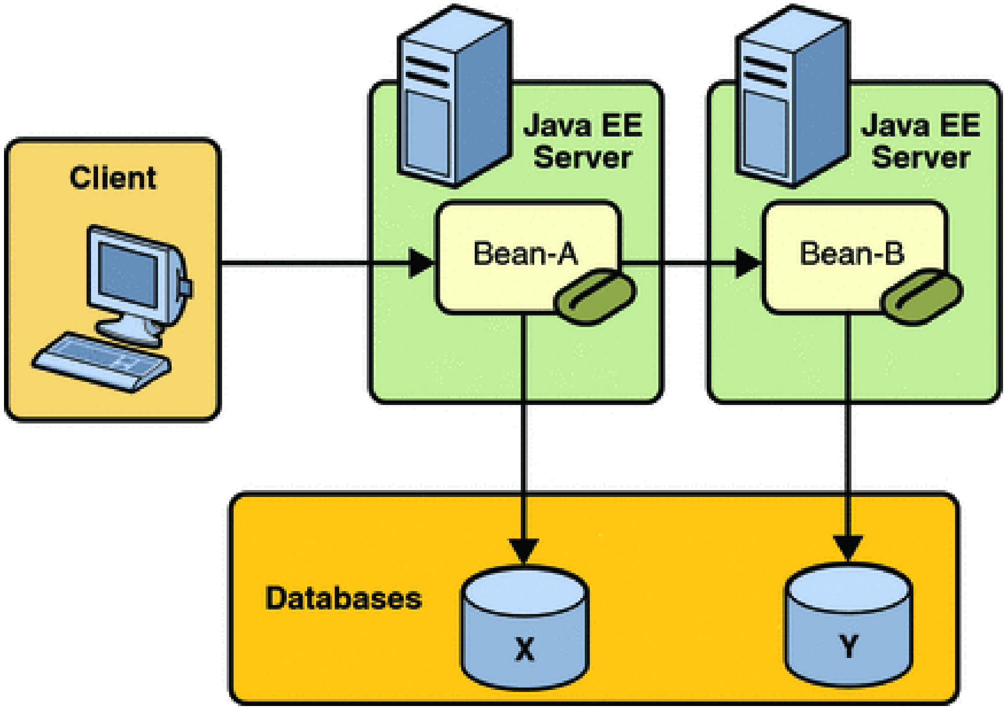
SQL Injection Attack Lab

Kailiang

❖ Background of DATABASE

Database:

Structured collection of data. The data are typically organized to model relevant aspects of reality, in a way that supports processes requiring this information.



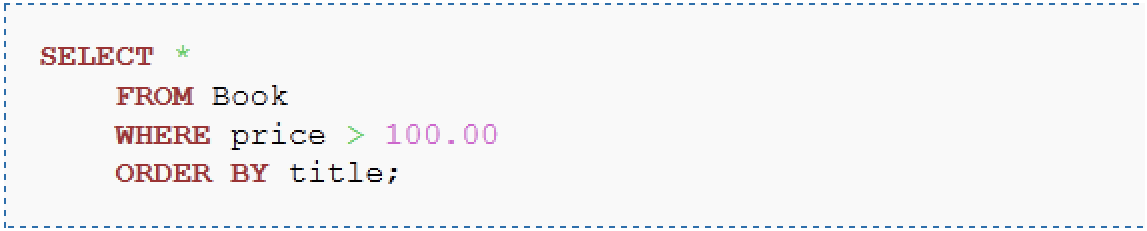
❖ Background of SQL

SQL (**Structured Query Language**):

A special-purpose programming language designed for managing data in **relational database** management systems (RDBMS).

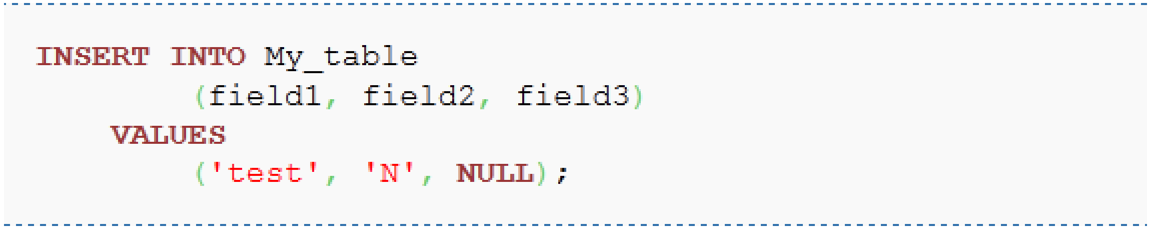
Data Manipulation:

1> Query:



```
SELECT *
    FROM Book
    WHERE price > 100.00
    ORDER BY title;
```

2> Insert:

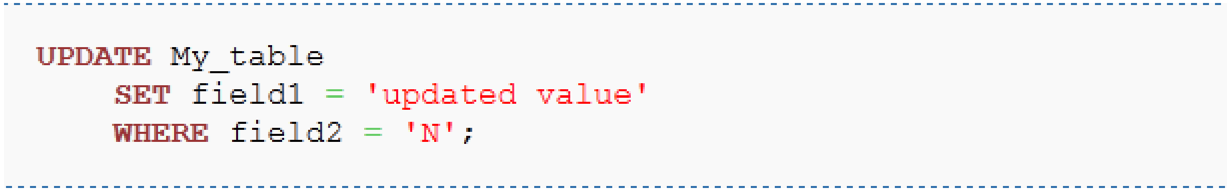


```
INSERT INTO My_table
        (field1, field2, field3)
    VALUES
        ('test', 'N', NULL);
```

Background of SQL

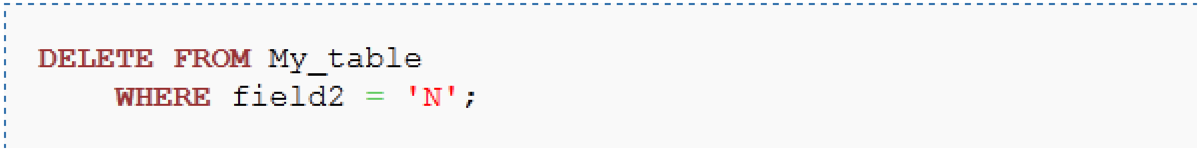
Data Manipulation:

3> Update:



```
UPDATE My_table
    SET field1 = 'updated value'
    WHERE field2 = 'N';
```

4> Delete:

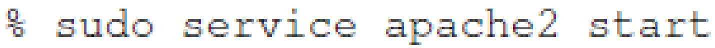


```
DELETE FROM My_table
    WHERE field2 = 'N';
```

W3C school is a good result to further study SQL statement

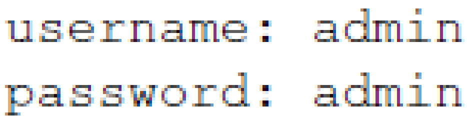
❖ Environment Configuration

1. Staring the Apache Server



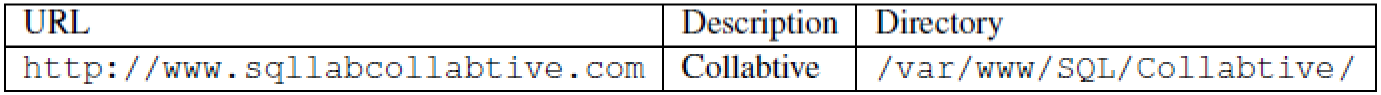
```
% sudo service apache2 start
```

2. The Collabtive Web Application and other accounts info are in admin post



```
username: admin
password: admin
```

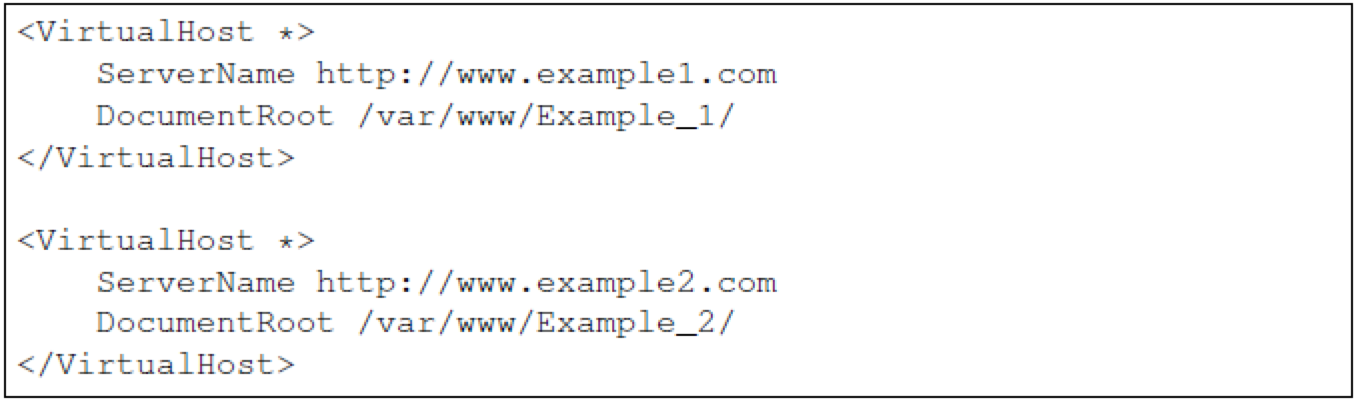
3. Configuring DNS



URL	Description	Directory
`http://www.sqllabcollabtive.com`	Collabtive	`/var/www/SQL/Collabtive/`

