МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации



**ЛАБОРАТОРНАЯ РАБОТА №1**

«Разработка SPA приложения»

по дисциплине: «*Технологии и методы программирования*»

|  |  |
| --- | --- |
| Выполнил:  Студент гр. «Абc-323», «АВТФ»  *Рязанов Владислав Андреевич*  «\_\_» \_\_\_\_\_\_ 2025г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Проверил:  *Старший преподаватель кафедры ЗИ*  *Медведев Михаил Александрович*  «\_\_\_» \_\_\_\_\_\_ 2025г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Новосибирск 2025

**Цель работы:**

1. Освоить процесс разработки одностраничного приложения (SPA) с помощью современного фреймворка или библиотеки (React/Vue/Angular).
2. Закрепить на практике концепции RESTful API: отправка запросов (GET/POST/PUT/DELETE), обработка JSON-ответов.
3. Реализовать базовый роутинг без перезагрузки страницы, управление состоянием и динамическую отрисовку интерфейса.

**Задачи:**

1. Подготовить проект по предметной области
2. Составить структуру и роутинг SPA
3. Организовать связь с сервером (RESTful API)
4. Провести работу с JSON-данными

НАСТРОЙКА ФРОНТЕНДА И БЕКЕНДА

Необходимо создать новый проект для фронтенд-разработки, используя современный инструмент сборки интерфейсов. Требуется установить все необходимые зависимости, совместимые с фреймворком Vue.js, после чего запустить клиентскую часть приложения для проверки его работоспособности (Рисунок 1).

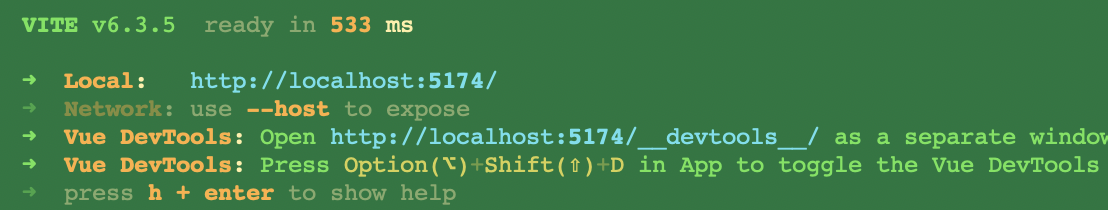


Рисунок 1 – Работа фронтенда

Требуется разработать серверную часть приложения на Python, используя фреймворк Flask. Необходимо создать проект, подключить актуальные зависимости, настроить базовую структуру сервера и запустить его, чтобы убедиться в корректной работе (Рисунок 2).



Рисунок 2 – Работа бэкенда

РАЗРАБОТКА РОУТИНГА

Главная страница веб-приложения доступна по адресу http://localhost:5000/ и представляет собой пользовательский интерфейс системы управления инцидентами. Она включает форму для обработки обращений, а также дополнительные элементы управления (Рисунок 3).

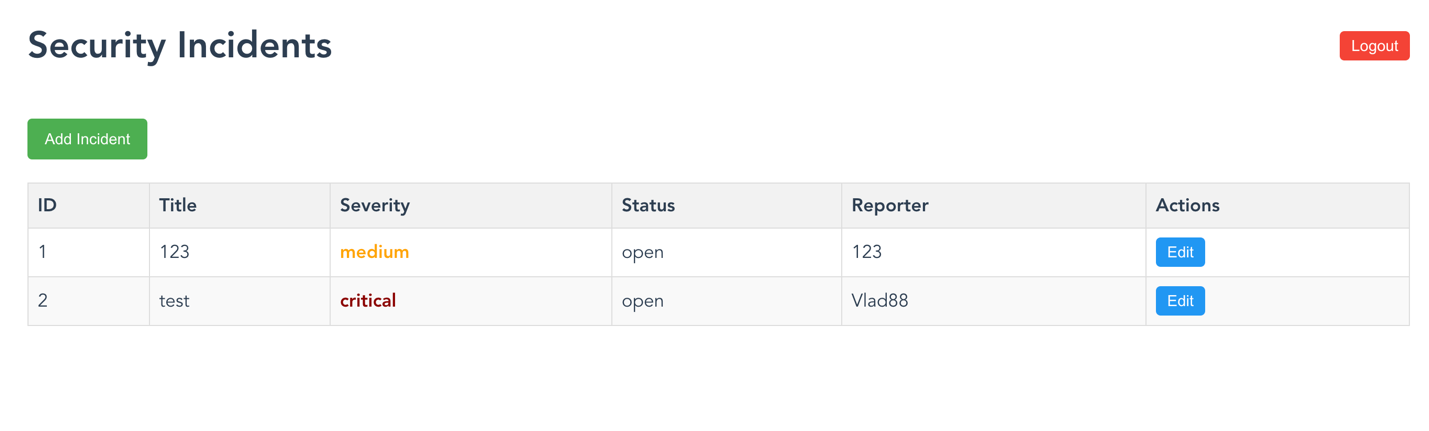


Рисунок 3 – Главная страница

Страница /login позволяет пользователям войти в систему под существующей учетной записью или зарегистрироваться, чтобы получить доступ ко всем возможностям приложения (Рисунок 4).

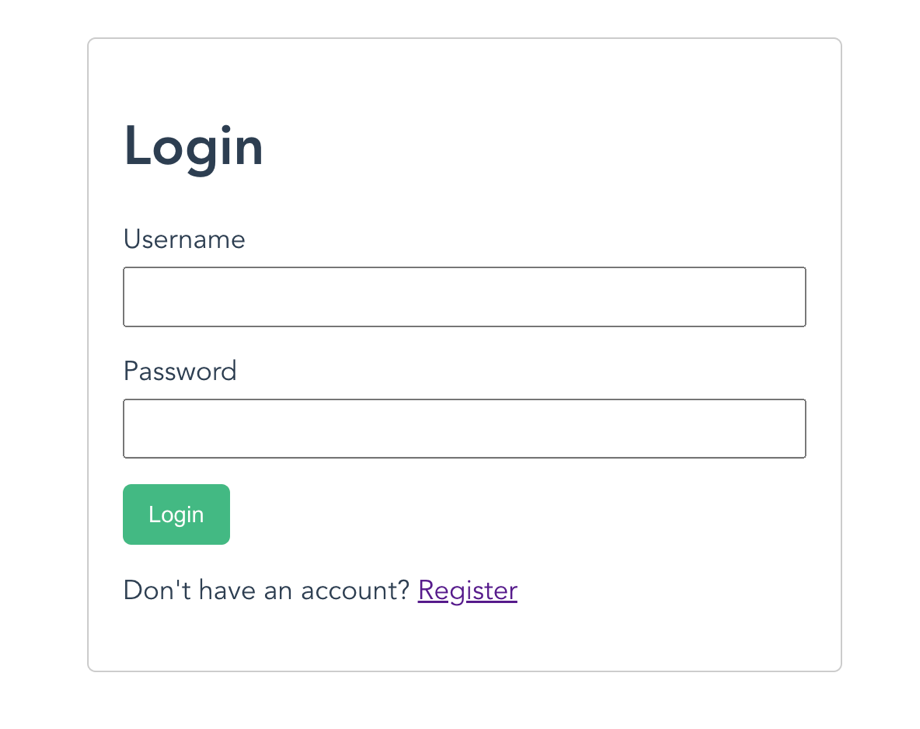


Рисунок 4 – Страница для входа

Страница /register предназначена для новых пользователей и позволяет создать учетную запись в системе через интуитивно понятный интерфейс (см. Рисунок 5).

