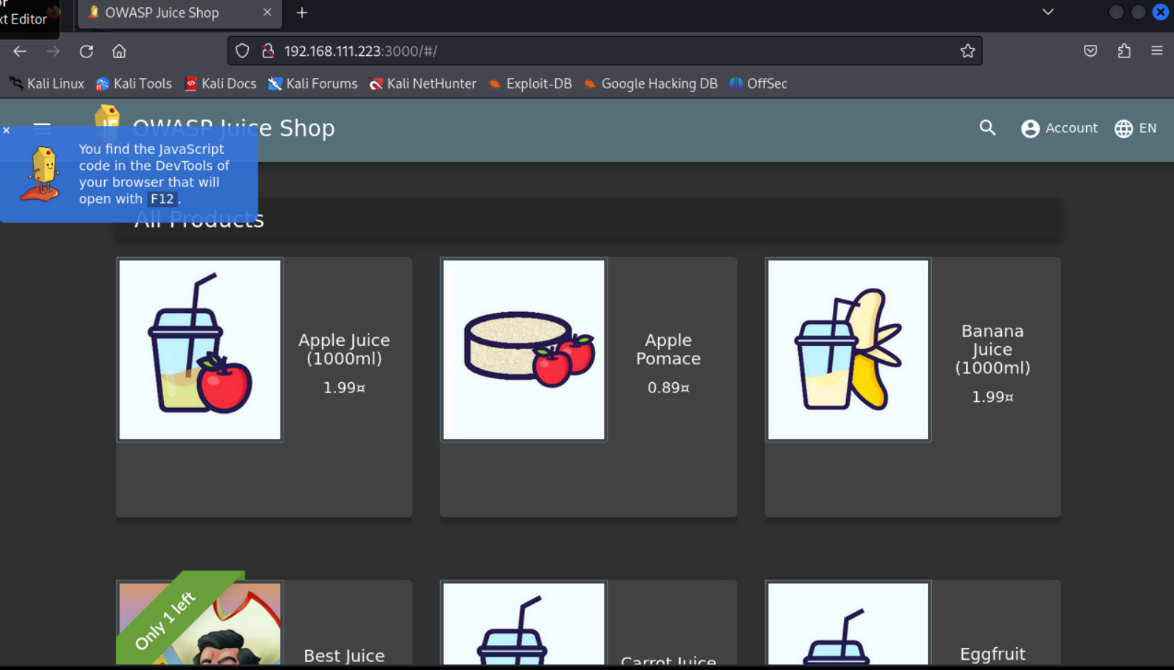
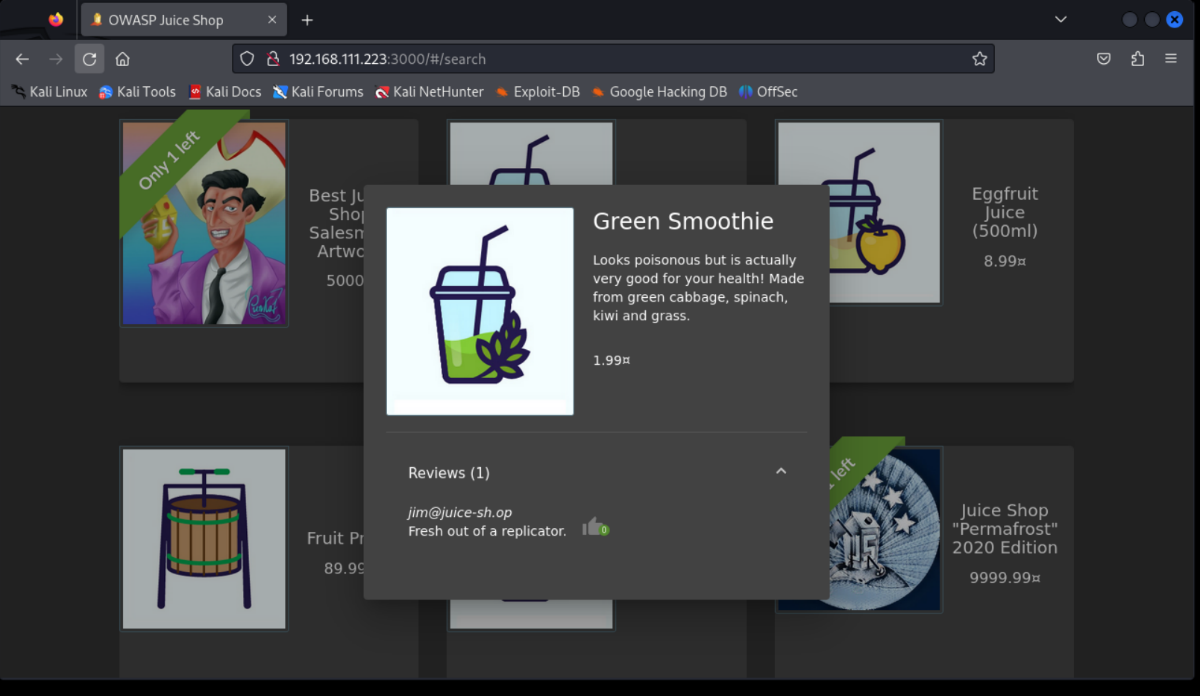
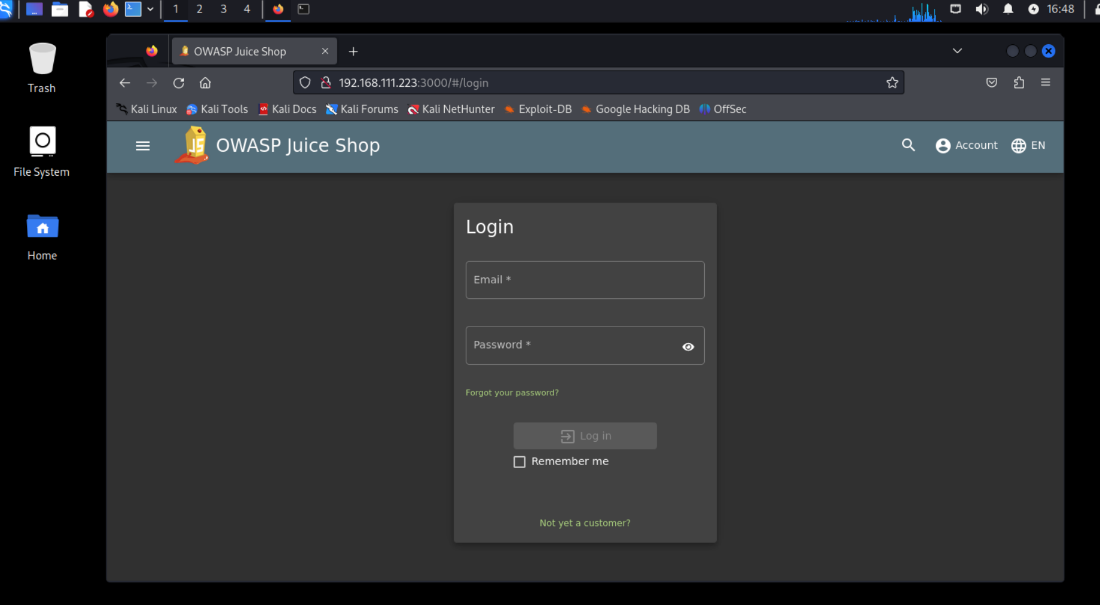
**Scoping out the site**



