

Curs 6

# Tehnici moderne de proiectare a aplicatiilor web

# CURS

I.	HTML si XHTML (recapitulare)	1 oră
II	CSS	2 ore
III	Baze de date, punct de vedere practic	1 oră
IV	Limbajul de interogare SQL	4 ore
V	PHP - HyperText Preprocessor	8 ore
VI	XML - Extended Mark-up Language si aplicatii	4 ore
VII	Conlucrare intre PHP/MySql, PHP/XML, Javascript/HTML	2 ore
VIII	Exemple de aplicatii	6 ore
	Total	28 ore

# LABORATOR

I.	Implementarea unui sistem de dezvoltare a aplicatiilor Web, instalare PHP, MySql, Apache si legaturile dintre ele	2 ore
II	Design web avansat folosind CSS	2 ore
III	Interogarea unei baze de date. Exercitii SQL	2 ore
IV	Utilizare PHP I	2 ore
V	Utilizare PHP II	2 ore
VI	Utilizare PHP pentru accesul la o baza de date	2 ore
VII	Aplicatie distribuita complexa	2 ore
	Total	14 ore

# CURS

I.	HTML si XHTML (recapitulare)	1 oră
II	CSS	2 ore
III	Baze de date, punct de vedere practic	1 oră
IV	Limbajul de interogare SQL	4 ore
V	PHP - HyperText Preprocessor	8 ore
VI	XML - Extended Mark-up Language si aplicatii	4 ore
VII	Conlucrare intre PHP/MySql, PHP/XML, Javascript/HTML	2 ore
VIII	Exemple de aplicatii	6 ore
	Total	28 ore

# Teme de proiect

- La toate temele **1p** din nota este obtinut de indeplinirea functionalitatii cerute.
- La toate temele forma paginii prezinta importanta (dependentă de dificultatea temei)

# Teme de nota 10

- Necesitatea conlucrării între 2 studenți cu două teme “pereche” (a și b). Fiecare tema se alocă **doar** unei perechi de studenți simultan.
- **Necesitatea** investigării posibilităților de îmbunătățire
- Baza de date cu care se lucrează să conțină minim **100** de înregistrări.
- Tema se **sustine** în săptămâna a 14-a
- Sustinerea este precedată obligatoriu de introducerea datelor într-o bază de date vidă (restaurare **back-up**) și copierea codului pe server-ul de referință
- Forma paginii este importantă

# Teme de nota 10+

- Cerinte
  - Necesitatea conlucrării între 3 studenți cu trei teme "pereche". Fiecare tema se alocă **doar** unei echipe de 3 studenți simultan.
  - **Necesitatea** investigării posibilităților de îmbunătățire
  - Baza de date cu care se lucrează să conțină minim **200** de înregistrări.
  - Necesitatea utilizării Javascript în aplicație (aplicație liberă dar cu efect **tehnic** nu estetic)
  - Tema se predă/trimite cu macar 1 zi înaintea susținerii ei
  - Tema se **sustine** în săptămâna a 14-a
  - Susținerea e precedată obligatoriu de introducerea datelor într-o bază de date vidă (restaurare **back-up**) și copierea codului pe server-ul de referință
  - Forma paginii trebuie să respecte cerințele "**F shape pattern**".
- Avantaje
  - **1p bonus** la nota finală (dacă tema e îndeplinită)
  - Acceptarea absentelor la laborator

# Exemplu

- 1. Galerie de imagini in care imaginile sunt ordonate dupa categorii.



