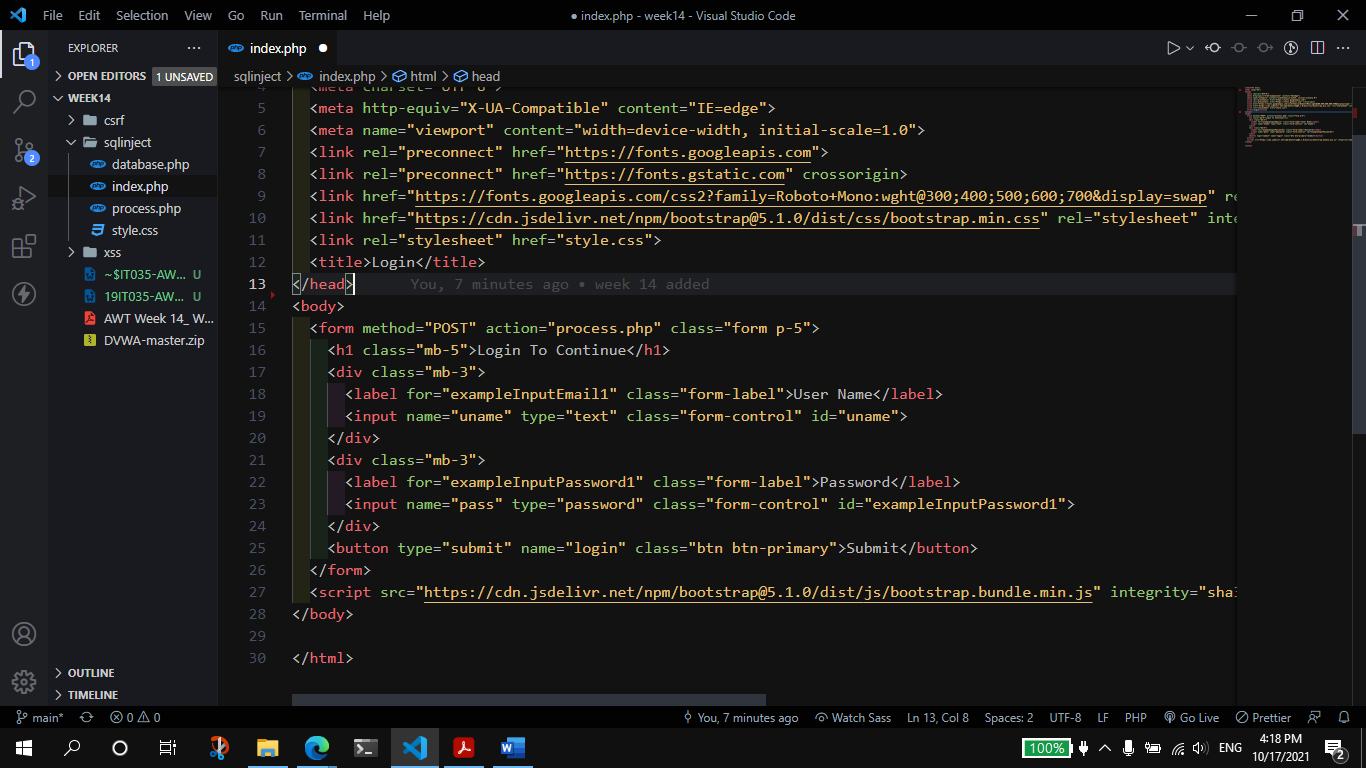
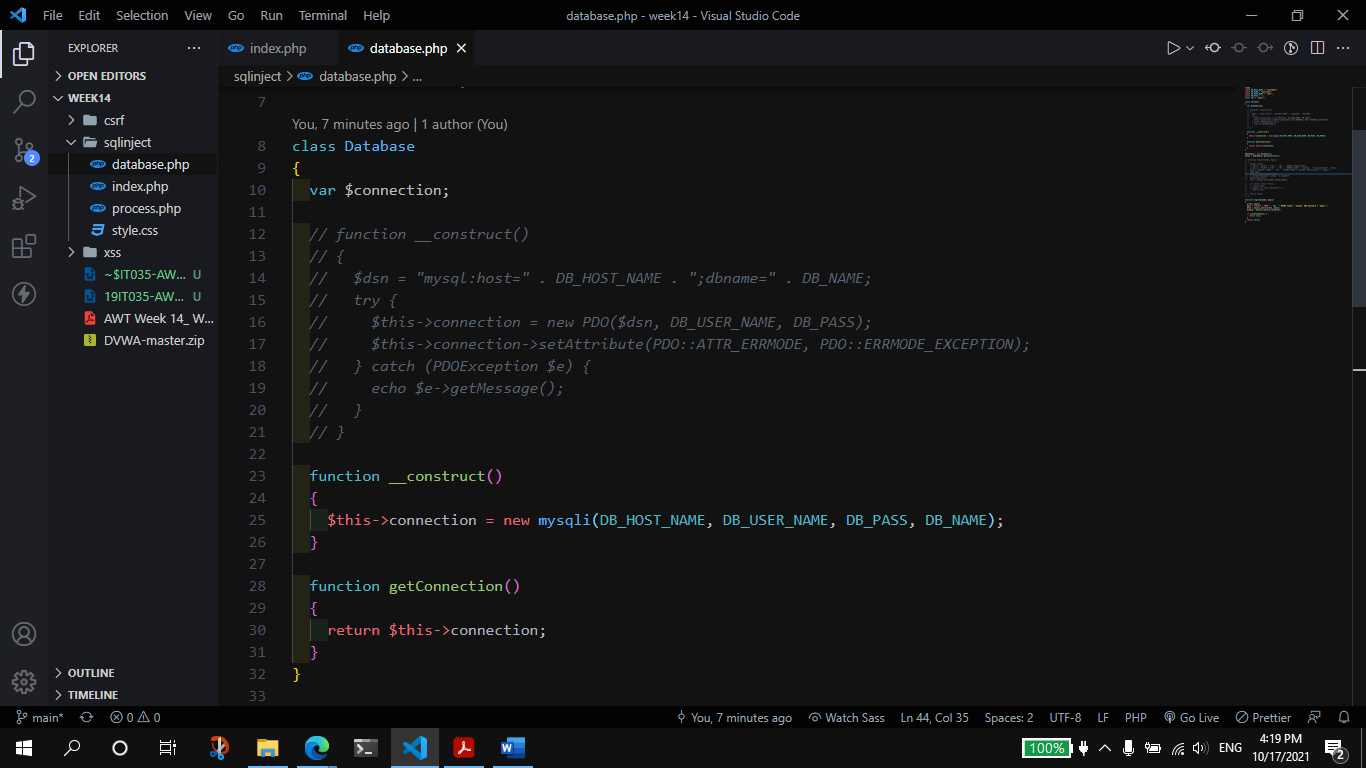
**AWT Practical *Week - 14 Task***

