MINISTERUL EDUCAȚIEI

INSPECTORATUL ȘCOLAR AL JUDEȚULUI COVANA

LICEUL TEORETIC „MIRCEA ELIADE” ÎNTORSURA BUZĂULUI

LUCRARE PENTRU ATESTAREA COMPETENŢELOR PROFESIONALE

PiXtreme

Elev: Oltean Costin-Petruț

Profesor îndrumǎtor: Bogdan-Ionut Ganea

2023

Cuprins

[1. Justificare proiect: 4](#_Toc133774907)

[2. Descrierea proiectului: 4](#_Toc133774908)

[3. Diagrama ERD: 4](#_Toc133774909)

[4. Tehnologii folosite: 5](#_Toc133774910)

[ MySQL : 5](#_Toc133774911)

[ HTML : 6](#_Toc133774912)

[ CSS : 6](#_Toc133774913)

[ PHP : 7](#_Toc133774914)

[ JavaScript : 7](#_Toc133774915)

[ Bootstrap : 8](#_Toc133774916)

[ GitHub 8](#_Toc133774917)

[ ChatGPT : 9](#_Toc133774918)

[5. Cerințe hardware si software : 9](#_Toc133774919)

[ Cerințe hardware: 9](#_Toc133774920)

[ Cerințe software: 10](#_Toc133774921)

[6. Structură și conținutul proiectului : 10](#_Toc133774922)

[ Structură HTML și CSS : 10](#_Toc133774923)

[ Index.css : 10](#_Toc133774924)

[ Login.css : 11](#_Toc133774925)

[ Sign\_up.css : 11](#_Toc133774926)

[ Wall.css : 12](#_Toc133774927)

[ Funcții de interogare : 14](#_Toc133774928)

[ Funcție pentru a introduce un utilizator în baza de date : 14](#_Toc133774929)

[ Funcție pentru a verifica dacă există un username sau un email deja existent : 16](#_Toc133774930)

[ Funcție pentru a prelua toate postăriile din baza de date : 17](#_Toc133774931)

[ Conectare la BD : 18](#_Toc133774932)

[ Codul sursă : 19](#_Toc133774933)

[ Index.php : Pagina principală a site-ului 19](#_Toc133774934)

[ Header.php : Pagina de header care se va regăsi în fiecare pagină 19](#_Toc133774935)

[ Footer.php : Pagina de footer care se va regăsi în fiecare pagină 20](#_Toc133774936)

[ Login.php : Formularul de login al site-ului 21](#_Toc133774937)

[ Signup.php : Formularul de sign up al site-ului 22](#_Toc133774938)

[ Code\_functions.php : Pagina care conține toate funcțiile utilizate in site 25](#_Toc133774939)

[ Code\_login.php : Codul care asigura logarea utilizatorului : 31](#_Toc133774940)

[ Code\_logout.php : Codul pentru a deloga utilizatorul 32](#_Toc133774941)

[ Code\_signup.php : Codul pentru a crea un cont nou 32](#_Toc133774942)

[ Signup\_confirmation : Pagina care îți confirmă faptul ca un cont a fost creat cu succes 35](#_Toc133774943)

[ Wall.php : Pagina în care se afla conținutul real al site-ului 36](#_Toc133774944)

[7. Bibliografie : 38](#_Toc133774945)

# Justificare proiect:

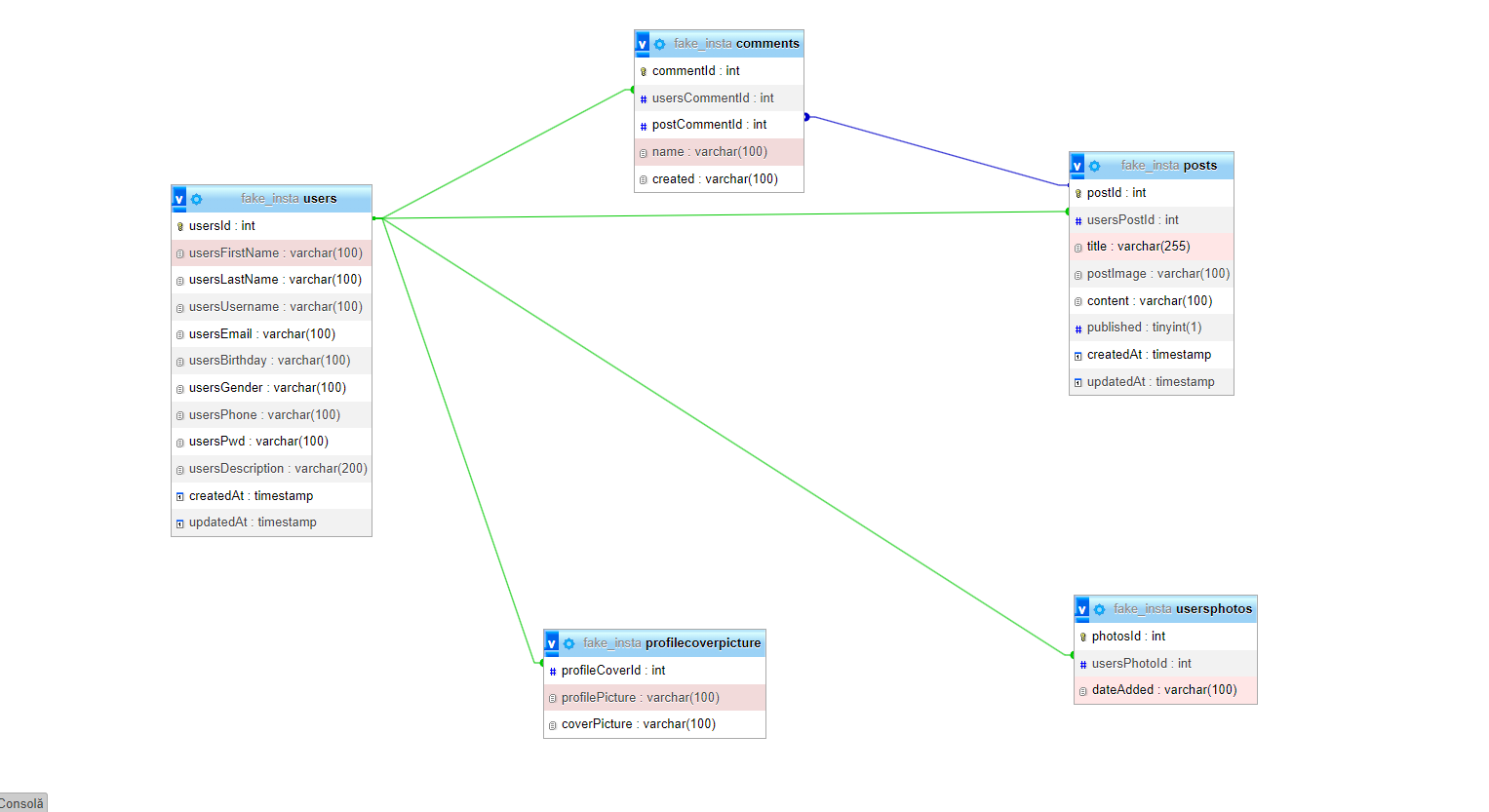
Am ales acest proiect pentru că, fiind un utilizator al aplicațiilor de tip social media, am vrut sa văd cum funcționează și care sunt necesitățiile pentru a crea o astfel de aplicație.

