UNIVERSITATEA “LUCIAN BLAGA” DIN SIBIU

FACULTATEA DE INGINERIE

DEPARTAMENTUL DE CALCULATOARE ŞI INGINERIE ELECTRICĂ

**PROIECT DE DIPLOMĂ**

Conducător științific : Crețulescu Radu

Îndrumător: Crețulescu Radu

Absolvent: Popa Doroteea Cristina

Specializarea: Tehnologia Informației

* Sibiu, 2024 –

UNIVERSITATEA “LUCIAN BLAGA” DIN SIBIU

FACULTATEA DE INGINERIE

DEPARTAMENTUL DE CALCULATOARE ŞI INGINERIE ELECTRICĂ

PLICAȚIE GESTIONARE DE PROIECTE DEP.CIE

Conducător științific: Crețulescu Radu

Îndrumător: Crețulescu Radu

Absolvent: Popa Doroteea Cristina

Specializarea: Tehnologia Informației

Plan tematic

CUPRINS

[1. Prezentarea temei 5](#_Toc169038568)

[1.1. Introducere 5](#_Toc169038569)

[1.2. Descrierea temei 5](#_Toc169038570)

[1.3. Scop si obiective 5](#_Toc169038571)

[1.4. Tehnologii utilizate 5](#_Toc169038572)

[1.5. Cazuri principale de utilizare a aplicației 5](#_Toc169038573)

[1.6. Organizarea lucrării 5](#_Toc169038574)

[2. Considerații teoretice 5](#_Toc169038575)

[2.1. PHP și PhpMyAdmin 5](#_Toc169038576)

[2.1.1. Introducere 5](#_Toc169038577)

[2.1.2. Caracteristici cheie ale limbajului PHP 6](#_Toc169038578)

[2.1.3. PHP în dezvoltarea web 7](#_Toc169038579)

[2.1.4. Securitatea 7](#_Toc169038580)

[2.1.5. Instrumente și resurse pentru dezvoltare 8](#_Toc169038581)

[2.1.6. PhpMyAdmin 8](#_Toc169038582)

[2.2. SQL și MySQL 9](#_Toc169038583)

[2.2.1. Introducere 9](#_Toc169038584)

[2.2.2. Structură și funcționalități 10](#_Toc169038585)

[2.2.3. Exemple de utilizare 10](#_Toc169038586)

[2.3. HTML 11](#_Toc169038587)

[2.3.1. Introducere 11](#_Toc169038588)

[2.3.2. Structură de bază a unui document 11](#_Toc169038589)

[2.3.3. Elemente HTML 12](#_Toc169038590)

[2.3.4. Practici de utilizare HTML 12](#_Toc169038591)

[2.4. CSS 13](#_Toc169038592)

[2.4.1. Introducere 13](#_Toc169038593)

[2.4.2. Sintaxa și selectoarele CSS 14](#_Toc169038594)

[2.4.3. Proprietăți CSS esențiale 15](#_Toc169038595)

[2.4.4. Practici de utilizare CSS 16](#_Toc169038596)

[2.5. XAMPP 17](#_Toc169038597)

[2.5.1. Introducere 17](#_Toc169038598)

[2.5.2. Instalarea și configurarea XAMPP 17](#_Toc169038599)

[2.5.3. XAMPP în dezvoltarea și testarea aplicațiilor web 18](#_Toc169038600)

# Prezentarea temei

## Introducere

## Descrierea temei

## Scop si obiective

## Tehnologii utilizate

## Cazuri principale de utilizare a aplicației

## Organizarea lucrării

# Considerații teoretice

# 

## PHP și PhpMyAdmin

### 2.1.1. Introducere

PHP (Hypertext Preprocessor) este un limbaj de scripting server-side, aceasta însemnând ca rulează la nivel de server, nu la nivel de client (browser). Acest limbaj e folosit în principal în dezvoltarea web. Acest limbaj de programare a fost creat de Rasmus Lerdorf în anul 1994 si este unul dintre cele mai utilizate limbaje in dezvoltarea web. PHP este preferat de către programatori deoarece oferă flexibilitate, este relativ simplu de utilizat și oferă posibilitatea de a lucra cu diferite baze de date, astfel asigurând dezvoltarea unor pagini web care oferă siguranță cibernetică si dinamicitate.

