

Университет Правительства Москвы
VIII Конкурс исследовательских и проектных работ обучающихся
образовательных
организаций города Москвы и Московской области
«Мегаполис XXI века – город для жизни» в 2023/2024 учебном году

Конкурсная работа

На тему: «TELEGRAM БОТ ДЛЯ ЭЛЕКТРОННОГО ДНЕВНИКА МЭШ»

Выполнил:

учащийся 11А класса

ГБОУ «Школа №2090»

Кузьмин Вячеслав Сергеевич

Подпись _____

Научные руководители работы:

Балицкая Анастасия Викторовна

Спирин Александр Борисович

Старший методист

ГБОУ «Школа №2090»

Сафонова Оксана Николаевна

Подпись _____

Москва

2023-2024

ОТЗЫВ

на конкурсную работу

Тема Telegram бот для электронного дневника МЕШ

Работу выполнил: Кузьмин Вячеслав Ученицы 11А класса ГБОУ «Школа №2090»

1 Объем выпускной квалификационной работы страниц, в т. ч.

а) количество страниц практической (расчетно-аналитической) части 18

б) количество страниц приложений нет

2 Оценка содержания работы, его положительные стороны и недостатки, выводы и предложения Автор работы показал, что тема бота, помогающего в работе системой «Электронной школы» необходима и актуальна.

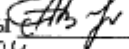

Структура и содержание работы соответствует теме. Работа состоит из четыре основных частей. Во введении определены актуальность темы, цель и задачи, решаемые в работе. Теоретическая часть посвящена анализу аналогичны решений и исследованию уже готовых ботов. Так же в теоретической части проанализированы сервисы «Электронной школы». Раскрытие теоретических аспектов темы проведено четко и полно, выявлены современные методы, необходимые для реализации поставленной задачи. В практической части работы был проведен анализ сайта и мобильного приложения, на основании которого было выбрано несколько вариантов достижения подавленной цели. Из этих вариантов был написан один бот, включающий все необходимы действия для работы с сервисом «Электронной школы». Так же была учтена и система безопасности личных данных учеников.

Практическая значимость полученных результатов показывает, удобство использования бота в работе «Электронным дневником», что показывает опрос пользователей данного бота.

При выполнении работы ученик проявил самостоятельность и организованность, работы выполнена в срок в полном объеме все поставленные задачи были решены.

Оформление работы в целом соответствуют требованиям ГОСТов.

Цель работы реализована.

Научные руководители работы  /А.В. Балицкая  /А.Б.Спирин/
«29» февраля 2024г.

СОДЕРЖАНИЕ

1. Терминология	4
-----------------------	---

2. Существующие аналоги.....	5
3. Ход работы.....	6
3.1. Выбор средств для реализации проекта.....	6
3.2. Инженерный анализ сайта и мобильного приложения МЭШ.	7
3.3. Микро-сервис 1 - Monitoring Package (Сервис мониторинга)	7
3.4. Микро-сервис 2 - HTTP REST API (Backend)	8
3.5. Микро-сервис 3 - Telegram бот	9
3.6. Система групп	13
3.7. Система цифровых учебных материалов	15
3.8. Авторизация через аккаунт МЭШ	16
3.9. Отказоустойчивость системы	17
3.10. Создание веб-сайта проекта	18
3.11. Страница статуса сервисов	22
3.12. Безопасность данных	23
4. Заключение.....	23

ВВЕДЕНИЕ

Сегодня чат-боты стали очень популярны, они помогают пользователям выполнять типичные рутинные действия в автоматическом режиме, значительно упрощая жизнь. Было принято решение создать чат-бота в мессенджере Telegram, который упрощал бы жизнь московских школьников.

Цель проекта – создать чат-бота для электронного дневника МЭШ в Telegram. Проект облегчит и ускорит работу школьников с электронным дневником. Работа с сервисом будет происходить внутри мессенджера Telegram, что позволяет избежать установки дополнительных приложений. Также бот имеет собственную базу данных, которая регулярно синхронизируется с дневником МЭШ, что позволит использовать его в любое время, даже когда сервисы МЭШ недоступны. Также бот имеет возможность загрузки электронных учебников, что позволяет ученикам находить их за пару нажатий.

Задачи проекта:

- Поиск и анализ существующих аналогов;
- Исследование необходимой литературы по теме;
- Определение плана, описания и целей проекта;
- Инженерный анализ сайта и мобильного приложения Дневника МЭШ (Анализ сетевых запросов);
- Извлечение конечных точек API, схемы запросов;
- Выбор технических средств, библиотек, языка программирования для реализации проекта;
- Написание кода Monitoring Package;
- Написание кода Bot Package;
- Написание кода Web Package;
- Написание кода Веб-сайта проекта;
- Тестирование бота;
- Развертывание проекта на сервере для полноценного запуска.

1. ТЕРМИНОЛОГИЯ

- Monitoring Package – Сервис, который занимается опросом (мониторингом) Дневника МЭШ через API и сохраняет полученные данные (Например, оценки ученика) в локальную базу данных.
- Bot Package – Сервис, отвечающий за работу самого Telegram бота и взаимодействие с пользователем. Преобразует данные из локальной базы данных в красивый интерфейс.
- Web Package – Сервис, необходимый для авторизации пользователей через аккаунт МЭШ в боте. В будущем может быть использован для интеграции с другими проектами, создания web версии проекта, страницы статуса сервисов.
- API – набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.
- Хэш-функция - Функция, осуществляющая преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины, выполняемое определённым алгоритмом.

2. СУЩЕСТВУЮЩИЕ АНАЛОГИ

2.1. Официальный навык «Дневник МЭШ» для Яндекс Алисы

Навык голосового Ассистента Алиса, позволяющий пользоваться дневником с помощью голосовых запросов.

Недостатки:

- Работает только в приложениях Яндекса и только с голосовым ассистентом Алиса, аудитория которого в несколько раз меньше.
- Нацелен в основном на голосовое управление, что не всегда удобно
- Имеет более ограниченный функционал в сравнении с ботом
- Нет возможности отправки уведомлений о новых оценках
- Не отображает вложенные файлы в домашних заданиях

2.2. Неофициальный Telegram бот «Школьный дневник»

Аналогичный Telegram бот, работающий с несколькими системами дневников.

Недостатки:

- Авторизация в аккаунте дневника (ввод логина и пароля от mos.ru) происходит на стороннем сайте, что не гарантирует безопасность
- Более скромный функционал
- Отсутствие обновлений

3. ХОД РАБОТЫ

3.1. Выбор средств для реализации проекта

В качестве основного языка программирования был использован JavaScript (NodeJS), ввиду высокого уровня владения им и удобства использования. Программная составляющая проекта состоит из 3 микро-сервисов: Сам Telegram бот, HTTP REST API Сервер (для авторизации пользователей и прочих веб сервисов), сервис мониторинга (синхронизирует данные из МЭШ с локальной базой данных).

