Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

Школа бакалавриата

ОТЧЕТ

По проекту
«Создание системы сканирования и анализа уязвимостей интернет-устройств»

по дисциплине «Проектный практикум»

Заказчик: Фамилия И.О.	Путинцева Т.А.
Куратор: Фамилия И.О. ученая степень, ученое звание, должность	Путинцева Т.А.
Студенты команды Babmouk Фамилия И.О.	Черкасов А.И.
Фамилия И.О.	Симонов А.А.
Фамилия И.О.	Косолапов С.А.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В наши дни сетевая инфраструктура часто подвержена кибератакам и в большинстве случаев имеет какие-либо уязвимости. Существует множество сервисов, позволяющих проводить сканирование интернет-вещей, но они не будут удобны в использовании для обычного пользователя, малого и среднего бизнеса в большинстве случаев, так как нужно обладать навыками в информационной безопасности.

Цель нашего проекта – создание веб-сервиса, позволяющего пользователям сканировать сетевую инфраструктуру и веб-приложения на предмет уязвимостей и получать отчеты в удобочитаемом формате. Из задач стоит отметить дать возможность пользователям проводить простые и удобные сканирования сетевой инфраструктуры.

Актуальность проекта заключается в создании удобного сервиса для интеллектуального сканирования, который будет удобен для каждого человека даже не знакомого с информационной безопасностью.

Программный продукт может быть использован как индивидуально пользователем, так и малым/средним бизнесом.

Ожидаемая конечная цель нашей работы – веб-приложение с возможностью запуска сканирований интернет-устройств, получения аналитики по итогам сканирований, создание задач периодических сканирований.

Основная часть

1.1 Работа каждого участника

Роли были распределены следующим образом на основе прошлых проектных работ и навыков участников:

1) Черкасов А.И. – Backend-разработка
2) Симонов А.А. – Frontend-разработка
3) Косолапов С.А. – Аналитика и тестирование

1.1.1 Черкасов А.И.

Александр реализовывал backend-часть проекта при помощи языка разработки python и фреймворка FastAPI. Развернул проект на тестовом стенде и на продакшн-версии. В его задачи входило:

1) разработка архитектуры проекта
2) разработка схемы базы данных
3) разработка основного API
4) работа со сканерами
5) исследование различных ии моделей
6) деплой приложения

1.1.2 Симонов А.А.

Александр работал над frontend-частью приложения используя фреймворк Vue.js. Работал над выбором цветов и стилей.

1) написание технического задания
2) выбор стилистики приложения
3) изучение и выбор оптимального фреймворка для frontend-разработки
4) реализация связи frontend с backend

1.1.3 Косолапов С.А.

Сергей работал над аналитикой проекта, поиском и ведением багов на тестовом стенде.

1) постановка и ведение задач
2) оформление документации
3) аналитика
4) тестирование

1.2 Требования к программному продукту, составление плана действий для достижения цели

1.2.1 Функциональные требования

Начальный экран

- Основная информация о сервисе
- Описание возможностей сервиса
- Предложения регистрации/входа

Основная страница

- Значок профиля пользователя, позволяющий перейти на страницу профиля
- Результаты последних проверок
- Возможность перейти на страницу сканирований
- Возможность добавить в профиль цели для сканирований
- Основная информация о сервисе, описание его возможностей

Профиль пользователя

- Отображение данных пользователя с возможностью их изменить
- Информация о статусе подписки
- Кнопка для привязки аккаунта Telegram

Страница сканирований

- Возможность создания задач на обычное/рекуррентное сканирование
- Возможность просмотра статуса задач
- Возможность просмотреть/скачать отчет по задаче
- Возможность посмотреть статистику задач с рекуррентным сканированием
- История сканирований

1.2.2 Нефункциональные требования

1) Минималистичный интуитивный интерфейс, понятный всем категориям

пользователей

2) Современная масштабируемая архитектура
3) Высокая скорость работы приложения
4) Интеграция с LLM
5) Интеграция с Telegram для реализации бота с уведомлениями
6) Разработка на Python FastAPI
7) Использование очереди сообщений RabbitMQ
8) База данных PostgreSQL для хранения информации
9) База данных Redis для хранения состояния пользователей и кеширования
10) Безопасное хранение и обработка персональных данных пользователей
11) Дизайн
    - Простая интуитивная навигация
    - Современный минималистичный интерфейс
    - Приятная цветовая палитра и вёрстка
    - Адаптивность
12) Этапы разработки

- Разработка архитектуры сервиса
- Разработка backend-части проекта и базы данных
- Разработка frontend-части проекта
- Деплой, тестирование и исправление багов, ошибок, недоработок, внедрение необходимых изменений

13) Ожидаемый результат
    - Полностью функционирующий веб-сервис для выявления уязвимостей инфраструктуры пользователя с удобными и понятными отчётами о сканированиях, с возможностью автоматизации проверок и уведомлениями через телеграм-бота.

