

Title: Software Testing MicroProject
Submitted in partial fulfilment of the requirements

of the Diploma of

in

COMPUTER ENGINEERING

by

Name	Roll no
Shrutika parmar	22203A0028
Samruddhi bajage	22203A0021
Vaishnavi parulekar	22203A0048

under the guidance of

Supriya Kadam maam



Department of Computer Engineering

Vidyalankar Polytechnic

Wadala (E), Mumbai -37

(Affiliated to MSBTE)

2022-2023

PART-A (About 2-3 Pages)

Part-A

Format for Micro-Project Proposal

For 1st to 4th Semester

Title of Micro Project: Manual Testing and Automation testing using selenium ide for Cross-Site Scripting (XSS) Vulnerabilities

1.0 Brief Introduction

Cross-Site Scripting (XSS) is a prevalent security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can manipulate the content of the page, steal sensitive information such as cookies, and perform unauthorized actions on behalf of the user. Given the increasing sophistication of cyber threats, it is crucial for web applications to implement robust security measures against XSS attacks.

This manual testing exercise focuses on identifying potential XSS vulnerabilities in the website <https://gresporcelan.in/>. By simulating various types of XSS payloads and evaluating the website's response, this testing aims to ensure that user inputs are properly sanitized and that the application does not expose users to risks associated with XSS vulnerabilities.

Automation testing for Cross-Site Scripting (XSS) using Selenium IDE helps identify vulnerabilities by simulating user actions and injecting potentially harmful scripts into web pages. This ensures that web applications are protected from XSS attacks, which can compromise user data and site functionality. By automating the process, testers can efficiently detect and fix vulnerabilities early, making the application more secure.

2.0 Aim of the Micro Project

The aim of this manual testing for XSS vulnerabilities on the website <https://gresporcelan.in/> is to systematically identify potential security weaknesses by testing various input fields and user-generated content areas for Cross-Site Scripting (XSS) vulnerabilities. This testing seeks to assess the effectiveness of existing input validation and output encoding mechanisms, ensuring that they adequately protect against XSS attacks. Additionally, the aim includes evaluating the security of sensitive cookies to determine if they can be compromised through XSS exploits, which could lead to unauthorized access to user sessions. Ultimately, this effort intends to enhance the website's security posture by providing actionable recommendations to address identified vulnerabilities and implement best practices for preventing XSS attacks, thereby ensuring a safe user experience.

The aim of automating XSS testing using Selenium IDE is to efficiently detect vulnerabilities by simulating user interactions and injecting malicious scripts. This ensures early identification and mitigation of XSS threats, enhancing the overall security of web applications without manual intervention.

Department of Computer Engineering

3.0 Action Plan (Sequence and time required for major activities for 8 weeks)

Sr. No.	Details of Activity	Planned Start Date	Planned Finish Date	Name of Responsible Team Members
1	Selection of topic	20/09/24	21/09/24	Vaishnavi Parulekar Samruddhi Bajage
2	Create a Test Plan	22/09/24	23/09/24	Shrutika Parmar
3	Design Test Cases	24/09/24	28/09/24	Samruddhi Bajage
4.	Execute Test Cases	29/09/24	01/10/24	Vaishnavi Parulekar
4	Find Vulnerabilities	01/10/24	01/10/24	Vaishnavi Parulekar
5	Log Defects	03/10/24	03/10/24	Samruddhi Bajage
6	Create Defect Report	06/10/24	09/10/24	Shrutika Parmar
7	Automation testing	07/10/24	09/10/24	Shrutika Parmar Vaishnavi Parulekar
8	Create Final Report	11/10/24	14/10/24	Shrutika Parmar

4.0 Resources Required (Such as raw material, some machining facility, software etc.)

Sr. No.	Name of Resource/Material	Specifications	Qty	Remarks
1	Laptop			
2	MS word			
3	Internet			
4	Selenium ide			

PART-B (Outcomes after Execution and Format for Micro-Project Report, About 6-10 Pages) For 1st to 4th Semester

Title of Micro Project: Manual Testing for Cross-Site Scripting (XSS) Vulnerabilities

1.0 Brief Description

Cross-Site Scripting (XSS) is a prevalent security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can manipulate the content of the page, steal sensitive information such as cookies, and perform unauthorized actions on behalf of the user. Given the increasing sophistication of cyber threats, it is crucial for web applications to implement robust security measures against XSS attacks.

