Содержание

[Введение 2](#_Toc192777037)

[1 Объект и предмет исследования 3](#_Toc192777038)

[1.1 Актуальность темы 3](#_Toc192777039)

[1.2 Цель и задачи работы 3](#_Toc192777040)

[1.3 методы исследования 4](#_Toc192777041)

[2 Теоретическая часть 6](#_Toc192777042)

[2.1 Обзор существующих решений для автоматизированного решения математических задач 6](#_Toc192777043)

[2.2 Основы работы с обработкой текста и распознаванием математических выражений 9](#_Toc192777044)

[2.3 Использование баз данных для хранения и обработки данных 10](#_Toc192777045)

[2.4 Инструменты и технологии разработки интерактивных веб-приложений 11](#_Toc192777046)

[3 Проектирование системы 14](#_Toc192777047)

[3.1 Требования к интерактивному сайту 14](#_Toc192777048)

[3.2 Проектирование базы данных 15](#_Toc192777049)

[3.3 Проектирование структуры веб-приложения 15](#_Toc192777050)

[4 Реализация интерактивного сайта 17](#_Toc192777051)

[4.1 Разработка механизма распознавания математических выражений 17](#_Toc192777052)

[4.2 Разработка интерфейса для ввода и обработки запросов 17](#_Toc192777053)

[4.3 Интеграция с базой данных 21](#_Toc192777054)

[Заключение 23](#_Toc192777055)

[Список использованных литературы 24](#_Toc192777056)

[Приложение А Справка о проверке на наличие заимствований 25](#_Toc192777057)

[Приложение Б Листинг программного кода 26](#_Toc192777058)

# Введение

В современном мире цифровые технологии играют ключевую роль в автоматизации вычислений и изучении математики. Важно предоставлять учащимся инструменты, способные облегчить выполнение задач и способствовать глубокому пониманию материала. Интерактивное веб-приложение для решения математических задач предоставляет удобный инструмент для получения ответов и изучения теоретического материала.

Цель проекта — создать веб-приложение, позволяющее школьникам и студентам решать математические задачи, изучать темы по школьной программе и проверять усвоенные знания. Приложение ориентировано на пользователей различного уровня подготовки, а также может быть полезным преподавателям для демонстрации решений и проверки знаний учеников.

Функционал включает базу знаний с теоретическими материалами и формулами, автоматическое решение задач, где предоставляется только ответ, а также генерацию задач для самостоятельной проверки знаний. Это помогает пользователям не только изучать новые темы, но и закреплять материал через практику.

Разработка велась с использованием HTML, CSS, JavaScript для клиентской части и Django (Python) для серверной логики. Использование Django обеспечивает удобную работу с базами данных и функциональность бэкенда. Приложение развернуто на Railway, что гарантирует доступность из любой точки мира. В процессе разработки особое внимание уделено удобному интерфейсу, точности вычислений и возможностям расширения. В перспективе планируется добавление графиков, улучшение алгоритмов обработки задач и внедрение интерактивных тестов.

Проект ориентирован на учеников и студентов, помогая им осваивать математику в доступной форме. Интерактивность, простота использования и образовательная направленность делают приложение полезным инструментом в учебном процессе, позволяя не только решать задачи, но и систематически изучать математические концепции. Это мотивирует пользователей к изучению математики, предоставляя удобный инструмент для самоподготовки и обучения.

1. Объект и предмет исследования
   1. Актуальность темы

В условиях цифровизации образования и науки растет потребность в доступных и эффективных инструментах для решения математических задач. Многие студенты, преподаватели и специалисты сталкиваются с необходимостью быстрого выполнения вычислений, анализа данных и проверки решений, однако традиционные методы, такие как бумажные конспекты, калькуляторы или громоздкие программные комплексы, не всегда удобны.

Разработка интерактивного веб-приложения для решения математических задач актуальна по нескольким причинам. Во-первых, она позволяет автоматизировать процесс вычислений, минимизируя вероятность ошибок. Во-вторых, подобное приложение доступно с любого устройства, имеющего выход в интернет, что делает его удобным инструментом для удаленного обучения и работы. В-третьих, оно способствует развитию цифровой грамотности пользователей, помогая осваивать современные технологии и улучшать навыки работы с математическими данными.

Функционал включает базу знаний с теоретическими материалами и формулами, автоматическое решение задач, где предоставляется только ответ, а также генерацию задач для самостоятельной проверки знаний. Это помогает пользователям не только изучать новые темы, но и закреплять материал через практику.

Разработка велась с использованием HTML, CSS, JavaScript для клиентской части и Django (Python) для серверной логики. Использование Django обеспечивает удобную работу с базами данных и функциональность бэкенда. Приложение развернуто на Railway, что гарантирует доступность из любой точки мира. В процессе разработки особое внимание уделено удобному интерфейсу, точности вычислений и возможностям расширения. В перспективе планируется добавление графиков, улучшение алгоритмов обработки задач и внедрение интерактивных тестов.

Проект ориентирован на учеников и студентов, помогая им осваивать математику в доступной форме. Интерактивность, простота использования и образовательная направленность делают приложение полезным инструментом в учебном процессе, позволяя не только решать задачи, но и систематически изучать математические концепции. Это мотивирует пользователей к изучению математики, предоставляя удобный инструмент для самоподготовки и обучения.

* 1. Цель и задачи работы

Разработка интерактивного сайта для изучения и решения математических задач направлена на создание удобной и эффективной образовательной платформы, которая поможет пользователям осваивать математические темы, выполнять вычисления и проверять свои знания.

Целью работы является создание веб-приложения, обеспечивающего пользователям доступ к учебным материалам, инструментам для решения задач и возможностям проверки знаний в удобном интерактивном формате. Для достижения этой цели необходимо разработать удобный интерфейс, который позволит пользователям легко ориентироваться в функционале сайта, быстро находить нужные разделы и комфортно взаимодействовать с системой. Важным аспектом является обеспечение высокой скорости работы, что позволит оперативно загружать страницы, обрабатывать пользовательские запросы и обеспечивать плавную работу веб-приложения.

Так как доступ к образовательным ресурсам должен быть максимально удобным, сайт должен функционировать через веб-браузер без необходимости установки дополнительного программного обеспечения, что обеспечит его доступность из любой точки мира. Для полноценного обучения пользователям необходимо предоставить доступ к базе знаний, содержащей теоретические материалы, примеры решений и пояснения по различным темам математики. Помимо изучения материала, система должна позволять проверять знания учеников, поэтому важным этапом разработки является внедрение инструментов автоматической генерации заданий, проверки решений и ведения истории выполнения упражнений.

Реализация всех этих аспектов обеспечит создание функционального и удобного инструмента для обучения математике, позволяющего пользователям не только изучать теоретический материал, но и применять знания на практике, тренироваться в решении задач и контролировать свой прогресс.

* 1. методы исследования

В процессе разработки интерактивного веб-приложения для изучения и решения математических задач были использованы различные методы исследования, позволяющие провести всесторонний анализ, обоснование выбора технологий и оценку эффективности созданной системы.

На первом этапе работы был применен теоретический анализ, включающий изучение существующих образовательных платформ, их функциональных возможностей и методик обучения математике в цифровой среде. Это позволило определить основные требования к разрабатываемому приложению, выделить ключевые функции и выбрать наиболее подходящие технологии для их реализации.