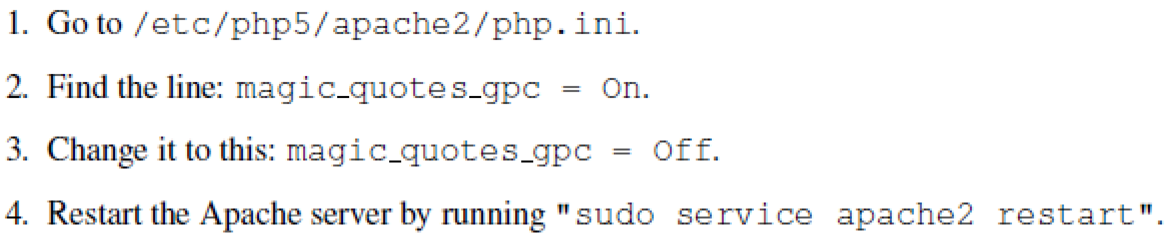
❖ Environment Configuration

4. Configuring Apache Server



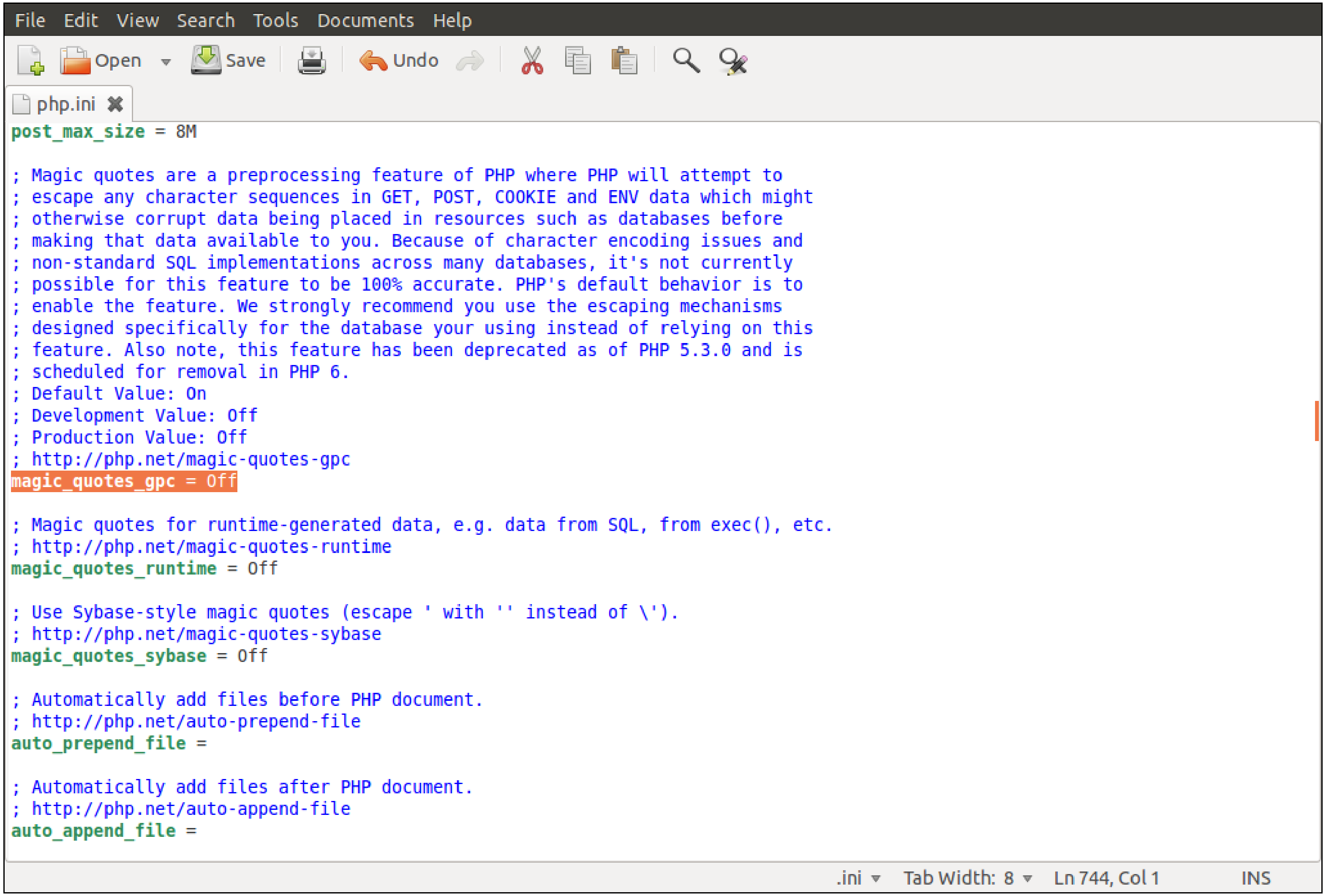
```
<VirtualHost *>
    ServerName http://www.example1.com
    DocumentRoot /var/www/Example_1/
</VirtualHost>

<VirtualHost *>
    ServerName http://www.example2.com
    DocumentRoot /var/www/Example_2/
</VirtualHost>
```

5. Turn off the Countermeasure



1. Go to `/etc/php5/apache2/php.ini`.
2. Find the line: `magic_quotes_gpc = On`.
3. Change it to this: `magic_quotes_gpc = Off`.
4. Restart the Apache server by running "`sudo service apache2 restart`".

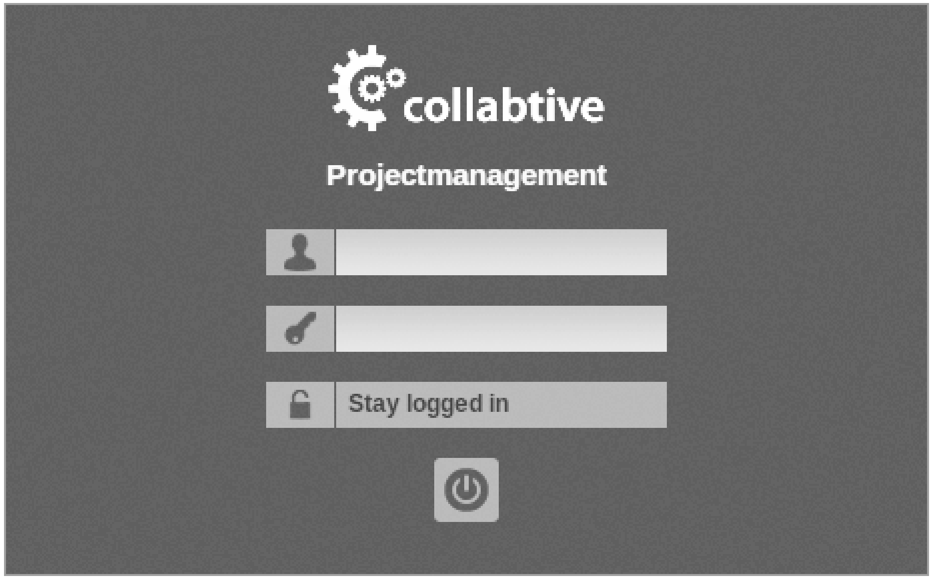
❖ Environment Configuration



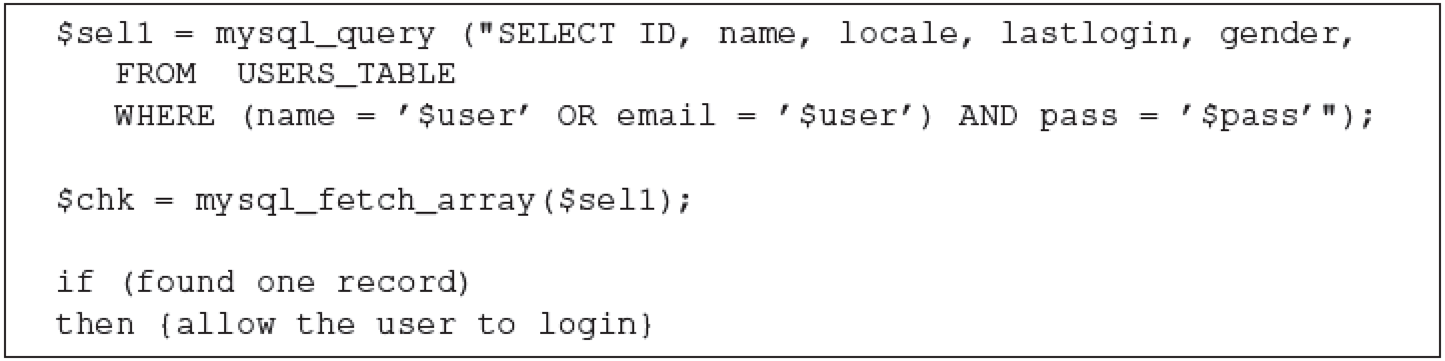
```
post_max_size = 8M

; Magic quotes are a preprocessing feature of PHP where PHP will attempt to
; escape any character sequences in GET, POST, COOKIE and ENV data which might
; otherwise corrupt data being placed in resources such as databases before
; making that data available to you. Because of character encoding issues and
; non-standard SQL implementations across many databases, it's not currently
; possible for this feature to be 100% accurate. PHP's default behavior is to
; enable the feature. We strongly recommend you use the escaping mechanisms
; designed specifically for the database your using instead of relying on this
; feature. Also note, this feature has been deprecated as of PHP 5.3.0 and is
; scheduled for removal in PHP 6.
; Default Value: On
; Development Value: Off
; Production Value: Off
; http://php.net/magic-quotes-gpc
magic_quotes_gpc = Off

; Magic quotes for runtime-generated data, e.g. data from SQL, from exec(), etc.
; http://php.net/magic-quotes-runtime
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ' with '' instead of \').
; http://php.net/magic-quotes-sybase
magic_quotes_sybase = Off

; Automatically add files before PHP document.
; http://php.net/auto-prepend-file
auto_prepend_file =

; Automatically add files after PHP document.
; http://php.net/auto-append-file
auto_append_file =
```

