Государственное бюджетное образовательное учреждение №1400
Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА - Российский технологический университет»

«ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АРЕНДЫ СЕРВЕРОВ С ПАНЕЛЬЮ УПРАВЛЕНИЯ»

Авторы:
ученик класса 10 «И» ГБОУ
Школы №1400
Арустамян Нарек
Андраникович

Научный руководитель:
Преподаватель Детского технопарка
«Альтаир» РТУ МИРЭА

Москва, 2026

Оглавление

Введение

Проект представляет собой сервис для аренды игровых и физических серверов с удалённым доступом.

Большая часть нынешних российских хостинг-проектов арендуют оборудование на удалённом доступе у крупных иностранных провайдеров (к примеру, hetzner или ovh), из-за чего в таких сервисах присутствует большой пинг и ограниченность (нет прямого доступа к оборудованию, нельзя масштабировать серверную своими оборудованием). Помимо этого, появляются дополнительные расходы на оплату аренды серверов.

Мы в свою очередь организовываем всё на своей серверной. Благодаря этому сервера для российских клиентов будут иметь маленький пинг, а мы можем легко улучшать оборудование для проекта и создавать индивидуальные решения (тарифные планы) для своих пользователей.

Цель и задачи

Наша цель — организовать инфраструктуру с веб-приложением для аренды серверов с панелью управления, которая позволит просматривать файловую систему своего сервера, менять её, включать и выключать сам сервер, а также писать консольные команды. Для этого нам нужно сделать следующее:

- Веб-сайт, где пользователь сможет узнать о нашем проекте, ознакомиться с тарифными планами, выйти на связь с поддержкой в случае возникновения вопросов, проблем или желании договорится об индивидуальном тарифном плане для него и обеспечить аренду сервера.
- Telegram-бот для быстрой аренды игрового сервера, связи с поддержкой в нужный момент, просмотр информации о проекте или своём сервере, получение данных для входа в панель управления и благотворительной деятельности для проекта.
- Поставить и настроить саму веб-панель для управления игровым сервером.

Ход работы

1. Порядок выполнения

Для работы нашего сервиса нужно организовать серверное оборудование обеспечить защищённое ssl-подключение клиента с нашим веб-приложением и панелью управления. После этого можно начать организацию клиентской части: написаниепанели управления, Telegram-бота и оформления веб-сайта.

2. Написание Telegram-бота

Список функций Telegram-бота: Регистрация пользователей с прохождением анкеты (нужно для сбора статистики с клиентов); просмотр списка арендованных игровых серверов; возможность быстрого создания сервера с выбором ядра и типа игры; просмотр данных для входа в панель управления серверами; возможности удалить свой сервер, сбросить пароль; команды для администрации, нужные для мобильного управления частью системы сервиса: написание sql-запросов в базу данных, ограничение доступа к функционалу для некоторых пользователей, получение информацию в виде json данных через API; информация о связи с поддержкой и возможность благотворительной деятельности для помощи в развитии проекта.

Чтобы написать Telegram-бота мы будем использовать язык программирования python, так как он имеет наиболее функциональные библиотеки для разработки ботов - к примеру, aiogram3. Для работы с API мы используем стандартную библиотеку requests. Первым нужно написать стартовый файл и подготовить архитектуру проекта для модульности проекта и простоты введения нового функционала в дальнейшем (рис 1).

Рисунок 1. Архитектура проекта и стартовый файл

Для хранения данных о пользователях мы создали локальную базу данных (рис 2) в боте (в дальнейшем будет заменена на онлайн базу данных для полноценной синхронизации с сайтом и другими сервисами). В базе данных будут храниться данные об идентификаторе аккаунта пользователя в телеграмме, панели управления, его пароле, статусе аккаунта (нужно для назначения прав и при необходимости их ограничения), и лимит бесплатных серверов, которые он может получить (по умолчанию — 1).

Подвальчик.hc Имя таблицы: Users

	Имя	Тип данных
1	id	INTEGER
2	telegram_id	INTEGER
3	pterodactyl_id	INTEGER
4	local_pterodactyl_password	TEXT
5	status	INTEGER
6	local_servers_count_limit	INTEGER

Рисунок 2. Структура базы данных

Директория для работы с базой данных разбита на 4 файла:

- __init__.py – основной файл-загрузчик базы данных, который создаёт таблицу пользователей в случае её отсутствия.
- servers.py – работа с серверами пользователей в связке с API панели управления.
- users.py – работа с пользователями: создание, удаления, изменение данных. Имеет связку с API панели управления.
- helpers.py – вспомогательные функции для отправки ручных sql-запросов через бота, проверки наличия регистрации пользователя.