Для формирования структуры системы использовался метод моделирования, с помощью которого были разработаны логические схемы взаимодействия компонентов, описаны сценарии работы пользователей и спроектирована архитектура базы данных. Это позволило структурировать процесс разработки и заранее предусмотреть возможные проблемы, связанные с функциональностью и удобством использования.

На этапе практической реализации применялся метод проектирования, включающий разработку пользовательского интерфейса, программную реализацию серверной и клиентской части веб-приложения, а также интеграцию математических инструментов, таких как калькулятор, генератор задач и система решения уравнений.

Для проверки работоспособности и надежности системы использовались эмпирические методы исследования, включая тестирование и эксперимент. В ходе тестирования проводилась проверка корректности вычислений, правильности обработки пользовательских запросов и устойчивости системы к возможным ошибкам. Также были проведены тесты на удобство использования, в рамках которых анализировалась скорость работы интерфейса, логичность навигации и удобство взаимодействия пользователей с основными функциями приложения.

Таким образом, применение комплекса методов исследования, включающих теоретический анализ, моделирование, проектирование, тестирование и эксперимент, позволило создать эффективное и удобное интерактивное веб-приложение, соответствующее современным требованиям к образовательным платформам.

2 Теоретическая часть

## 2.1 Обзор существующих решений для автоматизированного решения математических задач

Автоматизированные системы для решения математических задач развиваются на основе технологий машинного обучения, символьных вычислений и обработки естественного языка. Современные решения включают специализированные математические пакеты, онлайн-сервисы, мобильные приложения и интегрированные модули в образовательных платформах.

Одними из наиболее известных математических систем являются Wolfram Alpha, SymPy, Maxima и Mathematica. Wolfram Alpha использует мощный движок символьных вычислений и способен решать широкий спектр задач, включая алгебру, анализ, дифференциальные уравнения и статистику. SymPy представляет собой библиотеку на Python, которая предоставляет инструменты для символьных вычислений, упрощения выражений и решения уравнений.

Среди популярных онлайн-сервисов можно выделить Photomath и Microsoft Math Solver, использующие технологии компьютерного зрения для распознавания математических выражений с изображений. Эти системы применяют нейросетевые модели для интерпретации текста и дальнейшего вычисления результата.

В образовательных платформах, таких как Khan Academy и Brilliant, встроены механизмы пошагового объяснения решений, что делает их полезными для обучения. Google интегрировал решение математических выражений в свой поиск, позволяя пользователям быстро находить ответы без необходимости установки дополнительного программного обеспечения.

Несмотря на высокую точность и удобство существующих решений, у большинства из них есть ограничения. Они либо требуют постоянного подключения к интернету, либо не всегда корректно интерпретируют сложные математические записи. Кроме того, многие сервисы не предоставляют возможности полной интеграции в собственные проекты. В связи с этим разработка интерактивного сайта, позволяющего вводить текстовые математические выражения и автоматически их решать, становится актуальной задачей, особенно в образовательной и научной среде.

Photomath - мобильное приложение, которое использует камеру телефона для распознавания математических уравнений и отображения пошагового решения на экране, рисунок 1.



Рисунок 1 – Решение задач Photomath

Mathway — это онлайн-сервис, предназначенный для решения математических задач различного уровня сложности. Платформа охватывает широкий спектр математических разделов, включая арифметику, алгебру, геометрию, тригонометрию, математический анализ, статистику и другие направления. Одной из ключевых особенностей Mathway является возможность получения поэтапных решений с подробными пояснениями, что делает его полезным инструментом как для учащихся, так и для преподавателей.

Сервис предоставляет удобный интерфейс, позволяющий вводить математические выражения с клавиатуры или с использованием виртуальной панели с математическими символами. Пользователь может ввести уравнение, систему уравнений, неравенство или любую другую математическую задачу, после чего система автоматически анализирует запрос и предлагает решение. В отличие от простых калькуляторов, Mathway не только вычисляет результат, но и демонстрирует последовательность шагов, что особенно полезно для образовательных целей, рисунок 2.

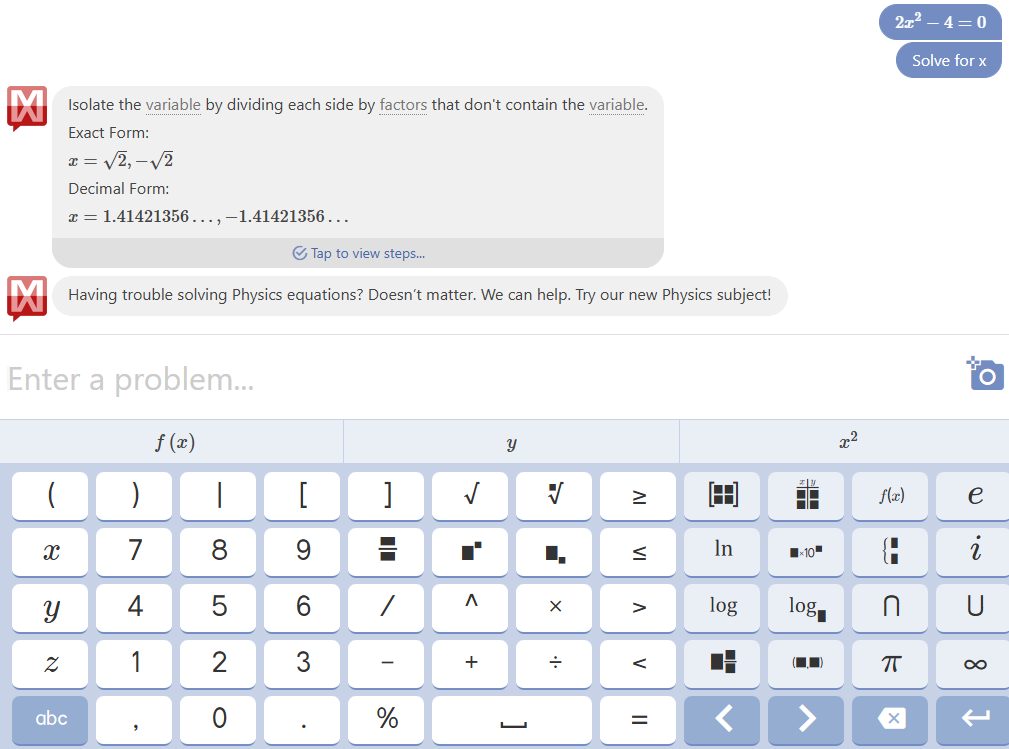


Рисунок 2 – Решение задачи Mathway

Microsoft Math Solver - инструмент от Microsoft, который предлагает пошаговые решения задач по алгебре, математическому анализу и другим темам, рисунок 3.

