Дисциплина: Разработка безопасных веб-приложений

Преподаватель: Мирвода Сергей Геннадьевич

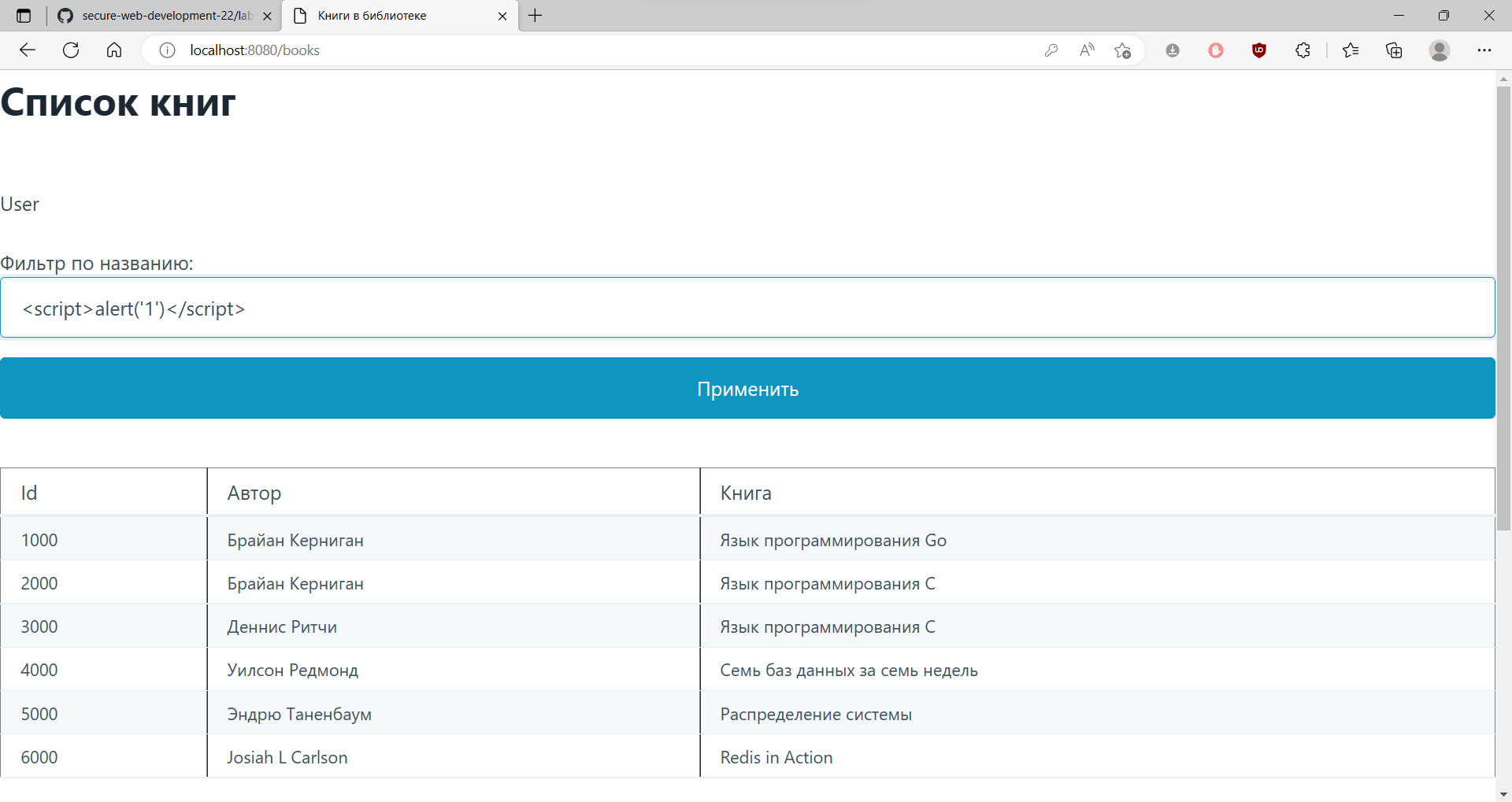
Группа: РИ-581220

Студенты: Фирсов Виктор, Ибрагимов Игорь

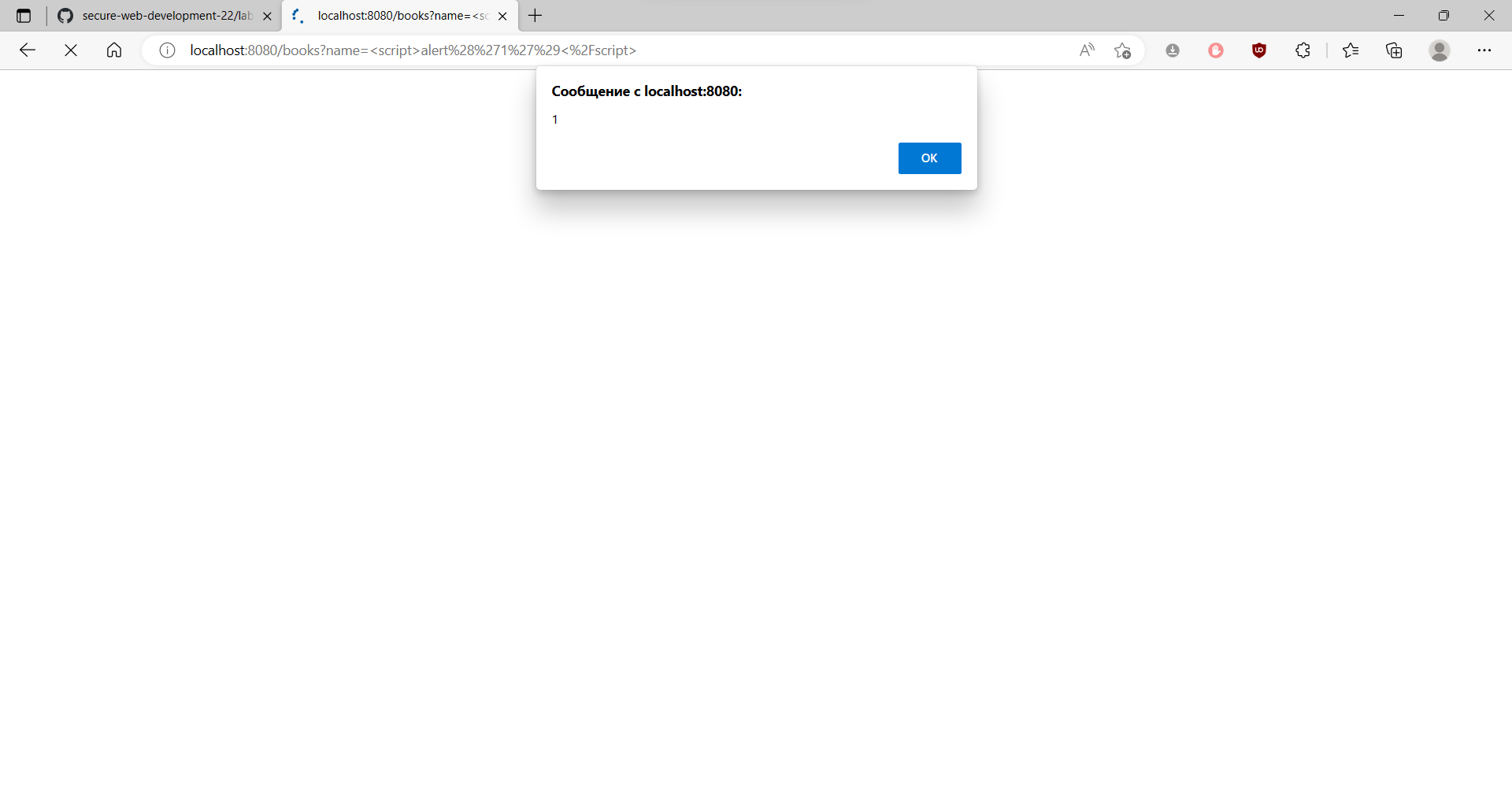
Лабораторная работа №4. XSS