Код в директориях handlers и keyboards отвечают за обработку сообщений и нажатий на различные кнопки в боте соответственно. Остаются лишь вспомогательные скрипты для генерации паролей, имён пользователей (generators.py) и отправки API-запросов в панель управления (pterodactyl.py). Помимо всех скриптов имеется директория configs (рис 3), где хранятся настройки для работы бота, все текстовые сообщений бота, которые он отправляет во время использования, конфигурация ядер и описания структуры сложных API-запросов (например, запросы на создание сервера).

Рисунок 3: Директория конфигурации

Остальные файлы вспомогательные и нужны лишь для удобной передачи файлов проекта, хранения библиотек в виртуальной среде и их список для лёгкой установки на сервер.

3. Реализация панели управления

В качестве панели управления для игровых серверов мы будем использовать проверенную годами технологию pterodactyl. Она позволяет развернуть на сервере простую панель управления с пользователями и серверами. Данная панель имеет свой API, который и поможет нам автоматизировать процесс создания пользователей и серверов на платформе. Разворачивать данную панель мы будем на отдельном от самих игровых серверов арендованном VDS, что бы в случае перегрузки серверов, наш клиентский сервис продолжал работать (на этом сервере также будет работать Telegram-бот и будущий веб-сайт). После базовой настройки панели мы ставим её на веб-сервер Nginx, который был заранее установлен на сервер и запускаем панель под ssl-сертификатом для защищенного соединения.

Как только мы убедимся, что всё стабильно работает, мы добавляем поддержку плагинов в нашу панель, что даст нам возможность увеличивать её фукнционал и стиль различными модификациями. После этого мы поставили

плагин на кастомизацию панели, что бы её дизайн подходил под наш проект (рис 4).

