Guidelines for implementation

laboratory work No. 6 on the course " Web technologies"

**Server scripts. Working with the database.**

**The purpose of the work** : to study the three-tier architecture of web applications (Client - Web server - Database server) and the language for writing server scripts php for building dynamic content of web sites.

**Rules for assessing works**

Completion of the basic task is estimated at a maximum of 6 points. To receive 10 points for the work, it is necessary to complete at least five points from the "Extended Task" section.

Points will be deducted from the maximum possible grade in the following situations:

1. **Deadline.** The report is posted in the personal account after the deadline - minus 1 point every 2 weeks from the date recorded in the personal account. The date of the defense does not affect;
2. **Defense.** "Gaps" in knowledge during the oral defense of the work: grade "good" - minus 1 point, grade "satisfactory" - minus 2 points;
3. **Report.** The report contains minor violations of requirements - minus 1 point, several report items are missing - minus 2 points or more;
4. **Work.** The completed task lacks minor elements - minus 1 point, several points of the task are missing - minus 2 points or more.

Protection of work is possible only if there is a report in the personal account.

**Basic task**

Apache+PHP+MySQL (WAMP – XAMPP , Denver, etc.) is allowed to be used as 3-tier architecture links . Server script language - PHP.

Prepare a relational database consisting of 2 tables linked by a one-to-many relationship. Each table must be in 3 normal form and contain at least 5 fields. The contents of the table fields must correspond to the site theme. It is prohibited to make tables "book", "author", "article" and their analogues. All data from the tables must be displayed on the page included in the site being developed. To generate a web page, use the echo command .

**Extended Task**

1.. do not display service fields (primary and secondary keys).

2.. display all fields in one table on the web page

3.. add the ability to enter a new row into the database table on the web page

4.. add the ability to edit a row in a database table on a web page

5.. add the ability to delete a row from a database table on a web page

6.. write a server script in PHP that performs the task with matrices from work #3.

7.. apply the design via css

printf ( ) function several times to generate web page

**Contents of the report**

1. front page;
2. purpose of work;
3. assignment option;
4. structure of database tables;
5. proof of finding tables in 3rd normal form
6. data in tables
7. HTML code
8. code of server scripts used on the site;
9. screenshots of site pages

**Technologies PHP, Apache , MySQL .**

PHP is a server-side scripting language. It allows you to implement various functions for the web, such as: building/searching/saving site data in database tables, dynamically building HTML pages, automatic mailing, etc. PHP supports working with many DBMS, including MySQL .

Example php script:

<html> <head>

<title>Example</title > < /head>

<body>

<?php echo "Hi, I'm a PHP script!" ; ? >

</body> </html>​​​​

php -script is embedded into html -page using special tags

*<? php* - opening tag

*?>* - closing tag

In this case, the file extension changes **from *.html* on *.php*** .***​***

The script supplements the text of the web page in which it is placed. These supplements are formatted as output to the screen via the echo command :

echo " You added flower : <strong> ".$ \_POST['flower']."</strong>< br />";

Strings are enclosed in double or single quotes. The type of quotes determines how the string is displayed. The string may contain html tags and variable names ($name). If the string is enclosed in **double quotes** , then the variable name **is replaced by its value** .

The string gluing operation is indicated by a dot. It is possible to glue not only strings, but also data of other types.

Also, to generate a web page, the printf ( ) function is used, similar to the function of the same name in the C language:

printf ( "%s (%s)\n" , $row [ 'Name' ], $row [ 'Population' ]);

PHP is executed on the web server side. That is, to execute the *add\_flower.php script it is not enough to enter* <file://C:/mysite/www/add_flower.php>in the browser

You need to use a request to the web server: <http://mysite.ru/add_flower.php>

Each variable in PHP is preceded by the dollar symbol $. The variable is declared using the assignment operator. Dynamic typing is used (the data type of the variable is determined by the type of the assigned value).