1.2.3 План действий для достижения цели

В план действий мы включили:

1) выбор ролей
2) постановка задач, их ведение в трекере
3) анализ
4) разработка backend
5) разработка frontend
6) тестирование
7) деплой приложения

1.3 Анализ и сопоставление аналогов

Таблица сравнения аналогов (Таблица 1):

Таблица 1 – сравнение аналогов

Решение	Функциональность	Доступность	Ценовая модель	Целевая аудитория
ScanFactory	Автоматизированное сканирование безопасности внешних веб-сайтов и сетевой инфраструктуры с помощью ~15 сканеров (open-source и коммерческих). Выявляет уязвимости веб и сетевых сервисов, утечки учетных данных, открытые порты; есть модули OSINT, брютфорс паролей, поиск поддоменов. Интегрируется с SIEM/SOAR и таск-трекерами, поддерживает выгрузку результатов через API.	B2B SaaS; развертывание в облаке или установка on-premises по выбору. Предназначен для организаций с ~20 до 1000+ узлов инфраструктуры.	Подписка (варианты лицензии по числу хостов). Базовый пакет включает ~19 сканеров и авто-отчеты; опции – ручная верификация экспертами и отчеты на русском для руководства. Стоимость рассчитывается индивидуально под клиента (зависит от числа хостов, рост >10% пересчитывается).	Коммерческие организации с внешними веб-ресурсами и сервисами. Чаще выбирается средним и крупным бизнесом, финтех и ИТ-компаниями, которым нужен постоянный аудит периметра. ИБ-отделы ценят широкие возможности и гибкость настройки сканирования.
MaxPatrol VM	Платформа управления уязвимостями от Positive Technologies. Автоматически выявляет актуальные уязвимости во всей ИТ-инфраструктуре (сети, ОС, БД, приложения) и приоритизирует их по критичности для бизнеса. Поддерживает непрерывную инвентаризацию	Предназначен для корпоративного сегмента; развёртывание локально (on-premise) на инфраструктуре заказчика. Имеет сертификаты ФСТЭК/ФСБ для критических систем, что позволяет использовать в госорганизациях и крупных	Лицензируется по числу узлов (перpetуальная лицензия с поддержкой или ежегодная подписка). Например, лицензия на 256 узлов стоила ~80 тыс. ₽ (без НДС); крупным организациям доступны корпоративные тарифы. Стоимость растёт с масштабом инфраструктуры (может достигать	Крупный бизнес, корпоративные ИТ-инфраструктуры, банки, телеком, госорганизации. Обычно используется командами ИБ для комплексного контроля уязвимостей и соответствия стандартам в больших сетях. Требуется

	активов и корреляцию с базой знаний (обнаруживает новые уязвимости без повторного сканирования благодаря обновлению базы). Имеет модули для сканирования без агентов, проверки конфигураций (комплаенс) и аудита безопасности**. Интегрируется с экосистемой PT (SIEM, Network Attack Discovery).	компаниях. B2B-решение, отсутствует как публичный облачный сервис.	нескольких миллионов ₽ в год для больших сетей).	выделенные ресурсы для развёртывания и поддержки решения.
BI.ZONE CPT	Continuous Penetration Testing – платформа непрерывного пен-тестинга. Проводит регулярное (частое) сканирование внешнего периметра компании на уязвимости с участием команды экспертов BI.ZONE. Объединяет автоматизацию (сканеры Nessus, OpenVAS) с ручной проверкой выявленных проблем. В платформе реализована коммуникация между заказчиком и pentest-экспертами (через комментарии) и полный цикл управления уязвимостями (от обнаружения до устранения).	Только облачный SaaS-сервис, предоставляемый как услуга от BI.ZONE. Ориентирован на модель B2B. Использование наиболее эффективно для больших компаний с широкой инфраструктурой; для малого бизнеса избыточен. Нет on-prem версии.	Подписка (как сервис от провайдера). Тарифы формируются индивидуально под клиента (учитывая размер периметра и требуемую частоту тестирования). Цены высокие – решение относится к премиум-сегменту услуг (контракт на год для крупной организации может составлять миллионы ₽). Оплата за доступ к платформе и работу экспертов на постоянной основе.	Крупные предприятия и холдинги, финансовый сектор, компании с повышенными требованиями к безопасности (банки, критическая инфраструктура). Обычно выбор ИБ-департаментов, которым нужен постоянный контроль и экспертная оценка уязвимостей.
CVM (Awillix)	Continuous Vulnerability Monitoring –	SaaS (полностью облачное решение,	Модель подписки: стоимость рассчитывается по	Бизнес любого размера, особенно

	облачный сервис 24/7 для автоматического тестирования инфраструктуры и веб-приложений на уязвимости. Включает модули инвентаризации (сканирует внешние хосты, домены, сервисы), мониторинга SSL-сертификатов, поиска уязвимостей (сканерами и вручную) и аналитики рисков. Использует комбинацию open-source и проприетарных сканеров, плюс собственные разработки. Возможны уведомления о критических уязвимостях в мессенджеры/e-mail и ручная верификация находок пентестерами.	установка агентов не требуется). Предоставляется компаниям в формате подписки, on-prem развёртывание не предусмотрено. Рассчитан на организации среднего размера и выше, которым нужен постоянный мониторинг безопасности. Соответствует требованиям регуляторов (например, ЦБ РФ) по регулярному анализу защищённости.	количеству активов (домены, IP) и частоте сканирований/ручных тестов в месяц. Гибкие тарифы под разные масштабы (без урезания функционала на младших тарифах). Точная цена определяется индивидуально; для ориентира, для десятков активов – сотни тысяч ₽ в год, для крупных периметров – выше.	компании без собственных средств непрерывного сканирования. Часто используется специалистами по ИБ/DevSecOps в финтех, e-commerce, ИТ-стартапах, которым нужен аутсорсинговый мониторинг уязвимостей на постоянной основе.