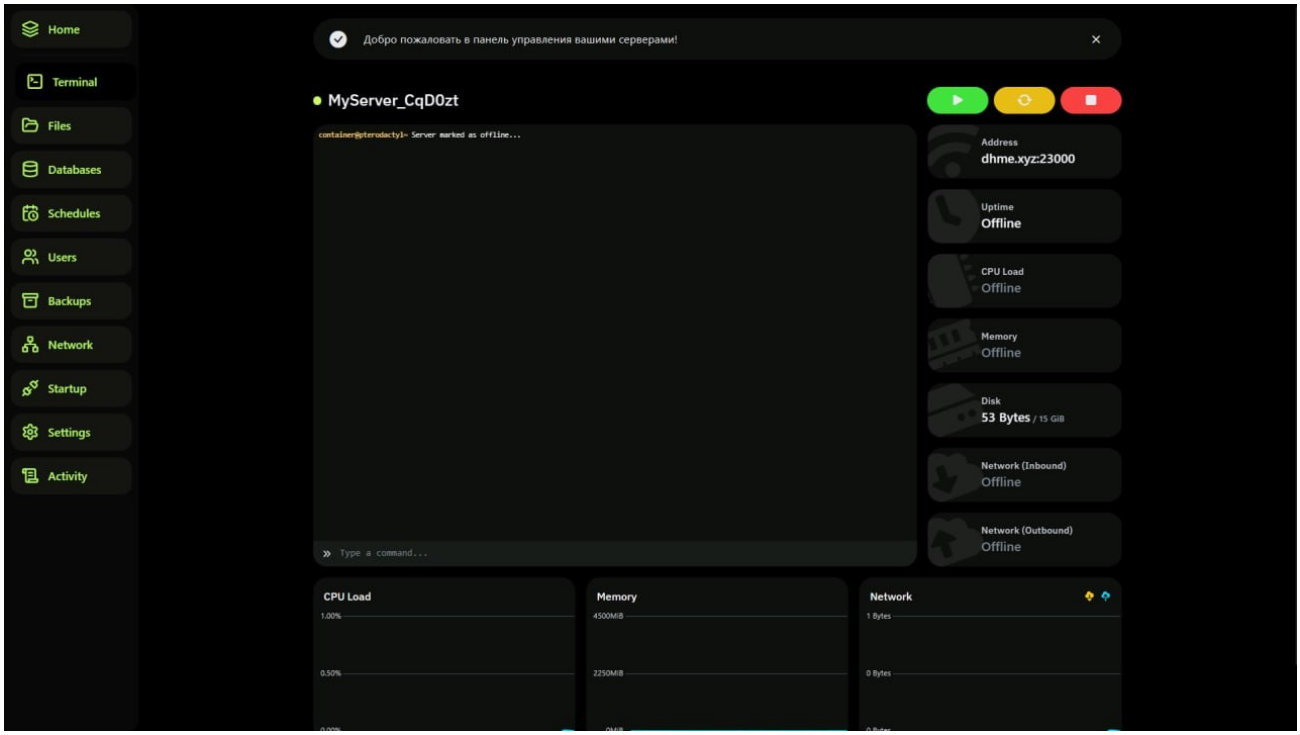

Рисунок 4. Панель управления сервером и её стиль

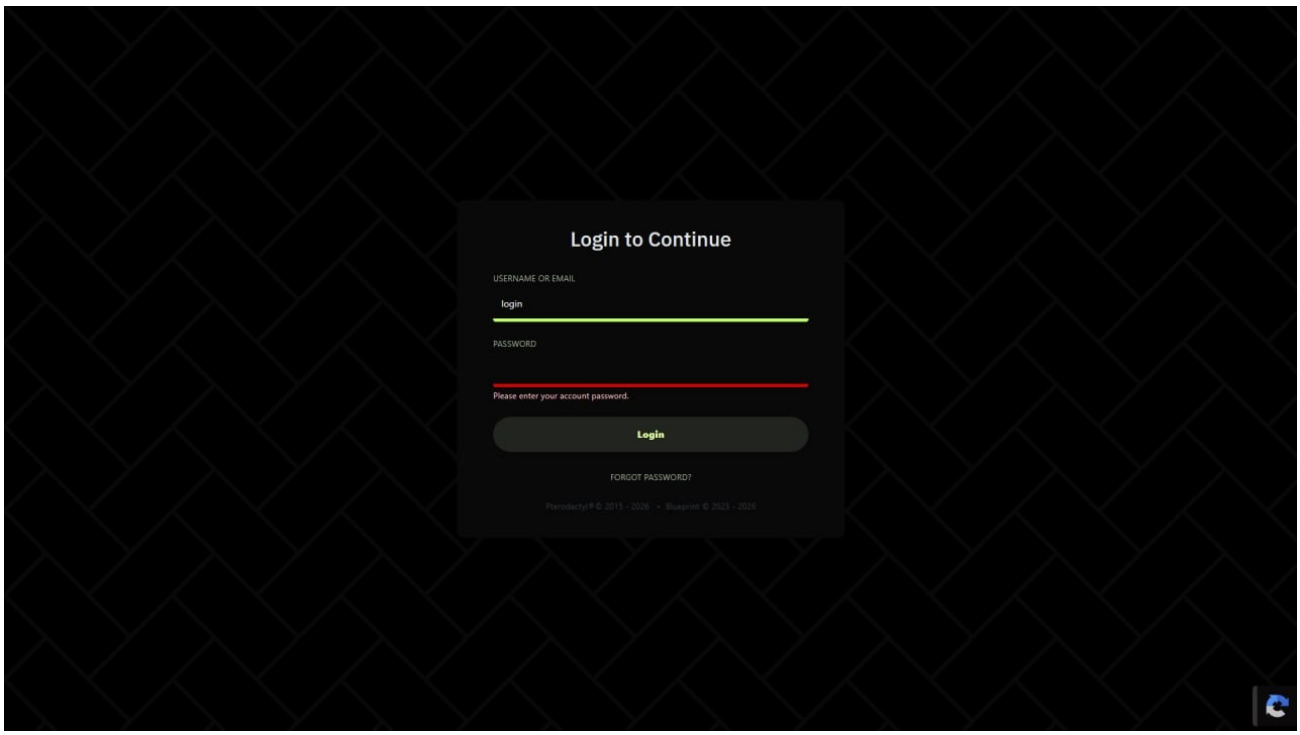

Рисунок 5. Страница входа в панель управления и её стиль

На этом можно прекратить работу с панелью, лишь изредка добовляя различные модификации по мере необходимости.