Unul dintre motivele pentru care PHP este un limbaj popular este faptul ca este open-source, acest lucru facilitând accesarea și modificarea codului sursă. Acest lucru a determinat formarea unei comunități de utilizatori care au îmbunătățit constant capacitățile PHP. De asemenea, PHP poate rula pe mai multe sisteme de operare, precum Linux, macOS si Windows si e compatibil cu servere web importante, cum ar fi Apache și Nginx.

Un alt avantaj al PHP-ului este sintaxa simplă, ușor de învățat, fiind accesibil pentru persoanele deja familiarizate cu limbaje de programare precum C, Java, Perl sau începătorilor care vor să învețe dezvoltare web. PHP oferă un proces simplificat de dezvoltare prin manipularea formularelor, operațiuni cu fișiere și gestionare de sesiuni.

### 2.1.2. Caracteristici cheie ale limbajului PHP

PHP are diverse caracteristici care îl fac să fie un limbaj atractiv pentru programatori. Câteva caracteristici importante a PHP-ului ar fi:

* Simplicitate și ușurință în învățare:

Sintaxa PHP e asemănătoare cu a altor limbaje de programare cunoscute (C, Perl), fiind astfel ușor de învățat pentru cei cu cunoștințe anterioare despre aceste limbaje. Sintaxa e simpla, ușor de învățat până și pentru persoanele care abia încep să învețe programare.

* Performanță și viteză:

PHP este un limbaj interpretat la nivel de server, aceasta însemnând că serverul execută codul PHP înainte de trimiterea paginii către client. Astfel, conținutul e generat dinamic, în timp real, fără un efect negativ semnificativ asupra performanței sau vitezei.

* Suport extins pentru baze de date:

PHP poate fi folosit ușor cu majoritatea sistemelor de gestionare a bazelor de date. Printre ele se regăsesc MySQL, PostgreSQL, Oracle si multe altele. Aceasta oferă dezvoltatorilor flexibilitate în alegerea bazei de date care e cea mai potrivită cu nevoile lor.

* Gestionarea sesiunilor:

Este o caracteristică esențială în dezvoltarea aplicațiilor web care folosesc autentificarea și păstrarea stării utilizatorilor. PHP oferă un mecanism simplu pentru a gestiona sesiunile, permițând urmărirea și menținerea informațiilor despre utilizatori pe parcursul mai multor pagini.

* Manipularea erorilor:

PHP conține funcții integrate pentru gestionare de excepții și erori. Dezvoltatorii pot folosi blocuri try-catch pentru gestionarea eficientă a excepțiilor și pot seta niveluri diferite pentru raportarea erorilor.

* Suport pentru programe orientate pe obiecte:

Dezvoltatorii care folosesc PHP își pot organiza și structura codul într-un mod modular și reutilizabil cu ajutorul programării orientate pe obiecte, suportată de limbaj. Astfel aceștia pot crea clase, obiecte, moștenire, polimorfism și se pot folosi de alte concepte OOP pentru a îmbunătății claritatea codului, performanța și viteza.

Alte caracteristici importante a PHP-ului ar fi portabilitatea și compatibilitatea cu diverse sisteme de operare (Linux, Windows, macOS), actualizări și îmbunătățiri continue de performanță de către o comunitate mare de dezvoltatori, precum și corecții de securitate, asigurând ca PHP rămâne un limbaj performant și relevant.

Toate aceste caracteristici determină popularitatea și accesibilitatea PHP-ului, fiind considerat și în prezent un limbaj adecvat pentru dezvoltarea aplicațiilor web interactive și dinamice.

### 2.1.3. PHP în dezvoltarea web

PHP este un limbaj important în dezvoltarea de aplicații web datorită scalabilității, flexibilității și posibilității oferite de a crea aplicații web dinamice și interactive. Cu ajutorul PHP se pot construi aplicații web sigure, rapide și ușor de întreținut datorită evoluții constante a limbajului. PHP facilitează diferite aspecte ale dezvoltării web, cum ar fi: generarea de conținut dinamic, integrarea cu baze de date, utilizarea framework-urilor PHP, resurse PHP, securitate și suport pentru servicii web.