❖ Task 1: SQL Injection Attack on SELECT Statements

Login Window:



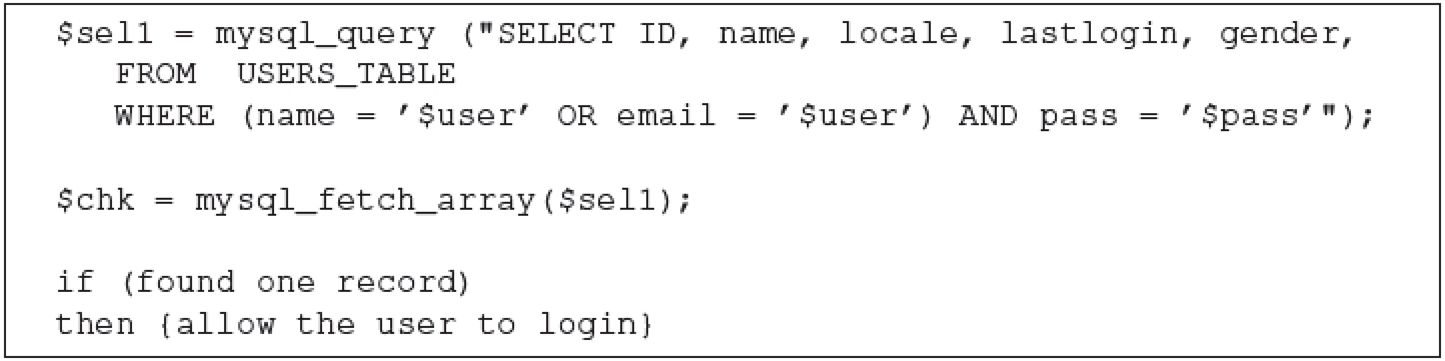
Login Code:



```
$sel1 = mysql_query ("SELECT ID, name, locale, lastlogin, gender,
   FROM  USERS_TABLE
   WHERE (name = ’$user’ OR email = ’$user’) AND pass = ’$pass’");

$chk = mysql_fetch_array($sel1);

if (found one record)
then {allow the user to login}
```

❖ Task 1: SQL Injection Attack on SELECT Statements

Login Code:



```
$sel1 = mysql_query ("SELECT ID, name, locale, lastlogin, gender,
   FROM  USERS_TABLE
   WHERE (name = '$user' OR email = '$user') AND pass = '$pass'");

$chk = mysql_fetch_array($sel1);

if (found one record)
then {allow the user to login}
```

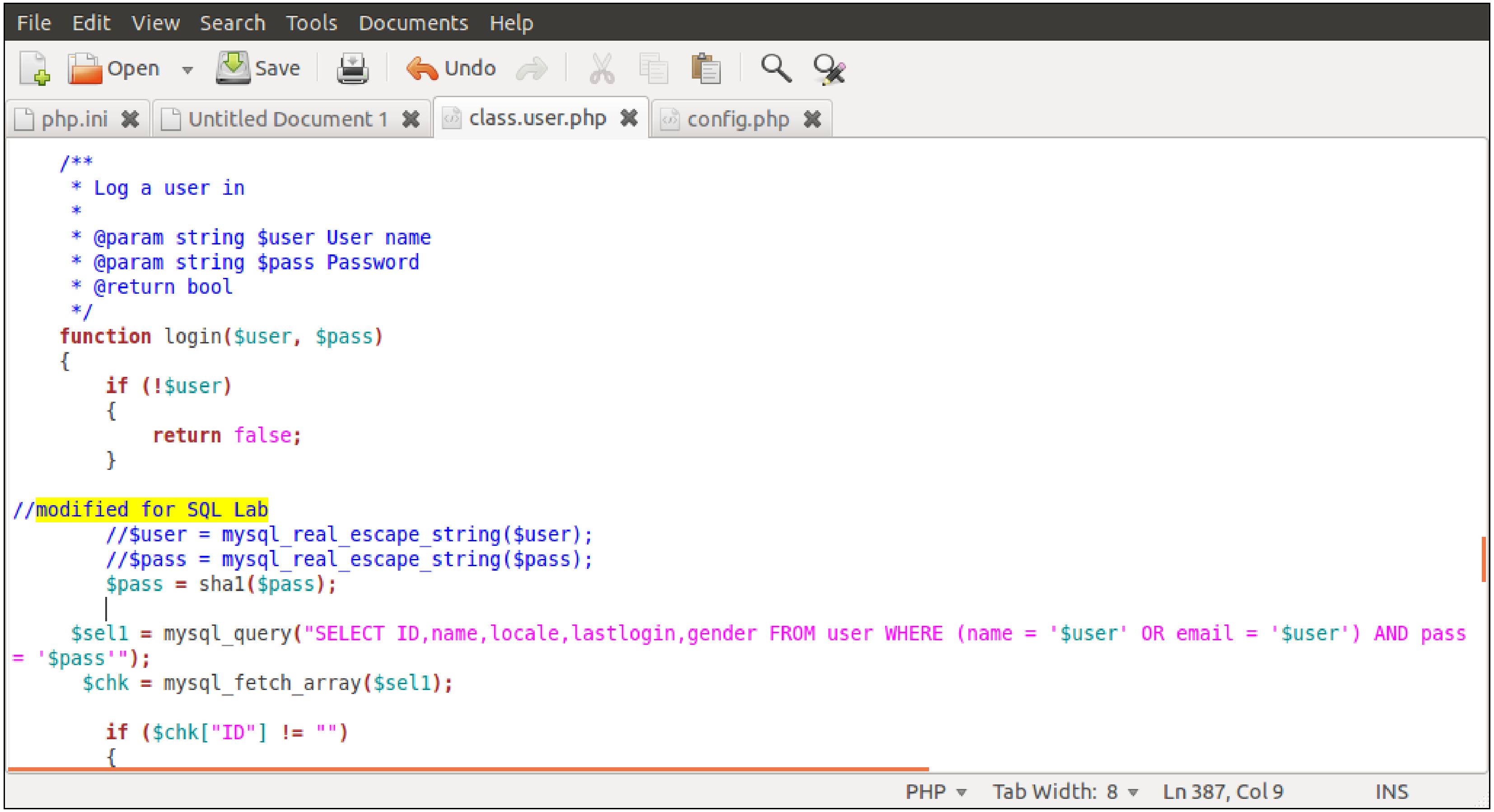
1> USERS_TABLE is a macro in PHP, and will be replaced by the users table;