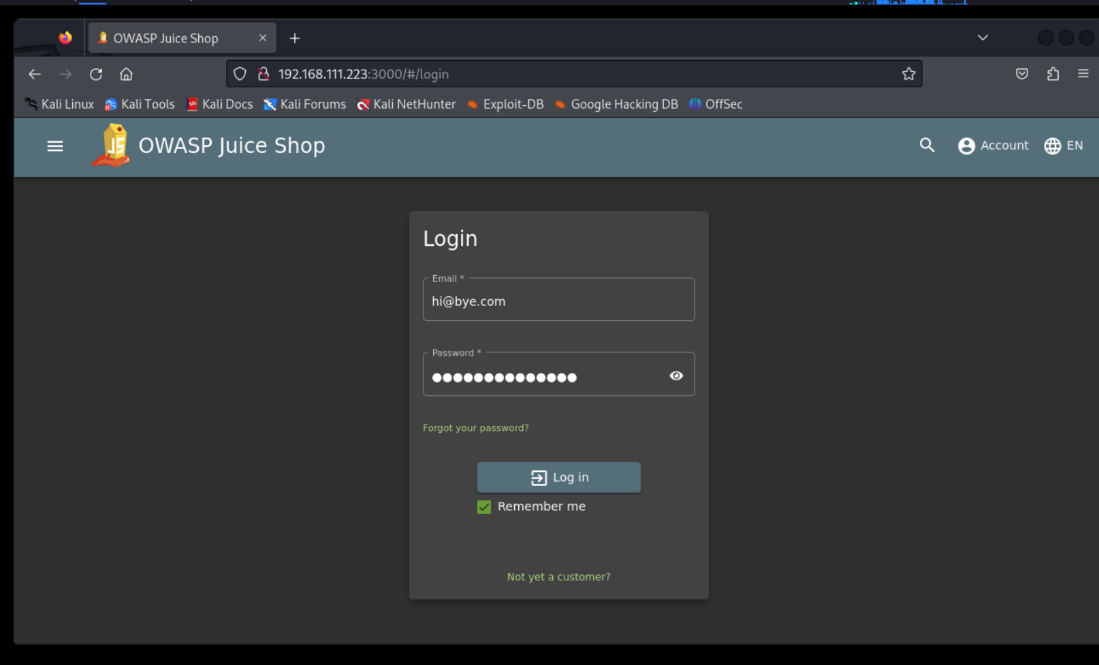
This here is the front page of juice box used to navigate the website where you can scroll through and see all the products that they sell on the website



