МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет математики и компьютерных наук**

**Кафедра вычислительной математики и информатики**

**Курсовая работа**

**Разработка электронной системы тестирования знаний**

Работу выполнил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.В. Кочергин

(подпись)

Направление подготовки 02.04.01 Математика и компьютерные науки курс 1

Направленность (профиль)  Вычислительная математика

Научный руководитель

канд. тех. наук, доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.А. Кирий

(подпись, дата)

Нормоконтролер

преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Лахтина

(подпись, дата)

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc167659317)

[1 Глава 1 5](#_Toc167659318)

[1.1 Тестирование в сфере образования 5](#_Toc167659319)

[1.2 Тестирование в корпоративной среде 6](#_Toc167659320)

[1.3 Система онлайн-тестирования 7](#_Toc167659321)

[1.4 Анализ существующих решений 7](#_Toc167659322)

[2 Глава 2 11](#_Toc167659323)

[2.1 Технологический стек 11](#_Toc167659324)

[Обоснование выбора данных технологий 11](#_Toc167659325)

[2.2 Фронтенд 12](#_Toc167659326)

[2.2.1 Описание архитектуры веб-приложения 12](#_Toc167659327)

[2.2.2 Краткое пояснение структуры кода и жизненного цикла в пределах основного кода приложения 12](#_Toc167659328)

[2.2.3 Подробнее о реализации рассылки результатов на почту тестируемого 18](#_Toc167659329)

[2.3 Бэкенд 22](#_Toc167659330)

[2.4 Функциональность приложения 23](#_Toc167659331)

[2.5 Дальнейшее развитие 27](#_Toc167659332)

[ЗАКЛЮЧЕНИЕ 28](#_Toc167659333)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc167659334)

[ПРИЛОЖЕНИЕ А 30](#_Toc167659335)

Введение

Процесс тестирования знаний сотрудников или учащихся во многом может быть улучшен за счет разработки автоматизированной системы тестирования. Данная работа будет описывать особенности разработки системы подобного типа, используемые технологии, а также анализирует современные подходы к разработке веб-сайтов и аналитических систем. Объектом исследования, лежащим в основе данной работы, является автоматизация процесса тестирования, а его предметом – разработка системы онлайн-тестирования знаний.

Задача создания такой системы предполагает наличие веб-сайта, который служит местом для создания и проведения тестов. Данные проведённых тестирований анализа копируются из реляционной БД сайта, которая содержит таблицы для хранения информации о тестах, их вопросах, вариантах ответа, ответов конкретного участника тестирования и т.д.

Задача составления и проведения теста всегда связана с некоторыми затратами, выраженными в материальном или временном выражении. Проведение тестов с печатными бланками связано как с материальными, так и с временными издержками, такими как бумага, чернила принтера, время поиска аудитории, время самого экзамена. Оценка результатов таких тестов также связана с временными издержками по сбору бланков и подсчету результатов. При этом время, затраченное на подведение итогов, растет вместе с числом участников теста. Для учителей, менеджеров и других людей, которые занимаются проведением различных тестов регулярно, минимизация потраченного на тестирование времени и ресурсов может оказаться критически важной.

Использование онлайн-системы тестирования позволит ускорить процесс создания типизированных тестов, осуществить их удаленное проведение и провести анализ их результатов с помощью персонального компьютера. Учитывая распространённость сети Интернет среди учащихся и сотрудников учебных и других организаций, можно сказать, что такая система станет доступна для многих. Разрабатываемая онлайн-система сможет стать решением задачи сокращения общего времени проведения тестов и увеличения качества аналитической оценки их результатов.

Разрабатываемая система должна минимизировать время на создание теста и анализ результатов за счет использования современных технологий в сфере веб-разработки. Выбор архитектуры системы и конкретного набора технологий должен быть осуществлен исходя из требования к максимальной скорости работы системы при взаимодействии с ней пользователей. В перспективе аналитическая составляющая системы также должна позволять ответить на наиболее востребованные вопросы пользователей при работе с результатами тестов.

Цель данной работы – разработать систему онлайн-тестирования с высокой скоростью работы и удобством использования. Вся работа может быть разделена на несколько последовательных задач:

• изучение подходов к тестированию в сфере образования и корпоративного среде

• изучение подходов к созданию универсальных онлайн-тестов

• разработка сценария использования проектируемой системы

• выявление требований к проектируемой онлайн-системе

• анализ существующих систем онлайн-тестирования

• разработка веб-сайта

• тестирование системы и исправление возможных ошибок

В первой главе данной работы дано описание современным подходам к разработке веб-приложений, предпосылки к проведению тестирования в корпоративной среде и сфере образования. В этой главе так же произведен анализ существующих решений для проведения тестирования.

Вторая глава содержит описание используемых технологий, а также описание функциональности реализуемого решения.

1 Глава 1

Проведение тестирования с использованием бумажных тестов давно является одним из приоритетных способов оценки успеваемости учеников или эффективности сотрудников в учебных заведениях и частных компаниях. В школах и университетах тесты могут быть использованы для промежуточного контроля, а также для оценки поведения, психики, профориентации или мотиваций учащихся. Некоторые государственные экзамены и олимпиады включают в себя тестовую часть с выбором ответа из нескольких вариантов. Проведение таких тестов связано с различными издержками, в первую очередь со временем на проверку, поиск места проведения, оповещения учащихся.

Разрабатываемая система онлайн-тестирования создается с целью совмещения процессов создания, проведения и анализа результатов тестов, тем самым повышая эффективность самого процесса тестирования.

