

КАТЕДРА: КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ
ДИСЦИПЛИНА: ИНФОРМАЦИОННИ СИСТЕМИ

ЛАБОРАТОРНО УПРАЖНЕНИЕ № 12

ТЕМА: Изпращане на емайли чрез SMTP сървър

ЦЕЛ:

Целта на упражнението е студентите да получат практически знания и умения за особеностите при изпращане на емайли от разработено приложение чрез HTML и PHP скрипт

I. ТЕОРЕТИЧНА ЧАСТ

При използването на контактна форма, форма за регистрация, поръчки, емайл бюлетин и други в уеб базирана система е необходимо тя да може да изпраща и емайли. За целта е необходимо да съществува SMTP сървър. За демонстрацията в настоящето упражнение ще се използва SMTP сървъра на google, като за изпращач се използва новосъздаден тестов gmail.

1. Конфигуриране на WAMP

Преди да можем да се изпращат емайли е необходимо да се направят на няколко конфигурации на wamp сървъра. Във файла php.ini трябва да се добави пътя до системното приложение, което ще се използва за трансфер на мейл. Ако се използва последната версия на wamp пакета, то в директорията на сървъра трябва да има папка с име sendmail, съдържаща приложението sendmail. Ако такава папка липсва или липсва това приложение е необходимо това приложение да се намери и запише в папката.

В лабораторно упражнение номер 10 вече се запознахте със файл php.ini. Реда, който трябва да се редактира се намира в раздела „Dynamic Extensions“ и подраздел mail function.

```
[mail function]
; For Win32 only.
; http://php.net/smtp
;SMTP = localhost
; http://php.net/smtp-port
;smtp_port = 25

; For Win32 only.
; http://php.net/sendmail-from
;sendmail_from = "admin@wampserver.invalid"

; For Unix only. You may supply arguments as well (default: "sendmail -t -i").
; http://php.net/sendmail-path
sendmail_path = "D:\wamp\sendmail\sendmail.exe -t -l";
```

Редовете, оцветени с червен цвят са редовете, които трябва да се редактират. Реда за SMTP и smtp_port трябва да се коментират (коментирането на ред става като се сложи точка и запетая пред реда.). В последния ред трябва да се премахне знака за коментар и в него да се добави пътя към sendmail.exe.

Следващата стъпка е да се конфигурира файлът sendmail.ini, които е записан в директория sendmail. Той съдържа следните редове.

```
[sendmail]
smtp_server=smtp.gmail.com
; smtp port (normally 25)
smtp_port=465
; SMTPS (SSL) support
; auto = use SSL for port 465, otherwise try to use TLS
; ssl = alway use SSL
; tls = always use TLS
; none = never try to use SSL
smtp_ssl=ssl
default_domain=localhost
error_logfile=error.log
debug_logfile=debug.log
auth_username=testmail@gmail.com
auth_password=*****
pop3_server=
pop3_username=
pop3_password=
force_sender=
force_recipient=
hostname=localhost
```

Редовете, оцветени с червен цвят са редовете, който са коригирани.

На първия ред за smtp_server се посочва SMTP сървъра който ще се използва. Необходимо е също да се зададе порт, домейна на сървъра и данните за емайла, от който ще се изпращат съобщенията.

2. Функция mail() в PHP

PHP разполага със специална функция, наречена mail(), която изпраща емайл-и. Тя има следния синтаксис:

mail(\$to,\$subject,\$msg,\$header,\$additional_parameters)

Описанието на параметрите е следното:

- *to* – Получателя на писмото. Трябва да съдържа коректен email адрес.
- *subject* – Темата на съобщението.
- *msg* – Съдържанието на самото съобщение
- *header* – Незадължителен параметър. Използва се за добавяне на допълнителни заглавия (From, Cc и Bcc). Няколко допълнителни заглавия трябва да бъдат разделени със CRLF (\r\n)

- *additional_parameters* – Незадължителен параметър. Може да се използва за подаване на допълнителни флагове под формата на аргументи на командния ред за програма.

Ако писмото е било изпратено функцията ще върне като резултат “true”, в противен случай – „false“.

II. ПРАКТИЧЕСКА ЧАСТ

!ВАЖНО: За да можете да тествате упражнението ви е необходим следния софтуер. Web server, Php интерпретатор, MYSQL база данни. Всеки един от тези софтуери ги има в пакетите WAMP или XAMP. Настоящото упражнение е тествано върху WAMP Version 3.2.0.

ЗАДАЧА1: Напишете PHP скрипт който да изпраща съобщение на избран от вас емайл със следното съобщение: „Send from localhost“, и следната тема: „LOCALHOST SUBJECT“. Включете допълнителна заглавка FROM:

Код за решение на задача:

```
<?php
$to="testmail@abv.bg";
$subject="LOCALHOST SUBJECT";
$message="Send from localhost";
$headers="FROM: testmail@gmail.com";
If(mail($to, $subject, $message, $headers))
echo "Email sent";
else echo "Email sending failed";
?>
```

Пояснение за задача 1

Предложеният код изпраща съобщение на „testmail@abv.bg“ със следните тема "LOCALHOST SUBJECT" и съобщение "Send from localhost", както и допълнително заглавие, което показва кой е изпращачът. Оператор if проверява дали съобщението е изпратено и извежда съответното съобщение в браузъра. Ако получите, че съобщението ви не е изпратено, то може да проверите грешката във файла „error.log“, който се намира в директорията на sendmail.

Задача 2: Създайте форма която предоставя възможност на потребителя да напише своето съобщение и да го изпрати при натискане на бутон.

Код за решение на задача:

```
<html>
<body>
<form action='<?php echo $_SERVER["PHP_SELF"]; ?>' method="Post">
<p><label for="to">To</label>
<input type="text" name="to" id="to"></p>
<p><label for="from">From</label>
<input type="text" name="from" id="from"
value="testmail@gmail.com"></p>
<p><label for="Subject">Subject</label>
```

```

        <input type="text" name="Subject" id="to"></p>
        <p><label for="Msg">Message</label>
        <textarea name="Msg" id="Msg"></textarea></p>
        <input type='submit' name='submit' />
    </form>
</body>
</html>
<?php
if(isset($_POST['submit']))
{
    $to=$_POST['to'];
    $subject=$_POST['Subject'];
    $message=$_POST['Msg'];
    $headers="FROM: ".$_POST['from'];
    If(mail($to, $subject, $message, $headers))
    echo "Email sent";
    else echo "Email sending failed";
}

```

Пояснение за задача 2

Чрез html кода се създава, форма която позволява на потребителя да изпраща съобщения. Тя е показана на фиг.1. Полето на подателя е попълнено автоматично. При изпращане на съобщението на посочения email адрес, се получава информация за това дали то е изпратено успешно или не.

Фиг.1. Форма за изпращане на емайл

III. Задача за самостоятелна работа.

1. Редактирайте задача 2 от практическата част, така че да се проверява коректно ли са попълнени всички полета.
2. Редактирайте задача 2 в част III от лабораторно упражнение номер 8, така че при регистрация на нов потребител да се изпраща email за потвърждение.
3. Редактирайте задача 3 в част III от лабораторно упражнение номер 8, така че при забравена парола да се изпраща емайл, който да предоставя възможност за ресет на паролата

Указание: В съобщението да се съдържа линк към форма, която предоставя възможност да си въведе нова парола