4. Веб-сайт для хостинга

Веб-сайт будет в основном выступать как основной источник информации. Там будут написаны посты с последними новостями, информация для связи с поддержкой в целях аренды физического сервера и ссылка на Telegram-бота для получения бесплатного игрового сервера, ссылки на официальный Telegram-канал, панель управления и калькулятор для расчёта примерной цены своего тарифного плана.

Для написания веб-сайта мы будем разворачивать два веб-сервера на языке программирования JavaScript для frontend и backend частей отдельно. Помимо этого frontent часть будет написана на языке разметки гипертекста HTML и языках программирования CSS и JavaScript, нужные для стилизации и анимации страниц на сайте. Помимо этого мы будем использовать фреймворк ReactJs и Express.

Есть пример главной страницы (рис 7) и архитектуры веб-сайта (рис 6).

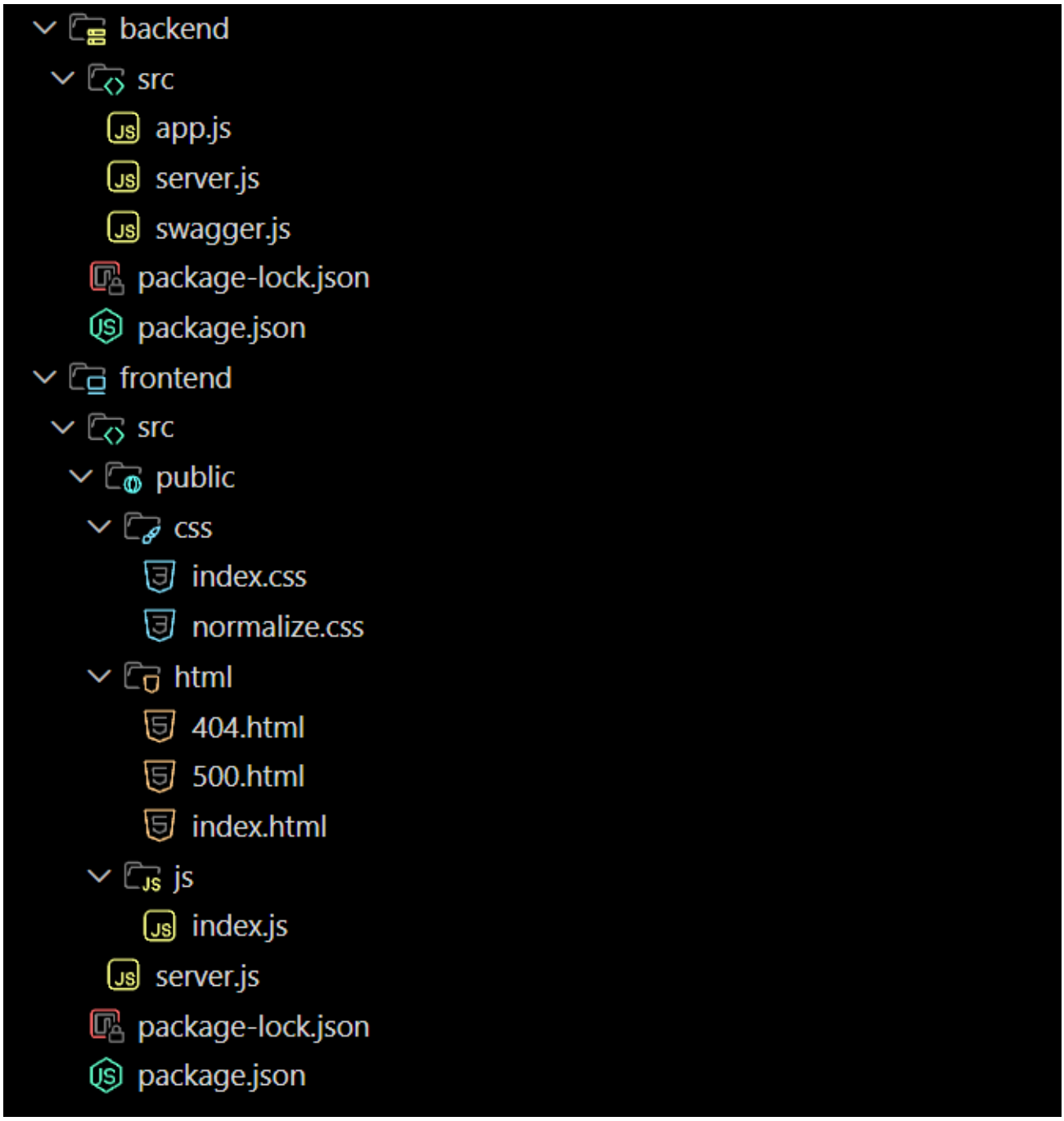

Рисунок 6. Архитектура написания веб-сайта

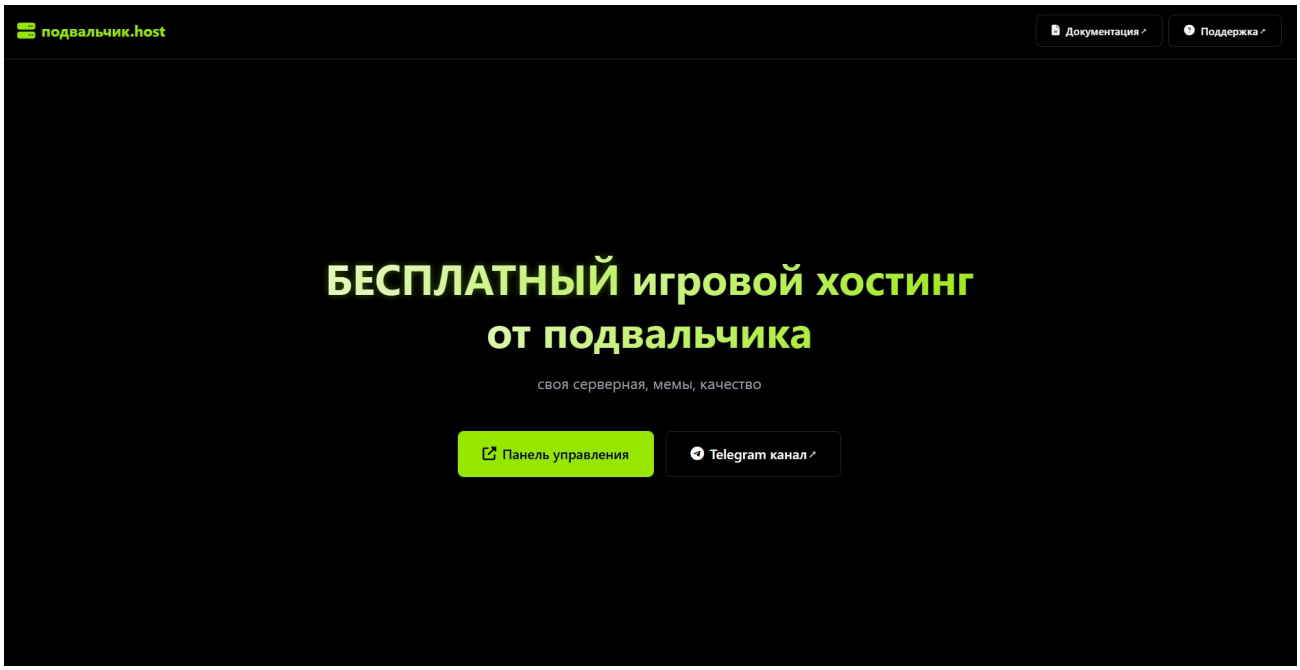

Рисунок 7. Пример главной страницы сайта

5. Подготовка оборудования

Список оборудования, которое мы использует для работы сервиса:

- Сервер Huawei RH1288v3
- Сервер Huawei RH2288v3
- Blade-система HP c7000 Enclosure
- Сервер Blade HP bl460c g7 (вскоре будет продан и заменён на g8)
- Сетевое оборудование Mikrotik L009UiGS
- Серверная стойка 42u

Основная настройка оборудования включает в себя развертывания операционной системы Linux на базе дистрибутива Debian 12. После установки операционной системы, сразу проводим из подсети /28 один ipv4 на сервер и открываем порты. Уже дальше приступаем к настройке правил браундмэра, смена стандартного порта SSH (22), настройка входа по ключам. Установка служб для мониторинга и подключения их к служебному серверу

Создания SWAP файла 30% от обьема оперативной памяти, а так же подключения ZRAM. Потребление памяти выставляется по приоретету: если кончается основная RAM, то используем ZRAM, а если кончается ZRAM, то используем SWAP. Это необходимо для того чтобы система не зависла в случае заполненности ОЗУ.

