

CAP. 1

INTRODUCERE ÎN PROGRAMARE

Internetul ne oferă tot felul de informații care ne pot ajuta să înțelegem mai bine programarea. Totuși, unui începător de obicei îi este foarte greu să găsească informația cu adevărat utilă și să înțeleagă anumite lucruri. Uneori este nevoie să citești zeci, sute de materiale ca să îți dai seama cum funcționează de fapt lucrurile.

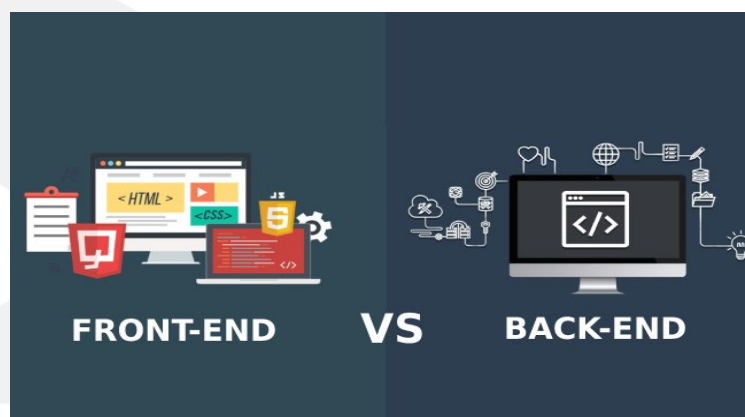
Scopul acestui curs este de a oferi informații bine structurate și ușor de înțeles pentru începători, aceștia având nevoie de mai puțin timp pentru a asimila cunoștințe decât dacă ar învăța pe cont propriu.

1.1 Care sunt părțile unei aplicații web?

O aplicație web are în general 2 componente: **front-end** și **back-end**.

Front-end

Front-end sau partea văzută de client (utilizator) este acea parte a site-ului sau a aplicației web, pe care vizitatorii o folosesc pentru a consuma conținut sau pentru a accesa informație. Aceasta include elemente precum elemente text, meniuri, butoane, tranziții, formulare, etc. De exemplu, un formular de autentificare poate fi vizibil vizitatorilor fără ca acesta să fie funcțional. Componenta de front-end este de obicei implementată cu ajutorul limbajelor **HTML**, **CSS** și **Javascript**.



Back-end

Back-end sau partea aplicației care este procesată pe server este acea parte care oferă funcționalități unui site. Aceasta de obicei are trei părți: un server, codul sursă al aplicației și o bază de date. Rolul principal al back-end-ului este cel de management de conținut, acesta gestionează modul în care sunt prezentate componentele și informațiile pentru un website, cum sunt administrate informațiile dintr-un magazin online, etc. De exemplu, un formular de autentificare care permite accesul într-o aplicație și care verifică dacă datele sunt corecte poate fi implementat cu ajutorul limbajelor de programare **PHP**, **Python**, **Java**, **C#** etc. și se folosește de baze de date **SQL** și/sau **NoSQL**.

1.2 Ce se întâmplă când accesezi o pagină web?

Ca să înțelegem ce se întâmplă de fapt în momentul în care un utilizator accesează o pagină web (proces redat în Figura 1.2), trebuie să cunoștem câteva noțiuni legate de ceea ce face un browser și care este protocolul care gestionează comunicația cu serverele web.

Web browser

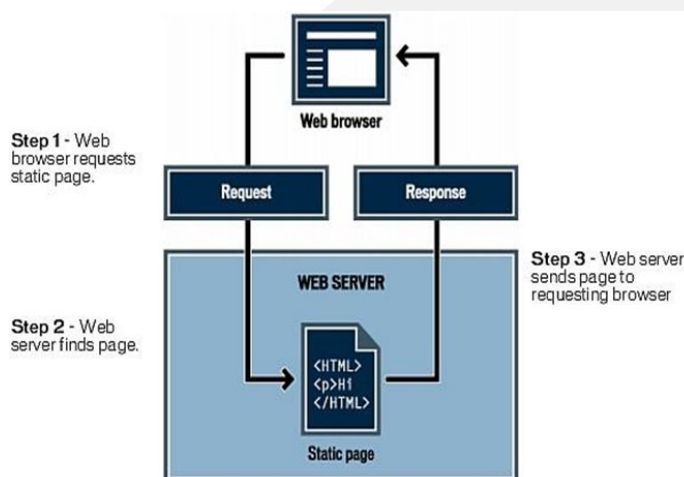
Este o aplicație ce permite utilizatorilor să afișeze text, imagini, video, sunet și alte informații ce se pot regăsi pe o pagină venită din Internet. Unele dintre cele mai cunoscute browsere sunt Google Chrome, Microsoft Internet Explorer, Mozilla Firefox, sau Apple Safari.