1.1 Тестирование в сфере образования

В России тестирование занимает срединное положение – оно используется одинаково часто как для оценки промежуточных результатов, так и для финальных экзаменов. Выпускные экзамены в школах после 9 и 11 классов (ОГЭ и ЕГЭ) частично основаны на тестах с выбором ответа из нескольких вариантов. Доля заданий подобного рода в контрольно-измерительных материалах (КИМ) зависит от сдаваемого предмета. Так, на ЕГЭ 2013 года в вариантах по математике не было ни одного задания с выбором ответа, а по биологии – не менее трети от общего числа. Использование тестов для промежуточного контроля в российских школах никак не регламентируется, хотя учителя все равно используют тесты по своей инициативе. Так, широко используются тесты по истории, географии, обществоведению и иностранным языкам.

В целом, тесты в образовании обладают следующими характеристиками:

• Разделение на секции

• Ограниченность по времени

• Различные типы вопросов

• Различные типы вариантов ответа

К наиболее распространенным типам вопросов относятся:

• Текстовые вопросы

• Вопросы с картинками

• Заполнение пропусков в тексте

Наиболее популярные типы ответов к таким вопросам являются:

• Единственный ответ из числа перечисленных

• Множественный выбор из числа перечисленных

• Написание текста

• Сопоставление

Проектируемая в рамках работы система должна иметь большинство из указанных распространенных типов ответов и вопросов, а также позволить проводить аналитику результатов.

1.2 Тестирование в корпоративной среде

Использование тестов в различных компаниях производится с целью оценки профессиональной подготовки сотрудников или выявления их скрытых талантов. В крупных компаниях процесс тестирования зачастую осуществляется сотрудниками ассессмент-центров (центров оценки). Процесс тестирования в таких центрах включает в себя как письменные тесты, так и упражнения в группах.

Разрабатываемая система позволит облегчить создание тестов для корпоративного использования в ассессмент-центрах. Она станет доступной многим, так как веб-сайт, служащий главным интерфейсом взаимодействия с создателями теста и тестируемыми, может быть открыт на многих устройствах, включая персональные компьютеры, смартфоны и планшеты.

1.3 Система онлайн-тестирования

Под системой онлайн-тестирования подразумевается веб-сайт, который предоставляет его посетителям различные функции для успешного осуществления процесса тестирования. Этот сайт, как правило, имеет несколько страниц с динамическим содержимым, хотя может содержать и статичные информационные страницы. Осуществление взаимодействия с пользователями производится за счет веб-страниц с различными элементами управления, такими как ссылки, поля для ввода, кнопки, веб-формы и другие.

Система тестирования с аналитическими возможностями должна обладать следующим набором функций:

• Регистрация и авторизация пользователей (не было реализовано в MVP версии)

• Просмотр, создание, редактирование тестов различных типов

• Удаленное проведение тестирования участников

• Просмотр и анализ результатов

Разрабатываемая в рамках работы система должна соответствовать указанному выше описанию.

1.4 Анализ существующих решений

Задача оптимизации времени, затраченного на тестирование уже была осуществлена некоторыми командами разработчиков. В сети Интернет присутствуют веб-сайты, схожие по замыслу и решаемым задачам с проектируемым приложением.

*QuizWorks Online Exam Builder*

Данный англоязычный веб-сервис предоставляет пользователям возможности по созданию онлайн-экзаменов различных типов. К типам относятся:

• Экзамен (Exam)

• Викторина (Quiz)

• Лабораторная работа (Assessment)

Интерфейс создания тестов на этом веб-сайте позволяет добавлять неограниченное количество вопросов. Разбиение вопросов на категории (секции) не допускается. При этом один вопрос может иметь не более четырех вариантов ответа и единственный верный вариант. Функционал сайта обеспечивает поддержку подсказок и пояснений к вопросам во время прохождения теста.

Возможности по аналитике ограничиваются следующими отчетами:

• Статистика по тесту

• Статистика по вопросу

• Итоговые результаты

Архитектура клиентской части приложения основана на статичной отрисовке элементов на стороне веб-сервера. Это означает, что веб-сайт работает дольше одностраничных сайтов за счет траты времени на обновление страницы в веб-браузере.

Адрес веб-сайта: <https://www.onlineexambuilder.com/>

*ClassMaker*

Эта англоязычная система онлайн-тестирования позволяет создавать тесты и наполнять их интерактивным содержимым. Данный сайт также предоставляет широкие возможности по аналитике результатов тестов.

Интерфейс создания тестов интуитивен для пользователя, а набор возможных типов вопросов для добавления довольно велик. Система поддерживает следующие типы вопросов:

• С несколькими вариантами ответа

• «Правда или ложь»

• Сопоставление

• Короткий ответ

• Эссе

• Заполнение пропусков

• Аудиовопрос

Допускается разбиение вопросов на категории, добавление подписка предопределенных групп тестируемых людей на приватные тесты и создание публичных тестов. Пользователи с премиум-подпиской могут также выдавать электронные сертификаты участникам теста, подтверждающие их результаты.

Аналитическая составляющая системы обладает следующими функциями:

• Вывод результатов

• Ранжирование по шкале оценок

• Отправка персонализированных E-Mail с анализом работы

Адрес веб-сайта: <https://www.classmarker.com/>

*MasterTest*

Данная система разработана группой российских разработчиков и направлена на распространение в коммерческих образовательных учреждениях. Каждому пользователю системы предоставляется возможность создавать учебные программы по различным предметам и настраивать периодические экзамены для проверки текущей результативности учеников на основе онлайн-тестов. Анализ результатов тесты возможна за счет использования стороннего ПО.

