PHP Form Handling

Introduction to PHP Global Variables - Superglobals

Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

Thus, Super global variables are built-in variables that are always available in all scopes.

The PHP superglobal variables are:

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_FILES
- $_ENV
- $_COOKIE
- $_SESSION

PHP $GLOBALS

- **$GLOBALS** is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).
- PHP stores all global variables in an array called $GLOBALS[*index*]. The ***index*** holds the name of the variable.
- The example below shows how to use the super global variable $GLOBALS:

```
<!DOCTYPE html>
<html>
<body>
<?php
$x = 75;
$y = 25;
function addition(){
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
?>
</body>
</html>
```

PHP $_SERVER

$_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

The example below shows how to use some of the elements in $_SERVER:

```
<!DOCTYPE html>
<html>
<body>
<?php
echo $_SERVER['PHP_SELF'];    //Returns the filename of the currently executing script
echo "<br>";
echo $_SERVER['SERVER_NAME']; //Returns the name of the host server
echo "<br>";
echo $_SERVER['HTTP_HOST']; //Returns the Host header from the current request.
echo "<br>";
echo $_SERVER['HTTP_REFERER']; //Returns the complete URL of the current page
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME']; //Returns the path of the current script
?>
</body></html>
```

Note:**HTTP_HOST:** It is fetched from HTTP request header obtained from the client request

PHP $_REQUEST

- PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.
- The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag.
- In this example, we point to this file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice.
- Then, we can use the super global variable $_REQUEST to collect the value of the input field:

```
<!DOCTYPE html>
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_POST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

PHP $_POST

- PHP $_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.
- The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag.
- In this example, we point to the file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice.
- Then, we can use the super global variable $_POST to collect the value of the input field:

```
<!DOCTYPE html>
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_POST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

PHP $_GET

- PHP $_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".
- $_GET can also collect data sent in the URL.
- Assume we have an HTML page that contains a hyperlink with parameters:

```
<html>
<body>

<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>

</body>
</html>
```

- When a user clicks on the link "Test $GET", the parameters "subject" and "web" are sent to "test_get.php", and you can then access their values in "test_get.php" with $_GET.

The example below shows the code in "test_get.php":

```
<!DOCTYPE html>
<html>
<body>

<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>

</body>
</html>
```

Form.php

```
<form action="display.php" method="post" id="nameform">
First Name: <input type="text" name="fname"></br>
Last Name: <input type="text" name="lname"></br>
<button type="submit" form="nameform" value="submit">Submit</button>
<button type="reset" value="Reset">Reset</button>
</form>
</form>
```

Note: When the user fills out the form above and clicks the submit button, the form data is sent for processing to a PHP file named “display.php". The form data is sent with the HTTP POST method.

Display.php(using POST method)

```
<?php

$first_name = $_POST['fname'];
$last_name = $_POST['lname'];

echo $first_name;
echo $last_name;

?>
```

OR

Display1.php(using GET method)

```
<?php

$first_name = $_GET['fname'];
$last_name = $_GET['lname'];

echo $first_name;
echo $last_name;

?>
```

GET vs. POST

- Both GET and POST create an array (e.g. array(key1 => value1, key2 => value2, key3 => value3, ...))
- Keys are the names of the form controls and values are the input data from the user.
- Both GET and POST are treated as $_GET and $_POST.
- These are superglobals, which means that they are always accessible, regardless of scope

