



Universitatea Tehnică de Construcții din București
Colegiul de Studii Tețiare Nonuniversitare UTCB
Domeniul: Informatică
Specializarea: Analist Programator

PROIECT

**PENTRU SUSȚINEREA EXAMENULUI DE CERTIFICARE A
COMPETENȚELOR PROFESIONALE DE NIVEL 5**

Coordonator științific:
Ing. Olteanu Gabriela-Valentina

Absolvent:
Manole Elisei

București, 2025



Universitatea Tehnică de Construcții din București
Colegiul de Studii Tețiare Nonuniversitare UTCB
Domeniul: Informatică
Specializarea: Analist Programator

APLICATIE WEB PENTRU CINEMA

Coordonator științific:
Ing. Olteanu Gabriela-Valentina

Absolvent:
Manole Elisei

București, 2025

Cuprins

Introducere	4
Motivarea alegerii temei	4
Obiectivele propuse in cadrul lucrarii	4
Structura	4
1. Tehnologii si instrumente folosite	6
1.1. Design si aspect vizual	6
1.2. Limbaje folosite	8
1.3. Back-end development	14
1.4. Gestionarea conținutului	17
1.5. Hosting si publicare	22
2. Studiu de caz. aplicatie web pentru cinema	23
2.1. Pagina Acasa	23
2.2. Pagina de Filme	24
2.3. Pagina Despre	25
2.4. Pagina de Contact	26
2.5. Autentificare/Inregistrare	27
3. Baza de date	29
4. Programare Orientata Obiect	34
4.1. Prezentarea diagramei UML	34
5. Conceperea Produselor Multimedia	37
6. Concluzii, contributii personale si perspective de viitor	38
6.1. Concluzii	38
6.2. Contributii personale	38
6.3. Perspective de viitor	39
Anexe	40
Bibliografie	63

INTRODUCERE

Motivarea alegerii temei

Motivul alegerii acestui proiect este dorința de a dezvolta o aplicație web care să îmbine tehnologia cu arta cinematografică, într-un domeniu ce are un impact semnificativ asupra culturii și societății. Filmele, prin natura lor, au influențat constant dezvoltarea culturală, iar pasiunea mea pentru cinematografie și tehnologie m-a determinat să aleg această temă.

Pentru a administra eficient un cinematograful modern, sunt necesare soluții digitale care să optimizeze procesele interne, să îmbunătățească interacțiunea cu publicul și să faciliteze operarea acestuia. Alegerea acestei teme îmi oferă posibilitatea de a combina creativitatea în designul interfeței cu tehnologiile necesare implementării unui sistem atractiv și funcțional.

De asemenea, am fost inspirat de provocarea de a crea o aplicație complexă, care să includă funcționalități precum rezervarea locurilor, gestionarea programului de proiecții și administrarea conturilor utilizatorilor. Acest proiect este o oportunitate de a aplica cunoștințele acumulate în programare și design, dar și de a învăța noi tehnologii și metode de lucru.

Obiectivele propuse în cadrul lucrării

Scopul principal al acestui proiect este să dezvolt o aplicație web care să eficientizeze managementul unui cinematograful. Aceasta va avea un design atractiv, ușor de utilizat și bine structurat, pentru a îmbunătăți experiența utilizatorilor în procesul de rezervare a biletelor și interacțiune cu serviciile oferite.

Un obiectiv esențial este să demonstrez abilitățile dobândite în programarea web și în proiectarea aplicațiilor complexe, aplicând cunoștințele acumulate în limbaje precum PHP, HTML, CSS și JavaScript. De asemenea, îmi propun să creez un sistem optimizat, care să permită rezervarea online a locurilor pentru proiecții, gestionarea programului și detaliilor despre filme, precum și administrarea conturilor utilizatorilor, în funcție de rolurile acestora (admin, angajat, client).

Prin realizarea acestui proiect, urmăresc să învăț și să aplic noi tehnologii de design și dezvoltare web, inclusiv crearea de interfețe responsive, pentru a asigura o experiență optimă pe orice dispozitiv. În plus, îmi propun să integrez tehnici de optimizare pentru performanță și securitate, precum și să organizez datele într-o bază de date bine structurată, pentru a asigura integritatea și gestionarea eficientă a acestora.

Structura

Lucrarea este structurată în șase capitole principale și un capitol final dedicat concluziilor, contribuțiilor personale și perspectivelor de viitor. Fiecare capitol este conceput pentru a reflecta etapele dezvoltării și implementării aplicației web pentru managementul unui cinematograful, abordând toate aspectele necesare pentru o soluție funcțională și modernă.

Primul capitol introduce tema lucrării, prezentând motivarea alegerii acesteia și importanța dezvoltării unui sistem digital care să eficientizeze gestionarea cinematografulor. Este explicat contextul în care digitalizarea devine o necesitate în industria de divertisment și modul în care proiectul răspunde acestor nevoi. Tot aici sunt discutate provocările și oportunitățile oferite

de crearea unei astfel de platforme, precum și impactul pe care aceasta îl poate avea asupra utilizatorilor și administrației unui cinematograf. Prin intermediul acestui capitol, se oferă o bază teoretică solidă pentru înțelegerea scopului lucrării.

Cel de-al doilea capitol se concentrează pe tehnologiile și instrumentele utilizate în dezvoltarea aplicației. Acesta este structurat în mai multe subcapitole care detaliază elementele esențiale ale procesului de implementare. În prima parte, sunt prezentate aspectele de design și estetica interfeței, explicând cum s-au folosit principii moderne de UX/UI pentru a crea o platformă atractivă și intuitivă. Urmează o secțiune dedicată limbajelor de programare utilizate, inclusiv HTML, CSS, JavaScript pentru partea de front-end și PHP pentru back-end. De asemenea, este descrisă utilizarea bazei de date MySQL pentru stocarea informațiilor. Capitolul continuă cu o prezentare detaliată a mediilor de dezvoltare și a instrumentelor utilizate pentru gestionarea conținutului, cum ar fi Visual Studio Code și XAMPP, subliniind avantajele lor în contextul dezvoltării proiectului.

Capitolul trei reprezintă un studiu de caz al aplicației web dezvoltate, descriind principalele pagini și funcționalități ale acesteia. Pagina de start este detaliată ca punct principal de interacțiune pentru utilizatori, oferind informații esențiale despre cinematograf și facilități de navigare. Pagina dedicată filmelor este descrisă ca fiind elementul central al aplicației, unde utilizatorii pot explora detalii despre proiecții, inclusiv trailere și afișe. Pagina "Despre noi" adaugă o notă de autenticitate, explicând povestea cinematografului și echipa din spatele acestuia. Capitolul include și o descriere detaliată a funcționalităților de autentificare și înregistrare, precum și a secțiunii de contact, care facilitează comunicarea dintre utilizatori și administrație.

Capitolul patru se concentrează pe baza de date utilizată pentru stocarea și gestionarea informațiilor despre utilizatori, filme, rezervări și proiecții. Este prezentată schema bazei de date, explicând relațiile dintre tabele și modul în care acestea susțin funcționalitățile aplicației. Se oferă detalii despre optimizările efectuate pentru a îmbunătăți performanța interogărilor și pentru a asigura integritatea datelor. Prin intermediul acestui capitol, cititorii pot înțelege infrastructura care stă la baza funcționării aplicației.

Capitolul cinci abordează programarea orientată pe obiect și modul în care acest concept a fost implementat în dezvoltarea aplicației. Este prezentată o diagramă UML care ilustrează relațiile dintre clasele principale, precum User, Admin, Movie și Booking. Explicațiile detaliate despre organizarea și reutilizarea codului demonstrează cum principiile OOP contribuie la crearea unei aplicații scalabile și ușor de întreținut.

În capitolul șase este descris procesul de creare și integrare a resurselor multimedia, cum ar fi afișele filmelor și trailerele video. Sunt explicate etapele de editare a imaginilor și optimizarea acestora pentru utilizare online, subliniind rolul lor în crearea unei experiențe captivante pentru utilizatori. Acest capitol evidențiază importanța produselor vizuale în construirea unei aplicații atractive și funcționale.

Ultimul capitol reunește concluziile lucrării, evidențiind contribuțiile personale aduse prin proiect, lecțiile învățate și impactul acestuia asupra dezvoltării mele profesionale. De asemenea, sunt propuse perspective viitoare pentru îmbunătățirea și extinderea funcționalităților aplicației, precum și pentru integrarea unor tehnologii moderne care să răspundă nevoilor în continuă schimbare ale utilizatorilor.

Prin această structură detaliată, lucrarea oferă o imagine completă a procesului de dezvoltare a aplicației, abordând toate aspectele relevante, de la alegerea temei până la implementarea concretă și concluziile finale [1].

1.TEHNologii SI INSTRUMENTE FOLOSITE

1.1. Design si aspect vizual

Secțiunea de design și aspect vizual joacă un rol esențial în dezvoltarea aplicațiilor web, contribuind semnificativ la atractivitatea și utilizabilitatea acestora. Pentru acest proiect, Adobe Photoshop a fost utilizat ca principal instrument de editare grafică, fiind ales datorită funcționalităților sale avansate și a versatilității sale în procesarea imaginilor.

1.1.1. Adobe Photoshop

Adobe Photoshop este un software de referință în industria creativă, dezvoltat de compania Adobe Systems. Recunoscut pentru capacitățile sale extinse de editare și manipulare a imaginilor, Photoshop oferă utilizatorilor un set variat de instrumente și funcționalități avansate, ideale pentru crearea și ajustarea resurselor vizuale necesare în dezvoltarea unei aplicații web.

Cu o interfață intuitivă și ușor de navigat, Photoshop este utilizat pe scară largă de profesioniști în domenii precum design grafic, fotografie și publicitate. În cadrul acestui proiect, Photoshop a fost utilizat pentru a realiza editări precise ale afișelor filmelor, pentru a optimiza imaginile utilizate în aplicație și pentru a crea bannere promoționale atractive. Aceste elemente vizuale sunt esențiale în asigurarea unei experiențe captivante pentru utilizatori.

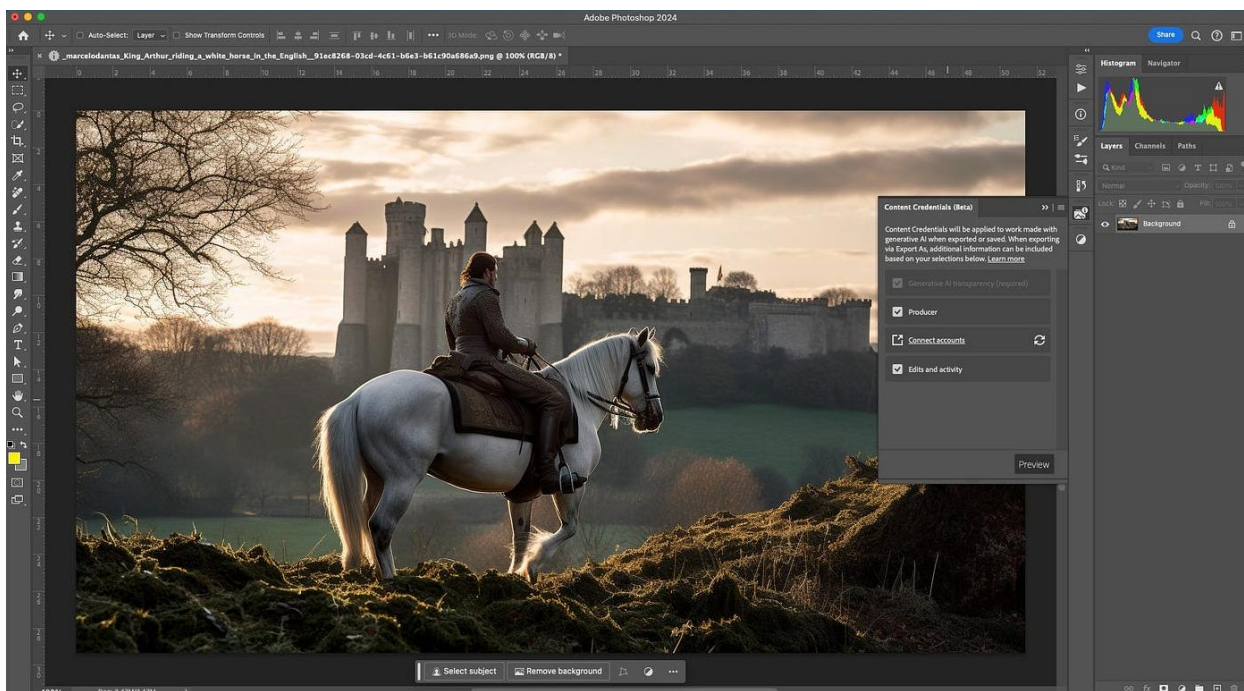


Fig.1.1 Adobe Photoshop

Adobe Photoshop a permis crearea compozițiilor vizuale prin utilizarea straturilor, o funcționalitate care oferă flexibilitate și control ridicat asupra fiecărui element grafic. Acest lucru a fost deosebit de util în personalizarea afișelor și a bannerelor, unde fiecare detaliu a fost ajustat separat pentru a obține un rezultat final armonios. Instrumentele de selecție și mascare au facilitat izolarea și retușarea unor porțiuni ale imaginilor, ceea ce a asigurat o precizie ridicată în editarea elementelor vizuale. De asemenea, ajustarea culorilor și tonurilor a fost utilizată pentru a asigura consistența cromatică între diferitele imagini, contribuind la estetica generală a aplicației.

Prin aplicarea unor efecte speciale, precum gradientele și filtrele de estompare, a fost posibilă crearea unor fundaluri elegante și atrăgătoare, care au accentuat identitatea vizuală a aplicației. Toate aceste elemente au fost integrate cu grijă pentru a îmbunătăți experiența vizuală a utilizatorilor. Photoshop a jucat un rol esențial în definirea aspectului vizual al aplicației, contribuind la crearea unei identități vizuale consistente și atractive. Prin utilizarea acestui software, a fost posibilă obținerea unui echilibru între designul artistic și cerințele tehnice ale platformei.

Această aplicație a Photoshop-ului nu se limitează doar la proiecte de design web, ci poate fi extinsă și în diverse alte domenii, precum crearea de materiale publicitare, design de produs sau branding vizual. În campaniile de marketing, pentru exemplu, utilizarea Photoshop-ului pentru personalizarea materialelor vizuale asigură coerența între diferitele canale de comunicare și permite adaptarea mesajului vizual în funcție de audiența țintă. De asemenea, pentru proiectele de dezvoltare de aplicații mobile, Photoshop joacă un rol important în crearea interfețelor atractive și user-friendly, esențiale pentru experiența utilizatorilor. Astfel, abilitățile dezvoltate în utilizarea Photoshop-ului pot fi aplicate eficient într-o gamă largă de proiecte, contribuind la succesul acestora în diferite domenii de activitate.

1.2. Limbaje folosite

1.2.1. Hypertext Markup Language

HTML (Hypertext Markup Language), cunoscut și sub denumirea de limbaj de marcare hipertext, este fundamentul oricărei experiențe online. Acesta reprezintă un standard esențial pentru structura și prezentarea conținutului pe paginile web, fiind utilizat de toți dezvoltatorii pentru a crea un cadru de bază al paginilor.

HTML permite dezvoltatorilor să definească structura semantică a unei pagini web prin utilizarea de etichete (tags) specifice. Aceste etichete, încadrate de paranteze unghiulare <>, oferă informații clare despre rolul fiecărui element din pagină. De exemplu, etichetele <h1> și <p> sunt utilizate pentru titluri și paragrafe, iar <a> este utilizată pentru link-uri. Prin intermediul atributelor, aceste etichete pot fi personalizate pentru a adăuga stiluri sau comportamente speciale, cum ar fi ajustarea culorilor, dimensiunilor sau conectarea la alte pagini.

```
57 </head>
58
59 <body style="background-color: black;">
60 <div class="outer">
61     <div class="content">
62         <h2>Bun venit la<span class="titlu"> CineMax</span></h2>
63         <p>Intra într-o lume unde magia filmelor a fermecat generatii intregi. Pasiunea pentru cinematografie transforma fiecare vizionare într-o experienta memorabila.
64             Descopera povestea noastra, traditia si dedicarea de a oferi momente de neuitat.</p>
65
66         <h2>O mostenire cinematografica unica</h2>
67         <p>Fa o incursiune in trecutul nostru si descopera cum a inceput povestea noastra. Inspirati de dragostea pentru povesti, am devenit un reper al cinematografiei.
68             Mostenirea noastra continua sa modeleze experientele culturale si de divertisment locale.</p>
69
70         <h2>In inima Romaniei</h2>
71         <p>Avem radacinile in inima vibranta a Bucurestiului, oferind un loc de intalnire pentru iubitorii de filme de toate varstele.
72             Alege dintr-o selectie variata de filme, de la cele mai noi lansari la capodoperele clasice, într-un cadru care celebreaza arta cinematografica.</p>
73
74         <h2>Experiente cinematografice deosebite</h2>
75         <p>Aici, fiecare detaliu conteaza. Bucura-te de imagini spectaculoase si sunet realist care te transpun direct in mijlocul actiunii.
76             Cu tehnologia noastra de ultima generatie, fiecare vizita devine o aventura de neuitat.</p>
```

Fig.1.2 Codul HTML al pagini de despre

O caracteristică esențială a HTML este capacitatea de a organiza informațiile într-o structură ierarhică, ceea ce facilitează citirea și interpretarea conținutului de către utilizatori și motoarele de căutare. Elementele de tip bloc, precum <div> sau <section>, sunt folosite pentru a organiza secțiuni mari ale paginii, în timp ce elementele de tip inline, precum sau , sunt utilizate pentru a marca elemente mai mici din text [2].

