Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет: Информационных технологий  
Кафедра «Информационная безопасность»

Направление подготовки/ специальность: 10.03.01 Информационная безопасность.

ОТЧЕТ

по проектной практике

Студент: Чарыев Аллагулы Группа: 241-353

Место прохождения практики: Московский Политех, кафедра Информационная безопасность

Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: Гневшев Александр Юрьевич

Москва 2025

**Оглавление**

[**ВВЕДЕНИЕ** 3](#_Toc198247026)

[**1.** **Общая информация о проекте** 3](#_Toc198247027)

[**2.** **Общая характеристика деятельности организации** 3](#_Toc198247028)

[**3.** **Описание задания по проектной практике** 4](#_Toc198247029)

[**4.** **Описание достигнутых результатов по проектной практике** 9](#_Toc198247030)

[**ЗАКЛЮЧЕНИЕ** 10](#_Toc198247031)

[**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 10](#_Toc198247032)

## **ВВЕДЕНИЕ**

Учебная практика, выполненная студентом группы 241-353 Чарыевым Аллагулы, была направлена на изучение и практическое применение навыков в области информационной безопасности. В процессе практики особое внимание было уделено работе с CTF-задачами, веб-разработке, разработка веб-мессенджера, а также анализу современных подходов к киберзащите, включая участие в мероприятиях от ведущих компаний отрасли.

## **Общая информация о проекте**

**Название проекта:** Киберполигон.

**Цели и задачи проекта. Цель:** Получение практических навыков в сфере информационной безопасности и веб-разработки, а также закрепление знаний через реализацию тематического проекта. **Задачи:** Изучить CTF-задачи из открытого репозитория AnaktaCTF/CTF; Ознакомиться с современными направлениями в области ИБ (Web, Crypto, Pwn и др.) Разработать статический сайт с учебной структурой. Реализовать веб-мессенджер с использованием Flask. Участвовать в мероприятиях по линии ИБ и познакомиться с продуктами компании R-Vision.

## **Общая характеристика деятельности организации**

**Наименование заказчика:** Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет» (Московский Политех)

**Организационная структура:** Московский Политех представляет собой крупный многопрофильный университет, включающий в себя различные институты, факультеты и кафедры. В структуру университета входит Факультет информационных технологий, в рамках которого работает кафедра «Информационная безопасность» — заказчик и куратор проектной практики. Практика студента проходила под научным руководством преподавателя кафедры и была организована в формате проектной работы с техническим и исследовательским уклоном.

**Описание деятельности:** Московский Политех активно занимается подготовкой специалистов в области информационной безопасности, информационных технологий и смежных дисциплин. Учебный процесс направлен на развитие как фундаментальных теоретических знаний, так и практических навыков, включая моделирование и анализ киберугроз, разработку программных решений, а также участие в соревнованиях и проектах, связанных с CTF-задачами и ИБ-аналитикой. Кафедра «Информационная безопасность» на регулярной основе реализует учебные курсы и практики, направленные на изучение инструментов киберзащиты, реагирования на инциденты, а также создание собственных прототипов средств защиты информации. В рамках данной практики кафедра выступила **заказчиком проекта «Киберполигон»**, предоставив техническое задание на разработку учебного информационного ресурса, основанного на CTF-задачах. Также в ходе практики студент участвовал в выездных мероприятиях и взаимодействовал с представителями профессионального сообщества, включая специалистов компании R-Vision. Таким образом, Московский Политех в лице своей кафедры не только обеспечил методическое сопровождение проектной практики, но и выступил как заинтересованная сторона, заинтересованная в применении и трансляции полученного результата для образовательных целей других студентов.

## **Описание задания по проектной практике**

**Базовая часть: создание простого веб-сайта**

**Общая информация о проекте**

* **Название проекта:** Разработка учебного информационного веб-сайта по материалам CTF-задач (на основе репозитория AnaktaCTF)
* **Цели и задачи проекта:** **Цель:** Получение практических навыков в области веб-разработки и информационной безопасности.  
  **Задачи:**
  + Изучить содержание репозитория CTF-задач.
  + Ознакомиться с категориями кибербезопасности (Web, Crypto, Pwn, и др.).
  + Разработать статический веб-сайт на HTML/CSS.
  + Визуализировать структуру и содержание изученного материала.

