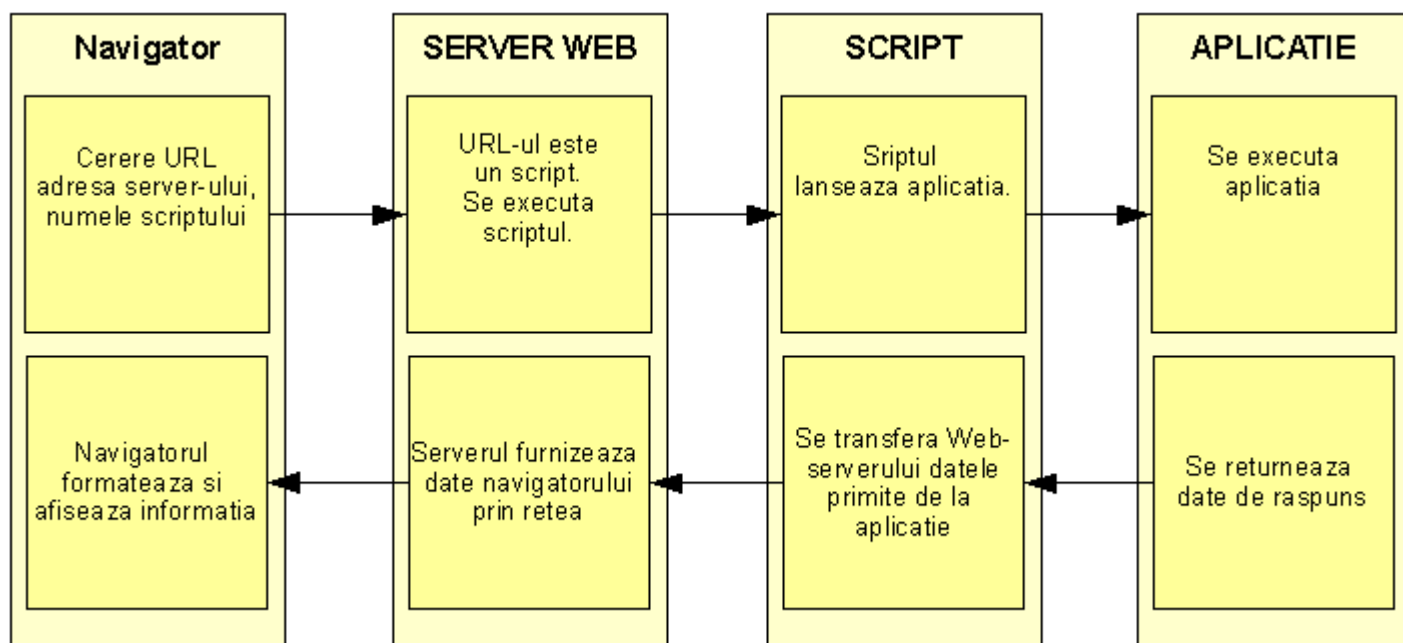


## Programarea aplicatiilor server side

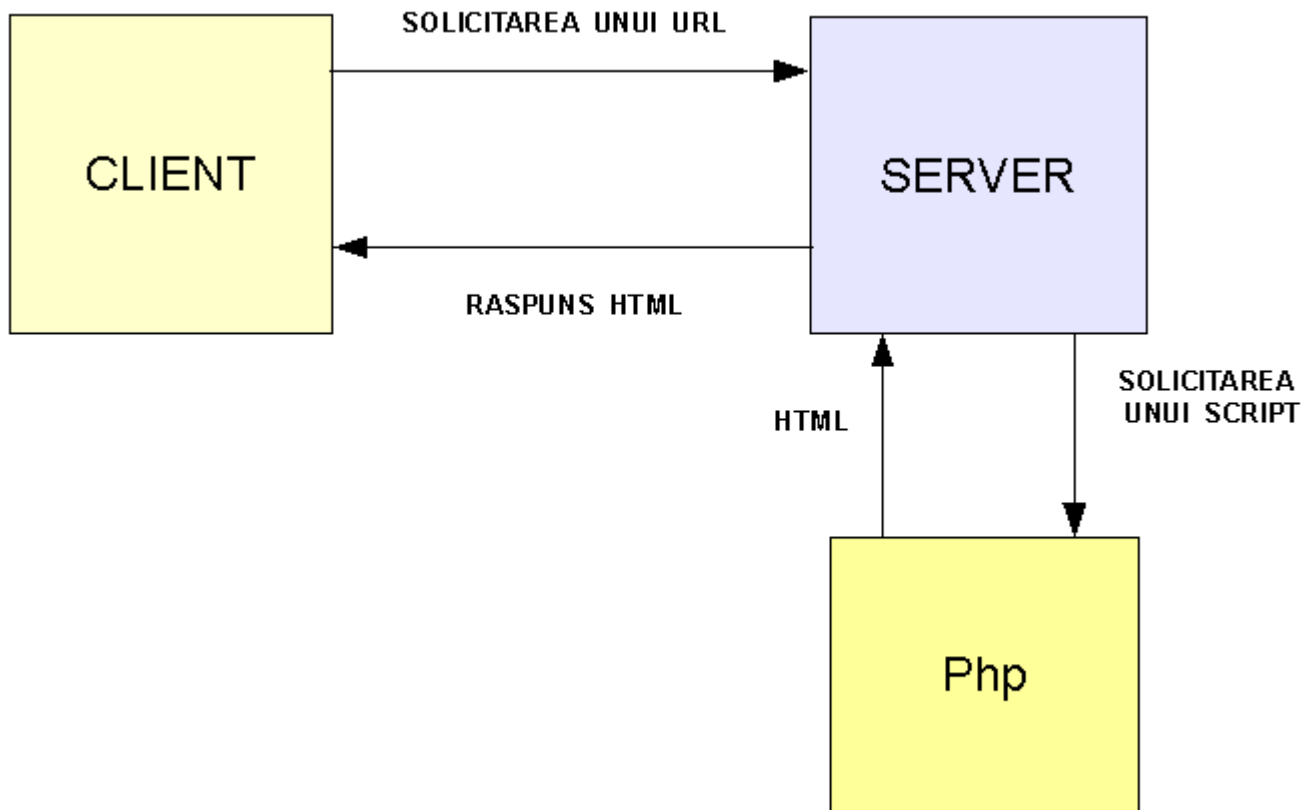
### Aplicatii server side

În momentul în care un client face o cerere sub formă 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 în care 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.

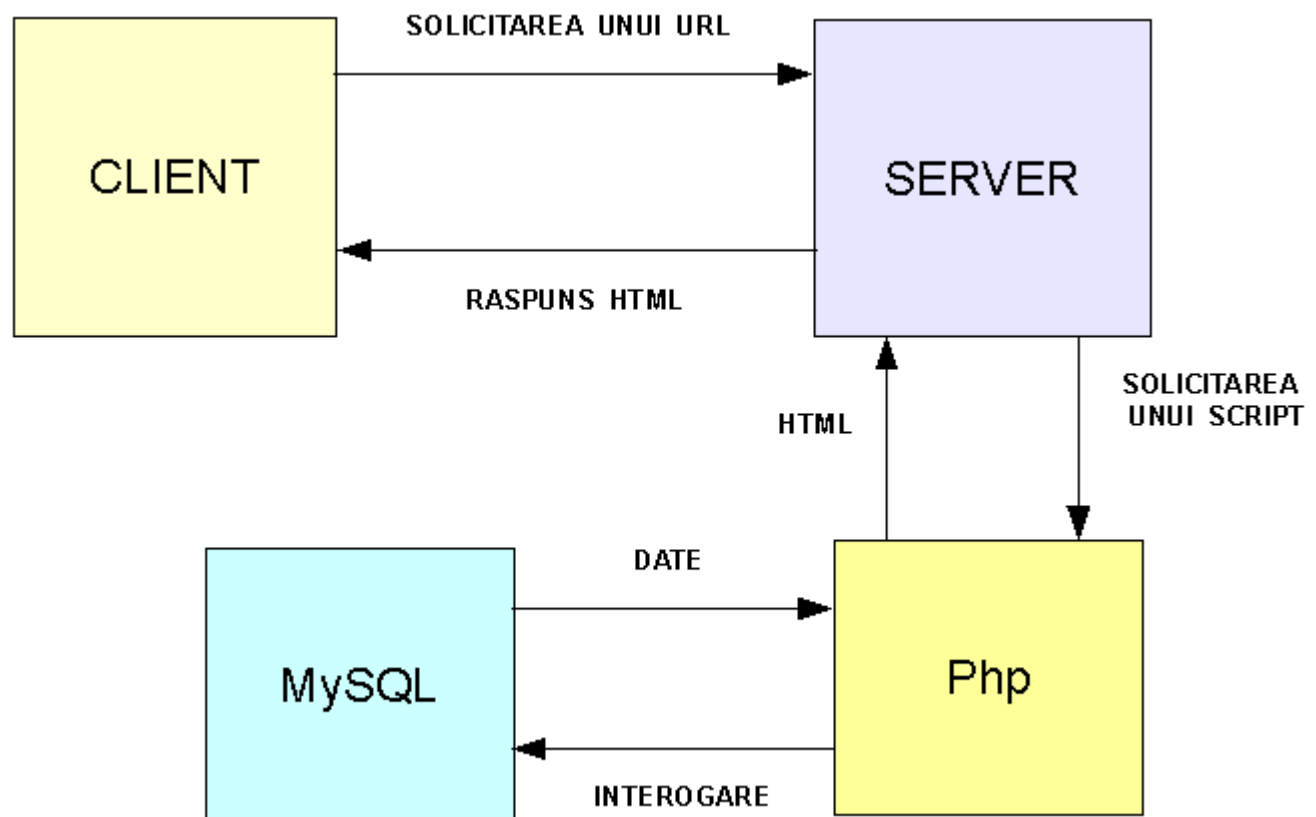


- **Aplicatii Php server side**

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



In majoritatea cazurilor, aplicatiile php apeleaza baze de date. Baza de date MySQL este cea mai utilizata baza de date accesata de aplicatiile php. In acest caz aplicatia client -server se poate reprezenta astfel:



## Programarea in Php

Php (Personal Home Page) este un limbaj de scriptare creat initial 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 inseamna astazi Hypertext Preprocessor. Este o tehnologie de scriptare interplatforma, rulant pe diverse sisteme de operare si pe diferite platforme hard. Limbajul are facilitati de operare cu diverse baze de date, fiind recomandat insa sa se lucreze cu baza de date MySQL.

- **Elemente de programare in Php**

Un limbaj de programare server side dispune pe langa un set de instructiuni de baza, de un mecanism de transmitere a datelor spre sin dinspre browser. Pentru inceput vom aborda elementele de baza ale limbajului, urmand sa completam ulterior cu mecanisme pentru comunicarea cu diverse browsere si accesul la baze de date.

- **Sintaxa de baza a limbajului**

Un script PHP incepe cu `< ? php` si se inchide cu `? >` . Scripturile PHP pot fi incluse in orice loc in cadrul unui document html. In general serverele accepta si forma prescurtata a unui script sub forma: `< ?....? >` . Un fisier PHP contine in mod normal Marcatori HTML si scripturi PHP. Dupa metoda consacrata, prima aplicatie tipareste un text de bun venit!

Urmatoarea aplicatie foloseste php pentru a afisa mesajul "Bine ati 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 ati 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, siruri 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
--

In PHP exista diferenta intre ghilimelele duble " si ghilimelele simple '. Daca in scriptul de mai sus inlocuim ghilimelele duble cu ghilimelele simple obtinem:

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$nr_cap=1;
$nume_1="limbajul PHP";
echo 'Capitolul :$nr_cap <br>Utilizarea variabilelor in :$nume_1' ;
?>
</td></tr></table></center>
</body>
</html>
```

Capitolul :\$nr_cap Utilizarea variabilelor in :\$nume_1
---

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

De multe ori este necesar sa introducem ghilimele in text. Daca dorim de exemplu sa atribuim unei variabile valoarea: UMFST " Emil Palade", va trebui sa folosim caracterul \ pentru a nu se interpreta ghilimelele din cadrul textului.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$nume_u="Universitatea \"Petru Maior\"";
echo " Studiez la :$nume_u " ;
?>
</td></tr></table></center>
</body>
</html>
```

Studiez la :Universitatea "Petru Maior"
---

In acest caz primul caracter " de dupa \ nu a fost interpretat.

## • Siruri de caractere

In PHP se definesc o serie de functii pe siruri de caractere, foarte utile tinand cont ca PHP este un limbaj de scriptare si trebuie sa gestioneze continut de pagini web. Printre cele mai importante functii se numara:

- concatenarea sirurilor
- determinarea lungimii unui sir
- cautarea unui subsir in cadrul unui sir
- Compararea a doua siruri

## Concatenarea sirurilor

Pentru concatenarea a doua siruri, se foloseste operatorul punct (.)

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$fc="Facultatea de Inginerie";
$sc="Sectia-Calculatoare";
$txt= 'Studiez la: '.$fc .'<br>'.$sc ;
echo $txt;
?>
</td></tr></table></center>
</body>
</html>
```

Studiez la: Facultatea de Inginerie Sectia-Calculatoare
--

## Determinarea lungimii unui sir

Pentru determinarea lungimii unui sir se foloseste functia: **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
---

## Cautarea unui subsir in cadrul unui sir

Pentru cautarea unui subsir in cadrul unui sir, se foloseste functia: **strpos()**

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
$fc="Facultatea de Inginerie";
echo 'Subsirul "Inginerie", se gaseste in sirul: "'. $fc . "'", incepand cu pozitia:'. strpos($fc ,
"Inginerie");
?>
</td></tr></table></center>
</body>
</html>
```

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

## Compararea a doua siruri

Pentru a compara doua siruri intre ele , se foloseste functia: **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 siruri egale.

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

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

## • Operatori utilizati in PHP

Operatorii utilizati in PHP sunt similari cu operatorii utilizati in Java Scripts 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

## • Instructiuni 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 instructiuni decizionale pentru a afisa un mesaj in functie de ora curenta.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
//$an=idate("Y");
date_default_timezone_set('Europe/Bucharest');
$an_c=date("Y");
if ($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: 2021 este an ne-bisect

Sintaxa instructiunii 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 instructiunea decizionala elseif pentru a afisa un salut in functie de ziua din saptamana.

```
<html>
<body>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<?php
date_default_timezone_set('Europe/Bucharest');
$d=date("D");
if ($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!

## • Instructiuni de ciclare

In PHP sunt definite patru instructiuni pentru realizarea buclelor repetitive de programare.

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

### Sintaxa instructiunii while:

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

Utilizand instructiunea de buclare **while** vom afisa primele 10 de numere naturale. Fiecare numar va fi afisat intr-o celula separata a unui tabel.

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

Primele 10 de numere naturale									
1	2	3	4	5	6	7	8	9	10

### Sintaxa instructiunii do..while:

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

Utilizand instructiunea de buclare **do...while** vom afisa 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 instructiunii for:

```

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

```

Reluam aplicatia anterioara insa vom folosi instructiunea de buclare **for** pentru afisa patratele 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 instructiunii foreach :

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

Urmatoarea aplicatie utilizeaza instructiunea de buclare **foreach** printru a afisa zilele saptamanii.

```

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

```

Zilele sapamanii sunt:

Ziua 1 : Luni  
 Ziua 2 : Marti  
 Ziua 3 : Miercuri  
 Ziua 4 : Joi  
 Ziua 5 : Vineri  
 Ziua 6 : Sambata  
 Ziua 7 : Duminica

## • Utilizarea functiilor in php

Vom utiliza in continuare functii in cardul sript-urilor php. Urmatoarea aplicatie defineste si utilizeaza functia patrat pentru a calcula patratul unui numar.

```

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

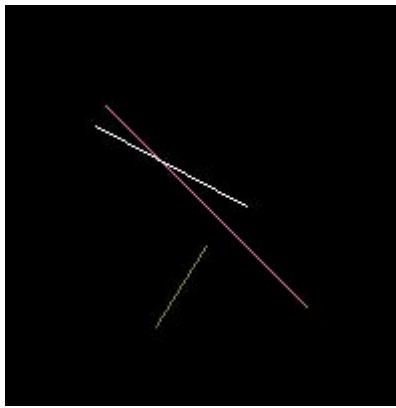
```

Patratul numarului 25este :625

## • Grafica in php

### - trasare linii

Vom realiza in continuare aplicatii grafice in php.Prima aplicatie traseaza 3 linii de culori diferite,



```
<?php

// Creare imagine 200 x 200
$canvas = imagecreatetruecolor(200, 200);

// definire culori
$pink = imagecolorallocate($canvas, 255, 105, 180);
$white = imagecolorallocate($canvas, 255, 255, 255);
$green = imagecolorallocate($canvas, 132, 135, 28);

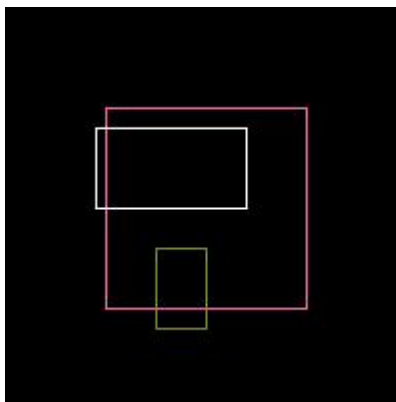
// trasare linii
imageline($canvas, 50, 50, 150, 150, $pink);
imageline($canvas, 45, 60, 120, 100, $white);
imageline($canvas, 100, 120, 75, 160, $green);

// Output and free from memory
header('Content-Type: image/jpeg');
imagejpeg($canvas);
imagedestroy($canvas);