Сильные и слабые стороны ключевых решений:

ScanFactory:

Плюсы:

- сочетание множества сканеров (Nessus, Nmap, Burp и др.) дает высокое покрытие уязвимостей и типов ресурсов; поддерживается как SaaS, так и on-prem, что удобно для разных требований инфраструктуры;
- есть интеграция с процессами клиента (SIEM, таск-трекеры) и оперативные оповещения (в т.ч. в Telegram) для быстрого реагирования.

Минусы:

- короткий пробный период (1 неделя) затрудняет полноценную оценку;
- документация на русском менее подробна, основные материалы доступны лишь в англоязычной версии продукта;
- недостаточно гибкая аналитика по устранённым уязвимостям (нет отдельного трекинга динамики их исправления);
- интерфейс отображает алерты списком без категоризации по типам проблем, что усложняет анализ при большом количестве находок.

MaxPatrol VM:

Плюсы:

- многолетний отечественный продукт с поддержкой всех основных платформ (Windows, Linux, Unix, сетевое оборудование, БД, веб-серверы и пр.);
- низкий процент ложных срабатываний по сравнению с аналогами; встроенные модули комплаенс-аудита помогают проверять соответствие стандартам безопасности (PCI DSS, ГОСТ и др.) прямо в системе;
- есть сертификаты ФСТЭК и ФСБ, необходимые для использования в госорганизациях и критической инфраструктуре.

Минусы:

- отсутствие SaaS-версии – развёртывание и поддержка требуют ресурсов и компетенций на стороне клиента;
- интеграция с внешними облачными сервисами ограничена (ориентирован преимущественно на внутреннюю сеть);
- для сканирования веб-приложений может потребоваться дополнительный модуль или продукт, т.к. фокус – на инфраструктурных уязвимостях;
- модель лицензирования довольно сложная (отдельные модули за отдельную плату, масштабирование может быть дорогостоящим при большом числе узлов).

BI.ZONE CPT:

Плюсы:

- сочетает автоматизацию и экспертизу – каждая уязвимость проходит проверку опытными пентестерами, что повышает качество результатов (минимум ложных находок и сразу рекомендации по исправлению);
- удобное взаимодействие с командой BI.ZONE через комментарии в платформе ускоряет устранение проблем;
- реализовано распределение прав и сегментация – можно мониторить уязвимости по подразделениям, ролям и филиалам в крупной компании;
- есть возможности сравнения отчётов во времени для отслеживания прогресса безопасности.

Минусы:

- предназначен в основном для крупных предприятий – для среднего и малого бизнеса экономически нецелесообразен;
- отсутствует автоматическая отправка уведомлений в сторонние каналы (мессенджеры), вся работа идет внутри самой платформы;
- стоимость решения высокая, и оно недоступно «из коробки» – требуется согласование объема работ с провайдером и время на онбординг команды.

CVM (Awillix):

Плюсы:

- “все-в-одном” платформа – помимо сканирования уязвимостей, ведется инвентаризация активов и мониторинг SSL, что сокращает поверхность атаки комплексно;
- гибкая настройка частоты сканирований для разных модулей позволяет адаптировать под нужды компании;

- при обнаружении критических проблем сразу приходят оповещения (в почту или мессенджер);
- найденные уязвимости при необходимости проверяются вручную командой квалифицированных экспертов, дается обратная связь по их устранению.

Минусы:

- функционал молодого сервиса – нет ролевого разграничения доступа (нельзя завести нескольких пользователей с разными правами);
- отсутствует публичный API для интеграции с другими системами;
- некоторые операции (например, изменение списка отслеживаемых активов или данных компании) требуют обращения в поддержку, что снижает автономность пользователя.
- платформа менее известна рынку, чем продукты крупных вендоров, что может вызывать осторожность у некоторых клиентов.

1.4 Архитектура программного продукта, описание основных компонентов

Проект можно разбить на следующие компоненты:

1) Основное API
2) Воркеры сканирований
3) Frontend
4) База данных PostgreSQL
5) Очередь сообщений RabbitMQ

Основное API отвечает за логику работы приложения, воркеры сканирований за работу сканирований, frontend отображает всю логику работы в браузере.

На представленной диаграмме (Рисунок 1) можно увидеть общий принцип работы приложения:

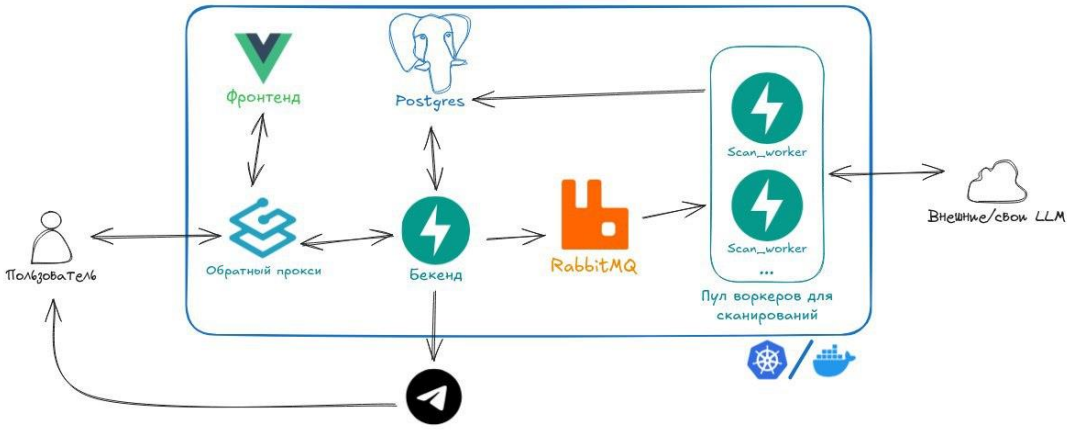

Рисунок 1 - Архитектура приложения

1.5 Описание процесса разработки

1.5.1 Backend

После распределения ролей и выбора архитектуры проекта, наша команда перешла к разработке. В первую очередь началась работа над серверной частью проекта, шаги разработки выполнялись в следующем порядке:

1) реализовано основное API
2) администрация базы данных
3) разработаны воркеры сканирований
4) разработаны конфигурации Docker compose для разных окружений

Ниже представлен список эндпоинтов основного API нашего проекта:

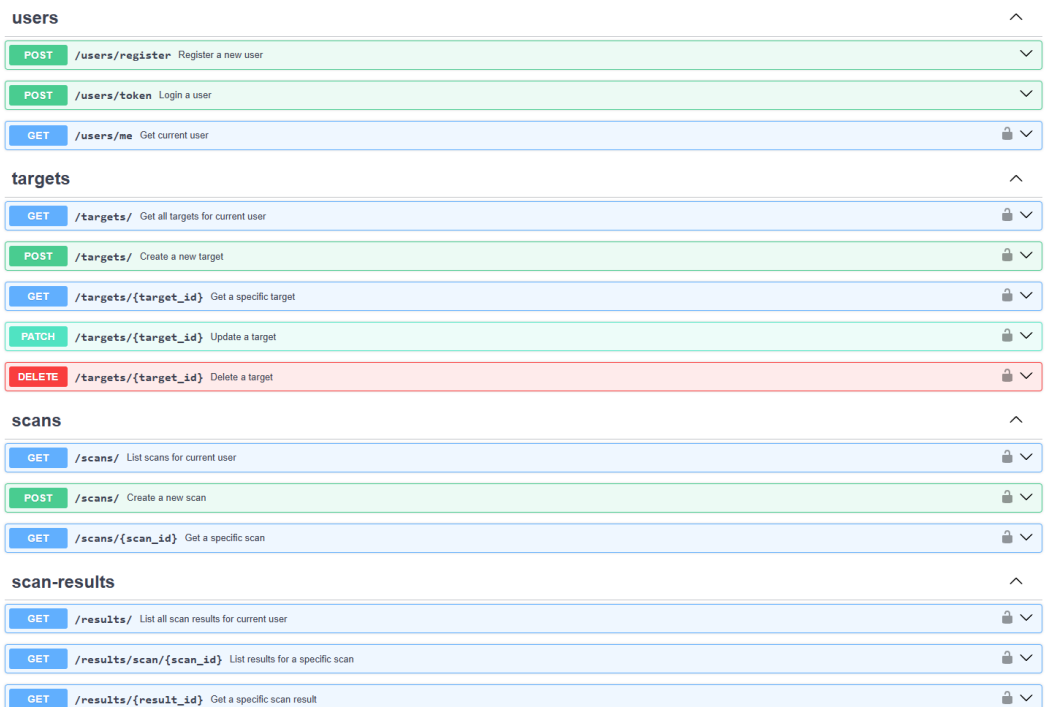

Рисунок 2 - Список эндпоинтов

1.5.2 Frontend

Далее в работу была взята визуальная часть нашего проекта. Из ключевых моментов работы над frontend стоит отметить:

1) Выбор минималистичных приятных цветов и стилей
2) Реализация страниц, который видит пользователь
3) Подключение API к фронтенду

На скриншотах ниже (Рисунки 3 – 10) представлен внешний вид приложения:

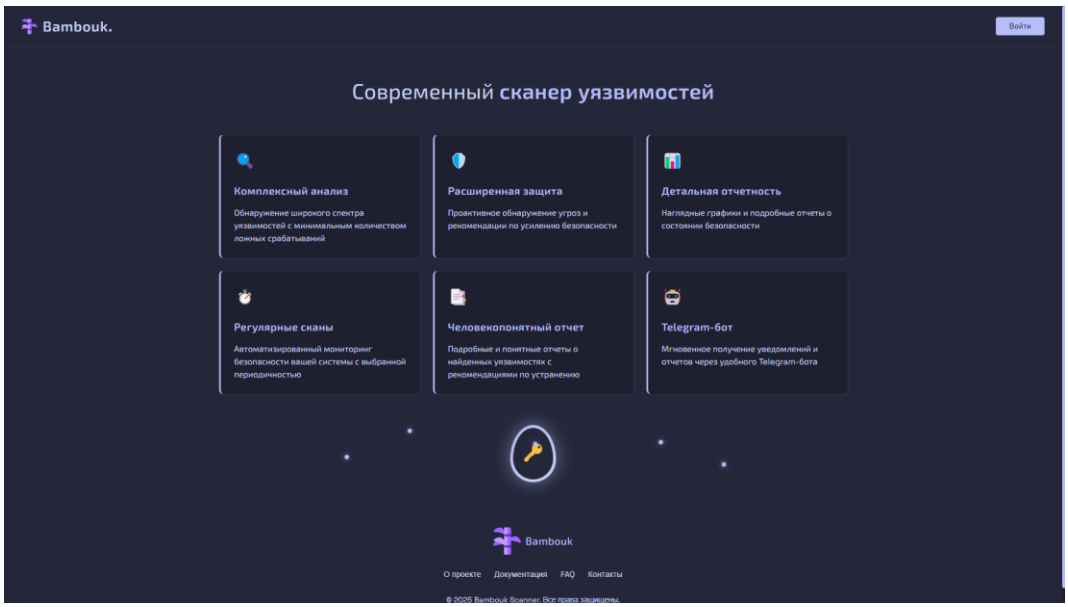

Рисунок 3 - Главная страница

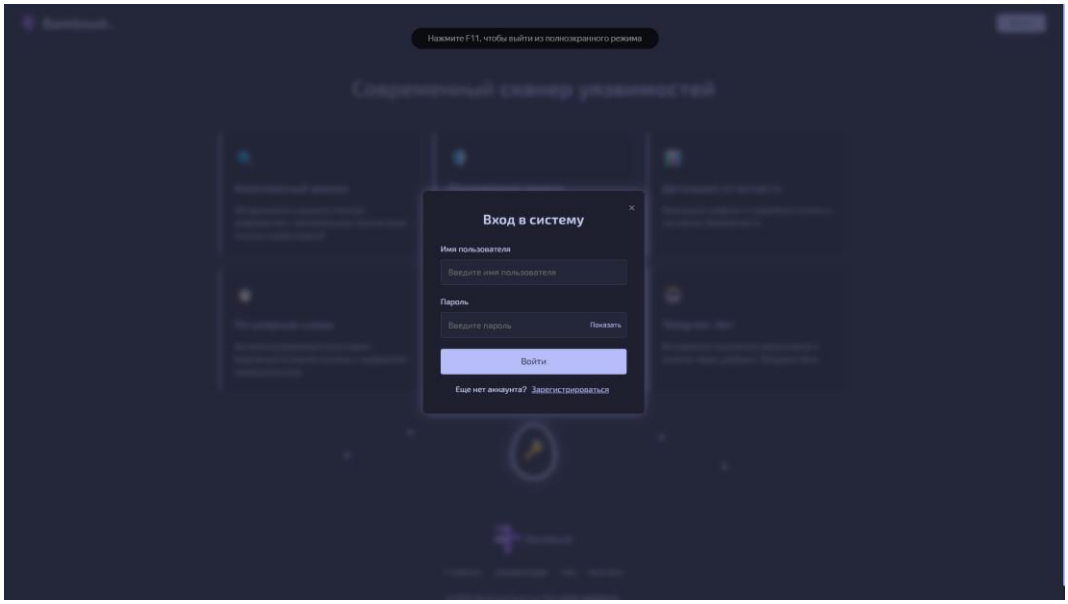

Рисунок 4 - Вход/Регистрация

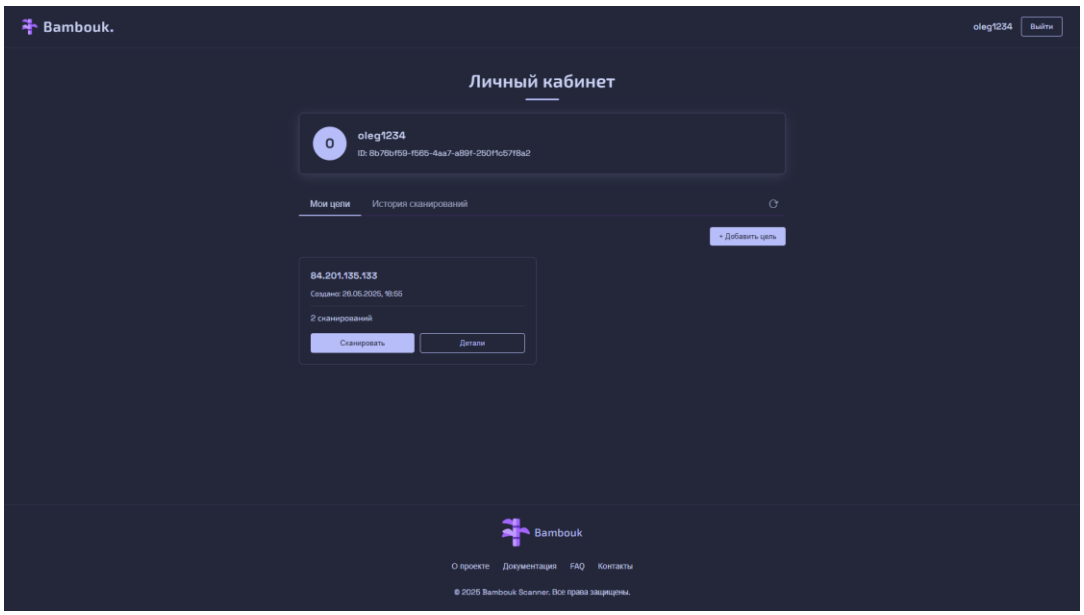

Рисунок 5 - Личный кабинет

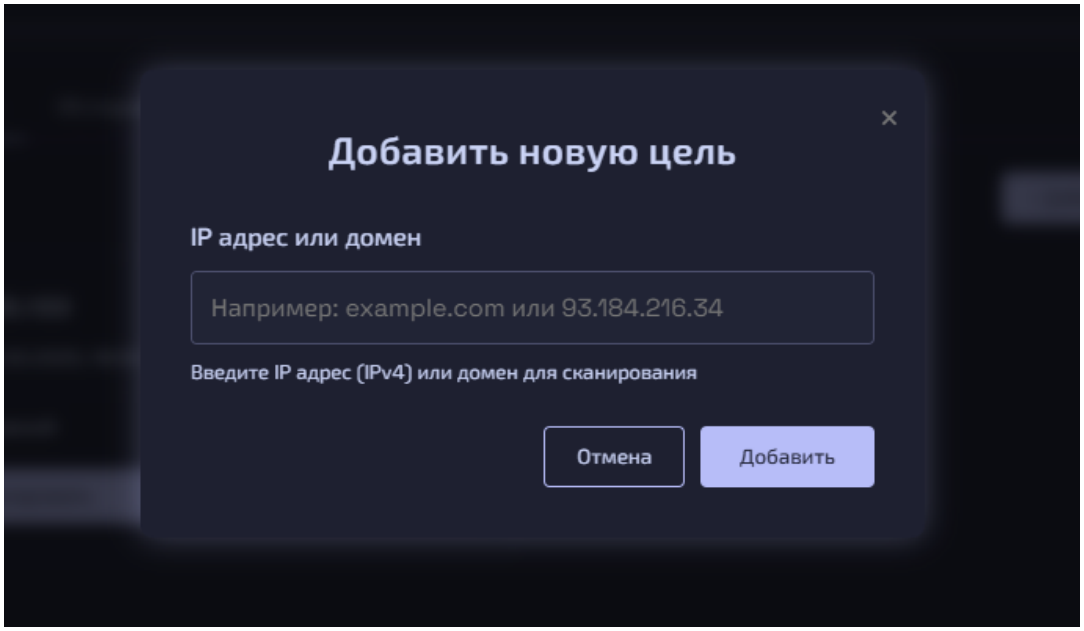

Рисунок 6 - Добавление цели