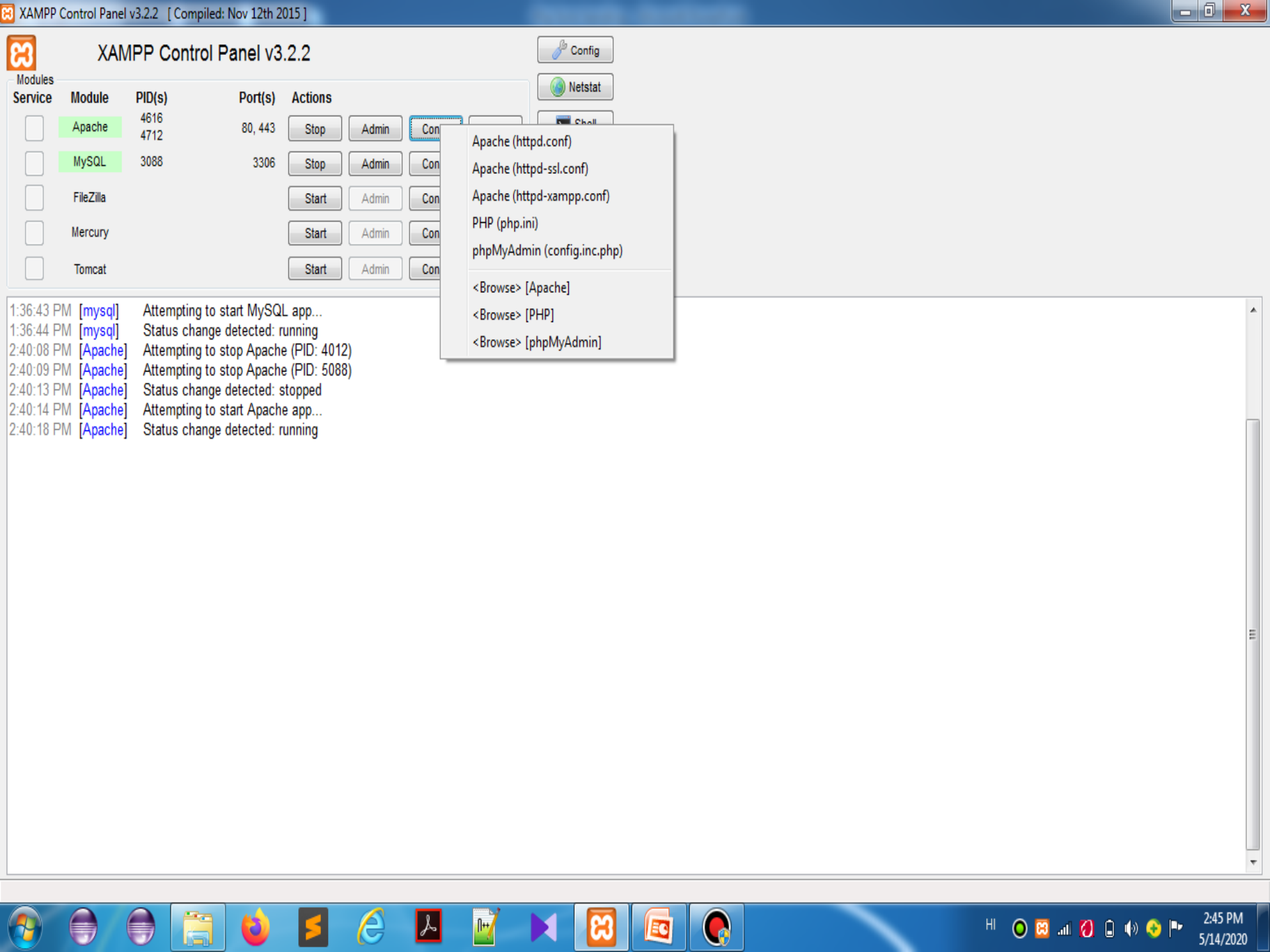
When to use GET?

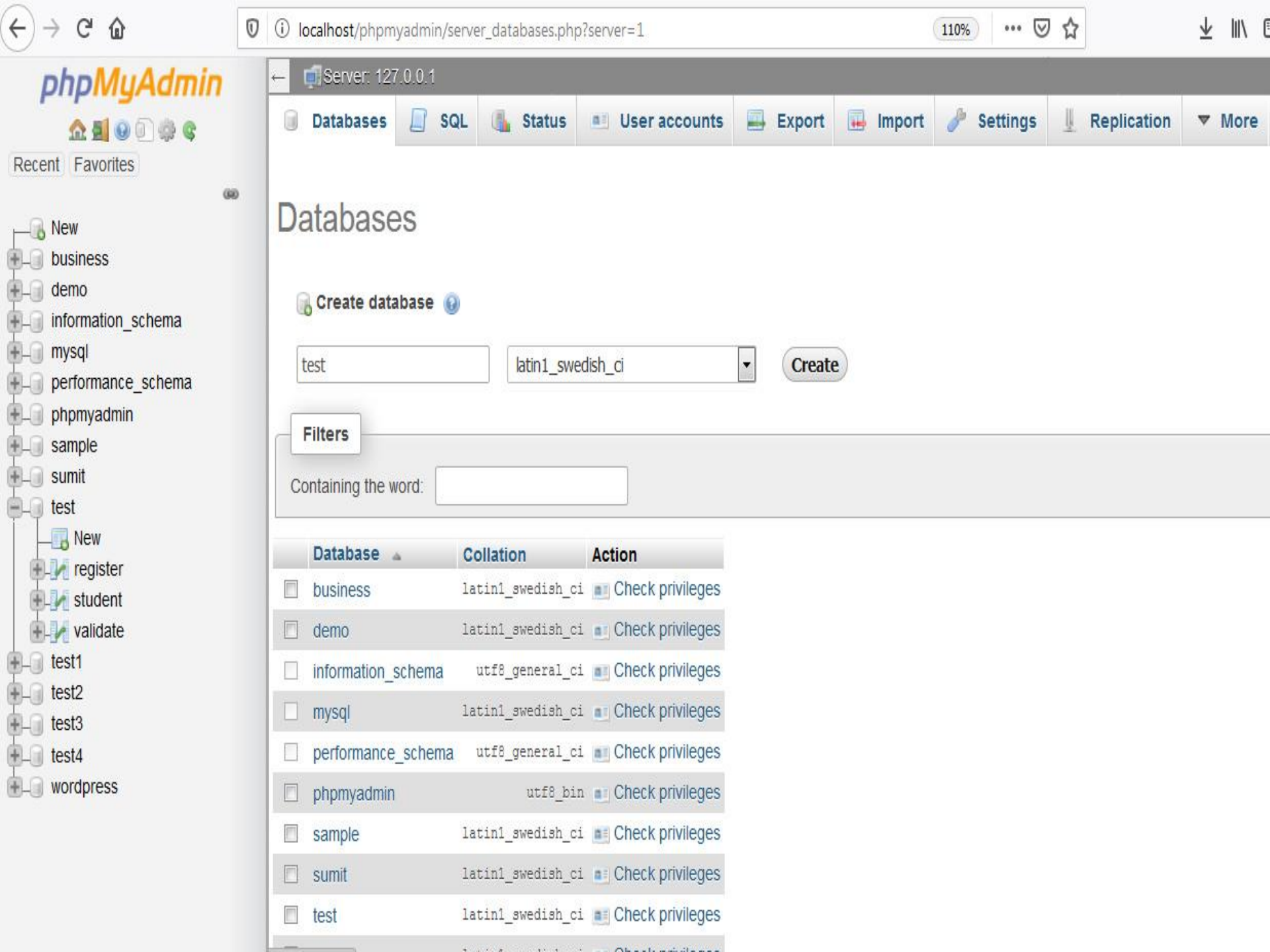
- Information sent from a form with the GET method is **visible to everyone** (all variable names and values are displayed in the URL).
- GET also has limits on the amount of information to send.
- The limitation is about 2000 characters.
- However, because the variables are displayed in the URL, it is possible to bookmark the page.
- GET may be used for sending non-sensitive data.