Spre deosebire de HTML care este static, paginile PHP au posibilitatea de a genera conținut personalizat în mod automat prin afișarea informațiilor dintr-o bază de date, personalizarea mesajelor pentru utilizatori sau prin actualizarea conținutului paginilor in timp real. Toate acestea contribuie la utilitatea și interactivitatea site-urilor web create folosind PHP.

Un framework PHP este creat cu ajutorul limbajului PHP care oferă caracteristici integrate care îmbunătățesc experiența unui dezvoltator software. Acestea ajută la creșterea vitezei de dezvoltare și a performanței în general. Câteva dintre framework-urile populare sunt: Symfony, Laravel și Codelgniter. Fiecare framework se focusează pe diferite aspecte, unele se focusează pe îmbunătățirea fluxului de lucru, pe când altele țin cont mai mult de oferirea scalabilității și a securității.

### 2.1.4. Securitatea

Un aspect important în dezvoltarea aplicațiilor web îl reprezintă securitatea pentru asigurarea securității PHP oferă mecanisme de protejare a aplicațiilor și de diminuare a vulnerabilităților. Pentru asigurarea securității aplicațiilor web e necesară o bună cunoștință a acestor mecanisme și aplicarea diferitelor măsuri de securitate.

Una dintre metodele de a asigura securitatea o reprezintă folosirea unor funcții precum filter\_input() și filter\_var(), cu ajutorul cărora programatorii pot verifica și valida datele primite de la utilizatori. Prin validarea și curățarea datelor de intrare se pot preveni atacuri de tip injecție sau exploatări bazate pe input rău intenționat. Injecțiile SQL sunt o tehnică comună de atac al aplicațiilor web. Pentru a le putea evita e nevoie ca programatorii să implementeze interogări care să asigure faptul că datele primite de la utilizatori sunt tratate ca parametri, nu ca parte a comenzii SQL. Acest lucru se poate realiza prin intermediul extensiilor PDO (PHP Data Objects) sau MySQLi, suportate de limbajul PHP.

Alte metode importante în asigurarea securității sunt gestionarea sigură a sesiunilor și folosirea de funcții de criptare și decriptare. Gestionarea sigură a sesiunilor e necesară pentru integritatea și protecția datelor utilizatorilor. PHP conține mecanisme integrate pentru a simplifica gestionarea sesiunilor și funcții cu ajutorul cărora se pot crea, distruge și regenera ID-uri de sesiune (exemplu: session\_start(), unset()). Funcția session\_regenerate\_id() poate fi folosită pentru a preveni atacuri de tip hijack al sesiunii. De asemenea, PHP pune la dispoziția programatorilor funcții pentru criptarea și decriptarea datelor cu caracter sensibil (exemplu: password\_hash() și password\_verify()). Aceste funcții sunt folosite pentru criptare și verificarea parolelor în siguranță, nepermițând accesul neautorizat.

Prin folosirea acestor metode se asigură dezvoltarea unor aplicații web robuste și sigure, protejate de atacuri cibernetice și care oferă integritatea datelor utilizatorilor.

### 2.1.5. Instrumente și resurse pentru dezvoltare

Dezvoltarea aplicațiilor web cu ajutorul PHP se poate realiza cu ajutorul unui set de instrumente și resurse ce simplifică procesul de scriere și gestionare a codului. Aceste instrumente includ editoare de cod IDE-uri, biblioteci și resurse online.

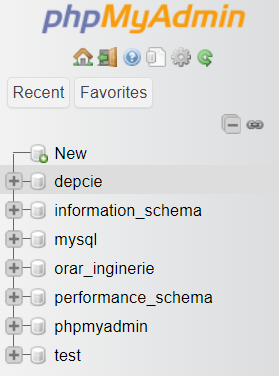
Printre cele mai populare editoare și IDE-uri se numără PHPStorm și Visual Studio Code. PHPStorm e un IDE dezvoltat de JetBrains, special pentru lucrul cu PHP, suportând framework-uri PHP, precum Laravel și Symfony. PHPStorm pune la dispoziție diferite facilități precum debugging ușor de realizat, integrarea bazelor de date, suport pentru testare și posibilitatea de refactorizare a codului și instrumente integrate de control al versiunilor (GIT, SVN). De asemenea, Visual Studio Code e un editor popular datorită flexibilității și numărului de extensii oferite. VS Code este un editor creat de Microsoft cu suport pentru PHP și multe alte limbaje de programare. La fel ca PHPStorm, VS Code are instrumente integrate care facilitează folosirea Git.