Ultima versiune majoră a HTML, HTML5, a adus o serie de îmbunătățiri semnificative. Printre acestea se numără suportul integrat pentru multimedia, prin etichetele <video> și <audio>, care permit redarea directă a conținutului media fără a necesita plugin-uri suplimentare. De asemenea, HTML5 a introdus elemente semantice noi, precum <article>, <header>, <footer> și <nav>, care îmbunătățesc organizarea și claritatea paginilor web.

Pe lângă suportul multimedia, HTML5 oferă și funcționalități avansate pentru formulare, cum ar fi validarea automată a datelor și utilizarea de noi tipuri de input, cum ar fi date, email și range. Aceste caracteristici simplifică procesul de dezvoltare și îmbunătățesc experiența utilizatorului [6].

HTML este utilizat în strânsă legătură cu alte tehnologii web esențiale, cum ar fi CSS (Cascading Style Sheets) și JavaScript. În timp ce HTML definește structura unei pagini, CSS controlează aspectul acesteia, iar JavaScript adaugă interactivitate. Împreună, aceste tehnologii formează baza dezvoltării web moderne, permițând crearea de site-uri dinamice și atractive.

În concluzie, HTML este indispensabil în dezvoltarea web, oferind o bază solidă pentru organizarea și prezentarea informațiilor. Prin utilizarea sa, dezvoltatorii pot crea pagini accesibile, ușor de înțeles și optimizate atât pentru utilizatori, cât și pentru motoarele de căutare. Evoluțiile constante ale acestui limbaj asigură că rămâne relevant și adaptabil în contextul rapidelor schimbări din domeniul tehnologic [7].



Fig.1.3 Logo HTML

1.2.2. Cascading Style Sheets

CSS (Cascading Style Sheets) este un limbaj esențial de stilizare utilizat în dezvoltarea web pentru a controla aspectul și prezentarea elementelor HTML. Acesta permite dezvoltatorilor să creeze un design personalizat pentru paginile web, separând stilurile de conținut și oferind o flexibilitate ridicată în personalizarea și optimizarea vizuală a unui site. CSS completează funcționalitatea HTML, permițând controlul estetic și uniformizarea stilurilor pe întreaga platformă.

Limbajul CSS folosește reguli de stilizare care constau dintr-un selector, ce identifică elementele HTML la care se aplică stilul, și un bloc de declarații, unde sunt definite proprietățile și valorile aplicate. Aceste proprietăți permit controlul unor aspecte precum culoarea, dimensiunea, fonturile, marginile, padding-ul, poziționarea și altele. Prin utilizarea claselor și a identificatorilor, CSS oferă o modalitate eficientă de a stiliza grupuri specifice de elemente sau de a aplica stiluri unice pentru anumite componente ale paginii [5].

```
232  /* carousel principal acasa */
233  .carousel-inner {
234      width: 100%;
235      height: 100%;
236  }
237
238  .carousel-item img {
239      width: 100%;
240      height: 100%;
241      object-fit: cover;
242  }
243
244  /* Media queries pentru ecrane mai mici */
245  @media screen and (max-width: 1200px) {
246      #carouselExampleAutoplaying {
247          height: 400px;
248      }
249  }
250
251  @media screen and (max-width: 768px) {
252      #carouselExampleAutoplaying {
253          height: 300px;
254      }
255  }
256
```

Fig.1.4 Cod CSS din pagina Style

Una dintre caracteristicile puternice ale CSS este capacitatea sa de a crea stiluri reutilizabile. Aceste stiluri pot fi centralizate într-un fișier extern, ceea ce permite gestionarea eficientă a întregului aspect al site-ului prin modificarea unui singur fișier.

Această abordare facilitează consistența vizuală pe întregul site și simplifică procesul de actualizare a designului [13].

CSS utilizează, de asemenea, conceptul de cascading, care determină modul în care stilurile sunt aplicate atunci când mai multe reguli sunt definite pentru același element. Specificitatea selectorilor, ordinea regulilor și utilizarea unor elemente precum `!important` permit dezvoltatorilor să controleze modul în care stilurile se suprapun, oferind un control precis asupra rezultatului final.

Odată cu evoluția sa, limbajul de programare a atins un nou nivel de sofisticare prin introducerea versiunii CSS3. Această versiune a adus funcționalități avansate, precum tranziții, animații, gradienturi, umbre și transparențe, oferind dezvoltatorilor instrumente noi pentru a crea designuri dinamice și atractive. CSS3 include, de asemenea, suport pentru media queries, permițând dezvoltarea designurilor responsive care se adaptează la diferite dispozitive și dimensiuni ale ecranului [8].

CSS funcționează complementar cu HTML și JavaScript, formând o triadă esențială pentru dezvoltarea web modernă. În timp ce HTML definește structura și conținutul paginii, CSS se ocupă de aspectul vizual, iar JavaScript adaugă interactivitate. Această combinație permite crearea de site-uri web interactive, estetice și optimizate pentru o varietate de platforme [17].

Importanța CSS în dezvoltarea web constă în capacitatea sa de a transforma pagini simple în experiențe vizuale captivante. Limbajul oferă flexibilitate și control complet asupra designului, contribuind la crearea unor site-uri bine structurate, estetice și ușor de utilizat.

CSS este indispensabil în realizarea unor platforme adaptabile și atractive, oferind utilizatorilor o experiență plăcută, indiferent de dispozitivul utilizat [21].



Fig.1.5 logoul

1.2.3. JavaScript

JavaScript este un limbaj de programare versatil și puternic, considerat unul dintre cei trei piloni fundamentali ai dezvoltării web, alături de HTML și CSS. Acesta joacă un rol esențial în adăugarea de interactivitate, funcționalitate și dinamică paginilor web, contribuind la crearea unor experiențe captivante și fluide pentru utilizatori. Popularitatea sa derivă din capacitatea de a fi interpretat direct de browser și de a fi integrat cu ușurință în codul HTML.

JavaScript permite dezvoltatorilor să manipuleze elementele paginii web, să gestioneze evenimente și să creeze interacțiuni în timp real. Spre exemplu, utilizatorii pot completa formulare a căror validare este realizată instantaneu, fără a fi necesară reîncărcarea paginii. De asemenea, JavaScript este utilizat pentru a adăuga animații, a controla carusele de imagini și a oferi răspunsuri rapide la acțiunile utilizatorilor, cum ar fi clicurile sau gesturile de tip drag and drop.

```
79
80 <!-- conectare bst -->
81 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
82 integrity="sha384-C6RzsynM9kWDrmMNeT87bh950GNYZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfl"crossorigin="anonymous">
83
84 const myCarouselElement = document.querySelector('#carouselExampleAutoplaying')
85
86 const carousel = new bootstrap.Carousel(myCarouselElement, {
87   interval: 10000, // Trecerea la următorul slide se va face la fiecare 10 secunde
88   touch: false
89 })
90 </script>
```

Fig.1.6 Codul JavaScript pentru carusel filme

Un punct forte al JavaScript este reprezentat de capacitatea sa de a lucra cu obiecte și funcții, permițând o abordare orientată pe obiecte în dezvoltarea aplicațiilor. Dezvoltatorii pot crea obiecte personalizate, adăugând proprietăți și metode specifice pentru a modela comportamente complexe. De asemenea, JavaScript facilitează modularitatea codului, ceea ce înseamnă că părțile componente ale aplicației pot fi reutilizate sau extinse cu ușurință. Acest lucru contribuie la o întreținere mai simplă a aplicațiilor și la scalabilitatea proiectelor.

JavaScript se află într-o continuă evoluție, cu standardele ECMAScript definind specificațiile limbajului. Versiuni recente, cum ar fi ECMAScript 6 (ES6) și următoarele, au introdus îmbunătățiri majore, inclusiv o sintaxă mai clară, suport pentru clase și module, precum și funcții avansate, cum ar fi promisiunile (Promises) și funcțiile asincrone (async/await). Aceste îmbunătățiri fac codul JavaScript mai intuitiv, mai eficient și mai ușor de întreținut.

Prin integrarea sa strânsă cu HTML și CSS, JavaScript reprezintă baza dezvoltării web moderne. Limbajul permite crearea unor site-uri web interactive, responsive și adaptabile, contribuind la îmbunătățirea experienței utilizatorilor pe toate platformele. De asemenea, este utilizat în aplicații complexe, cum ar fi cele pentru gestionarea cinematografelelor, unde este nevoie de interacțiuni dinamice, cum ar fi rezervările de bilete, actualizările în timp real și validarea formularelor [9].

JavaScript nu se limitează doar la dezvoltarea web, ci este utilizat și în dezvoltarea aplicațiilor mobile și desktop prin intermediul platformelor precum Node.js și Electron. Versatilitatea și puterea sa îl fac un limbaj indispensabil în ecosistemul modern al dezvoltării software, deschizând noi perspective pentru aplicații inovatoare și performante [18].

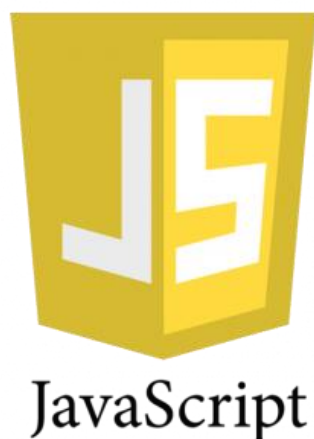


Fig.1.7 Logoul Java

1.3. Back-end development

1.3.1. Hypertext preprocessor

PHP (Hypertext Preprocessor) este un limbaj de programare interpretat, extrem de popular în dezvoltarea web, datorită versatilității sale și capacității de a crea aplicații dinamice. Inițial conceput pentru generarea paginilor web dinamice și interacțiunea cu bazele de date, PHP s-a extins semnificativ, devenind un instrument indispensabil pentru dezvoltarea diverselor aplicații și servicii web.

PHP este un limbaj server-side, ceea ce înseamnă că toate procesele sunt executate pe server, iar utilizatorii primesc în browser rezultatul final, sub formă de pagină web. Acest mecanism oferă dezvoltatorilor control complet asupra funcționalităților aplicației și permite crearea unor soluții complexe, cum ar fi autentificarea utilizatorilor, generarea conținutului dinamic sau gestionarea bazelor de date.

```
19 <?php
20 if ($conn->connect_error) {
21     die("Conexiunea a esuat: " . $conn->connect_error);
22 }
23
24 // Primele 8 filme din baza de date
25 $sql = "SELECT id_film, titlu, sursa_img FROM filme LIMIT 8";
26 $result = $conn->query($sql);
27
28 if ($result->num_rows > 0) {
29     // Inceperea containerului pentru toate filmele
30     echo "<div class='grid-container container'>";
31
32     while ($row = $result->fetch_assoc()) {
33         // Fiecare film are propriul card
34         echo "<div class='film-card'>";
35
36         // Crearea formularelor pentru fiecare film
37         echo "<form action='detalii.php' method='get'>";
38         echo "<input type='hidden' name='titlu' value='" . htmlspecialchars($row["titlu"]) . "'>";
39     }
36
37
38
39
```

Fig.1.8 Codul PHP pentru pagina de filme

Un avantaj important al PHP este simplitatea și accesibilitatea sintaxei sale. Acest lucru îl face ideal atât pentru începători, cât și pentru programatorii avansați, facilitând dezvoltarea rapidă și eficientă a aplicațiilor. Codul PHP nu necesită compilare prealabilă, ceea ce accelerează procesul de testare și implementare, oferind flexibilitate în realizarea modificărilor în timp real.

PHP beneficiază de o comunitate vastă și activă de utilizatori, care contribuie constant la dezvoltarea limbajului prin tutoriale, biblioteci și resurse gratuite. Această comunitate joacă un rol semnificativ în menținerea relevanței PHP și în oferirea de soluții inovatoare pentru problemele comune din dezvoltarea web.

Limbajul este compatibil cu o gamă variată de baze de date, inclusiv MySQL, PostgreSQL și SQLite, ceea ce îl face ideal pentru proiectele care implică stocarea și gestionarea datelor. Această compatibilitate, alături de ușurința integrării cu alte tehnologii web precum HTML, CSS și JavaScript, contribuie la crearea unor aplicații complexe și bine structurate.

De-a lungul anilor, PHP a evoluat semnificativ, cu introducerea unor versiuni precum PHP 7 și PHP 8, care au adus îmbunătățiri majore în performanță, securitate și funcționalități. Aceste versiuni moderne au inclus suport pentru tipuri de date stricte, funcții avansate și optimizări care permit gestionarea aplicațiilor la scară largă.

Un alt aspect esențial este existența unor framework-uri populare precum Laravel, Symfony și CodeIgniter, care simplifică dezvoltarea aplicațiilor complexe. Aceste framework-uri oferă structuri predefinite, biblioteci și instrumente care reduc timpul de dezvoltare și îmbunătățesc organizarea codului.

PHP este utilizat într-o gamă variată de aplicații, cum ar fi site-uri de comerț electronic, sisteme de gestionare a conținutului (CMS), platforme de socializare și servicii web bazate pe API-uri. În contextul proiectului, PHP a fost utilizat pentru gestionarea datelor stocate în baza de date, procesarea cererilor utilizatorilor și generarea dinamică a conținutului paginilor web, asigurând astfel funcționalitatea esențială a aplicației.

Datorită flexibilității și versatilității sale, PHP rămâne unul dintre cele mai utilizate limbaje de programare în dezvoltarea web. Capacitatea sa de a integra funcționalități avansate, compatibilitatea cu diverse baze de date și comunitatea activă îl fac o alegere excelentă pentru proiectele care necesită soluții eficiente, dinamice și scalabile.



Fig.1.9 Logoul PHP

1.3.2. Structured Query Language

MySQL este unul dintre cele mai populare și utilizate sisteme de gestionare a bazelor de date relaționale (RDBMS), cunoscut pentru performanța, scalabilitatea și fiabilitatea sa. Dezvoltat inițial de compania suedeză MySQL AB și ulterior integrat în portofoliul Oracle Corporation, acest sistem a devenit o alegere preferată pentru dezvoltatori și organizații care au nevoie de o soluție robustă pentru stocarea și manipularea datelor.

