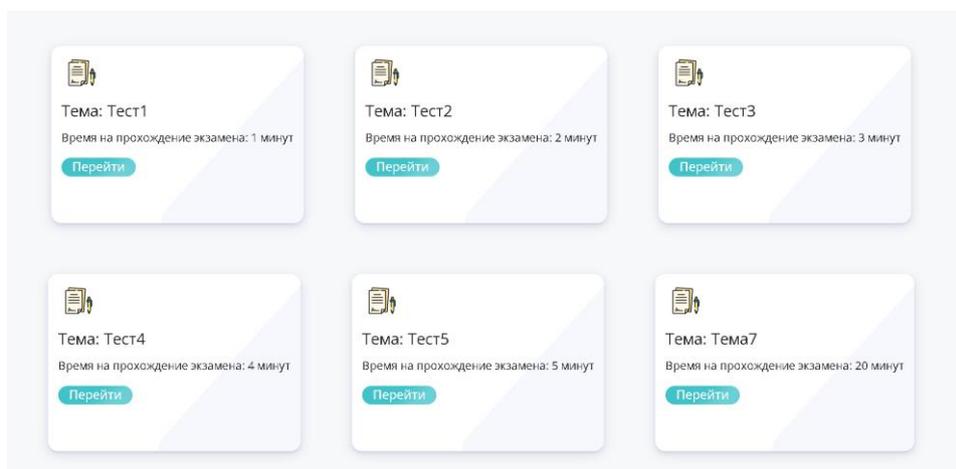


Рисунок 16 – Тесты

После прокрутки тестов будет расположен подвал сайта (рисунок 17) состоящая из (наименования колледжа, фотографии колледжа, ссылка на страницу техподдержки, ссылка на соцсеть колледжа и её почту).

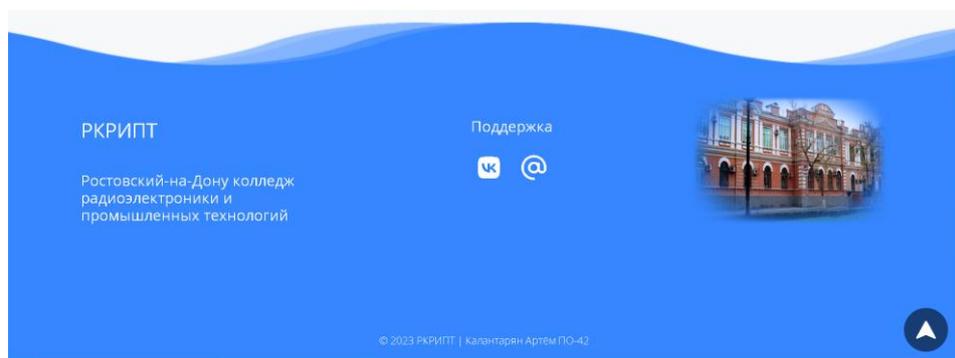


Рисунок 17 – Подвал

Вернёмся к тестам сайта, при нажатии на тест, произойдет перенаправление пользователя на прохождение теста по выбранной теме (рисунок 18). Сам тест состоит из:

- Шапки сайта;
- Подвал;

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

- Количество вопросов теста;
- Время прохождения;
- Вопросы;
- Кнопка перехода на следующий вопрос и на предыдущий.

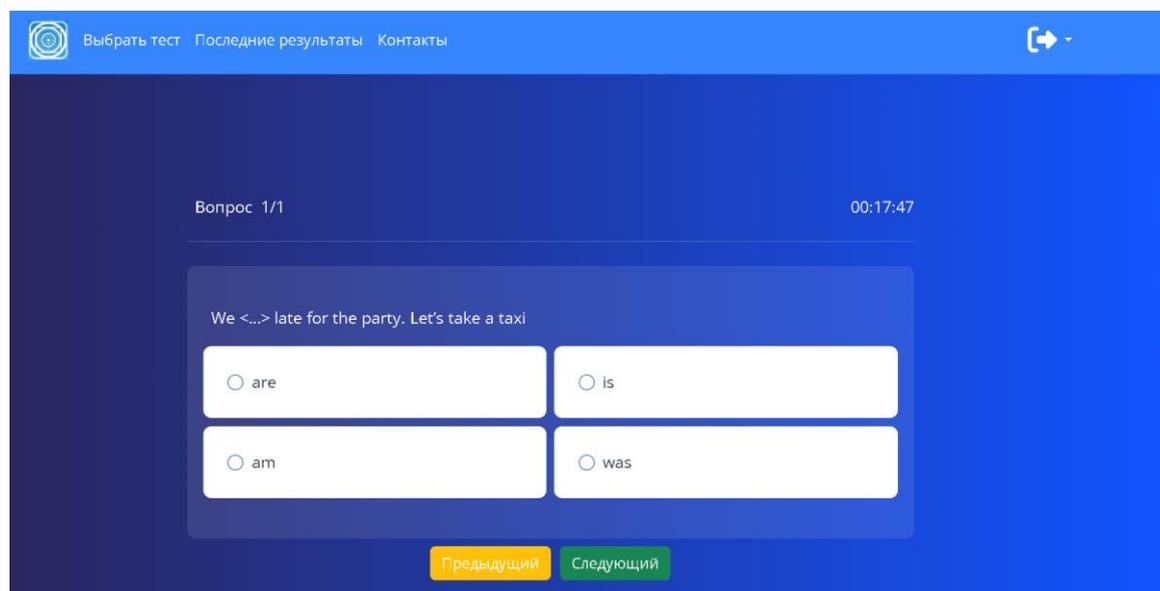


Рисунок 18 – Тест

После прохождения теста, произойдет перенаправление на страницу результат. В котором будет описано количество всего вопросов, количество правильных и не правильных ответов, баллы и информация, как происходит подсчёт баллов (рисунок 19).

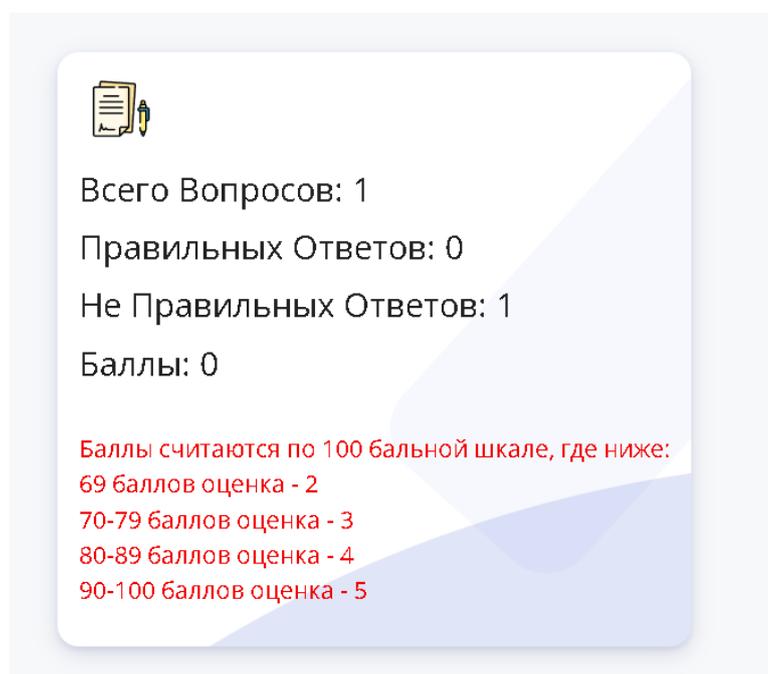


Рисунок 19 – Страница результата тестирования

После прохождения тестирования, студент может нажать в навигационном меню на раздел «Последние результаты» и перейти на страницу последних результатов тестов.

В ней будет информация по всем пройденным тестам (дата, никнейм пользователя, тема теста, всего вопросов, количество правильных и не правильных ответов, баллы) за всё время. Эти результаты также можно будет экспортировать в форматы xls и csv (рисунок 20-22).