2> $user holds the string typed in the Username textbox;

3> $pass holds the string typed in the Password textbox;

4> User inputs in these two textboxes are placed directly in the SQL query string.

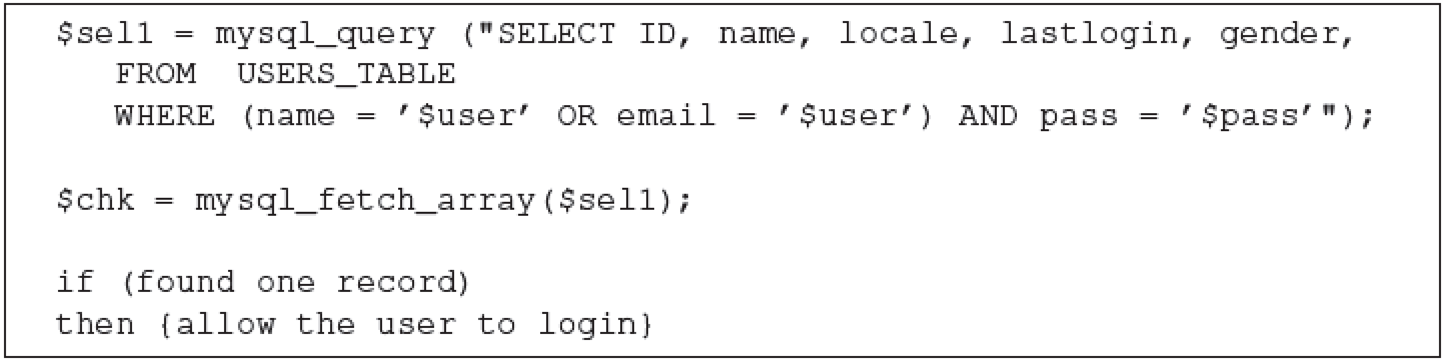
❖ Task 1: SQL Injection Attack on SELECT Statements



```
    /**
     * Log a user in
     *
     * @param string $user User name
     * @param string $pass Password
     * @return bool
     */
    function login($user, $pass)
    {
        if (!$user)
        {
            return false;
        }

//modified for SQL Lab
        //$user = mysql_real_escape_string($user);
        //$pass = mysql_real_escape_string($pass);
        $pass = sha1($pass);

     $sel1 = mysql_query("SELECT ID,name,locale,lastlogin,gender FROM user WHERE (name = '$user' OR email = '$user') AND pass
= '$pass'");
      $chk = mysql_fetch_array($sel1);

        if ($chk["ID"] != "")
        {
```

❖ Task 1: SQL Injection Attack on SELECT Statements

Login Code:



```
$sel1 = mysql_query ("SELECT ID, name, locale, lastlogin, gender,
   FROM  USERS_TABLE
   WHERE (name = ’$user’ OR email = ’$user’) AND pass = ’$pass’");

$chk = mysql_fetch_array($sel1);

if (found one record)
then {allow the user to login}
```

1> Online SQL Injection Tutorial

2> Attack Ways:

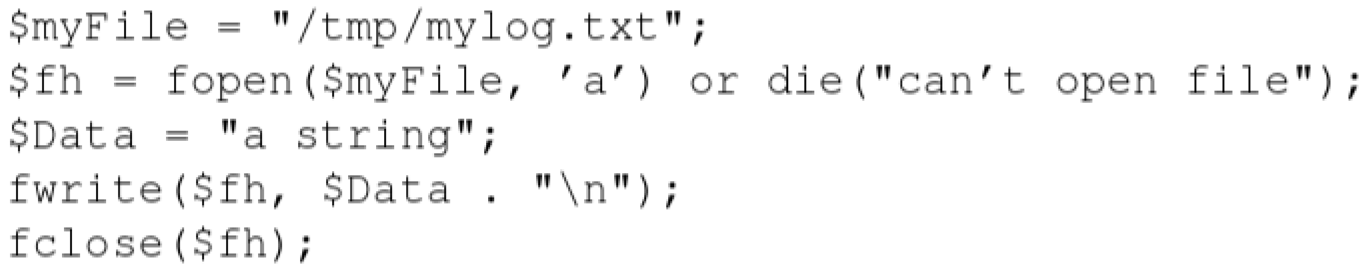
a. Comment out left part;

b. Make statement always true;