# Descrierea proiectului:

Proiectul meu constă în realizarea unei aplicații de social media, pe care am numit-o PiXtreme, în care să poată fii executate anumite lucruri simple, dar în același timp necesare pentru o astfel de aplicație, cum ar fi:

* Posibilitatea de a crea un cont;
* Posibilitatea de a te putea loga în aplicație cu contul creat;
* Posibilitatea de a putea posta postări care pot include poze, conținut text;
* Posibilitatea de a putea comenta și reacționa la astfel de postări;
* Posibilitatea de a de putea deconecta de pe aplicație.
* Etc.

# Diagrama ERD:



# Tehnologii folosite:

* MySQL : este un sistem open-source de gestionare a bazelor de date relaționale (RDBMS).Numele său este o combinație de „My”, numele fiicei co-fondatorului Michael Widenius My, și „SQL”, acronimul pentru Structured Query Language . O bază de date relaționalăorganizează datele în unul sau mai multe tabele de date în care datele pot fi legate între ele; aceste relații ajută la structurarea datelor. SQL este un limbaj folosit de programatori pentru a crea, modifica și extrage date din baza de date relațională, precum și pentru a controla accesul utilizatorilor la baza de date. Pe lângă bazele de date relaționale și SQL, un RDBMS precum MySQL funcționează cu un sistem de operare pentru a implementa o bază de date relațională în sistemul de stocare al unui computer, gestionează utilizatorii, permite accesul la rețea și facilitează testarea integrității bazei de date și crearea de copii de siguranță.MySQL este un software gratuit și cu sursă deschisă în conformitate cu termenii Licenței publice generale GNU și este disponibil și sub o varietate de licențe proprietare . MySQL a fost deținut și sponsorizat de compania suedeză MySQL AB , care a fost cumpărată de Sun Microsystems (acum Oracle Corporation ). În 2010, când Oracle a achiziționat Sun , Widenius a bifurcat proiectul open-source MySQL pentru a crea MariaDB .MySQL are clienți autonomi care permit utilizatorilor să interacționeze direct cu o bază de date MySQL utilizând SQL, dar mai des, MySQL este folosit cu alte programe pentru a implementa aplicații care au nevoie de capacitatea de baze de date relaționale.
* Apache : este un server HTTP de tip open source. Apache a jucat și joacă un rol important în dezvoltarea webului, fiind folosit în prezent în circa 65.2 % din paginile web.Cuvântul apache este numele apașilor, triburi de amerindieni.Apache este un server web cu o contribuție notabilă la dezvoltatea Internetului (world wide web). Apache a reprezentat prima alternativă viabilă la Netscape Communications Corporation, și a evoluat rapid în funcționalitate și performanță ca un rival competitiv pentru alte servere web bazate pe Unix.Apache este dezvoltat de o comunitate deschisă de programatori sub emblema Apache Software Foundation. Aplicația este disponibilă pentru o mare varietate de sisteme de operare incluzând Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows si OS/2.Serverul Apache este caracterizat ca fiind un software gratuit și open source, acesta făcând ca, începând din aprile 1996, el să fie cel mai popular server HTTP. Cu toate că în noiembrie 2005 a început să piardă din cota de piață, în aprilie 2008 Apache stătea încă la baza a peste 50 % din siturile web, iar în iunie 2013 a ajuns la 65.2%. Apache este folosit pentru 46,91% din totalul domeniului românesc.Prima versiune a serverului a fost creată de Robert McCool, care la vremea aceea era implicat în proiectul National Center for Supercomputing Applications, cunoscut pe scurt ca NCSA HTTPd. A doua versiune a serverului a fost o rescriere substanțială, de data aceasta punându-se accentul pe crearea unui layer prioritar (Apache Portable Runtime) și a suportului de module.
* HTML : HyperText Markup Language sau HTML este limbajul standard de marcare pentru documentele concepute pentru a fi afișate într-un browser web . Este adesea asistat de tehnologii precum Cascading Style Sheets (CSS) și limbaje de scripting, cum ar fi JavaScript .Browserele web primesc documente HTML de la un server web sau de la stocarea locală și redă documentele în pagini web multimedia. HTML descrie structura unei pagini web din punct de vedere semantic și a inclus inițial indicii pentru aspectul acesteia.Elementele HTML sunt elementele de bază ale paginilor HTML. Cu constructele HTML, imaginile și alte obiecte, cum ar fi formularele interactive , pot fi încorporate în pagina redată. HTML oferă un mijloc de a crea documente structurate prin denotarea semanticii structurale pentru text, cum ar fi titluri, paragrafe, liste, legături , citate și alte elemente. Elementele HTML sunt delimitate de etichete , scrise folosind paranteze unghiulare . Etichete precum și introduc direct conținut în pagină. Alte etichete precum și<img /><input /><p></p>înconjoară și oferă informații despre textul documentului și poate include etichete sub-element. Browserele nu afișează etichetele HTML, ci le folosesc pentru a interpreta conținutul paginii.HTML poate încorpora programe scrise într-un limbaj de scripting, cum ar fi JavaScript , care afectează comportamentul și conținutul paginilor web. Includerea CSS definește aspectul și aspectul conținutului. World Wide Web Consortium (W3C), fostul menținător al HTML și actualul menținător al standardelor CSS, a încurajat utilizarea CSS în defavoarea HTML-ului de prezentare explicit din 1997. O formă de HTML, cunoscută sub numele de HTML5 , este folosită pentru afișa video și audio, în principal folosind elementul , împreună cu JavaScript.
* CSS : Cascading Style Sheets ( CSS ) este un limbaj pentru foi de stil folosit pentru a descrie prezentarea unui document scris într-un limbaj de marcare precum HTML sau XML (inclusiv dialecte XML, cum ar fi SVG , MathML sau XHTML ). CSS este o tehnologie de bază a World Wide Web , alături de HTML și JavaScript.CSS este conceput pentru a permite separarea conținutului și a prezentării , inclusiv aspectul , culorile și fonturile .Această separare poate îmbunătăți accesibilitatea conținutului ; oferă mai multă flexibilitate și control în specificarea caracteristicilor de prezentare; permiteți mai multor pagini web să partajeze formatarea prin specificarea CSS-ului relevant într-un fișier .css separat, ceea ce reduce complexitatea și repetarea conținutului structural; și permiteți ca fișierul .css să fie stocat în cache pentru a îmbunătăți viteza de încărcare a paginii între paginile care partajează fișierul și formatarea acestuia.Separarea formatării și a conținutului face, de asemenea, posibilă prezentarea aceleiași pagini de marcare în stiluri diferite pentru diferite metode de randare, cum ar fi pe ecran, în tipărire, prin voce (prin browser sau cititor de ecran bazat pe vorbire) și pe baza Braille . dispozitive tactile. CSS are, de asemenea, reguli pentru formatarea alternativă dacă conținutul este accesat pe un dispozitiv mobil . Numele în cascadă provine din schema de prioritate specificată pentru a determina ce regulă de stil se aplică dacă mai multe reguli se potrivesc cu un anumit element. Această schemă de prioritate în cascadă este previzibilă.
