Ganpat University
॥ विद्यया समाजोत्कर्षः ॥

Department of Computer Science

A

Project Report

On

Vulnerability Assessment and Penetration Testing

Submitted By

Group No: 04

AJEYPALSINH JADEJA - 23084321007
RIYAKUMARI PATEL - 23084321017

M.Sc. IT (Cyber Security) Semester-III

Guided By

Internal: Dr. Amit Suthar

Submitted to

Department of Computer Science,

Ganpat University,

Ganpat Vidyanagar - 384012

A.Y. 2024-2025

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥

Department of Computer Science

Date: 06/12/2024

C E R T I F I C A T E

TO WHOM SO EVER IT MAY CONCERN

This is to certify that the following students of M.Sc. IT (Cyber Security) Semester-III have completed their project work titled **"Vulnerability Assessment and Penetration Testing"** satisfactorily and fulfil the requirement of M.Sc. IT (Cyber Security) Semester-III, Department of Computer Science, Ganpat University in the Academic Year, 2024-2025.

Sr. No.	Student Name	Enrollment No.
1.	AJEYPALSINH JADEJA	23084321007
2.	RIYAKUMARI PATEL	23084321017

Internal Guide	Project Coordinator	Program Coordinator	Dean
Dr. Amit Suthar	Dr. Amit Suthar	Dr. Ajay Patel	Dr. Nirbhay Chaubey

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to all those who have contributed to the successful completion of this project. Their guidance, encouragement, and support have been invaluable throughout this journey.

First and foremost, we would like to thank **Dr. Amit Suthar**, our project mentor, for their insightful guidance, constructive feedback, and unwavering support. Their expertise and advice have been instrumental in shaping this project and overcoming challenges.

We are also sincerely grateful to our college, **Ganpat University**, for providing us with the necessary resources and a conducive environment to carry out this work. Special thanks to the **Department of Computer Science** for fostering our learning and growth.

We extend our heartfelt appreciation to our family and friends for their constant motivation, patience, and encouragement during this endeavour. Their support has been a pillar of strength throughout this process.

Lastly, we are thankful to all the sources and individuals whose work, ideas, or resources have inspired and enriched this project.

Thank you all for your invaluable contributions.

With Regards,

AJEYPALSINH JADEJA

RIYAKUMARI PATEL

PREFACE

“Vulnerability Assessment and Penetration Testing” aims to address performed to identify and exploit vulnerabilities in an application, and the way it interacts and transfers data with the backend systems while applying the concepts and skills learned during our studies.

The project reflects our efforts to gain a deeper understanding of “**Vulnerability Assessment and Penetration Testing”** and showcases the practical application of theoretical knowledge.

We had tried to approach this work with dedication, ensuring the use of relevant tools, techniques, and resources to achieve the desired outcomes. We hope this work contributes to the academic discussion on the topic and provides valuable insights.

CONTENT		
SR. NO.	**PARTICULARS**	**PAGE NO.**

1.PROJECT PROFILE

PROJECT TITLE: -	**Vulnerability Assessment and Penetration Testing**
OBJECTIVE: -	To identify and address security weaknesses in a system, network, or application
OPERATING SYSTEM: -	Kali Linux (2024) Window (windows 10)
TOOLS: -	Burp-suite (2022.2.4) SQL Map Wappalyzer Wpscan Subzy
PERFORM BY: -	AJEYPALSINH JADEJA (E. NO – 23084321007) RIYAKUMARI PATEL (E. NO – 23084321017)
INTERNAL GUIDE: -	Dr. Amit Suthar
GROUP NO: -	04
SUBMITTED TO: -	Department of Computer Science, Ganpat University, Kherva

2.0 INTRODUCTION

- ✓ VAPT stands for Vulnerability Assessment and Penetration Testing. VAPT is a methodological approach to improving your organization's security posture by identifying, prioritizing, and mitigating vulnerabilities in its infrastructure. It also helps you stay compliant with various industry standards throughout the year.
- ✓ VAPT is the process of finding and exploiting all possible vulnerabilities in your infrastructure, with the primary goal of mitigating them. VAPT is done by security experts who are experts in offensive exploitation. Simply put, VAPT is a proactive "hacking" activity in which you hack your infrastructure before hackers come looking for loopholes.

2.1 OVERVIEW

- ✓ The adoption of technology to gain speedy growth of IoT, mobile applications, has made the networks more vulnerable than ever. VAPT methods are designed to support users to authenticate their enterprise-level security against the real-world threat, recognize the risks of the system and the network and know the consequences of these flaws.
- ✓ Every industry spends a fair amount of share in their security systems. Taking charge and confirming the reliability and robustness of the processes is highly important. VAPT services help improve networks and immune the security system and guard it against hackers.

2.2 BACKGROUND AND MOTIVATION

- ✓ The evolving tools, tactics and procedures used by cybercriminals to breach networks means that it is important to regularly test your organization's cyber security. VAPT helps to protect your organization by providing visibility of security weaknesses and guidance to address them.
- ✓ VAPT is increasingly important for organizations wanting to achieve compliance with standards including the GDPR, ISO 27001 and PCI DSS.
- ✓ Vulnerabilities that cyber attackers might exploit, allowing you to patch them before a breach occurs.

2.3 OBJECTIVE

- ✓ The objective of performing a Vulnerability Assessment is to create an overview of the security risks to a network and then use that overview as a guideline to resolve those threats.
- ✓ Performing regular assessments and routinely resolving all security risks provides a baseline security for the network.
- ✓ Ultimately, the goal is to identify security weaknesses in a network, machine, or piece of software.
- ✓ Once caught, the people maintaining the systems or software can eliminate or reduce the weaknesses before hostile parties discover them.

2.4 METHODOLOGY

3. HARDWARE AND SOFTWARE REQUIREMENT

HARDWARE REQUIREMENT

Processor	1.6 GHz to 2.5 GHz
RAM	Recommended 8 GB
Free Hard disk space	Minimum 256 GB

SOFTWARE REQUIREMENT

Operating System	Linux Window
Tools	Burp suite, Sql map, Nmap, Wappalyzer, Wpscan, Subzy,

4. FUNCTIONAL SPECIFICATION

Sr. No.	Vulnerability	Description
01.	SQL INJECTION	It allows an attacker to interfere with the queries that an application makes to its database.
02.	LOGIN BYPASS	It can allow attackers to gain unauthorized access to sensitive information.
03.	AUTHENTICATION BYPASS	It occurs when an attacker bypasses the authentication mechanisms of a device to gain unauthorized access.
04.	CROSS SITE REQUEST FORGERY (CSRF)	The end-user performs unwanted actions within a web application that has already granted them authentication.
05.	REFLECTED XSS	It occur when a malicious script is reflected off of a web application to the victim's browser.
06.	NO RATE LIMIT ON FORGOT PASSWORD PAGE	This can be used to send an unlimited number of forgot password requests to any random email.
07.	CLEAR TEXT PASSWORD SUBMISSION	Password is transmitted in a non-encrypted form during authentication processes.
08.	CLICK JACKING	It is attack that tricks a user into clicking a webpage element which is invisible.
09.	SERVER VERSION DISCLOSURE	The Server header describes the server application that handled the request.
10.	HTML INJECTION	It allows an attacker to inject HTML code into web pages that are viewed by other users.

5. TOOLS DESCRIPTION

1) Burp Suite: -

- Burp Suite is an integrated platform designed for web application security testing. It comprises various tools that work seamlessly together to facilitate the entire penetration testing process. Key components include the Proxy, Scanner, Intruder, and Repeater.
- **How is it Used in Web Pen testing?**
  - ✓ **Proxy:** Used for intercepting and modifying HTTP/S traffic between the browser and the target web application, allowing testers to analyze and manipulate requests.
  - ✓ **Scanner:** Automates the identification of vulnerabilities such as SQL injection and XSS, providing detailed insights into potential security issues with comprehensive scanning capabilities.
  - ✓ **Intruder:** Enables automated attacks with customizable payloads, valuable fortesting input validation and identifying potential injection points, aiding in the discovery of application vulnerabilities.
  - ✓ **Repeater:** Allows manual exploration and modification of individual requests, facilitating in-depth analysis of application behaviour and responses.

2) SQL Map: -

- SQL Map is an open-source tool specifically designed for the automated detection and exploitation of SQL injection vulnerabilities in web applications and databases. Its versatility extends to supporting a wide range of database management systems (DBMS) and providing powerful mechanisms for identifying and exploiting SQLinjection flaws.
- **How is it Used in Web Pen testing?**
  - ✓ **Detection:** SQL Map automates the identification of SQL injection vulnerabilities by thoroughly analyzing web application parameters, pinpointing potential entry points for exploitation.
  - ✓ **Exploitation:** The tool offers robust options for exploiting identified vulnerabilities, including the extraction of sensitive data from databases. This feature is crucial for simulating real-world scenarios and assessing the potential impact of SQL injection flaws.

- ✓ **Database Fingerprinting:** SQL Map excels in determining the type and version of the underlying database, providing testers with valuable insights for crafting targeted and effective attacks.

3) Nmap: -

- Nmap, short for Network Mapper, stands out as a versatile and powerful network scanning tool, primarily used for discovering hosts, open ports, and services within a network. Its reputation is built on flexibility, speed, and extensive capabilities in network reconnaissance.
- **How is it Used in Web Pen testing?**
  - ✓ **Host Discovery:** Nmap excels in identifying live hosts on a network, allowing penetration testers to delineate the scope of the target environment.
  - ✓ **Port Scanning:** It determines open ports and services on target systems, providing critical insights into the potential entry points for further investigation.
  - ✓ **Version Detection:** Nmap goes beyond port scanning by retrieving detailed information about the versions of services running on open ports, aiding invulnerability assessments.
  - ✓ **Scripting Engine:** Nmap features a scripting engine that permits the execution of custom scripts, enabling advanced testing and automation for specific testing scenarios.

4) Wappalyzer: -

- Wappalyzer is a browser extension designed to identify the technologies used by websites. It reveals insights into web servers, content management systems, and frameworks, aiding penetration testers during the reconnaissance phase.
- **How is it Used in Web Pen testing?**
  - ✓ **Real-time Analysis:** Wappalyzer provides real-time analysis of visited sites, identifying the underlying technologies and frameworks in use.
  - ✓ **Technology Stack Identification:** The extension reveals the web application's technology stack, aiding in targeted testing based on identified technologies.
  - ✓ **Browser Integration:** Wappalyzer operates seamlessly within web browsers, offering ease of use and immediate visibility into the technological aspects of web application.

5) wpscan: -

- Wpscan is a WordPress security scanner used to test WordPress installations and WordPress-powered websites. This is a command line tool used in Kali Linux. This tool can be used to find any vulnerable plugins, themes, or backups running on the site.
- **Used in Web Pen testing: -**
  - Detect the WordPress version
  - Detect sensitive files
  - Detect enabled features
  - Enumerate themes and plugins
  - Scan for usernames
  - Perform WordPress login brute force
  - Find default WordPress directories

6) subzy: -

- Subzy is the tool that identifies or checks the subdomain takeover on the target domain or multiple subdomains. This automated scanner can help you in bug bounty programs to find Subdomain Takeover bugs in the target website.
- **Used in Web Pen testing: -**
  - ✓ **Discovery of Subdomains:** The first step in subdomain takeover testing is discovering subdomains that belong to a target domain. This is typically done using tools like Sublist3r and Amass. Once the subdomains are discovered, Subzy checks if any of them are vulnerable to takeover by verifying the status of their associated external services.
  - ✓ **Subzy's Workflow's Verification:** Subzy scans subdomains to identify DNS records that point to external services (CNAME records to services like Heroku, AWS, GitHub, etc.). Service Check: It checks if the external service is still active or if the resource (like a Heroku app or S3 bucket) no longer exists. Vulnerability Detection: If Subzy finds that the subdomain is pointing to a service that is no longer in use or has been deleted, it marks the subdomain as vulnerable to takeover.
  - ✓ **Exploitation:** If Subzy detects a vulnerable subdomain, a pen tester can attempt to claim the service (e.g., by creating a new GitHub Pages site or S3 bucket) and take control of the subdomain. This allows the attacker to host content under the victim's domain, which can be used for phishing, malicious redirections, or other attacks.

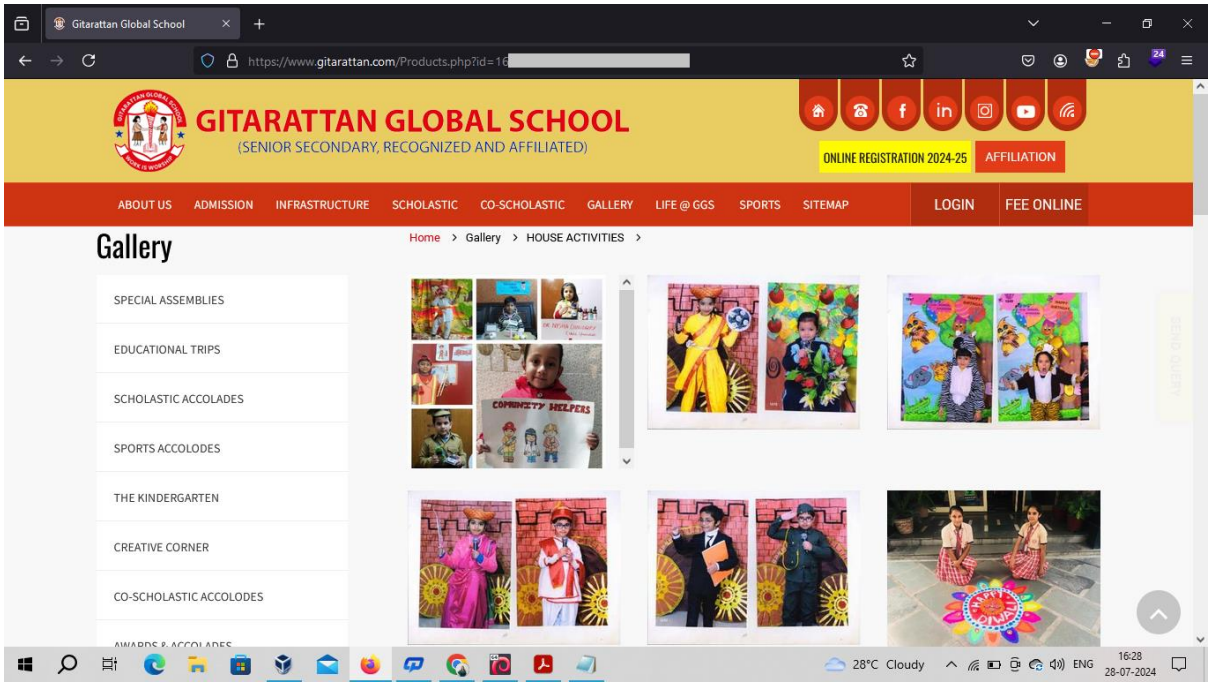
- ✓ **Reporting:** Once a vulnerable subdomain is identified, the pen tester would report the issue to the affected organization, providing proof of concept and details on how the takeover can be prevented (e.g., by removing the DNS entry or configuring the external service properly).

PROOF OF CONCEPTS

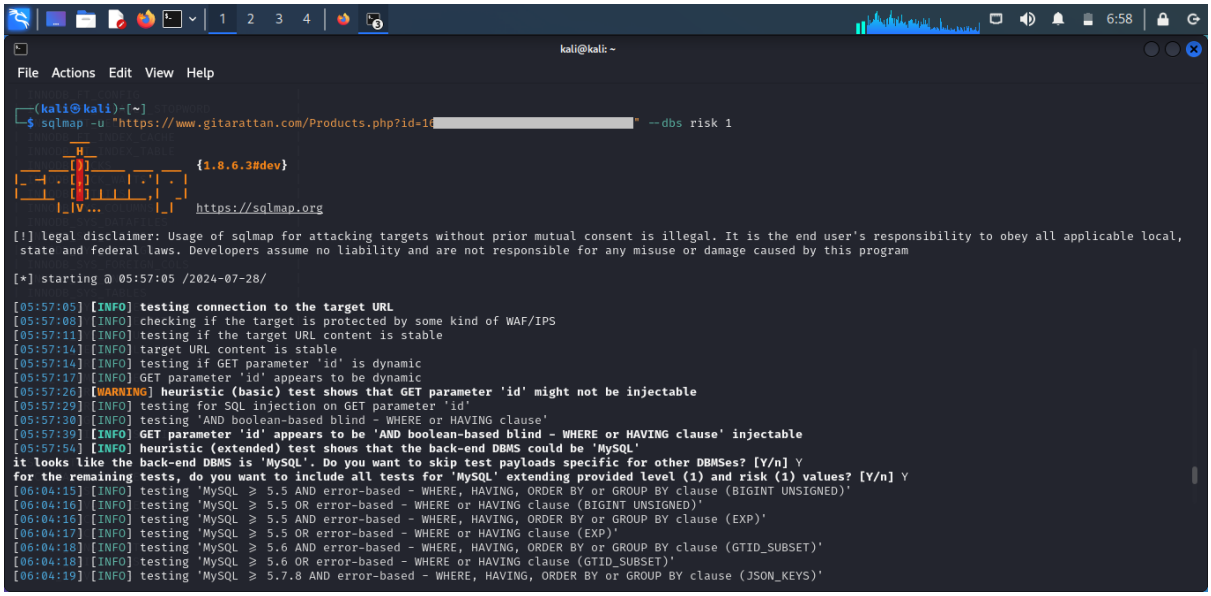
Sr. No.	Vulnerability Name	Severity
01.	SQL INJECTION	**CRITICAL**
02.	LOGIN BYPASS	**HIGH**
03.	AUTHENTICATION BYPASS	**HIGH**
04.	CROSS SITE REQUEST FORGERY (CSRF)	**HIGH**
05.	REFLECTED XSS	**HIGH**
06.	NO RATE LIMIT ON FORGOT PASSWORD PAGE	**MEDIUM**
07.	CLEAR TEXT PASSWORD SUBMISSION	**MEDIUM**
08.	CLICK JACKING	**MEDIUM**
09.	SERVER VERSION DISCLOSURE	**LOW**
10.	HTML INJECTION	**LOW**

01.SQL INJECTION	
Description	
It allows an attacker to interfere with the queries that an application makes to its database.	
Affected Resource / Parameter	**Severity**
https://gitarattan.com	**CRITICAL**
Impact / Consequences	
There is always risk that the user's privacy will be compromised. ✓ A successful attack may result in the unauthorized viewing of user lists, the deletion of entire tables and in certain cases, the attacker gaining administrative rights to a database ✓ So, this vulnerability is very critical!!	
Recommendations	
The only sure way to prevent SQL Injection attacks is input validation and parameterized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms. They must remove potential malicious code elements such as single quotes. It is also a good idea to turn off the visibility of database errors on your production sites. Database errors can be used with SQL Injection to gain information about your database.	
Tools Used	**References**
Sqlmap	https://sqlmap.org/
CWE	**OWASP Top 10**
89	A03:2021 – Injection
Proof of Vulnerability	

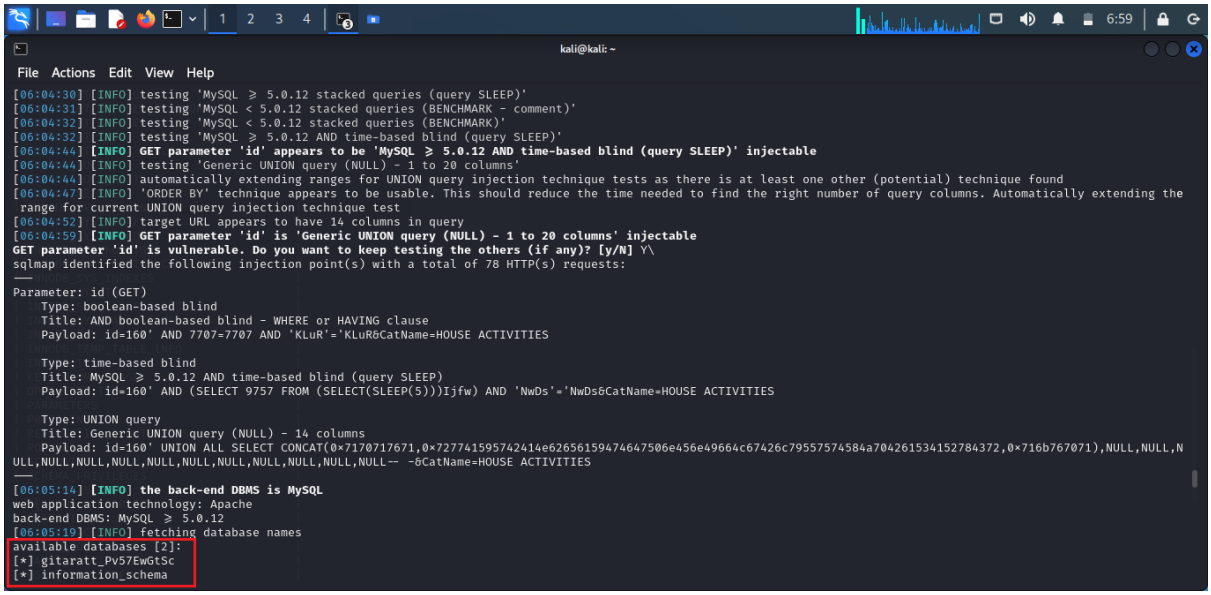
- Step 1: - visit website.



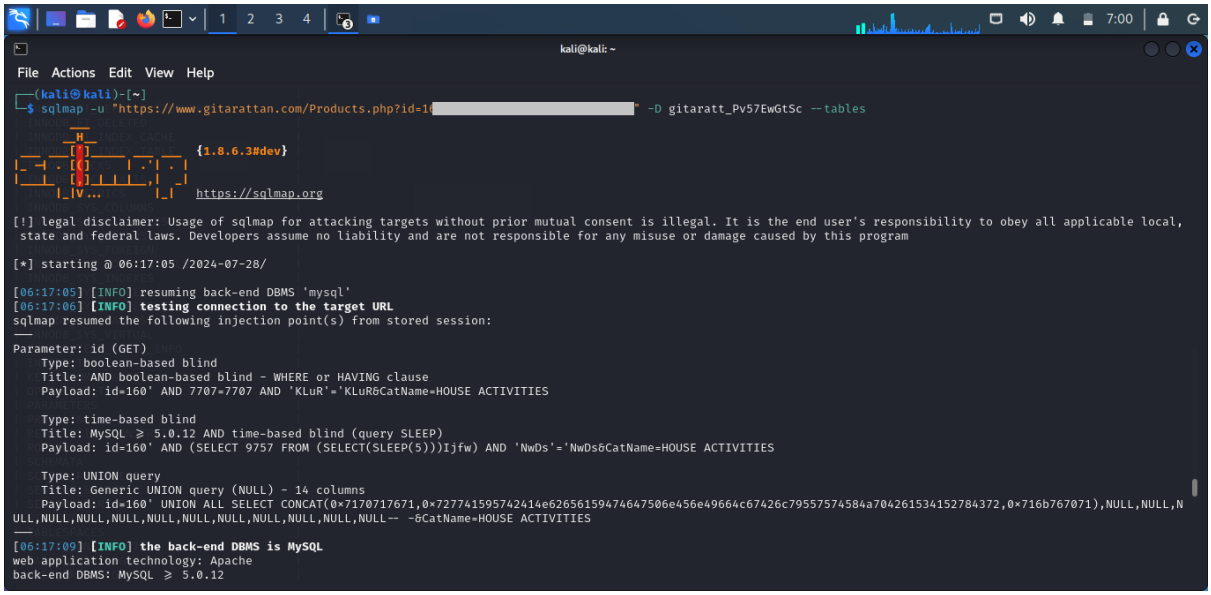
- Step 2: - Use SQL map tool in kali Linux. put -u before the URL then provide the URL. Use –dbs for getting the database control and use –risk 1 for increasing the risk level.



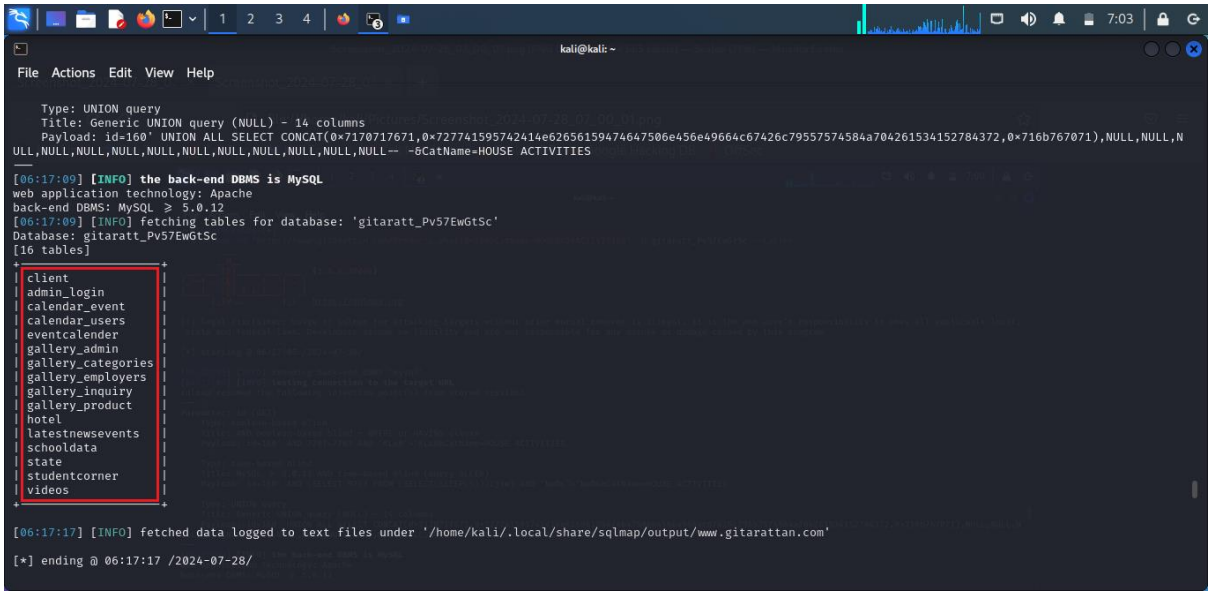
- Step 3: - Then we get the information that the backend database is MySQL.



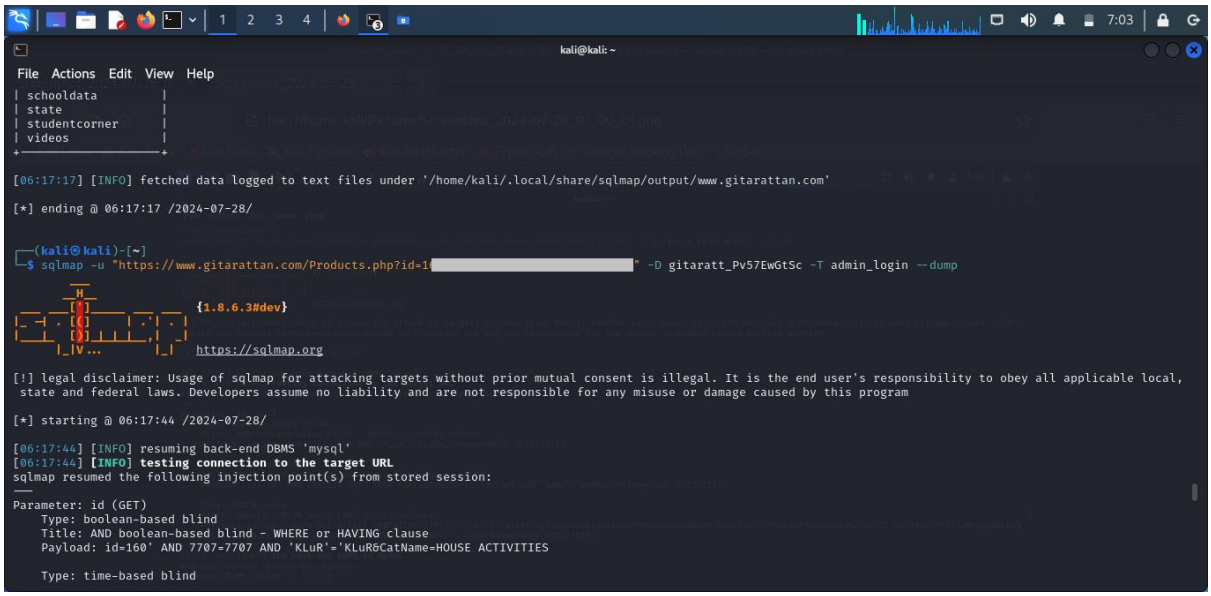
- Step 4: - After getting the database name use --tables command for retrieve all the tables name.



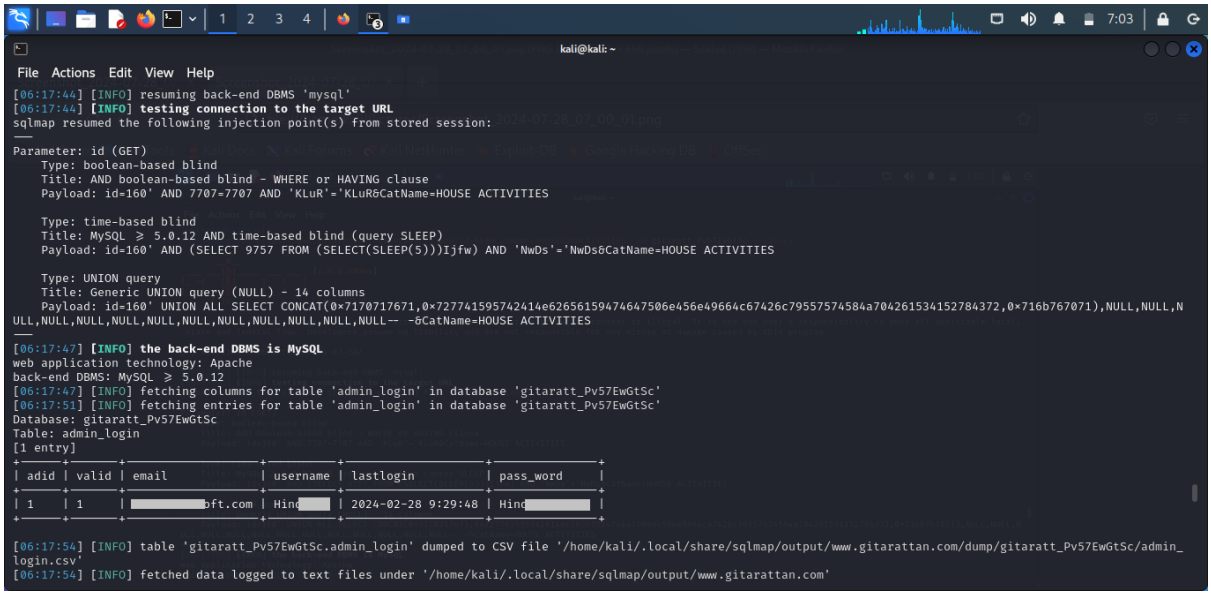
- Step 5: - Here you can see we are retrieve all the tables name.



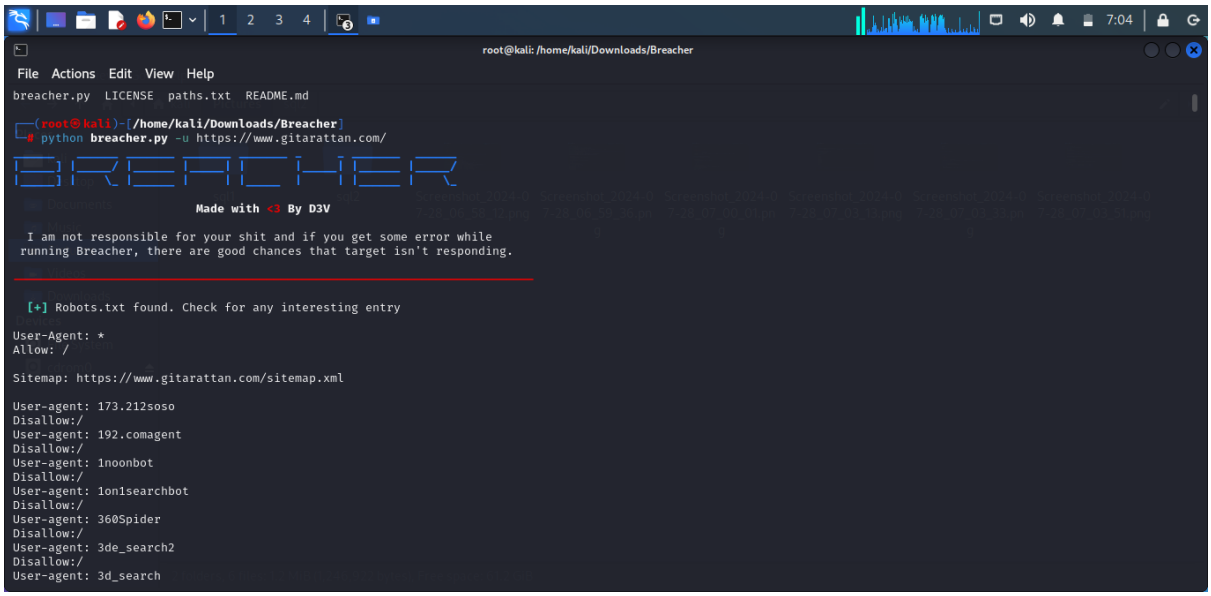
- Step 6: - After getting the tables we have to use –dump to get data.