c. …

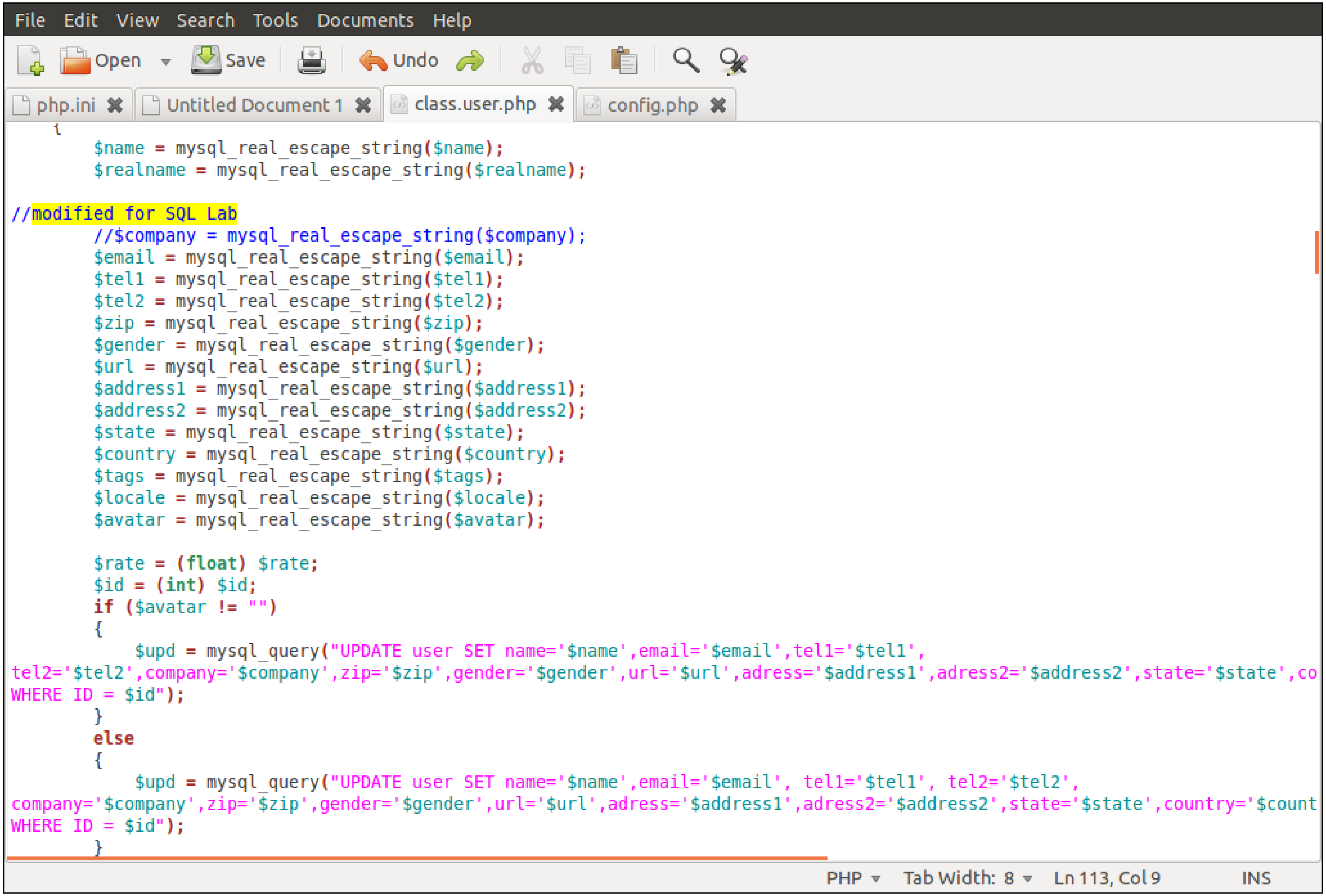
❖ Task 1: SQL Injection Attack on SELECT Statements

Debug in php:



```
$myFile = "/tmp/mylog.txt";
$fh = fopen($myFile, ’a’) or die("can’t open file");
$Data = "a string";
fwrite($fh, $Data . "\n");
fclose($fh);
```

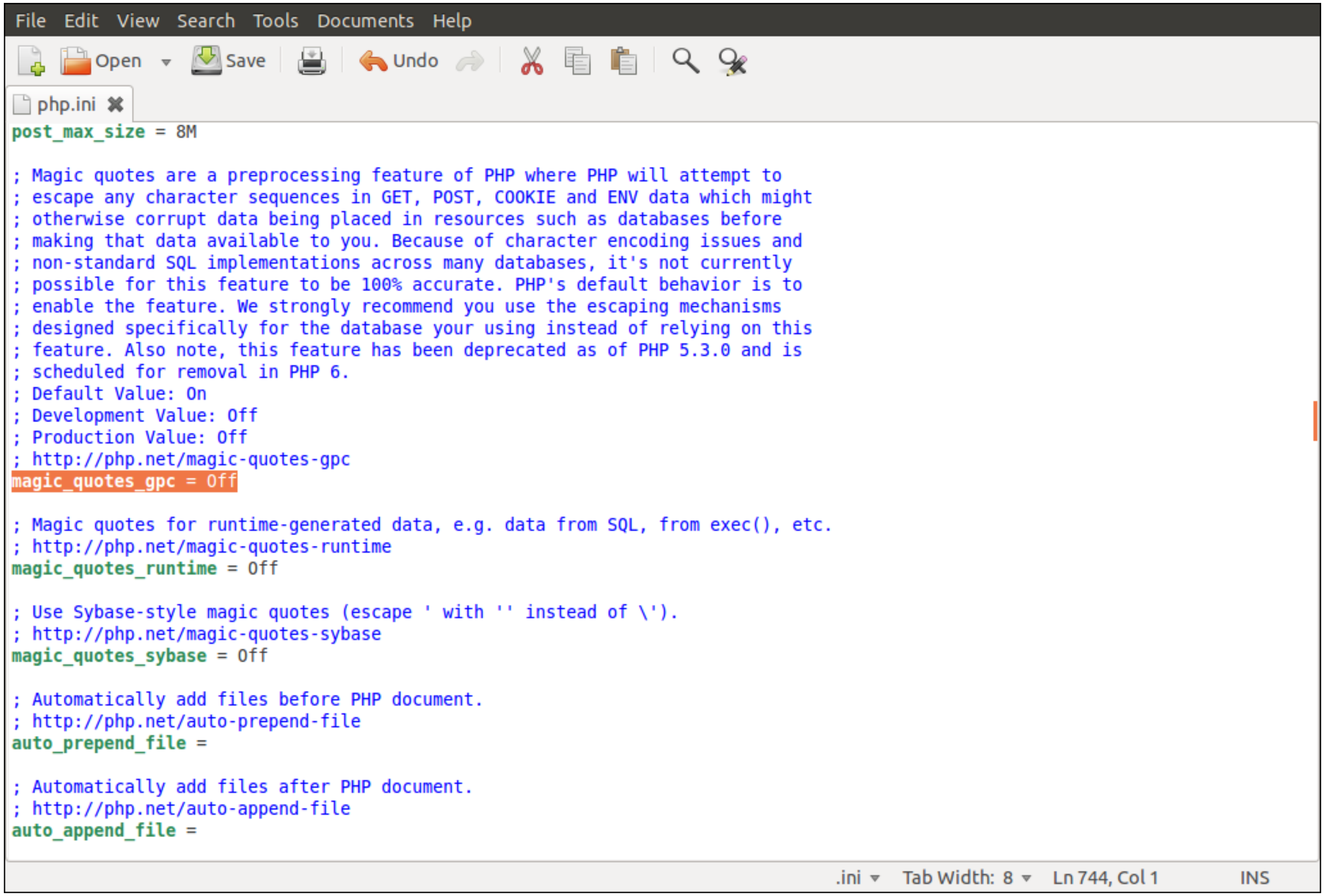
❖ Task 2: SQL Injection on UPDATE Statements



```
    {
        $name = mysql_real_escape_string($name);
        $realname = mysql_real_escape_string($realname);

//modified for SQL Lab
        //$company = mysql_real_escape_string($company);
        $email = mysql_real_escape_string($email);
        $tel1 = mysql_real_escape_string($tel1);
        $tel2 = mysql_real_escape_string($tel2);
        $zip = mysql_real_escape_string($zip);
        $gender = mysql_real_escape_string($gender);
        $url = mysql_real_escape_string($url);
        $address1 = mysql_real_escape_string($address1);
        $address2 = mysql_real_escape_string($address2);
        $state = mysql_real_escape_string($state);
        $country = mysql_real_escape_string($country);
        $tags = mysql_real_escape_string($tags);
        $locale = mysql_real_escape_string($locale);
        $avatar = mysql_real_escape_string($avatar);

        $rate = (float) $rate;
        $id = (int) $id;
        if ($avatar != "")
        {
            $upd = mysql_query("UPDATE user SET name='$name',email='$email',tel1='$tel1',
tel2='$tel2',company='$company',zip='$zip',gender='$gender',url='$url',adress='$address1',adress2='$address2',state='$state',co
WHERE ID = $id");
        }
        else
        {
            $upd = mysql_query("UPDATE user SET name='$name',email='$email', tel1='$tel1', tel2='$tel2',
company='$company',zip='$zip',gender='$gender',url='$url',adress='$address1',adress2='$address2',state='$state',country='$count
WHERE ID = $id");
        }
```