Web server

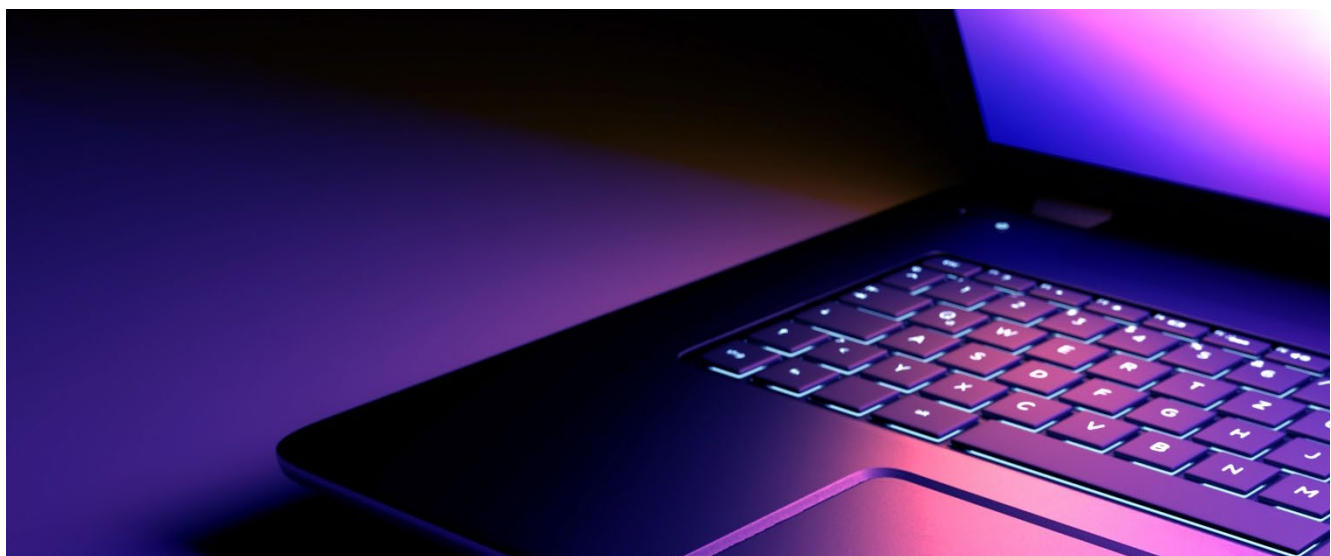
Serverul web (web server) este serverul fizic (un calculator specializat) care stochează (găzduiește) pagini web și le pune la dispoziția solicitanților prin protocolul Hypertext Transfer Protocol (HTTP).

HTTP

HTTP este protocolul cel mai des utilizat pentru accesarea informațiilor din Internet. Acesta conține o serie de reguli și mecanisme de comunicare prin care paginile web se pot transmite de la un server web aflat la distanță către clienți.



Pașii parcurși de protocolul HTTP pentru a furniza o pagină web unui client



Pasul 1

Utilizatorul, prin intermediul unui browser, vrea să acceseze o pagină web (de exemplu: <https://www.ateliereleilbah.ro/>). Browser-ul identifică adresa fizică a site-ului cerut folosind protocolul Domain Name System (DNS), și trimite un mesaj de tip HTTP către adresa IP a serverului (în cazul de față aceasta este **148.251.128.24** – vezi figura următoare).