Un plus al folosirii PHP îl reprezintă comunitatea extinsă si resursele online disponibile. PFP.net este site-ul oficial al limbajului PHP care oferă o gamă largă de resurse, ghiduri, documentație cu exemple de cod și specificații detaliate care simplifică experiența dezvoltatorilor experimentați sau începători. Un beneficiu este, bineînțeles, faptul că documentația este actualizată constant și secțiunea de comentarii care permit utilizatorilor să își ofere feedback-ul și să contribuie la corectarea sau îmbunătățirea conținutului.

### 2.1.6. PhpMyAdmin

PhpMyAdmin este un instrument software open-source făcut pentru a ușura gestionarea și administrarea bazelor de date MySQL și MariaDB. Această platformă este realizată în principal cu ajutorul PHP. Popularitatea sa se datorează interfeței sale prietenoase și ușor de utilizat care îl face un tool ideal pentru programatorii începători sau avansați.

PhpMyAdmin oferă o interfață grafică intuitivă pentru crearea și gestionarea bazelor de date; ]n partea stângă a platformei sunt poziționate bazele de date cu butonul de “new” pentru crearea unei noi baze de date (fig. 1.1), iar în partea de sus avem un meniu cu diferitele funcții oferite de PhpMyAdmin (fig. 1.2).



(fig. 1.1)



(fig. 1.2)

Utilizatorii pot executa interogări SQL direct din platforma PhpMyAdmin folosind butonul “SQL”. Acest lucru este foarte avantajos pentru crearea, modificarea și ștergerea datelor din baza de date, oferind un mod rapid de a interacționa cu baza de date. O altă caracteristică importantă o reprezintă suportul integrat pentru import și export al datelor în diferite formate, precum, CSV, SQL, JSON și XML. Aceasta asigură o bună realizare de backup-uri și integrarea datelor cu alte aplicații. PhpMyAdmin oferă și o funcționalitate de “designer” care e folosită pentru vizualizarea și gestionarea grafică a tabelelor din baza de date. Aceasta permite vizualizarea tabelelor și a relațiilor dintre ele și a conectării prin cheile străine. Utilizatorii pot genera și diagrame PDF ale structurii bazei de date, acest lucru fiind util în realizarea unei documentații sau pentru partajarea acesteia.

## SQL și MySQL

### 2.2.1. Introducere

MySQL este un sistem de gestionare al bazelor de date relaționale, popular printre dezvoltatorii de aplicații web. A fost dezvoltat de MySQLAB dar este în prezent deținut de Oracle Corporation. Este folosit pe scară largă deoarece oferă diferite beneficii, precum scalabilitate, performanță ridicată și flexibilitate, fiind ideal pentru o gestionare eficientă de date. Performanța ridica se datorează vitezei și eficienței în gestionarea volumelor mari de date care se pot extinde pe mai multe servere. Un alt avantaj îl reprezintă suportul pentru SQL pentru facilitarea manipulării și controlului de date.

SQL (Structured Query Language) este un limbaj standardizat de programare folosit în gestionarea bazelor de date. A fost creat de IBM în 1970, iar de atunci popularitatea sa a crescut, devenind standard pentru manipularea bazelor de date relaționale. SQL are o sintaxă relativ ușor de învățat. Sintaxa este declarativă, utilizatorii putând specifica acțiunile dorite asupra bazei de date fără a descrie detaliat modul de execuție al operației.

### 2.2.2. Structură și funcționalități

Structura MySQL e compusă din următoarele componente: serverul MySQL, motoare de stocare, clientele MySQL și biblioteci API. Serverul MySQL e nucleul care se ocupă de operațiile legate de baza de date, conexiunile utilizatorilor și gestionarea interogărilor. MySQL suportă diferite motoare de stocare, printre care InnoDB (implicit) și MyISAM (rapid, folosit mai ales pentru tabele care nu necesită tranzacții). Clientele MySQL sunt aplicații care fac posibilă interacțiunea dintre utilizatori și serverul MySQL (MySQL Command Line Client, MySQL Workbench). Bibliotecile API sunt oferite de MySQL pentru limbaje de programare PHP, Phyton, Java și altele, acestea ajutând în gestionare bazelor de date MySQL.

