**Государственное бюджетное образовательное учреждение «Школа №924»**

Проектная работа

«Аналитическая система генерации и адаптации экзаменационных заданий»

Выполнил: Ученик 11Л класса

Вершинин Сергей Алексеевич,

Научный руководитель:

Дарсавелидзе Алекандр Александрович,

учитель информатики

Москва, 2024 г.

# Оглавление

[Оглавление 2](#_Toc159276782)

[Введение 3](#_Toc159276783)

[Цель и задачи проекта 5](#_Toc159276784)

[Методика выполнения работы 6](#_Toc159276785)

[Результаты и обсуждение 11](#_Toc159276786)

[Описание завершенного продукта 15](#_Toc159276787)

[Список литературы 16](#_Toc159276788)

[Приложения 17](#_Toc159276789)

# Введение

В наше время, когда процесс обучения становится все более индивидуализированным, создание эффективных инструментов для помощи учителям в подготовке учащихся к экзаменам становится крайне актуальной задачей. С развитием веб-технологий все больше появляется онлайн-платформ, предоставляющих доступ к базам актуальных заданий, демонстрационных вариантов и пробных экзаменов. Однако, основной функционал этих ресурсов ограничивается просмотром задач, решений и ответов, а также генерацией случайных вариантов. Сгенерированные на сайтах, варианты не имеют единую медиану сложности заданий, из-за чего некоторые варианты получаются слишком не консистентными по уровню сложности, что требует высокого текущего уровня подготовки, даже при решении несложного варианта. Попытки учителей использовать такие сайты в процессе обучения часто сталкиваются с рядом проблем.

С одной стороны, сбор и составление разнообразных вариантов тестов или контрольных занимает значительное количество времени. С другой стороны, доступность заданий в открытом доступе приводит к тому, что учащиеся могут легко находить ответы в интернете, что затрудняет объективную оценку их знаний. Более того, при генерации вариантов могут возникать неравные по сложности задания, что делает процесс выставления оценок несправедливым и недостаточно объективным.

Размышляя над этими проблемами, мы пришли к выводу, что необходимо разработать веб-приложение, способное статистически рассчитывать сложность заданий, присваивать им соответствующий уровень, и генерировать варианты с сопоставимой сложностью. Такой подход позволит учителям индивидуализировать подход к каждому ученику, что часто оказывается невозможным в условиях общеобразовательных учреждений из-за большого числа учащихся с разным уровнем подготовки.

Актуальные проблемы:

* Неравномерность сложности заданий на сайтах для подготовки к экзаменам.
* Весьма ограниченный функционал telegram бота или его отсутствие
* Плохая отказоустойчивость из-за отсутствия брокера сообщений.

# Цель и задачи проекта

**Цель проекта** – разработка собственного веб-приложения и telegram бота для подготовки к экзаменам с учетом сложности заданий от времени и правильности выполнения.

**Задачи проекта:**

* Поиск и анализ нового материала по теме.
* Разработка формулы для расчета сложности каждого задания от времени и правильности выполнения конкретного задания.
* Создание веб-приложения.
* Разработка и подключение Telegram бота.
* Тестирование проекта на учащихся школы.
* Получение обратной связи от учащихся, корректировка продукта.
* Создание сервиса-аналитики.
* Выложить веб-приложение и Telegram бота на хостинг.

# Методика выполнения работы

**Создание концепции проекта**

Разработку проекта было решено начать с выбора языка программирования. Python был выбран для разработки веб-приложения из-за своей простоты и легкой читаемости кода, большого сообщества разработчиков, множества библиотек и фреймворков, скорости разработки, поддержки облачных вычислений, интеграции с другими технологиями и совместимости с различными операционными системами.

После определения языка программирования, была начата дискуссия по поводу выбора фреймворка, на котором будет написано веб-приложение. Это было гораздо сложнее, так как фреймворков существует немало, и все они поставляют примерно одинаковый функционал, но все же было решено выбрать Django. Помог сделать этот нелегкий выбор тот факт, что Django поставляет "из коробки" тот функционал, который пришлось бы реализовывать вручную с другими фреймворками. Например: администрирование базы данных, простое и понятное составление моделей, интуитивное обращение к записям в базе данных и отсутствие необходимости прямой работы с SQL-запросами. Телеграмм-бот было решено реализовать на библиотеке aiogram. Это одна из наиболее популярных библиотек в наше время. Использующая такую технологию как планирование I/O-задач - async. Данный бот должен играть роль notification-сервиса, реализовывать функционал вывода заданий, а также иметь функционал просмотра статистики по аккаунту юзера и иметь связь с сервисом аналитики.

