A PARADIGM SHIFT IN XSS-DOM MITIGATION VIA DCSP AND PROXY ORCHESTRATION

REPORT
IT5811 - PROJECT I

Submitted by

Krishnaa S (2020506045)
Jawahar A S (2020506035)
Thamizharasi M (2020506102)

Guided by

Dr. P. Kola Sujatha
Associate Professor

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600044

OCT 2023

INTRODUCTION:

Cross-Site Scripting (XSS) and Document Object Model (DOM) attacks are critical security concerns for web applications. XSS vulnerabilities allow attackers to inject malicious scripts into web pages viewed by other users, potentially compromising their data or session information. These attacks exploit the trust a user has in a website, enabling unauthorized access or data theft. DOM-based XSS attacks specifically manipulate a webpage's Document Object Model, which represents the structure of a web page in a hierarchical manner. In DOM-based XSS, malicious scripts manipulate the DOM of a web page dynamically, often targeting client-side scripts and affecting the behavior of web applications. Detecting and mitigating DOM-based XSS attacks is essential for safeguarding user data, maintaining application integrity, and preventing unauthorized access to sensitive information. Security measures involve input validation, output encoding, and monitoring for unusual script behavior to thwart these attacks and ensure a secure web environment.

Our focus is on using pattern matching techniques to detect XSS-DOM vulnerabilities, adding an extra layer of security to web applications. This proactive approach helps identify potential attacks early, enhancing overall security. We also explore preventive strategies, such as input validation and Content Security Policy (CSP), in detail. These strategies are effective in reducing the risks associated with XSS-DOM attacks, making web environments more secure against potential breaches.

OBJECTIVES:

Understanding Vulnerabilities: Gain insight into the nature and mechanics of Cross-Site Scripting (XSS) and DOM-based attacks.

Detection Techniques: Investigate methods (Pattern matching) for identifying XSS-DOM vulnerabilities, enabling proactive defense against potential attacks.

Preventive Strategies: Explore strategies such as input validation and Content Security Policy (CSP) to mitigate the risks of XSS-DOM attacks.

Real-world Insights: Analyze real-world cases of notable attacks to grasp the potential consequences and implications of these vulnerabilities.

PROBLEM STATEMENT:

Cross-Site Scripting (XSS) vulnerabilities persist in web applications, allowing malicious scripts to compromise user data and privacy.Existing Content Security Policy solutions struggle to effectively protect dynamic web pages where content changes based on user interactions.

Lack of adaptive security measures leaves dynamic pages exposed to XSS attacks, necessitating a novel solution.

TECH STACK:

- Python
- Flask
- Javascript
- Sqlite

System Requirements:

- Windows 10 and above
- VS Code
- Chrome Browser

IMPLEMENTATION MODULES:

Module 1: DETECTION

- In this module, we have implemented a set of regular expression patterns designed to enhance the security system's ability to identify potential XSS-DOM vulnerabilities.
- This pattern integration allows for more robust detection of security threats related to client-side scripting and DOM manipulation.

Module 2: MITIGATION

- Under this module, we explore the integration of Content Security Policy (CSP) as a proactive measure to mitigate the risks associated with client-side scripting and DOM manipulation.
- We employ a specific whitelist framework to ensure no scripts or resources are loaded from a source that we are unaware of.
- Additionally, we receive alerts when such an attack is detected and blocked.
- These alerts include detailed information about the detected threats and recommended actions for effective incident response.
- This combined approach ensures that not only are vulnerabilities identified, but also steps are taken to mitigate and respond to potential threats effectively.

NOVELTY:

Traditional Approach:

Existing Content Security Policy solutions struggle to effectively protect dynamic web pages where content changes based on user interactions. Lack of adaptive security measures are also not addressed.