Веб-сайт позволяет автоматизировать процесс обучения за счет интерактивной веб-платформы для создания протяженных во времени учебных программ с тестами. Функциональность системы поддерживает:

• Создание тестов

• Подписка на тесты по E-Mail и SMS

• Вывод данных в формате электронной таблицы

• Выставление оценок

• Генерация сертификатов

Система хорошо интегрируется со сторонним ПО для анализа данных, таким как MS Excel. С помощью специального инструмента в интерфейсе веб-сайта можно выгрузить информацию о результатах конкретного теста в текстовом формате.

Ссылка на сайт: <http://master-test.net/>

2 Глава 2

2.1 Технологический стек

Веб-приложение для тестирования знаний разработано с использованием HTML, CSS, JavaScript и PHP. HTML (HyperText Markup Language) является основным языком разметки для создания структуры веб-страниц. CSS (Cascading Style Sheets) применяется для оформления внешнего вида веб-страниц, включая стилизацию элементов и макет. JavaScript является клиентским языком программирования, который обеспечивает интерактивное поведение веб-приложения, такое как обработка событий пользователя и динамическое изменение содержимого страницы. PHP (Hypertext Preprocessor) используется для обработки серверных запросов, включая будущее взаимодействие с базой данных и выполнение бизнес-логики приложения.

Обоснование выбора данных технологий

Использование чистого JavaScript без фреймворков было выбрано с целью обеспечения гибкости и простоты разработки в условиях сжатых сроков. Это позволяет избежать зависимостей от сторонних библиотек и фреймворков, что упрощает поддержку и развертывание приложения. Кроме того, использование чистого JavaScript дает возможность полностью контролировать функциональность приложения и оптимизировать его производительность. Что касается выбора PHP, этот язык программирования широко распространен в веб-разработке и хорошо подходит для создания динамических веб-страниц и взаимодействия с базой данных. Его простота в изучении и широкая поддержка делают его привлекательным выбором для серверной части веб-приложения. Стоит сразу отметить тот факт, что реализуемое решение сразу же нашло своё применение в прикладной деятельности автора работы как преподавателя АНПОО «Краснодарский Институт Профессионального образования» для использования его на занятиях. Более того, основной модуль приложения был развернут и использовался в дипломной работе одного из студентов Факультета Математики и Компьютерных наук, что сразу же утвердило правильность выбранных технологий с отсутствием фреймворков, разработка с использованием которых заняла бы у автора значительно больше времени от начала до минимально работающей версии (MVP) приложения.

2.2 Фронтенд

2.2.1 Описание архитектуры веб-приложения

Веб-приложение для тестирования знаний построено на архитектурном шаблоне MVC (Model-View-Controller). HTML и CSS формируют представление (View), определяющее структуру и внешний вид веб-страниц. JavaScript отвечает за контроллер (Controller), обрабатывая события пользователя и управляя логикой взаимодействия с элементами. Такое разделение ответственности позволяет создать модульную и легко поддерживаемую структуру приложения.

JavaScript играет ключевую роль в клиентской части приложения, обеспечивая динамические возможности и взаимодействие пользователя с интерфейсом. Он используется для реализации различных функциональных элементов, таких как валидация форм, динамическое обновление содержимого страницы без перезагрузки, анимации и т.д.. Применение JavaScript позволяет создать интерактивное и отзывчивое веб-приложение, улучшая пользовательский опыт и функциональность приложения в целом.

2.2.2 Краткое пояснение структуры кода и жизненного цикла в пределах основного кода приложения

Я использовал в своём коде подход, который можно назвать объектно-ориентированным программированием (ООП). ООП - это парадигма программирования, которая фокусируется на создании объектов и управлении их взаимодействием для решения конкретной задачи. В данном коде используется несколько классов, каждый из которых представляет собой объект с определенными свойствами и методами, что является ключевой характеристикой ООП.

ООП был выбран в качестве подходящего подхода по нескольким причинам:

Модульность и переиспользуемость кода: ООП позволяет разбить код на отдельные классы, каждый из которых отвечает за определенную функциональность. Это делает код более модульным и переиспользуемым. Например, класс UI отвечает за управление интерфейсом пользователя, класс Game управляет состоянием игры, а класс QuizService занимается загрузкой и проверкой вопросов. Каждый класс имеет свою собственную логику и может быть использован в других проектах или расширен в будущем.

Инкапсуляция и абстракция: ООП позволяет инкапсулировать данные и методы в классах, скрывая детали реализации от других частей кода. Это обеспечивает абстракцию и упрощает понимание и использование классов. Например, класс Timer инкапсулирует логику таймера, скрывая внутреннюю реализацию от других классов.

Структурирование и организация кода: ООП позволяет структурировать код в логические блоки, что облегчает его чтение, понимание и обслуживание. Каждый класс имеет свою собственную роль и ответственность, что делает код более организованным и управляемым.

Разделение обязанностей: ООП поощряет разделение обязанностей между классами, что улучшает читаемость и поддерживаемость кода. Каждый класс фокусируется на конкретной задаче, что упрощает понимание его цели и поведения.

