



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA
CENTRUL UNIVERSITAR NORD DIN BAIA MARE
FACULTATEA DE ȘTIINȚE

LUCRARE DE LICENȚĂ

Aplicatie pentru gestiunea pacientilor

COORDONATOR ȘTIINȚIFIC:

Grad didactic, Prenume Nume

ABSORVENT:

Goloman Adrian Tanasie

Sesiunea Iulie 2026

**Declarație pe proprie răspundere privind
autenticitatea lucrării de licență/diplomă/disertație**

Subsemnatul

, legitimat cu _____ seria _____ nr. _____ (conform copiei anexate prezentei declarații, copie semnată și certificată „conform cu originalul”), autorul lucrării

elaborată în vederea susținerii examenului de finalizare a studiilor de _____

la Facultatea _____

Specializarea _____

din cadrul Universității Tehnice din Cluj-Napoca, sesiunea _____ a anului universitar _____, declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării, și în bibliografie.

Declar, că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că aceasta lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență/diplomă/disertație.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, anularea examenului de licență/diplomă/disertație.

Declar și sunt de acord ca, pe tot parcursul vieții, în cazul în care este necesar și se va dori verificarea autenticității lucrării mele să fiu identificat și verificat în baza datelor declarate de mine și conform copiei documentului de identitate menționat mai sus.

Nume, prenume

Data

Semnătura

Rezumat (RO)

[Imagine care definește lucrarea de licență/disertație realizată.]

Rezumatul lucrării în limba română, 150–250 de cuvinte (nu mai mult de pagina curentă).

Cuvinte-cheie: termen1; termen2; termen3; termen4; termen5.

Abstract (EN)

[Image that defines the completed bachelor's/dissertation work.]

Short abstract in English, 150–250 words.

Keywords: term1; term2; term3; term4; term5.

Cuprins

1 Introducere	10
1.1 Motivarea alegerii temei	10
1.2 Gradul de noutate al temei	10
1.3 Obiectivele lucrării	10
1.3.1 Obiectiv general	10
1.3.2 Obiective specifice	11
1.4 Relevanța rezultatelor	11
1.5 Metodologia utilizată	11
1.5.1 Metode de cercetare	11
1.5.2 Metode de dezvoltare	11
1.6 Descrierea soluției propuse	11
1.7 Structura lucrării	12
2 Analiza situației existente	13
2.0.1 Analiza comparativă din tabel	15
3 Tehnologii utilizate	16
3.1 Prezentare generală	16
3.2 Backend - PHP 8+	16
3.2.1 Caracteristicile PHP 8 utilizate	16
3.2.2 Securitate în PHP	16
3.3 Frontend - HTML5, CSS3 și JavaScript	16
3.3.1 HTML5	17
3.3.2 CSS3 și Responsive Design	17
3.3.3 JavaScript Vanilla	17
3.4 Baza de date - MariaDB	17
3.4.1 Modelul de date	17
3.4.2 Relații și integritate	17
3.4.3 Normalizare	18

3.5	Editor de text bogat - TinyMCE	18
3.5.1	Funcționalități ale TinyMCE	18
3.5.2	Integrarea cu LaTeX	18
3.6	Suport LaTeX - MathJax	18
3.6.1	Utilizare în aplicație	18
3.6.2	Avantaje	18
3.7	Servicii de email - PHPMailer	19
3.7.1	Funcționalități	19
3.7.2	Implementare în aplicație	19
3.8	Alte biblioteci și instrumente	19
3.8.1	Composer	19
3.8.2	Git și control de versiuni	19
3.9	Infrastructură de hosting	19
3.9.1	Cerințe minime	19
3.10	Comparăția cu alte stack-uri tehnologice	20
4	Specificații și cerințe	21
4.1	Introducere despre aplicație	21
4.2	Actori ai sistemului	21
4.2.1	Vizitator (Utilizator neautentificat)	21
4.2.2	Pacient (Utilizator autentificat cu rol de pacient)	21
4.2.3	Doctor (Utilizator autentificat cu rol de doctor)	22
4.2.4	Administrator (Utilizator cu rol de admin)	22
4.3	Cerințe non-funcționale	22
4.3.1	Performanță	22
4.3.2	Securitate	22
4.3.3	Utilizabilitate	22
4.3.4	Scalabilitate	23
4.3.5	Compatibilitate	23
4.3.6	Mantenabilitate	23
4.4	Fluxuri de lucru principale	23
4.4.1	Fluxul de înregistrare și autentificare	23

4.4.2	Fluxul de publicare articol	23
4.4.3	Fluxul de interacțiune cu articol	23
4.4.4	Fluxul de administrare utilizatori	24
4.5	Concluzii asupra specificațiilor și cerințelor	24
5	Implementare	25
5.1	Prezentare generală	25
5.2	Arhitectura sistemului	25
5.2.1	Structura directoarelor	25
5.2.2	Componenta de prezentare	25
5.2.3	Componenta de logică	26
5.2.4	Componenta de date	26
5.3	Diagrame și ilustrații	26
5.3.1	Diagrama Use Case	27
5.3.2	Diagrama de activitate	27
5.3.3	Workflow Diagram	28
5.4	Implementarea funcționalităților principale	28
5.4.1	Sistemul de autentificare și înregistrare	28
5.4.2	Sistemul de programări medicale	29
5.4.3	Sectiunea educațională	29
5.4.4	Panoul de administrare	29
5.4.5	Panoul doctorului	30
5.5	Detalii tehnice de implementare	30
5.5.1	Conexiunea la baza de date	30
5.5.2	Sistemul de încărcare imagini	30
5.5.3	Sistemul de email	30
5.5.4	Sanitizarea datelor	31
5.6	Interfața utilizator	31
5.6.1	Componentele vizuale	31
5.6.2	Feedback vizual	31
5.6.3	Organizarea sarcinilor	31
5.7	Screenshot-uri reprezentative	32

6 Testare și Validare	37
6.1 Introducere	37
6.2 Testare	37
6.2.1 Abordarea de testare	37
6.2.2 Tipuri de teste efectuate	37
6.2.3 Cazuri de testare	38
6.3 Concluzii privind validarea aplicației	44
7 Concluzii și direcții viitoare	45
7.1 Sinteza contribuților	45
7.2 Analiza rezultatelor obținute	45
7.3 Limitări identificate	46
7.4 Observații personale	46
7.5 Relevanța și impactul soluției	47
7.6 Direcții viitoare de dezvoltare	47
7.6.1 Sistem de notificări	47
7.6.2 Mesagerie internă	48
7.6.3 Telemedicină	48
7.6.4 Aplicație mobilă	48
7.6.5 Funcționalități de accesibilitate	48
7.6.6 Multilingvism	48
7.7 Concluzii finale	48
Bibliografie	51
Anexe	51
Anexa A - Listă de figuri	51
AAnexa B - Listă de tabele	51
Anexa C - Listă de abrevieri (optional)	52
Anexa D — Exemplificare cod/flux/etc. (exemplu)	52

1 Introducere

1.1 Motivarea alegerii temei

Aplicațiile pentru gestionarea pacienților în domeniul sanitar joacă un rol important în eficiența și gestionarea serviciilor de sănătate. În perioada actuală, digitalizarea a devenit mai mult o obligație decât un plus.

Digitalizarea serviciilor medicale rezolvă probleme precum gestionarea mult mai usoară a datelor medicale, deoarece informațiile pacienților sunt mult mai bine structurate și mai ușor de accesat și modificat. Comunicarea dificilă între pacienți și doctori devine mult mai accesibilă și rapidă printr-o aplicație în mediul online. De asemenea, sistemele actuale de programare consumă extrem de mult timp și generează o frustrare la nivelul pacienților, problemă ce poate fi remediată prin soluții digitale.

Soluționarea acestor probleme are un **impact direct asupra calității serviciilor medicale** și a **satisfacției utilizatorilor**, făcând tema extrem de relevantă pentru domeniul medical.

1.2 Gradul de noutate al temei

Lucrarea de licență propune o **platformă integrată** care combină mai multe funcționalități necesare pentru un mediu eficient. Aplicația oferă funcționalități diferite pentru fiecare tip de utilizator, având restricții în funcție de rolul fiecărui. Secțiunea educațională permite medicilor să își publice cercetările academice prin care să își expună descoperirile unui număr cât mai mare de utilizatori, aceștia din urmă putând reacționa prin comentarii și evaluări de tip like/dislike. Secțiunea de profil permite pacienților să își adauge poze de profil, date personale și medicale, dar și să își vizualizeze investigațiile și rezultatele anterioare. Tabela de bord a administratorului oferă control total asupra utilizatorilor și articolelor, permitându-i acestuia să editeze și să steargă conținut. Funcționalitatea de programare transformă această acțiune într-una simplă, fiind la un singur click distanță.

1.3 Obiectivele lucrării

1.3.1 Obiectiv general

Creeare unui platforme web de gestionare a cadrului sanitar într-un mediu online cu funcționalitate care să reducă timpul de așteptare pentru programari și vizite medicale, să imbunătățească comunicarea dintre pacient și doctor și să ajute la dezvoltarea cunoștinței populației prin intermediul funcționalității educationale.

1.3.2 Obiective specifice

Implementarea unui sistem de autentificare securizat cu verificare email și roluri diferențiate constituie prima țintă. Dezvoltarea modulului de creare a unui profil de utilizator cu suport pentru adăugare de date personale și biometrice este al doilea obiectiv. Crearea platformei educaționale cu posibilitate de publicare, editare și stergere de articole urmează ca al treilea scop. Implementarea sistemului de feedback pentru articole este esențială. Dezvoltarea tabelei de control a adminului cu funcții CRUD complete pe tabelele de utilizatori și articole asigură controlul total al sistemului. Integrarea editorului rich text (TinyMCE) cu suport LaTeX pentru articole științifice permite publicarea de conținut avansat. Proiectarea unei interfețe user-friendly și responsive pentru desktop și mobile garantează accesibilitate universală. În sfârșit, testarea și validarea funcționalităților critice ale platformei asigură calitatea și fiabilitatea aplicației.

1.4 Relevanța rezultatelor

Rezultatele lucrării aduc îmbunătățiri considerabile în mai multe arii. Pe plan practic, oferă o soluție ușoară și scalabilă care poate fi implementată în spitale pentru îmbunătățirea fluxului acestora. Pe plan educațional, ajută accesul la informații și noutăți medicale ale profesionistilor, contribuind la educația și combaterea dezinformării în spațiul medical a populației.

1.5 Metodologia utilizată

1.5.1 Metode de cercetare

Analiza documentară reprezintă studiul literaturii de specialitate privind sisteme de management medical. Analiza nevoilor identifică cerințele funcționale și non-funcționale ale utilizatorilor, asigurând o înțelegere profundă a scopului sistemului.

1.5.2 Metode de dezvoltare

Development-ul iterativ realizează implementarea în cicluri multiple cu feedback și ajustări continue. Database design-ul implică modelarea relațională a datelor cu normalizare și indexare pentru performanță. Frontend-ul utilizează HTML5, CSS3 și JavaScript vanilla pentru interactivitate fără dependență de framework-uri. Testing-ul implică testare manuală a funcționalităților, validare formulare și scenarii edge-case.

1.6 Descrierea soluției propuse

Soluția constă într-o **platformă web full-stack** construită cu tehnologiile amintite mai sus. Componentele principale ale sistemului sunt următoarele: Modulul de autentificare oferă Register, Login, și verificare prin email. Gestiona profilului permite editarea datelor personale, adăugarea unei poze de profil și vizualizare istoricului. Platforma EDU conține articole cu Like/Dislike și comentarii. Dashboard-ul doctorului permite publicare articole, vizualizare proprii postări și statistici. Dashboard-ul adminului oferă control asupra utilizatorilor și asupra artico-

lelor. Pagina principală prezintă articole populare și navigare intuitivă.

Arhitectura este **modulară și scalabilă**, permitând ușor extensiile cum ar fi sistem de notificări și chat.

1.7 Structura lucrării

Lucrarea de licență este structurată în șase capitole principale. Capitolul 1, Introducere, prezintă tema, motivația, obiectivele și structura lucrării, fiind capitolul curent. Capitolul 2, Analiza de caz și cerințe, oferă descriere detaliată a nevoilor utilizatorilor, actor-uri sistem, use case-uri și cerințe funcționale și non-funcționale. Capitolul 3, Arhitectură și design, cuprinde diagramele UML (Class, Sequence, Use Case), schema bazei de date și design pattern-uri utilizate. Capitolul 4, Implementare, detaliază componentele implementate, stack tehnologic și explicația codului semnificativ. Capitolul 5, Testare și validare, prezintă scenarii de test, rezultate, bugs identificate și resolved, precum și métrici de calitate. Capitolul 6, Concluzii și perspective viitoare, face sumarul contribuțiilor, identifică limitări și propune direcții de dezvoltare viitoare.

2 Analiza situației existente

Pentru a oferi o vizuire clară asupra pozițiilor relative ale soluțiilor existente, prezentăm în continuare un tabel comparativ care analizează caracteristicile principale ale aplicațiilor similare și poziționează soluția propusă în contextul acestora.