* PHP : este un limbaj de scripting de uz general orientat spre dezvoltarea web . A fost creat inițial de programatorul danez-canadian Rasmus Lerdorf în 1993 și lansat în 1995.Implementarea de referință PHP este acum produsă de Grupul PHP. PHP a fost inițial o abreviere pentru Personal Home Page , dar acum reprezintă inițialismul recursiv PHP: Hypertext Preprocessor .Codul PHP este de obicei procesat pe un server web de un interpret PHP implementat ca modul , demon sau ca executabil CGI ( Common Gateway Interface ). Pe un server web, rezultatul codului PHP interpretat și executat – care poate fi orice tip de date, cum ar fi datele HTML generate sau imagini binare – ar forma întregul răspuns sau o parte a unui răspuns HTTP . Diverse sisteme de șabloane web , sisteme de gestionare a conținutului web și cadre webexistă care pot fi folosite pentru a orchestra sau facilita generarea acelui răspuns. În plus, PHP poate fi folosit pentru multe sarcini de programare în afara contextului web, cum ar fi aplicații grafice independente și controlul dronei robotizate .Codul PHP poate fi, de asemenea, executat direct din linia de comandă .Interpretul standard PHP, alimentat de Zend Engine , este un software gratuit lansat sub Licența PHP . PHP a fost portat pe scară largă și poate fi implementat pe majoritatea serverelor web pe o varietate de sisteme de operare și platforme.
* JavaScript : este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor.Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser. Limbajul este binecunoscut pentru folosirea sa în construirea siteurilor web, dar este folosit și pentru accesul la obiecte încapsulate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către Brendan Eich de la Netscape Communications Corporation sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.În ciuda numelui și a unor similarități în sintaxă, între JavaScript și limbajul Java nu există nicio legătură. Ca și Java, JavaScript are o sintaxă apropiată de cea a limbajului C, dar are mai multe în comun cu limbajul Self decât cu Java.Până la începutul lui 2005, ultima versiune existentă a fost JavaScript 1.5, care corespunde cu Ediția a 3-a a ECMA-262, ECMAScript, cu alte cuvinte, o ediție standardizată de JavaScript. Versiunile de Mozilla începând cu 1.8 Beta 1 au avut suport pentru E4X, care este o extensie a limbajului care are de a face cu XML, definit în standardul ECMA-357. Versiunea curentă de Mozilla, 1.8.1 (pe care sunt construite Firefox și Thunderbird versiunile 2.0) suportă JavaScript versiunea 1.7.
* Bootstrap : Bootstrap este o bibliotecă HTML, CSS și JS care se concentrează pe simplificarea dezvoltării paginilor web informative (spre deosebire de aplicațiile web ). Scopul principal al adăugarii acestuia la un proiect web este de a aplica alegerile Bootstrap de culoare, dimensiune, font și aspect la acel proiect. Ca atare, factorul principal este dacă dezvoltatorii responsabili găsesc acele alegeri pe placul lor. Odată adăugat la un proiect, Bootstrap oferă definiții de stil de bază pentru toate elementele HTML . Rezultatul este un aspect uniform pentru proză, tabele și elemente de formular în browserele web. În plus, dezvoltatorii pot profita de clasele CSS definite în Bootstrap pentru a personaliza în continuare aspectul conținutului lor. De exemplu, Bootstrap a prevăzut tabele de culoare deschisă și închisă, titluri de pagină, ghilimele de tragere mai proeminente și text cu evidențiere.Bootstrap vine, de asemenea, cu mai multe componente JavaScript care nu necesită alte biblioteci precum jQuery . Acestea oferă elemente suplimentare de interfață cu utilizatorul, cum ar fi casete de dialog , sfaturi cu instrumente , bare de progres, meniuri derulante de navigare și carusele. Fiecare componentă Bootstrap constă dintr-o structură HTML, declarații CSS și, în unele cazuri, cod JavaScript însoțitor. Ele extind, de asemenea, funcționalitatea unor elemente de interfață existente, inclusiv, de exemplu, o funcție de auto-completare pentru câmpurile de intrare.
* GitHub : este un serviciu de găzduire pe internet pentru dezvoltarea de software și controlul versiunilor folosind Git . Oferă controlul distribuit al versiunii Git plus controlul accesului , urmărirea erorilor , solicitările de funcții software , managementul sarcinilor , integrarea continuă și wiki-uri pentru fiecare proiect.Cu sediul în California, este o subsidiară a Microsoft din 2018.Este folosit în mod obișnuit pentru a găzdui proiecte de dezvoltare software open source . În ianuarie 2023, GitHub a raportat că are peste 100 de milioane de dezvoltatori și mai mult de 372 de milioane de depozite , inclusiv cel puțin 28 de milioane de depozite publice.Este cea mai mare gazdă de cod sursă din noiembrie 2021 . GitHub, Inc. a fost inițial o organizație simplă , fără manageri de mijloc; cu alte cuvinte, „toată lumea este un manager” ( auto-management ). Angajații puteau alege să lucreze la proiecte care îi interesau ( alocare deschisă ), dar salariile erau stabilite de directorul executiv. (adică liderii executivi individuali sau grupuri de companii iau decizii cu privire la obiectivele și dezvoltarea proiectului, inclusiv finanțarea).În 2014, GitHub, Inc. a adăugat un nivel de management mediu ca răspuns la acuzațiile grave de hărțuire făcute împotriva conducerii sale superioare. Ca urmare a scandalului, Tom Preston-Werner a demisionat din funcția de CEO.

# Cerințe hardware si software :

## Cerințe hardware:

* Procesor: Intel Core i3 sau echivalent
* Memorie RAM: 4 GB sau mai mult
* Spațiu de stocare: minim 500 MB, dar depinde de dimensiunea aplicației și a datelor utilizatorilor
* Placă video: orice placă video integrată care suportă cel puțin OpenGL 2.0
* Rezoluție ecran: recomandat 1280 x 768 sau mai mare
* Conexiune internet: conexiune de bandă largă pentru încărcarea și descărcarea conținutului din rețea.

## Cerințe software:

* Sistem de operare: Windows 10, MacOS, sau Linux
* Browser web: Chrome, Firefox, Safari, sau Edge
* Server web: Apache sau Nginx
* Limbaj de programare: PHP, Python, Ruby sau orice alt limbaj de programare potrivit pentru dezvoltarea aplicațiilor web
* Bază de date: MySQL, PostgreSQL sau orice alt sistem de gestiune a bazelor de date compatibil cu serverul web.

# Structură și conținutul proiectului :

## Structură HTML și CSS :

### Index.css :

body{

    text-align: center;

}

### Login.css :

body{

    background: linear-gradient(to top right, orange, #aa06c7);

    background-repeat: no-repeat;

    background-attachment: fixed;

}

.login-form{

    height: 100% auto;

    width: 100px auto;

    text-align: center;

    border: dotted black;

    margin-top:100px auto;

}

### Sign\_up.css :

body{

    background-color: whitesmoke;

    text-align: center;

    margin:10px;

    font-family: "Press Start 2P", cursive;

}

.signup-form-form{

    margin: 0 auto;

    height: 100% auto;

    width: 50px auto;

    border:solid violet;

    border-radius: 410px;

    width: 1000px;

}

input{

    height: 70px;

    font-family: "Comic Sans", cursive;

    background-color: pink;

    border-radius: 8px;

}

.submit{

    font-family : "comic sans";

}

#error{

    color:red;

}

### Wall.css :

body{

  background-color: whitesmoke;

}

.sidenav {

    height: 100%;

    width: 0;

    position: fixed;

    z-index: 1;

    top: 0;

    left: 0;

    background-color: #111;

    overflow-x: hidden;

    transition: 0.5s;

    padding-top: 60px;

  }

  .sidenav a {

    padding: 8px 8px 8px 32px;

    text-decoration: none;

    font-size: 25px;

    color: #818181;

    display: block;

    transition: 0.3s;

  }

  .sidenav a:hover {

    color: #f1f1f1;

  }

  .sidenav .closebtn {

    position: absolute;

    top: 0;

    right: 25px;

    font-size: 36px;

    margin-left: 50px;

  }

  @media screen and (max-height: 450px) {

    .sidenav {padding-top: 15px;}

    .sidenav a {font-size: 18px;}

  }

  .container{

    border : solid black;

  }

## Funcții de interogare :

### Funcție pentru a introduce un utilizator în baza de date :

Acesta este un cod PHP care definește o funcție numită "createAccount". Această funcție primește ca parametri o conexiune la baza de date ($conn) și informații despre utilizatorul care își creează contul (primul nume, numele de familie, numele de utilizator, adresa de email, data nașterii, genul, numărul de telefon și parola).

Funcția începe prin construirea un șir de caractere SQL care va fi utilizat pentru a insera informațiile despre utilizator în baza de date. Apoi, se pregătește o declarație SQL utilizând funcția "mysqli\_stmt\_init()" și se verifică dacă declarația a fost pregătită cu succes cu ajutorul funcției "mysqli\_stmt\_prepare()". Dacă pregătirea declarației a eșuat, utilizatorul va fi redirecționat către o pagină de înregistrare cu un mesaj de eroare.

Dacă pregătirea declarației a fost efectuată cu succes, parola utilizatorului este hash-uită cu ajutorul funcției "password\_hash()" pentru a proteja informațiile utilizatorului în baza de date. Această valoare hash este apoi legată de declarația pregătită cu ajutorul funcției "mysqli\_stmt\_bind\_param()" și apoi declarația este executată cu ajutorul funcției "mysqli\_stmt\_execute()".

La final, declarația este închisă cu ajutorul funcției "mysqli\_stmt\_close()", utilizatorul este redirecționat către o pagină de înregistrare cu un mesaj de succes și scriptul PHP se termină cu ajutorul funcției "exit()".

function createAccount($conn, $firstname, $lastname, $username, $email, $birthday, $gender, $phone, $pwd1) {

    $sql = "INSERT INTO users(usersFirstName, usersLastName, usersUsername, usersEmail, usersBirthday, usersGender, usersPhone, usersPwd) VALUES(?,?,?,?,?,?,?,?)";

    $stmt = mysqli\_stmt\_init($conn);

    if(!mysqli\_stmt\_prepare($stmt, $sql)){

        header('location:../pagina\_signup.php?error=stmtfailed2');

        exit();

    }

        $hashedPwd = password\_hash($pwd1, PASSWORD\_DEFAULT);

        mysqli\_stmt\_bind\_param($stmt, "ssssssss", $firstname, $lastname, $username, $email, $birthday, $gender, $phone, $hashedPwd);

        mysqli\_stmt\_execute($stmt);

        mysqli\_stmt\_close($stmt);

        header('location:../pagina\_signup.php?error=none');

        exit();

}

        $hashedPwd = password\_hash($pwd1, PASSWORD\_DEFAULT);

        mysqli\_stmt\_bind\_param($stmt, "ssssssss", $firstname, $lastname, $username, $email, $birthday, $gender, $phone, $hashedPwd);

        mysqli\_stmt\_execute($stmt);

        mysqli\_stmt\_close($stmt);

        header('location:../pagina\_signup.php?error=none');

        exit();

}

### Funcție pentru a verifica dacă există un username sau un email deja existent :

Aceasta este o funcție PHP numită "usernameTaken", care primește ca parametri o conexiune la baza de date ($conn), un nume de utilizator ($username) și o adresă de email ($email). Scopul acestei funcții este de a verifica dacă numele de utilizator sau adresa de email sunt deja luate în baza de date.

Funcția începe prin construirea un șir de caractere SQL care selectează toate rândurile din tabela "users" unde numele de utilizator sau adresa de email se potrivesc cu parametrii furnizați. Apoi, se pregătește o declarație SQL utilizând funcția "mysqli\_stmt\_init()" și se verifică dacă declarația a fost pregătită cu succes cu ajutorul funcției "mysqli\_stmt\_prepare()". Dacă pregătirea declarației a eșuat, utilizatorul va fi redirecționat către o pagină de înregistrare cu un mesaj de eroare.

Dacă pregătirea declarației a fost efectuată cu succes, numele de utilizator și adresa de email sunt legate de declarația pregătită cu ajutorul funcției "mysqli\_stmt\_bind\_param()" și apoi declarația este executată cu ajutorul funcției "mysqli\_stmt\_execute()".

Rezultatele interogării sunt obținute cu ajutorul funcției "mysqli\_stmt\_get\_result()" și sunt stocate în variabila "$resultData". Dacă funcția "mysqli\_fetch\_assoc()" returnează un rând din aceste rezultate, înseamnă că numele de utilizator sau adresa de email există deja în baza de date, iar acestea sunt returnate prin intermediul instrucțiunii "return $row;". Dacă nu există niciun rând în rezultate, înseamnă că numele de utilizator și adresa de email nu sunt luate încă, iar valoarea "false" este returnată prin intermediul instrucțiunii "return $result;". La final, declarația este închisă cu ajutorul funcției "mysqli\_stmt\_close()".

function usernameTaken($conn, $username, $email) {

    $sql = "SELECT \* FROM users WHERE usersUsername = ? OR usersEmail = ?;";

    $stmt = mysqli\_stmt\_init($conn);

    if(!mysqli\_stmt\_prepare($stmt, $sql)){

        header('location:../pagina\_signup.php?error=stmtfailed1');

        exit();

    }

        mysqli\_stmt\_bind\_param($stmt, "ss", $username, $email);

        mysqli\_stmt\_execute($stmt);

        $resultData = mysqli\_stmt\_get\_result($stmt);

        if($row = mysqli\_fetch\_assoc($resultData)){

            return $row;

        }

        else{

            $result = false;

            return $result;

        }

        mysqli\_stmt\_close($stmt);

}

### Funcție pentru a prelua toate postăriile din baza de date :

Aceasta este o funcție PHP numită "getPublishedPosts", care nu primește niciun parametru. Scopul acestei funcții este de a obține toate postările publicate din baza de date și de a le returna ca un array asociativ.