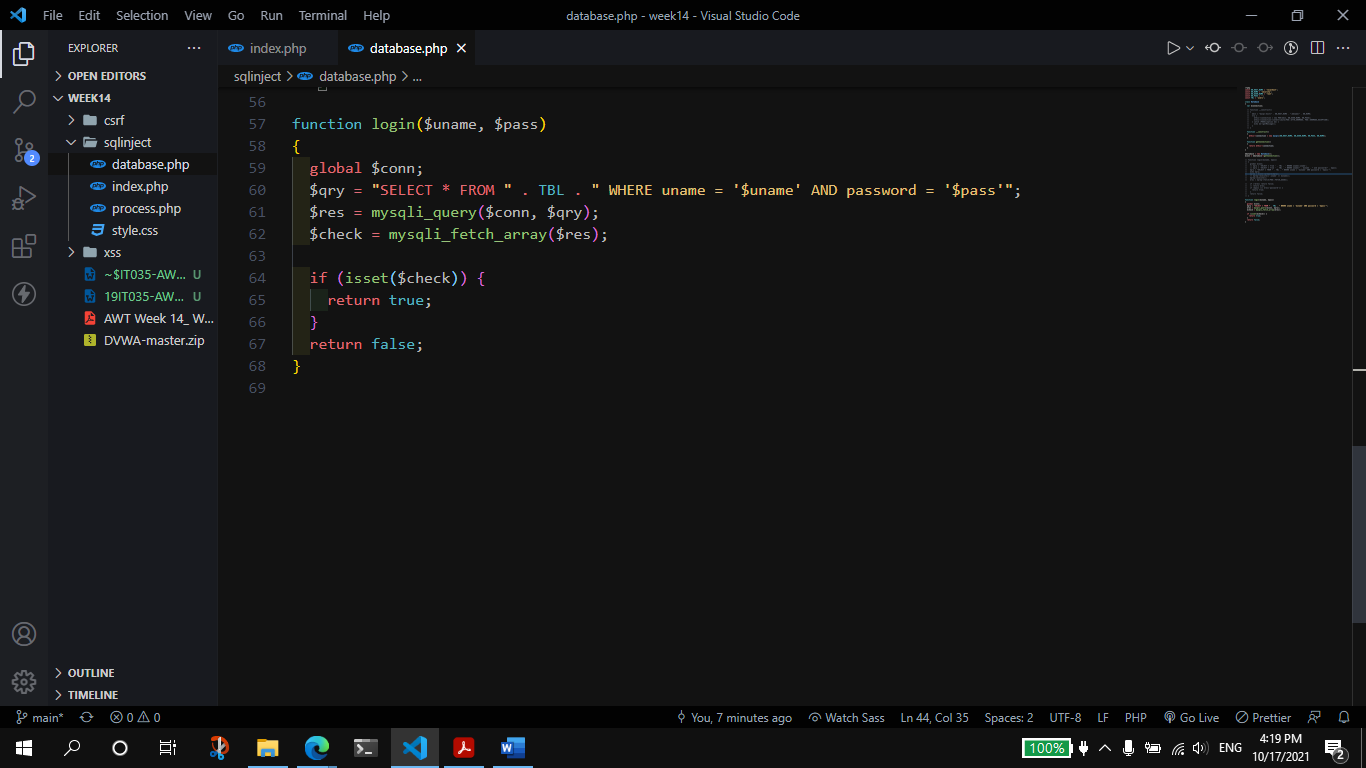
**Web Securities ( PHP )**

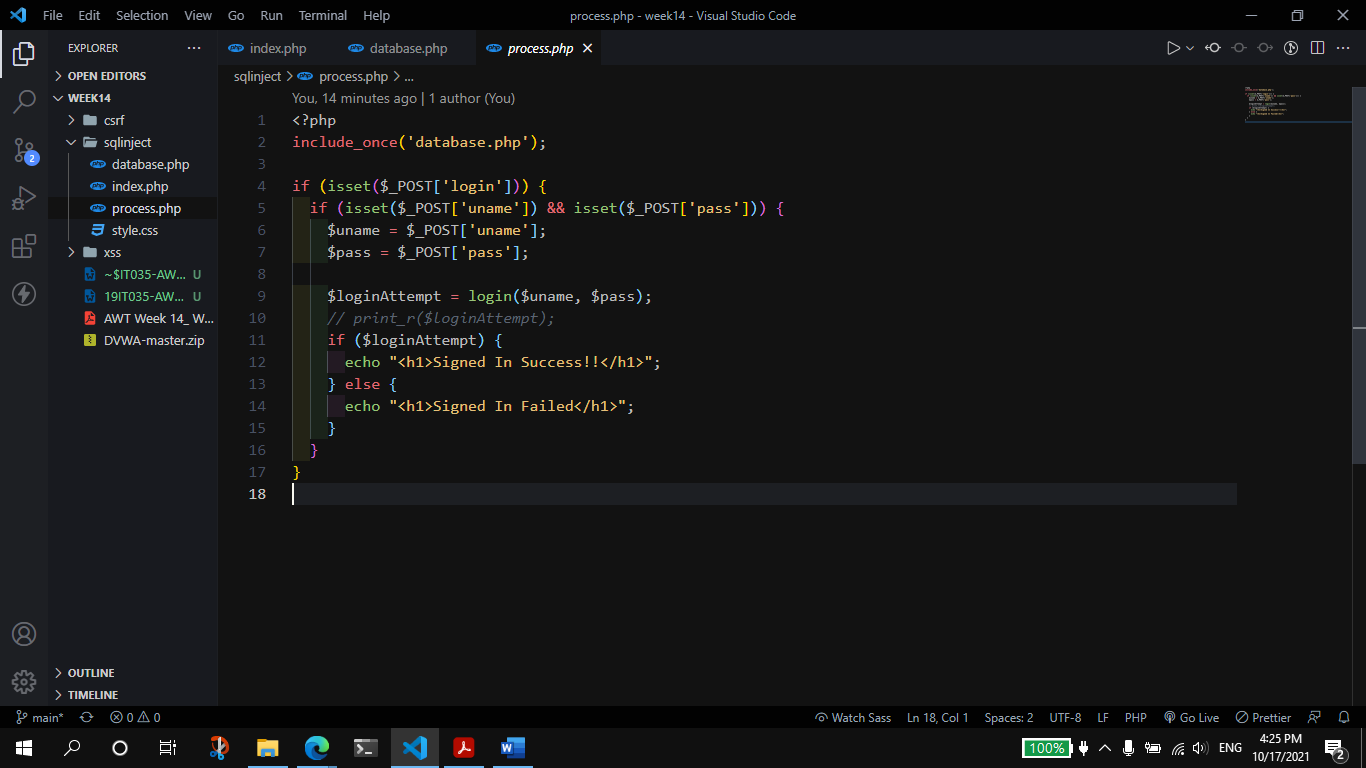
1. Demonstrate SQL injection and Prevention in PHP.

