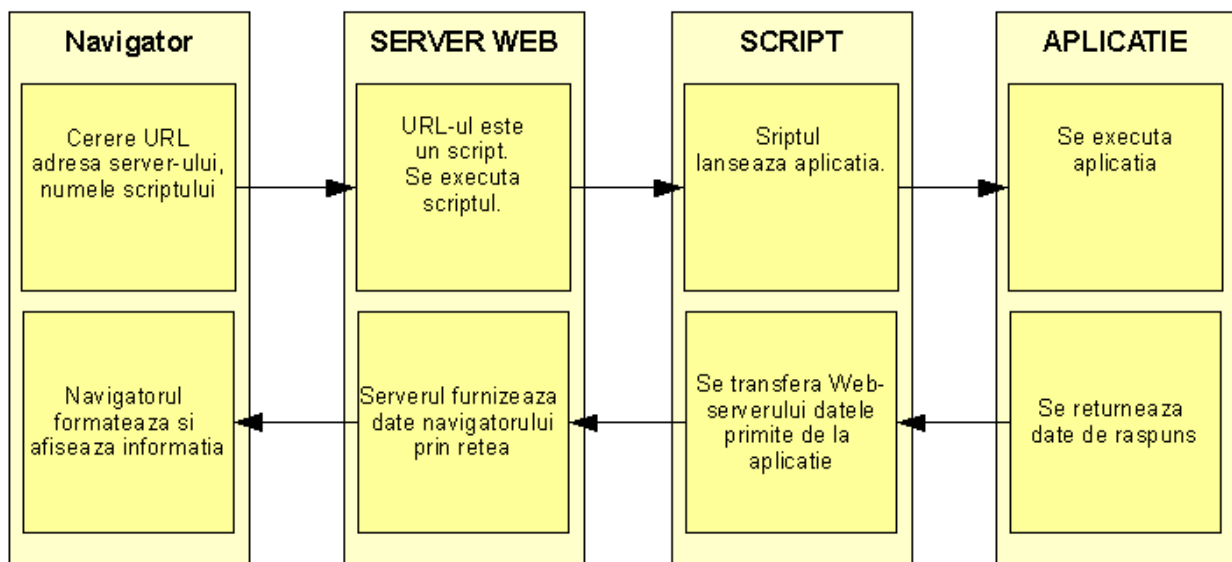


Programarea aplicațiilor server side

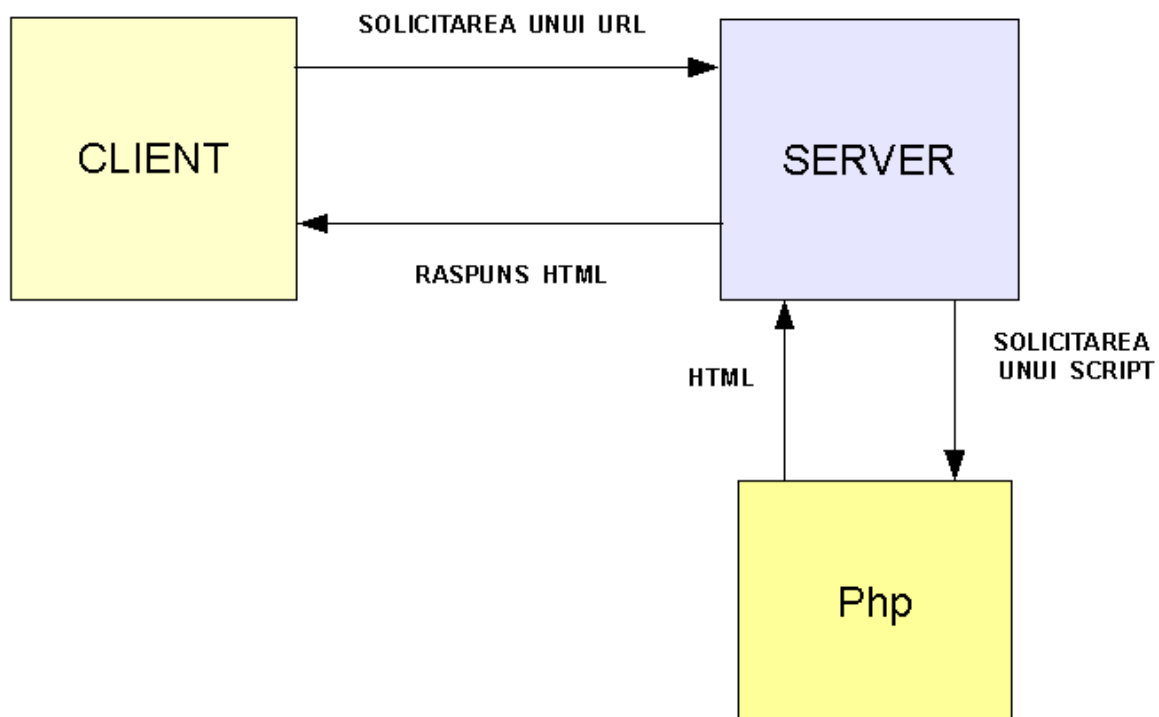
Aplicații server side

În momentul în care un client face o cerere sub forma unui URL, WEB serverul HTTP la care ajunge cererea, analizează dacă URL-ul se referă la o pagină html sau la un script. În cazul că cererea este un script , serverul HTTP pasează cererea spre o aplicație care poate interpreta scriptul respectiv. În acest moment avem de-a face cu o aplicație server side care se derulează după principiul afișat grafic mai jos.