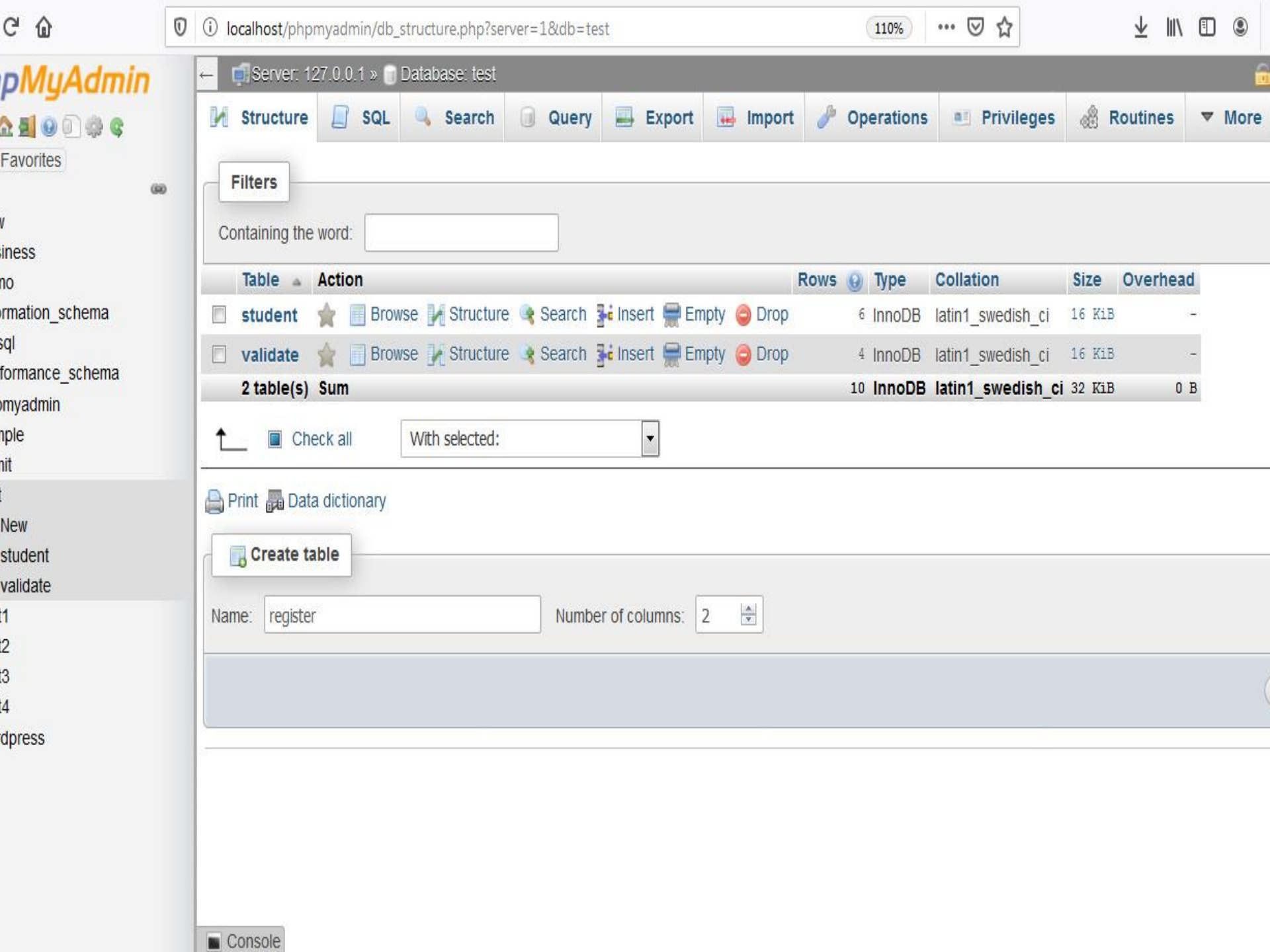
When to use POST?