**Code: Demo SQL Injection:**

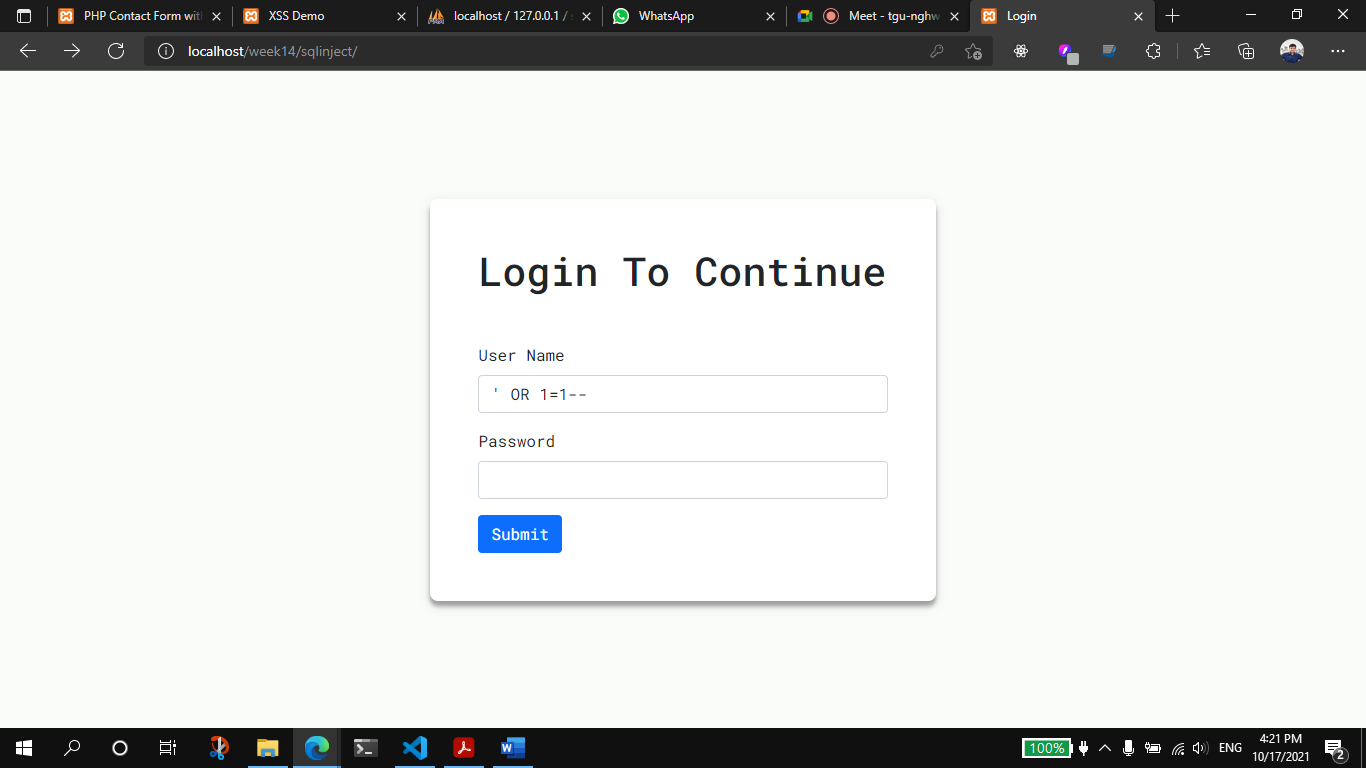


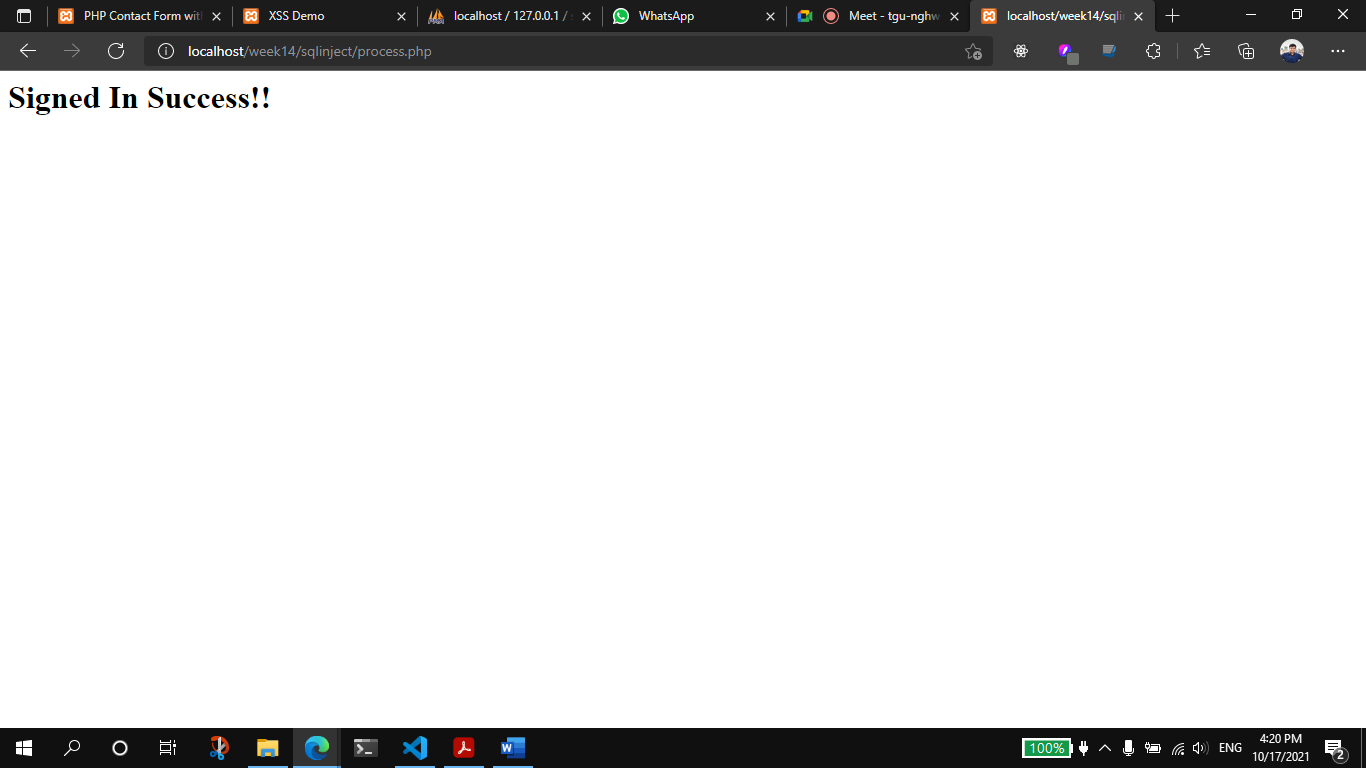






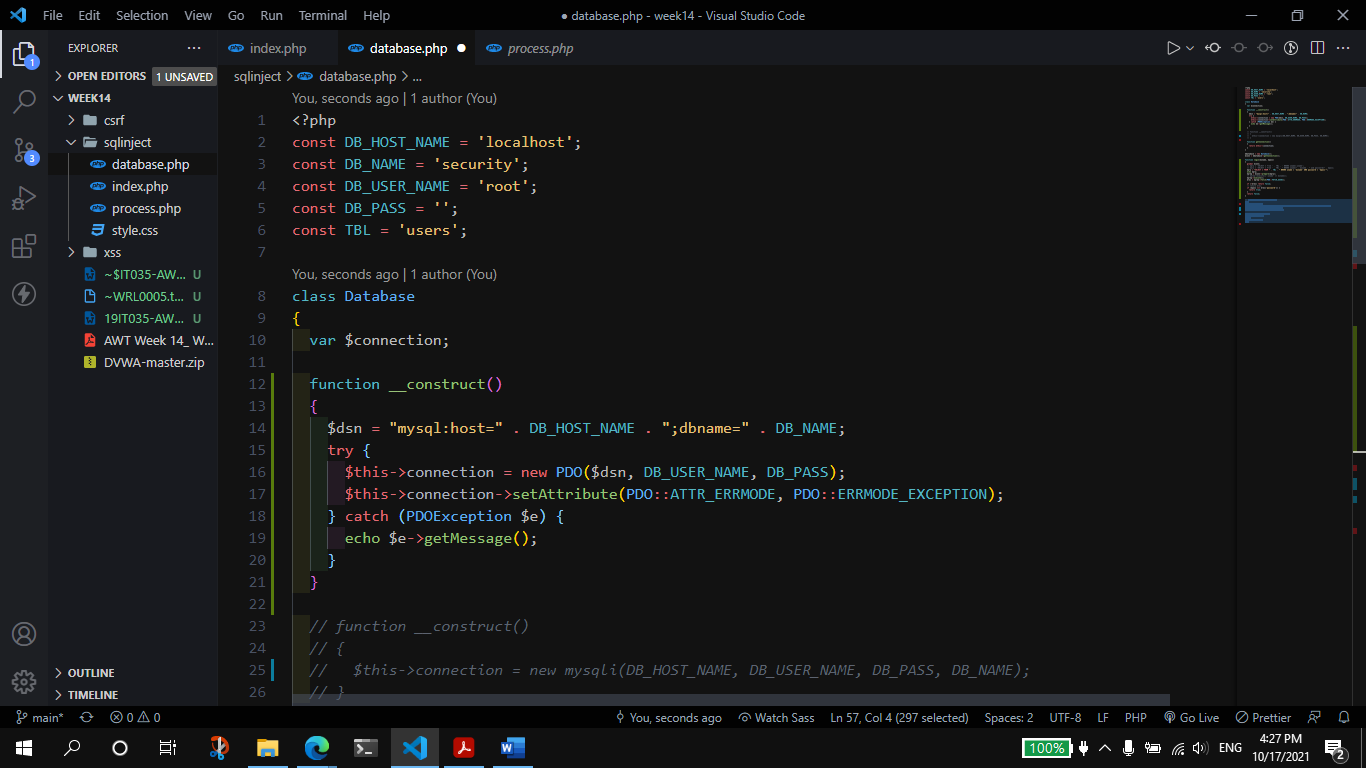
**Output:**

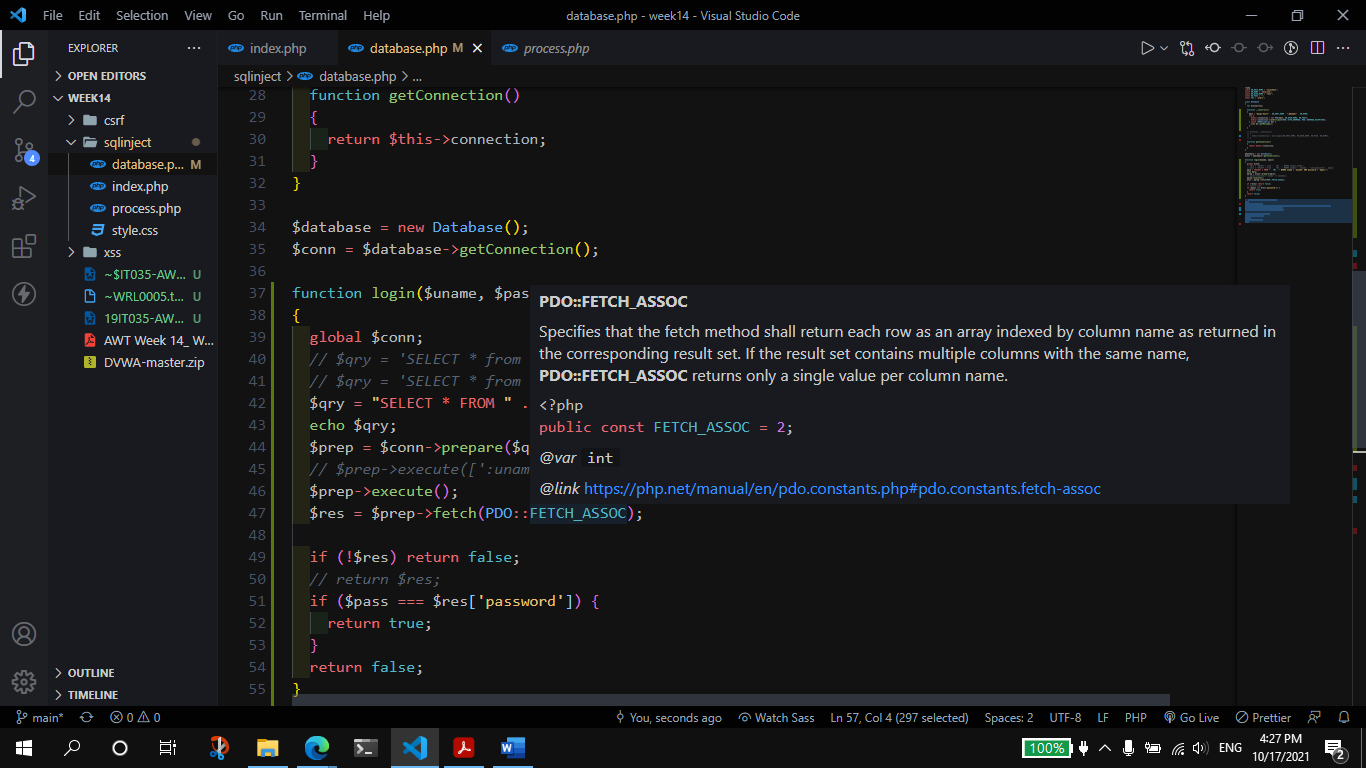




**Prevent SQL Injection:**

* Should not use the GET method, for submitting form, for sensitive data.
* Should avoid using the older functional approaches for, DB connection, instead use the PDO approach which by default add, single quotes to the input.
* Should use form validation, restrict user to use certain special character.





**Output:**



1. **What is CSRF Attack? Explain with examples with solutions.**

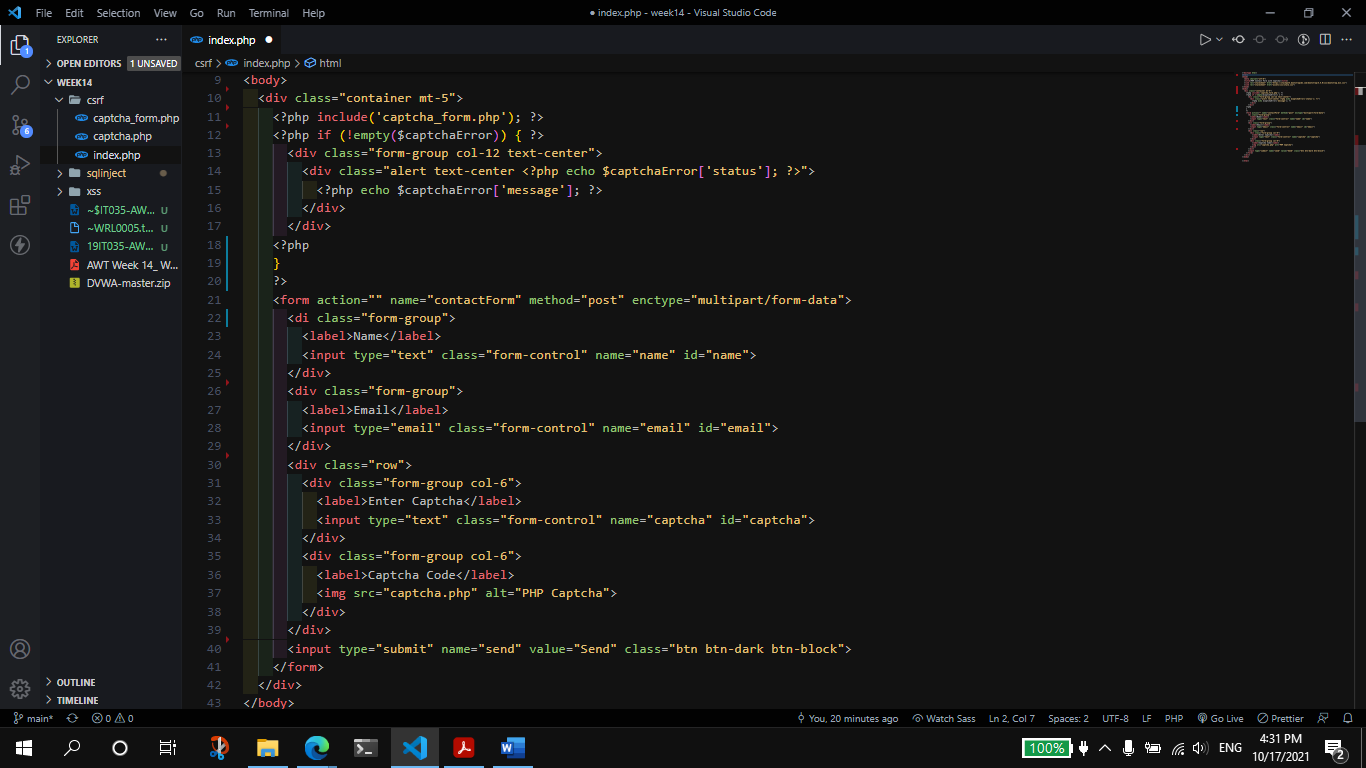