The screenshot shows a web interface titled "Find Website IP". It contains a text input field with "ateliereleilbah.ro" and a blue button labeled "Find Website IP". Below the button, it says "The submitted website resolves to" followed by a box containing the IP address "148.251.128.24 (ateliereleilbah.ro)".

Rezoluție DNS realizată cu ajutorul site-ului <https://www.ipvoid.com>

Pasul 2 Serverul web identifică pagina dorită, imaginile, fișierele ce țin de acest website.

Pasul 3 Web serverul trimite aceste informații prin intermediul protocolului HTTP către client.

Pasul 4 Browser-ul primește datele, assemblează și afișează pagina.

1.3 Ce este un dezvoltator pentru back-end?

Un dezvoltator de back-end este responsabil de implementarea funcționalităților unei aplicații ce poate fi reprezentată de o aplicație web, aplicație pentru telefoane inteligente, sau o aplicație desktop.

Responsabilitatea unui dezvoltator de back-end este să creeze codul și programele care interacționează cu serverele, bazele de date și alte aplicații ale acestora. În calitate de programator web de back-end, cea mai mare calitate este abilitatea de a crea un program curat și eficient care face tot ce îți dorești în cel mai scurt timp.

Un programator back-end este responsabil și pentru partea de baze de date. O bază de date stochează toată informația necesară site-ului. De exemplu: conturile utilizatorilor, coșul de cumpărături în cazul unui magazin online, ș.a..

1.4 Ce este un dezvoltator full-stack?

Un dezvoltator full-stack este combinația dintre un dezvoltator de front-end și un dezvoltator de back-end. El este persoana responsabilă de implementarea unei aplicații web cap-coadă.

El se ocupă de implementarea designului (partea de client) sau tot ceea ce vede un utilizator cât și de implementarea componentei care va sta pe server (partea de pe server), a bazei de date și

tot el face și legătura dintre front-end, back-end și baza de date. O reprezentare a tehnologiilor ce trebuie cunoscute de un dezvoltator full-stack este oferită în imaginea de mai jos.



1.5 Ce este PHP, MySQL, Laravel?

În cadrul acestui curs veți învăța să folosiți limbajele **PHP**, **MySQL** și framework-ul **Laravel** pentru PHP, necesare în procesul de dezvoltare a aplicațiilor web.

PHP

Ca idee generală, PHP-ul reprezintă un limbaj de programare ce permite modificarea paginilor web înainte ca acestea să fie transmise de server către browser-ele utilizatorilor.

PHP a prins viață datorita lui Rasmus Lerdorf în 1995. A fost creat inițial pentru a face paginile mai dinamice, oferind un surplus de dinamism și funcționalități, fiind inițial conceput pentru magazinele online. După ce este interpretat de către server codul PHP este generat ca și cod HTML. Pentru interpretarea limbajului este necesară doar instalarea unui server (**Apache** sau **Nginx**).

Deși inițial PHP nu a fost creat pentru a deveni un limbaj de programare, acesta a devenit oficial un limbaj începând cu versiunea 3 fiind oficial lansat în iunie 1998, făcând trecerea de la un produs open-source 100% la unul dedicat și pentru Enterprise, precum Zend Engine în 1999.

În data de 22 mai 2000 a fost lansat oficial PHP 4, versiunea sa stabilă fiind lansată în August 2008.

Pe 14 iulie 2004 a fost lansată versiunea PHP 5, având și posibilitatea de a scrie cod POO (programare orientate pe obiecte), având ulterior și suport pentru PDO (Programming Data Object) fiind o extensie a programării pe obiecte. Această versiune a reprezentat un salt imens pentru limbaj deoarece, din acel moment a devenit mai mult decât un alt limbaj procedural, putând ține pasul cu limbaje mai complexe.

PHP 6 a avut ca scop adaptarea limbajului în sistemul ICU (International Components for Unicode) fiind creat în 2009 și dezvoltând-se până în martie 2010.

PHP 7 este versiunea majoră ce este dezvoltată până în prezent, fiind creată în 2014 și lansând prima versiune stabilă în 2015. Cele mai multe schimbări semnificative pentru PHP sunt în această versiune de limbaj, permițând ca totul să fie cât mai modern atunci când programăm folosind cod PHP.