**Описание задания по проектной практике**

Задание заключалось в самостоятельном изучении репозитория AnaktaCTF/CTF, содержащего задачи по информационной безопасности, и в разработке на его основе простого веб-сайта с использованием только HTML и CSS. Сайт должен содержать структурированную информацию по категориям CTF и наглядно представлять ключевые аспекты информационной безопасности.

**Описание достигнутых результатов по проектной практике**

* Ознакомление с содержанием репозитория и разбором задач (Web, Crypto, Reverse, Pwn и др.).
* Разработка статического веб-сайта, включающего:
  + Главную страницу;
  + Категории CTF с описанием;
  + Простое оформление с использованием CSS;
* Публикация проекта в GitHub.

В процессе создания и анализа статического сайта научного журнала становится очевидным, что для повышения функциональности, удобства использования и масштабируемости проекта потребуется изучение и внедрение ряда дополнительных веб-технологий. Ниже перечислены ключевые направления, освоение которых может значительно улучшить как пользовательский опыт, так и технические характеристики сайта:

1. **HTML5 и семантическая вёрстка** Хотя базовый HTML уже используется, углублённое изучение **HTML5** позволит правильно структурировать контент с помощью семантических тегов (<header>, <article>, <section>, <nav>, и т.д.). Это улучшит читаемость кода, доступность и индексирование сайта поисковыми системами.
2. **CSS3 и адаптивная вёрстка (Responsive Design)** Для современного сайта важно, чтобы он корректно отображался на мобильных устройствах. Изучение **медиазапросов**, **Flexbox** и **CSS Grid** позволит создать адаптивный интерфейс. Также можно внедрить переменные CSS и кастомные свойства для более удобной настройки тем оформления.
3. **Системы контроля версий (Git/GitHub)** Для командной разработки, отслеживания изменений и развёртывания сайта желательно использовать **Git** и платформы вроде **GitHub**, что также упростит публикацию проекта через GitHub Pages.

В рамках проектной практики удалось изучить реальные материалы по кибербезопасности и закрепить навыки веб-разработки. Выполненное задание позволило систематизировать знания и получить опыт разработки проекта "с нуля". Сайт может быть использован в учебных целях другими студентами. Задача была выполнена полностью и соответствует поставленным целям. Полученные навыки имеют практическую ценность для заказчика (учебного заведения), поскольку позволяют применять теоретические знания на практике.

**О компании R-Vision и событиях конференции**

**R-Vision** — российская компания, специализирующаяся на разработке решений в области информационной безопасности. Основное направление деятельности компании — создание интегрированных платформ и инструментов, предназначенных для автоматизации процессов управления инцидентами ИБ, уязвимостями, активами, рисками и реагирования на киберугрозы. Благодаря своим передовым технологиям, R-Vision занимает прочные позиции на рынке кибербезопасности и активно сотрудничает с крупными государственными и коммерческими организациями.

Ключевым продуктом компании является **платформа R-Vision EVO**, представляющая собой гибкое, масштабируемое решение корпоративного уровня. Платформа поддерживает горизонтальное масштабирование, обеспечивает высокую отказоустойчивость, оптимизирована под работу с большими объемами данных и позволяет автоматизировать ключевые ИБ-процессы с минимальным участием человека. Благодаря встроенным инструментам no-code/low-code, заказчики могут легко адаптировать решения под свои внутренние процессы.

**Участие в конференции и ключевые события**

В рамках учебной практики состоялось и участие в **конференции, организованной компанией R-Vision**, где обсуждались актуальные темы в сфере кибербезопасности. Конференция включала в себя презентации, демонстрации продуктов, практические кейсы, а также открытые сессии с вопросами и обсуждениями.

Ключевые моменты конференции:

**Презентация платформы R-Vision EVO**: участникам подробно рассказали об архитектуре платформы, ее модульности, возможностях масштабирования и интеграции с другими ИБ-системами.

**Живые демонстрации**: представители компании продемонстрировали работу системы в реальном времени — как происходит автоматическое обнаружение инцидента, его анализ и запуск сценариев реагирования.

**Разбор инцидентов**: были рассмотрены реальные кейсы атак 2024–2025 годов, в том числе инциденты, связанные с фишингом, использованием вредоносного ПО и эксплуатацией уязвимостей в корпоративных системах.