Теперь подробно рассмотрим назначение классов и жизненный цикл элементов в коде, который приведен в виде ссылки на репозиторий GitHub в ПРИЛОЖЕНИИ А:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1 – Структура классов в коде основного модуля

Архитектура приложения основана на использовании нескольких классов:

UI: Этот класс отвечает за управление интерфейсом пользователя. Он содержит методы для отображения и скрытия элементов интерфейса, переключения между экранами, управления таймером и обработки пользовательского ввода.

Game: Этот класс управляет состоянием игры. Он хранит информацию о текущем индексе вопроса, категории викторины, набранных баллах и ответах пользователя.

Controller: Этот класс является контроллером приложения. Он управляет взаимодействием между классами UI и Game, обрабатывает события пользователя и управляет логикой викторины.

QuizService: Этот класс отвечает за загрузку вопросов викторины, их случайный выбор и проверку ответов пользователя.

Timer: Этот класс управляет таймером викторины.

SwiperHandler: Этот класс управляет слайдером Swiper, который используется для отображения результатов викторины.

**Жизненный цикл приложения:**

При запуске приложения создаются экземпляры классов UI, Game, Controller, QuizService и Timer.

Класс Controller загружает вопросы викторины с помощью QuizService и инициализирует игру.

Пользователь нажимает кнопку "Начать", что запускает таймер и отображает первый вопрос.

Пользователь отвечает на вопросы, нажимая на варианты ответов или вводя текст.

После каждого ответа класс Controller отправляет ответ на проверку в QuizService и обновляет состояние игры в классе Game.

Когда пользователь отвечает на все вопросы, таймер останавливается, и отображаются результаты викторины.

Пользователь может перезапустить викторину или закрыть приложение.

Каждый класс имеет определенную роль в приложении:

UI: Класс UI управляет отображением элементов интерфейса, таких как вопросы, ответы, таймер и результаты. Он также обрабатывает события пользователя, такие как нажатие кнопок и ввод текста.

Game: Класс Game хранит состояние игры, включая текущий вопрос, категорию викторины, набранные баллы и ответы пользователя.

Controller: Класс Controller координирует взаимодействие между классами UI и Game. Он обрабатывает события пользователя, запускает таймер, отправляет ответы на проверку и обновляет состояние игры.

QuizService: Класс QuizService отвечает за загрузку вопросов викторины, их случайный выбор и проверку ответов пользователя. Он обеспечивает логику для проверки правильности ответов.

Timer: Класс Timer управляет таймером викторины. Он запускает и останавливает таймер, а также обновляет индикатор времени на экране.

SwiperHandler: Класс SwiperHandler управляет слайдером Swiper из библотеки swiper js, который используется для отображения результатов викторины. Он инициализирует и уничтожает слайдер в зависимости от состояния приложения.

Модуль для работы с результатами теста представлен следующей структурой:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2 – Структура кода модуля для работы с результатами

Этот код является частью веб-приложения викторины и отвечает за обработку результатов викторины и отправку их по электронной почте. Функция processingResultsThenSendMail используется для сбора из элементов html страницы и форматирования результатов викторины, а затем отправки их по электронной почте с помощью сервиса emailjs.

Код начинает с получения результатов викторины из локального хранилища (localStorage). Результаты включают в себя общее количество правильных ответов, общее количество вопросов и подробную информацию о каждом ответе пользователя.

Форматирование результатов: Код обрабатывает результаты викторины и форматирует их в удобочитаемый формат. Он создает таблицу, в которой отображаются вопросы, тип вопроса, ответы пользователя и правильные ответы. Цвет фона строки в таблице зависит от того, был ли ответ пользователя правильным или неправильным.

Получение дополнительной информации: Код также получает дополнительную информацию, такую как имя пользователя, контактный адрес электронной почты, дату прохождения викторины и время начала и окончания викторины. Эта информация используется для персонализации сообщения электронной почты.

Отправка сообщения электронной почты: Код использует библиотеку emailjs для отправки сообщения электронной почты с результатами викторины. Он создает объект params, который содержит всю необходимую информацию, включая имя, адрес электронной почты, тему и сообщение. Затем он использует функцию emailjs.send для отправки сообщения.

**Подход, используемый в этом коде:**

Структурирование данных: Код структурирует результаты викторины в виде таблицы, что облегчает чтение и понимание результатов. Он использует HTML-разметку для форматирования таблицы и добавления стилей, таких как цвет фона и текст.

Обработка данных: Код обрабатывает результаты викторины, извлекая необходимую информацию и преобразуя ее в удобочитаемый формат. Он также выполняет некоторые вычисления, такие как вычисление процентного соотношения правильных ответов.

Интеграция с электронной почтой: Код интегрируется с сервисом emailjs для отправки сообщения электронной почты. Он использует объект params для передачи информации в функцию emailjs.send.

Использование локального хранилища: Код использует локальное хранилище (localStorage) для хранения и получения результатов викторины. Это позволяет сохранять результаты викторины между сессиями и получать их при необходимости.

В целом, этот код является важной частью веб-приложения викторины, поскольку он отвечает за обработку результатов викторины, их форматирование и отправку по электронной почте. Он обеспечивает персонализированную обратную связь пользователям, позволяя им видеть свои результаты и анализировать свои ответы. Этот подход позволяет структурировать и представлять результаты викторины в удобочитаемом формате, а также интегрировать их с электронной почтой для удобства пользователей.

2.2.3 Подробнее о реализации рассылки результатов на почту тестируемого

Для организации электронной рассылки результатов тестирования применяется сервис EmailJS. Это serverless решение, которое позволяет отправлять электронные письма без необходимости настройки собственного почтового сервера. EmailJS интегрируется с веб-приложением, позволяя отправлять электронные письма напрямую из браузера пользователя. Это обеспечивает простоту в настройке и использовании функционала отправки писем, сокращая время и ресурсы, необходимые для реализации данной функциональности.