**План выполнения проекта**

На выполнение проекта было отведено примерно 2 месяца. Подробный план проекта представлен в Таблице 1.

Таблица 1. План выполнения работы.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Действие | Стоимость | Затраченное время |  |  |  |  |
| 1 | Поиск и анализ аналогов | 0 руб. | 5 час |  |  |  |  |
| 2 | Изучение новых фреймворков | 0 руб. | 10 дней |  |  |  |  |
| 3 | Разработка собственного веб-приложения | 0 руб. | 15 дней |  |  |  |  |
| 4 | Создание и тестирование формулы подсчета сложности | 0 руб. | 4 дня |  |  |  |  |
| 5 | Разработка Telegram бота | 0 руб. | 3 дня |  |  |  |  |
| 6 | Подключение бота к веб-приложению | 0 руб. | 2 дня |  |  |  |  |
| 7 | Тестирование программного комплекса на учащихся школы №924 и получение обратной связи | 0 руб. | 10 дней |  |  |  |  |
| 8 | Внесение корректировок | 0 руб. | 5 дней |  |  |  |  |
| 9 | Аренда сервера и домена | 320 руб. | 1 час |  |  | 0 руб | 2 часа |
| 10 | Перенос программного комплекса на хостинг и подключение к домену | 0 руб. | 6 часа |  |  |  |  |
| 11 | Конечное тестирование | 0 руб. | 1 день |  |  |  |  |
| 12 | Создание проектной документации и презентации | 0 руб. | 5 дней |  |  |  |  |

**Реализация плана, корректировки**

В процессе планирования нашего проекта была принято решение реализовать архитектуру таким образом:

Сервисы:

* Телеграмм-бот
* веб приложение
* Сервис аналитики

Вспомогательные компоненты:

* Брокер сообщений – Kafka
* Скрипт-посредник между брокером сообщений и сервисом

Все сервисы не должны общаться напрямую (см. рис. 1). Они должны отсылать запрос вспомогательному компоненту-посреднику, он должен класть сообщение в брокера, потребитель в лице компонента-посредника другого сервиса должен приходить к брокеру, брать сообщение и передавать своему сервису.

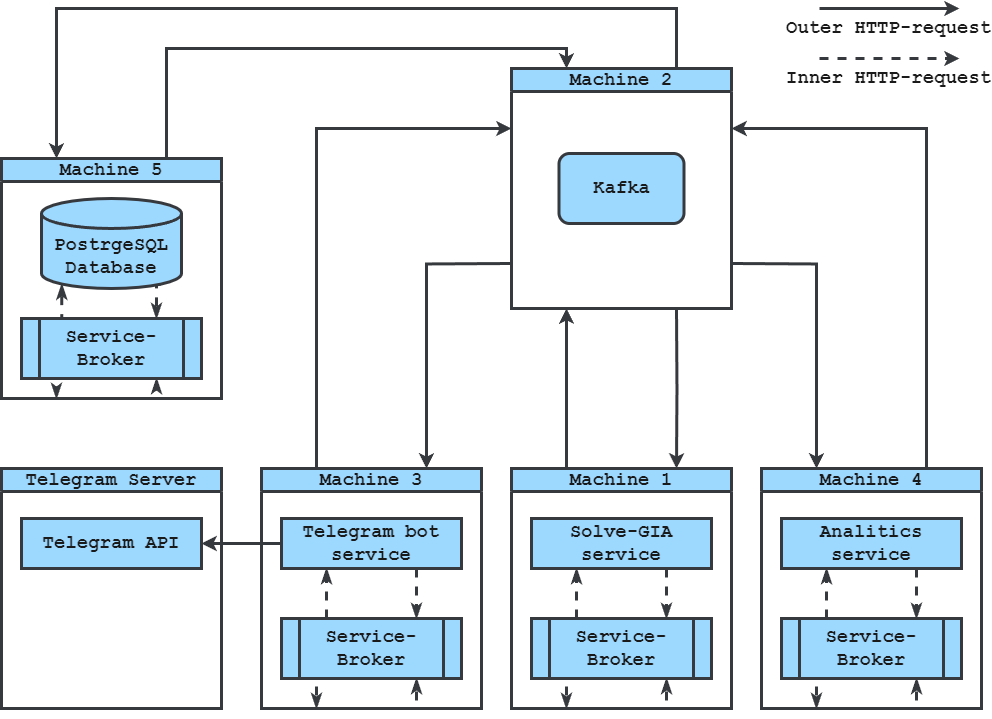


Рисунок 1. Схема работы Kafka

Таким образом, брокер сообщений служит прослойкой между 3-мя независимыми сервисами. Это дает, в первую очередь, отказоустойчивость системы. При выходе одного сервиса из строя - остальные продолжат работу в штатном режиме. В качестве брокера сообщения используется Kafka, схему работы представлена на рисунке 2. Также брокер сообщений поможет справиться даже с очень большой нагрузкой на сервер, т.к. написанный на scala, он имеет невероятную пропускную способность - около миллиона запросов в секунду. Помимо уже сказанного, брокер сообщений, также может обеспечить возможность горизонтального масштабирования, в будущем, посредством равномерного распределения нагрузки между несколькими независимыми машинами.