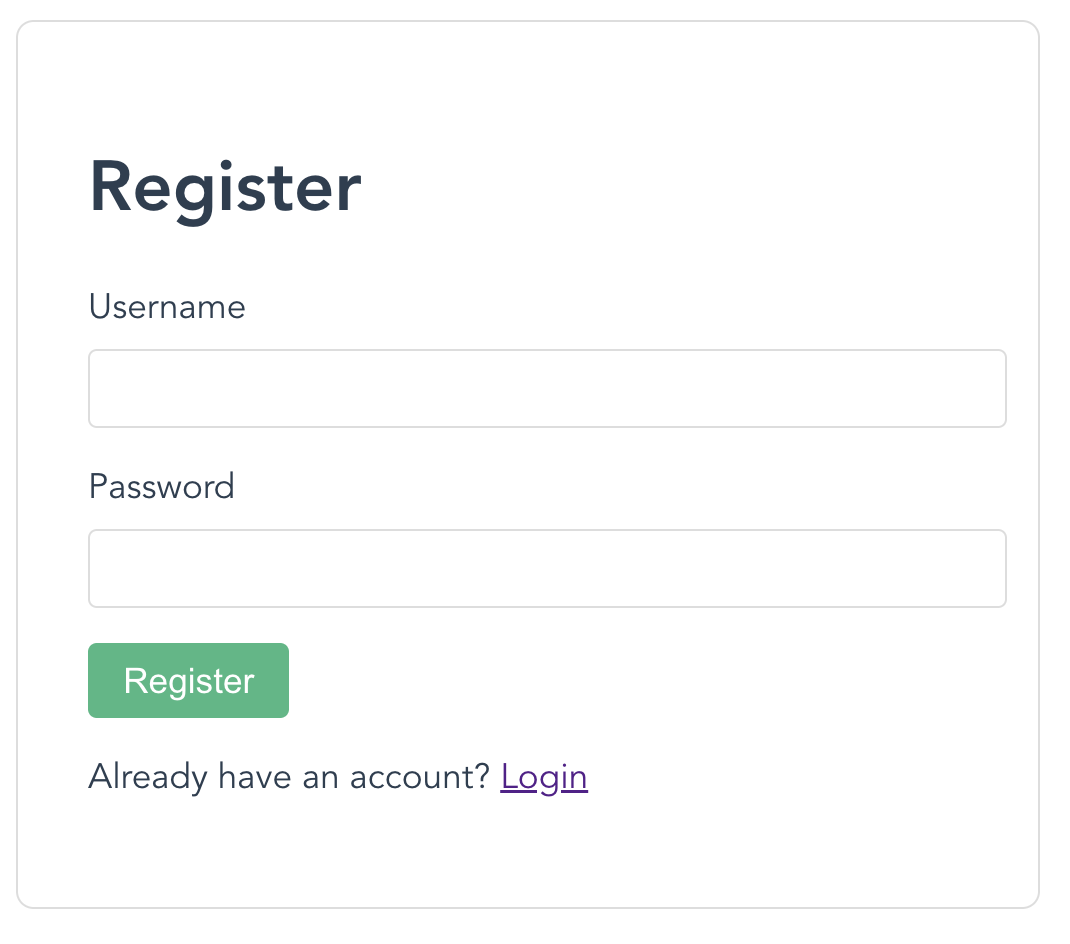


Рисунок 5 – Страница для регистрации

БЭКЕНД ЧАСТЬ

Серверный компонент системы разработан на языке Python с применением фреймворка Flask, который был выбран благодаря эффективности при создании веб-приложений и обеспечивает обработку HTTP-запросов и реализацию механизма маршрутизации. Для обеспечения кросс-доменного взаимодействия между клиентским приложением на Vue.js и сервером используется модуль. CORS.

Функционал регистрации новых пользователей, входа существующих, а так же работы с инцидентами, реализован через обработку POST, GET, DELETE, PUT-запросов, направляемых на специально предназначенные для этого эндпоинты (Рисунок 8-11).