?>
```

### - trasare dreptunghiuri

Urmatoarea aplicatie traseaza 3 dreptunghiuri de culori diferite,



```
<?php

// Creare imagine 200 x 200
$canvas = imagecreatetruecolor(200, 200);

// definire culori
$pink = imagecolorallocate($canvas, 255, 105, 180);
$white = imagecolorallocate($canvas, 255, 255, 255);
$green = imagecolorallocate($canvas, 132, 135, 28);

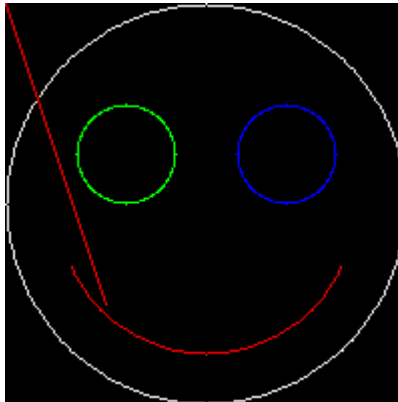
// trasare dreptunghiuri
imagerectangle($canvas, 50, 50, 150, 150, $pink);
imagerectangle($canvas, 45, 60, 120, 100, $white);
imagerectangle($canvas, 100, 120, 75, 160, $green);

// Output and free from memory
header('Content-Type: image/jpeg');
imagejpeg($canvas);
imagedestroy($canvas);

?>
```

### - trasare cercuri

Aplicatia traseaza arcuri cercuri si linii de culori diferite,





```
<?php
// creare imagine 200*200
$img = imagecreatetruecolor(200, 200);

// definire culori
$white = imagecolorallocate($img, 255, 255, 255);
$red   = imagecolorallocate($img, 255, 0, 0);
$green = imagecolorallocate($img, 0, 255, 0);
$blue  = imagecolorallocate($img, 0, 0, 255);

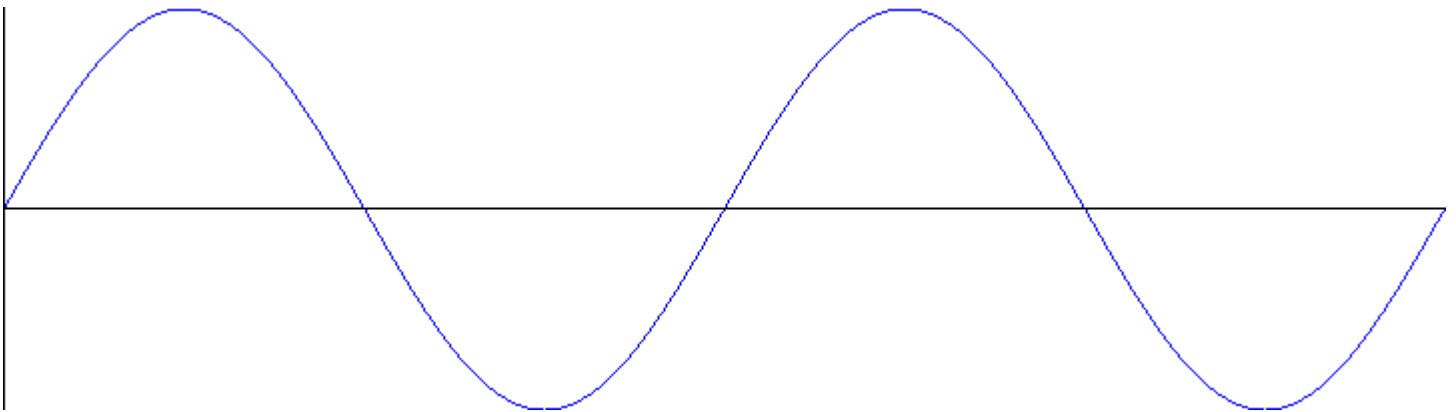
// trasare arcuri de cerc
imagearc($img, 100, 100, 200, 200, 0, 360, $white);
imagearc($img, 100, 100, 150, 150, 25, 155, $red);
imagearc($img, 60, 75, 50, 50, 0, 360, $green);
imagearc($img, 140, 75, 50, 50, 0, 360, $blue);
//trasare linie
imageline($img, 0, 0, 50, 150, $red);
// output image in the browser
header("Content-type: image/png");
imagepng($img);

// free memory
imagedestroy($img);

?>
```

### - trasare sinus

Aplicatia traseaza graficul functiei sinus,



```
<?php

Header("Content-type: image/png");

//creare obiect im

$im_Width=360*2;
$im_Height=200;
$im=ImageCreate($im_Width+1,$im_Height+1);

// Creare culori

$cWhite=ImageColorAllocate($im,255,255,255);
$cBlue=ImageColorAllocate($im,0,0,255);
$cBlack=ImageColorAllocate($im,0,0,0);

// creare fundal alb

ImageFilledRectangle($im,0,0,$im_Width+1,$im_Height+1,$cWhite);

// trasare sinus

$xv=0;
$yv=$im_Height/2;
for($pt=0;$pt<$im_Width;$pt++){
    $x=$xv+1;
    $y=($im_Height/2)+(1-sin(deg2rad($x))*($im_Height/2));
    ImageLine($im,$xv,$yv,$x,$y,$cBlue);
    $xv=$x;
    $yv=$y;
}

// trasare axe

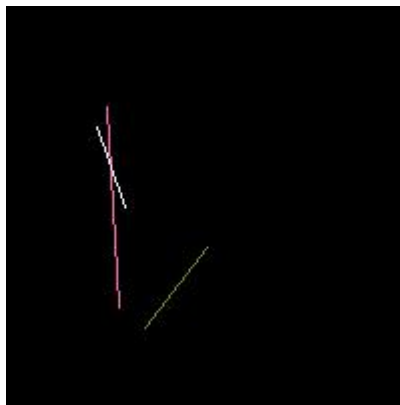
ImageLine($im,0,0,0,$im_Height,$cBlack);
ImageLine($im,0,$im_Height/2,$im_Width,$im_Height/2,$cBlack);

// creare imagine

ImagePNG($im);
ImageDestroy($im);

?>
```

### - Imagini dinamice



Start

Se realizeaza aplicatia "linii\_d.php" de forma:

```
<?php

// Creare imagine 200 x 200
$canvas = imagecreatetruecolor(200, 200);

// definire culori
$pink = imagecolorallocate($canvas, 255, 105, 180);
$white = imagecolorallocate($canvas, 255, 255, 255);
$green = imagecolorallocate($canvas, 132, 135, 28);

// trasare linii
imageline($canvas, 50, 50, rand(0,150), 150, $pink);
imageline($canvas, 45, 60, rand(0,120), 100, $white);
imageline($canvas, 100, 120, rand(0,75), 160, $green);

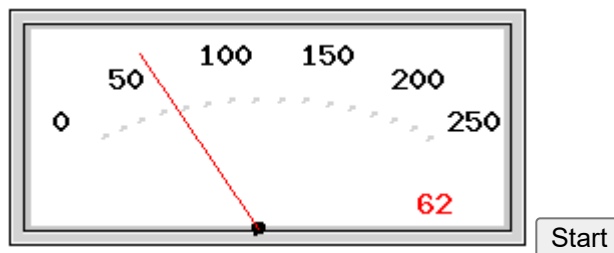
// Output and free from memory
header('Content-Type: image/jpeg');
imagejpeg($canvas);
imagedestroy($canvas);

?>
```

Dupa realizarea aplicatiei php "linii\_d.php", pentru apelare dinamica, se realizeaza urmatorul script:

```
<script>
function afis_im(){
    im.src="linii_d.php"+"?sid="+Math.random();
    setTimeout("afis_im()",300);
}
</script>
<div>
    <center>
        <INPUT onclick=afis_im() type=button value=Start>
    </center>
</div>
```

### - Instrument virtual



Se realizeaza aplicatia php "voltm.php".