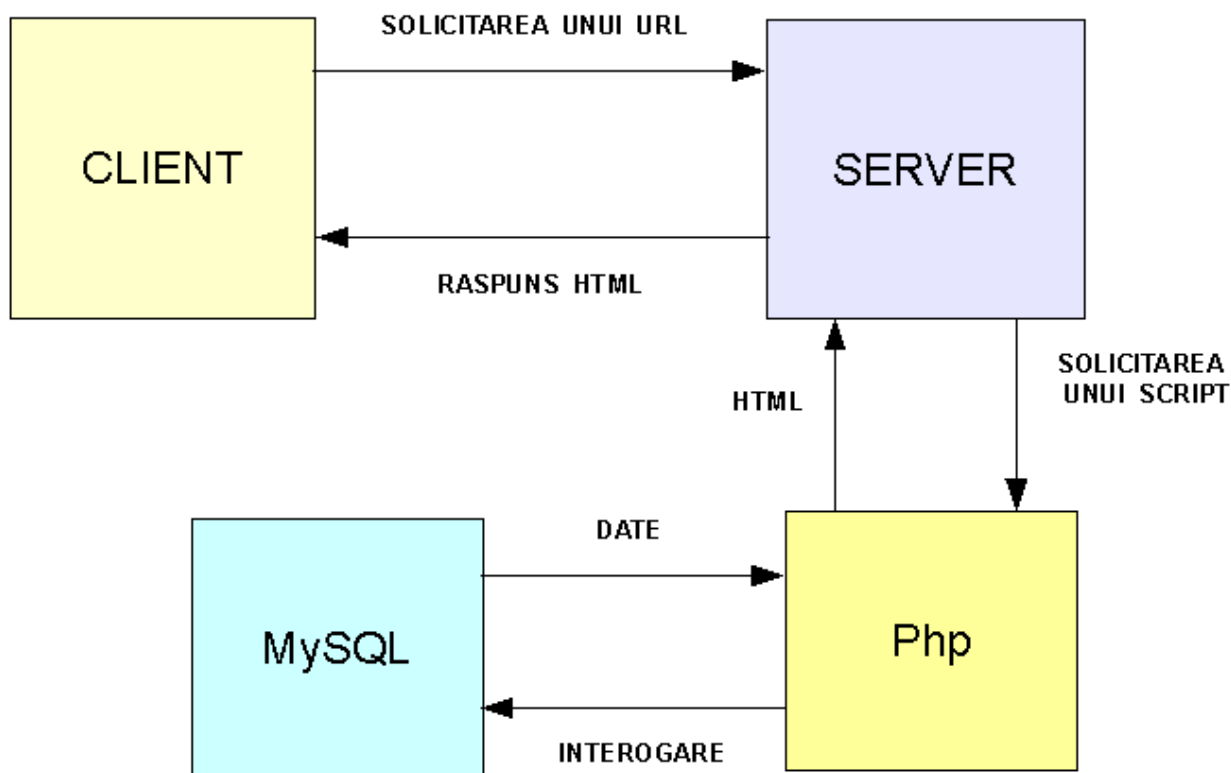


- **Aplicații Php server side**

În cazul în care scriptul este o aplicație Php procesul de execuție al aplicației server side este:



În majoritatea cazurilor, aplicațiile php apelează baze de date. Baza de date MySQL este cea mai utilizată bază de date accesată de aplicațiile php. În acest caz aplicația client -server se poate reprezenta astfel:



Programarea in Php

Php (Personal Home Page) este un limbaj de scriptare creat inițial de Rasmus Lerdorf pentru contorizarea vizitatorilor unei pagini. Limbajul a fost dezvoltat ulterior devenind un limbaj de scriptare pe partea server-ului (server side). PHP înseamnă astăzi Hypertext Preprocessor. Este o tehnologie de scriptare interplatformă, rulând pe diverse sisteme de operare și pe diferite platforme hard. Limbajul are facilități de operare cu diverse baze de date, fiind recomandat însa să se lucreze cu baza de date MySQL.

• Elemente de programare in Php

Un limbaj de programare server side dispune pe lângă un set de instrucțiuni de bază, de un mecanism de transmitere a datelor spre și dinspre browser. Pentru început vom aborda elementele de bază ale limbajului, urmând să completăm ulterior cu mecanisme pentru comunicarea cu diverse browsere și accesul la baze de date.

• Sintaxa de baza a limbajului

Un script PHP începe cu `< ? php` și se încheie cu `? >`. Scripturile PHP pot fi incluse în orice loc în cadrul unui document html. În general serverele acceptă și forma prescurtată a unui script sub forma: `< ?....? >`. Un fișier PHP conține în mod normal Marcatori HTML și scripturi PHP. După metoda consacrată, prima aplicație tipărește un text de bun venit!

Următoarea aplicație folosește php pentru a afișa mesajul "Bine ați venit!".

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
echo "Bine ati venit!";
?>
</td></tr></table></center>
</body>
</html>
```

Bine ați venit!

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
echo "Utilizarea comentariilor si a blocurilor de comentarii";
// Comentariu simplu
/* Bloc de
comentarii*/
?>
</td></tr></table></center>
</body>
</html>
```

Utilizarea comentariilor si a blocurilor de comentarii

• Utilizarea variabilelor

În orice limbaj de programare există posibilitatea definirii și utilizării variabilelor. Variabilele memorează valori, șiruri de caractere, tablouri etc.

PHP este un limbaj slab tipizat, permițând utilizarea variabilelor fără ca acestea să fie declarate în prealabil.

Forma generală pentru inițializarea unei variabile este:

```
$nume_variabila=valoare;
```

Să realizăm o aplicație care inițializează și utilizează variabile:

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
echo "Capitolul :";
$nr_cap=1;
echo $nr_cap;
echo "<br>Utilizarea variabilelor in :";
$nume_l="limbajul PHP";
echo $nume_l;
?>
</td></tr></table></center>
</body>
</html>
```

Capitolul :1
Utilizarea variabilelor in :limbajul PHP

Același rezultat obținem rulând următorul script:

```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$nr_cap=1;
$nume_l="limbajul PHP";
echo "Capitolul :$nr_cap <br>Utilizarea variabilelor in :$nume_l" ;
?>
</td></tr></table></center>
</body>
</html>

```

Capitolul :1
Utilizarea variabilelor in :limbajul PHP

În PHP exista diferența între ghilimelele duble " si ghilimelele simple '. Dacă în scriptul de mai sus înlocuim ghilimelele duble cu ghilimelele simple obținem:

```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$nr_cap=1;
$nume_l="limbajul PHP";
echo 'Capitolul :$nr_cap <br>Utilizarea variabilelor in :$nume_l' ;
?>
</td></tr></table></center>
</body>
</html>

```

Capitolul :\$nr_cap
Utilizarea variabilelor in :\$nume_l

În acest caz variabilele nu au mai fost interpretate ele comportându-se ca un simplu text.

De multe ori este necesar să introducem ghilimele în text. Dacă dorim de exemplu să atribuim unei variabile valoarea: Universitatea " Petru Maior", va trebui să folosim caracterul \ pentru a nu se interpreta ghilimelele din cadrul textului.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$nr_cap=1;
$nume_l="limbajul PHP";
echo 'Capitolul :$nr_cap <br>Utilizarea variabilelor in :$nume_l' ;
?>
</td></tr></table></center>
</body>
</html>
```

Studiez la :Universitatea "Petru Maior"

În acest caz primul caracter " de după \ nu a fost interpretat.

• Șiruri de caractere

În PHP se definesc o serie de funcții pe șiruri de caractere, foarte utile ținând cont că PHP este un limbaj de scriptare și trebuie să gestioneze conținut de pagini web. Printre cele mai importante funcții se număra:

- concatenarea șirurilor
- determinarea lungimii unui șir
- căutarea unui subșir în cadrul unui șir
- Compararea a doua șiruri

Concatenarea șirurilor

Pentru concatenarea a două șiruri, se folosește operatorul punct (.)

```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$nr_cap=1;
$nume_l="limbajul PHP";
echo 'Capitolul :$nr_cap <br>Utilizarea variabilelor in :$nume_l' ;
?>
</td></tr></table></center>
</body>
</html>

```

Studiez la: Facultatea de Inginerie
Sectia-Calculatoare

Determinarea lungimii unui șir

Pentru determinarea lungimii unui șir se folosește funcția: **strlen()**

```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$fc="Facultatea de Inginerie";
echo "Lungimea sirului : $fc , este: ".strlen($fc);
?>
</td></tr></table></center>
</body>
</html>

```

Lungimea sirului : Facultatea de Inginerie , este: 23

Căutarea unui subșir în cadrul unui șir

Pentru căutarea unui subșir în cadrul unui șir, se folosește funcția: **strpos()**


```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$fc="Facultatea de Inginerie";
echo "Lungimea sirului : $fc , este: ".strlen($fc);
?>
</td></tr></table></center>
</body>
</html>

```

Subsirul "Inginerie", se gaseste in sirul: "Facultatea de Inginerie", incepand cu pozitia:14

Compararea a două șiruri

Pentru a compara doua șiruri între ele , se folosește funcția: **strcmp()**

```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$fc="Facultatea de Inginerie";
echo 'Sirul "Inginerie", comparat cu sirul: "'. $fc. "', da rezultatul:'. strcmp($fc , "Inginerie");
?>
</td></tr></table></center>
</body>
</html>

```

Sirul "Inginerie", comparat cu sirul: "Facultatea de Inginerie", da rezultatul:-1

Vom compara acum doua șiruri egale.

```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$fc="Facultatea de Inginerie";
echo 'Sirul "Inginerie", comparat cu sirul: "'. $fc. '", da rezultatul:'. strcmp($fc , "Inginerie");
?>
</td></tr></table></center>
</body>
</html>

```

Sirul "Facultatea de Inginerie", comparat cu sirul:
"Facultatea de Inginerie", da rezultatul:0

• Operatori utilizați in PHP

Operatorii utilizați in PHP sunt similari cu operatorii utilizați in Java Scrips sa C++

Operatori aritmetici

Operator	Descriere
+	Adunare
-	Scadere
*	Inmultire
/	Impartire
%	Modulo
++	Increment
--	Decrement

Operatori de asignare

Operator	Exemplu	Echivalent cu
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Operatori de comparare

Operator	Descriere
==	este egal cu
!=	nu este egal cu
>	este mai mare decat
<	este mai mic decat
>=	este mai mare sau egal cu
<=	este mai mic sau egal cu

Operatori logici

Operator	Descriere
&&	and
	or
!	not

- **Instrucțiuni decizionale**

Sintaxa:

```
if (conditie)
{
    cod care va fi executat daca conditia e adevarata
}
else
{
    cod care va fi executat daca conditia e falsa
}
```

Vom folosi instrucțiuni decizionale pentru a afișa un mesaj in funcție de data curentă.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
//$an=idate("Y");
$an_c=date("Y");
if ($an_c%4==0)
echo 'Anul: ' . $an_c . ' este an bisect';
else
echo 'Anul: ' . $an_c . ' este an ne-bisect';
```

```
?>
</td></tr></table></center>
</body>
</html>
```

Anul: 2010 este an ne-bisect

Sintaxa instrucțiunii elseif

```
if (conditie)
{
    cod care va fi executat daca conditia e adevarata
}
elseif(conditie){
    cod care va fi executat daca conditia e adevarata
}
else
{
    cod care va fi executat daca conditia e falsa
}
```

Vom folosi instrucțiunea decizională elseif pentru a afișa un salut în funcție de ziua din săptămână.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$d=date("D");
if ($d=="Fri")
    echo "Weekend placut!";
elseif ($d=="Sun")
    echo "Duminica placuta!";
else
    echo "O zi buna!";
?>
</td></tr></table></center>
</body>
</html>
```

O zi buna!

• Instrucțiuni de ciclare

În PHP sunt definite patru instrucțiuni pentru realizarea buclilor repetitive de programare.