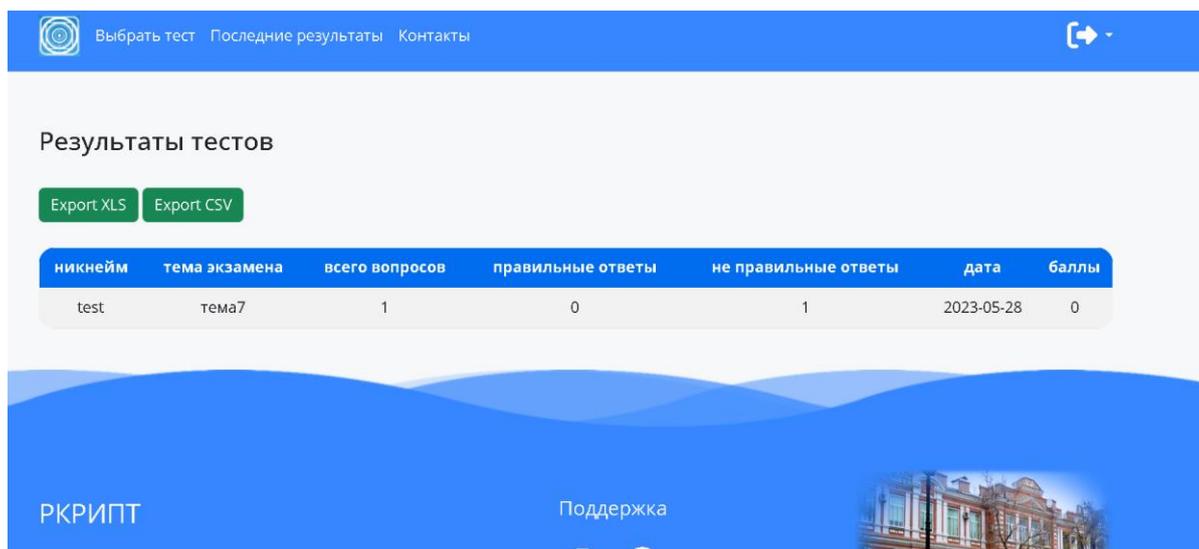


Рисунок 20 – Страница Последних результатов тестирования

| | A | B | C | D | E | F | G |
|---|---------|-----------|-----------|----------|----------|----------------|-------|
| 1 | никнейм | тема экза | всего вог | правильн | не прави | время экзамена | баллы |
| 2 | test | тема7 | 1 | 0 | 1 | 28.05.2023 | 0 |
| 3 | | | | | | | |
| 4 | | | | | | | |

Рисунок 21– Экспорт csv результатов тестирования

| | A | B | C | D | E | F | G |
|---|---------|---------------|----------------|--------------------|-----------------------|----------------|-------|
| 1 | никнейм | тема экзамена | всего вопросов | правильных ответов | не правильных ответов | время экзамена | баллы |
| 2 | test | тема7 | 1 | 0 | 1 | 28.05.2023 | 0 |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |

Рисунок 22 – Экспорт xls результатов тестирования

Нажатие в подвале на раздел «Поддержка» пользователя, перенаправит на страницу формы заявки, где студент может ввести своё имя, телефон или почту и задать свой вопрос (рисунок 23).

Рисунок 23 – Страница техподдержки

Нажав в подвале на раздел «Контакты» пользователя, перенаправит на главный сайт колледжа страницу контакты (рисунок 24).

Рисунок 24 – Страница контакты

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

Рассмотрим работу учителя с WEB-приложением. После авторизации и перейдя на главную страницу, учитель знакомится с шапкой сайта (рисунок 25). На шапке располагается иконка колледжа, ссылка на контакты колледжа и кнопка выхода из учетной записи.

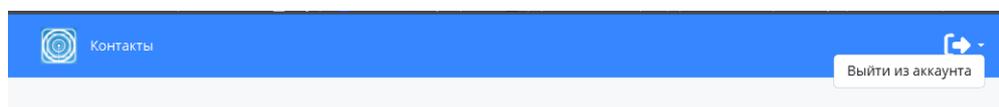


Рисунок 25 – Шапка главной страницы для учителя

После прокрутки вниз, учитель видит главный блок, в котором расположены результаты тестирования всех студентов. Сами результаты можно сортировать по группе студентов. При выборе группы и нажатия кнопки найти произойдет поиск по выбранной группе. Так же кроме сортировки по группе можно произвести экспорт в форматы xls и csv. Сами форматы также можно сортировать по определенной группе студентов и даты прохождения экзамены. При выборе неверного формата, учителя перенаправит на пустую страницу с выводом, что нет результатов для экспорта (рисунок 26-28).

Результаты тестов

Все

| никнейм | группа | тема экзамена | всего вопросов | правильные ответы | не правильные ответы | дата | баллы |
|-----------|--------|---------------|----------------|-------------------|----------------------|------------|-------|
| oskaldev1 | ПО-22 | тест1 | 2 | 2 | 0 | 2023-05-28 | 100 |
| oskaldev1 | ПО-22 | тест1 | 2 | 0 | 2 | 2023-05-28 | 0 |
| oskaldev | ПО-42 | тема7 | 1 | 0 | 1 | 2023-05-28 | 0 |
| oskaldev | ПО-42 | тема7 | 1 | 1 | 0 | 2023-05-28 | 100 |

Рисунок 26 – Результаты тестирования всех студентов

| | A | B | C | D | E | F | G | H | I |
|---|-----------|--------|---------------|----------------|--------------------|-----------------------|----------------|-------|---|
| 1 | никнейм | группа | тема экзамена | всего вопросов | правильных ответов | не правильных ответов | время экзамена | баллы | |
| 2 | oskaldev1 | ПО-22 | тест1 | 2 | 2 | 0 | 28.05.2023 | 100 | |
| 3 | oskaldev1 | ПО-22 | тест1 | 2 | 0 | 2 | 28.05.2023 | 0 | |
| 4 | | | | | | | | | |
| 5 | | | | | | | | | |
| 6 | | | | | | | | | |

Рисунок 27 – Экспорт по определённой группе и даты тестирования

| | A | B | C | D | E | F | G | H | I |
|----|-----------|--------|---------------|----------------|--------------------|-----------------------|----------------|-------|---|
| 1 | никнейм | группа | тема экзамена | всего вопросов | правильных ответов | не правильных ответов | время экзамена | баллы | |
| 2 | oskaldev1 | ПО-22 | тест1 | 2 | 2 | 0 | 28.05.2023 | 100 | |
| 3 | oskaldev1 | ПО-22 | тест1 | 2 | 0 | 2 | 28.05.2023 | 0 | |
| 4 | oskaldev | ПО-42 | тема7 | 1 | 0 | 1 | 28.05.2023 | 0 | |
| 5 | oskaldev | ПО-42 | тема7 | 1 | 1 | 0 | 28.05.2023 | 100 | |
| 6 | test | | тема7 | 1 | 0 | 1 | 28.05.2023 | 0 | |
| 7 | | | | | | | | | |
| 8 | | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |
| 11 | | | | | | | | | |

Рисунок 28 – Экспорт по всего списка тестирования

2.5.4 Порядок запуска административной части WEB-приложения

Открытие WEB-приложения для администратора происходит через переход по ссылке < <https://online-quiz.ru/admin/> >. Открывается форма с авторизацией (рисунок 29).

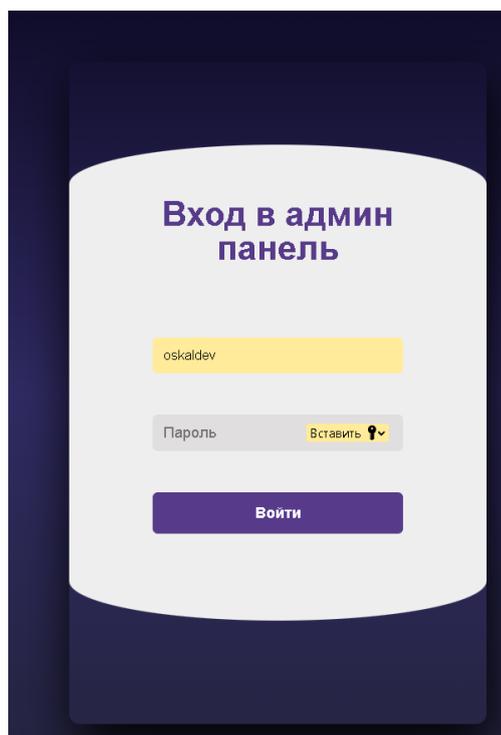


Рисунок 29 – Вход в административную панель

После авторизации и перейдя на главную страницу, администратор знакомится с боковым меню сайта, а также со всеми остальными страницами (рисунок 30). Боковое меню состоит из разделов:

- Добавить тему;
- Добавить тест;
- Добавить учителя;
- Результаты тестов;
- Список пользователей;
- Выйти из аккаунта.