В качестве основной базы данных была выбрана MySQL. Для хранения временных данных (кэша) – Redis. Для хранения и работы с данными о цифровых учебных материалах была использована поисковая система Elasticsearch, так как её удобно применить для поиска учебников по названию, автору, классу.

Все сервисы проекта разворачиваются на сервере под управлением ОС Ubuntu с использованием Docker контейнеров.

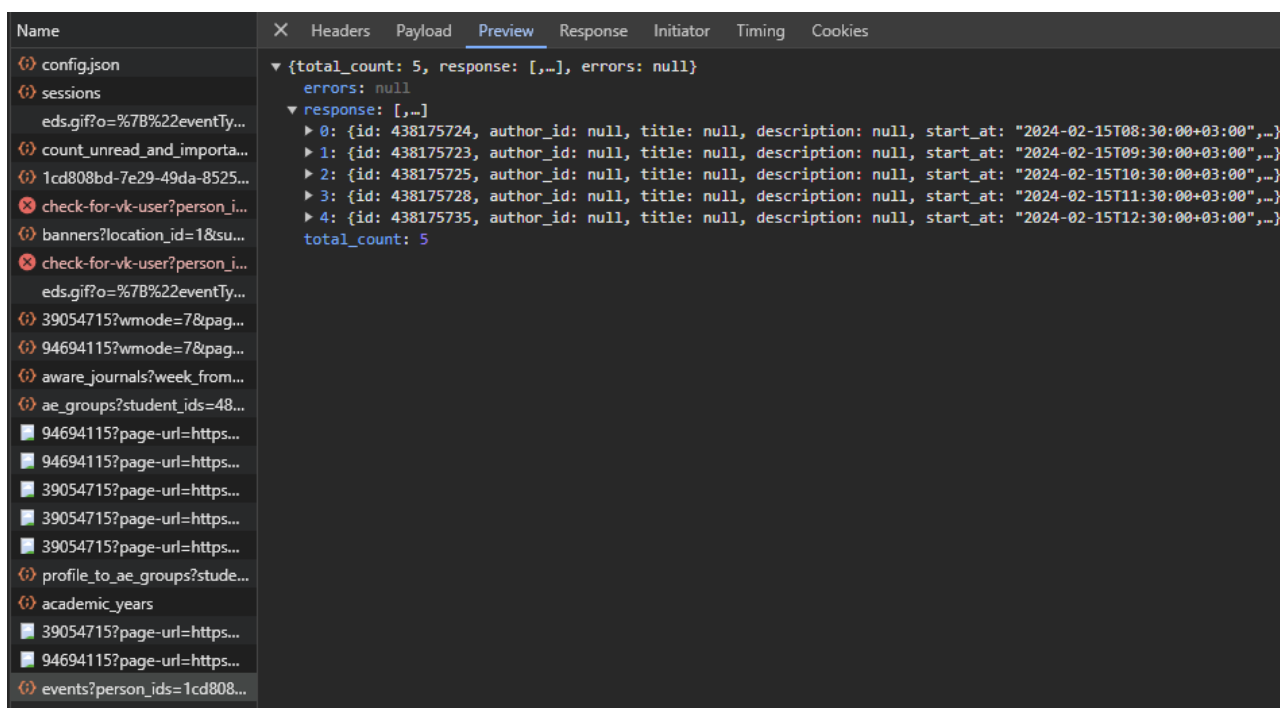
Основные библиотеки и ПО для проекта:

- СУБД MySQL (+ Библиотека Node JS «mysql2»)
- Кэш-хранилище Redis (+ Библиотека Node JS «redis»)
- Библиотека Axios (для выполнения HTTP запросов)
- Библиотека Telegraf JS (Для создания Telegram бота)
- Forever JS (Для запуска проекта в режиме процесса)
- ОС Ubuntu 22
- Node JS 20
- Docker (Сервис для развертывания проектов в виде автономных переносимых контейнеров)
- Библиотека Crypto (для работы с хэшами и шифрованием)
- Elasticsearch – поисковая система для эл. Учебников

- WEB фреймворк ExressJS
- Собственная Python программа для оптимизации цифровых учебников
- Wireguard VPN для организации защищенного канала между сервером проекта и компьютером, на котором ведется разработка или отладка проекта

3.2. Инженерный анализ сайта и мобильного приложения МЭШ

С помощью инструментов разработчика в браузере, а также, ПО «HTTP Toolkit» был проанализирован сетевой трафик официального приложения и сайта МЭШ. Были извлечены конечные точки API МЭШ. Были изучены запросы, извлечена схема тела и заголовков запросов. Конечная точка API – это путь страницы на сайте, куда можно отправить запрос и получить ответ в JSON формате (Например: www.test.ru/api/account).



3.3. Микро-сервис 1 - Monitoring Package (Сервис мониторинга)

Работа над проектом началась с идеи сделать такой сервис, который позволит просматривать домашние задания в то время, когда сервисы МЭШ недоступны или работают с большими сбоями (довольно острая проблема во времена пандемии коронавируса). На Node.js был написан скрипт, который отправляет HTTP запросы на полученные ранее точки API, а в ответ получает список уроков на заданные даты и задания к ним. Затем эти данные сохранялись в локальную MySQL базу данных, из которой их уже можно было извлекать без всяких сложностей.

Для осуществления любых запросов к сайту дневника необходимо пройти авторизацию (войти в аккаунт), после этого в браузере пользователя сохраняется ключ доступа (JWT токен), с помощью которого сервисы МЭШ идентифицируют пользователя при каждом запросе. Написанный скрипт извлекает заранее сохраненный JWT токен из локальной базы данных и отправляет его в заголовке каждого запроса. С точки зрения сайта МЭШ – это был обычный пользователь, который решил обновить страницу дневника и запросить данные.

Скрипт постепенно обрастал функционалом, была добавлена возможность получения из дневника оценок, вложений в домашних заданиях, рейтинга ученика (и других учеников) в классе.

3.4. Микро-сервис 2 - HTTP REST API (Backend)

Этот сервер используется для реализации функции авторизации через МЭШ и реализации сервиса электронных учебников. В будущем планируется добавление своего публичного API сервера, который позволит открыть доступ к закрытому API МЭШ для сторонних разработчиков.

В качестве основного фреймворка и веб сервера используется Express JS. Веб сервер отвечает за обработку и сохранение JWT токенов авторизации, полученных от пользователя. Также сервер отвечает за работу с электронными учебными материалами: сохраняет последнюю открытую страницу, закладки, примечания.