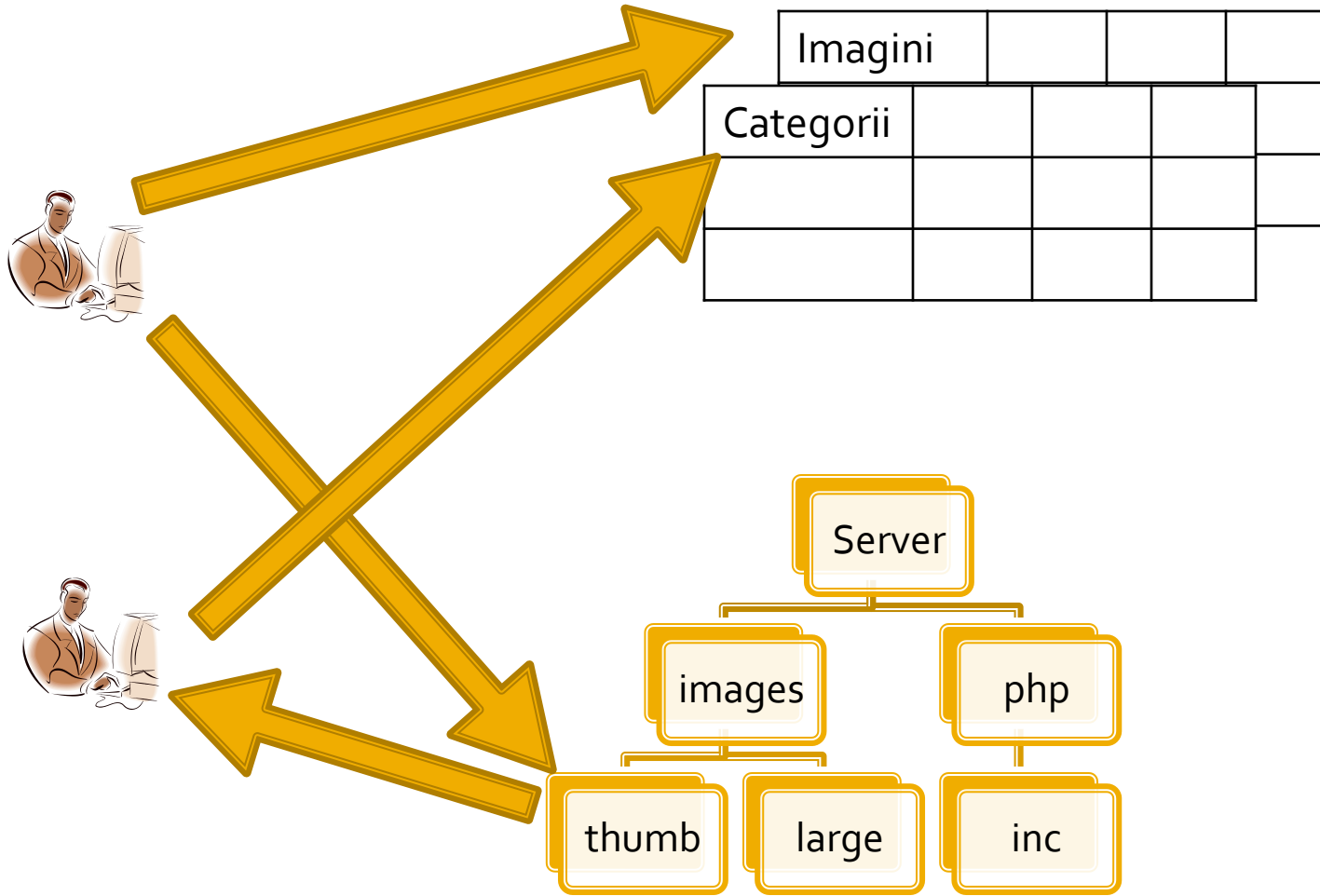
a. aplicatia pentru adaugarea de categorii si afisare a imaginilor (cu alegerea prealabila a categoriei si afisarea listei de imagini format mic)



b. aplicatia pentru adaugare de imaginilor (cu alegerea prealabila a categoriei si generarea prealabila a imaginii format mic)



# Exemplu



# Curs 5

- Scrierea datelor pe disc ca in orice limbaj de programare
- Format text sau binar
- Citirea si interpretarea datelor