This manual testing exercise focuses on identifying potential XSS vulnerabilities in the website <https://gresporcelan.in/>. By simulating various types of XSS payloads and evaluating the website's response, this testing aims to ensure that user inputs are properly sanitized and that the application does not expose users to risks associated with XSS vulnerabilities.

Automation testing for Cross-Site Scripting (XSS) using Selenium IDE helps identify vulnerabilities by simulating user actions and injecting potentially harmful scripts into web pages. This ensures that web applications are protected from XSS attacks, which can compromise user data and site functionality. By automating the process, testers can efficiently detect and fix vulnerabilities early, making the application more secure.

2.0 Aim of Micro Project:

The aim of this manual testing for XSS vulnerabilities on the website <https://gresporcelan.in/> is to systematically identify potential security weaknesses by testing various input fields and user-generated content areas for Cross-Site Scripting (XSS) vulnerabilities. This testing seeks to assess the effectiveness of existing input validation and output encoding mechanisms, ensuring that they adequately protect against XSS attacks. Additionally, the aim includes evaluating the security of sensitive cookies to determine if they can be compromised through XSS exploits, which could lead to unauthorized access to user sessions. Ultimately, this effort intends to enhance the website's security posture by providing actionable recommendations to address identified vulnerabilities and implement best practices for preventing XSS attacks, thereby ensuring a safe user experience.

The aim of automating XSS testing using Selenium IDE is to efficiently detect vulnerabilities by simulating user interactions and injecting malicious scripts. This ensures early identification and mitigation of XSS threats, enhancing the overall security of web applications without manual intervention.

3.0 Course Outcomes Integrated

- Prepare test cases for different types and levels of testing
- Prepare test plan for an application
- Identify bugs to create defect report of given application
- Test software for performance measures using automation testing tool

4.0 Actual Procedure followed

- Topic Selection
- Test Plan Creation
- Resource Identification
- Test Case Design
- Test Case Execution
- Vulnerability Identification
- Defect Logging
- Defect Report
- Automation testing
- Final Report Creation

- Project Closure

5.0 Actual Resources Used: (Mention the actual resources used)

Sr. No.	Name of Resource/Material	Specifications	Qty	Remarks
1	Laptop			
2	MS word			
3	Internet			
4	Selenium ide			

6.0 Outputs of the Micro Projects

TEST CASE

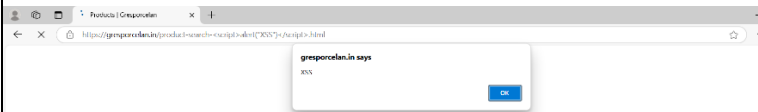
Test Case ID	Test Case Name	Steps	Expected Output	Actual Output	Status
TC-001	Basic XSS Payload	Navigate to the search bar. 2. Enter <code><script>alert('XSS');</script></code> . 3. Click Search.	Alert with "XSS" should not appear.	Alert with "XSS" appeared.	Fail
TC-002	XSS with Delay	Navigate to the search bar. 2. Enter <code><script>setTimeout(() => alert('XSS'), 1000);</script></code> . 3. Click Search.	Alert with "XSS" should not appear after 1 second	Alert with "XSS" appeared after 1 second	Fail
TC-003	Multiple Script Tags	Navigate to the search bar. 2. Enter <code><script>alert(1);</script><script>alert(2);</script></code> . 3. Click Search.	Alert with "1" and "2" should not appear.	Alert with "1" and "2" appeared.	Fail
TC-004	Comment Tag Injection	Navigate to the search bar. 2. Enter <code><!--<script>alert('XSS');</script>--></code> . 3. Click Search.	Alert with "XSS" should not appear.	Alert with "XSS" appeared	Fail
TC-005	Access Document Cookie	Navigate to the search bar. 2. Enter <code><script>alert(document.cookie);</script></code> . 3. Click Search.	It should not display the cookie.	It is displaying the cookie.	Fail
TC-006	Malicious Link	1. Navigate to the search bar. 2. Enter <code>Click here</code> . 3. Click Search.	Alert with "XSS" should not appear.	Alert with "XSS" appeared.	Fail
TC-007	Cookie Manipulation	1. Navigate to the search bar. 2. Enter <code><script>document.cookie='test=XSS'; alert(document.cookie);</script></code> . 3. Click Search.	Alert should not display cookie values or manipulated cookie.	It is displaying the Alert cookie values and manipulated cookie.	Fail