Выбор сервиса email JS вместо написания собственных скриптов для отправки почты был обоснован несколькими преимуществами:

Простота использования: Сервис email JS предоставляет простой и понятный интерфейс для отправки электронных писем из JavaScript-кода. Это позволяло бы сосредоточиться на бизнес-логике приложения, не тратя много времени на написание и настройку скриптов для отправки почты.

Легкая настройка: Email JS предлагает интуитивно понятный инструментарий для настройки отправки писем. Не нужно было настраивать и поддерживать собственный SMTP-сервер или устанавливать специальные библиотеки и зависимости. Все настройки и конфигурации выполняются через интерфейс веб-приложения email JS.

Интеграция с различными почтовыми сервисами: Email JS обеспечивает интеграцию с популярными почтовыми сервисами, такими как Google, Yandex и другими. Это означает, что разработчику не нужно беспокоиться о поддержке различных почтовых провайдеров и их API. Email JS берет на себя эту задачу, обеспечивая надежную доставку писем через различные почтовые системы.

Использование HTML-шаблонов для писем: Email JS позволяет создавать и использовать HTML-шаблоны для оформления электронных писем. Это делает процесс создания красиво оформленных и информативных писем гораздо проще и эффективнее, чем при написании собственных скриптов.

В конце концов, такое решение позволило вынести важную функциональность – отправку результатов на почту – из кода в графический интерфейс, что повысило удобство использования приложения для людей, не знакомых с техническими подробностями работы приложения. Так, например, от преподавателя бы требовалось только редактировать шаблон письма в самом сервисе, подставляя в него нужные названия переменных, которые получает приложение из элементов на html странице. К тому же система имеет удобный интерфейс для просмотра статистики отправленных писем.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 3 – Редактирование шаблона письма через редактор TinyMC(без кода)

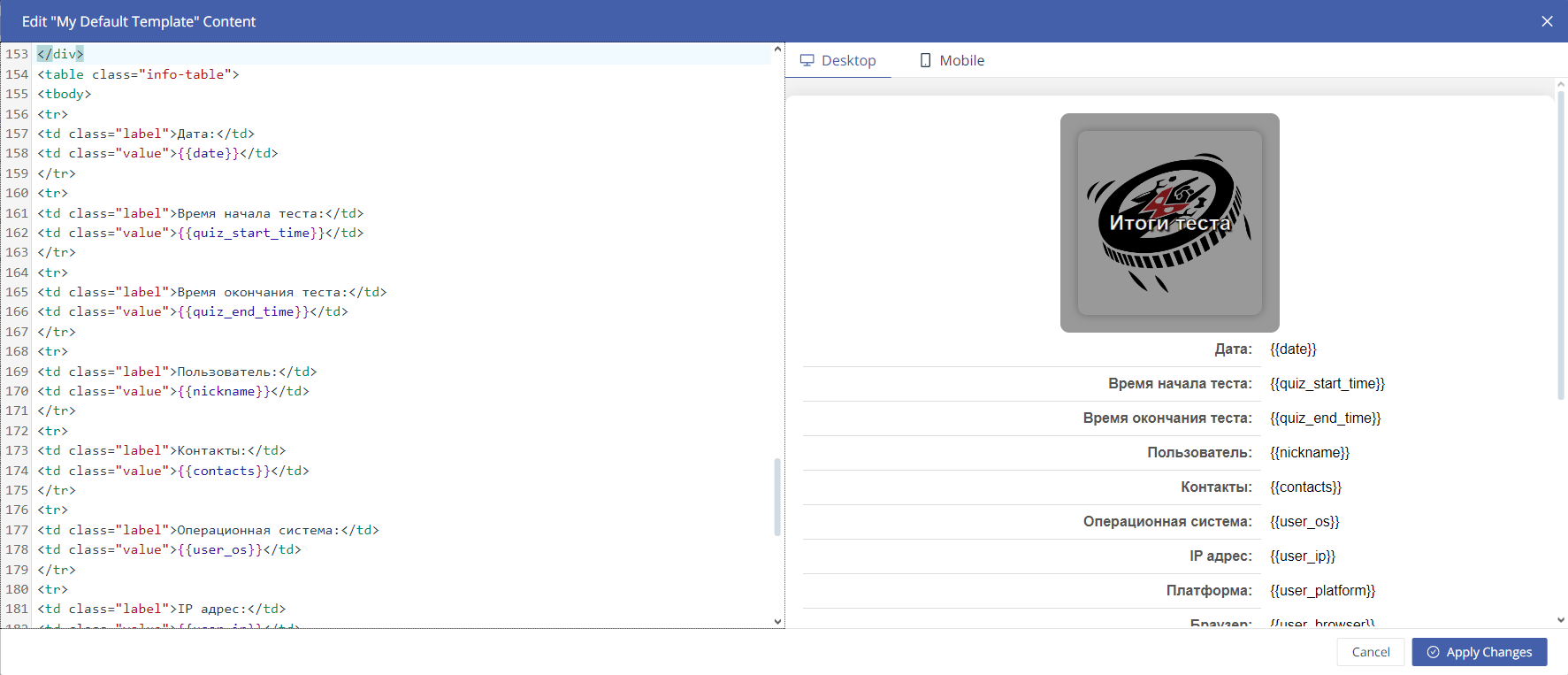


Рисунок 4 – Редактирование шаблона через HTML код с возможностью просмотра результатов в реальном времени

