

## **Упражнение 4. Инсталиране и конфигуриране на сървър за скриптова обработка – PHP**

### **1. Цел на упражнението.**

Целта на упражнението е да се получат практически знания и умения за инсталиране, конфигуриране и администриране на сървър за скриптова обработка на Web страници - PHP.

### **2. Теоретични сведения**

#### **2.1 Статични и динамични web страници**

Статичните web сайтове се състоят от фиксиран брой страници с фиксирано съдържание. Обикновено статичните web сайтове съдържат по-малко информация, по-бързо се разработват, но от друга страна за да промените съдържанието на дадена страница е необходимо да се промени кодът на страницата. Статичните сайтове предлагат по-базова функционалност и обикновено са написани на език HTML и CSS (Cascading Style Sheets). В някои случаи се добавя Java или JavaScript.

Динамичните сайтове предлагат много повече възможности и позволяват съдържанието на всеки потребител да бъде генерирано динамично, според неговите настройки, предпочитания или други параметри. Примери за динамични сайтове са Facebook, LinkedIn, почти всички електронни магазини и форуми. За да може да се изгради динамичен web сайт, част от страниците трябва да се напишат на скриптов език и на web сървъра трябва да бъде инсталиран и конфигуриран сървър за скриптова обработка.

Съществуват няколко популярни скриптов езика. Microsoft разработват и използват технологията с отворен код ASP.NET (Active Server Pages). [1]

Вероятно най-популярната технология за създаване на динамични web страници е PHP (Hypertext Preprocessor) – широко използван скриптов език с отворен код, специално насочен за разработка на web страници, който може да бъде вграден в кода на HTML. За разлика от Javascript кода, който се изпълнява в клиентската страна, PHP кодът се изпълнява на сървъра, където се генерира HTML код, който се изпраща към клиента.

PHP може да се използва на повечето операционни системи, включително Linux, Microsoft Windows, Mac OS X и други. PHP се поддържа от повечето съвременни web сървъри, като Apache, IIS, и много други. Така с PHP имате свободата да изберете операционна система и web сървър. Освен това може да се избира дали да се използва процедурно програмиране, обектно-ориентирано или комбинация от двете.

В момента се предлагат две главни версии на PHP - 5.6.27 и 7.0.12. Вариантите на версия 7 имат по-нови и добри възможности, но версия 5 е по-съвместима със стари проекти. В настоящото упражнение се използва PHP версия 7.0.12.

## 2.2 Инсталиране на PHP

Предлагат се пакети за инсталация на различни операционни системи. Под Windows може да се избира между 32 и 64 битова версия. Желаният пакет може да се изтегли от <http://windows.php.net/download#php-7.0>. За целите на настоящото упражнение се използва 64 битова версия.

Софтуерът има две разновидности за Windows - "Thread Safe" - ако ще се използва Apache Web Server и "Non Thread Safe" - ако ще ползваме IIS Web Server.

Ако на компютъра не е инсталиран пакетът Visual C++ Redistributable for Visual Studio 2015 x64, той трябва да се инсталира предварително. Пакетът може да се изтегли от <https://www.microsoft.com/en-us/download/details.aspx?id=48145>.

Изтегленият файл се разархивира в произволна папка на сървъра, например C:\PHP. За да се настрои PHP да работи заедно с Apache е необходимо в конфигурационния файл на Apache – httpd.conf да се направят следните настройки:

```
AddHandler application/x-httpd-php .php  
AddType application/x-httpd-php .php .html  
LoadModule php7_module "c:/php/php7apache2_4.dll"  
PHPIniDir "c:/php"
```

За да се отворят по подразбиране PHP файловете е необходимо да се редактира и следната секция по указания начин:

```
<IfModule dir_module>  
DirectoryIndex index.php index.html  
</IfModule>
```

Необходимо е да се рестартира web сървъра Apache, за да се възприемат промените.

## 2.3 Конфигуриране на PHP

Конфигурационният файл на PHP е C:\PHP\php.ini. Настройките се правят чрез написване на различни текстове в конфигурационните файлове. Редовете, започващи с „;“ са коментари. В някои случаи е необходимо част от тях да се активират, премахвайки символа „;“ в началото на реда. Ако файлът липсва, създайте го, като копирате файла php.ini-development като php.ini. Тук ще бъдат разгледани част от най-важните конфигурационни параметри във файла. Пълно описание на възможните директиви може да се види в [2].

**engine = On** – разрешава изпълнението на PHP скриптове в Apache;

**output\_buffering = 4096** – определя размера на буфера в байтове, който PHP пази, преди да изпрати данните към клиента;

**disable\_functions =** – позволява забрана за изпълнението на някои функции от съображения за сигурност;

**max\_execution\_time = 30** – максимално време за изпълнение на скрипт в секунди;

**memory\_limit = 128M** – ограничение на оперативната памет, която може да използва всеки скрипт;

**error\_reporting = E\_ALL** – указва типовете грешки, които да се показват;

**display\_errors = On** – определя дали да се показват грешки;

**log\_errors = On** – определя дали грешките да се записват във файл;

**log\_errors\_max\_len = 1024** – определя максималната дължина на записваните грешки;

**post\_max\_size = 8M** – определя максималният размер на изпратени данни с метод POST, които PHP да приема;

**default\_charset = "UTF-8"** – определя подразбиращия се набор символи. За поддръжка на кирилица е добре да бъде UTF-8;

**file\_uploads = On** – указва дали е позволено качването на файлове от PHP;

**upload\_tmp\_dir =** – определя папката, в която се качват файлове;

**upload\_max\_filesize = 2M** – определя максималният размер на качваните файлове;

**max\_file\_uploads = 20** – указва максималният брой файлове, които могат да се качат с една заявка;

**session.use\_cookies = 1** – определя дали да се използват „бисквитки“ (cookies);

#### **Връзка към база данни MySQL:**

**extension=php\_mysqli.dll** – трябва да бъде разкоментирано за поддръжка на база данни MySQL;

В секция [MySQLi]:

**mysqli.max\_persistent = -1** – максимален брой връзки към базата данни (-1 означава, че няма ограничение);

**mysqli.default\_port = 3306** – номер на порта на MySQL сървъра;

**mysqli.default\_host = 127.0.0.1** - адрес на MySQL сървъра;

**mysqli.default\_user = root** – Определя потребителя, от чието име ще се достъпва базата данни;

**mysqli.default\_pw =** - парола, която да се използва за достъп до базата данни. От съображения за сигурност не е добре да се запазва паролата в този файл;

#### **Изпращане на електронна поща от PHP:**

**SMTP = localhost** – определя адреса на сървъра за електронна поща;

`smtp_port = 25` – указва номера на порта на пощенския сървър, по подразбиране 25;

`sendmail_from = me@example.com` – указва e-mail адреса, от който ще се изпраща пощата;

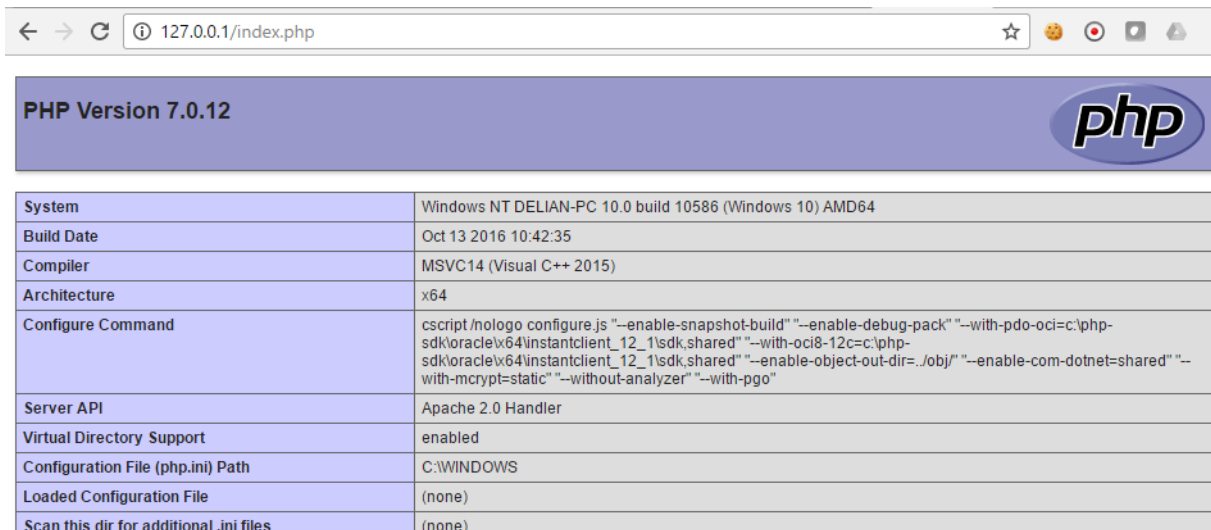
`mail.log = syslog` – указва къде ще се записва отчет за изпратените пощи (в примера – в Event Log на Windows);

## 2.4 Проверка на работата на PHP

Проверката дали PHP и Apache работят правилно може да се направи чрез копиране на следния програмен код във файл C:\Apache24\htdocs\index.php:

```
<?php
phpinfo();
?>
```

След това се отваря локалния web сайт с името на файла, както е показано на фигура 4.1.