Un aspect esențial al acestui RDBMS este utilizarea limbajului de interogare structurat, care permite crearea, actualizarea și gestionarea datelor. Acesta susține o gamă largă de funcționalități, facilitând majoritatea operațiunilor necesare în manipularea datelor stocate. Flexibilitatea și compatibilitatea sa cu standardele industriei îl fac ușor de integrat în proiecte și aplicații de diferite tipuri și dimensiuni.

```
1 CREATE TABLE `filme` (  
2   `id_film` int(11) NOT NULL,  
3   `titlu` varchar(255) NOT NULL,  
4   `descriere` text DEFAULT NULL,  
5   `sursa_img` varchar(255) DEFAULT NULL,  
6   `pret_film` decimal(10,2) DEFAULT NULL  
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
8  
9 CREATE TABLE `sala` (  
10  `id_sala` int(11) NOT NULL,  
11  `numeSala` varchar(255) NOT NULL,  
12  `id_film` int(11) DEFAULT NULL  
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Fig.1.10 Codul MySql pentru creare tabele

Acest sistem de gestionare a bazelor de date relaționale excelează prin performanța sa, fiind optimizat pentru acces rapid la date și gestionarea volumelor mari de informații. Viteza sa de operare este susținută de mecanisme avansate de indexare, care permit realizarea de interogări complexe și rapide chiar și în baze de date de dimensiuni considerabile. Acest avantaj îl face ideal pentru aplicațiile ce necesită acces în timp real la date, cum ar fi platformele de comerț electronic, rețelele sociale și aplicațiile de streaming [16].

Un alt avantaj major este scalabilitatea, care îl face potrivit atât pentru proiecte mici, cum ar fi site-urile personale, cât și pentru aplicații complexe ce gestionează milioane de înregistrări. Funcționalitățile avansate, precum gestionarea concurenței, asigură acces simultan la date de către mai mulți utilizatori, fără a compromite performanța sau integritatea bazei de date.

În ceea ce privește fiabilitatea, acest sistem oferă suport pentru replicare și clustering, funcționalități esențiale pentru menținerea disponibilității și securității datelor. Replicarea permite crearea de copii sincronizate ale bazei de date pe mai multe servere, oferind redundanță și posibilitatea de a recupera datele în cazul unei defecțiuni. Clusteringul, pe de altă parte, distribuie sarcinile de procesare pe mai multe servere, asigurând o disponibilitate constantă și o performanță îmbunătățită chiar și în medii de producție critice.

Fiind open-source, acest sistem beneficiază de o comunitate globală activă de dezvoltatori care contribuie la îmbunătățirea continuă a acestuia. Actualizările regulate și dezvoltarea de noi funcționalități asigură că rămâne în topul soluțiilor de gestionare a bazelor de date relaționale.

De-a lungul timpului, a fost utilizat în diverse scenarii, de la gestionarea bazelor de date pentru site-uri web dinamice până la aplicații complexe de întreprindere. Integrarea perfectă cu limbajele de programare web, precum PHP, și compatibilitatea cu diverse platforme îl fac o alegere preferată în dezvoltarea modernă. În cadrul proiectului nostru, această soluție este utilizată pentru gestionarea datelor despre utilizatori, filme, rezervări și programări, asigurând o funcționare eficientă și securizată a aplicației.

Acest sistem continuă să evolueze, aducând îmbunătățiri semnificative în ceea ce privește performanța și securitatea, consolidându-și poziția ca soluție de top pentru gestionarea bazelor de date relaționale [22].



Fig.1.11 Logoul MySQL

1.4. Gestionarea conținutului

1.4.1. Visual Studio Code

Visual Studio Code (VS Code) este un editor de cod sursă modern, open-source și gratuit, dezvoltat de Microsoft, destinat să ofere programatorilor un mediu de lucru eficient și adaptabil. Acesta este utilizat pe scară largă în dezvoltarea de aplicații web, mobile, și cloud datorită flexibilității și funcționalităților sale avansate.

Disponibil pe multiple platforme, inclusiv Windows, macOS și Linux, VS Code vine cu suport nativ pentru o gamă largă de limbaje de programare, cum ar fi HTML, CSS, JavaScript, Python, Java, C++ și multe altele. Acesta oferă funcționalități precum evidențierea sintaxei, completarea automată, refactorizarea codului și debugging, ceea ce simplifică procesul de scriere și întreținere a codului sursă [10].

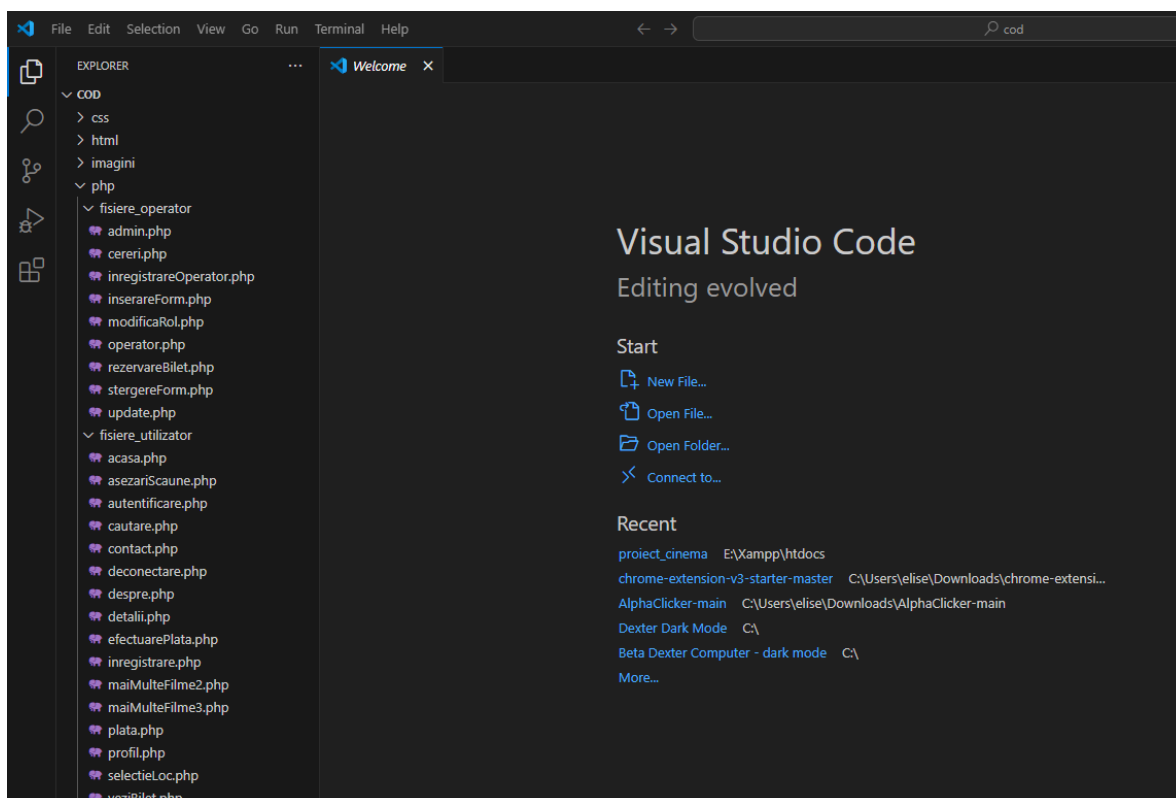


Fig.1.12 Interfața Visual Studio Code

Unul dintre principalele avantaje ale VS Code este interfața sa modulară și personalizabilă. Utilizatorii pot ajusta aspectul și funcționalitățile editorului în funcție de nevoile lor, alegând dintr-o gamă variată de teme și extensii disponibile în piața dedicată. Aceste extensii includ instrumente pentru integrarea cu diverse limbaje, framework-uri și platforme, permițând programatorilor să creeze un mediu de lucru adaptat proiectelor lor specifice [11].

Un alt punct forte al VS Code este integrarea instrumentelor și serviciilor de dezvoltare într-un singur mediu. Terminalul integrat elimină necesitatea de a comuta între aplicații, iar debugger-ul oferă o modalitate eficientă de identificare și corectare a erorilor. Funcționalități precum preview-ul fișierelor, suportul pentru testare și integrarea cu linter contribuie la un flux de lucru bine organizat și eficient. Această coerență ajută dezvoltatorii să își concentreze atenția asupra proiectelor, reducând timpul pierdut cu trecerea între diferite instrumente.

Integrarea cu Git reprezintă unul dintre cele mai apreciate aspecte ale VS Code. Această funcționalitate permite gestionarea eficientă a codului sursă, urmărirea modificărilor și facilitarea colaborării în echipe. Prin intermediul Git, dezvoltatorii pot iniția și rezolva conflicte de cod, pot gestiona ramurile proiectelor și pot implementa fluxuri de lucru CI/CD (integrare continuă și livrare continuă) direct din editor [14].

În cadrul acestui proiect, Visual Studio Code a fost utilizat pentru scrierea codului, gestionarea fișierelor și colaborarea eficientă asupra dezvoltării aplicației web. Suportul său pentru limbajele utilizate, cum ar fi PHP, HTML, CSS și JavaScript, a contribuit la asigurarea unei experiențe de dezvoltare fluide. Terminalul integrat și suportul pentru debug au fost esențiale în testarea și perfecționarea funcționalităților site-ului.

Un alt aspect notabil este comunitatea activă care susține Visual Studio Code. Aceasta oferă suport, extensii noi și actualizări frecvente, care asigură că editorul rămâne la zi cu cele mai recente tehnologii și cerințe ale dezvoltării moderne [15].

Visual Studio Code este mai mult decât un simplu editor de cod; este o platformă robustă și versatilă care răspunde nevoilor dezvoltatorilor din diverse domenii. Cu funcționalități avansate, o interfață personalizabilă și integrarea eficientă cu alte instrumente, VS Code contribuie la crearea unor proiecte de succes și la îmbunătățirea procesului de dezvoltare.

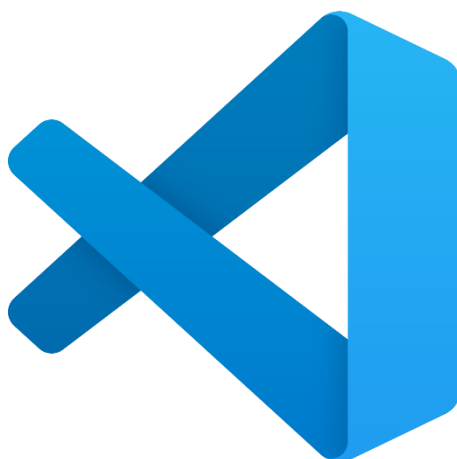


Fig.1.13 Logoul Visual Studio Code

1.4.2. XAMPP

XAMPP este un pachet software open-source destinat dezvoltării și testării aplicațiilor web pe un server local. Numele său este un acronim pentru Apache, MySQL, PHP și Perl, cele patru componente principale integrate în această platformă. XAMPP oferă un mediu complet și ușor de utilizat, care reunește toate instrumentele necesare pentru a dezvolta aplicații web funcționale și eficiente.

Apache, componenta principală a XAMPP, este un server web extrem de popular, cunoscut pentru stabilitatea și performanța sa. Acesta permite găzduirea și rularea aplicațiilor web, oferind suport pentru protocoale standard precum HTTP și HTTPS. MySQL, o altă componentă integrată, este un sistem de gestionare a bazelor de date relaționale utilizat pentru stocarea și administrarea datelor, esențial pentru funcționarea aplicațiilor dinamice. PHP este limbajul de programare server-side folosit pentru crearea paginilor web interactive și pentru interacțiunea cu bazele de date, în timp ce Perl adaugă flexibilitate prin scripturi suplimentare și funcționalități avansate.

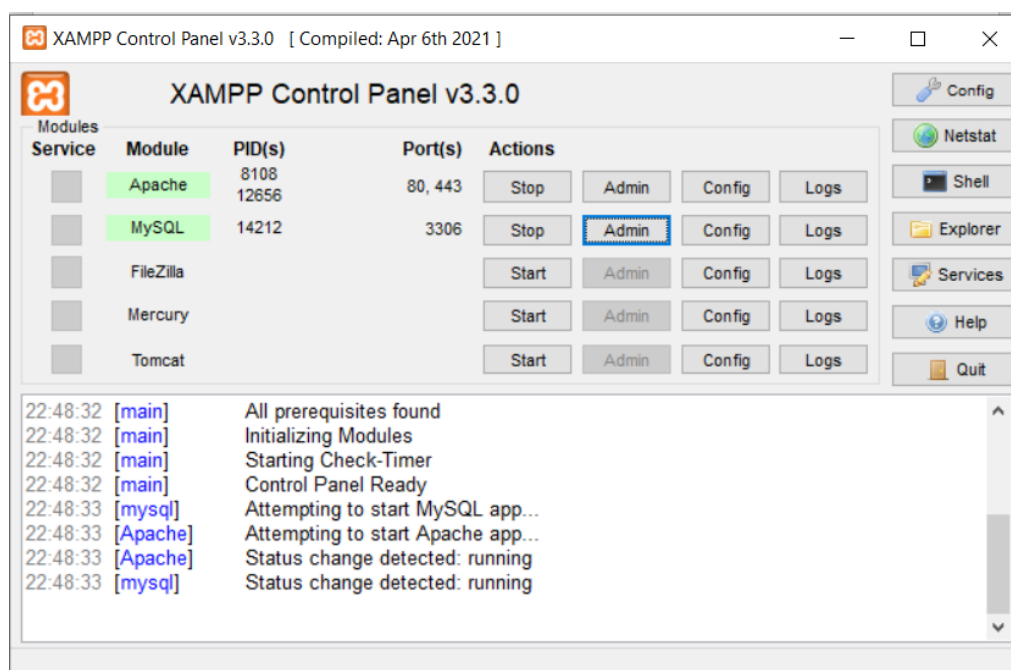


Fig.1.14 Interfata Xampp

Unul dintre principalele sale avantaje este simplitatea instalării și configurării. Prin instalarea acestui pachet software, dezvoltatorii beneficiază de un mediu complet configurat, care le permite să înceapă imediat lucrul. Această abordare elimină necesitatea configurării manuale a fiecărei componente individuale, economisind timp și reducând complexitatea procesului.

De asemenea, soluția este portabilă, fiind compatibilă cu diverse platforme, inclusiv Windows, macOS și Linux. Această portabilitate oferă dezvoltatorilor un mediu de lucru uniform, indiferent de sistemul de operare utilizat. Fie că este vorba despre testarea unui site local sau dezvoltarea unei aplicații complexe, XAMPP oferă instrumentele necesare pentru a lucra eficient pe orice platformă.

Este utilizat pe scară largă în etapa de dezvoltare și testare a aplicațiilor web. Prin rularea aplicațiilor pe un server local, dezvoltatorii pot testa funcționalitățile și identifica eventualele erori într-un mediu controlat, fără a afecta utilizatorii finali. Acest proces de testare este esențial pentru asigurarea calității aplicațiilor înainte de lansarea pe un server public.

Este important de menționat că această soluție este concepută în principal pentru dezvoltare și testare și nu este recomandat să fie utilizată direct într-un mediu de producție. Pentru utilizarea sa în producție, este necesară configurarea suplimentară a componentelor, precum securizarea accesului la server, configurarea certificatelor SSL și optimizarea setărilor pentru performanță.

În cadrul acestui proiect, pachetul software a jucat un rol esențial, servind drept mediu de dezvoltare pentru aplicația web. Serverul Apache a permis rularea aplicației local, iar baza de date MySQL a fost utilizată pentru stocarea și gestionarea datelor, cum ar fi informațiile despre utilizatori, rezervări și filme. PHP, componenta server-side, a fost responsabil pentru procesarea datelor și generarea conținutului dinamic al paginilor.

Această soluție rămâne o alegere preferată pentru dezvoltatorii care caută o metodă completă și accesibilă pentru gestionarea serverului local. Cu suport pentru principalele componente ale dezvoltării web și o comunitate activă care oferă suport și actualizări, este ideală pentru crearea de aplicații web funcționale și bine testate, contribuind la succesul proiectelor de orice dimensiune.



Fig.1.15 Logoul XAMPP

1.5. Hosting si publicare

1.5.1. Numele de domeniu

Un nume de domeniu reprezintă un identificator unic care facilitează accesul utilizatorilor la o aplicație web, eliminând necesitatea utilizării adreselor IP complexe sau a porturilor specificate. În contextul utilizării XAMPP ca mediu de dezvoltare, numele de domeniu atribuit serverului local joacă un rol important, oferind o modalitate simplă și convenabilă de a accesa și interacționa cu aplicațiile web dezvoltate.

Acest nume de domeniu poate fi personalizat pentru a reflecta identitatea proiectului și a fi ușor de reținut de utilizatori. De exemplu, în loc să se acceseze aplicația web printr-o adresă de tipul `http://localhost:8080`, utilizatorii pot folosi un nume de domeniu configurat local, precum `http://Eli's Cinemax.local`. Această funcționalitate contribuie la o experiență mai plăcută de testare și prezentare a proiectului, mai ales în cazul colaborării cu alți membri ai echipei sau al demonstrațiilor.

Un alt avantaj al utilizării unui nume de domeniu personalizat este simularea unui mediu de producție real. Prin asocierea unui nume de domeniu aplicației, dezvoltatorii pot identifica și corecta eventualele probleme legate de accesibilitate, structura URL-urilor sau alte detalii legate de implementare, asigurând astfel o tranziție lină către publicarea pe un server public.

În cadrul acestui proiect, un nume de domeniu local a fost utilizat pentru a facilita accesul la aplicația de gestionare a cinematografele dezvoltată cu ajutorul acestui pachet software. Configurarea numelui de domeniu a permis o testare mai rapidă și o prezentare mai clară a funcționalităților proiectului, eliminând complicațiile tehnice asociate cu utilizarea adreselor IP brute.

2. STUDIU DE CAZ. APLICATIE WEB PENTRU CINEMA

Eli's Cinemax reprezintă un exemplu practic de implementare a unei platforme dedicate pasionaților de film, care combină designul modern cu funcționalități esențiale pentru o experiență completă. Platforma este construită utilizând limbaje de programare precum PHP, HTML, CSS, JavaScript și SQL, fiind hostată local prin XAMPP. Proiectul a fost creat pentru a facilita accesul utilizatorilor la informații despre cele mai noi filme, trailere și recenzii, precum și pentru a simplifica procesul de rezervare a билетelor. Studiul de caz se concentrează pe integrarea bazelor de date pentru gestionarea utilizatorilor și a filmelor, optimizarea interfeței pentru navigare ușoară și respectarea principiilor de securitate și accesibilitate.

2.1. Pagina Acasa

Pagina de start a aplicației joacă un rol esențial, servind drept punct central pentru utilizatori. Aceasta este proiectată pentru a atrage atenția și a ghida navigarea către secțiuni importante ale site-ului. Designul combină elemente vizuale atractive cu funcționalități esențiale, cum ar fi navigarea rapidă și accesul direct la funcții cheie. În partea superioară a paginii se află meniul principal de navigare, care oferă linkuri către secțiuni importante precum filme, autentificare, contact și despre. În plus, pagina include un slider dinamic cu imagini, care prezintă cele mai recente sau populare filme disponibile în cinema, stârnind interesul utilizatorilor.

De asemenea, sunt incluse secțiuni promoționale ce anunță evenimente speciale, premiere de filme sau oferte. Un buton de autentificare proeminent facilitează accesul rapid la contul utilizatorului. Acest design simplu și intuitiv permite utilizatorilor să navigheze cu ușurință și să descopere conținutul relevant (vezi Fig.).

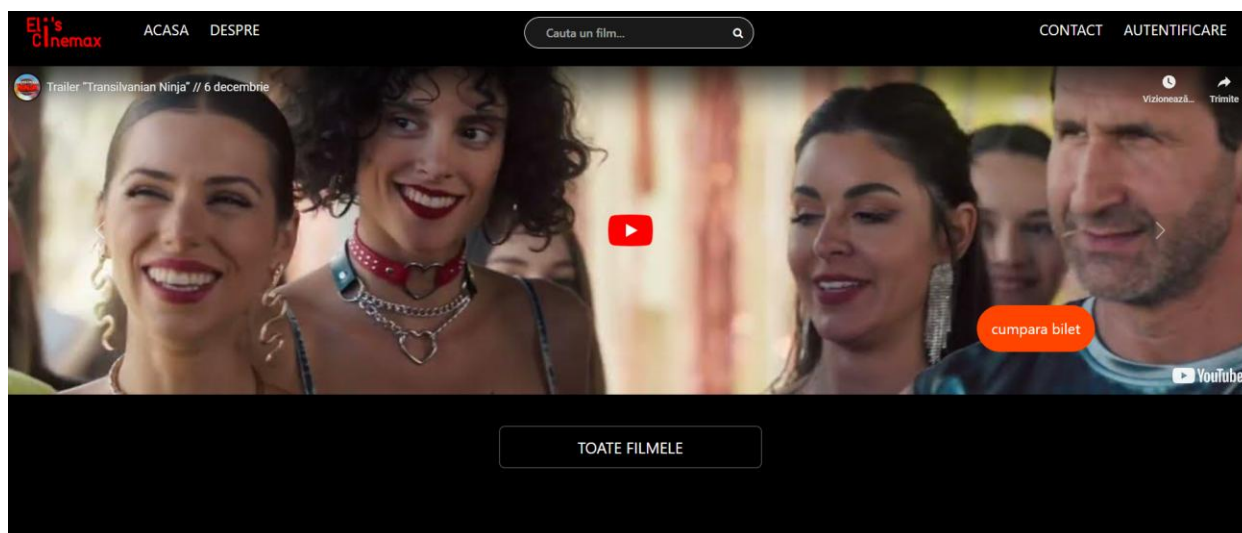


Fig.2.1 Pagina Acasa

2.2. Pagina de Filme