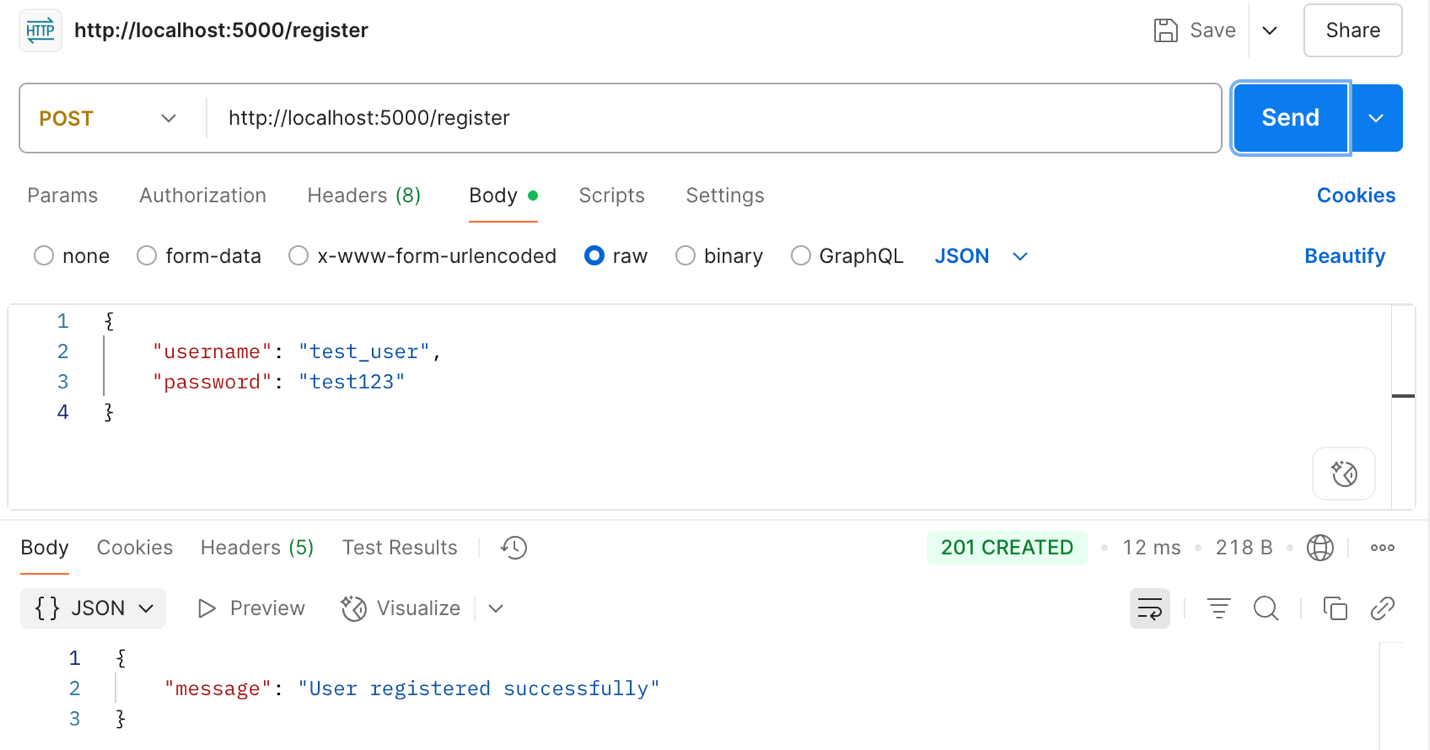


Рисунок 8 – Эндпоинт для регистрации

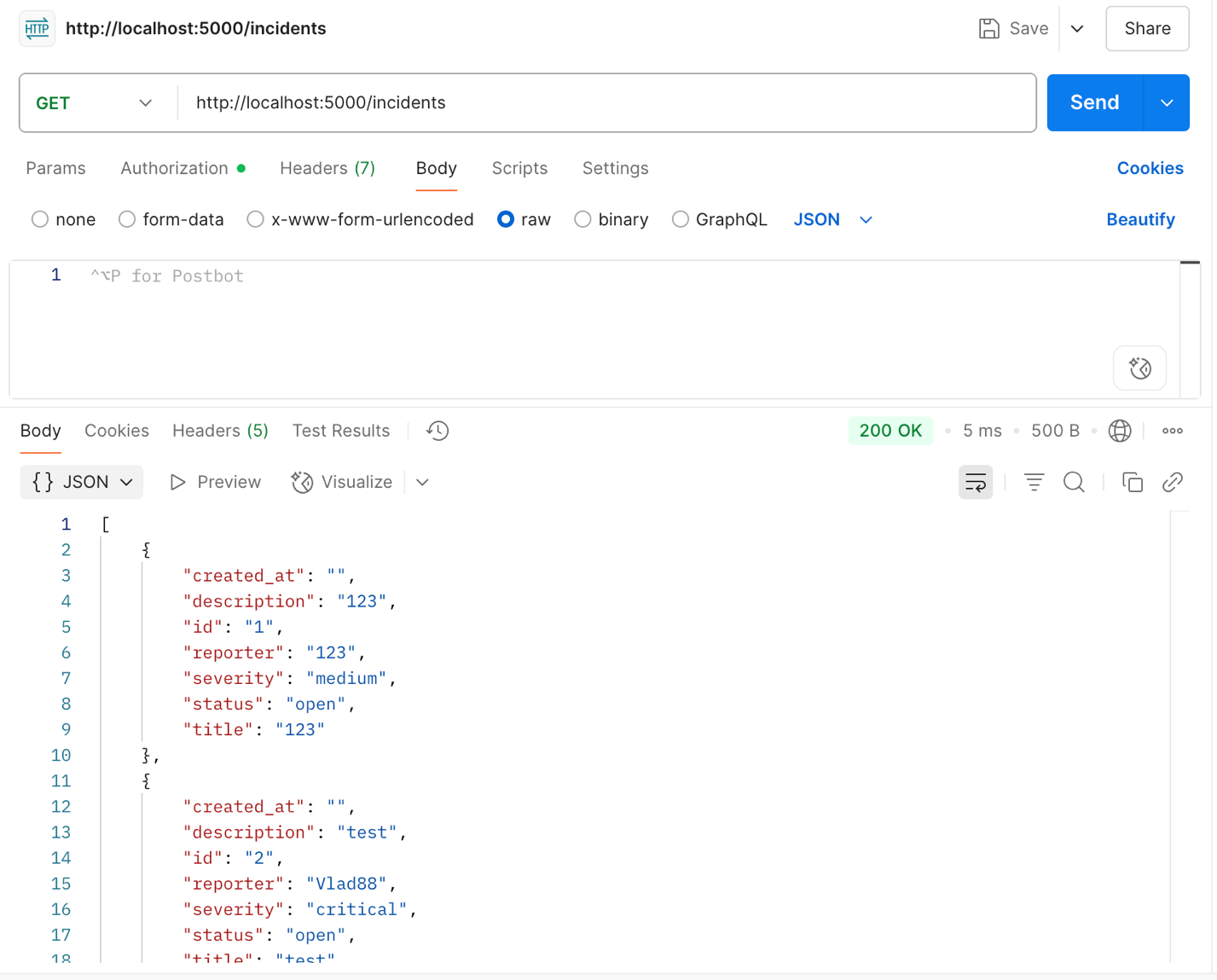


Рисунок 9 – Эндпоинт для получения всех инцидентов

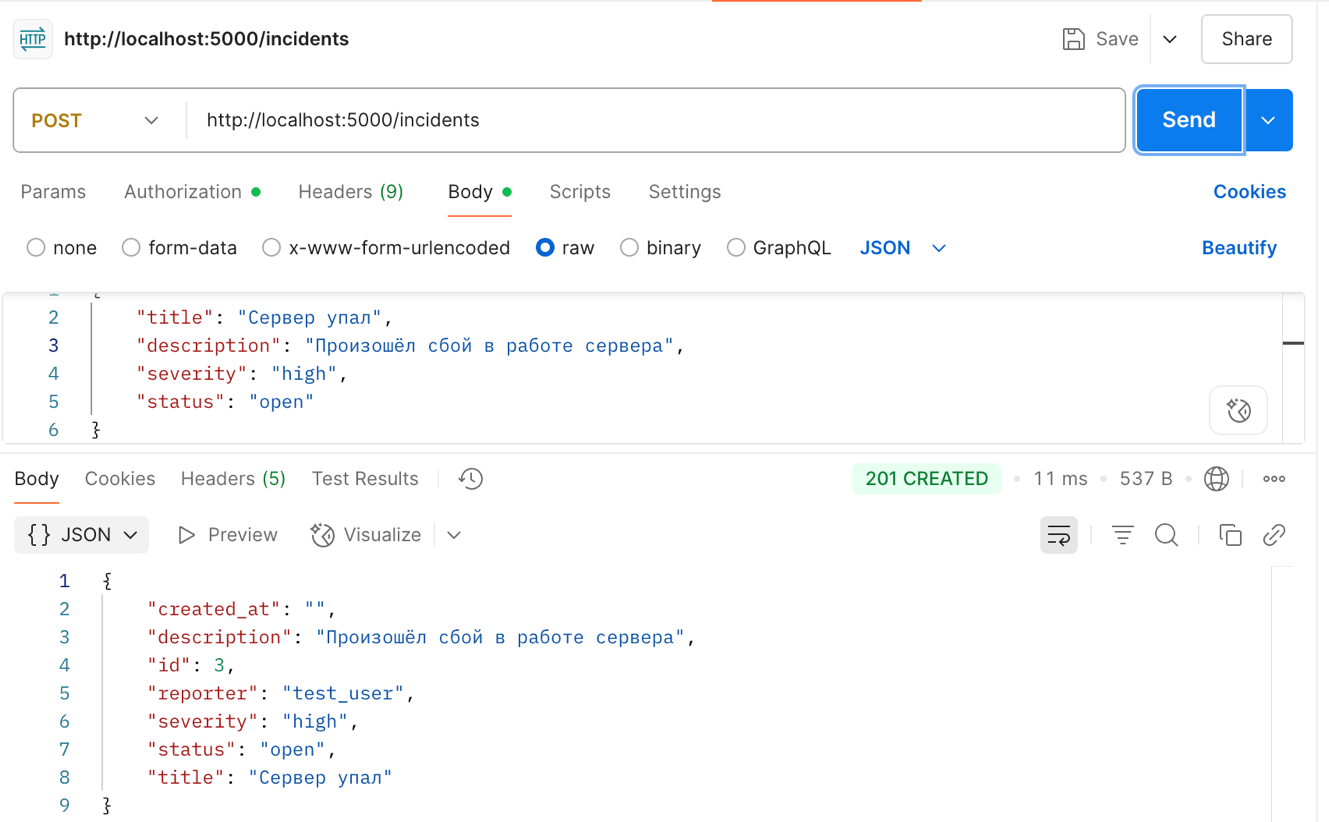


Рисунок 10 – Эндпоинт для добавления инцидентов

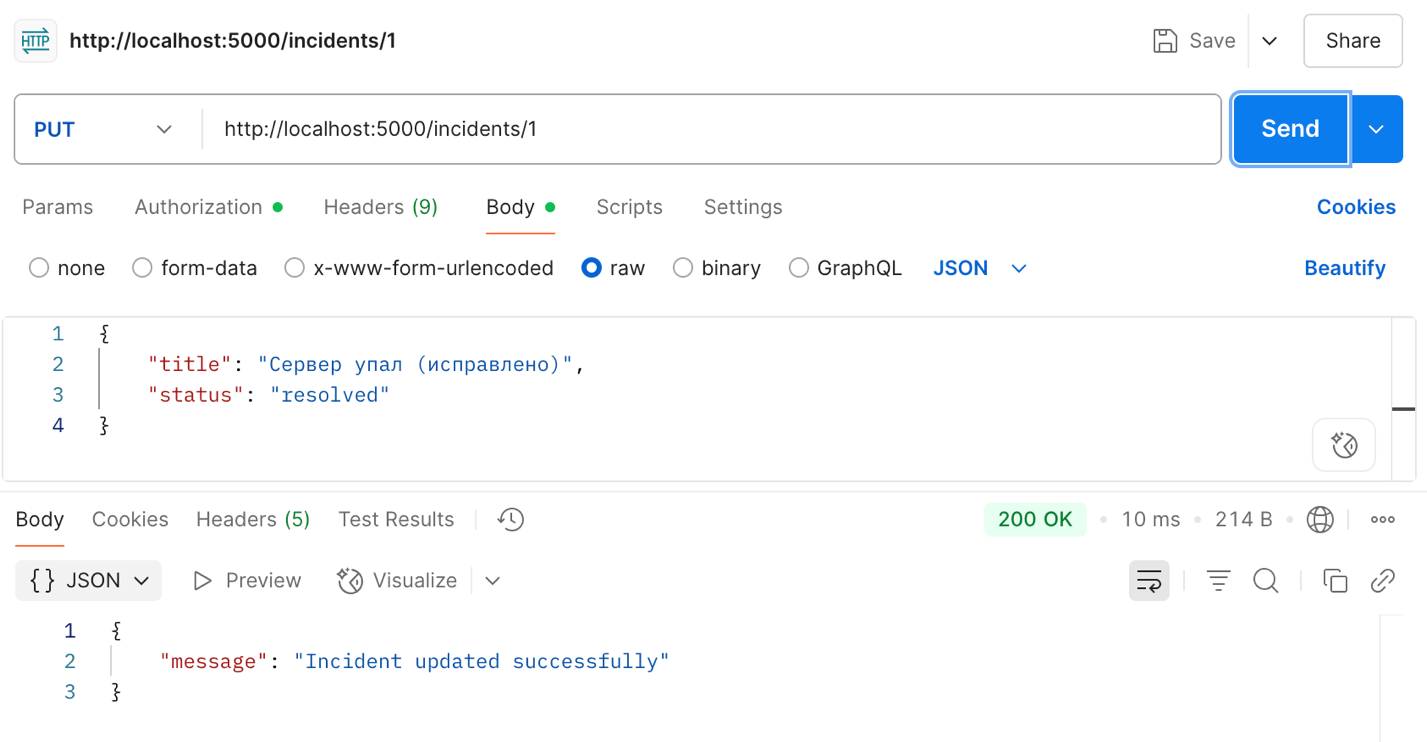


Рисунок 11 – Эндпоинт для обновления инцидента

РАБОТА С JSON

Для сохранения данных о новом инциденте через JSON-запрос необходимо преобразовать информацию в формат, соответствующий требованиям сервера (Рисунок 12).



Рисунок 12 – Преобразования для требований сервера

УПРАВЛЕНИЕ СОСТОЯНИЕМ

Состояние интерфейса поддерживается путем обновления и повторного отображения только тех компонентов, данные которых изменились, что обеспечивает актуальное представление информации (Рисунок 13-14).

