Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЁТ

к лабораторной работе №6

на тему

**РЕАЛИЗОВАТЬ ЗАЩИТУ ОТ АТАКИ МЕТОДОМ ВНЕДРЕНИЯ   
SQL-КОДА**

Выполнил: студент гр.253504 Давыдовский Д.В.

Проверил: ассистент кафедры информатики Герчик А.В.

Минск 2025

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc188986632)

[2 Ход работы 4](#_Toc188986633)

[Заключение 6](#_Toc188986634)

# 1 ЦЕЛЬ РАБОТЫ

Цель работы заключается в разработке программных средств для защиты от атак методом внедрения SQL-кода. В ходе исследования будут рассмотрены механизмы возникновения данной уязвимости, методы её эксплуатации злоумышленниками, а также существующие способы защиты и предотвращения подобных атак.

На первом этапе будет проведен анализ принципов работы SQL-запросов, механизмов взаимодействия с базами данных и причин, по которым возможны атаки внедрения. Рассматриваемые угрозы будут включать различные типы инъекций, такие как классические SQL-инъекции и их вариации. Также уделим внимание примерам реальных атак и их последствиям.

Далее будет разработана архитектура защитной системы, включающая современные методы предотвращения атак внедрения SQL-кода. Важными аспектами станут использование параметризованных запросов.

На следующем этапе будет реализовано программное решение, направленное на выявление и предотвращение атак методом внедрения SQL-кода. Оно должно обеспечивать безопасное взаимодействие с базами данных, предотвращать неправомерный доступ и исключать возможность выполнения вредоносных команд.

Ожидаемыми результатами работы являются разработка эффективных методов защиты от атак внедрения SQL-кода и создание надежного программного инструмента. Это обеспечит защиту от одной из наиболее распространенных категорий уязвимостей, повышая общую устойчивость программного обеспечения к потенциальным угрозам.

# 2 ХОД РАБОТЫ

В ходе работы был разработан два программных продукта на основе *Node.js* с использованием *Express* и *PostgreSQL*. Один из них защищён от *SQL*-инъекций, а второй — не защищённый. Данная уязвимость может быть проиллюстрирована с помощью примеров кода, который позволяет пользователям регистрироваться и авторизовываться в системе.

Исходный код беззащитной программы представлен нижу. В этом коде используются небезопасные *SQL*-запросы, где пользовательские данные вставляются напрямую в строку запроса. Например, в функции *unsafe-login* пользователь вводит логин и пароль, которые затем подставляются в *SQL*-запрос. Это создаёт возможность для злоумышленника выполнить *SQL*-инъекцию, что может привести к компрометации данных.