Pagina de Filme este punctul central al aplicației, oferind utilizatorilor posibilitatea de a explora întreaga listă de filme disponibile în cinema. Structura acestei pagini a fost concepută pentru a facilita navigarea și selecția rapidă a filmelor preferate. Filmele sunt organizate în categorii, în funcție de genuri (acțiune, comedie, dramă etc.) sau alte criterii, ceea ce permite utilizatorilor să găsească rapid ceea ce caută. Fiecare film este reprezentat printr-un card care include un poster atrăgător, titlul filmului, o scurtă descriere sau trailer și un buton „Detalii” pentru informații suplimentare. Pagina include, de asemenea, funcționalități de filtrare și căutare, care permit utilizatorilor să sorteze filmele în funcție de diverse criterii, cum ar fi data premierei, popularitatea sau evaluările. Această pagină este integrată cu baza de date, permițând afișarea dinamică a filmelor și actualizarea informațiilor în timp real.

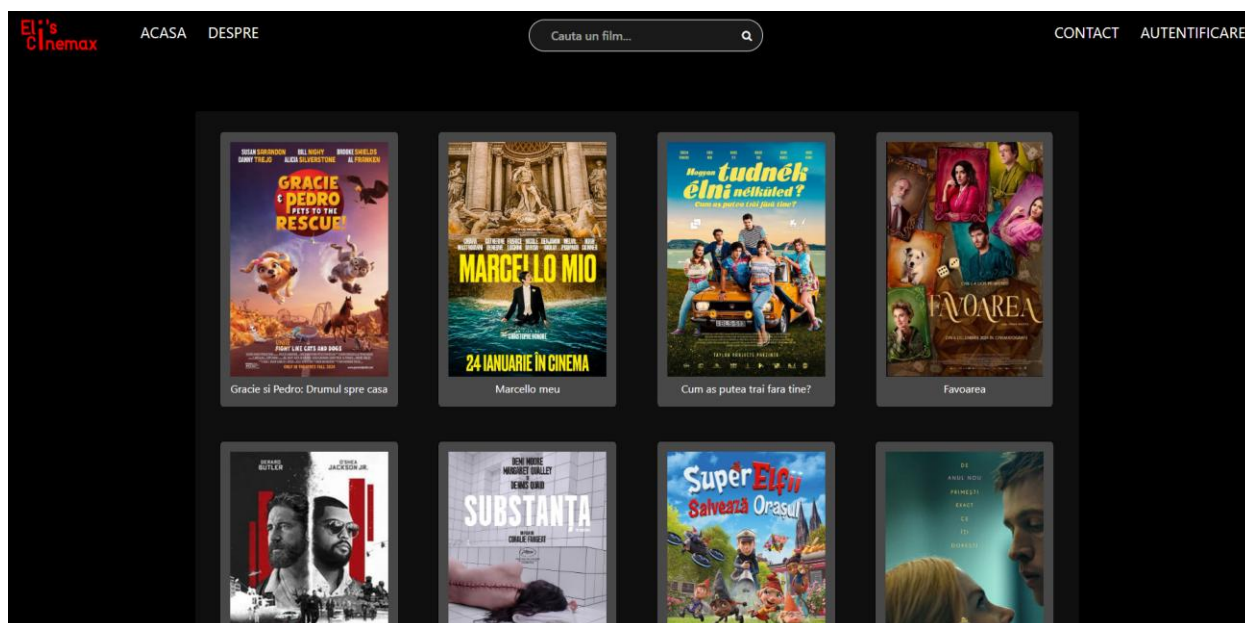


Fig.2.2 Pagina Filme

2.3. Pagina Despre

Pagina Despre este dedicată furnizării de informații esențiale despre cinema, având rolul de a construi încredere și de a adăuga un aspect personal aplicației. Aceasta oferă o scurtă poveste despre începuturile și evoluția cinematografului, subliniind momentele importante care au marcat dezvoltarea sa. De asemenea, include o prezentare a misiunii și valorilor cinema-ului, detaliind obiectivele și angajamentul față de oferirea unei experiențe cinematografice de calitate. Pagina este completată de fotografii reprezentative, care ilustrează interioarele cinema-ului, sălile de proiecție și alte facilități. Astfel, utilizatorii pot obține o imagine completă și autentică a cinema-ului, ceea ce întărește legătura cu aceștia.

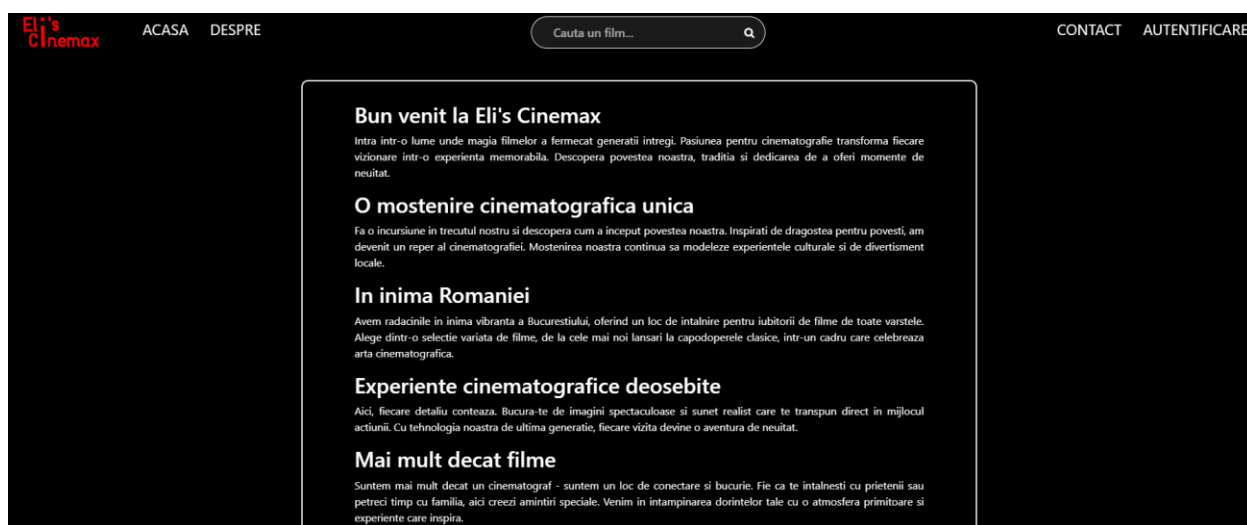


Fig.2.3 Pagina Despre

2.4. Pagina de Contact

Pagina de Contact oferă utilizatorilor un mod simplu și eficient de a intra în legătură cu echipa Eli's Cinemax. Aceasta include informații esențiale care facilitează comunicarea și rezolvarea oricăror întrebări sau cerințe din partea clienților.

În secțiunea „Telefon”, utilizatorii pot găsi numărul de contact al cinema-ului, oferind o modalitate rapidă de a obține asistență sau informații suplimentare.

„Adresa” indică locația exactă a cinema-ului, în București, pe Aleea Pravat nr 1, iar „Mail” oferă o adresă de contact pentru orice solicitări prin e-mail.

De asemenea, harta interactivă integrată pe pagină le permite utilizatorilor să localizeze rapid cinema-ul și să obțină indicații precise de orientare, contribuind astfel la o experiență de utilizare mai ușoară și mai eficientă.

Această pagină este ușor accesibilă din meniul principal, asigurând un punct de contact rapid și clar între utilizatori și echipa de suport, pentru orice întrebare sau problemă.

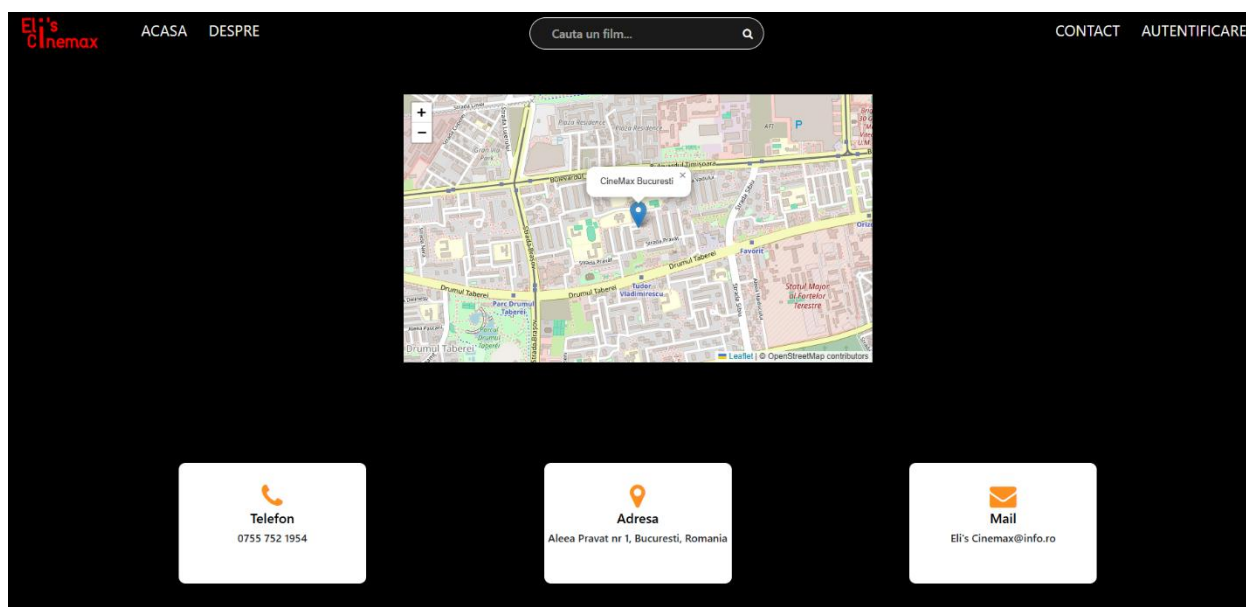


Fig.2.4 Pagina de Contact

2.5. Autentificare/Înregistrare

Sistemul de autentificare și înregistrare este crucial pentru gestionarea utilizatorilor și accesul la funcționalități personalizate.

2.5.1. Pagina de Autentificare

Pagina de Autentificare este o componentă esențială a aplicației, permițând utilizatorilor să acceseze funcționalitățile personalizate ale platformei. Aceasta oferă o interfață intuitivă și bine structurată, care simplifică procesul de conectare la conturile existente.

Pagina de autentificare permite utilizatorilor să acceseze conturile lor într-un mod rapid și sigur. Aceasta include câmpuri clare pentru introducerea datelor de autentificare, cum ar fi adresa de email și parola asociată contului. Fiecare câmp este bine marcat și validat pentru a preveni eventualele erori de introducere a datelor.

Pentru cazurile în care utilizatorii uită parola, pagina include o opțiune vizibilă „Am uitat parola”, care le permite să inițieze procesul de resetare a parolei. Prin intermediul acestei funcționalități, utilizatorii primesc un email securizat în care sunt ghidați pentru a-și crea o parolă nouă, asigurându-se astfel accesul rapid și sigur la conturile lor. După completarea câmpurilor de autentificare, utilizatorii pot apăsa butonul „Login”, care trimite datele și declanșează procesul de validare a acestora. Dacă autentificarea este reușită, utilizatorii sunt redirecționați către pagina principală a platformei sau către secțiunile disponibile exclusiv celor conectați.

Pentru o experiență optimă, pagina este optimizată atât pentru desktop, cât și pentru dispozitive mobile, asigurând accesibilitate și ușurință în utilizare indiferent de platforma folosită. În plus, designul său modern și minimalist asigură o interfață vizuală atractivă, care se aliniază cu identitatea aplicației.

Această pagină contribuie la securitatea aplicației prin autentificare și gestionează accesul utilizatorilor la secțiuni dedicate, consolidând astfel experiența generală a platformei.

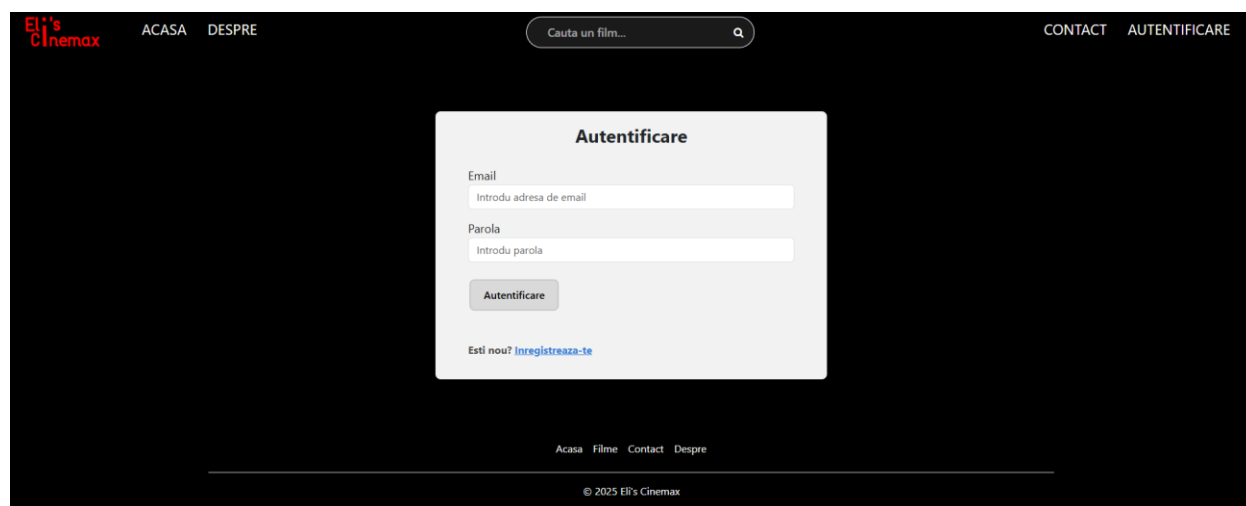


Fig.2.5 Pagina de Autentificare

2.5.2. Pagina de Înregistrare

Pagina de Înregistrare este destinată celor care doresc să își creeze un cont nou pe platformă, oferindu-le acces personalizat la funcționalitățile aplicației. Aceasta este concepută pentru a fi intuitivă și sigură, punând un accent deosebit pe validarea corectă a datelor introduse.

Utilizatorii completează un formular care include câmpuri esențiale pentru crearea unui cont, precum numele complet, adresa de email validă, parola personalizată și confirmarea parolei pentru a preveni eventualele greșeli.

Formularul include verificări automate care asigură corectitudinea și securitatea datelor introduse. Printre aceste verificări se numără validarea formatului adresei de email, respectarea criteriilor de securitate pentru parolă (cum ar fi o lungime minimă și utilizarea caracterelor speciale) și confirmarea că adresa de email nu este deja asociată unui alt cont existent.

Pentru a ajuta utilizatorii să corecteze eventualele erori, pagina include mesaje de feedback în timp real. Dacă datele introduse sunt incorecte sau incomplete, aceștia sunt informați imediat prin mesaje vizuale prietenoase.

După completarea formularului, utilizatorii pot apăsa butonul de înregistrare pentru a trimite datele către server. Dacă datele sunt valide, un cont nou este creat și salvat în baza de date, iar utilizatorul poate accesa funcționalitățile platformei.

Pagina de înregistrare este proiectată să funcționeze optim pe toate dispozitivele, asigurând accesibilitate și o experiență plăcută, indiferent dacă utilizatorii folosesc un computer desktop sau un dispozitiv mobil.

Odată finalizat procesul de înregistrare, utilizatorii primesc un mesaj de confirmare și sunt redirecționați automat către pagina de autentificare sau către o pagină de bun venit, unde li se oferă informații despre funcționalitățile platformei și următorii pași pe care îi pot urma.

Prin intermediul acestei pagini, platforma facilitează o interacțiune ușoară și sigură, asigurând utilizatorilor un început plăcut și personalizat în explorarea aplicației.

The image shows a web page for registering an account on the El's Cinemax platform. The page has a dark theme. At the top, there is a navigation bar with the logo 'El's Cinemax' on the left, and links for 'ACASA', 'DESPRE', a search bar with the placeholder 'Cauta un film...', 'CONTACT', and 'AUTENTIFICARE' on the right. In the center of the page is a white registration form titled 'Inregistreaza un cont'. The form contains five input fields: 'Prenume' (First Name) with placeholder 'Introdu prenumele', 'Nume' (Last Name) with placeholder 'Introdu numele', 'Email' with placeholder 'Introdu adresa de email', 'Parola' (Password) with placeholder 'Creeaza o parola', and 'Confirma parola' (Confirm password) with placeholder 'Introdu parola din nou'. Below these fields is a 'Trimite' (Send) button. At the bottom of the form, there is a link that says 'Ai deja un cont? Autentifica-te'.

Fig.2.6 Pagina de Înregistrare

3. BAZA DE DATE

Baza de date joacă un rol esențial în gestionarea și stocarea tuturor informațiilor necesare pentru funcționarea eficientă a aplicației. În acest context, sunt stocate informațiile despre filme, utilizatori, rezervări și programări, toate acestea fiind interconectate printr-o structură bine definită de tabele și relații. Vom detalia fiecare tabel și legăturile dintre ele pentru a înțelege cum sunt organizate datele și cum interacționează între ele [3].

Definiția unei baze de date relaționale

O bază de date relațională este un sistem de gestionare a datelor care utilizează tabele pentru a organiza informațiile. Aceste tabele sunt formate din coloane (câmpuri) și rânduri (înregistrări), iar datele sunt stocate sub formă de valori care sunt structurate și interconectate între tabele prin relații. În cazul aplicației Eli's Cinemax, fiecare tabel conține informații esențiale despre filme, utilizatori, rezervări și altele, iar legăturile dintre tabele sunt realizate prin chei primare și chei externe [4].

Teorie - structura bazei de date

A. Tabelul bookings

Acest tabel stochează informațiile legate de rezervările efectuate de utilizatori. Fiecare rezervare este identificată printr-un ID unic și este legată de utilizator, sală, film și locuri. De asemenea, sunt înregistrate detalii despre ora rezervării și statusul acesteia.

Coloane principale:

- booking_id (ID unic al rezervării - cheie primară)
- user_id (ID-ul utilizatorului - cheie externă)
- rooms_id (ID-ul sălii - cheie externă)
- movie_id (ID-ul filmului - cheie externă)
- seat_id (ID-ul locului - cheie externă)
- showtime_id (ID-ul orei de proiecție - cheie externă)
- booking_time (Ora rezervării)
- status (Statusul rezervării: 0 - inactiv, 1 - activ)

B. Tabelul movies

Acest tabel stochează informațiile despre filmele disponibile. Fiecare film este identificat printr-un ID unic, iar informațiile includ titlul, descrierea, imaginea asociată și prețul filmului.

Coloane principale:

- movie_id (ID unic al filmului - cheie primară)
- title (Titlul filmului)
- description (Descrierea filmului)
- image_path (Calea către imaginea asociată filmului)
- movie_price (Prețul filmului)

C. Tabelul rooms

Acest tabel stochează informațiile despre sălile de cinema. Fiecare sală este identificată printr-un ID unic și este asociată cu un film specific.

Coloane principale:

- rooms_id (ID unic al sălii - cheie primară)
- RoomName (Numele sălii)
- movie_id (ID-ul filmului proiectat în sală - cheie externă)

D. Tabelul seats

Acest tabel stochează informațiile despre locurile disponibile în sălile de cinema. Fiecare loc este identificat printr-un ID unic și este asociat cu o sală specifică.

Coloane principale:

- seat_id (ID unic al locului - cheie primară)
- room_id (ID-ul sălii - cheie externă)
- seat_number (Numărul locului)
- status (Statusul locului: 'available', 'booked')

E. Tabelul showtimes

Acest tabel stochează informațiile despre orele de proiecție ale filmelor. Fiecare oră de proiecție este asociată cu un film și o sală.

Coloane principale:

- showtime_id (ID unic al orei de proiecție - cheie primară)
- movie_id (ID-ul filmului - cheie externă)
- room_id (ID-ul sălii - cheie externă)
- showtime (Data și ora proiecției)

F. Tabelul tickets

Acest tabel stochează informațiile despre bilete. Fiecare bilet este legat de o rezervare și include informații despre utilizator.

Coloane principale:

- ticket_id (ID unic al biletului - cheie primară)
- booking_id (ID-ul rezervării - cheie externă)
- customer_name (Numele clientului)
- customer_lname (Prenumele clientului)
- customer_email (Email-ul clientului)
- customer_phone (Telefonul clientului)
- customer_address (Adresa clientului)

G. Tabelul users

Acest tabel stochează informațiile despre utilizatorii aplicației. Fiecare utilizator este identificat printr-un ID unic și are roluri definite pentru a controla accesul la funcționalitățile aplicației.

Coloane principale:

- id (ID unic al utilizatorului - cheie primară)
- firstname (Prenumele utilizatorului)
- lastname (Numele utilizatorului)
- email (Email-ul utilizatorului)
- password (Parola utilizatorului)
- reg_date (Data înregistrării utilizatorului)
- role (Rolul utilizatorului: 'user', 'worker', 'admin')

H. Tabelul workers

Acest tabel stochează informațiile despre angajații cinema-ului. Fiecare angajat este identificat printr-un ID unic.

Coloane principale:

- id (ID unic al angajatului - cheie primară)
- firstname (Prenumele angajatului)
- lastname (Numele angajatului)
- email (Email-ul angajatului)
- password (Parola angajatului)
- reg_date (Data înregistrării angajatului)

Legături între Tabele

Relația între tabelele movies și bookings

Tabelul bookings include o cheie externă (movie_id) care face referire la tabelul movies. Fiecare rezervare este asociată cu un film specific din lista de filme existente în sistem. Cheie externă: bookings.movie_id → movies.movie_id.

Relația între tabelele users și bookings

Tabelul bookings include o cheie externă (user_id) care face referire la tabelul users. Fiecare rezervare este asociată cu un utilizator specific. Sistemul poate urmări toate rezervările făcute de un anumit utilizator. Cheie externă: bookings.user_id → users.id.

Relația între tabelele rooms și bookings

Tabelul bookings include o cheie externă (rooms_id) care face referire la tabelul rooms. Fiecare rezervare este asociată cu o sală specifică. Cheie externă: bookings.rooms_id → rooms.rooms_id.

Relația între tabelele rooms și movies

Tabelul rooms include o cheie externă (movie_id) care face referire la tabelul movies. Fiecare sală este asociată cu un film care rulează în respectiva sală. Cheie externă: rooms.movie_id → movies.movie_id.

Relația între tabelele seats și rooms

Tabelul seats include o cheie externă (room_id) care face referire la tabelul rooms. Fiecare loc este asociat cu o sală specifică. Cheie externă: seats.room_id → rooms.rooms_id.

Relația între tabelele showtimes și movies

Tabelul showtimes include o cheie externă (movie_id) care face referire la tabelul movies. Fiecare oră de proiecție este asociată cu un film specific. Cheie externă: showtimes.movie_id → movies.movie_id.

Relația între tabelele showtimes și rooms