```
<?php
date_default_timezone_set('Europe/Helsinki');
$dimx=250;
$dimy=120;
```

```

pimmy-120,
$alfa=20;
$alfa_s=$alfa+180;
$val_max=250;
$nr_n=$val_max/5;
$nr_d=$nr_n/4;
$img = imagecreatetruecolor($dimx+10, $dimy);
$white = imagecolorallocate($img, 255, 255, 255);
$red = imagecolorallocate($img, 255, 0, 0);
$black = imagecolorallocate($img, 0, 0, 0);
$grey = imagecolorallocate($img, 211, 211, 211);
$l_grey = imagecolorallocate($img, 200, 200, 200);

imagefill($img, 0, 0, $white);

// desen rama

imagesetthickness($img, 1);
imagerectangle($img, 1, 1, $dimx+8, $dimy-2, $black);
imagesetthickness($img, 4);
imagerectangle($img, 5, 5, $dimx+4, $dimy-6, $grey);
imagesetthickness($img, 1);
imagerectangle($img, 8, 8, $dimx+1, $dimy-9, $black);
imagesetthickness($img, 2);
imagerectangle($img, 10, 10, $dimx-1, $dimy-11, $l_grey);

//imagearc($img, $dimx/2, $dimy/2, $dimx-30, $dimy-30, $alfa_s, 180-$alfa_s, $black);
imagefilledarc($img, $dimx/2, $dimy-10, 7, 7, 0, 0, $black, IMG_ARC_PIE);

for ($zz = 0; $zz <= $val_max; $zz++) {
    $digitCoords['x'][] = ($dimx-30)/2.1 * cos(deg2rad(((180+$alfa)+(180-
2*$alfa)*$zz/$val_max))) + ($dimx-20)/2;
    $digitCoords['y'][] = ($dimy-10)/2.1 * sin(deg2rad(((180+$alfa)+(180-
2*$alfa)*$zz/$val_max))) + ($dimy-10)/1.5;
}

for ($zz = 0; $zz <= $val_max; $zz++) {
    if ($zz % $nr_n == 0)
        imagestring($img, 5, 7+$digitCoords['x'][$zz], $digitCoords['y'][$zz] - 6, $zz, $black);
    if (($zz % $nr_d == 0) && ($zz > 2*$nr_d) && ($zz < $val_max - 2*$nr_d))
        imagefilledarc($img, 15+$digitCoords['x'][$zz], 25+$digitCoords['y'][$zz], 3, 3, 0, 0,
$grey, IMG_ARC_PIE);
}
/*
$r_x = $dimx/2.3;
$r_y = $dimy/2.3;

for ($zz = 0; $zz <= $val_max; $zz += $nr_d) {
    $x_val = $r_x * cos(deg2rad(((170+$alfa)+(200-2*$alfa)* $zz/$val_max))) + $dimx/2;
    $y_val = $r_y * sin(deg2rad(((170+$alfa)+(200-2*$alfa)* $zz/$val_max))) + $dimy/2;
    imagesetthickness($img, 1);
    imageline($img, $dimx/2, $dimy-10, $x_val, $y_val, $black);

    $x_vl = 0.8*$r_x * cos(deg2rad(((170+$alfa)+(200-2*$alfa)* $zz/$val_max))) + $dimx/2;
    $y_vl = 0.8*$r_y * sin(deg2rad(((170+$alfa)+(200-2*$alfa)* $zz/$val_max))) + $dimy/2;
    imagesetthickness($img, 8);
    imageline($img, $dimx/2, $dimy-10, $x_vl, $y_vl, $white);
}
*/
$val = rand(0, $val_max);

//$val = 0;
//$val = $val_max;

$r_x = $dimx/2.5;

```

```

$r_y = $dimy/2.5;
$x_val = $r_x * cos(deg2rad(((170+$alfa)+(200-2*$alfa)*$val/$val_max))) + $dimx/1.9;
$y_val = $r_y * sin(deg2rad(((170+$alfa)+(200-2*$alfa)*$val/$val_max))) + $dimy/2;
imagesetthickness($img, 1);
imageline($img, $dimx/2, $dimy-10, $x_val, $y_val, $red);
imagestring($img,5,$dimx-45,$dimy-30, $val, $red);

header("Content-type: image/png");
imagepng($img);

imagedestroy($img);

?>

```

Pentru apelare dinamica, se realizeaza urmatorul script:

```

<script>
function afis_instr1(){
    instrum1.src="voltm.php"+"?sid="+Math.random();
    setTimeout("afis_instr1()",1000);
}
</script>
<div>
    <center>
        <INPUT onclick=afis_instr1() type=button value=Start>
    </center>
</div>

```

## Formuri in php

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

- Utilizarea formurilor**

Vom defini 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="Trimite datele">
</form>
</td></tr></table></center>
</body>
</html>

```

<b>Numele:</b> <input type="text"/>
<b>Prenumele:</b> <input type="text"/>
<input type="button" value="Trimite datele"/>

La actionarea butonului de tip SUBMITT numit "Trimite datele", dupa cum se vede si in sursa aplicatiei, datele sunt prelucrate si afisate de aplicatia pe partea server-ului "apl\_s1.php". Sursa aplicatiei server o puteti 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 utilizata pentru a colecta valorile variabilelor dintr-un form trimis cu metoda="post".

Aplicatia anterioara utilizeaza metoda "post" pentru a trimite valorile variabilelor "numa" si "p\_numa". Aplicatia pe partea server "apl\_s1.php" are acces la valorile acestor variabile prin intermediul variabilei \$\_POST.

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

Informatiile trimise prin metoda "post" sunt invizibile pentru utilizatori. Nu exista limitare in ceea ce priveste dimensiunea informatiilor trimise.

## • Variabila \$\_GET

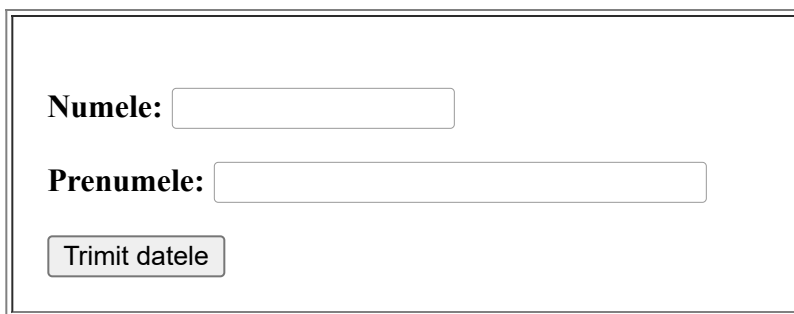
Variabila \$\_GET este utilizata pentru a colecta valorile variabilelor dintr-un form trimis cu metoda="get".

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

Informatiile trimise prin metoda "post" sunt in acest caz invizibile pentru utilizatori, fiind afisate in cadrul URL-ului iar dimensiunea informatiei transmise este limitata la 100 de caractere.

Aplicatia urmatoare utilizeaza metoda "get" pentru a trimite valorile variabilelor "nume" si "p\_nume". Aplicatia 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>
```

A screenshot of a web browser displaying a form. The form is enclosed in a double-bordered box. It contains two text input fields. The first field is labeled "Numele:" and has a width of approximately 150 pixels. The second field is labeled "Prenumele:" and has a width of approximately 300 pixels. Below these fields is a button labeled "Trimit datele".

La actionarea butonului de tip SUBMITT numit "Trimit datele", se trimit datele spre aplicatia server "apl\_s2.php".

Pentru a putea vedea informatiile transmise prin metoda "get" va trebui sa deschidem aplicatia intr-o noua fereastră. [Vezi aplicatia intr-o fereastră noua](#).

Datele sunt deci, prelucrate si afisate de aplicatia pe partea server-ului "apl\_s2.php". Sursa aplicatiei server o puteti 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 utilizeaza o serie de tehnici pentru realizarea paginilor WEB dinamice.

- Cele mai importante metode pentru realizarea paginilor web dinamice sunt:
  - utilizarea si incorporarea fisierelor externe
  - manipularea formularelor si prelucrarea datelor continute de acestea
  - utilizarea functiilor proprii
  - utilizarea functiilor speciale

- **Utilizarea si incorporarea fisierelor externe**

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

- zona antet
- zona meniu
- continut principal
- baza paginii



Paginile web dinamice pastreaza zonele enumerate mai sus fixe , mai putin zona de continut principal. Pentru realizarea continutului celorlalte zone este utila definirea lor o singura data in fisiere externe si includerea acestor fisiere de fiecare data cand se lanseaza o pagina.

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

Fisierele externe se includ in pagina curenta prin utilizarea functiei "include".

Forma generala pentru includerea unui fisier extern este:

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

Sa realizam o pagina web dinamica utilizand php care sa contina o zona antet, o zona meniu, un continut principal si o zona la baza paginii. Zona de antet, zona meniu, si zona de la baza paginii vor fi descrise in fisiere separate si vor fi incluse in pagina web.

### Utilizarea limbajului Php - realizarea paginilor dinamice

[Home](#) | [Nume\\_prenume](#) | [Calculator\\_tva](#) | [Calculator\\_p](#) |

### Utilizarea PHP pentru realizarea paginilor WEB dinamice

- Aplicatiile web dinamice sunt realizate prin:
  - utilizarea si incorporarea fisierelor externe
  - manipularea formularelor si prelucrarea datelor continute de acestea
  - utilizarea functiilor proprii
  - utilizarea functiilor speciale

Succes in realizarea paginilor dinamice !



© Autor: [lector univ.dr. Traian Turc](#) 2009

Programul principal "pag\_din\_1.php" contine includerea fisierelor externe, dupa 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>
</h1><font color="blue" font size="2">
</b><br><br>
<ul>
<li>Aplicatiile web dinamice sunt realizate prin:<br><br>
<ul>
<li> utilizarea si incorporarea fisierelor externe
<li> manipularea formularelor si prelucrarea datelor continute de acestea
<li> utilizarea functiilor proprii
<li> utilizarea functiilor speciale
</ul>
</ul>
<br><br><font color="red" font size="2"> Succes in realizarea paginilor dinamice !<br><br>
</td></tr></table>
</div>
<div id="Baza_pag">';
include ('./includes/baza_pag.html');
echo '</div>';
?>

```

Programul principal "pag\_din\_1.php" include in primul rand antetul paginii prin instructiunea: **include ('./includes/header.php');**

Continutul fisierului "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 observa ca fisierul header.php include la randul lui un fisier CSS dupa care scrie un text in div-ul Header si deschide div-ul Continut in care vor fi afisate meniul si continutul principal.

