



**FACULTATEA TEHNOLOGII INFORMAȚIONALE ȘI STATISTICĂ
ECONOMICĂ**
**DEPARTAMENTUL TEHNOLOGIA INFORMAȚIEI ȘI MANAGEMENT
INFORMAȚIONAL**

PROIECT DE AN

“Elaborarea unei platforme de mesagerie interactive”

Specialitatea 0410.4 Cibernetică și Informatică Economică

Autor:

Student gr. CIB 211 fr,

(învățământ cu frecvență/frecvență redusă)

Slutu Laurențiu

(semnătura)

Conducător științific:

Dr. Lector Universitar Lozan Victoria

(semnătura)

Chișinău – 2024

Cuprins

INTRODUCERE.....	3
Context și motivație.....	3
Obiectivele proiectului.....	4
1. PARTEA CONCEPTUALĂ	5
2.1 Cadrul teoretic	5
2.2 Obiectivele studiului/cercetării.....	6
2.3 Analiza situației în domeniul platformelor de mesagerie	7
2.4 Analiza situației în domeniul TIC în Republica Moldova	7
2.5 Aspectele de securitate și confidențialitate	7
2. PARTEA APLICATIVĂ	9
3.1 Infracstructura tehnologica	9
3.2 Interacționarea.....	12
CONCLUZII	14
REFERINȚE BIBLIOGRAFICE.....	16
ANEXE	17

INTRODUCERE

Context și motivație

Într-o eră dominată de avansul tehnologic și conectivitatea globală, necesitatea unei comunicări eficiente și interactive devine tot mai evidentă. În acest context, dezvoltarea unei platforme de mesagerie interactive devine o inițiativă cheie pentru a răspunde cerințelor și a oferi o soluție modernă utilizatorilor. Platforma noastră de mesagerie interactivă vine să rezolve anumite probleme identificate în peisajul actual al comunicațiilor online. Astfel, luând în considerare creșterea rapidă a utilizatorilor online, este imperativ să oferim o alternativă care să depășească limitările platformelor existente. Aceasta include necesitatea unei interacțiuni în timp real, securitatea datelor și o experiență intuitivă pentru utilizatori. Motivația din spatele acestui proiect constă în dorința de a aduce inovație în domeniul comunicării online și de a crea un instrument care să faciliteze interacțiunile interpersonale. Prin dezvoltarea acestei platforme cu framework-ul Laravel, avem în vedere nu doar rezolvarea unor probleme practice, ci și crearea unei baze pentru dezvoltarea ulterioară a funcționalităților și extinderii capacităților platformei. De asemenea, într-un mediu în care se observă o creștere a dependenței de comunicarea digitală, se pune un accent deosebit pe asigurarea securității datelor. Platforma noastră prioritizează implementarea măsurilor de securitate pentru a proteja confidențialitatea utilizatorilor și pentru a construi o încredere solidă în serviciul oferit. În concluzie, dezvoltarea acestei platforme de mesagerie interactivă nu se rezumă doar la tehnologie, ci are în spate o viziune de a transforma modul în care oamenii interacționează online. Această viziune este alimentată de recunoașterea nevoilor actuale ale utilizatorilor și dorința de a oferi o soluție care să răspundă acestor cerințe într-un mod inovator și eficient.

Obiectivele proiectului

Obiectivele proiectului nostru sunt aliniate cu viziunea noastră de a dezvolta o platformă de mesagerie interactivă cu impact semnificativ în mediul digital. Aceste obiective sunt definitorii pentru succesul proiectului și sunt formulate într-un mod clar și concis.

Crearea unei Platforme Robuste de Mesagerie Interactive Primul obiectiv al proiectului este să dezvoltăm o platformă de mesagerie interactivă robustă și fiabilă. Aceasta trebuie să ofere o experiență fluidă utilizatorilor, facilitând comunicarea în timp real și furnizând un mediu sigur pentru schimbul de informații.

Utilizarea Eficientă a Framework-ului Laravel Obiectivul nostru este să demonstrăm eficiența și puterea framework-ului Laravel în dezvoltarea aplicațiilor web. Vom utiliza caracteristicile specifice ale Laravel pentru a optimiza codul, a gestiona bazele de date și a asigura securitatea aplicației noastre.