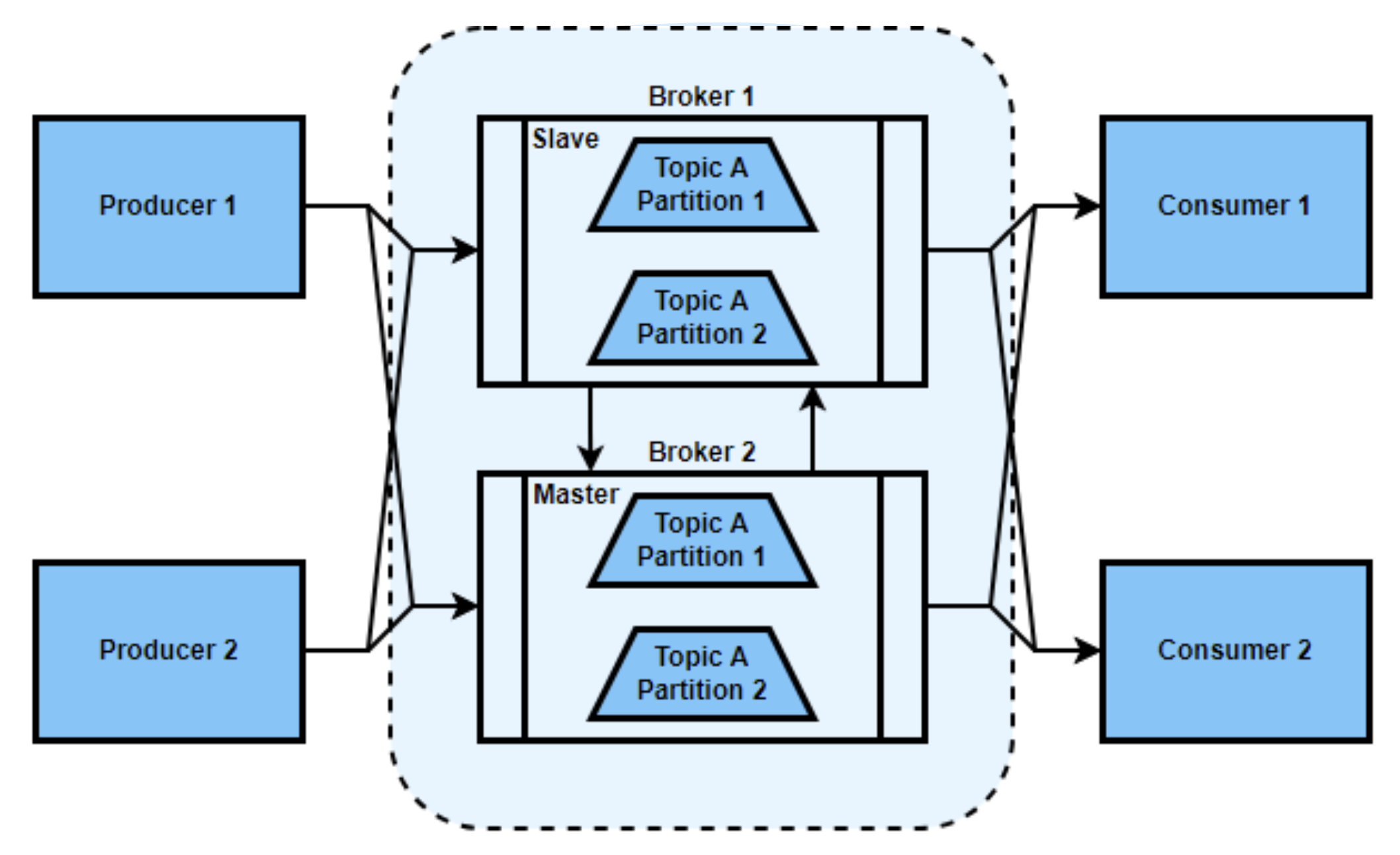


Рисунок 2. Схема работы брокера сообщений

В качестве основного сервиса, в нашей архитектуре, выступает непосредственно веб-приложение. Данный продукт написан на языке python с использованием таких библиотек как Django, Jinja2, Pillow. Изначально, выбор пал на такой фреймворк как Flask, однако, в дальнейшем было принято решение мигрировать проект на фреймворк Django по причине отсутствия необходимости выдумывать собственную внутреннюю архитектуру веб-приложения. Также к несомненным превосходствам Django относятся batteries-included-фича, такие как: превосходная надстройка над SQL (см. рис. 3) - Django ORM, шаблонизатор основанный на Jinja2, система отладки, генератор паттерна архитектуры и прочее.