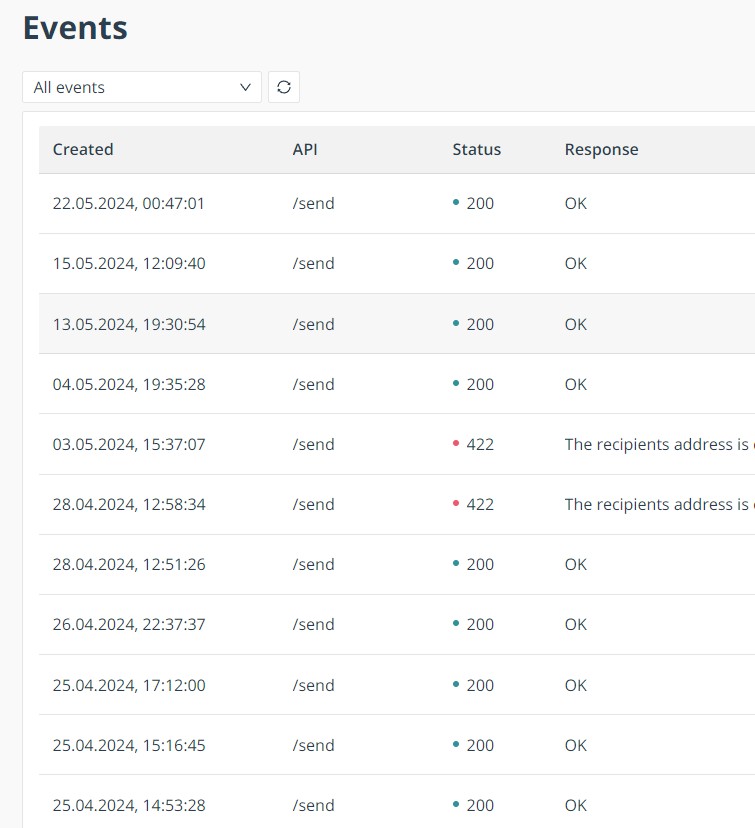


Рисунок 5 – Статистика отправленных писем (1\2)

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

Рисунок 6 – Статистика отправленных писем (2\2)

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

Рисунок 7 – Подробное содержимое и технические детали отправленного письма

2.3 Бэкенд

Серверная часть приложения реализована с использованием PHP. PHP обрабатывает запросы от клиента, включая создание новых тестовых вопросов и будущее взаимодействие с базой данных, как только генерируемые вопросы будут храниться в базе данных, а не в директориях сервера. К тому же, будущее внедрение возможности авторизации пользователя наиболее простым способом было бы реализовать именно с помощью PHP в силу обилия справочных материалов.

2.4 Функциональность приложения

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 8 – Главная страница приложения

Минимально жизнеспособная версия приложения расположена по адресу <https://trelawney.online>. Приложение предоставляет пользователям возможность создавать и проходить тесты по различным темам с ограничением во времени.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 9 – Приветственная карточка с заглавием теста

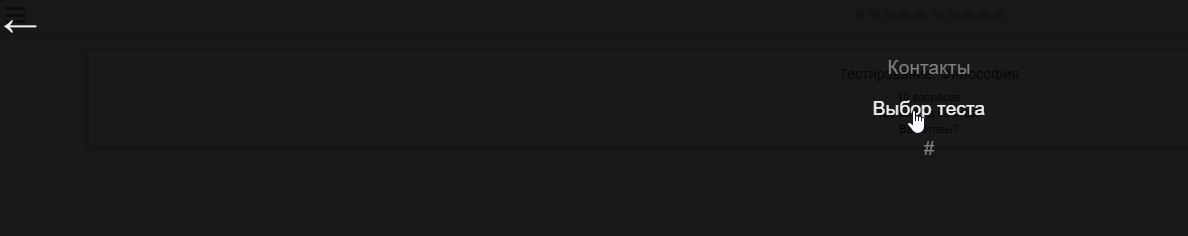


Рисунок 10 – Возможность выбора разных тестов(1\2)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Возможность выбора разных тестов(2\2)

Пользователи могут создавать тесты с разнообразными типами вопросов и форматами ответов, что позволяет адаптировать тесты под конкретные потребности и цели обучения.