**Обсуждение современных угроз**: эксперты поделились аналитикой по текущим кибертрендам, включая рост атак с использованием искусственного интеллекта, социальную инженерию, а также рост интереса к малозаметным, но долгосрочным вторжениям (APT).

**Секция вопросов и ответов**: студенты могли напрямую пообщаться с ведущими специалистами компании, задать вопросы по технологиям, карьерным возможностям, а также по развитию практических навыков в области ИБ.

Также участникам представили **планы развития платформы**, включая будущие модули, расширение аналитических возможностей, внедрение ИИ-инструментов и расширение автоматизации процессов SOC (Security Operations Center).

**Значение конференции для учебной практики**

Участие в мероприятии позволило не только расширить понимание современных решений в области ИБ, но и получить представление о том, как работает отрасль «изнутри». Студенты познакомились с реальными сценариями кибератак и узнали, какие подходы применяются для их предотвращения и анализа. Это дало ценную практическую основу для дальнейшей проектной работы и анализа реальных инцидентов в рамках практики.

**Вариативная часть: Разработка базового веб-сервера с использованием Flask и реализация простого веб-мессенджера.**

Цель проекта — разработка веб-мессенджера с использованием технологий Flask, Socket.IO и SQLite. Мессенджер предоставляет пользователям возможность регистрироваться, входить в систему и обмениваться текстовыми сообщениями в реальном времени.

**1. Описание архитектуры**

Проект состоит из двух основных частей:

1. **Backend (серверная часть)** — реализован с использованием веб-фреймворка **Flask**, который обрабатывает HTTP-запросы и отвечает за взаимодействие с клиентом. Для хранения данных используется **SQLite** (легковесная база данных). Для обмена сообщениями в реальном времени применяется **Socket.IO**.
2. **Frontend (клиентская часть)** — состоит из веб-страниц, написанных с использованием **HTML** и **CSS** для оформления, а также **JavaScript** для реализации асинхронного обмена сообщениями через **Socket.IO**.

**2. Структура проекта**

Проект организован в виде нескольких каталогов и файлов:

* **/static** — папка для хранения статичных файлов, таких как стили (CSS) и изображения.
* **/templates** — папка с HTML-шаблонами для разных страниц.
* **app.py** — основной файл серверной части, где происходит обработка логики и маршрутизация.
* **chat.db** — база данных SQLite, которая хранит информацию о пользователях и сообщениях.

В папке **templates** находятся такие страницы, как:

* **index.html** — главная страница, на которой отображается список пользователей.
* **login.html** — форма входа для уже зарегистрированных пользователей.
* **register.html** — форма регистрации новых пользователей.
* **chats.html** — страница чата, где отображаются сообщения и осуществляется обмен сообщениями.

**3. Описание компонентов проекта**

**3.1. База данных**

Для хранения данных используется **SQLite**. Проект включает две основные сущности:

1. **Пользователь (User)**: хранит имя пользователя и хешированный пароль.
2. **Сообщение (Message)**: хранит текст сообщения, отправителя и получателя, а также временную метку.

Каждое сообщение связывается с двумя пользователями (отправителем и получателем), а также записывается с временной меткой.

**3.2. Регистрация и вход**

Для работы с пользователями реализованы две основные функции:

* **Регистрация**: новый пользователь вводит своё имя и пароль, который перед сохранением в базе данных хешируется для безопасности.
* **Вход**: при входе в систему пользователи вводят логин и пароль, после чего система проверяет их на соответствие данным в базе (пароль сравнивается с хешем).

При успешной аутентификации пользователь перенаправляется на главную страницу, где отображается список доступных для общения пользователей.

**3.3. Чаты и обмен сообщениями**

Для обеспечения обмена сообщениями в реальном времени используется **Socket.IO**, который позволяет устанавливать двустороннюю связь между клиентом и сервером. Каждый пользователь подключается к уникальной «комнате», которая зависит от его идентификатора и идентификатора собеседника. В рамках этой комнаты происходят обмены сообщениями между участниками.

Когда пользователь отправляет сообщение, оно сохраняется в базе данных и тут же отображается в чате у другого пользователя с использованием **Socket.IO** для обновления страницы в реальном времени.