# Exemplu

- fiecare produs e caracterizat de:
  - nume
  - pret
  - (eventual) descriere
  - cantitate comandata
- matricea asociata:  
`$produse["produs"] = ["caracteristici"]`

```
$produse = array ( 'carte' => array ("descr" => "mai multe pagini scrise", "pret" => 100, "cant" => 0),  
                  'caiet' => array ("descr" => "mai multe pagini goale", "pret" => 50, "cant" => 0),  
                  'penar' => array ("descr" => "loc de depozitat instrumente", "pret" => 150, "cant" => 0),  
                  'stilou' => array ("descr" => "instrument de scris albastru", "pret" => 125, "cant" => 0),  
                  'creion' => array ("descr" => "instrument de scris gri", "pret" => 25, "cant" => 0)  
                );
```

# Scrierea datelor pe disc

- Structura datelor este total dependentă de algoritmul utilizat
- Modificările sunt dificil de realizat și implică schimbarea fișierului de date și schimbarea codului în întregime
- Exemplu: gruparea elementelor pe categorii
  - `$produse["produs"] = ["caract."]`
  - `$produse["categorie"]["produs"] = ["caract."]`
- index-ul suplimentar introdus transformă o matrice bidimensională în una tridimensională
  - dificil de implementat cu fișiere text (caract/linii – 2D)

# Rezolvare

- modificarea fisierului text cu introducerea categoriei ca prim parametru in fisier

carte	mai multe pagini scrise legate	100	0	
caiet	mai multe pagini goale legate	75	0	
penar	loc de depozitat instrumente de scris	150	0	
stilou	instrument de scris albastru	125	0	
papetarie	carte	mai multe pagini scrise legate	100	0
papetarie	caiet	mai multe pagini goale legate	75	0
instrumente	penar	loc de depozitat instrumente de scris	150	0
instrumente	stilou	instrument de scris albastru	125	0

# Dezavantaj

- date redundante: numele categoriei apare listat pentru fiecare produs in parte
- sensibilitate la erori si nevoie de precizie in realizarea fisierului
  - numele categoriei trebuie scris exact de fiecare data
  - o mica greseala ("instrument" in loc de "instrumente") are ca efect crearea unei categorii suplimentare
- Daca numarul de produse e mare schimbarea e dificil de realizat manual, si complicat de realizat prin cod
- Produsele care nu apartin nici unei categorii nu vor putea fi diferite

# Cod

- Codul ramane in mare parte acelasi
- Cateva modificari sunt necesare


```
$matr=file("produse.txt");
foreach ($matr as $linie)
{
    $valori=explode("\t",$linie,4);
    $produse[$valori[0]]=array ("descr" => $valori[1], "pret" => $valori[2], "cant" => $valori[3]);
}
```

```
$matr=file("produse.txt");
foreach ($matr as $linie)
{
    $valori=explode("\t",$linie,5);
    $produse[$valori[0]] [$valori[1]]=array ("descr" => $valori[2], "pret" => $valori[3], "cant" =>
$valori[4]);
}
```

# Acces la date

- In principiu, cu doua bucle foreach

```
$index=1;
foreach ($produse as $prod => $detalii)
    {?>
<tr><td><?php echo $index;?></td><td><?php echo ucfirst(strtolower($prod));?></td><td><?php echo
$detalii['descr'];?></td><td align="center"><?php echo $detalii['pret'];?></td></tr>
<?php $index++;
    }
```



```
$index=1;
foreach ($produse as $categ => $lista_categ)
    foreach ($lista_categ as $prod => $detalii)
        {?>
<tr><td><?php echo $index;?></td><td><?php echo $categ;?></td><td><?php echo
ucfirst(strtolower($prod));?></td><td><?php echo $detalii['descr'];?></td><td align="center"><?php
echo $detalii['pret'];?></td></tr>
<?php $index++;
        }
```