**Implement Image Captcha Using PHP to Prevent CSRF Attack.**

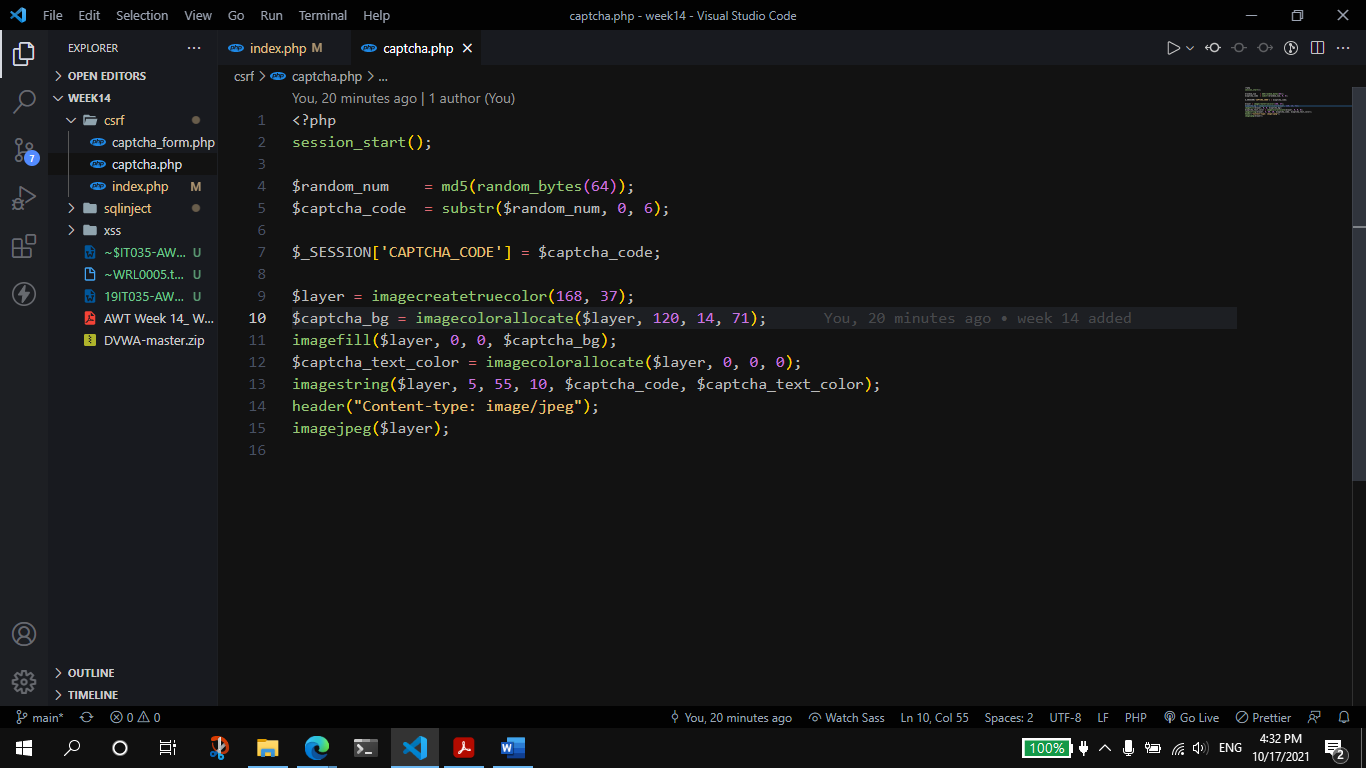
* Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they’re currently authenticated.
* If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.
* A CSRF attack is limited to the permissions of the targeted end user. An end user with limited permissions can be forced into changing email addresses, or transferring funds, while an admin account can be forced to compromise an entire web application.\
* For example, a user might receive an email or a text message with a link, which deploys malware or injects malicious code into a web page. Once the user clicks the link, attackers use the malware or injected code to send requests to the web application on the user’s behalf.