**3.4. Интерфейс пользователя**

Каждая страница приложения строится на основе базового шаблона, который включает общие элементы интерфейса, такие как меню навигации, кнопки и стили. Страница чата отображает историю сообщений между пользователями, а также поле для ввода нового сообщения.

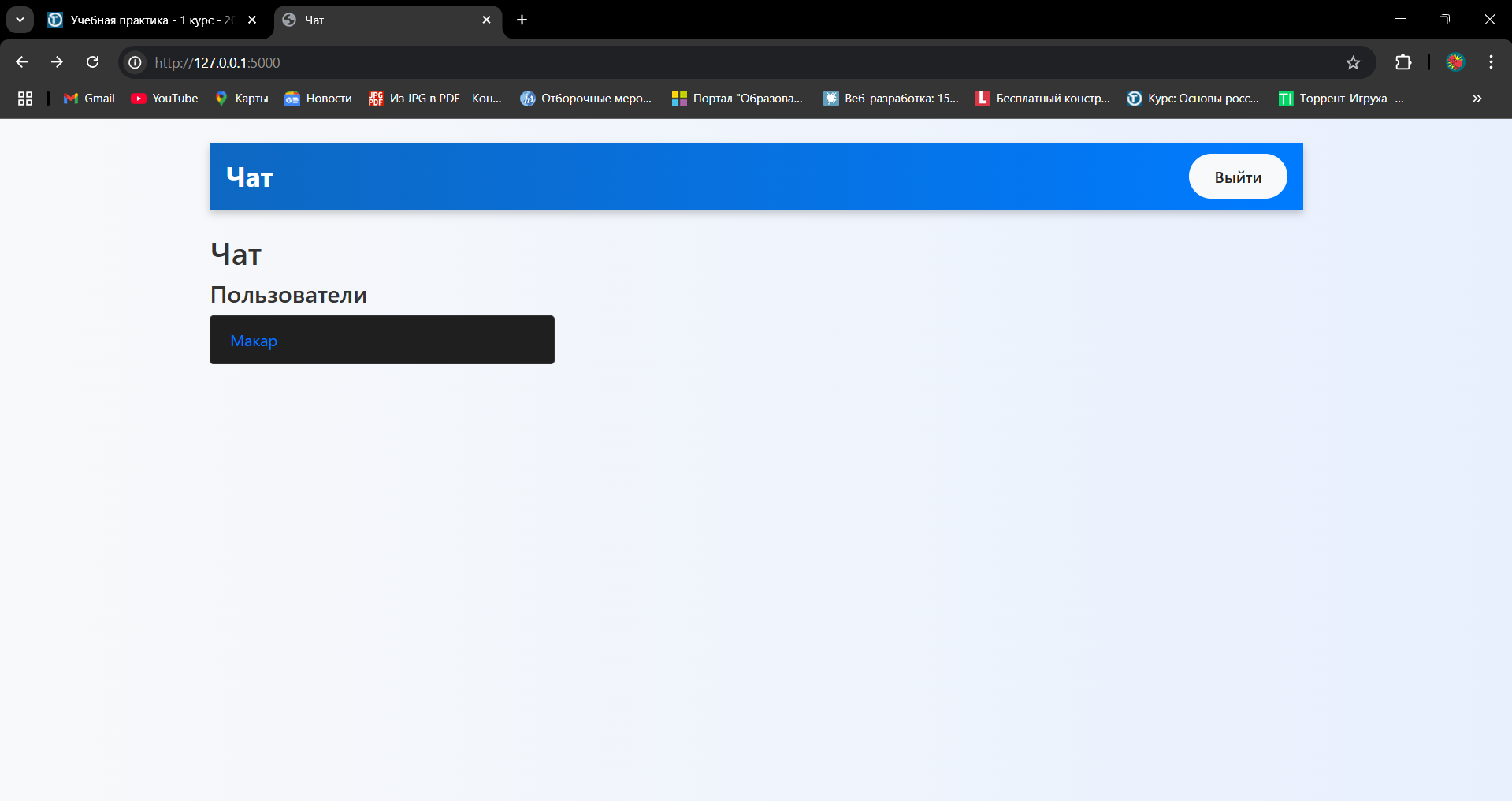


Рисунок Интерфейс пользователя

**4. Технологии, использованные в проекте**

1. **Flask**: используется для создания серверной части веб-приложения. Он обрабатывает HTTP-запросы, маршруты и рендеринг шаблонов.
2. **SQLAlchemy**: ORM (Object Relational Mapper), который позволяет работать с базой данных на уровне объектов Python, упрощая создание, извлечение, обновление и удаление записей в базе.
3. **Socket.IO**: библиотека для реализации обмена сообщениями в реальном времени между клиентом и сервером.
4. **SQLite**: легковесная база данных, использованная для хранения информации о пользователях и сообщениях.
5. **HTML/CSS**: стандартные технологии для построения структуры и стилизации веб-страниц.
6. **JavaScript**: используется для обработки асинхронных запросов и взаимодействия с сервером через **Socket.IO**.

