

Laborator 7

Proiectarea unei aplicații cu baze de date online

Un sistem de baze de date are în arhitectura client-server două componente distincte și anume: **serverul** care are instalat sistemul de gestiune al bazelor de date și baza de date în sine și **clientul** care are instalate programe de aplicație (un program dedicat sau un browser) care interacționează cu baza de date printr-o interfață software (PHP, ASP, JAVA, etc.). Serverul are rolul de a procesa și returna rezultatul cererilor de la clienți.

Configurația de tip client-server poate fi de trei tipuri:

- un singur client și un singur server
- mai mulți clienți și un singur server
- mai mulți clienți și mai multe servere

Clientul are rolul de a administra interfața cu utilizatorul, de a accepta și verifica sintaxa intrărilor utilizatorului, procesează aplicațiile generând cerințele pentru baza de date, cerințe ce vor fi mai apoi transmise către server și de asemenea de a trimite răspunsul înapoi utilizatorului.

Serverul primește și procesează cerințele clienților pentru baza de date, verifică autorizarea, asigură respectarea constrângerilor de integritate, efectuează procesarea interogarea/reactualizarea și transmitere răspunsului către client, întreține catalogul de sistem și oferă acces simultan la baza de date.

După accesul la baza de date, SGBD pot fi clasificate în două categorii și anume:

- Baze de date monolitice pentru care motorul BD și interfața se află pe același calculator; acestea sunt baze de date de talie și performanță redusă
 - Ex. Windows: Microsoft Access, Fox Pro, MySQL, etc.
 - Linux: (în general, Open Source) MySQL, PostgreSQL, etc.
- Baze de date client-server pentru care motorul BD și interfața se află pe calculatoare diferite ; acestea sunt BD de talie și performanțe considerabile.
 - Ex. ORACLE, Microsoft SQL SERVER, IBM DB2, INFORMIX, etc

Driver de acces la BD: intermediaza interacțiunea dintre interpretorul de scripturi și BD; sunt instrumente software specializate care nu sunt vizibile nici programatorului nici utilizatorului [5].

Baza de date: constă dintr-un SGBDR instalat fie pe același calculator cu serverul Web, fie este accesibil prin rețea sau Internet. [5]

Ex:

- MySQL, SQL Server și Oracle
- Fox Pro și Access nu se pretează pentru BD web folosite intens.

Fișiere: informații cu caracter static trimise utilizatorilor la cerere[5]

În cadrul acestui capitol se va crea un site web care va reprezenta interfața on-line cu baza de date creată anterior. În acest scop se va utiliza MySQL pentru a crea baza de date și pentru a o gestiona și limbajul PHP pentru crearea interfeței web ce poate fi folosită de utilizator pentru a vizualiza informațiile dorite.

Limbajul PHP are multe avantaje, iar dintre acestea pot fi amintite:

- **simplicitate:** simplu de folosit, accesibil ne-programatorilor. Sintaxa asemănătoare limbajului C
- **eficiența:** include tehnici de POO
- **creșterea productivității:** un programator PHP poate folosi module scrise de un alt programator fără să fie nevoit să cunoască detalii specifice de implementare a codului.
- **cross-platform :** există implementări ale limbajului pe mai multe sisteme de operare, cele mai populare fiind cele de pe Linux și Windows. Un cod PHP care rulează pe Linux, poate fi ulterior folosit pe un server Windows
- **gratis :** PHP reprezintă un program Open Source, oricine poate folosi acest limbaj fără a fi nevoit să plătească ceva. Utilizatorii au acces la sursele PHP (scrise în limbajul C).

Ca dezavantaj în folosirea PHP se poate aminti faptul că programarea interfețelor complexe în PHP presupune cunoștințe avansate în programarea C++/Java.

Ca și sintaxă, scripturile scrise în limbajul PHP pot fi editate cu orice editor de text. Alte caracteristici sunt următoarele:

- terminarea unei instrucțiuni se face obligatoriu cu caracterul ;
- numele funcțiilor NU SUNT «case Sensitive» (nu se face diferența între litere mari/mici)
- numele variabilelor SUNT Case Sensitive
- ghilimelele se folosesc pereche "..." sau '... '
- variabilele sunt precedate de \$
- Codul PHP este delimitat de 2 seturi de etichete de deschidere și închidere:

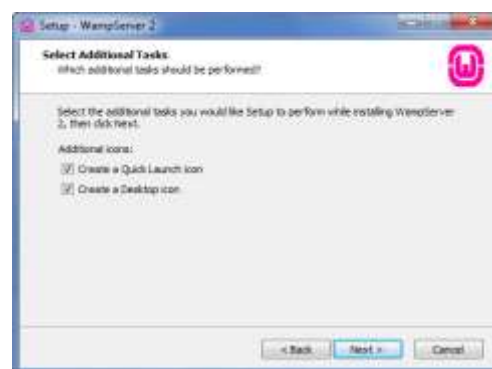
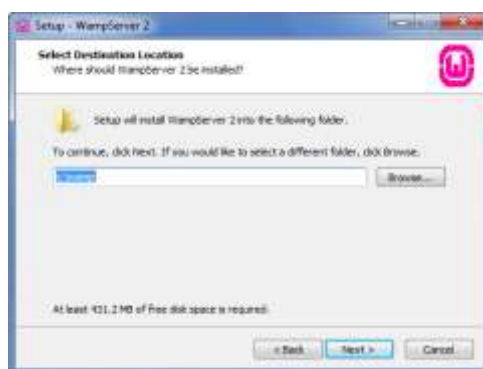
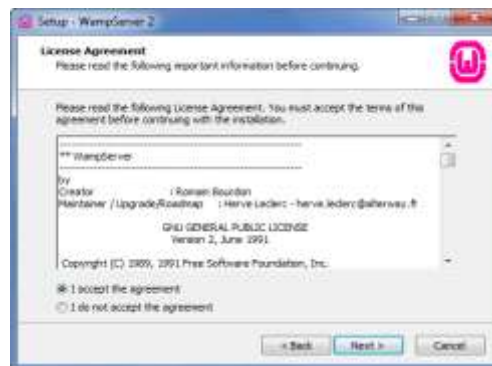
```
<?php sau <script language="php"?>  
.....  
?> </script>
```

Exemplu:

```
<?php  
echo 'Salut, acesta este primul meu script PHP';  
// comentariu PHP  
# comentariu PHP  
?>
```

Diferența dintre o aplicație html și una php, este aceea că o aplicație php conține date dinamice, ce au nevoie de existența unui server web care să știe să interpreteze conținutul dinamic. Un client, de obicei un web browser, va emite o cerere HTTP pentru a primi o resursă; dacă totul merge corect, serverul de HTTP va fi cel care va răspunde cu resursa dorită. Serverul HTTP cel mai utilizat pentru aplicațiile php este Apache HTTP Server. Proiectul Apache HTTP Server este un program colaborativ ce pune la dispoziție un HTTP (Web) Server robust, de calitate ridicată și gratuit. Un mare avantaj al Apache-ului este acela că el poate să fie instalat pe toate platformele. Serverul Wamp este un mediu de dezvoltare web, ce permite crearea de aplicații web folosind Apache, PHP și MySQL. Serverul Wamp instalează automat cele 3 servicii, permite configurarea lor fără a fi nevoie să edităm fișiere de configurare. Odată ce serverul wamp a fost instalat, este posibilă instalarea mai multor instanțe Apache, MySQL și PHP. [8]

În cazul aplicației de față s-a instalat wampserver 2.4-x64 care este gratuit. Pașii de instalare sunt prezentați în cele ce urmează:



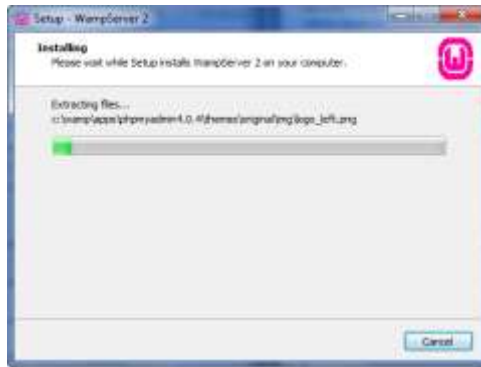



Fig.1. Instalarea și configurarea serverului WAMP

S-ar putea ca WAMP să ne pună să alegem browserul cu care vom lucra, iar în acest caz vom alege explorer.exe. Următorul pas este pornirea serverului WAMP. Dacă acesta funcționează, în bara de meniu va apărea semnul  colorat verde, în caz contrar acesta rămâne colorat roșu.