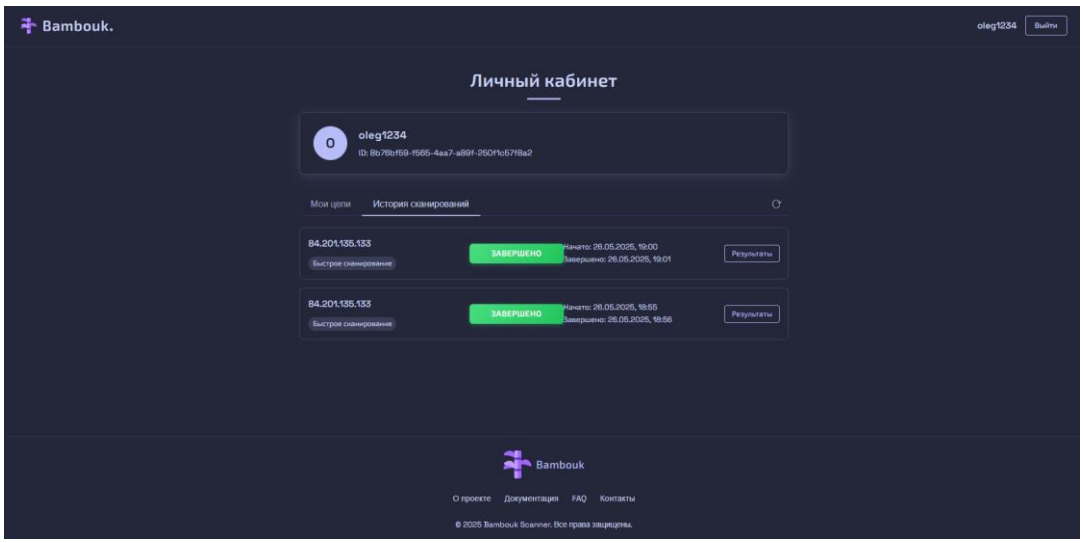

Рисунок 7 - История сканирований

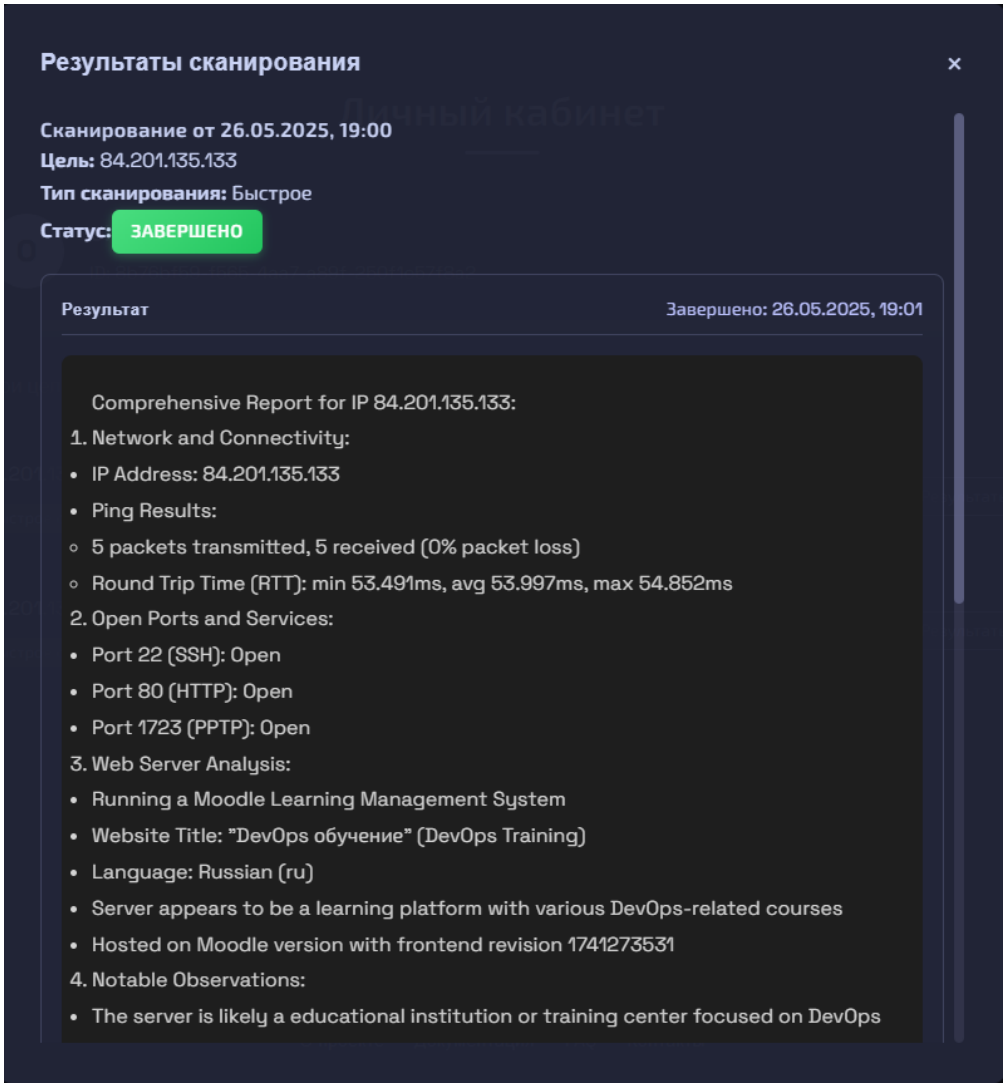

Рисунок 8 - Результаты сканирования

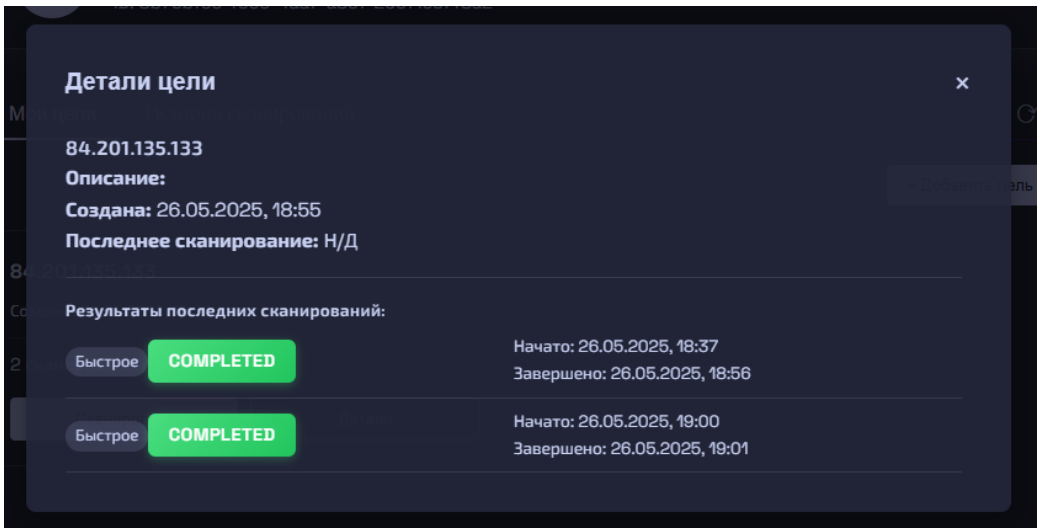

Рисунок 9 - Информация о цели сканирования

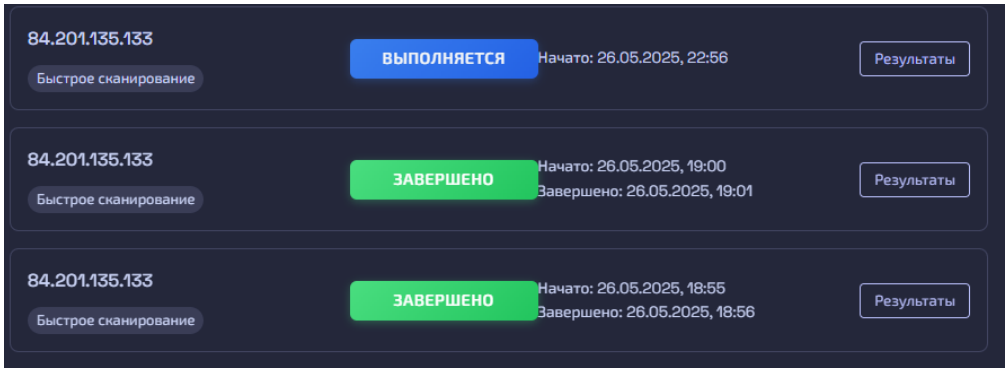

Рисунок 10 - Статусы сканирований

1.5.3 Развёртывание

После того, как MVP была готова и протестирована, мы приступили к развёртыванию проекта. Для этого нам потребовалось настроить виртуальный сервер, установить на него Docker, собрать образ, после чего запустить все компоненты приложения.

1.5.4 Методология разработки и тестирование

В качестве методологии разработки был выбран Agile. Мы вели учёт задач в трекере YouTrack. Нами ставились задачи и велись их статусы. Тестирование проходило следующим образом: после деплоя нового функционала на тестовый стенд, задача отправлялась в столбец QA, после чего она тестировалась и отправлялась дальше, или же при наличии багов отправлялась в статус «В работе».

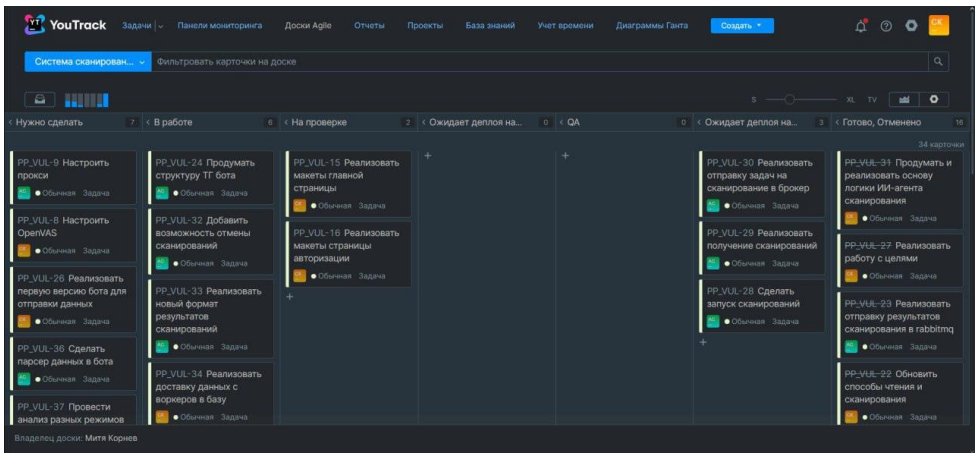

Рисунок 11 - Трекер задач

1.6 Планирование деятельности и распределение задач между участниками команды

Эффективное выполнение проекта потребовало чёткого планирования и скоординированного распределения задач между участниками. С самого начала была выбрана гибкая методология Agile, обеспечивающая адаптацию к изменяющимся требованиям и постоянную обратную связь. Все задачи фиксировались в системе управления проектами YouTrack, где велось отслеживание их статуса (Backlog, В работе, QA, Готово).

Планирование деятельности происходило на еженедельных встречах команды, где формировался список задач и определялись приоритеты. Каждый участник получил определённую роль на проекте, но при необходимости помогал коллегам на смежных направлениях.

Распределение задач:

- **Backend (Черкасов А.И.)**: реализация API, сканеры, архитектура, деплой, работа с базами данных.
- **Frontend (Симонов А.А.)**: дизайн интерфейса, реализация клиентской части, интеграция с API.
- **Аналитика и тестирование (Косолапов С.А.)**: формулировка требований, QA, поиск и фиксация ошибок, ведение документации.

Такой подход позволил команде синхронно продвигаться по проекту, минимизируя риски сбоев и задержек.

ЗАКЛЮЧЕНИЕ

В конечном итоге у нас получилось MVP сканера интернет-устройств, соответствующего всем основным поставленным требованиям. В сервисе можно зарегистрироваться/войти, выбрать цель, произвести сканирование, посмотреть свои цели, изучить результаты сканирований, понятных и читаемых благодаря интеграции с ИИ-моделями.

На данный момент, при тестировании системы грубых ошибок не было выявлено, основные проблемы были устранены во время разработки, чему способствовало использование гибкой методологии разработки Agile.

По нашему мнению, проект является очень перспективным, и имеет большое количество предпосылок к развитию. На стадии разработки сейчас находится телеграм-бот с тем же функционалом, что представлен в веб-приложении, рекуррентные сканирования (такие сканирования выполняются с определенной периодичностью). Они нужны для того, чтобы пользователь мог с нужной периодичностью сканировать какое-то устройство и в случае появлении угрозы, быстро об этом узнать и среагировать.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ScanFactory - Сканфэктори [Электронный ресурс] – URL: https://scan-factory.ru (дата обращения: 09.03.2025).

2. MaxPatrol VM — система управления уязвимостями [Электронный ресурс] – URL: https://ptsecurity.com/ru-ru/products/mp-vm/ (дата обращения: 09.03.2025).

3. BI.ZONE CPT [Электронный ресурс] – URL: https://bi.zone/catalog/products/continuous-penetration-testing/ (дата обращения: 09.03.2025).

4. CVM – Awillix [Электронный ресурс] – URL: https://cvm.awillix.com (дата обращения: 09.03.2025).

5. FastAPI [Электронный ресурс] – URL: https://fastapi.tiangolo.com (дата обращения: 17.04.2025).

6. RabbitMQ Documentation [Электронный ресурс] – URL: https://www.rabbitmq.com/docs (дата обращения: 22.04.2025).

7. Vue.js - The Progressive JavaScript Framework | Vue.js [Электронный ресурс] – URL: https://vuejs.org (дата обращения: 03.05.2025).