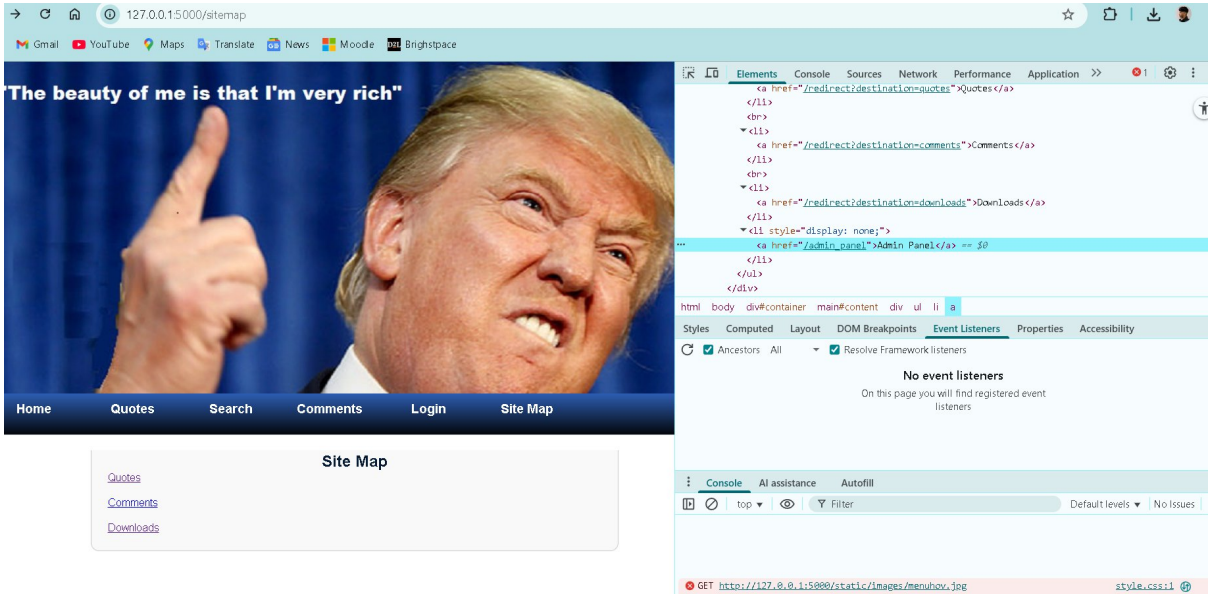
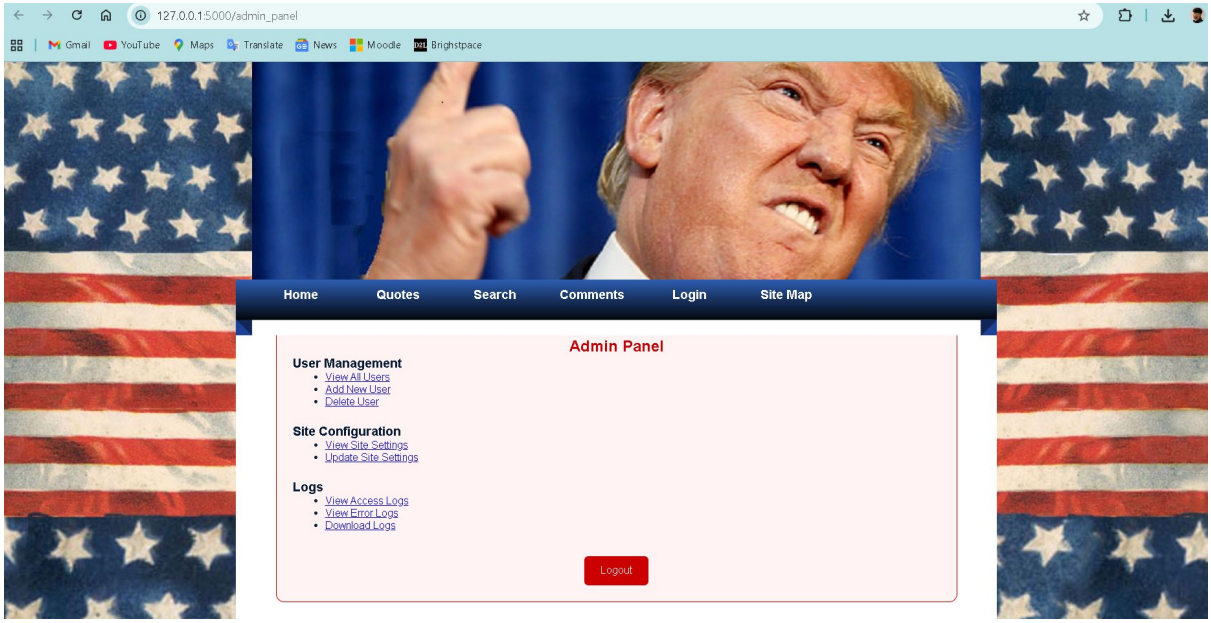
Access Control Vulnerability was found as within the sitemap in the page source there is a link to the admin panel