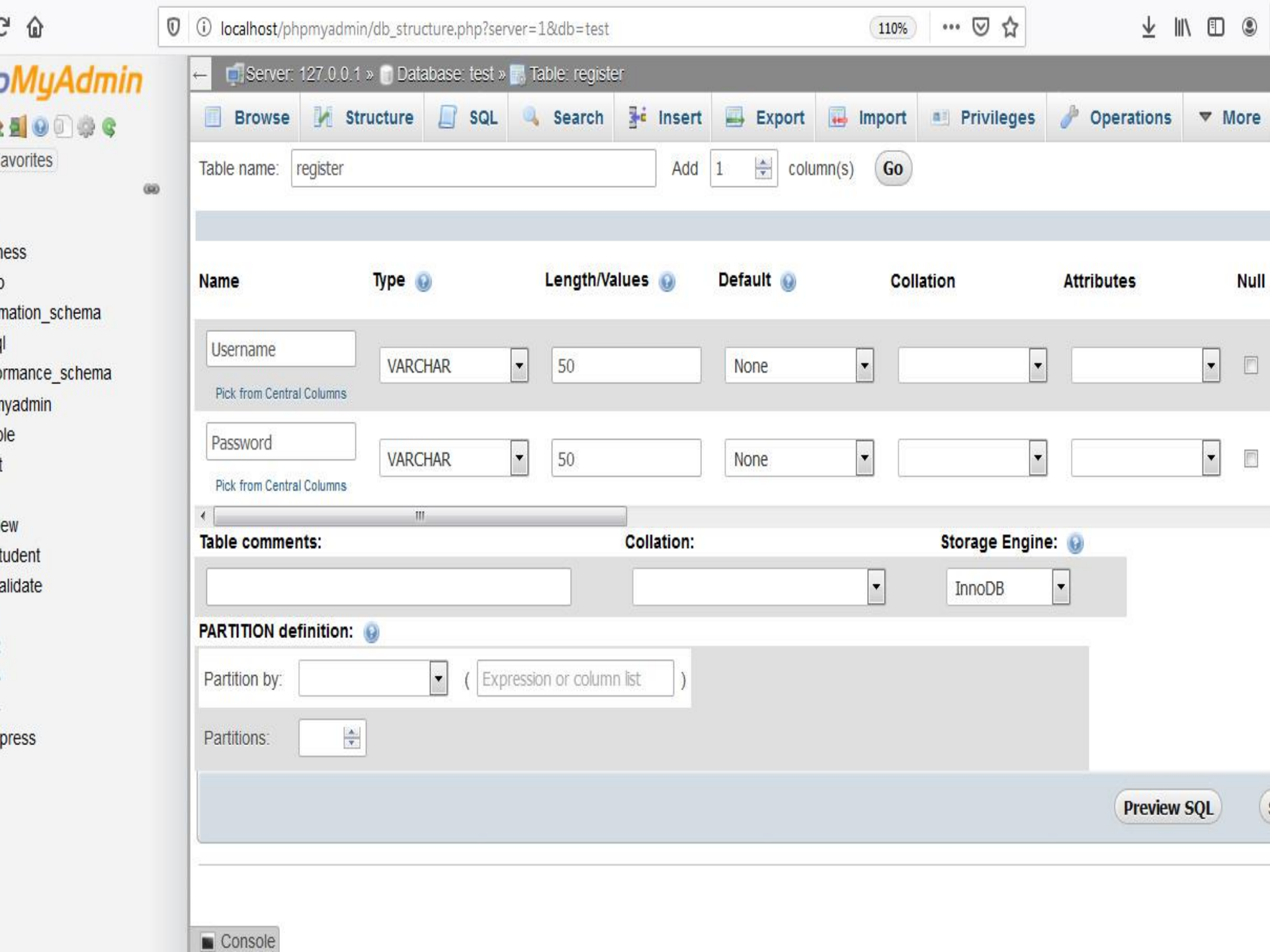
- Information sent from a form with the POST method is **invisible to others** (all names/values are embedded within the body of the HTTP request)
- **No limits** on the amount of information to send.
- Because the variables are not displayed in the URL, it is not possible to bookmark the page.