После базовой настройки производим установку Docker. На всех серверах у нас включена виртуализация через BIOS, а служба по созданию контейнеров позволяет создавать нам изолированные, независимые друг от друга игровые сервера по различным играм. Установка Wings - служба необходимая для Pterodactyl что бы подвязать узел (сервер).

Веб-часть у нас размещена на отдельном арендованном VDS сервере (Панель, сайт, бот). Купленный нами домен подвязывается к CloudFlare и к ip сервера через DNS запись. Подписанный SSL-сертификат через certbot, с установкой автообновления в случае истечения.

7. Сроки реализации

На реализацию всего проекта до нынешнего результата ушло два месяца, а на реализацию недоделанных задач проекта уйдёт около ещё двух месяцев.

Результаты

1. Результаты в кратком описании

На данный момент уже подготовлено оборудование для стабильной работы, имеется ssl-сертификат, разработан Telegram-бот, начальная веб-страница и подключенная к ней панель управления. Помимо этого, мы улучили нашу сетевую составляющую, проведя дополнительные 16 ip-адресов к

серверам, что даст возможность создания дополнительных услуг для клиентов и позволит организовать маршрутизацию трафика.

2. Работа с аудиторией и статистика

До этого мы открылись за краткие сроки для тестирования нашего оборудования и смогли набрать минимальную аудиторию. Ниже предоставлены фотографии со статистикой (Рисунок 9-10).