API сервер для сторонних разработчиков. Например, вы хотите интегрировать функции дневника МЭШ в свой проект, такой как мобильное приложение, сайт, систему умного дома или вообще вывести расписание уроков на свой телевизор. Официально в МЭШ нет никаких инструментов и поддержки для сторонних разработчиков, так что сделать это будет сложно. Сервис «Telegram бот для электронного дневника МЭШ» использует отлаженные схемы и инфраструктуру для обращения к закрытому API дневника, а также использует собственные разработки и занимается поддержкой их работоспособности. Было сочтено логичным дать доступ и сторонним разработчикам, чтобы они могли, воспользовавшись данными разработками, легко интегрировать МЭШ в свои сервисы и создавать новые полезные проекты.

3.5. Микро-сервис 3 - Telegram бот

В качестве библиотеки для работы с Telegram была выбрана Telegraf JS, она является лучшей и самой полноценной на сегодняшний день. В боте за каждым пользователем закреплена сессия, хранящаяся в кэш-хранилище сервера.

При первом запуске бота создается запись в MySQL базе данных, где сохраняется базовая информация о пользователе и внутренний идентификатор в системе. Если пользователь перешел в бот по ссылке приглашения, то он автоматически подключается к группе, в ином случае, пользователю будет предложено войти в бот через аккаунт МЭШ или ввести код приглашения вручную.

Реализовано интуитивное главное меню, из которого можно легко попасть во все разделы бота:

- Оценки по дням или средний балл по предметам
- Расписание уроков
- Домашние задания
- Настройки профиля
- Приглашение друзей
- Информация о боте
- Рейтинг ученика по классу
- Сервис цифровых учебников

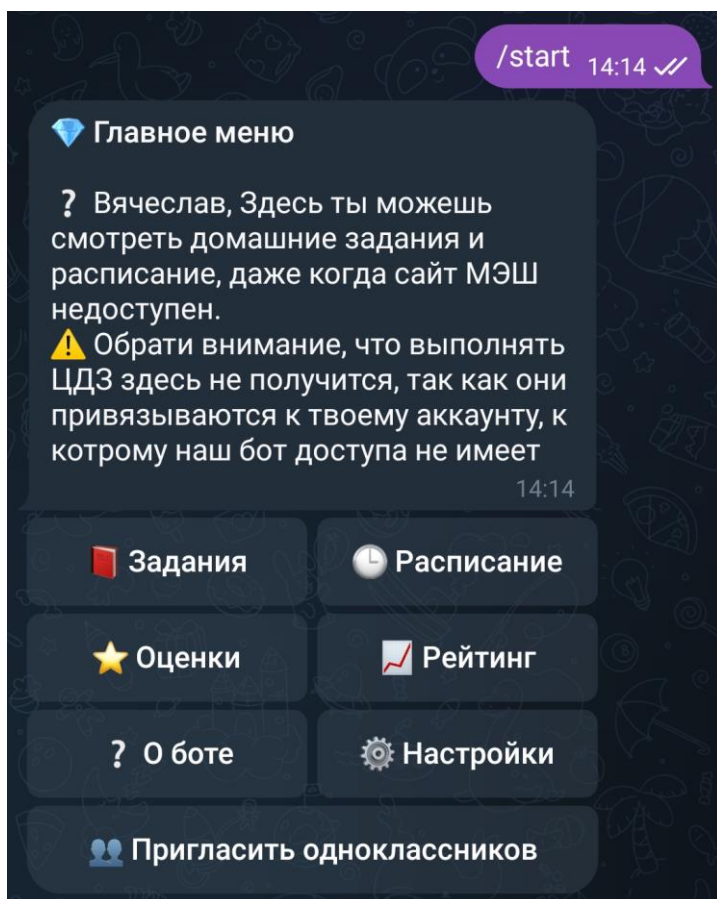


Рис. 3.5.1. Главное меню бота

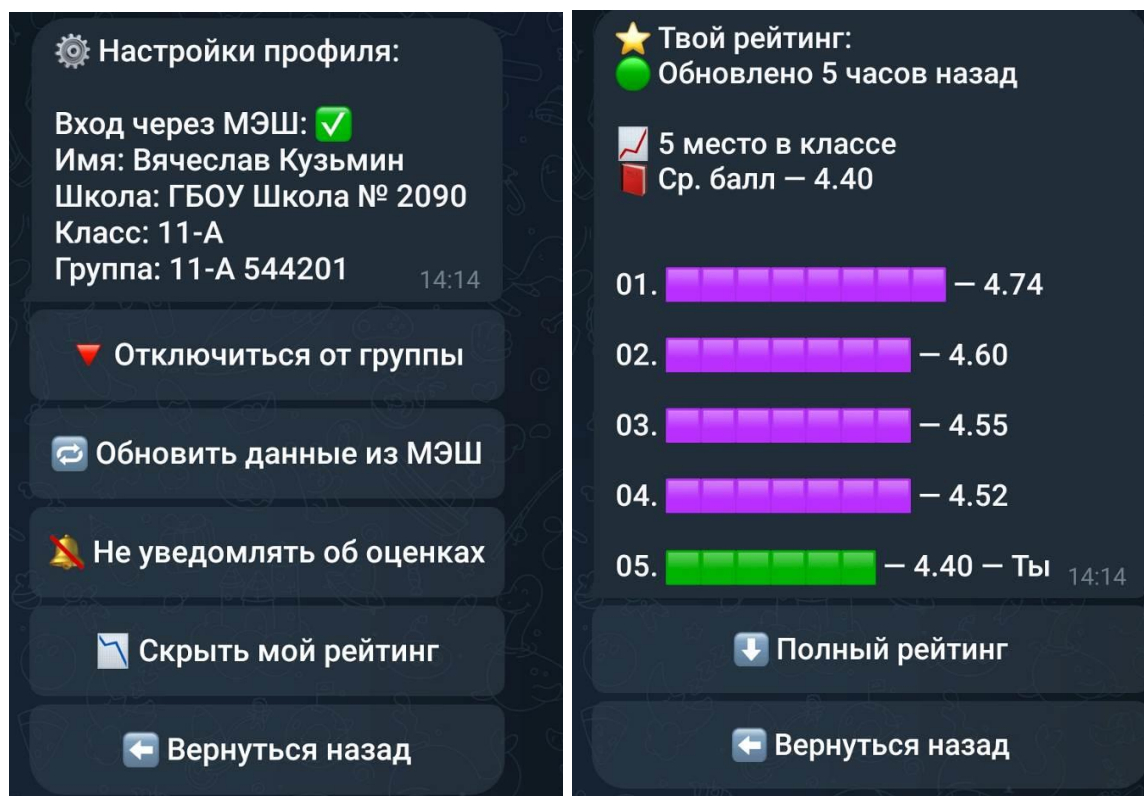


Рис. 3.5.2. Разделы настроек профиля и рейтинга по классу

В настройках профиля можно вручную синхронизировать данные из МЭШ, отключиться от группы (класса), управлять уведомлениями о новых оценках, скрыть или показать свое имя в рейтинге класса.