When clicked on, the admin panel is displayed and there is no proper authorisation or authentication checks



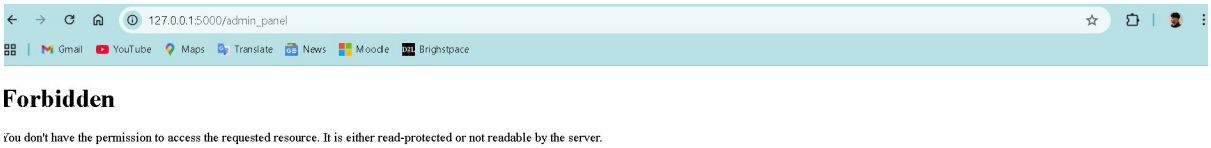
The backend has no authentication whatsoever, and who ever wants to access it is free to do so

```
50
51  @app.route('/admin_panel')
52  def admin_panel():
53      return render_template('admin_panel.html')
54
```

To fix this, the code now has a key for the administrator, so if anybody was now able to try and login wihout the code it will now not be possible.

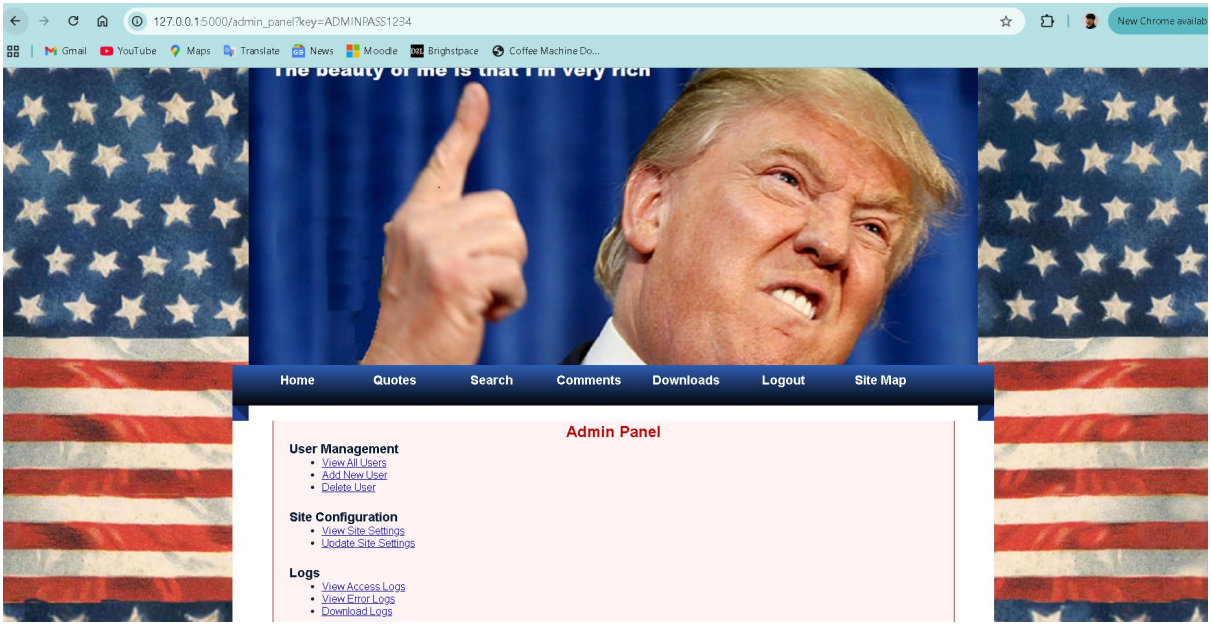
```python
@app.route('/admin_panel')
def admin_panel():
    key = request.args.get('key')
    if key != 'ADMINPASS1234':
        abort(403)  # Forbidden
    return render_template('admin_panel.html')
```