MySQL oferă funcționalități necesare în dezvoltarea de aplicații web. MySQL permite crearea, ștergerea, modificarea bazelor de date și inserarea, interogarea, actualizarea și ștergerea datelor din tabelele din baza de date . MySQL oferă posibilitatea de a gestiona utilizatorii și permisiunile oferite acestora.

### 2.2.3. Exemple de utilizare

Pasul inițial în folosirea MySQL este crearea unei baze de date si a tabelelor conținute de aceasta. Aceasta se poate realiza folosind comenzile SQL:

CREATE DATABASE magazinOnline;

USE magazinOnline;

CREATE TABLE produse\_din\_magazin (

id INT AUTO\_INCREMENT PRIMARY KEY,

nume VARCHAR(200),

preț INT,

stoc INT

);

(Ex. 1.1.)

Folosind aceste comenzi, am creat baza de date magazinOnline care conține tabelul produse\_din\_magazin. Tabelul conține un id, care e setat ca cheie primară care e incrementat automat și alte trei atribute care descriu produsele din magazin: nume, preț și stoc.

O altă operație des folosită este inserarea de date în tabele. În exemplul următor sunt inserate produsele telefon, cu prețul de 2000 și stocul de 35 și televizor cu prețul de 2500 și stocul de 20:

INSERT INTO produse\_din\_magazin(nume, preț, stoc) VALUES (‘telefon’, 2000, 35);

INSERT INTO produse\_din\_magazin(nume, preț, stoc) VALUES (‘televizor’, 2500, 20);

(Ex. 1.2.)

Alte operații importante în gestionarea bazelor de date sunt selectarea(SELECT), actualizarea(UPDATE) și ștergerea(DELETE) de date din tabele. Acestea sunt reprezentate în următorul exemplu unde selectăm toate produsele din tabelul de produse\_din\_magazin, actualizăm prețul telefonului la 2200 și ștergem televizorul:

SELECT \* FROM produse\_din\_magazin;

UPDATE produse\_din\_magazin SET preț = 2200 WHERE nume = ‘telefon’;

DELETE FROM produse\_din\_magazin WHERE nume = ‘televizor’;

(Ex. 1.3.)

## HTML

### 2.3.1. Introducere

HTML, prescurtare pentru HyperText Markup Language, este limbajul standard utilizat pentru crearea și structurarea paginilor web. Folosind "tag-uri", HTML permite dezvoltatorilor să organizeze și să prezinte conținutul pe internet într-un mod structurat. Inventat de Tim Berners-Lee în 1991, HTML a început cu doar 18 elemente. De atunci, a evoluat semnificativ, cu actualizări importante precum HTML 2.0 în 1995, HTML 3.2 în 1997 care a adus suport pentru tabele, HTML 4.01 în 1999 care a introdus suport pentru scripturi și HTML5 în 2014 care a adăugat capabilități pentru audio, video și elemente grafice avansate.

### 2.3.2. Structură de bază a unui document

Un document HTML are o structura standard, cu elemente specifice care definesc aspectul paginii web. La această structură se adaugă ulterior elemente noi în funcție de dorințele și cerințele dezvoltatorilor web. Structura unui document este afișată și descrisă mai jos.



Tipul documentului și versiunea HTML este declarată folosind <!DOCTYPE html>. Tagul <html> este elementul rădăcină, care conține toate celelalte elemente HTML. Tagul <head> include metadate despre document, ca de exemplu titlu și link-uri către resurse externe. Tagul <title> e folosit pentru a da titlul browserului, <meta charset=”UTF-8”> e utilizat pentru a specifica standardul de codare a caracterelor și în interiorul tagurilor <body> e structurat conținutul efectiv al paginii web (text, imagini, link-uri și alte elemente).

