

# Secure Coding

## Code-CSE2010

### Lab-5

Name- Jayadeep Vinod  
Registration number-18BCN7019  
Date-27/02/2021  
Slot- L23-L24

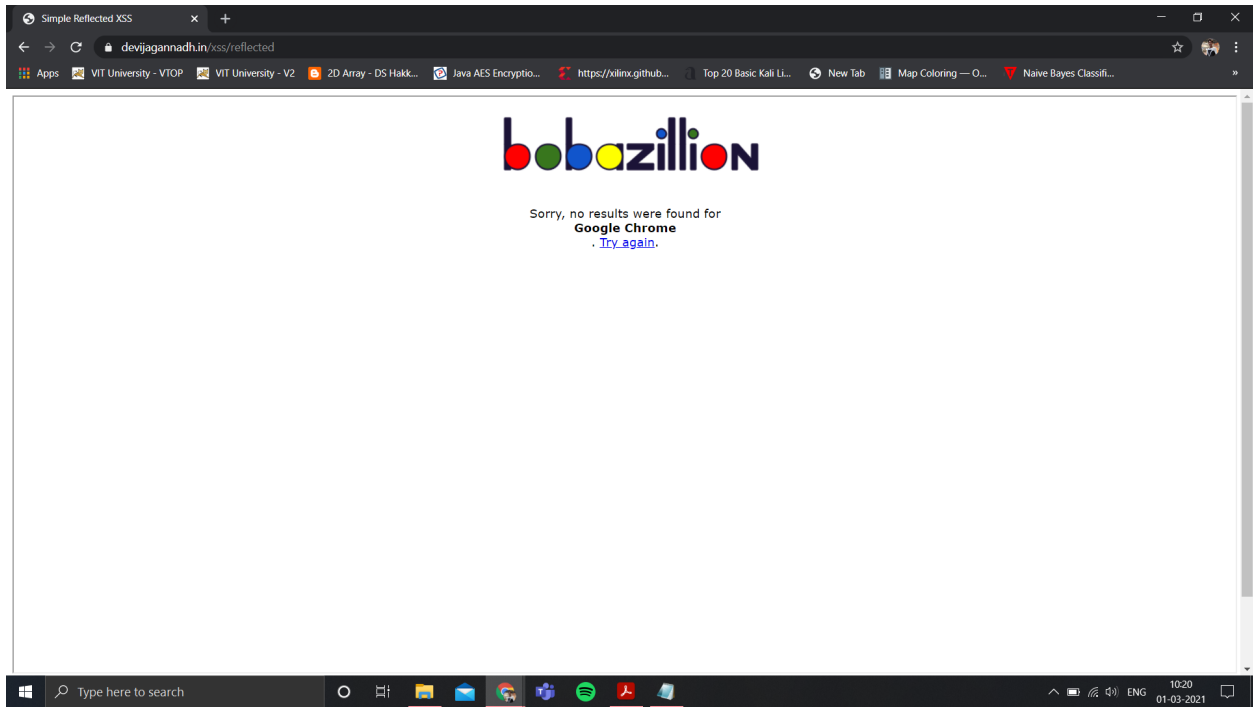
#### **How secure coding related to XSS?**