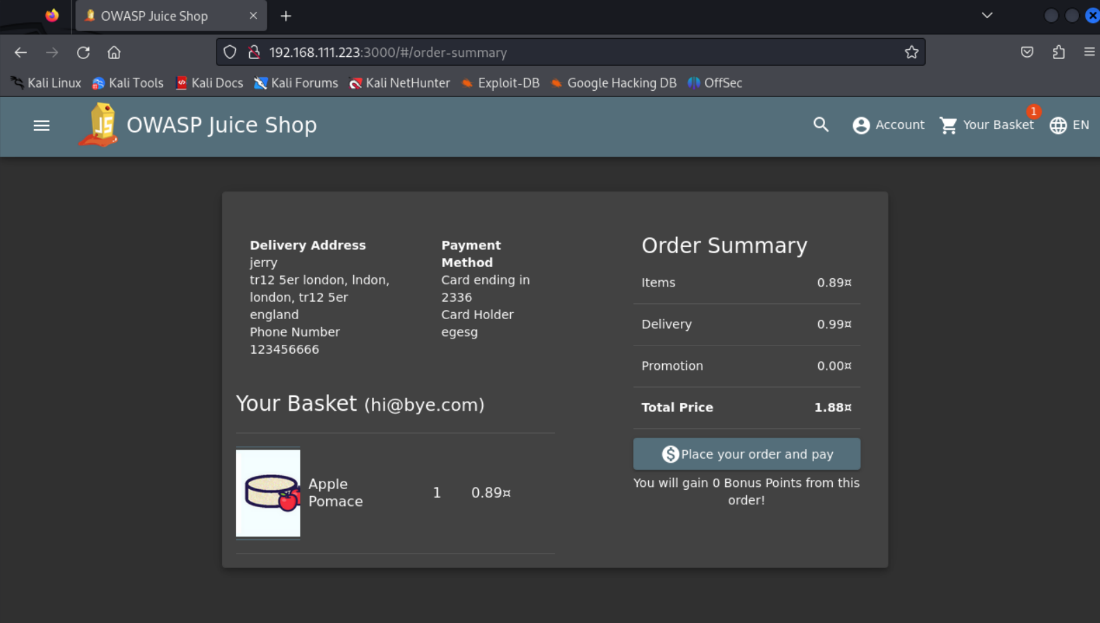
All the product icons on the website are clickable and when you click on them, they come up with a brief description of the product along with a review of it