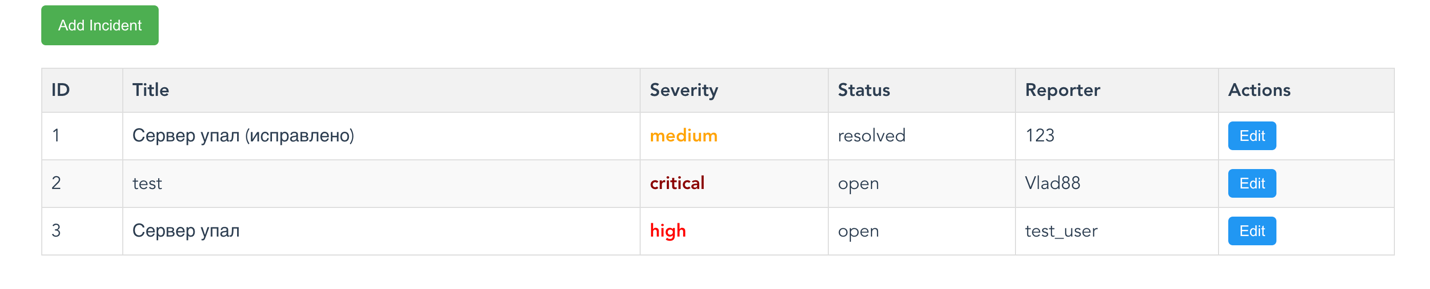


Рисунок 13 – Инциденты до обновления

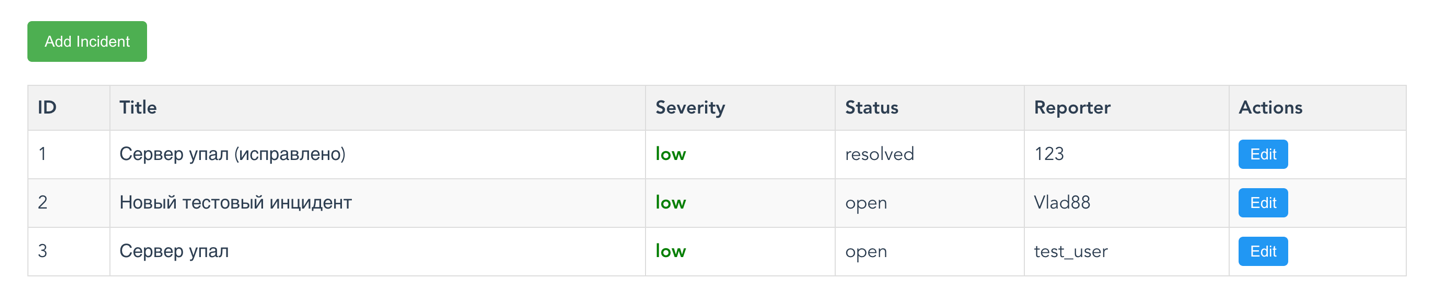


Рисунок 14 – Инциденты после обновления

ДОПОЛНИТЕЛЬНЫЕ ТРЕБОВАНИЯ

На стороне клиента реализована система проверки ввода: контролируется корректность пароля и логина, а также выполняется валидация данных (Рисунок 15-16). Кроме того, во время обработки и загрузки данных в интерфейсе отображается индикатор загрузки (спиннер) непосредственно на кнопке (Рисунок 17).