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

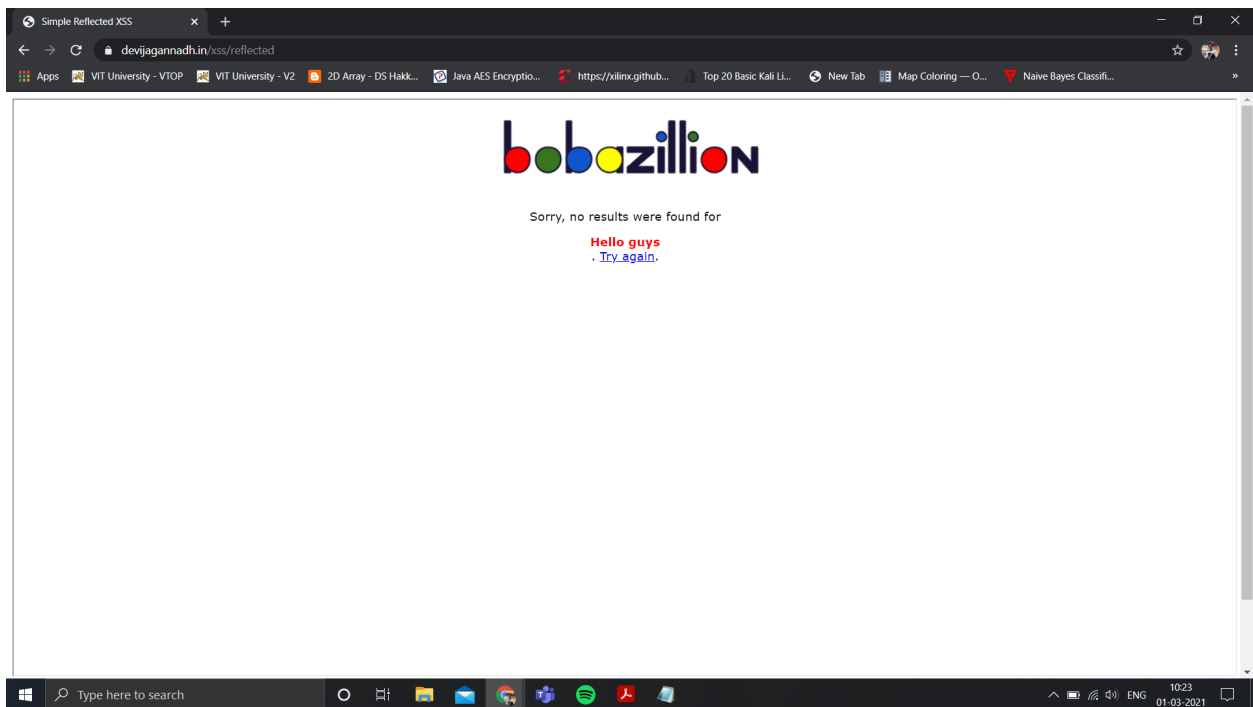
An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