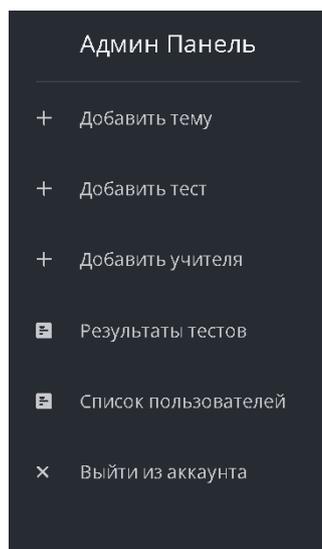


Рисунок 30 – Боковое меню административной панели

При нажатии на раздел «Добавить тему», откроется страница добавление темы для теста, состоящая из (название темы и времени). Чуть правее будут расположены все названия тем, их время, изменение времени и удаления. После ввода всех полей, в базу данных занесётся тема и время теста (рисунок 31).

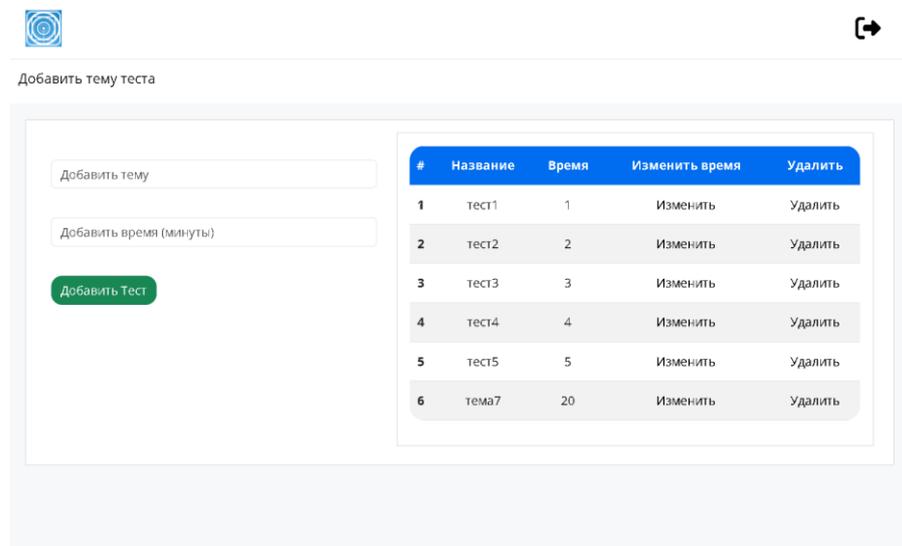
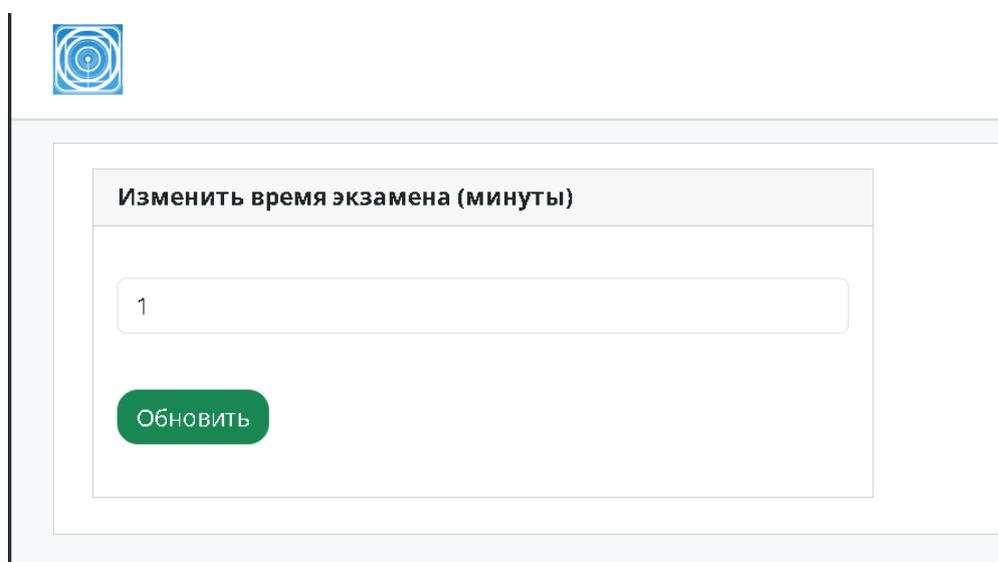


Рисунок 31 – Страница добавить тему

При нажатии на кнопку Изменить откроется страница изменения времени (рисунок 32).

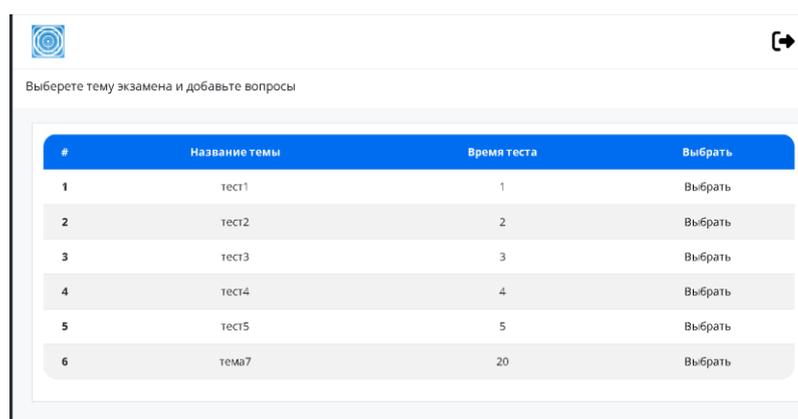


Изменить время экзамена (минуты)

Обновить

Рисунок 32 – Страница изменение времени для темы теста

При нажатии на раздел «Добавить тест», откроется страница добавление теста, в которой расположена название всех тем теста, которые администратор добавил, время, номер теста и кнопка выбора. (рисунок 33).



Выберете тему экзамена и добавьте вопросы

| # | Название темы | Время теста | Выбрать |
|---|---------------|-------------|---------|
| 1 | тест1 | 1 | Выбрать |
| 2 | тест2 | 2 | Выбрать |
| 3 | тест3 | 3 | Выбрать |
| 4 | тест4 | 4 | Выбрать |
| 5 | тест5 | 5 | Выбрать |
| 6 | тема7 | 20 | Выбрать |

Рисунок 33 – Страница добавить тест

После того, как администратор выбрал тему и нажал на кнопку «Выбрать» откроется страница с четырьмя вариантами добавления вопросов для темы. (рисунок 34-36) Каждый вариант вопросов различается между с собой. Чуть ниже после добавления вопросов, отображается все добавленные вопросы, которые можно изменить (поменять вопрос и ответы на него) (рисунок 37) и кнопка удаления этого вопроса из базы данных.

