



PHP

Email app
Visitor counter app
Using Wamp
Database handling
SQL basic operations



PHP

Sending email
here is the form

```
<h1>Comments</h1>
<form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="post">
  <p>Name: <input type="text" name="name" /></p>
  <p>Email: <input type="text" name="email" /></p>
  <p>Comments:</p>
  <p><textarea name="comments"></textarea></p>
  <p><input type="submit" value="Send!" /></p>
</form>
```



PHP

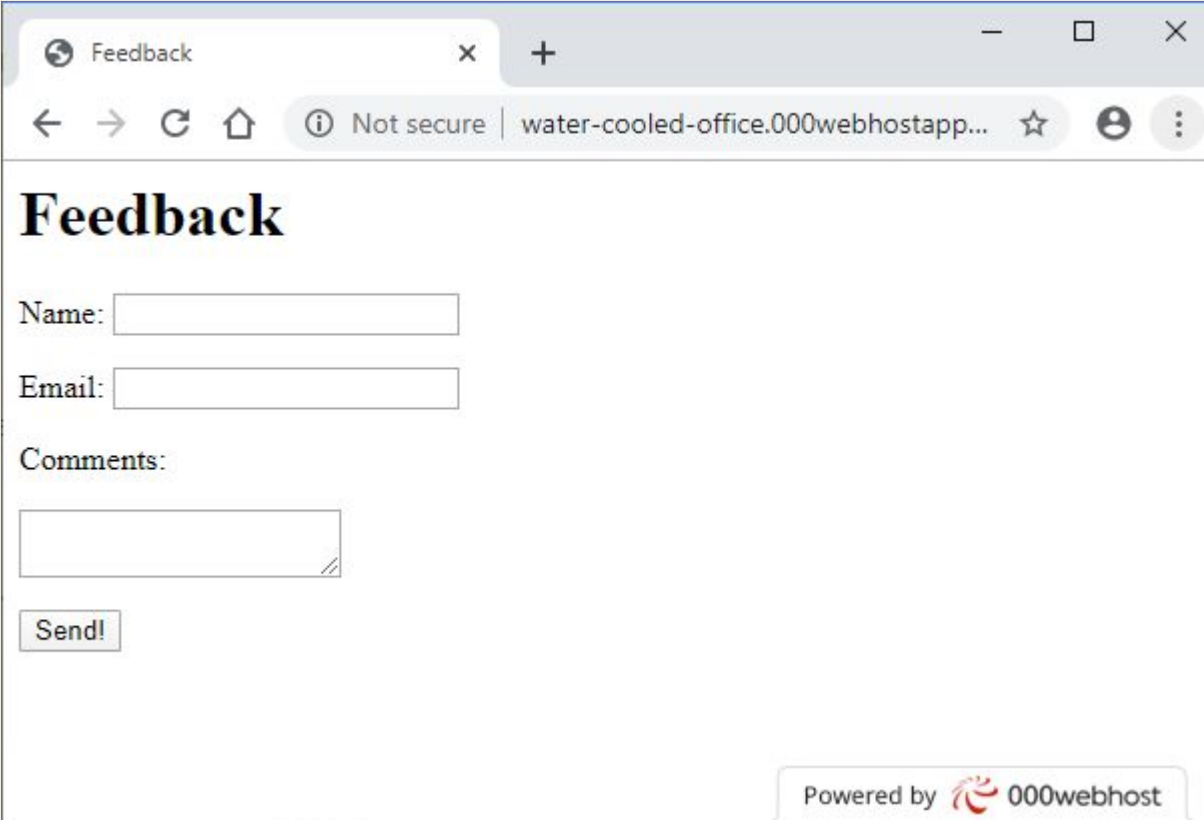
Sending email
here is sending code

```
if ($_POST)
{
    $name = $_POST['name'];
    $email = $_POST['email'];
    $comments = $_POST['comments'];
    $headers = "From: donald.duck@disney.com";

    // Here is the message:
    $message = "Comments : \n$comments";
    mail("darry@darry.com", "Feedback", $message, $headers);
    echo "Thank you!\n";
}
```

PHP

Sending email
- put the code to
web server




The screenshot shows a web browser window with a single tab titled "Feedback". The address bar indicates the URL is "water-cooled-office.000webhostapp..." and the connection is "Not secure". The page content features a heading "Feedback" in a large, bold, black serif font. Below the heading are three input fields: "Name:" followed by a text box, "Email:" followed by a text box, and "Comments:" followed by a larger text area with a diagonal line in the bottom right corner. A "Send!" button is positioned below the comments field. At the bottom right of the page, there is a footer that reads "Powered by" followed by the 000webhost logo and the text "000webhost".

Feedback

Name:

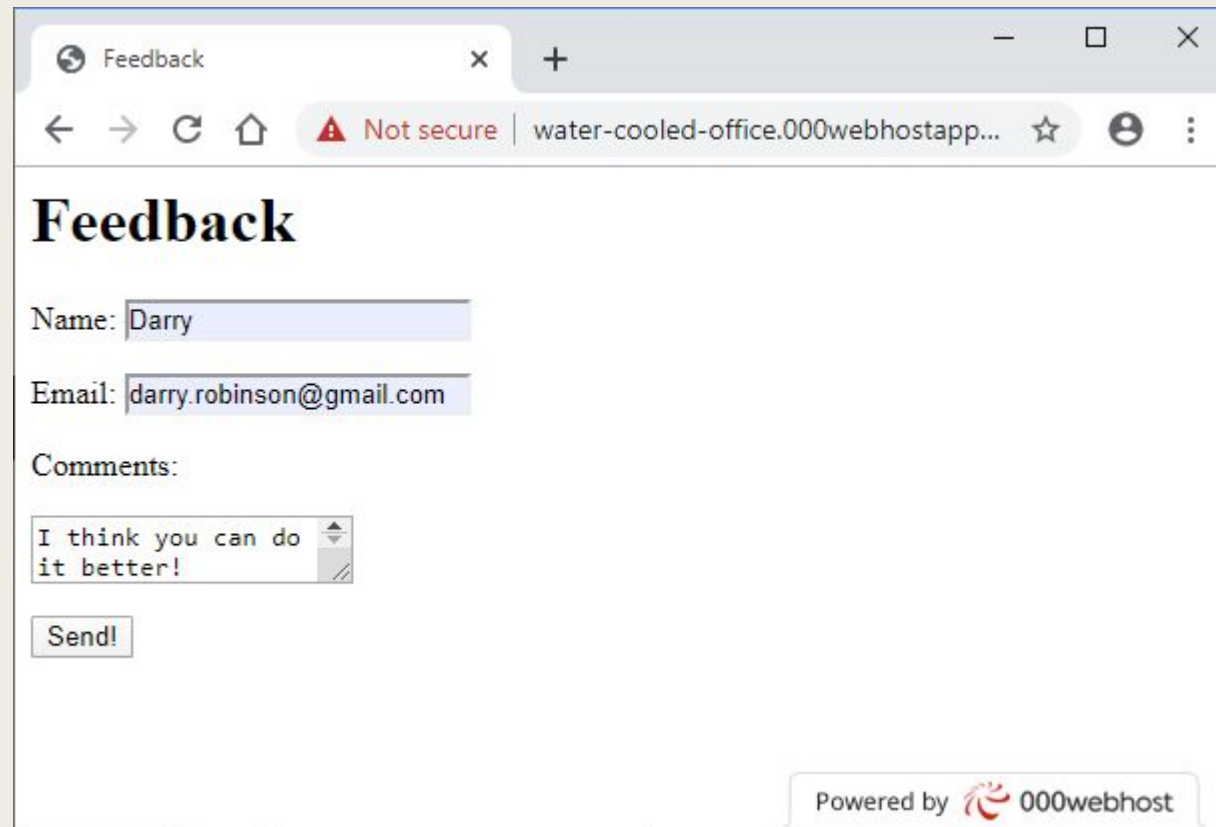
Email:

Comments:

Powered by  000webhost

PHP

Sending email
- test run



The screenshot shows a web browser window with a single tab titled 'Feedback'. The address bar indicates the URL is 'water-cooled-office.000webhostapp...' and shows a 'Not secure' warning. The page content includes a heading 'Feedback' followed by three input fields: 'Name:' with the value 'Darry', 'Email:' with the value 'darry.robinson@gmail.com', and 'Comments:' with the text 'I think you can do it better!'. A 'Send!' button is located below the comments field. At the bottom right, there is a footer that reads 'Powered by 000webhost'.

Feedback

Name: Darry

Email: darry.robinson@gmail.com

Comments:

I think you can do it better!

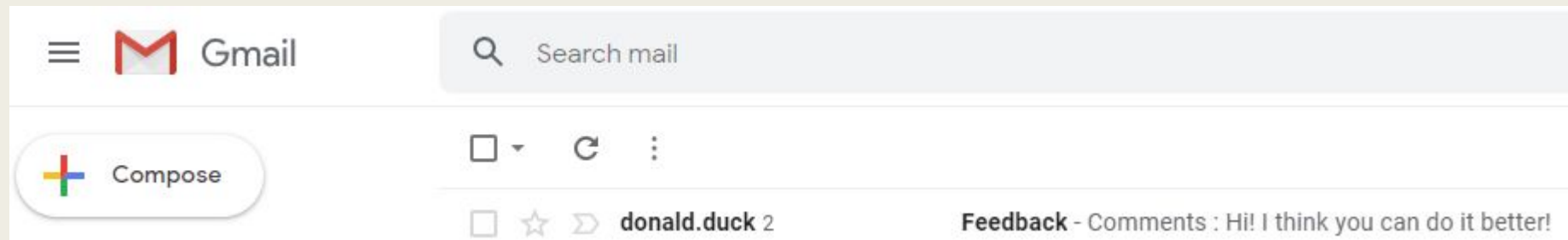
Send!

Powered by 000webhost



PHP

Sending email
- test run



Yeah, got email from Donald
Duck!



PHP

Own chat:
UI form here:

Darry's Chat





Nickname?:

Your message:



PHP

Own chat:
Textfile has to have write
rights

Name	Size	Type	Date Modified	Permissions
 ./	-	Current	Dec 16 2019 18:21:37	drwxr-xr-x (755)
 ../	-	Parent	Dec 16 2019 18:20:04	drwxr-xr-x (755)
 chat.php*	694 B	PHP script	Dec 16 2019 18:25:13	-rwxr--r-- (744)
 chat.txt	0 B	Text file	Dec 16 2019 18:21:37	-rw-rw-rw- (666)



PHP

Own chat:
Testing

Darry's Chat

Nickname?:

Your message:

Send

Ricky: I am here!