# Rxss on demo website

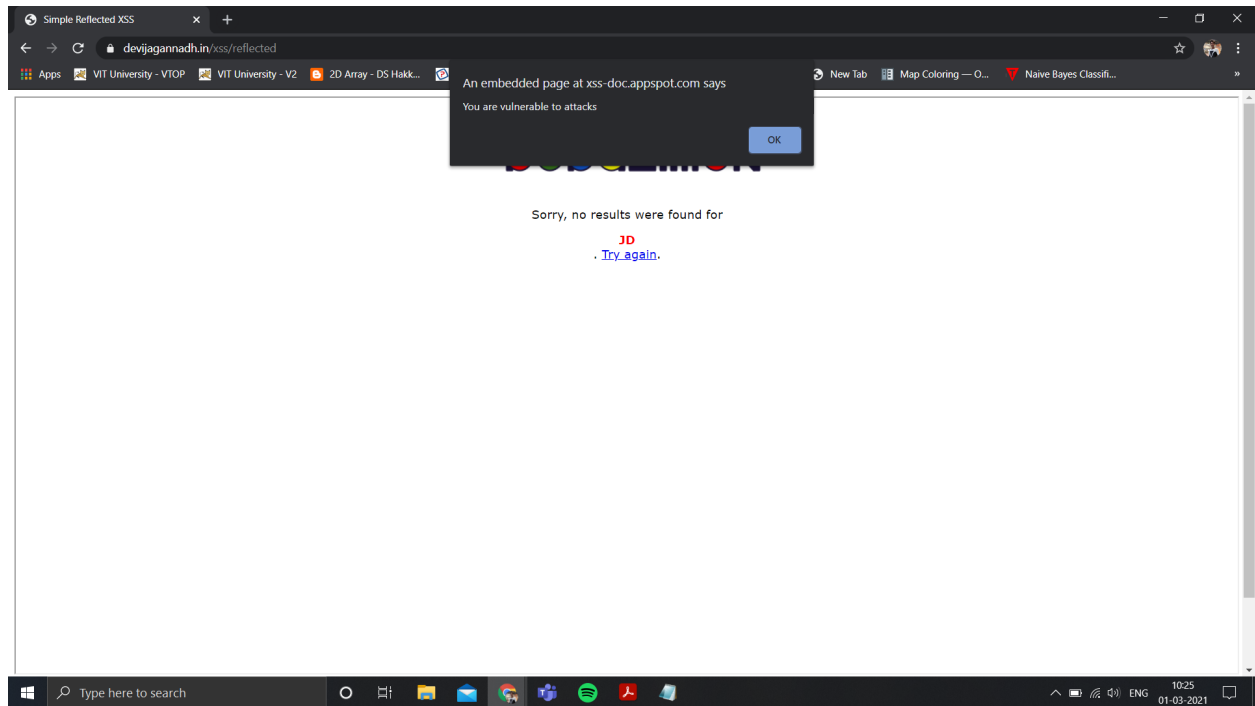
## 1.<br>Google Chrome</br>



## 2.<div style=color:red>Hello guys</div>

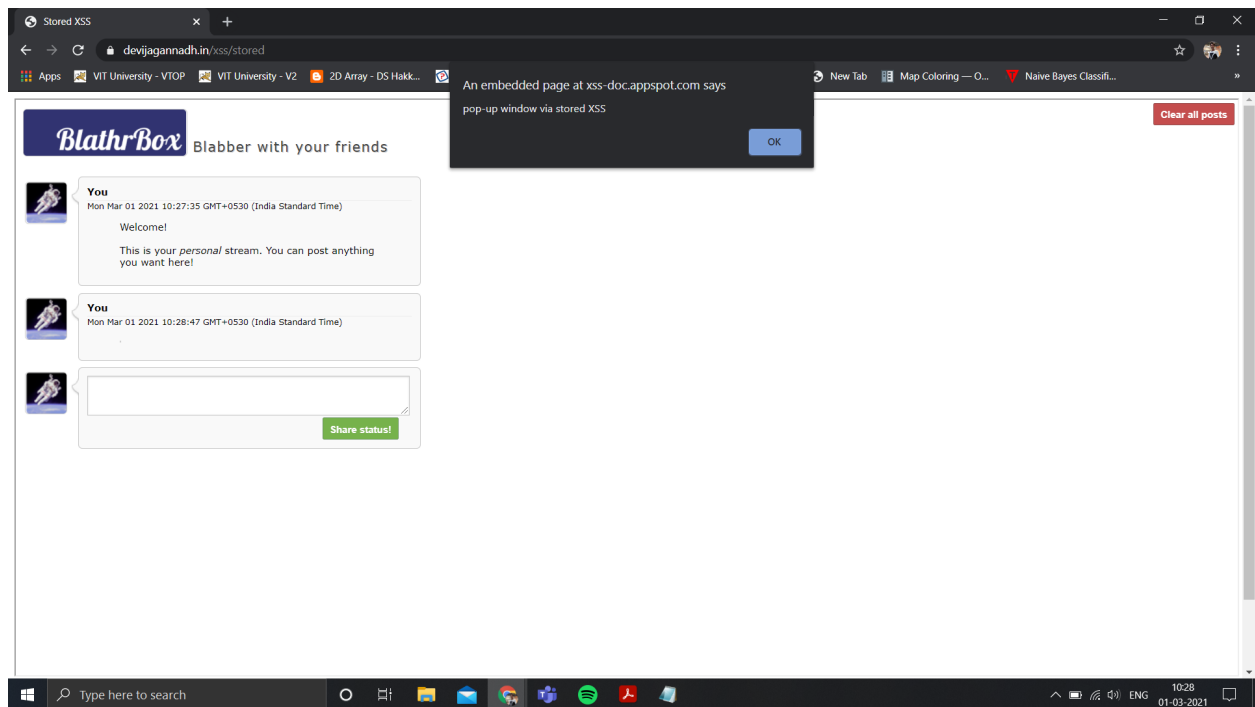


3.<div style=color:red>JD</div> <script>alert("You are vulnerable to attacks")</script>



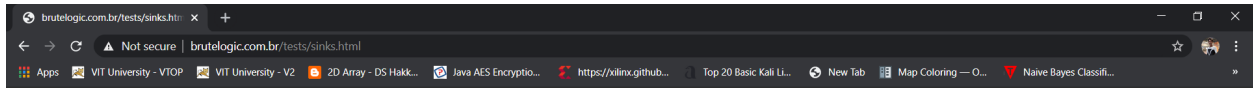
## Stored xss on demo website

1.<img src=x onerror="alert('pop-up window via stored XSS');"