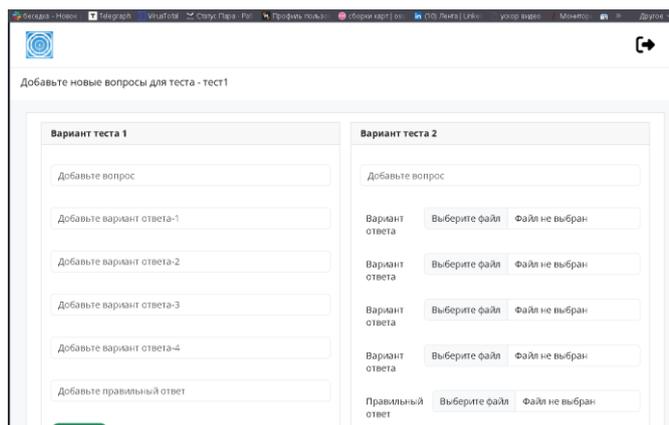


Рисунок 34 – Вариант вопросов 1-2

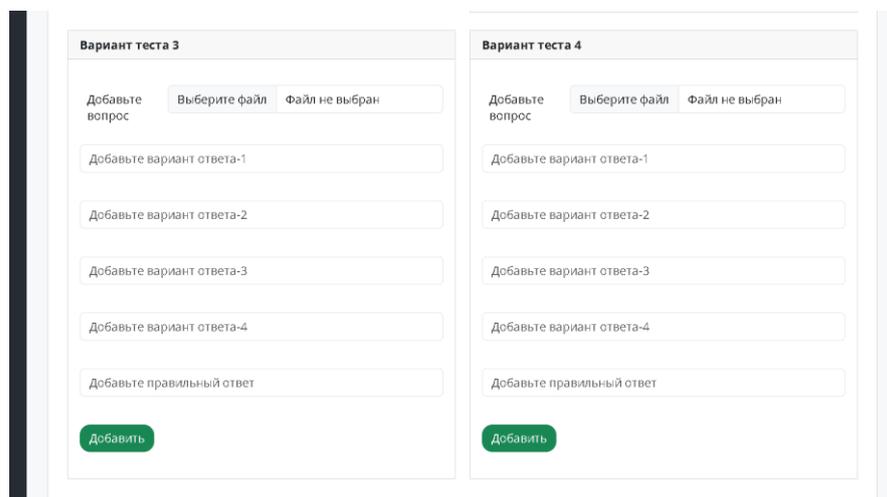


Рисунок 35 – Вариант вопросов 3-4

| ID | Вопросы | Ответ-1 | Ответ-2 | Ответ-3 | Ответ-4 | Правильный ответ | Изменить | Удалить |
|----|--|---------|---------|---------|---------|------------------|----------|---------|
| 1 | We <...> late for the party. Let's take a taxi | are | is | am | was | are | Изменить | Удалить |
| 2 | I <input type="text" value="heard"/> | | | | | heard | Изменить | Удалить |

Рисунок 36 – Все добавленные вопросы

Обновить вопросы

Изменить вопросы

Изменить вопрос: We <...> late for the party. Let's take a taxi

Изменить вариант ответа-1: are

Изменить вариант ответа-2: is

Изменить вариант ответа-3: am

Изменить вариант ответа-4: was

Изменить правильный ответ: are

Обновить

Рисунок 37 – Страница редактирование вопросов

При нажатии на раздел «Добавить учителя», откроется страница добавление учетной записи для учителя, в которой расположена название всех добавленных учетных записей и кнопка удаления их. Для защиты предусмотрено, что администратор не может вводить одинаковый никнейм два раза (рисунок 38).

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

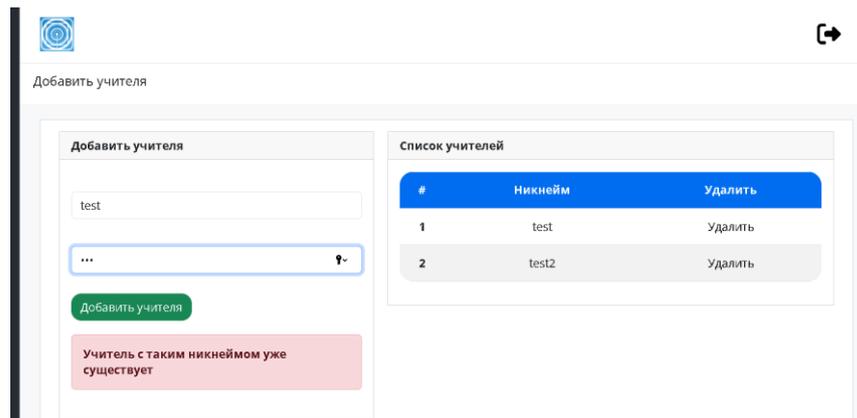


Рисунок 38 – Страница добавления учителя

При нажатии на раздел «Результаты тестов», откроется страница результатов тестирования всех студентов за всё время, эти результаты можно экспортировать в двух форматах xls и csv (рисунок 39).

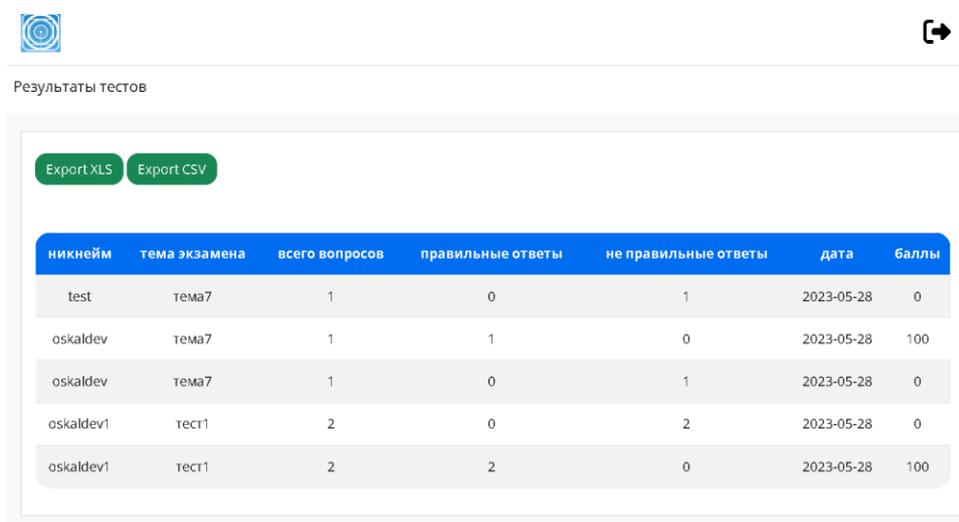


Рисунок 39 – Страница результаты тестов

При нажатии на раздел «Список пользователей», откроется страница всех зарегистрированных студентов с параметрами (никнейм, имя, фамилия, группа,

почта), этот список можно сортировать по группе, удалять несколько сразу учетных записей или одну, экспортировать весь список в форматах xls и csv (рисунок 40-41).

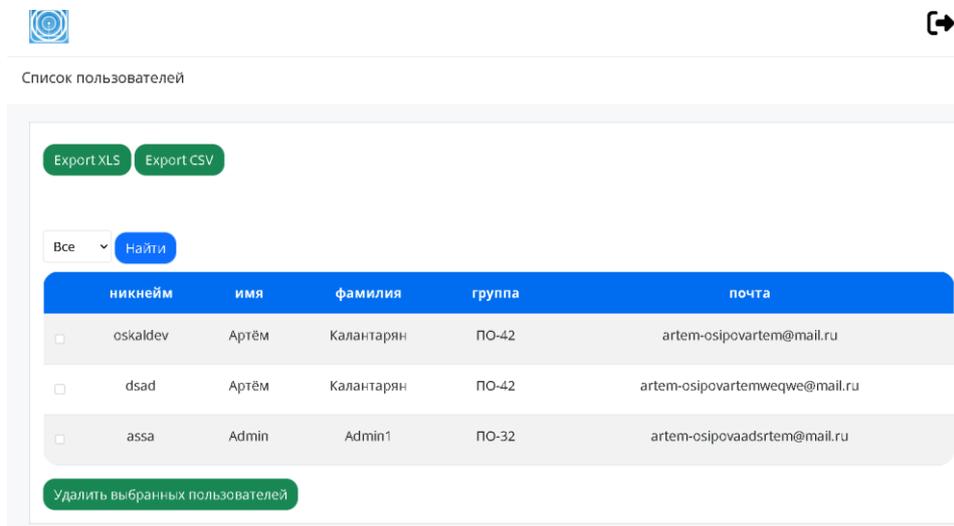


Рисунок 40 – Страница список пользователей

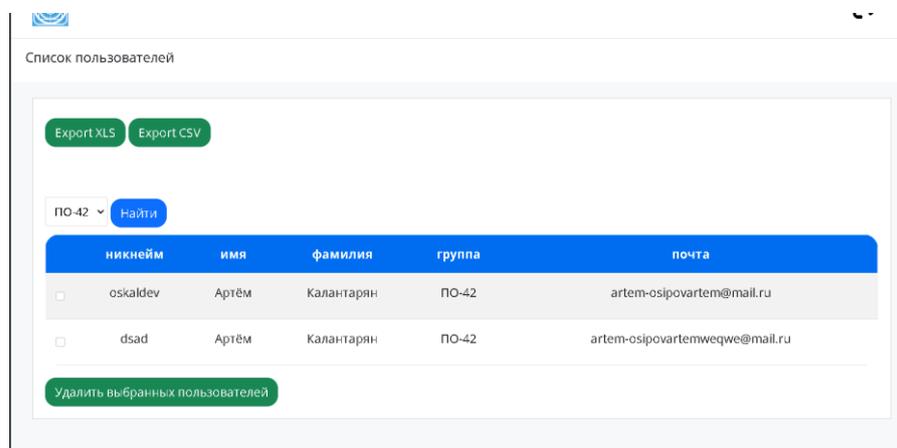


Рисунок 41 – Сортировка пользователей по группе

При нажатии на раздел «Выйти из аккаунта», произойдет закрытие сессии аккаунта и перенаправление на страницу авторизации, без доступа до авторизации перехода по другим страницам.