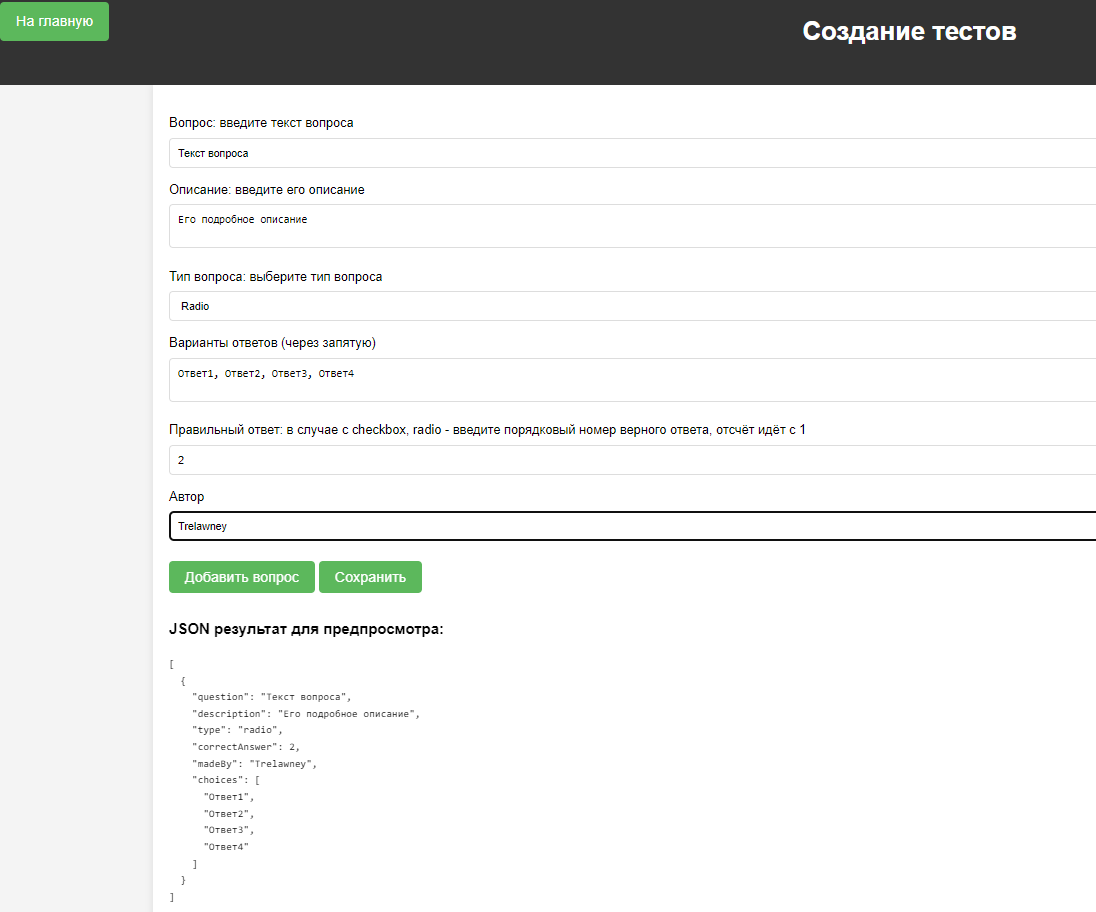


Рисунок 12 – Страница создания вопросов к тесту

Приложение поддерживает разнообразные типы вопросов, такие как одиночный выбор, множественный выбор, ввод текста и другие. Каждый тип вопроса имеет свой формат ответа, который определяет правильные и неправильные.

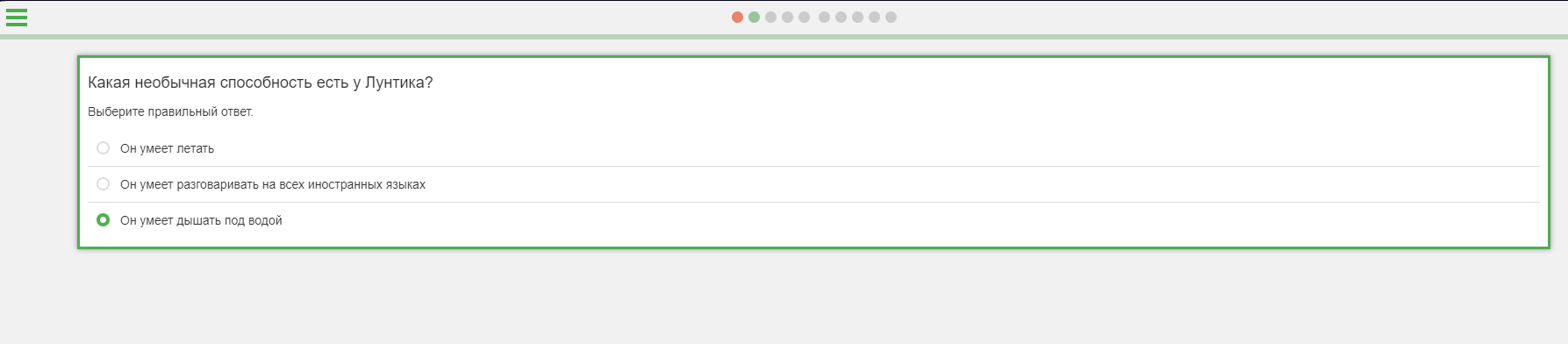


Рисунок 13 – Интерфейс карточки с типом ответа «один из многих»

При начале прохождения теста, пользователю предлагается указать (что, впрочем необязательно) данные электронной почты для получения результатов тестирования.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 14 – Модальное окно с предложением ввести адрес электронной почты для обратной связи с пользователем

Наконец сами результаты тестирования пользователь может как просмотреть сразу же на странице приложения, узнав кол-во правильных ответов и посмотрев все правильные ответы, так и получить на почту, указанную в форме на Рисунке 14, подробную статистику.