Integrarea Funcționalităților Avansate Platforma noastră va fi echipată cu funcționalități avansate pentru a satisface nevoile diversificate ale utilizatorilor. Aceste funcționalități pot include mesagerie în timp real, gestionarea grupurilor de conversație, funcții de căutare și alte caracteristici inovatoare pentru o experiență îmbunătățită.

Focus Pe Experiența Utilizatorului Obiectivul nostru este să oferim o experiență de utilizare plăcută și intuitivă. Interfața utilizator va fi proiectată astfel încât să fie ușor de navigat, cu un design prietenos și interacțiuni clare. Aceasta contribuie la crearea unui mediu propice pentru comunicare și colaborare.

Implementarea Măsurilor de Securitate Securitatea datelor este o prioritate. Obiectivul nostru este să implementăm măsuri robuste de securitate pentru a proteja informațiile utilizatorilor. Acest lucru include criptarea datelor, gestionarea autentificării și prevenirea amenințărilor cibernetice. Aceste obiective reflectă direcția strategică a proiectului nostru și servesc ca piloni pentru dezvoltarea și evaluarea continuă a platformei de mesagerie interactivă. Prin atingerea acestor obiective, aspirăm să oferim o soluție de calitate care să răspundă nevoilor utilizatorilor și să aducă o contribuție semnificativă în domeniul comunicării online.

1. PARTEA CONCEPTUALĂ

2.1 Cadrul teoretic

Cadrul teoretic reprezintă prima componentă a proiectului și pune bazele pentru înțelegerea contextului și a fundalului conceptual necesare pentru dezvoltarea unei platforme de mesagerie interactivă. Acest capitol va explora și integra contribuții teoretice relevante din literatura de specialitate, construind astfel fundamente solide pentru proiect.

Se va realiza o analiză detaliată a literaturii de specialitate din domeniul comunicațiilor online și al platformelor de mesagerie interactivă. Se vor identifica principalele teorii, concepte și modele care au stat la baza dezvoltării unor astfel de platforme, evidențiind evoluția acestora în timp.

Situația actuală în domeniu, aceasta aduce în prim-plan situația actuală a platformelor de mesagerie interactive, analizând tendințele și inovațiile recente. Se va investiga cum aceste platforme au evoluat pentru a răspunde nevoilor în schimbare ale utilizatorilor.

Formularea scopului și obiectivelor, pe baza analizei literaturii de specialitate și a situației actuale, se va formula scopul general al proiectului și obiectivele specifice care vor ghida dezvoltarea platformei. Această formulare va fi susținută de contribuțiile teoretice relevante și va arăta direcția în care se îndreaptă cercetarea și dezvoltarea.

Integrarea concluziilor din cadrul teoretic, această va demonstra abilitatea autorului de a integra diverse contribuții teoretice într-o viziune coerentă și înțelegere comprehensivă a domeniului. Se vor evidenția concluziile și ipotezele elaborate pe baza analizei teoretice, oferind astfel o bază solidă pentru partea practică a proiectului.

Evaluarea critică și independența, un aspect crucial al acestei va fi abordarea critică și independentă a literaturii de specialitate. Se vor evidenția puncte de vedere divergente, limitări ale teoriilor existente și va propune propriile observații, contribuind astfel la dezvoltarea unei perspective critice și inovatoare asupra subiectului.

Prin această abordare detaliată a cadrului teoretic, se va asigura că fundamentul teoretic al proiectului este rezistent, că scopurile și obiectivele sunt bine definite și că contribuția originală este clar evidențiată în proiect.

Ca instrumente tehnologice de lucru sa efectuat o cercetare a pieții și sa identificat urmatoarele instrumente:

- Laravel

- Mysql
- Pusher
- Node.Js
- Composer

2.2 Obiectivele studiului/cercetării

Obiectivele studiului și cercetării în cadrul proiectului nostru sunt esențiale pentru a aduce inovație și contribuții semnificative în domeniul platformelor de mesagerie interactive. Acest set de obiective definesc direcțiile noastre specifice de cercetare și explorare.