Fisierului CSS "layout.css" defineste div-urile si elementele de design, continutul fisierului 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;
    }
    #Meniu {
        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" contine de asemenea si 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" cellpadding="0" cellspacing="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>

```

Urmeaza apoi in programul principal "pag\_din\_1.php" continutul primei pagini si la baza este inclusa informatia zin 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>

```

La optiunea "Nume\_prenume" din meniul principal sa apelat formul "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 sa se incadreze in sablonul general deci trebuie sa fie asemanator 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_num" size="30" maxlength="30" value=" " />
</p>
  <p>    <INPUT type = SUBMIT value="Trimite datele">
</form>
</td></tr></table></center>
</div>
<div id="Baza_pag">';
include ('./includes/baza_pag.html');
echo '</div>';
?>
</body>
</html>
```

Chiar si aplicatia de partea server care prelucreaza datele trimise trebuie sa se incadreze in acelasi sablon general.

```
<html>
<head>
<title>Pagina dinamica</title>
</head>
<body>
<?php # Script : num_pr_s.php
$page_title = 'Numele si prenumele';
// Includere fisier header.php
include ('./includes/header.php');
echo '<body bgcolor="8fbc8f">';
// Includere fisier meniu.php
include ('./includes/meniu.php');
echo '<div id="Continut_p">';
?>
<br>
<br>
<br>
<center><table border =1 width="400">
<tr><td height="50" background="../../Images/imm_s.bmp">
<center>
<font color= "white" size=5>Prelucreare nume si prenume </font>
<br><font color= "white" size=3>Afisare in pagina dinamica </font>
</center>
</td></tr>
<tr><td>
<br>
<br>
<br><center>
<?php
echo 'Bine ai venit !! ' .$_GET["p_nume"];
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 prelucrearea datelor continute de acestea**

In aplicatia "num\_pr.php" pentru a prelucrea datele trimise s-a realizat o noua aplicatie care sa prelucreze aceste date si sa afiseze rezultatul intr-o noua pagina. Ar fi de preferat sa includem intr-o singura aplicatie atat form-ul initial cat si prelucrearea datelor. Avantajul acestei abordari permite pastrarea datelor introduse si revenirea asupra lor in cazul in care trebuiesc reintroduse anumite campuri.

La optiunea "Calculator" din meniul principal este inclusa o astfel de aplicatie care manipuleaza form-ul initial si prelucreaza datele trimise.



# Calcularea valorii si TVA-ului

Introduceti cantitatea, pretul si cota TVA.

Calculator valoare si TVA	
Cantitate:	<input type="text"/>
Pret:	<input type="text"/>
Cota TVA (%):	<input type="text"/>
<input type="button" value="Calculeaza!"/>	



© 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 mimimala 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 functiilor proprii si a functiilor speciale**

O alta modalitate de a realiza pagini dinamice, este utilizarea functiilor proprii si a functiilor speciale. Vom realiza in continuare o aplicatie care calculeaza puterea aparenta, puterea activa si reactiva cunoscand tensiunea, curentul si 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

Aplicatia foloseste o functie proprie "calc\_pa" pentru calculul puterii aparente.

```
<?php # Script: - putere.php
$page_title = 'Putere activa si reactiva';
// Includere fisier header.php
include ('./includes/header.php');
echo '<body bgcolor="8fbc8f">';
// Includere fisier meniu.php
include ('./includes/meniu.php');
function calc_pa () { // Functia pentru calculul puterii aparente
    Return ($_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.
        $pa=calc_pa(); // putere aparenta
        $p_a = $pa * cos($f); // putere activa
        $p_r = $pa * sin($f); // putere reactiva
        // 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>':

```

- **Utilizarea tehnologiei AJAX**

Vom utiliza in continuare tehnologia AJAX pentru a realiza pagini dinamice. Vom realiza o aplicatie care sugereaza nume prememorate in cazul in care se completeaza un camp cu numele sau prenumele unei persoane.

Utilizarea tehnologiei AJAX	
Nume:	<input type="text"/>
Sugestie:	

Aplicatia este asemanatoare cu aplicatiile scrise in Java-Script in care se foloseste tehnologia AJAX. Diferenta este ca URL-ul va fi de data aceasta o aplicatie php care raspunde in functie de parametrii trimisi in URL.

```
<html>
<body>
<script>
var xmlhttp
function showHint(str)
{
if (str.length==0)
{
document.getElementById("txtHint").innerHTML="";
return;
}
xmlhttp=GetXmlHttpRequestObject();
if (xmlhttp==null)
{
alert ("Browser-ul nu suporta XMLHTTP!");
return;
}
var url="sugestie.php";
url=url+"?q="+str;
url=url+"&sid="+Math.random();
xmlhttp.onreadystatechange=stateChanged;
xmlhttp.open("GET",url,true);
xmlhttp.send(null);
}

function stateChanged()
{
if (xmlhttp.readyState==4)
{
document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
}
}

function GetXmlHttpRequestObject()
{
if (window.XMLHttpRequest)
{
// code for IE7+, Firefox, Chrome, Opera, Safari
return new XMLHttpRequest();
}
if (window.ActiveXObject)
{
// code for IE6, IE5
return new ActiveXObject("Microsoft.XMLHTTP");
}
return null;
}
</script>
<form>
<center><table bgcolor="WhiteSmoke" border =1 width="400"><tr><td>
<center>Utilizarea tehnologiei AJAX </center></td></tr>
<tr><td align="center">
Nume: <input type="text" id="txt1" onkeyup="showHint(this.value)" />
</td></tr></table>
<p>Sugestie: <span id="txtHint"></span></p>
</td></tr></table>
</body>
</html>
```

Dupa cum se poate observa URL-ul transmis metodelor xmlhttp este de forma **url=url+"?q="+str;**, initial URL-ul fiind definit : **var url="sugestie.php";**

Continutul fisierului "sugestie.php" fiind:

```
<?php
// Setarea variabilelor cu diverse nume
$a[]="Andrea";
$a[]="Anca";
$a[]="Bogdan";
$a[]="Bianca";
$a[]="Camelia";
$a[]="Diana";
$a[]="Eugenia";
$a[]="Florin";
$a[]="Flavia";
$a[]="Gabriela";
$a[]="Horatiu";
$a[]="Ionel";
$a[]="Julia";
$a[]="Kitty";
$a[]="Liliana";
$a[]="Nicoleta";
$a[]="Otilia";
$a[]="Poliana";
$a[]="Paul";
$a[]="Anamaria";
$a[]="Raul";
$a[]="Carmen";
$a[]="Dumitru";
$a[]="Enescu";
$a[]="Emil";
$a[]="Sandu";
$a[]="Tamara";
$a[]="Marin";
$a[]="Violeta";
$a[]="Liza";
$a[]="Elena";
$a[]="Catrinel";
$a[]="Wenche";
$a[]="Victoria";

//preluarea parametrului q din URL
$q=$_GET["q"];

//preluarea sugestiilor din tablou daca lungimea lui q>0
if (strlen($q) > 0)
{
    $hint="";
    for($i=0; $i<count($a); $i++)
    {
        if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))
        {
            if ($hint=="")
            {
                $hint=$a[$i];
            }
            else
            {
                $hint=$hint." , ".$a[$i];
            }
        }
    }
}
```

```
// Trimiterea textului " nu sunt sugestii"
// sau trimiterea sugestiilor
if ($hint == "")
{
    $response=" nu sunt sugestii";
}
else
{
    $response=$hint;
}

//trimiterea raspunsului
echo $response;
?>
```

Aplicatia poate fi imbunatatita, sugestiile putand fi luate dintr-o baza de date.