# Rezultat

**Magazin**

**Firma X SRL**

## Magazin online Firma X SRL

### Lista Produse

Nr.	Categorie	Produs	Descriere	Pret
1	papetarie	Carte	mai multe pagini scrise legate	100
2	papetarie	Caiet	mai multe pagini goale legate	75
3	papetarie	Hartie scris	mai multe pagini goale NElegate	50
4	instrumente	Penar	loc de depozitat instrumente de scris	150
5	instrumente	Stilou	instrument de scris albastru	125
6	instrumente	Creion	instrument de scris gri	25
7	instrumente	Marker	instrument de scris verde	25
8	audio-video	Cd	canta	50
9	audio-video	Dvd	vizual	100
10	audio-video	Blue ray	vizual extrem	500

[Comanda](#)

**Accesul la metode externe de  
stocare eficiente a datelor**

---

# XML

- XML - eXtensible Markup Language
- O forma a SGML - Standard Generalized Markup Language (ISO 8879:1986 SGML)
- O metoda de a descrie structura si importanta datelor si continutul lor fara a da indicatii despre afisare
- XSLT - XSL Transformations (Extensible Stylesheet Language) limbaj de conversie a XML in alte tipuri de documente XML cu sau fara reprezentare grafica

# HTML/XHTML vs XML

- XML
  - proiectat pentru a **descrie** datele
  - orientat spre **continutul** datelor respective
  - o metoda de a transmite informatiile **independent** de platforma si hardware
- HTML/XHTML
  - proiectat pentru a **afisa** datele
  - orientat spre **forma** pe un ecran a datelor respective
  - o metoda de a **afisa uniform** datele indiferent de platforma si hardware

# XML

- In conceptie asemanator cu XHTML
  - etichete XHTML ("tag" - EN)
  - elemente XML ("element" - EN) descrise de etichete ("tag" - EN)
- Elementele XML accepta attribute (similar XHTML)
- Conceptele de scriere a documentului similar XHTML
- Diferenta majora:
  - HTML – etichetele si attributele sunt predefinite si orientate spre modalitatea de afisare a datelor
  - XML – etichetele de identificare a elementelor si attributele sunt la latitudinea creatorului documentului, introducand structura in date

# Avantaje

- **Redundanta**
  - fiecare element XML trebuie introdus complet
  - aceasta permite detectia si corectarea facila a erorilor
- **Auto descriptiv**
  - XML este un limbaj bazat pe text, insesi elementele si attributele descriu datele
  - usor de citit/corectat pentru utilizatori umani
- **Generalitate**
  - orice fisier XML poate fi citit de orice aplicatie XML
  - anumite aplicatii necesita o anumita structura a datelor

# Reguli XML

- Aproape orice caracter UNICODE poate fi utilizat
- 107000 caractere, 90 scrieri diferite
- exceptii:
  - &lt; <
  - &gt; >
  - &amp; &
  - &quot; "
  - &apos; `

# Reguli XML

- etichetele de definire a elementelor **trebuie** închise sau elementul declarat ca vid
  - `<descriere> ... </descriere>`
  - `<descriere></descriere>`
  - `<descriere />`
- atributele **trebuie** scrise între ghilimele
  - `<categorie nume="papetarie">`
- etichetele și atributele sunt **Case Sensitive**
  - greșit -> `<descriere> ... </Descriere>`
  - greșit -> `<descriere> ... </descriere><DESCRIERE> ... </DESCRIERE>`



# Reguli XML

- Documentele XML creaza o structura ierarhica foarte stricta
- Nu sunt permise etichete suprapuse
  - `<x><y></y></x>` → permis
  - `<x><y></x></y>` → interzis
- Trebuie sa existe un singur element radacina care sa le cuprinda pe toate celelalte
  - similar cu `<html></html>`

# Structura unui document XML

- prima linie – definitia tipului de document
  - `<?xml ... ?>`
  - `<?xml version="1.0" encoding="utf-8"?>`
- element radacina
  - `<radacina> ...[elemente]... </radacina>`

# XML Concepte

- comentariile pot fi introduse oriunde in interiorul documentului cu conditia sa fie in exteriorul oricarui element
  - similare cu comentariile HTML: intre `<-- si -->`
- Sectiuni de date neinterpretate
  - intre `<![CDATA[ si ]]>`
  - pentru a putea introduce date care ar putea contine caracterele interzise
    - cod
    - date binare oarecare