2.6 Охрана труда пользователей программного продукта

2.6.1 Общие положения

Охрана труда является одной из важнейших составляющих процесса создания программного продукта. Она направлена на обеспечение безопасности труда пользователей программного продукта и снижение рисков для их здоровья.

Рабочее место программиста — это место, где программист выполняет свою работу, разрабатывая, тестируя и поддерживая программное обеспечение. Оно включает в себя компьютер, монитор, клавиатуру, мышь, а также другие необходимые устройства и инструменты.

Вредный или опасный фактор на рабочем месте программиста — это любой фактор, который может негативно повлиять на здоровье и безопасность программиста при выполнении его профессиональных обязанностей. Это могут быть факторы, связанные с эргономикой рабочего места, напряжением глаз, плохим освещением, длительным сидением, повторяющимися движениями, электробезопасностью, а также факторы, связанные с психологическим стрессом и рабочей нагрузкой.

2.6.2 Меры безопасности для предотвращения вредного или опасного фактора на рабочем месте

Эргономические требования к рабочему месту программиста учитывают аспекты комфорта, безопасности и эффективности работы. Важно обеспечить правильную организацию рабочего места программиста с целью максимального

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 59 |

комфорта и снижения риска для предотвращения вредного или опасного фактора на рабочем месте [19].

Некоторые принципы эргономики включают (рисунок 42):

– используйте стул с регулируемой высотой и спинкой, чтобы поддерживать правильное положение спины и шеи. Рабочий стол также должен быть регулируемым по высоте, чтобы обеспечить удобное положение рук и предотвратить напряжение.

– программист должен сидеть прямо, с ногами на полу и спиной, опирающейся на спинку стула. Руки и предплечья должны быть параллельны полу, а локти согнуты под прямым углом. Клавиатура и мышь должны быть расположены на уровне предплечья.

– используйте подставку для запястий или эргономическую клавиатуру и мышь с поддержкой запястий, чтобы предотвратить напряжение и травмы.

– рекомендуется поддерживать оптимальное расстояние между глазами программиста и монитором компьютера. Рекомендуемое расстояние составляет около 50-70 см. Это помогает уменьшить напряжение глаз и предотвращает возникновение усталости;

– предоставьте достаточное пространство на рабочем столе для размещения необходимых устройств и документов. Убедитесь, что рукава программиста не зажаты и свободно двигаются.

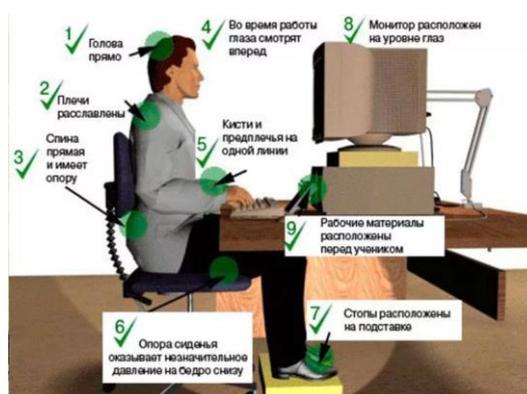


Рисунок 42 – Эргономические требования

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

РКРИПТ.09.02.0301.4205.000ПЗ

Лист

60

Параметры микроклимата также играют важную роль в обеспечении комфортных и безопасных условий на рабочем месте программиста.

Рабочее место программиста должно поддерживать оптимальную температуру. Рекомендуется, чтобы температура находилась в диапазоне 20-24°C. Это помогает предотвратить перегрев или переохлаждение организма, что может привести к дискомфорту и снижению производительности.

Влажность воздуха: Оптимальный уровень влажности воздуха на рабочем месте составляет около 40-60%. Низкая влажность может вызвать сухость глаз и кожи, а высокая влажность может способствовать развитию плесени и вызывать дискомфорт. Регулярная вентиляция помогает поддерживать подходящий уровень влажности.

Циркуляция воздуха: Хорошая циркуляция свежего воздуха на рабочем месте программиста является важным параметром микроклимата. Это может быть достигнуто через регулярную вентиляцию помещения или использование системы кондиционирования воздуха.

2.6.3 Способы защиты от опасных и вредных факторов

Для защиты от опасных и вредных факторов необходимо:

- использовать специальные программы для защиты от вредного воздействия электромагнитных излучений и снижения уровня шума на рабочем месте;
- обеспечить правильное хранение и использование электронного оборудования, чтобы избежать риска поражения электрическим током;
- обеспечить оптимальные условия освещения рабочего места, что снизит напряжение глаз и уменьшит вероятность возникновения головных болей и проблем со зрением;

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 61 |

- использовать удобную и эргономичную мебель, которая обеспечивает правильную позу тела и уменьшает нагрузку на позвоночник и мышцы;
- регулярно проводить перерывы и зарядку, чтобы уменьшить усталость и напряжение мышц;
- использовать специальные подставки и коврики для мыши и клавиатуры, чтобы снизить риск развития синдрома карпального канала.

2.6.4 Выводы по разделу «Охрана труда»

Эргономические требования к рабочему месту программиста для охраны труда являются неотъемлемой частью процесса создания программного продукта.

Они направлены на обеспечение безопасности труда пользователей программного продукта и снижение рисков для их здоровья. Для этого необходимо обеспечить оптимальные условия освещения, использовать эргономичную мебель, проводить перерывы и зарядку, использовать специальные программы и средства защиты от опасных и вредных факторов.

Это позволит снизить вероятность возникновения заболеваний опорно-двигательной системы, глаз и нервной системы у пользователей программного продукта, а также повысит эффективность работы программиста.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 62 |

3 Экономическая часть

3.1 Определение трудоемкости разработки проекта

На разработку WEB-приложения тестирования по английскому языку для ГБПОУ РО «РКРИПТ» потребовалось 24 рабочих дня по 6 часов, что составляет 144 часа. Распределение времени на разработку представлено в таблице 8.

Таблица 8 – Распределение времени на разработку WEB-приложения тестирования по английскому языку для ГБПОУ РО «РКРИПТ»

| Виды работ (перечислить свои виды работ) | Трудоемкость, н.час. (количество часов) |
|---|--|
| 1. Анализ предметной области | 6 |
| 2. Сбор теоретических материалов | 10 |
| 3. Проектирование сайта (разработка технического задания) | 12 |
| 4. Проектирование структуры сайта | 16 |
| 5. Проектирование макета сайта | 6 |
| 6. Подбор контента | 7 |
| 7. Создание макетов страниц | 14 |
| 8. Разработка программной логики | 22 |
| 9. Создание программных модулей | 35 |
| 10. Тестирование | 10 |
| 11. Создание пользовательской документации | 6 |
| ВСЕГО: | 144 |

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

3.2 Определение величины затрат на создание

Величину затрат на создание WEB-приложения, можно определить на основе метода калькуляции себестоимости. В этом случае затраты определяются расчетом по отдельным статьям расходов и их последующим суммированием.

Затраты в свою очередь делятся на две категории:

- постоянные издержки;
- переменные издержки.

Рассмотрим постоянные издержки. В данную категорию издержек при создании интернет-магазина входят:

- амортизационные отчисления на вычислительную технику.

Подчитаем амортизационные отчисления.