Example: $ my\_name = “ John ”;

**Working with HTML Forms Using PHP**

POST, GET requests send data from the client ( html page) to the server (PHP script). GET – usually used to search for data (without changing the data on the server, has a limited number of parameters)

POST – used to modify data (has an unlimited number of parameters)

To transfer data from an HTML page to a PHP script, use:

1 Tag *form* and his attributes *action* , *method*

*<FORM ACTION=" email.php " METHOD="POST">*

The tag specifies the screen form in which the data to be transferred will be specified. *Action* – PHP script to which data from the HTML page will be transferred. *Method* – request type (GET or POST)

2 Tags *input , select , textarea* and their *name attributes*

*<INPUT TYPE="text" NAME=" flowerColor " value="blue">*

Specify the data to be transmitted. In this case, the parameter “ *flowerColor* ” will be transmitted, the value of the parameter is: “ *blue* ”

3. *input tag* type *submit* :

*<INPUT TYPE=" submit " VALUE="Submit request!">*

Used to send form data to a PHP script.

**Example HTML page *example.html***

Contains a screen form that passes to the *add\_flower.php script POST* method two parameters – *flower* and *color* :

<html>

<head>

<title>Example</title>

</head>

<body>

<form action="add\_flower.php" method="post">

<input type="text" name="flower" value="роза">

<input type="text" name="color" value="красный">

<input type="submit" value="Отправить!">

</form>

</body>

</html>

In the php script, the data transferred from the html forms are stored in the associative arrays *$\_GET* and *$\_POST* respectively. To access the element, the name of the transferred parameter is used: $\_POST[' color '].

Access to the transferred values is by the name of *the name attribute* tags *input , select , textarea*

When debugging a program, you can view the contents of these arrays using the *<? php function. phpinfo ( 32); ?>*

*add\_flower.php* file

< html >

< body >

<p>

<? php

/\* This script uses values from the $\_POST array \*/

echo " You added flower : <strong> ".$ \_POST['flower']."</strong>< br />";

echo " His color : <strong> ".$ \_POST['color']."</strong>"; ?>

</p>

</body>​​

</ html >

On screen:

You have added a flower: **rose** .

Its color is **red** .

**php operators**

Most of the operators are similar to those in the C language.

To iterate through the array elements, a special foreach cycle has been introduced . Let's look at its syntax. There are two possible ways to write it:

foreach ( array as $ value )

{}

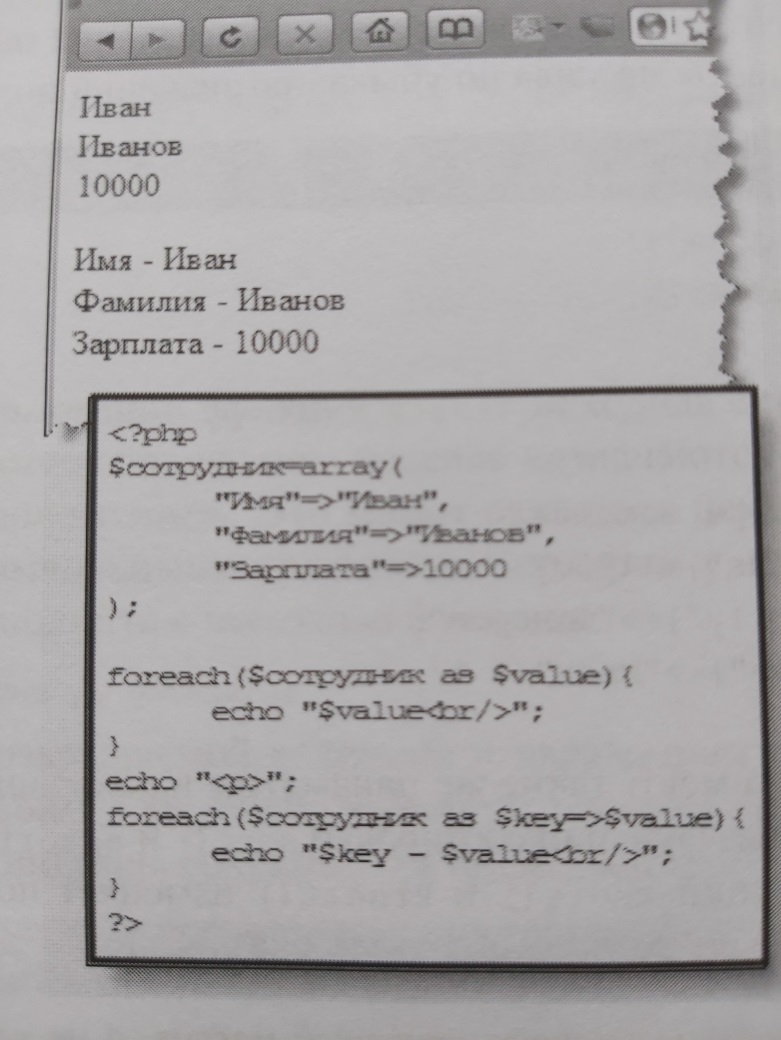
foreach ( array as $index => $value)