If we check now, the website does not allow access to the admin panel



+3

However if we try with the key in the address bar



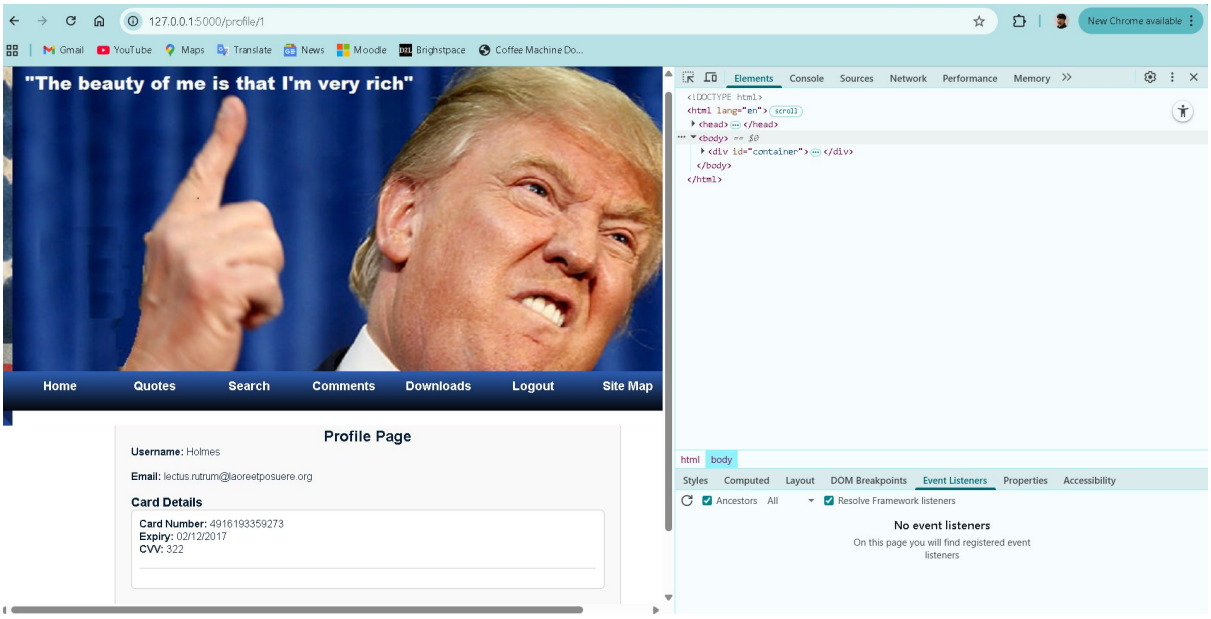
We now have access to the admin panel

Insecure Direct Object Reference vulnerability was found. To do this we first needed any login detail. since the website does not have a sign up option, ssing the sql file provided we got one login,