### 2.3.3. Elemente HTML

Elementele HTML se referă la blocurile de bază ale unui document HTML. Fiecare bloc este creat folosind un tag de deschidere (< >) și unul de închidere (</ >) și conținutul este plasat între aceste taguri. Titlurile pot fi specificare, folosind tagurile ‘h1’ până la ‘h6’ (exemplu: <h1> Titlu</h1>). Un paragraf se poate defini prin tagul ‘p’ (<p>Paragraful respectiv</p>). Listele ordonate și neordonate pot fi reprezentate cu tagurile ‘ul’, respective ‘ol’ și ‘li’ definește un element dintr-o listă. Listele sunt elemente importante a HTML, putând fi utilizate în construcția meniului unei pagini web. Alte elemente HTML importante sunt link-rile (<a href=”https://www.link.com”>link.com</p>) și imaginile (<img src=”imagine.jpg” alt=”descriere imagine” >).

Atributele au și ele un rol important în HTML: ‘href’ specifică URL-ul care conduce către link, ‘src’ specifică sursa imaginii, ‘alt’ e folosit pentru a oferi un text alternativ afișat dacă nu se poate încărca imaginea dorită, ‘class’ și ‘id’ sunt folosite pentru a adăuga stil și identificator unic elementelor HTML.

### 2.3.4. Practici de utilizare HTML

Pentru a crea cod ușor de înțeles și menținut, există diferite practici bune care se utilizează. Structurarea clară a codului e absolut necesară pentru a obține un astfel de cod. Utilizarea de indentare corespunzătoare și structurarea codului HTML în mod logic, cu subsecțiuni clare sunt tehnici apreciate. O altă practică comună este utilizarea tag-urilor semantice, precum “<header>”, “<nav>”, “<footer>”, “<section>”, “<article>”, acestea îmbunătățind claritatea și accesibilitatea codului HTML. Un use case al acestora e reprezentat in Ex. 1.4.



(Ex. 1.4.)

Alte practici aplicate sunt minimizarea codului HTML prin eliminarea spațiilor și comentariilor nefolositoare. Aceasta duce la reducerea fișierului HTML și ca urmare, la optimizarea și creșterea performanței.

Toate acestea fac din HTML un limbaj fundamental în dezvoltarea de aplicații web. Aplicarea acestor practici e esențială în crearea de site-uri web eficiente, moderne, accesibile și interactive.

## CSS

### 2.4.1. Introducere

CSS (Cascading Style Sheets) este un limbaj de stil utilizat pentru a descrie aspectul unui document HTML. CSS facilitează gestionarea paginilor web deoarece permite programatorilor să separe conținutul paginilor de aspectul acestora. Prima versiune CSS a fost creată în 1996 de către W3C. Acesta conținea elemente de aspect fundamentale, precum culori, fonturi, margini. În anul 1998 a apărut versiunea CSS2, care a introdus funcționalități avansate, cum ar fi, poziționarea relativă și absolută. Versiunea CSS2.1, apărută în 2011, a corectat unele erori ale versiunii anterioare și a îmbunătățit specificațiile. Versiunea CSS3, care se află în curs de dezvoltare, permite dezvoltarea mai rapidă a funcționalităților, prin aducerea de îmbunătățiri semnificative. Această versiune include selectoare avansate, efecte de tranziții și animații.

### 2.4.2. Sintaxa și selectoarele CSS

CSS are o sintaxă simplă, creată din selectoare și declarații. Declarațiile sunt formate din proprietăți și valori. Această structură este reprezentată în exemplul Ex. 1.1.

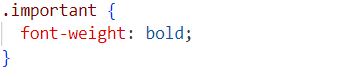
Selectoarele CSS sunt folosite pentru a selecta elementele HTML pe care dezvoltatorul dorește să le stilizeze. Există diverse selectoare CSS care sunt utilizate în funcție de ceea ce se dorește a fi selectat. Selectoarele de tip sunt folosite pentru a selecta toate elementele de tipul respectiv (Ex. 1.2.). Selectoarele de clasă selectează toate elementele cu o anumită clasă (Ex. 1.3.). Selectoarele de ID selectează un element unic cu un anumit ID (Ex. 1.4.). Selectoarele de atribut selectează elementele pe baza unui atribut (Ex. 1.5.). Selectoarele de descendență selectează elemente pe baza relațiilor lor cu alte elemente (Ex. 1.6.). Pseudo- clasele selectează elementele pe baza uni anumit statut sau condiție (Ex. 1.7.). Pseudo-elementele sunt utilizate pentru a stiliza anumite părți dintr-un element (Ex. 1.8.).