**Reflected XSS:**

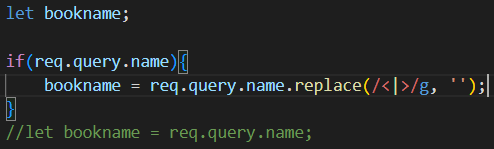
Нам нужно поместить скрипт в поле для поиска и нажать «Применить»:



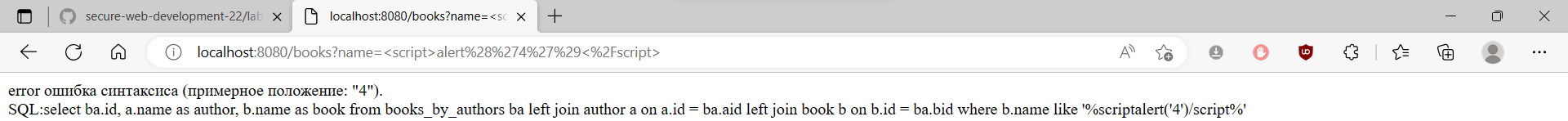
Скрипт исполнится сайтом один раз:



Для устранения данной уязвимости можно воспользоваться командой replace, которая будет менять символы «<» и «>» на ничего. Для этого нужно немного поменять код сервера:



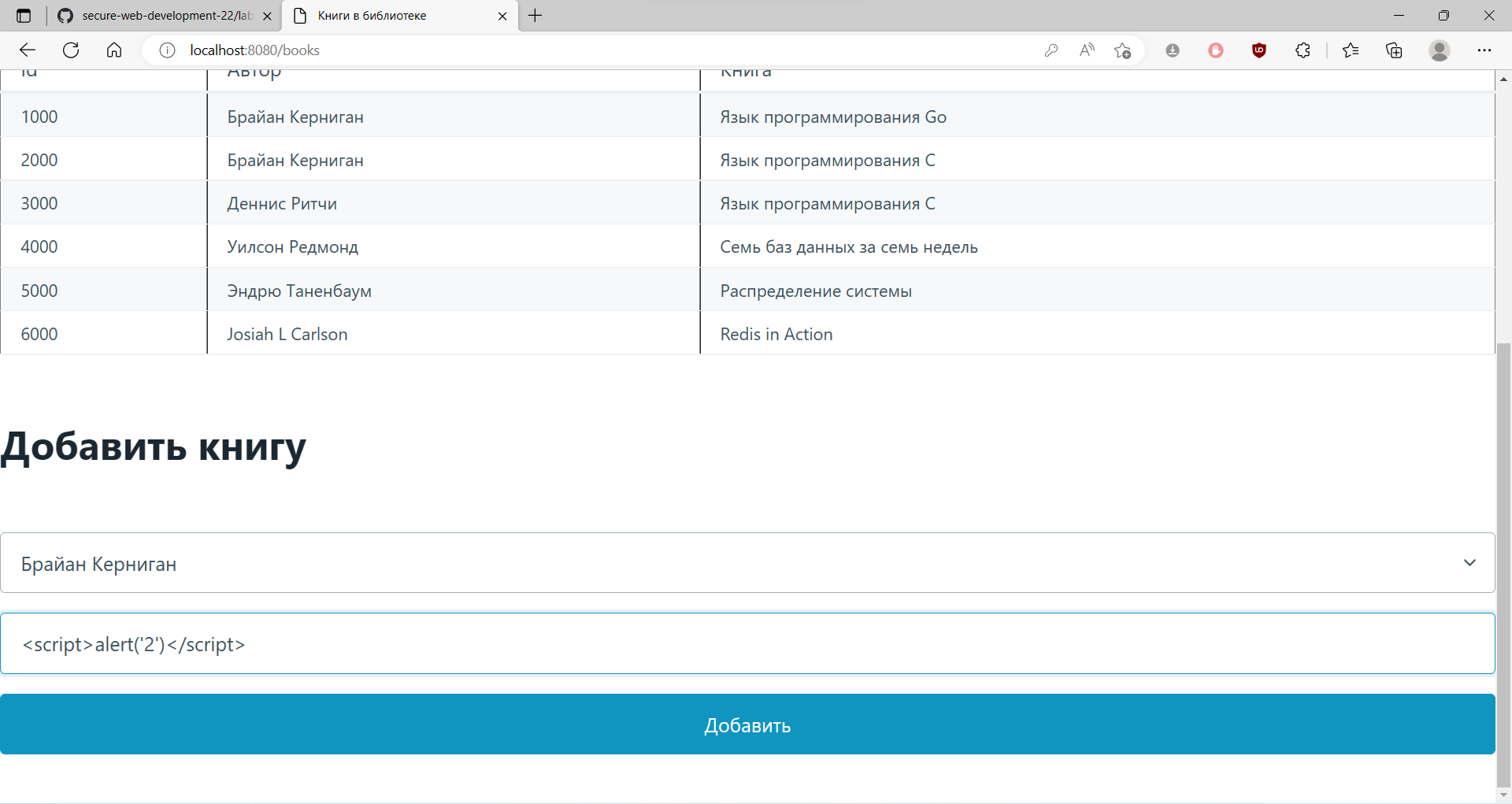
Теперь при попытке вставить скрипт в поле поиска, будет следующий результат:



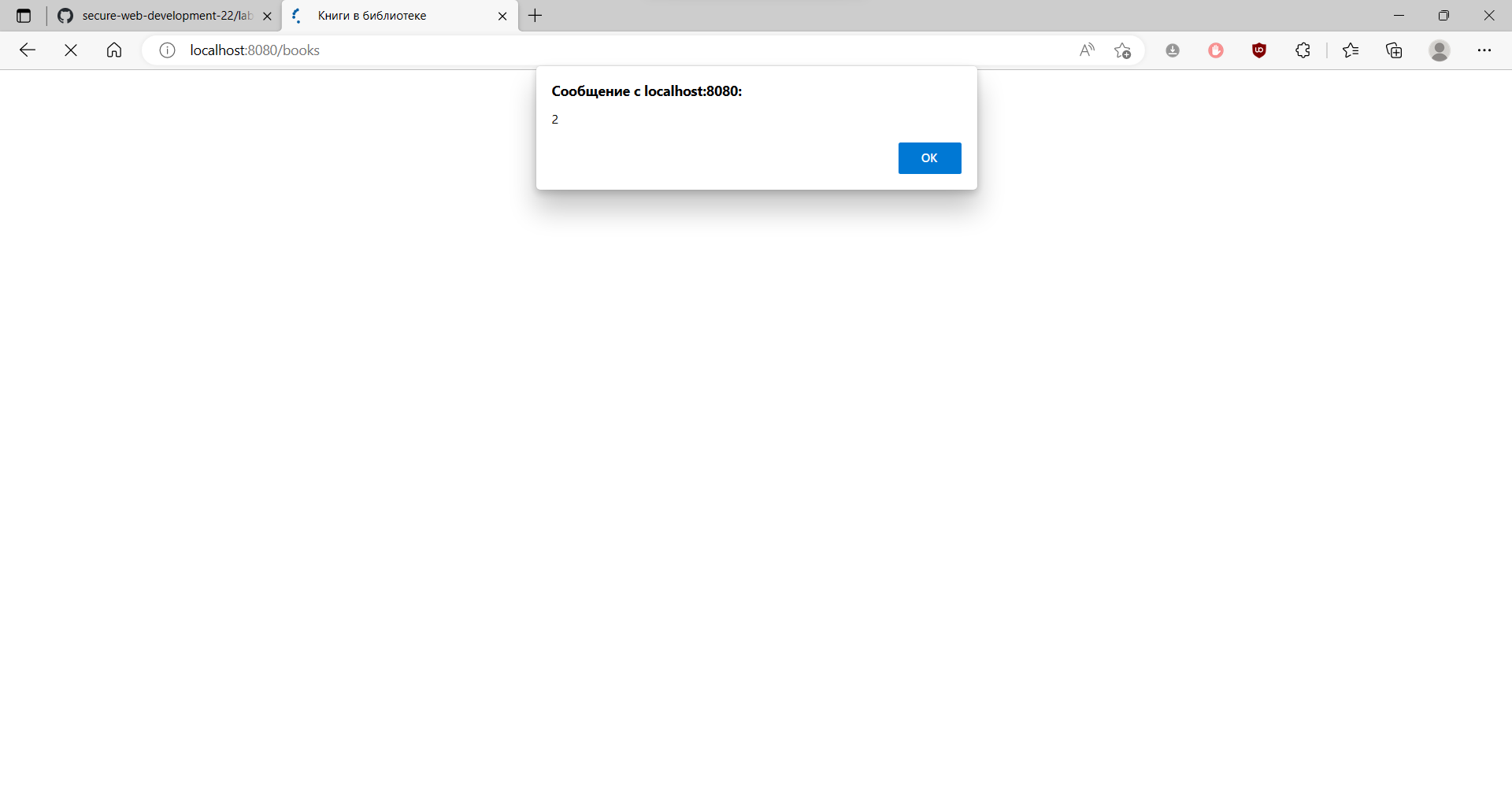
Как можно заметить, из пользовательского ввода удалились треугольный скобки и сервер, не выполнив скрипт, просто выдал ошибку sql-запроса.