- **Elemente de programare obiect in php**

Definitia generala a unei clase in php este:

```
<?php
    class nume_clasa {
        function fmembru1 (variabile) {
            instructiuni;
        }
        function fmembru2 (variabile) {
            instructiuni;
        }
        .
        .
        .
        function fmembru1 (variabile) {
            instructiuni;
        }
        function __construct( $v1, $v2,...,$vm ) {
            $this->var1 = $v1;
            $this->var2 = $v2;
            .
            .
            .
            $this->varm = $vm;
        }
    }
?>
```



```
$this->var1,$this->var2,...,$this->varn -- variabile membru
fmembru1,fmembru2,...,fmembrun -- functii membru
function __construct -- constructorul
```

Crearea unui obiect:

```
$nume_obiect= new nume_clasa(valori).
```

Invocarea unei metode:

```
$nume_obiect->fmembru(valori)
```

Vom crea in continuare clasa Instrument de forma:

```
<?php
class Instrum {
    /* Functii membru */
    function set_titlu($tit){
        $this->titlu = $tit;
    }

    function afis_titlu(){
        echo $this->titlu . "--";
    }

    function set_um($um){
        $this->u_m = $um;
    }

    function afis_um(){
        echo $this->u_m . " <br/>";
    }
}
$voltm = new Instrum;
$amp = new Instrum;
$frecv = new Instrum;

$voltm->set_titlu( "Voltmetru" );
$amp->set_titlu( "Ampermetru" );
$frecv->set_titlu( "Frecventmetru" );

$voltm->set_um( "V" );
$amp->set_um( "A" );
$frecv->set_um( "Hz" );

$voltm->afis_titlu();
$voltm->afis_um();
$amp->afis_titlu();
$amp->afis_um();
$frecv->afis_titlu();
$frecv->afis_um();

?>
```

In urma rularii scriptului de sus, rezulta:

Voltmetru--V  
Ampermetru--A  
Frecventmetru--Hz

Vom defini o noua clasa "Instrumm" in care vom folosi constructor

```
<?php
class Instrumm {

    /* Constructor */

    function __construct($tit,$um){

        /* Variable membru */

        $this->titlu = $tit;
        $this->u_m = $um;
    }
    /* Functie membru */
    function set_titlu($tit){
        $this->titlu = $tit;
    }

    function afis_titlu(){
        echo $this->titlu . "--";
    }

    function set_um($um){
        $this->u_m = $um;
    }

    function afis_um(){
        echo $this->u_m . " <br/>";
    }
}
$voltm = new Instrumm("Voltmetru","V");
$ampm = new Instrumm("Ampermetru","A");
$frecvm = new Instrumm("Frecventmetru","Hz");

$voltm->afis_titlu();
$voltm->afis_um();
$ampm->afis_titlu();
$ampm->afis_um();
$frecvm->afis_titlu();
$frecvm->afis_um();
?>
```

In urma rularii scriptului de sus, rezulta:

Voltmetru--V  
Ampermetru--A  
Frecventmetru--Hz

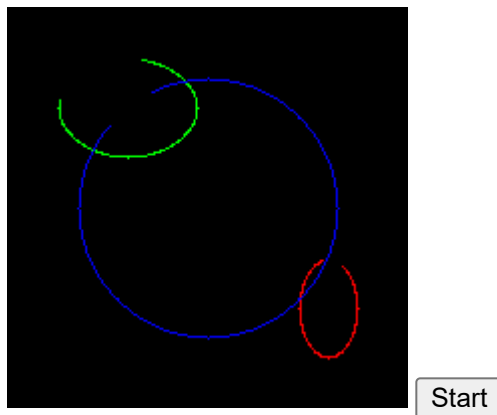
Dupa redefinirea unitatii de masura ( "Volt" ) si ( "Amp" ), folosind metoda set\_um(), obtinem:

Voltmetru--Volt

Ampermetru--Amp

Frecventmetru--Hz

#### - Utilizarea clasei "cerc"



Codul sursa:

```

<?php
// creare imagine 200*200
$img = imagecreatetruecolor(200, 200);

// definire culori
$white = imagecolorallocate($img, 255, 255, 255);
$red    = imagecolorallocate($img, 255, 0, 0);
$green  = imagecolorallocate($img, 0, 255, 0);
$blue   = imagecolorallocate($img, 0, 0, 255);
class cerc {

    /* Constructor */

    function __construct($i,$x,$y,$w,$h){

        // Variabile membru

        $this->imag = $i;
        $this->x0 = $x;
        $this->y0 = $y;
        $this->wx = $w;
        $this->hy = $h;

    }

    function set_val($ui,$u,$c){

        imagearc($this->imag, $this->x0 , $this->y0, $this->wx, $this->hy, $ui, $u, $c);

    }

}

$cerc1 = new cerc($img, 60, 50, 70, 50);
$cerc1->set_val(rand(100,360),rand(100,360), $green);

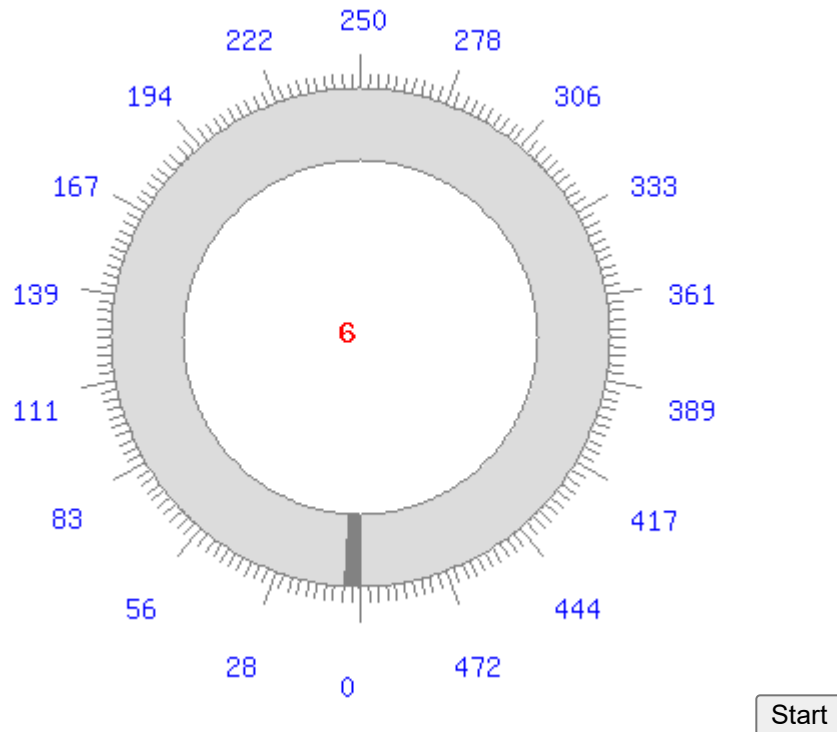
$cerc2 = new cerc($img, 160, 150, 30, 50);
$cerc2->set_val(rand(100,360),rand(100,360), $red);

$cerc3 = new cerc($img, 100, 100, 130, 130);
$cerc3->set_val(rand(100,360),rand(100,360), $blue);
// output image in the browser
header("Content-type: image/png");
imagepng($img);

// free memory
imagedestroy($img);
?>