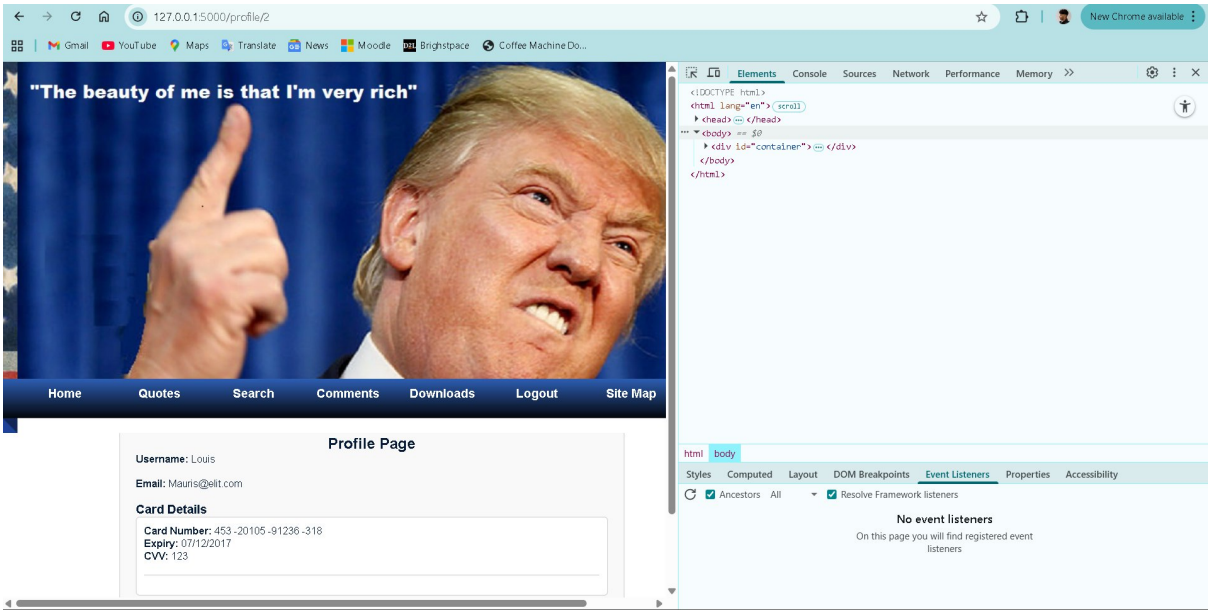
I decided to use username ” Holmes” and his password “MEC15DBF3XD”in this isntance

```
13    `user` TEXT DEFAULT NULL
14
15
16 •  SERT INTO `users` (`username`,`password`,`email`,`about`) VALUES ("Holmes","MEC15DBF3X
17 •  SERT INTO `users` (`username`,`password`,`email`,`about`) VALUES ("Jakeem","FXL38OBR0A
18 •  SERT INTO `users` (`username`,`password`,`email`,`about`) VALUES ("Stuart","AOO42CAE6S
19 •  SERT INTO `users` (`username`,`password`,`email`,`about`) VALUES ("Kamal","XUG95SXT6EY
20 •  SERT INTO `users` (`username`,`password`,`email`,`about`) VALUES ("Bruno","RKV77YJT3UO
```

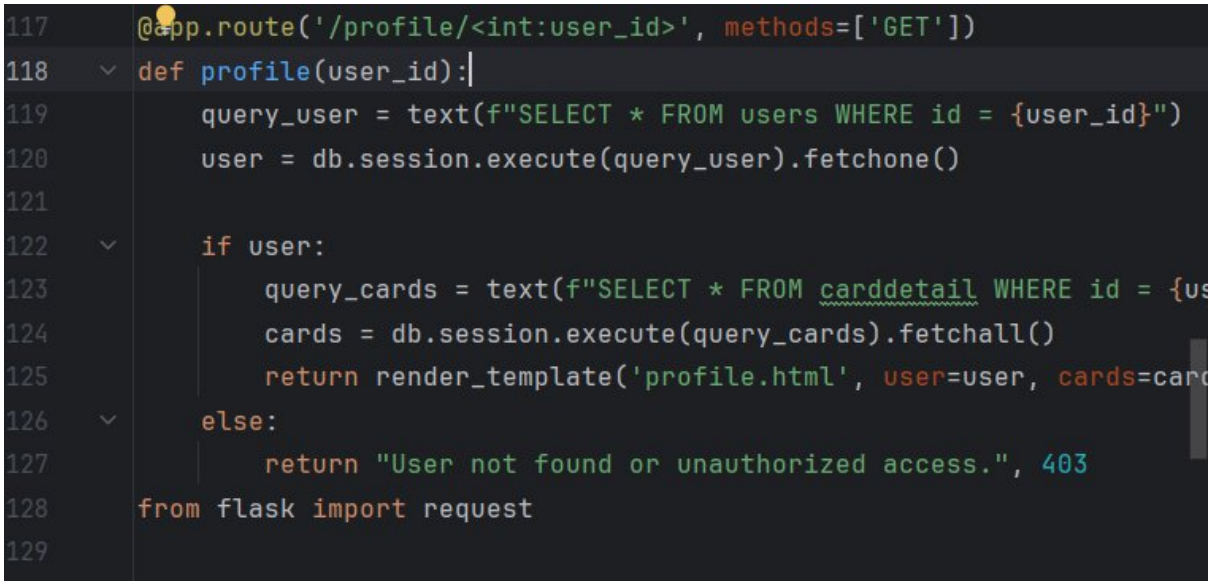
Once I logged in this was the page I had access to:



On this page in the url bar it shows us that profile id is 1, using this, we decided to change profile by changing the 1 to a 2 and refreshing the page.