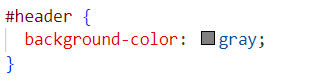
(Ex.1.1)



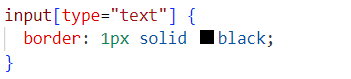
(Ex. 1.2.)



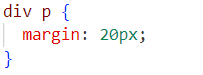
(Ex. 1.3.)



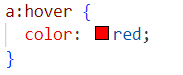
(Ex. 1.4.)



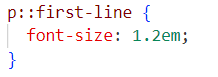
(Ex. 1.5.)



(Ex. 1.6.)



(Ex. 1.7.)

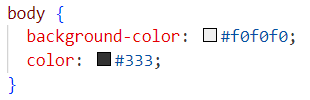


(Ex. 1.8.)

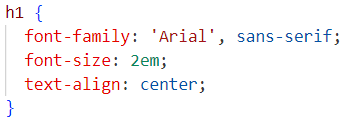
### 2.4.3. Proprietăți CSS esențiale

Câteva dintre cele mai importante proprietăți CSS folosite în stilizarea paginilor web sunt setarea de culori și fundaluri (Ex. 1.9.), fonturi și text (Ex. 2.1.), margini și spațieri (Ex. 2.2.), borduri și umbre (Ex. 2.3.), afișare și vizibilitate (Ex. 2.4.).

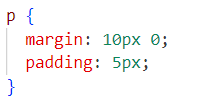
Pentru setarea de culori și fonturi se folosesc următoarele : ‘body’ cu care se setează culoarea textului unui element, ‘background-color’ setează culoarea de fundal al unui element, ‘background-image’ adaugă o imagine de fundal al unui element. Pentru setarea de fonturi și text sunt utilizate proprietățile: ‘font-family’ care specifică familia de fonturi pentru text, ‘font-size’ setează dimensiunea textului, ‘font-weight’ controlează grosimea fontului, ‘text-align’ aliniază textul în interiorul unui element, ‘text-decoration’ adaugă sau elimină decorarea textului. Pentru setarea de margini și spațieri se folosesc proprietățile: ‘margin’ setează spațiul exterior al unui element, ‘padding’ setează spațiul interior al elementelor. Pentru setarea de bordurilor și a umbrelor se folosesc: ‘border’ setează toate proprietățile bordurii (lățime, stil, culoare etc.), ‘border-radius’ setează colțurile rotunjite ale unui element, ‘box-shadow’ adaugă o umbra unui element. Afișarea și vizibilitatea sunt gestionate cu ‘display’ care controlează modul în care este afișat un element și ‘visibility’ care controlează vizibilitatea unui element.



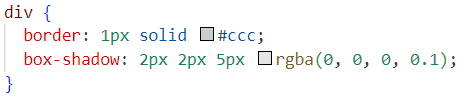
(Ex. 1.9.)



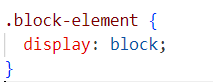
(Ex. 2.1.)



(Ex. 2.2.)



(Ex. 2.3.)



(Ex. 2.4.)

### 2.4.4. Practici de utilizare CSS

Pentru a asigura obținerea unui cod clar, eficient și ușor de gestionat este esențial să se urmeze anumite practici și tehnici bine definite. De asemenea, utilizarea de framework-uri CSS ajută la îmbunătățirea procesului de dezvoltare a aplicațiilor web interactive și cu un aspect modern.

O practică bună des întâlnită de organizare a codului CSS este păstrarea codului CSS in fișiere separate de cele de HTML. Aceasta determină obținerea de cod reutilizabil și ușor de întreținut și totodată îmbunătățește performanța, reducând timpul de încărcare a paginilor. O altă tehnică apreciată este utilizarea comentariilor pentru a oferi explicații suplimentare unde este cazul. Comentariile ajută atât programatorii noi care doresc să continue sau să modifice codul, cât și autorul codului. Utilizarea de clase și ID-uri conferă reutilizabilitate și lizibilitate codului, acestea putând fi aplicate pentru mai multe elemente.