**Stored XSS:**

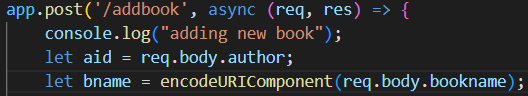
Нам нужно поместить скрипт в поле для названия книги и нажать «Добавить»:



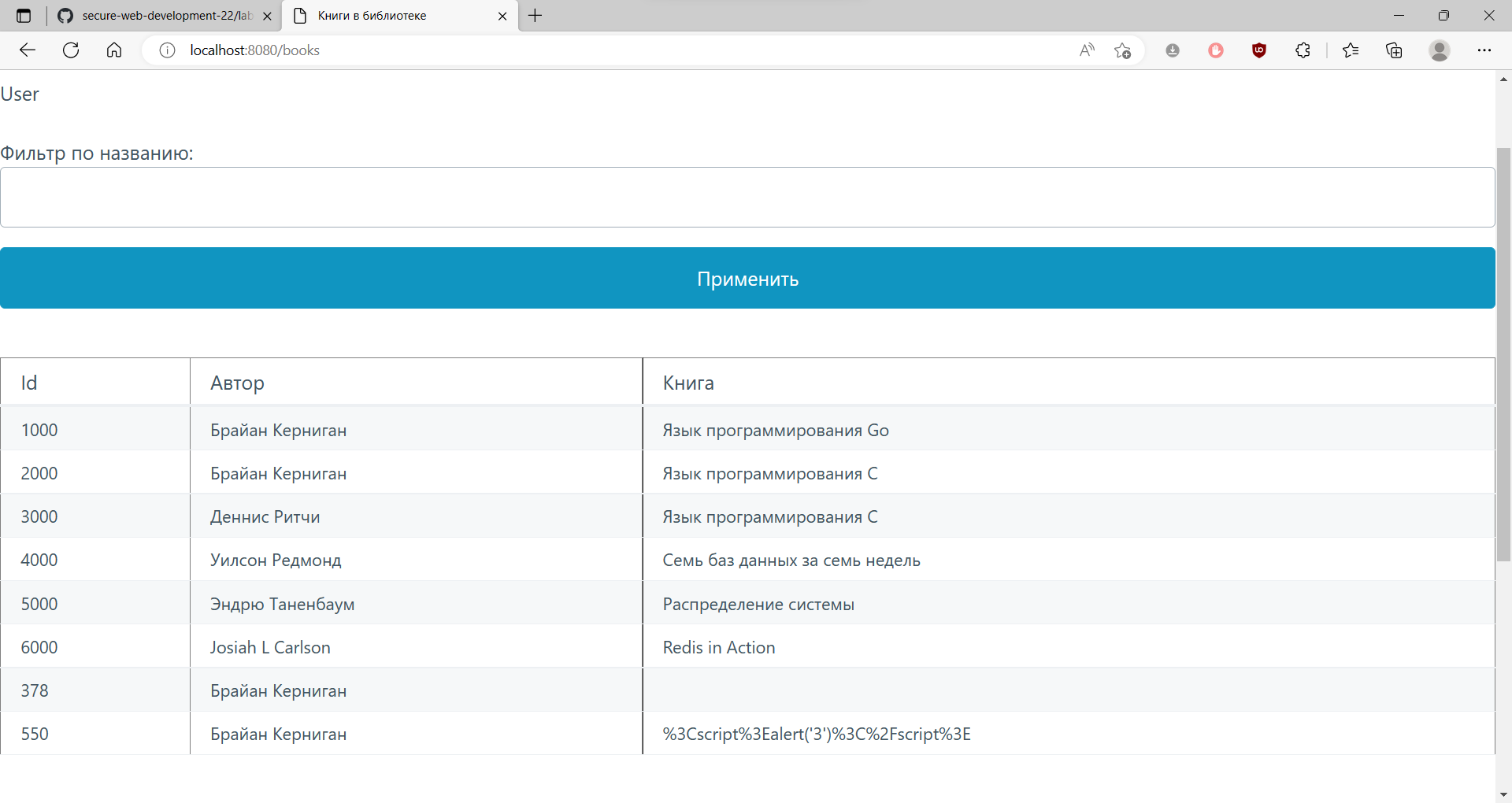
Скрипт будет исполняться при каждом открытии этой страницы:



Для устранения этой уязвимости можно воспользоваться функцией encodeURIComponent(), которая заменяет служебные символы мнемониками. У этой функции есть минус: она кодирует таким образом и русские символы, поэтому уместно воспользоваться функцией replace, как в предыдущем пункте, однако если пользовательский ввод не предполагает русских символов, ей все же можно воспользоваться, поэтому мы ее все-таки рассмотрим. Эту функцию необходимо применить для пользовательского ввода req.body.bookname, считывающегося переменной bname:

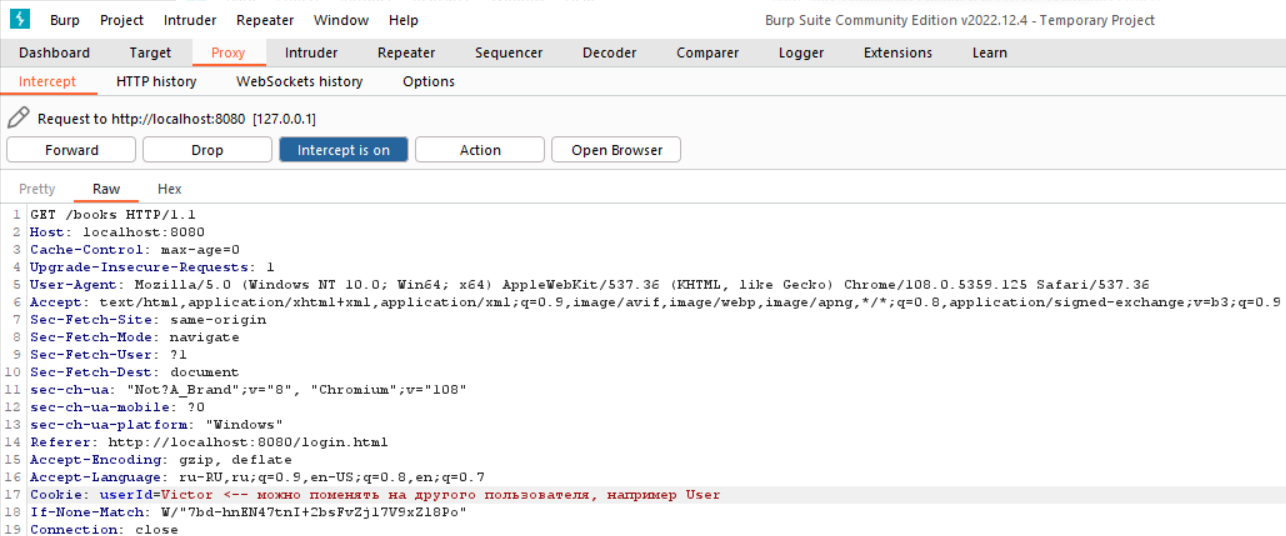


Тогда скрипт не будет исполняться, так как служебные символы пользовательского ввода в базе данных будут выглядеть следующим образом:

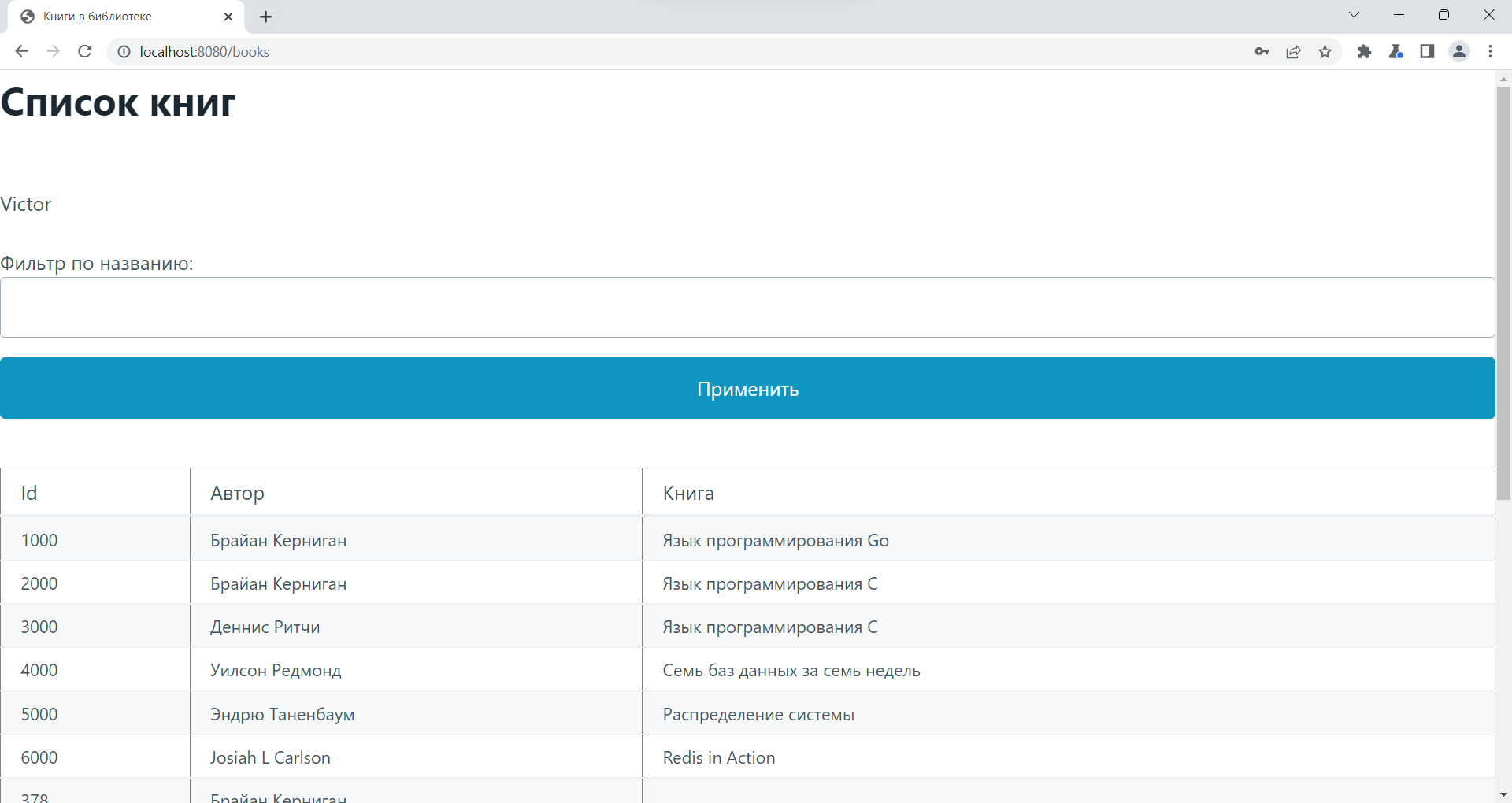


**Cookie Injection:**

Потенциальная уязвимость находится при отправке запросов на сервер. Там можно поменять значение cookie на другого пользователя:

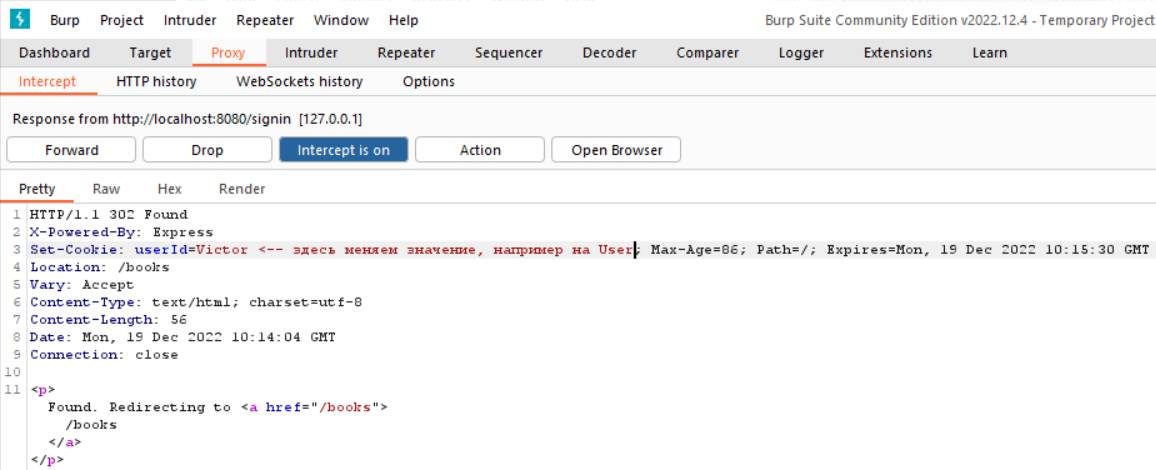


Однако это лишь потенциальная уязвимость, так как на ответ сервера она не влияет и активным пользователем остался Victor:



**Session hijacking:**

Эта уязвимость находится в ответах сервера. Здесь в ответе можно поменять cookie на другого пользователя, и все последующие запросы будут происходить от его имени, то есть как раз таки произойдет перехват сессии:



Теперь запрос отправляется от имени другого пользователя:



Сервер принимает этот запрос, и авторизация происходит от имени другого пользователя, то есть сессия успешно перехвачена:

