ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

503073

WEB PROGRAMMING & APPLICATIONS

LECTURE 10

Instructor: Mai Van Manh

OUTLINE

1. URL Rewriting
2. Most Common Web Application Attacks
   - SQL Injection
   - Cross site scripting
   - Cross site request forgery
3. Composer and External Library
4. MVC

URL Rewriting

What is URL Rewriting?

- URL rewriting is the process of modifying Uniform Resource Locators (URLs) for various purposes.
- Changing the URL can help with user access and site visibility.
- Webmasters may want to rewrite a URL for readability.

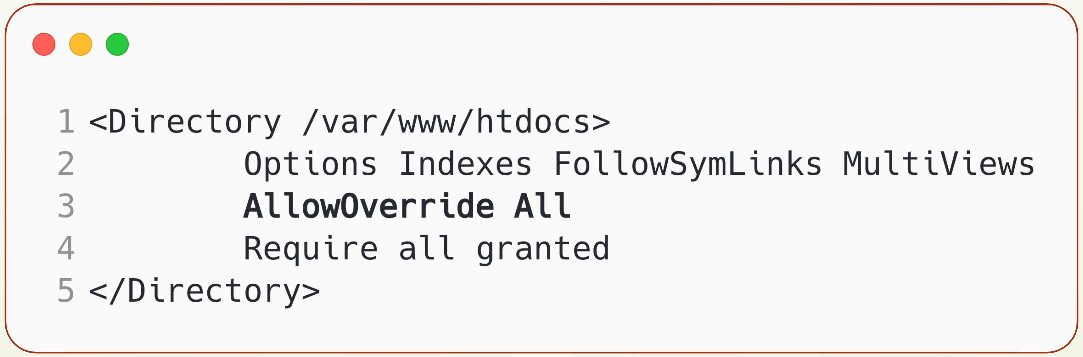
http://www.example.com/Blog/Posts.php?Year=2006&Month=12&Day=19

http://www.example.com/Blog/2006/12/19/

URL Rewriting in Apache

- In Apache, Mod_rewrite is a powerful module that provides URL manipulation capability.
- Mode_rewrite may not be enabled by default. To enable it
  - uncomment the following line in httpd.conf:

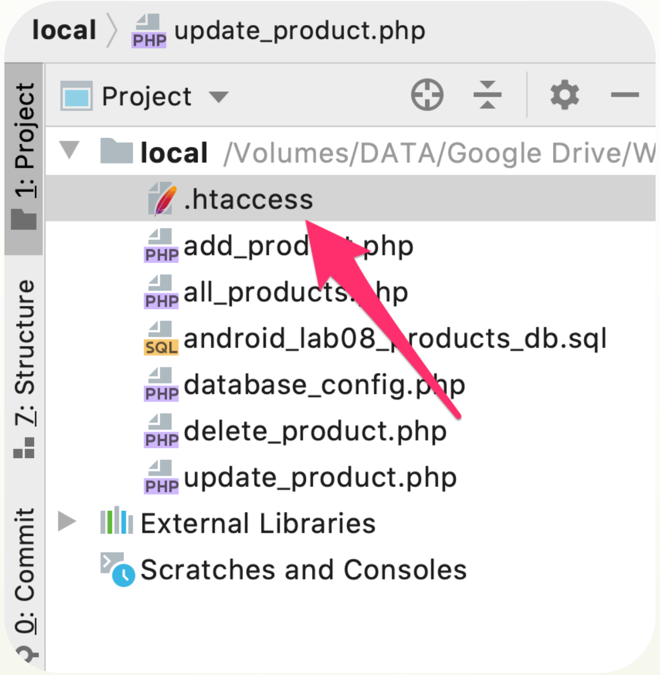
    LoadModule rewrite_module modules/mod_rewrite.so
  - Change configuration of each virtual host file as follow:



```
<Directory /var/www/htdocs>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Require all granted
</Directory>
```

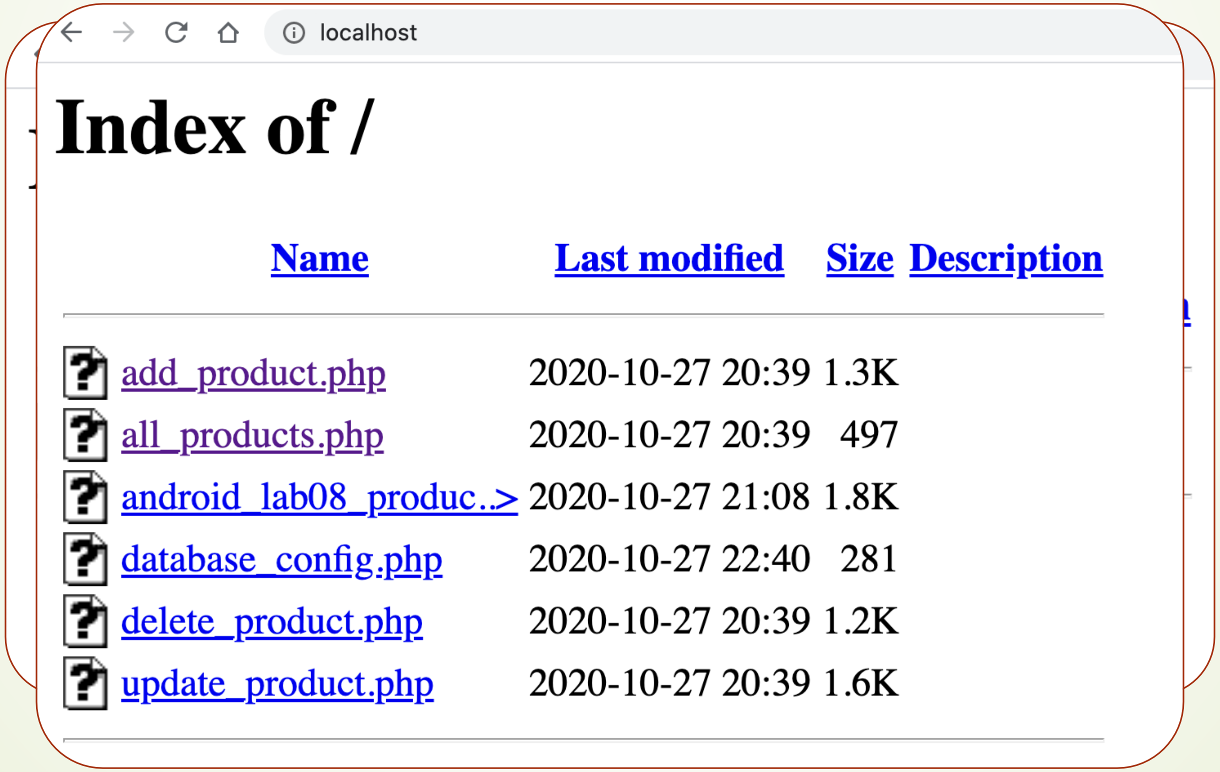
How To Use URL Rewriting?

- Create a .htaccess file in document root directory.
- Use htaccess syntax to modify URL



HTACCESS Examples

- IndexIgnore: hide a specific file extension when list a directory.

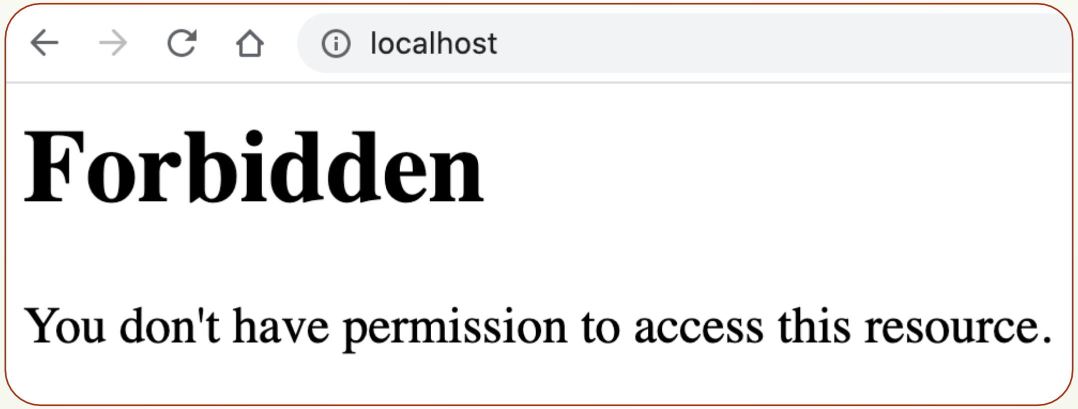


Index of /

```
        Name                  Last modified    Size  Description
--------------------------------------------------------------------
 add_product.php              2020-10-27 20:39 1.3K
 all_products.php             2020-10-27 20:39  497
 android_lab08_produc..>      2020-10-27 21:08 1.8K
 database_config.php          2020-10-27 22:40  281
 delete_product.php           2020-10-27 20:39 1.2K
 update_product.php           2020-10-27 20:39 1.6K
--------------------------------------------------------------------
```

HTACCESS Examples

- Options -Indexes: Disable directory listing.

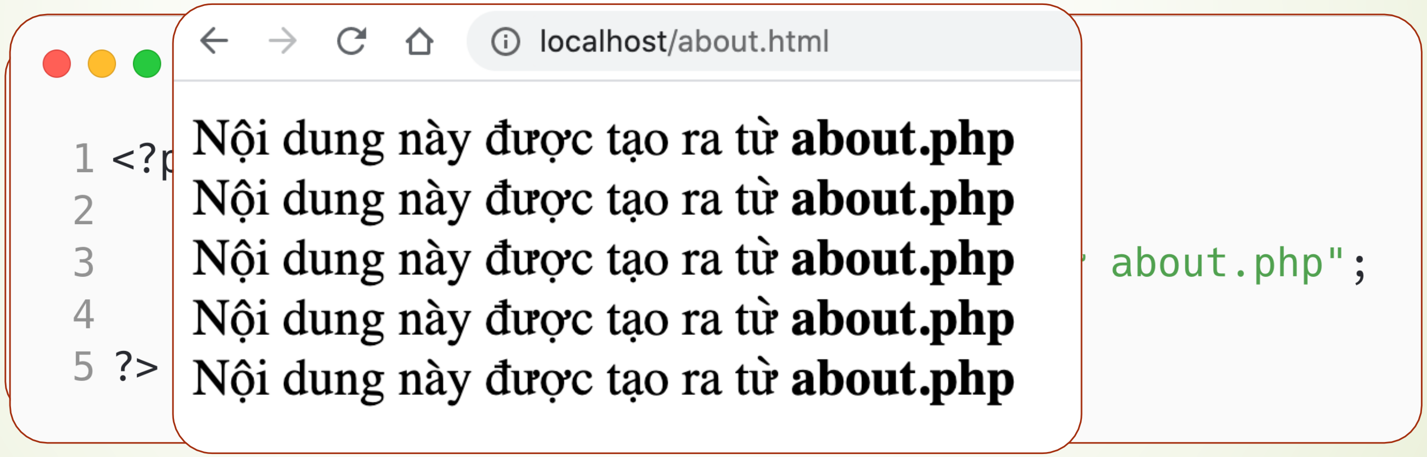


Forbidden

You don't have permission to access this resource.

HTACCESS Examples

- RewriteEngine On
- RewriteRule: Control URL mapping.



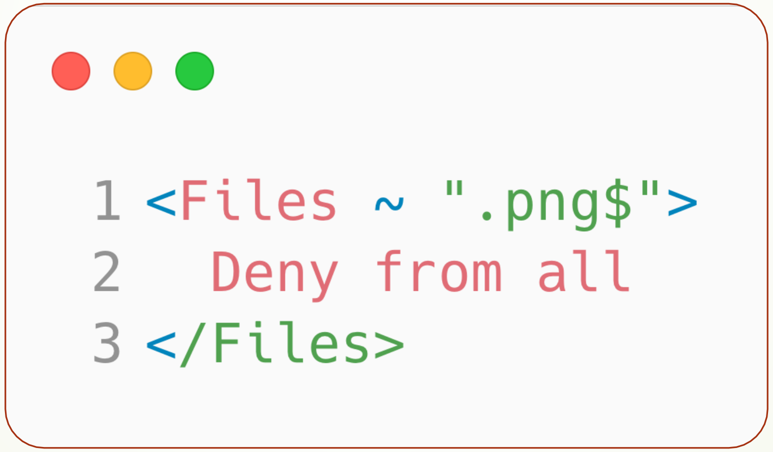
```
<?p
      about.php";
?>
```

localhost/about.html

Nội dung này được tạo ra từ **about.php**
Nội dung này được tạo ra từ **about.php**
Nội dung này được tạo ra từ **about.php**
Nội dung này được tạo ra từ **about.php**
Nội dung này được tạo ra từ **about.php**

HTACCESS Examples

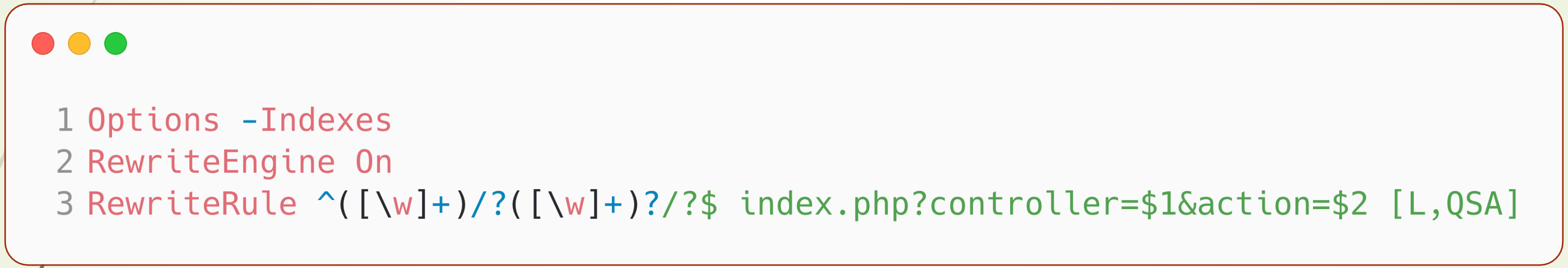
- Disable access to all png images.



```
<Files ~ ".png$">
  Deny from all
</Files>
```

HTACCESS Examples

- Create beautiful URL with controller & action names.



```
Options -Indexes
RewriteEngine On
RewriteRule ^([\w]+)/?([\w]+)?/?$ index.php?controller=$1&action=$2 [L,QSA]
```

http://localhost/index.php?controller=**account**&action=**login**

http://localhost/account/login

SQL Injection

SQL Injection Definitions

- An application security weakness that allows attackers to control an application’s database.
- A web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.
- A code injection technique in which malicious SQL statements are inserted into an entry field for execution.

When SQL Injection occured?

- SQL injection weaknesses occur when an application uses untrusted data.
- When user input is incorrectly filtered.

How SQL Injection Works?

- The attack works on dynamic SQL statements.
- Attackers try to trick the web application into run unexpected SQL commands.

What attackers can do with SQL Injection?

- Extract sensitive information, like Social Security numbers, or credit card details.
- Enumerate the authentication details of users registered on a website.
- Delete data or drop tables.
- Inject further malicious code.
- Stealing business confidential data.

SQL Injection Examples

- Vulnerability SQL command:

```
select * from account where user = 'admin' and password = '123456’
```

- Modified version:

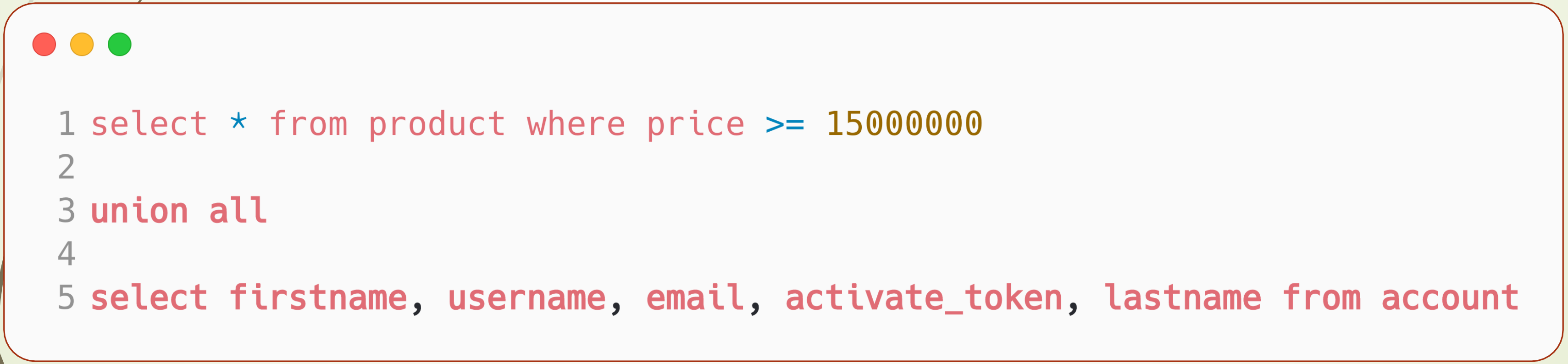
```
select * from account where user = 'admin' and password = '123456’ or 1 = 1

select * from account where user = 'admin’ or 1 = 1 -- and password = '123456’
```

SQL Injection Examples

- Vulnerability SQL command:

  select * from product where price >= $price

- Modified version:

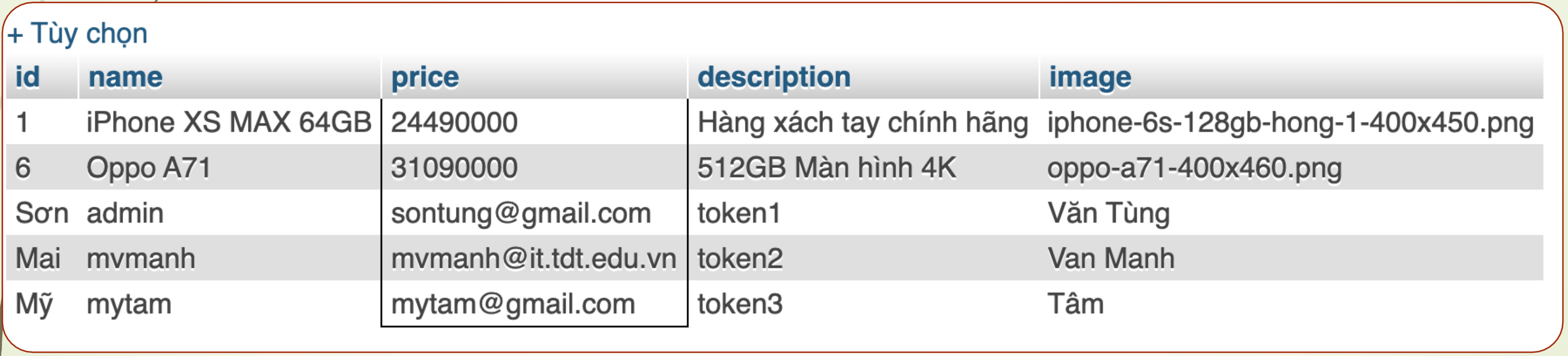


```
select * from product where price >= 15000000

union all

select firstname, username, email, activate_token, lastname from account
```

SQL Injection Examples

- Vulnerability SQL command:

```
select * from product where price >= $price
```



+ Tùy chọn

id	name	price	description	image
1	iPhone XS MAX 64GB	24490000	Hàng xách tay chính hãng	iphone-6s-128gb-hong-1-400x450.png
6	Oppo A71	31090000	512GB Màn hình 4K	oppo-a71-400x460.png
Sơn	admin	sontung@gmail.com	token1	Văn Tùng
Mai	mvmanh	mvmanh@it.tdt.edu.vn	token2	Van Manh
Mỹ	mytam	mytam@gmail.com	token3	Tâm

How to prevent SQL Injection attacks

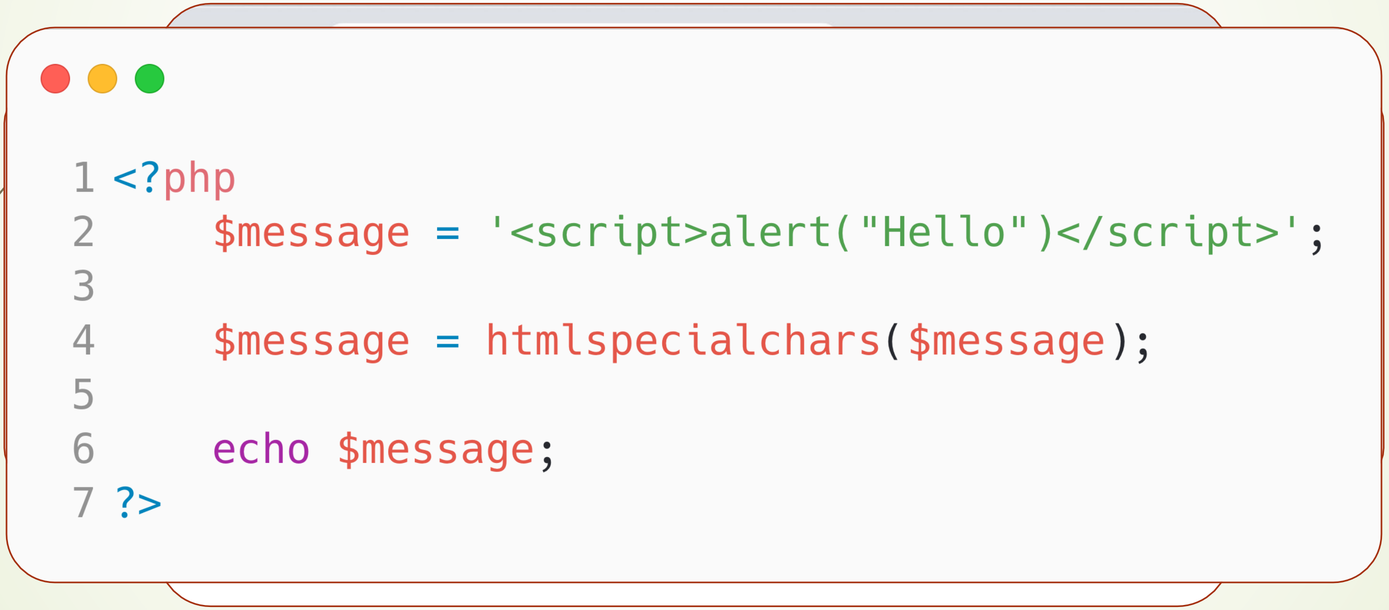
- Carefully sanitize input data.
- Use Prepared Statement for all SQL Query.

Sanitize Input Data

- htmlspecialchars()
- *filter_var()*
- *filter_input()*

Sanitize Input Data

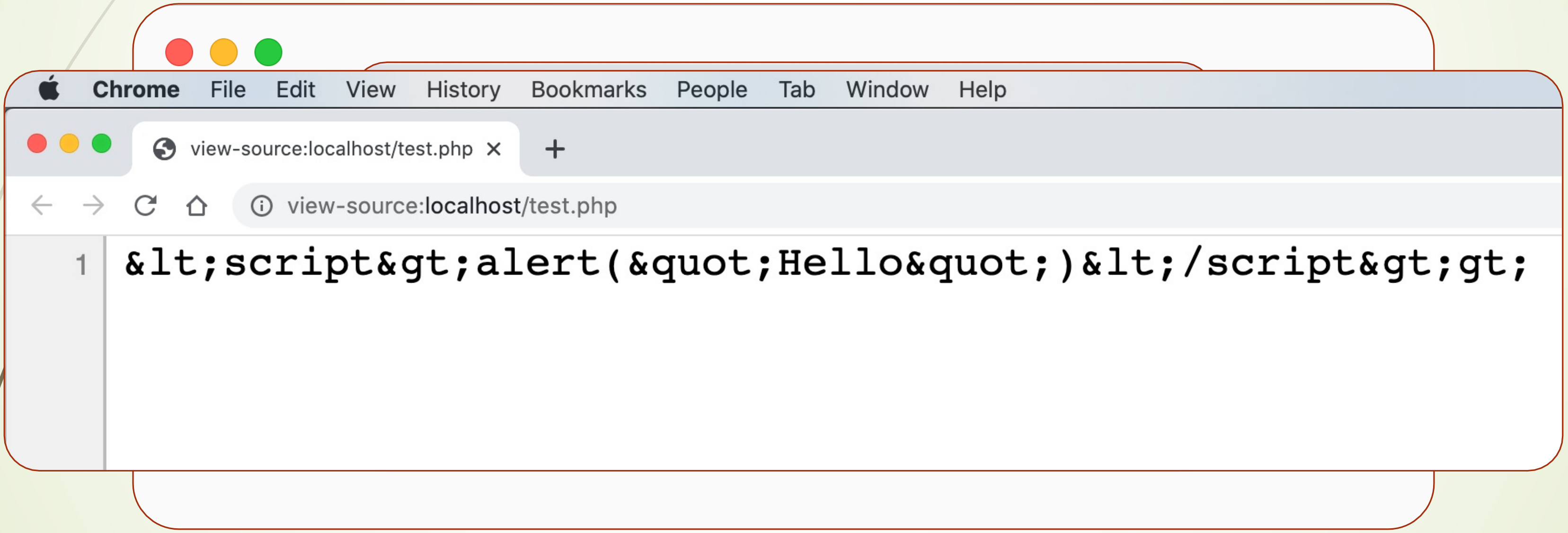
- htmlspecialchars()



```php
<?php
    $message = '<script>alert("Hello")</script>';

    $message = htmlspecialchars($message);

    echo $message;
?>
```