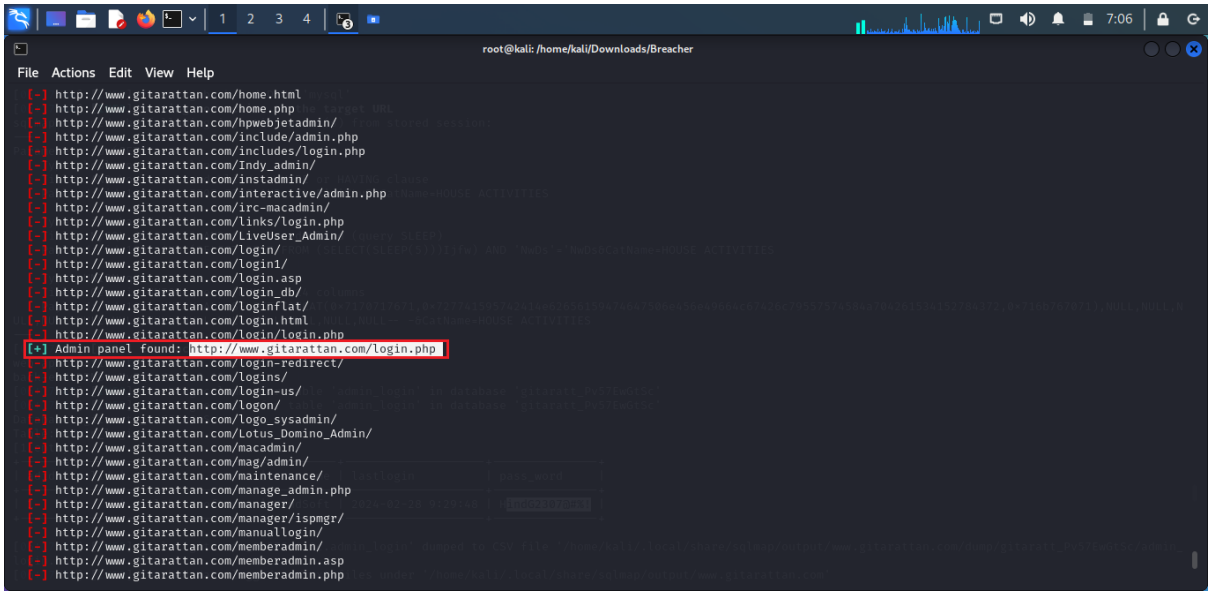
- Step 7: - As a result, we found the data of those table. Here you can see username and password also.



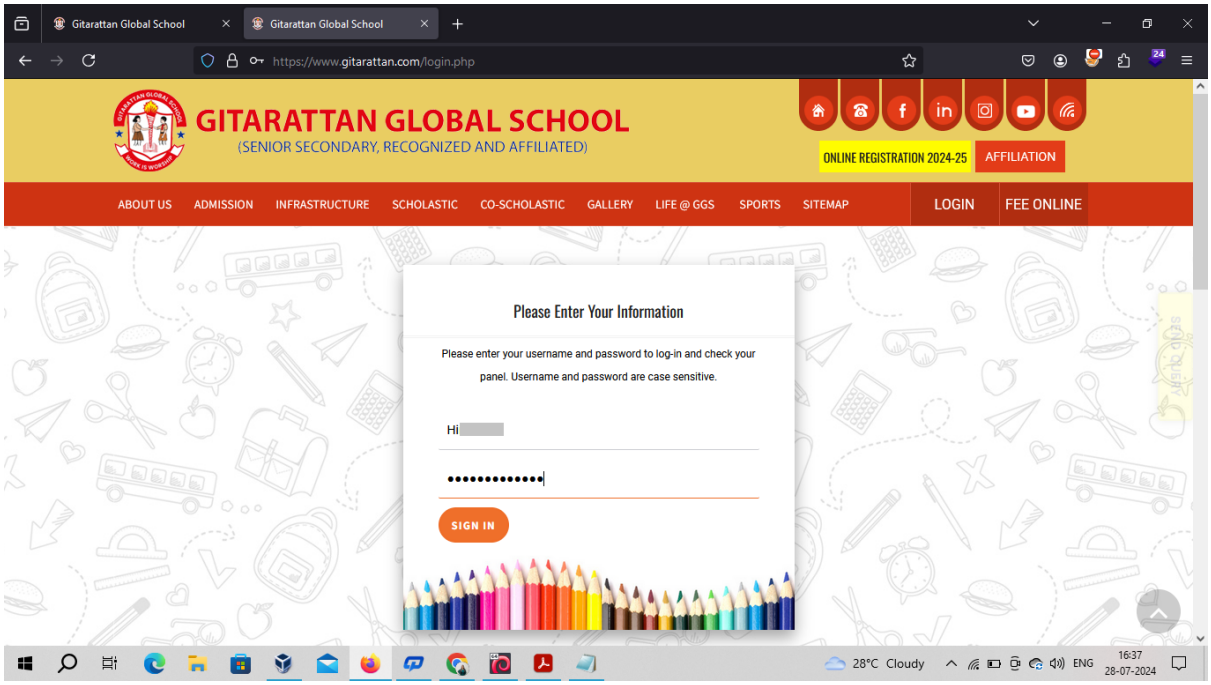
- Step 8: - Here we are using Breacher tool. Breacher is a python script to find admin login pages.



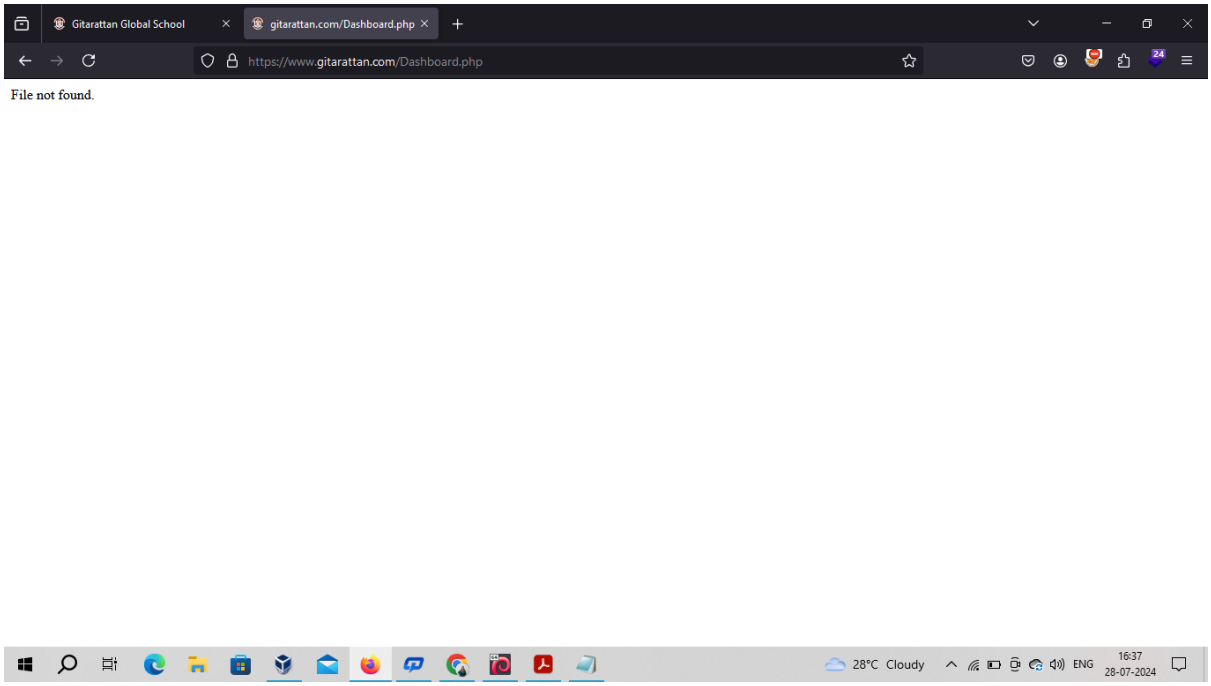
- Step 9: - Here you can see the admin login URL.



- Step 10: - Here we go to that admin log in page and write that username and password which we are found. And click on sign in button.



- Step 11: - Here you can see we are successfully log in the admin page. But in the dashboard, there no any file available.

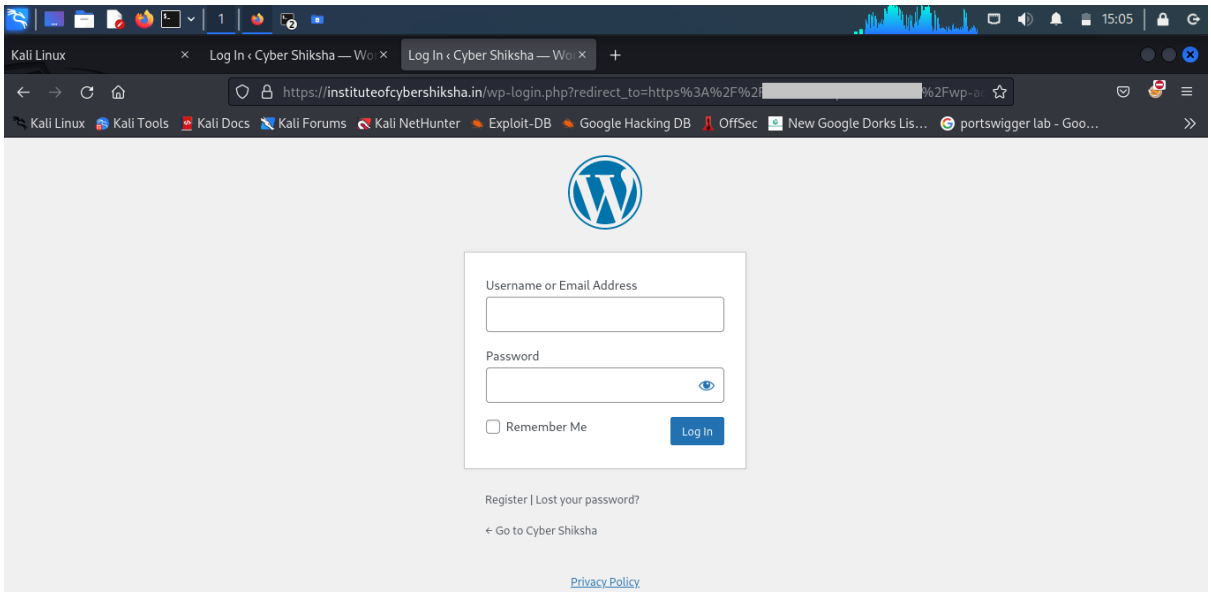


Observation
It was observed that a SQL injection attack consists of the insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands. It is observed that the username attribute is vulnerable to SQL injection.

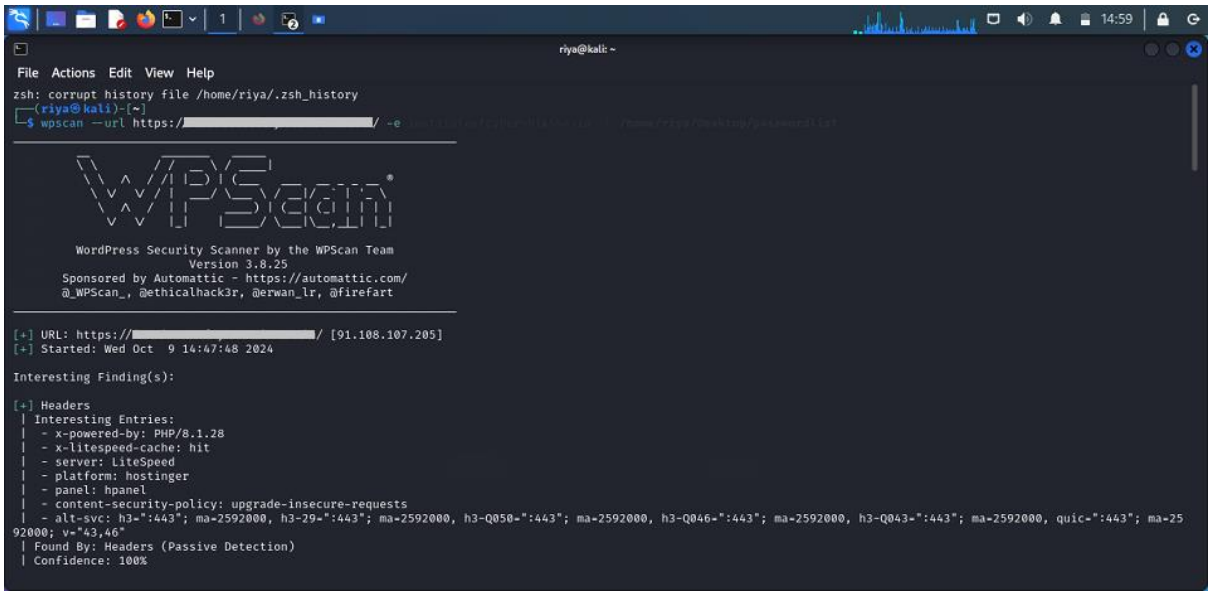
02.LOGIN BYPASS	
Description	
It can allow attackers to gain unauthorized access to sensitive information.	
Affected Resource / Parameter	**Severity**
https://instituteofcybershiksha.in/	**HIGH**
Impact / Consequences	
✓ Unauthorized access to any User account.	
Recommendations	
✓ Input Validation and Sanitization ✓ Strengthen CAPTCHA and Anti-Bot Measures ✓ Secure Forgot Password and Account Recovery Processes ✓ Encrypt and Hash Credentials ✓ Implement Strong Authentication Mechanisms	
Tools Used	**References**
Burp-suit	https://portswigger.net/burp/communitydownload
CWE	**OWASP Top 10**
307	WSTG-ATHN-04
Proof of Vulnerability	

- Step 1: - Visit the Website.

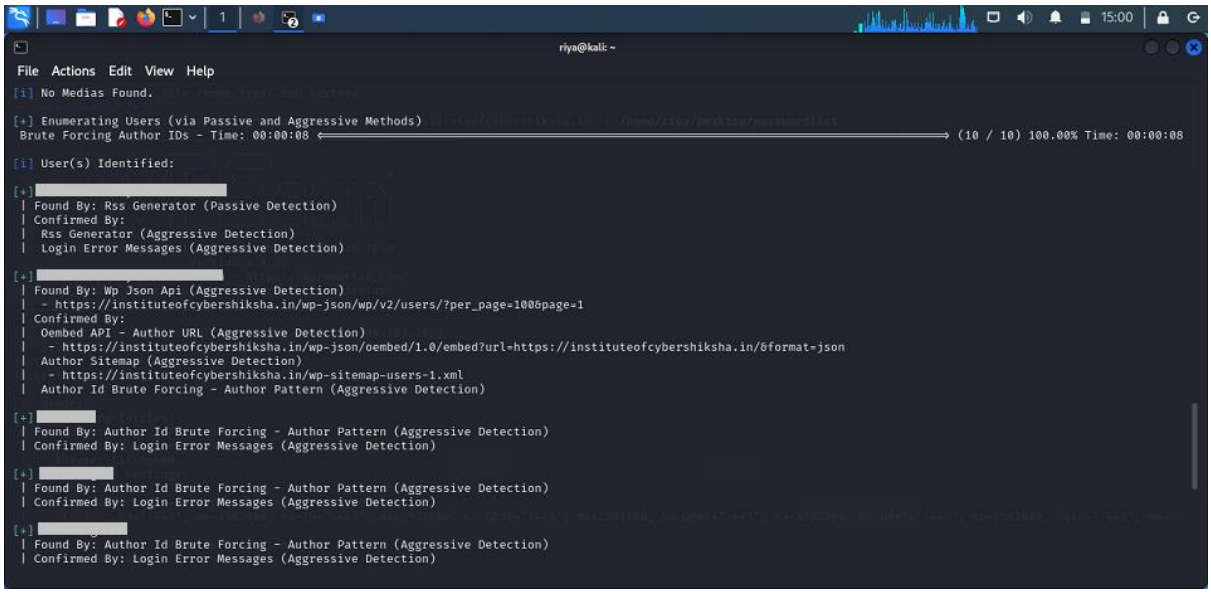
- Step 2: - Here you can see the login page.

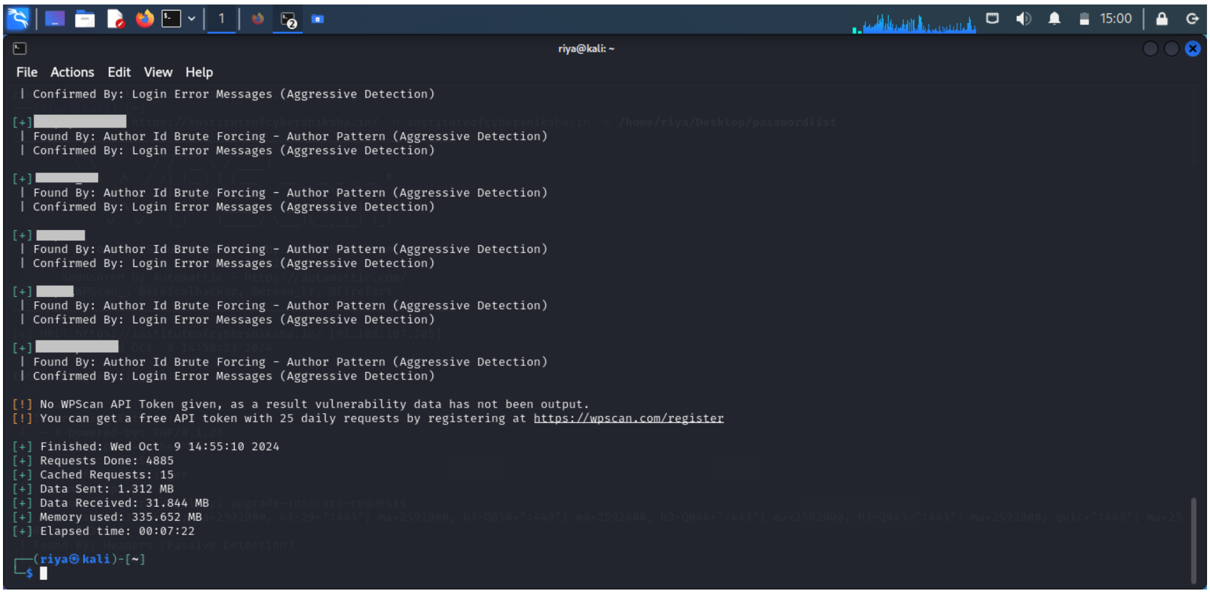


- Step 3: - Here I used wpscan tool for finding the username and password. Here I used command **wpscan –url <URL> -e** for fetching Username.

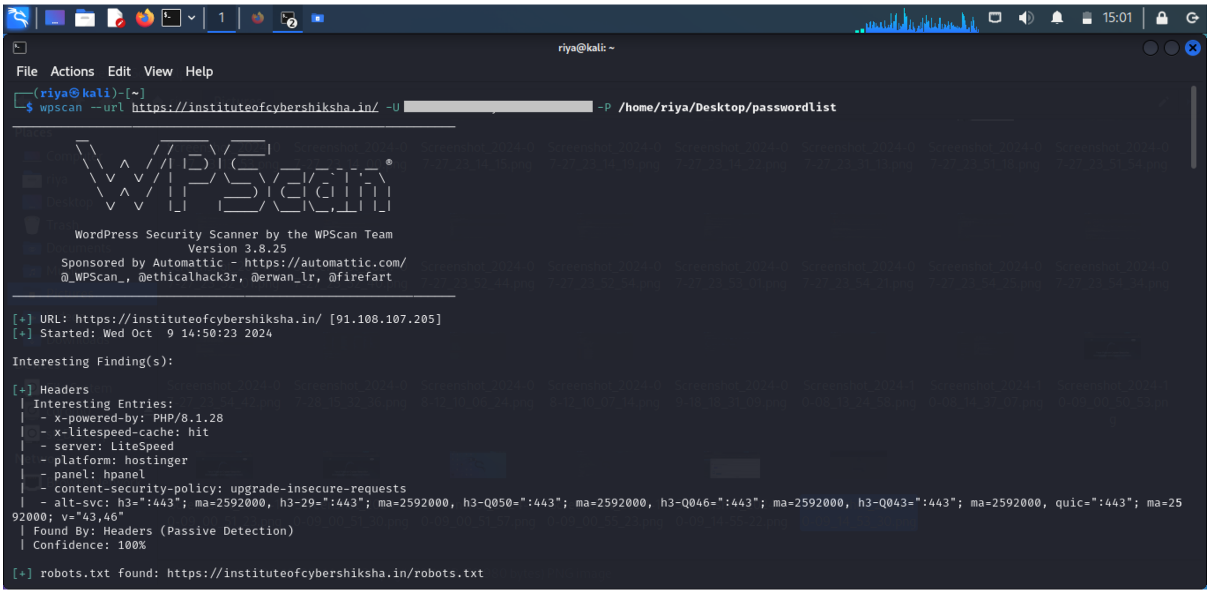


- Step 4: - Here I found the all Username. Now I will try to fetch Correct Username and Password.

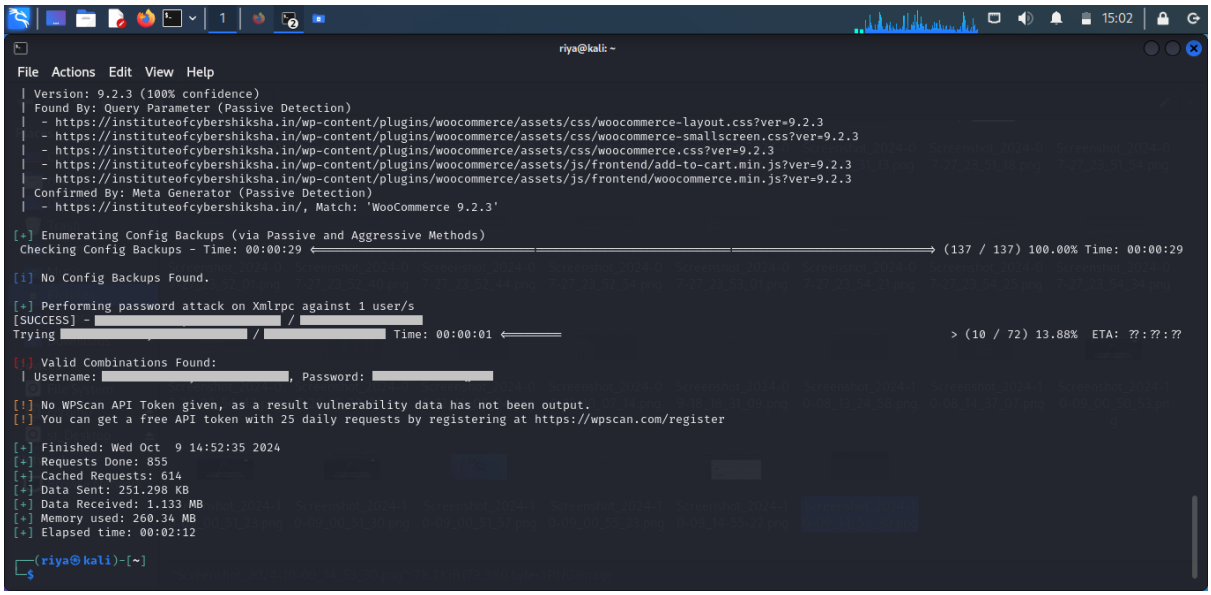




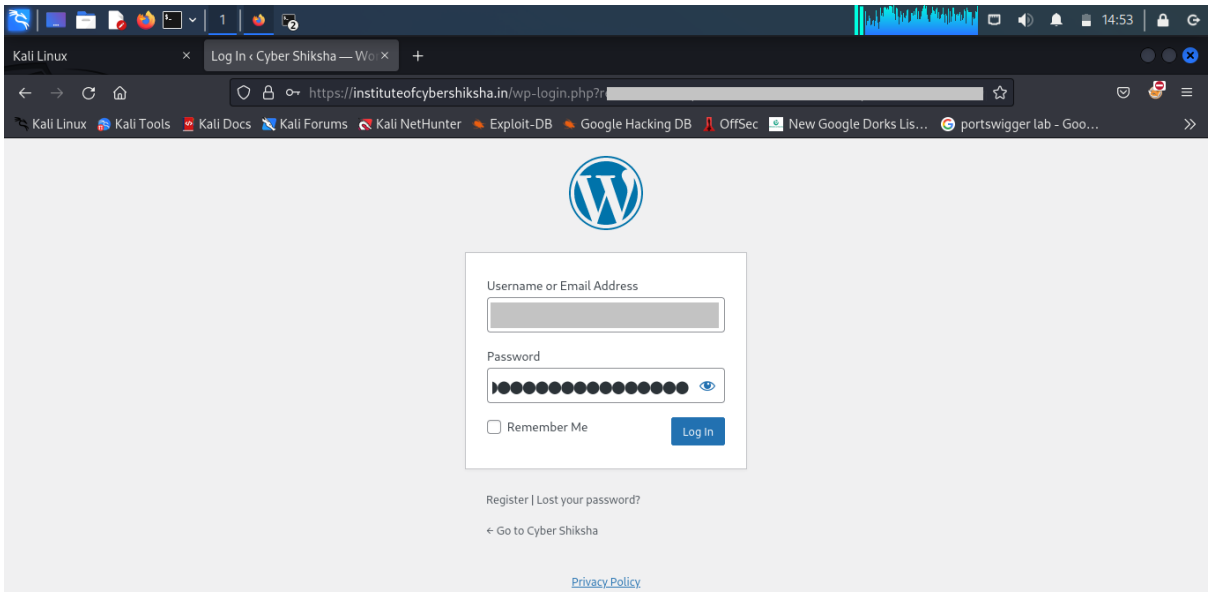
- Step 5: - Here I used the username and also used the Password list for fetch the password. For fetch password I write command **wpcan –url <URL> -U <username> -P <path of passwordlist>**.



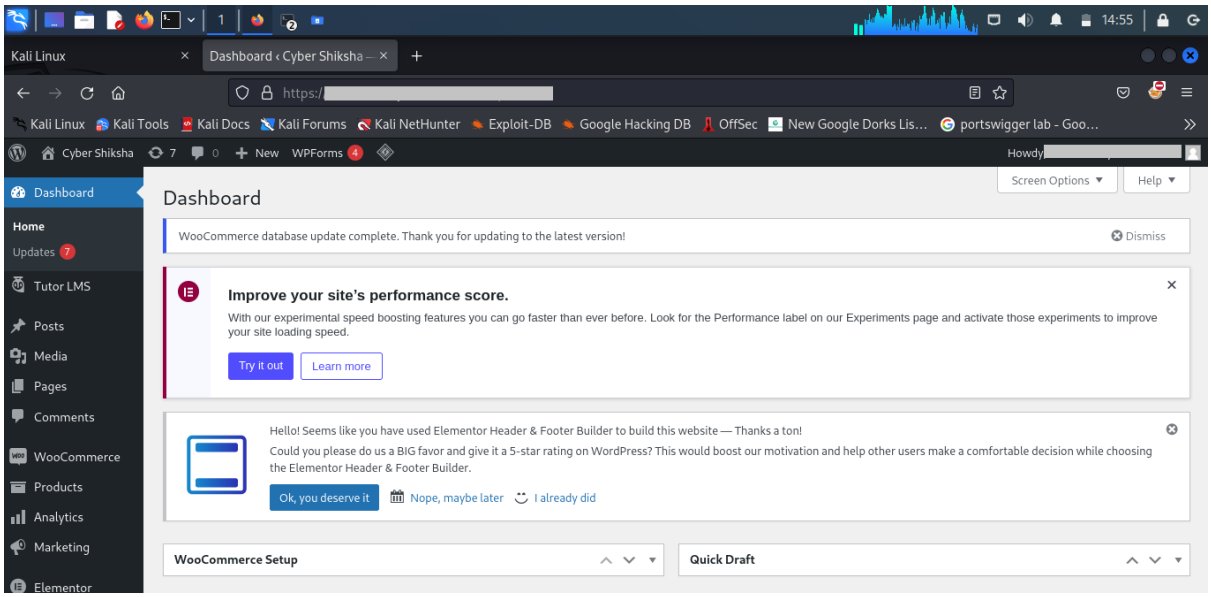
- Step 6: - Here you can see the successfully found the Username and Password.

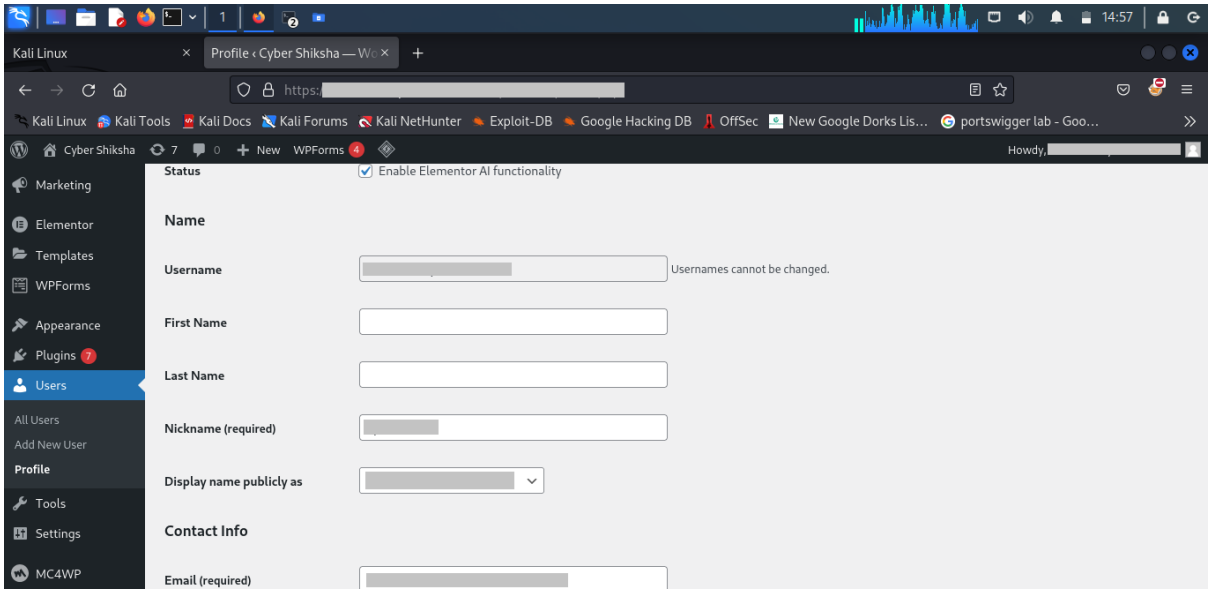


- Step 7: - In login page I will try that username and password which I found.



- Step 8: - Here you can see the successfully log in using that username and password. In this page I will modify the username and password. And anything I will modify.

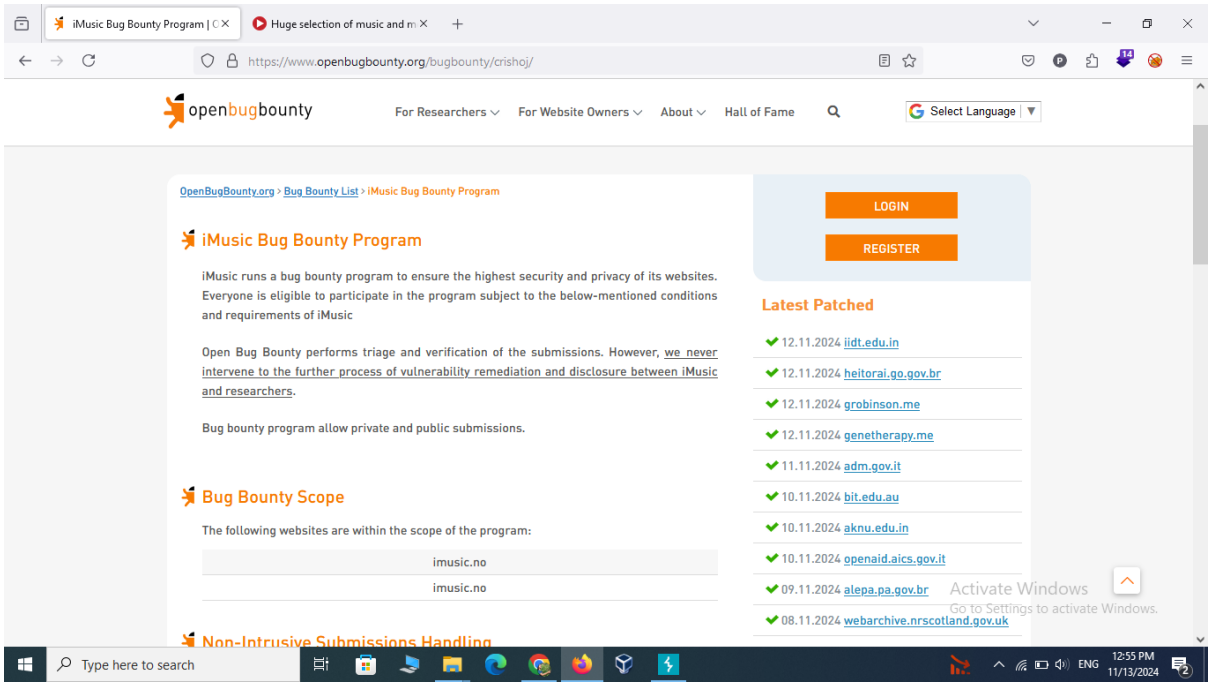




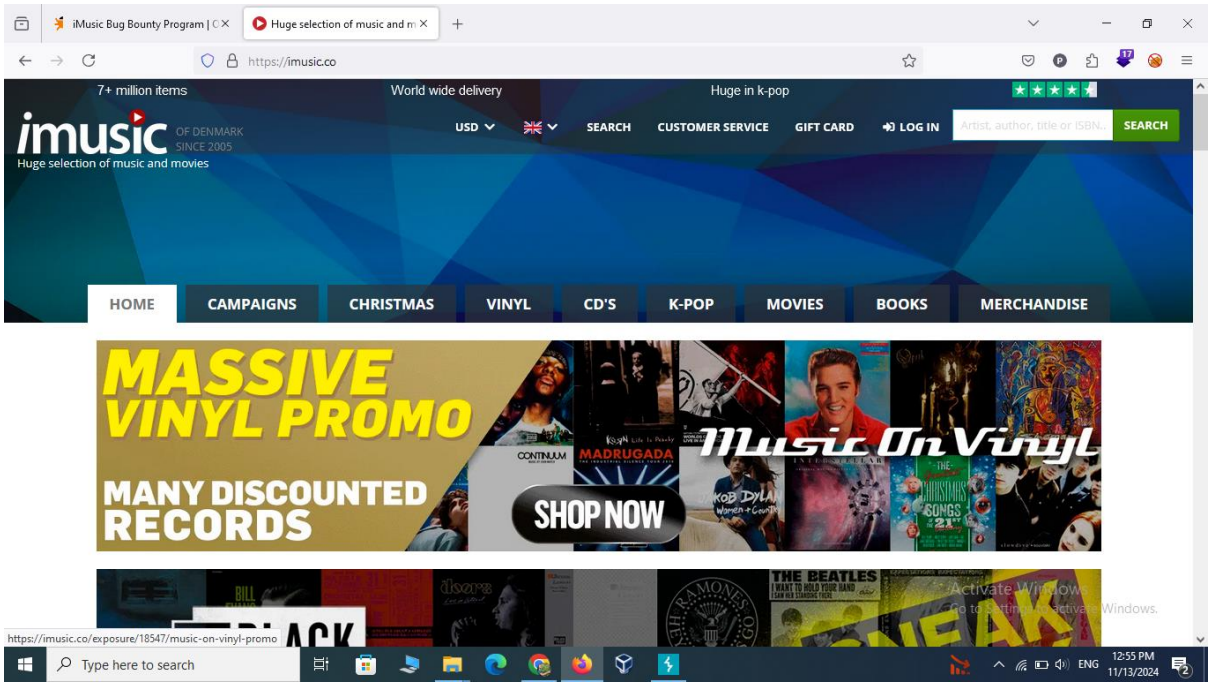
Observation
High rates of failed login attempts might indicate brute force or credential stuffing. Login forms revealing specifics like "username not found" or "password incorrect" can aid attackers. Checking if MFA can be bypassed by directly accessing resources without verification.

03. AUTHENTICATION BYPASS	
Description	
It occurs when an attacker bypasses the authentication mechanisms of a device to gain unauthorized access.	
Affected Resource / Parameter	**Severity**
https://imusic.no/	**HIGH**
Impact / Consequences	
The consequences of successful authentication bypass can be severe: 1. Unauthorized access to sensitive data 2. Compromised user accounts 3. Financial losses 4. Damage to reputation 5. Compliance violations	
Recommendations	
1. Implement Multi-Factor Authentication (MFA) 2. Use Strong, Secure Authentication Protocols 3. Use Input Validation 4. Enforce complex password requirements and regular password changes 5. Use secure, randomly generated session tokens and implement proper session timeout mechanisms	
Tools Used	**References**
Burp-suite	https://portswigger.net/burp
CWE	**OWASP Top 10**
288	A07:2021 – Identification and Authentication Failures
Proof of Vulnerability	

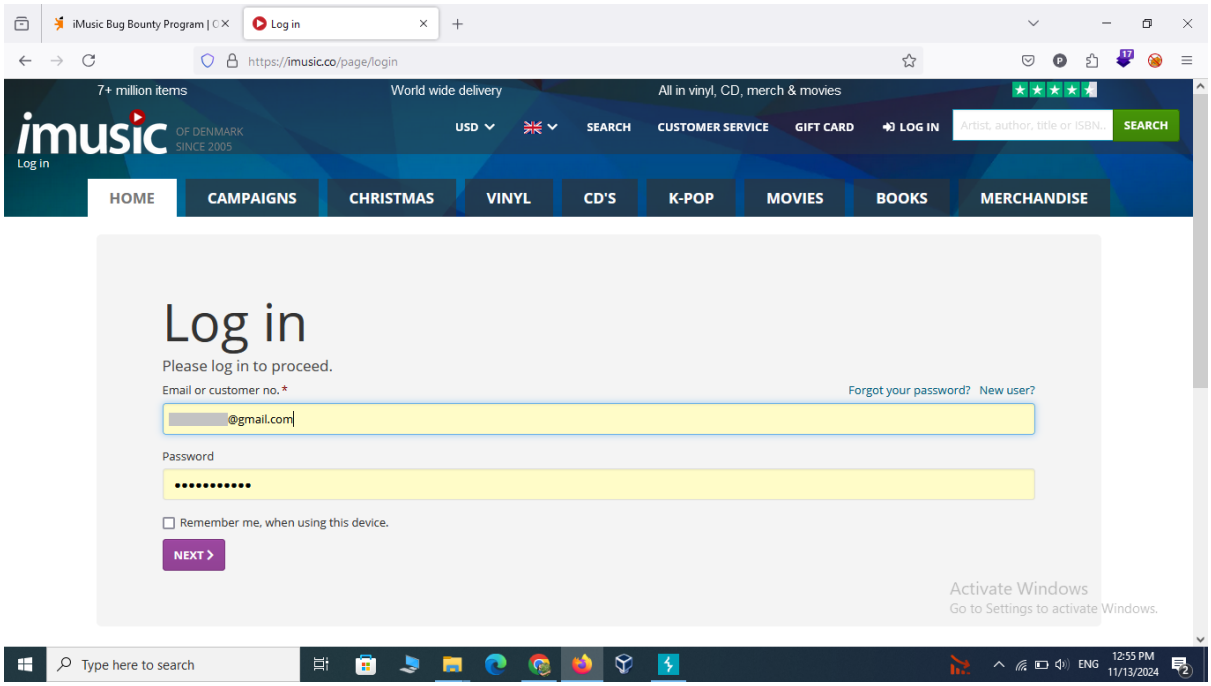
→ Here you can see the website of openbugbounty that is use for vulnerability testing.



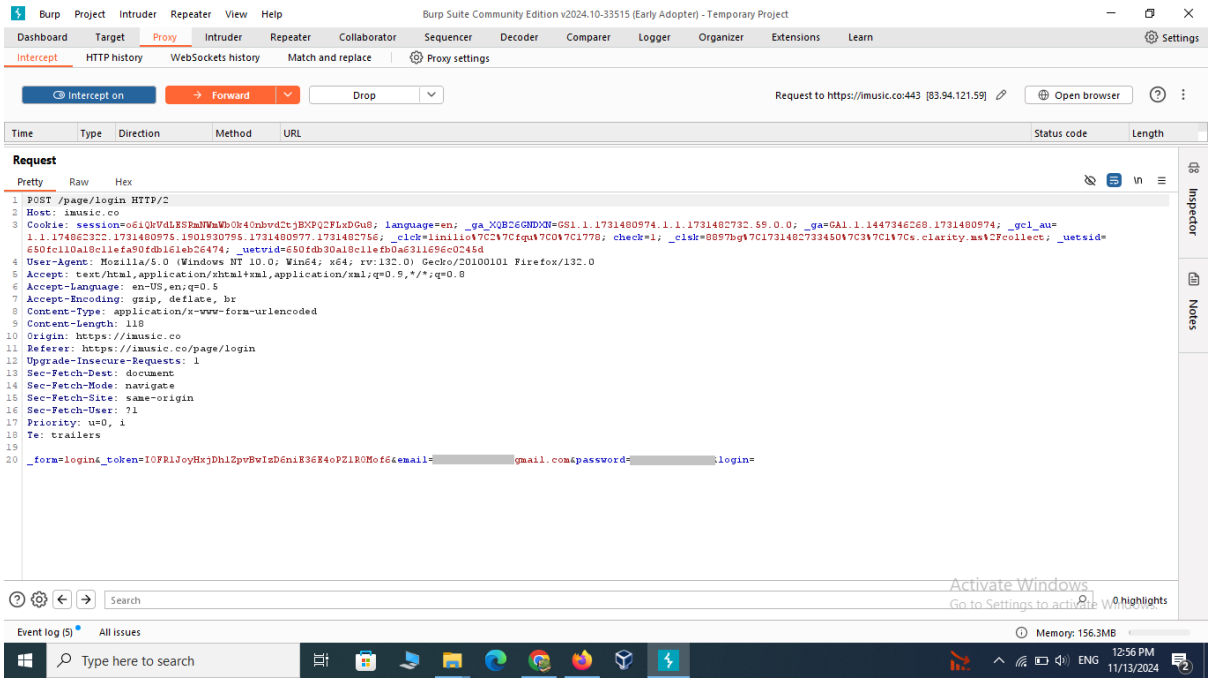
- Step 1: - Visit the Website.



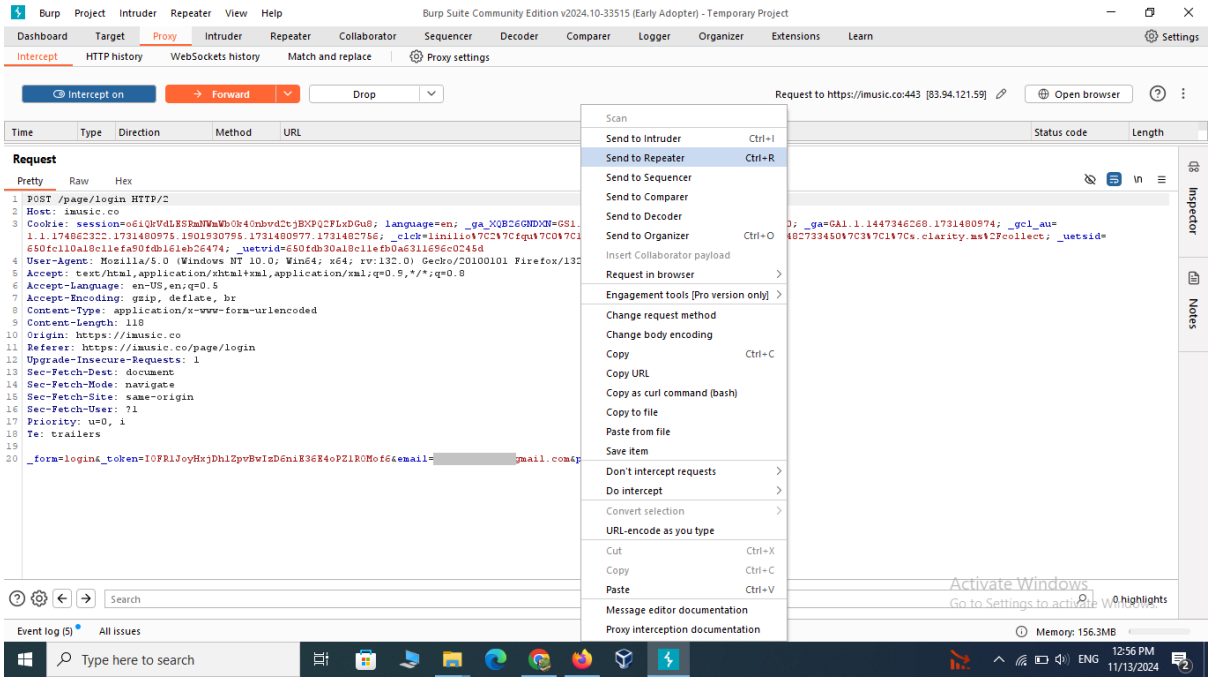
- Step 2: - Here I register first. I am login with correct mail and password and capture the request in burp suite.



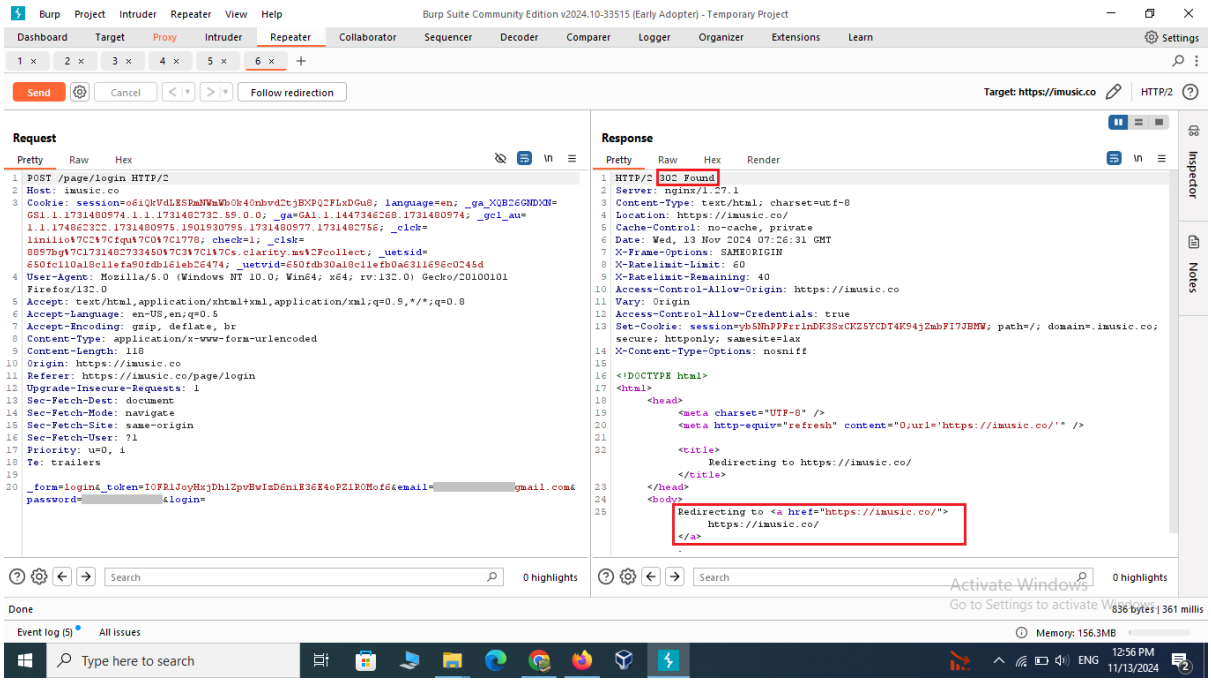
- Step 3: - See the request that I captured.



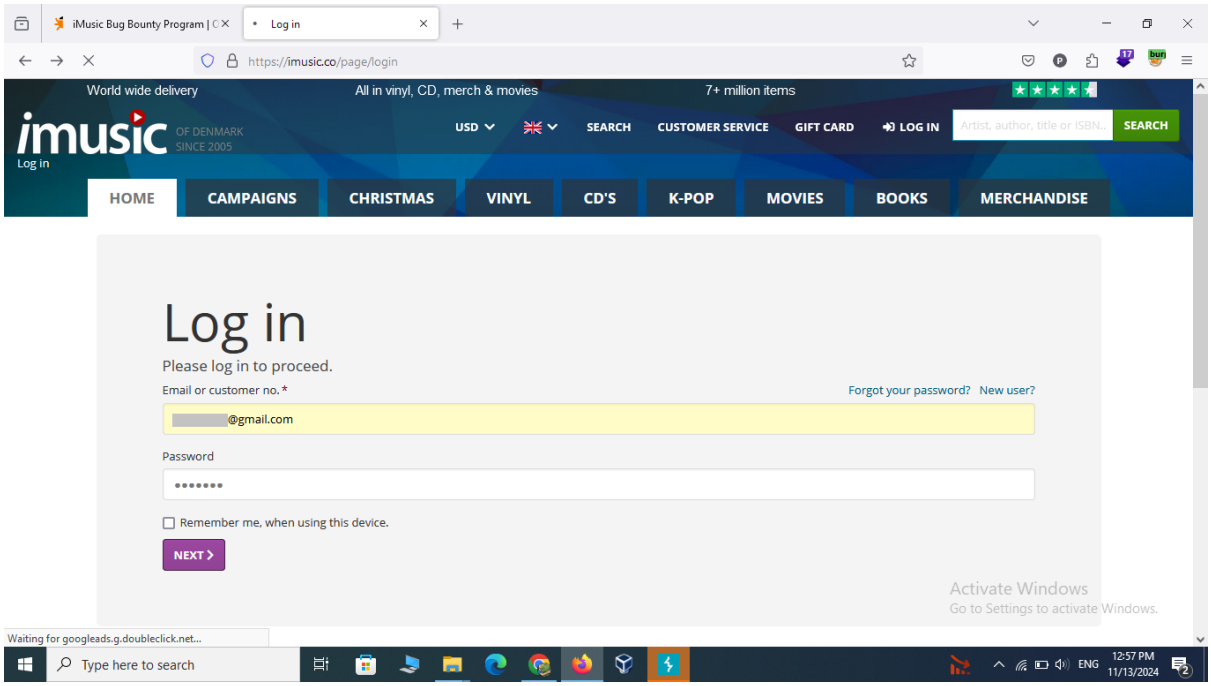
- Step 4: - Here, I send request to Repeater.



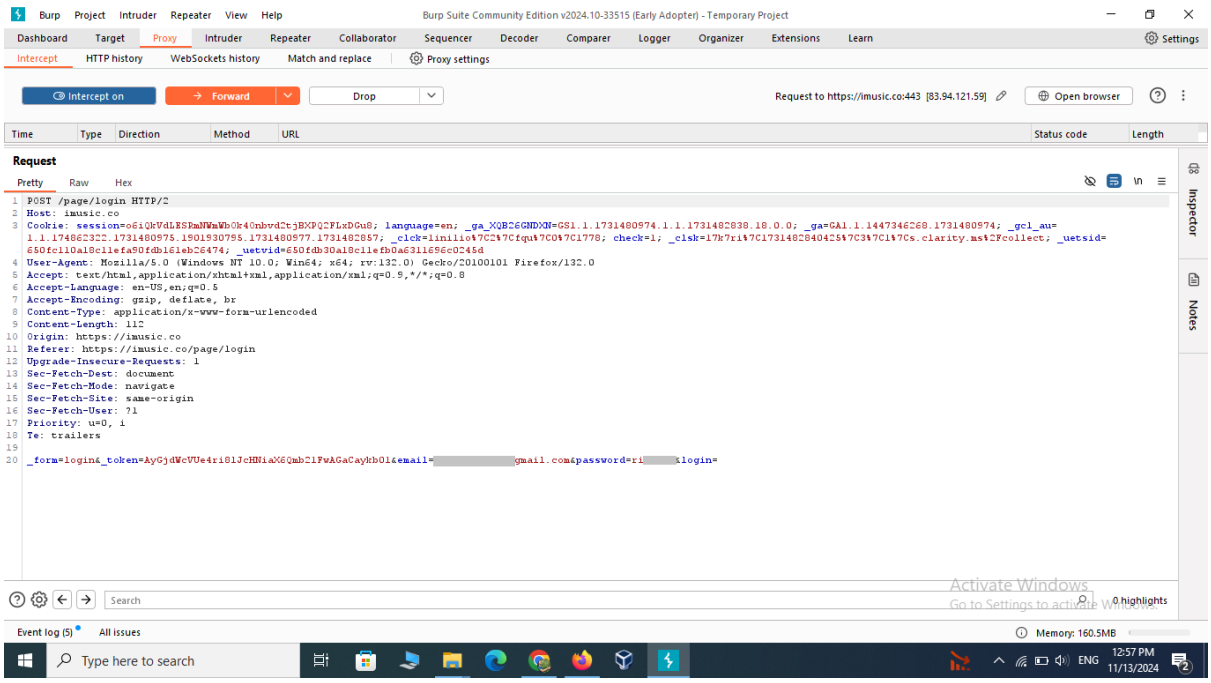
- Step 5: - Here I click on send button and see the response so it is found the page and redirect to that website.



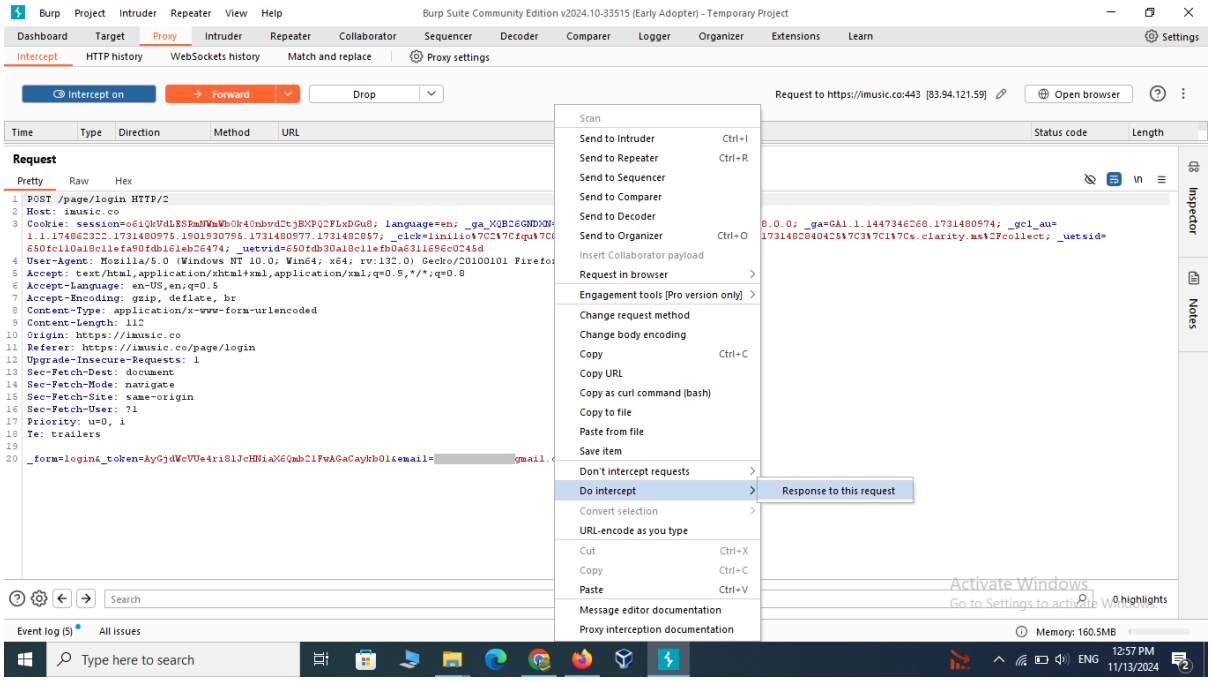
- Step 6: - Now, Here I will try capture the request **with wrong password.**



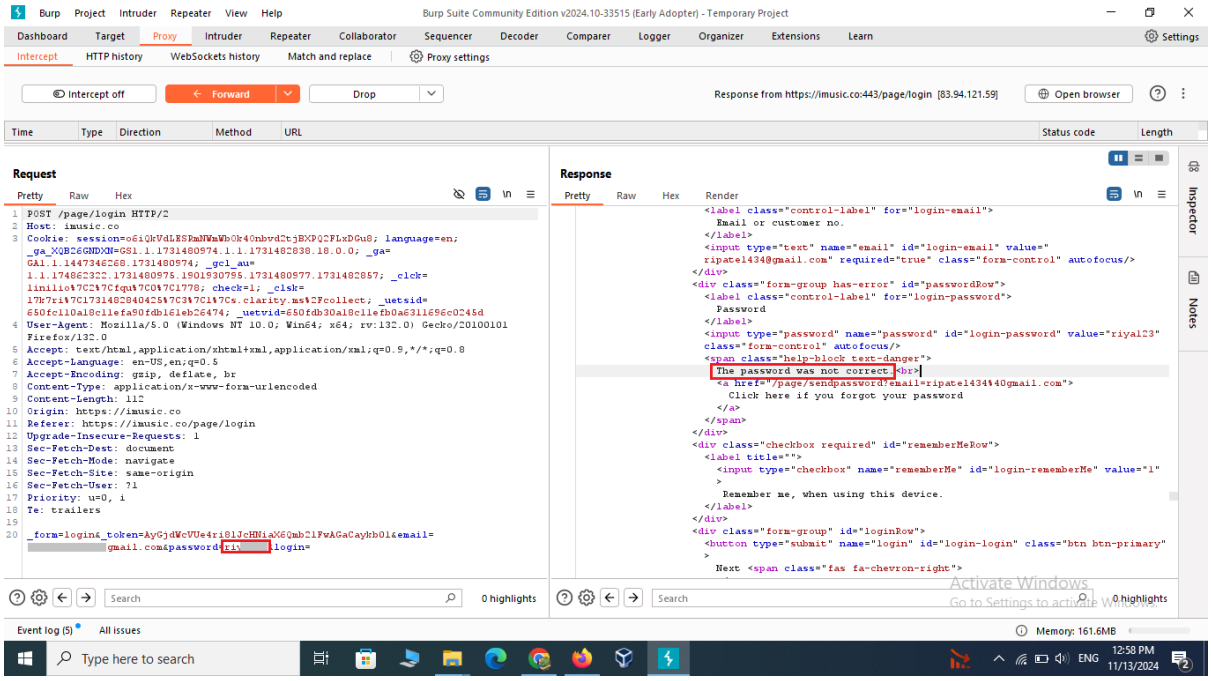
- Step 7: - Here you can see the request.



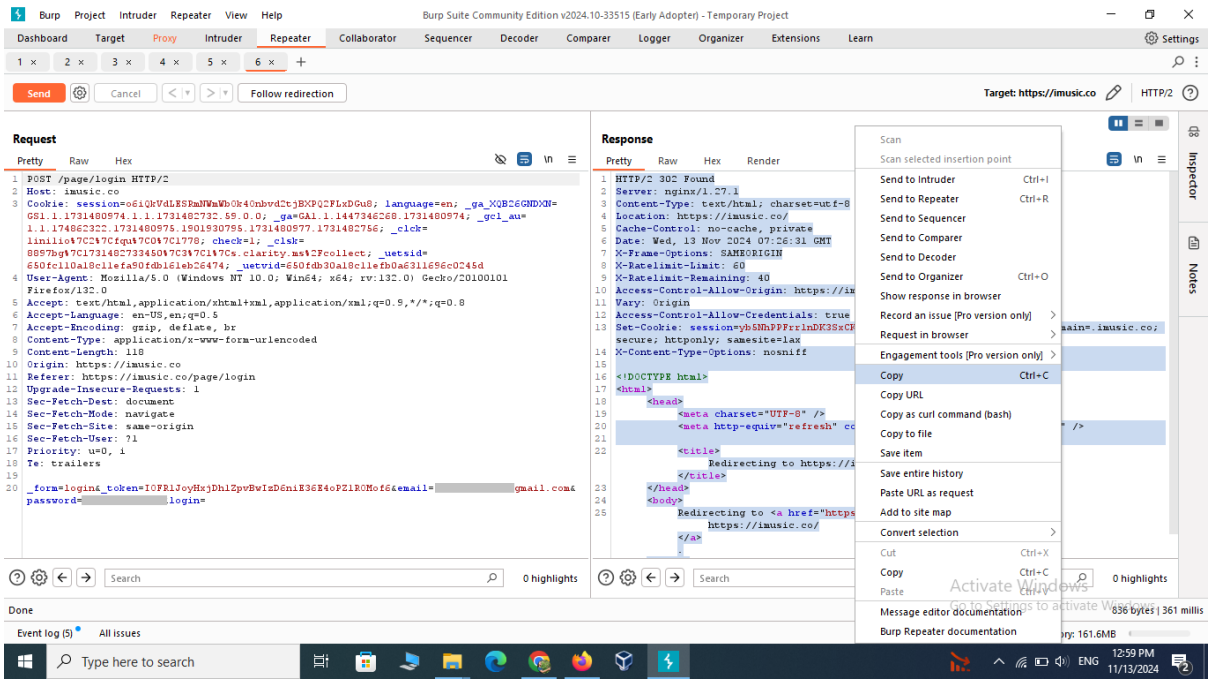
- Step 8: - Here I right click and Do intercept and Response to this request.



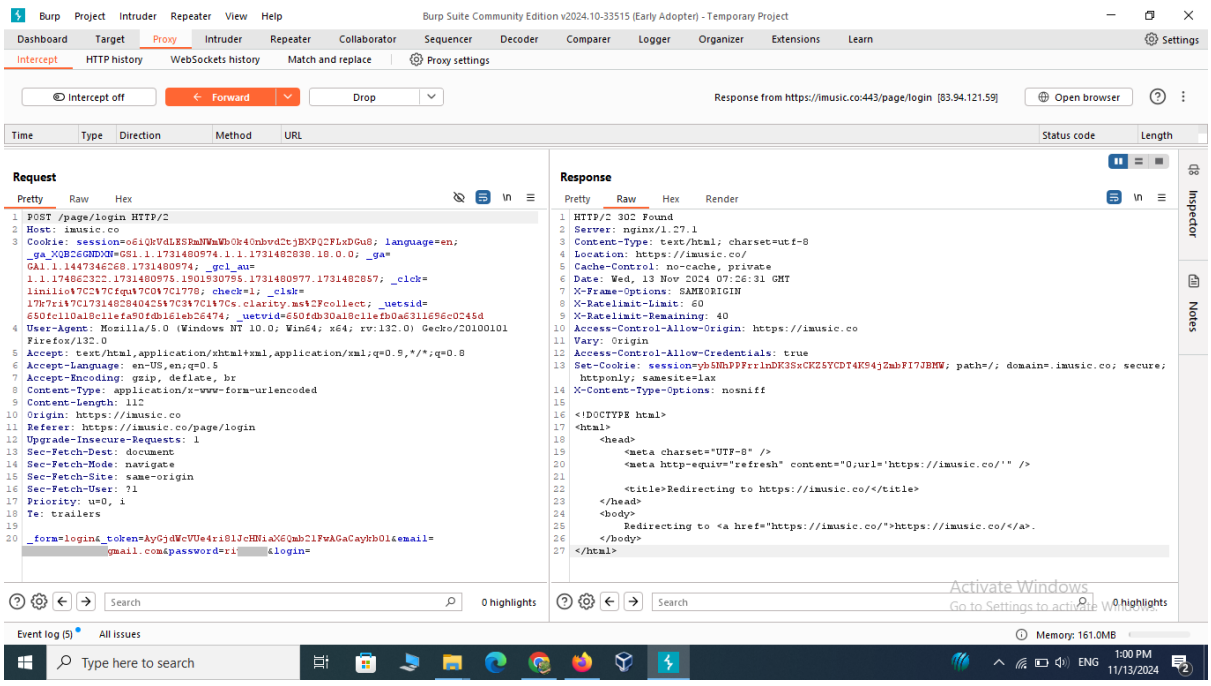
- Step 9: - Here you can see the message in Response that is the password was not correct.



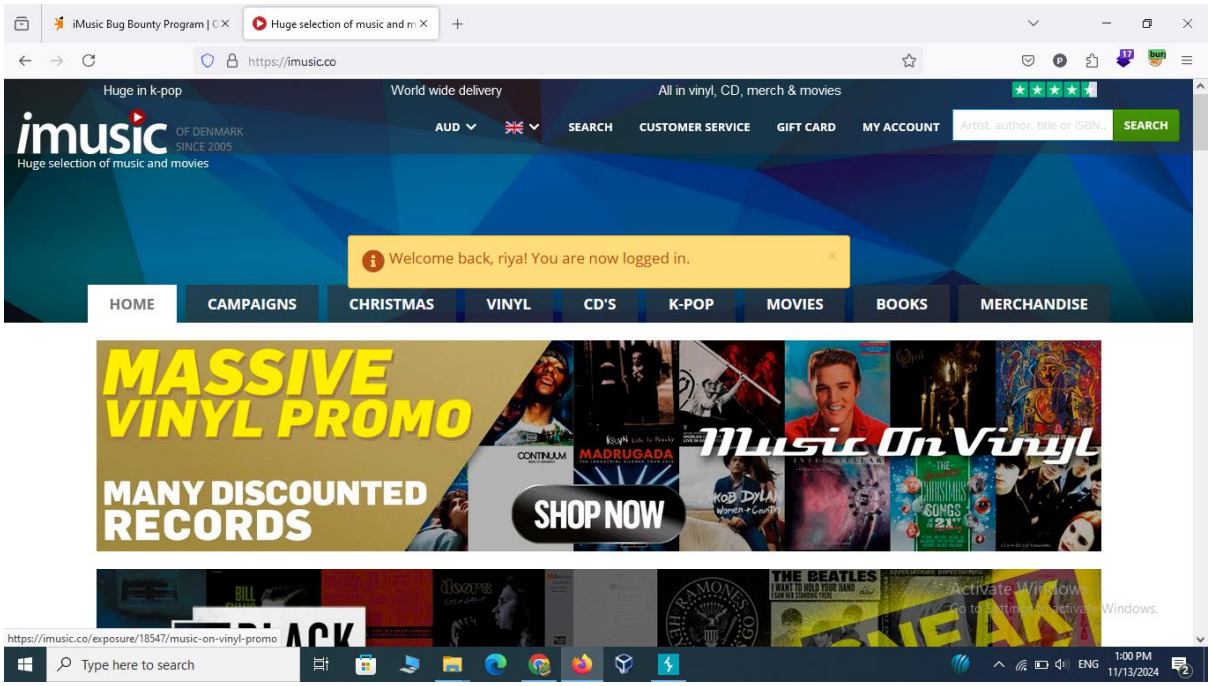
- Step 10: - Here copy the response. (This poc from step 5 which is login with correct password)



- Step 11: - Here I paste the Response which I copied. (This POC from step 10 which response with wrong password.) And I forward the Request.



- Step 12: - Here you can see the successfully login.



Observation
Authentication bypass vulnerability allows hackers to perform malicious activities by bypassing the authentication mechanism of the devices.

04. CROSS SITE REQUEST FORGERY (CSRF)

Description

The end-user perform unwanted actions within a web application that has already granted them authentication.

Affected Resource / Parameter	Severity
https://livingamalfi.com/	**HIGH**

Impact / Consequences

The impact of a successful CSRF attack can therefore be quite severe. Attackers often rely on social engineering techniques (e.g., phishing, usually by sending a link by email, which aims to redirect a user to a malicious site) to increase the chances of success.

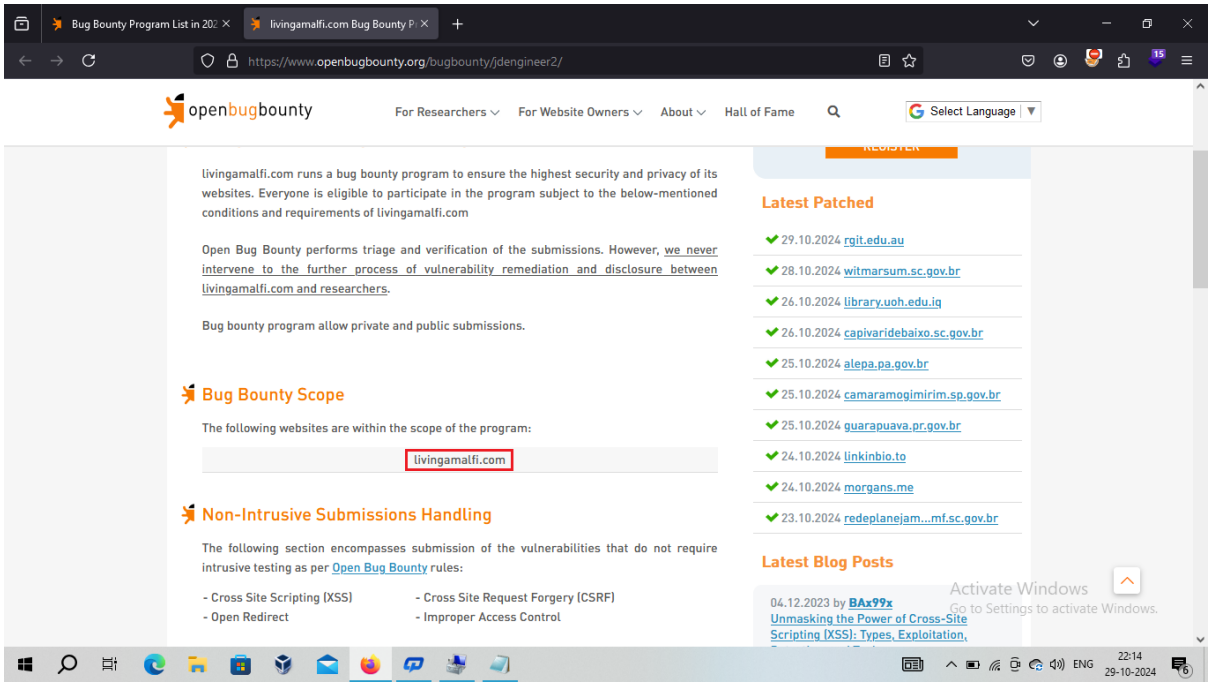
Recommendations

- ✓ Implementing CSRF token to prevent attacks
- ✓ Using the SameSite attribute on cookies to counter CSRF attacks
- ✓ Use Advanced Validation Techniques to Reduce CSRF

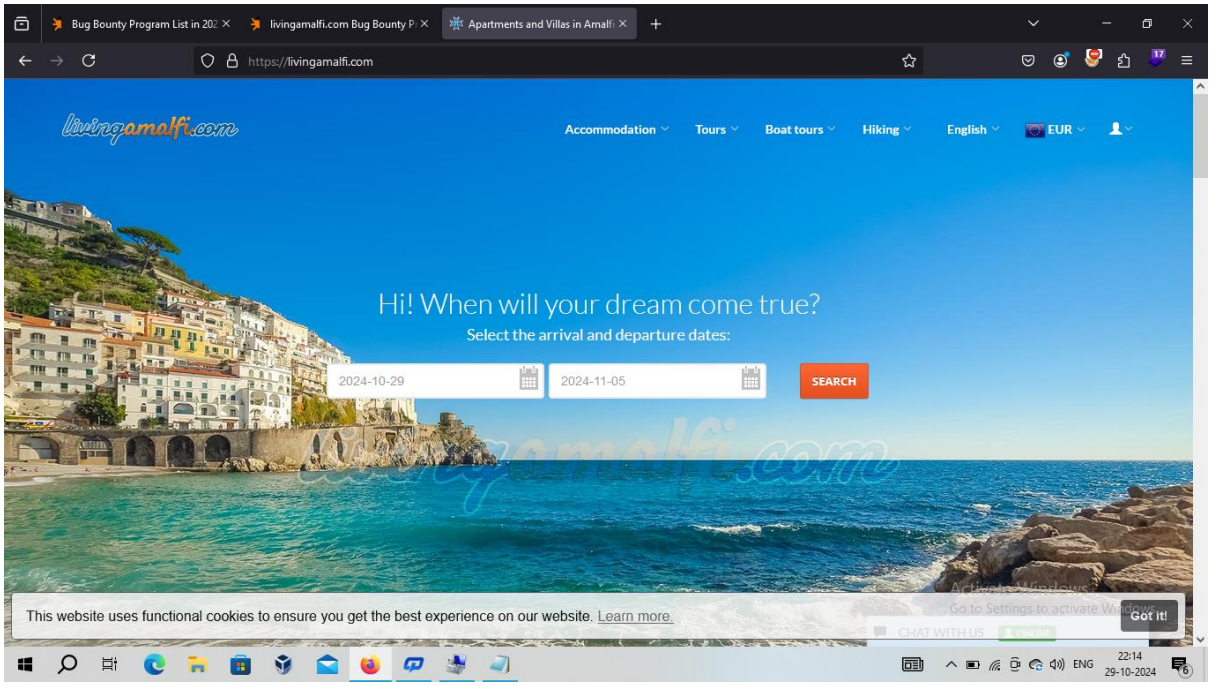
Tools Used	References
Burp-suite Vs Code	https://portswigger.net/burp https://code.visualstudio.com/download
CWE	**OWASP Top 10**
352	A8:2013 – Cross-Site Request Forgery

Proof of Vulnerability

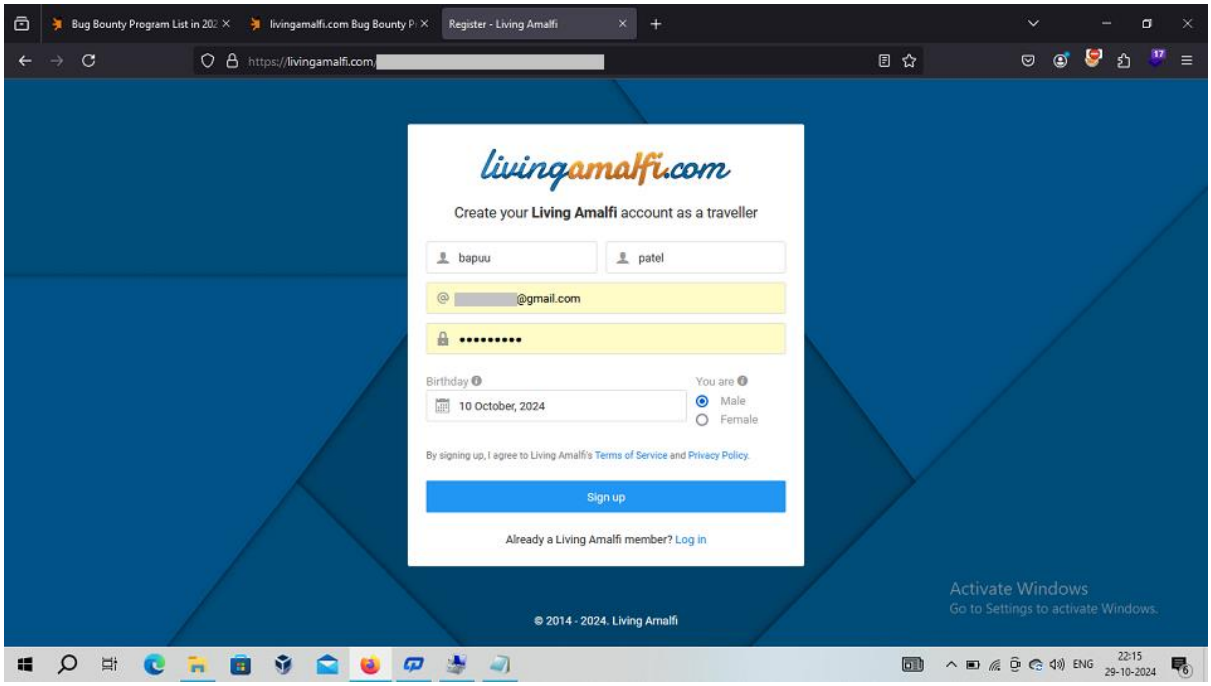
→ Here you can see the website of openbugbounty that is use for vulnerability testing.



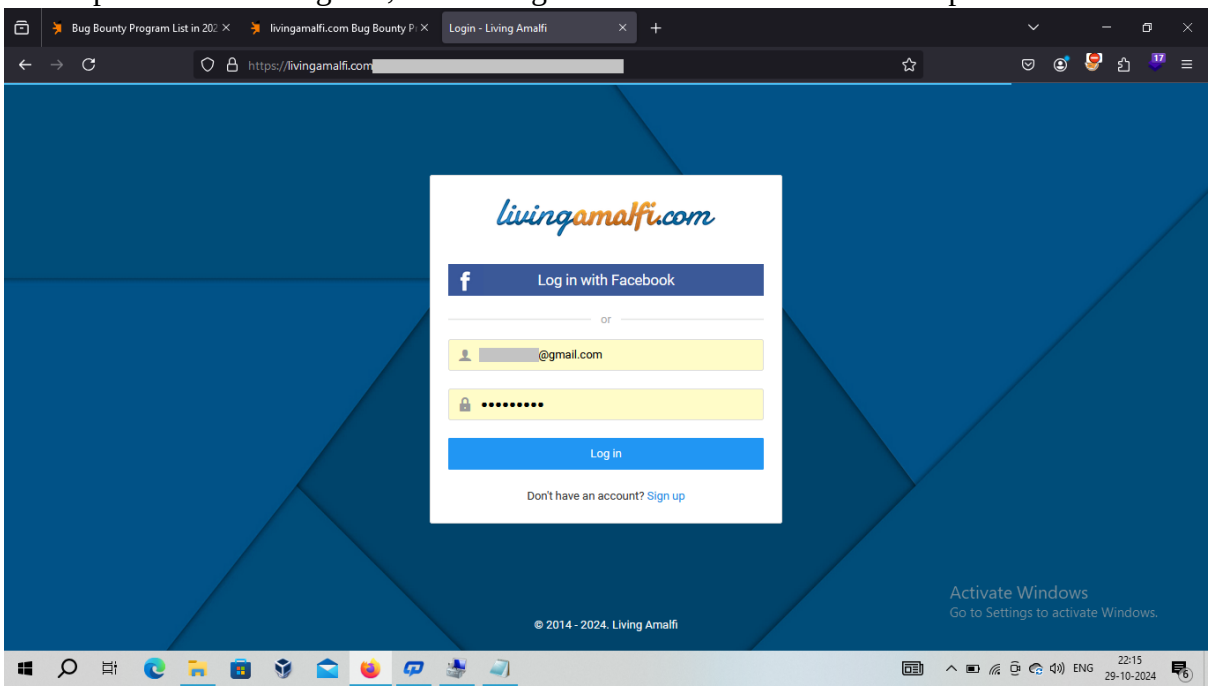
- Step 1: - Visit Website.



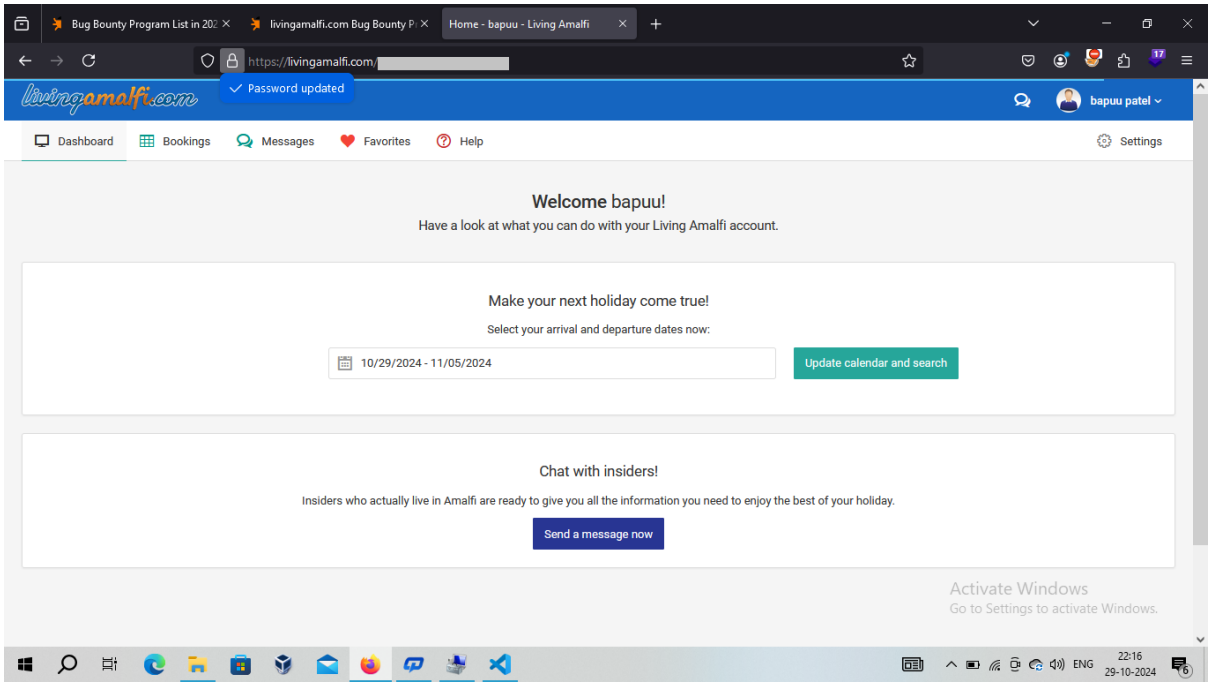
- Step 2: - Here, we are Register the account.



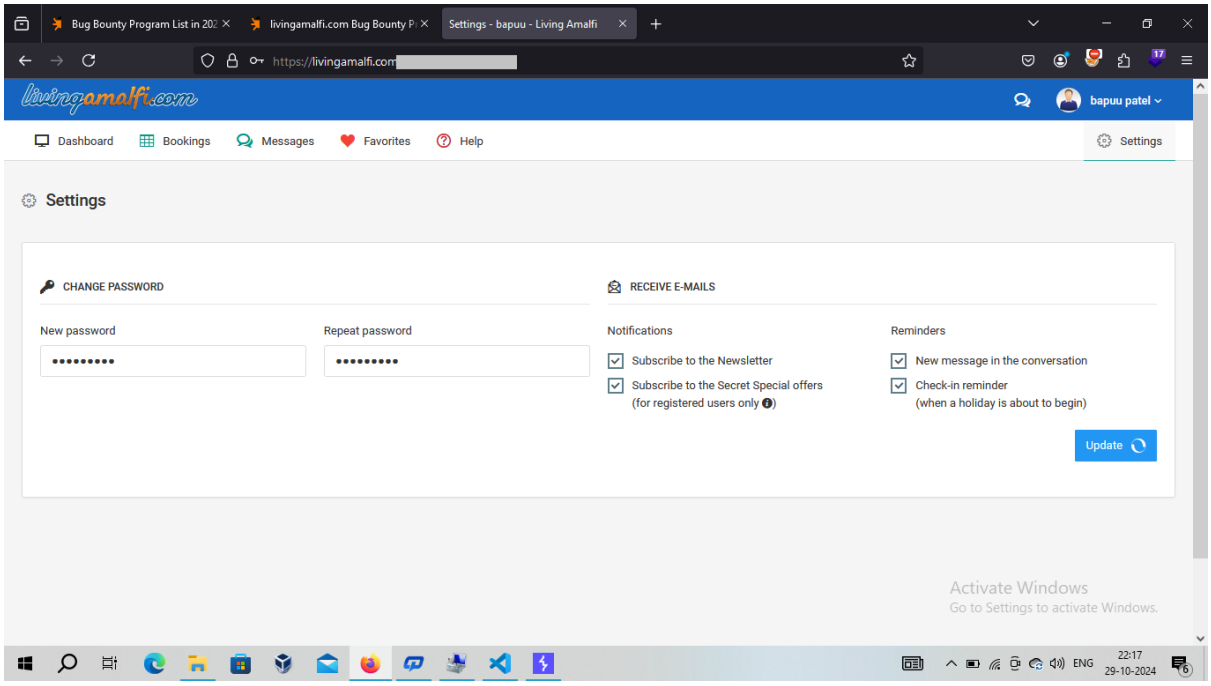
- Step 3: - After the register, we are Log in this account with email and password.



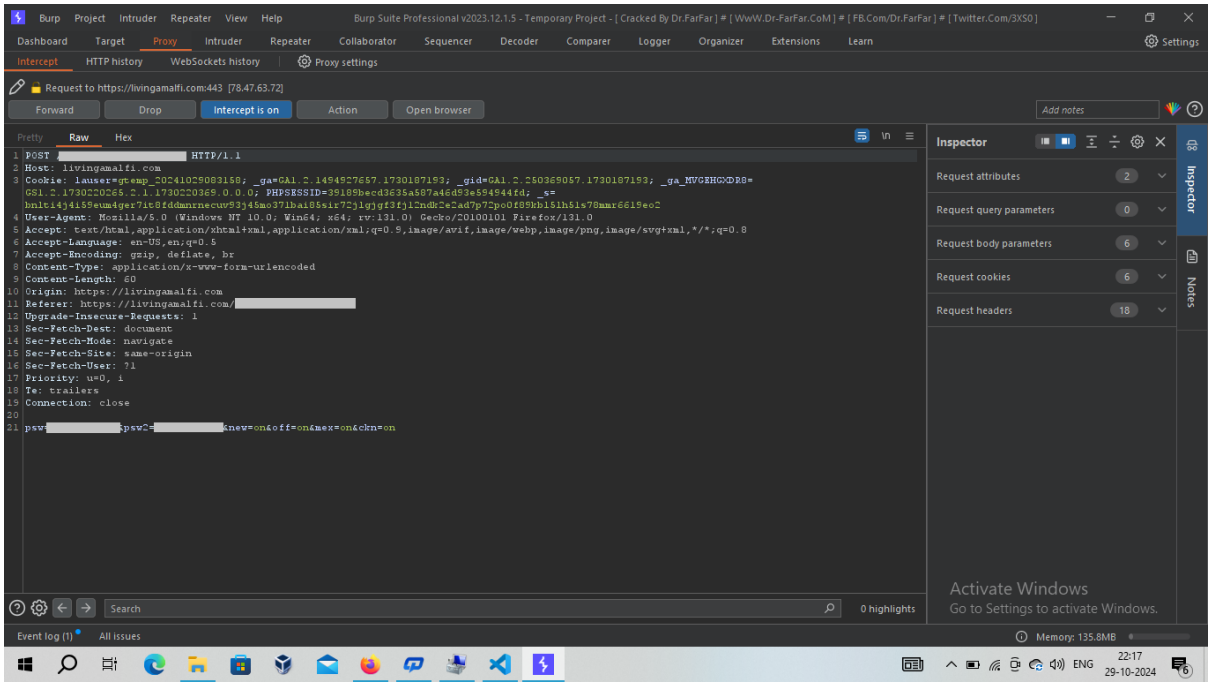
- Step 4: - Here you can see the successfully log in website.



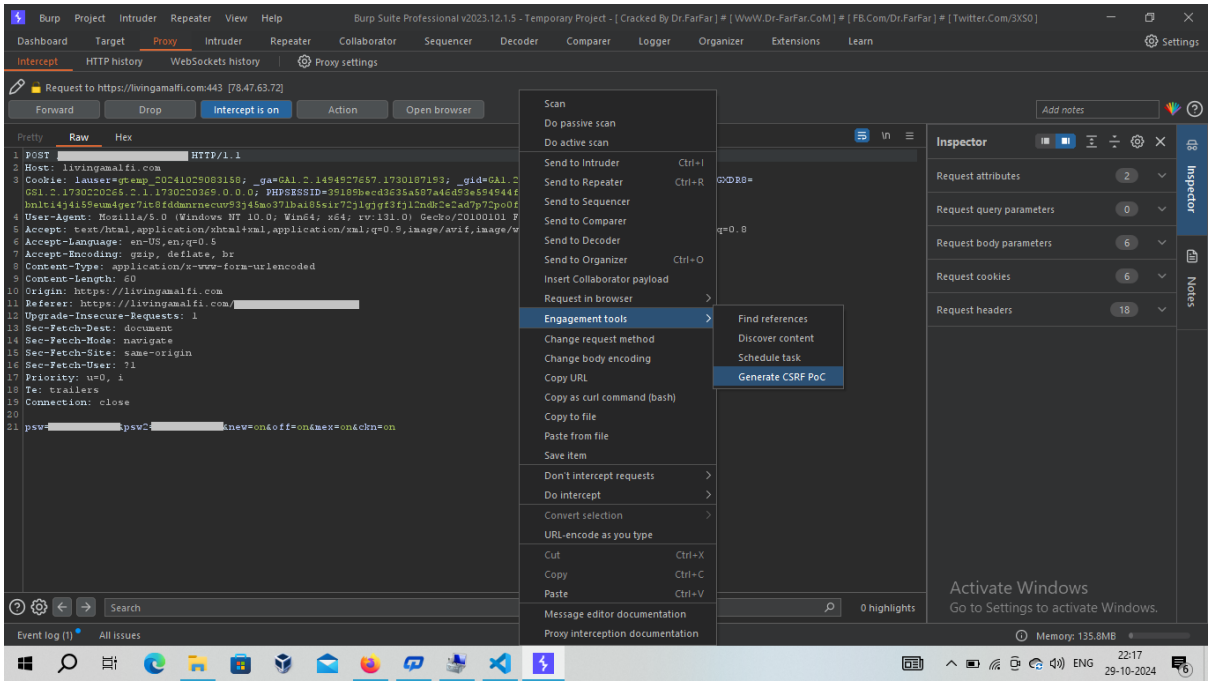
- Step 5: - Now, we are writing a new password and click on the update button.



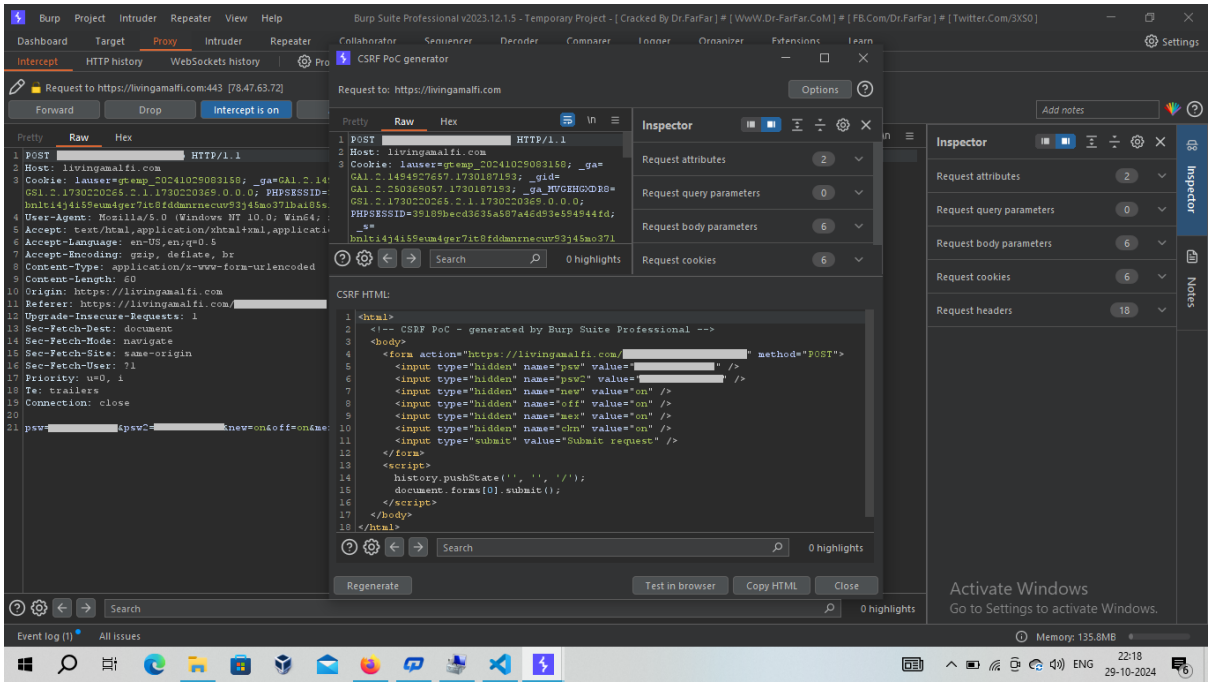
- Step 6: - Here we are capture the request of new password.



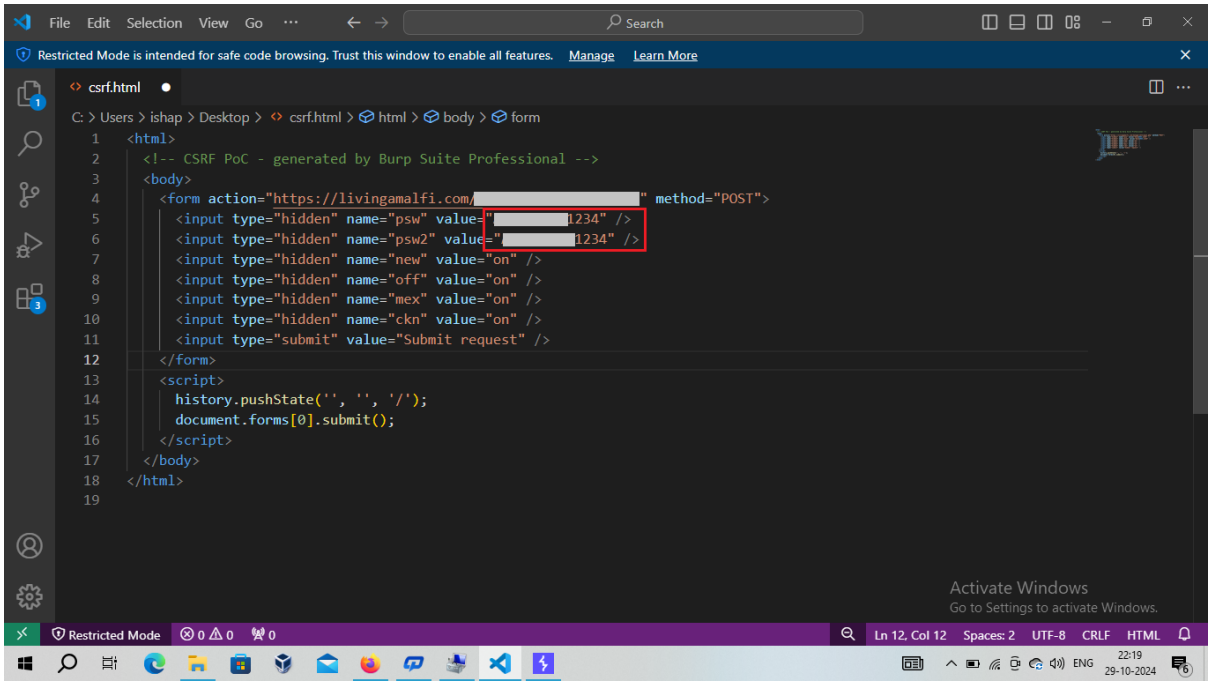
- Step 7: - After Go to Engagement tools and Generate CSRF POC.



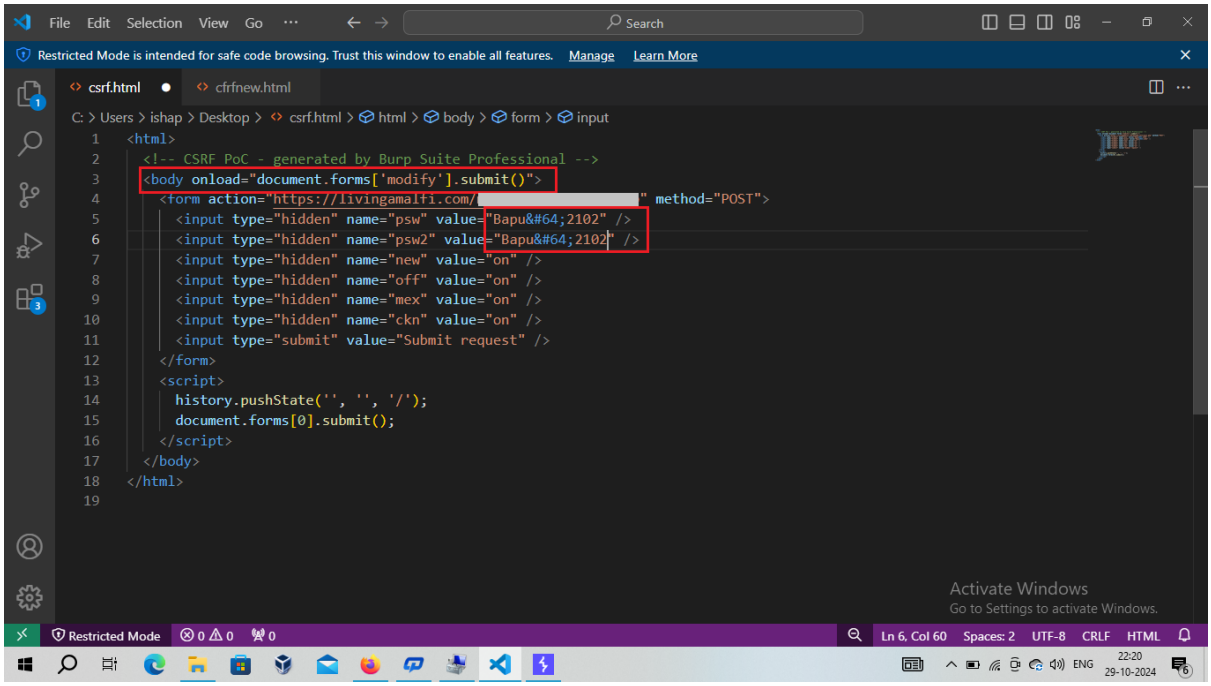
- Step 8: - Here you can see the CSRF token Generated and copy this HTML code.