Pentru a verifica dacă MySQL este configurat corect, iar php are acces la datele din MySQL, trebuie să se deschidă un browser și să se verifice dacă linkul <http://localhost/phpmyadmin/> furnizează următoarea pagină (Fig. 15.2), iar introducând username-ul și password-ul pentru MySQL, utilizatorul va determina afișarea paginii din figura 15.3 care conține interfața grafică pentru MySQL ce este pusă la dispoziție de către php și este folosită pentru gestiunea bazelor de date MySQL care conține în arborele din stânga și baza de date creată de către noi.

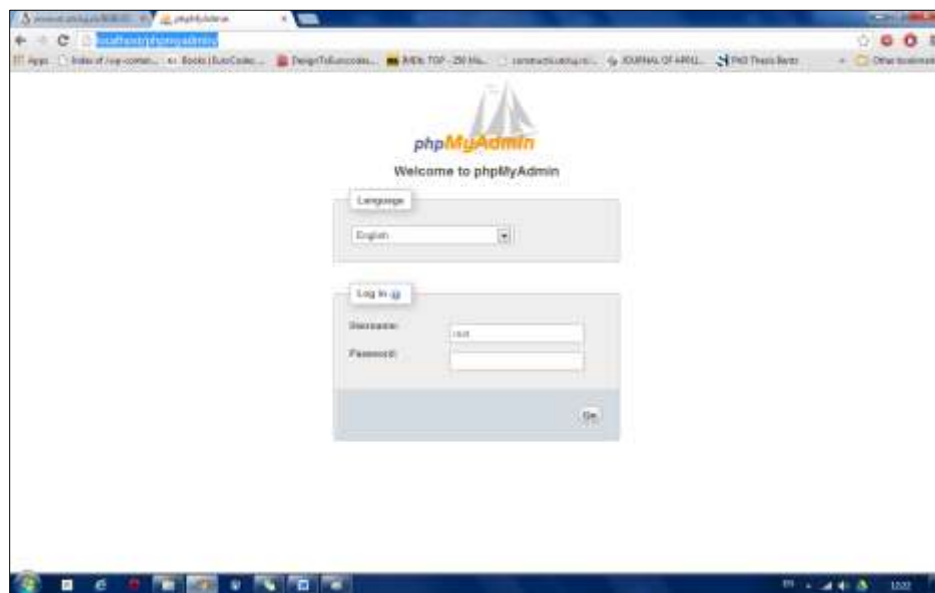


Fig. 2. Pagina de logare la PHPmyadmin.

În momentul instalării WAMP, în locația în care acesta s-a instalat s-a creat un fișier WAMP, iar în interiorul acestuia se găsește un fișier WWW; aceasta este locația unde sunt salvate toate scripturile PHP și anume într-un nou fișier denumit **sitebd1**.(Figura 15.4)

Pentru a vizualiza aplicația creată, se va scrie în browserul de internet **localhost/sitebd1/index.php**. Dacă apare pagina din Figura 15.5, atunci înseamnă că aplicația a fost instalată corect.

Caracteristicile de bază pentru modul de afișare au fost scrise în site.css. De asemenea, deoarece header-ul și footer-ul rămân aceleași, s-au creat fișiere php individuale care sunt apelate apoi în fiecare dintre scripturile php folosite.