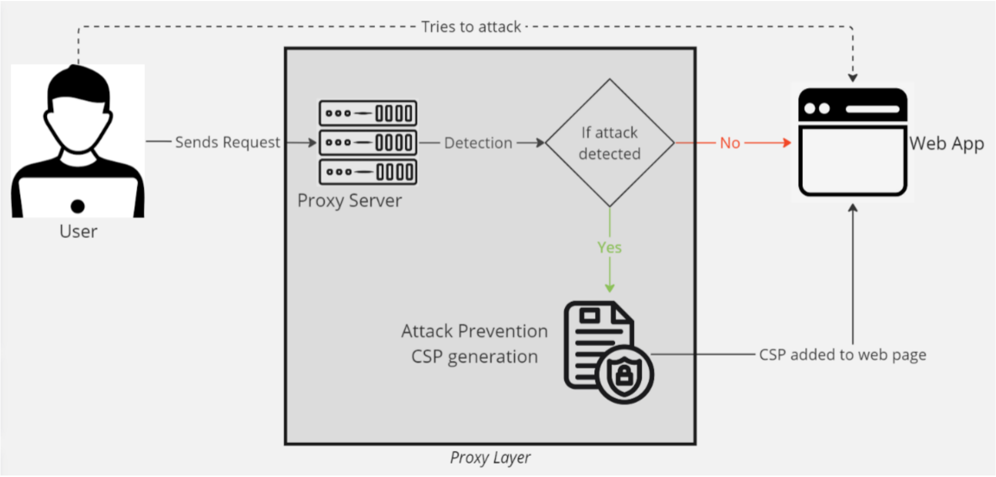
Our Approach:

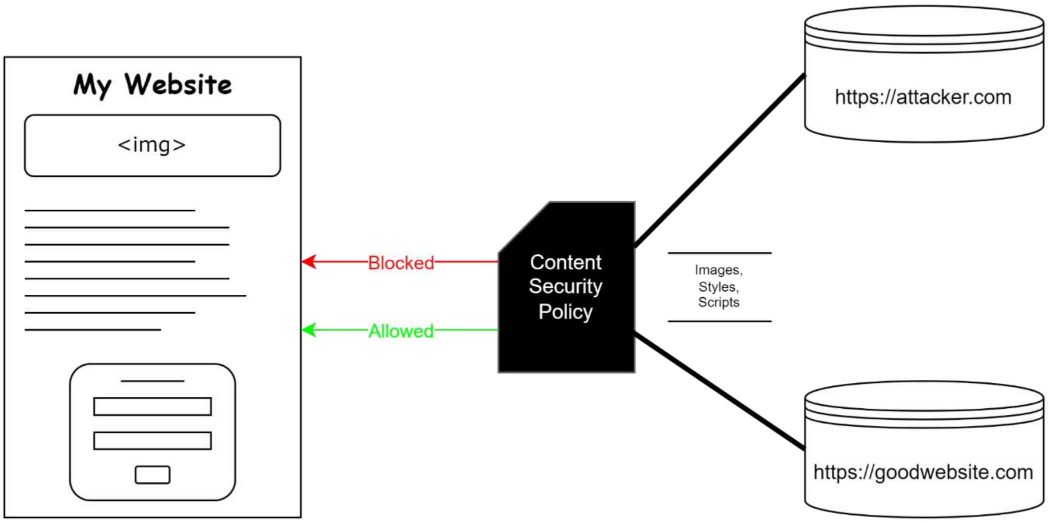
1. **Novel Approach**: The project takes a novel approach to address the critical security concerns of Cross-Site Scripting (XSS) and Document Object

Model (DOM) attacks in web applications, emphasizing pattern matching techniques to proactively detect vulnerabilities.

2. **Early Detection**: By utilizing pattern matching techniques, the project aims to identify XSS-DOM vulnerabilities at an early stage, providing an extra layer of security that can help prevent potential attacks before they occur, enhancing overall web application security.
3. **Comprehensive Strategies**: In addition to detection, the project delves into comprehensive preventive strategies such as input validation and Content Security Policy (CSP), offering a holistic approach to reducing the risks associated with XSS-DOM attacks and strengthening the security of web environments.
4. **Mitigating Dynamic Attacks**: DOM-based XSS attacks dynamically manipulate web page structure, and the project focuses on detecting and mitigating these dynamic threats, which often target client-side scripts and affect the behaviour of web applications.
5. **Enhanced Web Security**: The project's emphasis on detecting and preventing XSS-DOM attacks contributes to safeguarding user data, maintaining application integrity, and preventing unauthorized access to sensitive information, ultimately ensuring a more secure web environment.