Department of Computer Engineering

TC-008	Alert with Document Cookie	1. Navigate to the search bar. 2. Enter <code><script>console.log(document.cookie); alert('Cookie Check');</script></code> . 3. Click Search.	It should not display the cookie.	It is displaying the cookie.	Fail
TC-009	Multiple Cookie Access Attempts	1. Navigate to the search bar. 2. Enter <code><script>for (var i = 0; i < 10; i++) { alert(document.cookie); }</script></code> . 3. Click Search.	Alerts should not show cookie values multiple times.	It is showing Alerts cookie values multiple times.	Fail

DEFECT REPORT

ID	DEF_01
Project	Manual Testing for Cross-Site Scripting (XSS) Vulnerabilities
Product	https://gresporcelan.in/
Release Version	-
Module	Search bar
Detected Build Version	-
Summary	XSS vulnerability identified in search bar
Description	An XSS vulnerability was identified in the search bar of the website, allowing scripts to be executed, which can compromise user data and session integrity.

Steps to Replicate	<ol style="list-style-type: none"> 1. Open the Firefox browser and navigate to https://gresporcelan.in/. 2. Enter <script>alert('XSS');</script> in the search bar. 3. Click on the Search button. 4. Observe the alert pop-up.
Actual Results	The alert is displayed, confirming that the XSS vulnerability is present.
Expected Results	The alert should not display, and the website should prevent script execution to safeguard against XSS attacks.
Attachments	
Remarks	The presence of an XSS vulnerability poses significant security risks, including potential session hijacking, data theft, and unauthorized actions performed on behalf of users.
Defect Severity	High
Defect Priority	High
Reported By	Shrutika Parmar

Assigned To	Supriya Angne
Status	Assigned

TEST PLAN

1	Test _Identifier	Plan	TP_01
2	Introduction		The purpose of the document is to create a test plan for https://gresporcelan.in/ . By doing Manual testing of the website in the search bar
3	Test Items		Working with Search bar
4	Features to be tested		Search bar
5	Approach		<ul style="list-style-type: none"> ○ On the test object: o functional o nonfunctional ○ According to the requirements positive negative ○ By degree of preparedness - intuitive testing (adhoc)
6	Item Pass/ Fail Criteria		<ul style="list-style-type: none"> ○ All test cases with high priority are closed with the result - pass. ○ The test coverage is checked and sufficient, where the criterion of sufficiency is not less than 99% of the coverage of requirements by tests. ○ The test report was compiled and approved by the team lead and customer
7	Suspension Criteria: Resumption Criteria:		<ul style="list-style-type: none"> ○ The appearance and entering the bug-tracking system of blocking bugs. ○ Closing the blocking bug in the bug tracking system
8	Test Deliverable		Test plan, test case specification, test case , defect report
9	Test Tasks		<ul style="list-style-type: none"> ○ Writing a test plan ○ Writing test cases ○ Development of criteria for the success of testing. ○ Conducting the testing and evaluation of the results ○ Creating test reports

Department of Computer Engineering

Polytechnic

10	Environmental needs	<ul style="list-style-type: none">○ Devices: Smartphones (iOS and Android),Tablets○ Operating Systems: Android 10 and above, iOS14 and above○ Network: Stable internet connection		
11	Responsibilities	Functionality and Responsibilities		sponsible
		Arch Bar	Test Engineer 1	
12	Staffing and Training Needs	<ul style="list-style-type: none">○ To perform the tasks, you need to have thefollowing knowledge and skills:○ knowledge of practical application of the defect.○ knowledge and ability to apply in practice the basic techniques of test design.○ Knowledge of various types of testing including functional and non-functional.		
13	Schedule	The deadline for completion of all works and deliveryof the project is 25/11/2024 by 2.00pm		
14	Risks and Contingencies Possible risks during testing	o Changing the requirements for the product		
15	ApprovalsTeam	<ul style="list-style-type: none">○ Lead Test engineer 1		