Tabelul showtimes include o cheie externă (room_id) care face referire la tabelul rooms. Fiecare oră de proiecție este asociată cu o sală specifică unde se rulează filmul. Cheie externă: showtimes.room_id → rooms.rooms_id.

Relația între tabelele tickets și bookings

Tabelul tickets include o cheie externă (booking_id) care face referire la tabelul bookings. Fiecare bilet este legat de o rezervare specifică. Cheie externă: tickets.booking_id → bookings.booking_id.

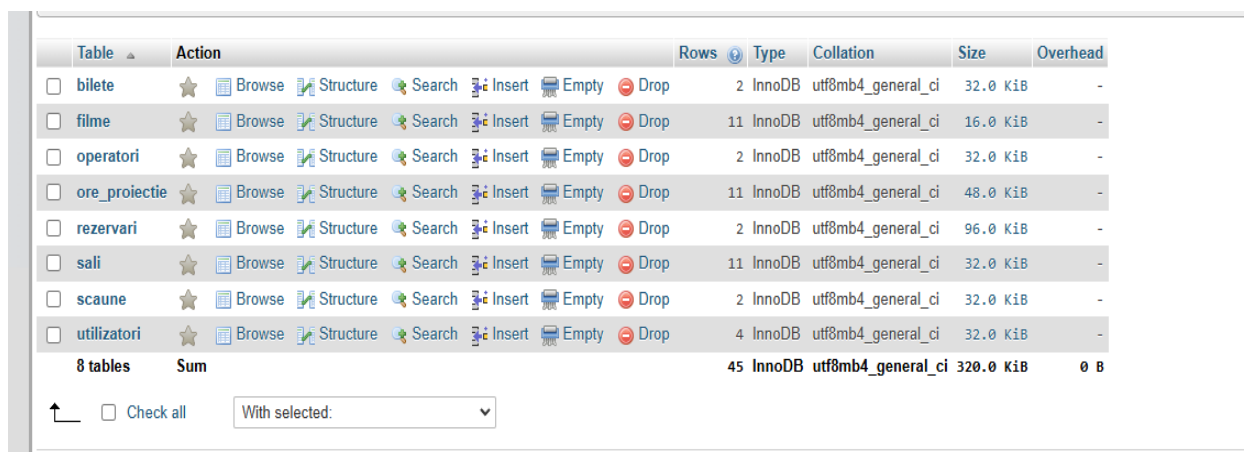


Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> bilete	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> filme	★ Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> operatori	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> ore_proiectie	★ Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> rezervari	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	96.0 KiB	-
<input type="checkbox"/> sali	★ Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> scaune	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> utilizatori	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
8 tables	Sum	45	InnoDB	utf8mb4_general_ci	320.0 KiB	0 B

☐ Check all With selected: ▾

Fig.3.1 Continutul bazei de date

4. PROGRAMARE ORIENTATA OBIECT

În cadrul dezvoltării aplicației Eli's Cinemax, am folosit principiile programării orientate pe obiect (OOP) pentru a organiza și structura eficient datele și funcționalitățile. Programarea orientată pe obiecte permite crearea unor componente (clase) care definesc comportamentele și atributele obiectelor din aplicație. În acest context, utilizatorii, filmele și rezervările sunt reprezentate de obiecte care interacționează între ele într-un mod structurat și coerent. Diagrama UML (Unified Modeling Language) este un instrument esențial pentru vizualizarea acestei structuri, deoarece ajută la descrierea componentelor aplicației și relațiilor dintre acestea.

4.1. Prezentarea diagramei UML

Diagrama UML (Unified Modeling Language) este un tip de diagramă care ajută la descrierea structurii unui sistem software, identificând clasele, atributele acestora, metodele și relațiile dintre ele. În cadrul aplicației Eli's Cinemax, diagramele UML sunt folosite pentru a reprezenta clasele principale, cum ar fi Utilizator, Film și Rezervare, și pentru a arăta interacțiunile dintre acestea. Diagrama UML este esențială în etapele de design ale aplicației și ajută dezvoltatorii să vizualizeze structura aplicației înainte de a scrie codul efectiv.

Diagrama UML a aplicației Eli's Cinemax include următoarele clase și relații:

1. **Clasa Utilizator:** Reprezintă utilizatorii aplicației, incluzând informații precum numele, adresa de email, parola și rolul. Aceștia pot avea mai multe rezervări.
2. **Clasa Film:** Reprezintă filmele disponibile în cinema, cu atribute cum ar fi titlul, descrierea, genul, durata, rating-ul și data lansării.
3. **Clasa Rezervare:** Legătura dintre un utilizator și un film, stocând informații despre numărul de locuri rezervate, data proiecției și filmul asociat.

4.2. Explicația diagramei UML

Diagrama UML a aplicației Eli's Cinemax reflectă relațiile și interacțiunile dintre clasele principale. În această diagramă, fiecare clasă este reprezentată sub forma unui dreptunghi care conține atributele și metodele relevante pentru acea clasă.

1. **Clasa Utilizator** este conectată la clasa Rezervare printr-o relație de tip 1:M (un utilizator poate face mai multe rezervări). Această relație este indicată printr-o linie care leagă cele două clase și este marcată cu o cifră "1" lângă clasa Utilizator și "M" lângă clasa Rezervare.
2. **Clasa Film** este, de asemenea, conectată la **Clasa Rezervare** printr-o relație 1:M, deoarece un film poate fi rezervat de mai mulți utilizatori.
3. **Clasa Rezervare** conține două chei externe, `id_utilizator` și `id_film`, care leagă rezervarea de utilizatorul care a făcut-o și filmul rezervat.

Această structură permite urmărirea tuturor rezervărilor realizate de un utilizator și asocierea acestora cu filmele corespunzătoare.

Explicația Relațiilor

- **Relația dintre Utilizator și Rezervare:** Un utilizator poate face mai multe rezervări, astfel că există o relație de tip 1:M între Utilizator și Rezervare.
- **Relația dintre Film și Rezervare:** Un film poate fi rezervat de mai mulți utilizatori, astfel că există o relație de tip 1:M între Film și Rezervare.
- **Chei externe:** Tabelul Rezervare conține două chei externe - id_utilizator și id_film - care leagă rezervarea de utilizatorul care a făcut-o și de filmul rezervat.

Această diagramă UML este esențială pentru înțelegerea arhitecturii aplicației și pentru implementarea corectă a funcționalităților acesteia. Relațiile între clase sunt clare, iar această structură face ca gestionarea datelor să fie eficientă și ușor de extins pe măsură ce aplicația crește.

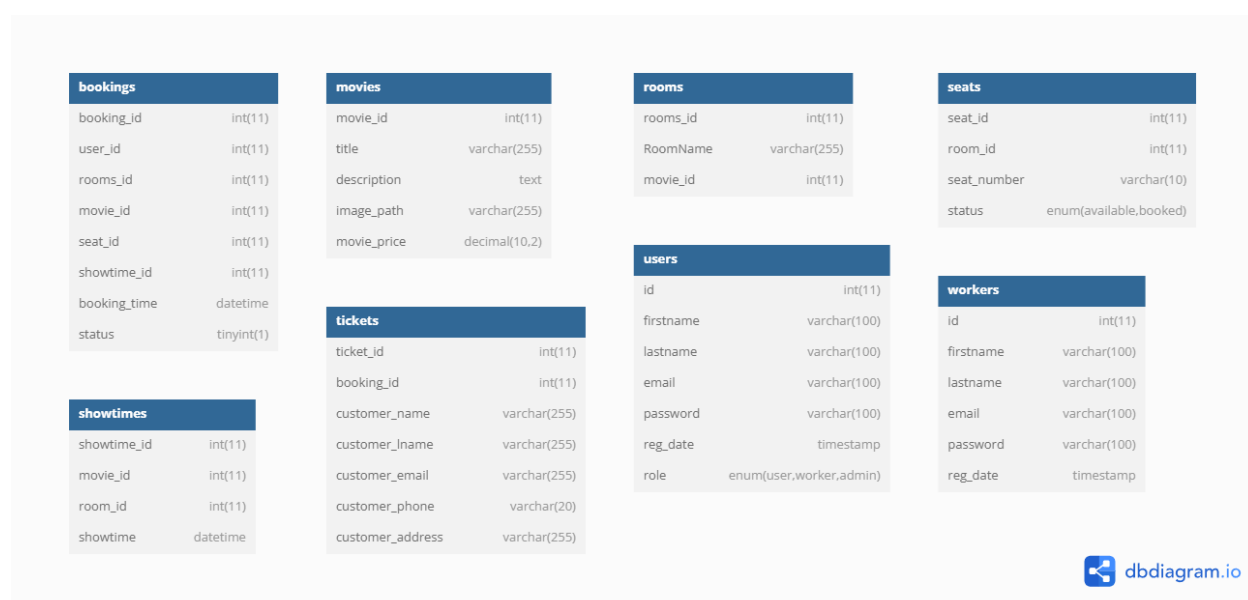


Fig.4.1 Diagrama UML a site-ului

Diagrama UML oferă o reprezentare vizuală clară a structurii și relațiilor dintre entitățile cheie ale aplicației dezvoltate pentru gestionarea unui cinema. Această diagramă ajută la înțelegerea modului în care componentele aplicației interacționează între ele și cum sunt organizate din punct de vedere logic.

Entitățile principale din diagrama UML sunt: film reprezintă un film disponibil pentru vizionare, cu proprietăți precum titlu, gen, durată, descriere, rating și imagine (poster). User (Utilizator) reprezintă un utilizator al aplicației (client, worker sau admin), cu proprietăți precum nume, adresă de email, rol și parolă. Ticket (Bilet) reprezintă un bilet emis pentru un anumit film, cu proprietăți precum număr unic, preț, loc și oră de vizionare. Showtime (Oră de proiecție) reprezintă o sesiune de proiecție pentru un film, cu proprietăți precum oră, dată și sală de proiecție. CinemaRoom (Sală de cinema) reprezintă o sală de proiecție din cadrul cinema-ului, cu proprietăți precum număr alocat, capacitate și echipamente disponibile. Booking (Rezervare) reprezintă o rezervare efectuată de un utilizator, cu proprietăți precum număr de rezervare, utilizator asociat și bilete incluse.

Relațiile dintre entități sunt următoarele: Relația între User și Booking este o relație de agregare, deoarece un utilizator poate efectua mai multe rezervări, dar fiecare rezervare aparține unui singur utilizator. Relația între Movie și Showtime este o relație de asociere, deoarece un film poate avea mai multe ore de proiecție, iar fiecare oră de proiecție este legată de un singur film. Relația între Showtime și CinemaRoom este o relație de asociere, deoarece o sesiune de proiecție este programată într-o singură sală, iar o sală poate găzdui mai multe sesiuni. Relația între Showtime și Ticket este o relație de agregare, deoarece fiecare sesiune de proiecție are mai multe bilete asociate, iar un bilet este legat de o singură sesiune. Relația între Booking și Ticket este o relație de compoziție, deoarece biletele sunt parte integrantă a unei rezervări și nu pot exista independent de aceasta.

Explicația relațiilor din diagrama UML: Relațiile de agregare indică faptul că entitățile sunt conectate, dar fiecare poate exista independent. De exemplu, un utilizator poate exista chiar dacă nu are nicio rezervare. Relațiile de asociere subliniază legături simple între entități, fără a implica o dependență strictă, cum ar fi relația dintre film și ora de proiecție. Relațiile de compoziție indică o legătură puternică, în care entitatea dependentă nu poate exista fără entitatea principală. De exemplu, biletele sunt parte integrantă a unei rezervări și nu pot exista fără aceasta.

Importanța diagramei UML:

Diagrama UML oferă o privire de ansamblu asupra structurii aplicației, facilitând procesul de proiectare și dezvoltare. Prin definirea entităților și a relațiilor dintre ele, această diagramă asigură o bază solidă pentru implementarea bazei de date și a logicii aplicației.

Această reprezentare vizuală ajută echipa de dezvoltare să înțeleagă mai bine fluxurile de lucru, să identifice eventualele conflicte sau redundanțe și să optimizeze structura generală a aplicației pentru a răspunde nevoilor utilizatorilor.

5. CONCEPEREA PRODUSELOR MULTIMEDIA

Pentru a îmbunătăți aspectul vizual și atractivitatea aplicației, au fost utilizate tehnici avansate de editare grafică prin intermediul programului Adobe Photoshop. Acest proces a contribuit la optimizarea imaginilor incluse în platformă, atât din punct de vedere estetic, cât și funcțional.



Fig.5.1 Pregătire imagine cover film

6. CONCLUZII, CONTRIBUTII PERSONALE SI PERSPECTIVE DE VIITOR

6.1. Concluzii

Prin dezvoltarea acestei aplicații, am reușit să ating obiectivele propuse, contribuind la crearea unei soluții complete și eficiente pentru gestionarea și organizarea activităților unui cinema. Implementarea funcționalităților esențiale, precum rezervarea билетelor online, administrarea filmelor și gestionarea utilizatorilor, a demonstrat viabilitatea și utilitatea proiectului.

Un aspect important al proiectului a fost optimizarea interfeței utilizator, care asigură o navigare intuitivă și o experiență plăcută pentru utilizatori. Am acordat o atenție deosebită designului, funcționalităților dinamice și integrării tehnologiilor moderne, pentru a oferi o platformă adaptabilă și accesibilă.

Un alt rezultat semnificativ al proiectului este implementarea unei baze de date robuste și bine structurate. Aceasta permite gestionarea eficientă a datelor privind utilizatorii, filmele și rezervările, contribuind la o performanță ridicată și la reducerea timpilor de răspuns.

Prin utilizarea unor tehnologii moderne precum PHP, MySQL și JavaScript, am reușit să creez o aplicație web dinamică și interactivă. Acest proiect a oferit o oportunitate valoroasă de a învăța și aplica cele mai bune practici în dezvoltarea software, consolidându-mi competențele tehnice și abilitățile de rezolvare a problemelor.

6.2. Contributii personale

Pe parcursul dezvoltării acestui proiect, am adus contribuții semnificative care au îmbunătățit funcționalitatea și performanța aplicației. Am conceput și implementat o interfață utilizator prietenoasă și intuitivă, adaptată atât nevoilor utilizatorilor obișnuiți, cât și celor ale administratorilor platformei. În acest scop, am utilizat limbaje și tehnologii moderne, cum ar fi HTML, CSS și JavaScript, pentru a asigura o experiență de utilizare fluidă și plăcută.

Am proiectat și realizat o arhitectură a aplicației clară și scalabilă, care permite extinderea ulterioară a funcționalităților fără a afecta stabilitatea sau performanța sistemului. Totodată, am creat o structură de baze de date robustă, folosind MySQL, care stochează și gestionează eficient informațiile legate de utilizatori, filme, rezervări și administrare.

De asemenea, am dezvoltat funcționalități personalizate, cum ar fi un sistem de autentificare pentru utilizatori cu diferite niveluri de acces, gestionarea dinamică a rezervărilor și notificările pentru utilizatori. Aceste funcționalități au fost esențiale în transformarea platformei într-un instrument complex și util, capabil să răspundă nevoilor diversificate ale utilizatorilor.

Această experiență mi-a permis să aprofundez cunoștințele în limbaje de programare și tehnologii web moderne, consolidându-mi competențele tehnice și capacitatea de a gestiona proiecte complexe. Fiecare etapă a proiectului a contribuit la dezvoltarea mea profesională, oferindu-mi o mai bună înțelegere a procesului de creare a aplicațiilor web complete și funcționale.

6.3. Perspective de viitor

Pentru dezvoltările viitoare ale platformei, există multiple direcții de extindere și optimizare care pot aduce valoare adăugată utilizatorilor și administratorilor. O prioritate majoră este implementarea unui sistem de plată online, care să permită utilizatorilor achiziționarea de bilete direct de pe platformă, sporind astfel accesibilitatea și eficiența. Această funcționalitate va facilita procesul de rezervare și va reduce efortul necesar pentru utilizatori.

De asemenea, integrarea unui sistem de recomandări personalizate, bazat pe preferințele și istoricul utilizatorilor, ar putea contribui la o experiență mai interactivă și plăcută. Prin utilizarea tehnicilor de inteligență artificială, platforma ar putea sugera filme relevante și atrăgătoare, consolidând astfel angajamentul utilizatorilor.

Optimizarea suplimentară a platformei pentru dispozitive mobile reprezintă o altă direcție importantă. Crearea unei aplicații mobile dedicate sau îmbunătățirea versiunii web pentru o navigare mai rapidă și mai intuitivă pe dispozitivele mobile ar asigura o accesibilitate sporită și ar răspunde cerințelor actuale ale utilizatorilor.

Un alt aspect important îl constituie introducerea unui sistem avansat de raportare, care să ofere administratorilor informații detaliate despre vânzările de bilete, veniturile generate și alte statistici relevante. Aceste date ar facilita luarea deciziilor și optimizarea activităților administrative [12].

Pe termen lung, este esențială integrarea unui sistem de gestionare a stocurilor și a comenzilor, care să asigure o monitorizare eficientă și să prevină eventualele probleme de disponibilitate a resurselor. În plus, îmbunătățirea securității datelor utilizatorilor prin implementarea unor protocoale avansate de protecție reprezintă o prioritate în contextul creșterii preocupărilor legate de confidențialitatea informațiilor.

Aceste perspective de viitor oferă oportunități valoroase pentru dezvoltarea și perfecționarea platformei, contribuind la crearea unei experiențe mai complexe și mai satisfăcătoare pentru toți utilizatorii implicați.

ANEXE

Această secțiune oferă o explicație detaliată a fiecărei părți din codul scris. Structurat pentru a reprezenta entitățile principale, cum ar fi filmele, utilizatorii, biletele, rezervările și sălile de cinema.

Pagina acasa.php

Fișierul acasa.php reprezintă pagina principală oferind utilizatorilor o listă de filme populare preluate din baza de date. Fiecare film este afișat sub forma unui card cu imagine și titlu, permițând accesul rapid la detalii suplimentare despre film. De asemenea, pagina integrează navigația între secțiunile site-ului și suport pentru paginare.