The screenshot shows a web browser window with the address bar containing "127.0.0.1/index.php". The page content is a purple header with "PHP Version 7.0.12" and the PHP logo. Below the header is a table with the following data:

System	Windows NT DELIAN-PC 10.0 build 10586 (Windows 10) AMD64
Build Date	Oct 13 2016 10:42:35
Compiler	MSVC14 (Visual C++ 2015)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-sdk\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=.\obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)

Фиг. 4.1. Страница, показваща информация за PHP

На тази страница може да се намери доста подробна информация за PHP и компютъра, на който е инсталиран.

### Глобални променливи

PHP поддържа няколко глобални променливи, част от които са `$_SERVER`, `$_POST`, `$_GET`. Следният код може да се използва за проверка на стойностите на глобалните променливи:

```
<html>
  <head>
    <title>Globals</title>
```

```
</head>
<body>
<?php
    print_r($_SERVER);
    print "<br />";
    print_r($_GET);
    print "<br />";
    print_r($_POST);
?>
</body>
</html>
```

## 2.4 Пакет за достъп до база данни MySQL – phpMyAdmin

Популярен пакет, написан на езика PHP е безплатния инструмент за работа с MySQL бази данни – phpMyAdmin. Той може да бъде изтеглен от сайта на проекта [phpmyadmin.net](http://phpmyadmin.net). Към момента на написване на това упражнение актуалната версия е 4.6.6. Инсталационния пакет се изтегля и разархивира в папка под папката с HTML документи на Web сървъра, например C:\Apache\htdocs\phpmyadmin.

Във файла `php.ini` трябва да се премахне знакът за коментар пред следните редове:

```
extension=php_bz2.dll
extension=php_mbstring.dll
```

Конфигурационния файл на пакета е `config.inc.php`, но съвременните версии могат да бъдат конфигурирани извиквайки предварително подготвения скрипт в папката `/setup` и може да бъде стартирана през web browser чрез:

<http://127.0.0.1/phpmyadmin/setup>

След настройка на необходимите параметри трябва да се изтегли направения от процедурата конфигурационен файл на локалния компютър чрез бутона `Download` и после да бъде копиран в папката `phpmyadmin` на сървъра ръчно или чрез FTP.

Влизането без парола е забранено по подразбиране в phpMyAdmin от съображения за сигурност. За да се позволи влизане без парола, в конфигурационния файл на phpMyAdmin – `config.inc.php` трябва да се добави следния ред:

```
$cfg['Servers'][$i]['AllowNoPassword'] = TRUE;
```

Забележка: В упражнението за улеснение се конфигурира достъп без зададена парола за потребителя `root` на `mysql`. В реална среда това представлява заплаха за сигурността и трябва да се използва достатъчно силна парола.

### 3. Задачи за изпълнение

1. Стартирайте виртуална машина с инсталирани Web сървър Apache и база данни MySQL. Инсталирайте PHP, настройте го и проверете функционалността на PHP чрез функцията `phpinfo()`.
2. Проверете стойностите на глобалните променливи, използвайки кода по-горе.
3. Инсталирайте и конфигурирайте уеб базирания клиент за база данни phpMyAdmin. Проверете функционалността му.
4. Чрез phpMyAdmin създайте база данни **University** и таблица **Student** с три колони – **Student\_id** (INT auto increment), **name** (varchar (30)), **Speciality** (varchar (8)).
5. Създайте няколко записа като използвате phpMyAdmin (Избира се бутона Insert).
6. Създайте потребител **student**, задайте му парола **stud123parola** и му задайте права за четене и запис към базата данни **University**.
7. Изведете съдържанието на таблицата чрез phpMyAdmin.

### 4. Контролни въпроси

1. Защо е необходим PHP ?
2. Може ли един PHP скрипт (програма) да се свърже към два сървъра за бази от данни?
3. Коя програма може да замени phpMyAdmin?

### 5. Допълнителни ресурси

1. Microsoft, ASP.NET, <https://www.asp.net/>, дата на използване 05.11.2016 г.
2. Списък на директивите на `php.ini`, <http://php.net/manual/bg/ini.list.php>, дата на използване 05.11.2016 г.
3. phpMyAdmin's Documentation, <https://docs.phpmyadmin.net/en/latest/>, дата на използване 11.11.2016 г.
4. PHP 5 Tutorial, <http://www.w3schools.com/php/>, дата на използване 11.11.2016 г.