Funcția începe prin accesarea variabilei globale $conn, care reprezintă conexiunea la baza de date. Apoi, se construiește un șir de caractere SQL care selectează toate rândurile din tabela "posts" unde valoarea coloanei "published" este "true". Această instrucțiune SQL este executată utilizând funcția "mysqli\_query()", care primește conexiunea la baza de date ($conn) și șirul de caractere SQL ($sql).

Rezultatele interogării sunt stocate în variabila $result, iar apoi funcția "mysqli\_fetch\_all()" este utilizată pentru a obține toate rândurile rezultatului sub formă de array asociativ ($posts). În cele din urmă, acest array asociativ este returnat prin intermediul instrucțiunii "return $posts;".

function getPublishedPosts(){

        global $conn;

        $sql = "SELECT\* FROM posts WHERE published = true";

        $result = mysqli\_query($conn, $sql);

        $posts = mysqli\_fetch\_all($result, MYSQLI\_ASSOC);

        return $posts;

    }

### Conectare la BD :

Acesta este un fișier PHP care stabilește o conexiune la o bază de date MySQL folosind funcția "mysqli\_connect()". Conexiunea se face cu următorii parametri: numele serverului ("localhost"), numele de utilizator al bazei de date ("root"), parola ("mysql") și numele bazei de date ("fake\_insta").

După stabilirea conexiunii, fișierul definește două constante: "ROOT\_PATH" și "BASE\_URL". Prima constantă stochează calea absolută către fișierul curent (fișierul PHP curent), în timp ce cea de-a doua constantă stochează adresa de bază a site-ului ("http://localhost/atestat\_costin" în acest caz).

În plus, fișierul inițiază o sesiune folosind funcția "session\_start()". Sesiunea va fi utilizată pentru a stoca informații de autentificare ale utilizatorilor și alte date legate de sesiune.

<?php

    session\_start();

    $conn = mysqli\_connect("localhost","root","mysql","fake\_insta");

    if(!$conn){

        die("Eroare la conectare" .mysqli\_connect\_error());

    }

define('ROOT\_PATH' , realpath(\_\_FILE\_\_));

define('BASE\_URL','http://localhost/atestat\_costin');

?>

## Codul sursă :

### Index.php : Pagina principală a site-ului

<?php require("config.php"); ?>

<?php include("includes/header.php"); ?>

<link rel="stylesheet" href="static/index.css">

    <div class ="first\_page">

    <h1>L.T.M.E</h1>

    <br>

    <a href="login.php"> Log in </a>

    <br>

    <a href="signup.php"> Sign up </a>

    </div>

### Header.php : Pagina de header care se va regăsi în fiecare pagină

<!DOCTYPE html>

<html lang="ro">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>PiXtreme</title>

</head>

<body>

</body>

</html>

### Footer.php : Pagina de footer care se va regăsi în fiecare pagină

<footer>

    <h2 class = "footer">Copyright &copy; <a href="https://www.instagram.com/coxtin\_/" target="\_blank">Costin</a><?php echo " ".date('Y'); ?></h2>

</footer>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.7/dist/umd/popper.min.js" integrity="sha384-zYPOMqeu1DAVkHiLqWBUTcbYfZ8osu1Nd6Z89ify25QV9guujx43ITvfi12/QExE" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.min.js" integrity="sha384-Y4oOpwW3duJdCWv5ly8SCFYWqFDsfob/3GkgExXKV4idmbt98QcxXYs9UoXAB7BZ" crossorigin="anonymous"></script>

</body>

</html>

### Login.php : Formularul de login al site-ului

<?php include("includes/header.php"); ?>

<link rel="stylesheet" href="static/login.css">

<section class = "login-form">

    <h2>Log in</h2>

    <div class = "login-form-form">

        <form action="includes\_signup\_login/code\_login.php" method = "POST">

            <input type="text" name = "username" placeholder = "Email/Username"> <br>

            <input type="password" name = "pwd" placeholder = "Password"> <br>

            <button type = "submit" name = "submit">Log in</button>

        </form>

    </div>

    <?php

         if(isset($\_GET['error'])){

            if ($\_GET['error'] == "emptyInput"){

                echo "<p>Fill in all fields</p>";

            }

            elseif($\_GET['error'] == "wrongLogin"){

                echo "<p>Incorrect login information</p>";

            }

        }

    ?>

</section>

<?php include("includes/footer.php"); ?>

### Signup.php : Formularul de sign up al site-ului

<?php require("config.php"); ?>

    <link rel="stylesheet" href="static/sign\_up.css">

    <link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Press+Start+2P&display=swap" rel="stylesheet">

        <section class = "signup-form-form">

            <h2>PiXstreme</h2>

            <div class="signup-form">

                <h3>Sign up</h3><br>

                <p>Sign in to see videos and photos from friends</p>

                <form action="includes\_signup\_login/code\_signup.php" method = "POST">

                    <input type="text" name="firstname" placeholder="First name." title = "First Name"> <br>

                    <input type="text" name="lastname" placeholder="Last name." title = "Last Name" > <br>

                    <input type="text" name="username" placeholder="Username" title = "Username"> <br>

                    <div>

                        <input type="radio" id = "masc" name="gender" value = "Male">

                        <label for="masc">Male</label>

                        <input type="radio" id = "fem" name="gender" value ="Female">

                        <label for="fem">Female</label>

                    </div>

                    <br>

                    <input type="tel" name="phone" placeholder="Phone number" title = "Phone"> <br>

                        <h3>Enter your birthday</h3>

                    <input type="date" name = "birthday"><br>

                    <input type="email" name="email" placeholder="Email" title="Email"> <br>

                    <input type="password" name="pwd1" placeholder="Password"> <br>

                    <input type="password" name="pwd2" placeholder="Repeat password"> <br>

                    <button type="submit" name="submit" >Submit</button>

                </form>

            </div>

            <?php

        if(isset($\_GET['error'])){

            if ($\_GET['error'] == "emptyInput"){

                echo "<p id='error'>Fill in all fields</p>";

            }

            elseif($\_GET['error'] == "invalidAge"){

                echo "<p id='error'>Your age is too young</p>";

            }

            elseif($\_GET['error'] == "invalidUsername"){

                echo "<p id='error'>Choose a proper username!</p>";

            }

            elseif($\_GET['error'] == "invalidEmail"){

                echo "<p id='error'>Choose a proper email</p>";

            }

            elseif($\_GET['error'] == "pwdDoesNotMatch"){

                echo "<p id='error'>The 2 passwords does not match</p>";

            }

            elseif($\_GET['error'] == "pwdLenghtTooShort"){

                echo "<p id='error'>Password needs at least 8 characters</p>";

            }

            elseif($\_GET['error'] == "pwdSpecialCharactersNeeded"){

                echo "<p id='error'>Password needs at least one special character</p>";

            }

            elseif($\_GET['error'] == "pwdCapsCharactersNeeded"){

                echo "<p id='error'>Password needs at least one capital letter</p>";

            }

            elseif($\_GET['error'] == "pwdNumberNeeded"){

                echo "<p id='error'>Password needs at least one number</p>";

            }

            elseif($\_GET['error'] == "usernameTaken"){

                echo "<p id='error'>Username already taken</p>";

            }

            elseif($\_GET['error'] == "stmtfailed1" || $\_GET['error'] == "stmtfailed2"){

                echo "<p id='error'>Something went wrong.Try again</p>";

            }

            elseif($\_GET['error'] == "none"){

                header('location:includes\_signup\_login/signup\_confirmation.php');

            }

        }

    ?>

        </section>

<?php include("includes/footer.php");?>

### Code\_functions.php : Pagina care conține toate funcțiile utilizate in site

<?php

    global $conn;

function emptyInputSignup($firstname, $lastname, $username, $gender, $phone, $birthday, $email, $pwd1, $pwd2) {

    if (empty($firstname) || empty($lastname) || empty($username) || empty($gender) || empty($phone) || empty($birthday) || empty($email) || empty($pwd1) || empty($pwd2)){

        $result = true;

    }

    else {

        $result = false;

    }

    return $result;

}

function userAge($birthday){

    $birthday\_date = new datetime($birthday);

    $today = new datetime();

    $age = $today->diff($birthday\_date)->y;

    if ($age < 11) {

        $result = true;

    }

    else{

        $result = false;

    }

    return $result;

}

function invalidUsername($username) {

    if (!preg\_match("/^[a-zA-Z]\*$/", $username)){

        $result = true;

    }

    else {

        $result = false;

    }

    return $result;

}

function invalidEmail($email){

    if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)){

        $result = true;

    }

    else {

        $result = false;

    }

    return $result;

}

function pwdDoesNotMatch($pwd1, $pwd2) {

    if ($pwd1 !== $pwd2){

        $result = true;

    }

    else {

        $result = false;

    }

    return $result;

}

function pwdLenght($pwd1){

    if(strlen($pwd1) < 8){

        $result = true;

    }

    else{

        $result = false;

    }

    return $result;

}

function pwdSpecialCharacters($pwd1){

    if(!preg\_match('/[\'^£$%&\*()}{@#~?><>,|=\_+¬-]/', $pwd1)){

        $result = true;

    }

    else{

        $result = false;

    }

    return $result;

}

function pwdCapsCharactes($pwd1){

    if (!preg\_match('/[A-Z]/', $pwd1)){

        $result = true;

    }

    else{

        $result = false;

    }

    return $result;

}

function pwdNumbers($pwd1){

    if(!preg\_match('/\d/', $pwd1)){

        $result = true;

    }

    else{

        $result = false;

    }

    return $result;

}

function usernameTaken($conn, $username, $email) {

    $sql = "SELECT \* FROM users WHERE usersUsername = ? OR usersEmail = ?;";

    $stmt = mysqli\_stmt\_init($conn);

    if(!mysqli\_stmt\_prepare($stmt, $sql)){

        header('location:../pagina\_signup.php?error=stmtfailed1');

        exit();

    }

        mysqli\_stmt\_bind\_param($stmt, "ss", $username, $email);

        mysqli\_stmt\_execute($stmt);

        $resultData = mysqli\_stmt\_get\_result($stmt);

        if($row = mysqli\_fetch\_assoc($resultData)){

            return $row;

        }

        else{

            $result = false;

            return $result;

        }

        mysqli\_stmt\_close($stmt);

}

function createAccount($conn, $firstname, $lastname, $username, $email, $birthday, $gender, $phone, $pwd1) {

    $sql = "INSERT INTO users(usersFirstName, usersLastName, usersUsername, usersEmail, usersBirthday, usersGender, usersPhone, usersPwd) VALUES(?,?,?,?,?,?,?,?)";

    $stmt = mysqli\_stmt\_init($conn);

    if(!mysqli\_stmt\_prepare($stmt, $sql)){

        header('location:../pagina\_signup.php?error=stmtfailed2');

        exit();

    }

        $hashedPwd = password\_hash($pwd1, PASSWORD\_DEFAULT);

        mysqli\_stmt\_bind\_param($stmt, "ssssssss", $firstname, $lastname, $username, $email, $birthday, $gender, $phone, $hashedPwd);

        mysqli\_stmt\_execute($stmt);

        mysqli\_stmt\_close($stmt);

        header('location:../pagina\_signup.php?error=none');

        exit();

}

function emptyInputLogin($username, $pwd) {

    if (empty($username) || empty($pwd)){

        $result = true;

    }

    else {

        $result = false;

    }

    return $result;

}

    function loginUser($conn, $username, $pwd) {

        $usernameExists = usernameTaken($conn, $username, $username);

        if($usernameExists == false){

            header('location:../login.php?error=wronglogin');

            exit();

        }

        $pwdhashed = $usernameExists['usersPwd'];

        $checkPwd = password\_verify($pwd, $pwdhashed);

        if($checkPwd == false){

            header('location:../login.php?error=wronglogin');

            exit();

        }

        elseif ($checkPwd == true){

            session\_start();

            $\_SESSION["usersId"] = $usernameExists['usersId'];

            $\_SESSION["usersUsername"] = $usernameExists['usersUsername'];

            header('location:../wall.php');

            exit();

        }

    }

    function getPublishedPosts(){

        global $conn;

        $sql = "SELECT\* FROM posts WHERE published = true";

        $result = mysqli\_query($conn, $sql);

        $posts = mysqli\_fetch\_all($result, MYSQLI\_ASSOC);

        return $posts;

    }

?>

### ­­­­Code\_login.php : Codul care asigura logarea utilizatorului :

<?php

if(isset($\_POST['submit'])){

    $username = $\_POST['username'];

    $pwd = $\_POST['pwd'];

    require\_once('../config.php');

    require\_once('code\_functions.php');

    if(emptyInputLogin($username, $pwd) !== false){

        header('location:../pagina\_login.php?error=emptyInput');

        exit();

    }

    loginUser($conn, $username, $pwd);

}

else{

    header('location:../login.php');

    exit();

}

### Code\_logout.php : Codul pentru a deloga utilizatorul

<?php

    session\_start();

    session\_unset($\_SESSION['user']);

    session\_destroy();

    header('location:../index.php');