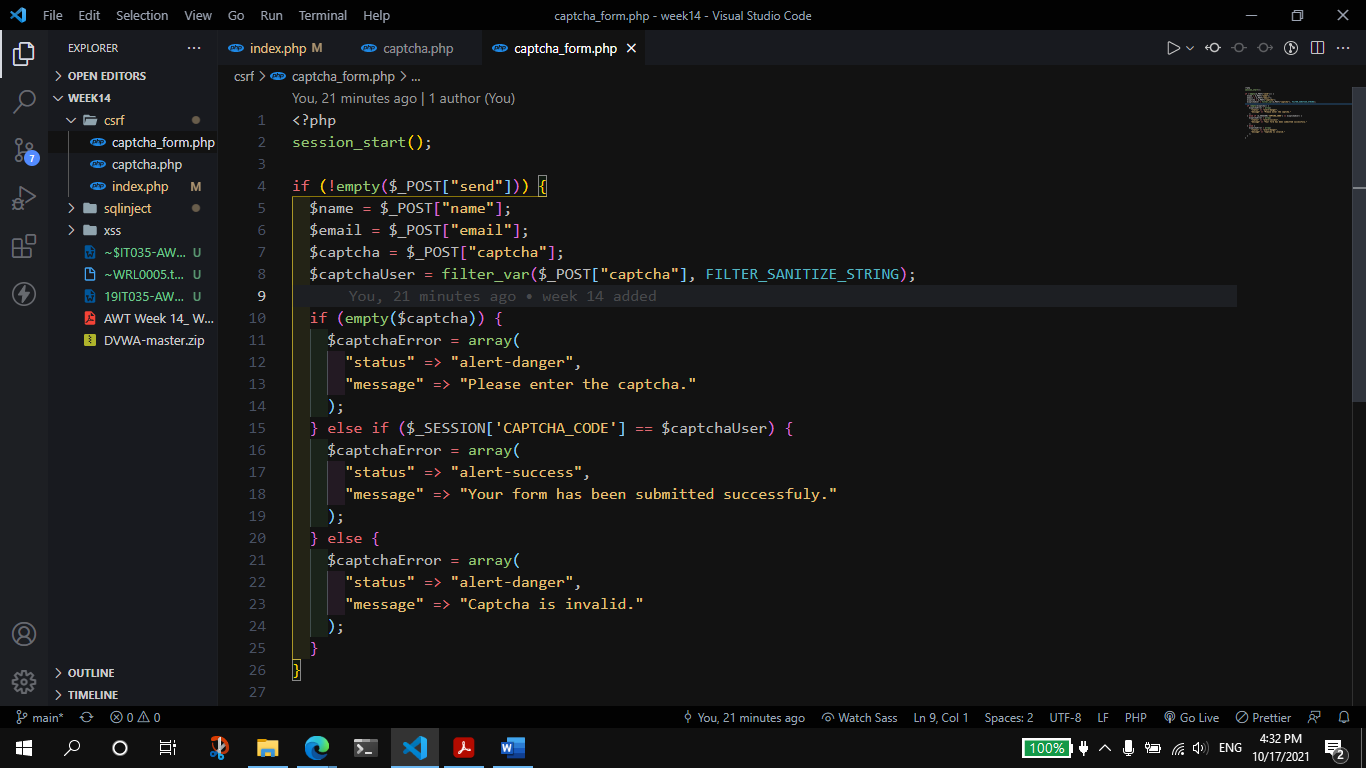
**Solutions:**

* Captcha
* Anti-CSRF Token
* Use Same site flag in cookies
* Multifactor authentication

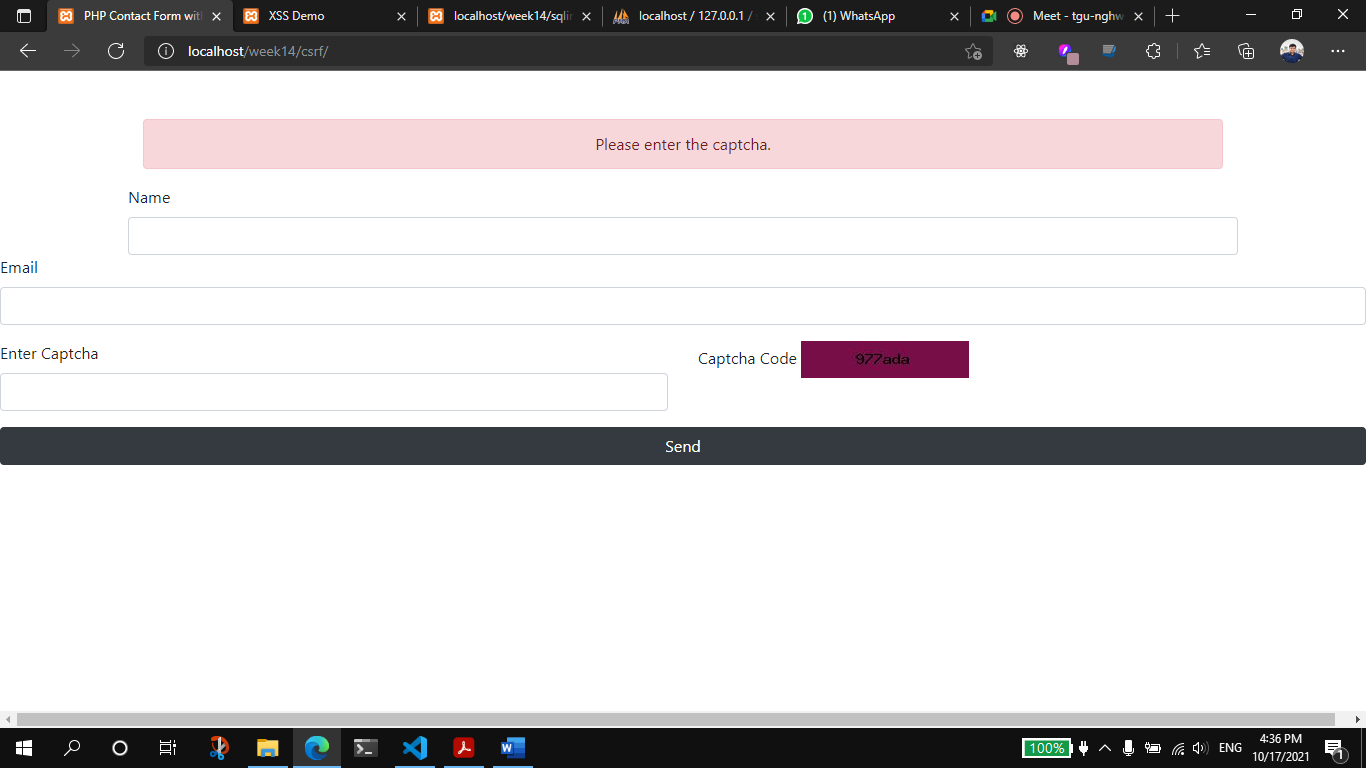
**Code**:

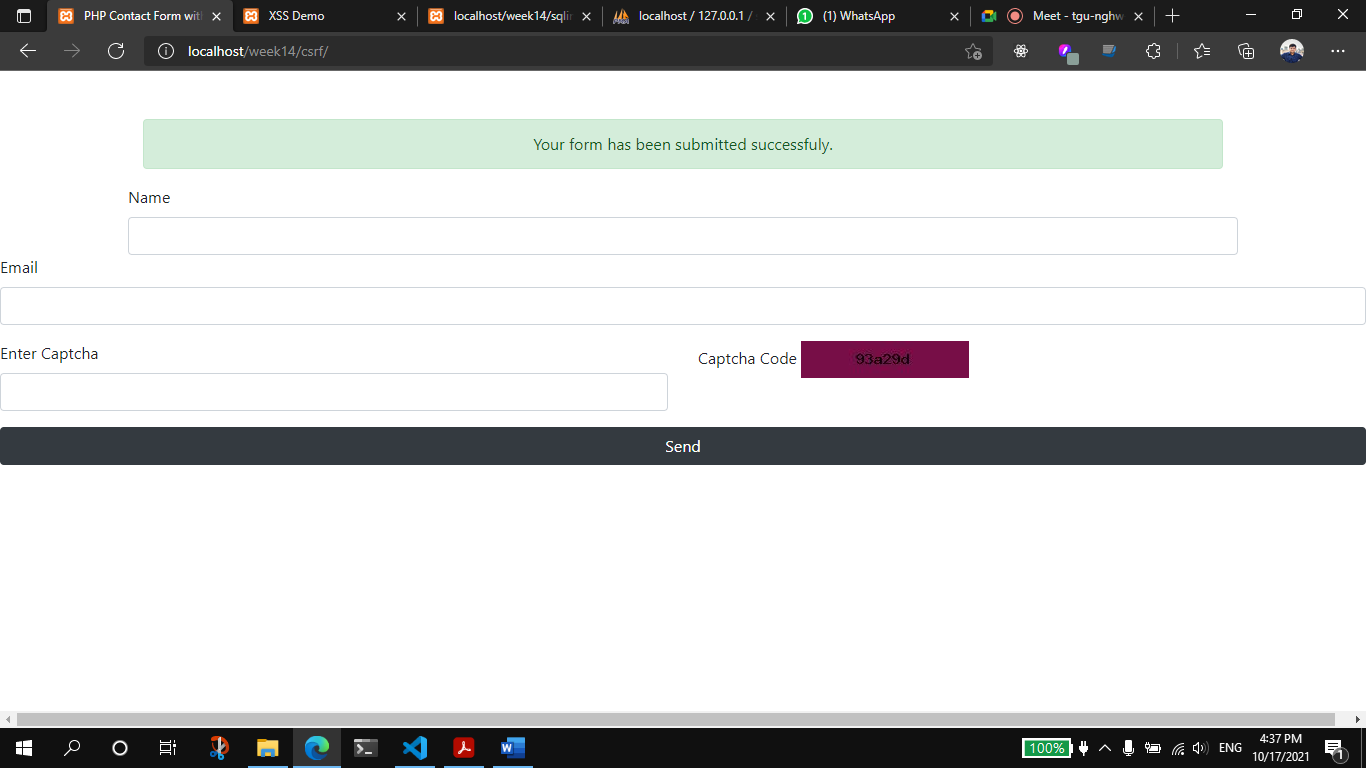






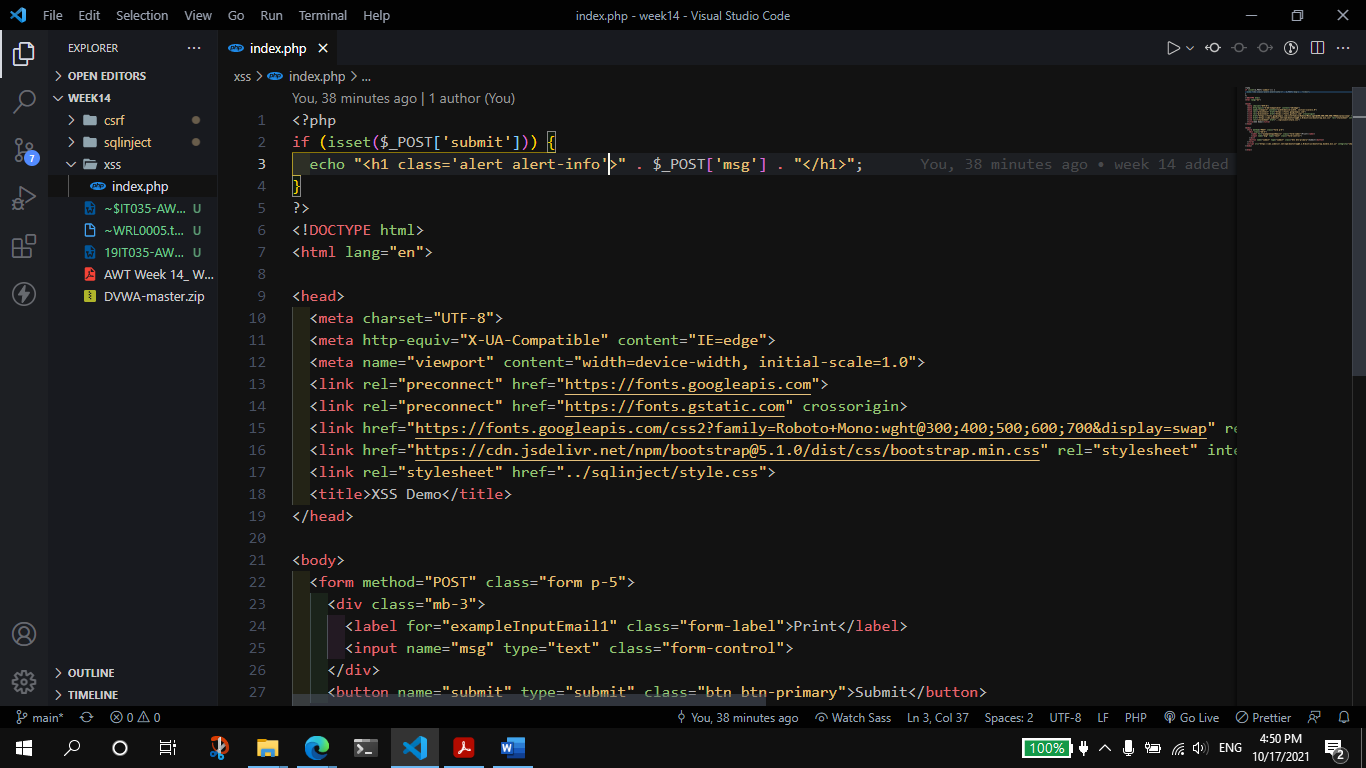
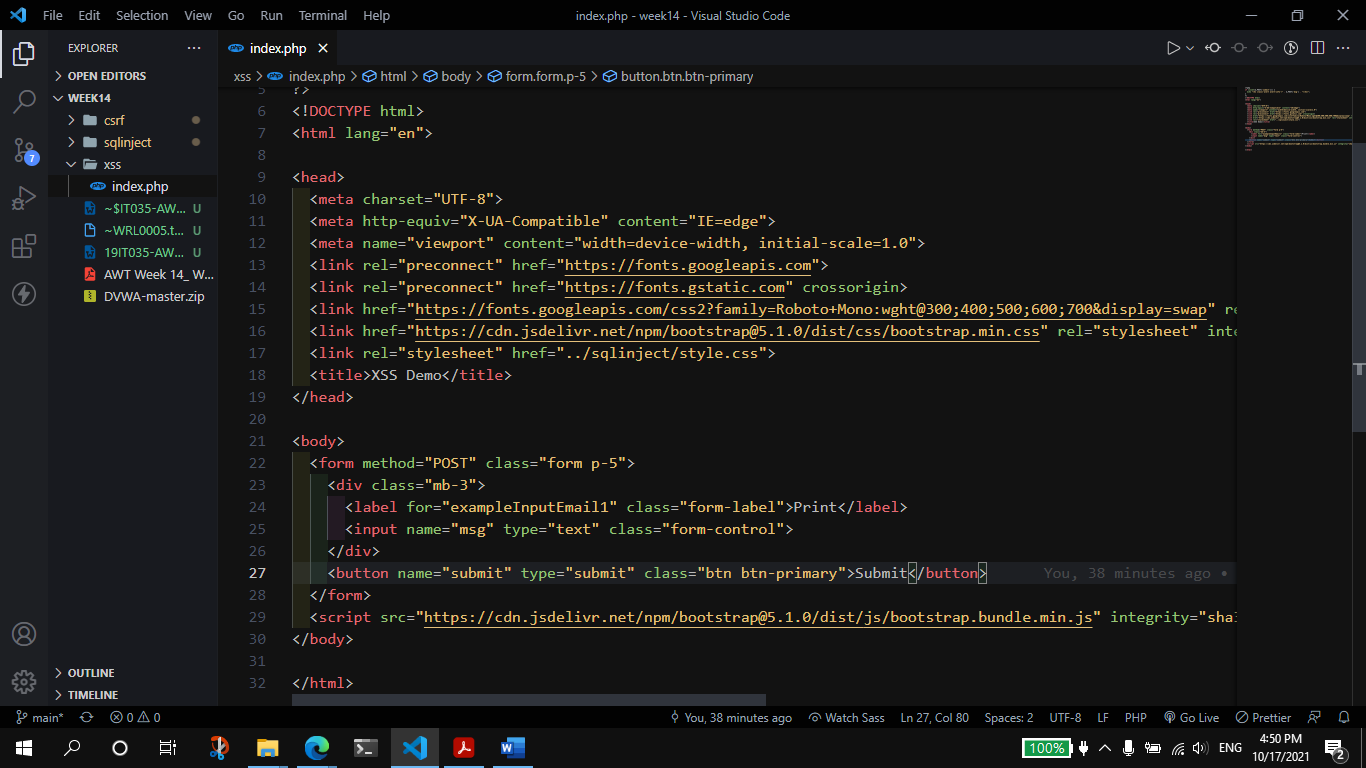
**Output:**