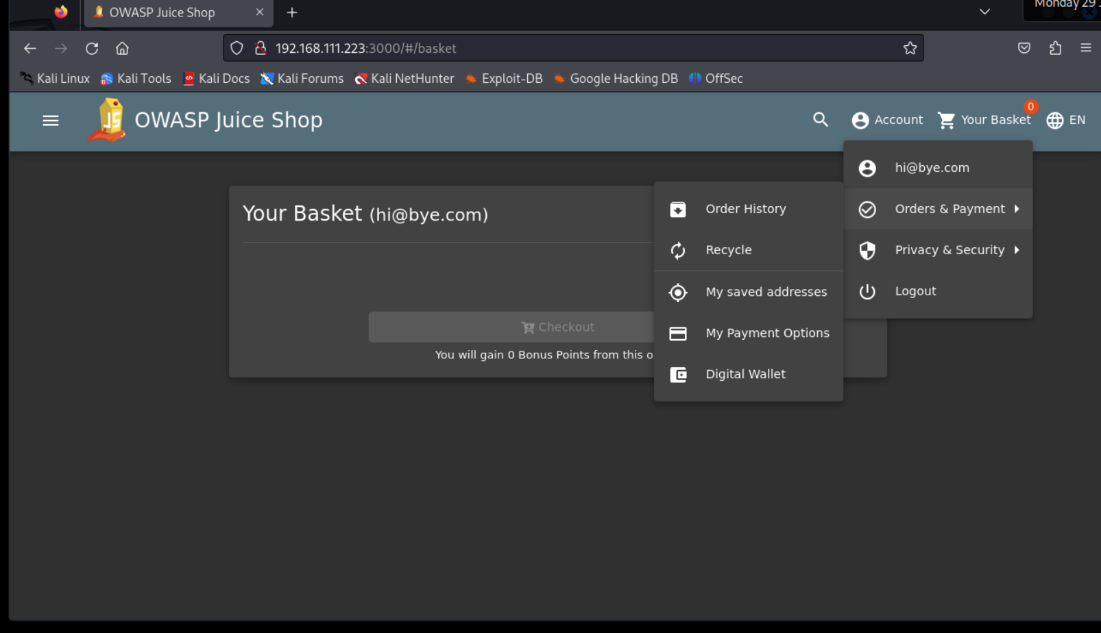
There's also a log in page for you to log in with an account



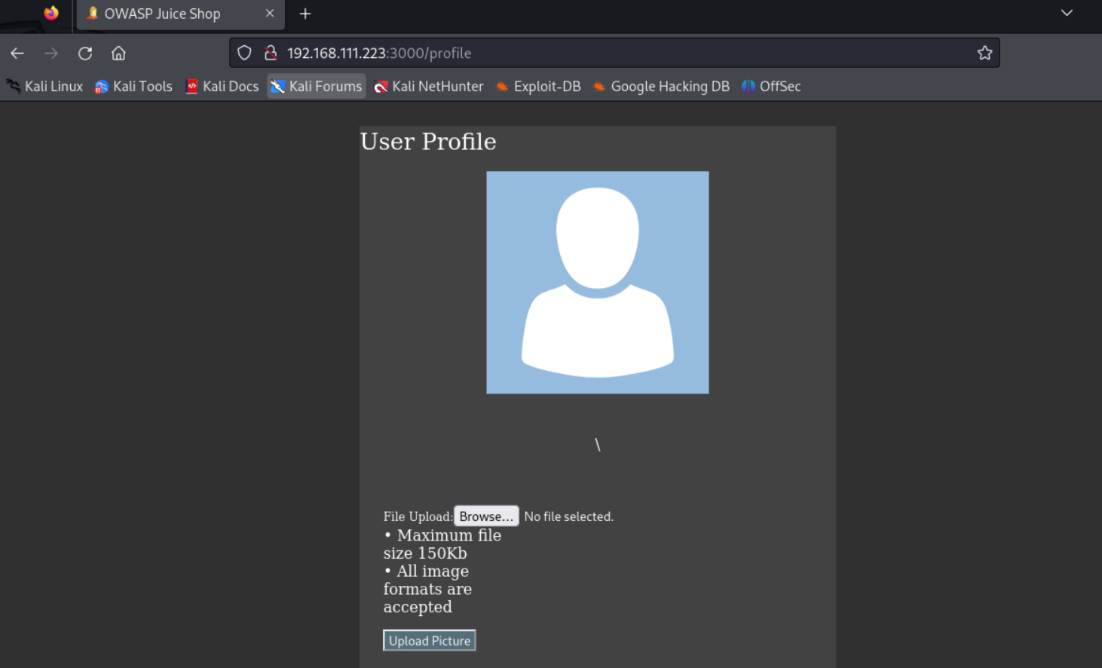
After you have created an account you can log into that account and add items to your basket