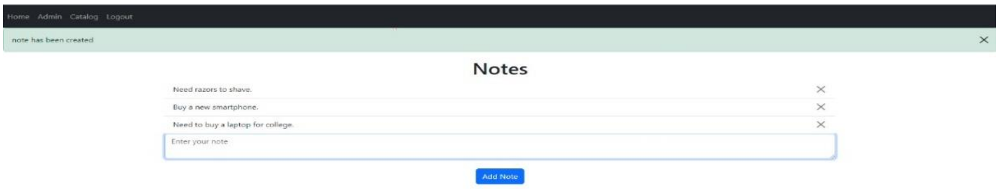
ARCHITECTURE DIAGRAM:





OUTPUT SCREENSHOTS:

Notes:



note has been created

Notes

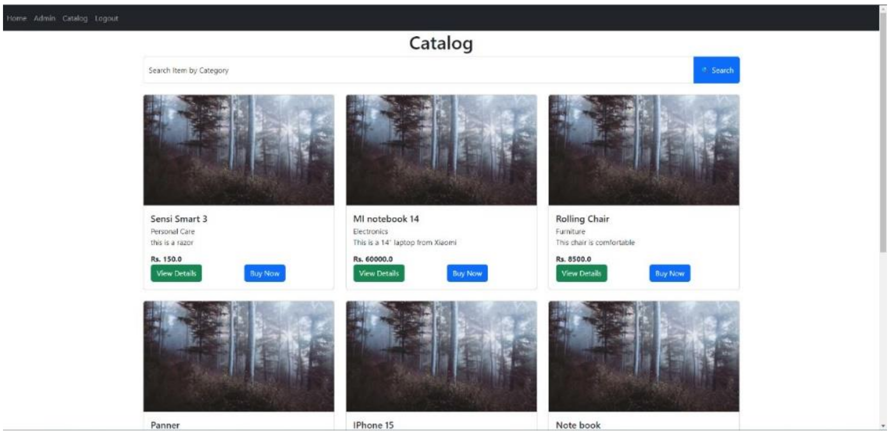
Need razors to shave.

Buy a new smartphone.

Need to buy a laptop for college.

Enter your note

Catalog:



Catalog

Search Item by Category

Sensi Smart 3
Personal Care
this is a razor
Rs. 150.0

MI notebook 14
Electronics
This is a 14" laptop from Xiaomi
Rs. 60000.0

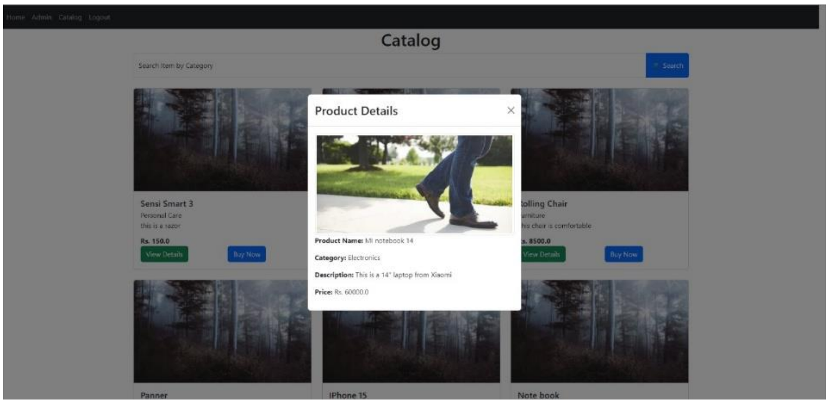
Rolling Chair
Furniture
This chair is comfortable
Rs. 8500.0

Panner

IPhone 15

Note book

Product Viewing:



Catalog

Search Item by Category

Sensi Smart 3
Personal Care
this is a razor
Rs. 150.0

olling Chair
urniture
his chair is comfortable
s. 8500.0

Product Details

Product Name: MI notebook 14

Category: Electronics

Description: This is a 14" laptop from Xiaomi

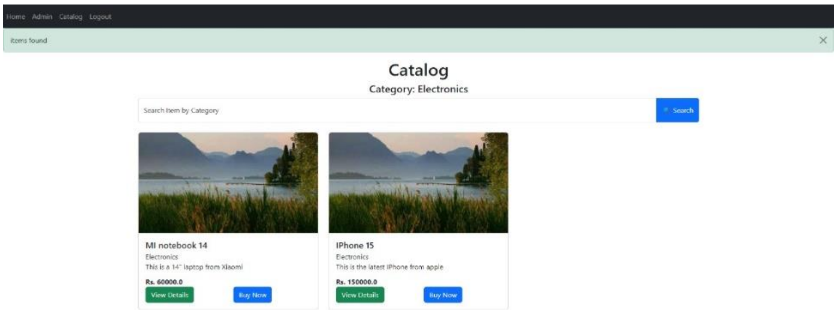
Price: Rs. 60000.0

Panner

IPhone 15

Note book

Searching Products:



items found

Catalog

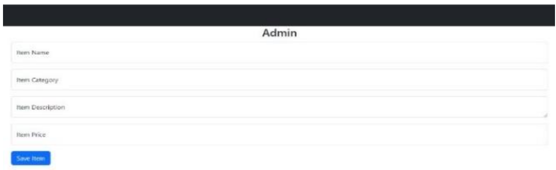
Category: Electronics

Search Item by Category

MI notebook 14
Electronics
This is a 14" laptop from Xiaomi
Rs. 60000.0

IPhone 15
Electronics
This is the latest IPhone from apple
Rs. 150000.0

Admin,Login and Signup pages:



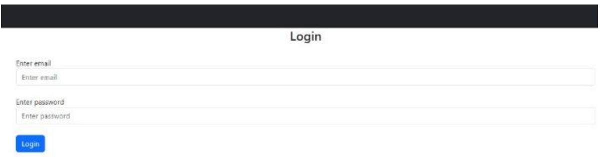
Admin

Item Name

Item Category

Item Description

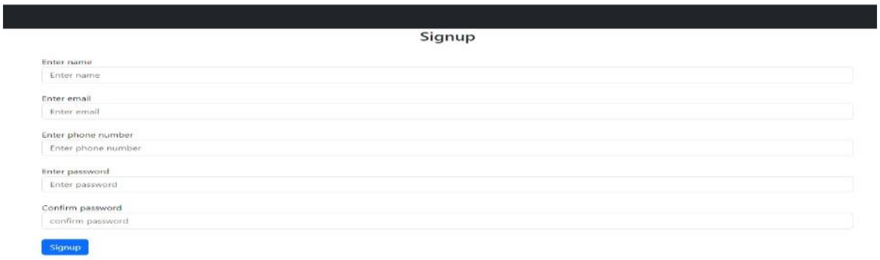
Item Price



Login

Enter email
Enter email

Enter password
Enter password



Signup

Enter name
Enter name

Enter email
Enter email

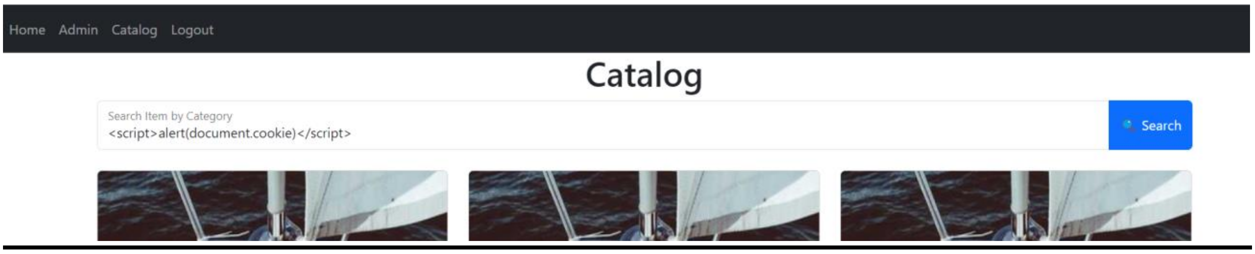
Enter phone number
Enter phone number

Enter password
Enter password

Confirm password
confirm password

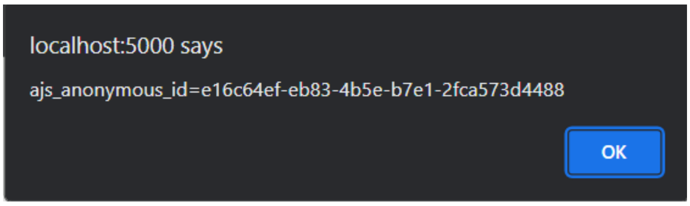
Injecting Script in various ways:

Via search bar



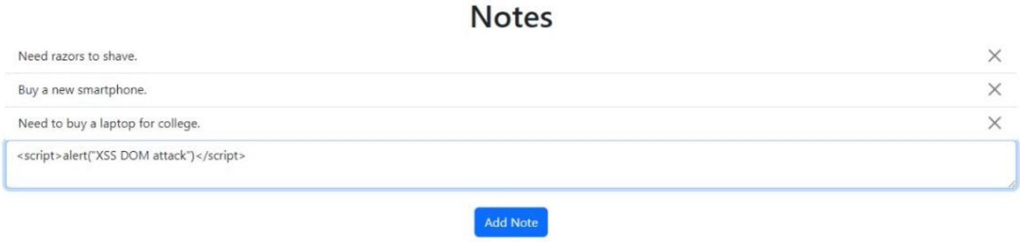
Catalog

Search Item by Category
<script>alert(document.cookie)</script>


localhost:5000 says

ajs_anonymous_id=e16c64ef-eb83-4b5e-b7e1-2fca573d4488

Via Notes page

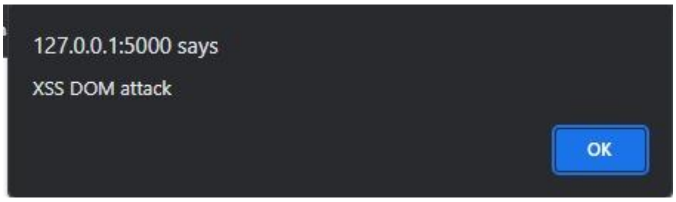

Notes

Need razors to shave.

Buy a new smartphone.

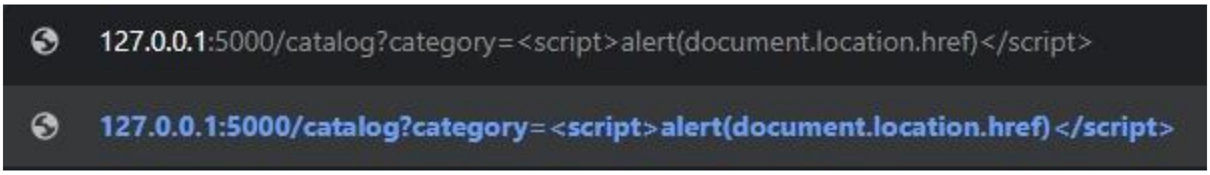
Need to buy a laptop for college.

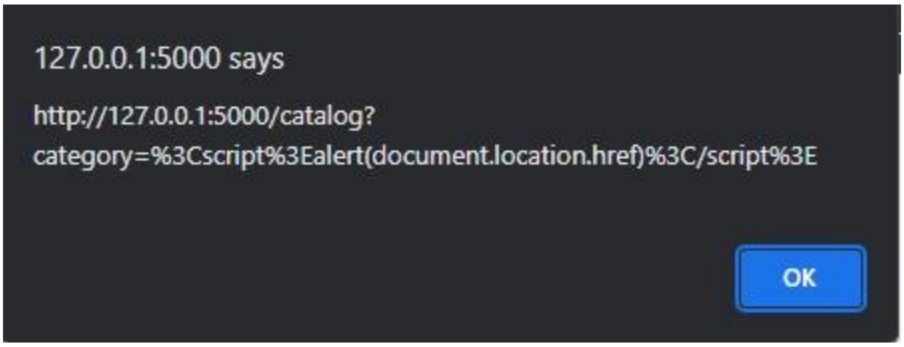
<script>alert("XSS DOM attack")</script>