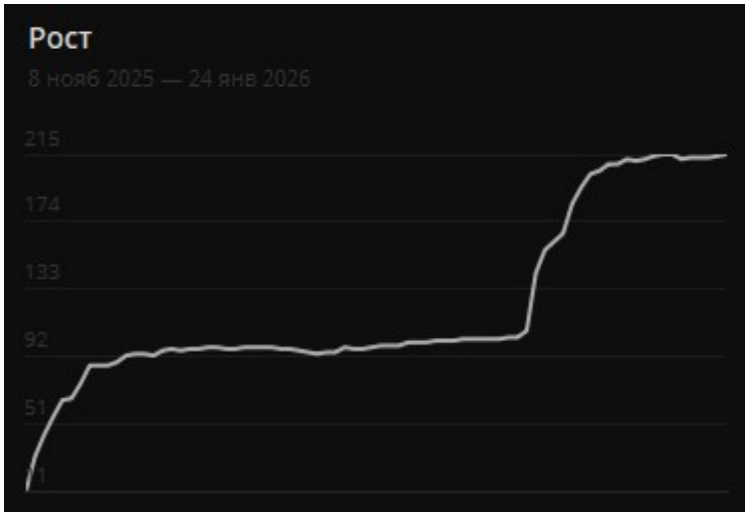

Рисунок 9. Статистика пользователей с момента создания проекта

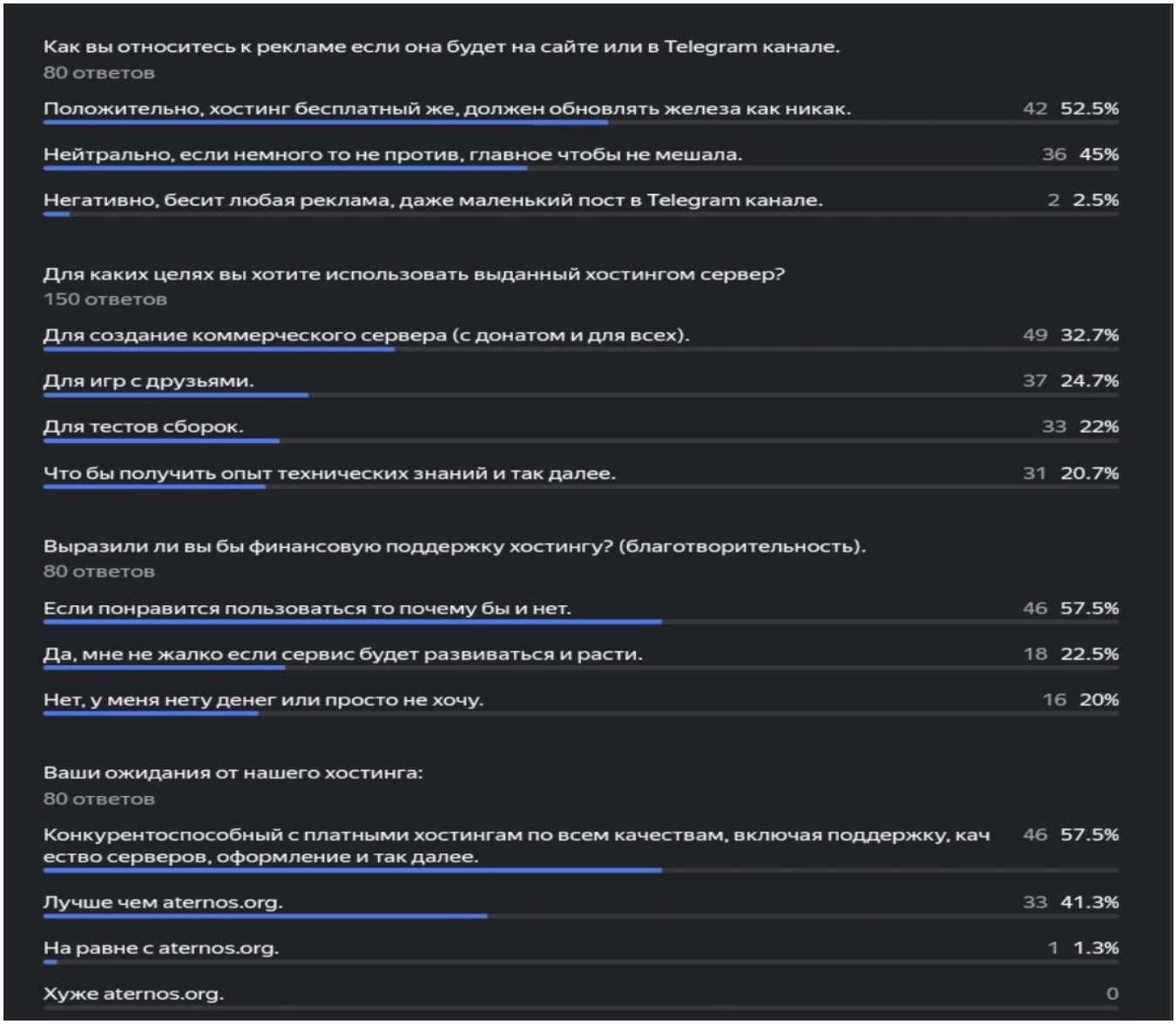

Рисунок 10. Прочая статистика, собранная с пользователей

3. Telegram-бот

Бот был успешно и полностью разработан. Теперь на него изредка разрабатываются обновления для увеличения функционала и мобильности. Был полностью разработан личный кабинет, список серверов и остальные фукнции, указанные ранее (Рисунок 11-12):