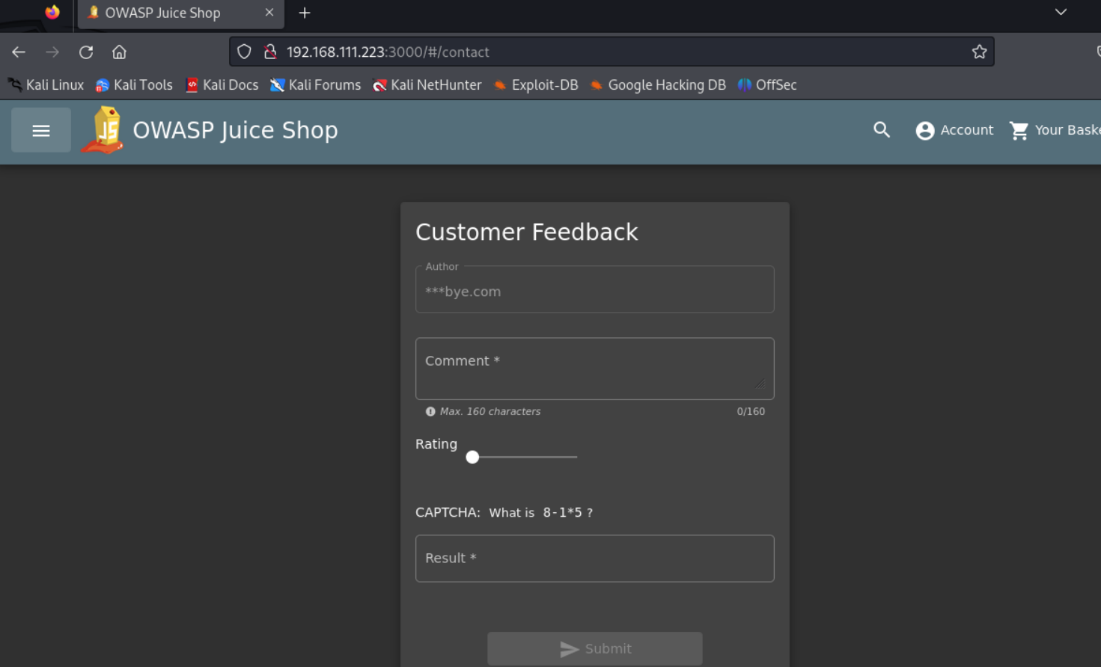
After putting in details when filling out the checkout requirements you can place an order for a particular item