PHP poate genera conținut HTML pe baza unor fișiere existente sau pornind de la zero, poate să afișeze o imagine sau orice alt conținut accesibil prin web, sau să redirectioneze utilizatorul către alte pagini. În cadrul acestui proces, PHP poate consulta baze de date, fișiere externe sau alte resurse, poate trimite email-uri sau executa comenzi ale sistemului de operare. Întrucât procesarea se realizează la nivelul serverului web, înainte ca paginile web să ajungă în browser, PHP este considerat un limbaj de programare server-side.

În trecut, PHP era echivalent cu dezvoltarea web. Cunoscut și ca limbajul Internetului, și acum este folosit într-o formă sau alta de către 80% dintre site-uri.

PHP este ușor de folosit și de învățat, fiind foarte popular printre programatorii web de back-end aflați la început, dar și în rândul site-urilor mici, oferind astfel multe oportunități de muncă de tip freelance pentru cei care cunosc PHP.

MySQL

MySQL este un sistem de gestiune a bazelor de date, având propria structură, funcții și cuvinte cheie pentru prelucrarea datelor. MySQL poate crea și gestiona baze de date (cum ar fi informații despre angajați, inventar produse și multe altele).

MySQL este o bază de date relațională. Ce înseamnă asta? Datele găzduite sunt interconectate prin intermediul unor relații stocate în baza de date. Fiecare bază de date conține tabele. Fiecare tabelă conține una sau mai multe categorii de date stocate în coloane. Fiecare rând (denumit, de asemenea, o înregistrare) conține o informație unică (identificată printr-o cheie) pentru categoriile definite în coloane.

Laravel

Majoritatea site-urilor, pe lângă pagini, conțin o serie de module software care îndeplinesc aceleași sarcini tipice: lucrul cu baza de date, afișarea paginilor bazată pe șabloane, controlul accesului, protecția împotriva atacurilor și altele. Și din moment ce astfel de sarcini sunt aceleași în toate site-urile, puteți să nu scrieți aceste module de fiecare dată, dar să utilizați un produs software care le are încorporate. Astfel de produse software sunt framework-urile.

Laravel a devenit cel mai popular framework gratuit (open-source) de PHP din lume. Deoarece poate trata aplicații web complexe în siguranță, într-un ritm considerabil mai rapid decât alte framework-uri. Laravel simplifică procesul de dezvoltare prin ușurarea sarcinilor obișnuite, cum ar fi rutarea, gestionarea sesiunilor, memorarea în cache sau autentificarea.

1.6 Instrumente software necesare

Pentru a crea un fișier PHP avem la dispoziție 2 metode:

1. Folosirea unui mediu de dezvoltare sau editor IDE (Integrated Development Environment). Acesta este o aplicație desktop cu interfața grafică care ușurează munca prin numeroasele funcții pe care le oferă. Ne ajută să fim mai eficienți prin funcțiile de auto completare a codului, putem vedea mai ușor greșelile și erorile, pe lângă multe alte avantaje pe care le veți observa pe parcursul cursului.
2. Prin folosirea editorului text pus la dispoziție de sistemul de operare. Această metodă nu prea mai este folosită în ziua de azi dar este bine de știut.

Pentru ambele metode, avem diferite alternative gratuite dar și cu plată, cele mai populare fiind:

- Visual Studio Code (VsCode) - <https://code.visualstudio.com/>
- NotePad ++ - <https://notepad-plus-plus.org/downloads/>
- Sublime - <https://www.sublimetext.com/>

Cu plata:

- PhpStorm - <https://www.jetbrains.com/phpstorm/>
- WebStorm - <https://www.jetbrains.com/webstorm/>

Observație: Pentru acest curs vom folosi și implicit va trebui să instalăm următoarele aplicații:

1. Un editor de cod (IDE), VsCode;
2. Un browser (Chrome sau Firefox);
3. Un mediu de dezvoltare local (Laragon), ce conține aplicațiile de tip server necesare pentru găzduirea aplicațiilor dezvoltate.