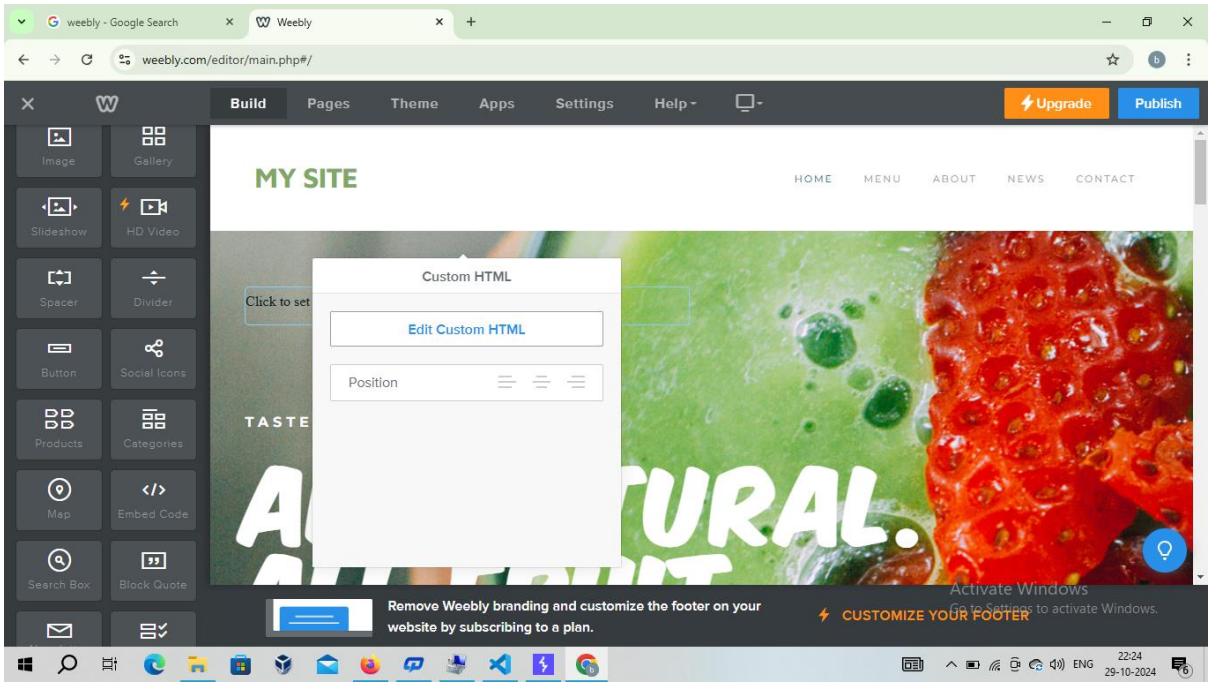
- Step 9: - In the vs code we are paste the CSRF token.



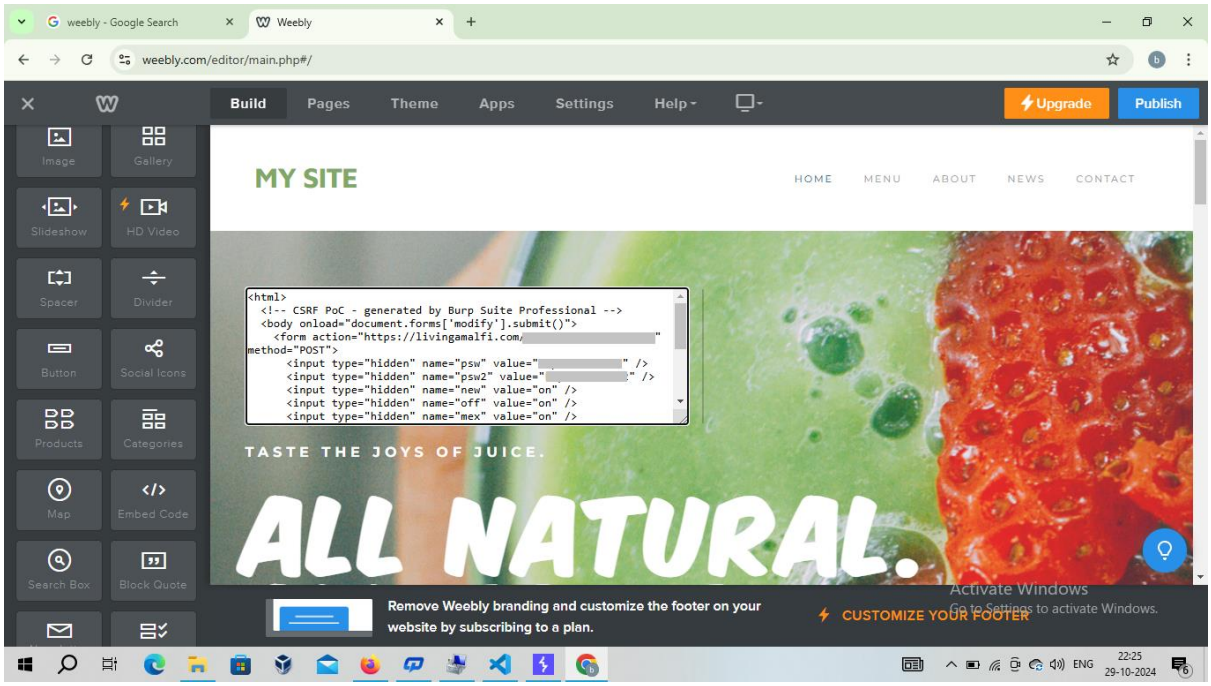
- Step 10: - Here we are change the password and also add the payload **<onload=”document.forms[‘modify’].submit()”>** in body tag.



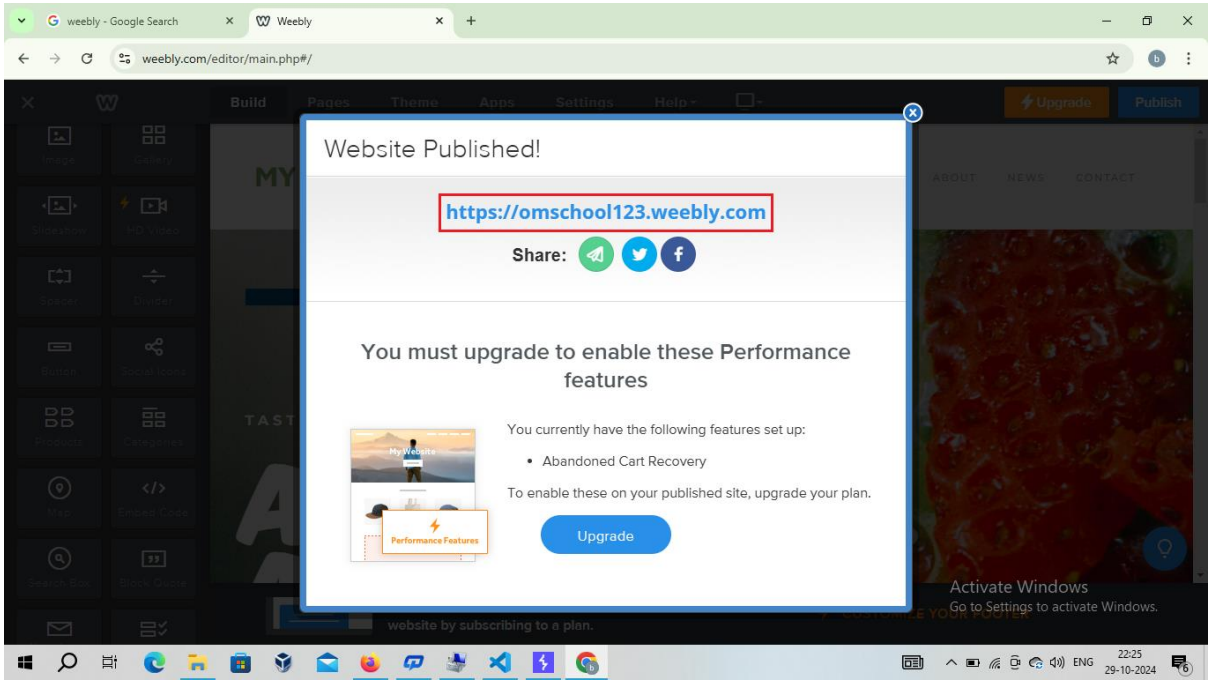
- Step 11: - This is weebly website for create our website.



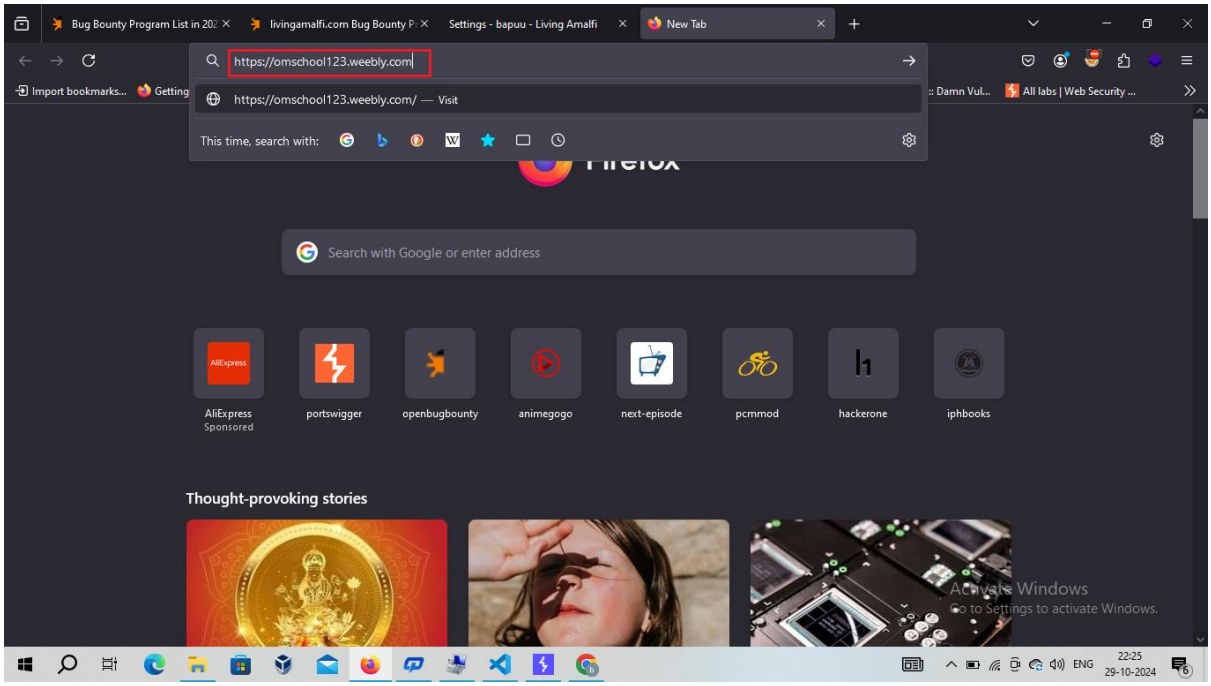
- Step 12: - Here we are add the CSRF token in the embed code.



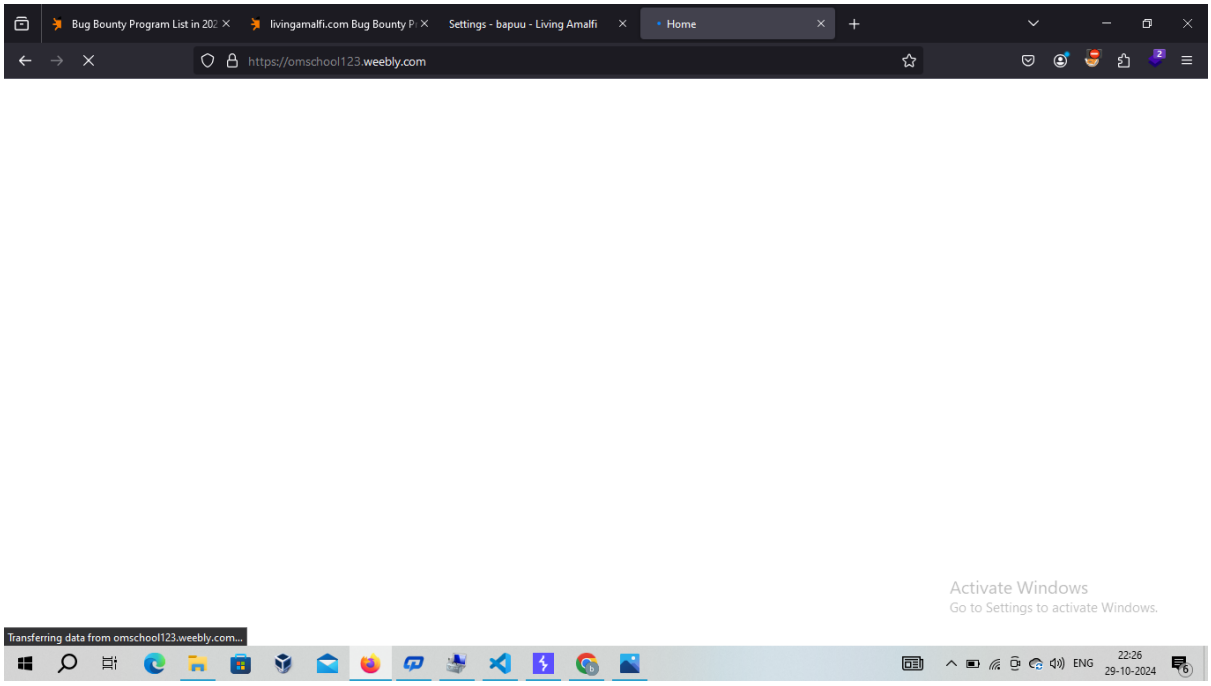
- Step 13: - Here you can see the website is published.



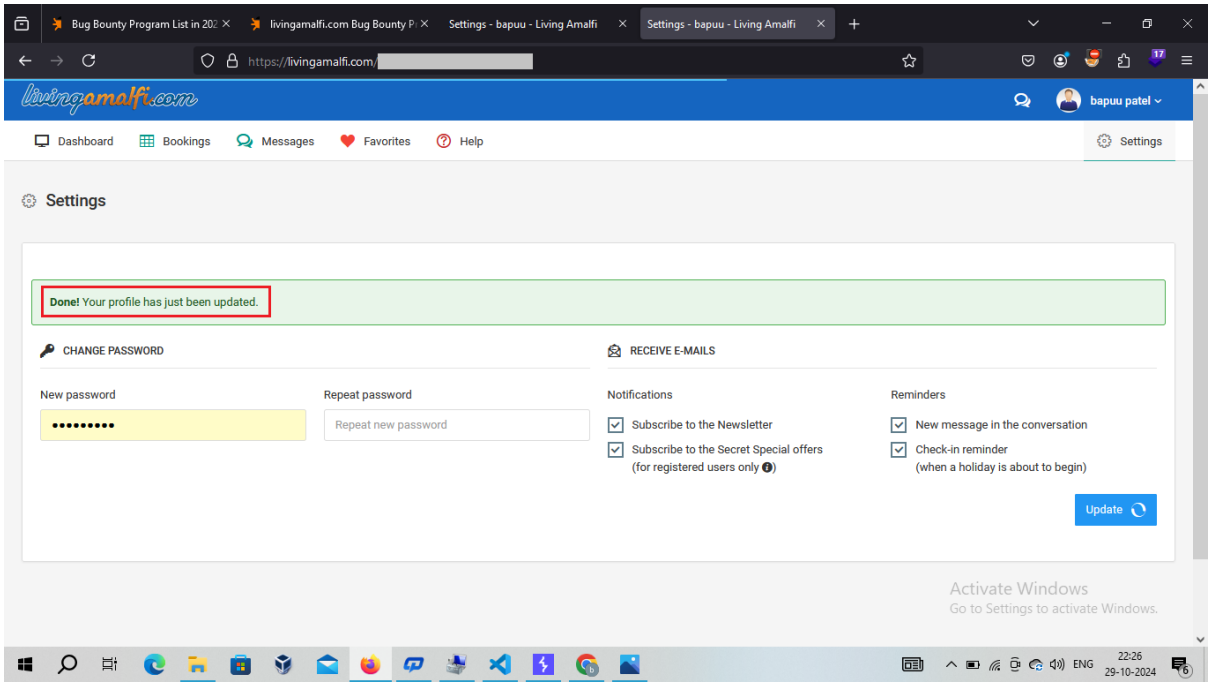
- Step 14: - After, open the browser and open the our website.



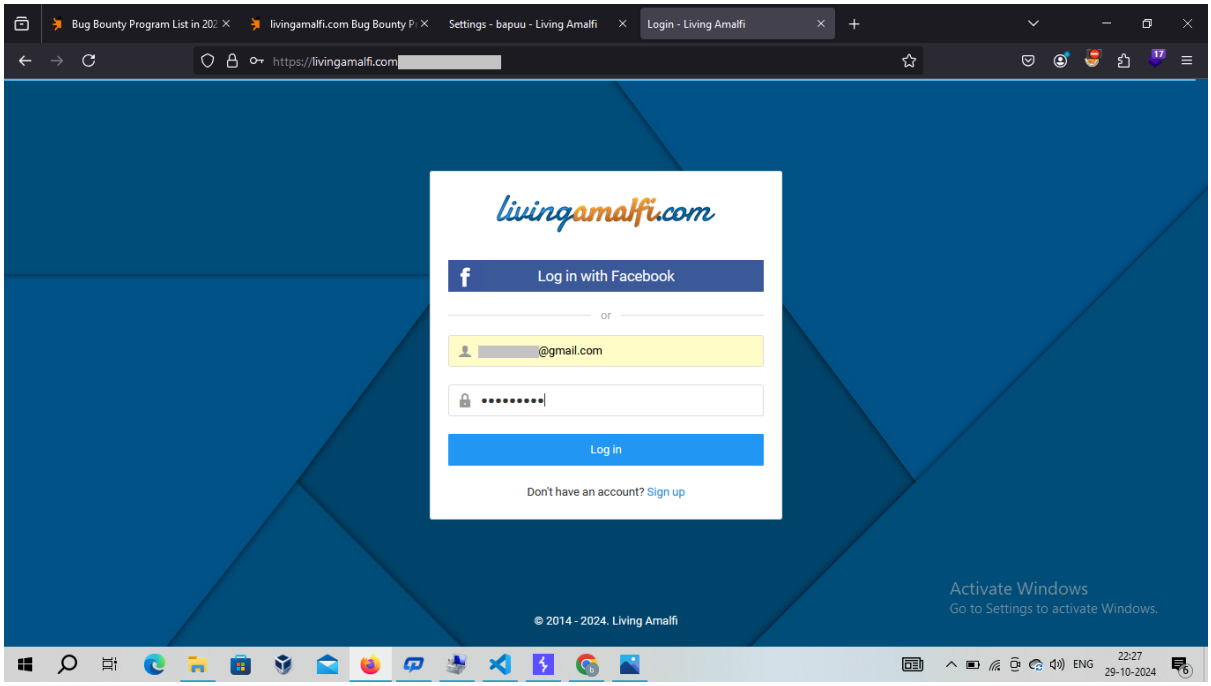
- Step 15: - Here you can see that website is loading.



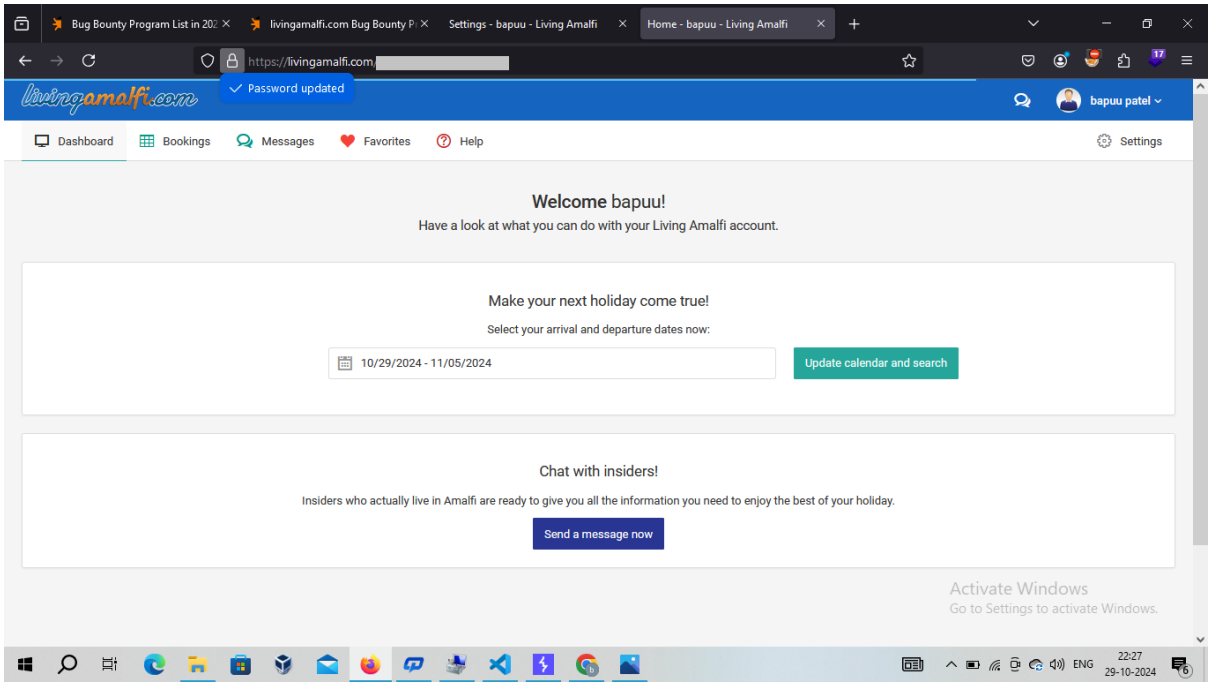
- Step 16: - Now you can see the published website is redirect on that vulnerable website. And see the password is successfully changed.



- Step 17: - Here we are log in with mail id and that updated password.



- Step 18: - Here you can see the website is successfully log in.

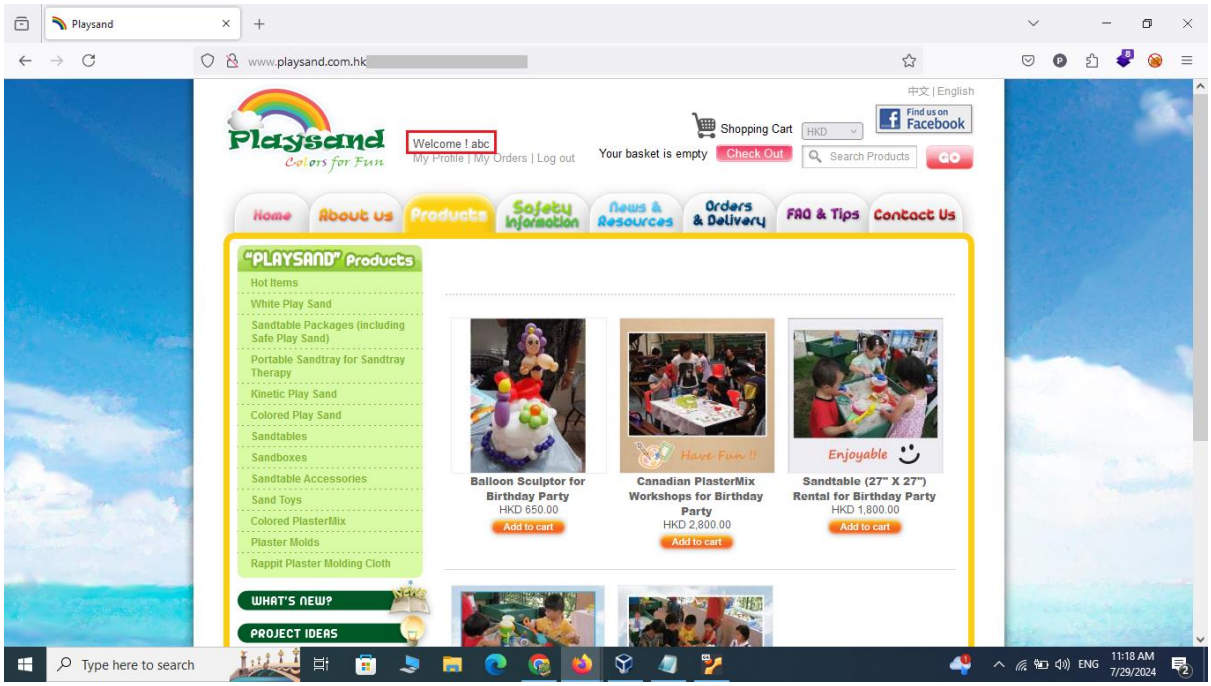


Observation
CSRF vulnerability depends on how the HTTP protocol manages web requests and processes. In a CSRF attack, the attacker tricks the authenticated user into performing malicious action on a web application without the user's knowledge.

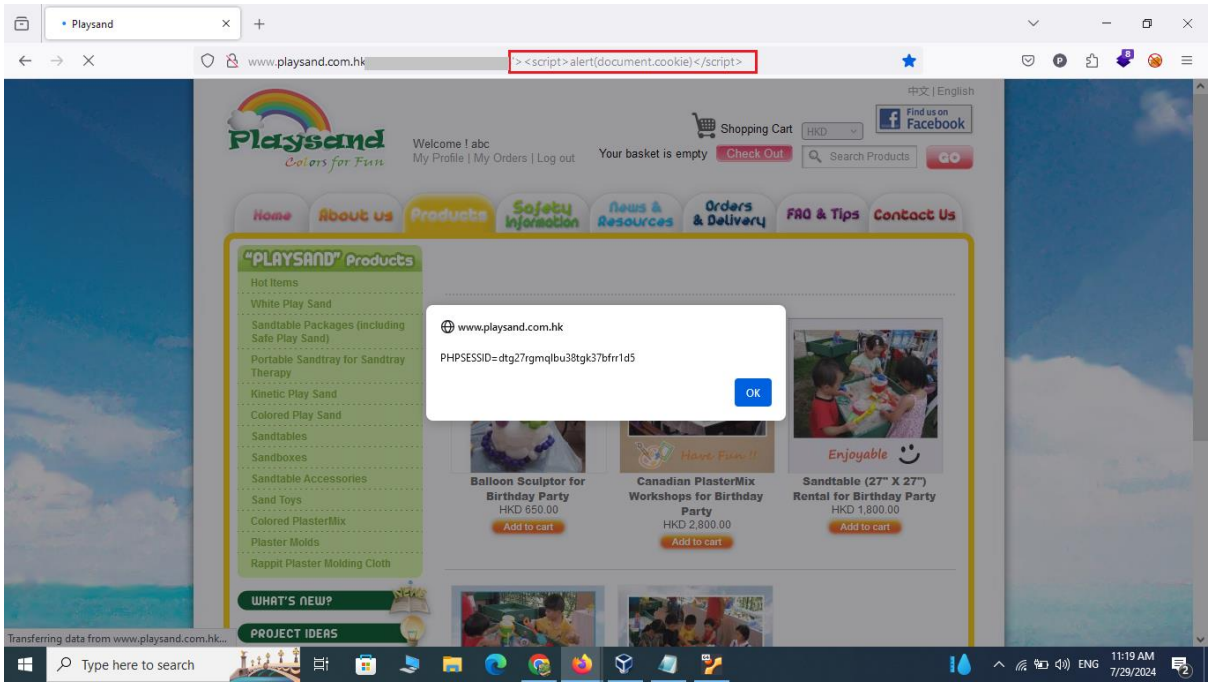
05. REFLECTED XSS	
Description	
It occur when a malicious script is reflected off of a web application to the victim's browser.	
Affected Resource / Parameter	**Severity**
http://www.playsand.com.hk	**HIGH**
Impact / Consequences	
XSS stands for Cross-Site Scripting and it is a web-based vulnerability in which an attacker can inject malicious scripts in the application. ✓ Codes injected into a vulnerable application can exfiltrate data or install malware on the user's machine. ✓ Attacker can get the cookie and Attackers can masquerade as authorized users via session cookies, allowing them to perform any action allowed by the user account.	
Recommendations	
Filter input on arrival. At the point where user input is received, filter as strictly as possible based on what is expected or valid input. ✓ Encode data on output. At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.	
Tools Used	**References**
Wappalyzer	https://www.wappalyzer.com/
CWE	**OWASP Top 10**
79	A7:2017-Cross-Site Scripting (XSS)
Proof of Vulnerability	

- Step 1: - Visit the Website.

- Step 2: - Here, Register first then login with username and password. And you can see the login abc user.

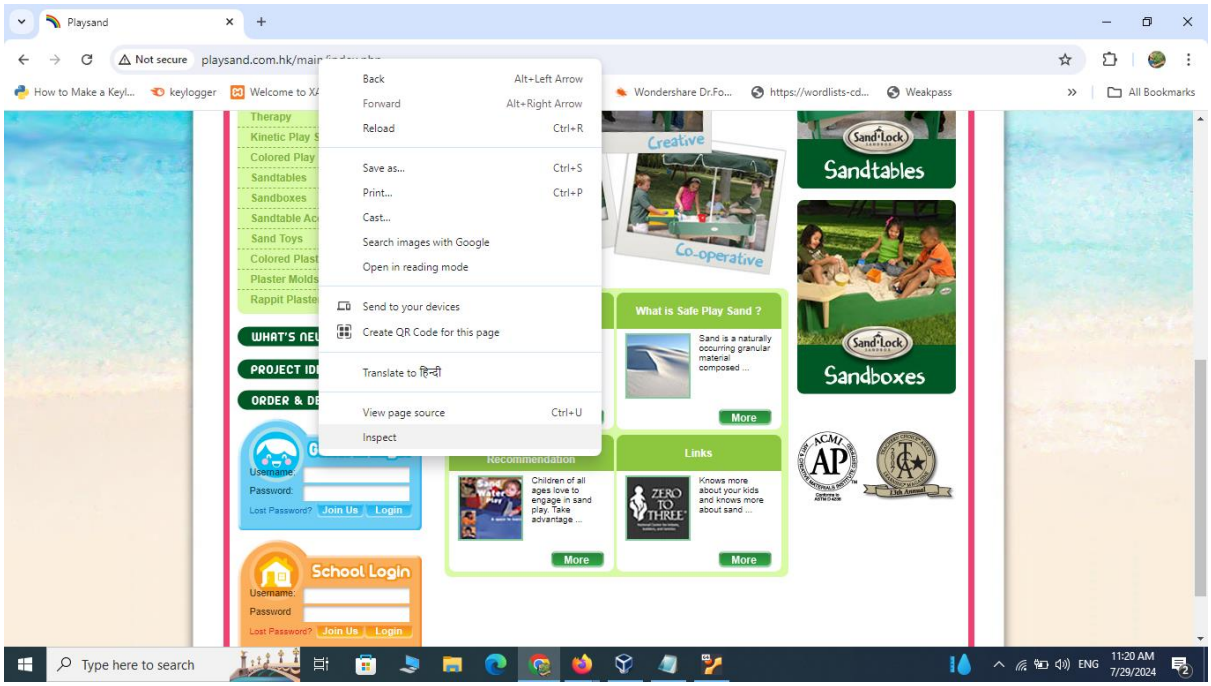


- Step 3: - Here I write JavaScript in URL like this **'><script>alert(document.cookie)</script>** to inject cookie. After this we get a pop up and with the content is (cookie) Which one I injected in the parameter.