Сумма годовых амортизационных отчислений определяется по формуле (1):

$$A_{\Gamma} = C_{\Pi} \times N_{\text{ам}}, \quad (1)$$

где A_{Γ} – сумма годовых амортизационных отчислений, руб;

C_{Π} – балансовая стоимость компьютера, руб./шт.;

$N_{\text{ам}}$ – норма амортизации, %.

$$A_{\Gamma} = 80000 \times 0,33 = 26400 \text{ руб.}$$

Сумма амортизационных отчислений за период создания интернет-магазина будет равняться произведению амортизационных отчислений в день на количество дней эксплуатации компьютера и программного обеспечения при создании интернет-магазина, определяется по формуле (2):

$$A_{\text{ЭВМ}} = (C_{\text{п}} \times N_{\text{ам}}) \times n / 365, \quad (2)$$

где $A_{\text{ЭВМ}}$ – сумма амортизационных отчислений за период создания интернет-магазина дней, руб.; n – количество рабочих дней (24 дня).

$$A_{\text{ЭВМ}} = (80000 \times 0,33) \times 24 / 365 = 2393,42 \text{ руб.}$$

Амортизационные отчисления на компьютер и программное обеспечение производятся ускоренным методом. Норма амортизации на компьютеры равна 33%, на программное обеспечение равна 25%. Первоначальная стоимость компьютера включает отпускную цену, расходы на транспортировку, монтаж оборудования и его наладку и вычисляется по формуле (3):

$$C_{\text{п}} = C_{\text{р.ц.}} + Z_{\text{уст}}, \quad (3)$$

где $C_{\text{п}}$ – первоначальная стоимость ПЭВМ, руб.;

$C_{\text{р.ц.}}$ – рыночная стоимость компьютера, руб./шт.;

$Z_{\text{уст}}$ – затраты на доставку и установку компьютера, %.

Компьютер, на котором велась работа, был приобретен до создания интернет-магазина по цене 80000 руб., затраты на установку и наладку были включены в стоимости компьютера.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 65 |

Следовательно, $C_{п} = 80000 \times 1 = 80000$ руб./шт.

Программное обеспечение было также приобретено до создания веб сайта, общая цена составила 12000 руб. На программное обеспечение производятся, как и на компьютеры, амортизационные отчисления.

$$A_{по} = (10000 \times 0,25) \times 24 / 365 = 164,38 \text{ руб.}$$

Общая амортизация за время эксплуатации компьютера и программного обеспечения при проектировании интернет-магазина вычисляется по формуле (4):

$$A_{об} = A_{эвм} + A_{по}, \quad (4)$$

где $A_{эвм}$ – амортизационные отчисления на компьютер за время его эксплуатации;

$A_{по}$ – амортизационные отчисления на программное обеспечение за время его эксплуатации.

$$A_{об} = 2393,42 + 164,38 = 2557,80 \text{ руб.}$$

Рассмотрим переменные издержки. В данную категорию издержек при создании веб-сайта входят:

- затраты на приобретение материалов;
- затраты на потребляемую электроэнергию;

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 66 |

Рассчитаем затраты на приобретение материалов. В стоимость материальных затрат включаются транспортные расходы (5% от стоимости материалов). Расчет статьи «Материалы» приводится в таблице 9.

Таблица 9– Материалы при разработке WEB-приложения

| Наименование | Единица измерения | Количество | Цена за единицу, руб. | Стоимость, руб. |
|---------------------|-------------------|------------|-----------------------|-----------------|
| Домен | | 1 | 300 | 300 |
| Хостинг | | 1 | 197 | 197 |
| Бумага для принтера | пачка | 1 | 400 | 400 |
| Тонер для принтера | шт. | 1 | 500 | 500 |
| Итого: | – | – | 1397 | 1397- |

Необходимо рассчитать затраты на материалы и покупные изделия с учетом транспортных расходов.

$$З_m = 1397 \times 1,05 = 1466,85 \text{ руб.}$$

Рассчитаем затраты на потребляемую электроэнергию.

Рабочий день обучающегося равен шести часам, следовательно, стоимость электроэнергии за период работы компьютера во время создания веб сайта будет вычисляться по формуле (5):

$$З_{эл. пер} = P_{эвм} \times n \times 6 \times C_{эл}, \quad (5)$$

где n – время эксплуатации компьютера при создании интернет–магазина, дней.

$$Z_{\text{эл. пер}} = 0,4 \times 24 \times 6 \times 7 = 403,2 \text{ руб.}$$

Согласно техническому паспорту компьютера $P_{\text{эвм}} = 0,4$ кВт, а стоимость одного кВт/ч электроэнергии на предприятии $C_{\text{эл}} = 7$ руб.

$$\text{Общие затраты: } Z = 2557,8 + 1466,85 + 403,2 = 4428 \text{ руб.}$$

3.3 Определение себестоимости разработанного проекта

WEB-приложение – высокоэффективный и малозатратный инструмент для успешного ведения тестирования по английскому языку для ГБПОУ РО «РКРИПТ».

Рассмотрим, за какое время окупится веб-приложение. Доход - денежные средства или материальные ценности, полученные государством, физическим или юридическим лицом в результате какой-либо деятельности за определённый период времени. Например, по статистике в среднем на сайт будет заходить в месяц около 1000 студентов. Умножив на 1%. Доход с одного студента, после вычета налогов и накладных расходов, составит 10000 руб. В месяц при использовании веб-приложения получим (6):

$$10000 \times 10 = 100000 \text{ руб. дохода.} \quad (6)$$

Как было показано, на создание веб-сайта было потрачено 4428 руб. ($Z = Z_{\text{пост.}} + Z_{\text{пер.}}$), то можно подсчитать, что сайт окупится через 0,04 месяца (затраты/доход).

Таким образом, можно заключить, что предложенные выше мероприятия экономически эффективны и целесообразны для WEB-приложения.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 68 |

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы (ВКР) был разработан проект WEB-приложение тестирование по английскому языку для ГБПОУ РО «РКРИПТ» (<https://online-quiz.ru/>), позволяющий:

- проходить студентам тестирование по английскому языку;
- учителям отслеживать и анализировать успеваемость студентов;
- осуществлять выгрузку результатов тестирования в разные форматы xls, csv;
- осуществлять создание, изменение, удаление и редактирование тестов администратором сайта.

В процессе выполнения ВКР были выполнены следующие виды работ:

- проанализирована предметная область;
- разработана схема пользовательского интерфейса;
- разработана структура базы данных;
- спроектирован пользовательский интерфейс;
- спроектирована база данных;
- реализовано WEB-приложение;
- подключена защита данных;
- выполнена отладка и тестирование WEB-приложения;
- составлена программная документация.

Разработанное WEB-приложение устойчиво выполняет все свои функции:

- предоставлен доступ к регистрации и авторизации;
- прохождение тестирования;
- выгрузка данных;
- редактирование, изменение и удаление данных;
- добавление учителей администратором сайта;

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 69 |

Результаты выполнения ВКР могут быть использованы ГБПОУ РО «РКРИПТ» для проведения тестирования по английскому языку.

Выполненная работа представляет собой полноценное WEB-приложение тестирование по английскому языку, которое эффективно поддерживает образовательный процесс и позволяет студентам проходить тестирование по английскому языку, а учителям управлять и анализировать результаты студентов.

Дальнейшее развитие данного проекта может включать расширение функциональности, улучшение интерфейса и адаптацию под потребности других образовательных учреждений.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 70 |

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Федорова Г.Н. Участие в интеграции программных модулей: учеб. пособие для студ. учреждений сред. проф. образования / Г.Н. Федорова. –М.: Издательский центр «Академия», 2020. – 304 с. – ЭБС Знаниум: <https://znanium.com/>

2 Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник для студ. учреждений сред. проф. образования / Г.Н. Федорова. – 4-е изд., перераб. – М.: Издательский центр «Академия», 2020. – 384 с.

3 Федорова Г.Н. Основы проектирования баз данных: учебник для студ. учреждений сред. проф. образования / Г.Н. Федорова. – 4-е изд., перераб. – М.: Издательский центр «Академия», 2020. – 224 с.

4 Рудаков А.В. Технология разработки программных продуктов: учебник для студ. учреждений сред. проф. образования / А.В. Рудаков. – 12-е изд., стер. – М.: Издательский центр «Академия», 2018. – 208 с.

5 Федорова Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: Учебное пособие для СПО.- М.: КУРС, 2020.- 333с. - ЭБС Знаниум: <https://znanium.com/> Гниденко И.Г. Технология разработки программного обеспечения: учебное пособие для СПО / И.Г. Гниденко, Ф.Ф. Павлов, Д.Ю. Федоров.- М.: Юрайт-М, 2019. – 235 с.– ЭБС Юрайт: <https://urait.ru/ebs>

6 Голицына О.Л. Основы проектирования баз данных: учебное пособие для СПО/ О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 2-е изд., перераб. и доп. — Москва: ФОРУМ: ИНФРА-М, 2021. — 416 с.