**5. Этапы разработки**

**1. Настройка окружения:**

* Установлены все необходимые библиотеки и зависимости для работы с Flask, SQLAlchemy и Socket.IO.
* Создана структура проекта с папками для шаблонов и статичных файлов.

**2. Реализация базы данных:**

* Определены модели для пользователей и сообщений.
* Настроено подключение к базе данных SQLite через SQLAlchemy.

**3. Создание логики регистрации и входа:**

* Реализованы формы для регистрации и входа с проверкой правильности введённых данных.
* Добавлена хешировка паролей с использованием библиотеки **Werkzeug** для обеспечения безопасности.

**4. Реализация чатов в реальном времени:**

* Настроен **Socket.IO** для создания комнат, в которые могут входить два пользователя для общения.
* Создан механизм обмена сообщениями в реальном времени с автоматическим отображением новых сообщений у обоих пользователей.

**5. Разработка интерфейса:**

* Разработаны страницы для регистрации, входа, чатов и главной страницы с использованием **HTML** и **CSS**.
* Использованы простые и интуитивно понятные элементы управления для пользователей.

**6. Тестирование и отладка:**

* Проверено корректное сохранение и отображение сообщений.
* Осуществлены тесты на функциональность регистрации, входа и обмена сообщениями.
* Исправлены ошибки и улучшены взаимодействия между сервером и клиентом.

**6. Запуск проекта**

Для запуска проекта необходимо выполнить следующие шаги:

1. Установить все необходимые библиотеки через менеджер пакетов **pip**.
2. Инициализировать базу данных с помощью команд Flask.
3. Запустить серверное приложение с помощью команды Python.

После этого веб-мессенджер будет доступен по локальному адресу, и пользователи смогут взаимодействовать друг с другом через чат.

Проект представляет собой веб-мессенджер, в котором реализована регистрация пользователей, их аутентификация и возможность обмениваться сообщениями в реальном времени. Использование **Flask**, **Socket.IO** и **SQLite** позволило создать легковесное, но функциональное приложение. В дальнейшем проект можно развивать, добавляя новые возможности и улучшая интерфейс для пользователей.