1.7 Instalare PHP

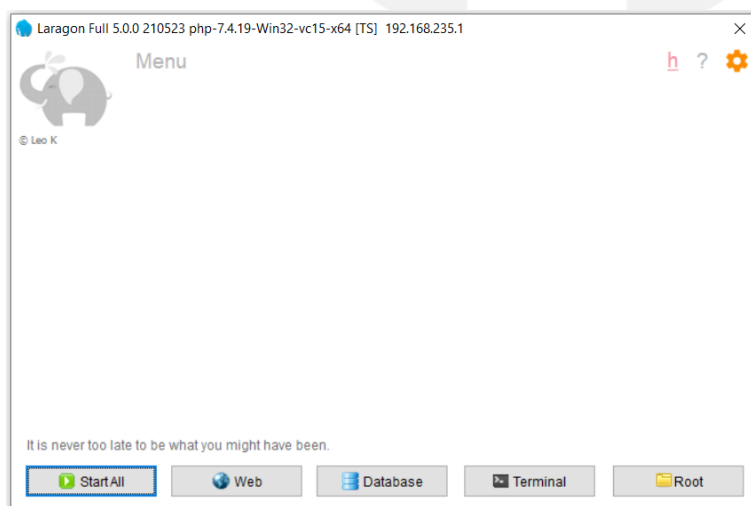
Pentru a putea rula codul de PHP avem nevoie de acces la un server. Avem posibilitatea de a lucra pe un server local sau pe un server aflat la distanță. În acest curs vom folosi un server local.

Majoritatea programatorilor prefera Xampp sau Wamp atunci când este vorba de instalarea unui server local pe Windows, dar noi vom folosi **Laragon**.

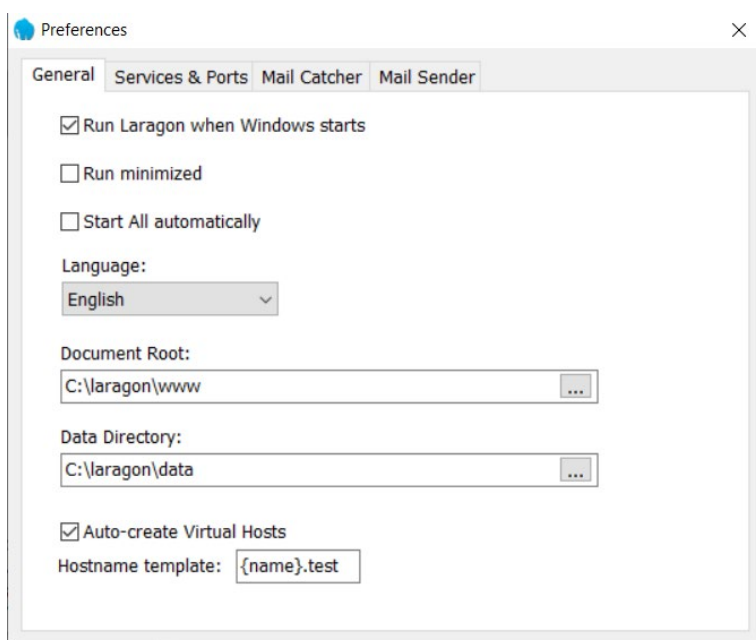
Pasul 1 Descărcați pachetul software pentru Laragon. Îl puteți obține accesând <https://laragon.org/download/>.

Pasul 2 Instalați Laragon.

Pasul 3 După ce instalăm programul, îl deschidem și dăm click pe roțița din partea dreaptă sus (evidențiată în Figura 1.5) pentru a afla calea în care va trebui să plasăm codul PHP pentru a fi procesat de către serverul web.