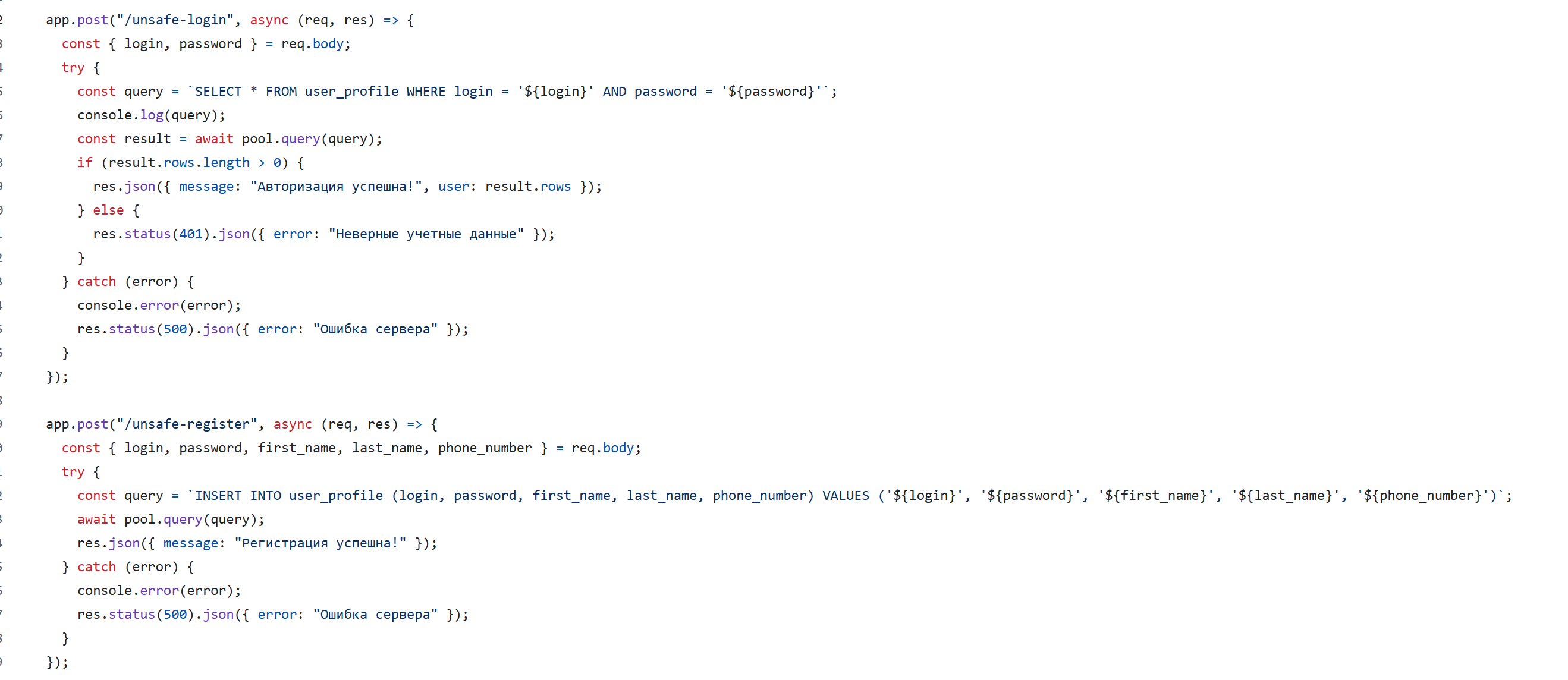


Рисунок 1 – Исходный код беззащитного участка

На практике, если пользователь введёт вредоносные данные, такие как  *admin 'OR '1'='1*, это может вызвать несанкционированный доступ к системе. Результат выполнения этой программы демонстрирует уязвимость, позволяя злоумышленнику получить доступ к аккаунту администратора. Результат работы небезопасного участка кода представлен на рисунке 2.

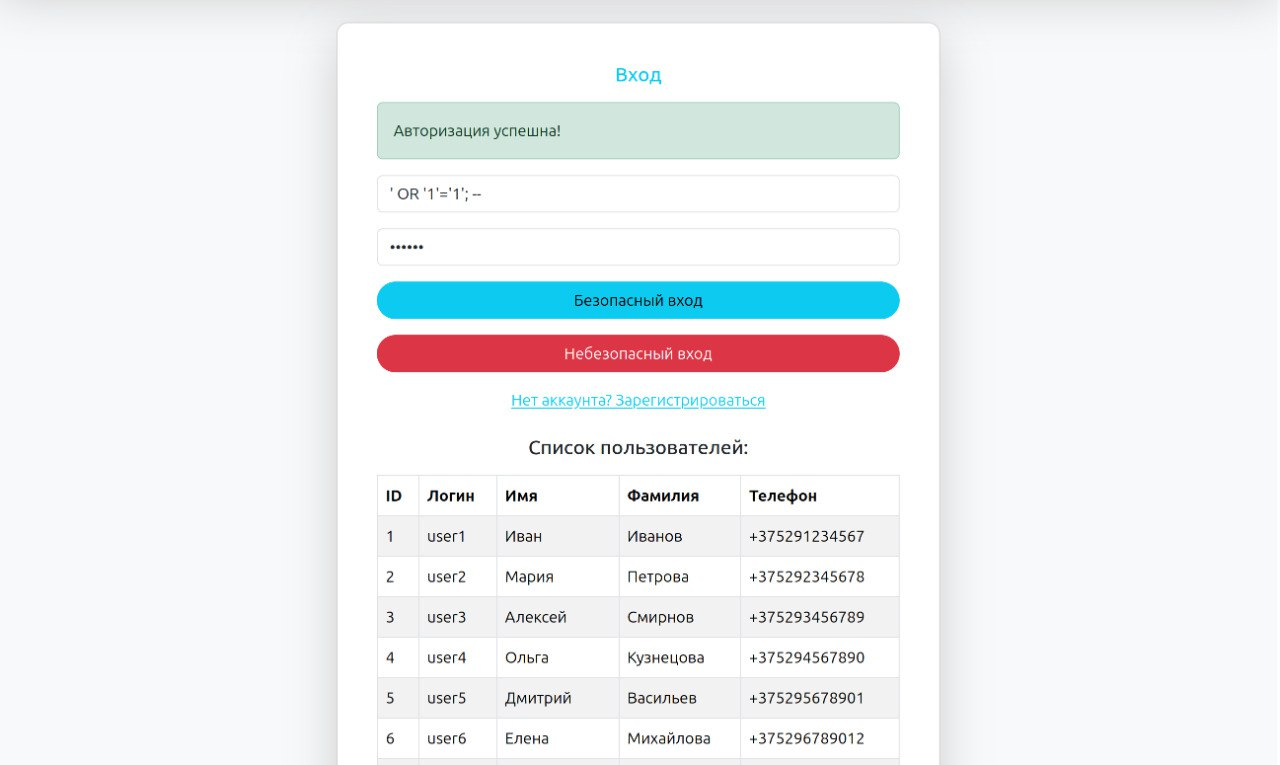


Рисунок 2 – Результат работы небезопасного участка

Теперь рассмотрим защищённый код, который использует параметризованные запросы. В этом варианте функции *login* и *register* применяют параметры $1, $2 и т.д., которые безопасно экранируют вводимые данные. Это предотвращает возможность *SQL*-инъекций, так как пользовательские данные не могут быть интерпретированы как часть *SQL*-запроса. На рисунке 3 представлен исходный код с защитой от *SQL*-атаки.