Jekaterina: Anybody here?

Jekaterina: ??



PHP

Own chat:
Code 1)

```
<?
session_start();
?>

<h1> Darry Chat </h1>
<form action = "chat.php" method = "post">

Nickname?: <br>
<input type=text name="nick"> <br>
Your message: <br>
<input type=text name="mess"> <br>
<input type=submit name="send" value="Send"> <br>
</form>
```



PHP

Own chat:
Code 2)

```
<?
// Write to the text file

if (isset($_POST['nick']))|
{
    $nick = $_POST['nick'];
    if ( !$_POST['nick'] == "")
        $_SESSION["nick"] = $nick;
}

if (isset($_POST['mess']))
{
    $v = $_POST['mess'];
    $tt = fopen("chat.txt", "a");
    fwrite($tt, $_SESSION["nick"] . ": " . $v . "\n");
    fclose($tt);
}

$filename = file("chat.txt");
$rows = count($filename);
for ($k = $rows - 1; $k > 0; $k--)
    print $filename[$k] . "<br>";

?>
```



PHP

Visitor counter

File amount.txt is updated when new user visit the page

```
<?
// read previous amount from textfile
$tt = fopen("amount.txt", "r");
$row = fgets($tt, 10);
fclose($tt);
// add 1
$row = $row + 1;
print "Amount of visitors now: ". $row;

// write new value to the textfile
$tt = fopen("amount.txt", "w");
fwrite($tt, $row);
fclose($tt);
?>
```



PHP

Visitor counter

Remember permissions!

File Name	Size	Type	Modified
<input checked="" type="checkbox"/> amount.txt	5 B	Text file	Dec 16 2019 18:41:34
<input type="checkbox"/> counter.php	278 B	PHP script	Dec 16 2019 18:41:08
Total: 2 files, Total file size: 283 B			

Set Permissions

Owner: ☒ Read ☒ Write ☐ Execute

Group: ☒ Read ☐ Write ☐ Execute

Public: ☒ Read ☒ Write ☐ Execute

Numeric value: ☐ Set Recursive

Set

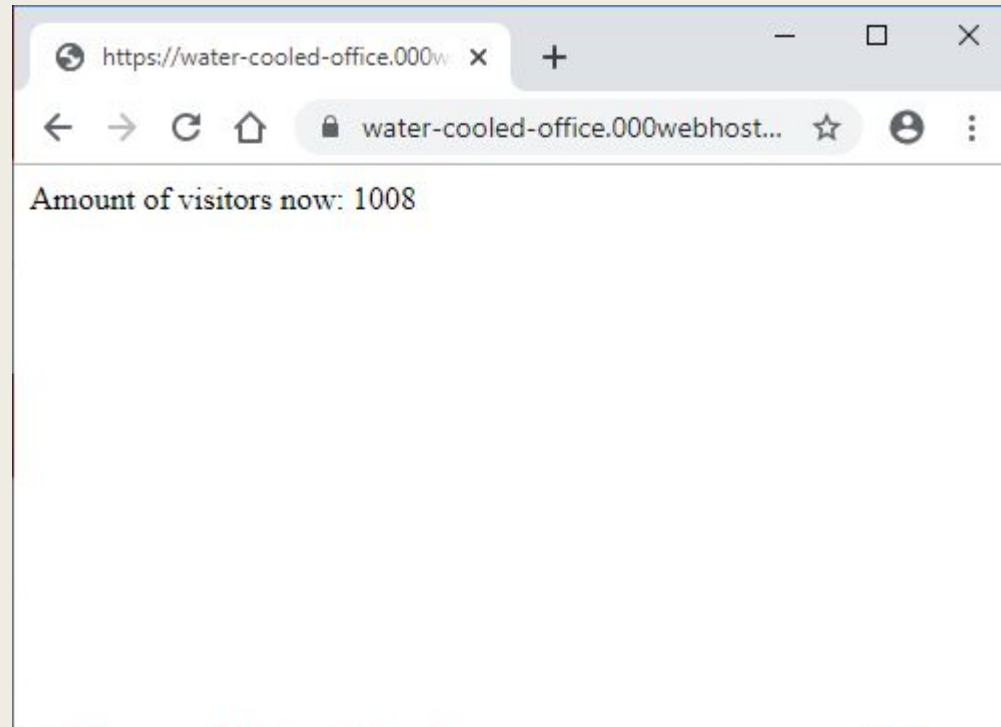
Cancel



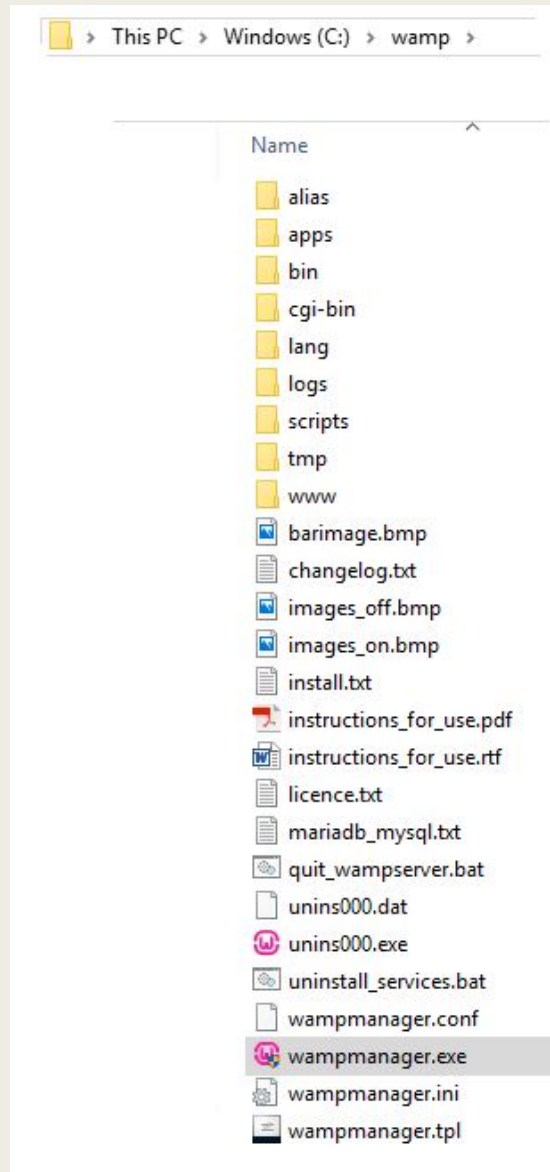
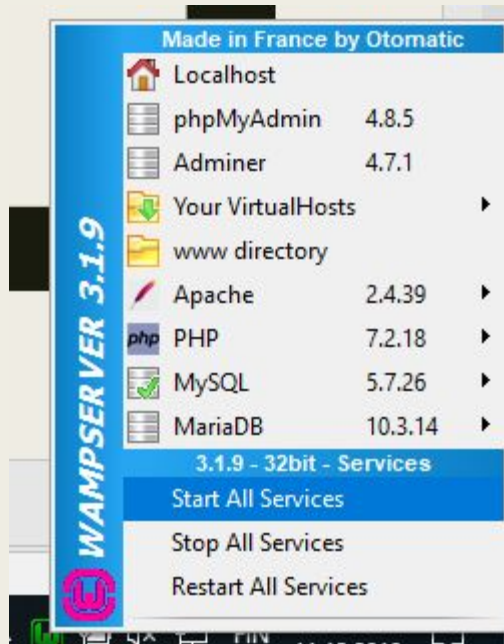
PHP

Visitor counter

Test!

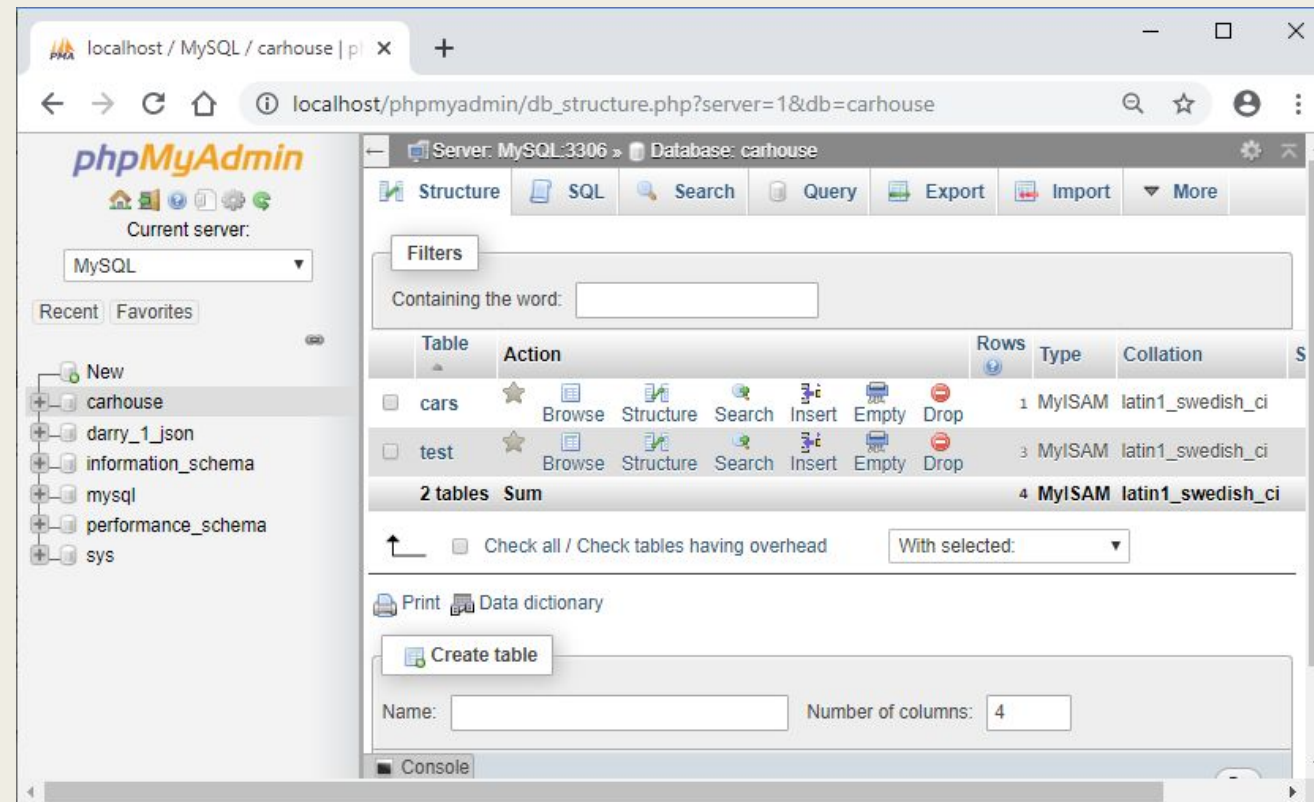


PHP & MySQL Start WAMP

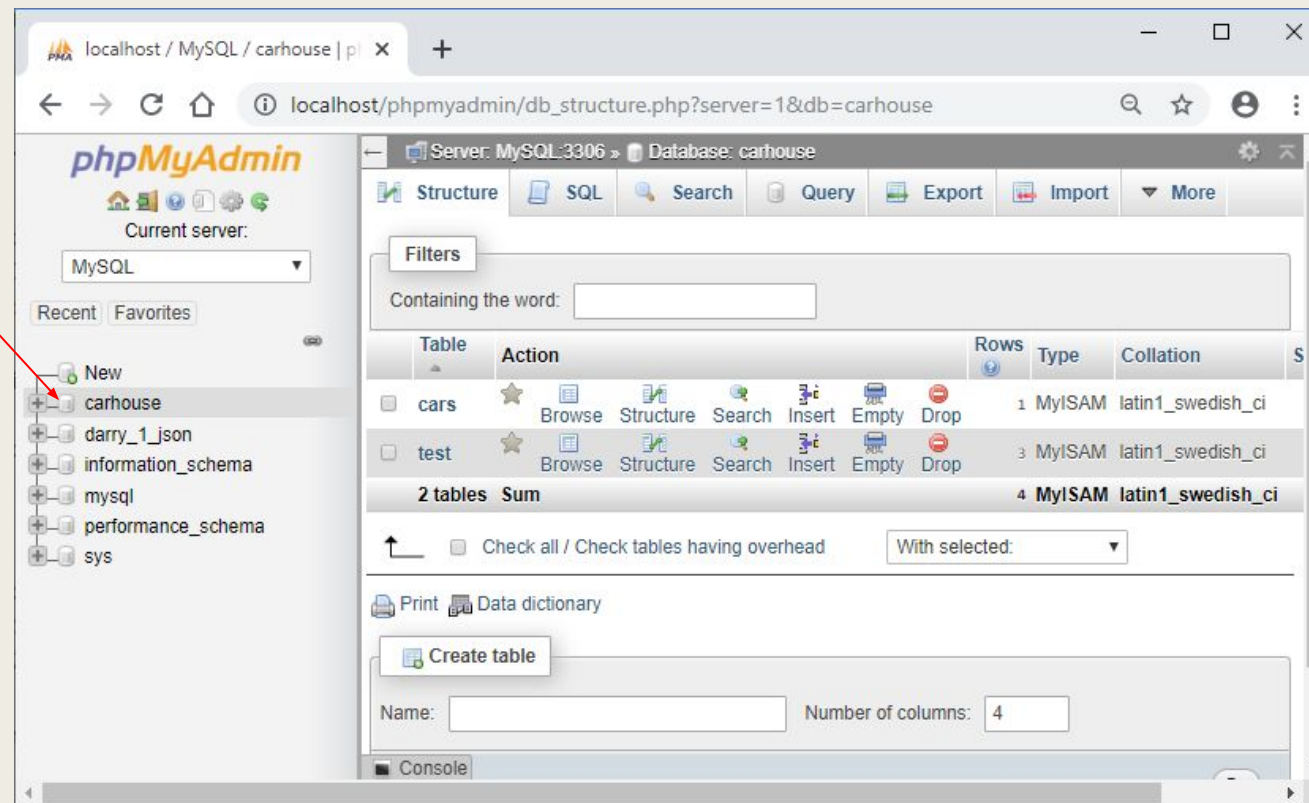


PHP & MySQL

Test and open
phpMyAdmin



PHP & MySQL
Create database
carhouse





PHP & MySQL

Connection code

```
<?php
// host
$host = "localhost";
// database
$user = "root";

// database password
$pass = "";

// database name
$dbase = "carhouse";

$connection = mysqli_connect($host, $user, $pass);
if (!$connection)
{
    echo '<p>Can not connect now<p>';
    //exit();
}

if (!mysqli_select_db($connection,$dbase)) {
    exit('Can not choose database.');
```

```
    else
        print 'Connected :)';
```

```
?>
```

PHP & MySQL

Create cars
table

```
create_cars_table.php - Notepad
File Edit Format View Help
<?php
include 'carhouse_connect.php';

$sql = "CREATE TABLE cars (
    regnr varchar(10),
    make varchar(30),
    yearmodel int,
    mileage int
);";

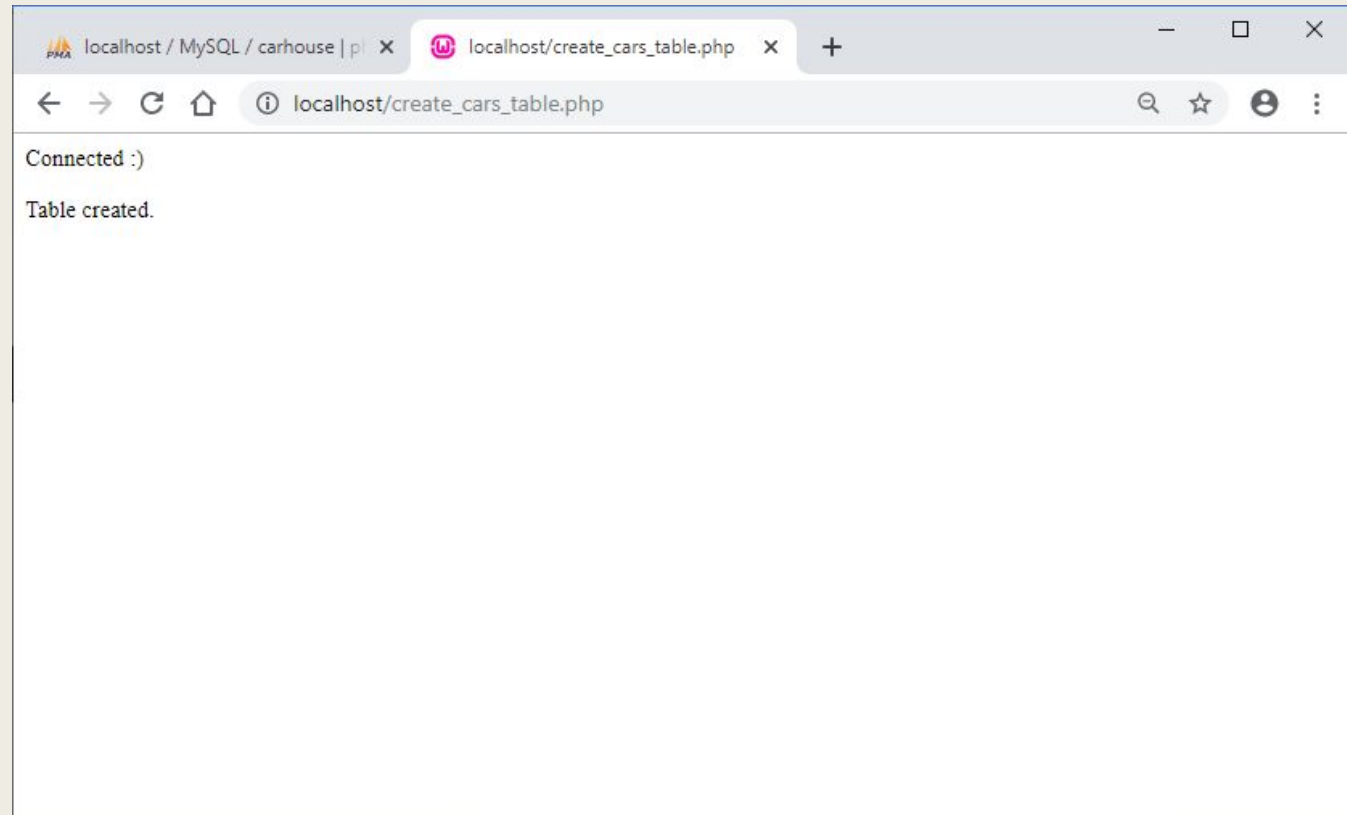
if (mysqli_query($connection, $sql)) {
    echo '<p>Table created.</p>';
}
else
{
    echo '<p>Failed: check the sql:';
    echo $sql;
}

?>
```



PHP & MySQL

Create cars
table
- test



PHP & MySQL

Create cars
table
- test

The screenshot shows the phpMyAdmin web interface in a browser. The address bar indicates the URL is `localhost/phpmyadmin/sql.php?db=carhouse&table=cars...`. The interface is divided into several sections:

- Left Panel:** Displays the database structure. Under the 'carhouse' database, the 'cars' table is selected. Other databases listed include 'darry_1_json', 'information_schema', 'mysql', 'performance_schema', and 'sys'.
- Top Panel:** Shows the current server as 'MySQL' and the selected database as 'carhouse'.
- Right Panel:** Displays the table structure for 'cars'. The columns are 'regnr', 'make', 'yearmodel', and 'mileage'. Below the structure, a message states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)'. The SQL query shown is `SELECT * FROM 'cars'`. Below the message, there are links for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'.

A line from the text 'Create cars table - test' points to the 'cars' table in the left panel.



PHP & MySQL

Add
data

A screenshot of a web browser window displaying a form titled "Give car info:". The form contains several input fields and two buttons. The browser's address bar shows "localhost/add_data_to_cars.php".

Give car info:
Register number:

Make:

Yearmodel:

Mileage:

Submit:

Clear:

Connected :)

PHP & MySQL

Add
data

+ Options

regnr	make	yearmodel	mileage
ABC-112	Volvo	2000	12345

localhost/add_data_to_cars.php x +

localhost/add_data_to_cars.php

Give car info:

Register number:

Make:

Yearmodel:

Mileage:

Submit:

Clear:

Connected :)

Data added.



PHP & MySQL

Add data - codes

```
<?php
$this_filename = $_SERVER['PHP_SELF'];
?>
<html>
<body>

<form action="<?php echo $this_filename; ?>" method="post">
Give car info: <br>

Register number:<br>
<input type="text" name="reg" size="10"> <br>
Make: <br>
<input type="text" name="make" size="30"> <br>
Yearmodel:<br>
<input type="text" name="year" size="5"> <br>
Mileage:<br>
<input type="text" name="miles" size="10"> <br>
Submit:<br>
<input type="submit" value="Submit"> <br>
Clear:<br>
<input type="reset" value="Clear All">

</form>
```




PHP & MySQL

Add data -
codes

```
<?php
include 'connect_to_carhouse.php';

if (isset($_POST['reg']))
{ $m1 = $_POST['reg']; }
if (isset($_POST['make']))
$m2 = $_POST['make'];
if (isset($_POST['year']))
$m3 = $_POST['year'];
if (isset($_POST['miles']))
$m4 = $_POST['miles'];
settype($m3, "int");settype($m4, "int");
if (isset($_POST['reg']))
{
$sql =
"INSERT INTO `cars`(`regnr`, `make`, `yearmodel`,`mileage` )
VALUES ('$m1','$m2',$m3,$m4)";

if (mysqli_query($connection, $sql)) {
echo '<p>Data added.</p>';
}
else
{
echo '<p>Failed: ';
// echo $sql; for testing ...
}

}
?>
```



PHP & MySQL

Try it!

Then add operations

printing

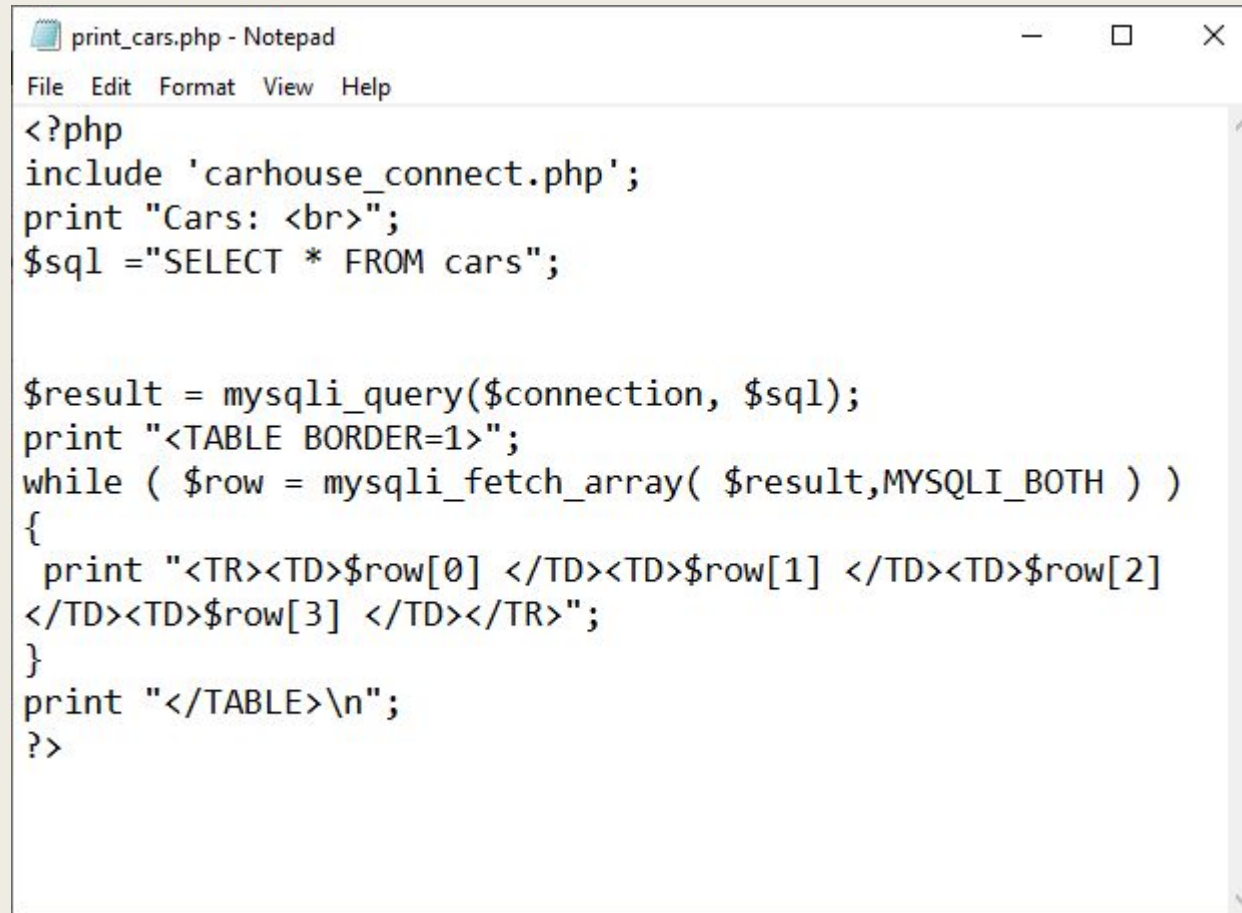
searching

deleting

updating

PHP & MySQL

Database operations:
printing

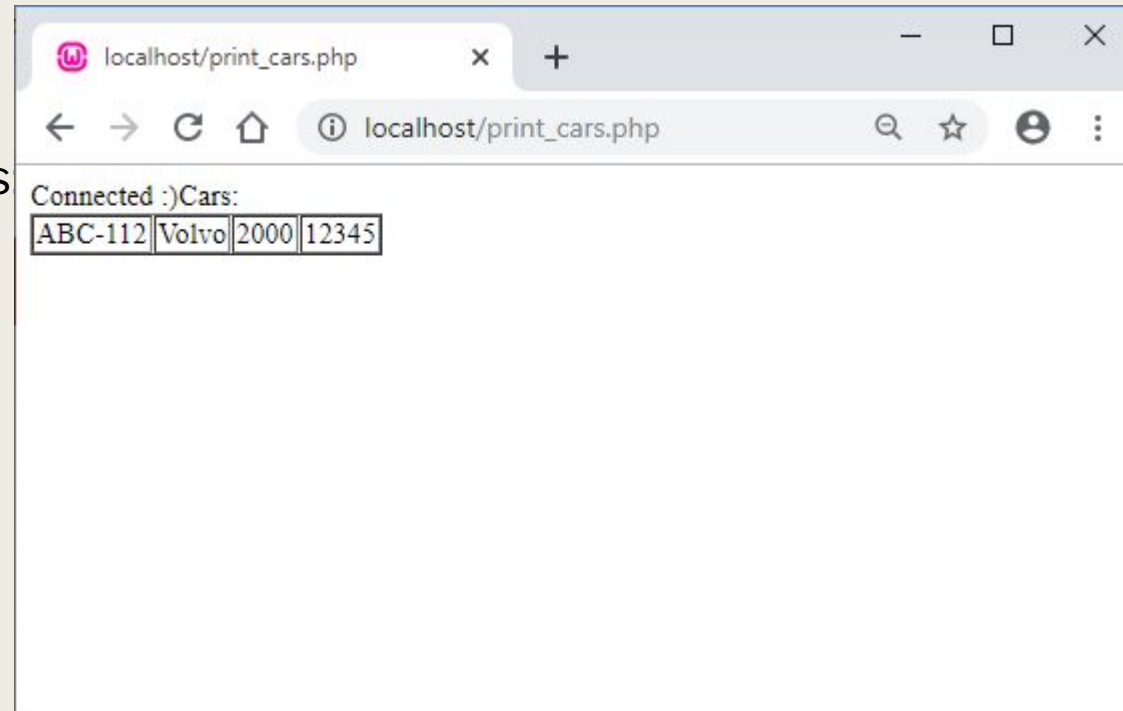


```
print_cars.php - Notepad
File Edit Format View Help
<?php
include 'carhouse_connect.php';
print "Cars: <br>";
$sql ="SELECT * FROM cars";

$result = mysqli_query($connection, $sql);
print "<TABLE BORDER=1>";
while ( $row = mysqli_fetch_array( $result,MYSQLI_BOTH ) )
{
    print "<TR><TD>$row[0] </TD><TD>$row[1] </TD><TD>$row[2]
</TD><TD>$row[3] </TD></TR>";
}
print "</TABLE>\n";
?>
```

PHP & MySQL

Database operations
printing





PHP & MySQL

Database operations:
searching

```
<head>
<title>Search cars</title>
</head>
<body>
<?php
include 'carhouse_connect.php';
$this_filename = $_SERVER['PHP_SELF'];
?>

<form action="<?php echo $this_filename; ?>" method="post">
<br>
Give reg.number: <br>
<input type="text" name="regnr" size="20">
<input type="submit" value="Search">
<input type="reset" value="Clear All">
</form>
```