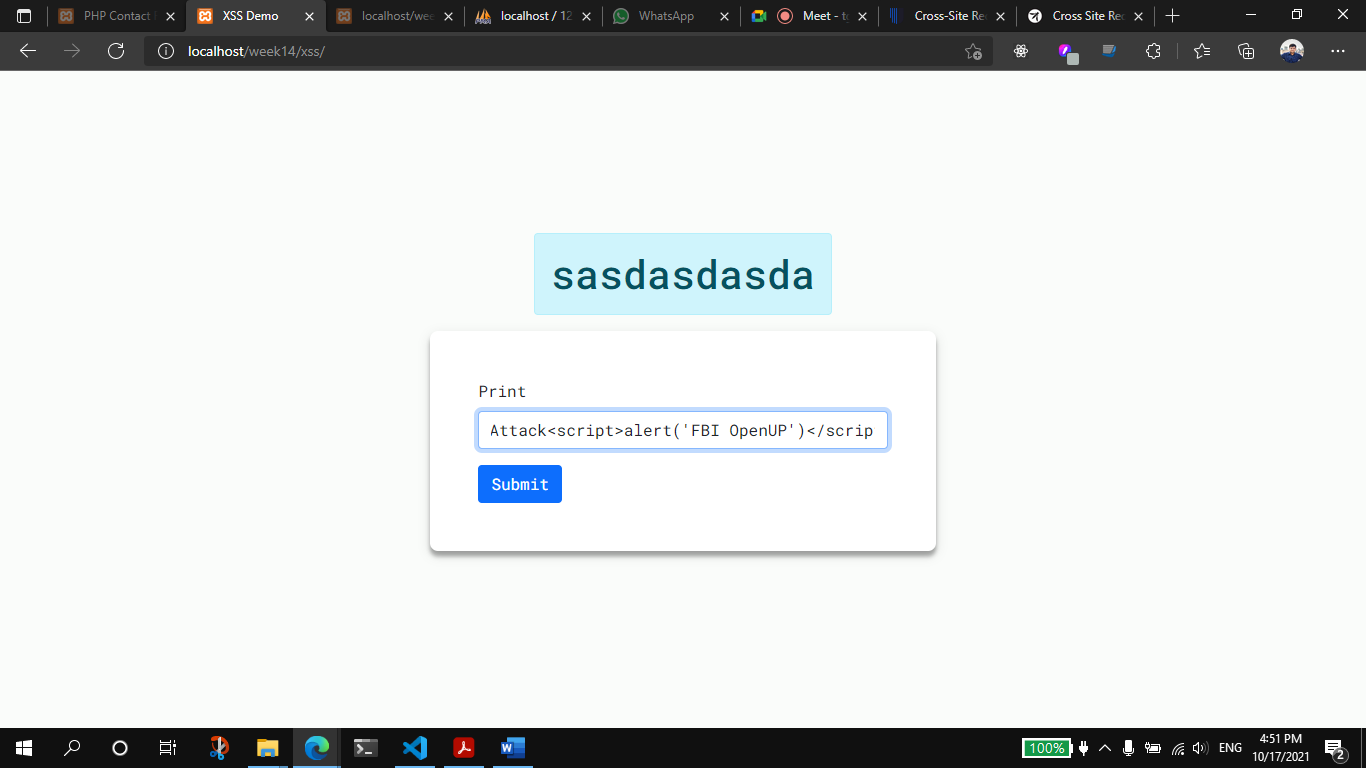


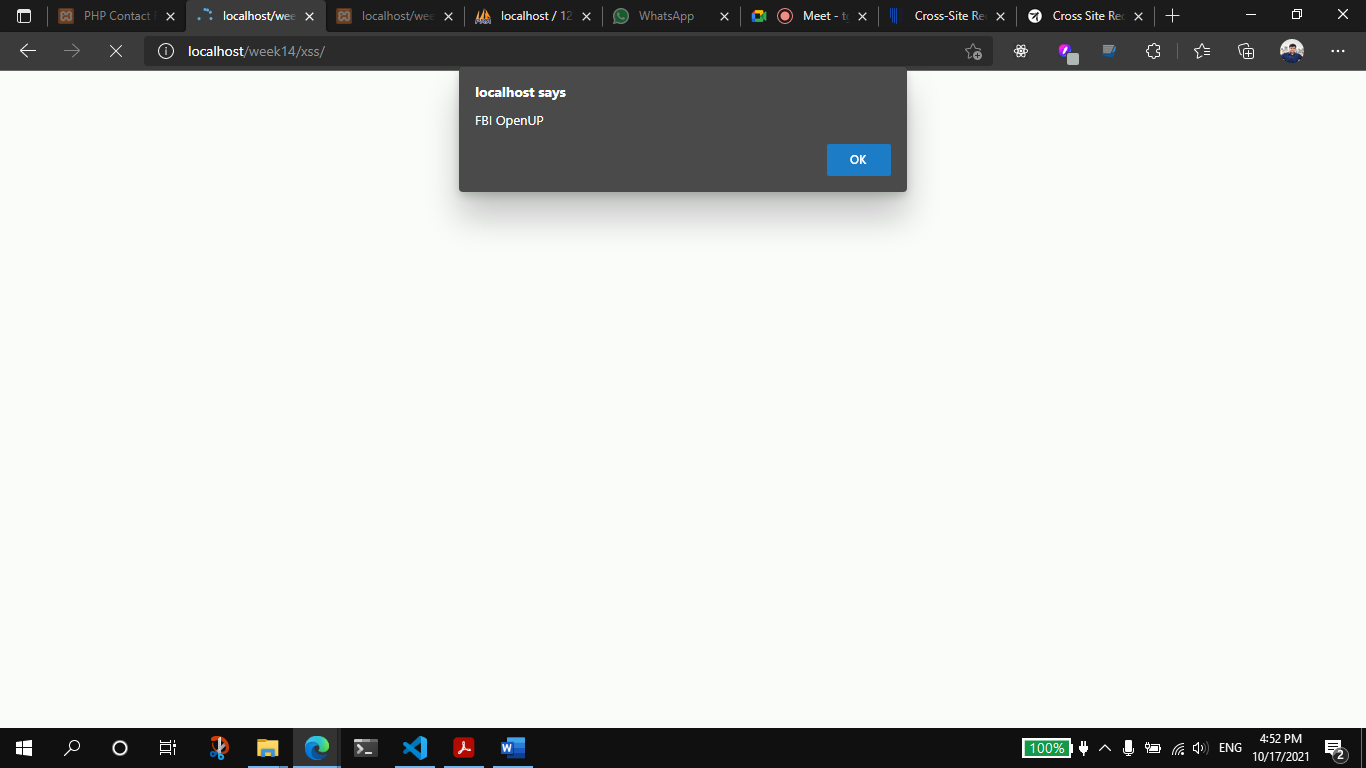
1. Demonstrate and Prevent XSS Attack in PHP App.