### Code\_signup.php : Codul pentru a crea un cont nou

<?php

    if(isset($\_POST['submit'])) {

        $firstname = $\_POST['firstname'];

        $lastname = $\_POST['lastname'];

        $username = $\_POST['username'];

        $gender = $\_POST['gender'];

        $phone = $\_POST['phone'];

        $birthday = $\_POST['birthday'];

        $email = $\_POST['email'];

        $pwd1 = $\_POST['pwd1'];

        $pwd2 = $\_POST['pwd2'];

    require\_once ('../config.php');

    require\_once ('code\_functions.php');

    if(emptyInputSignup($firstname, $lastname, $username, $gender, $phone, $birthday, $email, $pwd1, $pwd2) !== false){

        header('location:../pagina\_signup.php?error=emptyInput');

        exit();

    }

    if(userAge($birthday) !== false){

        header('location:../pagina\_signup.php?error=invalidAge');

        exit();

    }

    if(invalidUsername($username) !== false){

        header('location:../pagina\_signup.php?error=invalidUsername');

        exit();

    }

    if(invalidEmail($email) !== false){

        header('location:../pagina\_signup.php?error=invalidEmail');

        exit();

    }

    if(pwdDoesNotMatch($pwd1, $pwd2) !== false){

        header('location:../pagina\_signup.php?error=pwdDoesNotMatch');

        exit();

    }

    if(pwdLenght($pwd1) !== false){

        header('location:../pagina\_signup.php?error=pwdLenghtTooShort');

        exit();

    }

    if(pwdSpecialCharacters($pwd1) !== false){

        header('location:../pagina\_signup.php?error=pwdSpecialCharatersNeeded');

        exit();

    }

    if(pwdCapsCharactes($pwd1) !== false){

        header('location:../pagina\_signup.php?error=pwdCapsCharactersNeeded');

        exit();

    }

    if(pwdNumbers($pwd1) !== false){

        header('location:../pagina\_signup.php?error=pwdNumbersNeeded');

        exit();

    }

    if(usernameTaken($conn, $username, $email) !== false){

        header('location:../pagina\_signup.php?error=usernameTaken');

        exit();

    }

    createAccount($conn, $firstname, $lastname, $username, $email, $birthday, $gender, $phone, $pwd1);

    }

    else{

        header("location: ../pagina\_signup.php");

        exit();

    }

### Signup\_confirmation : Pagina care îți confirmă faptul ca un cont a fost creat cu succes

<html lang="ro">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Te-ai inregistrat cu succes</title>

</head>

<body>

        <div class = "countdown">

        <h2>You signed up</h2>

        <br>

        <p>You will be redirected to the login page <span id="countdown">5 seconds</span></p>

        </div>

        <?php

            $redirect\_url = '../login.php';

            $countdown\_time = 5;

            echo '<script>setTimeout(function(){ window.location.href = "' . $redirect\_url . '"; }, ' . ($countdown\_time \* 1000) . ');</script>';

        ?>

</body>

</html>

### Wall.php : Pagina în care se afla conținutul real al site-ului

<?php require\_once    ("config.php"); ?>

<?php include("includes\_signup\_login/code\_functions.php");?>

<?php $posts = getPublishedPosts();

// echo '<pre>';

// echo print\_r($posts);

// echo '</pre>';

?>

<?php include("includes/header.php");?>

<link rel="stylesheet" href="static/wall.css">

<div id="mySidenav" class="sidenav">

  <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

  <a href="#">Profile</a>

  <a href="#">Search</a>

  <a href="#">Messages</a>

  <a href="#">Add a post</a>

</div>

<span style="font-size:30px;cursor:pointer" onclick="openNav()">&#9776; Menu</span>

<script>

  function openNav() {

    document.getElementById("mySidenav").style.width = "250px";

  }

  function closeNav() {

    document.getElementById("mySidenav").style.width = "0";

  }

</script>

<div class="container justify-content-center ">

<?php foreach($posts as $posts\_inside):?>

  <div class="card">

    <div class="card-body">

      <h3 class="card-title"><?php echo  $posts\_inside['title'] ?></h3>

      <img src="static/images/<?php echo  $posts\_inside['postImage'] ?>" alt="">

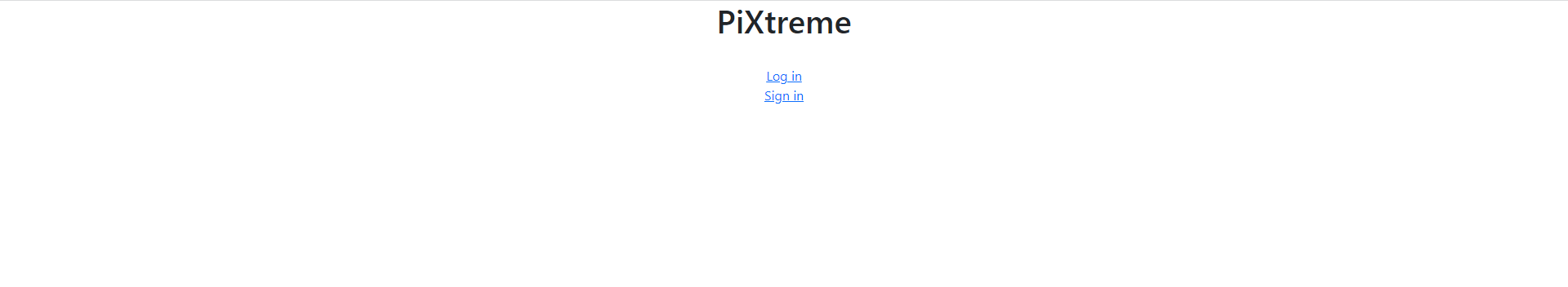
      <p><?php echo  $posts\_inside['content'] ?></p>