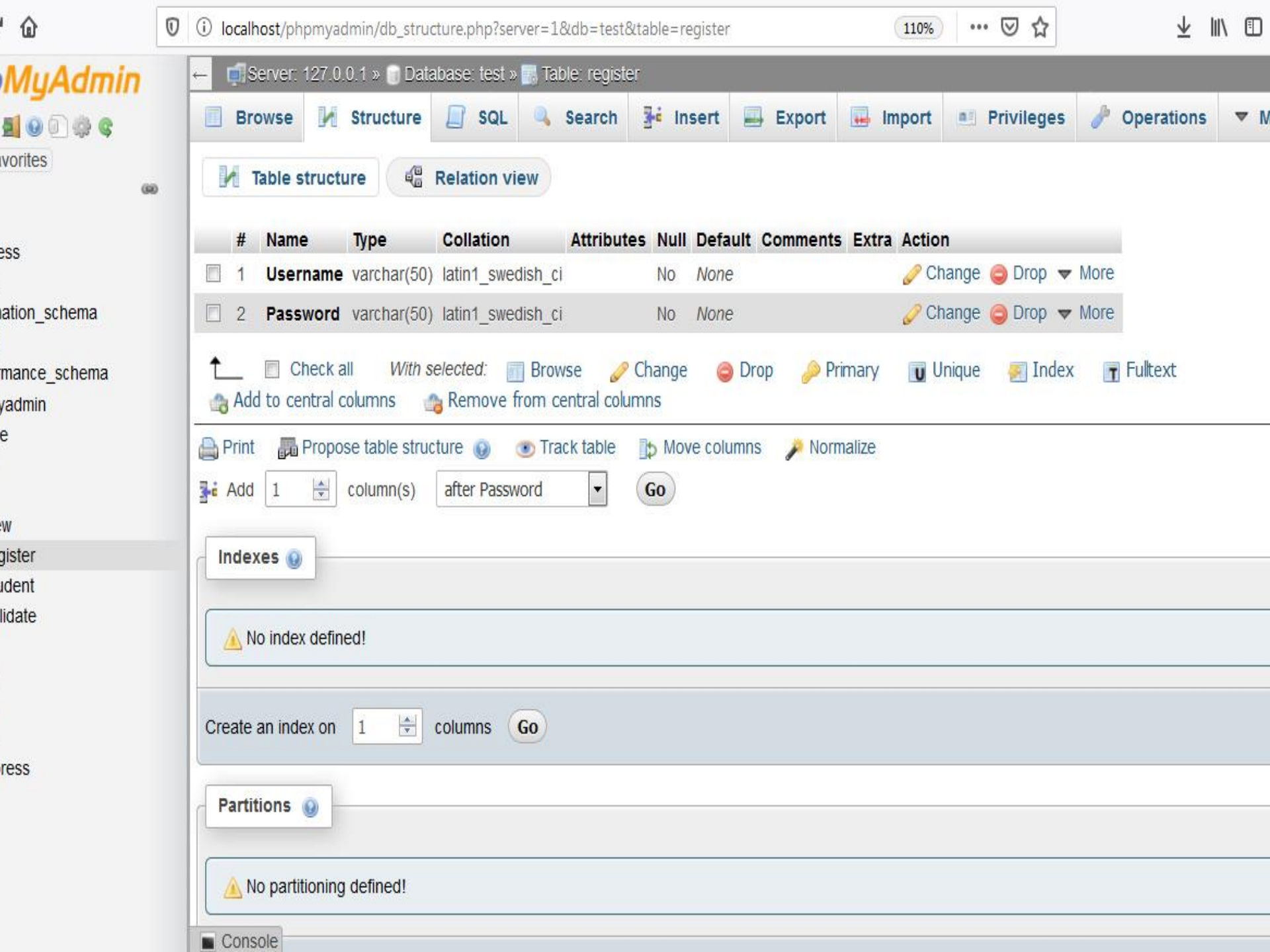
Database Connectivity


XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Config
Netstat
Modules
Service
Module
PID(s)
Port(s)
Actions
Apache
4616
4712
80, 443
Stop
Admin
MySQL
3088
3306
Stop
Admin
FileZilla
Start
Admin
Mercury
Start
Admin
Tomcat
Start
Admin
Apache (httpd.conf)
Apache (httpd-ssl.conf)
Apache (httpd-xampp.conf)
PHP (php.ini)
phpMyAdmin (config.inc.php)
<Browse> [Apache]
<Browse> [PHP]
<Browse> [phpMyAdmin]
1:36:43 PM [mysql] Attempting to start MySQL app...
1:36:44 PM [mysql] Status change detected: running
2:40:08 PM [Apache] Attempting to stop Apache (PID: 4012)
2:40:09 PM [Apache] Attempting to stop Apache (PID: 5088)
2:40:13 PM [Apache] Status change detected: stopped
2:40:14 PM [Apache] Attempting to start Apache app...
2:40:18 PM [Apache] Status change detected: running
HI
2:45 PM
5/14/2020


localhost/phpmyadmin/server_databases.php?server=1
110%
phpMyAdmin
Recent
Favorites
New
business
demo
information_schema
mysql
performance_schema
phpmyadmin
sample
sumit
test
New
register
student
validate
test1
test2
test3
test4
wordpress
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
More
Databases
Create database
test
latin1_swedish_ci
Create
Filters
Containing the word:
Database
Collation
Action
business latin1_swedish_ci Check privileges
demo latin1_swedish_ci Check privileges
information_schema utf8_general_ci Check privileges
mysql latin1_swedish_ci Check privileges
performance_schema utf8_general_ci Check privileges
phpmyadmin utf8_bin Check privileges
sample latin1_swedish_ci Check privileges
sumit latin1_swedish_ci Check privileges
test latin1_swedish_ci Check privileges


localhost/phpmyadmin/db_structure.php?server=1&db=test
110%
pMyAdmin
Favorites
Server: 127.0.0.1 » Database: test
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Filters
Containing the word:
Table
Action
Rows
Type
Collation
Size
Overhead
student
Browse
Structure
Search
Insert
Empty
Drop
6
InnoDB
latin1_swedish_ci
16 KiB
-
validate
Browse
Structure
Search
Insert
Empty
Drop
4
InnoDB
latin1_swedish_ci
16 KiB
-
2 table(s)
Sum
10
InnoDB
latin1_swedish_ci
32 KiB
0 B
Check all
With selected:
Print
Data dictionary
Create table
Name:
register
Number of columns:
2
New
student
validate
Console