Just by doing that we had access to another users account this was beacuase of IDOR, the site only checked what the user asked for and it never checked wether the user was allowed to ask for it, ie no authorisation – code below is missing authorisation



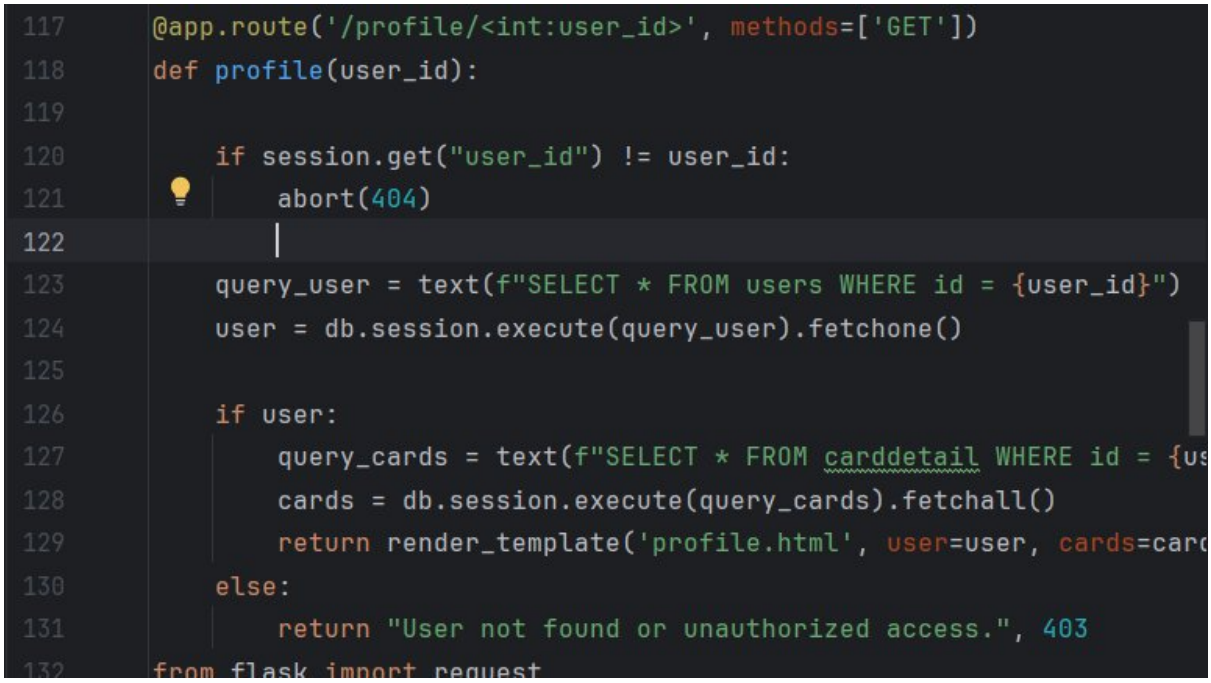
```
@app.route('/profile/<int:user_id>', methods=['GET'])
def profile(user_id):
    query_user = text(f"SELECT * FROM users WHERE id = {user_id}")
    user = db.session.execute(query_user).fetchone()

    if user:
        query_cards = text(f"SELECT * FROM carddetail WHERE id = {us
        cards = db.session.execute(query_cards).fetchall()
        return render_template('profile.html', user=user, cards=card
    else:
        return "User not found or unauthorized access.", 403
from flask import request
```

To allow for authorisation we have to add a line to the script

if session.get("user_id") != user_id:

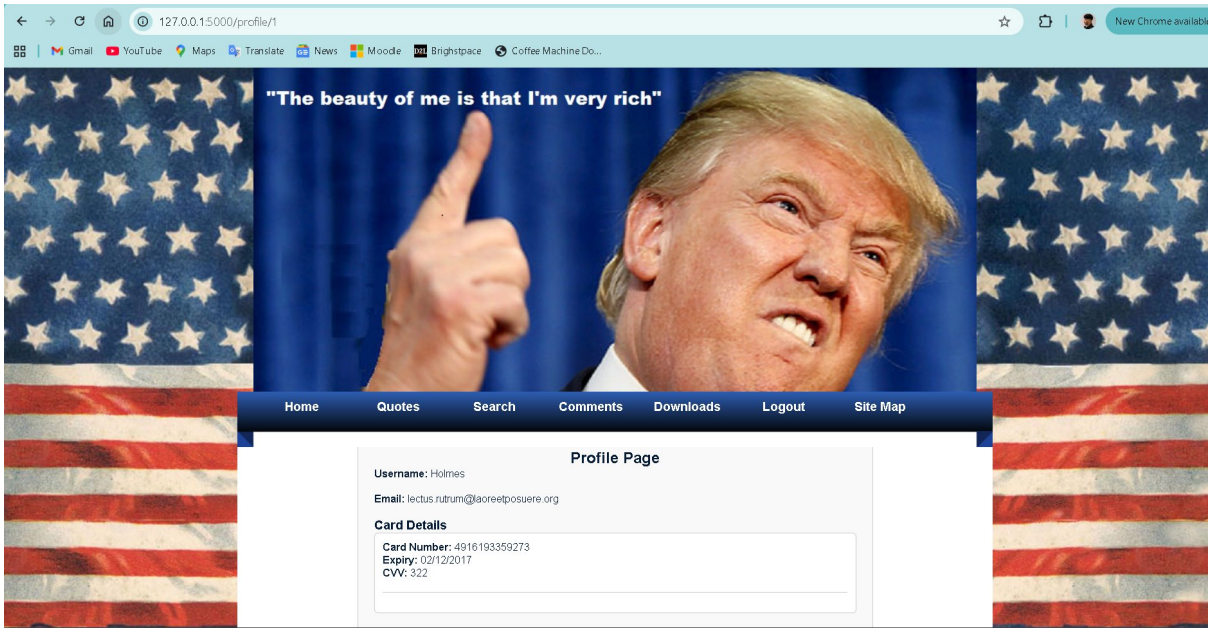
    abort(403)

this compares the session ids and aborts if it is incorrect

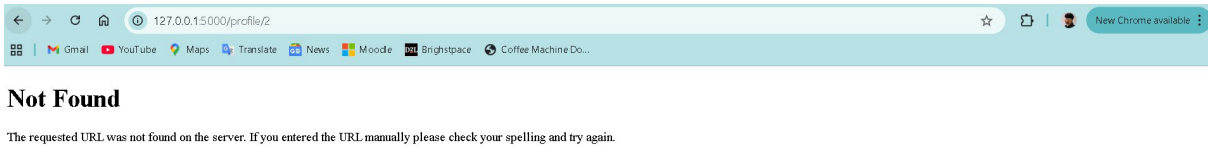


```
@app.route('/profile/<int:user_id>', methods=['GET'])
def profile(user_id):

    if session.get("user_id") != user_id:
        abort(404)

    query_user = text(f"SELECT * FROM users WHERE id = {user_id}")
    user = db.session.execute(query_user).fetchone()

    if user:
        query_cards = text(f"SELECT * FROM carddetail WHERE id = {us
        cards = db.session.execute(query_cards).fetchall()
        return render_template('profile.html', user=user, cards=card
    else:
        return "User not found or unauthorized access.", 403
from flask import request
```

