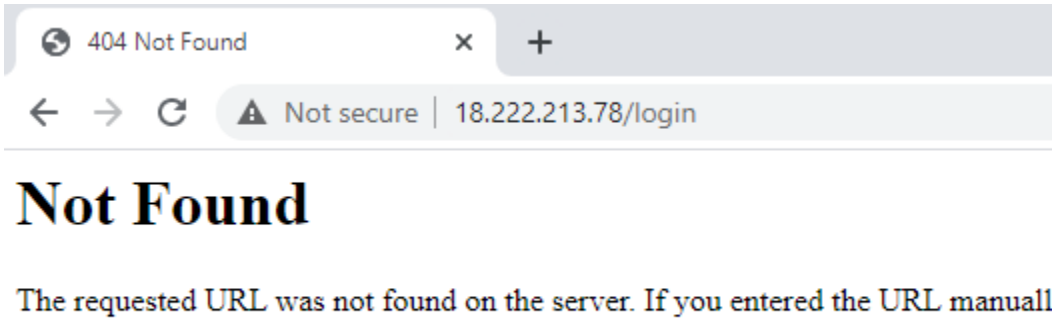
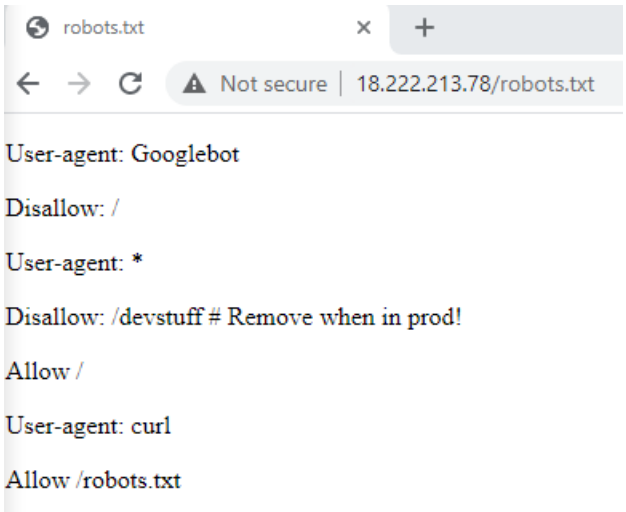
Web Bug Bounty
Part 1

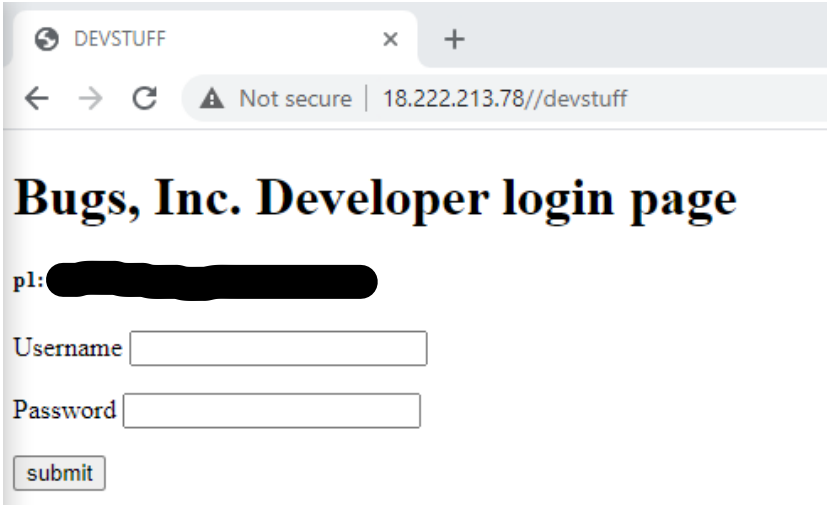
For the first part, I tried to append /login the the URL



Then I appended /robots.txt



Following http://18.222.213.78//devstuff



The first flag is [redacted]

Part 2

For the second part, i first tried default credentials
admin
admin

Next i tried to do an SQL injection
admin') 1==1 --
Password

Not secure | 18.222.213.78/devstuff

Bad login info!

Refresh your page to try again

Upon inspecting the page, i found a link to a .js file

LOGIN FAILED | view-source:18.222.213.78//dev | 18.222.213.78/devstuff/clientsi

Not secure | view-source:18.222.213.78//devstuff

Line wrap

```
<!DOCTYPE html>
<html>
<head>
    <title>DEVSTUFF</title>
</head>
<body>
    <!-- For security reasons, we are no longer using this clientside validation! -->
    <script src="/devstuff/clientside_login.js"></script>
    <!-- Logins are handled on the server side! -->

    <h1>Bugs, Inc. Developer login page</h1>
    <h5>p1: [redacted]</h5>

    <form action="/devstuff" method = "POST">
        <p>Username <input type = "text" name = "username" /></p>
        <p>Password <input type = "text" name = "password" /></p>
        <p><input type = "submit" value = "submit" /></p>
    </form>

</body>
</html>
```

Following it, we get

Not secure | 18.222.213.78/devstuff/clientside_login.js

// CLIENT SIDE JS LOGIN VALIDATION // ^^ REMOVE IN PROD!! INSECURE!! // CORPORATE SAID NO PLAINTEXT PASSWORD -> MOVED TO 64-BIT ENCRYPTION // The atob() and btoa() methods allow authors to transform content to and from the encoding. function CheckUsername(username_input) { username_encoded = btoa(username_input) // using JS enCRYPTion! if(username_encoded == "Y3JhaWcxCg==") { // if encoded username_input == encoded good username alert('Username good!') return true; } else { alert('Bad username!') return false; } } function CheckUsername(password_input) { password_encoded = btoa(password_input) // using JS enCRYPTion! if(password_encoded == "Y29va2llbTBuc3Qzcgo=") { // if encoded password_input == encoded good password alert('Password good!') return true; } else { alert('Bad password!') return false; } }