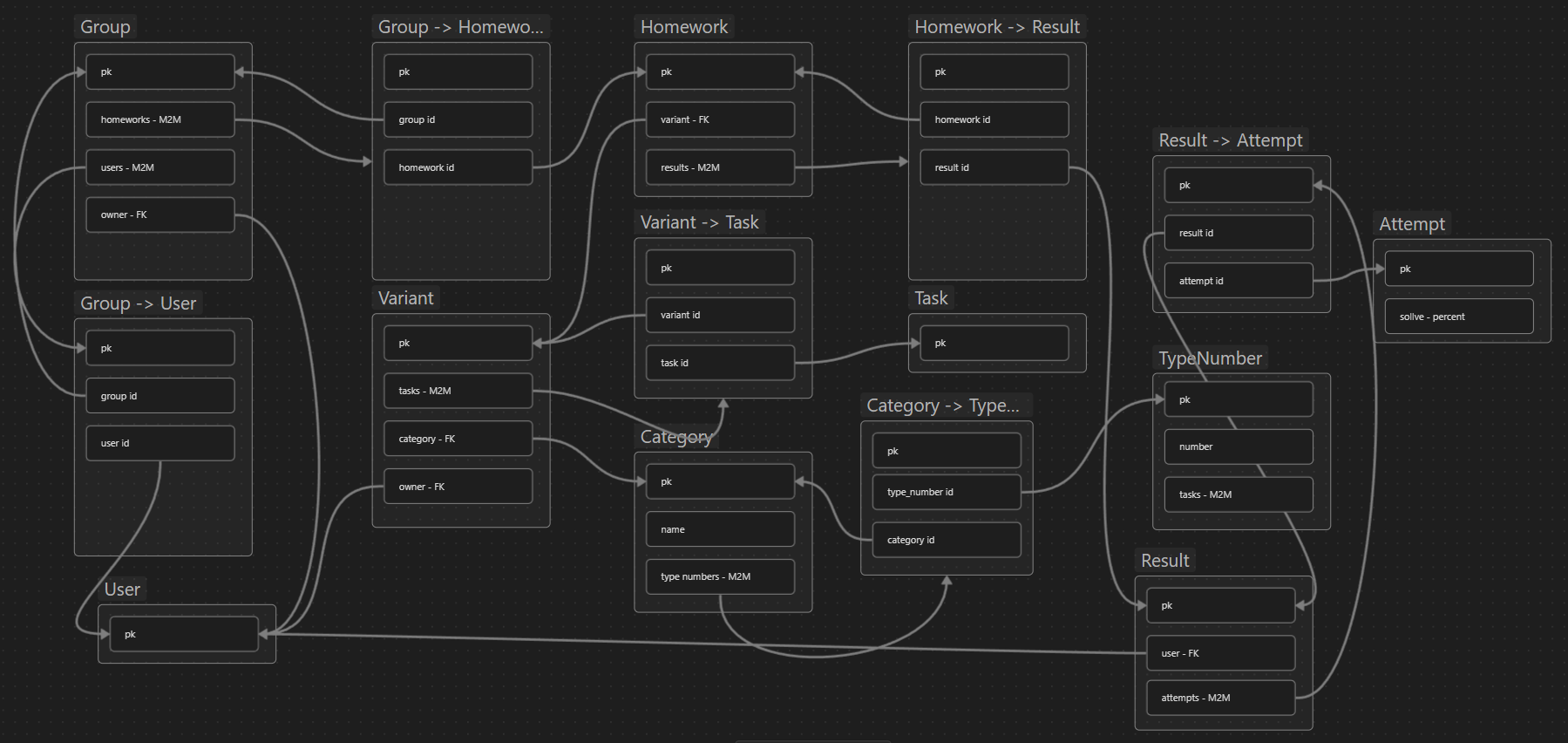


Рисунок 3. Схема связей в базе данных

Второстепенными сервисами в данной связке являются телеграмм-бот и сервис аналитики.

Телеграмм-бот было решено реализовать на библиотеке aiogram. Это одна из наиболее популярных библиотек в наше время. Использующая такую технологию как планирование I/O-задач - async. Данный бот должен играть роль notification-сервиса, реализовывать функционал вывода заданий, а также иметь функционал просмотра статистики по аккаунту юзера и иметь связь с сервисом аналитики.