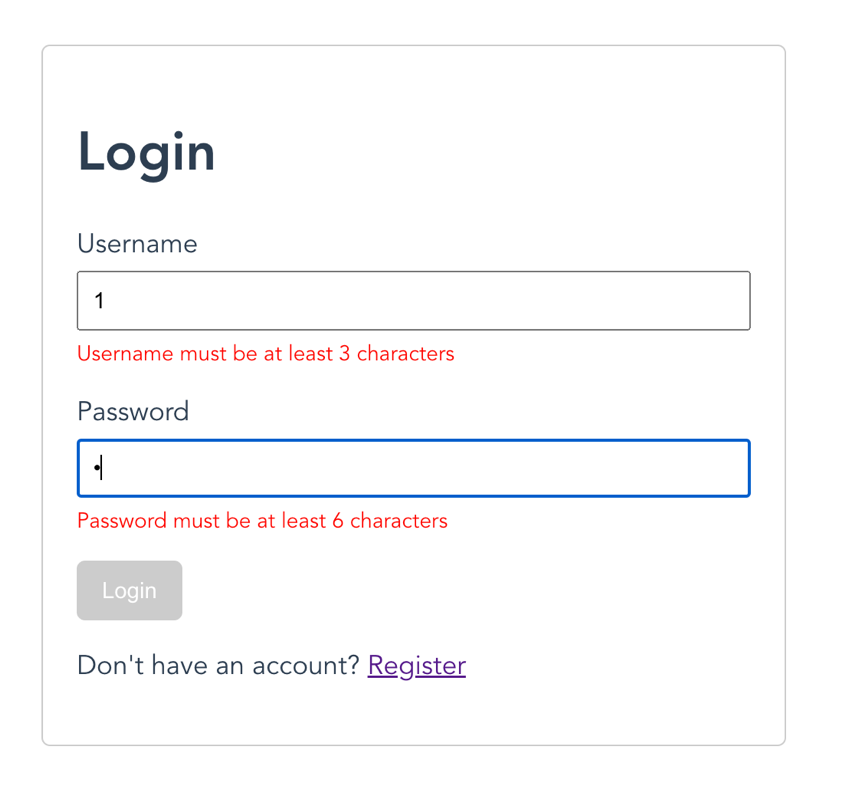


Рисунок 15 – Валидация формы для входа

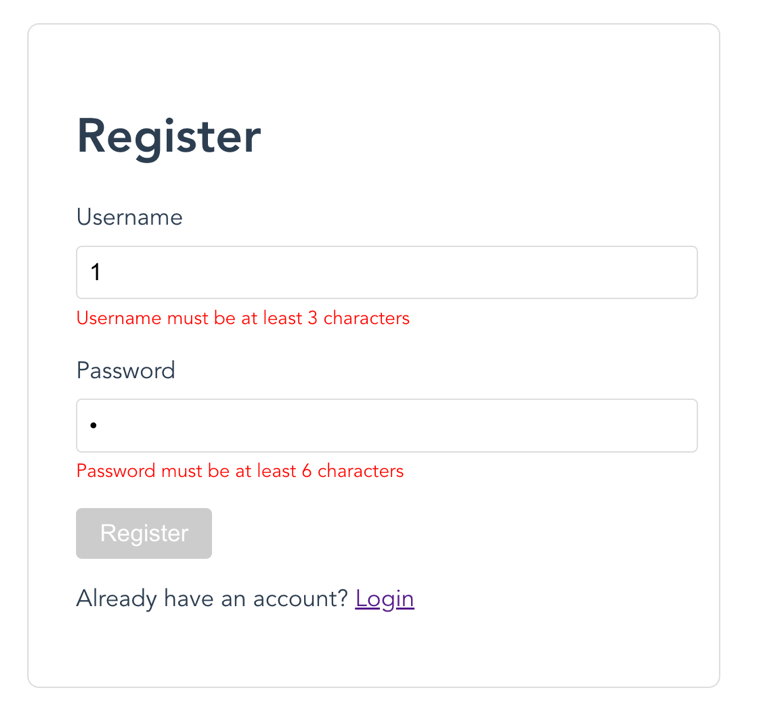


Рисунок 16 – Валидация формы для регистрации

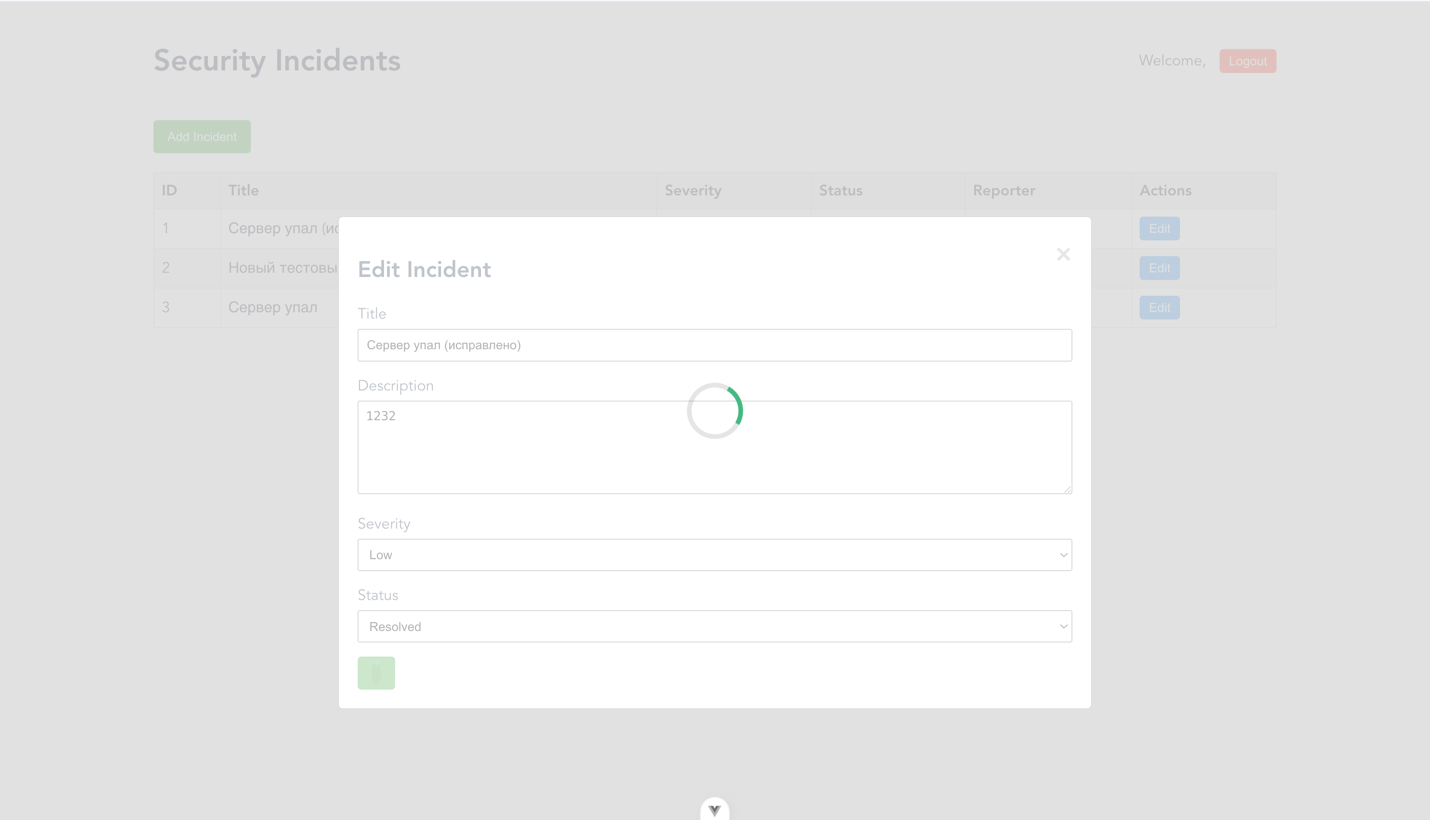


Рисунок 17 – Спинер на главной странице

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

Веб-приложение эффективно реализует функционал для регистрации, а так же последующего управления инцидентами кибербезопасности и обеспечивает вход и регистрацию пользователей. Работа серверной части (Рисунок 18) и фронтенда (Рисунок 1).

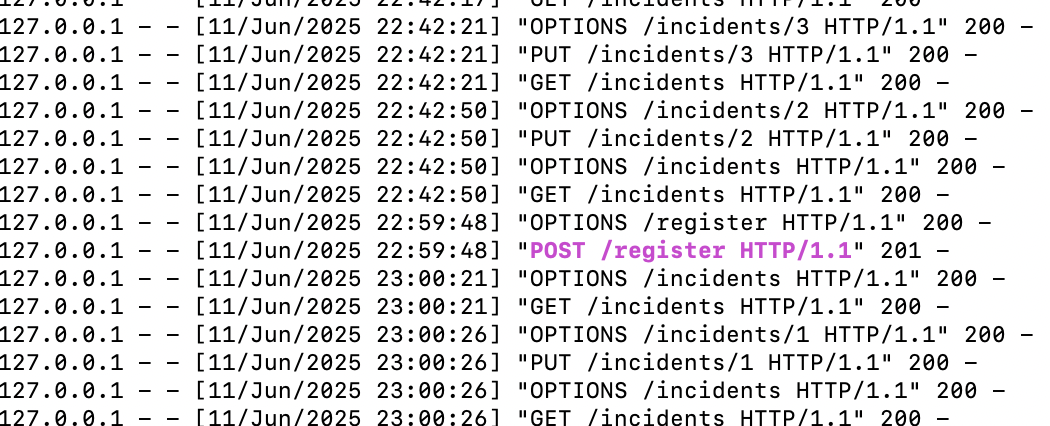


Рисунок 18 – Работа серверной части

ОТЧЕТ О ПРОДЕЛАННОЙ РАБОТЕ

Был реализован почти весь необходимый функционал, который включает в себя регистрацию пользователей, а также управление, регистрацию случаев нарушения безопасности (Рисунок 19-23).