127.0.0.1:5000 says

XSS DOM attack

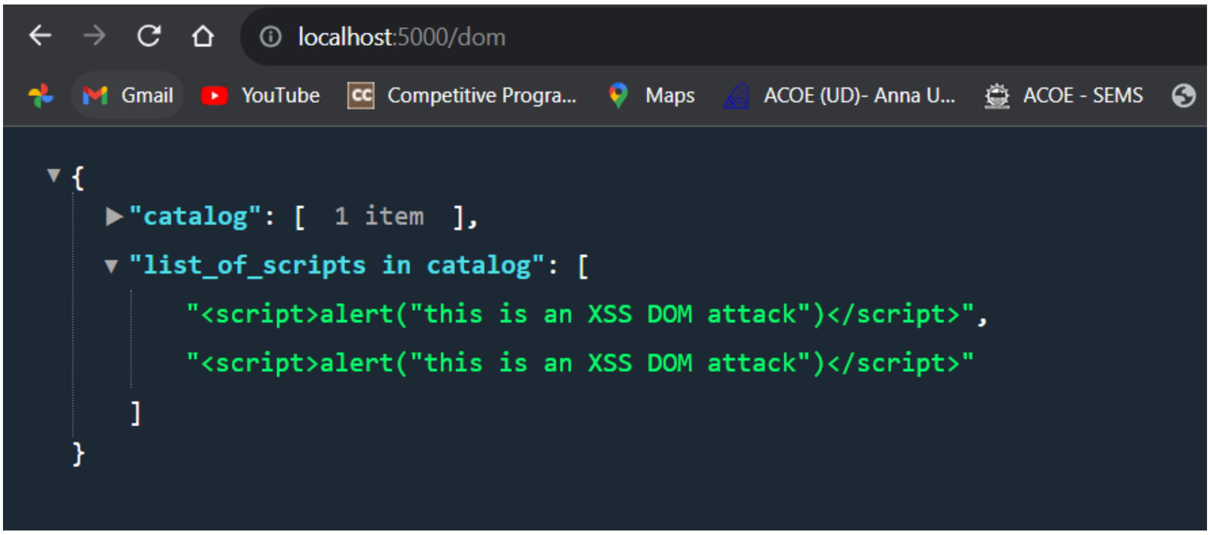
Via URL


```
127.0.0.1:5000/catalog?category=<script>alert(document.location.href)</script>
127.0.0.1:5000/catalog?category=<script>alert(document.location.href)</script>
```


127.0.0.1:5000 says

http://127.0.0.1:5000/catalog?category=%3Cscript%3Ealert(document.location.href)%3C/script%3E

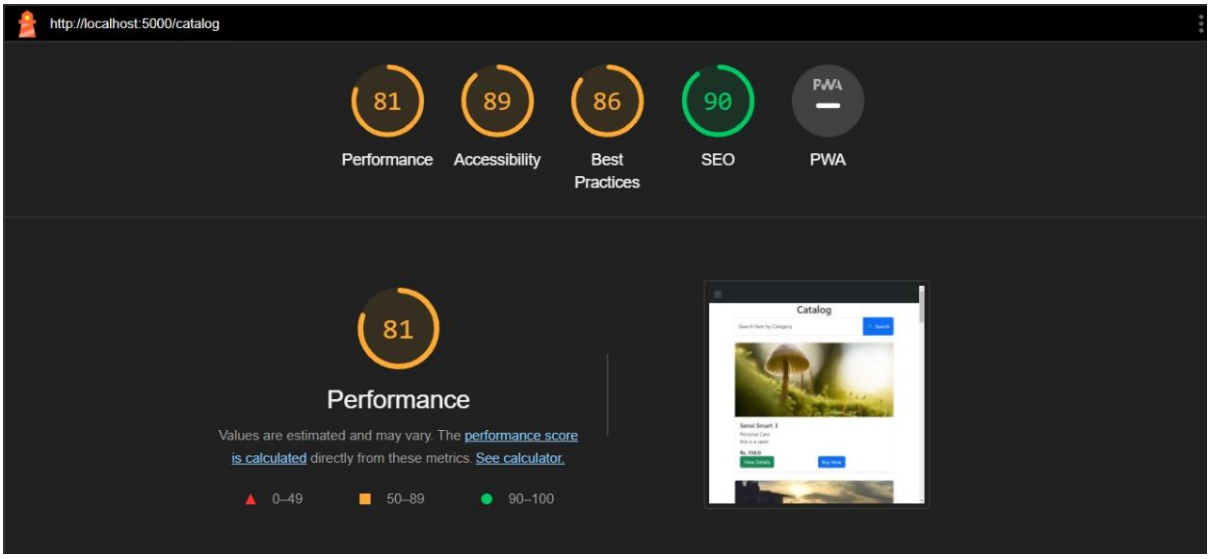
Detection of XSS DOM attacks via RegEx (pattern matching):



```
▼{
  ▶"catalog": [ 1 item ],
  ▼"list_of_scripts in catalog": [
      "<script>alert("this is an XSS DOM attack")</script>",
      "<script>alert("this is an XSS DOM attack")</script>"
  ]
}
```