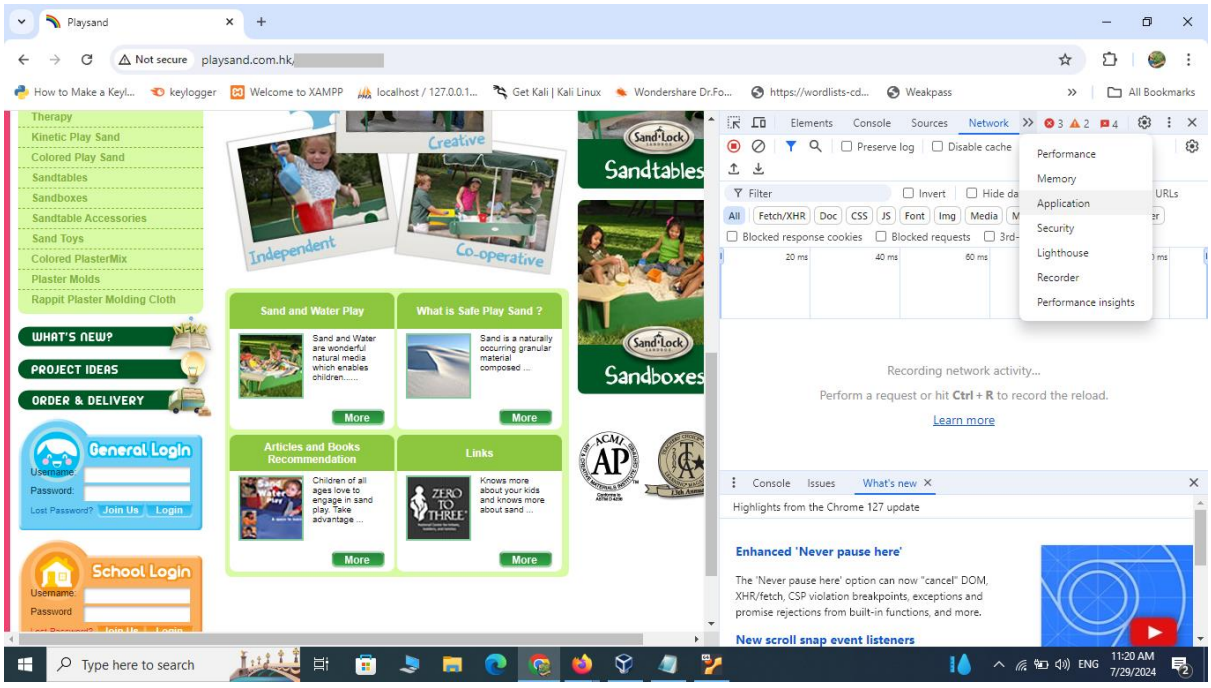


- Step 4: - Now I go to google chrome and open that website.

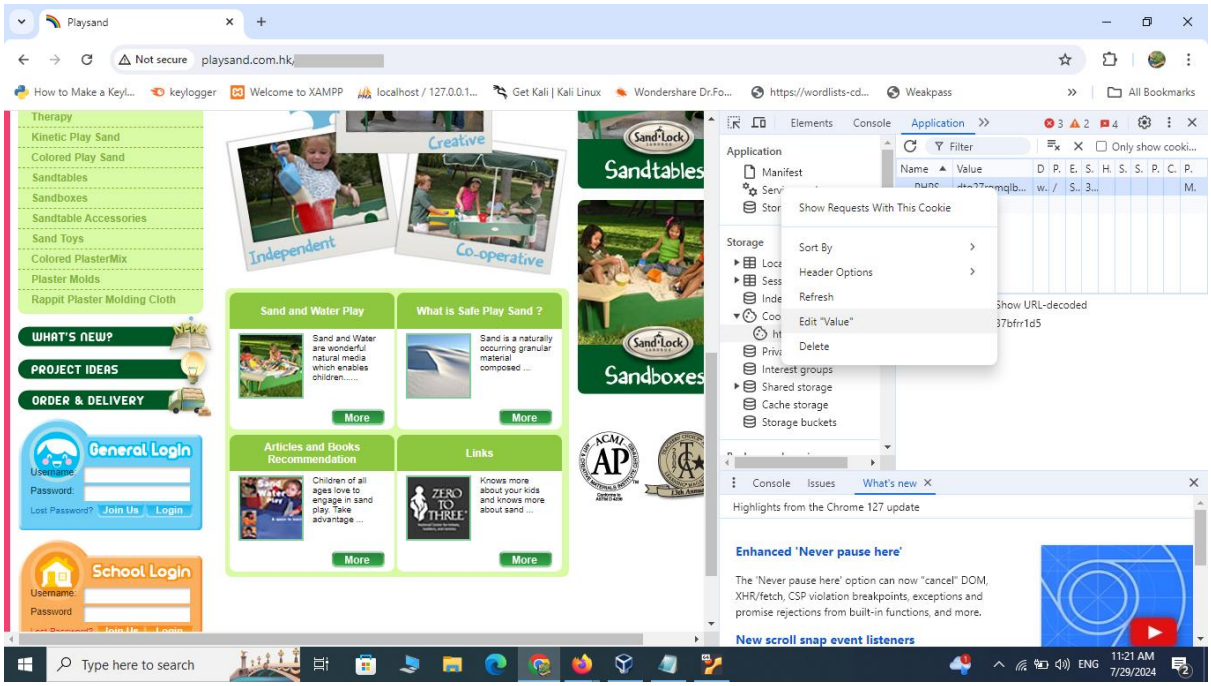
- Step 5: - Here I right click and go to Inspect.



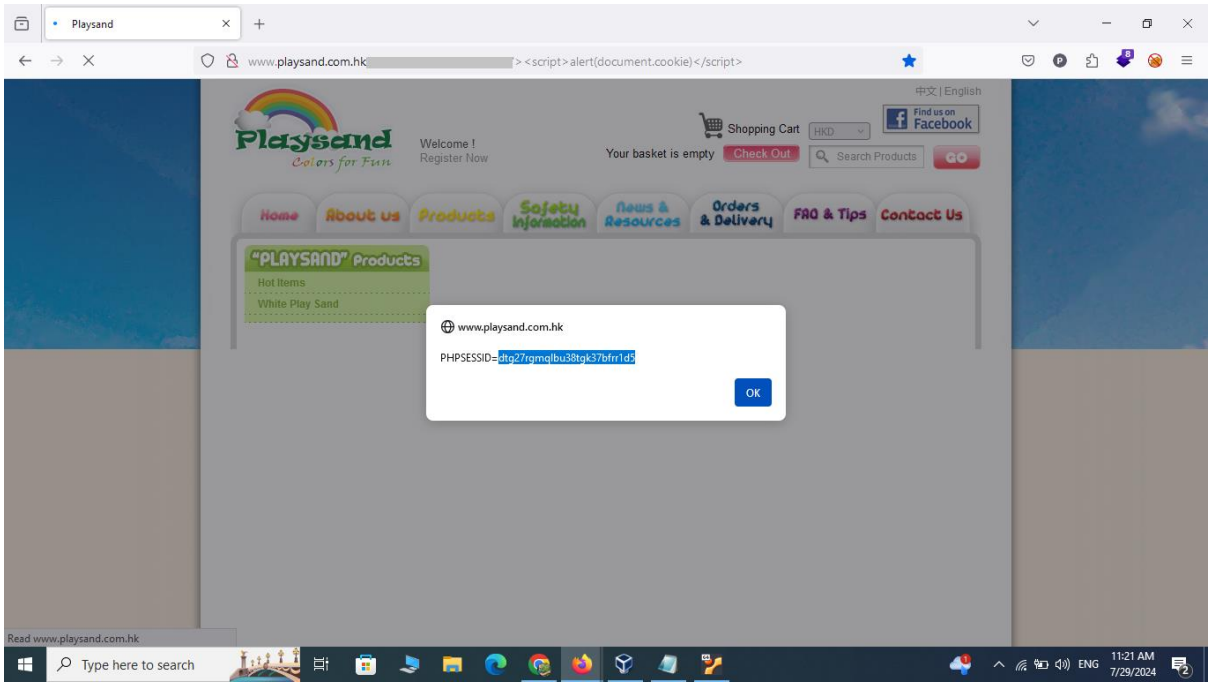
- Step 6: - And now I go to application.



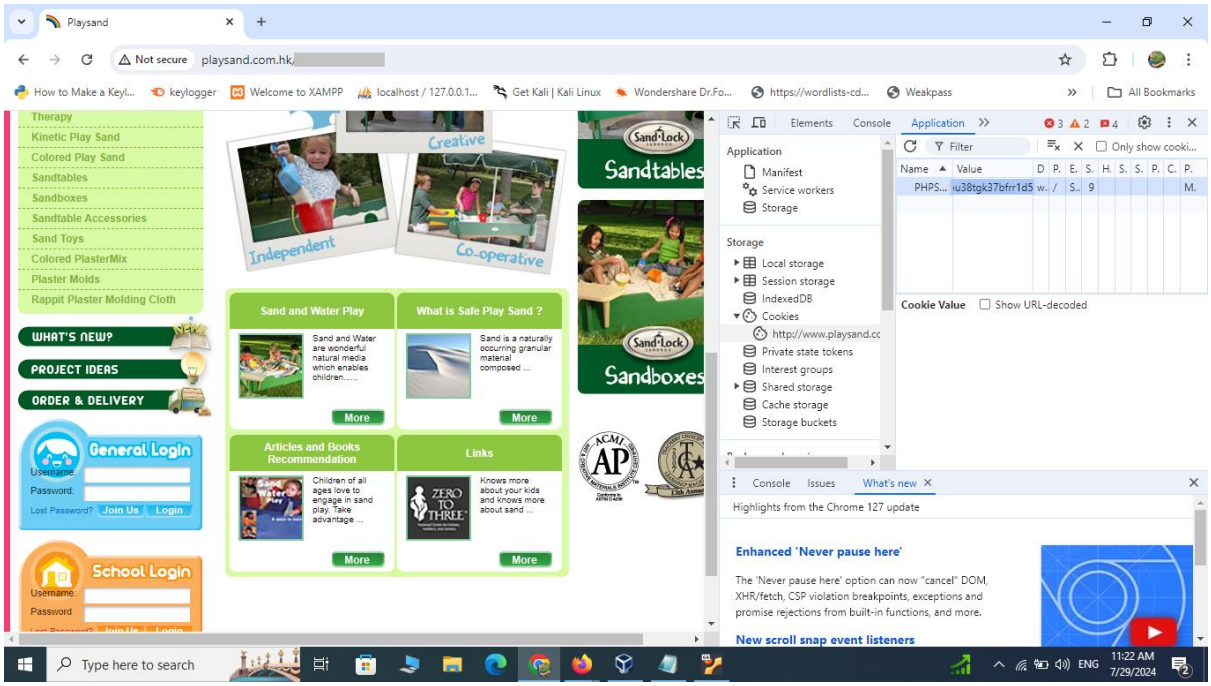
- Step 7: - Then I go to Edit “value”.



- Step 8: - Here I copy the cookie from the firefox.



- Step 9: - Then I paste cookie here and press enter.

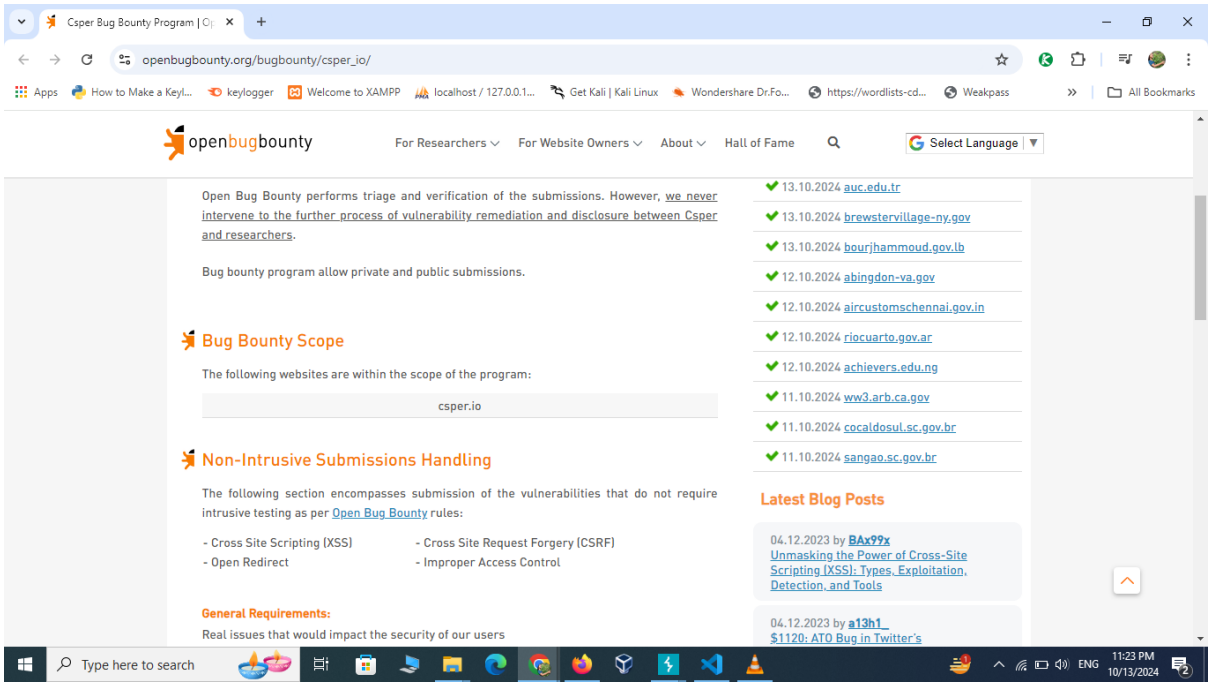


- Step 10: - So here you can see the successfully login the abc user.

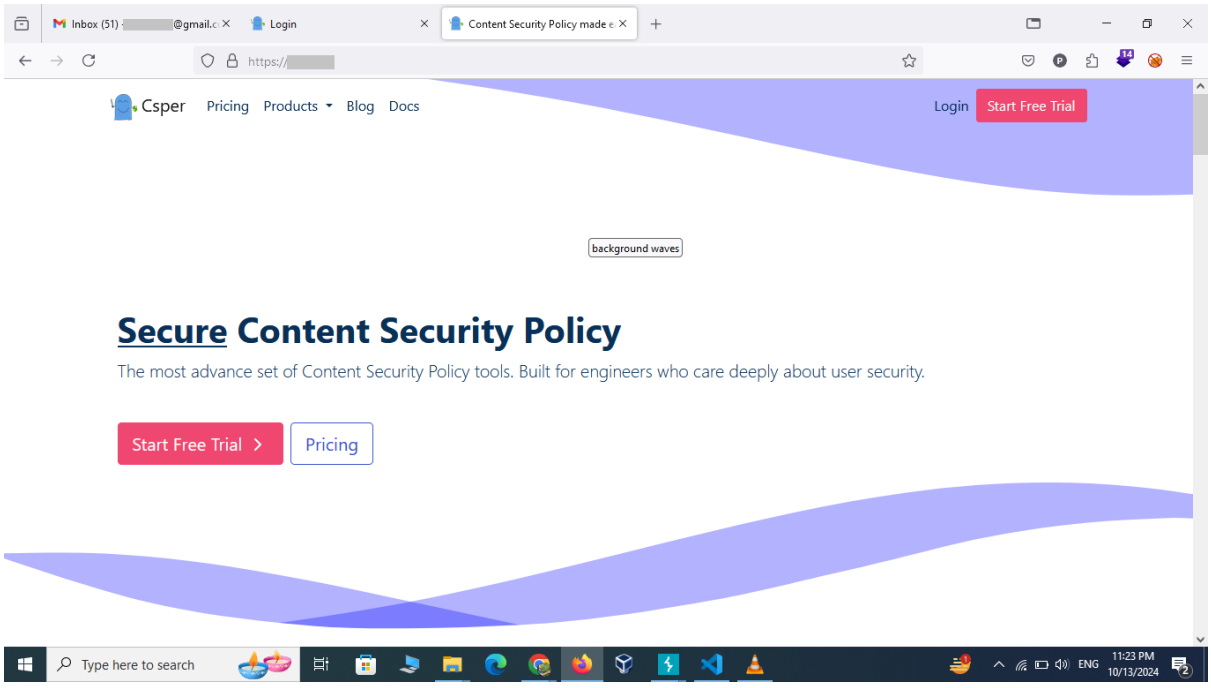
Observation
It was observed that the website is missing X-XSS-Protection which means that the website may be at risk of Cross-site Scripting (XSS) attacks. The HTTP 'X-XSS-Protection' response header is a feature of modern browsers that allows websites to control their XSS auditors.

06. NO RATE LIMIT ON FORGOT PASSWORD PAGE	
Description	
This can be used to send an unlimited number of forgot password requests to any random email.	
Affected Resource / Parameter	**Severity**
https://csper.io/	**MEDIUM**
Impact / Consequences	
There is no rate limit enabled for the "Old Password" field on changing password on your website. A malicious minded user can continually try to brute force an account password. If a user forgets to logout an account in some public computer, then the attacker is able to know the correct password, and also able to change the password to a new one by inputting a large number of payloads.	
Recommendations	
✓ Monitoring API activity against your rate limit. ✓ Catching errors caused by rate limiting. ✓ Reducing the number of requests. ✓ Extra precautions are taken with login, otp, vouchers etc.	
Tools Used	**References**
Burp-suite	https://portswigger.net/burp
CWE	**OWASP Top 10**
307	WSTG-ATHN-09
Proof of Vulnerability	

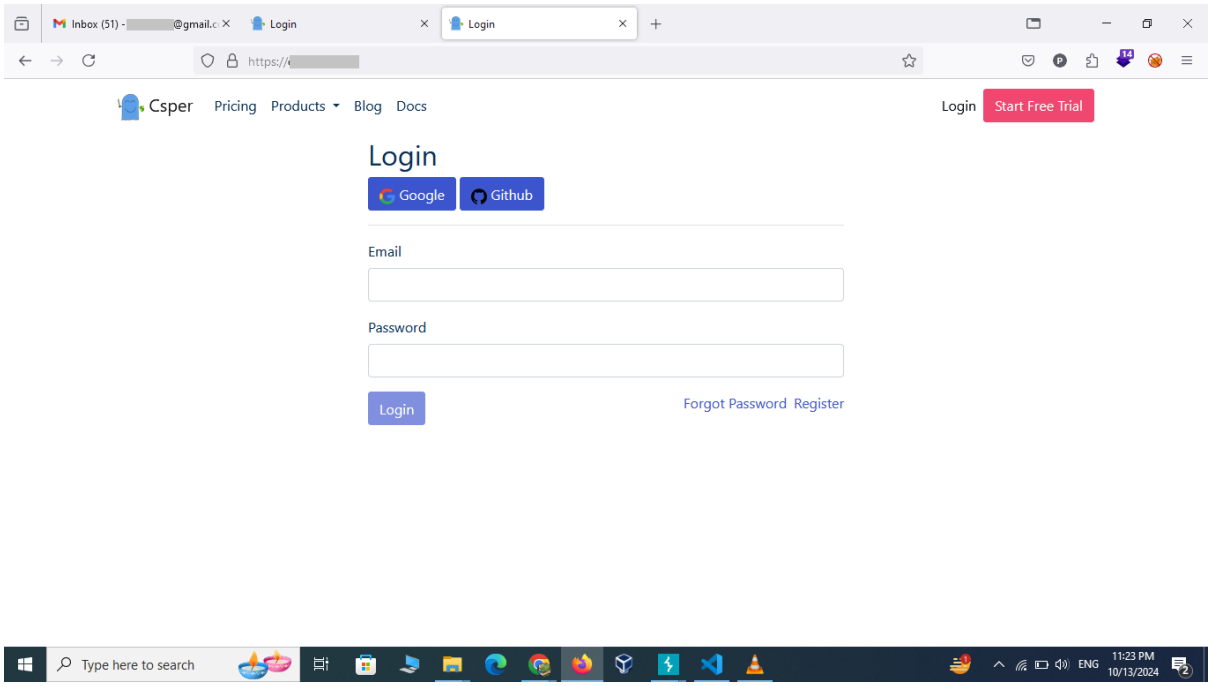
→ Here you can see the website of openbugbounty that is use for vulnerability testing.



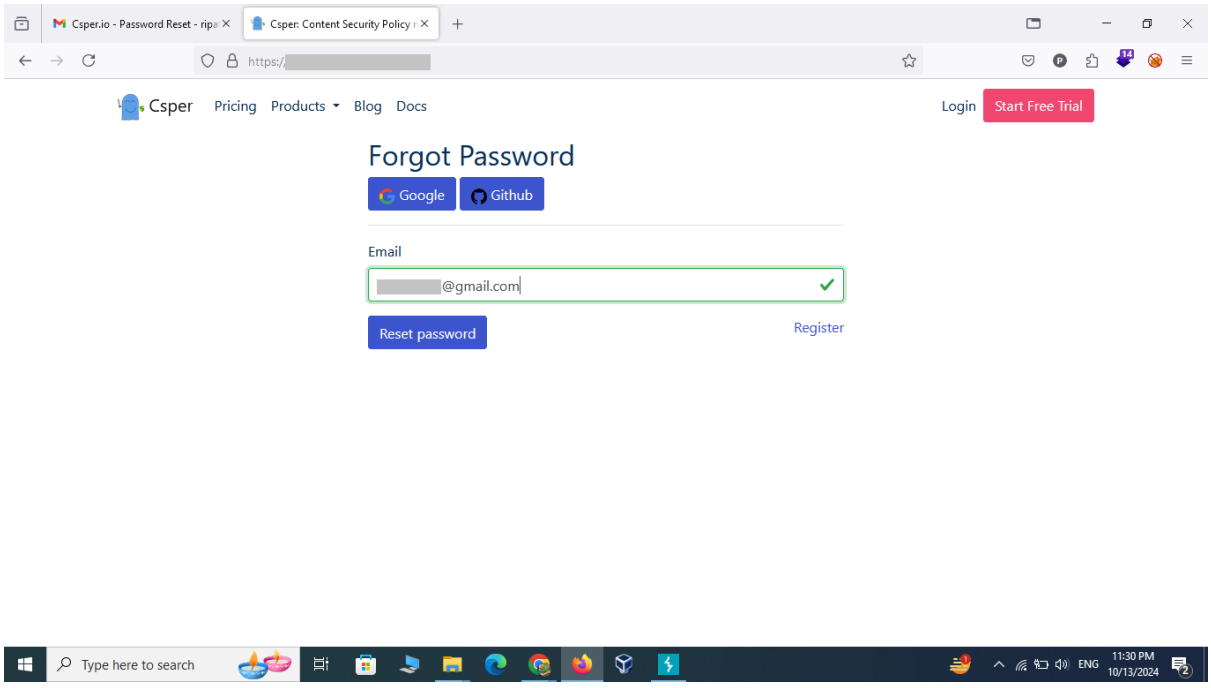
- Step 1: - Visit the website.



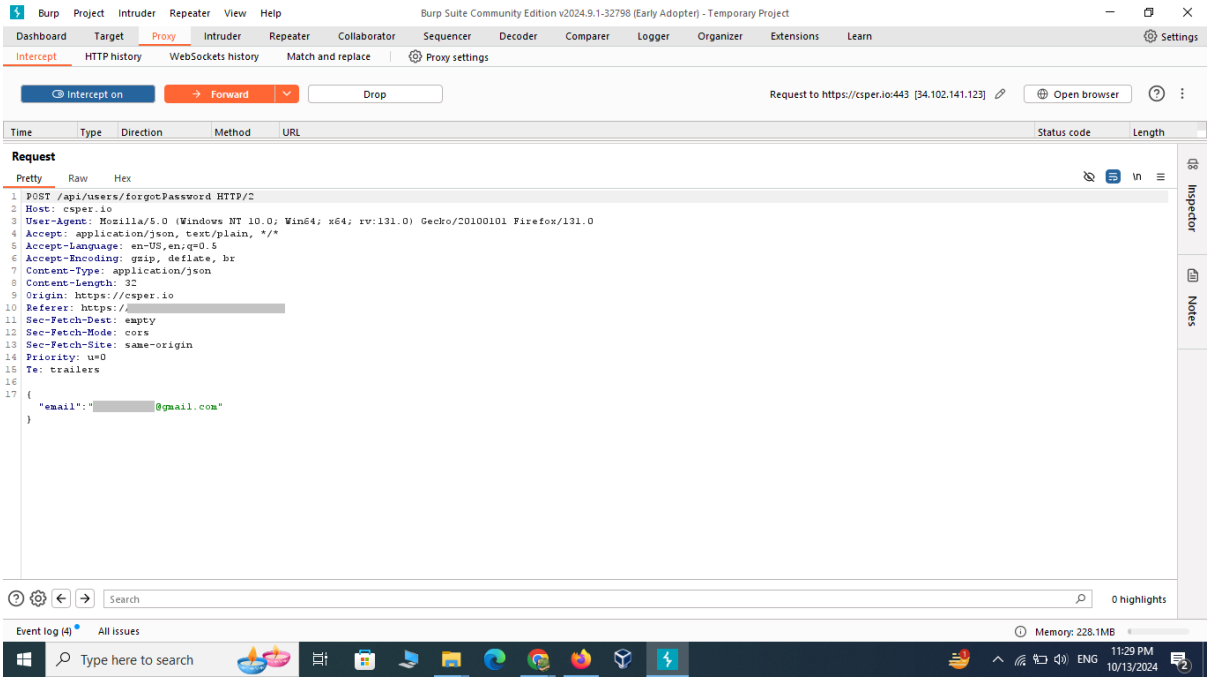
- Step 2: - Here you can see the login page and click on forgot password.



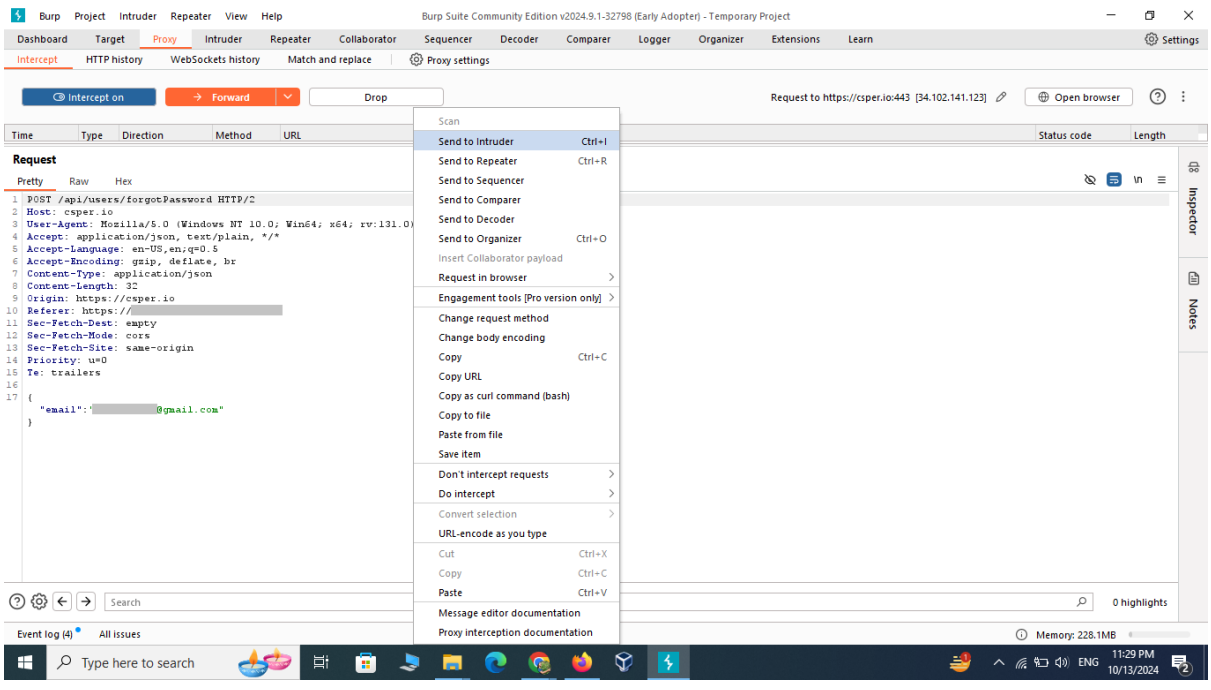
- Step 3: - Here, we write the mail-id and capture the request in burp suite while click on Reset password.



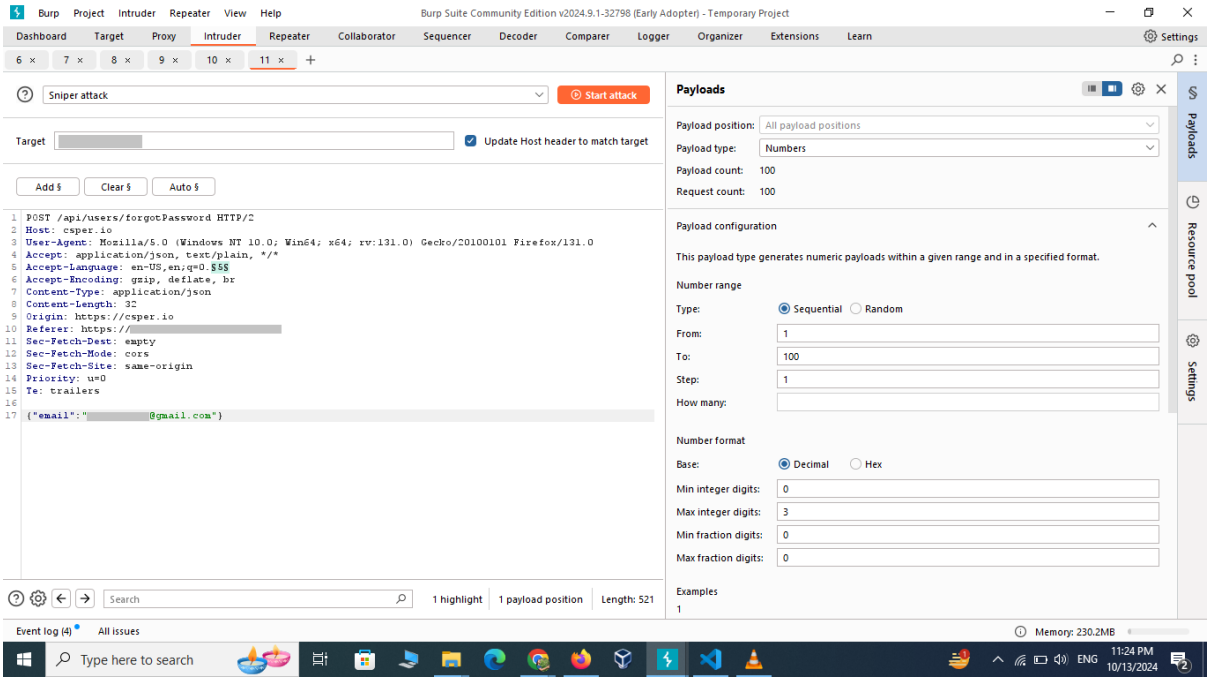
- Step 4: - Here you can see the reset password request.



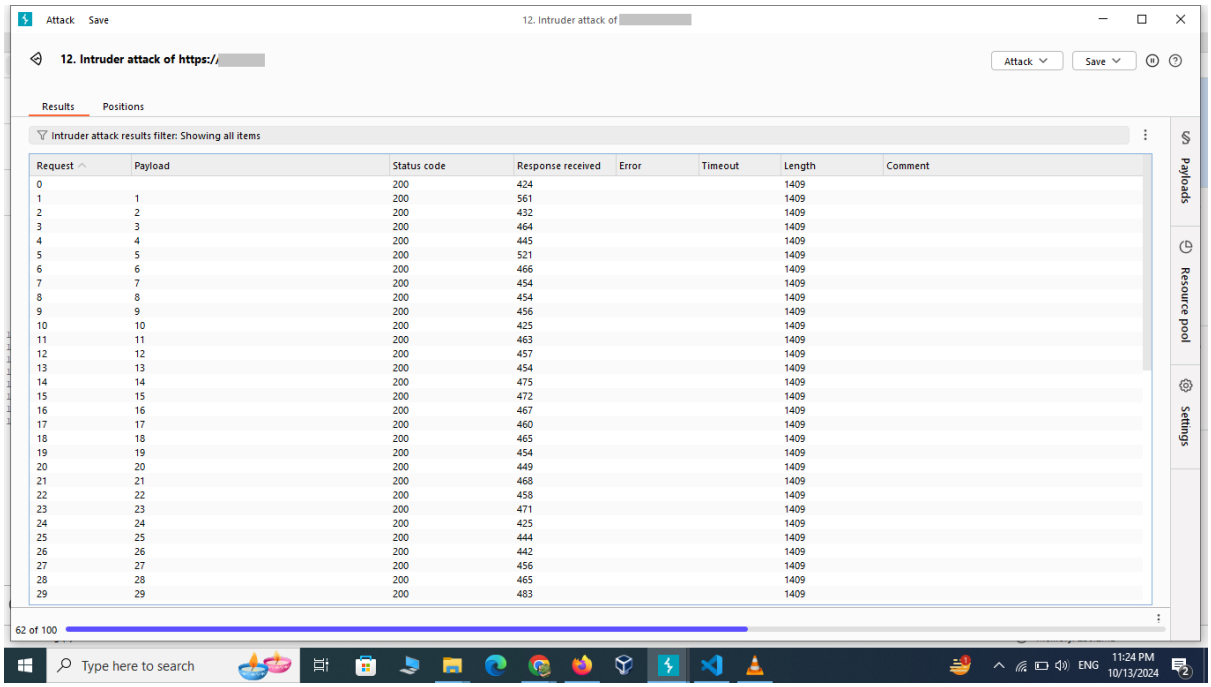
- Step 5: - Here I send the request to Intruder.