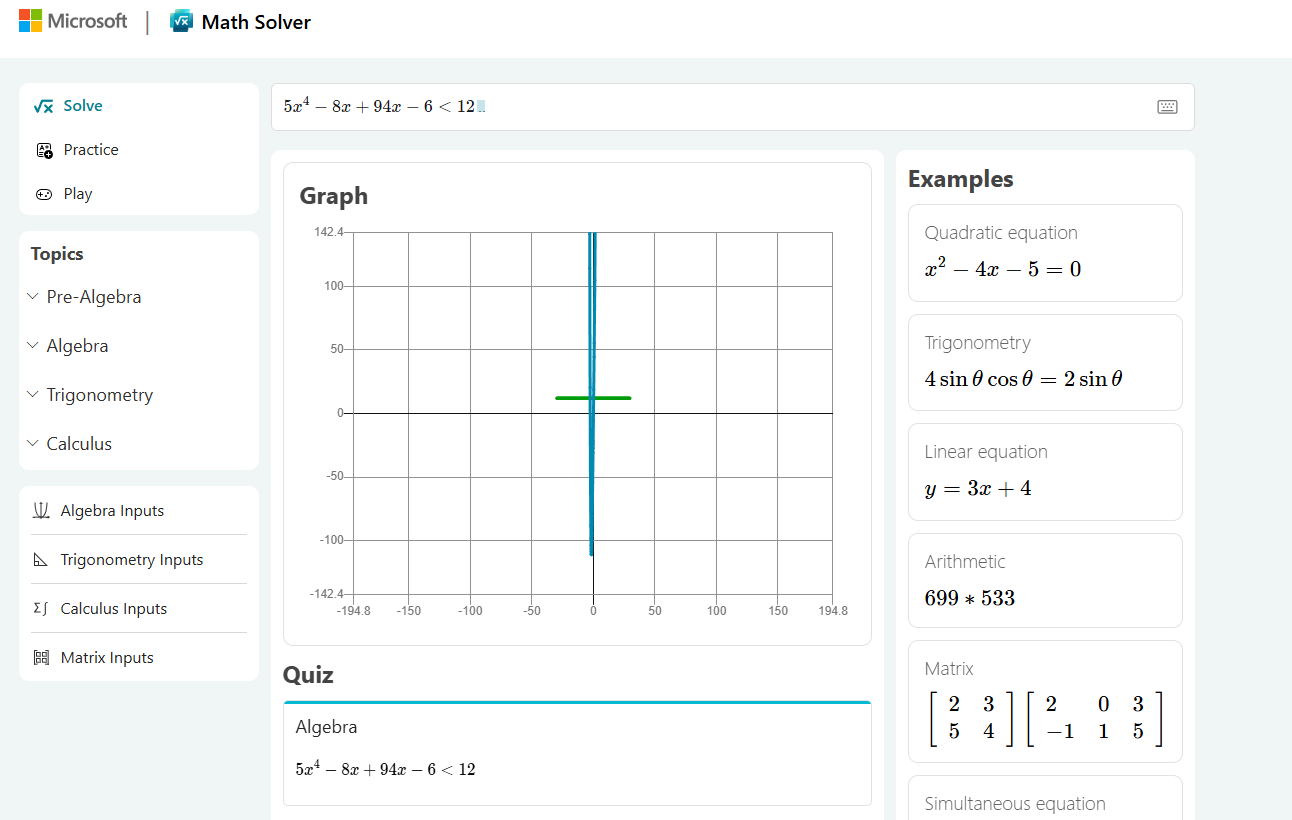


Рисунок 3 – решение задачи Math solver

## 2.2 Основы работы с обработкой текста и распознаванием математических выражений

Обработка текста и распознавание математических выражений являются ключевыми этапами при разработке интерактивного сайта для решения математических задач. Этот процесс включает разбор введённого пользователем текста, его интерпретацию в математическую форму и последующее вычисление результата.

Анализ введённого текста осуществляется с помощью методов обработки естественного языка, позволяющих распознавать числовые значения, математические операторы, функции и другие символы. Простые математические выражения, такие как 2 + 2, могут быть обработаны встроенными механизмами языков программирования, однако более сложные конструкции требуют применения регулярных выражений или специализированных библиотек.

Токенизация представляет собой разбиение текста на отдельные элементы, включая числа, операторы, переменные и функции. Для этого используются регулярные выражения или инструменты, такие как библиотека Tokenize в Python. Например, выражение sin(x) + 3x^2 можно разделить на sin, (, x, ), +, 3, x, ^, 2.

Синтаксический разбор превращает последовательность токенов в структурированное дерево, удобное для вычислений. Для этого применяются деревья разбора и стеки, позволяющие учитывать приоритет операций. Одним из популярных алгоритмов является обратная польская нотация, помогающая упростить выполнение математических операций.

Распознавание математических функций необходимо для корректного интерпретирования выражений, содержащих тригонометрические, логарифмические, степенные и другие функции. Здесь используются специализированные библиотеки, такие как SymPy и Math.js, которые позволяют работать с символьной математикой и выполнять сложные вычисления.

Обработка ошибок и проверка корректности выражений играют важную роль, поскольку важно предусмотреть механизм обнаружения синтаксических ошибок, таких как незакрытые скобки или некорректные операции вида 3++4. В этом помогают парсеры с генерацией исключений и встроенные валидаторы.

Таким образом, успешная реализация механизма обработки текста и распознавания математических выражений требует комплексного подхода, включающего обработку естественного языка, парсинг, работу с математическими библиотеками и алгоритмы проверки синтаксической корректности.

## 2.3 Использование баз данных для хранения и обработки данных

База данных играет ключевую роль в организации работы системы, поскольку позволяет сохранять историю вычислений, кэшировать результаты, управлять пользовательскими запросами и их решениями, а также обеспечивать безопасность и стабильность работы сервиса.

Выбор типа базы данных зависит от требований системы, объемов данных и особенностей их обработки. Реляционные базы данных, такие как PostgreSQL и MySQL, подходят для хранения структурированной информации, например, данных пользователей, истории вычислений, параметров выражений и записей о пользовательских сессиях. Они обеспечивают надежность хранения данных и возможность выполнения сложных запросов. В то же время NoSQL-решения, такие как MongoDB и Redis, могут применяться для хранения временных данных, кэширования результатов и ускорения обработки часто повторяющихся запросов. Это особенно актуально для высоконагруженных систем, где требуется минимизировать задержки и повысить скорость отклика.

Структура базы данных включает несколько ключевых компонентов, обеспечивающих функциональность системы. Хранение данных о пользователях необходимо в случае поддержки авторизации, что предполагает наличие информации об учетных записях, идентификаторах, адресах электронной почты, дате регистрации и других параметрах, например, роли пользователя в системе. История запросов фиксирует введённые выражения, их обработанные версии, полученные результаты и временные метки вычислений. Это позволяет пользователям отслеживать свои решения, повторно обращаться к ранее введенным данным и анализировать их.

Дополнительным инструментом оптимизации работы системы является кэширование выражений. Этот механизм позволяет существенно сократить время обработки повторяющихся вычислений, снижая нагрузку на сервер. Например, если пользователь вводит sqrt(16), система сначала проверяет наличие результата в кэше и мгновенно возвращает 4, исключая необходимость повторного вычисления. Для реализации кэширования могут использоваться такие инструменты, как Redis или встроенные механизмы кэширования Django, позволяющие хранить временные результаты и ускорять работу приложения.

Взаимодействие с базой данных осуществляется через ORM (Object-Relational Mapping), что позволяет разработчикам работать с данными, используя объектно-ориентированные подходы без необходимости написания сложного SQL-кода. В случае использования Django ORM или SQLAlchemy процесс взаимодействия с базой данных становится более удобным, уменьшается вероятность ошибок, упрощается поддержка кода и повышается безопасность за счет встроенных механизмов защиты, таких как автоматическое экранирование SQL-запросов.