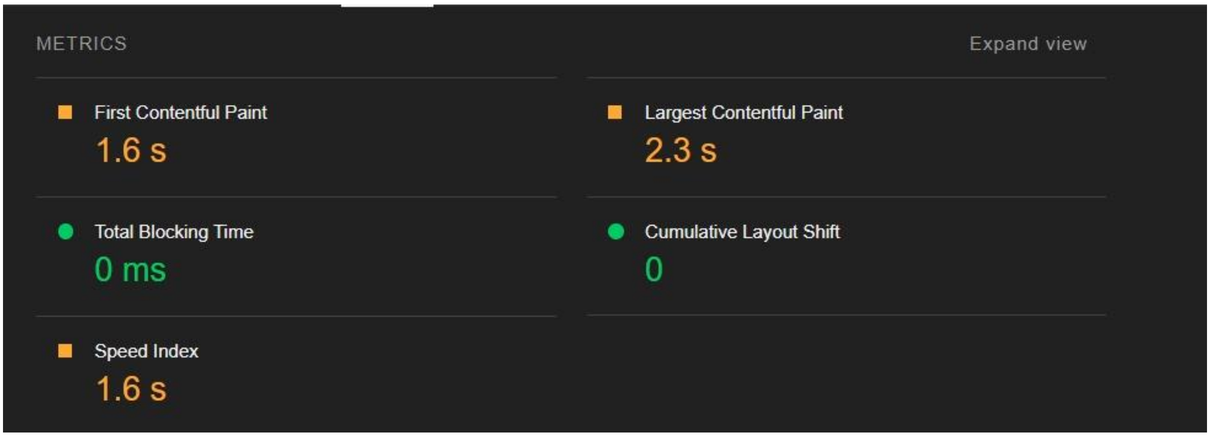
EVALUATION METRICS:

Evaluated website using Google Lighthouse



Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100



■ First Contentful Paint 1.6 s	■ Largest Contentful Paint 2.3 s
● Total Blocking Time 0 ms	● Cumulative Layout Shift 0
■ Speed Index 1.6 s	

RELATED WORK:

BASE PAPER:

G. Xu et al., "JSCSP: A Novel Policy-Based XSS Defense Mechanism for Browsers," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 862-878, 1 March-April 2022, doi: 10.1109/TDSC.2020.3009472.

REFERENCES:

1. K. Ali, A. Abdel-Hamid and M. Kholief, "Prevention Of DOM Based XSS Attacks Using A White List Framework," 2014 24th International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, 2014, pp. 68-75, doi: 10.1109/ICCTA35431.2014.9521633.
2. S. K. Mahmoud, M. Alfonse, M. I. Roushdy and A. -B. M. Salem, "A comparative analysis of Cross Site Scripting (XSS) detecting and defensive techniques," 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, 2017, pp. 36-42, doi: 10.1109/INTELCIS.2017.8260024.
3. J. C. Pazos, J. -S. Légaré and I. Beschastnikh, "XSnare: Application-specific client-side cross-site scripting protection," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 2021, pp. 154-165, doi: 10.1109/SANER50967.2021.00023.
4. Z. Jingyu, H. Hongchao, H. Shumin and L. Huanruo, "A XSS Attack Detection Method Based on Subsequence Matching Algorithm," 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou, China, 2021, pp. 83-86, doi: 10.1109/AIID51893.2021.9456515.