Using https://www.10bestdesign.com/dirtymarkup/js/ to clean this up, the code looks like

```javascript
// CLIENT SIDE JS LOGIN VALIDATION
// ^^ REMOVE IN PROD!! INSECURE!!
// CORPORATE SAID NO PLAINTEXT PASSWORD -> MOVED TO 64-BIT ENCRYPTION
// The atob() and btoa() methods allow authors to transform content to and from the encoding
function CheckUsername(username_input) {
    username_encoded = btoa(username_input)
    // using JS enCRYPTion!
    if (username_encoded == "Y3JhaWcxCg==") { // if encoded username_input == encoded good username alert('Username good!') return true; }
        else {
            alert('Bad username!') return false;
        }
    }

    function CheckUsername(password_input) {
        password_encoded = btoa(password_input) // using JS enCRYPTion!
        if (password_encoded == "Y29va2llbTBuc3Qzcgo=") { // if encoded password_input == encoded good password alert('Password good!')
            return true;
        } else {
            alert('Bad password!')
            return false;
        }
    }
```

While the password credentials are encoded, we can see from the comments

// CORPORATE SAID NO PLAINTEXT PASSWORD -> MOVED TO 64-BIT ENCRYPTION
// The atob() and btoa() methods allow authors to transform content to and from the encoding

That they use 64 bit encryption and that we can use atob() and btoa() to decode/encode
Using an online script,

index.html × styles.css × script.js ×

```javascript
import 'bootstrap@4.6.0'

console.log(atob("Y3JhaWcxCg=="));
console.log(atob("Y29va2llbTBuc3Qzcgo="));
```

CONSOLE ×

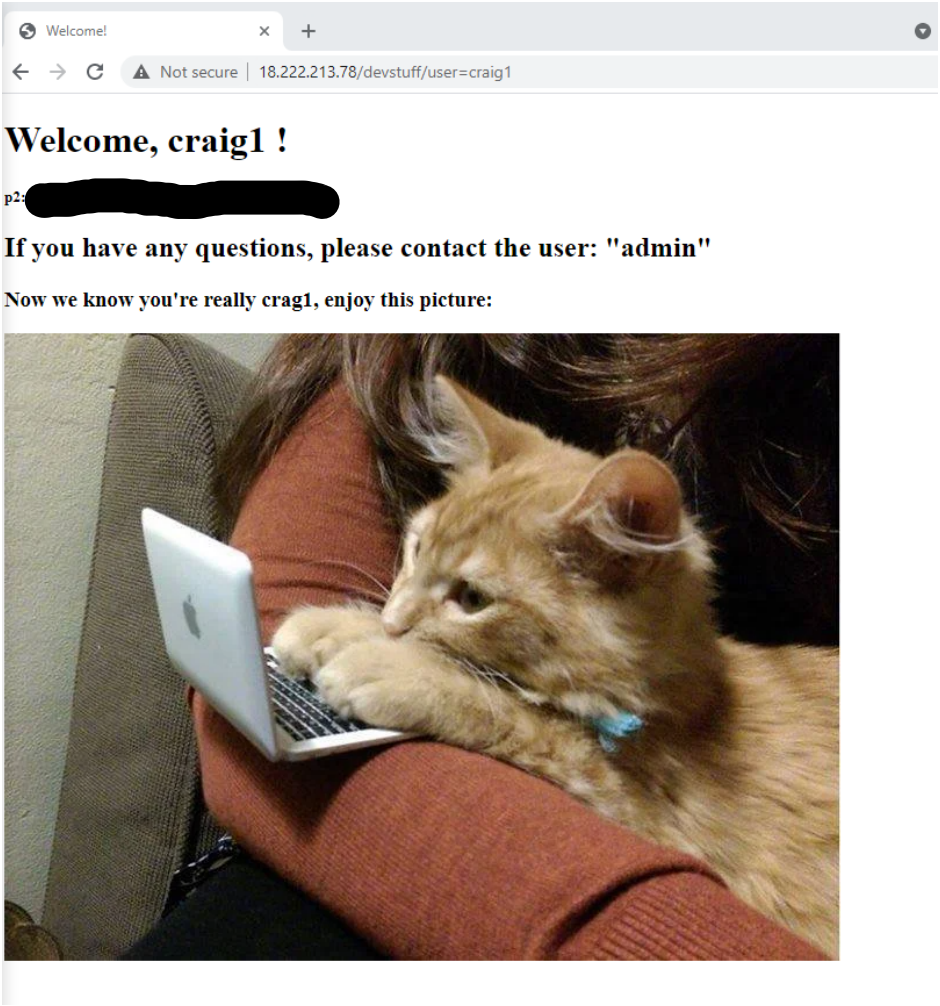
```
craig1
cookiem0nst3r
```

I found the credentials to be

`craig1`

`cookiem0nste3r`

This didn't work at first, so I got confused but then I realized I had to refresh the page and I got in.



The flag is [redacted]

Part 3

For part 3, to login as admin, I noticed the URL for craig1 was

http://18.222.213.78/devstuff/user=craig1

The `user=craig1` is interesting so i changed it to admin

http://18.222.213.78/devstuff/user=admin

Welcome!

Not secure | 18.222.213.78/devstuff/user=admin

Welcome, admin !

p3[redacted]

Great job! Here's another cat pic:

The flag is [redacted]