Automation testing using selenium IDE

Side script

```
{
  "id": "f980fa3c-ef5b-4be9-a3f8-14a69e939a05",
  "version": "2.0",
  "name": "Xss",
  "url": "https://gresporcelan.in",
  "tests": [{
    "id": "b00656f2-ac49-4a59-8d70-c6f77d1be409",
    "name": "tc_01",
    "commands": [{
      "id": "51e41d17-4742-46df-b171-774d32fbeaad",
      "comment": "",
      "command": "open",
      "target": "/",
      "targets": [],
      "value": ""
    }], {
      "id": "3e5af1b0-6049-4865-b8ac-6246e8456424",
      "comment": "",
      "command": "setWindowSize",
      "target": "1552x832",
      "targets": [],
      "value": ""
    }, {
      "id": "eaceafde-adaa-434b-b07b-e4695339058f",
      "comment": "",
      "command": "click",
      "target": "css=.header-icon",
      "targets": [
        ["css=.header-icon", "css:finder"],
        ["xpath=//nav/div/ul/li[5]", "xpath:position"]
      ],
      "value": ""
    }, {
      "id": "71a1a4fd-f997-4fd9-839e-4b62e0a9de83",
      "comment": "",
      "command": "click",
      "target": "name=search",
      "targets": [
        ["name=search", "name"],
        ["css=.search-field", "css:finder"],

```

Department of Computer Engineering

```
["xpath=//input[@name='search']", "xpath:attributes"],
["xpath=//form[@id='search']/input", "xpath:idRelative"],
["xpath=//input", "xpath:position"]
],
"value": ""
}, {
  "id": "e37473af-ccca-47a7-802a-eb9882437716",
  "comment": "",
  "command": "type",
  "target": "name=search",
  "targets": [
    ["name=search", "name"],
    ["css=.search-field", "css:finder"],
    ["xpath=//input[@name='search']", "xpath:attributes"],
    ["xpath=//form[@id='search']/input", "xpath:idRelative"],
    ["xpath=//input", "xpath:position"]
  ],
  "value": "<script>alert(document.cookie)</script>"
}, {
  "id": "25a421c4-e049-41a3-af87-3a7aed60eadc",
  "comment": "",
  "command": "sendKeys",
  "target": "name=search",
  "targets": [
    ["name=search", "name"],
    ["css=.search-field", "css:finder"],
    ["xpath=//input[@name='search']", "xpath:attributes"],
    ["xpath=//form[@id='search']/input", "xpath:idRelative"],
    ["xpath=//input", "xpath:position"]
  ],
  "value": "${KEY_ENTER}"
}]
}],
"suites": [{
  "id": "db64eb02-955d-435f-ae59-ad599d8f989d",
  "name": "Default Suite",
  "persistSession": false,
  "parallel": false,
  "timeout": 300,
  "tests": ["b00656f2-ac49-4a59-8d70-c6f77d1be409"]
}],
"urls": ["https://gresporcelan.in/"],
"plugins": []
}
```

Department of Computer Engineering

7.0 Skill Developed/Learning out of this Micro Project

- 7.1.1 **Understanding of XSS Vulnerabilities:** Gained a comprehensive understanding of Cross- Site Scripting (XSS) vulnerabilities, including types (reflected, stored, and DOM-based) and their potential impact on web applications.
- 7.1.2 **Manual Testing Techniques:** Enhanced skills in manual testing methodologies, particularly in identifying and exploiting XSS vulnerabilities using various payloads and techniques.
- 7.1.3 **Test Case Development:** Developed the ability to create effective test cases, including defining clear steps, expected results, and actual results, tailored for security testing scenarios.
- 7.1.4 **Automation testing:** Using Selenium IDE.
- 7.1.5 **Defect Reporting:** Learned to document defects comprehensively, including details such as environment, severity, reproduction steps, and proposed solutions, which is crucial for communicating issues to development teams.

Annexure-IIA

Name of Student:

Enrollment No:

Name of Programmed:

Semester:

Course Title:

Code:

Title of the Micro Project:

Course Outcomes Achieved:

Micro Project Evaluation Sheet

Process Assessment		Product Assessment		Total Marks 10
Part-A Project Proposal (Mark-2)	Project Methodology (Mark-2)	Part-B Project Report/ Working Model (Marks-2)	Individual Presentation/ Viva (Marks-4)	

Note: Every course teacher is expected to assign marks for group evolution in first 3 columns and individual in 4th columns for each group of students as per rubrics.

Comments/Suggestions about team work/leadership/inter-personal communication (if any)

--

-

Any other Comments:

-

-

Name and Designation of Faculty Members