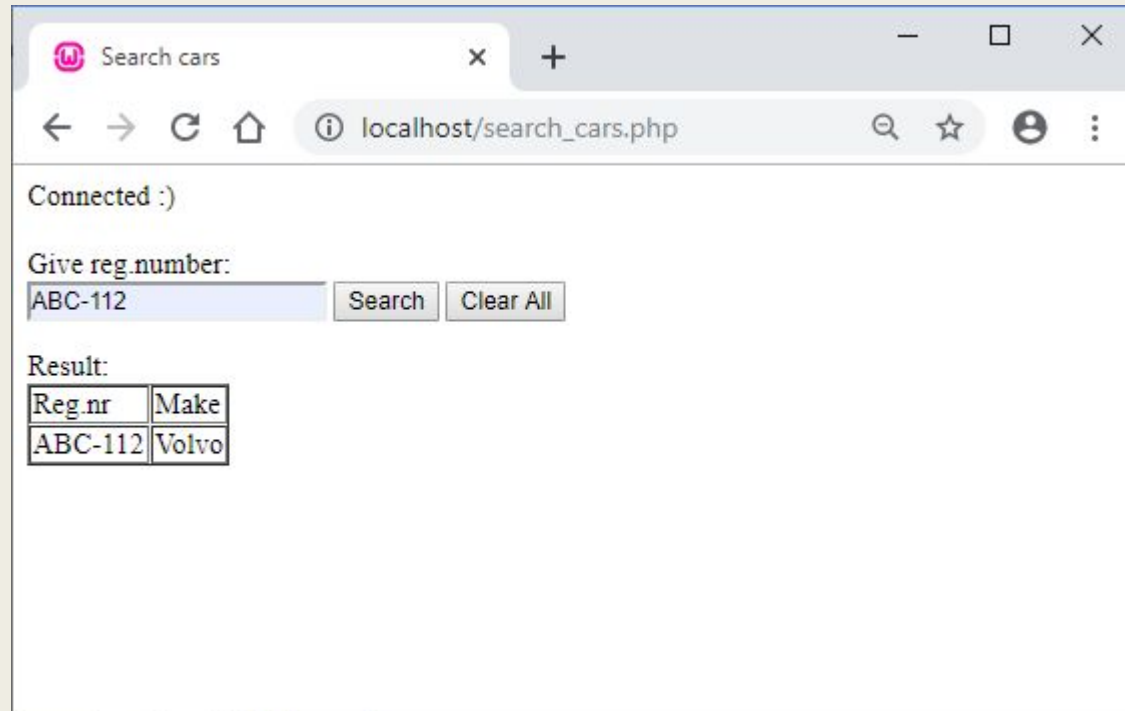
PHP & MySQL

Database operations:
searching

```
<?php
$sql = "";
if (isset($_POST['regnr']))
{
    $code = $_POST['regnr'];
    $sql ="SELECT * FROM cars WHERE regnr = '$code'";
    if ($result = mysqli_query($connection, $sql))
    {
        print "Result: <br>";
        print "<TABLE BORDER=1>";
        print "<TR><TD>Reg.nr</TD><TD>Make</TD></TR>";
        while ( $row = mysqli_fetch_array( $result ) )
        {
            print "<TR><TD>$row[0] </TD><TD>$row[1] </TD></TR>";
        }
        print "</TABLE>\n";
    }
else
{
    echo 'Can not print ..';
} }
?>
</body>
</html>
```

PHP & MySQL

Database operations:
searching



Search cars

localhost/search_cars.php

Connected :)

Give reg.number:

ABC-112 Search Clear All

Result:

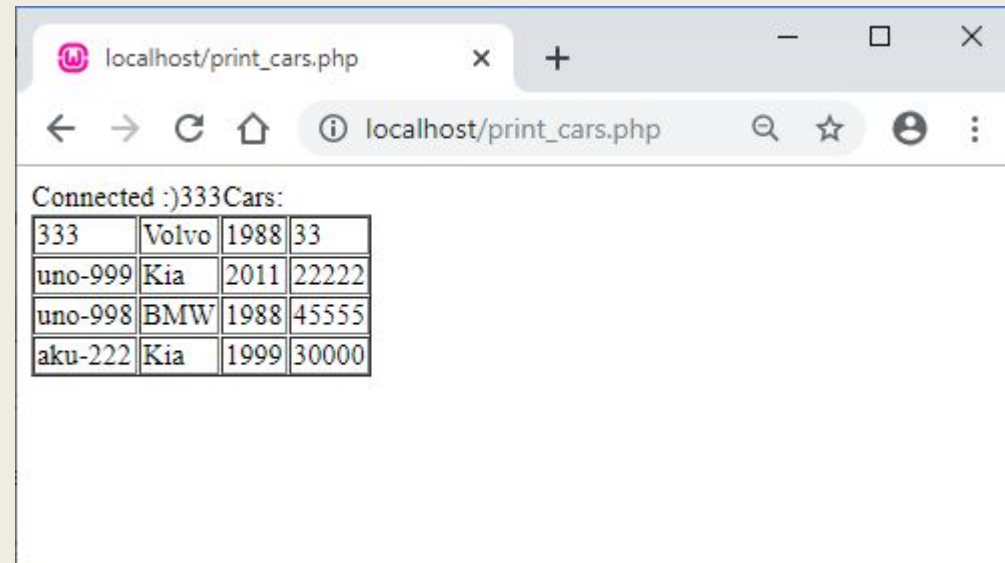
Reg.nr	Make
ABC-112	Volvo



PHP & MySQL

Database operations
deleting

We have
these cars



The screenshot shows a web browser window with the address bar displaying 'localhost/print_cars.php'. The page content includes the text 'Connected :)333Cars:' followed by a table with four rows of car data. The table has four columns: license plate, brand, year, and price.

Connected :)333Cars:			
333	Volvo	1988	33
uno-999	Kia	2011	22222
uno-998	BMW	1988	45555
aku-222	Kia	1999	30000



PHP & MySQL

Database operations deleting

Code

```
<?php
include 'carhouse_connect.php';

$this_filename = $_SERVER['PHP_SELF'];

print "<h4>Remove this car:</h4>";
?>

<form action="<?php echo $this_filename; ?>" method="post">

<label>Register Nr:<br/>
<input type = "text" name = "regnr" ><br>
<input type="submit" value="SUBMIT" /> <br>
<input type="reset" value="RESET" />
</form>
<?php

$code = "";
if (isset($_POST['regnr']))
{ $code = $_POST['regnr'];
}
$sql = "DELETE FROM cars WHERE regnr = '$code'";
if ($result = mysqli_query($connection, $sql))
{
    print "<br>Deleted...: <br>";
}
else
    print "Not found...: <br>";
?>
```



PHP & MySQL

Database operations
deleting

Testing

The screenshot shows a web browser window with the address bar displaying 'localhost/delete_cars.php'. The page content includes a status message 'Connected :)', a heading 'Remove this car:', a text input field labeled 'Register Nr:' containing the value 'uno-998', and two buttons labeled 'SUBMIT' and 'RESET'. Below these elements is a label 'Deleted...:'.

localhost/delete_cars.php

Connected :)

Remove this car:

Register Nr:
uno-998

SUBMIT

RESET

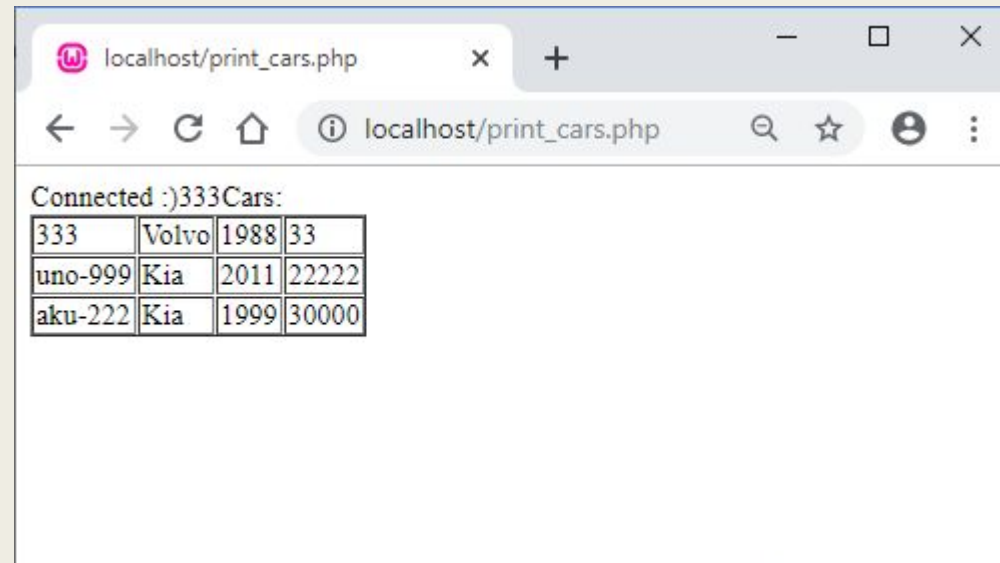
Deleted...:



PHP & MySQL

Database operations
deleting

Testing –
removed...



The screenshot shows a web browser window with the address bar displaying 'localhost/print_cars.php'. The page content shows the text 'Connected :)333Cars:' followed by a table of car data.

333	Volvo	1988	33
uno-999	Kia	2011	22222
aku-222	Kia	1999	30000



PHP & MySQL

Database operations updating

A screenshot of a web browser window displaying a form for updating car information. The browser's address bar shows the URL `localhost/update_this_car.php`. The page content includes a message stating that the car's info is to be updated and that the file `"cars_update.php"` shows the car's info and it can be updated. Below this, there is a label `Reg.nr:` followed by a text input field containing the value `uno-999`. At the bottom of the form, there are two buttons: `SUBMIT` and `RESET`.

localhost/update_this_car.php

← → ↻ 🏠 ⓘ localhost/update_this_car.php 🔍 ☆ 👤 ⋮

This car's info is to be updated...
File "cars_update.php" shows car's info and it can be updated...