Рейтинг теперь не анонимный. Функция просмотра рейтинга по классу была улучшена, в сравнении с той, что предлагает сама МЭШ. В рейтинге теперь можно увидеть имена других учеников, которые так же авторизовались в боте через аккаунт МЭШ и не скрыли в настройках свое имя в рейтинге (по умолчанию отображение имени включено). Многие ученики хотят сравнить свой прогресс с прогрессом своих конкретных одноклассников.

В рейтинге за каждой позицией закреплен обезличенный идентификатор ученика. Бот ищет эти идентификаторы в локальной базе пользователей сервиса и отображает имена найденных учеников, если они разрешили это в настройках.

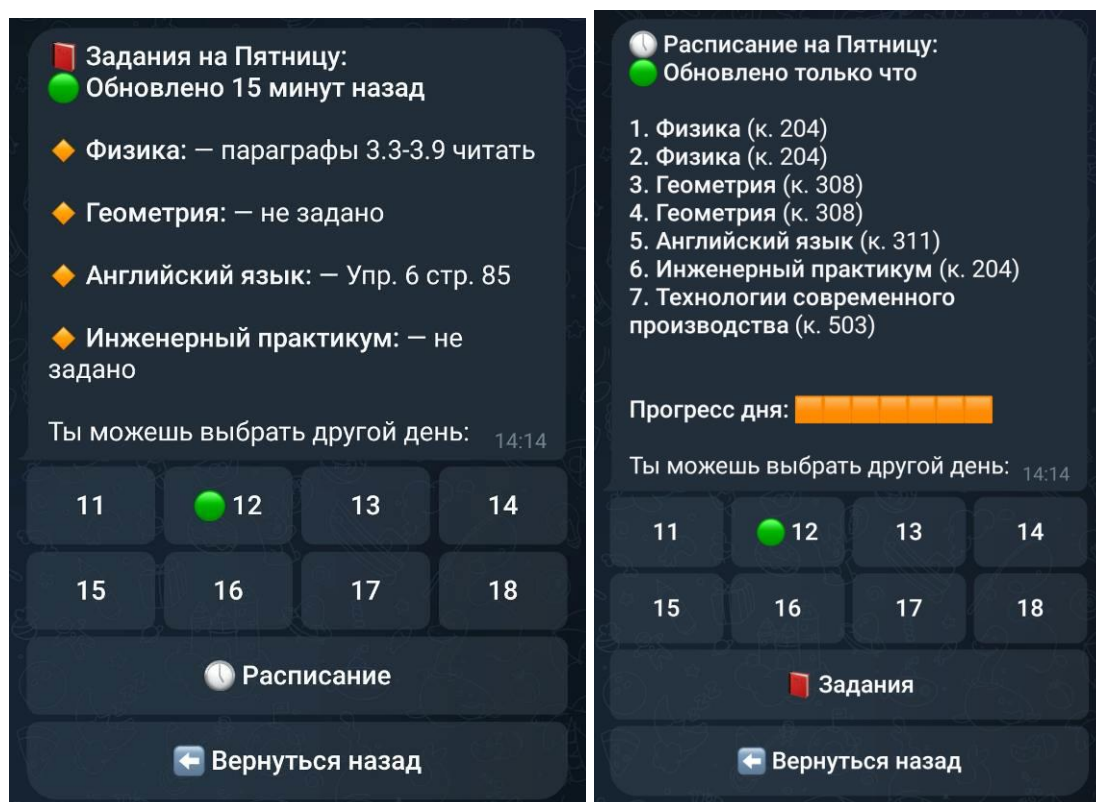


Рис. 3.5.3. Разделы домашних заданий и расписания

В разделе домашних заданий можно выбрать день, за который нужно увидеть задания (до 7 дней вперед). Бот поддерживает вложения к заданиям и множественные задания на один урок. В разделе указано время последней синхронизации данных с МЭШ, чтобы пользователь всегда мог убедиться в их актуальности.

В разделе расписания также можно выбрать желаемый день. В расписании отображаются уроки и внеурочные курсы, если они имеются в программе ученика. Отображаются номера кабинетов, где проходят занятия, а также, замены, если они есть. Реализован декоративный индикатор «Прогресс дня», который визуально отображает, сколько уроков уже прошло, а сколько впереди. По мере прохождения уроков, индикатор будет заполняться зеленым цветом. Также, прошедшие уроки зачёркиваются, а предстоящий урок для удобства отдельно выделяется специальной меткой.

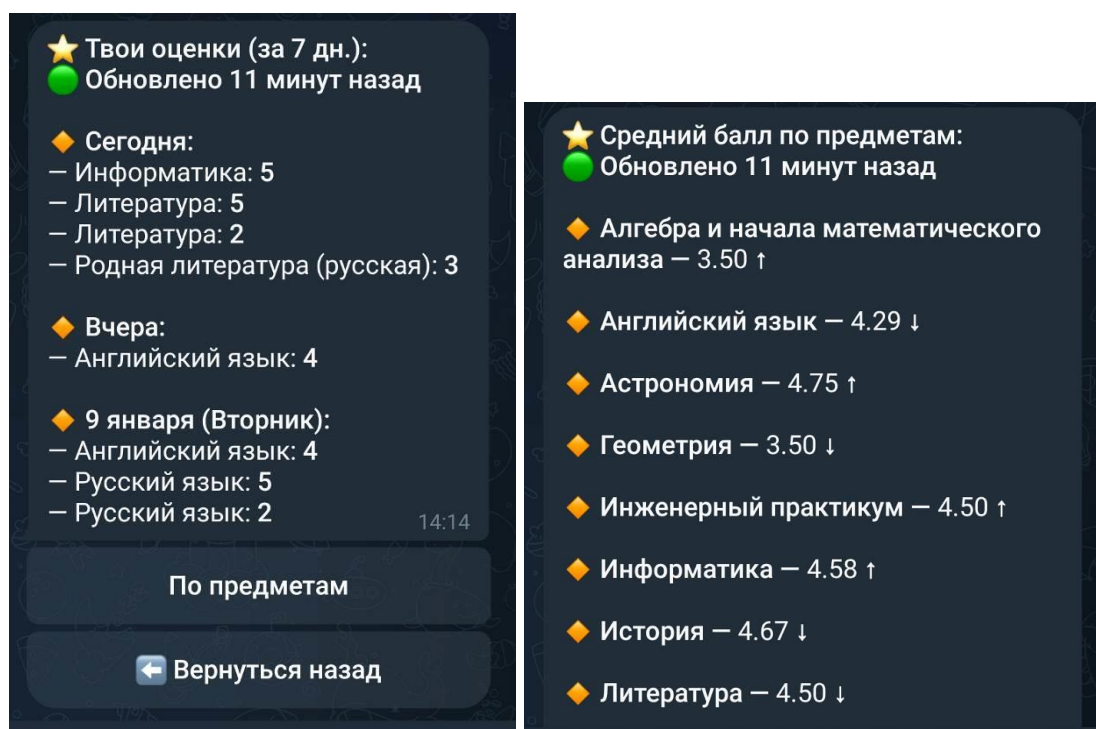
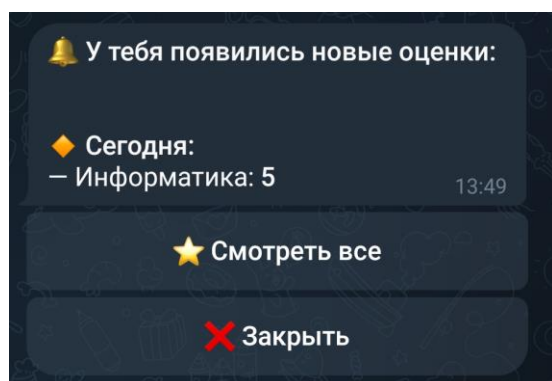