Сервис аналитики — это сервис, предоставляющий пользователю множество различных данных по его аккаунту в структурированном виде. Предположительно в виде графиков. Также планируется интеграция нейронной сети, помогающей пользователю выявить слабые места и выделить моменты, которым стоит уделить особое внимание при подготовке. Данный сервис должен реализовывать как web-GUI, так и API для обмена данными между сервисами.

Также на каждой машине должен быть развернут скрипт, реализующий посредника между брокером сообщений и сервисом посредством HTTP-соединений. Он будет слушать на глобальном адресе, преобразуя команды из брокера сообщений в HTTP-запросы к gateway на локальный адрес сервиса.

# Результаты и обсуждение

Таким образом, в ходе разработки был создан продукт для подготовки к государственной итоговой аттестации с динамической медианой сложности (см. рис. 8), зависимой от количества людей, прошедших то или иное задание, времени, затраченного на выполнение задания, правильности выполнения. В качестве дополнительных сервисов был разработан telegram-bot и будет разработан сервис аналитики для получения важной информации о ученике и его результатах по тому или иному предмету. Благодаря использованию формулы для вычисления динамической сложности, учителя могут составлять варианты с единой медианой сложности для любого количества учеников, а обучающиеся в свою очередь могут сосредоточиться на тех темах, которые вызывают у них наибольшие трудности, и улучшить свои знания в этих областях.  
После проведенной работы было проведено тестирование веб-приложения на учителях информатики и учащихся ГБОУ школы №924, в ходе которого собиралась важная информация о недостатках и возможных улучшениях нашего сайта, а также общая оценка нашего продукта. В результате опроса мы получили некоторые данные: 70% опрошенных учителей считают, что сайт решает поставленные задачи и имеет удобный и понятный интерфейс, 20% считают, что проект нуждается в доработке, 10% воздержались от оценивания веб-приложения (см. рис. 4). Все учащиеся отметили, что проектная работа решает проблему с вариантами разной сложности и очень нужна для процесса обучения. 83% опрошенных отметили, что интерфейс веб-приложения достаточно удобен для использования в качестве образовательной платформы (см. рис. 5). 74% указали, что функционала сайта хватает и 26%, что сайт нуждается в большем количестве возможностей (см. рис. 6). Тем не менее, продолжать пользоваться ботом собираются примерно 79% людей (см. рис. 7).

В целом, веб-приложение для подготовки к государственным аттестационным работам с формулой динамической сложности является полезным инструментом для учителей и учащихся, который помогает им улучшить свои знания и успешно подготовиться к экзаменам.

Однако, несмотря на достигнутые результаты, есть некоторые аспекты, которые можно улучшить в будущем. Например, можно расширить базу данных, чтобы включить больше материалов для подготовки к различным дисциплинам. Также можно добавить дополнительные функции, такие как возможность создания индивидуальных планов обучения.