Reg.nr:

uno-999

SUBMIT

RESET



PHP & MySQL

Database operations
updating
Code 1) which car
to update?

```
<html>

<body>

This car's info is to be updated... <br>
File "cars_update.php" shows car's info and it can be updated... <br>

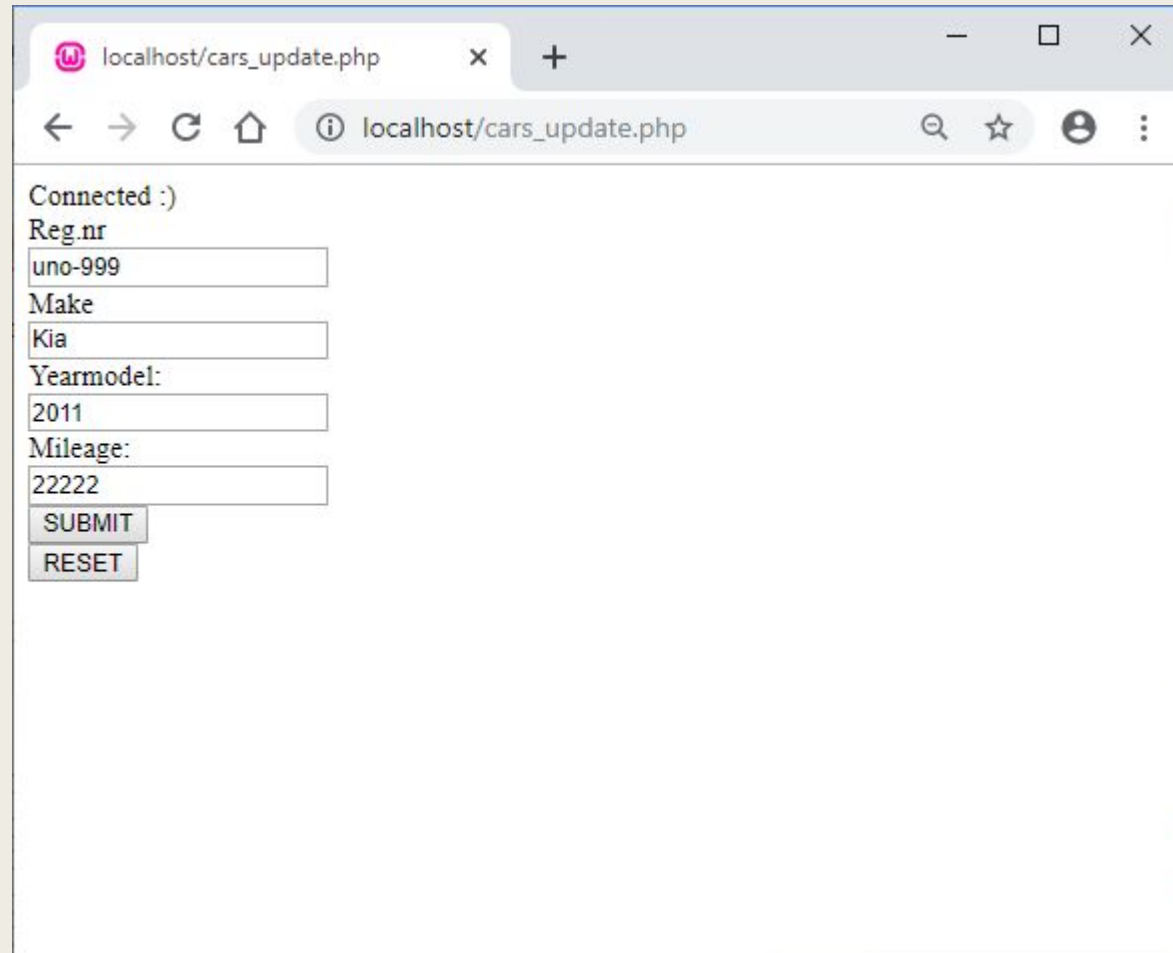
<form action="cars_update.php" method="post">

<label>Reg.nr:<br/>
<input type = "text" name = "y" ><br>
<input type="submit" value="SUBMIT" /> <br>
<input type="reset" value="RESET" />
</form>

</body>
</html>
```

PHP & MySQL

Database operations
updating



localhost/cars_update.php

localhost/cars_update.php

Connected :)

Reg.nr

Make

Yearmodel:

Mileage:



PHP & MySQL

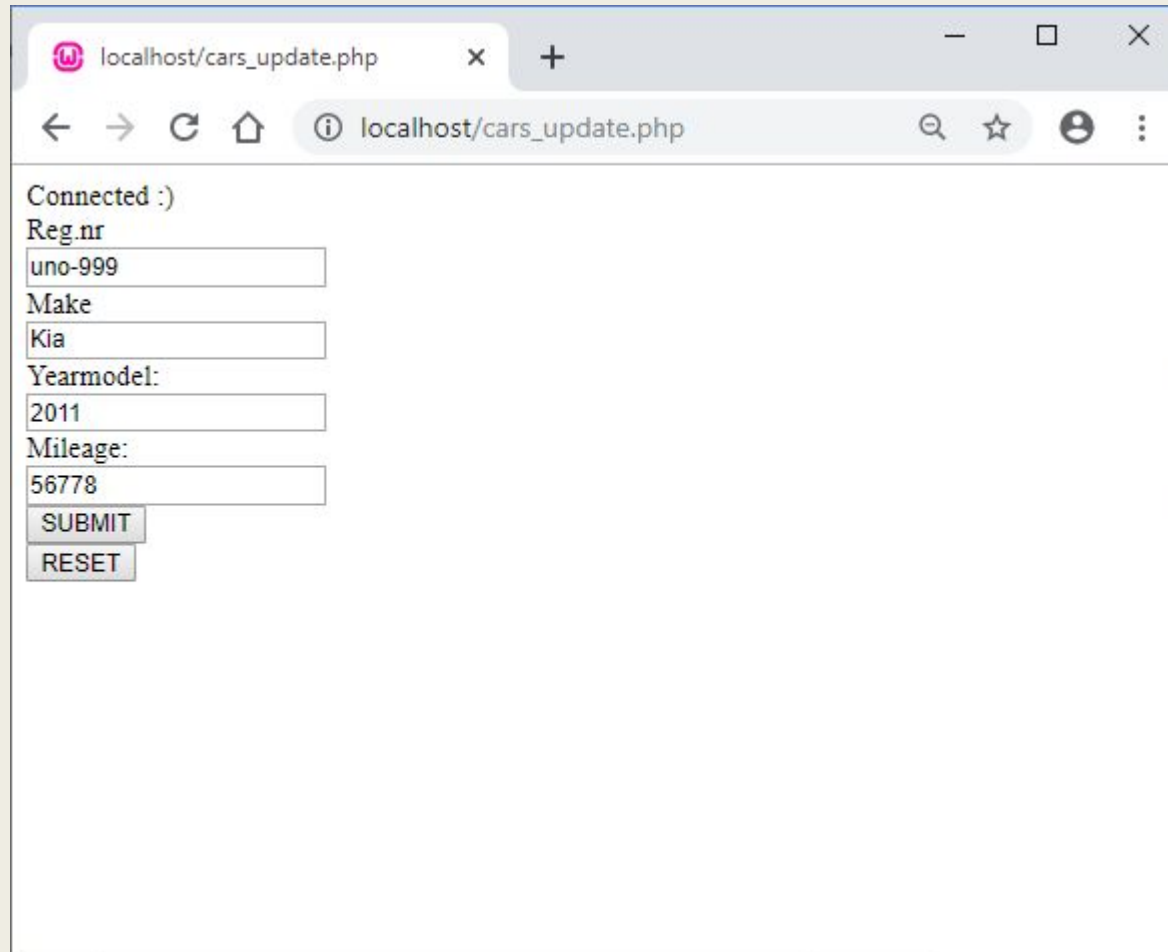
Database operations
updating, code 2)
all data shown

```
<?php
include 'carhouse_connect.php';
$this_filename = $_SERVER['PHP_SELF'];
// car's info is shown, then user can update info

if (isset($_POST['y']))
{ $my = $_POST['y'];
$sql = "SELECT * FROM cars WHERE regnr = '$my'";
$result = mysqli_query($connection, $sql);
?>
<form action="<?php echo $this_filename; ?>" method="post">
<?php
while ($row = mysqli_fetch_array( $result ))
{
?>
<label>Reg.nr<br/>
<input type = "text" name = "1" value = "<?php echo $row[0]; ?>" ><br>
<label>Make<br/>
<input type = "text" name = "2" value = "<?php echo $row[1]; ?>" ><br>
<label>Yearmodel:<br/>
<input type = "text" name = "3" value = "<?php echo $row[2]; ?>" ><br>
<label>Mileage:<br/>
<input type = "text" name = "4" value = "<?php echo $row[3]; ?>" ><br>
<input type="submit" value="SUBMIT" /> <br>
<input type="reset" value="RESET" />
</form>
```

PHP & MySQL

Database operations
updating



A screenshot of a web browser window displaying a form for updating car information. The browser's address bar shows the URL `localhost/cars_update.php`. The form contains the following fields and buttons:

- Connected :)** A status message.
- Reg.nr**: A text input field containing the value `uno-999`.
- Make**: A text input field containing the value `Kia`.
- Yearmodel:**: A text input field containing the value `2011`.
- Mileage:**: A text input field containing the value `56778`.
- SUBMIT**: A button to submit the form.
- RESET**: A button to reset the form.



PHP & MySQL

Database operations
updating, code 3)

```
<?php
if (isset($_POST['1']))
    $m1 = $_POST['1'];
if (isset($_POST['2']))
    $m2 = $_POST['2'];
if (isset($_POST['3']))
    $m3 = $_POST['3'];
if (isset($_POST['4']))
    $m4 = $_POST['4'];

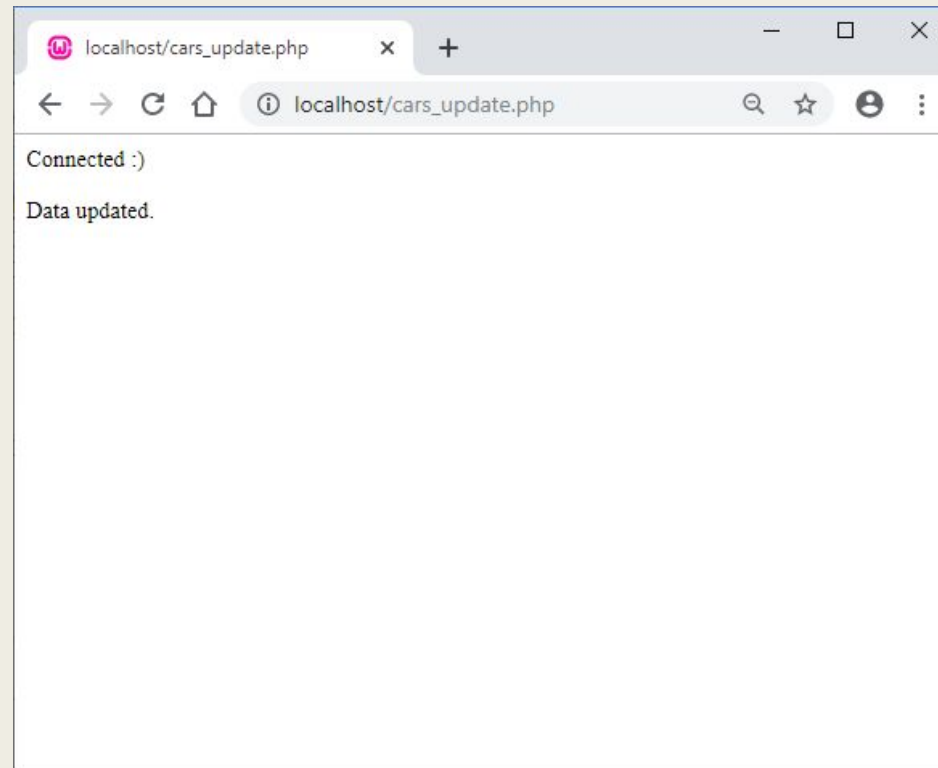
if (isset($_POST['1']))
{
    $sql = "UPDATE cars SET make = '$m2', yearmodel = $m3,
        mileage= $m4 WHERE regnr = '$m1'";

    if (mysqli_query($connection, $sql)) {
        echo '<p>Data updated.</p>';
    }
    else
    {
        echo '<p>Failed ';
        mysqli_error($connection) . '</p>';
    } }
?>
```