Помимо хранения вычислений и кэширования, база данных может использоваться для организации логирования системы. Это необходимо для отслеживания активности пользователей, выявления ошибок, сбора статистики и дальнейшего анализа работы сервиса. Логирование помогает разработчикам быстро находить и исправлять возможные проблемы, а также оптимизировать производительность приложения.

Таким образом, грамотно спроектированная база данных играет ключевую роль в обеспечении быстродействия, надежности и удобства работы интерактивного математического сервиса. Она не только хранит информацию о вычислениях, но и помогает оптимизировать процесс обработки выражений, снижая нагрузку на сервер и повышая эффективность работы приложения. Интеграция современных механизмов кэширования, логирования и безопасного хранения данных делает систему устойчивой к нагрузкам и удобной для пользователей.

## 2.4 Инструменты и технологии разработки интерактивных веб-приложений

Для создания интерактивного сайта, способного распознавать и решать математические выражения, применяются современные веб-технологии, охватывающие серверную часть, клиентский интерфейс, базу данных и механизмы кэширования. Комплексное использование этих инструментов позволяет создать удобное, производительное и масштабируемое веб-приложение, обеспечивающее комфортную работу пользователей и высокую точность обработки математических данных.

Серверная часть разрабатывается на Django — мощном фреймворке на языке Python, который предоставляет удобные инструменты для обработки HTTP-запросов, работы с базой данных, кэширования и интеграции с внешними библиотеками. Django отличается гибкостью и безопасностью, что особенно важно при разработке образовательных сервисов. Для выполнения математических вычислений используется библиотека SymPy, предназначенная для символьной математики. Она позволяет решать уравнения, упрощать выражения, дифференцировать, интегрировать и выполнять другие математические операции. В случае необходимости выполнения числовых вычислений, особенно при работе с большими массивами данных, применяется NumPy, который ускоряет математические операции за счёт оптимизированных алгоритмов работы с многомерными массивами.

Обработка пользовательского ввода представляет собой одну из ключевых задач системы, так как вводимые выражения могут быть сложными и содержать различные математические конструкции. Для корректного восприятия математических выражений используются регулярные выражения и специализированные парсеры, такие как Lark и PLY. Они помогают интерпретировать вводимые пользователем формулы, проверять их на ошибки и преобразовывать в структуру, пригодную для дальнейшей обработки сервером. Дополнительно возможно использование алгоритмов машинного обучения для распознавания рукописного ввода, что повысит удобство работы с приложением.

Клиентская часть разрабатывается с применением HTML, CSS и JavaScript. Для динамичного обновления данных и удобного взаимодействия с пользователем используется React.js — современный фреймворк, позволяющий создавать высокоотзывчивые интерфейсы. Благодаря архитектуре компонентов React упрощает поддержку и расширение функционала веб-приложения. Визуализация математических выражений выполняется с помощью библиотеки MathJax, которая обеспечивает точное отображение формул в веб-интерфейсе, используя разметку LaTeX. Это делает ввод и вывод данных максимально приближенным к традиционному математическому написанию, что особенно полезно для образовательных платформ.

База данных является важнейшим компонентом системы, обеспечивающим хранение пользовательских данных, истории вычислений и кэширования результатов. В качестве основного хранилища данных используется PostgreSQL — реляционная база данных, обладающая высокой производительностью и надежностью. Она поддерживает сложные запросы, индексы, транзакции и механизмы защиты данных, что делает её идеальным решением для хранения структурированной информации. Для ускорения работы системы применяется Redis — высокопроизводительное хранилище данных в оперативной памяти, предназначенное для кэширования результатов вычислений и временного хранения пользовательских сессий. Это позволяет существенно снизить нагрузку на сервер и сократить время ответа на повторяющиеся запросы.

Развертывание и интеграция веб-приложения осуществляется с использованием контейнеризации на базе Docker, что позволяет упаковать все компоненты системы в изолированные контейнеры и упростить процесс развертывания на различных серверах. Для управления серверными процессами используется Gunicorn в сочетании с NGINX. Gunicorn обрабатывает запросы на уровне приложения, обеспечивая асинхронное выполнение задач, а NGINX выполняет роль обратного прокси-сервера, оптимизируя распределение нагрузки, повышая безопасность системы и обеспечивая быструю отдачу статических файлов.

Для хостинга приложения могут использоваться облачные платформы, такие как Railway, Heroku, AWS или DigitalOcean. Эти сервисы позволяют масштабировать вычислительные мощности в зависимости от нагрузки, а также обеспечивают надежное хранение данных и высокую доступность приложения. Для мониторинга работы сервера и сбора статистики используются инструменты, такие как Prometheus и Grafana, которые позволяют отслеживать производительность системы и своевременно выявлять возможные узкие места.

Дополнительно, для обеспечения безопасности и защиты пользовательских данных применяются механизмы аутентификации и авторизации на основе Django Authentication. Это позволяет разграничивать доступ пользователей к различным разделам системы, обеспечивая сохранность данных и защиту от несанкционированного использования. Также внедряется шифрование передаваемых данных с помощью SSL/TLS, что предотвращает перехват конфиденциальной информации при взаимодействии клиента с сервером.

Таким образом, использование современных технологий и инструментов позволяет создать мощное, удобное и безопасное веб-приложение, способное эффективно обрабатывать математические выражения, обеспечивать корректную работу с пользовательскими запросами и предоставлять решения в удобном формате. Интеграция механизмов кэширования, контейнеризации и облачного хостинга делает систему масштабируемой, надежной и доступной для пользователей по всему миру.

1. Проектирование системы
   1. Требования к интерактивному сайту

Интерактивный сайт для решения математических задач должен обеспечивать корректное распознавание выражений, точное вычисление и удобный интерфейс для взаимодействия с пользователем. Он должен быть не только функциональным, но и соответствовать ряду важных требований, включая быстродействие, надёжность, масштабируемость, производительность и продуманный сдержанный дизайн.

Быстродействие системы играет ключевую роль, так как пользователи ожидают мгновенного отклика на введённые запросы. Время обработки выражений должно быть минимальным даже для сложных уравнений, а интерфейс должен оставаться плавным и отзывчивым. Это достигается за счёт оптимизации вычислений, кэширования ранее полученных результатов и эффективного взаимодействия с сервером.

Надёжность системы заключается в её устойчивости к ошибкам и некорректным данным. Приложение должно корректно обрабатывать любой пользовательский ввод, предупреждая возможные ошибки и выдавая понятные сообщения, если выражение содержит недопустимые символы или синтаксические ошибки. Система не должна зависать или выдавать некорректные результаты при работе с математическими операциями любой сложности.

Масштабируемость позволяет поддерживать стабильную работу даже при увеличении количества пользователей. Серверная часть должна быть готова к высоким нагрузкам, обеспечивая равномерное распределение вычислений и эффективное использование ресурсов. Это особенно важно для онлайн-сервисов, где доступ должен быть возможен из любой точки мира без задержек и перебоев в работе.

Производительность приложения связана с минимальным временем обработки запросов и оптимизированной работой базы данных. Вычисления должны выполняться без задержек, а результаты – появляться на экране сразу после обработки. Использование технологий кэширования и продвинутых алгоритмов обработки выражений позволяет снизить нагрузку на сервер и ускорить выдачу ответов.

Сдержанный дизайн является важным элементом интерфейса, поскольку пользователи должны сосредотачиваться на решении математических задач, а не отвлекаться на излишние визуальные элементы. Интерфейс должен быть удобным, интуитивно понятным и адаптивным, чтобы обеспечивать комфортную работу на разных устройствах.