# Exemplu

```
<?xml version="1.0" encoding="utf-8"?>
<gallery titlu="Photomagic" thumbDir="./fotografii/thumbnails/" imageDir="./fotografii/">
  <category nume="VIATA">
    <image>
      <desc>curiozitate</desc>
      <img>foto33.jpg</img>
      <thumb>foto33TH.jpg</thumb>
    </image>
  </category>
  <category nume="NUNTA">
    <image>
      <desc>asteptare</desc>
      <img>foto132.jpg</img>
      <thumb>foto132TH.jpg</thumb>
    </image>
  </category>
  <category nume="ORASE">
    <image>
      <desc>Atena</desc>
      <img>foto28.jpg</img>
      <thumb>foto28TH.jpg</thumb>
    </image>
  </category>
</gallery>
```

# Acces prin cod PHP la date XML

- Biblioteci PHP de acces la XML
  - DOM XML
  - XML Parser (Reader, Writer)
  - Simple XML

# Simple XML

- `$xml = simplexml_load_file('cale');`
- `$xml = simplexml_load_string($string);`
- Functiile care creaza un obiect din clasa predefinita **SimpleXMLElement** prin incarcarea si interpretarea documentului sau sirului XML
- obiectul creat (`$xml`) contine elementul radacina
  - exemplul anterior → `$xml` va contine **`<gallery></gallery>`**

# Simple XML

- fiecare **tip** de element descendent creaza o proprietate a clasei cu acel nume
  - proprietatea clasei (descendent) e o matrice de obiecte **SimpleXMLElement** daca nu e unicul descendent
  - matricea e caracterizata de elementele implicite de indexare (index intreg, pornind de la 0) dar e recomandata iteratia folosind `foreach()`

# Exemplul anterior

- elementul <gallery> contine un numar mare de descendentii de un singur tip, <category>
- obiectul radacina a fost redenumit prin operatia de deschidere a fisierului
  - `$xml = simplexml_load_file('cale');`
- obiectul `$xml` va contine ca proprietate
  - `$xml->category`
  - `$xml->category` este o matrice deoarece nu este o singura categorie
  - descendentii se pot accesa cu
    - `$xml->category[0], $xml->category[1], ...`
    - `foreach ($xml->category as $categ_curent) { // $categ_curent obiect de tip SimpleXMLElement, aceleasi reguli }`



# Simple XML

- attributele XML pot fi accesate in obiectul corespunzator ca si cum ar fi elementele unei matrici cu indice text, indicele fiind acelasi cu numele atributului cautat
- In exemplul anterior fiecare element `<category>` are un atribut `nume`
  - `<category nume="VIATA">`
  - ```
foreach ($xml->category as $categ_curent)
{
    echo $categ_curent['nume'];
}
```

# Recursivitate

- Fiecare proprietate a clasei **SimpleXMLElement** este de asemenea un obiect din clasa **SimpleXMLElement** daca mai are descendenti
  - `$xml->category[o]` este de tip obiect:  
`$xml->category[o] ->image ...`
- Proprietatea clasei contine datele elementului daca nu mai sunt alti descendenti
  - `echo $xml->category[o]->image[o]->img // foto33.jpg`
- Pentru ca proprietatea sa fie tratata ca fiind de tip string (implicit la afisare) trebuie fortata conversia
  - `if ((string)$xml->category[o]->image[o]->img=="foto33.jpg") ...`

# Modificare de cod XML

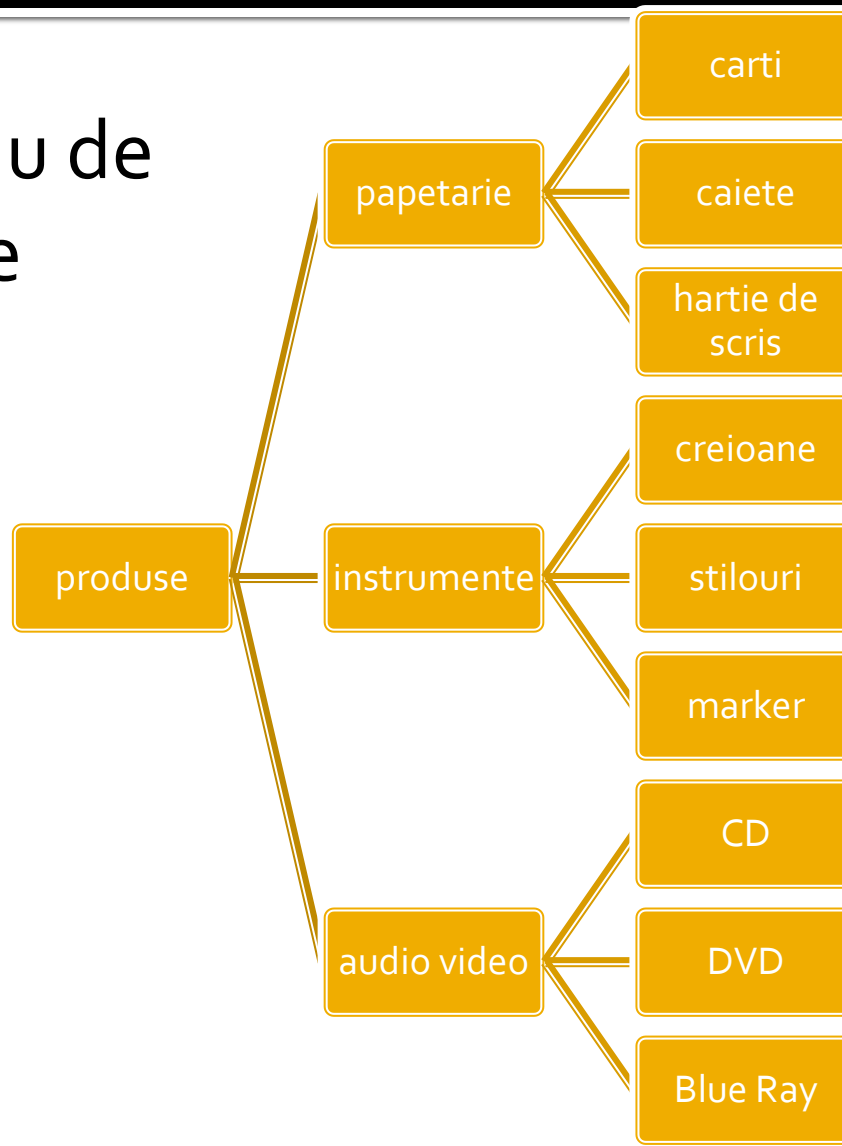
- incepand cu PHP 5.1.3.
- SimpleXMLElement::addChild(string nume, string valoare)
- SimpleXMLElement::addAttribute(string nume, string valoare)
- SimpleXMLElement::asXML('filename') scrie in fisierul 'filename' rezultatul sau il returneaza ca un sir
- schimbarea continutului unui element
  - `$xml->category[0]->image[0]->img = "foto33.jpg"`

# Laborator 6

- Sa se continue magazinul virtual cu:
  - produsele sunt grupate pe categorii de produse
  - sa prezinte utilizatorului o lista de grupe de produse pentru a alege
  - sa prezinte utilizatorului o lista de produse si preturi in grupa aleasa
  - lista de produse si preturi se citeste dintr-un fisier **XML**
  - se preia comanda si se calculeaza suma totala
  - **se creaza o pagina prin care vanzatorul poate modifica preturile si produsele**

# Laborator 6

- exemplu de grupare



# Contact

- Laboratorul de microunde si optoelectronica
- <http://rf-opto.etti.tuiasi.ro>
- [rdamian@etti.tuiasi.ro](mailto:rdamian@etti.tuiasi.ro)