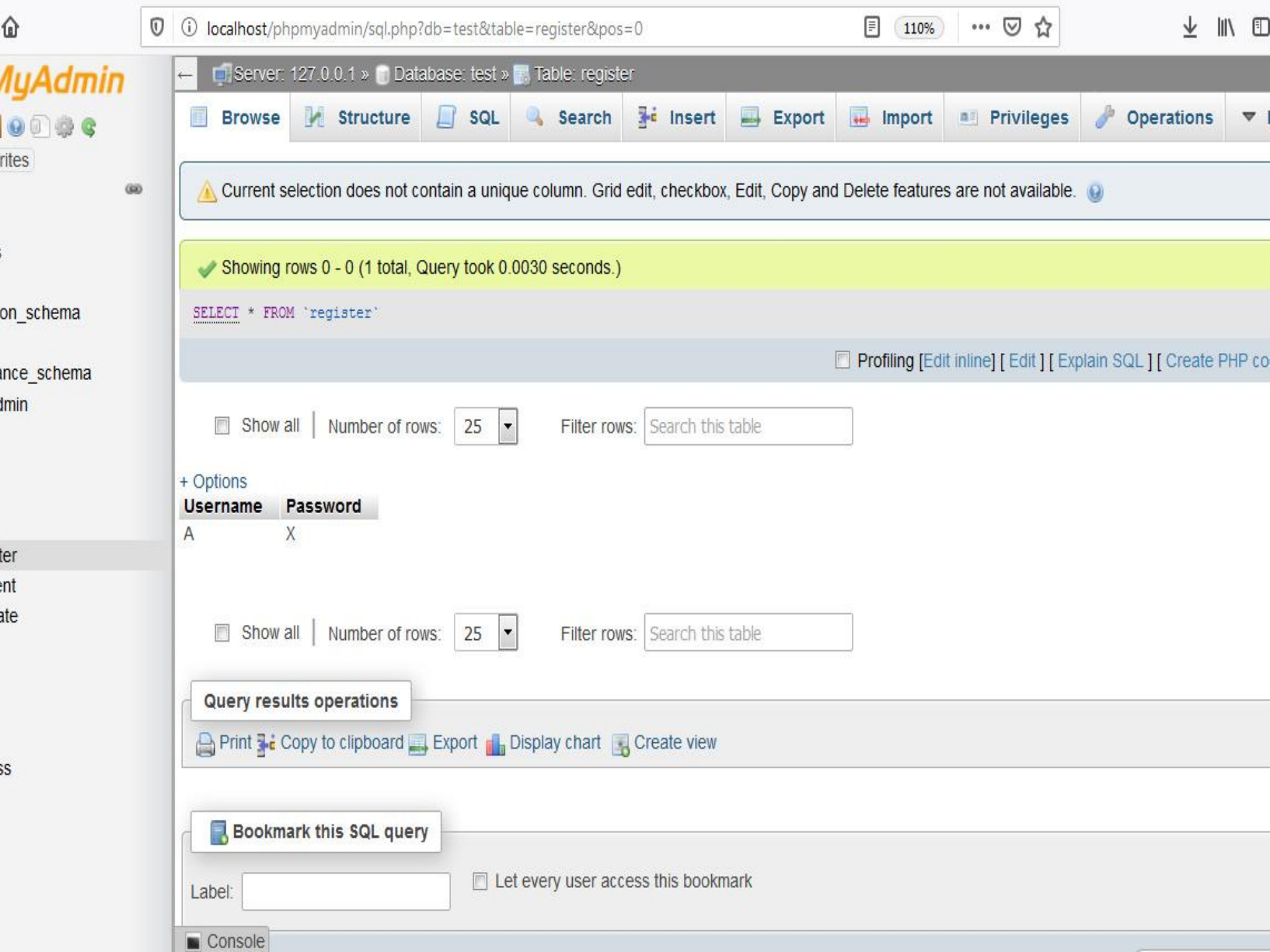

localhost/phpmyadmin/db_structure.php?server=1&db=test
110%
MyAdmin
avorites
ness
mation_schema
rmance_schema
nyadmin
ew
tudent
alidate
press
Server: 127.0.0.1 » Database: test » Table: register
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
More
Table name: register
Add 1 column(s)
Go
Name
Type
Length/Values
Default
Collation
Attributes
Null
Username
VARCHAR
50
None
Pick from Central Columns
Password
VARCHAR
50
None
Pick from Central Columns
Table comments:
Collation:
Storage Engine:
InnoDB
PARTITION definition:
Partition by:
Expression or column list
Partitions:
Preview SQL
Console


localhost/phpmyadmin/db_structure.php?server=1&db=test&table=register
110%
MyAdmin
Server: 127.0.0.1 » Database: test » Table: register
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Table structure
Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 Username varchar(50) latin1_swedish_ci No None Change Drop More
2 Password varchar(50) latin1_swedish_ci No None Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Fulltext
Add to central columns Remove from central columns
Print Propose table structure Track table Move columns Normalize
Add 1 column(s) after Password Go
Indexes
No index defined!
Create an index on 1 columns Go
Partitions
No partitioning defined!
Console