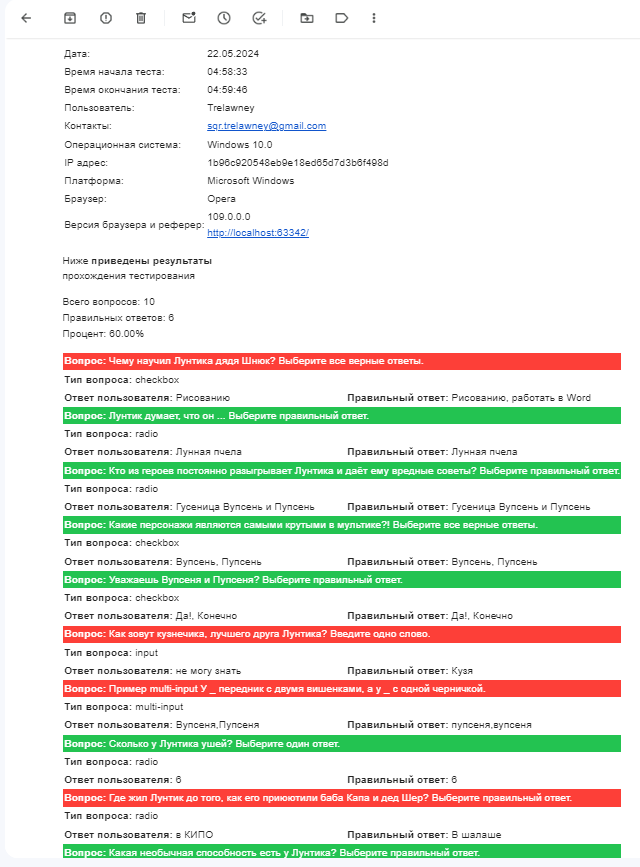


Рисунок 15 – Подробная статистика результатов тестирования, полученная на электронную почту

2.5 Дальнейшее развитие

Планы по дальнейшему развитию приложения и добавлению новых функций таковы:

В дальнейшем развитии приложения планируется расширение его функциональности и добавление новых возможностей. Например, можно рассмотреть внедрение дополнительных типов вопросов и форматов ответов для расширения гибкости при создании тестов. Т.е. тесты с вариантами ответа с выбором изображений, а так же сопоставления объектов. Однако прежде всего видится логичным реализовать систему авторизации пользователей с хранением данных учётной записи авторизированного пользователя в базе данных. На данный момент создание теста и манипуляция с файлами, содержащими вопросы теста, доступна любому пользователю, перешедшему по ссылке, что является очевидной уязвимостью в информационной безопасности приложения. К тому же стоит перенести хранение вопросов тестов из файлов формата JSON в базу данных, оперируя с ними путём совершения запросов к БД.

Второстепенным мне видится настройка внешнего вида приветственной страницы, с наполнением её подробными инструкциями, слайдами и, возможно, видеоматериалами по использованию платформы.

Также стоит добавить, что используемая архитектура хоть и позволяет избежать работу с фреймворками, при этом сохранив быстродействие и избавившись от необходимости перезагрузки страницы для продвижения по вопросам теста, тем не менее не является идеальной и не рассчитывалась на большое кол-во пользователей, что может привести к проблемам в будущем. Однако реализуемое мной решение позиционируется скорее как решение для узкого круга лиц с возможностью предоставления «исключительного доступа» группам лиц (обучающиеся и прочие тестируемые). Поэтому работа с последним замечанием видится мне наименее существенной.

ЗАКЛЮЧЕНИЕ

В ходе проведенной курсовой работы была проанализирована проблема тестирования в корпоративной среде и образовании. Анализ показал, что тесты используется для проведения различных экзаменов и олимпиад и для оценки результативности сотрудников в ассессмент-центрах.

На основании проблем, связанных с проведением тестирования с использованием бумажных бланков, были выделены требования к разрабатываемой системе онлайн-тестирования. Во-первых, было принято решение о максимизации скорости работы пользовательского интерфейса при этом без необходимости использовать фреймворки.

В работе был дан обоснованный выбор набора технологий, используемых для разработки веб-сайта, а также было проведено краткое описание архитектуры основного модуля.

Конечный продукт позволит мне, а также другим заинтересованным лицам, создающим тесты или участвующих в тестировании, повышать свою эффективность, сводя к минимуму время, затрачиваемое на создание и реализацию тестов. Он также должен улучшить качество тестирования, предоставляя аналитические инструменты для своих пользователей.

В результате этого проекта было создано онлайн-приложение, прикладная значимость которого, как я считаю, не может быть недооценена, потому что решение реализовано в виде уже работающей системы, которая была протестирована на обучающихся отделения СПО АНПОО «Кубанский Институт Профессионального образования» и так же была использована в одной из работ студента Педагогического направления Факультета Математики и Компьютерных наук, как техническое решение по реализации возможности тестирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. American Psychological Association, Assessment Centers Help Companies Identify Future Managers. Электронный ресурс. Дата посещения 18.04.2024. <http://www.apa.org/research/action/managers.aspx>
2. Chua Yan Piaw. Replacing Paper-based Testing with Computer-based Testing in Assessment: Are we Doing Wrong? Электронный ресурс. Дата посещения 16.03.2022.

<http://www.sciencedirect.com/science/article/pii/S1877042812050549>

1. Сорокин А.А. Объектно-ориентированное программирование: учебное пособие (курс лекций) / А.А. Сорокин; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет», Министерство образования и науки Российской Федерации. – Ставрополь : СКФУ, 2014. – 174с. : ил
2. Паттерны для новичков: MVC vs MVP vs MVVM. − URL: https://habr.com/post/215605/ (Дата обращения 01.03.2024).
3. Kyle Simpson. You Don’t Know JS. Серия книг. 2013-2017гг
4. Prompt Engineering Guide Электронный ресурс. Дата посещения: 14.05.2024 <https://www.promptingguide.ai/ru/introduction/examples>

ПРИЛОЖЕНИЕ А

Ссылка на гит-репозиторий с кодом приложения:

<https://github.com/EsquireTrelawney/JustAQuizApp.git>

Ссылка на веб-приложение в сети Интернет: <https://trelawney.online>