Рис. 3.5.4. Разделы оценок и среднего балла по предметам

В разделе оценок отображаются оценки по всем предметам за последние 7 дней. Если оценка имеет коэффициент, то он будет отображен в скобках рядом с оценкой, например: 5(2). Есть возможность просмотра среднего балла по предметам. Рядом с каждым предметом будет стрелка вверх, вниз, или точка – это отображение динамики среднего балла. Например, если средний балл понизился, то рядом будет стрелка вниз. Данные об оценках обновляются раз в 15 минут.

Предусмотрена возможность отправки ботом уведомлений при выставлении новых оценок:



3.6. Система групп

«Группой» в боте может быть как целый класс, так и один единственный пользователь. В группе может быть неограниченное количество пользователей. Группы различаются между собой следующими параметрами:

- Номер школы
- Класс
- Подгруппа в классе
- Список предметов и учителей по ним

Чтобы создать группу, необходимо авторизоваться в боте через аккаунт МЭШ ученика. Сервис получит из дневника данные о школе, классе, список предметов и учителей. Список предметов и учителей сортируется и преобразуется с помощью хэш-функции SHA-1 в хэш-строку для экономии хранилища базы данных, так как эти данные не несут особой ценности для функционирования бота, но необходимы для различия групп между собой. Хэш-функция всегда вернет уникальное и постоянное значение фиксированной длины для любых входных данных. Например, строка «Hello, world!» преобразуется в «943a702d06f34599aee1f8da8ef9f7296031d699».

Длина хэш-строки (следовательно, и объем, занимаемый в памяти) всегда фиксированная, независимо от длины (объема) входных данных.

Когда группа уже создана и имеет хотя-бы одного активного пользователя, который вошел через аккаунт МЭШ, к этой группе могут подключаться другие пользователи, не входя через аккаунт МЭШ, используя уникальный код приглашения, который отображается в боте у всех участников группы.

Вышеописанный сервис мониторинга получает список групп, данные в которых обновлялись более 15 минут назад (или 35 в ночное время), получает

наиболее свежий токен любого участника группы и совершает необходимые HTTP API запросы к сервисам МЭШ, используя выбранный токен.

Если запросы прошли успешно, полученные данные сохраняются в локальную базу данных, а в специальном поле таблицы с группами записывается время последнего обновления данных группы. Этот процесс выполняется циклически и круглосуточно.

Основным преимуществом системы групп является то, что нет необходимости входить через МЭШ каждому ученику, достаточно хотя-бы одного входа, а остальные подключатся по коду приглашения, однако для них функционал бота будет ограничен (Функции просмотра домашних заданий, расписания, электронных учебников будут доступны).

3.7. Система цифровых учебных материалов

Этот сервис позволяет ученикам использовать электронные учебники и другие материалы внутри бота. Поиск материалов осуществляется в интерфейсе бота. Нужные учебники можно добавить в избранное, чтобы иметь к ним быстрый доступ.

Материалы берутся в исходном формате PDF, после чего с помощью собственной Python программы преобразуются в набор изображений в формате WEBP (1 страница = 1 изображение). Это сделано для оптимизации скорости загрузки материала. Мы не загружаем сразу весь файл, содержащий множество страниц, объем которого может достигать сотен мегабайт. Обычно пользователю нужно просмотреть всего несколько страниц за время сеанса. Страницы подгружаются с сервера по мере их востребованности, т.е., по мере пролистывания учебника. Объем файла каждой страницы – в среднем не более 300 кб. С такой системой учебник загружается за доли секунды, даже при медленном соединении.

Для просмотра учебного материала Telegram бот высылает специальную ссылку в виде кнопки, которую можно открыть в любом браузере, все сразу будет готово к работе. Система автоматически сохраняет номер страницы, на которой остановился пользователь: эта страница откроется при следующем запуске учебника, даже на другом устройстве. Предусмотрена быстрая навигация по страницам: в специальное поле можно ввести номер страницы учебника, которую нужно открыть.

В будущем планируется добавление функции пометок и закладок на страницах учебников: можно будет подчеркнуть самое важное или оставить текстовые примечания.

3.8. Авторизация через аккаунт МЭШ

Для работы с сервисами МЭШ необходимы JWT Bearer токены авторизации, полученные от лица ученика. Токен – это длинная строка, подписанная цифровой подписью, в которой закодирован уникальный идентификатор пользователя, срок действия токена и прочие служебные данные. С помощью токенов сервер МЭШ понимает, кто отправляет запросы.

При первом запуске бота ученику предлагается войти с использованием аккаунта МЭШ. После нажатия на кнопку входа, в браузере открывается страница авторизации через официальный сайт дневника. Бот не запрашивает учетных данных от сервиса mos.ru. Весь процесс авторизации и выпуска токенов происходит на стороне официального сайта дневника и сайта mos.ru. Выпущенный токен имеет ограниченные права доступа и действует только в рамках сервиса МЭШ. После успешного входа и выпуска токена доступа через протокол OAuth, пользователь перенаправляется на сервисную страницу на сайте бота, где происходит проверка и сохранение токена в базе данных. Во время передачи данных используется защищенное SSL соединение, что гарантирует безопасность данных. После авторизации пользователь может закрыть

сервисную страницу и вернуться к боту, от бота придет соответствующее уведомление об успешной привязке аккаунта.

Токены доступа нужны для того, чтобы сервис мониторинга мог получать данные из дневника МЭШ. После авторизации токен каждого пользователя отдельно сохраняется в базе данных бота и используется при дальнейших запросах к МЭШ. Раз в сутки сервис мониторинга отправляет запросы на обновление токенов, так как они имеют ограниченный срок действия. Теоретически, регулярно обновляемый токен будет активен вечно, поэтому не придется входить в аккаунт каждый раз заново.