Insert Data into Database

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test"; // Go to localhost/phpmyadmin and then create the database test
    there
// Creating connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO register (Username, Password) VALUES ('A', 'X')";

if ($conn->query($sql) === TRUE) {
   echo "New record created successfully";
} else {
   echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```


localhost/phpmyadmin/sql.php?db=test&table=register&pos=0
110%
MyAdmin
Server: 127.0.0.1 » Database: test » Table: register
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.
Showing rows 0 - 0 (1 total, Query took 0.0030 seconds.)
SELECT * FROM `register`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP co
Show all | Number of rows: 25 Filter rows: Search this table
+ Options
Username Password
A X
Show all | Number of rows: 25 Filter rows: Search this table
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Console

Fetch data from the database

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT Username, Password from register";
$result = $conn->query($sql);
```

Contd..

```
if ($result->num_rows > 0) {
   // output data of each row
   while($row = $result->fetch_assoc()) {
      echo "Username: " . $row["Username"]. " - Password: " .
     $row["Password"]."<br>";
   }
} else {
   echo "0 results";
}
$conn->close();
?>
```

Note: $result = $conn->query($sql);

This line of code runs the query and puts the resulting data into a variable called $result.

The function **num_rows()** checks if there are more than zero rows returned.

$row = $result->fetch_assoc()

The function fetch_assoc() puts all the results into an associative array that we can loop through.

ssword: X

Inserting values into database using forms

Form1.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
   <form method="post" action= "insert.php" >
      <table border="1" >
         <tr>
            <td><label for="users_email">Email</label></td>
            <td><input type="text"

              name="users_email" id="users_email"></td>
         </tr>
```

Contd..

Contd...

Contd..

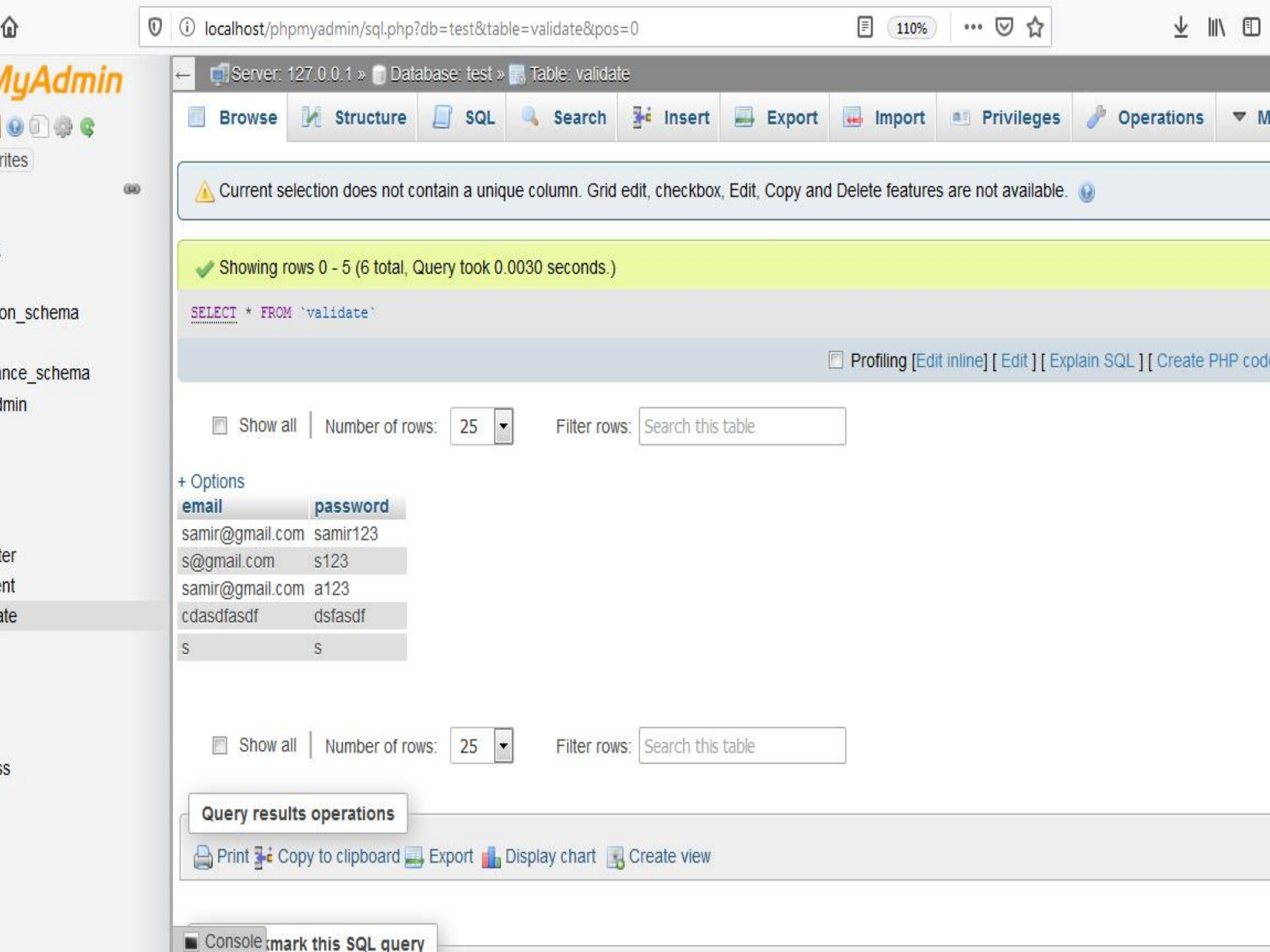
```
        <tr>
            <td><label for="users_pass">Password</label></td>
            <td><input name="users_pass"

              type="password" id="users_pass"></input></td>
        </tr>
        <tr>
            <td><input type="submit" value="Submit"/>
            <td><input type="reset" value="Reset"/>
        </tr>
    </table>
  </form>
</body>
</html>
```