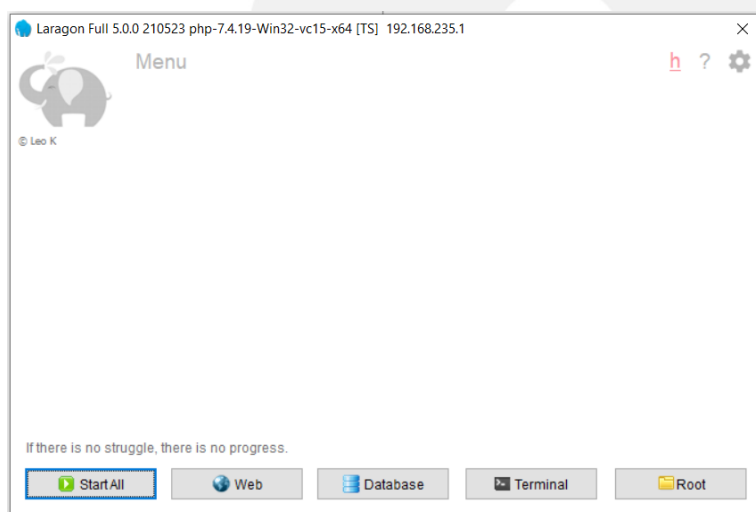


Accesarea meniului de configurare

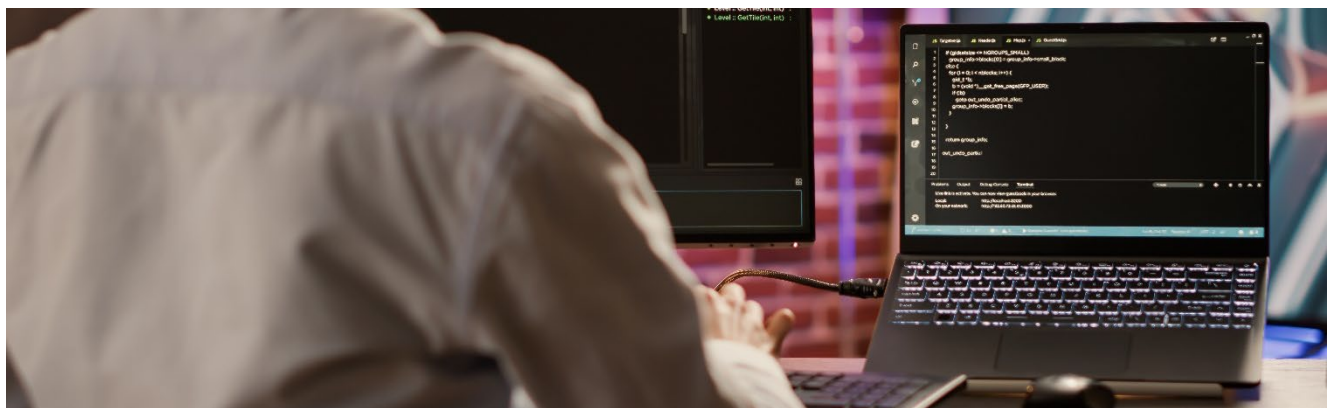


Locația în care va trebui să punem codul ce va fi interpretat de server

Pasul 5 Pornim serverele (Apache și MySQL) folosind butonul evidențiat în figura următoare.



Pornirea serverelor Apache și MySQL



1.8 Prima aplicație PHP

Creați un fișier index.php cu următorul conținut și plasați-l în directorul identificat la Pasul 4 din subcapitolul anterior:

```
<!DOCTYPE html>
<html>

<body>

    <h1>My first PHP page</h1>

    <?php
    echo "Hello World!"; // first line

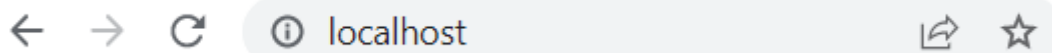
    ?>

</body>

</html>
```

Observație: Este posibil să existe deja un fișier **index.php** în directorul respectiv. În acest caz va trebui să înlocuiți fișierul cu cel creat de voi.

După ce v-ați asigurat că ați pornit în prealabil serverul **Apache** (**Pasul 5** din subcapitolul anterior), deschideți un browser și tastați în caseta de URL: **localhost**.



Caseta URL în care apare domeniul DNS al site-urilor accesate

Dacă toți pașii au fost îndepliniți, ar trebui să fiți întâmpinați de următoarea pagină web.

My first PHP page

Hello World!

Prima pagină web generată cu ajutorul PHP