Такая схема авторизации используется в официальном приложении и на сайте МЭШ. Бот, по сути, заменяет собой приложение, отправляя такие же запросы.

3.9. Отказоустойчивость системы

Все данные, полученные ботом из сервисов МЭШ дублируются и синхронизируются с определенным интервалом в локальную базу данных сервиса. Бот может существовать автономно. Даже если сайт МЭШ будет полностью недоступен, пользователи смогут без проблем использовать все возможности бота: смотреть свои оценки, задания, расписание. Это поможет ученикам во время пиковых нагрузок на дневник. Как только сайт МЭШ станет доступен, данные сразу автоматически синхронизируются. Синхронизация происходит даже в ночное время, так как обычно в это время проблем в работе дневника нет, значит можно без проблем получить актуальные данные. Такой механизм работы обеспечивает высокую отказоустойчивость системы, что является важнейшей её особенностью.

3.10. Создание веб-сайта проекта

Сайт полностью доступен и размещен по адресу: mesh-bot.ru

На сайте размещена информация о сервисе, такая как: основные возможности, ответы на частые вопросы, статус доступности сервиса, основные преимущества, qr код и ссылка для запуска telegram бота.

Сайт написан на HTML с применением JavaScript и css. Было принято решение не использовать различные фреймворки по типу React JS, так как сайт имеет довольно простую структуру и всего несколько страниц. Был разработан современный, адаптивный и удобный дизайн сайта, имеющий возможность выбора цветовой темы оформления (темная и светлая темы).

Были использованы JS библиотеки Axios (Для HTTP API запросов), JQuery (Для удобства работы с интерфейсом сайта), Tailwind CSS (Отвечает за внешний вид сайта).

Сайт обеспечивает функционирование авторизации через МЭШ. Создана отдельная сервисная страница, куда перенаправляется пользователь после авторизации. Страница передает данные в базу данных бота через HTTP API.

Также на сайте размещена сервисная страница, которая открывается в telegram боте при его первом запуске. На ней кратко размещено описание основных возможностей бота в карточках с анимацией и коротким текстом. Такой формат позволит пользователю легко получить основную информацию о боте, притом не создавая перегрузку текстовой информацией. Пользователь переходит между карточками с помощью кнопки внизу. Всего сделано 5 карточек, в последней карточке пользователю будет предложено авторизоваться через аккаунт МЭШ или код приглашения.

Сервисные страницы скрыты от индексирования поисковыми системами и не добавлены в главное меню сайта. Они открываются лишь при определённых сценариях, таких как первичный запуск бота или авторизации через МЭШ.

Сайт был размещен в объектном хранилище от Yandex cloud, также была подключена сеть доставки контента (CDN) от Yandex cloud и Cloudflare, с целью оптимизации скорости загрузки сайта. Такой подход позволяет добиться почти 100% доступности сайта, а также, высокого уровня быстродействия и возможности масштабирования мощностей.

Благодаря почти полному отсутствию аналогов проекта и грамотной поисковой оптимизации, сайт удалось вывести в топ поиска в поисковой системе Яндекс. Ежедневно сайт посещает около десятка пользователей.

Поисковая оптимизация заключается в грамотной установке названия сайта (название должно содержать ключевые слова); составлении описания; установке иконки сайта и ключевых слов в специальных тегах. Также, сайт повышается в топе поиска по мере того, как другие люди заходят на него.