insert.php

```
<?php
// Grab User submitted information
$email = $_POST["users_email"];
$pass = $_POST["users_pass"];

// Connect to the database
$conn = new mysqli("localhost","root","","test");
// Make sure we connected successfully
if($conn->connect_error)
{
   die('Connection Failed'.$conn->connect_error);
}
$sql= "insert into validate(email, password) Values('$email','$pass')";
$result = $conn->query($sql);
if($result)
     echo"Record added";
else echo"Record Could not be added";
```


localhost/phpmyadmin/sql.php?db=test&table=validate&pos=0
110%
MyAdmin
Server: 127.0.0.1 » Database: test » Table: validate
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.
Showing rows 0 - 5 (6 total, Query took 0.0030 seconds.)
SELECT * FROM `validate`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP cod
Show all
Number of rows: 25
Filter rows: Search this table
+ Options
email password
samir@gmail.com samir123
s@gmail.com s123
samir@gmail.com a123
cdasdfasdf dsfasdf
s s
Show all
Number of rows: 25
Filter rows: Search this table
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Console

Checking for login after fetching values from the database

Form1.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
   <form method="post" action= "login.php" >
      <table border="1" >
         <tr>
            <td><label for="users_email">Email</label></td>
            <td><input type="text"

              name="users_email" id="users_email"></td>
         </tr>
```

Contd..

Contd...

Contd..

```
        <tr>
            <td><label for="users_pass">Password</label></td>
            <td><input name="users_pass"
              type="password" id="users_pass"></input></td>
        </tr>
        <tr>
            <td><input type="submit" value="Submit"/>
            <td><input type="reset" value="Reset"/>
        </tr>
    </table>
  </form>
</body>
</html>
```

login.php

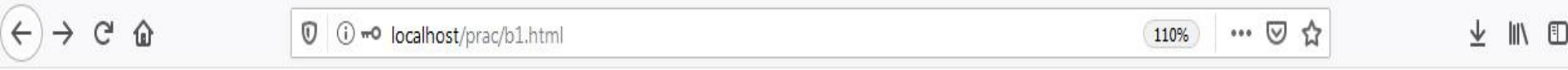
```
<?php
// Grab User submitted information
$email = $_POST["users_email"];
$pass = $_POST["users_pass"];

// Connect to the database
$conn = new mysqli("localhost","root","","test");
// Make sure we connected successfully
if($conn->connect_error)
{
   die('Connection Failed'.$conn->connect_error);
}
```

Contd..

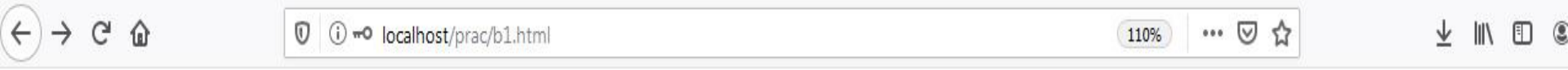
Contd..

```
$sql="select email, password from validate where email='$email'";
$result = $conn->query($sql);
if ($result-> num_rows>0)
{
    while($row=$result->fetch_assoc())
    {
        if($row['email']==$email && $row['password']==$pass)
            echo"You are a validated user";
        else
            echo"Wrong password";
    }
}
else echo "User does not exist";
?>
```



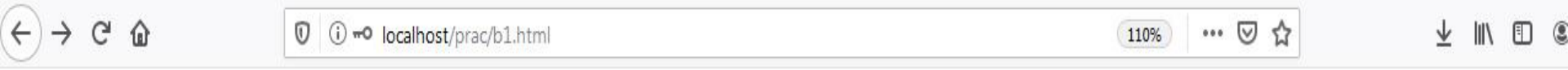
Email	s@gmail.com
Password	••
Submit	Reset

Wrong password



Email	s1@gmail.com
Password	••••
Submit	Reset

User does not exist



Email	s@gmail.com
Password	••••
Submit	Reset

You are a validated user