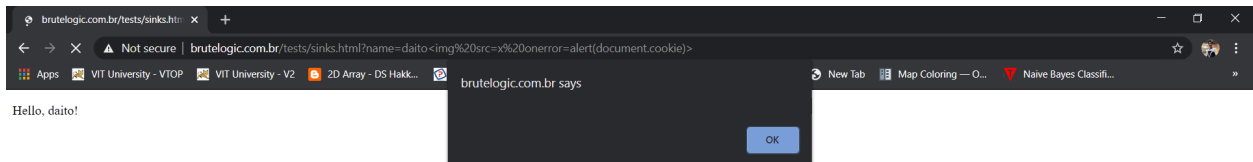


## DOM xss on demo website

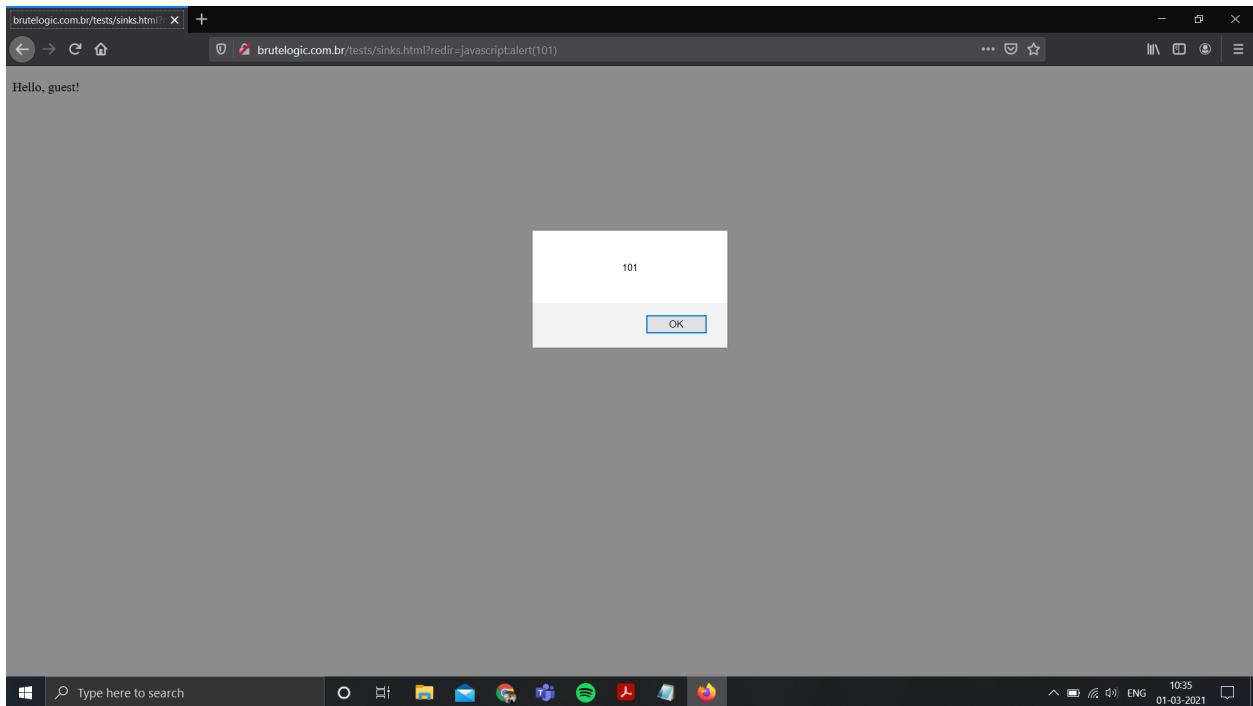
1. <http://brutelogic.com.br/tests/sinks.html>



2. [http://brutelogic.com.br/tests/sinks.html?name=daito<img src=x onerror=alert\(document.cookie\)>](http://brutelogic.com.br/tests/sinks.html?name=daito<img src=x onerror=alert(document.cookie)>)



### 3. [http://brutelogic.com.br/tests/sinks.html?redir=javascript:alert\(101\)](http://brutelogic.com.br/tests/sinks.html?redir=javascript:alert(101))



## Challenge : [alf.nu/alert1](http://alf.nu/alert1)