Analiza tendințelor actuale în comunicația online, primul obiectiv este să analizăm și să înțelegem tendințele actuale în comunicarea online. Această cercetare ne va ajuta să identificăm cerințele ale utilizatorilor și să integrăm aceste observații în dezvoltarea platformei noastre.

Evaluarea soluțiilor existente și identificarea lacunelor, cercetarea va include evaluarea critică a soluțiilor existente în domeniul platformelor de mesagerie. Identificarea lacunelor și a limitărilor acestor soluții ne va oferi o bază solidă pentru a dezvolta o platformă inovatoare și performantă.

Explorarea metodologiilor eficiente de dezvoltare cu laravel, un alt obiectiv important este să explorăm și să adoptăm metodologii eficiente de dezvoltare cu framework-ul Laravel. Aceasta implică studiul detaliat al celor mai bune practici, a principiilor SOLID, și a modurilor optime de organizare a codului.

Identificarea oportunităților de integrare a tehnologiilor, cercetarea va viza identificarea și evaluarea tehnologiilor care pot aduce valoare adăugată platformei noastre. Aici, ne propunem să explorăm cum tehnologii precum machine learning sau blockchain pot fi integrate pentru îmbunătățirea funcționalităților

Contribuții la îmbunătățirea eficienței comunicației online, obiectivul final este să aducem contribuții semnificative la îmbunătățirea eficienței comunicației online prin implementarea de funcționalități inovatoare și soluții la problemele identificate în cercetare. Dorim să oferim o platformă care să redefinim standardele în domeniul mesageriei interactive. Prin atingerea acestor obiective de cercetare, ne propunem să aducem o contribuție distinctă și valoroasă în domeniul platformelor de mesagerie interactive. Cercetarea noastră nu se oprește doar la implementarea unei soluții practice, ci vizează să extindă cunoștințele în acest domeniu și să ofere

o perspectivă inovatoare asupra comunicației online. În figura 2.1 este prezentat metoda simplificată a trimiterii unui mesaj între 2 persoane.

2.3 Analiza situației în domeniul platformelor de mesagerie

Tendențele actuale în comunicația online vizează identificarea și înțelegerea tendințelor actuale în ceea ce privește comunicarea online. Se va analiza modul în care utilizatorii interacționează în mediul digital, incluzând preferințele lor în materie de funcționalități, interfețe și tipuri de conținut consumat în cadrul platformelor de mesagerie.

Evoluția platformelor de mesagerie, vom explora istoricul acestora, evidențiind momentele cheie care au marcat schimbări semnificative în modul în care oamenii interacționează digital. Această privire ne va oferi o perspectivă mai profundă asupra modului în care aceste platforme s-au adaptat și inovat în timp.

Inovații recente în domeniu, vom explora tehnologii inovatoare, funcționalități noi și abordări creative adoptate de platformele de mesagerie interactive, evaluând modul în care acestea ar putea influența dezvoltarea și design-ul platformei noastre.

Un aspect tare important mai este impactul schimbărilor sociale și tehnologice asupra utilizării platformelor de mesagerie. Vom investiga cum factori precum evoluția comportamentului utilizatorilor sau schimbările tehnologice au modelat cerințele și așteptările acestora.

2.4 Analiza situației în domeniul TIC în Republica Moldova

Cum se știe în ultimul timp în Republica Moldova sa efectuat o creștere majoră în domeniul TIC , ceea ce reprezintă că populația este mai dotată cu diferite gadgeturi ce au ieșiri de conexiune la internet. Tot odată RM este printre primele 10 țări din lume cu privire la accesibilitatea și costul al accesului la internet Gigabit, cu un cost de aproximativ 15 euro pe lună , asta prezentând că Moldova are o perspectivă mare în domeniul TIC . Cum se știe orice aplicație are nevoie de acces la internet și de un dispozitiv, mai sunt și alte componente în Figura 2.2

2.5 Aspectele de securitate și confidențialitate

În urma studierii aplicațiilor de mesagerie au fost depistate momente de securitate cum ar fi :

- Autentificare în aplicație
- Criptarea datelor de transmitere

- Protecția datelor personale
- Gestiunea accesului si privilegiilor
- Monitorizarea si raportarea incidentelor de securitate
- Limitarea conectărilor de pe un singur dispozitiv
- Autentificarea prin 2 pași