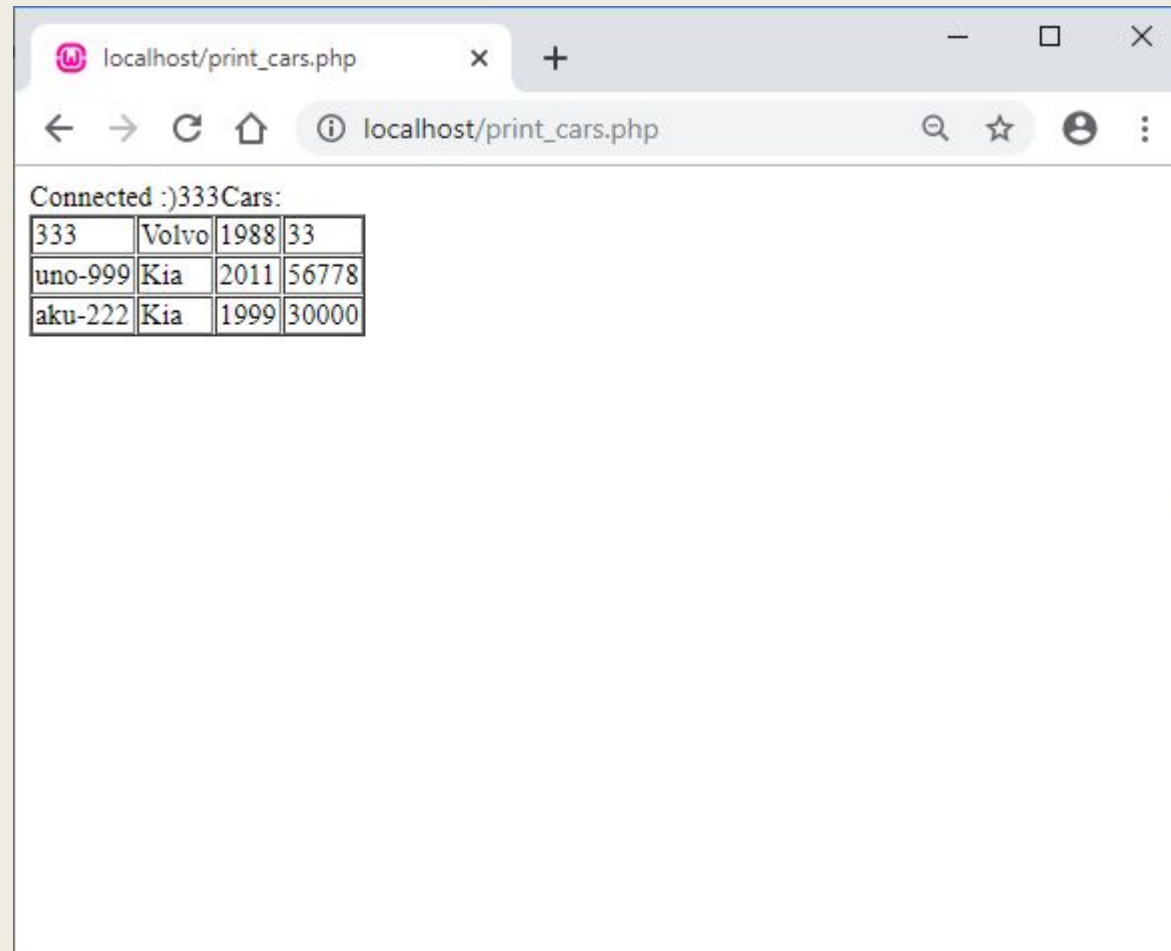
PHP & MySQL

Database operations -
testing update



PHP & MySQL

Database operations -
testing update



The screenshot shows a web browser window with the address bar displaying 'localhost/print_cars.php'. The page content shows the text 'Connected :)333Cars:' followed by a table of car data. The table has four columns and three rows of data.

Connected :)333Cars:			
333	Volvo	1988	33
uno-999	Kia	2011	56778
aku-222	Kia	1999	30000

PHP & MYSQL snaps

About SQL Injection

Example with a table that has only admin data:

Admins table



The image shows a screenshot of a database management tool. On the left, a tree view displays a database named 'carhouse' containing several tables: 'New', 'admins', 'cars', 'darry', and 'user'. The 'admins' table is selected and highlighted. A red line points from the 'admins' table in the tree view to a detailed view of the table on the right. This detailed view shows the table's structure with two columns: 'username' and 'password'. Below the column headers, a single row of data is visible, with 'admin' in the 'username' column and 'jekaterina' in the 'password' column.

username	password
admin	jekaterina

There we have then this info:

```
SELECT * FROM admins where username = 'admin' AND password = 'jekaterina'
```

☐ Show all | Number of rows: 25 ▼ | Filter rows:

+ Options

username	password
admin	jekaterina

Welcome page if admin info was given correctly

```
<?php  
print "<h1>" . "welcome admin!" . "</h1>";  
?>
```

If username and/or password where wrong, another page is shown

```
<?php  
  
print "<h1>failed</h1>";  
  
print "<a href = " . "login2.php" . ">back</a>";  
  
?>
```

Page: login2.php

```
<form method="post">
<h2>ADMIN LOGIN </h2>
User name:<br>
<input type = "text" name = "usr" ><br>
Password: <br>
<input type = "text" name = "pass" ><br>
<input type="submit" value="SUBMIT" /> <br>
<input type="reset" value="RESET" />
</form>
<?php
if (!empty($_POST))
{
if (isset($_POST['usr']))
$n = $_POST['usr'];
if (isset($_POST['pass']))
$s = $_POST['pass'];
$sql = "SELECT * FROM admins where  username = '$n' AND password = '$s' ";
include 'connect_to_db_server.php';

$result = mysqli_query($connection, $sql);
if($result->num_rows)
    header("Location: http://localhost/admin_welcome.php");
else
    header("Location: http://localhost/wrong_trial.php");
}
?>
```

Trial 1:

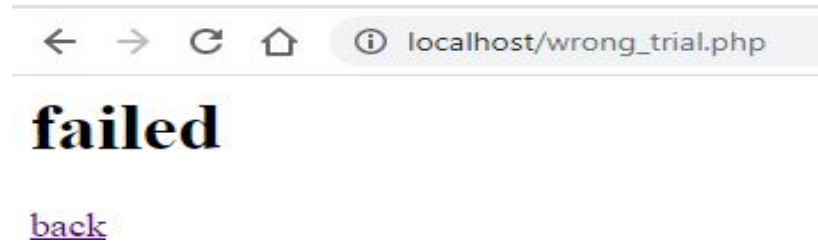
← → ↻ 🏠 ⓘ localhost/login2.php

ADMIN LOGIN

User name:

Password:

Failed



Trial 2:

A screenshot of a web browser window. The address bar shows 'localhost/login2.php'. The main content area displays the title 'ADMIN LOGIN' in a large, bold, black serif font. Below the title, there is a form with two input fields. The first field is labeled 'User name:' and contains the text 'admin'. The second field is labeled 'Password:' and contains the text 'trialpassword' OR '1=1'. Below the password field, there are two buttons: 'SUBMIT' and 'RESET'.

You can see now that Password field contains SQL command OR and condition $1 = 1$ is true.

Successful!



welcome admin!

SO, it was allowed to use SQL commands **without escaping**.

Escaping strings

NOW we add function `mysqli_real_escape_string` with.

Example here (<https://www.php.net/manual/en/mysqli.real-escape-string.php>)

E.g. `$city = mysqli_real_escape_string($link, $city);`

Code

```
]<?php
include 'connect_to_db_server.php';
if (!empty($_POST))
[
    if (isset($_POST['usr']))
        $n = $_POST['usr'];
    if (isset($_POST['pass']))
        $s = $_POST['pass'];
    $n = mysqli_real_escape_string($connection, $n);
    $s = mysqli_real_escape_string($connection, $s);
    $sql ="SELECT * FROM admins where  username = '$n' AND password = '$s' ";

    $result = mysqli_query($connection, $sql);

    if($result->num_rows)
        header("Location: http://localhost/admin\_welcome.php");
    else
        header("Location: http://localhost/wrong\_trial.php");
}
```

And now we can not use commands in SQL (there are many other SQL injection types, too...)

ADMIN LOGIN

User name:

Password:

← → ↻ 🏠 ⓘ localhost/wrong_trial.php

failed

[back](#)

SQL injection example 2: several SQL statements

Here is a table (name is darry) that is to be removed by the user that logs in:



The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, with the 'carhouse' database selected. The 'darry' table is highlighted. The main panel shows the 'SQL' tab with the query `SELECT * FROM `darry``. Below the query, the results are displayed as a table with two rows: John (id 1) and Don (id 2).

id	name
1	John
2	Don

So, user types another sql statement after username: if escaping is not working, s even that another sql command is exexuted. Now we remove a table (of course some user that wants to cause harm may have to use wildcards with tablenamees OR try to print first all tablenamees of the database....)