```

**- Instrument virtual bazat pe clasa "manometru"**



Codul sursa:

```
<?php
// creare imagine 200*200
$img = imagecreatetruecolor(400, 400);
// definire culori
$white = imagecolorallocate($img, 255, 255, 255);
// fundal alb imagine
imagefill($img, 0, 0, $white);
$vmax=500; // Valoarea maxima afisata
class manometru {

    function __construct($i,$x,$y,$w,$mx){

        /* Variabile membru */

        $this->imag = $i;                // imaginea
        $this->x0 = $x;                  // pozitia pe x in cadrul imaginii
        $this->y0 = $y;                  // pozitia pe y in cadrul imaginii
        $this->wx = $w;                  // latimea imaginii
        $this->val_max = $mx;            // valoarea maxima afisata
        $this->xc=$x+$w/3+20;            // Coordonata x a centrului arcului de cerc
        $this->yc=$y+$w/3+20;            // Coordonata y a centrului arcului de cerc

        // variabile publice pentru toata clasa

        $this->white = imagecolorallocate($this->imag, 255, 255, 255);
        $this->blue = imagecolorallocate($this->imag, 0, 0, 255);
        $this->red = imagecolorallocate($this->imag, 255, 0, 0);
        $this->gray = imagecolorallocate($this->imag, 130, 130, 130);
        $this->l_gray = imagecolorallocate($this->imag, 220, 220, 220);
        $this->black = imagecolorallocate($this->imag, 0, 0, 0);

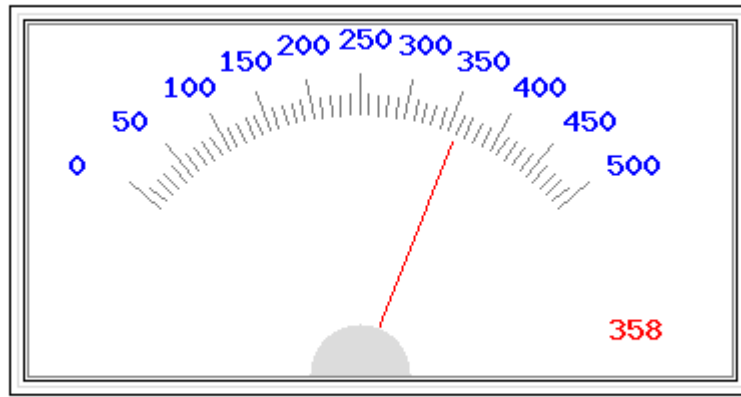
    }
    function desen_grd()
    {
        /* Variabile locale */
```

```

    // variabile locale //
    $lg = 5; // latimea gradatiilor
    $raza = $this->wx / 3; // raza gradatiilor
    $val_a = 0; // valoarea pentru afisat a gradatiilor
    $nrd = 0; // numarul curent al diviziunii
    $alfa_gr = 630; // alfa_gr unghiul in grade
    while ($alfa_gr > 270) // afisarea se face in sensul acelor de ceasornic adica
de la 630 gr la 270
    {
        if ($nrd % 10 == 0)
        {
            $xt = $this->xc + ($raza-0*$lg) * cos(deg2rad($alfa_gr))-10; // coordonate text
            $yt = $this->yc - ($raza-0*$lg) * sin(deg2rad($alfa_gr));
            $x1 = $this->xc + ($raza-5*$lg) * cos(deg2rad($alfa_gr)); // coordonate
sfarsit gradatii mari
            $y1 = $this->yc - ($raza-5*$lg) * sin(deg2rad($alfa_gr));
            $val_a=round(2*$nrd*($this->val_max)/360); // calculare valoare
gradatie
            imagestring($this->imag, 4, $xt,$yt, $val_a, $this->blue); // scriere
valoare gradatie
        }
        else
        {
            $x1 = $this->xc + ($raza-7*$lg) * cos(deg2rad($alfa_gr)); // coordonate
sfarsit gradatii mici
            $y1 = $this->yc - ($raza-7*$lg) * sin(deg2rad($alfa_gr));
        }
        $x2 = $this->xc + ($raza - 9 * $lg) * cos(deg2rad($alfa_gr)); // coordonate
inceput gradatii mari sau mici
        $y2 = $this->yc - ($raza - 9 * $lg) * sin(deg2rad($alfa_gr));
        imageline($this->imag, $x1, $y1, $x2, $y2, $this->gray); // trasare
gradatii
        $alfa_gr= $alfa_gr- 2;
        $nrd=$nrd+1;
    }
}
function set_val( $vl){
    $wd=$this->wx/7; // latimea arcului de cerc
    $alfa_i = 90; // unghiul initial
    $alfa = $alfa_i+360*($vl/$this->val_max); // unghiul curent
    imagefilledarc($this->imag, $this->xc , $this->yc, $this->wx/2, $this->wx/2, 0, 360,
$this->l_gray,4);
    imagefilledarc($this->imag, $this->xc , $this->yc, $this->wx/2, $this->wx/2,
$alfa_i, $alfa, $this->gray,4);
    imagefilledarc($this->imag, $this->xc , $this->yc, $this->wx/2-$wd,$this->wx/2-$wd,
0, 360, $this->white,4);
    imagearc($this->imag, $this->xc , $this->yc, $this->wx/2, $this->wx/2, 0, 360,$this-
>gray);
    imagearc($this->imag, $this->xc , $this->yc, $this->wx/2-$wd, $this->wx/2-$wd, 0,
360,$this->gray);
    imagestring($this->imag, 5, $this->xc-10 , $this->yc-10,$vl , $this->red);
}
}
$manon = new manometru($img, 0, 0, 500, $vmax);
$manon->desen_grd();
$manon->set_val(rand(0,$vmax));
// output image in the browser
header("Content-type: image/png");

```

### - Instrument virtual bazat pe clasa "voltmetru"




Codul sursa:

```
<?php
$dimx=550;
$dimy=350;
$img = imagecreatetruecolor($dimx+10, $dimy);
class voltmtru {

    /* Constructor */

    function __construct($i,$x,$y,$w,$u,$vm){

        /* Variabile membru */

        $this->imag = $i;
        $this->x0 = $x;
        $this->y0 = $y;
        $this->wx = $w;
        $this->hy = $w/1.7;
        $this->alfa = $u;
        $this->val_max = $vm;

        /* Variabile locale */

        $this->white = imagecolorallocate($this->imag, 255, 255, 255);
        $this->blue = imagecolorallocate($this->imag, 0, 0, 255);
        $this->red = imagecolorallocate($this->imag, 255, 0, 0);
        $this->gray = imagecolorallocate($this->imag, 130, 130, 130);
        $this->l_gray = imagecolorallocate($this->imag, 220, 220, 220);
        $this->black = imagecolorallocate($this->imag, 0, 0, 0);
    }

    function init_instr()
    {
        $lg = 5;
        $xc = $this->x0 + $this->wx / 2;
        $yc = $this->y0 + $this->hy-20;
        $raza = $this->wx / 2;
        $val_a = 0; // valoarea pentru afisat

        // desen rama

        imagefill( $this->imag, 0, 0, $this->white);
        imagerectangle( $this->imag, $this->x0, $this->y0+1, $this->x0+$this->wx+20, $this->y0+$this->hy-10, $this->black );
    }
}
```

```

    imagerectangle( $this->imag, $this->x0+4, $this->y0+5, $this->x0+$this->wx+16, $this->y0+$this->hy-14, $this->l_gray);
    imagerectangle( $this->imag, $this->x0+7, $this->y0+8, $this->x0+$this->wx+13, $this->y0+$this->hy-17, $this->black);
    imagerectangle( $this->imag, $this->x0+9, $this->y0+10, $this->x0+$this->wx+11, $this->y0+$this->hy-19, $this->gray);
    imagesetthickness($this->imag, 1);

    // alfa_gr unghiul in grade

    $alfa_gr =180-$this->alfa;
    $nrd = 0;
    while ($alfa_gr >= $this->alfa)
    {
        if ($nrd % 5 == 0)
        {
            $xt = $xc + ($raza-0*$lg) * cos(deg2rad($alfa_gr))-10;
            $yt = $yc - ($raza-0*$lg) * sin(deg2rad($alfa_gr));
            $x1 = $xc + ($raza-5*$lg) * cos(deg2rad($alfa_gr));
            $y1 = $yc - ($raza-5*$lg) * sin(deg2rad($alfa_gr));
            imagestring($this->imag, 5, $xt,$yt, $val_a, $this->blue);
        }
        else
        {
            $x1 = $xc + ($raza-7*$lg) * cos(deg2rad($alfa_gr));
            $y1 = $yc - ($raza-7*$lg) * sin(deg2rad($alfa_gr));
        }
        $x2 = $xc + ($raza - 9 * $lg) * cos(deg2rad($alfa_gr));
        $y2 = $yc - ($raza - 9 * $lg) * sin(deg2rad($alfa_gr));
        imageline($this->imag, $x1, $y1, $x2, $y2, $this->gray);
        $val_a=$val_a+round(2*$this->val_max/((180-2*$this->alfa)));
        $alfa_gr= $alfa_gr- 2;
        $nrd=$nrd+1;
    }
    // redefinesc valoarea maxima in functie de maximul posibil de afisat pe ecran
    $val_a=$val_a-round(2*$this->val_max/((180-2*$this->alfa)));
    $this->val_max=$val_a;
}
function setval($val)
{
    $lg = 5;
    $xc = $this->x0 + $this->wx / 2;
    $yc = $this->y0 + $this->hy-20;
    $raza = $this->wx / 2;

    // $alfa_gr unghiul in grade

    $alfa_gr =(180-$this->alfa)-($val*(round((180-2*$this->alfa)/$this->val_max,2)));
    $x = $xc + ($raza-10*$lg) * cos(deg2rad($alfa_gr));
    $y = $yc - ($raza-10*$lg) * sin(deg2rad($alfa_gr));
    imageline($this->imag, $xc, $yc, $x, $y, $this->red);
    imagefilledarc($this->imag, $xc, $yc, 50, 50, 180, 0, $this->l_gray,
    IMG_ARC_PIE);
    imagestring($this->imag, 5, $xc+$raza-50,$yc-30, $val, $this->red);
}
}
$instr1 = new virt_instrum($img, 50, 50, 450, 40,500);
$instr1->voltmetru();
$instr1->setval(rand(0,500));

// output image in the browser
header("Content-type: image/png");
imagepng($img);

```



## Pagini dinamice web responsive

Vom folosi un meniu web responsiv si vom crea o pagina dinamica

### Utilizarea PHP pentru realizarea paginilor WEB dinamice

- Aplicatiile web dinamice sunt realizate prin:
  - utilizarea si incorporarea fisierelor externe
  - manipularea formularelor si prelucrarea datelor continute de acestea
  - utilizarea functiilor proprii
  - utilizarea functiilor speciale

Succes in realizarea paginilor dinamice !