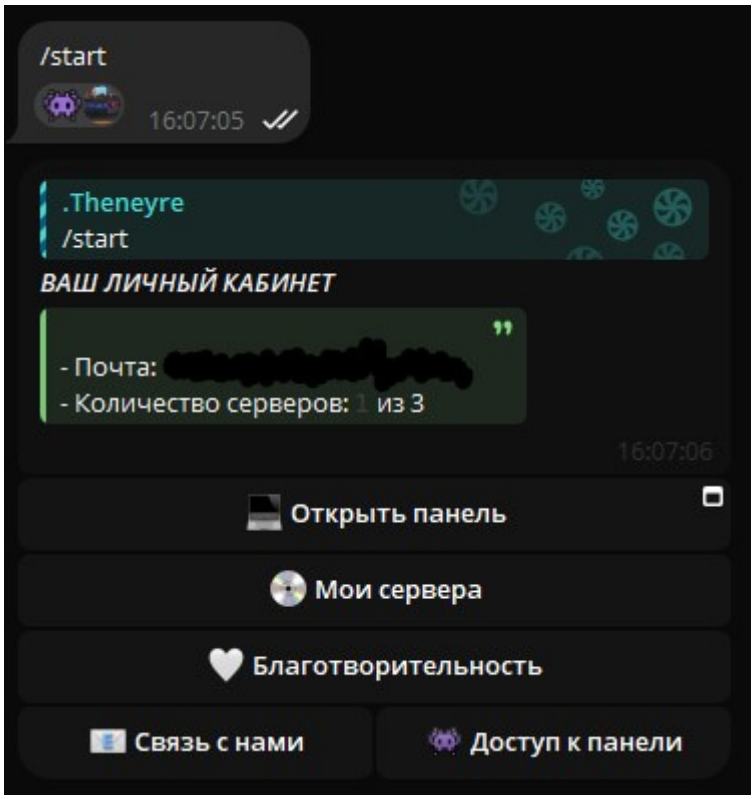

Рисунок 11. Личный кабинет в Telegram-боте

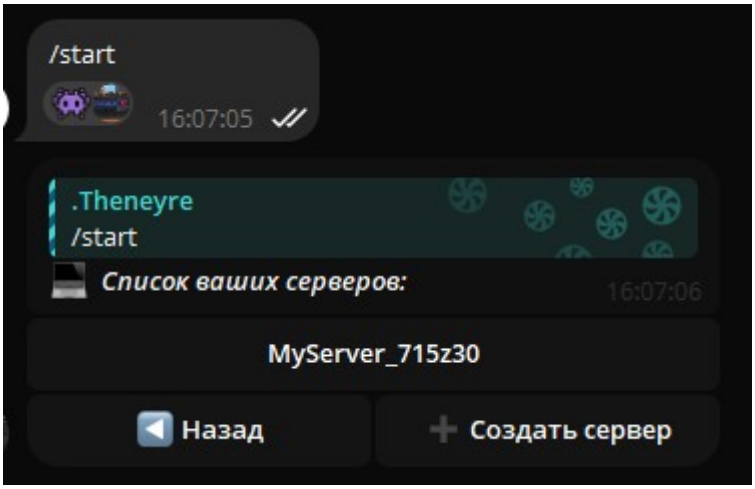

Рисунок 12. Список серверов в кабинете пользователя и функция создания сервера

4. Невыполненные задачи

Осталось реализовать до конца сам веб-сайт, расписать там информацию об тарифных планах и частично автоматизировать процесс аренды физического сервера и интегрировать аренду игрового сервера на сайт (процесс аренды игровых серверов автоматизирован через Telegram-бота для удобства клиентов)

Неготовность веб-сайта вызвана в основном из-за технических издержек. Было потрачено много времени на написание Telegram-бота и обновления для него, стабилизацию всех систем и обеспечение полной работоспособности панели управления, а также решение проблем в поддержке сервиса, разработка дизайна, логотипа. Немалое влияние на скорость работы также оказала учёба)

Выводы

После открытых тестов на реальной аудитории мы смогли сделать вывод, что несмотря на незаконченность проекта, он пользуется популярностью благодаря ряду следующих факторов:

- Telegram-бот действительно удобен в качестве инструмента для быстрого решения малых задач от клиента (рис 13), такие как быстрая аренда бесплатного игрового сервера, переход от личного кабинета в панель управления и просмотр различной информации о сервисе
- Панель управления работает всегда стабильно благодаря использованной и годами проверенной технологией Pterodactyl. В дальнейшем мы будем больше модифицирован нашу панель в целях добавления нового функционала и оптимизации (к примеру, автовыключение неиспользуемых серверов для снижения потребления общих ресурсов серверной)

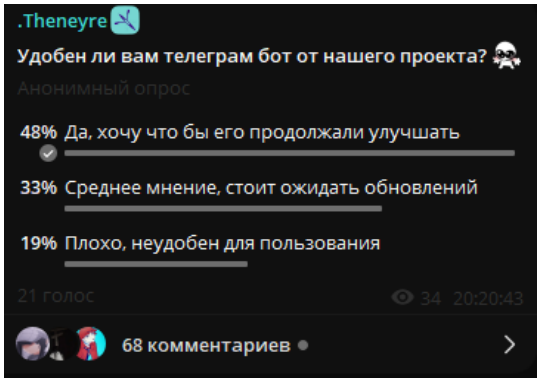

Рисунок 13. Опрос по поводу удобства Telegram-бота

Список литературы

- Документация по Pterodactyl API [Электронный ресурс] // netvpx : [сайт]. — URL: https://pterodactyl-api-docs.netvpx.com/docs/intro (дата обращения: 24.01.2026).
- Документация по Aiogram3 [Электронный ресурс] // aiogram : [сайт]. — URL: https://docs.aiogram.dev/en/latest/ (дата обращения: 24.01.2026).
- Документация по Requests [Электронный ресурс] // readthedocs : [сайт]. — URL: https://requests.readthedocs.io/en/latest/index.html (дата обращения: 24.01.2026).
- Документация по ReactJS [Электронный ресурс] // react : [сайт]. — URL: https://react.dev/learn (дата обращения: 24.01.2026).