- **while**
- **do...while**
- **for**
- **foreach**

Sintaxa instrucțiunii **while**:

```
while (conditie) {  
    bloc de instructiuni ce va fi executat;  
}
```

Utilizând instrucțiunea de buclare **while** vom afișa primele 10 de numere naturale. Fiecare număr va fi afișat într-o celulă separată a unui tabel.

```
<html>  
<body>  
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>  
<center>  
Primele 100 de numere naturale  
<hr>  
<?php  
echo"<table border=1>";  
$i=1;  
while ($i<=10) {  
    echo"<td>";  
    echo $i."</td>";  
    $i+=1;  
}  
echo"</table>";  
?>  
</td></tr></table></center>  
</body>  
</html>
```

Primele 10 de numere naturale

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Sintaxa instrucțiunii do..while:

```
do {  
    bloc de instructiuni ce va fi executat;  
}  
while (conditie);
```

Utilizând instrucțiunea de buclare **do...while** vom afișa primele 100 de numere naturale.

```
<html>  
<body>  
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>  
<center>  
    Primele 100 de numere naturale  
<hr>  
<?php  
echo"<table border=1>";  
$i=1;  
$j=1;  
$k=1;  
do {  
    echo"<tr>";  
    $i=1;  
    do {  
        echo"<td>";  
        echo $k."</td>";  
        $i+=1;  
        $k+=1;  
    } while ($i<=10);  
    $j+=1;  
    echo"</tr>";  
    } while ($j<=10);  
echo"</table>";  
?>  
</td></tr></table></center>  
</body>  
</html>
```

Primele 100 de numere naturale

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Sintaxa instrucțiunii for:

```
for (init; cond; incr) {
    bloc de instructiuni ce va fi executat;
}
```

Reluăm aplicația anterioară însă vom folosi instrucțiunea de buclare **for** pentru a fișă pătratele primelor 100 de numere naturale.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
echo"<table border=1>";
$k=1;
for ($j=1;$j<=10;$j++){
    echo"<tr>";
    $i=1;
    for ($i=1;$i<=10;$i++){
        echo"<td>";
        echo $k*$k."</td>";
        $k+=1;
    }
    echo"</tr>";
```

```

}
echo"</table>";
?>
</td></tr></table></center>
</body>
</html>

```

Patratele primelor 100 de numere naturale									
1	4	9	16	25	36	49	64	81	100
121	144	169	196	225	256	289	324	361	400
441	484	529	576	625	676	729	784	841	900
961	1024	1089	1156	1225	1296	1369	1444	1521	1600
1681	1764	1849	1936	2025	2116	2209	2304	2401	2500
2601	2704	2809	2916	3025	3136	3249	3364	3481	3600
3721	3844	3969	4096	4225	4356	4489	4624	4761	4900
5041	5184	5329	5476	5625	5776	5929	6084	6241	6400
6561	6724	6889	7056	7225	7396	7569	7744	7921	8100
8281	8464	8649	8836	9025	9216	9409	9604	9801	10000

Sintaxa instrucțiunii foreach :

```

foreach (tabel as variabila){
    bloc de instructiuni ce va fi executat;
}

```

Următoarea aplicație utilizează instrucțiunea de buclare **foreach** pentru a afișa zilele săptămânii.


```

<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$zile=array("Luni", "Marti", "Miercuri", "Joi", "Vineri", "Sambata", "Duminica");
echo "Zilele sapamanii sunt:\n ";
foreach ($zile as $zi){
echo $zi . "<br />";
}
?>
</td></tr></table></center>
</body>
</html>

```

<p style="text-align: center;">Zilele sapamanii sunt:</p> <p>Ziua 1 : Luni</p> <p>Ziua 2 : Marti</p> <p>Ziua 3 : Miercuri</p> <p>Ziua 4 : Joi</p> <p>Ziua 5 : Vineri</p> <p>Ziua 6 : Sambata</p> <p>Ziua 7 : Duminica</p>

• Utilizarea funcțiilor in php

Vom utiliza in continuare funcții în cadrul script-urilor php. Următoarea aplicație definește și utilizează funcția patrat pentru a calcula pătratul unui număr.

```

<html>
<body><center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
function patrat($x){
$patr = $x*$x;
return $patr;
}
$nr=25;
echo 'Patratal numarului '$nr . 'este :'. patrat($nr);
?>
</td></tr></table></center>
</body></html>

```

<p>Patratal numarului 25este :625</p>

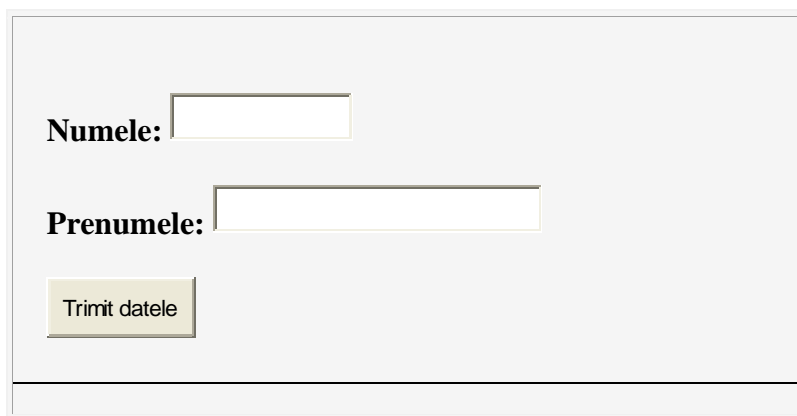
Form-uri în php

În momentul definirii unui form trebuie precizată aplicația care va prelucra datele trimise, precum și metoda prin care se trimit datele.

- **Utilizarea form-urilor**

Vom realiza un form care cere numele și prenumele, și precizează aplicația care va prelucra datele trimise. Mai jos este prezentată sursa aplicației care definește form-ul.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<form action="apl_s1.php" method="post"><br>
<p> <b>Numele:</b> <input type="text" name="nume" size="15" maxlength="15" value=" "
/></p>
<p> <b>Prenumele:</b> <input type="text" name="p_nume" size="30" maxlength="30"
value=" " /></p>
<p> <INPUT type = SUBMIT value="Trimit datele">
</form>
</td></tr></table></center>
</body>
</html>
```



La acționarea butonului de tip SUBMIT numit "Trimit datele", după cum se vede și în sursa aplicației, datele sunt prelucrate și afișate de aplicația pe partea server-ului "apl_s1.php". Sursa aplicației server o puteți vedea mai jos.

```

<html>
<head>
<title>Aplicatie Server</title>
</head>
<body>
<br>
<br>
<br>
<center><table border =1 width="400">
<tr><td height="50" background="../../Images/imm_s.bmp">
<center><font color= "white" size=5>Aplicatie PHP de tipul Server-Side </center>
</td></tr>
<tr><td>
<br>
<br>
<br><center>
<?php
echo 'Bine ai venit !! ' .$_POST["p_numa"];
echo '<br> Numele tau este :' .$_POST["numa"];
?>
</center>
<br>
<br>
<br>
</td></tr>
</table></center>
</body>
</html>

```

• Variabila \$_POST

Variabila \$_POST este utilizată pentru a colecta valorile variabilelor dintr-un form trimis cu metoda="post".

Aplicația anterioară utilizează metoda "post" pentru a trimite valorile variabilelor "nume" si "p_numa". Aplicația pe partea server "apl_s1.php" are acces la valorile acestor variabile prin intermediul variabilei \$_POST.

Variabila \$_POST este de fapt un tablou care conține numele variabilelor si valoarea acestora, trimise prin metoda "post" de către HTTP. Astfel pentru a afla numele trimis s-a utilizat \$_POST["numa"], iar pentru prenume \$_POST["p_numa"].

Informațiile trimise prin metoda "post" sunt invizibile pentru utilizatori. Nu exista limitare in ceea ce privește dimensiunea informațiilor trimise.

- **Variabila \$_GET**

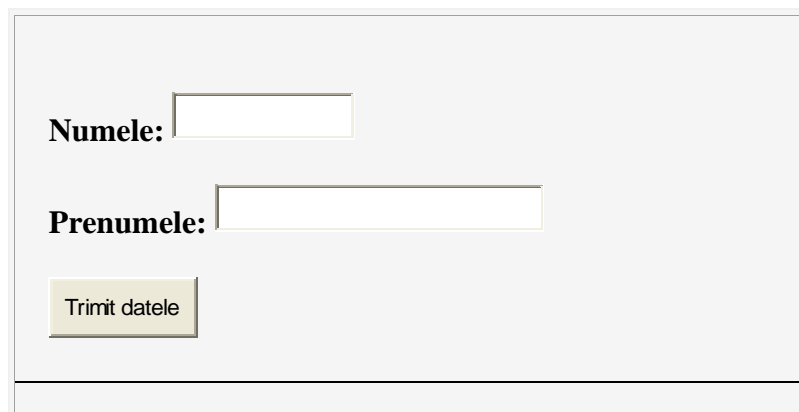
Variabila \$_GET este utilizată pentru a colecta valorile variabilelor dintr-un form trimis cu metoda="get".

Variabila \$_GET similar cu variabila \$_POST este un tablou care conține numele variabilelor și valoarea acestora, trimise prin metoda "get" de către HTTP.

Informațiile trimise prin metoda "post" sunt în acest caz invizibile pentru utilizatori, fiind afișate în cadrul URL-ului iar dimensiunea informației transmise este limitată la 100 de caractere.

Aplicația următoare utilizează metoda "get" pentru a trimite valorile variabilelor "nume" și "p_nume". Aplicația pe partea server "apl_s2.php" are acces la valorile acestor variabile prin intermediul variabilei \$_GET. Astfel pentru a afla numele trimis s-a utilizat \$_GET["nume"], iar pentru prenume \$_GET["p_nume"].

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<form action="apl_s2.php" method="get"><br>
<p> <b>Numele:</b> <input type="text" name="nume" size="15" maxlength="15" value=" "
/></p>
<p> <b>Prenumele:</b> <input type="text" name="p_nume" size="30" maxlength="30"
value=" " /></p>
<p> <INPUT type = SUBMIT value="Trimit datele">
</form>
</td></tr></table></center>
</body>
</html>
```

The image shows a web browser window displaying a form. The form is centered on a light gray background. It consists of two text input fields. The first field is preceded by the label 'Numele:' and the second by 'Prenumele:'. Below these fields is a button with the text 'Trimit datele'. The entire form is enclosed in a thin black border.

La acționarea butonului de tip SUBMIT numit "Trimit datele", se trimite datele spre aplicația server "apl_s2.php".

Pentru a putea vedea informațiile transmise prin metoda "get" va trebui să deschidem aplicația într-o nouă fereastră.

Datele sunt deci, prelucrate și afișate de aplicația pe partea server-ului "apl_s2.php". Sursa aplicației server o puteți vedea mai jos.

```

<html>
<head>
<title>Aplicatie Server</title>
</head>
<body>
<br>
<br>
<br>
<center><table border =1 width="400">
<tr><td height="50" background="../../Images/imm_s.bmp">
<center>
<font color= "white" size=5>Aplicatie PHP de tipul Server-Side </font>
<br><font color= "white" size=3>Se utilizeaza metoda <b> get </b> </font>
</center>
</td></tr>
<tr><td>
<br>
<br>
<br><center>
<?php
echo 'Bine ai venit !! ' .$_GET["p_nume"];
echo '<br> Numele tau este :' .$_GET["nume"];
?>
</center>
<br>
<br>
<br>
</td></tr>
</table></center>
</body>
</html>

```

Utilizarea PHP pentru realizarea paginilor WEB dinamice

In PHP se utilizează o serie de tehnici pentru realizarea paginilor WEB dinamice.

- Cele mai importante metode pentru realizarea paginilor web dinamice sunt:
 - utilizarea și încorporarea fișierelor externe
 - manipularea formularelor și prelucrarea datelor conținute de acestea
 - utilizarea funcțiilor proprii
 - utilizarea funcțiilor speciale

- **Utilizarea și incorporarea fișierelor externe**

În majoritatea paginilor web se pot distinge mai multe zone relativ fixe cum ar fi:

- zona antet
- zona meniu
- conținut principal
- baza paginii

Paginile web dinamice păstrează zonele enumerate mai sus fixe, mai puțin zona de conținut principal. Pentru realizarea conținutului celorlalte zone este utilă definirea lor o singură dată în fișiere externe și includerea acestor fișiere de fiecare dată când se lansează o pagină.

Facilitățile SSI (Server Side Includes) sunt utilizate pentru a crea deci headers, footers, sau elemente ce se includ în mai multe pagini.

Fișierele externe se includ în pagina curentă prin utilizarea funcției "include".

Forma generală pentru includerea unui fișier extern este:

```
include("nume_fis";
```

Să realizăm o pagină web dinamică utilizând php care să conțină o zona antet, o zona meniu, un conținut principal și o zonă la baza paginii. Zona de antet, zona meniu, și zona de la baza paginii vor fi descrise în fișiere separate și vor fi incluse în pagina web.


Utilizarea limbajului Php - realizarea paginilor dinamice

[Home](#) | [Nume_prenume](#) | [Calculator_tva](#) | [Calculator_p](#) |

Utilizarea PHP pentru realizarea paginilor WEB dinamice

- Aplicațiile web dinamice sunt realizate prin:
 - o utilizarea și incorporarea fișierelor externe
 - o manipularea formularelor și prelucrarea datelor conținute de acestea
 - o utilizarea funcțiilor proprii
 - o utilizarea funcțiilor speciale

Succes în realizarea paginilor dinamice !

 UNIVERSITATEA
Petru Maior
TARGU-MURES

© Autor: lector univ.dr. Traian Turc 2009

Programul principal "pag_din_1.php" conține includerea fișierelor externe, după cum se poate vedea mai jos.

```
<?php # Script : pag_din_1.php
$page_title = 'Pagina web dinamica';
include ('./includes/header.php');
echo '<body bgcolor="8fbc8f">';
// Includere fisier meniu.php.
include ('./includes/meniu.php');
echo '
<div id="Continut_p">
<br><br>
<center><table border =1 width="80%">
<tr><td height="50" background=".../Images/imm_s.bmp">
<center><font color="white" font size="3">
<b> Utilizarea PHP pentru realizarea paginilor WEB dinamice</b>
</center>
</td></tr><tr><td>
</td></tr></table>
</div>
<div id="Baza_pag">';
include ('./includes/baza_pag.html');
echo '</div>';
?>
```

Programul principal "pag_din_1.php" include in primul rând antetul paginii prin instrucțiunea:
include ('./includes/header.php');
Conținutul fișierului "header.php" este:

```

<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <title>Pagini web dinamice</title>
<style type="text/css" media="screen">@import "../includes/layout.css";</style>
</HEAD>
<body>
<div id="Header">Utilizarea limbajului Php - realizarea paginilor dinamice</div>
<div id="Continut">
<!------->
<!-- End of Header -->

```

Se observă că fișierul header.php include la rândul lui un fișier CSS după care scrie un text în div-ul Header și deschide div-ul Conținut în care vor fi afișate meniul și conținutul principal.

Fișierul CSS "layout.css" definește div-urile și elementele de design, conținutul fișierului CSS "layout.css" fiind:

```

body {
    margin:0px;
    padding:0px;
    font-family:verdana, arial, helvetica, sans-serif;
    color:Blue;
    /*background-color:AliceBlue*/
    background-color:WhiteSmoke
}
h1 {
    margin:0px 0px 15px 0px;
    padding:0px;
    font-size:20px;
    line-height:20px;
    font-weight:800;
    color:#9370d8;
}
h2 {
    margin-bottom:.0001pt;
    text-align:justify;
    text-justify:inter-ideograph;
    line-height:150%;
    page-break-after:avoid;
    font-size:12.0pt;
    font-family:"Book Antiqua";
    margin-left:0cm; margin-right:0cm; margin-top:0cm
}
h5 {

```



```

        margin:0px 0px 0px 0px;
        padding:0px;
        font-size:11px;
        line-height:11px;
        font-weight:400;
        color:navy;
    }
p {
    font:11px/20px verdana, arial, helvetica, sans-serif;
    margin:0px 0px 16px 0px;
    padding:0px;
}
a {
    color:#09c;
    font-size:8px;
    text-decoration:none;
    font-weight:300;
    font-family:verdana, arial, helvetica, sans-serif;
}
a:link {color: #000fff; font-size: 8pt; font-family: arial, helvetica, sans-serif; text-
decoration:none;}
a:visited {color: #0000ff; font-size: 8pt; font-family: arial, helvetica, sans-serif; text-
decoration:none;}
a:hover{color: #ff0000; font-size: 8pt; font-family: arial, helvetica, sans-serif; text-
decoration:underline;}

#Header {
    font-size:24px;
    margin:0px 0px 0px 0px;
    padding:10px 0px 0px 10px;
    height:40px;
    border-style:solid;
    border-color:black;
    border-width:1px 0px;
    line-height:11px;
    background-image: url(imm_s.bmp);
}
#Continut {
    position:absolute;
    top:45px;
    left:0px;
    right:0px;
    width:100%;
    padding:0px;
}
#Menu {
    position:absolute;
    top:0px;

```

```

        left:0px;
        width:100%;
        padding:0px;
        voice-family: "\"}\"";
        voice-family:inherit;
        width:100%;
    }
#Continut_p {
    position:relative;
    top:40px;
    left:0px;
    right:0px;
    height:400px;
    width:100%;
    padding:0px;
}
#Baza_pag {
    position:relative;
    top:10px;
    left:0px;
    right:0px;
    width:100%;
    padding:0px;
}
#Error {
    color:#FF0033;
    font-size:14px;
}

```

Programul principal "pag_din_1.php" conține de asemenea și meniul principal inclus cu **include ('./includes/meniu.php');**.

```

<div id="Meniu">
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr bgcolor="Blue">
<td>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr bgcolor="D9E3EE" Background="./includes/imm_m.bmp" >
<td Background="./includes/imm_m.bmp" class="forTexts" vliagn>
|<a href="pag_din_1.php" title="Intrare pe baza de parola">Home</a> |
<a href="num_pr.php" title="Date personale">Nume_prenume</a> |
<a href="calculator.php" title="Calculator Valoare si TVA">Calculator_tva</a> |
<a href="putere.php" title="Calculator putere activa si reactiva">Calculator_p</a> |
<font face="arial" color="#084a8d" size="1">

```

```

        </font>
    </td>
</tr>
</table>
</td>
</tr>
</table>
</div>

```

Urmează apoi in programul principal "pag_din_1.php" conținutul primei pagini și la baza este inclusa informația din zona baza_paginii cu
include ('./includes/baza_pag.html');

```

<table width="100%" cellpadding="1" cellspacing="0" align="lef" bgcolor="#999999">
  <tr><td>
    <table width="100%" cellpadding="1" cellspacing="0" align="center">
      <tr bgcolor="#d3d3d3"><td Background=" ../Images/fondj.bmp">
        <table align="center" cellpadding="1" cellspacing="1"
Background=" ../Images/fondj.bmp" >
          <tr>
            <td align="center" >
              <!--/Start upm.ro/-->
              <a href="http://www.upm.ro" target="parent" ></a>
              <!--/End upm.ro/-->
            </td><td>
              <!--/Start mail/-->
              <font color="red">
                <SMALL>© <strong><font color="red" face="Arial"size=2>Autor:</strong>
<A HREF=mailto:traian.turc@yahoo.com > lector univ.dr. Traian Turc</A> 2009</SMALL>
              <!--/End mail/-->
            </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
</td>
</tr>
</table>


```

La opțiunea "Nume_prenume" din meniul principal sa apelat form-ul "num_pr.php".

Utilizarea limbajului Php - realizarea paginilor dinamice				
Home	Nume_prenume	Calculator_tva	Calculator_p	

Numele:

Prenumele:

© Autor: lector univ.dr. Traian Turc 2009

Acest form trebuie să se încadreze în șablonul general deci trebuie să fie asemănător cu programul principal "pag_din_1.php". Diferă doar zona de conținut principal care conține efectiv form-ul care cere numele și prenumele.

```
<html>
<body>
<?php # Script : num_pr.php
$page_title = 'Numele si prenumele';
// Includere fisier header.php
include ('./includes/header.php');
echo '<body bgcolor="#8fbc8f">';
// Includere fisier meniu.php
include ('./includes/meniu.php');
echo '
<div id="Continut_p">
<br><br><br>
<center><table bgcolor="#D9E3EE" border =1 width="400"><tr><td>
```

```

<form action="num_pr_s.php" method="get"><br>
<p> <b>Numele:</b> <input type="text" name="nume" size="15" maxlength="15" value=" "
/></p>
<p> <b>Prenumele:</b> <input type="text" name="p_nume" size="30" maxlength="30"
value=" " /></p>
<p> <INPUT type = SUBMIT value="Trimit datele">
</form>
</td></tr></table></center>
</div>
<div id="Baza_pag">;
include ('./includes/baza_pag.html');
echo '</div>';
?>
</body>
</html>

```

Chiar și aplicația de partea server care prelucrează datele trimise trebuie să se încadreze în același șablon general.

```

<html>
<head>
<title>Pagina dinamica</title>
</head>
<body>
<?php # Script : num_pr_s.php
$page_title = 'Numele si prenumele';
// Includere fisier header.php
include ('./includes/header.php');
echo '<body bgcolor="8fbc8f">';
// Includere fisier meniu.php
include ('./includes/meniu.php');
echo '<div id="Continut_p">';
?>
<br>
<br>
<br>
<center><table border =1 width="400">
<tr><td height="50" background=".../Images/imm_s.bmp">
<center>
<font color= "white" size=5>Prelucrare nume si prenume </font>
<br><font color= "white" size=3>Afisare in pagina dinamica </font>
</center>
</td></tr>
<tr><td>
<br><br><br><center>

```

```

<?php
echo 'Bine ai venit !! ' . $_GET["p_ume"];
echo '<br> Numele tau este : ' . $_GET["nume"];
</center>
<br><br><br></td></tr>
</table></center>
</div>
<div id="Baza_pag">
<?php
include ('./includes/baza_pag.html');
echo '</div>';
?>
</body>
</html>

```

• Manipularea formularelor si prelucrarea datelor continute de acestea

În aplicația "num_pr.php" pentru a prelucra datele trimise s-a realizat o noua aplicație care să prelucreze aceste date și să afișeze rezultatul într-o noua pagină. Ar fi de preferat să includem într-o singură aplicație atât form-ul inițial cât și prelucrarea datelor. Avantajul acestei abordări permite păstrarea datelor introduse și revenirea asupra lor în cazul în care trebuiesc reintroduse anumite câmpuri.

La opțiunea "Calculator" din meniul principal este inclusa o astfel de aplicatie care manipulează form-ul inițial si prelucrează datele trimise.

Utilizarea limbajului Php - realizarea paginilor dinamice

[Home](#)
[Nume_prenume](#)
[Calculator_tva](#)
[Calculator_p](#)

Calcularea valorii si TVA-ului

Introduceti cantitatea, pretul si cota TVA.


Calculator valoare si TVA

Cantitate:

Pret:

Cota TVA (%):

Calculează!


UNIVERSITATEA
Petru Maior
TARGU-MURES

© **Autor:** lector univ.dr. Traian Turc 2009

```

<?php # Script: - calculator.php
$page_title = 'Calculator';
// Includere fisier header.php
include ('./includes/header.php');
echo '<body bgcolor="8fbc8f">';
// Includere fisier meniu.php
include ('./includes/meniu.php');
echo '
<div id="Continut_p">';
// Se verifica daca form-ul a fost trimis
if (isset($_POST['trimis'])) {

    // Validare minimala form.
    if ( is_numeric($_POST['cantit']) && is_numeric($_POST['pret']) &&
is_numeric($_POST['cota_tva']) ) {

        // Calcularea rezultatelor.
        $tva = $_POST['cota_tva'] / 100; // Turn 5% into .05.
        $total = ($_POST['cantit'] * $_POST['pret']) * ($tva + 1);

        // Afisarea rezultatelor.
        echo '<h1>Total Cost</h1>
<p>Costul total al celor ' . $_POST['cantit'] . ' bucati, este: ' . number_format ($total, 2) . '
lei , din care TVA ' . $_POST['cota_tva'] . '% = ' . number_format
($_POST['pret']*$_POST['cantit']*$tva, 2) . ' .</p><p><br /></p>';