```
<?php session_start(); // Inceperea sesiunii pentru a retine ID-ul si numele complet al utilizatorului
require "../Tables-MakeDB/makeDBConnection.php";
require "../html/navBar.php";
require "../html/carusel.php";
?>
<head>
    <link rel="stylesheet" href="../css/navBar.css">
    <link rel="stylesheet" href="../css/imagesStyle.css">
    <link rel="stylesheet" type="text/css" href="../css/style.css">
    <link rel="icon" type="image/png" href="../imagini/clapperboard.png">
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Eli's Cinemax</title>
</head>
<body style="background-color: black;">
    <?php
    if ($conn->connect_error) {
        die("Conexiunea a esuat: " . $conn->connect_error);
    }
    // Primele 8 filme din baza de date
    $sql = "SELECT movie_id, title, image_path FROM movies LIMIT 8";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        // Inceperea containerului pentru toate filmele
        echo "<div class='grid-container container'>";
        while ($row = $result->fetch_assoc()) {
            // Fiecare film are propriul card
            echo "<div class='movie-card'>";
            // Crearea formularelor pentru fiecare film
            echo "<form action='detalii.php' method='get'>";
            echo "<input type='hidden' name='title' value='" . htmlspecialchars($row["title"]) . "'>";
```



```

// Container pentru imagine si titlu pentru o mai buna stilizare
echo "<div class='movie-img-container'>";
echo "<button type='submit' class='movie_img'>";
echo "<img src='" . htmlspecialchars($row["image_path"]) . "' alt='Imagine Film'>";
echo "</button>";
echo "<div>" . htmlspecialchars($row["title"]) . "</div>";
echo "</div>";
echo "</form>";
// Inchiderea div-ului pentru cardul filmului
echo "</div>";
}
// Inchiderea div-ului pentru grid-container
echo "</div>";
} else {
    echo "<div>Nu s-au gasit rezultate</div>";
}
?>
<div class="num-container">
    <form action="acasa.php" method="GET">
        <button type="submit" class="num-button current" id="Page1">1</button>
    </form>
    <form action="maiMulteFilme2.php" method="GET">
        <button type="submit" class="num-button" id="Page2">2</button>
    </form>
    <form action="maiMulteFilme3.php" method="GET">
        <button type="submit" class="num-button" id="Page3">3</button>
    </form>
</div>

<div class="container">
    <footer class="py-3 my-4">
        <ul class="nav justify-content-center border-bottom pb-3 mb-3">
            <li class="nav-item"><a href="acasa.php" class="nav-link px-2 text-white">Acasa</a></li>
            <li class="nav-item"><a href="maiMulteFilme2.php" class="nav-link px-2 text-white">Filme</a></li>
            <li class="nav-item"><a href="contact.php" class="nav-link px-2 text-white">Contact</a></li>
            <li class="nav-item"><a href="despre.php" class="nav-link px-2 text-white">Despre noi</a></li>
        </ul>
        <p class="text-center text-white">&copy; <?php echo date("Y"); ?> Eli's Cinemax</p>
    </footer>

```

```

</div>
<?php
// Verificare daca formularul de cautare a fost trimis si campul movie_name nu este gol
if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET["movie_name"])) {
    $movieName = $_GET["movie_name"];

    if (!empty($movieName)) {
        header("Location: cautare.php?movie_name=" . urlencode($movieName));
        exit();
    } else {
        echo 'Cautare invalida';
    }
}
?>
</body>
</html>

```

Asezari scaune

Fișierul `asezari_scaune.php` gestionează afișarea și selectarea locurilor disponibile pentru un anumit film, oră de proiecție și sală. Acesta oferă utilizatorilor posibilitatea de a rezerva un scaun într-un mod interactiv, afișând un plan vizual al sălii de cinema. Sistemul validează selecția locurilor și redirecționează utilizatorii autentificați către pagina de plată.