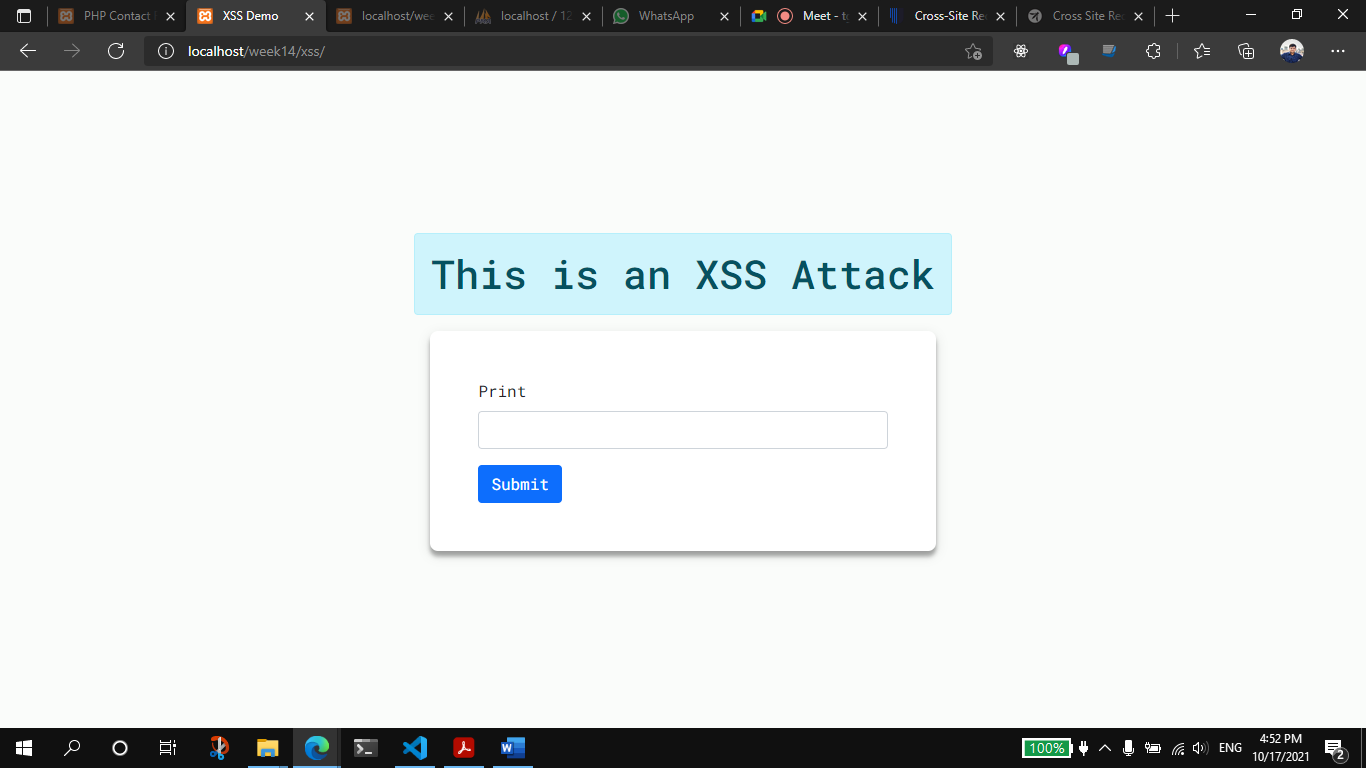
**Code:**



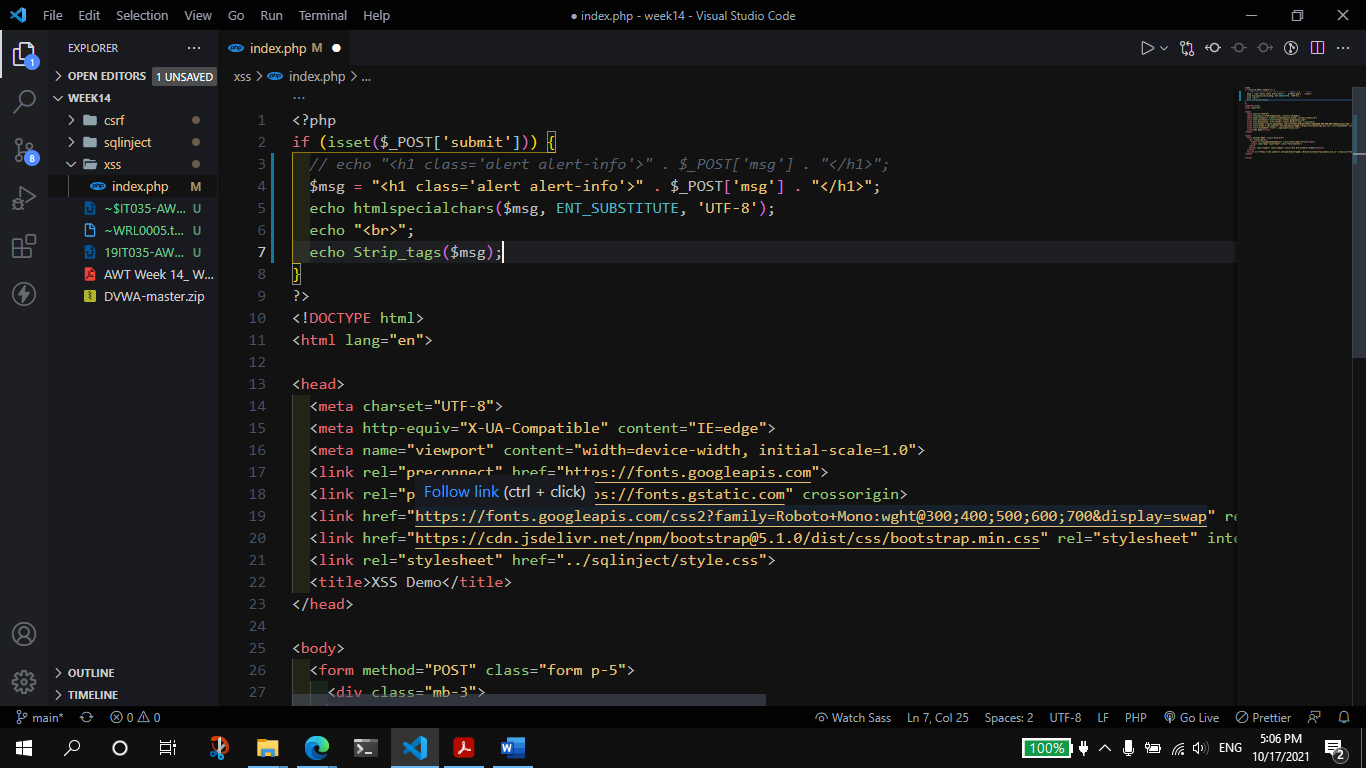
**Output:**

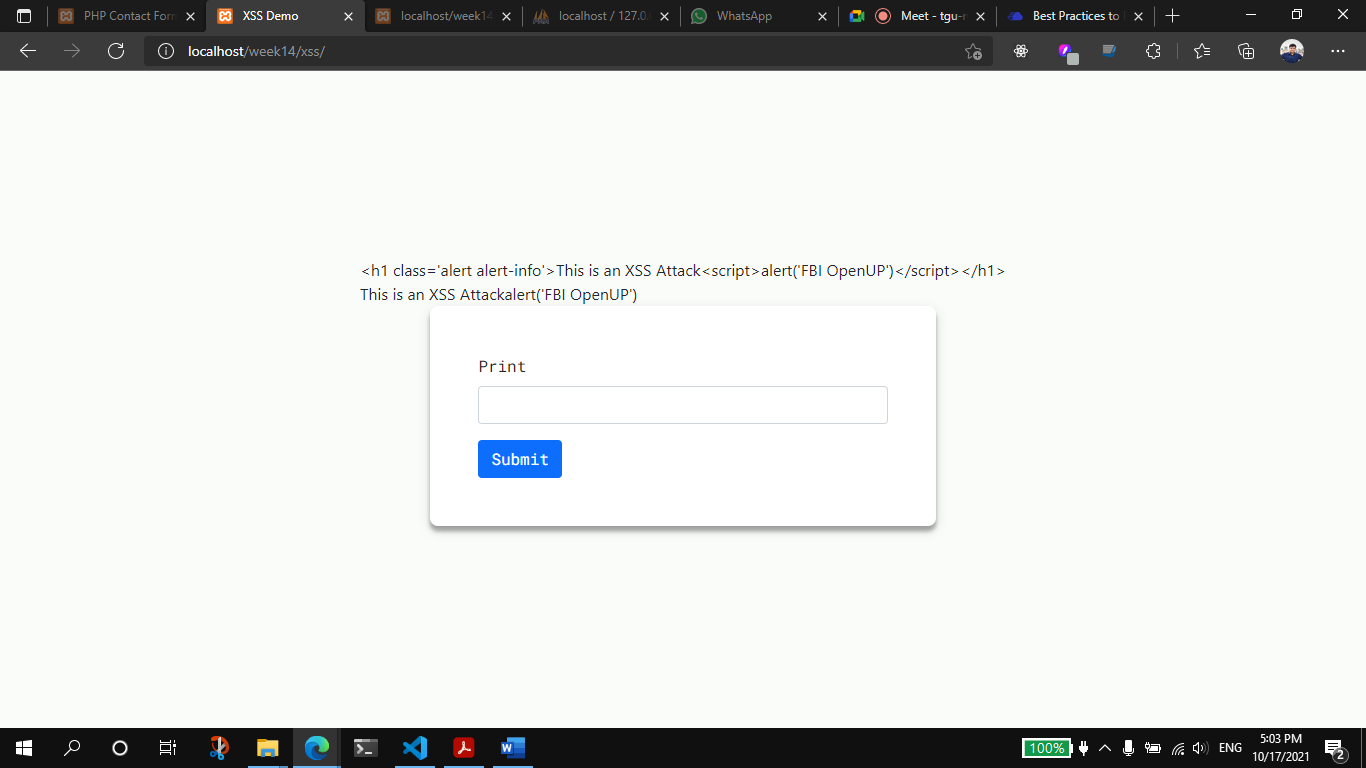






**Prevention:**





* Whenever need to print something should use htmlspecialchars() function to print, this function returns the string, and the code does not execute In this string.
* Should use strip\_tag() function to print the msg, this function remove the HTML tags and returns the string.