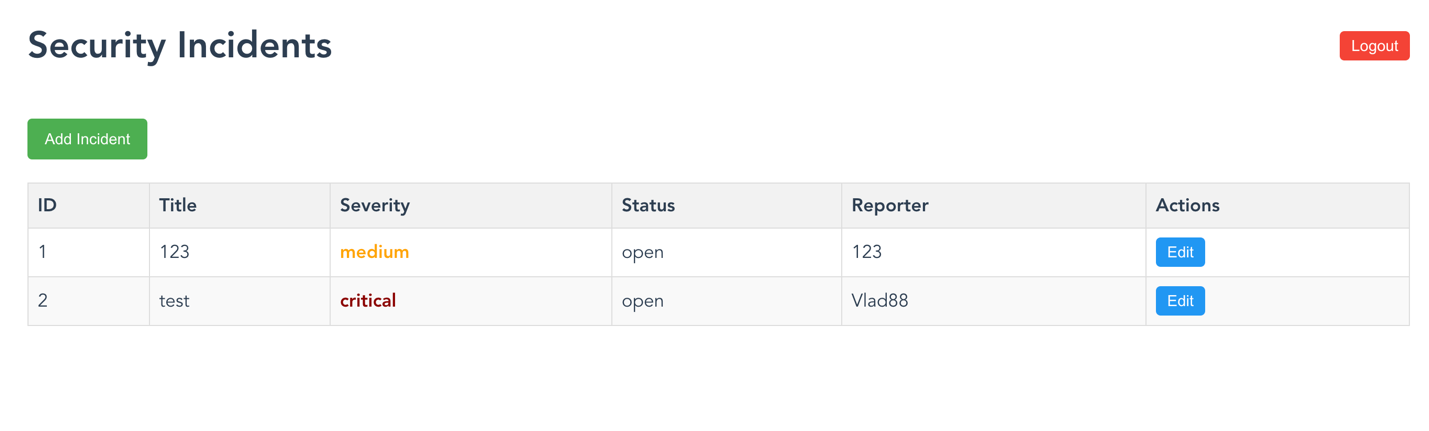


Рисунок 19 – Главная страница

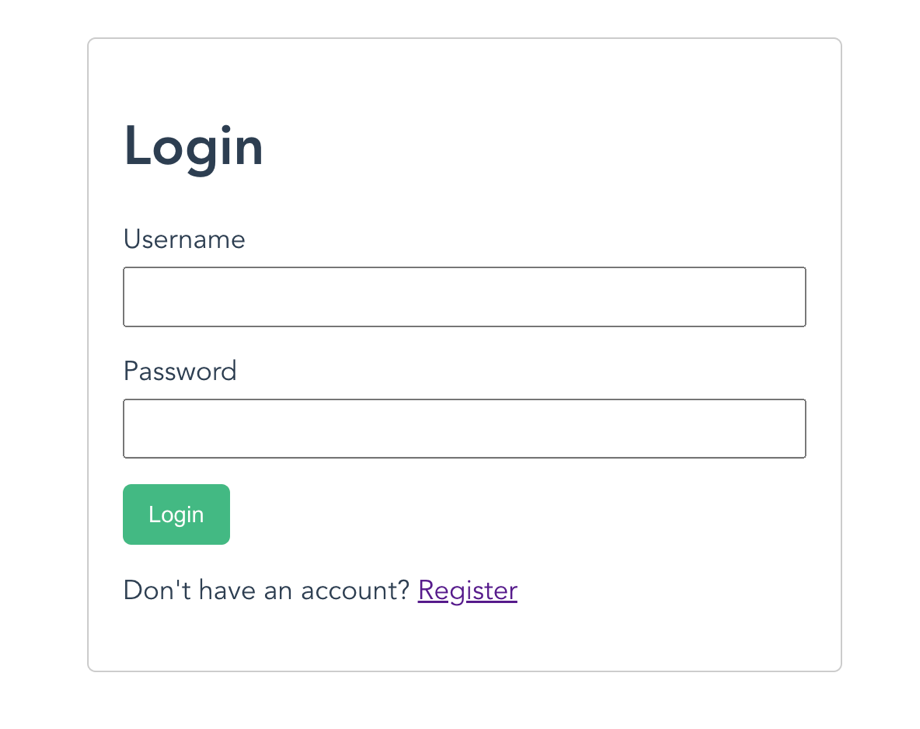


Рисунок 20 – Страница для входа

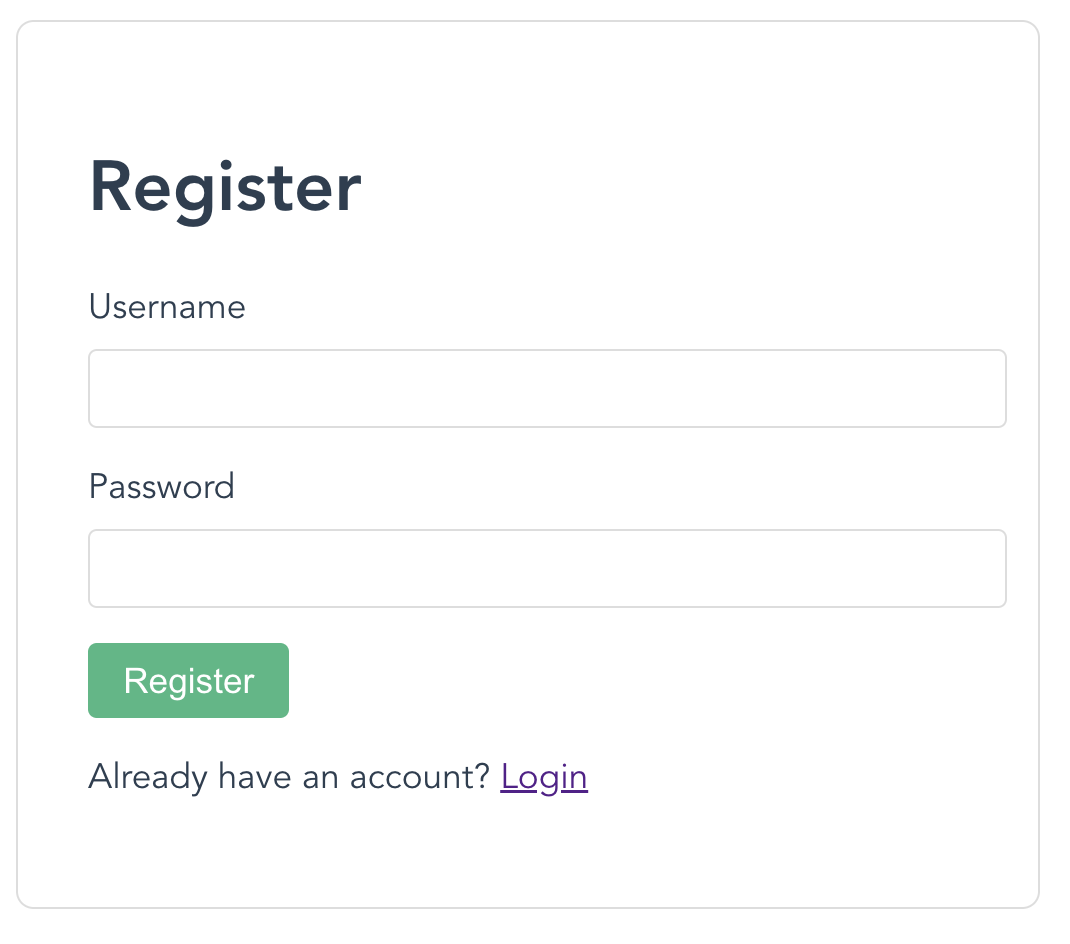


Рисунок 21 – Страница для регистрации

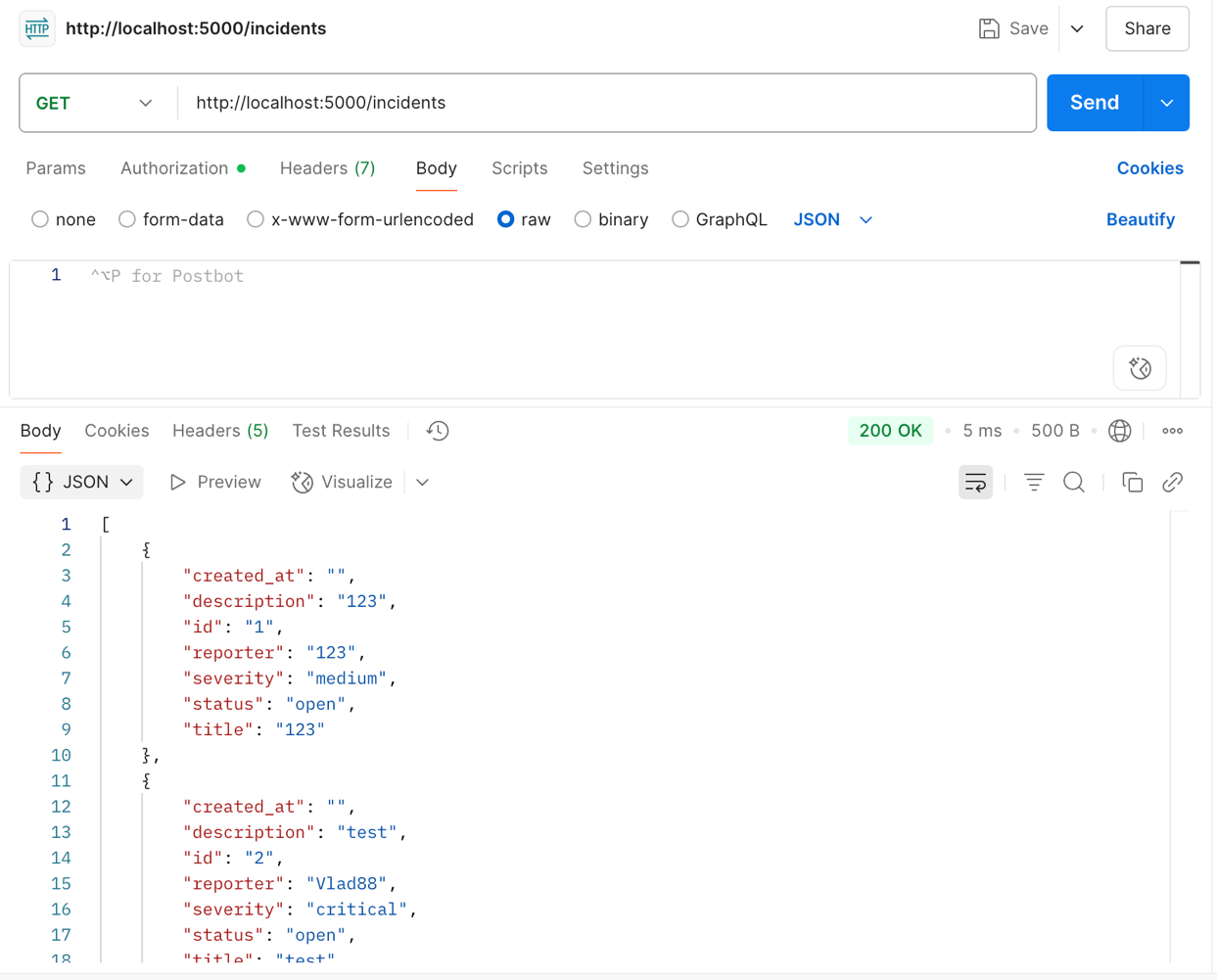


Рисунок 22 – Эндпоинт для получения всех инцидентов

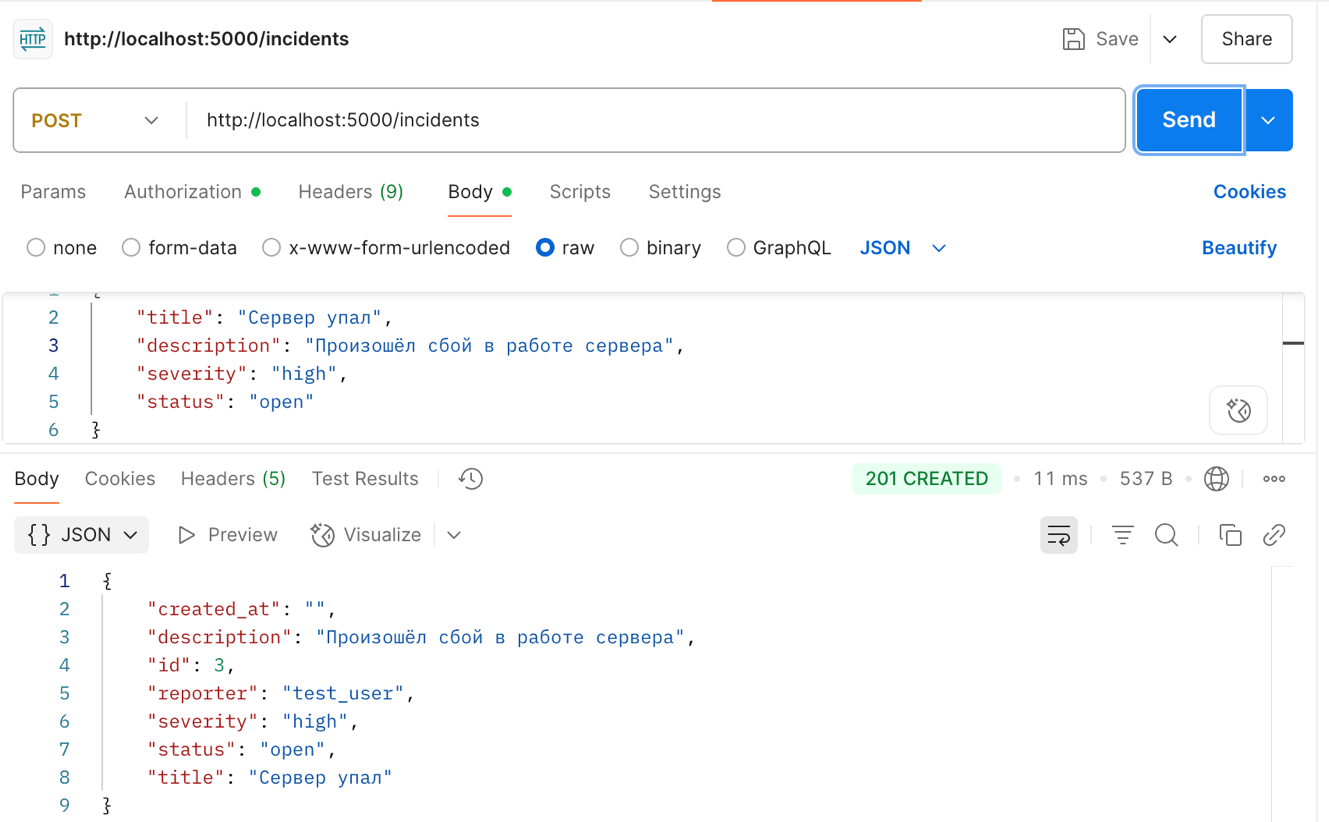


Рисунок 23 – Эндпоинт для добавления инцидентов

ВЫВОД

В рамках проекта разработано веб-приложение для фиксации инцидентов нарушения кибербезопасности на предприятии. Клиентская часть реализована на Vue.js, что обеспечивает динамическое отображение данных без перезагрузки страницы. Серверная часть создана на Python с обработкой ошибок и возвратом соответствующих HTTP-статусов для обеспечения стабильной работы системы.

Приложение предоставляет REST API с полным набором CRUD-операций, позволяя управлять записями об инцидентах. Маршрутизация во Vue.js обеспечивает динамическое обновление контента. Обмен данными между клиентом и сервером осуществляется в формате JSON, что обеспечивает структурированность передачи информации.

Функционал приложения включает регистрацию новых инцидентов, их редактирование, просмотр и удаление. Решение позволяет систематизировать учет нарушений кибербезопасности, что способствует своевременному выявлению угроз и повышению уровня защиты корпоративных данных.