    } else { // S-au trimis valori invalide.
        echo '<h1>Eroare !</h1>
<p class="error">Va rog introduceti cantitatea, pretul, si cota TVA.</p><p><br
/></p>';
    }

} else {
    echo '<h1>Calcularea valorii si TVA-ului </h1>
<p class="error">Introduceti cantitatea, pretul si cota TVA.</p><p><br /></p>';
}
?>
<center><table bgcolor="#D9E3EE" border =1 width="400"><tr><td>
<h2><center>Calculator valoare si TVA</center></h2><hr>
<form action="calculator.php" method="post">
    <table align ="center"><tr><td>
        <p>Cantitate:</td><td><input type="text" name="cantit" size="5" maxlength="10"
value="<?php if (isset($_POST['cantit'])) echo $_POST['cantit']; ?>" /></p>
        </td></tr><tr><td>
        <p>Pret: </td><td><input type="text" name="pret" size="5" maxlength="10"
value="<?php if (isset($_POST['pret'])) echo $_POST['pret']; ?>" /></p>
        </td></tr><tr><td>
        <p>Cota TVA (%): </td><td><input type="text" name="cota_tva" size="5"

```

```

maxlength="10" value="<?php if (isset($_POST['cota_tva'])) echo $_POST['cota_tva']; ?>" />
</p>
</td></tr></table><hr>
<p><input type="submit" name="trimite" value="Calculeaza!" /></p>
<input type="hidden" name="trimis" value="TRUE" />
</form>
</td></tr></table></center>
</div>
<div id="Baza_pag">
<?php
include ('./includes/baza_pag.html');
echo '</div>';
?>

```

• Utilizarea funcțiilor proprii și a funcțiilor speciale

O altă modalitate de a realiza pagini dinamice, este utilizarea funcțiilor proprii și a funcțiilor speciale. Vom realiza în continuare o aplicație care calculează puterea aparentă, puterea activă și reactivă cunoscând tensiunea, curentul și defazajul dintre ele.

Utilizarea limbajului Php - realizarea paginilor dinamice			
Home	Nume_prenume	Calculator_tva	Calculator_p

Calcularea puterii active si reactive


Introduceti tensiunea, curentul si defazajul in radiani .

Calcularea puterii active si reactive

Tensiune:

Curent:

Defazaj (radiani):



© Autor: lector univ.dr. Traian Turc 2009

Aplicația folosește o funcție proprie "calc_pa" pentru calculul puterii aparente.


```

<?php # Script: - putere.php
$page_title = 'Putere activa si reactiva';
// Includere fisier header.php
include ('./includes/header.php');
echo '<body bgcolor="8fbc8f">';
// Includere fisier meniu.php
include ('./includes/meniu.php');
function calc_pa () { // Functia pentru calculul puterii aparente
    global $pa;
    $pa = ($_POST['tens'] * $_POST['crnt']);
}

echo '
<div id="Continut_p">';
// Se verifica daca form-ul a fost trimis
if (isset($_POST['trimis'])) {
    if (is_numeric($_POST['tens'])) {
        $u = $_POST['tens'];
    } else {
        echo '<font color="red" size="2">Introduceti valoarea tensiunii electrice
!</font><br>';
        $u = FALSE;
    }
    if (is_numeric($_POST['crnt'])) {
        $i = $_POST['crnt'];
    } else {
        echo '<font color="red" size="2">Introduceti valoarea curentului electric
!</font><br>';
        $i = FALSE;
    }
    if (is_numeric($_POST['defaz'])) {
        $f = $_POST['defaz'];
    } else {
        echo '<font color="red" size="2">Introduceti valoarea defazajului intre curent si
tensiune !</font><br>';
        $f = FALSE;
    }

    // Validare minimala form.
    if ($u && $i && $f) { // Au fost introduse corect toate datele

        // Calcularea rezultatelor.
        calc_pa();
        $p_a = $pa * cos($defaz);
        $p_r = $pa * sin($defaz);
        // Afisarea rezultatelor.
        echo '<h1>Puterile calculate:</h1><font size="2">
Puterea aparenta este : <font color="red">' . number_format ($pa, 2) . '</font> VA <br>

```

```

Puterea activa este :<font color="red">' . number_format ($p_a, 2) .'</font> Watt <br>
Puterea reactiva este :<font color="red">' . number_format ($p_r, 2) .'</font> VAR
</font>';
    } else { // S-au trimis valori invalide.
        echo '<p class="error">Va rog, incercati inca o data !.</p><p><br /></p>';
    }

} else {
    echo '<h1>Calcularea puterii active si reactive </h1>
    <p class="error">Introduceti tensiunea, curentul si defazajul in radiani .</p><p><br
/></p>';
}
?>
<center><table bgcolor="#D9E3EE" border =1 width="400"><tr><td>
<h2><center>Calcularea puterii active si reactive</center></h2><hr>
<form action="putere.php" method="post">
    <table align ="center"><tr><td>
        <p>Tensiune:</td><td><input type="text" name="tens" size="5" maxlength="10"
value="<?php if (isset($_POST['tens'])) echo $_POST['tens']; ?>" /></p>
    </td></tr><tr><td>
        <p>Curent: </td><td><input type="text" name="crnt" size="5" maxlength="10"
value="<?php if (isset($_POST['crnt'])) echo $_POST['crnt']; ?>" /></p>
    </td></tr><tr><td>
        <p>Defazaj (radiani): </td><td><input type="text" name="defaz" size="5"
maxlength="5" value="<?php if (isset($_POST['defaz'])) echo $_POST['defaz']; ?>" /> </p>
    </td></tr></table><hr>
    <p><input type="submit" name="trimite" value="Calculeaza!" /></p>
    <input type="hidden" name="trimis" value="TRUE" />
</form>
</td></tr></table></center>
</div>
<div id="Baza_pag">
<?php
include ('./includes/baza_pag.html');
echo '</div>';
?>

```