{}

Here $value is the name of the variable that contains the values of the array elements. And $index is the variable that contains the index (key) of the array element.

foreach( $\_POST as $key=>$value)

echo "$ key =$ value < br />";



**Working with the** XAMPP package.

There are quite a lot of different packages with a set of tools for simulating the work of a web server ( wamp - packages). The XAMPP package works stably, but it can easily be replaced with another one. The package is free, it is installed quickly.

Before you start working with the web server and database, you need to start the Appache server and MySQL DBMS ( start buttons on the control panel).

To create your own website in the XAMPP package, you need to create a folder in the C:\xampp\htdocs folder, for example wordpress , so that in the future you can access this website at <http://localhost/wordpress/> To do this, on the XAMPP control panel, you need to click the Explorer button , which opens the package directory.

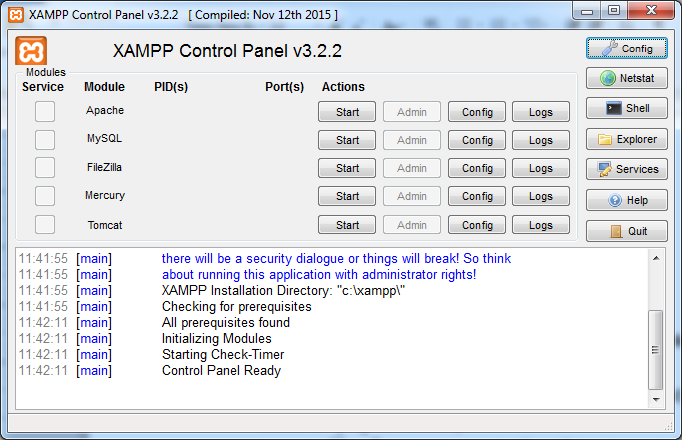


Fig. 1 XAMPP package window

To work with the database via the phpMyAdmin window , type localhost in the browser address bar when the XAMPP package is running . The window shown in Fig. 2 will appear. In this window, click the phpMyAdmin button , after which the window shown in Fig. 3 will appear.