Bootstrap este un framework des folosit, creat de Twitter. Acest framework oferă un set mare de componente predefinite și flexibilitate care simplifică procesul de dezvoltare a site-urilor web. Bootstrap este ușor de utilizat și este avantajos deoarece oferă multe componente reutilizabile, dar are ca dezavantaj dimensiunea mare a fișierelor CSS și JS. Alte framework-uri populare sunt Foundation, Bulma și Tailwind CSS.

## XAMPP

### 2.5.1. Introducere

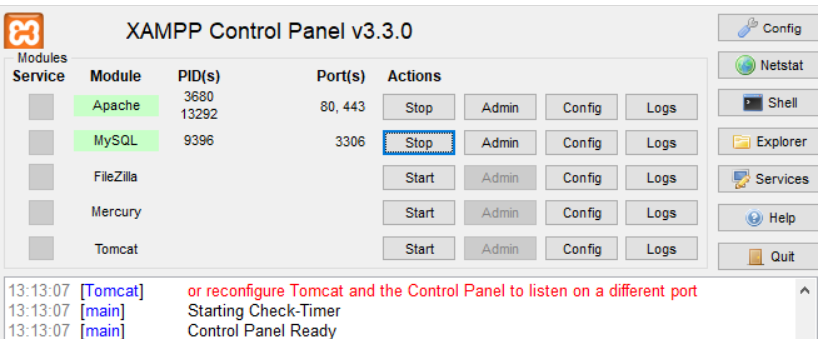
XAMPP este un pachet de software open-source care oferă o soluție pentru dezvoltarea și testarea aplicațiilor web. Numele XAMPP este acronim pentru X(platformă multiplă), Apache, MySQL, PHP și Perl. XAMPP este un pachet software ce pune la dispoziția dezvoltatorilor web un server Apache, un sistem prin care se pot gestiona bazele de date MySQL și suport pentru limbajele de programare PHP și Perl. Toate acestea fac din XAMPP un software ideal pentru dezvoltatorii web.

XAMPP a fost dezvoltat de compania Apache Friends în anul 2002. De-a lungul anilor a evoluat, devenind unul dintre cele mai populare pachete de software folosite în dezvoltarea aplicațiilor și a site-urilor web. XAMPP este ușor de utilizat și este suportat pe mai multe sisteme de operare, inclusiv Windows, Linux și macOS. XAMPP este actualizat constant pentru a suporta noile tehnologii și versiuni ale componentelor pe care le pune la dispoziție.

### 2.5.2. Instalarea și configurarea XAMPP

XAMPP poate fi descărcat de pe site-ul Apache Friends oficial, unde se pot găsi versiuni pentru Windows, Linux și macOS. Pentru instalare trebuie urmați câțiva pași simpli: selectarea componentelor care se doresc instalate, alegerea directorului de instalare (implicit, C:\xampp).

Panoul de Control XAMPP este interfața principală de unde sunt gestionate serviciile. După cum se poate observa din figura fig. 1.1., panoul de control conține butoane de start și stop al componentelor Apache, MySQL, FileZilla, Mercury și Tomcat (Apache și MySQL sunt pornite). În plus, fiecare serviciu are un buton care deschide fișierele de configurare, deci permite editarea și configurarea acestora. În consola de jos se pot observa mesaje relevante.



Apache poate fi configurat accesând fișierul principal de configurare (httpd.conf), unde se pot seta diferite configurații, precum porturile pe care rulează. De exemplu, schimbarea portului de la 80 (implicit) la 8080 se poate face folosind: Listen 8080. Configurarea MySQL se face în fișierul my.ini; acesta poate fi accesat apăsând pe butonul de config de la MySQL sau din directorul ‚xampp/mysql/bin’. De aici se pot modifica setările de memorie, cache și alte setări avansate. De exemplu, pentru a schimba portul de la 3306 (portul implicit) la 3307 se folosește: port=3307. Configurarea PHP se face în fișierul php.ini, care poate fi accesat la butonul de config din dreptul Apache sau în directorul ‚xampp’. Aici se pot configura memoria, extensiile încărcate și alte opțiuni. De exemplu pentru mărirea limitei de încărcare a fișierelor la 50M se utilizează: upload\_max\_filesize=50M;

### 2.5.3. XAMPP în dezvoltarea și testarea aplicațiilor web