Aplicație	Descriere	Avantaje (Pro)	Limitări (Contra)	Cum poate fi aplicația mea mai bună
Regina Maria	Platformă medicală privată care permite programări online, consultații și acces la istoricul medical.	Interfață modernă și intuitivă Acces rapid la servicii și rezultate	Limitată pentru utilizare în sistemul public Fără funcționalități educationale Contact în caz de urgență nu este automat	Adaugarea funcționalităților educationale în care medicii pot publica articole și ghiduri de preventie pentru pacienți.
Hipocrate	Sistem informatic folosit în spitale publice pentru gestiunea pacienților și activităților medicale interne.	Integrează mai multe module interne Folosit pe scară largă în spitalele din stat	Interfață învechită și greu de utilizat Poate fi dificil de personalizat sau extins Limitări în comunicarea cu pacienții	Aplicația mea va avea interfață simplă și modulară pentru extindere ușoară.
Medicover eHealth	Aplicație pentru pacienți din rețeaua Medicover, permite programări și vizualizarea analizelor.	Aplicație orientată către pacient Permite comunicare și vizualizare rezultate online	Nu este orientată pentru management intern al spitalului Funcționalități educationale limitate sau inexistente Dependență de spitalul/rețeaua medicală	Aplicația mea va limba accesul pacientului cu gestionarea internă a datelor medicale și va integra module educationale completepe.
Platforma Integrată de Management Sanitar (Propusă)	Platformă web unificată care integrează management sanitar, educație și comunicare medic-pacient.	Interfață modernă și responsivă Integrează complet management și educație Funcționalități sociale (Like/Dislike, comentarii) ¹⁴	Fără integrări cu echipamente medicale specializate (versiune inițială) Conformitate HIPAA necesită dezvoltare Comunitate mai	N/A - Aceasta este soluția propusă

2.0.1 Analiza comparativă din tabel

Din analiza tabelului comparativ, se pot extrage mai multe concluzii importante. Platformele existente sunt în general specialized pe anumite aspecte: Regina Maria și Medicover sunt orientate către pacient și servicii medicale private, Hipocrate este concentrat pe gestionarea internă a spitalelor, iar sistemele internaționale sunt fie prea costisitoare, fie prea generice.

Niciuna dintre soluțiile existente nu combină în mod eficient aspectele de management sanitar cu funcționalități educaționale în cadrul unei platforme single. Aceasta creează o oportunitate clara pentru soluția propusă, care va oferi o abordare integrată și modernă.

Avantajele competitive ale soluției propuse sunt evidente: combinația dintre funcționalități medicale și educaționale, interfață modernă și responsivă, modularitatea care permite extensii ușoare, și costul redus datorită utilizării tehnologiilor open-source. În plus, sistemul permite interacțiunea și feedback între utilizatori, ceea ce nu este bine suportat de sistemele tradiționale.

3 Tehnologii utilizate

3.1 Prezentare generală

Arhitectura sistemului se bazează pe o selecție de tehnologii open-source corelate pentru a asigura scalabilitatea și mențenanța pe termen lung. Alegerea acestui stack tehnologic a fost determinată de necesitatea de a procesa volume mari de date medicale și de a asigura un timp de răspuns optim în interfața utilizatorului.

3.2 Backend - PHP 8+

PHP este limbajul de programare ales pentru componenta server-side a aplicației. Versiunea PHP 8 și versiunile ulterioare oferă îmbunătățiri semnificative în ceea ce privește performanța, tipizarea și siguranța codului.

3.2.1 Caracteristicile PHP 8 utilizate

Tipizarea cu declarare strictă (strict types) este utilizată pentru a asigura siguranța tipurilor și pentru a preveni erorile silentioase. Operatorul nullsafe (?->) simplifică manipularea valorilor null și reduce necesitatea verificărilor repetitive. Named arguments permit o citire mai ușoară a codului și facilitează menținerea lui pe termen lung. Union types și mixed type oferă flexibilitate în declararea tipurilor parametrilor și valorilor de întoarcere.

3.2.2 Securitate în PHP

Prepared statements sunt utilizate în toate cererile către baza de date pentru a preveni injecțiile SQL. Funcția htmlspecialchars() este aplicată la toate datele care sunt afișate în HTML pentru a preveni atacurile XSS. Validarea și sanitizarea inputului utilizatorului sunt efectuate pe server-side, iar mesajele de eroare sunt generice pentru a nu expune detalii interne ale sistemului. Session management este implementat cu parametri siguri, inclusiv regenerarea ID-ului sesiunii după autentificare.

3.3 Frontend - HTML5, CSS3 și JavaScript

Frontend-ul aplicației este construit utilizând tehnologiile standard web fără a depinde de framework-uri grele precum React, Vue sau Angular. Aceasta oferă o încărcare rapidă, o footprint mic și o dependență minimă de biblioteci externe.

3.3.1 HTML5

HTML5 este utilizat pentru a structura semantic conținutul. Elementele semantice cum ar fi `<header>`, `<nav>`, `<main>`, `<article>`, `<section>` și `<footer>` sunt folosite corect pentru a oferi o structură clară și accesibilă. Formulare HTML5 includ validări nativ pe client-side, cu atribută cum ar fi `required`, `type="email"`, `min`, `max` și altele.

3.3.2 CSS3 și Responsive Design

CSS3 este utilizat pentru stilizare și layout, cu accent pe design responsive care funcționează pe ecrane de diferite dimensiuni. Flexbox și Grid sunt utilizate pentru layout-uri complexe și adaptabile. Media queries sunt implementate pentru a ajusta stilurile în funcție de lățimea ecranului, asigurând că aplicația funcționează bine pe desktop, tabletă și telefon. Custom properties (CSS variables) sunt utilizate pentru a mări tema aplicației și pentru a permite schimbări rapide de stil pe întreg site-ul.

3.3.3 JavaScript Vanilla

JavaScript vanilla este utilizat pentru interactivitate. Event listeners sunt utilizate pentru a răspunde la acțiuni ale utilizatorului cum ar fi clickuri, submit de formulare și schimbări de input. Fetch API este utilizat pentru a efectua cereri asincrone către server fără a necesita o bibliotecă externă. DOM manipulation este efectuată direct pentru a afișa sau ascunde elemente, a valida formulare și a afișa mesaje utilizatorului.

3.4 Baza de date - MariaDB

MariaDB este sistemul de gestiune a bazelor de date relaționale ales pentru aplicație. Este un fork open-source al MySQL, cu performanță îmbunătățită și compatibilitate completă cu MySQL.

3.4.1 Modelul de date

Baza de date conține cinci tabele principale: User, UserDetails, EduPosts, Comments și Likes. Tabelul User stochează informații de autentificare și rol. Tabelul UserDetails stochează detaliile personale ale utilizatorilor, inclusiv imagine de profil. Tabelul EduPosts stochează articolele educative publicate de medici. Tabelul Comments stochează comentariile și răspunsurile la articole. Tabelul Likes stochează informații despre aprecieri și dezaprecieri ale articolelor.

3.4.2 Relații și integritate

Foreign keys sunt utilizate pentru a asigura integritatea referențială între tabele. Când un utilizator este șters, toate articolele și comentariile sale sunt șterse automat datorită relației ON DELETE CASCADE. În deces sunt creați pe coloane frecvent căutate cum ar fi email, $user_i \rightarrow post_i \rightarrow comments_i$ pentru a obține performanță în recuperarea informațiilor.

3.4.3 Normalizare

Baza de date este normalizată până la a treia formă normală (3NF) pentru a evita redundanța datelor și pentru a asigura integritatea. De exemplu, detaliile utilizatorului sunt stocate într-o tabelă separată pentru a evita duplicarea datelor și pentru a ușura actualizarea informațiilor.

3.5 Editor de text bogat - TinyMCE

TinyMCE este un editor WYSIWYG (What You See Is What You Get) integrat în formularul de creare a articolelor educative. Permite medicilor să creeze conținut formatat cu ușurință, fără a fi nevoie să cunoască HTML.

3.5.1 Funcționalități ale TinyMCE

Editorul permite formatarea textului (bold, italic, underline), crearea listelor (ordonate și neordonate), inserarea de linkuri și imagini. Funcția de undo/redo permite revenirea la versiuni anterioare ale conținutului. Vizualizarea codului sursă permite utilizatorilor avansați să editeze HTML direct. Counter de cuvinte oferă feedback utilizatorului cu privire la lungimea conținutului.

3.5.2 Integrarea cu LaTeX

Deși TinyMCE în sine nu suportă LaTeX nativ, integrarea cu MathJax permite utilizatorilor să scrie ecuații matematice în format LaTeX, iar MathJax le randează în browser-ul utilizatorului. Aceasta permite medicilor și cercetătorilor să publice articole științifice cu ecuații și formule complexe.

3.6 Suport LaTeX - MathJax

MathJax este o bibliotecă JavaScript care randează notație matematică din LaTeX, MathML și AsciiMath în pagini web. Este utilizată pentru a afișa ecuații și formule în articolele educative.

3.6.1 Utilizare în aplicație

Utilizatorii pot introduce ecuații LaTeX în format inline (între \$ și \$) sau display (între \$\$ și \$\$). MathJax le detectează și le randează ca imagini matematice frumoase. De exemplu, o ecuație precum $E = mc^2$ va fi să fie corectă într-un articol.

3.6.2 Avantaje

MathJax este gratuit și open-source. Funcționează în orice browser modern fără a necesita plug-in-uri. Suportă o gamă largă de notații matematice și științifice. Permite utilizatorilor să copieze ecuații și să le utilizeze în alte documente.

3.7 Servicii de email - PHPMailer

PHPMailer este o bibliotecă PHP care facilitează trimitera de email-uri cu ușurință. Este utilizată pentru a trimite email-uri de verificare a contului și posibil alte notificări.

3.7.1 Funcționalități

PHPMailer suportă trimitera de email-uri prin SMTP, ceea ce oferă mai mult control și fiabilitate decât funcția nativă `mail()` a PHP. Suportă autentificare SMTP cu criptare TLS sau SSL pentru securitate. Permite adăugarea de attachments și imagini inline la email-uri.

3.7.2 Implementare în aplicație

La înregistrarea unui utilizator, un email de verificare este trimis cu un link care conține un token unic. Utilizatorul trebuie să acceseze link-ul pentru a verifica adresa de email. Token-ul expiră după un anumit timp pentru a preveni utilizarea lor pe termen lung.

3.8 Alte biblioteci și instrumente

3.8.1 Composer

Desi nu este utilizat în versiunea curentă, Composer ar putea fi introdus pentru gestionarea dependențelor în versiuni viitoare. Ar permite gestionarea facilă a bibliotecilor externe și a versiunilor lor.

3.8.2 Git și control de versiuni

Codul sursă este gestionat cu Git, permitând urmărirea modificărilor, colaborarea între dezvoltatori și revenirea la versiuni anterioare dacă este necesar.

3.9 Infrastructură de hosting

Aplicația este construită pentru a funcționa pe orice server web care are suport pentru PHP 8 și MySQL/MariaDB. Nu necesită o infrastructură specială sau un cloud computing expensive. Poate fi hostată pe un shared hosting standard, pe un VPS ieftin sau pe propriul server.

3.9.1 Cerințe minime

PHP 8.0 sau versiune superioară cu extensiile mysqli și mbstring activate. MySQL 5.7 / MariaDB 10.2 sau versiuni superioare. Minim 100 MB spațiu de stocare pentru aplicație și baza de date (mai mult pentru upload-uri). Lățime de bandă rezonabilă pentru servirea conținutului.

3.10 Comparația cu alte stack-uri tehnologice

Pentru context, este util să comparăm alegerea noastră cu alte abordări populare. Stackurile moderne cum ar fi MEAN (MongoDB, Express, Angular, Node.js), MERN (MongoDB, Express, React, Node.js) sau serverless architectures sunt mai complexe și mai costisitoare să fie desfășurate. Necesită mai multă experiență pentru a fi menținute.

Stack-ul Laravel cu Blade template engine ar fi o alegere mai modernă, dar ar introduce o complexitate suplimentară și o dependență de un framework monolic. Rails sau Django ar fi potrivite pentru aplicații mai mari, dar sunt excesive pentru un sistem de management medical de talie medie.

Alegerea noastră pentru PHP vanilla cu MySQL oferă un echilibru optim: simplitate, performanță, cost redus și ușurință de mențenanță. Permite dezvoltatorilor să înțeleagă exact ce se întâmplă în aplicație.

4 Specificații și cerințe

4.1 Introducere despre aplicație

Platforma Integrată de Management Sanitar cu Funcționalități Educaționale este o aplicație web complexă, construită pentru a facilita interacțiunea dintre medici și pacienți, gestionarea datelor medicale și diseminarea informațiilor educaționale în domeniul sănătății. Aplicația se adresează instituțiilor medicale de dimensiuni mici și medii, cabinetelor medicale independente și organizațiilor care doresc să implementeze un sistem digital de management sanitar fără a investi în soluții costisitoare.

Scopul principal al aplicației este de a centraliza într-o singură platformă mai multe funcționalități esențiale: gestionarea utilizatorilor, evitarea nevoiei de prezentare fizică pentru realizarea unei programări, publicarea și gestionarea articolelor educative, interacțiunea socială prin sistemul de comentarii, administrarea centralizată a întregului conținut și utilizatorilor. Aplicația operează în domeniul healthcare IT și e-learning medical, combinând elementele ambelor secțoare într-o soluție unitară și coerentă.

4.2 Actori ai sistemului

Aplicația este construită pe un model de roluri care permite diferențierea clară a permisiunilor și funcționalităților. Există trei tipuri principale de actori în sistem, fiecare cu un set distinct de funcționalități și responsabilități.

4.2.1 Vizitator (Utilizator neautentificat)

Vizitatorul este orice persoană care accesează aplicația fără a fi autentificată în sistem. Acesta poate vizualiza pagina principală care prezintă articolele populare, poate naviga către pagina de înregistrare pentru a crea un cont nou, poate accesa pagina de autentificare pentru a se conecta la un cont existent, și poate consulta articolele educaționale din secțiunea EDU fără a putea interacționa cu acestea.

4.2.2 Pacient (Utilizator autentificat cu rol de pacient)

Pacientul este un utilizator înregistrat care are acces la funcționalități de programare, vizualizare și interacțiune. Poate să vizualizeze și să editeze propriul profil, inclusiv date personale și biometrice. Poate accesa și citi toate articolele educaționale publicate de medici. Poate interacționa cu articolele prin sistemul de like și dislike. Poate adăuga comentarii la articole și poate răspunde la comentarii existente.

4.2.3 Doctor (Utilizator autentificat cu rol de doctor)

Doctorul este un utilizator specializat care are toate funcționalitățile pacientului plus capacitatea de a trimite rezultate pacienților de a crea conținut educațional. Poate crea articole noi cu titlu, conținut formatat și imagine asociată. Poate edita și șterge propriile articole publicate. Poate vizualiza statistici despre articolele sale, inclusiv numărul de vizualizări și rating-ul. Poate interacționa cu articolele altor doctori prin like, dislike și comentarii. Are acces la un dashboard personalizat unde poate gestiona toate articolele sale.

4.2.4 Administrator (Utilizator cu rol de admin)

Administratorul este utilizatorul cu cele mai multe permișii în sistem. Are control complet asupra tuturor utilizatorilor, putând să îi vizualizeze, să îi editeze și să îi steargă. Poate gestiona toate articolele din sistem. Poate șterge comentarii inadecvate sau spam. Poate vizualiza statistici globale despre platformă. Are acces la un dashboard dedicat cu funcționalități avansate de administrare.

4.3 Cerințe non-funcționale

Cerințele non-funcționale descriu calitățile și caracteristicile sistemului din perspectiva performanței, securității, utilizabilității și menținabilității.

4.3.1 Performanță

Aplicația trebuie să se încarce rapid, cu un timp de răspuns sub trei secunde pentru pagina principală și sub cinci secunde pentru pagini complexe cu multe articole. Cerele către baza de date sunt optimizate prin utilizarea de indecsi și prin limitarea numărului de join-uri complexe. Imaginele sunt comprimate pentru a reduce dimensiunea lor fără a afecta semnificativ calitatea vizuală.

4.3.2 Securitate

Securitatea este o prioritate majoră. Toate parolile sunt hash-uite folosind bcrypt cu un cost factor adecvat. Toate cererile către baza de date folosesc prepared statements pentru a preveni injecțiile SQL. Datele afișate în HTML sunt sanitizate folosind htmlspecialchars pentru a preveni atacurile XSS. Sesiunile sunt gestionate securizat cu regenerarea ID-ului după autentificare și cu parametri de cookie securizați. Upload-ul de fișiere este restricționat la anumite extensii și dimensiuni pentru a preveni încărcarea de scripturi malicioase.

4.3.3 Utilizabilitate

Interfața utilizatorului este intuitivă și ușor de navigat. Design-ul este consistent pe întreaga aplicație, cu scheme de culori și layout-uri similare. Mesajele de eroare și de succes sunt clare și ajută utilizatorul să înțeleagă ce s-a întâmplat. Formulare sunt validate atât pe client-side cât și pe server-side, oferind feedback imediat utilizatorului.

4.3.4 Scalabilitate

Aplicația este construită modular, permitând adăugarea de noi funcționalități fără a necesita restructurări majore. Baza de date este normalizată și indexată corespunzător pentru a suporta un număr mare de utilizatori și articole. Codul este bine documentat și urmează principiile SOLID pentru a facilita extinderea și menținabilitatea.

4.3.5 Compatibilitate

Aplicația funcționează pe toate browserele moderne, inclusiv Chrome, Firefox, Safari și Edge. Design-ul responsive asigură că interfața se adaptează la diferite dimensiuni de ecran, de la smartphone-uri la desktop-uri. Aplicația nu necesită plug-in-uri sau extensii speciale pentru a funcționa corect.

4.3.6 Menținabilitate

Codul sursă este organizat logic în directoare și fișiere clare. Funcționalitățile sunt separate în componente distincte pentru a facilita debugging-ul și actualizările. Comentariile în cod explică logica complexă și ajută viitorii dezvoltatori să înțeleagă rapid funcționarea sistemului.

4.4 Fluxuri de lucru principale

Pentru a înțelege mai bine funcționarea aplicației, prezentăm câteva fluxuri de lucru principale care ilustrează interacțiunea utilizatorului cu sistemul.

4.4.1 Fluxul de înregistrare și autentificare

Utilizatorul nou accesează pagina principală și selectează opțiunea de înregistrare. Completează formularul cu email și parolă, apoi trimită formularul. Sistemul validează datele, creează contul și trimită un email de verificare. Utilizatorul verifică email-ul, accesează link-ul de confirmare și este redirecționat către pagina de login. Se autentifică cu credențialele și este redirecționat către dashboard corespunzător rolului său.

4.4.2 Fluxul de publicare articol

Medicul autentificat accesează dashboard-ul și selectează tab-ul de creare articol. Completează titlul, scrie conținutul folosind editorul TinyMCE, încarcă o imagine și trimită formularul. Sistemul validează datele, salvează articoul în baza de date și redirecționează medicul către lista articolelor sale. Articolul nou publicat apare imediat în secțiunea EDU și poate fi accesat de toți utilizatorii.

4.4.3 Fluxul de interacțiune cu articol

Utilizatorul autentificat accesează secțiunea EDU și navighează printre articole. Selectează un articol de interes și îl citește. După lectură, decide să dea like articoului apăsând butonul

corespunzător. Sistemul înregistrează like-ul și actualizează scorul articolului în timp real. Utilizatorul decide să adauge un comentariu, scrie mesajul în formularul de comentarii și îl trimit. Comentariul apare imediat sub articol și notifică autorul despre noul feedback.

4.4.4 Fluxul de administrare utilizatori

Administratorul se autentifică și accesează dashboard-ul de administrare. Selectează tab-ul de gestiune utilizatori și vizualizează lista completă. Identifică un utilizator care trebuie editat, selectează opțiunea de editare și modifică datele necesare. Salvează modificările iar sistemul actualizează informațiile în baza de date. Administratorul decide să steargă un utilizator inactiv, selectează opțiunea de ștergere, confirmă acțiunea și utilizatorul este eliminat împreună cu toate datele asociate.

4.5 Concluzii asupra specificațiilor și cerințelor

Acest capitol a prezentat în detaliu specificațiile și cerințele aplicației de management sanitar dezvoltate. Actorii sistemului au fost identificați și rolurile lor au fost definite clar. Cerințele funcționale și non-funcționale au fost enumerate și explicate, oferind o imagine completă asupra a ceea ce aplicația poate și trebuie să facă. Fluxurile de lucru principale au ilustrat modul în care utilizatorii interacționează cu sistemul în scenarii reale. Împreună, aceste informații oferă o bază solidă pentru înțelegerea completă a aplicației și a modului în care aceasta răspunde nevoilor utilizatorilor din domeniul medical.

5 Implementare

5.1 Prezentare generală

Aplicația web este făcută să faciliteze interacțiunea dintre clienți și personalul medical. Principalele avantaje ale platformei includ accesibilitatea ridicată prin interfața web responsivă care se adaptează la orice dispozitiv, securitatea datelor personale prin mecanisme avansate de autentificare și verificare email, precum și facilitarea comunicării dintre medici și pacienți prin intermediul secțiunii educaționale interactive cu sistem de comentarii și aprecieri.

Aplicația implementează un sistem de roluri care face diferența clară între administrator, doctor, pacient și vizitator, oferind fiecărei categorii acces la funcționalități specifice și relevante pentru nevoile lor.

5.2 Arhitectura sistemului

Arhitectura aplicației urmează modelul pe trei niveluri, separând clar responsabilitățile între componenta de prezentare, logica de afaceri și persistența datelor. Această abordare asigură menținabilitatea codului și posibilitatea extinderii funcționalităților în iterații viitoare.

5.2.1 Structura directoarelor

Proiectul este organizat într-o structură logică de directoare care facilitează navigarea și întreținerea codului. Directorul principal conține fișierele PHP ale paginilor accesibile utilizatorilor, inclusiv index.php pentru pagina principală, login.php și register.php pentru autentificare, edu.php pentru secțiunea educațională și post.php pentru vizualizarea articolelor individuale.

Subdirectorul config găzduiește fișierele de configurare pentru conexiunea la baza de date și pentru sistemul de trimitere email prin PHPMailer. Directorul css conține folie de stil organizate pe componente reutilizabile în subdirectorul components și stiluri specifice paginilor în subdirectorul pages.

Directorul includes conține componentele PHP reutilizabile precum bara de navigare și formularul de programări. Directorul uploads este structurat în subdirectoare pentru imaginile de profil și pentru imaginile atașate postărilor educaționale.

5.2.2 Componenta de prezentare

Nivelul de prezentare este responsabil pentru interfața utilizator și interacțiunea cu aceasta. Paginile PHP generează conținut HTML5 dinamic, utilizat prin foloarea stilurilor CSS3 care implementează Flexbox și Grid pentru layouturi moderne și responsive.

Interfața utilizator respectă principiile designului responsiv, adaptându-se automat la dimensiuni-

nile ecranului dispozitivului. Tema vizuală utilizează o paletă de culori consistentă cu accentul principal roșu definit prin codul hexazecimal #d82323, care oferă identitate vizuală distinctă și facilitează recunoașterea elementelor interactive.

5.2.3 Componenta de logică

Logica aplicației este implementată în fișierele PHP utilizând paradigma procedurală cu funcții bine definite. Autentificarea și autorizarea sunt gestionate prin mecanisme de sesiuni PHP native, cu verificări de roluri la nivelul fiecărei pagini protejate.

Procesarea formularelor utilizează metoda POST pentru transmiterea datelor sensibile și implementează validări atât pe partea de client prin JavaScript, cât și pe partea de server pentru asigurarea integrității datelor. Toate interogările către baza de date utilizează declarații pregătite MySQLi pentru prevenirea atacurilor de tip SQL Injection.

5.2.4 Componenta de date

Persistența datelor este asigurată de sistemul de gestiune a bazelor de date MariaDB. Schema bazei de date este normalizată și include tabele pentru utilizatori, detalii utilizatori, postări educaționale, comentarii și aprecieri.

Tabela User stochează informațiile de autentificare inclusiv email, parola hash-uită și rolul utilizatorului. Tabela UserDetails extinde profilul cu informații precum numele, prenumele, data nașterii și calea către imaginea de profil. Tabela EduPosts conține articolele educaționale cu titlu, conținut formatat HTML, imagine reprezentativă și referință către autor.

Relațiile dintre tabele sunt implementate prin chei străine care asigură integritatea referențială. Stergerea unui utilizator declanșează stergerea în cascadă a postărilor și comentariilor asociate.

5.3 Diagrame și ilustrații

Pentru o înțelegere vizuală mai bună a funcționalităților și a fluxurilor din aplicație, prezentăm în continuare o serie de diagrame reprezentative.

5.3.1 Diagrama Use Case

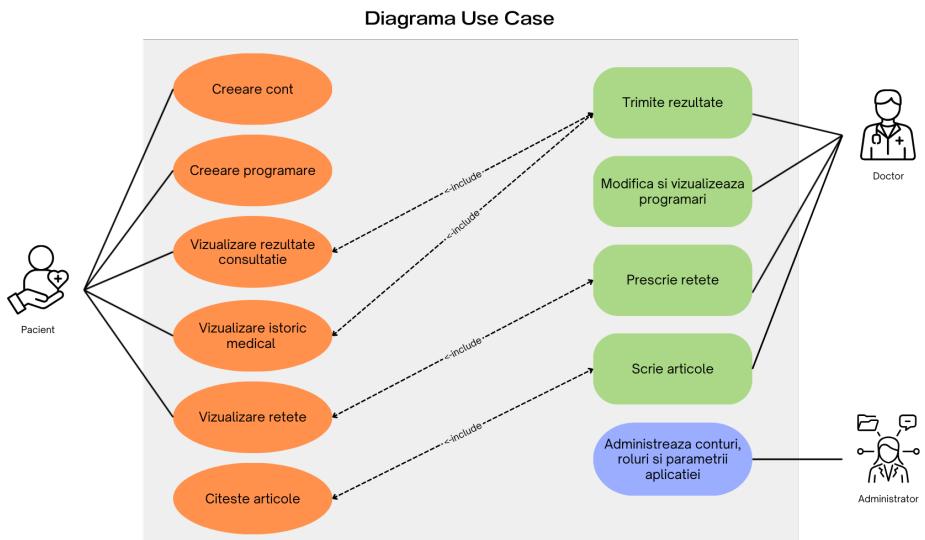


Diagrama Use Case ilustrează interacțiunile principale dintre actori și sistem. Aici putem observa actorii: administratorul cu cel mai înalt nivel de permisiuni, doctorul care poate crea conținut educațional și trimite rezultate și pacientul care consumă conținut și interacționează cu acesta.

5.3.2 Diagrama de activitate

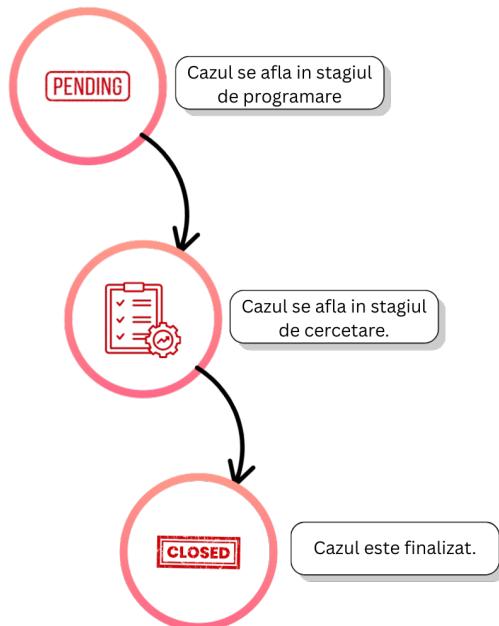


Diagrama de stare descrie starile unui caz.

5.3.3 Workflow Diagram

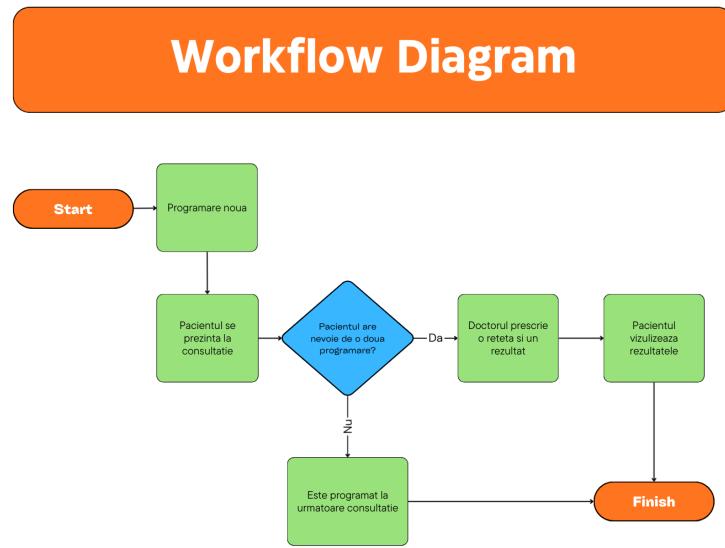


Diagrama Workflow ilustrează fluxul complet de la început până la sfârșit pentru procesul din aplicație.

Pacientul inițiază procesul prin înregistrarea unei programări. După efectuarea programării, aceasta îi apare și medicului în sistemul de gestiune. După terminarea vizitei medicale, medicul este nevoit să încarce concluzia vizitei în aplicație pentru ca pacientul să o poată vizualiza și el, asigurând astfel actualizarea dosarului medical electronic.

5.4 Implementarea funcționalităților principale

5.4.1 Sistemul de autentificare și înregistrare

Sistemul de autentificare reprezintă componenta fundamentală care asigură securitatea platformei. Implementarea utilizează funcțiile native PHP pentru gestionarea sesiunilor și funcția password_hash cu algoritmul bcrypt pentru stocarea securizată a parolelor.

Procesul de înregistrare începe cu completarea formularului care solicită numele, prenumele, adresa de email și parola. Sistemul validează formatul emailului și verifică unicitatea acestuia în baza de date. Parola este validată pentru a respecta cerințele minime de complexitate incluzând lungimea de cel puțin opt caractere.

După validarea cu succes, sistemul generează un token unic de verificare utilizând funcția bin2hex aplicată pe bytes aleatori generați criptografic. Tokenul are o durată de viață limitată la 24 de ore. Utilizatorul primește un email cu linkul de activare care conține tokenul generat.

La autentificare, parola introdusă este verificată contra hash-ului stocat utilizând password_verify. Sistemul verifică și starea de confirmare a contului, refuzând accesul pentru conturile neconfirmate și afișând un mesaj explicativ.

5.4.2 Sistemul de programări medicale

Modulul de programări permite pacienților să rezerve consultații medicale direct din platformă. Interfața prezintă un formular structurat care solicită selectarea specialității medicale, alegerea medicului disponibil, specificarea datei și intervalului orar preferat.

Sistemul implementează validări pentru prevenirea programărilor în trecut și pentru asigurarea disponibilității medicului în intervalul selectat. Interogările verifică absența suprapunerilor cu programări existente înainte de confirmarea rezervării.

Pacientul poate vizualiza propriile programări într-o secțiune dedicată și are posibilitatea de a anula programările viitoare. Medicii pot vizualiza în panoul lor de control lista programărilor asociate, organizate cronologic.

5.4.3 Secțiunea educațională

Secțiunea educațională reprezintă nucleul funcțional al platformei, oferind un spațiu unde medicii pot publica conținut informativ și pacienții pot accesa informații medicale de încredere.

Crearea articolelor utilizează editorul TinyMCE versiunea 6 care oferă o experiență de editare similară procesoarelor de text desktop. Configurația personalizată activează plugin-urile pentru formatare avansată inclusiv stilizarea textului, inserarea listelor și tabelelor, adăugarea linkurilor și încărcarea imaginilor inline.

Pentru articolele cu conținut științific, editorul permite inserarea formulelor matematice prin sintaxa LaTeX. Biblioteca MathJax încărcată în paginile de vizualizare interpretează și randează ecuațiile într-un format vizual profesional, esențial pentru explicarea conceptelor medicale complexe.

Sistemul de interacțiuni implementează aprecieri pozitive și negative pentru fiecare articol, cu prevenirea votului multiplu de la același utilizator. Comentariile permit discuții structurate, iar autorii primesc feedback direct asupra conținutului publicat.

5.4.4 Panoul de administrare

Panoul de administrare oferă control complet asupra platformei pentru utilizatorii cu rol de administrator. Interfața este organizată în tab-uri care separă funcționalitățile pentru gestionarea utilizatorilor, gestionarea articolelor și vizualizarea statisticilor.

Secțiunea de gestionare utilizatori permite vizualizarea listei complete de conturi, editarea informațiilor și rolurilor, precum și ștergerea conturilor cu confirmare. Administratorul poate modifica rolul oricărui utilizator între pacient și doctor.

Secțiunea de gestionare articole oferă acces la toate postările educaționale, indiferent de autor. Administratorul poate edita conținutul articolelor pentru corectarea erorilor sau actualizarea informațiilor și poate șterge articolele care încalcă regulamentul platformei.

5.4.5 Panoul doctorului

Panoul dedicat medicilor oferă funcționalități specifice pentru gestionarea activității profesionale pe platformă. Interfața prezintă vizualizarea programărilor pacienților organizate pe zile și intervale orare.

Secțiunea de gestionare a articolelor proprii permite doctorului să vizualizeze toate postările publicate, să le editeze pentru actualizări sau corecții și să le șteargă dacă nu mai sunt relevante. Fiecare articol afișează statisticile de interacțiune inclusiv numărul de aprecieri și comentarii.

5.5 Detalii tehnice de implementare

5.5.1 Conexiunea la baza de date

Conexiunea la baza de date este centralizată într-un fișier de configurare dedicat care definește parametrii de conectare și initializează obiectul mysqli. Configurația include numele serverului, numele bazei de date, utilizatorul și parola, permitând modificarea facilă a acestor parametri pentru diferite medii de dezvoltare sau producție.

Toate operațiunile cu baza de date utilizează declarații pregătite pentru prevenirea atacurilor SQL Injection. Parametrii sunt legați explicit folosind metoda bind_param cu specificarea tipurilor de date corespunzătoare pentru stringuri, întregi și alte formate.

5.5.2 Sistemul de încărcare imagini

Încărcarea imaginilor implementează validări multiple pentru asigurarea securității. Tipul fișierului este verificat atât prin extensie cât și prin analiza header-elor MIME pentru prevenirea încărcării de fișiere malicioase deghizate.

Dimensiunea fișierelor este limitată la un maxim configurabil pentru prevenirea consumului excesiv de spațiu de stocare. Numele fișierelor sunt regenerate utilizând identificatori unici pentru evitarea conflictelor și suprascrierii accidentale.

Imaginiile de profil sunt stocate în directorul uploads/profiles iar imaginile articolelor în uploads/posts. La ștergerea unui articol sau profil, imaginile asociate sunt eliminate din sistemul de fișiere pentru eliberarea spațiului.

5.5.3 Sistemul de email

Trimitera emailurilor utilizează biblioteca PHPMailer care oferă suport pentru protocolul SMTP și autentificare securizată. Configurația include serverul SMTP, portul, credențialele de autentificare și setările de criptare TLS.

Emailurile de verificare sunt formatare în HTML pentru o prezentare profesională și includ linkul de activare cu tokenul generat. Sistemul gestionează erorile de trimitere și oferă mesaje explicative utilizatorului în caz de eșec.

5.5.4 Sanitizarea datelor

Toate datele primite de la utilizatori sunt sanitizate înainte de procesare și afișare. Funcția `htmlspecialchars` este aplicată pentru prevenirea atacurilor Cross-Site Scripting, convertind caracterele speciale HTML în entități sigure.

Datele introduse în formulare sunt validate pentru format și lungime. Adresele email sunt verificate utilizând filter `_var` cu filtrul `FILTER_VALIDATE_EMAIL`. Parolele sunt verificate pentru lungime minimă și complexitate.

5.6 Interfața utilizator

5.6.1 Componentele vizuale

Sistemul de componente vizuale asigură consistență în întreaga aplicație. Butoanele utilizează stiluri uniforme cu tranzitii animate la interacțione. Formularele prezintă câmpuri cu etichete clare și mesaje de eroare vizibile.

Cardurile articolelor afișează imaginea reprezentativă, titlul, un fragment din conținut, avatarul și numele autorului, precum și statisticile de interacțione. Hero section-ul paginii principale prezintă platforma cu un design atractiv pe fundal alb care evidențiază elementele importante.

5.6.2 Feedback vizual

Aplicația oferă feedback vizual pentru toate acțiunile utilizatorului. Mesajele de succes și eroare sunt afișate în zone dedicate cu culori distinctive pentru verde în caz de succes și roșu pentru erori. Butoanele prezintă stări de hover și active pentru confirmarea interacționii.

Încărcarea paginilor și procesarea formularelor pot afișa indicatori de progres pentru informarea utilizatorului despre starea operațiunii în curs.

5.6.3 Organizarea sarcinilor

Proiectul a fost împărțit în module funcționale independente care au fost dezvoltate și testate separat. Primul modul a cuprins sistemul de autentificare și gestionarea utilizatorilor. Al doilea modul a implementat secțiunea educațională cu editorul de articole. Al treilea modul a dezvoltat sistemul de programări medicale. Al patrulea modul a finalizat panourile de administrare și doctor.

Fiecare modul a parcurs etapele de analiză a cerințelor, implementare, testare și integrare. Problemele identificate au fost corectate înainte de trecerea la modulul următor pentru evitarea acumulării datoriei tehnice.

5.7 Screenshot-uri reprezentative

În continuare sunt incluse capturile de ecran din aplicație care ilustrează vizual funcționalitățile descrise anterior.



Figura 5.1: Pagina principală cu hero section și articolele populare

Pagina principală prezintă mesajul de întâmpinare și butoanele de acțiune către secțiunile principale. Sub acesta sunt afișate cele mai apreciate articole educaționale în format card, oferind acces rapid la conținutul popular.

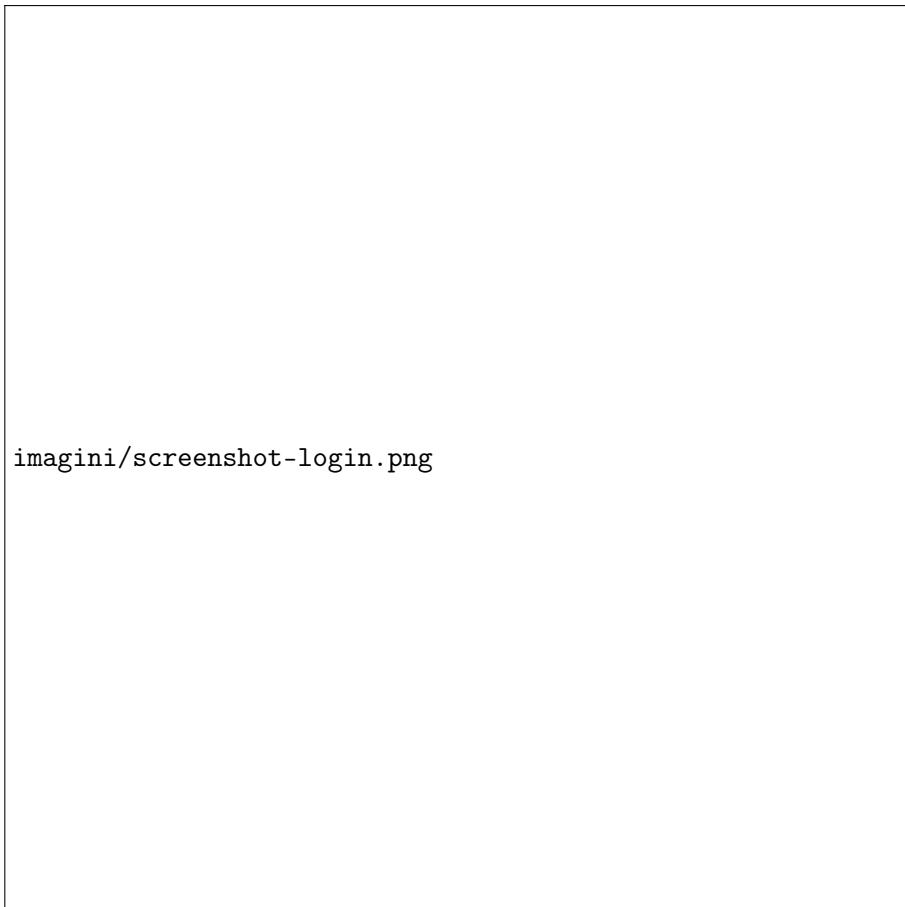


Figura 5.2: Formularul de autentificare cu design centrat

Formularul de autentificare prezintă un design modern centrat pe pagină, cu logo-ul platformei vizibil în partea superioară. Câmpurile pentru email și parolă sunt clar etichetate, iar linkurile către înregistrare și recuperare parolă sunt accesibile.

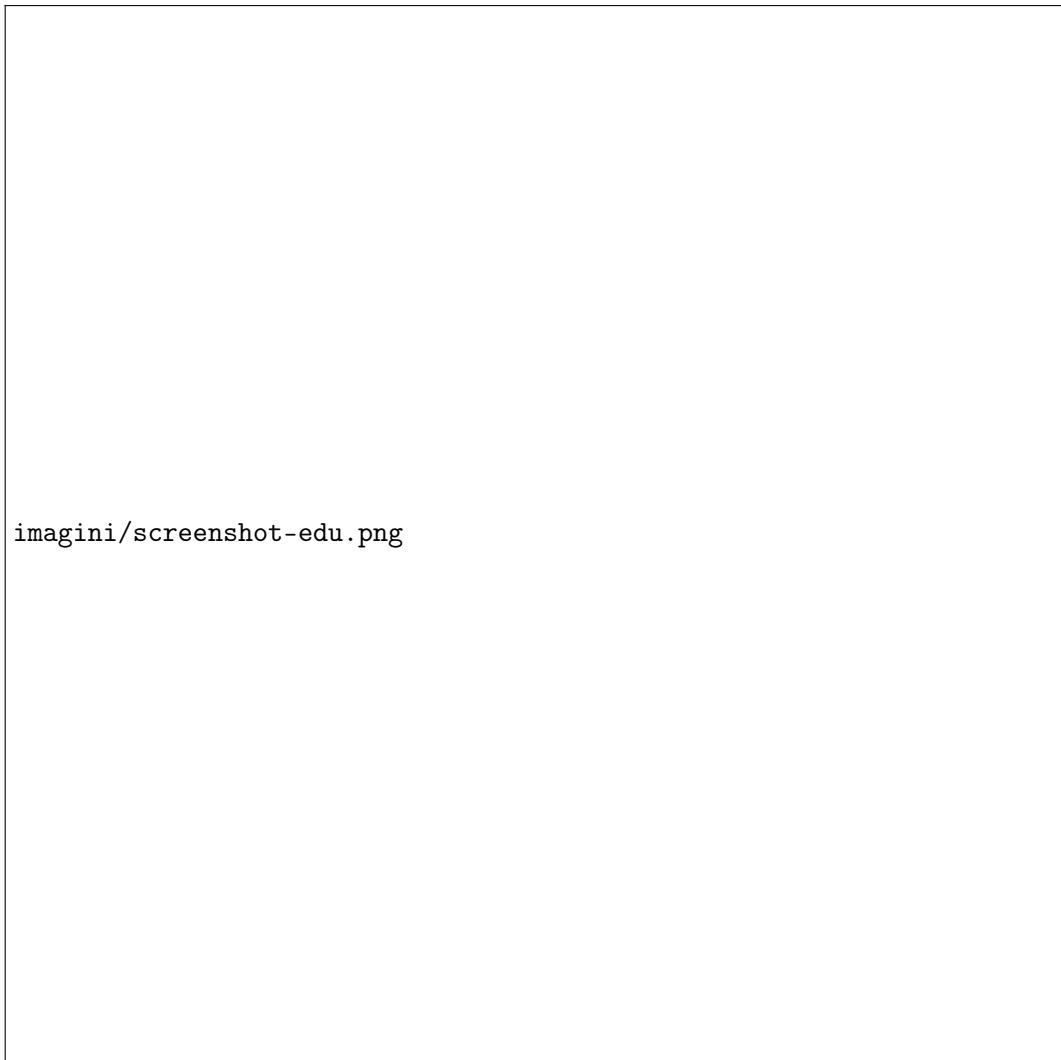


Figura 5.3: Sectiunea educatională cu lista articolelor

Sectiunea educatională afisează articolele în format card cu imaginea reprezentativă, titlul, fragmentul de conținut și informațiile despre autor. Avatarul și numele autorului sunt vizibile pentru fiecare articol, împreună cu numărul de aprecieri.

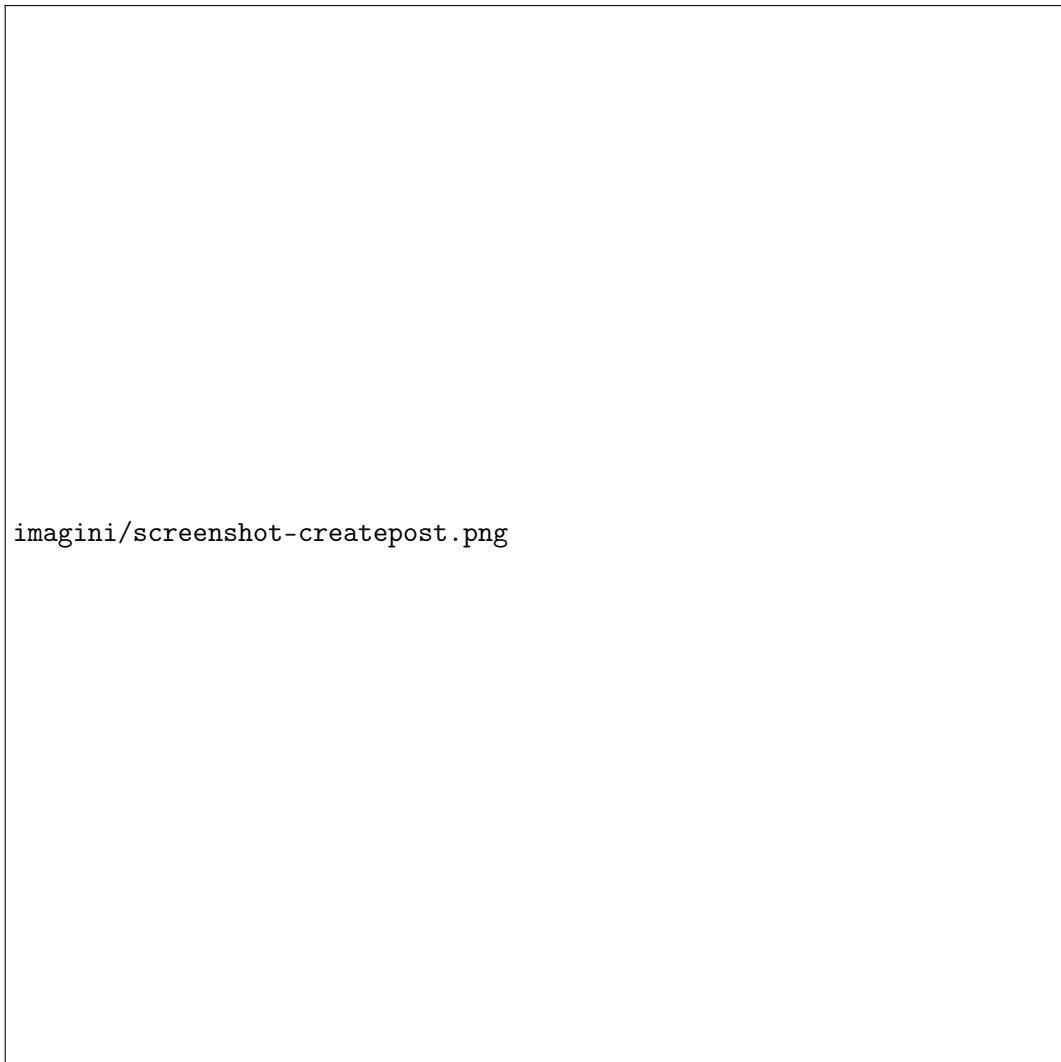


Figura 5.4: Interfața de creare articol cu editorul TinyMCE

Interfața de creare articol prezintă editorul TinyMCE cu bara de instrumente pentru formatare. Câmpul pentru titlu și zona de încărcare imagine sunt vizibile deasupra editorului, iar butonul de publicare este poziționat în partea inferioară.

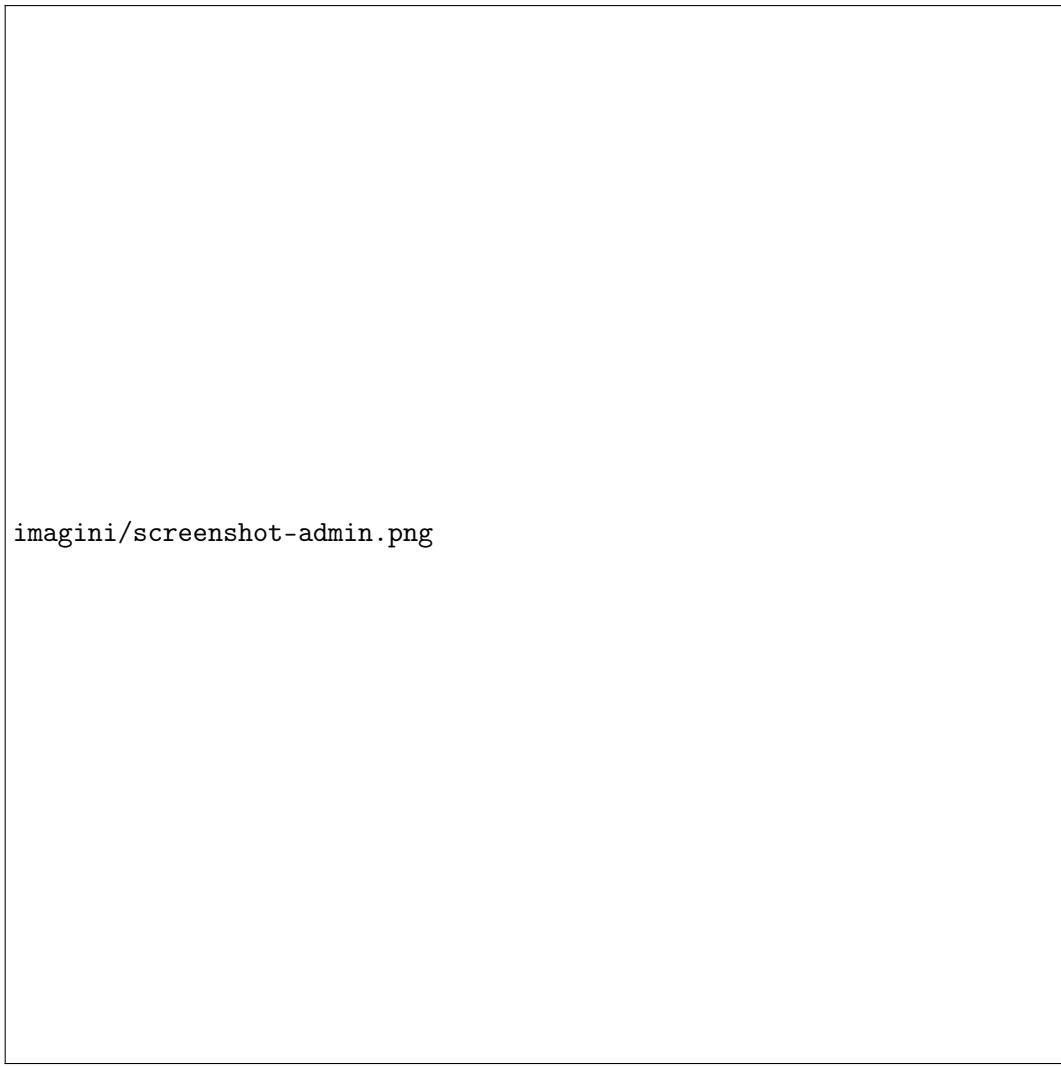


Figura 5.5: Panoul de administrare cu gestionarea utilizatorilor

Panoul de administrare afișează tab-urile pentru navigare între secțiuni și tabelul cu lista utilizatorilor. Fiecare rând prezintă informațiile utilizatorului și butoanele de acțiune pentru editare și stergere.

6 Testare și Validare

6.1 Introducere

Testarea și validarea sunt etape critice în procesul de dezvoltare software, asigurând că aplicația funcționează corect, este fiabilă și răspunde cerințelor utilizatorilor. Acest capitol prezintă abordarea utilizată pentru testarea platformei de management sanitar, tipurile de teste efectuate, cazurile de testare dezvoltate și rezultatele obținute.

6.2 Testare

6.2.1 Abordarea de testare

Pentru testarea aplicației a fost adoptată o abordare sistematică care combină mai multe tipuri de testare pentru a acoperi toate aspectele funcționale și non-funcționale ale sistemului. Testarea a fost efectuată manual, parcurgând fiecare funcționalitate și verificând comportamentul aplicației în diverse scenarii.

6.2.2 Tipuri de teste efectuate

Testarea funcțională

Testarea funcțională a verificat că toate funcționalitățile aplicației operează conform specificațiilor. Fiecare modul al aplicației a fost testat individual, inclusiv sistemul de autentificare și înregistrare, gestionarea profilului utilizatorului, platforma educațională EDU, dashboard-ul doctor, dashboard-ul administrator și navigarea generală.

Testarea validării datelor

Testarea validării datelor a verificat comportamentul aplicației la introducerea de date invalide sau neașteptate. Scenariile testate includ introducerea de caractere speciale în câmpuri de text, introducerea de litere în câmpuri numerice precum înălțime și greutate, introducerea de email-uri cu format invalid fără simbolul @ sau fără domeniu, încărcarea de fișiere cu extensii nepermise, lăsarea câmpurilor obligatorii goale și introducerea de parole care nu coincid în formularul de înregistrare.

Testarea proceselor specifice

Au fost testate procesele specifice ale aplicației pentru a verifica comportamentul în scenarii complexe. Acestea includ verificarea că un utilizator neautentificat nu poate accesa pagi-

nile protejate, verificarea că un pacient nu poate accesa dashboard-ul de administrator, testarea fluxului complet de înregistrare cu verificare email, testarea publicării și editării articolelor educaționale și testarea sistemului de like și dislike pentru articole.

Testarea securității

Testarea securității a verificat că aplicația este protejată împotriva atacurilor comune. Au fost testate protecția împotriva injectiilor SQL prin introducerea de cod SQL în câmpuri de input, protecția împotriva atacurilor XSS prin introducerea de cod JavaScript în formulare, verificarea că sesiunile expiră corect după logout și verificarea că utilizatorii nu pot accesa resurse care nu le aparțin.

6.2.3 Cazuri de testare

Tabelele următoare prezintă cazurile de testare dezvoltate pentru validarea aplicației, organizate pe categorii funcționale.

Teste pentru modulul de autentificare și înregistrare

Tabela 6.1: Plan de testare – Autentificare și Înregistrare

Funcționalitate	Întrebare test	Versiune	Data	Status
Logare	Deschide aplicația. Du-te pe logare. Introdu un username și parolă validă. A reușit logarea?	1.0	08/11/2025	PASS
Logare	Deschide aplicația. Du-te pe logare. Introdu un username greșit. Ai primit mesaj că username-ul nu există?	1.0	08/11/2025	PASS
Logare	Accesează login.php. Completează email corect. Completează parolă greșită. Click Intră în cont.	1.0	08/11/2025	PASS
Logare	Înregistrează un cont nou. NU verifică emailul. Încearcă să te loghezi.	1.0	08/11/2025	PASS
Logare	Accesează login.php. Completează Email sau nume fără @. Completează parola corectă. Click Intră în cont.	1.0	08/11/2025	PASS
Înregistrare	Accesează register.php. Completează email valid test@example.com. Nume Test User. Parolă Test123. Confirmă parola. Click Înregistrează-te.	1.0	08/11/2025	PASS
Înregistrare	Accesează register.php. Completează cu un email deja existent în BD. Click Înregistrează-te.	1.0	08/11/2025	PASS
Înregistrare	Accesează register.php. Completează Email testexample.com fără @. Click Înregistrează-te.	1.0	08/11/2025	PASS
Înregistrare	Accesează register.php. Parolă Test123. Confirmă parola Test456 (diferită). Click Înregistrează-te.	1.0	08/11/2025	PASS
Înregistrare	Accesează register.php. Lasă unul sau mai multe câmpuri goale. Click Înregistrează-te.	1.0	08/11/2025	PASS

Teste pentru verificare email

Tabela 6.2: Plan de testare – Verificare Email

Funcționalitate	Întrebare test	Versiune	Data	Status
Verificare email	Înregistrează un cont nou. Accesează emailul primit. Click pe linkul de verificare din email.	1.0	08/11/2025	PASS
Verificare email	Înregistrează un cont. Așteaptă expirarea token-ului (24h). Click pe linkul din email.	1.0	08/11/2025	PASS
Verificare email	Verifică un cont prin link. Click din nou pe același link.	1.0	08/11/2025	PASS
Sign up	Deschide mail. Apasă pe confirmare mail. Merge să te loghezi?	1.0	08/11/2025	PASS

Teste pentru profil și setări

Tabela 6.3: Plan de testare – Profil și Setări

Functionalitate	Întrebare test		Versiune	Data	Status
Profil	Loghează-te cu orice cont. Click pe numele tău din navbar și Profilul meu. Verifică secțiunea Informații personale.		1.0	08/11/2025	PASS
Profil	Accesează profilul. Verifică secțiunea Locație.		1.0	08/11/2025	PASS
Profil	Accesează profilul. Verifică secțiunea Informații fizice.		1.0	08/11/2025	PASS
Profil	Loghează-te cu cont care are poză încărcată. Accesează profilul.		1.0	08/11/2025	PASS
Profil	Loghează-te cu cont fără poză. Accesează profilul.		1.0	08/11/2025	PASS
Setări	Intră în cont pe setări. Adaugă date. Au fost actualizate pe profil?		1.0	08/11/2025	PASS
Setări	Deschide setări după ce ai introdus datele. Schimbă-le. Au fost schimbată?		1.0	08/11/2025	PASS
Setări	Accesează Settings. Modifica Nume, Telefon, Oraș. Click Salvează modificările.		1.0	08/11/2025	PASS
Setări	Accesează Settings. Click Alege fișier. Selecțează o imagine JPG/PNG sub 5MB. Click Salvează.		1.0	08/11/2025	PASS
Setări	Accesează Settings. Încearcă să încarci un PDF SAU imagine peste 5MB. Click Salvează.		1.0	08/11/2025	PASS
Setări	Accesează Settings. Modifică emailul cu unul care aparține altui utilizator. Click Salvează.		1.0	08/11/2025	PASS
Schimbare parolă	Intră pe setări. Schimbă parola. Încearcă să te loghezi. A funcționat?		1.0	08/11/2025	PASS

Teste pentru navigare și interfață

Tabela 6.4: Plan de testare – Navigare și Interfață

Funcționalitate	Întrebare test	Versiune	Data	Status
Navbar	Accesează orice pagină. Scroll în jos pe pagină. Observă navbar-ul.	1.0	08/11/2025	PASS
Navbar	Loghează-te. Hover cu mouse-ul peste numele tău din navbar (dreapta sus).	1.0	08/11/2025	PASS
Navbar	Fiind pe orice pagină. Click pe logo-ul din stânga navbar-ului.	1.0	08/11/2025	PASS
Navbar doctor	Loghează-te cu cont role doctor. Verifică navbar-ul.	1.0	08/11/2025	PASS
Logout	Click pe numele tău apoi ieșire. Verifică re-direct.	1.0	08/11/2025	PASS
Responsive	Redimensionează browser-ul la lățime mai mică de 768px SAU deschide site-ul pe telefon.	1.0	08/11/2025	PASS

Teste pentru permisiuni și securitate

Tabela 6.5: Plan de testare – Permisii și Securitate

Funcționalitate	Întrebare test	Versiune	Data	Status
Permisii	Intră în cont de pacient. Încearcă să introduci URL-ul pentru admin dashboard. Ai primit eroare?	1.0	08/11/2025	PASS
Delogare	Intră în cont de admin. Deloghează-te. Încearcă să intre pe admin dashboard delogat.	1.0	08/11/2025	PASS
Admin	Loghează-te ca pacient/doctor. Încearcă să accesezi direct admindashboard.php.	1.0	08/11/2025	PASS

Teste pentru dashboard administrator

Tabela 6.6: Plan de testare – Dashboard Administrator

Funcționalitate	Întrebare test	Versiune	Data	Status
Admin	Loghează-te cu cont role admin. Accesează admindashboard.php.	1.0	08/11/2025	PASS
Admin	Verifică tab Administrează conturi.	1.0	08/11/2025	PASS
Admin	Click pe tab Administrează conturi. Verifică conținutul.	1.0	08/11/2025	PASS
Admin	În admin dashboard. Click pe Editează la un utilizator.	1.0	08/11/2025	PASS
Admin	Click pe butonul roșu Șterge. Confirmă în dialog JavaScript.	1.0	08/11/2025	PASS
Admin	Utilizatori cu poză încărcată. Verifică în admin dashboard.	1.0	08/11/2025	PASS
Admin	Verifică utilizatori cu role diferite.	1.0	08/11/2025	PASS

Teste pentru programări

Tabela 6.7: Plan de testare – Programări

Funcționalitate	Întrebare test	Versiune	Data	Status
Programare	Intră în contul de user. Fă o programare. A funcționat?	1.0	08/11/2025	PASS
Programare	Intră în contul de user. Fă o programare. Îți apare în cont?	1.0	08/11/2025	PASS
Programare	Fă o programare. Intră pe contul de doctor. Își apare programarea doctorului corespunzător?	1.0	08/11/2025	PASS
Programare	Intră în contul de doctor. Pe o programare realizată. Adaugă un fișier invalid. A apărut eroarea?	1.0	08/11/2025	PASS
Programare	Intră în contul de doctor. Anulează o programare. A reușit anularea?	1.0	08/11/2025	PASS
Programare	Intră în contul de doctor. Finalizează o programare. Intră pe contul de user.	1.0	08/11/2025	PASS

6.3 Concluzii privind validarea aplicației

Procesul de testare și validare a confirmat că aplicația de management sanitar cu funcționalități educaționale este funcțională, fiabilă și pregătită pentru utilizare. Toate funcționalitățile principale au fost testate și validate cu succes.

Sistemul de autentificare oferă un nivel adecvat de securitate prin verificarea email-ului și gestionarea corectă a sesiunilor. Gestionarea profilurilor și setărilor permite utilizatorilor să își personalizeze conturile fără dificultăți. Sistemul de permișii bazat pe roluri funcționează corect, separând clar funcționalitățile disponibile pentru pacienți, doctori și administratori. Interfața utilizatorului este intuitivă și responsivă, funcționând corect pe diverse dispozitive și browsere.

Testarea a demonstrat că aplicația gestionează corect erorile, oferind feedback clar utilizatorilor în cazul introducerii de date invalide. Validarea datelor pe server-side asigură integritatea informațiilor stocate în baza de date.

În concluzie, aplicația a trecut cu succes toate testele planificate și este pregătită pentru deployment într-un mediu de producție. Se recomandă continuarea testării pe măsură ce noi funcționalități sunt adăugate și monitorizarea feedback-ului utilizatorilor pentru identificarea eventualelor probleme care nu au fost detectate în această fază de testare.

7 Concluzii și direcții viitoare

7.1 Sinteza contribuțiilor

Lucrarea de față a prezentat dezvoltarea unei Platforme Integrate de Management Sanitar cu Funcționalități Educaționale, o aplicație web care răspunde nevoilor concrete ale sistemului medical modern de a facilita comunicarea dintre personalul medical și pacienții și de a oferi acces la informații medicale de calitate.

Contribuția principală a acestui proiect constă în crearea unui sistem unificat care îmbină două componente esențiale: gestionarea programărilor medicale și distribuirea conținutului educațional de către profesioniști din domeniul sănătății. Această abordare integrată elimină necesitatea utilizării mai multor platforme separate și oferă utilizatorilor o experiență coerentă și eficientă.

Sistemul de autentificare implementat asigură securitatea datelor personale prin utilizarea algoritmilor moderni de hash-urile a parolelor și prin mecanismul de verificare a adresei de email. Această arhitectură previne accesul neautorizat și garantează că fiecare utilizator are un cont validat înainte de a putea interacționa cu funcționalitățile protejate ale platformei.

Secțiunea educațională reprezintă o contribuție semnificativă prin oferirea unui spațiu în care medicii pot publica articole informative cu formatare avansată, inclusiv suport pentru formule matematice prin sintaxa LaTeX. Sistemul de interacțiuni prin aprecieri și comentarii facilitează dialogul între autori și cititori, transformând platforma dintr-un simplu depozit de informații într-o comunitate activă de învățare.

Arhitectura bazată pe roluri diferențiază clar între administrator, doctor, pacient și vizitator, oferind fiecărei categorii acces granular la funcționalități relevante. Această separare asigură atât securitatea sistemului cât și o experiență de utilizare adaptată nevoilor specifice ale fiecărui tip de utilizator.

7.2 Analiza rezultatelor obținute

Evaluarea aplicației dezvoltate a demonstrat îndeplinirea obiectivelor stabilite inițial. Testarea manuală extensivă, care a cuprins 51 de cazuri de test distribuite pe categorii funcționale precum autentificare, gestionarea profilului, sistemul de programări și secțiunea educațională, a confirmat funcționarea corectă a tuturor componentelor implementate, cu o rată de succes de 100 la sută.

Interfața utilizator s-a dovedit intuitivă și accesibilă, beneficiind de designul responsiv care asigură funcționarea optimă pe dispozitive cu dimensiuni variate ale ecranului. Utilizatorii pot naviga cu ușurință între secțiuni, iar fluxurile de lucru pentru operațiuni precum crearea contului, publicarea articolelor sau programarea consultărilor sunt liniare și clare.

Performanța aplicației s-a menținut la parametri acceptabili pe parcursul testării, cu timpi de răspuns reduși pentru operațiunile frecvente. Utilizarea declarațiilor pregătite pentru interoga-riile bazei de date asigură atât securitatea cât și eficiența în gestionarea datelor.

Editorul de text avansat TinyMCE, integrat în formularul de creare articole, oferă medicilor instrumente profesionale pentru formatarea conținutului. Suportul pentru formule matematice prin MathJax extinde posibilitățile de prezentare a informațiilor științifice complexe într-un format vizual accesibil.

7.3 Limitări identificate

Pe parcursul dezvoltării și testării au fost identificate anumite limitări care ar putea fi adresate în iterării viitoare ale proiectului.

Sistemul actual de programări nu include un mecanism de notificare automată prin email sau SMS pentru a reaminti pacienților despre consultațiile viitoare. Această funcționalitate ar reduce rata de neprezentare și ar îmbunătăți eficiența activității medicale.

Absența unui sistem de mesagerie directă între medici și pacienți limitează comunicarea la interacțiunile prin comentariile articolelor educaționale. Un modul de conversații private ar facilita discuțiile confidențiale și ar oferi un canal suplimentar de comunicare.

Platforma nu include în prezent funcționalități de analiză și raportare avansată. Statisticile disponibile sunt limitate la contorizarea aprecierilor și comentariilor, fără vizualizări grafice sau rapoarte exportabile care ar putea oferi perspective valoroase asupra utilizării platformei.

Procesul de recuperare a parolei nu este complet implementat, utilizatorii care uită credențialele trebuind să contacteze administratorul pentru resetarea manuală a accesului. Automatizarea acestui flux ar îmbunătăți experiența utilizatorilor și ar reduce încărcarea administrativă.

Testarea a fost realizată preponderent în mediul de dezvoltare local, fără simularea condițiilor de încărcare ridicată care ar putea apărea într-un mediu de producție cu mulți utilizatori simultani.

7.4 Observații personale

Dezvoltarea acestui proiect a reprezentat o experiență valoroasă de învățare care a permis aprofundarea cunoștințelor în multiple domenii ale dezvoltării web. Lucrul cu tehnologii precum PHP, MySQL, JavaScript și integrarea bibliotecilor externe a oferit o perspectivă practică asupra procesului complet de creare a unei aplicații funcționale.

Alegerea arhitecturii procedurale PHP, deși mai puțin elegantă decât framework-urile moderne, s-a dovedit adecvată pentru scopul proiectului și a permis înțelegerea fundamentală a mecanismelor din spatele abstracțiilor oferite de framework-uri. Această experiență constituie o bază solidă pentru adoptarea ulterioară a unor soluții mai avansate precum Laravel sau Symfony.

Implementarea măsurilor de securitate a evidențiat importanța gândirii defensive în dezvolta-

rea software. Prevenirea atacurilor SQL Injection prin declarații pregătite și protecția contra Cross-Site Scripting prin sanitizarea datelor sunt practici esențiale care trebuie integrate de la începutul oricărui proiect.

Proiectul a subliniat și importanța designului centrat pe utilizator. Deciziile privind structura navigării, amplasarea elementelor interactive și feedback-ul vizual au fost ghidate de principiul accesibilității și ușurinței în utilizare.

Colaborarea dintre domeniul medical și cel tehnologic prezintă provocări specifice, în special în ceea ce privește prezentarea informațiilor științifice într-un format accesibil publicului larg. Integrarea editorului cu suport pentru formule matematice răspunde acestei nevoi, permitând medicilor să comunice concepte complexe fără compromisuri asupra acurateței.

7.5 Relevantă și impactul soluției

Platforma dezvoltată adresează o nevoie reală în contextul digitalizării serviciilor medicale. Accesul la informații medicale de calitate, validate de profesioniști, reprezintă un deziderat important într-o eră în care dezinformarea în domeniul sănătății poate avea consecințe grave.

Prin oferirea unui spațiu controlat în care doar personalul medical autorizat poate publica conținut, platforma contribuie la combaterea fenomenului de răspândire a informațiilor medicale eronate. Pacienții beneficiază de o sursă de încredere pentru educația în domeniul sănătății.

Componenta de programări simplifică procesul de accesare a serviciilor medicale, eliminând necesitatea apelurilor telefonice și reducând timpul de așteptare. Această funcționalitate are potențialul de a îmbunătăți eficiența activității medicale și satisfacția pacienților.

Arhitectura modulară a aplicației permite extinderea și adaptarea pentru diverse contexte de utilizare, de la cabinete medicale individuale până la clinici cu mai multe specialități. Flexibilitatea sistemului de roluri facilitează personalizarea conform structurii organizaționale specifice.

7.6 Direcții viitoare de dezvoltare

Platforma dezvoltată constituie o fundație solidă care poate fi extinsă și îmbunătățită în multiple direcții pentru a răspunde unor nevoi suplimentare și pentru a oferi funcționalități avansate.

7.6.1 Sistem de notificări

Implementarea unui sistem complet de notificări ar îmbunătăți semnificativ experiența utilizatorilor. Notificările prin email pentru programările viitoare, cu posibilitatea de configurare a intervalului de reamintire, ar reduce rata de neprezentare. Notificările pentru comentarii noi la articolele proprii ar menține autorii informați despre interacțiunile cititorilor. Un sistem de notificări în aplicație ar oferi actualizări în timp real fără necesitatea reîncarcării paginii.

7.6.2 Mesagerie internă

Dezvoltarea unui modul de mesagerie directă între utilizatori ar facilita comunicarea confidențială dintre medici și pacienți. Acest sistem ar putea include funcționalități precum atașarea fișierelor pentru partajarea rezultatelor analizelor, istoricul conversațiilor pentru referințe ulterioare și opțiuni de arhivare a discuțiilor relevante.

7.6.3 Telemedicină

Integrarea funcționalităților de telemedicină ar extinde considerabil utilitatea platformei. Consultațiile video în timp real, programate prin același sistem de rezervări, ar oferi o alternativă convenabilă pentru pacienții aflați la distanță. Implementarea ar necesita integrarea cu servicii de streaming video și asigurarea conformității cu reglementările privind confidențialitatea datelor medicale.

7.6.4 Aplicație mobilă

Dezvoltarea unei aplicații mobile native sau hibride ar îmbunătăți accesibilitatea platformei. Utilizatorii ar beneficia de notificări push, acces offline la articolele salvate și o experiență optimizată pentru interacțiunea pe dispozitive mobile. Sincronizarea cu platforma web ar asigura consistența datelor între dispozitive.

7.6.5 Funcționalități de accesibilitate

Îmbunătățirea accesibilității pentru utilizatorii cu dizabilități ar extinde audiența platformei. Implementarea conformității cu standardele WCAG, suportul pentru cititoare de ecran, opțiunile de contrast ridicat și dimensionare a textului ar asigura că platforma poate fi utilizată de toți pacienții, indiferent de abilitățile lor.

7.6.6 Multilingvism

Adăugarea suportului pentru mai multe limbi ar permite extinderea platformei în contexte internaționale sau în regiuni cu comunități multilingve. Sistemul de traduceri ar acoperi atât interfața aplicației cât și posibilitatea de a publica articole în mai multe limbi.

7.7 Concluzii finale

Proiectul prezentat în această lucrare demonstrează fezabilitatea și utilitatea unei platforme integrate care combină gestionarea programărilor medicale cu distribuirea conținutului educațional de calitate. Obiectivele stabilite inițial au fost îndeplinite, rezultând o aplicație funcțională care răspunde nevoilor identificate.

Tehnologiile alese s-au dovedit adecvate pentru scopul propus, oferind un echilibru între complexitate și funcționalitate. Arhitectura implementată permite extinderea și adaptarea aplicației pentru contexte diverse de utilizare.

Experiența acumulată pe parcursul dezvoltării constituie o bază valoroasă pentru proiecte vii-

toare mai complexe. Provocările întâmpinate și soluțiile identificate au contribuit la consolidarea competențelor tehnice și la înțelegerea aprofundată a procesului de dezvoltare software.

Platforma dezvoltată are potentialul de a contribui pozitiv la îmbunătățirea accesului la informații medicale de calitate și la eficientizarea procesului de programare a consultațiilor. Directiile viitoare de dezvoltare identificate oferă perspective clare pentru evoluția continuă a soluției.

În concluzie, proiectul reprezintă o contribuție practică în domeniul digitalizării serviciilor medicale, demonstrând că soluțiile tehnologice pot facilita semnificativ interacțiunea dintre profesioniștii din domeniul sănătății și pacienți. **[exemplu_carte]** **[exemplu_online]**

Multumiri (optional)

Aici se pot formula mulțumiri către coordonator, colegi, instituții sau finanțatori.

Anexe

Anexa A - Listă de figuri

Listă de figuri

5.1	Pagina principală cu hero section și articolele populare	32
5.2	Formularul de autentificare cu design centrat	33
5.3	Sectiunea educațională cu lista articolelor	34
5.4	Interfața de creare articol cu editorul TinyMCE	35
5.5	Panoul de administrare cu gestionarea utilizatorilor	36
7.1	Acesta este un grafic și text pe aceeași linie	55
7.2	Exemplu de grafic mai mic, centrat	55
7.3	Exemplu de două imagini pe aceeași linie	56
7.4	Exemplu de subfiguri (a) și (b)	56
7.5	Acesta este un grafic afisat pe toată latimea paginii	57
7.6	Aceasta este o imagine ce ocupa 0.7 din latimea paginii	58

Anexa B - Listă de tabele

Listă de tabele

6.1	Plan de testare – Autentificare și Înregistrare	39
6.2	Plan de testare – Verificare Email	40
6.3	Plan de testare – Profil și Setări	41
6.4	Plan de testare – Navigare și Interfață	42
6.5	Plan de testare – Permișuni și Securitate	42
6.6	Plan de testare – Dashboard Administrator	43
6.7	Plan de testare – Programări	43
7.1	Exemplu - Plan de testare – Studii de caz	52
7.2	Exemplul 1 - un model tabel cu caption sub tabel	58

7.3 Exemplul 2 - un model tabel cu caption deasupra	59
7.4 Exemplul 3 - model tabel	59

Anexa C - Listă de abrevieri (optional)

API — Interfață de Programare a Aplicațiilor;

DB — Bază de date;

ETL — Extract, Transform, Load.

Anexa D — Exemplificare cod/flux/etc. (exemplu)

Anexa E — Plan de testare

Tabela 7.1: Exemplu - Plan de testare – Studii de caz

Funcționalitate	Întrebare test	Versiune	Data	Status
Logare	Deschide aplicația. Mergi la logare. Introdu un username și o parolă valide. A reușit logarea?	1.0.1	04/09/2020	PASS
Logare	Deschide aplicația. Introdu un username greșit. Sistemul afișează mesaj că username-ul nu există?	1.0.1	04/09/2020	FAIL
Înregistrare	Completează formularul cu date valide. Contul se creează cu succes?	1.0.1	05/09/2020	PASS
Înregistrare	Lasă câmpul „email” gol. Aplicația afișează mesaj de eroare corespunzător?	1.0.1	05/09/2020	PASS
Upload fișier	Încarcă un fișier cu extensie invalidă. Sistemul refuză încărcarea și oferă un mesaj clar?	1.0.2	06/09/2020	FAIL
Procesare date	Încarcă un fișier valid. Sistemul procesează datele fără erori?	1.0.2	06/09/2020	PASS
Raportare	Generează raportul. Datele afișate sunt corecte și complete?	1.0.3	07/09/2020	PASS

Exemplu de structură și formatare

Acest document folosește un şablon care respectă cerințele de redactare ale facultății. Este gândit pentru tipărire față-verso, astfel încât capitolele noi și cuprinsul să poată începe pe pagină impară folosind comanda `\cleardoublepage`, dacă este nevoie.

Setările principale ale documentului (margini, font, spațiere, stilul titlurilor) sunt definite în preambulul fișierului `LATEX`. O modificare făcută acolo (de exemplu titlul lucrării sau numele autorului) se va reflecta automat în copertă, în antet/picior de pagină și, unde este cazul, în metadata PDF-ului.

Acest capitol explică modul în care se lucrează cu acest şablon: cum se structurează textul, cum se fac citările, cum se inserează referințe la figuri, tabele și listări de cod și unde se găsesc modelele complete. **Modelele efective pentru imagini, tabele, cod și planul de testare sunt incluse în aceasta secțiune.**

Structura capitolelor, secțiunilor și subsecțiunilor

Fiecare capitol începe cu:

```
\chapter{Titlul capitolului}
```

Secțiunile și subsecțiunile moștenesc numerotarea capitolului, de forma:

```
\section{Titlul secțiunii}      % Exemplu: 3.1  
\subsection{Titlul subsecțiunii} % Exemplu: 3.1.1
```

Sub-subsecțiunile pot exista, dar nu sunt recomandate pentru a evita fragmentarea textului.

Titlurile sunt deja formate prin pachetul `titlesec`, nu este necesar să fie stilizate manual.

Aceasta este o secțiune

Fiecare capitol este numerotat, iar secțiunile moștenesc numărul capitolului urmat de un punct și numărul secțiunii (de exemplu „3.1”). Figurile, tabelele și ecuațiile pot fi și ele numerotate pe capitole, pentru a fi mai ușor de referit în text.

Aceasta este o subsecțiune

Subsecțiunile sunt și ele numerotate (de forma „3.1.1”). În general este recomandat să nu se folosească o ierarhie prea adâncă de titluri (`\paragraph{}` și altele), deoarece textul devine fragmentat și mai greu de urmărit.

Aceasta este o sub-subsecțiune

Aceste titluri nu mai sunt numerotate, dar pot fi utile pentru a evidenția scurt anumite idei în interiorul unei subsecțiuni. Adâncimea maximă de numerotare poate fi controlată prin setarea `\setcounter{secnumdepth}{...}` în preambul.

Citarea surselor bibliografice (biblatex)

Acest şablon foloseşte biblatex, ceea ce înseamnă că toate sursele sunt incluse în fișierul `bibliography.bib` și citate astfel:

Citare normală în text:

Conform studiului prezentat în `\cite{exemplu_carte}..`

Citare multiplă:

Mai multe lucrări abordează subiectul `\cite{exemplu_carte, exemplu_online}.`

Citarea se actualizează automat în lista bibliografică generată la final:

`\printbibliography`

Referințe la figuri, tabele și listări de cod

Modelele complete pentru figuri, tabele, diagrame UML, listări de cod și planul de testare SUNT INCLUSE ÎN ANEXE (Anexa A, B, D, E).

Test de font și simboluri

Simbolurile și diacriticile pot fi scrise direct în document, fără comenzi speciale: à, â, î, š, ţ. De asemenea, putem avea diferite tipuri de cratimă și liniuță: - (minus simplu), – (liniuță en dash), — (liniuță em dash).

Acesta este un mic test de font:

Arial / Helvetica normal

Arial / Helvetica italic

Arial / Helvetica bold

Arial / Helvetica bold italic

Linkurile pot fi introduse cu pachetul `hyperref`, de exemplu: <https://www.utcluj.ro/>. Acestea vor fi clikcabile în PDF și pot apărea colorate sau subliniate, în funcție de setările definite în preambul.

Exemple de inserare a imaginilor

Aici aveti un Exemplu referinta Figura 7.6



imagini/image.png

Aici poti scrie un paragraf de text explicativ pentru imagine: ce reprezinta graficul, ce variabile sunt folosite, ce se observa important etc.

Figura 7.1: Aceasta este un grafic si text pe aceiasi linie

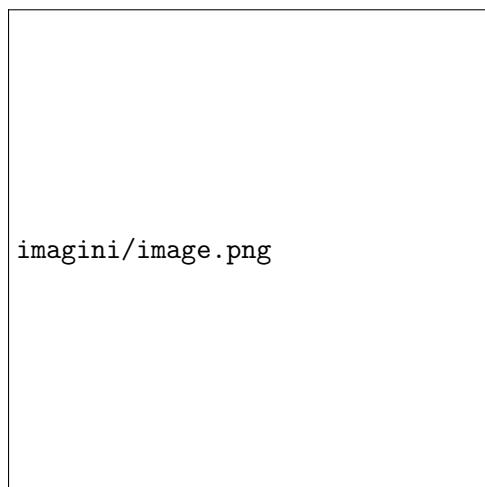


Figura 7.2: Exemplu de grafic mai mic, centrat

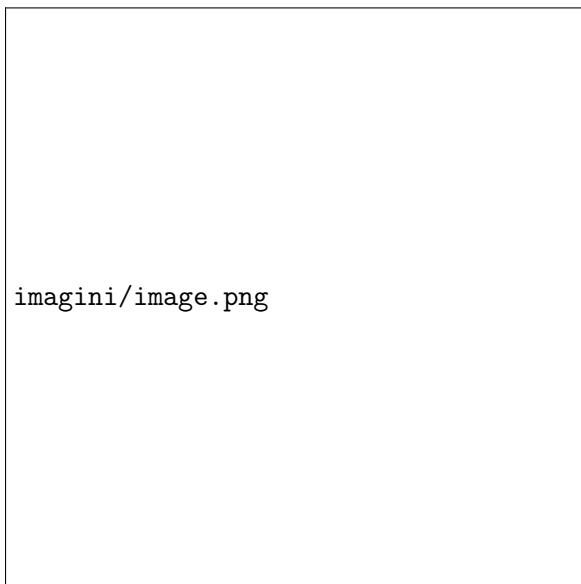


imagini/image.png



imagini/figura_use_case.png

Figura 7.3: Exemplu de doua imagini pe aceasi linie



imagini/image.png

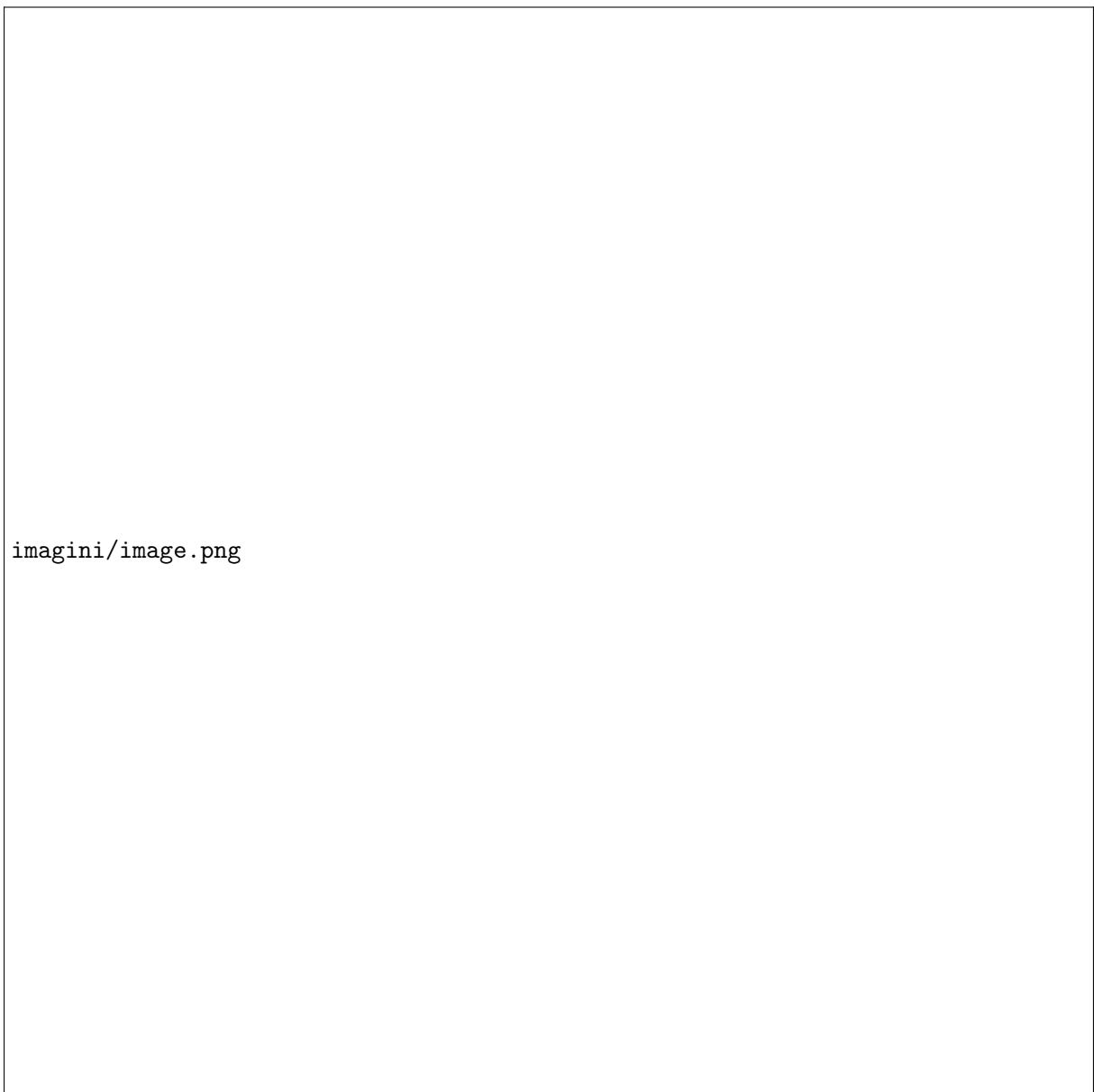


imagini/figura_use_case.png

(a) Grafic

(b) Diagrama de cazuri de utilizare

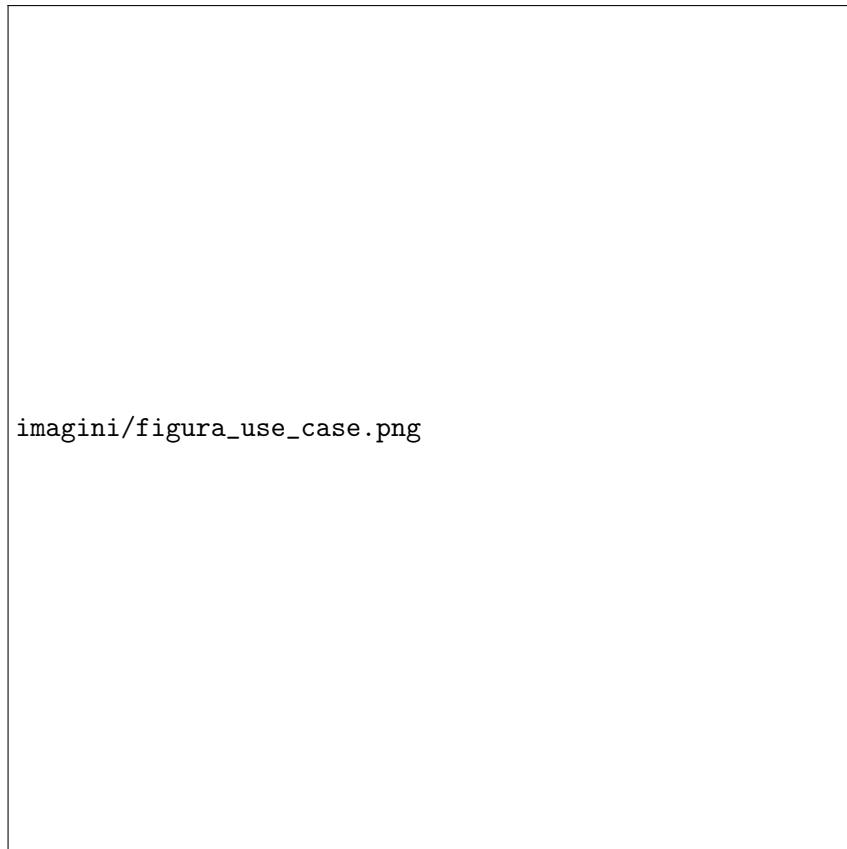
Figura 7.4: Exemplu de subfiguri (a) și (b)



imagini/image.png

Figura 7.5: Acesta este un grafic afisat pe toata latimea paginii

Exemplu referinta Figura 7.6



imagini/figura_use_case.png

Figura 7.6: Aceasta este o imagine ce ocupa 0.7 din latimea paginii

Exemple de tabele

Exemplu referinta la un tabel - Tabelul 7.2.

ID	Cerintă	Prioritate	Tip	Descriere suplimentară
FR-1	Autentificare utilizator	Ridicată	Funcțională	Utilizatorul se conectează folosind email și parolă.
FR-2	Gestionare cont	Medie	Funcțională	Utilizatorul poate actualiza datele personale.
FR-3	Căutare în aplicație	Ridicată	Funcțională	Sistemul oferă rezultate relevante după cuvinte cheie.
FR-4	Export date	Scăzută	Funcțională	Export în format PDF/CSV.
FR-5	Notificări	Medie	Funcțională	Alerte trimise prin email sau notificări interne.

Tabela 7.2: Exemplul 1 - un model tabel cu caption sub tabel

Tabela 7.3: Exemplul 2 - un model tabel cu caption deasupra

Algoritm	Acuratețe	Precizie	Recall
SVM	0.92	0.90	0.91
Random Forest	0.95	0.94	0.93
XGBoost	0.96	0.95	0.94
Logistic Regression	0.88	0.87	0.86

Tabela 7.4: Exemplul 3 - model tabel

Endpoint	Metodă	Descriere
/api/login	POST	Primeste email și parolă, validează credențialele și returnează un token JWT pentru autentificare.
/api/users	GET	Returnează lista completă a utilizatorilor. Acces permis doar administratorilor.
/api/users/id	GET	Returnează detalii despre un utilizator specificat prin ID.
/api/upload	POST	Permite utilizatorului să încarce un fișier pe server pentru procesare (ex: imagine, document, date brute).
/api/report	GET	Generează și returnează un raport în format JSON, pe baza datelor procesate anterior.

Cod. «Nr.» — Exemplu de listare de cod

```
1 def suma(a, b):
2     return a + b
3 print(suma(2, 3))
```

Listing 7.1: Exemplu de funcție

```
1 # Legend
2 legend_top = grid_y + len(rows)*row_h + 40
3
4 legend_items = [("C", "Consultat"), ("R", "Responsabil"), ("A", "Raspunzator"), ("I", "Informat")]
5 spacing = (W - 2*margin) // len(legend_items)
6 for i, (ltr, label) in enumerate(legend_items):
7     cx = margin + spacing*i + spacing//2
8     cy = legend_top + 30
9     draw_letter_circle(cx, cy, ltr)
10    draw.text((cx, cy+55), label, font=FONT_SMALL,
11               fill=text_dark, anchor="mm")
```

Listing 7.2: Exemplu pseudo-cod pentru legendă