❖ Task 3: Countermeasures

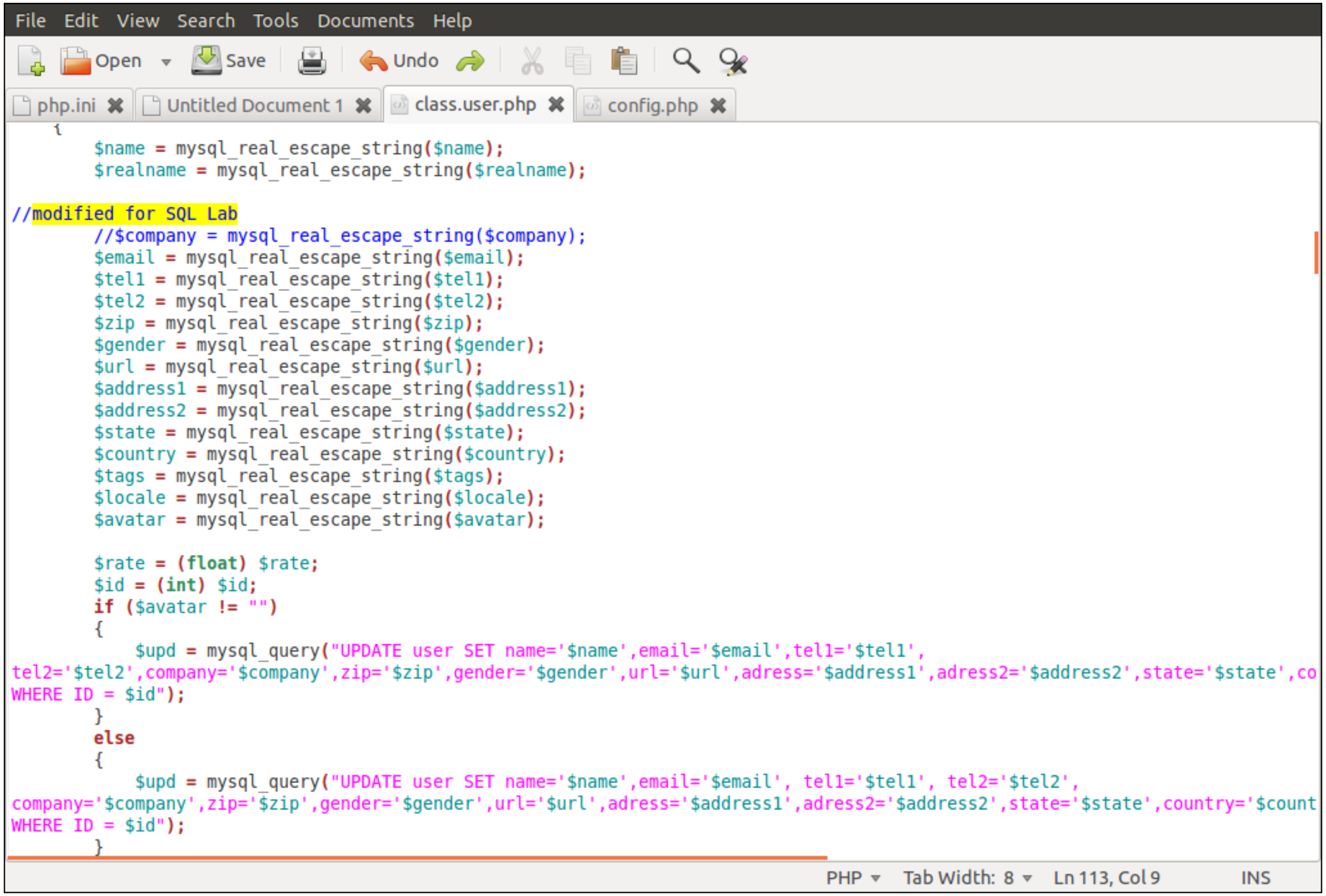
1> Escaping Special Characters using magic_quotes_gpc (global)



```
post_max_size = 8M

; Magic quotes are a preprocessing feature of PHP where PHP will attempt to
; escape any character sequences in GET, POST, COOKIE and ENV data which might
; otherwise corrupt data being placed in resources such as databases before
; making that data available to you. Because of character encoding issues and
; non-standard SQL implementations across many databases, it's not currently
; possible for this feature to be 100% accurate. PHP's default behavior is to
; enable the feature. We strongly recommend you use the escaping mechanisms
; designed specifically for the database your using instead of relying on this
; feature. Also note, this feature has been deprecated as of PHP 5.3.0 and is
; scheduled for removal in PHP 6.
; Default Value: On
; Development Value: Off
; Production Value: Off
; http://php.net/magic-quotes-gpc
magic_quotes_gpc = Off

; Magic quotes for runtime-generated data, e.g. data from SQL, from exec(), etc.
; http://php.net/magic-quotes-runtime
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ' with '' instead of \').
; http://php.net/magic-quotes-sybase
magic_quotes_sybase = Off

; Automatically add files before PHP document.
; http://php.net/auto-prepend-file
auto_prepend_file =

; Automatically add files after PHP document.
; http://php.net/auto-append-file
auto_append_file =
```