Here is php code:

```
<html>

<body>
<form method="post">
<h2>LOG IN </h2>
User name:<br>
<input type = "text" name = "usr" ><br>

Password: <br>
<input type = "text" name = "pass" ><br>
<input type="submit" value="SUBMIT" /> <br>
<input type="reset" value="RESET" />
</form>
<?php
if (!empty($_POST))
{
if (isset($_POST['usr']))
$n = $_POST['usr'];
if (isset($_POST['pass']))
$s = $_POST['pass'];
$sql ="SELECT * FROM user where  userid = '$n' AND password = '$s' ";
include 'connect_to_db_server.php';
print $sql;
$result = mysqli_query($connection, $sql);
```


Let's try:

First normal logging:

LOG IN

User name:

Password:

It was ok, and sql is printed for testing purposes

LOG IN

User name:

Password:

SUBMIT

RESET

```
SELECT * FROM user where userid = 'aa' AND password = 'aa'
```

Extra SQL statement added after password

Now user adds there a new sql statement after username (drop table darry):

LOG IN

User name:

Password:

Situation before



Now SQL statement "drop table darry" is added there:

LOG IN

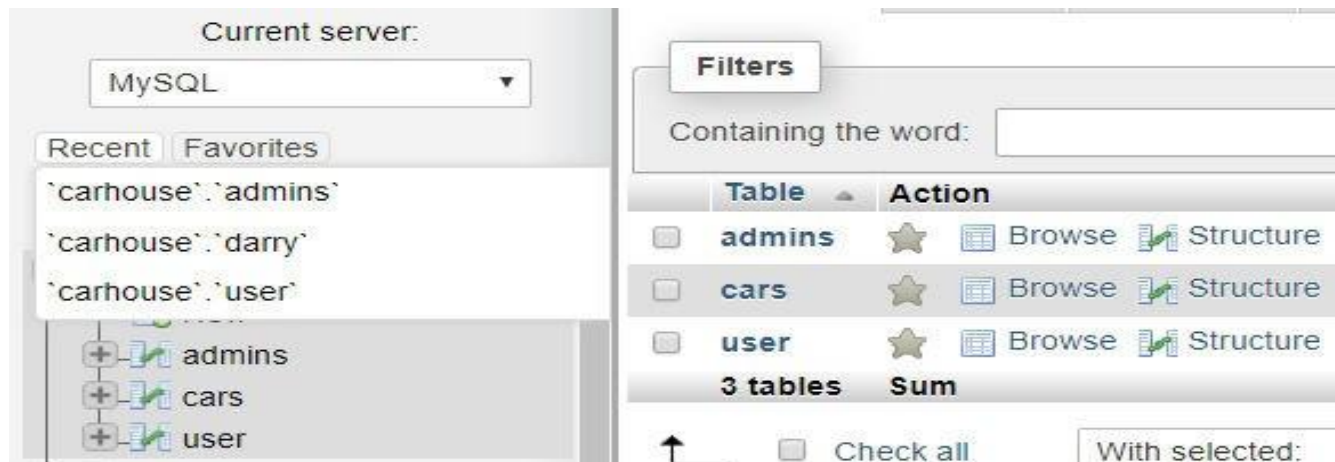
User name:

Password:

`SELECT * FROM user where userid = 'aa' AND password = 'aa';drop table darry#'`

Result

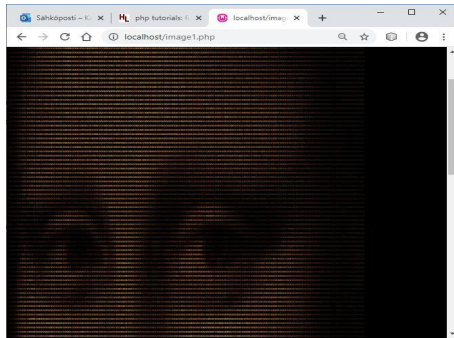
Table darry is gone!



Try it! OR may be I have to say avoid it!

OR make sure it cannot be happened!

As a prize for your watching I show a pic example (**code from php.net**): Image to html



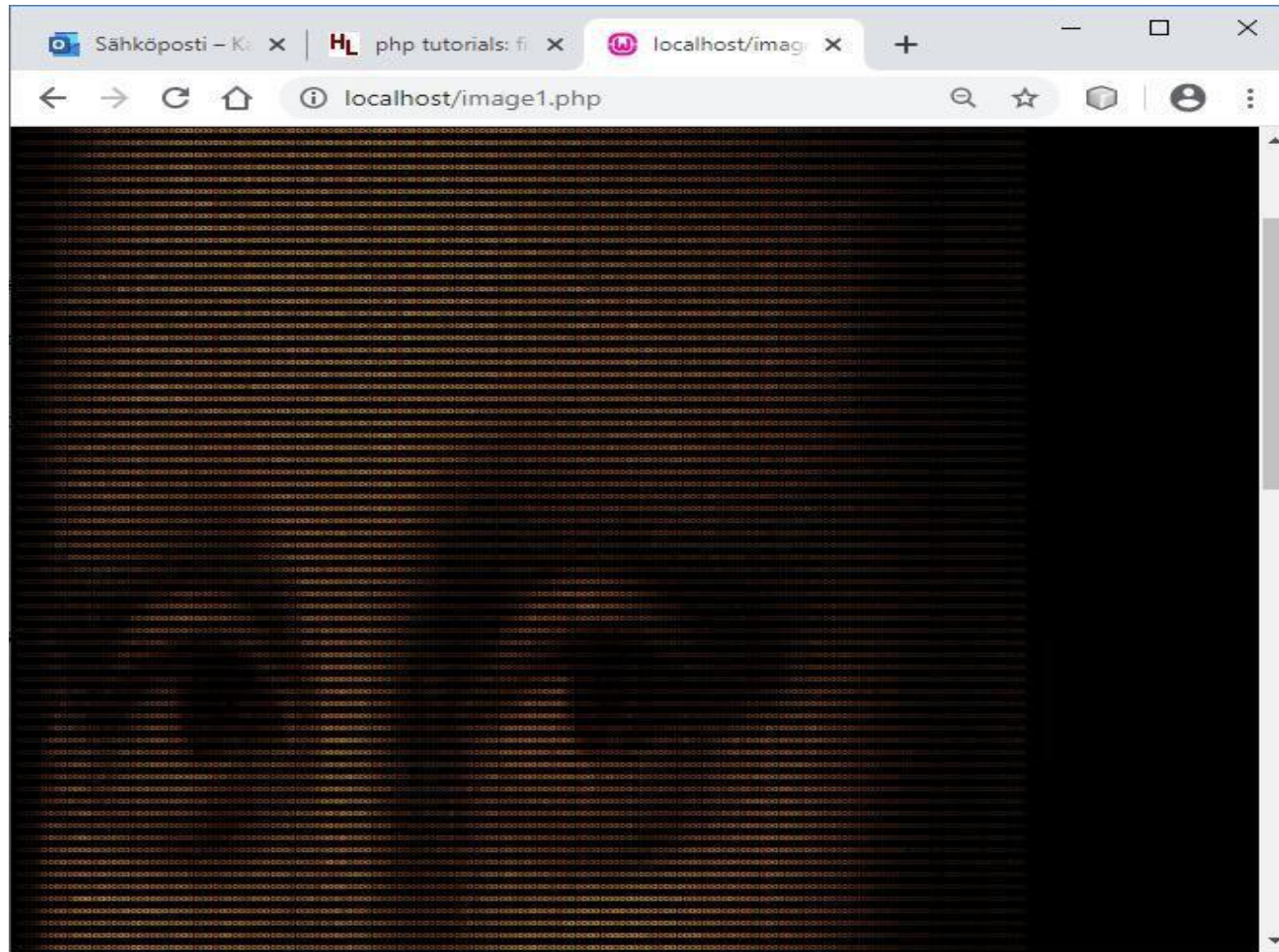
```
<?php

$im = imagecreatefromjpeg("monalisa2.jpg");
$rgb = ImageColorAt($im, 169, 197);

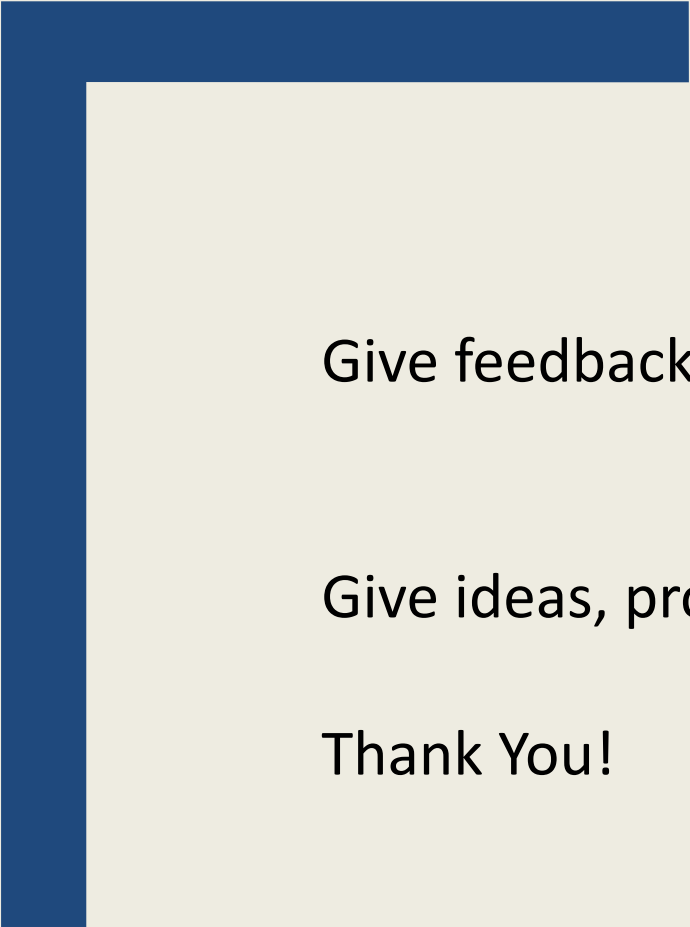
$res = 1;
$old=0;

print "<html>";
print "<head>";
    print "</head>";
    print "<body bgcolor='000000' >";
    print "<pre>" ;
    for ($Y = 10; $Y < 197; $Y = $Y + $res)
    {
        for ($X = 10; $X < 169; $X = $X + $res)
        {
            $rgb = ImageColorAt($im, $X, $Y);
            if ($rgb != $old)
            {
                printf ("</font><font color='#%06x'>", $rgb);
                $old = $rgb;
            }
            print "o";
        }
        print "<br>";
    }
    print "</font>";
    print "</pre>";
    print "</body>";
    print "</html>";

?>
```

Thank You!



Give feedback, please!

Give ideas, proposals for new guides!

Thank You!



Thank You!