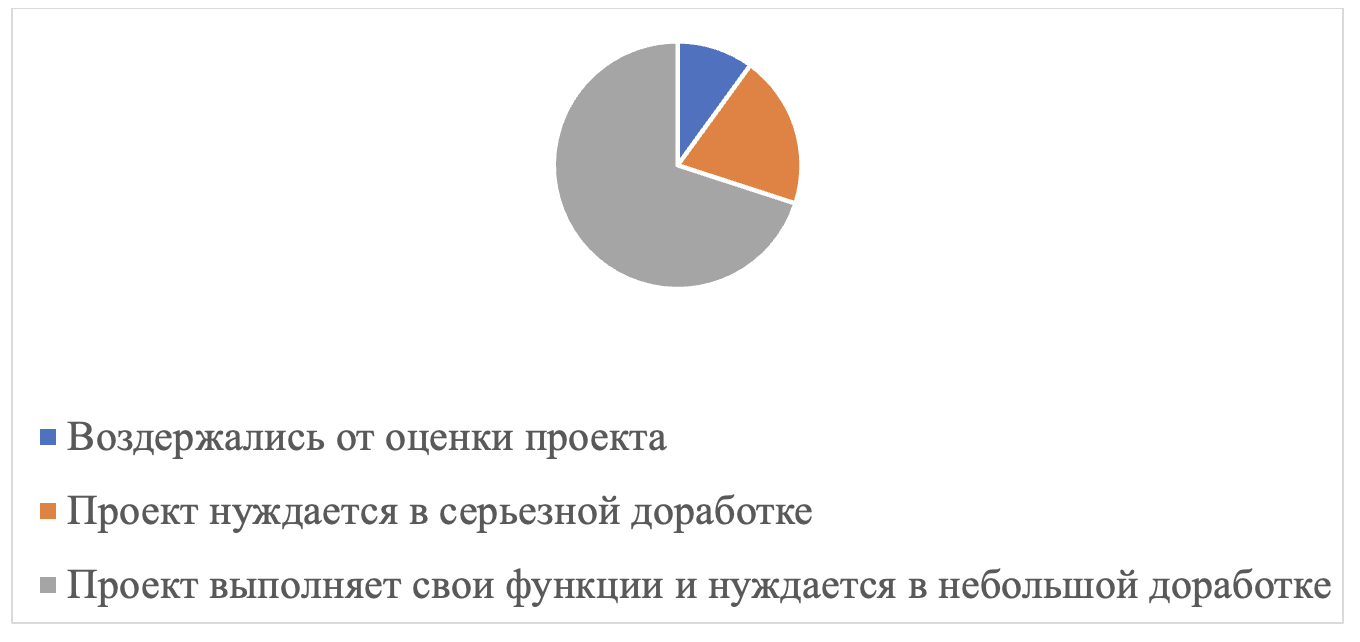


Рисунок 4. Оценка продукта

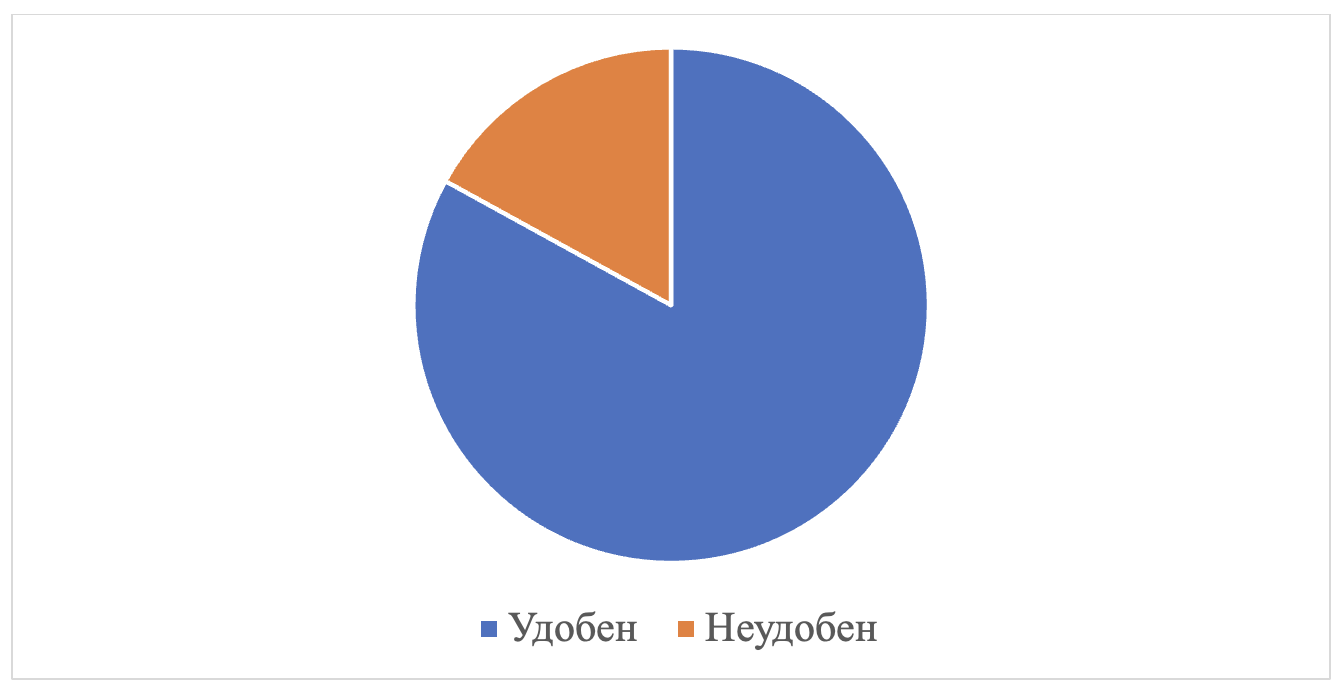


Рисунок 5. Оценка удобства

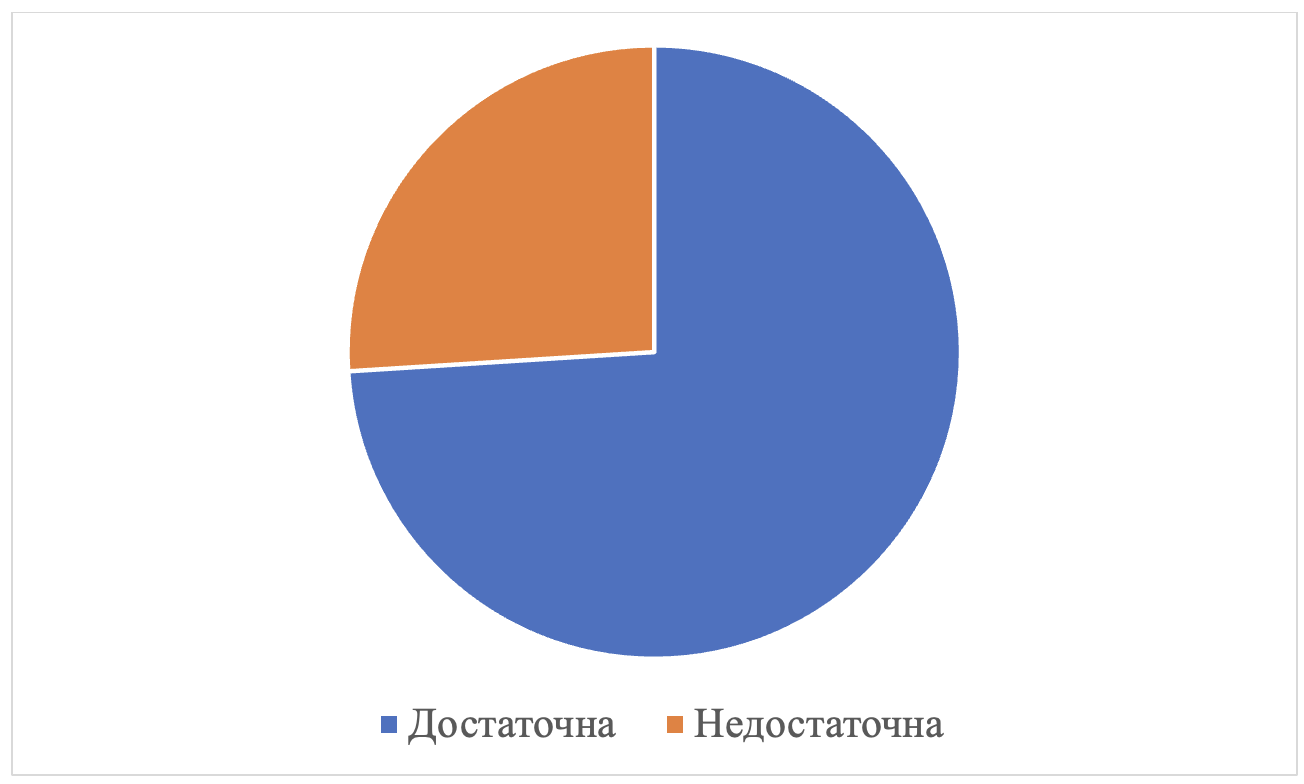


Рисунок 6. Оценка функциональности

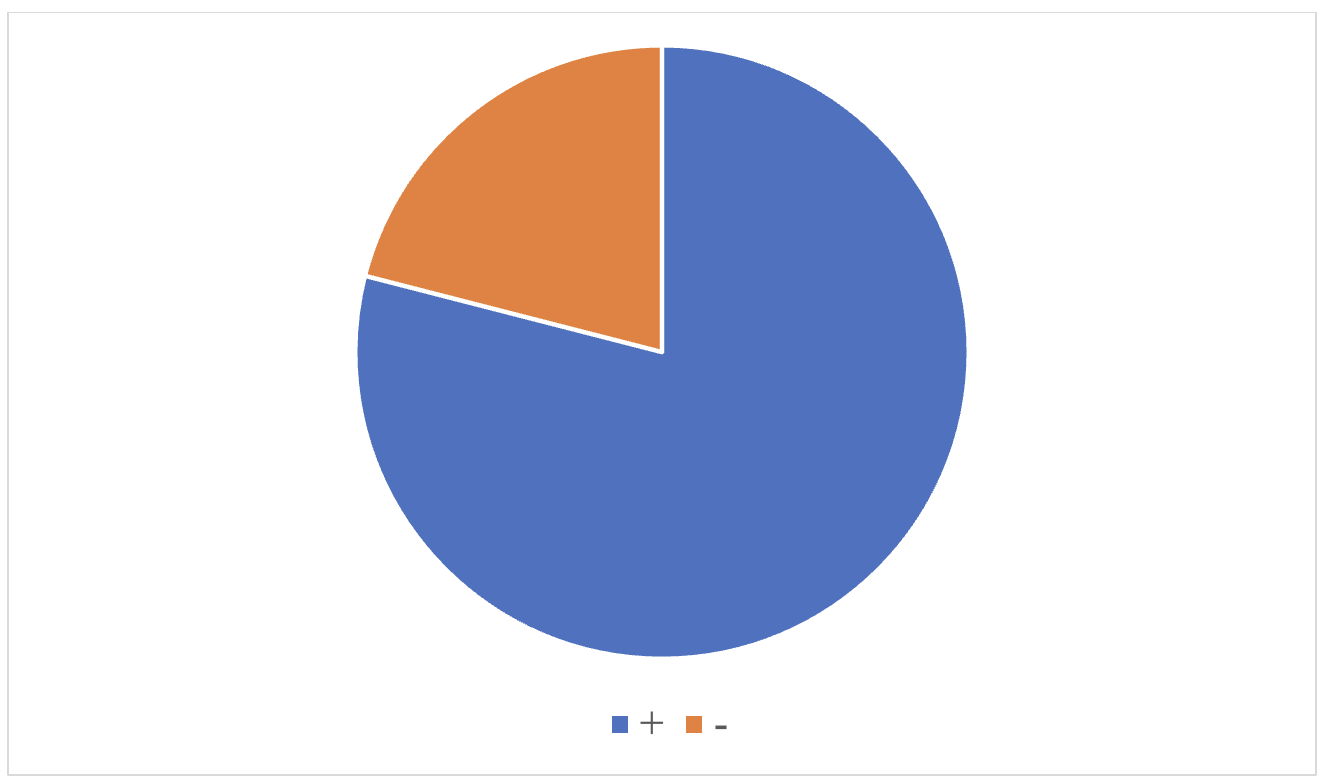


Рисунок 7. Удержание аудитории



Рисунок 8. График сложности варианта

# Описание завершенного продукта

По окончанию работы мы получили продукт - аналитическую систему генерации и адаптации экзаменационных заданий для подготовки к государственным аттестационным работам с формулой вычисления динамической сложности. Приложение предоставляет доступ к обширной базе данных, содержащей материалы для подготовки к различным дисциплинам. Основной особенностью данного проекта является использование формулы вычисления динамической сложности. Эта формула позволяет вычислить наиболее точную сложность каждого задания и адаптировать процесс обучения под индивидуальные потребности студента. Благодаря этому, студенты могут сосредоточиться на тех темах, которые