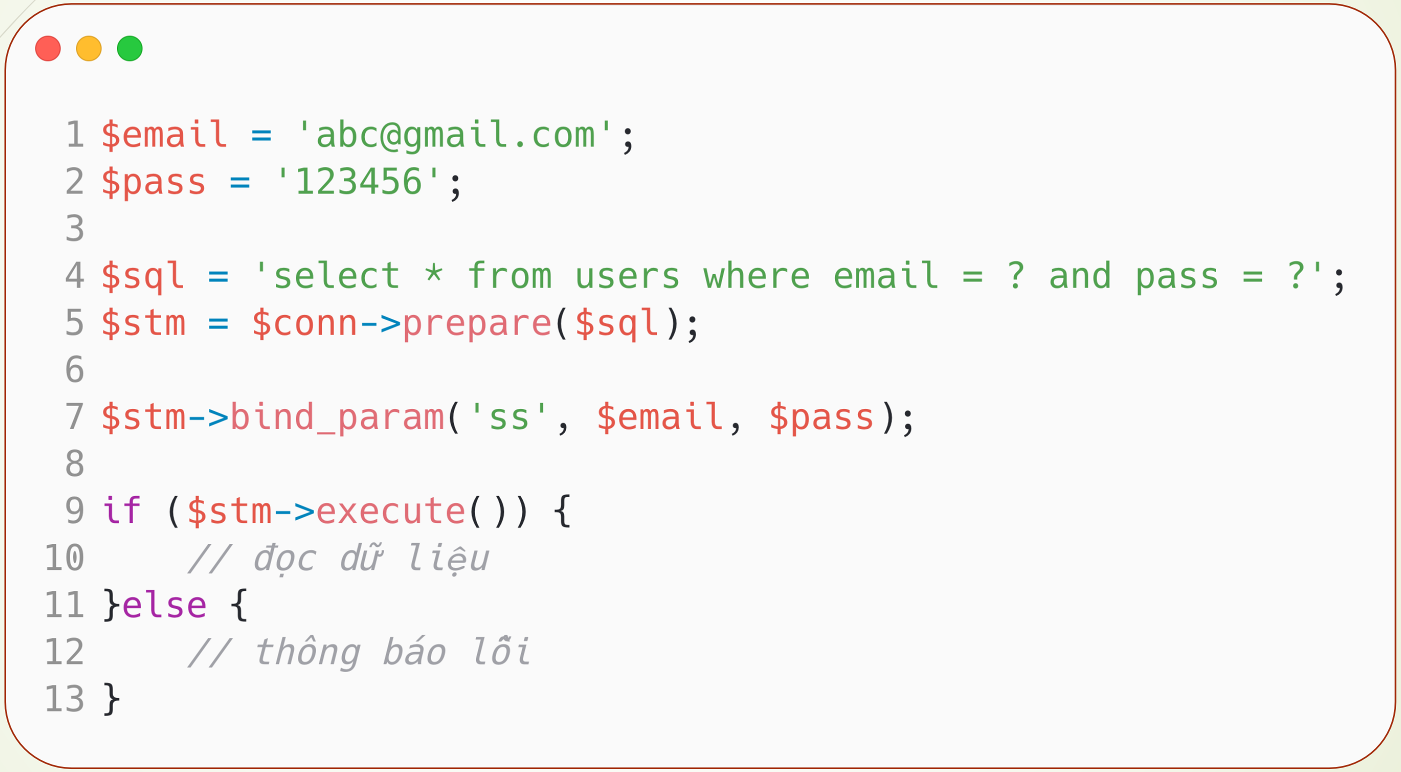
Sanitize Input Data

- htmlspecialchars()


Chrome
File
Edit
View
History
Bookmarks
People
Tab
Window
Help
view-source:localhost/test.php
view-source:localhost/test.php

```
&lt;script&gt;alert(&quot;Hello&quot;)&lt;/script&gt;gt;
```

Prepared Statement



```php
$email = 'abc@gmail.com';
$pass = '123456';

$sql = 'select * from users where email = ? and pass = ?';
$stm = $conn->prepare($sql);

$stm->bind_param('ss', $email, $pass);

if ($stm->execute()) {
    // đọc dữ liệu
}else {
    // thông báo lỗi
}
```

Composer

MVC

CMS

Content Management System

CMS - Wordpress

CMS - Drupal

CMS - Joomla

Web Development Trends

Web Development Trends

- Single Page Application
- Progressive Web Apps
- WebAssembly
- Voice Search and Navigation