Рисунок 3 – Исходный код участка с защитой

Результат работы защищённой программы показывает, что даже при попытке внедрения вредоносного кода доступ к данным остаётся закрытым. Это достигается благодаря использованию безопасных методов работы с базой данных, что гарантирует защиту от атак. На рисунке 4 представлен результат работы

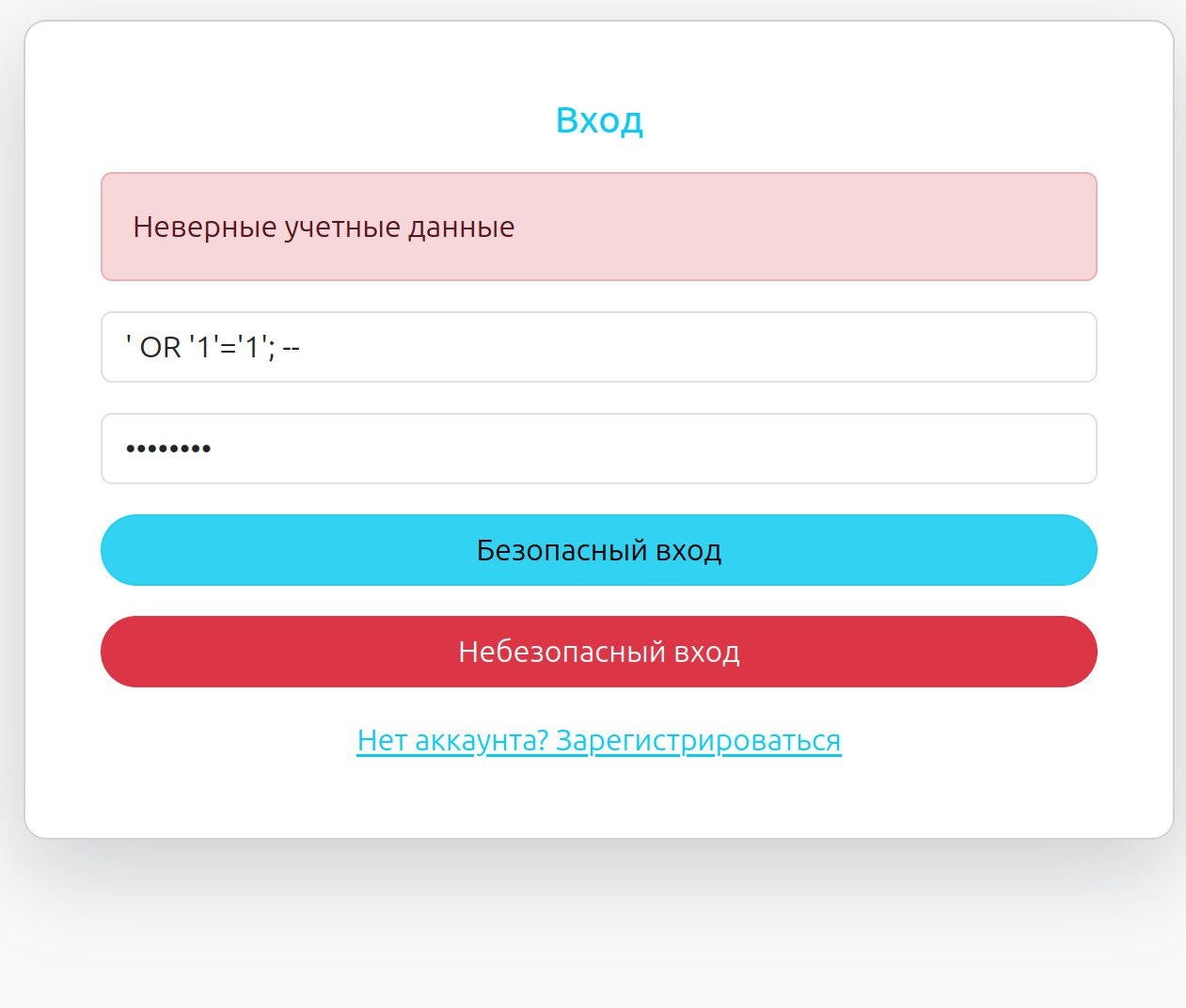


Рисунок 4 – Результат работы участка с защитой

Таким образом, реализация защиты от *SQL*-инъекций в разработанном программном продукте демонстрирует важность безопасного программирования и подтверждает, что применение параметризованных запросов значительно улучшает безопасность приложений, защищая их от одной из наиболее распространённых уязвимостей.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы была разработана и протестирована система защиты от атак типа SQL-INJECTION. Для примера был разработан сервер с 2 парами функций, одни безопасные другие нет. Для пользователя был разработан веб-сайт, для удобного пользовательского ввода.