вызывают у них наибольшие трудности, и улучшить свои знания в этих областях. Веб-приложение имеет простой и понятный интерфейс, который позволяет легко ориентироваться в материалах и быстро находить нужную информацию. На сайте реализованы личные кабинеты учащихся, учителей и администрации. Учителя, в свою очередь могут задавать домашние задания учащимся, создавать группы (классы) для проведения тестирований и для более удобного мониторинга учащихся. Личные кабинеты, также позволяют собирать аналитику каждого пользователя, для более правильного процесса обучения, это позволит увидеть динамику в обучении или ее отсутствие.

Для того, чтобы им пользоваться, не нужно скачивать какие-либо сторонние приложения, достаточно иметь любой HTTP-клиент.

# Список литературы

Документация к фреймворку Django [Электронный ресурс].

Ссылка: https://django.fun/docs/django/5.0/

Документация к фреймворку Aiogram [Электронный ресурс].

Ссылка: https://docs.aiogram.dev/en/latest/

Документация к брокеру сообщений Apache Kafka [Электронный ресурс]. Ссылка: https://kafka.apache.org/documentation/

Видео курс по фреймворку Django от автора selfedu [Электронный ресурс]. Ссылка: https://youtube.com/@selfedu\_rus?si=yBrCru\_HBTxcwt6L

Видео курс по фреймворку Django от автора Senior Pomidor Developer [Электронный ресурс]. Ссылка: https://youtube.com/@SeniorPomidorDeveloper?si=X-UGrSyPZ24dUa4R

Видео курс системе управления версиями GIT от автора Backend artist [Электронный ресурс]. Ссылка: https://youtube.com/@backend\_artist?si=KAwxLC\_yYLBjqrmJ

Видео курс по подключению хостинга от автора Диджитализируй! [Электронный ресурс]. Ссылка: https://youtube.com/@backend\_artist?si=KAwxLC\_yYLBjqrmJ

# Приложение А

Файлы и приложения проекта. Ссылка:

<https://disk.yandex.ru/d/tpM99c1S6CL_AA>

Код проекта [Электронный ресурс]. Ссылка:

https://github.com/ImCocos/Solve-GIA-Remake/  
Видео работы аналитической системы генерации и адаптации экзаменационных заданий [Электронный ресурс]. Ссылка:

https://rutube.ru/video/1ffacf62b893d4895a1f0f72b00365d3/