There is also vast drop down menus for the user to navigate through

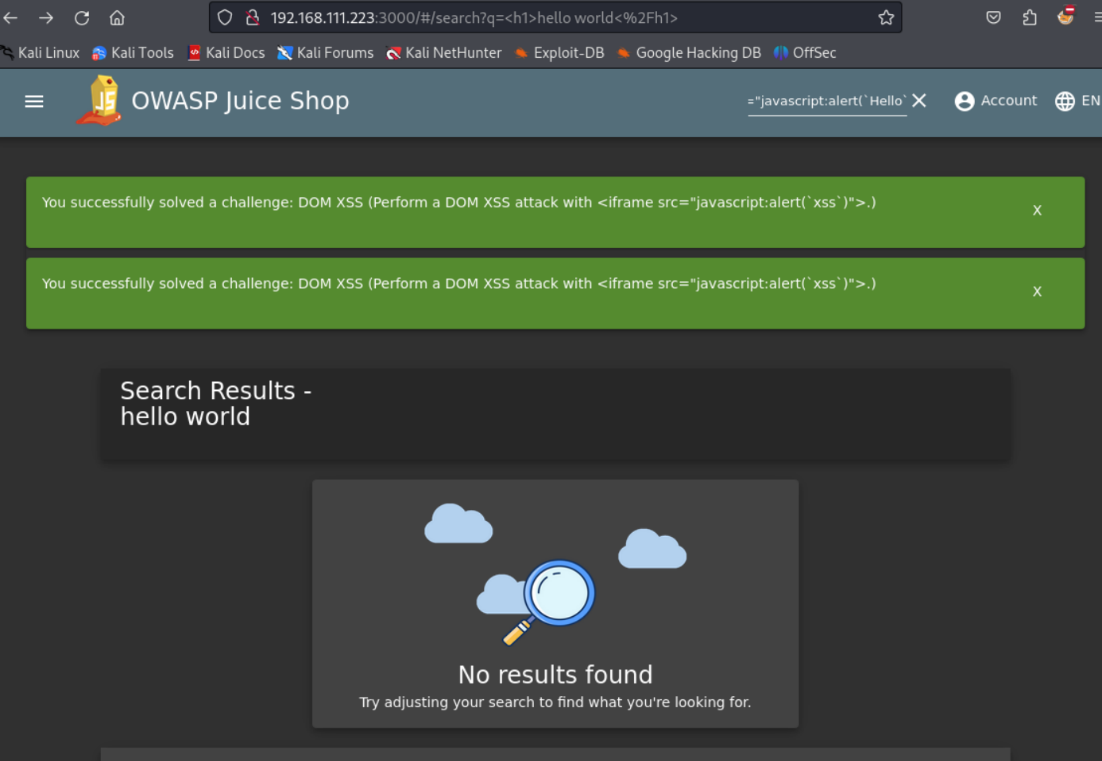


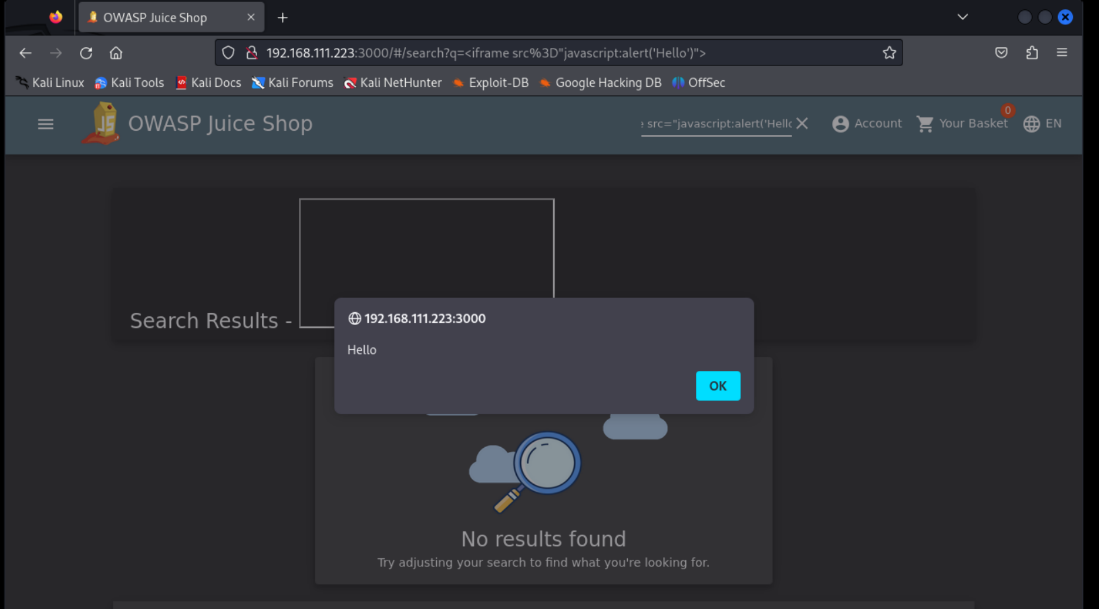
Along with a user profile that you can customise



There is also a customer feedback tab along with a complaints tab

**Requirement 3: XSS**





Firstly, I inputted <h1>hello world</h1> into the search bar to embed it into the page. I then injected the java script of <iframe src="javascript:alert(`xss`)"> into the search bar and got the pop up saying hello back at me once the website has executed the java I typed out which creates a pop up box showing a message, you can run other more malicious java scripts instead of just a pop up saying hello which makes it vital websites defend against XSS attacks.

**Requirement 4: SQL injection**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

For the SQL injection I identified the login page as a potential point of injection as it used SQL queries to authenticate. I used this weakness to gain access to the admin’s account by inputting the email: “ ‘or 1=1 – “ and a password: “ ‘ “. I used foxy proxy to route traffic to burp suite as well for my tools to gain access. Once I pressed login with these as the details it logged me into the admins account without having to have the correct email or password.