❖ Task 3: Countermeasures

2> Escaping Special Characters using mysql_real_escape_string (local)

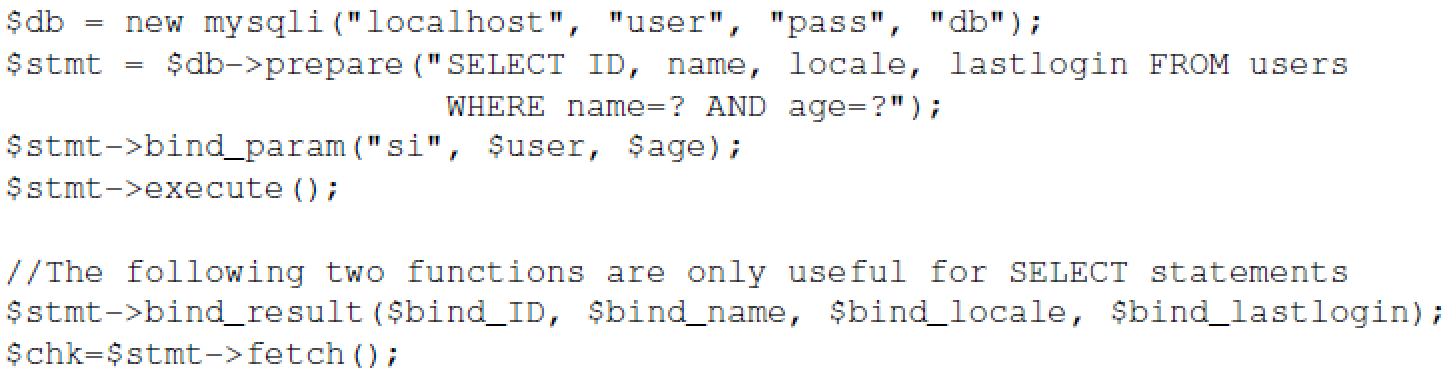


```
    {
        $name = mysql_real_escape_string($name);
        $realname = mysql_real_escape_string($realname);

//modified for SQL Lab
        //$company = mysql_real_escape_string($company);
        $email = mysql_real_escape_string($email);
        $tel1 = mysql_real_escape_string($tel1);
        $tel2 = mysql_real_escape_string($tel2);
        $zip = mysql_real_escape_string($zip);
        $gender = mysql_real_escape_string($gender);
        $url = mysql_real_escape_string($url);
        $address1 = mysql_real_escape_string($address1);
        $address2 = mysql_real_escape_string($address2);
        $state = mysql_real_escape_string($state);
        $country = mysql_real_escape_string($country);
        $tags = mysql_real_escape_string($tags);
        $locale = mysql_real_escape_string($locale);
        $avatar = mysql_real_escape_string($avatar);

        $rate = (float) $rate;
        $id = (int) $id;
        if ($avatar != "")
        {
            $upd = mysql_query("UPDATE user SET name='$name',email='$email',tel1='$tel1',
tel2='$tel2',company='$company',zip='$zip',gender='$gender',url='$url',adress='$address1',adress2='$address2',state='$state',co
WHERE ID = $id");
        }
        else
        {
            $upd = mysql_query("UPDATE user SET name='$name',email='$email', tel1='$tel1', tel2='$tel2',
company='$company',zip='$zip',gender='$gender',url='$url',adress='$address1',adress2='$address2',state='$state',country='$count
WHERE ID = $id");
        }
```

❖ Task 3: Countermeasures

3> Prepare Statement

SELECT:

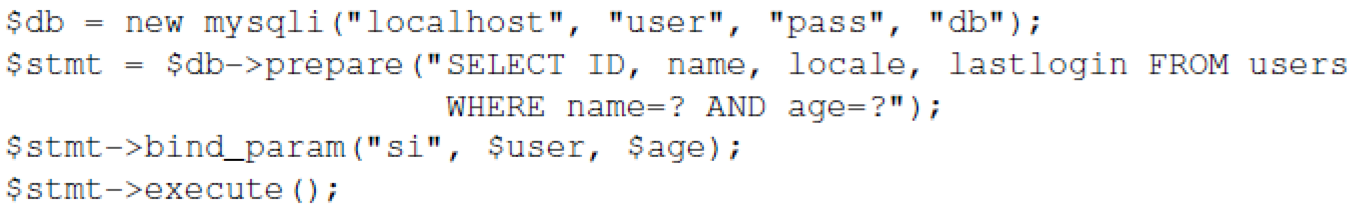


```
$db = new mysqli("localhost", "user", "pass", "db");
$stmt = $db->prepare("SELECT ID, name, locale, lastlogin FROM users
                      WHERE name=? AND age=?");
$stmt->bind_param("si", $user, $age);
$stmt->execute();

//The following two functions are only useful for SELECT statements
$stmt->bind_result($bind_ID, $bind_name, $bind_locale, $bind_lastlogin);
$chk=$stmt->fetch();
```

❖ Task 3: Countermeasures

3> Prepare Statement

UPDATE:



```
$db = new mysqli("localhost", "user", "pass", "db");
$stmt = $db->prepare("SELECT ID, name, locale, lastlogin FROM users
                      WHERE name=? AND age=?");
$stmt->bind_param("si", $user, $age);
$stmt->execute();
```

- **Grade Criteria**

Task1: 30 %
Task2: 30 %
Task4: 40 %