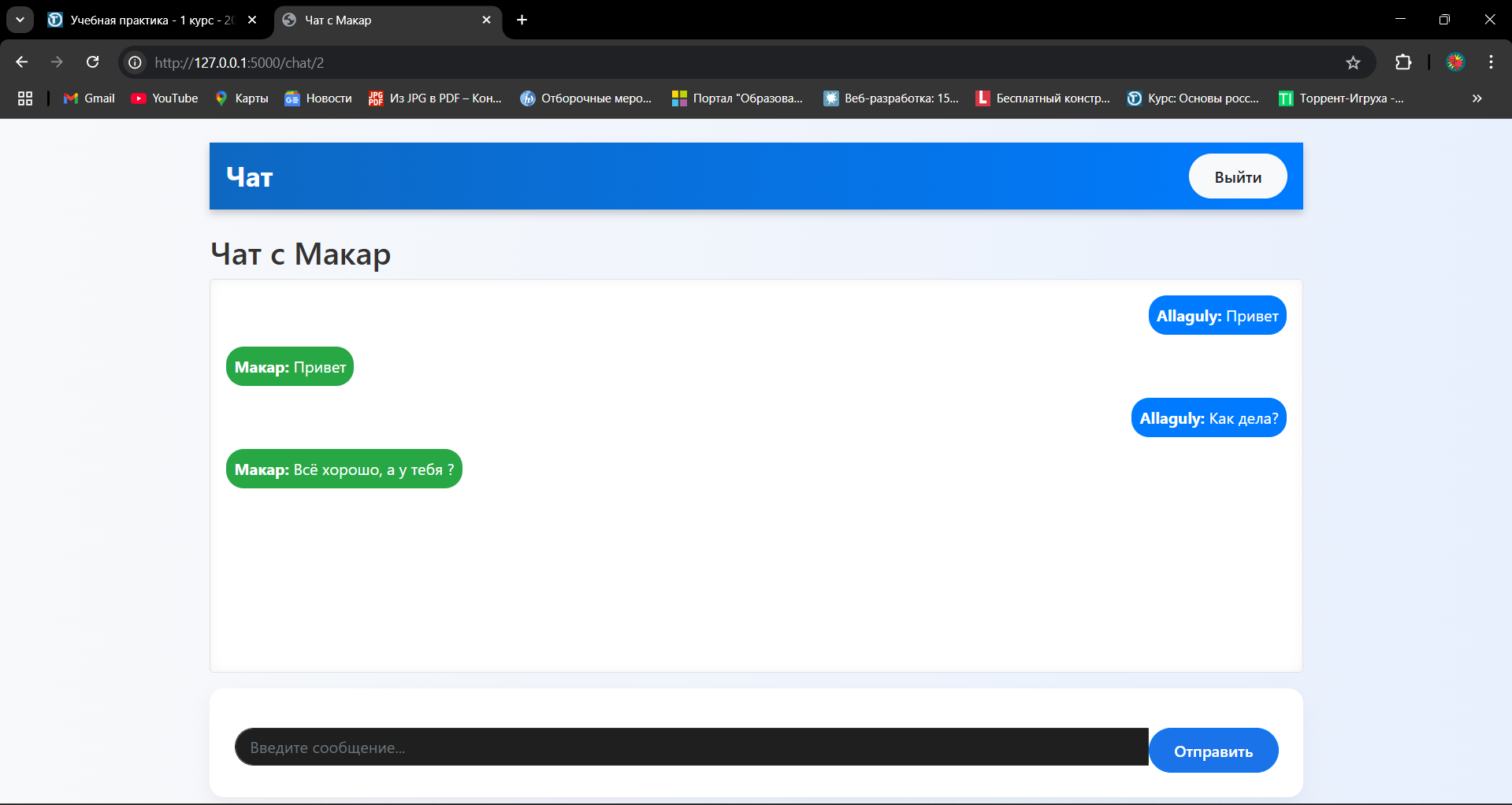


Рисунок Чат с пользователем от имени первого пользователя

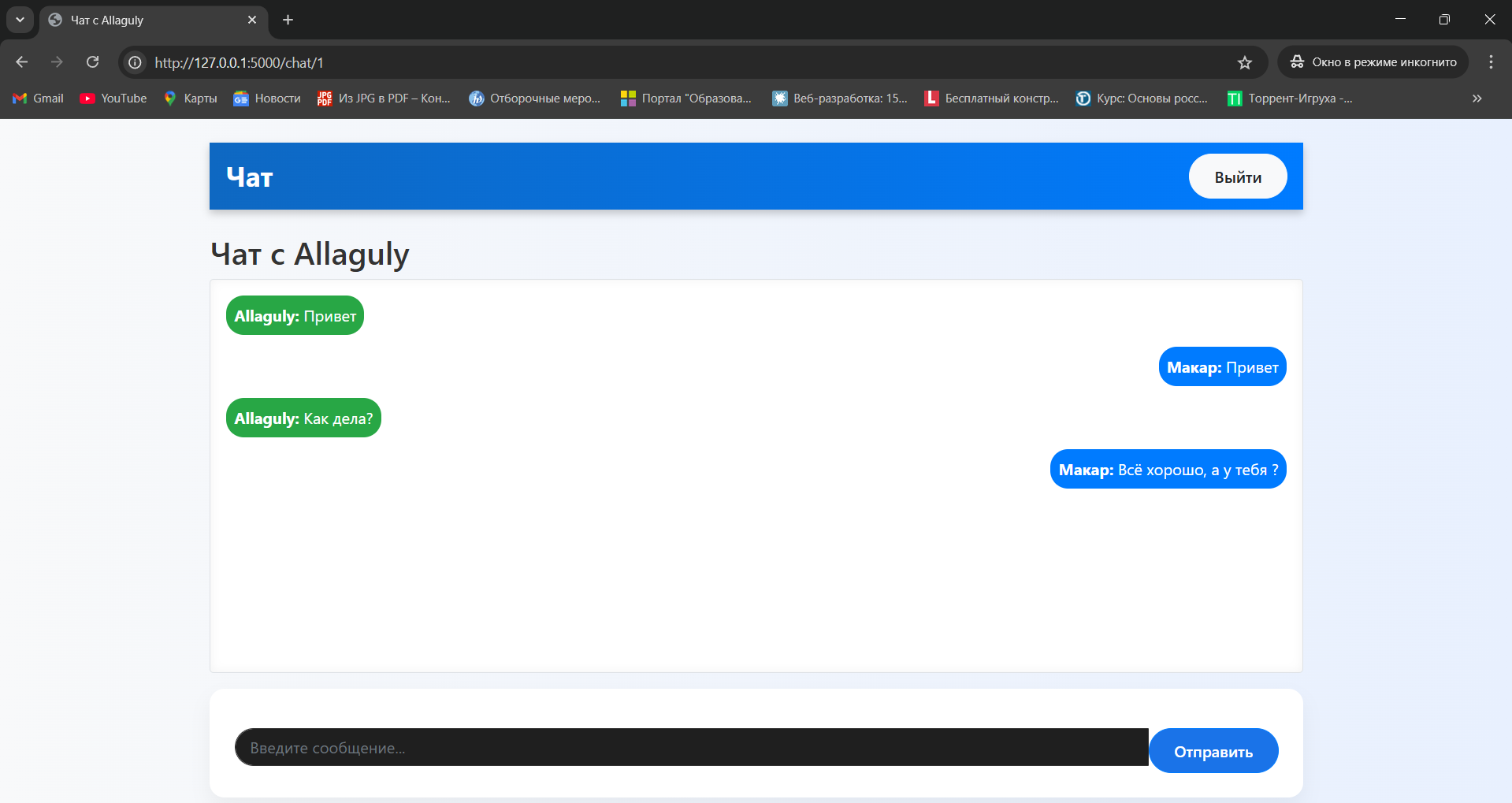


Рисунок Чат с пользователем от имени второго пользователя

## **Описание достигнутых результатов по проектной практике**

В результате практики были достигнуты следующие результаты: Проведен анализ и структурирование CTF-задач (Web, Crypto, Reverse, Pwn и др.) Создан статический сайт с учебными материалами, оформленный с помощью HTML и CSS. Разработан работающий веб-мессенджер на Flask с реализацией авторизации, чатов, базы данных и WebSocket-соединения через Socket.IO. Принято участие в мероприятии компании R-Vision, где были получены знания о современных технологиях киберзащиты и автоматизации процессов реагирования на инциденты. Получен опыт работы с реальными задачами, клиент-серверной архитектурой, регистрацией пользователей, хранением и обработкой данных.

## **ЗАКЛЮЧЕНИЕ**

В процессе прохождения проектной практики мне удалось получить значительный объем как теоретических, так и практических знаний. Работа охватывала сразу несколько ключевых направлений — веб-разработка, информационная безопасность, клиент-серверные взаимодействия, а также анализ реальных киберинцидентов. Особую ценность для меня представляло участие в мероприятии от компании R-Vision, где я смог познакомиться с корпоративными ИБ-решениями и их применением в реальных сценариях. Также работа над проектом «Киберполигон» позволила глубже понять структуру CTF-задач, укрепить навыки в HTML/CSS и реализовать собственное приложение на Flask. Полученные знания и навыки представляют значимую ценность как для меня лично, так и для образовательного учреждения, поскольку они демонстрируют практическое применение учебного материала и могут быть использованы в дальнейшем обучении и карьерном росте.

## **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Репозиторий AnaktaCTF/CTF // Git Hub URL: <https://github.com/AnaktaCTF/CTF>(дата обращения: 01.04.2025).
2. Документация по HTML: // Документация URL: <https://developer.mozilla.org/ru/docs/Web/HTML> (дата обращения: 05.04.2025).
3. Документация по CSS: // Документация URL: <https://developer.mozilla.org/ru/docs/Web/CSS> (дата обращения: 05.04.2025).
4. Учебные материалы по CTF (внутренние ресурсы кафедры / курса)
5. R-vision: <https://rvision.ru/>