**Requirement 5: Broken access control**

**A screenshot of a computer

Description automatically generated**

After having logged into the admin account using the SQL injection from the previous requirement I went to the URL displayed at the top of the page, I had previously tried this but got an error message saying I couldn’t access it because I wasn’t an admin. Once I was on the admin page I discovered that I had the ability to delete customers reviews and to see all the registered users along with their email addresses.

**Requirement 6: Authentication Bypass vulnerability**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A screenshot of a login box

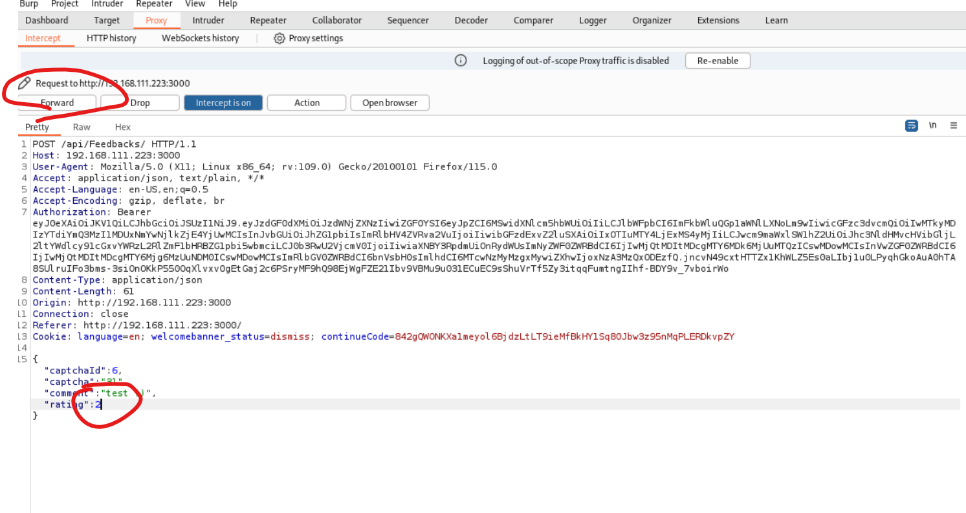
Description automatically generated**

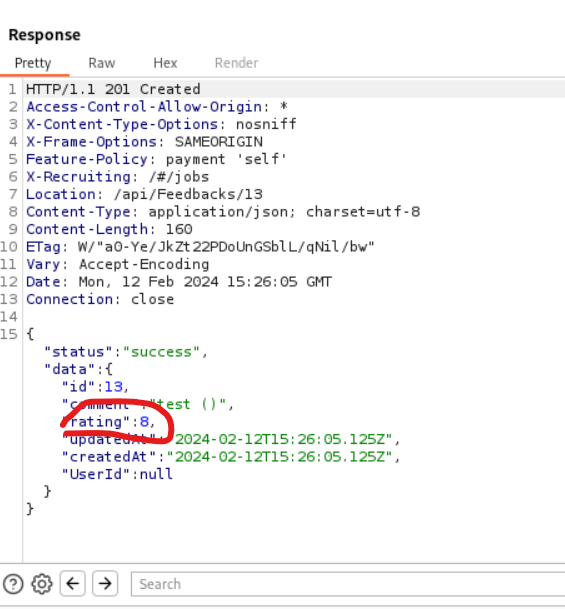
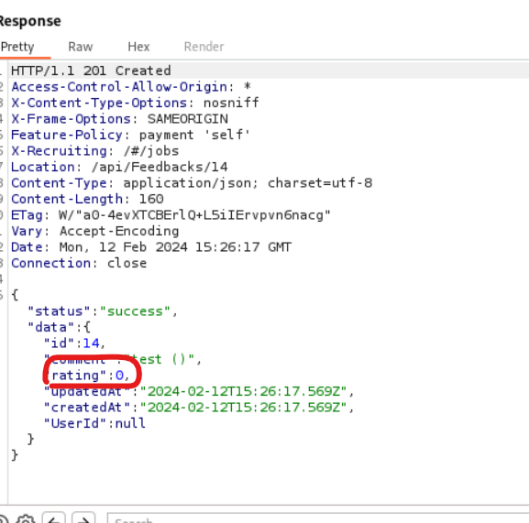
A screenshot of a computer