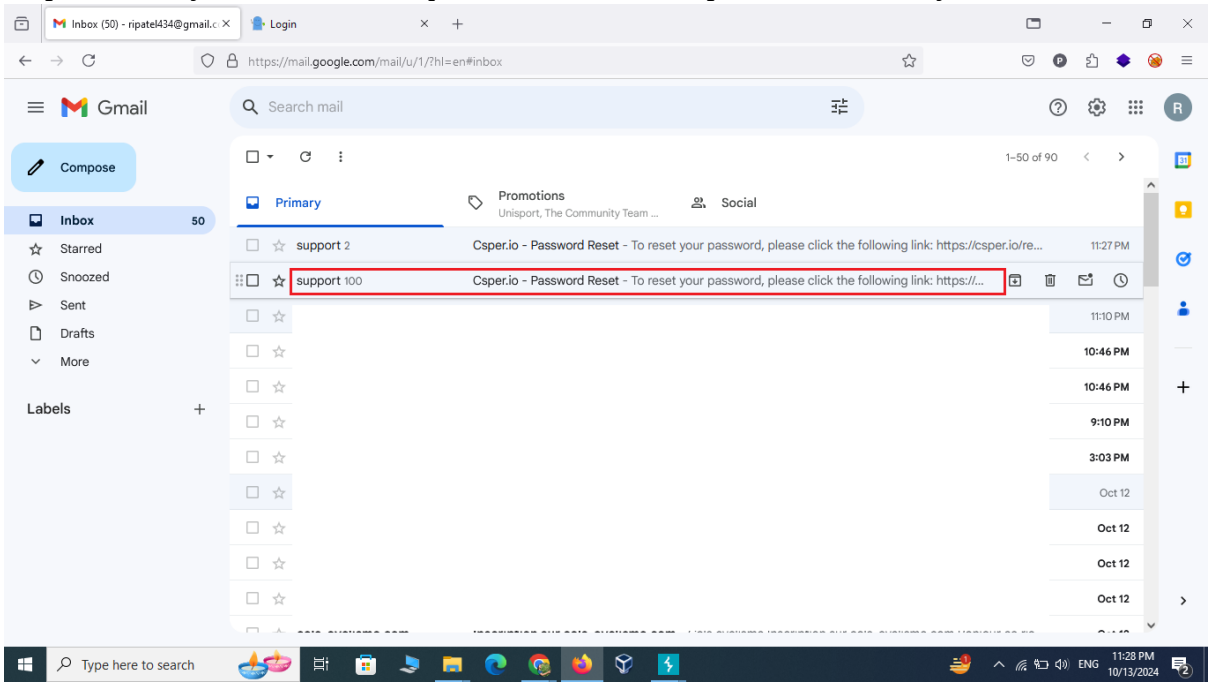
- Step 6: - Here you can see the payloads in which I select the Numbers and set the range from 1 to 100 which is send in mail for reset password for 100 times. And I click on start attack button.



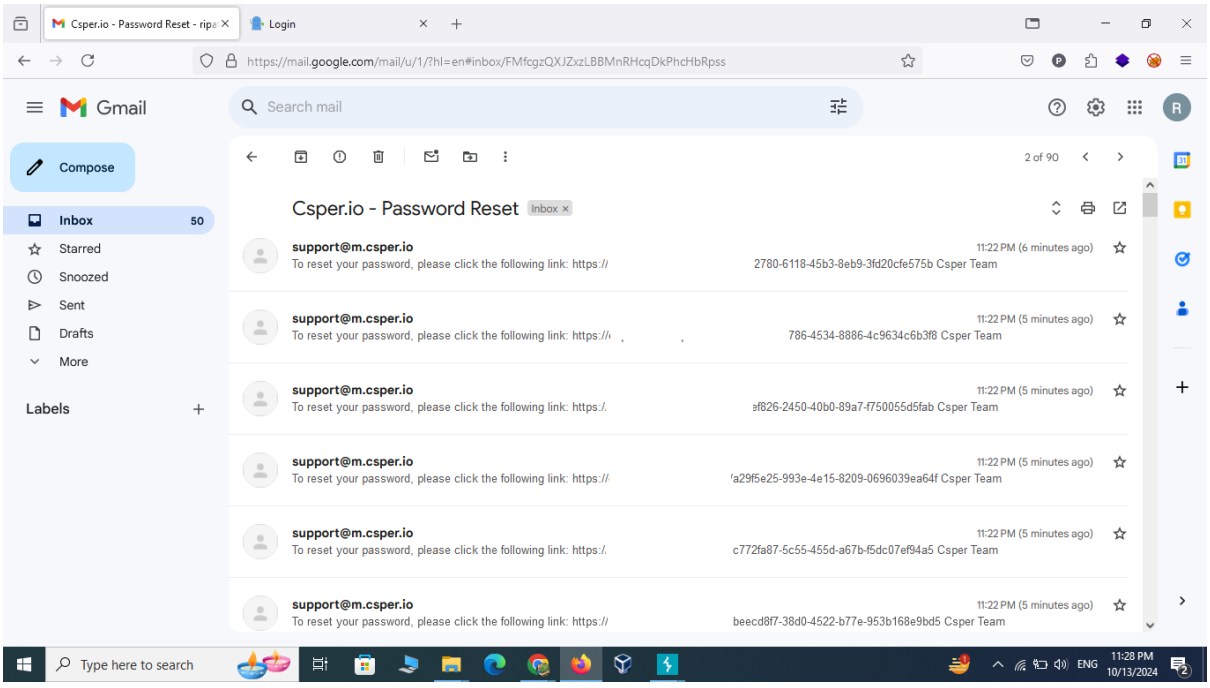
- Step 7: - Here you can see the start attack and send the request on that mail.



- Step 8: - Here you can see the request in mail for reset password for many times.



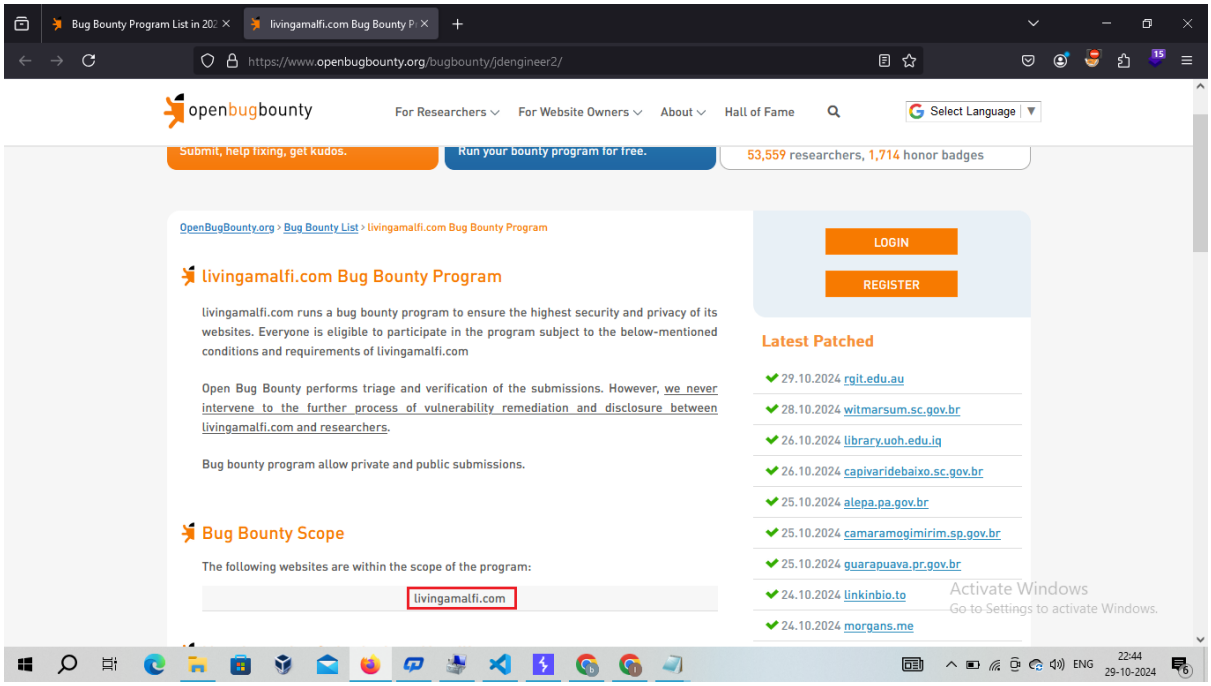
- Step 9: - See the all request Here for Reset password.



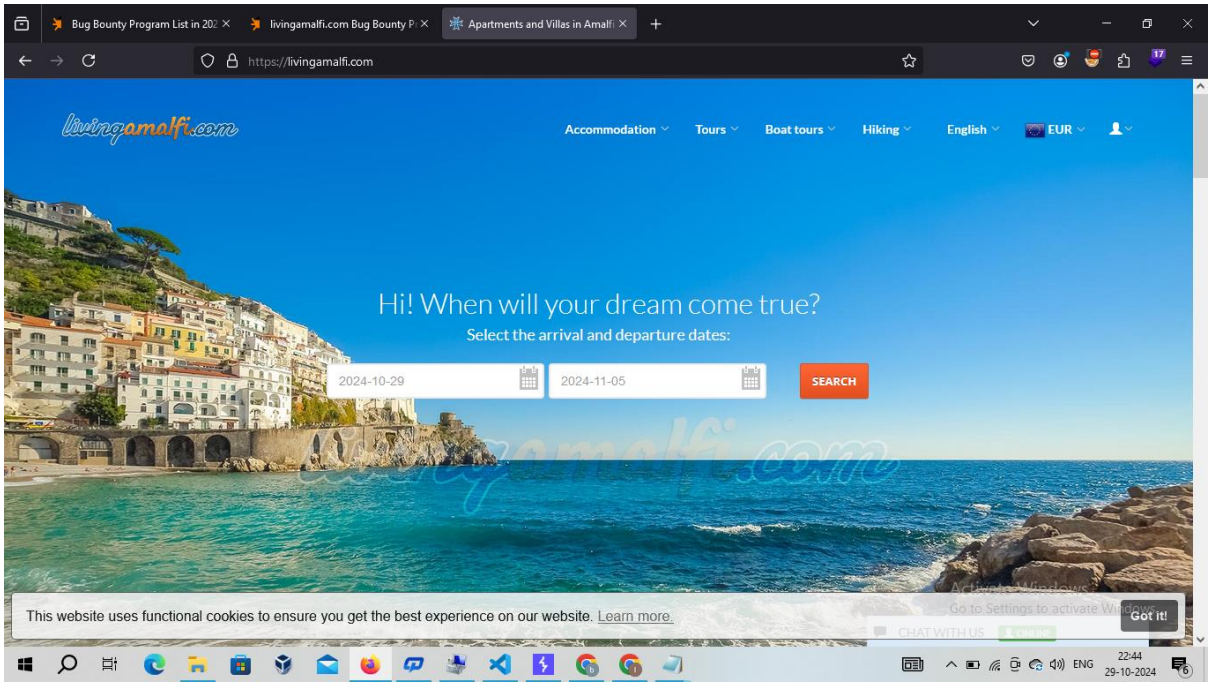
Observation
One or more password validation mechanisms were designed or configured in a way that did not offer sufficient protection to prevent password guessing attacks.

07. CLEAR TEXT PASSWORD SUBMISSION	
Description	
Password is transmitted in a non-encrypted form during authentication processes.	
Affected Resource / Parameter	**Severity**
https://livingamalfi.com/	**MEDIUM**
Impact / Consequences	
Once the attacker knows the credentials of the victim, an attacker can be able to access victims account and could perform malicious activity.	
Recommendations	
It is recommended to use asymmetric encryption to encrypt the sensitive parameters to prevent from being modified or exposed in plain text.	
Tools Used	**References**
Burp-suite	https://portswigger.net/burp
CWE	**OWASP Top 10**
319	A3:2017-Sensitive Data Exposure
Proof of Vulnerability	

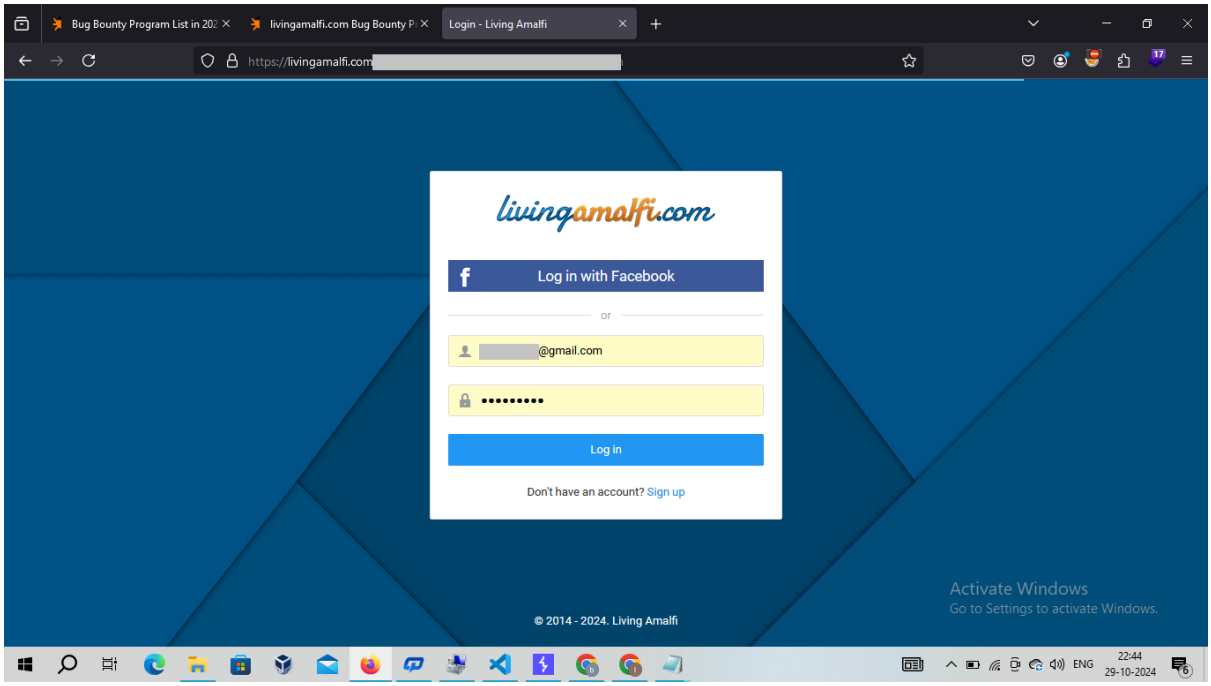
→ Here you can see the website of openbugbounty that is use for vulnerability testing.



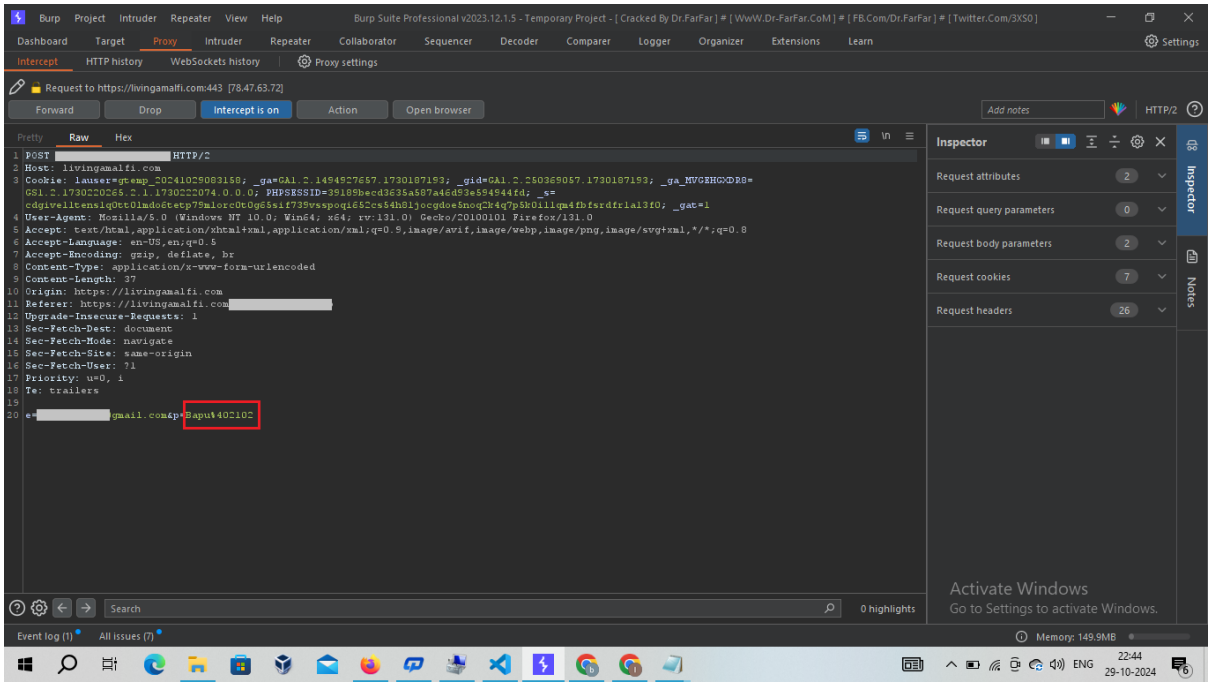
- Step 1: - Visit the website.



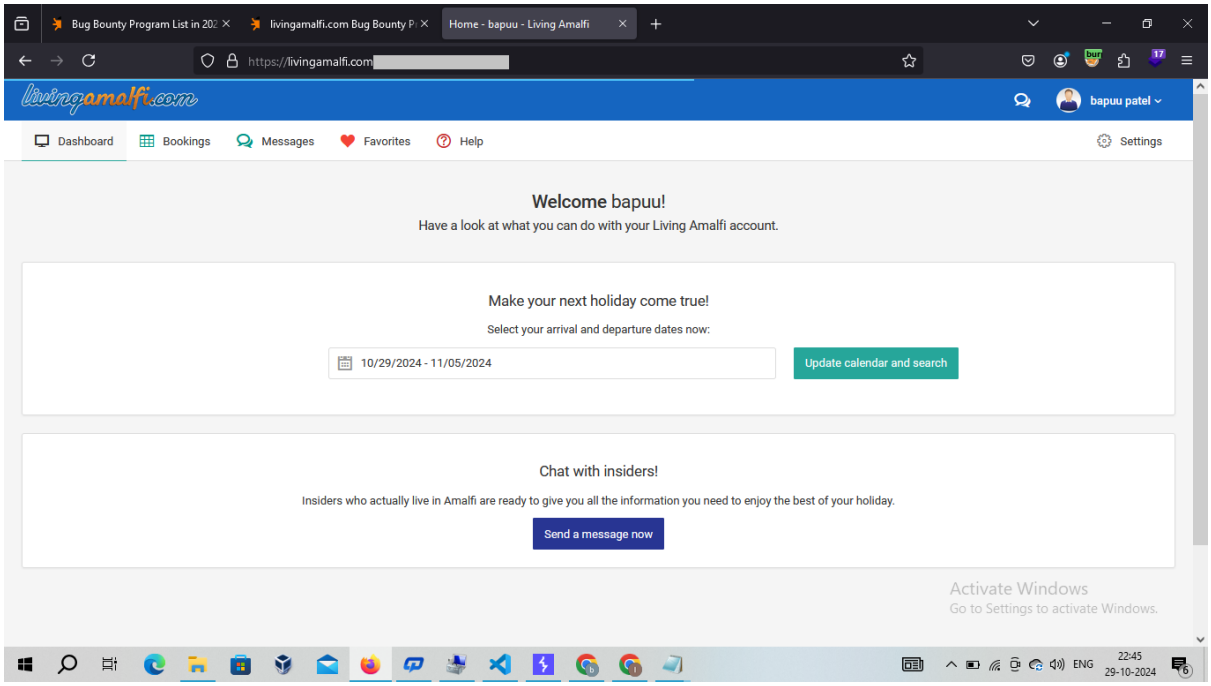
- Step 2: - Here we are log in with email and password.



- Step 3: - Here we are capture the request of log in page. And see password will not be in encrypted form.



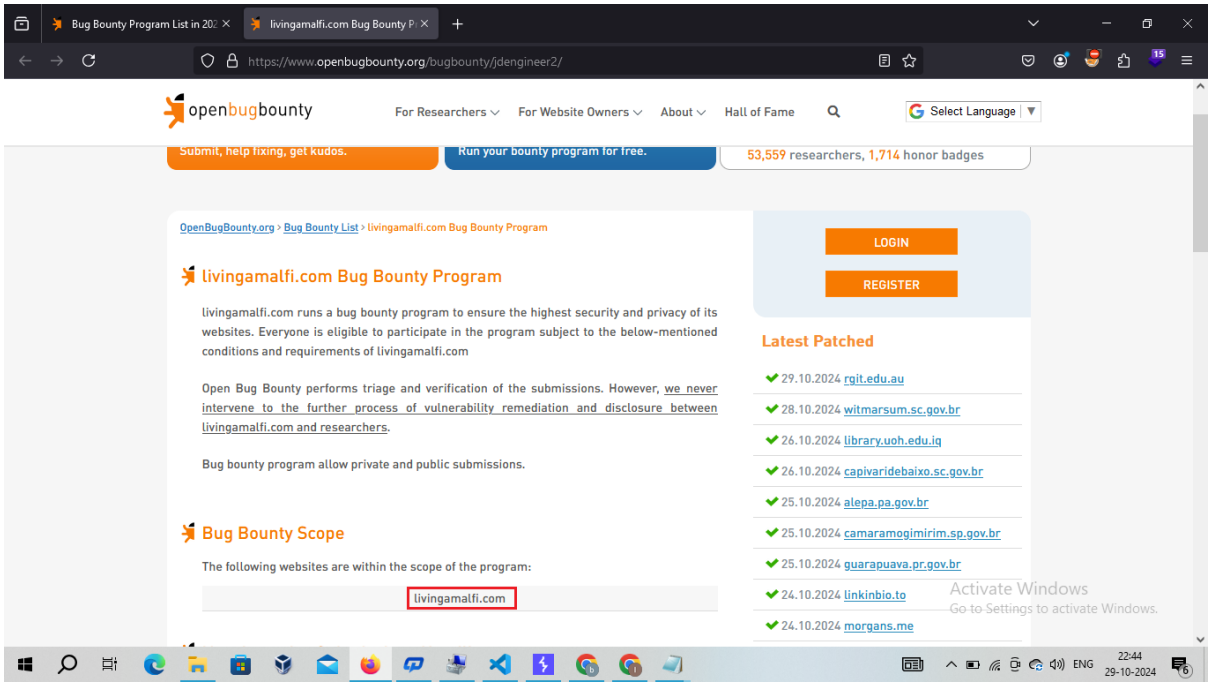
- Step 4: - Here you can see the website is successfully log in.



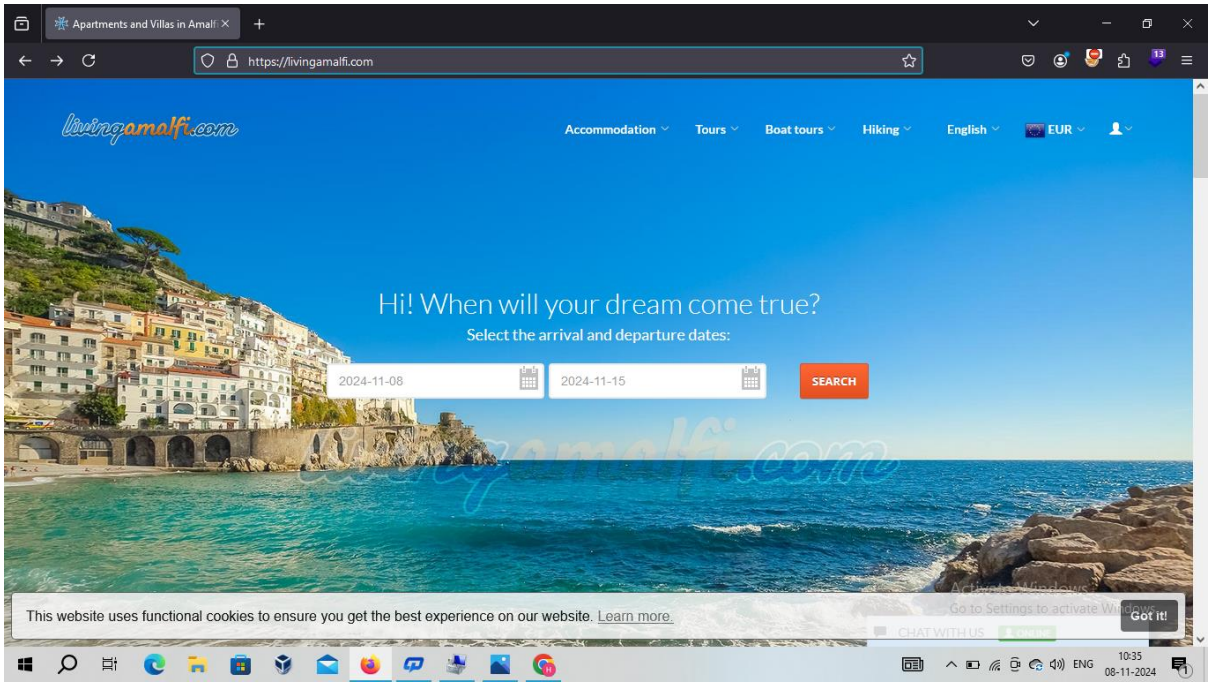
Observation
It was observed that application transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors. To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer.

08. CLICK JACKING	
Description	
It is attack that tricks a user into clicking a webpage element which is invisible.	
Affected Resource / Parameter	**Severity**
https://livingamalfi.com/	**MEDIUM**
Impact / Consequences	
✓ Click jacking is a vulnerability through which users are tricked to click some buttons or UI elements of the parent page, but they are clicking something in the vulnerable web application, because that is being hidden behind the UI of the parent page. ✓ The User assumes that they are entering their information into usual form but they are entering it in fields, Hackers will target passwords, credit card numbers and any other valuable data they can exploit.	
Recommendations	
✓ Properly setting authentication cookies with Same Site = Strict, unless they explicitly need None.	
Tools Used	**References**
Vs code	https://code.visualstudio.com/
CWE	**OWASP Top 10**
1021	A6:2017 – Security Misconfiguration
Proof of Vulnerability	

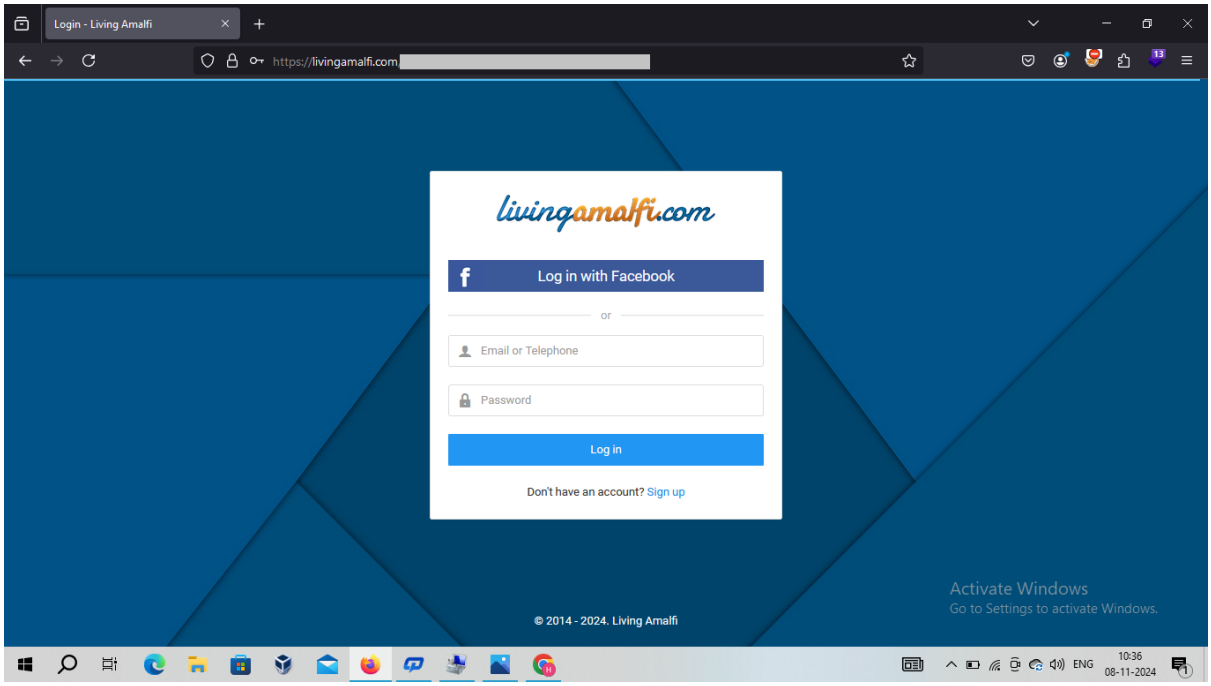
→ Here you can see the website of openbugbounty that is use for vulnerability testing.



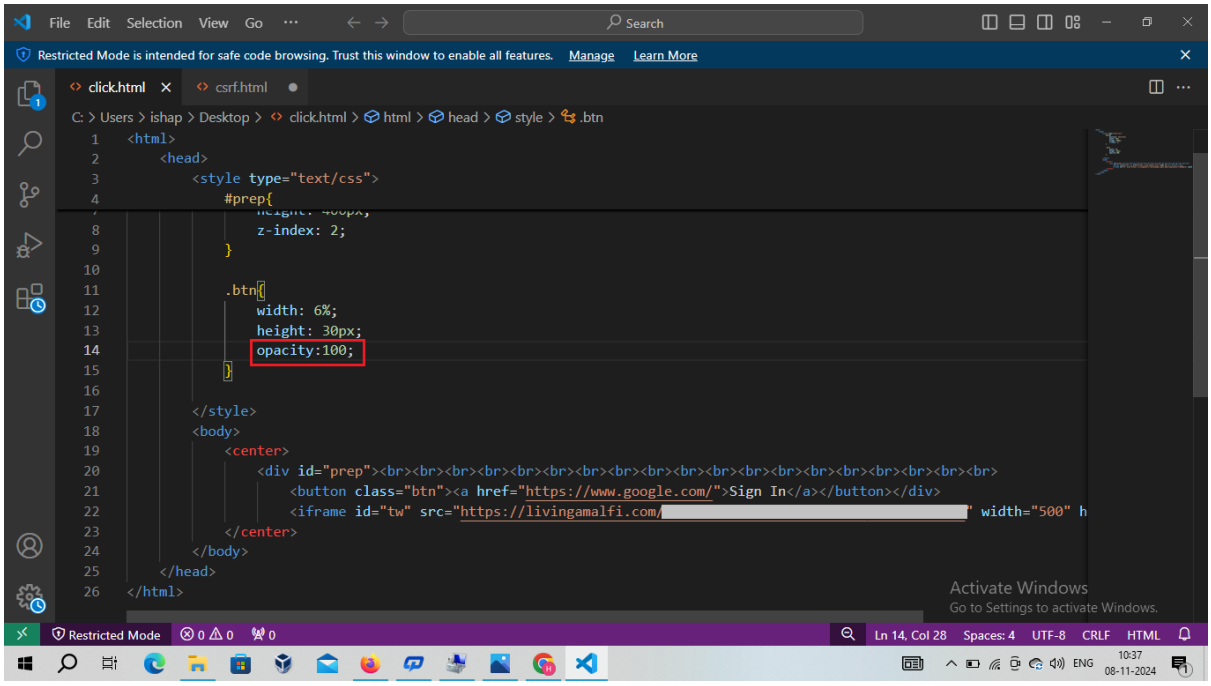
- Step 1: - visit the website.