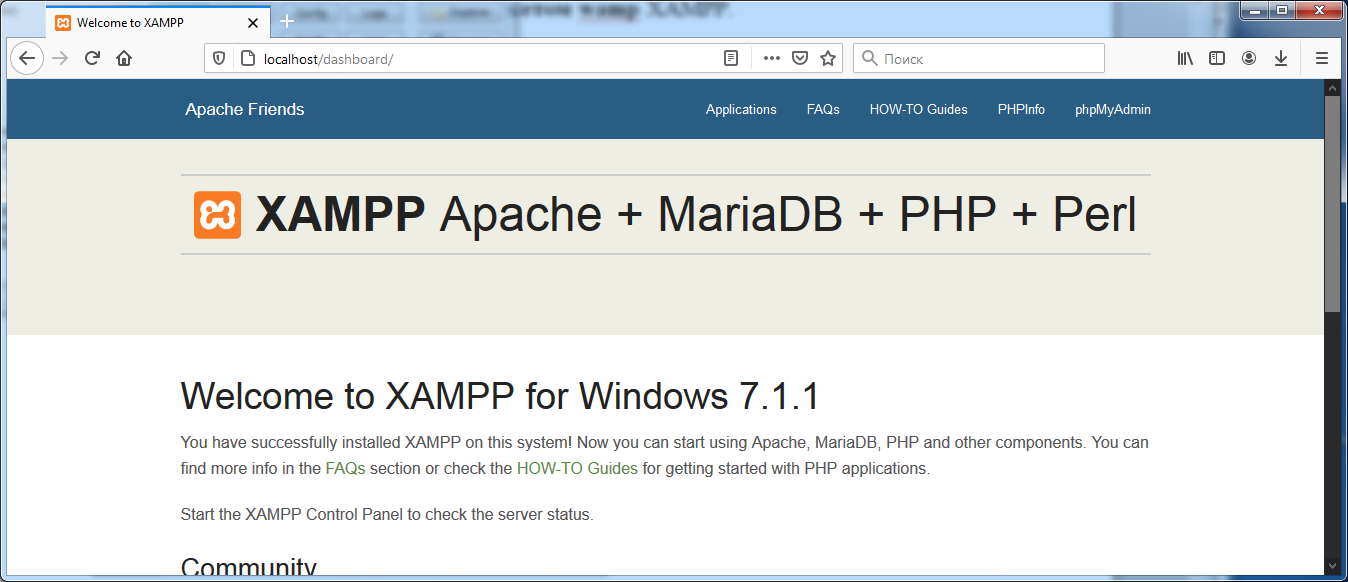


Fig. 2. XAMPP package window in the browser

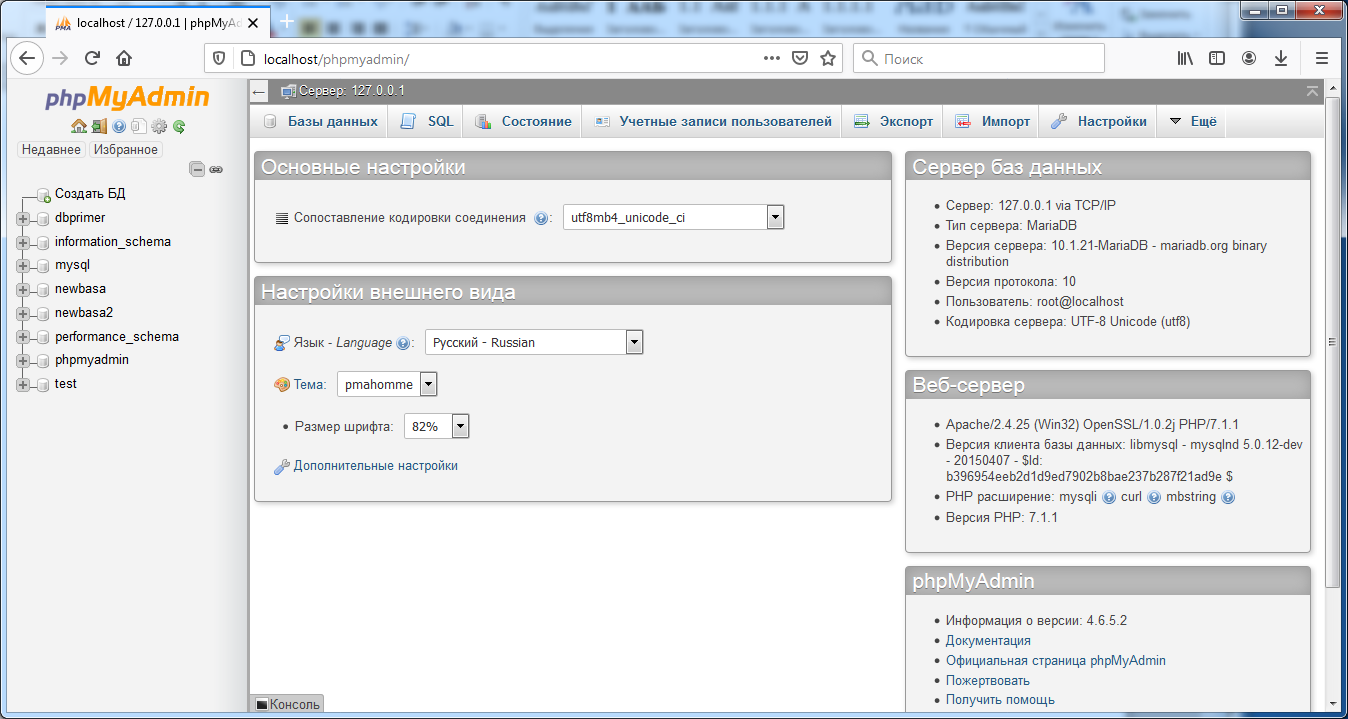


Fig. 3 DBMS window in the browser

*structured* query language) is used to access the contents of database tables. Example of a query for the *MY\_FLOWERS table* :

*SELECT \* FROM MY \_FLOWERS WHERE COLOR = “Blue”*

**Sample PHP code to display the contents of a database table**

/\* create connection \*/

mysql\_connect ($ hostname,$ username,$password ) OR DIE(" Not Can create connection ");

/\* choose base data . If will happen error - output its \*/ mysql\_select\_db ($ dbName ) or die( mysql\_error ());

/\* create a query that will select all Apple clients \*/

$query = "SELECT \* FROM $ userstable WHERE choice = ' Apples '";

/\* Execute the request. If an error occurs, display it. \*/

$res = mysql\_query ($query) or die( mysql\_ error ( ));

/\* How many of these were found \*/

$ number = mysql\_num\_rows ($ res );

/\* Print all in a beautiful form\*/

if ($ number == 0) { echo "<CENTER><P>No apple lovers</CENTER>"; } else { echo "<CENTER><P>Number of apple lovers: $ number <BR><BR>";

/\* Get one row at a time from the table into the $ row array until there are no more rows \*/

while ($row= mysql\_fetch\_array ($res))

{ echo " Customer ".$row['name']." likes Apples .<BR>"; echo " His Email: ".$row['email']; } echo "</CENTER>"; }