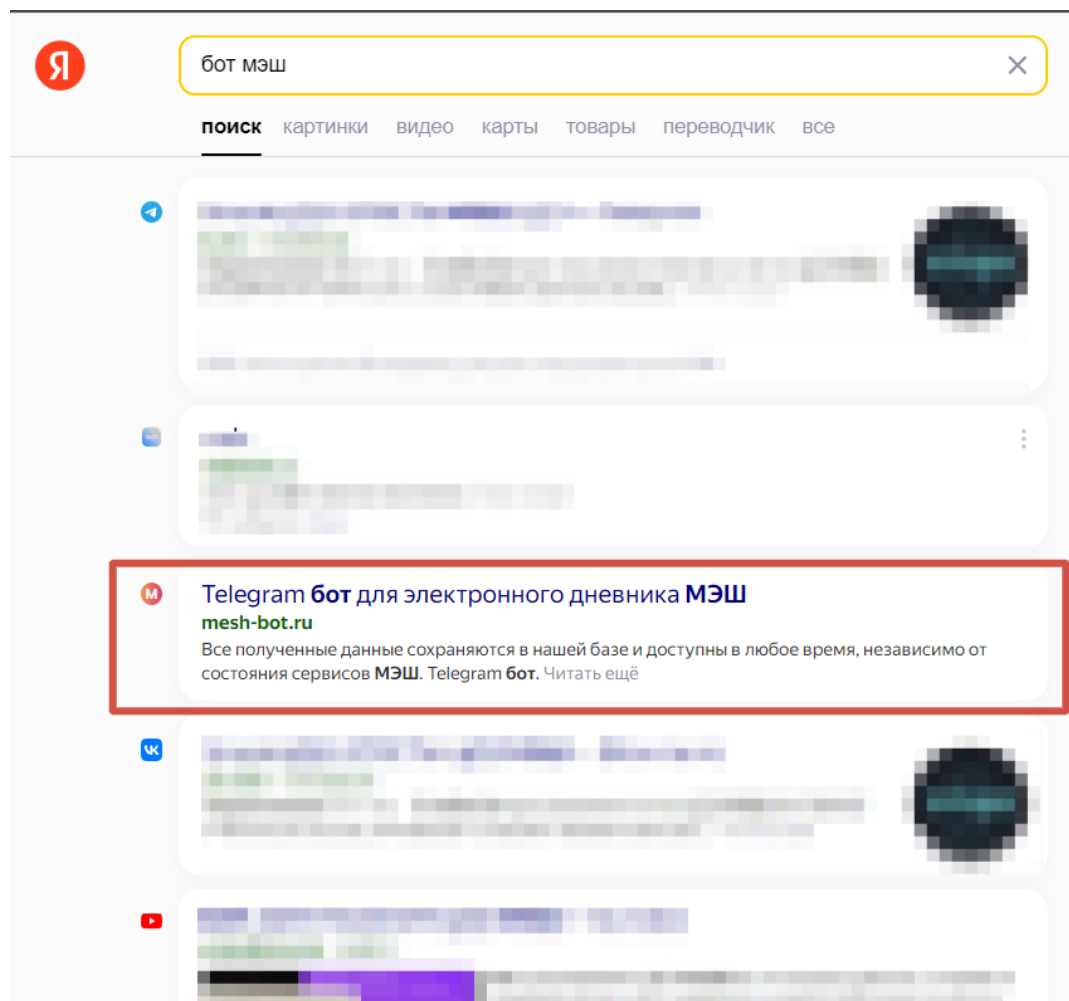


Рис. 3.10.1. Сайт проекта попадает в топ-3 поиска

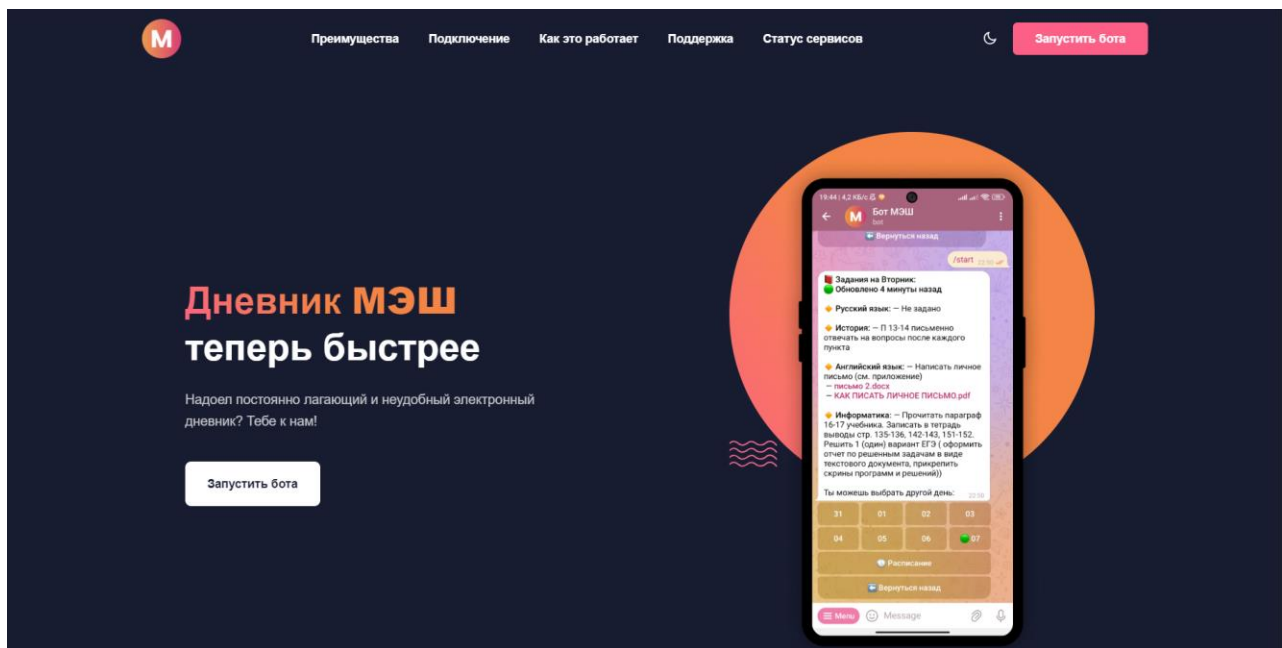


Рис. 3.10.2. Главная страница сайта

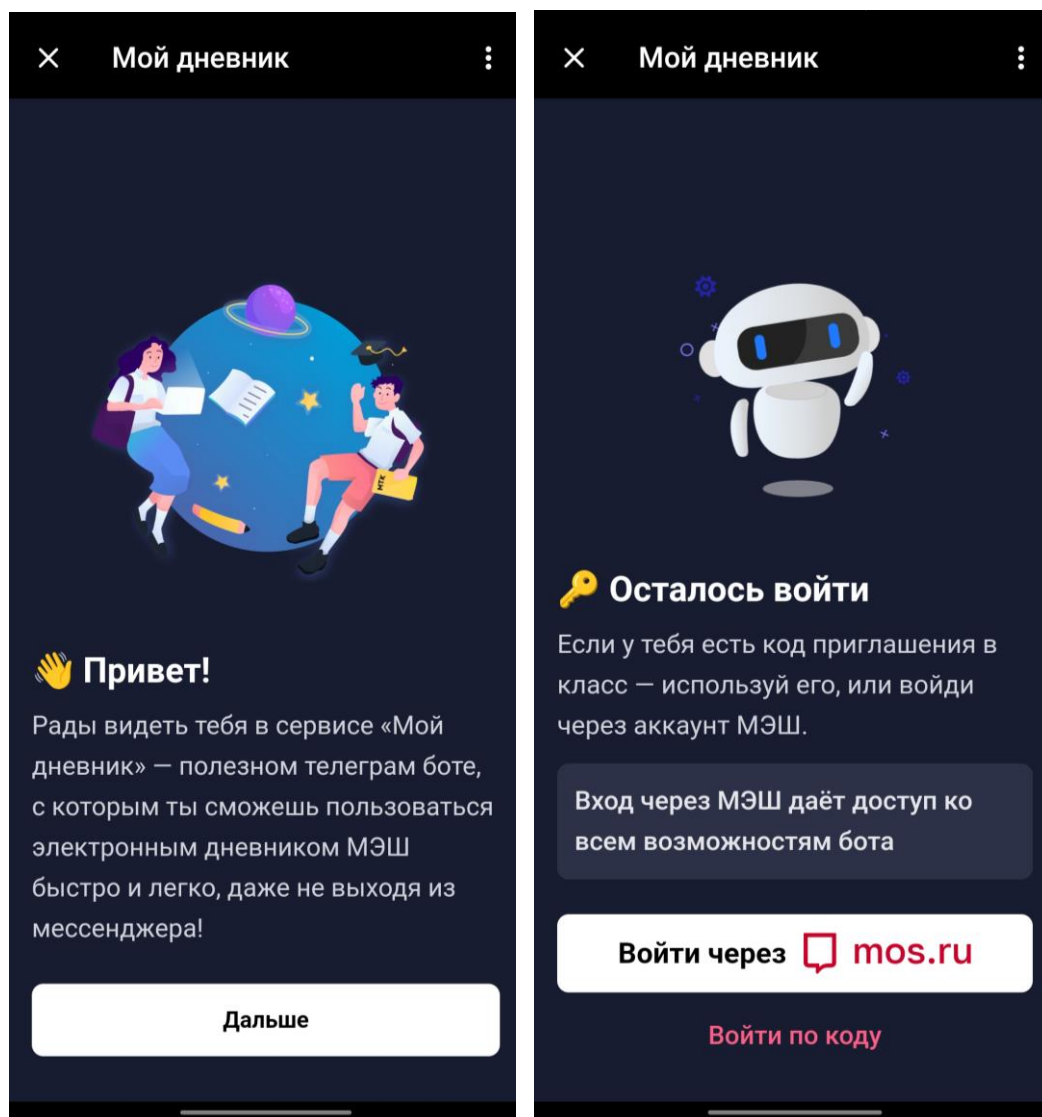


Рис. 3.10.3. Приветственная сервисная страница, которая открывается при первом запуске Telegram бота

Приветственная сервисная страница адаптирована под отображение в вертикальной ориентации, то есть, будет отлично выглядеть как на мобильных устройствах, так и на компьютерах и планшетах. Но большая часть аудитории проекта пользуются им с мобильных устройств.