7 Ляпина О.П. Стандартизация, сертификация и техническое документооборот: учебник для СПО / О.П. Ляпина, О.Н. Перлова.- М.: Академия, 2020.- 208с.

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 71 |

8 Ананьева Т.Н. Стандартизация, сертификация и управление качеством программного обеспечения: Учебное пособие для СПО / Т.Н. Ананьева, Н.Г. Новикова, Т.Н. Исаев. - М.: Инфра-М, 2019. – 232 с.– ЭБС Знаниум: <https://znanium.com/>

9 Методология веб разработки. – 2018 [Электронный ресурс]. URL: <https://vc.ru/flood/42084-agile-ili-waterfall-sravnenie-metodologiy-veb-razrabotki#:~:text=Waterfall%20-%20это%20четко%20запланированный%20и,изменений%20на%20каждом%20этапе%20проекта.> [Дата обращения: 20.05.2023].

10 Тестирование – 2021 [Электронный ресурс]. URL: <https://scilead.ru/article/231-testirovanie-web-prilozhenij>

11 Методология проектирования. – 2023 [Электронный ресурс]. URL: <https://asana.com/ru/resources/project-management-methodologies> [Дата обращения: 20.05.2023].

12 Защита данных и WEB-приложения. – 2023 [Электронный ресурс]. URL: <https://gardatech.ru/articles/smi/informatsionnaya-bezopasnost-web-prilozheniy-sovremennye-resheniya/> [Дата обращения: 24.05.2023].

13 Agile – 2023 [Электронный ресурс]. URL: <https://cloud.yandex.ru/blog/posts/2022/10/agile-and-project-management>[Дата обращения: 26.05.2023].

14 Waterfall. – 2023 [Электронный ресурс]. URL: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.html [Дата обращения: 24.05.2023].

15 CMS – 2023 [Электронный ресурс]. URL: <https://help.reg.ru/support/hosting/cms/chto-takoye-cms.> [Дата обращения: 24.05.2023].

16 СУБД – 2021 [Электронный ресурс]. URL: <https://medium.com/nuances-of-programming/как-выбрать-субд-для-решения-ваших-задач-386d91f53dda> [Дата обращения: 24.05.2023].

17 Отладка – 2021 [Электронный ресурс]. URL: https://addphp.ru/materials/base/1_18.php [Дата обращения: 27.05.2023].

18 Логирование VS Code – 2023 [Электронный ресурс]. URL: <https://www.cloudways.com/blog/php-debug/> [Дата обращения: 27.05.2023].

19 Эргономические требования к рабочему месту – 2023 [Электронный ресурс]. URL: <https://habr.com/ru/companies/ozontech/articles/712928/> [Дата обращения: 27.05.2023].

20 Среда разработки VS Code – 2022 [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/vsyo-o-visual-studio-code/> [Дата обращения: 27.05.2023].

21 Методика тестирования WEB-приложений – 2022 [Электронный ресурс]. URL: <https://habr.com/ru/companies/ruvds/articles/676752/> [Дата обращения: 24.05.2023].

| | | | | | | |
|------|------|----------|---------|------|------------------------------|------|
| | | | | | РКРИПТ.09.02.0301.4205.000ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 73 |

ПРИЛОЖЕНИЕ А

(Обязательное)

Диаграмма вариантов использования

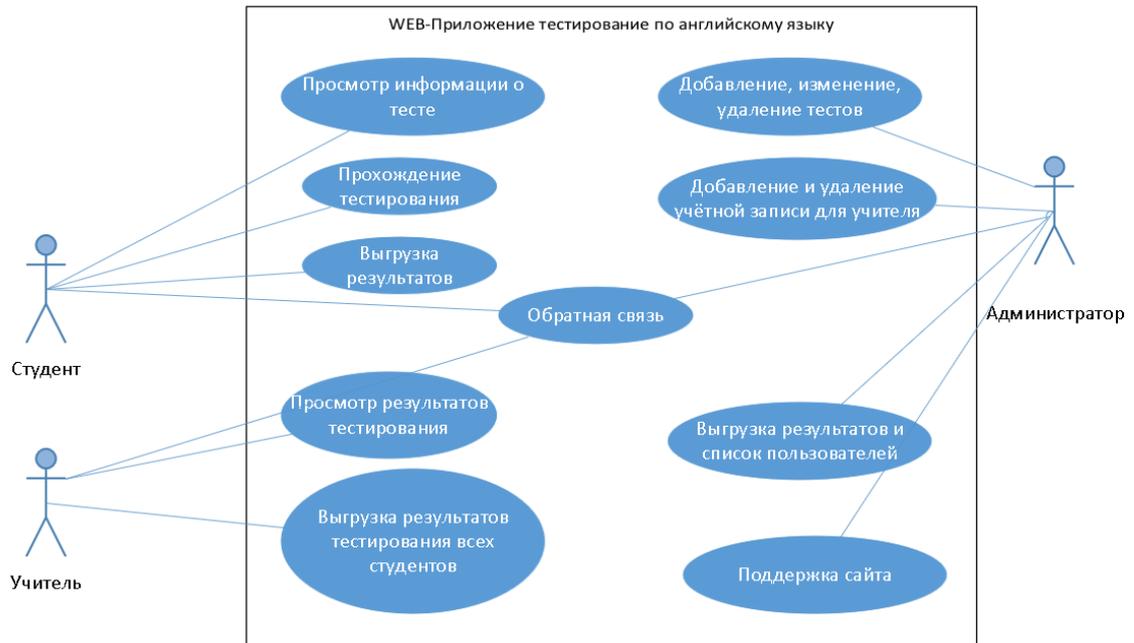


Рисунок А.1 – Диаграмма вариантов использования

| Изм. | Лист | № докум. | Подпись | Дата |
|------|------|----------|---------|------|
| | | | | |

ПРИЛОЖЕНИЕ Б

(Обязательное)