Now we can test to see if it works

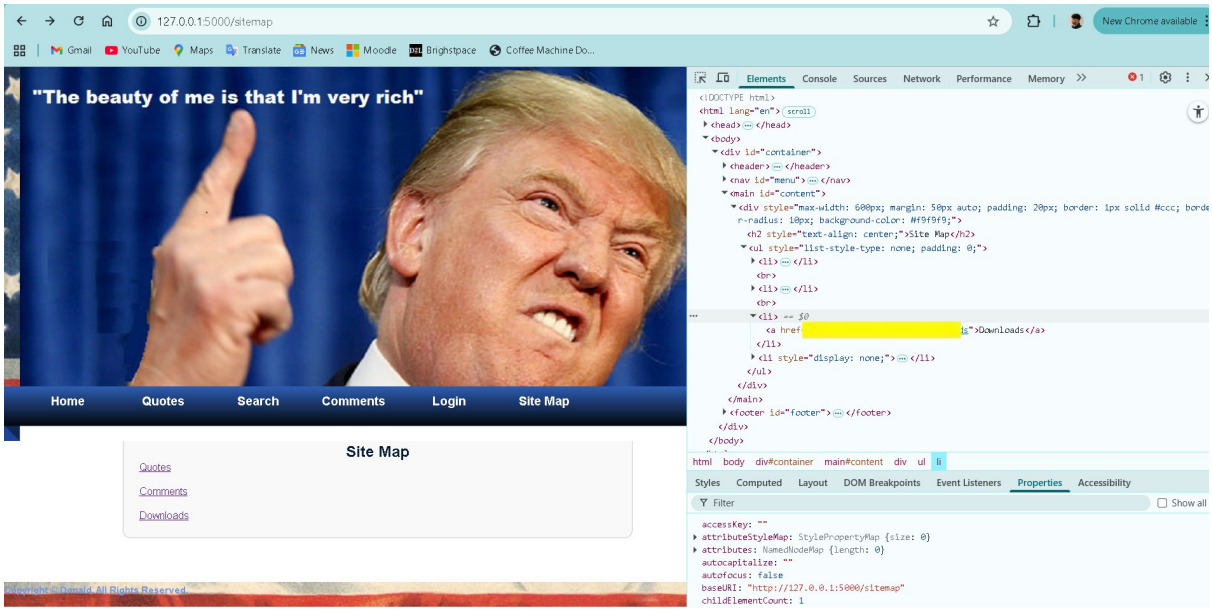


We are in profile 1, if we try change it to profile 2 like earlier:

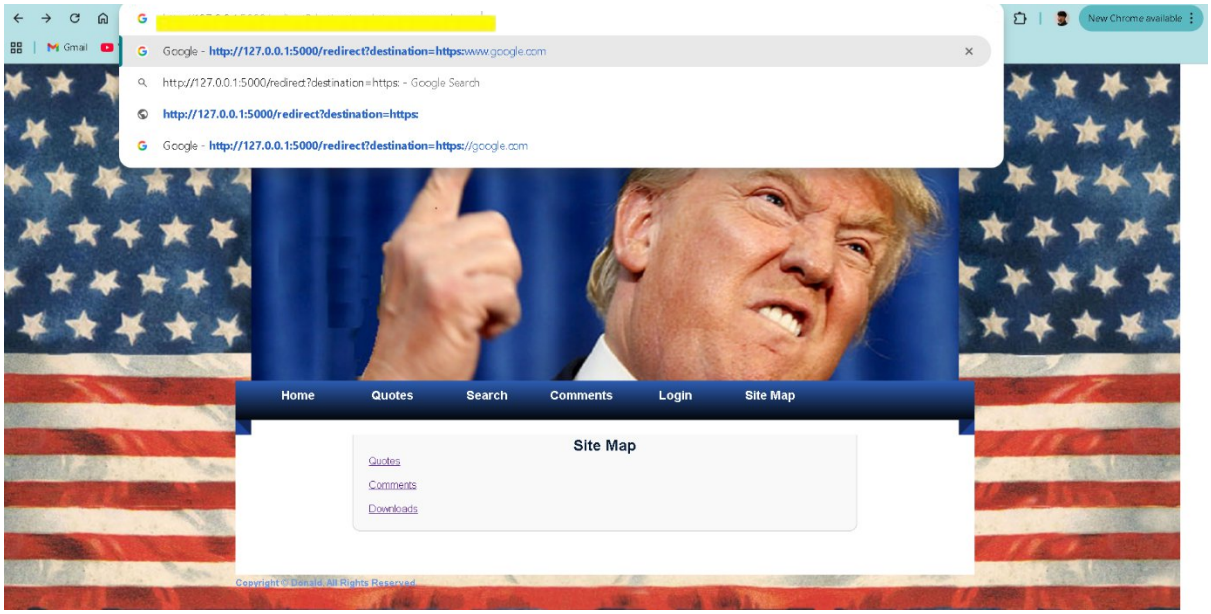


The System Aborts

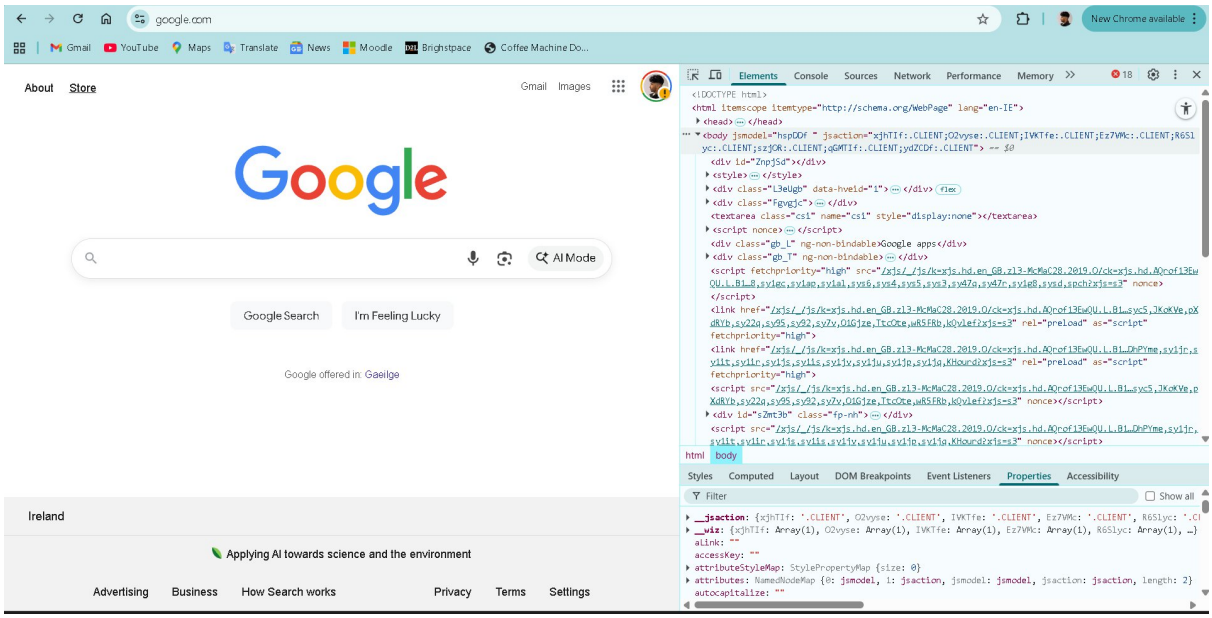
An open redirect vulnerability was discovered within the sitemap of the page. The redirect element underneath(downloads/comments/quotes) allows for redirection of the webpage to an external source without any input validation



Same vulnerability was discovered in the url tab, allowing for a redirect into other websites without input validation



Both when clicked on or pressed enter redirected to google as shown below



Problem in the code:

As you can see down below the code not check for any input validation and just redirects when asked

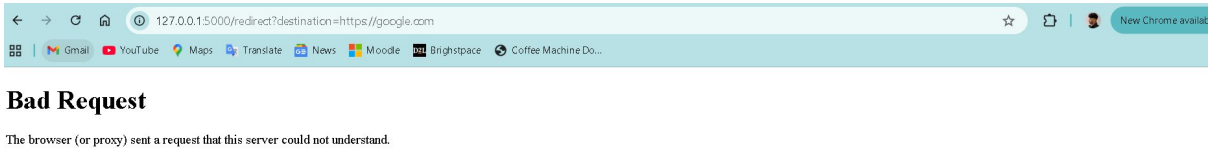
```
@app.route('/redirect', methods=['GET'])
def redirect_handler():
    destination = request.args.get('destination')

    if destination:
        return redirect(destination)
    else:
        return "Invalid destination", 400
```

To fix this: we add a line in the code which only allows for redirection to happen if its within the website or followed by the ” / “sign

```
# Route to handle redirects based on the destination query
@app.route('/redirect', methods=['GET'])
def redirect_handler():
    destination = request.args.get('destination')

    if not destination.startwith("/"):
        abort(400)


    return redirect(destination)
```

And finally to prevent the code from aborting when we click on the sitemap to downlaods, comments or quotes we have to go into html file for sitemap and then add a “/” before each of the locations

And now when we try to execute open redirect this time it fails



Bad Request

The browser (or proxy) sent a request that this server could not understand.