3.11. Страница статуса сервисов

Статусная страница позволяет пользователям увидеть, насколько стабильно работает сервис. На странице отмечаются все значительные сбои, приводящие к полной или частичной недоступности сервиса. Также на статусной странице можно создать уведомление о технических работах на сервисе, которое увидят все пользователи. Статусная страница размещена на отдельном независимом сервере, реализована с помощью ПО с открытым кодом – Uptime Kuma.

Доступна история сбоев за последние 90 дней. Автоматически подсчитывается Uptime в процентах - это показатель, отражающий то, сколько времени был доступен сервис за последние n дней.

Например, если из 4 суток наблюдений сервис был полностью доступен только 2 суток, то его Uptime будет равен 50%. Серьезные проекты имеют Uptime около 99.9%, то есть, доступны почти всегда.

В настоящее время осуществляется мониторинг стабильности вебсайта и telegram бота. Сервис мониторинга стабильности регулярно отправляет http запросы на сайт и http api сервера telegram бота, если был получен корректный ответ, то наблюдаемый сервис считается доступным.

Однако, этот способ не всегда является точным и часто требует тонкой настройки. Ответ от сервиса не всегда означает его полную работоспособность.

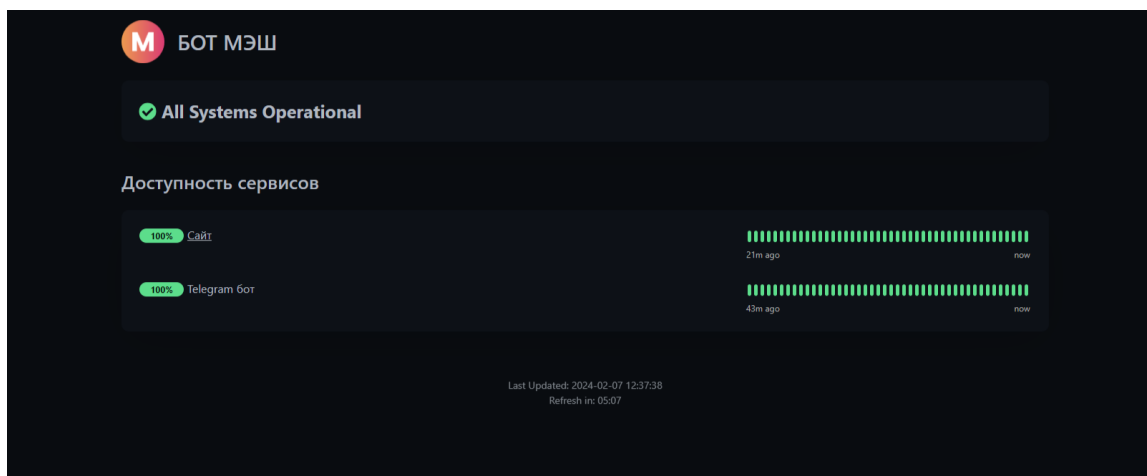


Рис. 3.11.1. Страница статуса сервисов

3.12. Безопасность данных

Поскольку бот хранит секретные ключи доступа учеников, важно обеспечить их безопасность, чтобы не допустить утечки данных. Все данные, передаваемые между сервером и пользователем, защищены сквозным шифрованием (SSL). База данных на сервере полностью изолирована от внешней сети. Подключиться к базе для проведения технических работ можно только через защищенный канал – VPN туннель. Реальный сетевой IP адрес сервера (используется для сайта) скрыт за прокси-защитой, поэтому потенциальный злоумышленник даже не сможет увидеть цель атаки. Прокси-защита фильтрует вредоносные запросы и защищает основной сервер от атак, не позволяет подключиться к консоли управления сервером. Консоль управления сервером защищена с помощью RSA ключей шифрования, подобрать которые невозможно.

3.13. Заключение

В ходе работы был создан востребованный и современный продукт, которым пользуются уже около 450 школьников (на момент 17.02.2024). Цели и задачи были выполнены. Дальнейшее развитие продукта предполагает создание универсальной образовательной платформы с полезными сервисами для школьников.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Gartner Inc. // [Электронный ресурс]. URL: <http://www.gartner.com/newsroom/id/3215217/> (дата обращения: 16.10.23);
2. Интернет 2022-2023 в мире и в России: Статистика и тренды // [Электронный ресурс]: URL: <https://www.web-canape.ru/business/internet-2022-2023-vmire-i-v-rossii-statistika-i-trendy/> (дата обращения: 21.12.23)
3. Официальная страница мобильного приложения «Школьный дневник» // [Электронный ресурс]. URL: <https://school.mos.ru/> (дата последнего обращения: 15.02.24)
4. Моделирование на UML. Общие диаграммы // [Электронный ресурс]: Моделирование на UML. URL: http://book.uml3.ru/sec_1_5#p7 (дата обращения: 09.05.2023).
5. Использование компонентной архитектуры в веб приложениях // [Электронный ресурс]: URL: <https://fwdays.com/en/event/js-frameworks-day2015/review/komponentnaia-arkhitektura-v-web-prilozheniiakn> (дата обращения: 29.01.24).
6. Архаков, Д. PHP: Делаем кнопки в Telegram API (inline- keyboards) // [Электронный ресурс]: URL: <https://archakov.im/post/nodejs-make-buttons-on-telegramapi.html> (дата обращения: 29.01.24).
7. Баночкин П.И. Методы и средства проектирования информационных систем и технологий: учебное пособие / П.И. Баночкин; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2019. – 92 с.
8. Матвеева Н. Ю., Технологии создания и применения чат-ботов [Электронный ресурс] / Н. Ю. Матвеева, А. В. Золотарюк. // Научные записки молодых исследователей. – 2018. – №1. – с. 28-30. // [Электронный ресурс]: URL: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov> (дата обращения: 29.06.23).
10. Yii2 framework. Обзор экосистемы // [Электронный ресурс]: Habr. URL:

<https://habr.com/company/simpleweek/blog/254681/> (дата обращения: 23.04.23).

11. Официальный сайт, фреймворка Yii2 // [Электронный ресурс]: URL:

<https://www.yiiframework.com/> (дата обращения: 20.04.23)

12. Официальный сайт Telegam API // [Электронный ресурс]: URL:

<https://core.telegram.org/api>. (дата обращения: 29.04.23).

13. Официальный сайт Maria DB // [Электронный ресурс]: URL:

<https://mariadb.org/>. (дата обращения: 29.04.23).