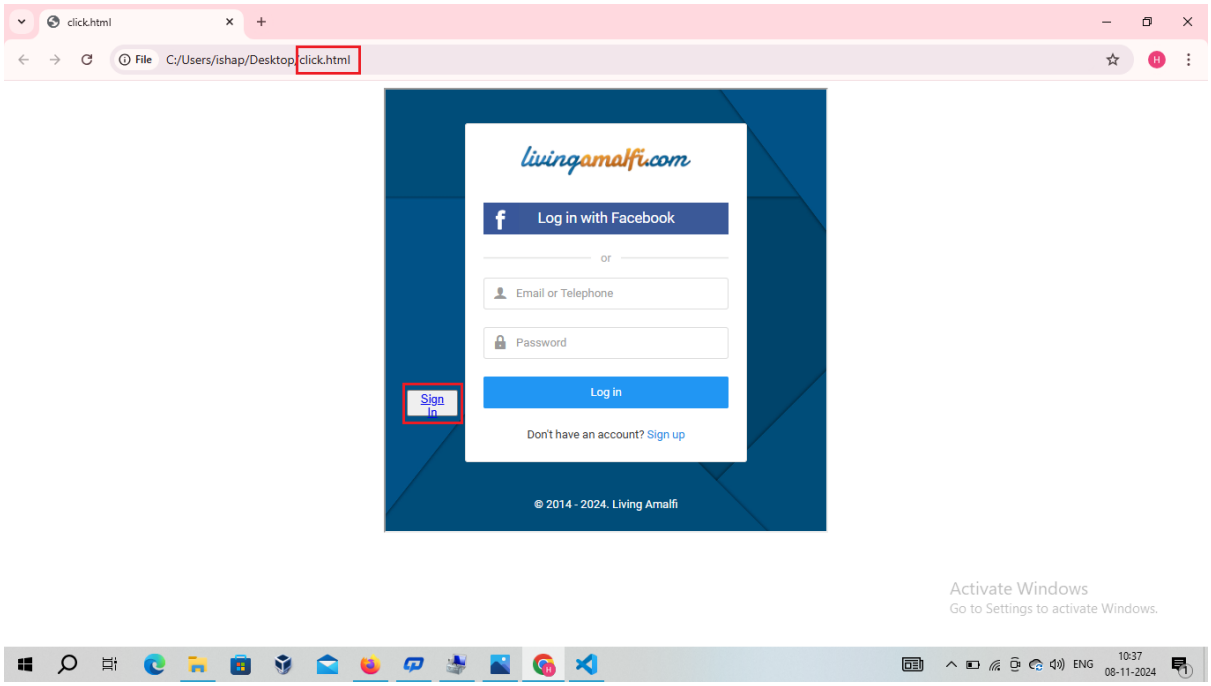
- Step 2: - Here you can see the log in page.



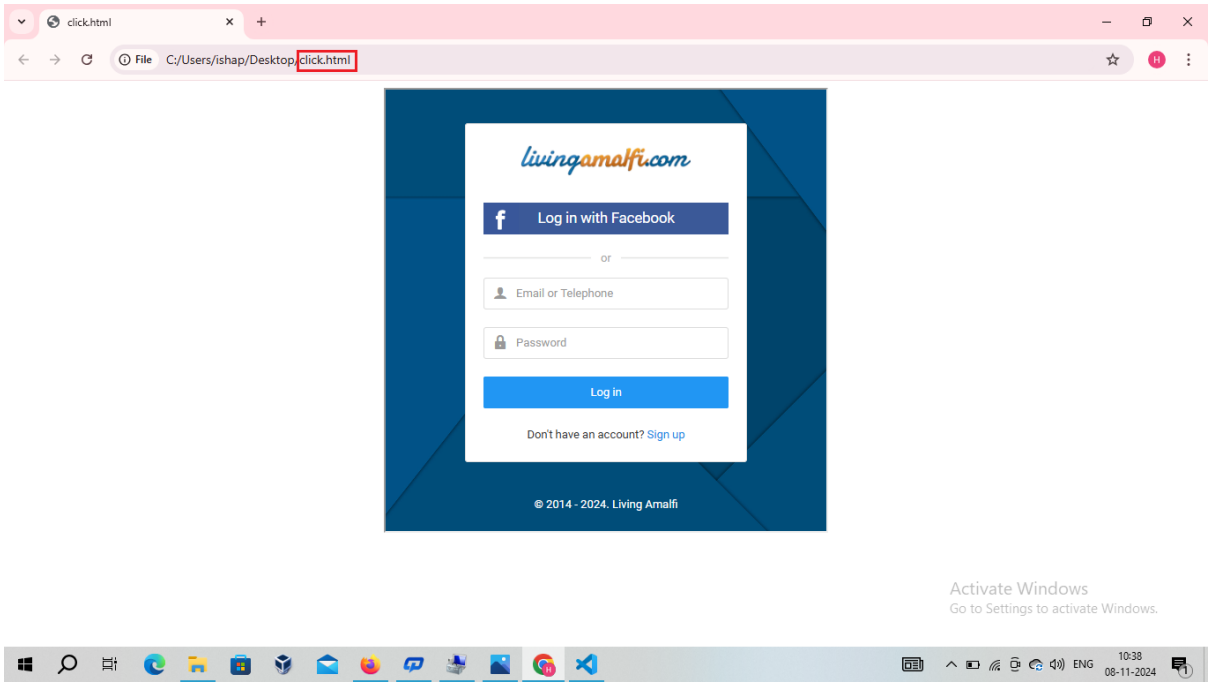
- Step 3: - Here enter the script like this.



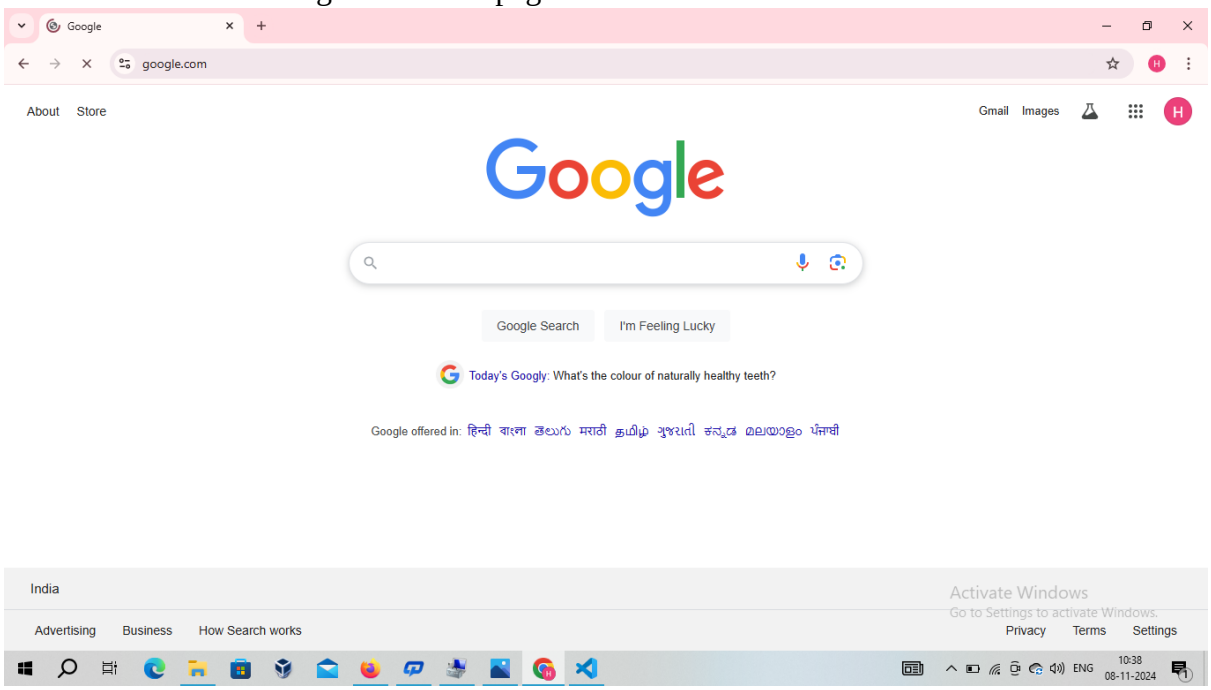
- Step 4: - After This script run and see its looks like this because it is i-frame. You can see the sign in button because the opacity is 100.



- Step 5: - Here we have changed the opacity by 0.0001. So, it would not be visible to the user.



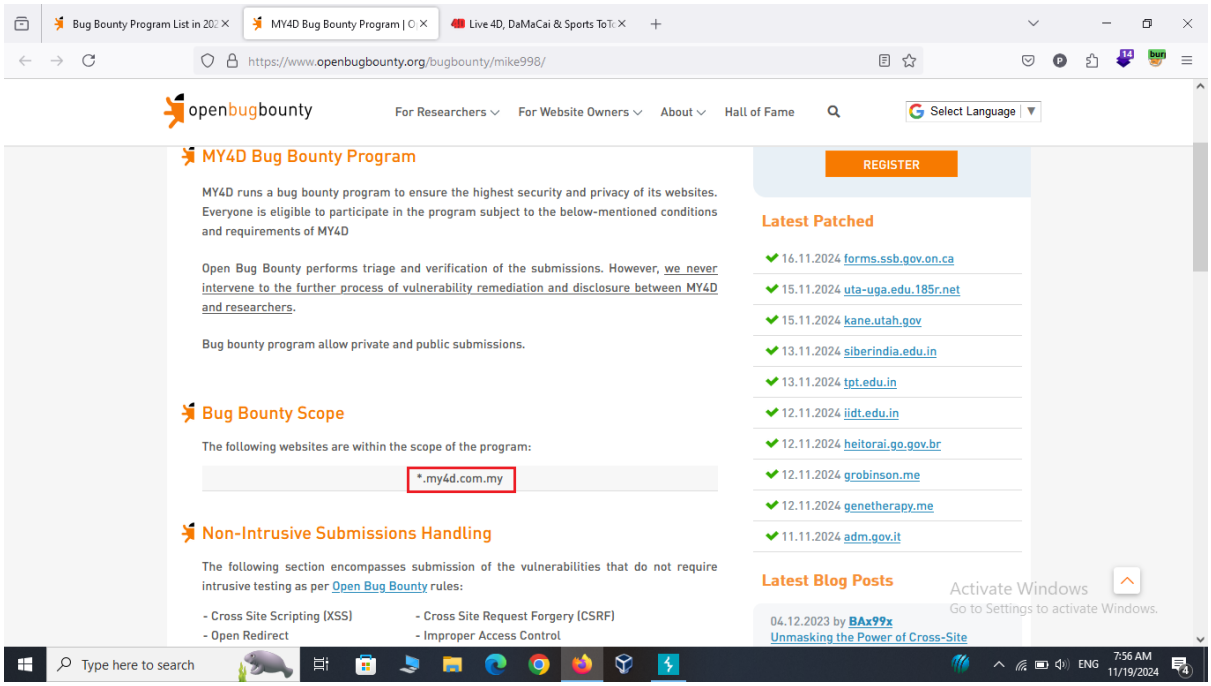
- Step 6: - As a result, when the user will enter their information and click on Login they will be redirected to the given created page.



Observation
It was observed that the server didn't return an `X-Frame-Options` header which means that this website could be at risk of a clickjacking attack. Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

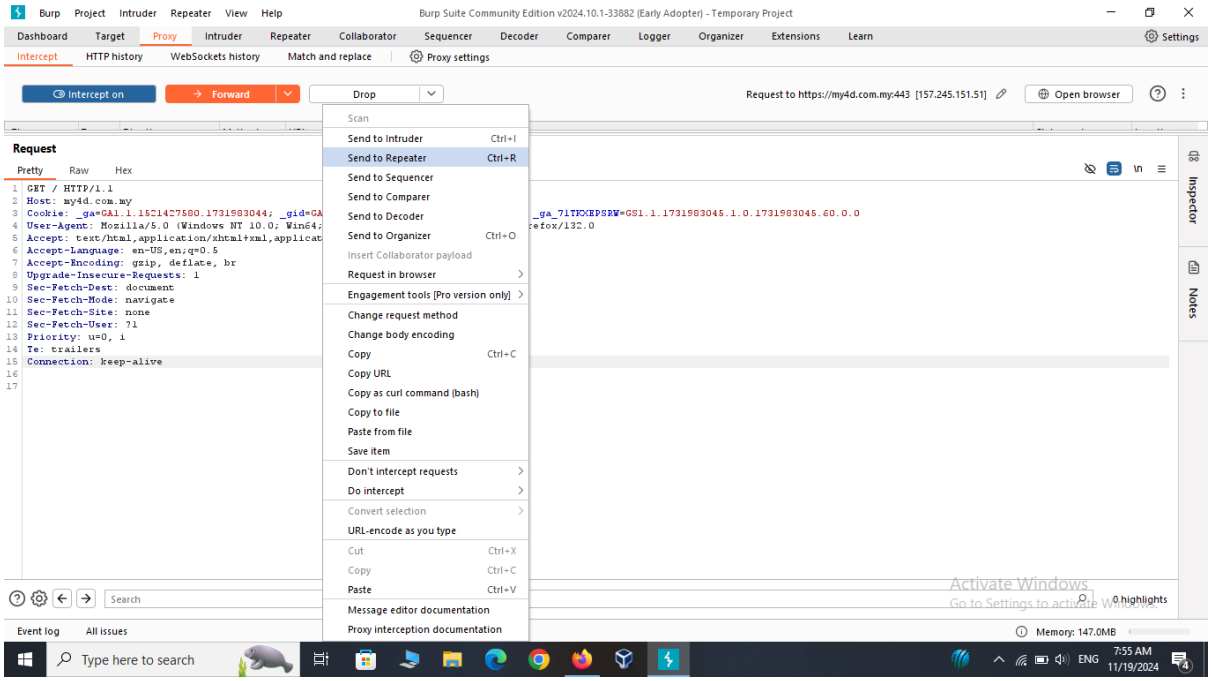
09. SERVER VERSION DISCLOSURE	
Description	
The Server header describes the server application that handled the request.	
Affected Resource / Parameter	**Severity**
https://my4d.com.my/	**LOW**
Impact / Consequences	
✓ An attacker can use the disclosed the information to harvest specific vulnerability for the version identified.	
Recommendations	
✓ Configure your web server to prevent information leakage from the SERVER header of its HTTP response.	
Tools Used	**References**
Burp-suite	https://portswigger.net/burp
CWE	**OWASP Top 10**
200	-
Proof of Vulnerability	

→ Here you can see the website of openbugbounty that is use for vulnerability testing.

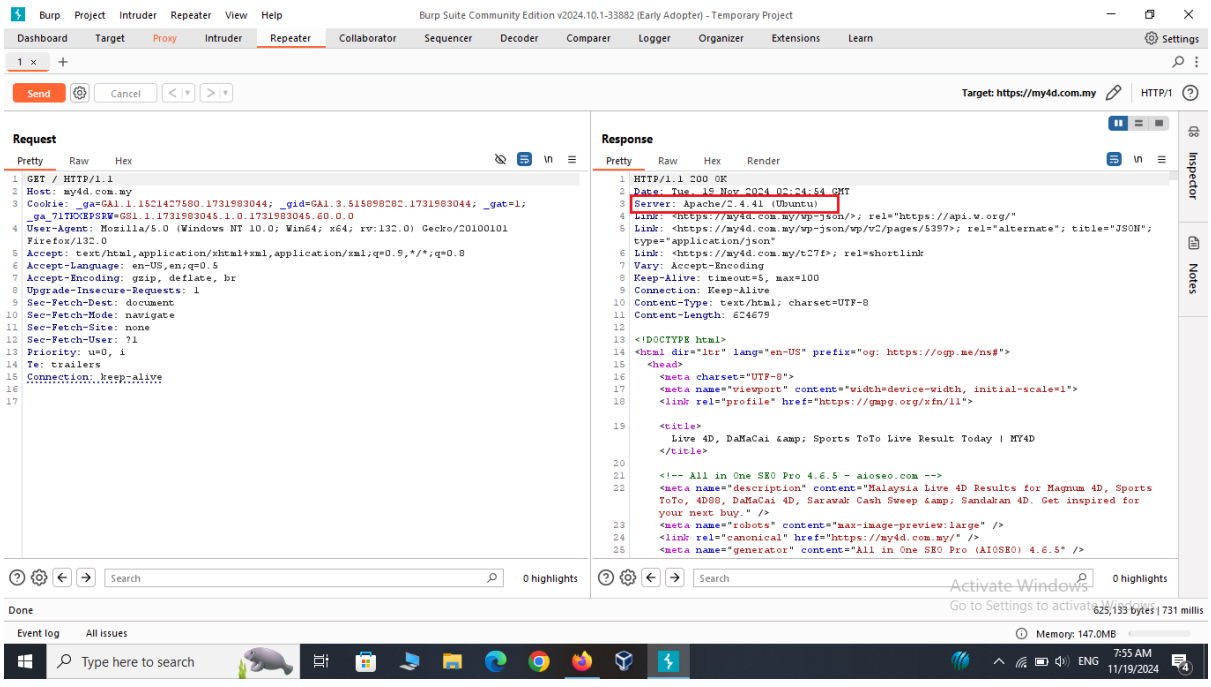


- Step 1: - visit the website.

- Step 2: - Here we have capture the request and sent it to Repeater.



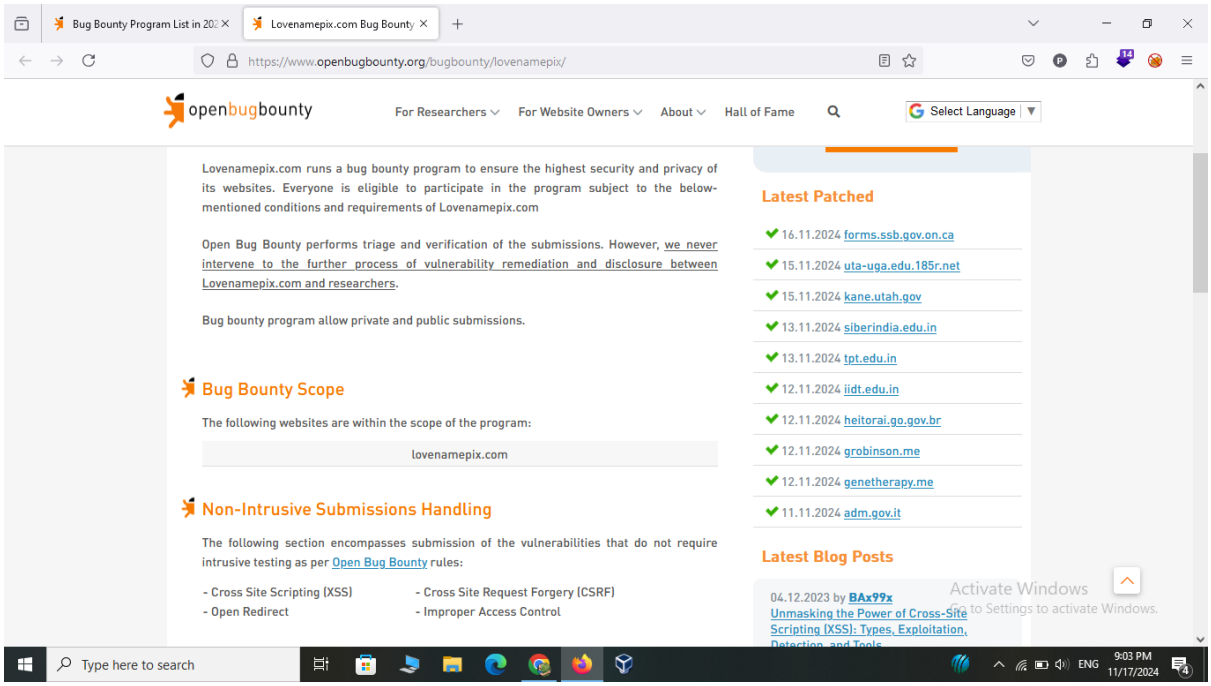
- Step 3: - Now in repeater click on send and then we can see the server version of the site.



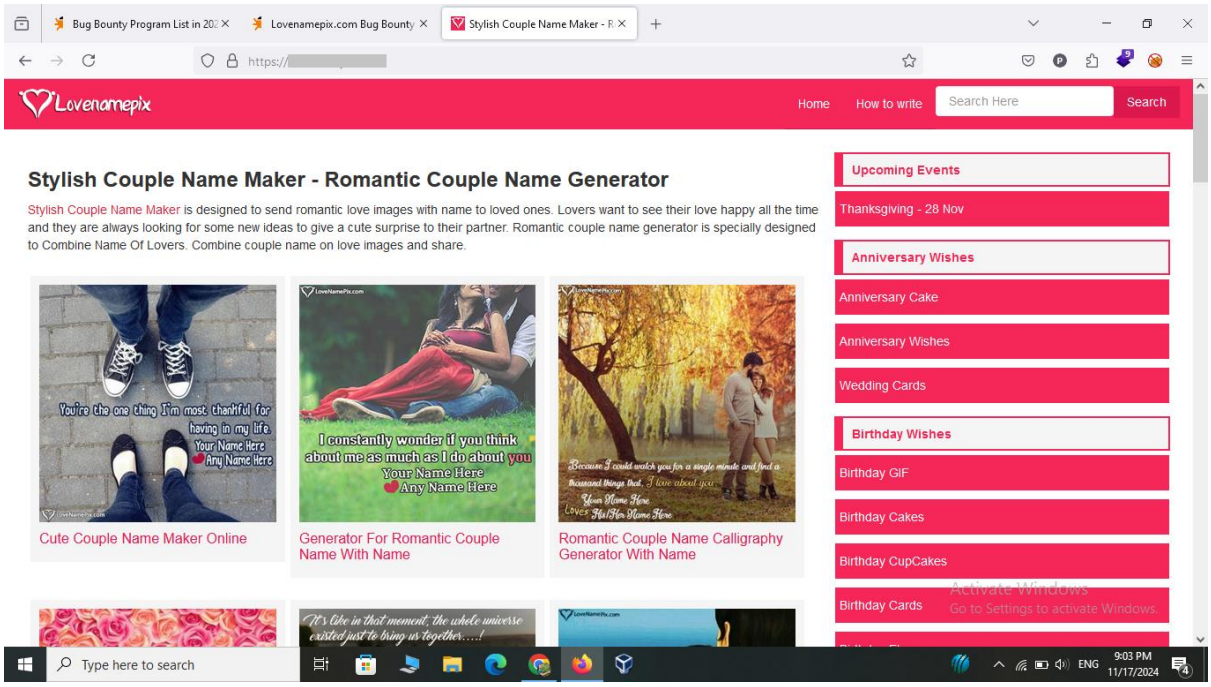
Observation
It was observed that identified a version disclosure in the target web server's HTTP response. This information can help an attacker gain a greater understanding of the systems in use and potentially develop further attacks targeted at the specific version.

10. HTML INJECTION	
Description	
It allows an attacker to inject HTML code into web pages that are viewed by other users.	
Affected Resource / Parameter	**Severity**
https://lovenamepix.com/	**LOW**
Impact / Consequences	
HTML Injection allow an attacker to modify the page. ✓ Using HTML Form, it can steal another person's identity. ✓ Attacker crafts malicious links, including his injected HTML content, and sends it to a user via email. ✓ At the end it can be a very serious vulnerability.	
Recommendations	
Use regular-expressions and filter the character like <, >, /, //, \\, \, etc. so the tag like <a> it cannot be work. ✓ Use Post method mainly to hide the URL and the vulnerable parameter from attacker. ✓ Use the proper sanitizer the input fields. Ex. You can use mysqli_real_escape_string function at database side.	
Tools Used	**References**
Wappalyzer	https://www.wappalyzer.com/
CWE	**OWASP Top 10**
80	A03:2021 – Injection
Proof of Vulnerability	

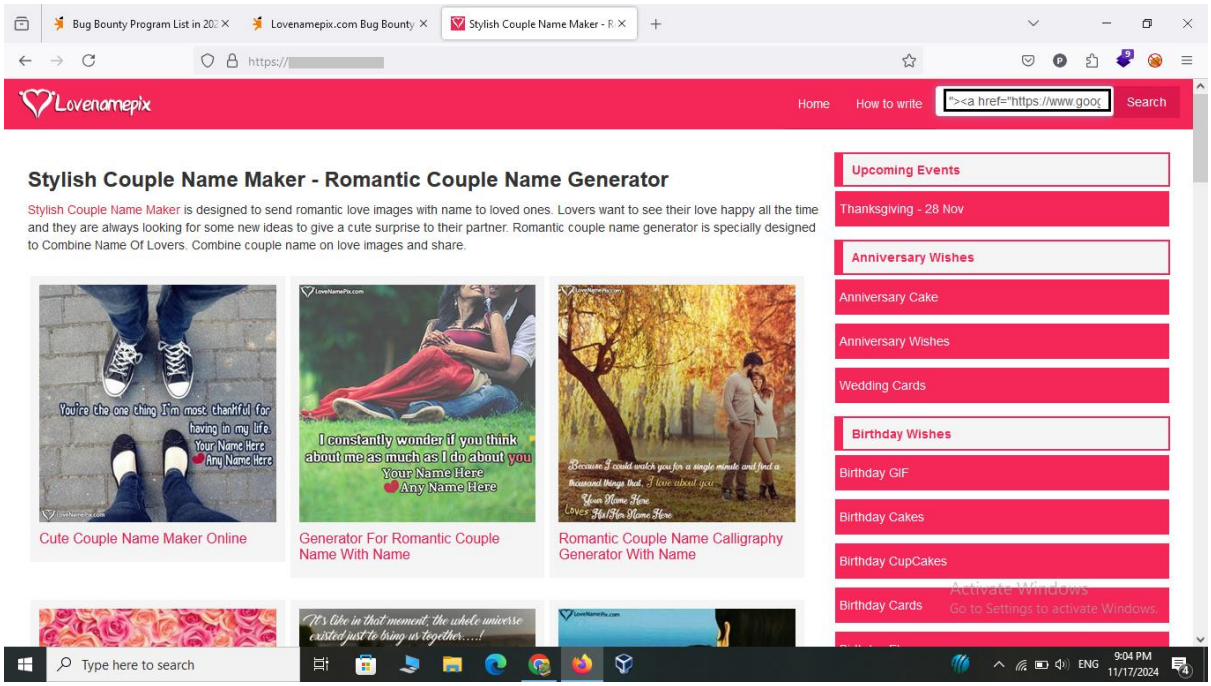
→ Here you can see the website of openbugbounty that is use for vulnerability testing.



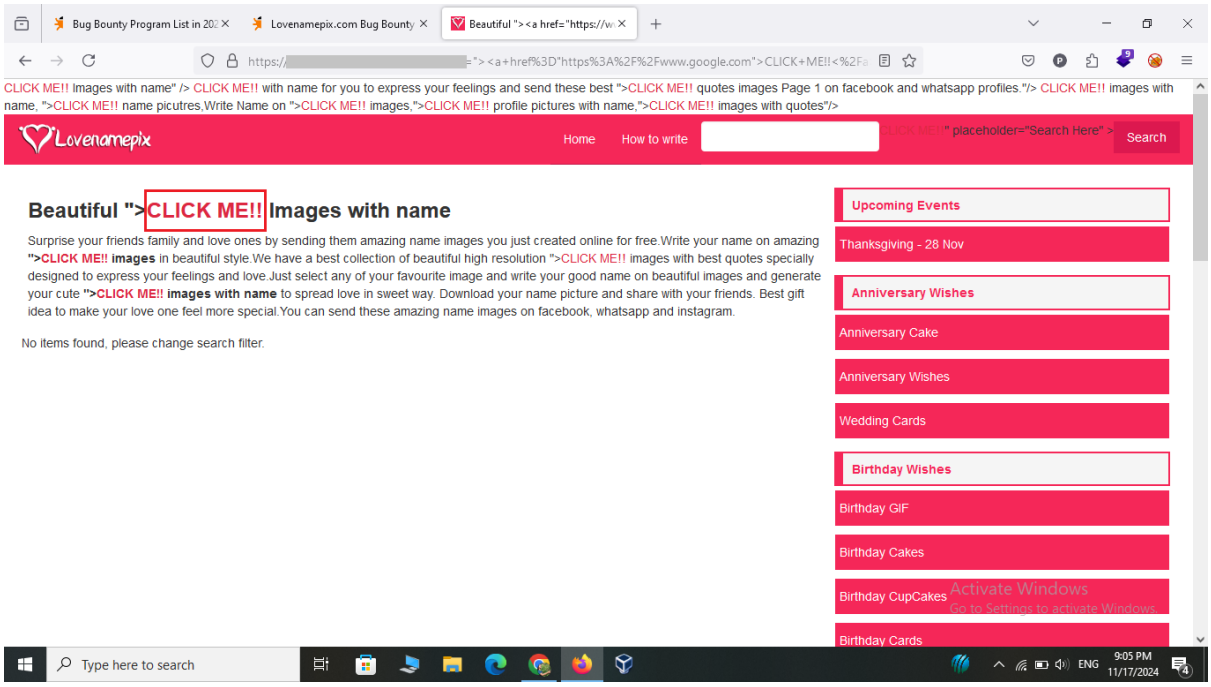
- Step 1: - Visit website.



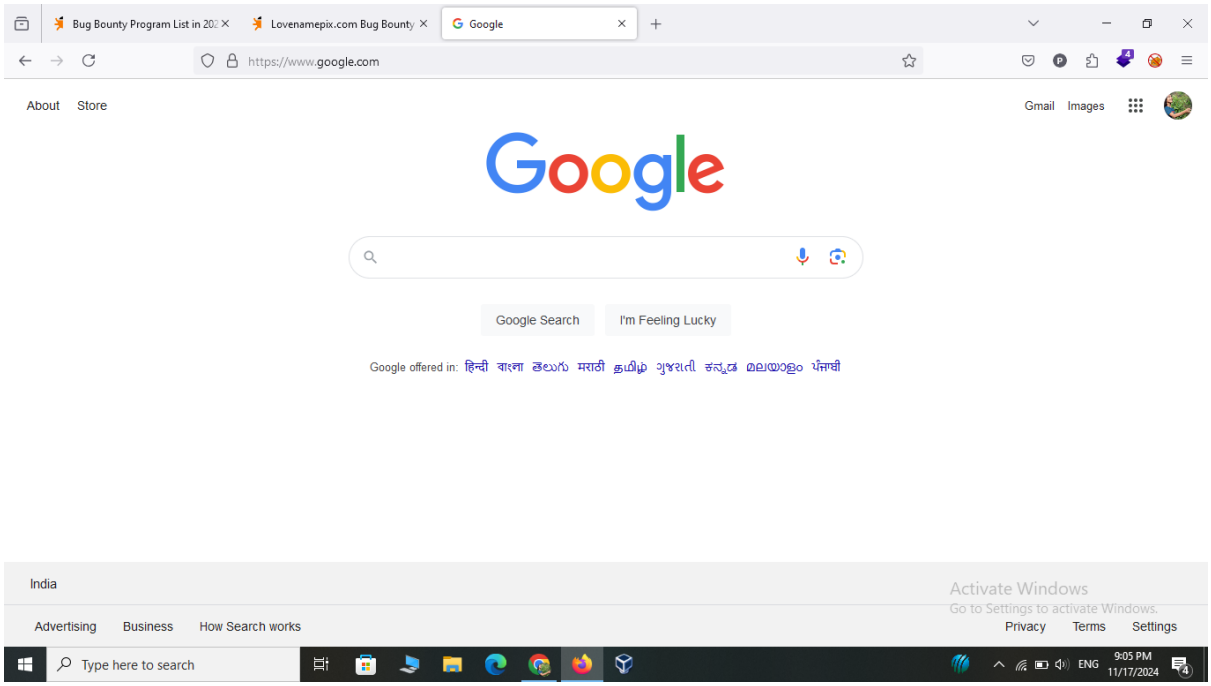
- Step 2: - Here, we are write html code to create link and redirect the page. Payload : **“>CLICK ME**



- Step 3: - Here you can see the CLICK ME!! Link.



- Step 4: - When you click on CLICK ME!! Then you will redirect to the www.google.com.



Observation
It was observed that the web application was vulnerable to HTML Injection. This type of injection attack includes injecting HTML code through the vulnerable parts of the website. The Malicious user sends HTML code through any vulnerable field with the purpose to change the website's design or any information displayed to the user. As the result, the user may see the data sent by the malicious user. Therefore, in general, HTML Injection is just the injection of markup language code to the document of the page.

7. FUTURE SCOPE

- Security testing is a type of software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders.
- Security testing of any system is about finding all possible loopholes and weaknesses of the system that may result into a loss of information, revenue, and repute at the hands of the employees or outsiders of the organization
- Major driving factor for growth of the penetration testing is the ability to provide security to industries from various cyber-attacks, as increasing incidence of cyber-attacks can increase the vulnerability of critical data stored by organizations and adversely impact the revenue.

8. REFERENCE

- Openbugbounty: https://www.openbugbounty.org/
- Burp-suite: https://portswigger.net/burp/communitydownload
- Owasp: https://owasp.org/
- Portswigger lab : https://portswigger.net/web-security/all-labs
- google dork: https://github.com/sushiwushi/bug-bounty-dorks.git