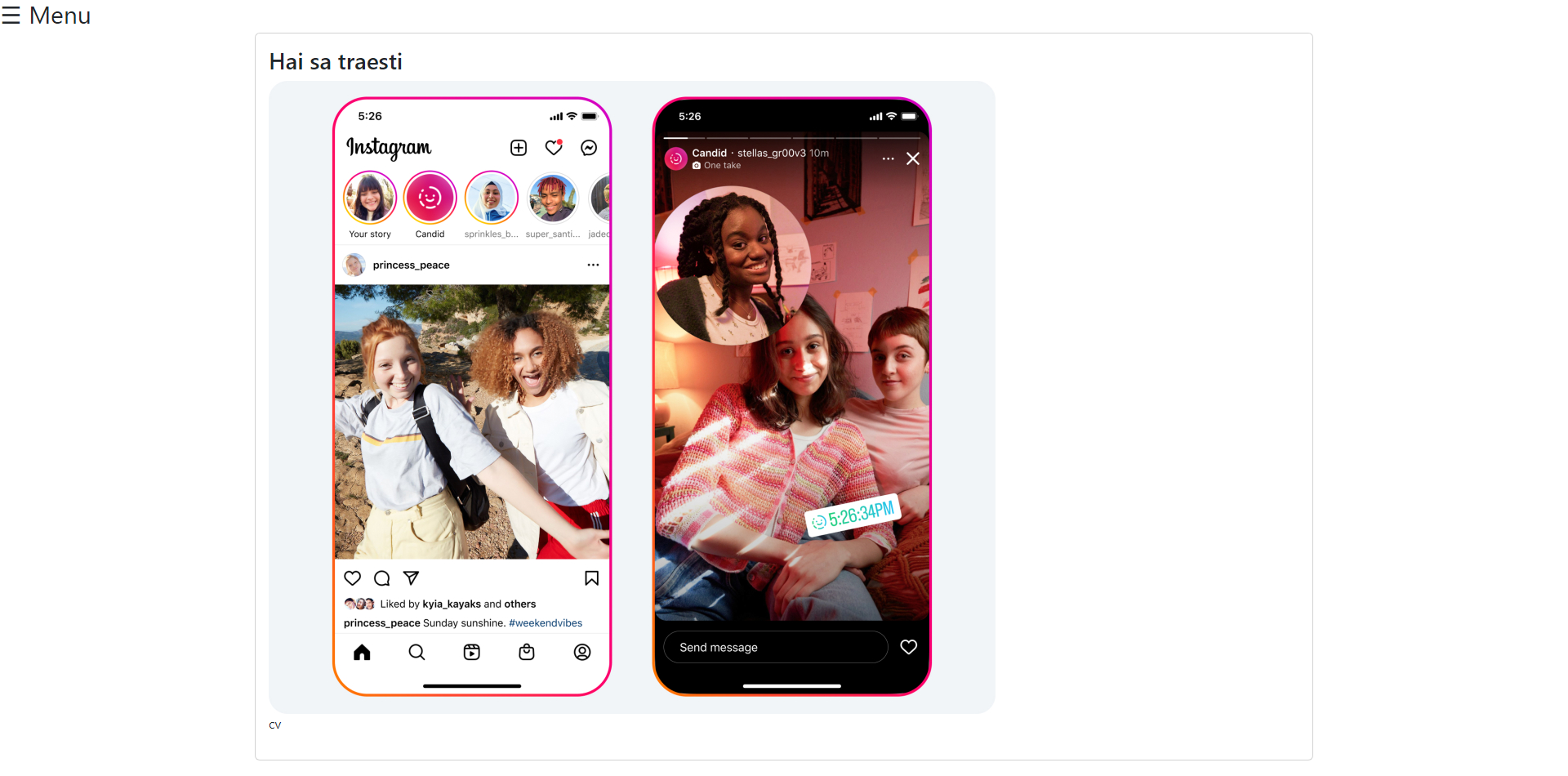
    </div>

  </div>

    <?php endforeach ?>

</div>



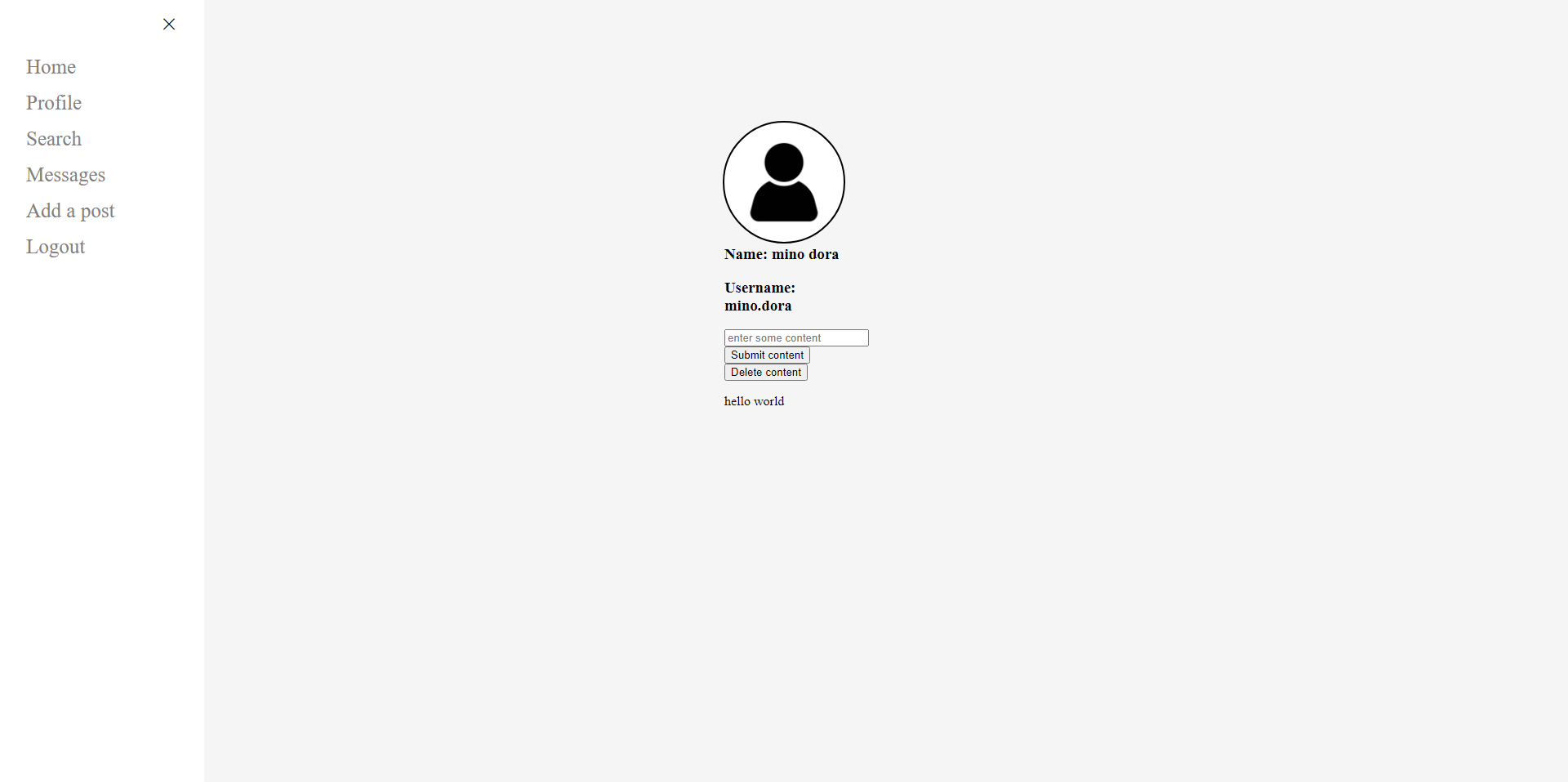


O imagine care conține grafic

Descriere generată automat

O imagine care conține grafic, diagramă

Descriere generată automat



# Bibliografie :

<https://getbootstrap.com/docs/5.3/components/card/#body>

<https://youtu.be/gCo6JqGMi30>

<https://github.com/Coxtin/atestat_costin_socialmedia>

https://www.w3schools.com/html/default.asp