**Using mysqli functions .**

The updated version of the functions is now recommended for accessing the database. The letter i is added to their name . The general logic is the same as the older version, but there are some differences. Here is a simple script that connects to a MySQL server , sends a query to the server using this connection, prints the query results, and then releases the query result set and closes the connection.

<? php   
  
/\* Connect to MySQL server \*/   
$ link = mysqli\_connect (  
            ' localhost ' , /\* The host we are connecting to \*/  
            ' user ' , /\* Username \*/  
            ' password ' , /\* Password to use \*/  
            ' world ' ); /\* Default database for queries \*/   
  
if (! $ link ) {  
   printf ( "Unable to connect to database. Error code: %s\n" , mysqli\_connect\_error ());  
   exit ;   
}   
  
/\* Send request to server \*/   
if ( $ result = mysqli\_query ( $ link , 'SELECT Name , Population FROM City ORDER BY Population DESC LIMIT 5' )) {  
  
    print ( "Very large cities:\n" );  
  
    /\* Fetch query results \*/  
    while ( $ row = mysqli\_fetch\_assoc ( $ result )){  
        printf ( "%s (%s)\n" , $ row [ ' Name ' ], $ row [ ' Population ' ]);   
}  
  
    /\* Free up the memory used \*/  
    mysqli\_free\_result ( $ result );   
}   
  
/\* Close the connection \*/   
mysqli\_close ( $ link );   
?>

The above script should output the following data:

Very large cities :   
  
Mumbai ( Bombay ) ( 10500000 )   
Seoul ( 9981619 )   
Sao Paulo ( 9968485 )   
Shanghai ( 9696300 )   
Jakarta ( 9604900 )