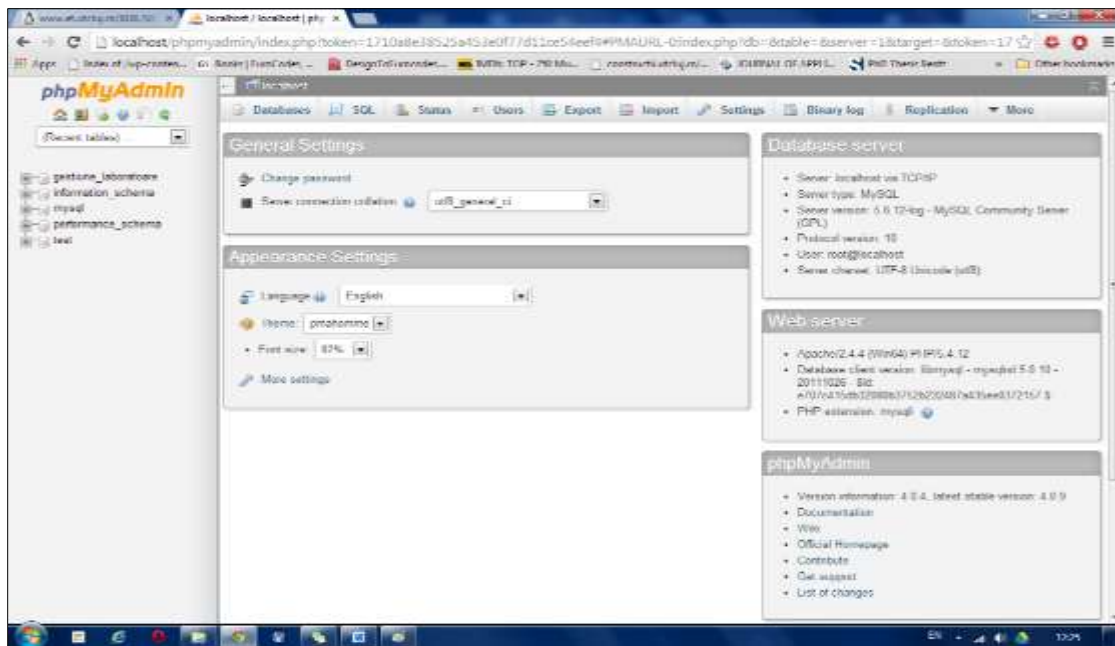


Fig.3. Interfața grafică pentru MySQL ce este pusă la dispoziție de către php.



Fig. 4.Fișiere create pentru aplicația web



Fig. 5. Afișarea ferestrei corespunzătoare index.php

Pagina care conține datele de contact va conține informații despre laboratoarele existente și va arăta precum în imaginea următoare:



Fig. 6. Pagina corespunzătoare about.php

Legătura la baza de date creată se face prin apelarea fisierului connection.php ce se află într-un nou fisier denumit includes si care conține codul:

```
<?php
$dbhost='localhost';
$dbuser='root';
$dbpass="";
$db='gestiune_laboratoare';
$conn=mysql_connect($dbhost,$dbuser,$dbpass);
mysql_select_db($db);
?>
```

În fiecare dintre fișierele care conțin o interogare asupra bazei de date se va scrie codul de verificare a conexiunii cu baza de date:

```
<?php  
include 'includes/connection.php';  
?>
```

Conexiunea la MySQL este verificată numai în momentul în care se selectează una dintre opțiunile din meniu, acesta fiind momentul în care se folosește pentru prima dată conectarea la baza de date. Pentru fiecare dintre meniurile care nu au fost amintite până acum s-a creat o pagină care va conține o afișare a înregistrărilor care sunt mai importante, dar și butoane care odată accesate vor face interogări asupra bazei de date și vor afișa informațiile cerute.

Exemplu:

Ca un exemplu vom lua tabelul responsabili. Fereastra de bază care se deschide la accesarea meniului responsabili conține atât informații despre responsabilii existenți și anume numele și prenumele acestora, dar și 3 butoane:

- primul buton afișează numele, prenumele, vârsta și adresa responsabililor și se numește **info_responsabili**
- cel de-al doilea buton va afișa numele și prenumele responsabililor, alături de denumire și descrierea funcțiilor acestora, ceea ce înseamnă că interogarea se va face asupra a 2 tabele din baza noastră de date și anume responsabili și funcții; acest buton este denumit **responsabili_cu_funcții**
- cel de al treilea buton denumit **responsabili_cu_laboratoare** va afișa numele și prenumele responsabililor, alături de denumirea laboratorului de care este răspunzătoare persoana afișată alături de adresa laboratorului amintit, interogarea făcându-se asupra a 2 tabele și anume laboratoare și responsabili



Fig. 7. Afișarea ferestrei corespunzătoare accesării meniului responsabili



Fig. 8. Afişarea ferestrei corespunzătoare accesării butonului **info_responsabili**



Fig. 9. Afişarea ferestrei corespunzătoare accesării butonului **responsabili_cu_functii**



Fig. 10. Afişarea ferestrei corespunzătoare accesării butonului **responsabili_cu_laboratoare**

Codul folosit pentru crearea acestei aplicații web se regăsește în Anexa 6. Baza de date creată se poate îmbunătăți prin implementarea unui site care să fie accesibil tuturor cadrelor didactice pentru o mai bună organizare a orelor de laborator. De asemenea, interpretarea datelor

poate duce la o statistică a funcționalității aparatelor pe o perioadă îndelungată de timp și astfel acest lucru poate duce la o îmbunătățire a aparaturii de laborator.