Fisierul meniu fiind:

```

<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <title>Meniu dinamic</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js">
</script>
</head>
<body>
<div class="slideshow-container">
<!--<font color="red" size="4"> Meniu dinamic Web responsive</h3> </font-->
    <nav class="navbar navbar-inverse">
        <div class="container-fluid">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#myNavbar">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">UMFST</a>
            </div>
            <div class="collapse navbar-collapse" id="myNavbar">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="start_resp.php">Home</a></li>
                    <li class="dropdown">
                        <a class="dropdown-toggle" data-toggle="dropdown" href="#">Grafica<span
class="caret"></span></a>
                        <ul class="dropdown-menu">
                            <li><a href="manom_din.php">Manometru</a></li>
                            <li><a href="voltm_din.php">Voltmetru</a></li>
                            <li><a href="sinus_afis.php">Sinus</a></li>
                        </ul>
                    </li>
                    <li class="dropdown">
                        <a class="dropdown-toggle" data-toggle="dropdown" href="#">Forms<span class="caret">
</span></a>
                        <ul class="dropdown-menu">
                            <li><a href="sah.php">Tabla sah </a></li>
                            <li><a href="calculator.php">Calc_tva </a></li>
                            <li><a href="manom_form.php">Presiune</a></li>
                        </ul>
                    </li>
                </ul>
                <ul class="nav navbar-nav navbar-right">
                    <li><a href="#"><span class="glyphicon glyphicon-user"></span> Sign Up</a></li>
                    <li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
                </ul>
            </div>
        </div>
    </nav>

```

Toate paginile trebuie sa includa fisierul meniu. De exemplu pagina initiala "start\_resp.php" fiind:

```
<?php # Script : start_resp.php
include ('menu.php');
echo '
<center><table border =1 width="500">
<tr><td height="50" bgcolor="magenta">
<center><font color="white" font size="3">
<b> Utilizarea PHP pentru realizarea paginilor WEB dinamice</b>
</center>
</td></tr><tr><td>
</h1><font color="blue" font size="2">
</b><br><br>
<ul>
<li>Aplicatiile web dinamice sunt realizate prin:<br><br>
<ul>
<li> utilizarea si incorporarea fisierelor externe
<li> manipularea formularelor si prelucrarea datelor continute de acestea
<li> utilizarea functiilor proprii
<li> utilizarea functiilor speciale
</ul>
</li>
</ul>
<br><br><font color="red" font size="2"> Succes in realizarea paginilor dinamice !<br><br>
</td></tr></table>
</div>';
?>
```

La fel pagina "calculator.php" pentru calculul tva-ului include fisierul menu.

```

<?php # Script: - calculator.php
include ('menu.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 '<center><h2>Total Cost</h2>
        <p>Costul total al celor ' . $_POST['cantit'] . ' bucati, este: ' . number_format ($total,
2) . ' lei <br> Din care TVA ' . $_POST['cota_tva'] . '% = ' . number_format
($_POST['pret']*$_POST['cantit']*$_tva, 2) . ' .</p></center>';

    } else { // S-au trimis valori invalide.
        echo '<center><h2>Eroare !</h2>
        <p>Va rog introduceti cantitatea, pretul, si cota TVA.</p><br>
        </center>';
    }
} else {
    echo '<center><h2>Calcularea valorii si TVA-ului </h2>
    <p>Introduceti cantitatea, pretul si cota TVA.</p><br>
    </center>';
}
?>
<center><table border =1 width="400"><tr><td bgcolor="#D9E3EE">
<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>

```

Pentru afisarea valorii presiunii avem nevoie de mai multe fisiere, primul fiind "manom\_form":

```

<?php # Script: - manom_form.php
include ('menu.php');
echo '
<div id="Continut_p">';
// Se verifica daca form-ul a fost trimis
if (isset($_POST['trimis'])) {

    // Validare mimimala form.
    if ( is_numeric($_POST['val_max']) && is_numeric($_POST['val']) ) {

        // Calcularea rezultatelor.

        $total = ($_POST['val_max'] * $_POST['val']) ;

        // Afisarea rezultatelor.
        echo '<center>';

        } else { // S-au trimis valori invalide.
            echo '<center>
<p class="error">Va rog introduceti valoarea maxima, si valoarea de afisat</p>';
        }

    } else {
        echo '<center>';
    }

?>

<center><table border =1 width="400"><tr><td bgcolor="#D9E3EE">
<form action="manom_form.php" method="post">
    <table align ="center"width="300"><tr><td align="left">
        Valoarea maxima: </td><td><input type="text" name="val_max" size="5" maxlength="10"
value="100" />
    </td></tr><tr><td align="left">
        Valoarea: </td><td><input type="text" name="val" size="5" maxlength="10" value="25" />
    </td></tr><tr><td align="left">
        <input type="submit" name="trimite" value="Afiseaza!" />
        <input type="hidden" name="trimis" value="TRUE" />
    </td></tr></table>
</form>
</td></tr></table></center>
</div>

```

manometru.php fiind:

```

<?php
include ('manom_clss.php');
// creare imagine 250*200
$img = imagecreatetruecolor(250, 250);
// definire culori
$white = imagecolorallocate($img, 255, 255, 255);
// fundal alb imagine
imagefill($img, 0, 0, $white);
if (isset($_GET['vm'])) {
    $vmax=$_GET['vm'];
}
else{
    $vmax=500; // Valoarea maxima afisata
}
if (isset($_GET['vl'])) {
    $val=$_GET['vl'];
}
else{
    $val=rand(0,$vmax); // Valoarea maxima afisata
}
$manon = new manometru($img, 0, 0, 300, $vmax);
$manon->desen_grd();
$manon->set_val($val);
// output image in the browser
header("Content-type: image/png");
imagepng($img);

// free memory
imagedestroy($img);
?>

```

Clasa manometru:

```

<?php
class manometru {

    function __construct($i,$x,$y,$w,$mx){

        /* Variabile membru */

        $this->imag = $i;           // imaginea
        $this->x0 = $x;             // pozitia pe x in cadrul imaginii
        $this->y0 = $y;             // pozitia pe y in cadrul imaginii
        $this->wx = $w;             // latimea imaginii
        $this->val_max = $mx;       // valoarea maxima afisata
        $this->xc=$x+$w/3+20;       // Coordonata x a centrului arcului de cerc
        $this->yc=$y+$w/3+20;       // Coordonata y a centrului arcului de cerc

        // variabile publice pentru toata clasa

        $this->white = imagecolorallocate($this->imag, 255, 255, 255);
        $this->blue  = imagecolorallocate($this->imag, 0, 0, 255);
        $this->red   = imagecolorallocate($this->imag, 255, 0, 0);
        $this->gray  = imagecolorallocate($this->imag, 130, 130, 130);
        $this->l_gray = imagecolorallocate($this->imag, 220, 220, 220);
        $this->black = imagecolorallocate($this->imag, 0, 0, 0);

    }

    function desen_grd()
    {

```

```

/* Variabile locale */
$lg = 5; // latimea gradatiilor
$raza = $this->wx / 3+16; // raza gradatiilor
$val_a = 0; // valoarea pentru afisat a gradtiilor
$nrd = 0; // numarul curent al diviziunii
$alfa_gr = 630; // alfa_gr unghiul in grade
while ($alfa_gr > 270) // afisarea se face in sensul acelor de ceasornic adica
de la 630 gr la 270
{
    if ($nrd % 10 == 0)
    {
        $xt = $this->xc + ($raza-2*$lg) * cos(deg2rad($alfa_gr))-10;//
        coordonate text
        $yt = $this->yc - ($raza-2*$lg) * sin(deg2rad($alfa_gr));
        $x1 = $this->xc + ($raza-5*$lg) * cos(deg2rad($alfa_gr)); // coordonate
        sfarsit gradatii mari
        $y1 = $this->yc - ($raza-5*$lg) * sin(deg2rad($alfa_gr));
        $val_a=round(2*$nrd*($this->val_max)/360); // calculare
        valoare gradatie
        imagestring($this->imag, 4, $xt,$yt, $val_a, $this->blue);// scriere
        valoare gradatie
    }
    else
    {
        $x1 = $this->xc + ($raza-7*$lg) * cos(deg2rad($alfa_gr)); // coordonate
        sfarsit gradatii mici
        $y1 = $this->yc - ($raza-7*$lg) * sin(deg2rad($alfa_gr));
    }
    $x2 = $this->xc + ($raza - 9 * $lg) * cos(deg2rad($alfa_gr)); // coordonate
    inceput gradatii mari sau mici
    $y2 = $this->yc - ($raza - 9 * $lg) * sin(deg2rad($alfa_gr));
    imageline($this->imag, $x1, $y1, $x2, $y2, $this->gray); // trasare gradatii
    $alfa_gr= $alfa_gr- 2;
    $nrd=$nrd+1;
}
}
function set_val( $vl)
{
    $wd=$this->wx/7; // latimea arcului de cerc
    $alfa_i = 90; // unghiul initial
    $alfa = $alfa_i+360*($vl/$this->val_max); // unghiul curent
    imagefilledarc($this->imag, $this->xc , $this->yc, $this->wx/2, $this->wx/2, 0, 360,
    $this->l_gray,4);
    imagefilledarc($this->imag, $this->xc , $this->yc, $this->wx/2, $this->wx/2,
    $alfa_i, $alfa, $this->red,4);
    imagefilledarc($this->imag, $this->xc , $this->yc, $this->wx/2-$wd,$this->wx/2-$wd,
    0, 360, $this->white,4);
}

```