```

<?php
session_start(); // Inceperea sesiunii pentru a retine ID-ul si numele complet al utilizatorului
require "../Tables-MakeDB/makeDBConnection.php";
require "../html/navBar.php";
?>
<!DOCTYPE html>
<html>
<head>
    <title>Plan scaunuri Cinema</title>
    <link rel="stylesheet" type="text/css" href="../css/style.css">
    <link rel="stylesheet" href="../css/navBar.css">
    <link rel="stylesheet" href="../css/imagesStyle.css">
<body style="background-color: black;">
    <style>
        .seating-plan {
            display: grid;
            grid-template-columns: repeat(8, calc((100% - 70px) / 8));
            gap: 10px;
            max-width: 600px;
            margin: 0 auto;

```

```

background-color: lightgray;
padding: 10px;
grid-template-rows: repeat(4, 1fr);
position: relative;
margin-top: 80px;
border-radius: 6px;
}
.seat {
  display: none;
}
.seat+label {
  appearance: none;
  padding: 10px;
  text-align: center;
  background-color: #f0f0f0;
  border: 1px solid #ddd;
  cursor: pointer;
  display: inline-block;
  margin: 0;
  box-sizing: border-box;
  display: flex;
  align-items: center;
  justify-content: center;
}
.seat+label:hover {
  background-color: #a8a8a8;
}
.seat:checked+label {
  background-color: #a0a0f0;
}
.Isle {
  visibility: hidden;
}
.view_title {
  text-align: center;
  margin-bottom: 55px;
  margin-top: 35px;
  color: white;
}
.movie-details {
  color: white;
  margin-left: 90px;
}

```

```

.seating-plan {
  position: relative;
  margin-top: 80px;
}
.seating-plan::before {
  content: 'Ecran';
  position: absolute;
  top: -60px;
  left: 0;
  right: 0;
  height: 50px;
  line-height: 40px;
  background-color: #333;
  border-radius: 50%;
  color: #fff;
  text-align: center;
  font-size: 28px;
  z-index: -1;
}
@media (max-width: 600px) {
  .seating-plan {
    width: 95%;
  }
  .seat+label {
    padding: 5px;
    font-size: 13px;
  }
}
.seatbtn {
  margin-top: 17px;
  margin-bottom: 10px;
  background-color: #333;
  color: white;
  padding: 12px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
.seatbtn:hover {
  background-color: #555;
}
</style>

```

```

</head>
<?php
if (isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true) {
    if ($_SERVER["REQUEST_METHOD"] == "GET") {
        $title = isset($_GET['title']) ? htmlspecialchars($_GET['title']) : "";
        $showtime = isset($_GET['showtime']) ? htmlspecialchars($_GET['showtime']) : "";
        $room = isset($_GET['room']) ? htmlspecialchars($_GET['room']) : "";
        echo "<div class='movie-details'>";
        echo "<h2 class='movie-title'>Rezervare pentru: $title</h2>";
        echo "<p class='movie-showtime'>Ora: $showtime</p>";
        echo "<p class='movie-room'>Sala: $room</p>";
        echo "</div>";
    }
}

?>
    <form action="plata.php" method="get" id="bookingForm" onsubmit="return
validateForm()">
        <input type="hidden" name="title" value="<?php echo $title; ?>">
        <input type="hidden" name="showtime" value="<?php echo $showtime; ?>">
        <input type="hidden" name="room" value="<?php echo $room; ?>">
        <div class="seating-plan">
            <?php
            $seats = array(
                "scaun 1", "scaun 2", "scaun 3", "Isle", "Isle", "scaun 4", "scaun 5", "scaun 6",
                "scaun 7", "scaun 8", "scaun 9", "Isle", "Isle", "scaun 10", "scaun 11", "scaun 12",
                "scaun 13", "scaun 14", "scaun 15", "Isle", "Isle", "scaun 16", "scaun 17", "scaun 18",
                "scaun 19", "scaun 20", "scaun 21", "Isle", "Isle", "scaun 22", "scaun 23", "scaun 24"
            );
            foreach ($seats as $index => $seat) {
                if ($seat == "Isle") {
                    echo "<div class='isle-seat'></div>";
                } else {
                    echo "<input type='radio' id='seat$index' class='seat' name='seat' value='$seat'>";
                    echo "<label for='seat$index'>$seat</label>";
                }
            }
        ?>
        <div class="seatbtn-container">
            <input type="submit" name="submit" class="seatbtn">
        </div>
    </div>
</form>
<?php
} else {

```

```

        echo "<script>alert('Trebuie sa fii autentificat pentru a rezerva un bilet!');
window.scaunation.href = 'autentificare.php';</script>";
        echo "<p>Bine ai venit, vizitator! <a href='autentificare.php'>Autentifica-te</a> sau <a
href='inregistrare.php'>Inregistreaza-te</a> pentru mai multe functionalitati.</p>";
    }
    echo "<h2 class='view_title'>VEZI SI</h2>";
    $sql = "SELECT movie_id, title, image_path
FROM movies
ORDER BY RAND()
LIMIT 8;";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        echo "<div class='grid-container container'>";
        while ($row = $result->fetch_assoc()) {
            echo "<div class='movie-card'>";
            echo "<form action='detalii.php' method='get'>";
            echo "<input type='hidden' name='title' value='" . htmlspecialchars($row["title"]) . "'>";
            echo "<div class='movie-img-container'>";
            echo "<button type='submit' class='movie_img'>";
            echo "<img src='" . htmlspecialchars($row["image_path"]) . "' alt='Imagine Film'>";
            echo "</button>";
            echo "<div class='movie-title'>" . htmlspecialchars($row["title"]) . "</div>";
            echo "</div>";
            echo "</form>";
            echo "</div>";
        }
        echo "</div>";
    } else {
        echo "<div>Nu s-au gasit rezultate</div>";
    }
    ?>
<script>
    function validateForm() {
        var seats = document.querySelectorAll('.seat');
        for (var i = 0; i < seats.length; i++) {
            if (seats[i].checked) {
                return true;
            }
        }
        alert('Va rugam selectati un scaun inainte de a trimite!');
        return false;
    }
</script>

```

```

<div class="container">
    <footer class="py-3 my-4">
        <ul class="nav justify-content-center border-bottom pb-3 mb-3">
            <li class="nav-item"><a href="acasa.php" class="nav-link px-2 text-white">Acasa</a></li>
            <li class="nav-item"><a href="maiMulteFilme2.php" class="nav-link px-2 text-white">Filme</a></li>
            <li class="nav-item"><a href="contact.php" class="nav-link px-2 text-white">Contact</a></li>
            <li class="nav-item"><a href="despre.php" class="nav-link px-2 text-white">Despre</a></li>
        </ul>
        <p class="text-center text-white">&copy; <?php echo date("Y"); ?> Eli's Cinemax</p>
    </footer>
</div>
</body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</html>

```

Pagina autentificare.php

Fișierul autentificare.php permite utilizatorilor să se conecteze în sistem prin intermediul unei interfețe securizate. Acesta verifică email-ul și parola utilizatorului în baza de date, asigurând accesul pe baza rolului (admin, worker, user). În cazul unei autentificări reușite, utilizatorii sunt redirecționați către secțiunea corespunzătoare a aplicației, iar erorile sunt comunicate direct în interfață pentru o experiență ușor de utilizat.

```

<?php
session_start();
require "../Tables-MakeDB/makeDBConnection.php";
require "../html/navBar.php";

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

$message = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (!empty($_POST["email"]) && !empty($_POST["password"])) {
        $email = test_input($_POST["email"]);
        $password = test_input($_POST["password"]);

```

```

    $sql_users = "SELECT id, firstname, lastname, email, password, role FROM users WHERE
email = ?";
    $stmt_users = $conn->prepare($sql_users);
    $stmt_users->bind_param("s", $email);
    $stmt_users->execute();
    $result_users = $stmt_users->get_result();
    if ($result_users->num_rows > 0) {
        $row = $result_users->fetch_assoc();
        if (password_verify($password, $row["password"])) {
            $_SESSION["loggedin"] = true;
            $_SESSION["userid"] = $row["id"];
            $_SESSION["username"] = $row["firstname"] . " " . $row["lastname"];
            $_SESSION["role"] = $row["role"];
            switch ($row["role"]) {
                case "admin":
                    header("Location: ../fisiere_operator/admin.php");
                    break;
                default:
                    header("Location: acasa.php");
                    break;
            }
            exit();
        } else {
            $message = "Parola invalida. Va rugam sa incercati din nou.";
        }
    } else {
        $sql_workers = "SELECT id, firstname, lastname, email, password FROM workers
WHERE email = ?";
        $stmt_workers = $conn->prepare($sql_workers);
        $stmt_workers->bind_param("s", $email);
        $stmt_workers->execute();
        $result_workers = $stmt_workers->get_result();
        if ($result_workers->num_rows > 0) {
            $row = $result_workers->fetch_assoc();
            if (password_verify($password, $row["password"])) {
                $_SESSION["loggedin"] = true;
                $_SESSION["userid"] = $row["id"];
                $_SESSION["username"] = $row["firstname"] . " " . $row["lastname"];
                $_SESSION["role"] = "worker";
                header("Location: ../fisiere_operator/operator.php");
                exit();
            } else {
                $message = "Parola invalida. Va rugam sa incercati din nou.";
            }
        }
    }
}

```



```

        }
    } else {
        $message = "Utilizatorul nu a fost gasit. Verificati emailul si incercati din nou.";
    }
    $stmt_workers->close();
}
$stmt_users->close();
$conn->close();
} else {
    $message = "Va rugam sa furnizati atat email-ul cat si parola pentru autentificare.";
}
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <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">
    <link rel="stylesheet" type="text/css" href="../../css/style.css">
    <link rel="stylesheet" href="../../css/navBar.css">
    <style>
        .outer-container {background-color: #f2f2f2; margin: 30px auto; max-width: 600px; border-
radius: 8px; box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);}
        .form-container {max-width: 550px; margin: auto; padding-bottom: 20px; padding: 20px;
border-radius: 10px;}
        .btnn {color: black; background-color: #d9d9d9; border: 2px solid #c9c9c9; border-radius:
8px; padding: 10px 20px; text-align: center; text-decoration: none; font-size: 16px; margin: 10px
2px; transition: all 0.2s; cursor: pointer;}
        .btnn:hover {background-color: #cccccc; border-color: #bbbbbb;}
        .btnn:active {background-color: #bdbdbd; border-color: #acacac;}
        .btnn:focus {outline: none;}
        label {font-size: 18px; color: #212121;}
    </style>
    <title>Autentificare</title>
</head>
<body style="background-color: black;">
    <div class="outer-container">
        <div class="form-container" style="margin-top:70px;">

```

```

        <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post" class="w3-container w3-card-4 w3-light-grey w3-round-large" style="max-
width:500px; margin:auto">
        <h3 style="text-align: center;"><b>Autentificare</b></h3>
        <p><label for="email">Email</label><input type="email" class="form-control"
id="email" name="email" placeholder="Introdu adresa de email"></p>
        <p><label for="password">Parola</label><input type="password" class="form-
control" id="password" name="password" placeholder="Introdu parola"></p>
        <input type="submit" value="Autentificare" class="btn">
        <p><h6 style="color:#4d4d4d;"><b>Esti nou? <a href="inregistrare.php" style="color:
#3982D8;">Inregistreaza-te</b></a></h6></p>
        <?php if (!empty($message)) {echo "<p style='color: red;'>$message</p>";} ?>
    </form>
</div>
</div>
<div class="container" style="margin-top: 70px;">
    <footer class="py-3 my-4">
        <ul class="nav justify-content-center border-bottom pb-3 mb-3">
            <li class="nav-item"><a href="acasa.php" class="nav-link px-2 text-
white">Acasa</a></li>
            <li class="nav-item"><a href="maiMulteFilme2.php" class="nav-link px-2 text-
white">Filme</a></li>
            <li class="nav-item"><a href="contact.php" class="nav-link px-2 text-
white">Contact</a></li>
            <li class="nav-item"><a href="despre.php" class="nav-link px-2 text-
white">Despre</a></li>
        </ul>
        <p class="text-center text-white">&copy; <?php echo date("Y"); ?> Eli's Cinemax</p>
    </footer>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ"
crossorigin="anonymous"></script>
</body>
</html>

```

Pagina cautare.php

Fișierul cautare.php permite utilizatorilor să găsească rapid un film în baza de date utilizând numele său. Interfața acceptă o căutare flexibilă prin potriviri parțiale și oferă redirecționare automată către detaliile filmului găsit. Dacă nu există rezultate, utilizatorul este notificat în interfață. De asemenea, pagina include sugestii de filme afișate aleatoriu.

```

<?php
session_start();

```

```

?>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="../../css/style.css">
    <link rel="stylesheet" href="../../css/navBar.css">
    <link rel="icon" type="image/x-icon" href="../../images/logo.png">
    <link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet"                                     integrity="sha384-
T3c6CoLi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
    <link          rel="stylesheet"          href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <style>
        .Error {color: white; text-align: center; font-size: 35px; display: flex; justify-content: center;
align-items: center; height: 20vh;}
        .return {float: right; margin-right: 30%; font-size: 20px; margin-top: -6px;}
    </style>
    <title>Cautare</title>
</head>
<body style="background-color: black;">
    <?php
require "../Tables-MakeDB/makeDBConnection.php";
if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['movie_name'])) {
    $movieName = trim($_GET['movie_name']);
    if (!empty($movieName)) {
        $movieName = $conn->real_escape_string($movieName);
        $sql = "SELECT movie_id, title FROM movies WHERE title LIKE ?";
        $stmt = $conn->prepare($sql);
        $searchTerm = '%' . $movieName . '%';
        $stmt->bind_param("s", $searchTerm);
        $stmt->execute();
        $result = $stmt->get_result();
        if ($result->num_rows > 0) {
            $row = $result->fetch_assoc();
            header("Location: detalii.php?title=" . urlencode($row['title']));
            exit();
        } else {
            echo "<div class='Error'>Niciun film gasit cu acest nume.</div>";
        }
    } else {
        header("Location: acasa.php");
        exit();
    }
}

```

```

    }
}
?>
<div class="nav-container">
    <div class="wrapper">
        <form action="search.php" method="get">
            <input type="text" name="movie_name" placeholder="Cauta un film...">
            <button class="searchbtn" type="submit"><i class="fa fa-search"></i></button>
        </form>
    </div>
</div>
<a href="acasa.php" class="return">Inapoi</a>
<section style="margin-top:60px"></section>
<?php
if ($conn->connect_error) {
    die("Conexiunea a esuat: " . $conn->connect_error);
}
$sql = "SELECT movie_id, title, image_path FROM movies ORDER BY RAND() LIMIT 8";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    echo "<div class='grid-container container'>";
    while ($row = $result->fetch_assoc()) {
        echo "<div class='movie-card'>";
        echo "<form action='detalii.php' method='get'>";
        echo "<input type='hidden' name='title' value='" . htmlspecialchars($row["title"]) . "'>";
        echo "<button type='submit' class='movie_img'>";
        echo "<img src='" . htmlspecialchars($row["image_path"]) . "' alt='Imagine Film'>";
        echo "<div> " . htmlspecialchars($row["title"]) . "</div>";
        echo "</button>";
        echo "</form>";
        echo "</div>";
    }
    echo "</div>";
} else {
    echo "<div>Niciun rezultat gasit</div>";
}
?>
<div class="container">
    <footer class="py-3 my-4">
        <ul class="nav justify-content-center border-bottom pb-3 mb-3">
            <li class="nav-item"><a href="acasa.php" class="nav-link px-2 text-
white">Acasa</a></li>

```

```

        <li class="nav-item"><a href="maiMulteFilme2.php" class="nav-link px-2 text-
white">Filme</a></li>
        <li class="nav-item"><a href="contact.php" class="nav-link px-2 text-
white">Contact</a></li>
        <li class="nav-item"><a href="despre.php" class="nav-link px-2 text-
white">Despre</a></li>
    </ul>
    <p class="text-center text-white">&copy; <?php echo date("Y"); ?> Eli's Cinemax</p>
</footer>
</div>
</body>
</html>

```

Pagina contact.php

Fișierul contact.php oferă utilizatorilor informații esențiale despre cum să ia legătura cu Eli's Cinemax. Pagina include detalii precum numărul de telefon, adresa fizică și adresa de email. De asemenea, integrează o hartă interactivă bazată pe Leaflet care marchează locația.

```

<?php
session_start();
require "../Tables-MakeDB/makeDBConnection.php";
require "../html/navBar.php";
?>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact</title>
    <link rel="stylesheet" type="text/css" href="../../css/style.css">
    <link rel="stylesheet" href="../../css/navBar.css">
    <link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css" />
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet">
    <style>
        .fa-phone, .fa-map-marker, .fa-envelope {color: #FD8F1E; font-size: 40px;}
        .contact-container {display: grid; grid-template-columns: repeat(7, 1fr); margin: auto;
margin-top: 150px;}
        .box {background-color: white; margin: auto; text-align: center; border-radius: 10px; width:
280px; height: 180px;}
        .inner {margin-top: 30px;}
        .box1 {grid-column-start: 2;}
        .box2 {grid-column-start: 4;}
        .box3 {grid-column-start: 6;}
        #map {width: 700px; margin: auto; margin-top: 40px;}
        .leaflet-top, .leaflet-bottom {position: absolute; z-index: 500; pointer-events: none;}
        .box {margin-bottom: 20px;}
    </style>

```

```

    @media screen and (max-width: 900px) { .contact-container { grid-template-columns:
repeat(2, 1fr); gap: 10px; } #map { width: 80%; margin: auto; margin-top: 40px; } }
    @media screen and (max-width: 600px) { .contact-container { grid-template-columns: 1fr; }
#map { width: 400px; margin: auto; margin-top: 40px; } }
    </style>
</head>
<body style="background-color: black;">
    <div id="map" style="height: 400px;"></div>
    <div class="contact-container">
        <div class="box box1">
            <div class="inner">
                <i class="fa fa-phone" aria-hidden="true"></i>
                <h5 style="color: black;">Telefon</h5>
                <h6>0755 752 1954</h6>
            </div>
        </div>
        <div class="box box2">
            <div class="inner">
                <i class="fa fa-map-marker" aria-hidden="true"></i>
                <h5 style="color: black;">Adresa</h5>
                <h6>Aleea Pravat nr 1, Bucuresti, Romania</h6>
            </div>
        </div>
        <div class="box box3">
            <div class="inner">
                <i class="fa fa-envelope" aria-hidden="true"></i>
                <h5 style="color: black;">Mail</h5>
                <h6>Eli's Cinemax@info.ro</h6>
            </div>
        </div>
    </div>
    <div class="container" style="margin-top: 90px;">
        <footer class="py-3 my-4">
            <ul class="nav justify-content-center border-bottom pb-3 mb-3">
                <li class="nav-item"><a href="acasa.php" class="nav-link px-2 text-
white">Acasa</a></li>
                <li class="nav-item"><a href="maiMulteFilme2.php" class="nav-link px-2 text-
white">Filme</a></li>
                <li class="nav-item"><a href="contact.php" class="nav-link px-2 text-
white">Contact</a></li>
                <li class="nav-item"><a href="despre.php" class="nav-link px-2 text-
white">Despre</a></li>
            </ul>

```

```

        <p class="text-center text-white">&copy; <?php echo date("Y"); ?> Eli's Cinemax</p>
    </footer>
</div>
<script>
    document.addEventListener('DOMContentLoaded', function() {
        var map = L.map('map').setView([44.425294, 26.041090], 15);
        L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {attribution: '&copy;
OpenStreetMap contributors'}).addTo(map);
        var marker = L.marker([44.425294, 26.041090]).addTo(map);
        marker.bindPopup('Eli's Cinemax Bucuresti').openPopup();
    });
</script>
<script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"
crossorigin="anonymous"></script>
</body>
</html>

```

Pagina deconectare.php

Fișierul deconectare.php gestionează închiderea sesiunii utilizatorului. Prin această pagină, utilizatorii sunt deconectați din aplicație, toate variabilele de sesiune sunt eliminate, iar aceștia sunt redirecționați către pagina principală. Acest proces asigură securitatea și confidențialitatea datelor utilizatorilor.

```

<?php
session_start();
$_SESSION = array();
session_destroy();
header("Location: acasa.php");
exit;

```

Pagina despre.php

Fișierul despre.php prezintă utilizatorilor povestea Eli's Cinemax, de la începuturile sale până la angajamentul de a oferi experiențe cinematografice memorabile. Pagina subliniază tradiția, inovația și dedicarea echipei de a crea un mediu care celebrează arta cinematografică și conectează oamenii prin pasiunea pentru filme.

```

<?php
session_start();
require "../Tables-MakeDB/makeDBConnection.php";
require "../html/navBar.php";
?>
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Despre Eli's Cinemax</title>
<link rel="stylesheet" type="text/css" href="../../css/style.css">
<link rel="stylesheet" href="../../css/navBar.css">
<style>
    .outer {margin: auto; margin-top: 20px; max-width: 53%; padding: 30px; text-align: justify;
border-radius: 10px; border: 2px solid white; background-color: transparent; backdrop-filter:
blur(12px);}
    .content {max-width: 90%; margin: auto; color: white;}
    @media (max-width: 1200px) {.outer {max-width: 80%;} p {font-size: 16px;} h2 {font-size:
24px;} .title {font-size: 28px;}}
    .title {color: red; font-size: 30px; font-weight: 400;}
</style>
</head>
<body style="background-color: black;">
<div class="outer">
    <div class="content">
        <h2>Bun venit la<span class="titlu"> Eli's Cinemax</span></h2>
        <p>Intra intr-o lume unde magia filmelor a fermecat generatii intregi. Pasiunea pentru
cinematografie transforma fiecare vizionare intr-o experienta memorabila. Descopera povestea
noastra, traditia si dedicarea de a oferi momente de neuitat.</p>
        <h2>O mostenire cinematografica unica</h2>
        <p>Fa o incursiune in trecutul nostru si descopera cum a inceput povestea noastra. Inspirati
de dragostea pentru povesti, am devenit un reper al cinematografiei. Mostenirea noastra continua
sa modeleze experientele culturale si de divertisment locale.</p>
        <h2>In inima Romaniei</h2>
        <p>Avem radacinile in inima vibranta a Bucurestiului, oferind un loc de intalnire pentru
iubitorii de filme de toate varstele. Alege dintr-o selectie variata de filme, de la cele mai noi lansari
la capodoperele clasice, intr-un cadru care celebreaza arta cinematografica.</p>
        <h2>Experiente cinematografice deosebite</h2>
        <p>Aici, fiecare detaliu conteaza. Bucura-te de imagini spectaculoase si sunet realist care te
transpun direct in mijlocul actiunii. Cu tehnologia noastra de ultima generatie, fiecare vizita devine
o aventura de neuitat.</p>
        <h2>Mai mult decat filme</h2>
        <p>Suntem mai mult decat un cinematograf - suntem un loc de conectare si bucurie. Fie ca
te intalnesti cu prietenii sau petreci timp cu familia, aici creezi amintiri speciale. Venim in
intampinarea dorintelor tale cu o atmosfera primitoare si experiente care inspira.</p>
        <h2>Angajamentul nostru pentru tine</h2>
        <p>Ne mandrim cu o echipa dedicata care se asigura ca fiecare vizita este impecabila.
Indiferent daca esti un cinefil experimentat sau descoperi magia filmelor pentru prima data, aici
vei gasi intotdeauna un loc unde esti binevenit.</p>
        <h2>Multumim ca faci parte din povestea noastra</h2>

```


<p>Suntem recunoscatori ca ai ales sa faci parte din universul nostru. Continuum sa lucram pentru a aduce cele mai bune momente cinematografice in viata ta. Relaxeaza-te, bucura-te de spectacol si lasa magia filmelor sa-ti inspire imaginatia.</p>

</div>

</div>

<div class="container" style="margin-top: 90px;">

<footer class="py-3 my-4">

<ul class="nav justify-content-center border-bottom pb-3 mb-3">

<li class="nav-item">Acasa

<li class="nav-item">Filme

<li class="nav-item">Contact

<li class="nav-item">Despre

<p class="text-center text-white">© <?php echo date("Y"); ?> Eli's Cinemax</p>

</footer>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-

ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe" crossorigin="anonymous"></script>

</body>

</html>

Pagina detalii.php

Fișierul detalii.php oferă utilizatorilor informații detaliate despre un anumit film selectat. Pagina include descrierea filmului, prețul biletelor, sala în care rulează, precum și orele disponibile pentru vizionare. De asemenea, utilizatorii pot rezerva locuri pentru o anumită oră direct din această pagină. În plus, sunt afișate alte filme sugerate pentru a menține interacțiunea utilizatorului.

<?php

session_start();

require "../Tables-MakeDB/makeDBConnection.php";

require "../html/navBar.php";

?>

<head>

<title>Detalii Film</title>

<link rel="stylesheet" type="text/css" href="../css/style.css">

<link rel="stylesheet" href="../css/navBar.css">

<link rel="stylesheet" href="../css/imagesStyle.css">

<style>

.view_title {text-align: center; margin-bottom: 55px; margin-top: 35px; color: white;}

```

        .movie-details {margin: 2% 10%; padding: 20px; color: white; display: flex; gap: 20px;
background-color: #909090; border-radius: 10px;}
        .movie-image {flex-shrink: 0; width: 260px; height: 340px; margin-top: 2%;}
        .movie-info {display: flex; flex-direction: column; align-items: flex-start; gap: 10px; flex-
grow: 1; text-align: justify;}
        .movie-meta {margin-top: 10px; display: flex; flex-direction: column;}
        @media (max-width: 968px) {.movie-details {flex-direction: column; align-items: center;}
.movie-image {margin-top: 0;} .movie-info {text-align: center; align-items: center;}}
    </style>
</head>
<?php
$movieId = null;
$movieTitle = null;
if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['title'])) {
    $movieTitle = $_GET['title'];
    $sql = "SELECT movies.*, rooms.RoomName, GROUP_CONCAT(DISTINCT
showtimes.showtime ORDER BY showtimes.showtime SEPARATOR ', ') AS ShowTimes
FROM movies JOIN rooms ON movies.movie_id = rooms.movie_id JOIN showtimes ON
movies.movie_id = showtimes.movie_id WHERE movies.title = ? GROUP BY
movies.movie_id";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $movieTitle);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $showtimes = explode(', ', $row["ShowTimes"]);
            echo "<div class='movie-details'>";
            echo "<img src='" . htmlspecialchars($row["image_path"]) . "' alt='Imagine Film'
class='movie-image'>";
            echo "<div class='movie-info'>";
            echo "<h1>" . htmlspecialchars($row["title"]) . "</h1>";
            echo "<div class='movie-description'>Descriere: " . htmlspecialchars($row["description"])
. "</div>";
            echo "<div class='movie-meta'>";
            echo "<p>Pret: " . htmlspecialchars($row["movie_price"]) . "</p>";
            echo "<p>Sala: " . htmlspecialchars($row["RoomName"]) . "</p>";
            echo "<p>Disponibilitate:</p>";
            foreach ($showtimes as $showtime) {
                echo "<form action='asezariScaune.php' method='get' class='showtime-form'>";
                echo "<input type='hidden' name='title' value='" . htmlspecialchars($row['title']) . "'>";
                echo "<input type='hidden' name='showtime' value='" . htmlspecialchars($showtime) .
"">";

```

```

        echo "<input type='hidden' name='room' value='" .
htmlspecialchars($row['RoomName']) . "'>";
        echo "<input type='hidden' name='image' value='" .
htmlspecialchars($row['image_path']) . "'>";
        echo "<button type='submit' class='showtime-button'>" . htmlspecialchars($showtime)
. "</button>";
        echo "</form>";
    }
    echo "</div>";
    echo "</div>";
    echo "<div style='clear:both;'></div>";
    echo "</div>";
}
} else {
    echo "<div>Niciun film gasit</div>";
}
} else {
    echo "<div>Parametrul 'title' lipseste sau este invalid.</div>";
}
if ($conn->connect_error) {
    die("Conexiunea a esuat: " . $conn->connect_error);
}
echo "<h2 class='view_title'>VEZI SI</h2>";
$sql = "SELECT movie_id, title, image_path FROM movies ORDER BY RAND() LIMIT 8;";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    echo "<div class='grid-container container'>";
    while ($row = $result->fetch_assoc()) {
        echo "<div class='movie-card'>";
        echo "<form action='detalii.php' method='get'>";
        echo "<input type='hidden' name='title' value='" . htmlspecialchars($row["title"]) . "'>";
        echo "<div class='movie-img-container'>";
        echo "<button type='submit' class='movie_img'>";
        echo "<img src='" . htmlspecialchars($row["image_path"]) . "' alt='Imagine Film'>";
        echo "</button>";
        echo "<div>" . htmlspecialchars($row["title"]) . "</div>";
        echo "</div>";
        echo "</form>";
        echo "</div>";
    }
    echo "</div>";
} else {
    echo "<div>Nu s-au gasit rezultate</div>";
}

```

```

}
?>
<body style="background-color: black;">
    <div class="container">
        <footer class="py-3 my-4">
            <ul class="nav justify-content-center border-bottom pb-3 mb-3">
                <li class="nav-item"><a href="acasa.php" class="nav-link px-2 text-white">Acasa</a></li>
                <li class="nav-item"><a href="maiMulteFilme2.php" class="nav-link px-2 text-white">Filme</a></li>
                <li class="nav-item"><a href="contact.php" class="nav-link px-2 text-white">Contact</a></li>
                <li class="nav-item"><a href="despre.php" class="nav-link px-2 text-white">Despre</a></li>
            </ul>
            <p class="text-center text-white">&copy; <?php echo date("Y"); ?> Eli's Cinemax</p>
        </footer>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>

```

Pe lângă paginile principale ale proiectului, sistemul include și alte pagini care contribuie la funcționalitatea completă a platformei. Fiecare dintre acestea joacă un rol important în administrarea utilizatorilor, filmelor, rezervărilor și plăților. Mai jos enumerăm câteva dintre aceste pagini, împreună cu o scurtă descriere și un exemplu de cod relevant pentru fiecare.

Pagina efectuarePlata.php

```

session_start();
require "../Tables-MakeDB/makeDBConnection.php";
$userId = $_SESSION["userid"];
$total = $_POST["total"];
$sql = "INSERT INTO payments (user_id, amount) VALUES (?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("id", $userId, $total);
$stmt->execute();
echo "Plata de " . htmlspecialchars($total) . " RON a fost efectuată cu succes!";

```

Pagina inregistrare.php

```

session_start();
require "../Tables-MakeDB/makeDBConnection.php";
$email = htmlspecialchars($_POST["email"]);
$password = password_hash($_POST["password"], PASSWORD_BCRYPT);

```

```

$sql = "INSERT INTO users (firstname, lastname, email, password) VALUES (?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ssss", $_POST["firstname"], $_POST["lastname"], $email, $password);
$stmt->execute();
echo "Înregistrare completă! Bine ai venit, " . htmlspecialchars($_POST["firstname"]) . "!";

```

Pagina maiMulteFilme2.php

```

session_start();
require "../Tables-MakeDB/makeDBConnection.php";
$offset = isset($_GET["page"]) ? ($_GET["page"] - 1) * 10 : 0;
$sql = "SELECT * FROM movies LIMIT 10 OFFSET ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $offset);
$stmt->execute();
$result = $stmt->get_result();
while ($row = $result->fetch_assoc()) {
    echo "<div class='movie-card'>" . htmlspecialchars($row["title"]) . "</div>";
}

```

Pagina plata.php

```

session_start();
require "../Tables-MakeDB/makeDBConnection.php";
$bookingId = $_GET["booking_id"];
$sql = "SELECT * FROM bookings WHERE booking_id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $bookingId);
$stmt->execute();
$result = $stmt->get_result();
$row = $result->fetch_assoc();
echo "Total: " . htmlspecialchars($row["total_price"]) . " RON";

```

Pagina profil.php

```

session_start();
require "../Tables-MakeDB/makeDBConnection.php";
$userId = $_SESSION["userid"];
$sql = "SELECT firstname, lastname, email FROM users WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $userId);
$stmt->execute();
$result = $stmt->get_result();
$row = $result->fetch_assoc();
echo "Nume: " . htmlspecialchars($row["firstname"]) . " " . htmlspecialchars($row["lastname"]);

```

Pagina admin.php

```
session_start();
require "../Tables-MakeDB/makeDBConnection.php";
$sql = "SELECT * FROM users";
$result = $conn->query($sql);
while ($row = $result->fetch_assoc()) {
    echo "<div>Utilizator: " . htmlspecialchars($row["email"]) . " - Rol: " .
    htmlspecialchars($row["role"]) . "</div>";
}
```

Pagina rezervareBilet.php

```
session_start();
require "../Tables-MakeDB/makeDBConnection.php";
$userId = $_SESSION["userid"];
$movieId = $_POST["movie_id"];
$seatId = $_POST["seat_id"];
$sql = "INSERT INTO bookings (user_id, movie_id, seat_id) VALUES (?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("iii", $userId, $movieId, $seatId);
$stmt->execute();
echo "Rezervare realizată cu succes!";
```

BIBLIOGRAFIE

- [1] David Flanagan, "JavaScript: The Definitive Guide", O'Reilly Media, 2020.
- [2] Elmasri, Ramez, and Shamkant B. Navathe. "Fundamentals of Database Systems", Pearson, 2016.
- [3] Martin Fowler, "Patterns of Enterprise Application Architecture", Addison-Wesley Professional, 2002.
- [4] Note de curs "Baze de date", Universitatea Tehnica de Constructii Bucuresti, 2016
- [5] Ethan Marcotte, "Responsive Web Design", A Book Apart, 2011.
- [6] <https://www.safaribooksonline.com/library/view/rt-essentials/0596006683/ch01.html>
- [7] https://ro.wikipedia.org/wiki/HyperText_Markup_Language
- [8] https://ro.wikipedia.org/wiki/Cascading_Style_Sheets
- [9] <https://ro.wikipedia.org/wiki/JavaScript>
- [10] <https://ro.wikipedia.org/wiki/SQL>
- [11] <https://ro.wikipedia.org/wiki/Model-view-controller>
- [12] [https://ro.wikipedia.org/wiki/Ajax_\(programare\)](https://ro.wikipedia.org/wiki/Ajax_(programare))
- [13] [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
- [14] https://en.wikipedia.org/wiki/ASP.NET_Razor
- [15] https://ro.wikipedia.org/wiki/Microsoft_Visual_Studio
- [16] https://en.wikipedia.org/wiki/Microsoft_SQL_Server
- [17] Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke, No Starch Press, 2020.
- [18] Robert C. Martin, "Clean Code: A Handbook of Agile Software Craftsmanship", Prentice Hall, 2008.
- [19] Jeremy McPeak, "Beginning JavaScript, 5th Edition", Wrox, 2015
- [20] Eric Evans, "Domain-Driven Design: Tackling Complexity in the Heart of Software", Addison-Wesley, 2003.
- [21] "Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages" by Elisabeth Robson, Eric Freeman, O'Reilly Media, 2012.
- [22] "SQL in 10 Minutes, Sams Teach Yourself" by Ben Forta, Sams Publishing, 2012.