Диаграмма взаимодействия

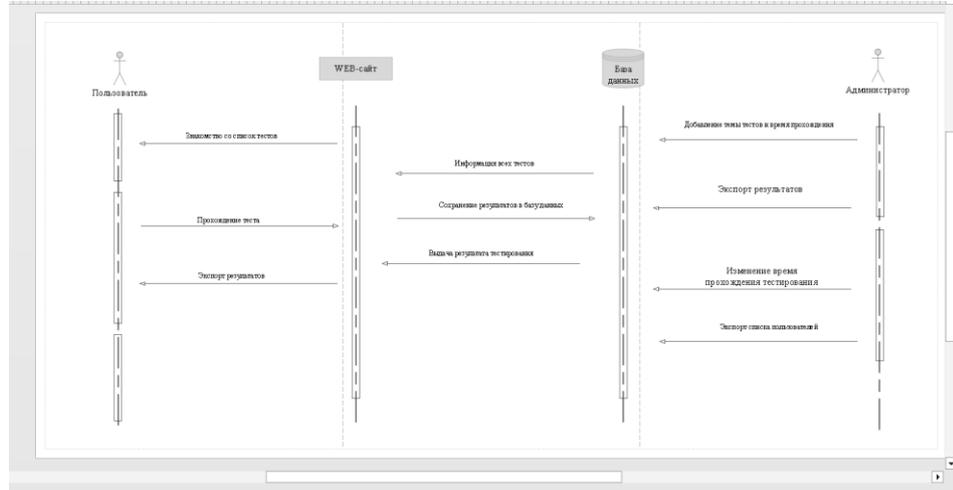


Рисунок Б.1 – Диаграмма взаимодействия

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

РКРИПТ.09.02.0301.4205.000ПЗ

Лист

75

ПРИЛОЖЕНИЕ В
(Обязательное)
ER-диаграмма

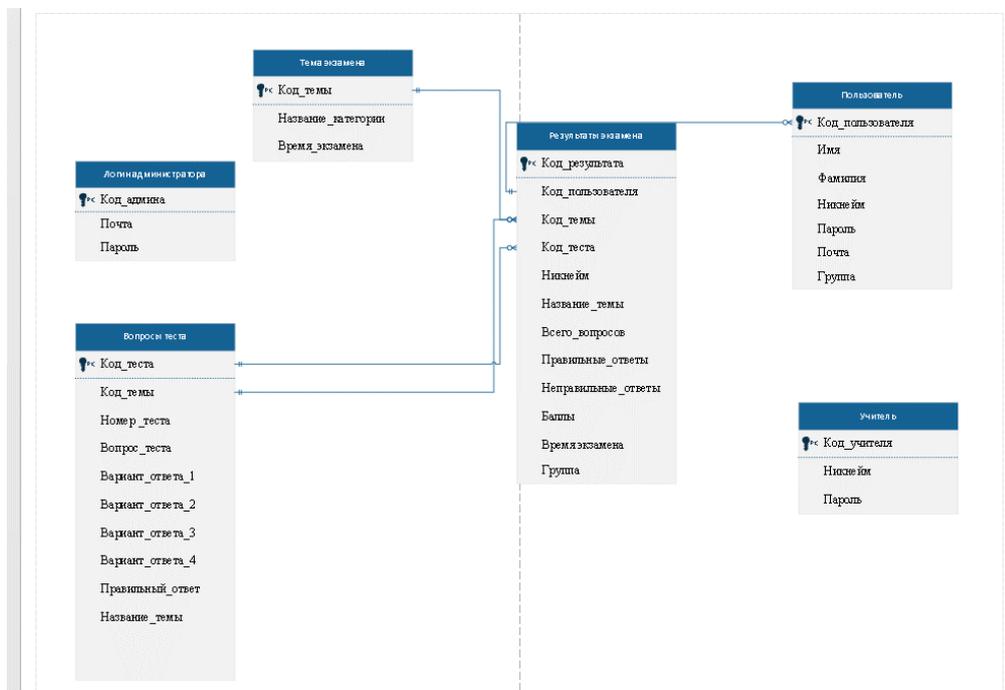


Рисунок В.1 – ER-диаграмма

| | | | | |
|------|------|----------|---------|------|
| Изм. | Лист | № докум. | Подпись | Дата |
|------|------|----------|---------|------|

ПРИЛОЖЕНИЕ Г

(Обязательное)

Диаграмма компонентов

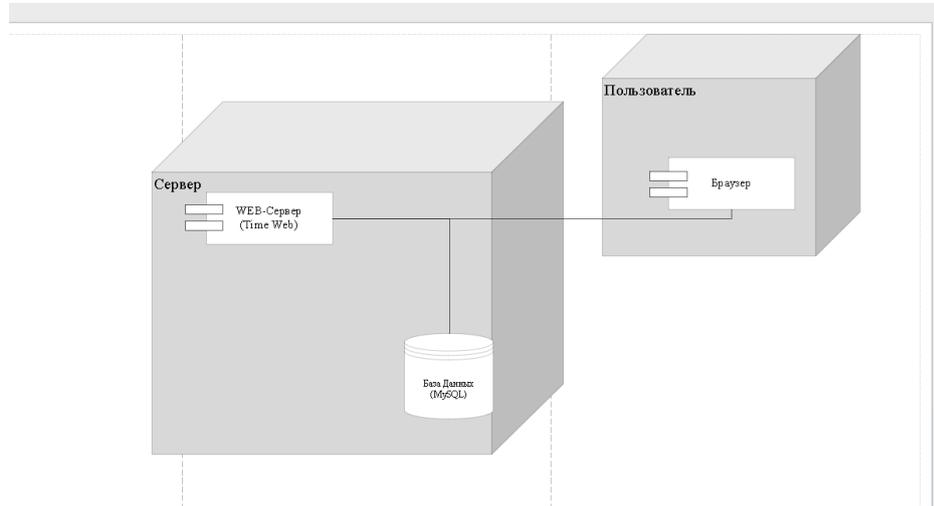


Рисунок Г.1 – Диаграмма компонентов

ПРИЛОЖЕНИЕ Д

(Обязательное)

Схема пользовательского интерфейса

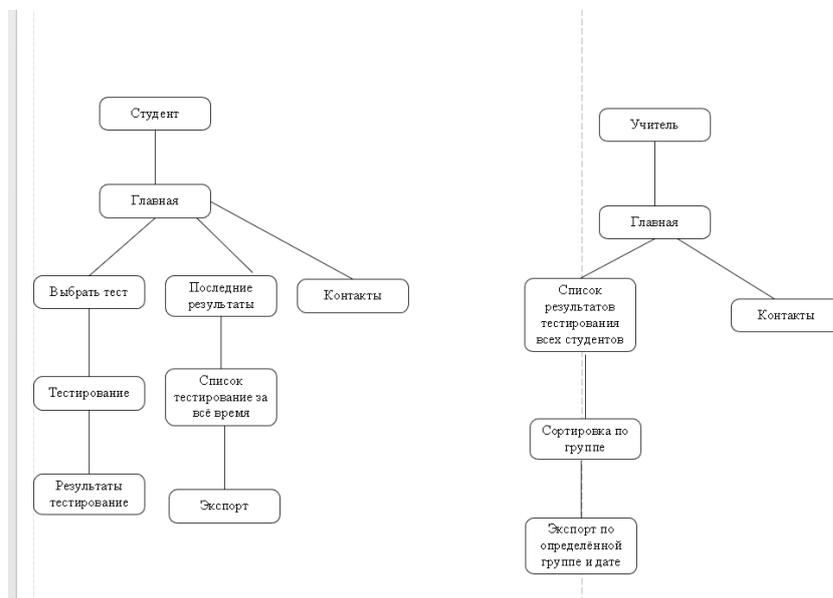


Рисунок Д.1 – Схема пользовательского интерфейса

| | | | | |
|------|------|----------|---------|------|
| | | | | |
| Изм. | Лист | № докум. | Подпись | Дата |

ПРИЛОЖЕНИЕ Е

(Обязательное)

Диаграмма развертывания

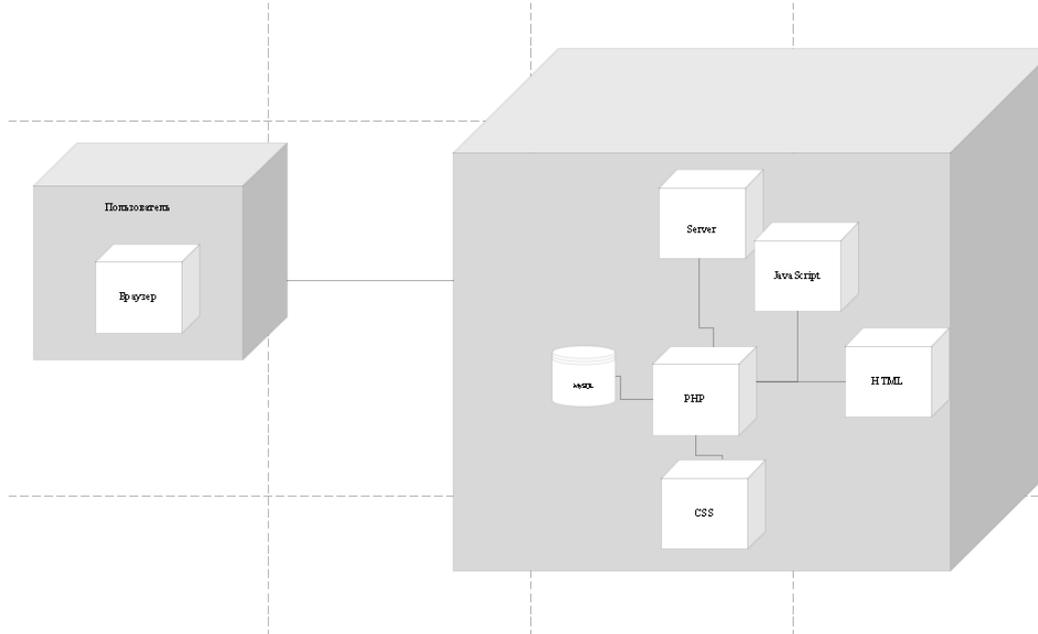


Рисунок Е.1 – Диаграмма взаимодействия

| Изм. | Лист | № докум. | Подпись | Дата |
|------|------|----------|---------|------|
| | | | | |

РКРИПТ.09.02.0301.4205.000ПЗ

Лист

79