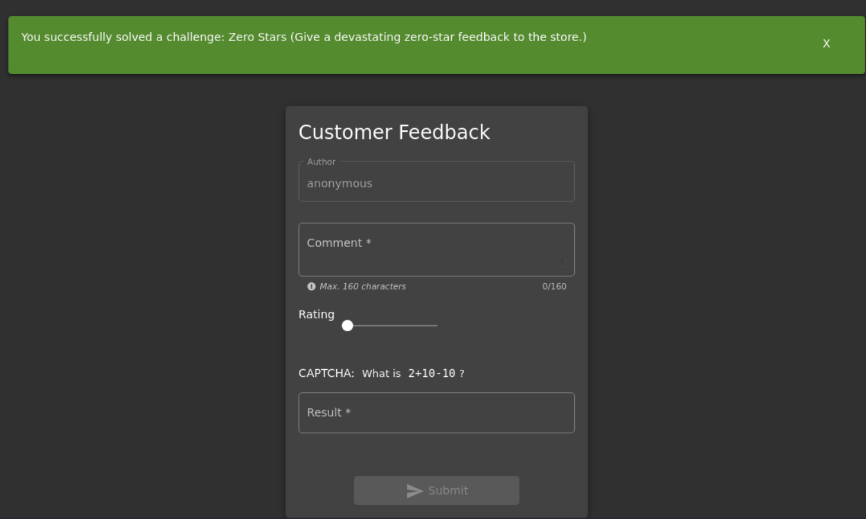
Description automatically generated

For this attack I used burp suite and foxy proxy again. In the screenshots you can see how I set up my burp suite to intercept me putting in a random password and then use that to send an intruder attack. It did this by taking a file I made of common passwords and entering all those passwords along with the already known admin email until it got a password that was a success, as shown with ‘admin123’ with the status code of 200 meaning success. Once I knew what the password was I could then return to my juice shop and enter the password and press log in and gain access to the account.

**Requirement 7: Exploit an Improper Input Validation vulnerability**

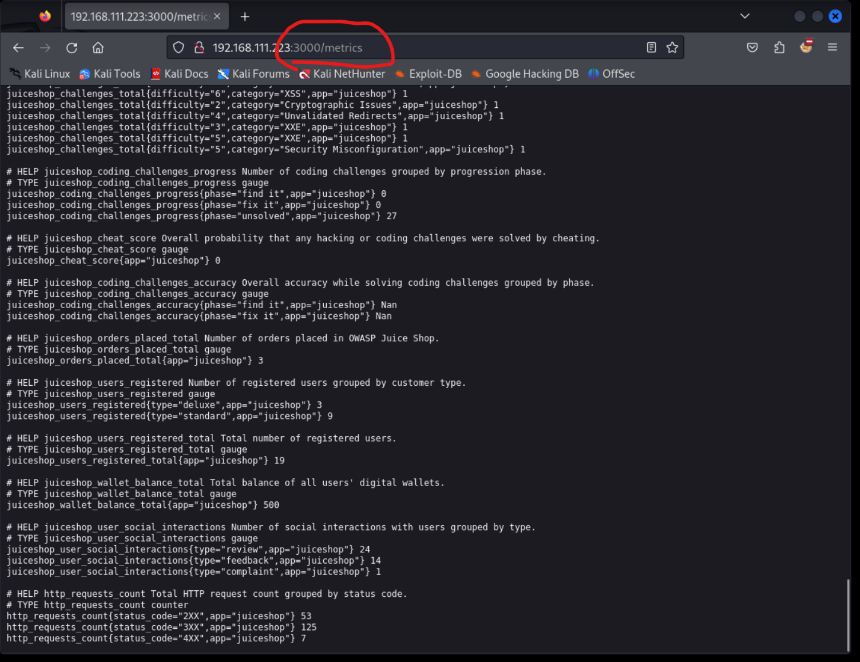






For this exploit I used burp suite again and used the same configuration of targeting the juice shop URL then intercepting my original customer feedback with a 2-star rating. I then edited the star rating to a 0 (or any other invalid number for example 8) and then pressed forward to have the site run it and then I got the challenge successfully completed message at the top of my screen showing I was able to enter an invalid input.

**Requirement 8: Exploit a Sensitive Data Exposure vulnerability**



For this exploit I found online that you can add /metrics on the end of the URL and it directs you to a new page displaying lots of sensitive data that shouldn't be publicly accessible such as how many users they have on the web page.