Это особенно важно для пользователей, изучающих математику, так как позволяет им лучше понять принципы вычислений. Решения должны быть точными, безошибочными и доступными для анализа, а сам процесс вычислений – максимально быстрым и удобным.

## 3.2 Проектирование базы данных

В качестве базы данных для интерактивного сайта по решению математических задач мы будем использовать SQLite. Этот выбор обусловлен её удобством, высокой скоростью работы и минимальными требованиями к ресурсам.

SQLite не требует отдельного серверного процесса, что значительно упрощает развертывание и администрирование. База данных представлена одним файлом, который легко копировать и переносить, а это удобно для резервного копирования и миграции. При этом SQLite полностью поддерживает транзакции, гарантируя целостность данных даже в случае сбоя.

Благодаря минимальному потреблению памяти и быстродействию, SQLite идеально подходит для веб-приложений, где не требуется одновременная обработка большого количества запросов. Индексация, встроенная оптимизация и поддержка SQL делают её достаточно мощным инструментом для хранения истории вычислений, пользовательских данных и кеширования результатов.

Ещё одним важным преимуществом является кроссплатформенность: SQLite одинаково хорошо работает на Windows, Linux, macOS и мобильных устройствах. Это позволяет легко развернуть приложение в любой среде.

Таким образом, SQLite — это оптимальный выбор для нашего проекта. Она обеспечивает простоту, надёжность, высокую скорость работы и удобство в управлении данными, что особенно важно для веб-приложения, ориентированного на обработку математических выражений.

## 3.3 Проектирование структуры веб-приложения

Структура интерактивного сайта для решения математических задач включает несколько ключевых компонентов: интерфейс для ввода выражений, серверную обработку, модуль вычислений и базу данных.

Описание логики работы системы  
Пользователь вводит математическое выражение в текстовом формате. Система анализирует введённый текст, определяет математические операции и преобразует выражение в формат, удобный для вычислений. Далее происходит вычисление результата с возможным пошаговым объяснением. Итоговый ответ отображается пользователю в удобочитаемом виде, при необходимости дополняясь графиками. Если функция истории включена, вычисления сохраняются в базе данных.

Алгоритм обработки пользовательских запросов  
Сначала сервер получает выражение от фронтенда и выполняет предварительную проверку синтаксиса, выявляя возможные ошибки, такие как незакрытые скобки или некорректные символы. Затем осуществляется токенизация и парсинг выражения, после чего выполняется его оптимизация, включая упрощение и приведение к стандартной форме. Далее система вычисляет результат с использованием библиотеки SymPy или её аналогов. Полученный ответ форматируется в LaTeX или MathML для удобного отображения. Если предусмотрено сохранение истории, результат заносится в базу данных. Наконец, сервер отправляет ответ клиенту.

Протоколы взаимодействия с базой данных  
Для удобства работы с реляционной базой данных используется Django ORM, обеспечивающая высокоуровневый доступ к данным. Взаимодействие между фронтендом и сервером осуществляется через REST API или GraphQL, что даёт гибкость и масштабируемость. Хранение истории запросов позволяет быстро получать повторные результаты без необходимости их пересчёта. Для снижения нагрузки на сервер активно применяется Redis, который кэширует часто используемые вычисления.

Продуманная архитектура системы обеспечивает удобный интерфейс, быструю обработку выражений и стабильную работу сервиса, делая его эффективным инструментом для решения математических задач.

# 4 Реализация интерактивного сайта

## 4.1 Разработка механизма распознавания математических выражений

Разработка интерактивного сайта для решения математических задач начинается с настройки серверной части, которая отвечает за обработку пользовательских запросов, вычисления и взаимодействие с базой данных. Серверная часть реализуется с использованием веб-фреймворка Django, который обеспечивает удобную работу с запросами и данными, а также упрощает взаимодействие с базой данных.

Основной механизм работы системы основан на передаче пользовательского ввода на сервер, его анализе и обработке с последующей отправкой результата обратно пользователю. После ввода математического выражения оно передаётся на сервер, где происходит его синтаксический разбор. Для этого используется библиотека SymPy, способная интерпретировать математические выражения, выполнять преобразования и вычислять числовые результаты.

Далее обработанное выражение проверяется на корректность. В случае ошибок система выдаёт пользователю сообщение о некорректном вводе. Если выражение верно, оно передаётся в вычислительный модуль, где происходит выполнение операций. Полученный результат затем форматируется и отправляется пользователю.

Для повышения производительности в системе предусмотрен механизм кэширования повторяющихся вычислений. Если выражение уже было введено ранее, его результат извлекается из базы данных, что позволяет сократить время обработки запроса и снизить нагрузку на сервер.

Таким образом, серверная часть веб-приложения организована так, чтобы обеспечивать быстрое и точное решение математических задач, используя мощные алгоритмы символьных вычислений и эффективное управление данными.

## 4.2 Разработка интерфейса для ввода и обработки запросов

Интерактивный сайт для решения математических задач разработан таким образом, чтобы пользователи могли легко находить нужную информацию, выполнять вычисления и тренироваться в решении задач. Интерфейс приложения интуитивно понятен и удобен, что делает процесс обучения комфортным и эффективным.

Главная страница является отправной точкой для работы с сайтом. В верхней части расположено название курсовой работы, отражающее суть и цель проекта. Ниже представлены основные элементы навигации в виде кнопок, каждая из которых ведет в отдельный раздел. Пользователь может выбрать один из доступных режимов: изучение тем, использование встроенного калькулятора, генерацию задач, решение уравнений или отработку базовых арифметических операций, таких как сложение и вычитание. Такое структурирование интерфейса позволяет быстро переходить к нужной функции и ориентироваться в возможностях приложения, рисунок 4.

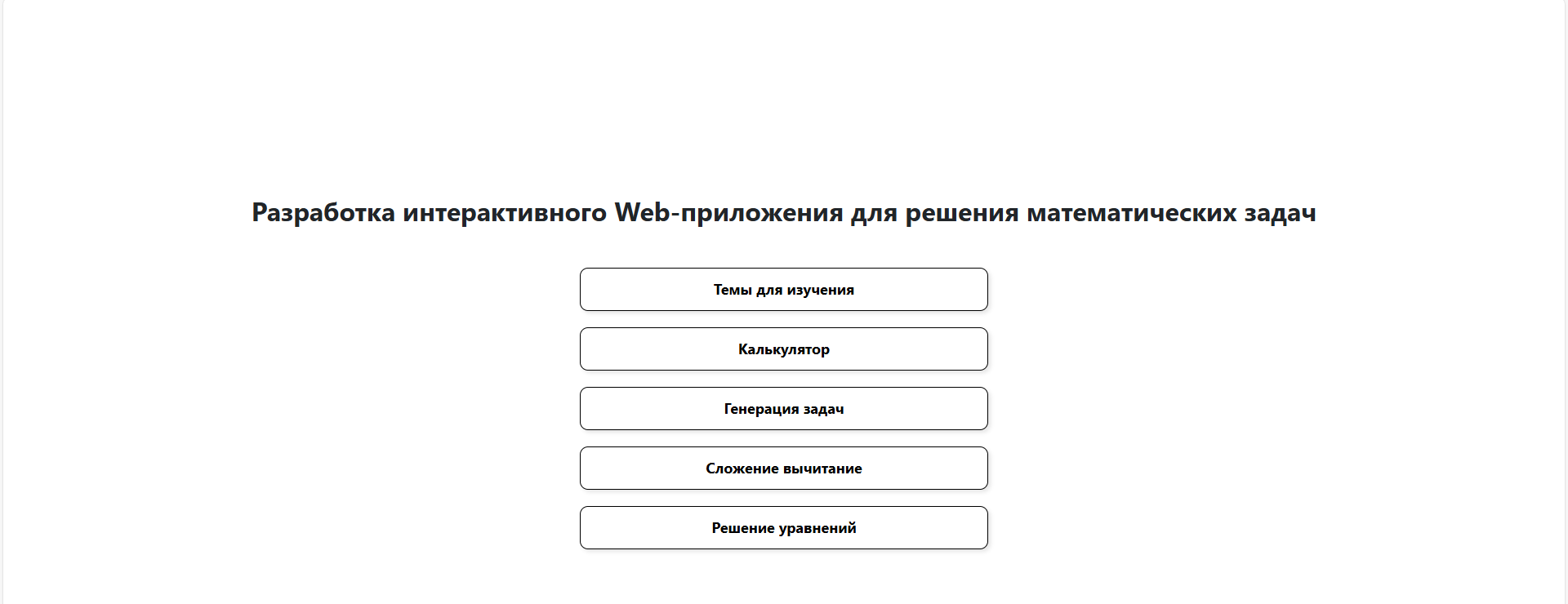


Рисунок 4 – главная страница веб-приложения

Раздел «Темы для изучения» представляет собой подборку математических материалов, сгруппированных по классам с 1 по 11. Здесь пользователи могут выбрать интересующую их тему, изучить теоретический материал, познакомиться с примерами решений и потренироваться в решении задач. Каждая тема представлена в виде отдельной кнопки, что упрощает навигацию. Также на этой странице предусмотрена кнопка «Домой», позволяющая в любой момент вернуться на главную страницу, рисунок 5.

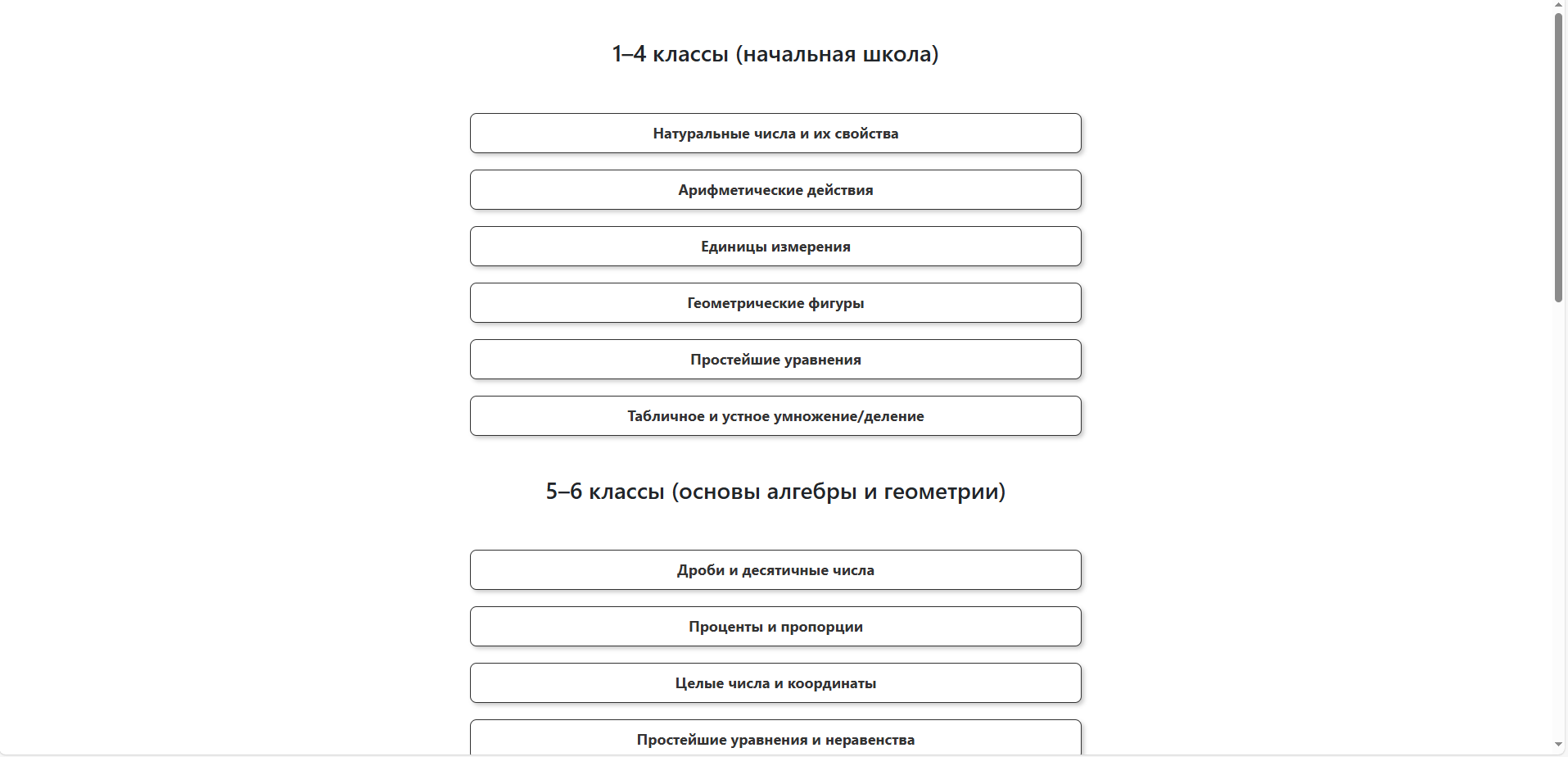


Рисунок 5 – раздел базы знаний со всеми темами за 1-11 классы.

Для выполнения базовых вычислений в приложении предусмотрен калькулятор. Он реализует стандартные арифметические операции, обеспечивая пользователям удобный инструмент для работы с числами. Простота интерфейса позволяет использовать его как для быстрых расчетов, так и в качестве вспомогательного средства при решении более сложных задач, рисунок 6.

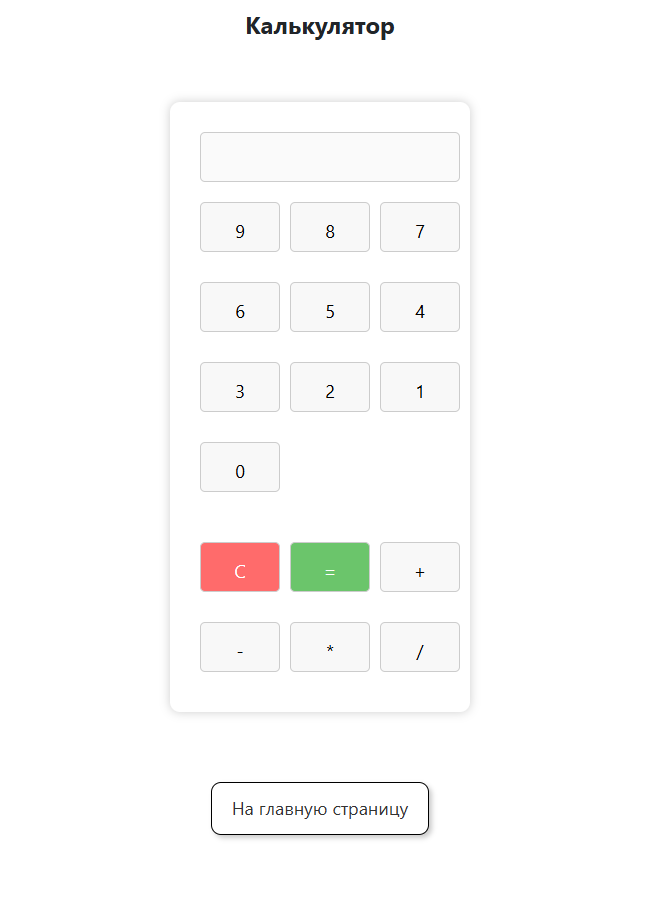


Рисунок 6 – обыкновенный калькулятор

Раздел «Генерация задач» разработан для автоматического подбора математических упражнений различной сложности. Пользователь может выбрать желаемый уровень сложности из выпадающего списка, после чего на экране отобразится случайно сгенерированная задача. Под задачей расположено поле ввода, куда пользователь может ввести свой ответ. Если представленное задание кажется слишком сложным, можно воспользоваться кнопкой генерации новой задачи, что делает процесс обучения более гибким и адаптивным к индивидуальному уровню знаний, рисунок 7.

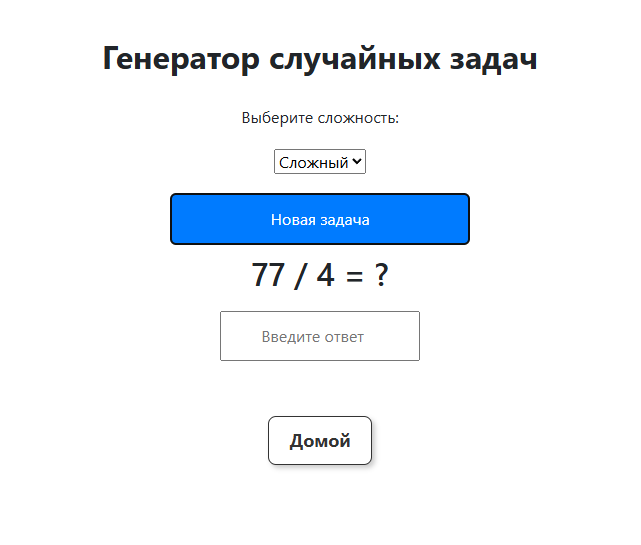


Рисунок 7 – раздел генерация случайных задач

Для работы с уравнениями в приложении предусмотрен отдельный раздел «Решение уравнений». В нем пользователь вводит математическое выражение в текстовом поле, после чего система автоматически вычисляет его корни и отображает результат. Это позволяет не только проверять правильность решений, но и анализировать различные методы решения, рисунок 8.

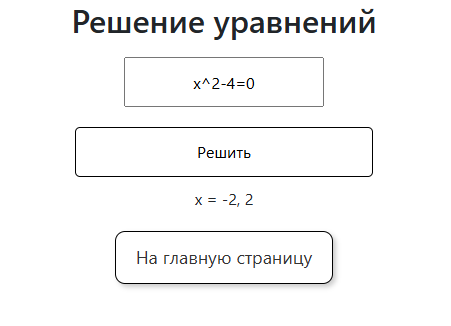


Рисунок 8 – решений уравнений

Таким образом, интерфейс сайта сочетает удобство и широкий функционал, обеспечивая пользователям комфортное взаимодействие с системой. Каждый раздел продуман с точки зрения логики работы и доступности информации, что делает процесс обучения математике максимально эффективным.

## 4.3 Интеграция с базой данных

Интерактивный сайт для решения математических задач использует базу данных для хранения истории вычислений, пользовательских данных (при наличии регистрации) и логов обработки выражений. Интеграция с базой данных позволяет улучшить производительность, ускорить обработку повторяющихся запросов и предоставить пользователям возможность просматривать историю решений.

Для работы с данными применяется реляционная база данных, такая как SQLite. Связь между серверной частью и базой данных осуществляется через Django ORM, что упрощает выполнение запросов и управление структурами данных.

Процесс интеграции включает несколько этапов. Первым шагом является проектирование структуры базы данных, включающей таблицы для хранения выражений, их результатов и времени выполнения. Далее происходит настройка соединения с базой данных в серверной части приложения, после чего реализуются механизмы записи, чтения и обновления данных.

При каждом новом запросе система проверяет, встречалось ли данное выражение ранее. Если оно уже есть в базе данных, результат извлекается без повторного вычисления, что снижает нагрузку на сервер. В случае отсутствия в базе выражение передаётся в вычислительный модуль, после чего результат сохраняется для последующего использования.

Таким образом, интеграция с базой данных обеспечивает надёжное хранение информации, улучшает производительность системы и предоставляет пользователям доступ к истории решений, делая работу сайта более удобной и эффективной.

Основной сущностью является пользователь (User), который может иметь разные и роли и соответствующие им возможности, одна дает право на просмотр, другая на изменение. Таблица пользователей испольцуется для отслеживания посетителей веб-приложения, а также их результатов, таблица 1.

Таблица 1.

Сущность Users.

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Связь |
| id | AutoField (PK) | Первичный ключ |
| username | CharField (max\_length=150) | Унаследовано от AbstractUser |
| first\_name | CharField (max\_length=150) | Унаследовано от AbstractUser |
| last\_name | CharField (max\_length=150) | Унаследовано от AbstractUser |
| password | CharField (max\_length=128) | Унаследовано от AbstractUser |
| permission | CharField (max\_length=10, choices=ROLE\_CHOICES, default='student') | Унаследовано от permission в таблице groups |
| image | ImageField (upload\_to='users\_images', null=True, blank=True) | - |

Таблица Group используется для управления ролями пользователей в веб-приложении. Каждая группа представляет собой определенный набор прав (permissions), которые определяют возможности пользователя. Например, одна группа может предоставлять доступ только к просмотру материалов, а другая — к их изменению. Пользователь может состоять в одной или нескольких группах, что позволяет гибко настраивать уровень доступа. Связь между пользователями и группами организована через отношение многие ко многим, таблица 2.

Таблица 2.

Сущность Groups.

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | AutoField(Pk) | |  | | --- | |  |  |  | | --- | | Первичный ключ | |
| name | CharField (max\_length=150, unique=True) | |  | | --- | |  |  |  | | --- | | Уникальное название группы | |
| permissions | ManyToManyField (Permission) | |  | | --- | |  |  |  | | --- | | Связь многие ко многим с правами | |

# Заключение

В условиях цифровизации образования растет потребность в доступных инструментах для решения математических задач. Многие студенты и преподаватели нуждаются в быстром выполнении вычислений и проверке решений, однако традиционные методы не всегда удобны.

Целью данной работы является разработка интерактивного веб-приложения, способного автоматически решать математические задачи, предоставлять базу знаний и генерировать задания для закрепления материала. Для этого проведен анализ существующих решений, спроектирована архитектура клиентской и серверной частей, разработан удобный интерфейс и реализованы алгоритмы обработки выражений. Интеграция специализированных математических библиотек, таких как SymPy, позволила добиться высокой точности вычислений.

Исследование включало обзор аналогичных приложений, выявление их преимуществ и недостатков, проектирование структуры системы и тестирование её работы. Были реализованы механизмы обработки ошибок и валидации данных для повышения надежности. Разработка велась с использованием Django для серверной части, HTML, CSS и JavaScript для клиентской. Приложение развернуто на Railway, что обеспечивает его доступность.

В ходе работы создано интерактивное веб-приложение, позволяющее пользователям вводить математические выражения и получать точные вычисления. Тестирование подтвердило корректность работы алгоритмов и удобство интерфейса. Реализована система хранения истории вычислений для повышения удобства использования. В перспективе планируется расширение функционала, интеграция дополнительных модулей и разработка мобильной версии.

Таким образом, поставленные задачи успешно выполнены, а разработанная система представляет собой полезный инструмент для автоматизированного решения математических задач.

Список использованных литературы

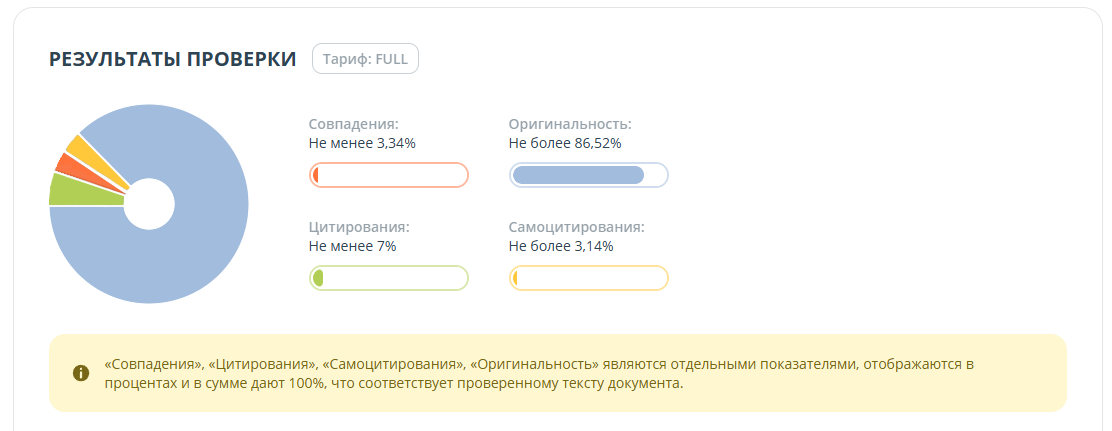
1. Баранов А.С., Васильев В.И. Основы разработки веб-приложений. - М.: Наука, 2015. - 420 с.
2. Громов Д.В. Веб-программирование: серверные и клиентские технологии. - СПб.: Питер, 2018. - 368 с.
3. Иванов К.П. Проектирование и разработка баз данных. - М.: БИНОМ, 2017. - 512 с.
4. Петров Н.С. Современные методы кэширования данных. - Казань: Университетская книга, 2019. - 290 с.
5. Смирнов В.А. Алгоритмы обработки математических выражений. - Новосибирск: Сибирское научное издательство, 2020. - 340 с.
6. Фёдоров И.Н. Django и современные веб-технологии. - М.: ДМК Пресс, 2021. - 456 с.
7. Сидоров Л.Г. Оптимизация баз данных: теория и практика. - СПб.: Питер, 2016. - 384 с.
8. Кузнецов П.В. Архитектура веб-приложений. - М.: Мир, 2018. - 410 с.
9. Орлов Д.С. PostgreSQL: администрирование и оптимизация. - Екатеринбург: Уральский университет, 2019. - 372 с.
10. Алексеев Ю.М. Современные алгоритмы обработки данных. - Новосибирск: СибАК, 2020. - 298 с.
11. Беляев В.Н. Разработка кроссплатформенных веб-приложений. - СПб.: Питер, 2021. - 435 с.
12. Захаров О.П. Основы JavaScript и клиентского программирования. - М.: Лаборатория знаний, 2017. - 320 с.
13. Тимофеев С.Р. Введение в машинное обучение для анализа данных. - СПб.: Научная книга, 2022. - 460 с.
14. Морозов А.И. Технологии облачных вычислений. - Казань: Академия, 2020. - 355 с.
15. Ершов Д.А. Компьютерная алгебра и символьные вычисления. - Новосибирск: Сибирское научное издательство, 2021. - 312 с.

Приложение А Справка о проверке на наличие заимствований

НАО «Карагандинский технический университет имени Абылкаса Сагинова»

Кафедра «Информационно-вычислительные системы»

Данная справка свидетельствует, что студент группы ИС-22-2с, Алпыспаев Тамирлан, прошел антиплагиат на курсовую работу по дисциплине «Разработка информационных систем» с процентом оригинальности %.



Студент гр.ИС-22-2с \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алпыспаев Т.А.

И.О. Доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мутовина Н.В.

Зав. Кафедры ИВС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Калинин А.А.

Приложение Б Листинг программного кода

Листинг функции генерации задач с уровнями сложности

<body class="container">

    <h1>Генератор случайных задач</h1>

    <label for="difficulty">Выберите сложность:</label>

    <select id="difficulty">

        <option value="easy">Легкий</option>

        <option value="medium">Средний</option>

        <option value="hard">Сложный</option>

    </select>

    <button onclick="generateProblem()">Новая задача</button>

    <h2 id="problem"></h2>

    <input type="number" id="answer" placeholder="Введите ответ" onkeypress="checkEnter(event)">

    <p id="result"></p>

    <script>

        let correctAnswer;

        function generateProblem() {

            let difficulty = document.getElementById("difficulty").value;

            let num1, num2, operator;

            if (difficulty === "easy") {

                num1 = Math.floor(Math.random() \* 10) + 1;

                num2 = Math.floor(Math.random() \* 10) + 1;

                operator = "+";

                correctAnswer = num1 + num2;

            } else if (difficulty === "medium") {

                num1 = Math.floor(Math.random() \* 50) + 1;

                num2 = Math.floor(Math.random() \* 50) + 1;

                operator = Math.random() > 0.5 ? "+" : "-";

                correctAnswer = operator === "+" ? num1 + num2 : num1 - num2;

            } else {

                num1 = Math.floor(Math.random() \* 100) + 1;

                num2 = Math.floor(Math.random() \* 10) + 1;

                operator = Math.random() > 0.5 ? "\*" : "/";

                correctAnswer = operator === "\*" ? num1 \* num2 : Math.floor(num1 / num2);

            }

            document.getElementById("problem").textContent = `${num1} ${operator} ${num2} = ?`;

            document.getElementById("answer").value = "";

            document.getElementById("result").textContent = "";

        }

        function checkAnswer() {

            let userAnswer = parseInt(document.getElementById("answer").value);

            if (userAnswer === correctAnswer) {

                document.getElementById("result").textContent = "Правильно!";

                setTimeout(generateProblem, 1000);

            } else {

                document.getElementById("result").textContent = "Неправильно. Попробуйте снова!";

            }

        }

        function checkEnter(event) {

            if (event.key === "Enter") {

                checkAnswer();

            }

        }

        generateProblem();

    </script>

    <div class="nav-links"><a href="/">Домой</a></div>

</body>