Toate aceste aspecte enumerate fac parte dintr-o componentă importantă a oricărui produs nu doar celui de mesagerie, de aceea este important și necesar ca aspectele sa fie prezenta în aplicație. Clientul simplu accesând aplicația ar trebui să fie în siguranță din toate punctele, asta va duce la o eficiență mai mare și va duce la creșterea de clienți, un exemplu din viața actuală este telegram care are o securitate înaltă și efectuează criptarea datelor ceea ce duce la evitarea scurgerii datelor prin internet.

2. PARTEA APLICATIVĂ

3.1 Infrastructura tehnologica

După analiza pieții au fost identificate diverse soluții pentru realizarea unui portal de messengerie, iar pentru proiectul dat s-au aplicat tehnologiile prezentate în cadrul secțiunii teoretice.

Una din aceste tehnologii este Laravel, este o platformă open-source bazată pe PHP, specializată în crearea aplicațiilor web complexe. Acest framework ca concept de bază are model-view-controller (MVC) și se bazează pe arhitectura de Symfony, oferind o serie largă de caracteristici ce ușurează și structurează dezvoltarea aplicațiilor web.

Unele dintre avantajele acestei tehnologii sunt:

1. Sistem modular și gestionarea dependențelor(Composer). Având acest sistem ca bază, putem integra diferite module.
2. Suport pentru baze de date relaționale. Platforma dată permite suportul pentru conectarea la diverse tipuri de baze de date, ceea ce duce la lucrul cu datele într-un mod flexibil și eficient.
3. Interfață de linii de comandă (CLI) integrată. Prin intermediul CLI proprii Laravel facilitează efectuarea unor acțiuni diverse atât în cadrul aplicației, cât și în cadrul altor componente ale proiectului, ceea ce duce la o productivitate asupra dezvoltării proiectului.
4. Documentație și comunitate activă. Laravel beneficiază de o documentație largă și de o comunitate activă de dezvoltatori, ceea ce înseamnă că putem găsi rapid soluții și suport pentru remedierea problemelor în timpul dezvoltării.
5. Testarea automată. Frameworkul are deja integrat o gamă largă de testare automată a softului, ceea ce duce la crearea și rularea testelor unit, ca rezultat având o aplicație securizată de diferite tipuri de atacuri.

Prin utilizarea la Laravel în proiectul nostru ne asigurăm că platforma de mesagerie va avea o mentenanță pe un timp îndelungat, de o gamă largă de resurse și de o aplicație ușor de întreținut.

Laravel a fost ales ca framework de bază din compartimentul că are o comunitate largă, suport și mentenanță, actualizări de framework tabelul 3.1, tabelul dat demonstrează faptul că vor fi și alte versiuni mai performante.

Ca una din tehnologiile de utilizate a gestionării bazei de date a fost folosit Mysql.

Mysql este un limbaj de interogare structural pentru a gestiona și manipula datele în baza de date relaționale. Oferă suport pentru multiple sisteme de operare (windows, linux și macOS) ceea ce duce la o alegere bună pentru dezvoltarea unei aplicații. Tot odată oferă suport pentru limbajele de programare cum ar fi unele din cele mai populare PHP, Python, Java și multe altele.

Unele avantaje ale Mysql:

- Open-source
- Performanță ridicată
- Stabilitate
- Comunitate activă și suport extins
- Integrare

Node.js este un mediu de execuție JavaScript construit pe motorul V8 JavaScript dezvoltat de Google. Acesta este adesea utilizat în aplicațiile web pentru a construi servere de tipul "event-driven", care sunt eficiente și scalabile în gestionarea unui număr mare de conexiuni simultane.

În cadrul proiectului, Node.js este integrat cu scopurile:

- Utilizarea pentru sarcini asincrone și în timp real. De exemplu, poate fi utilizat în implementarea funcționalității de o mesagerie sau de notificări în browser în timp real în cadrul frameworkului Laravel.
- Gestionarea proceselor pe fundal și a sarcinilor în coadă. Node.js poate fi utilizat cu scopul de crearea serviciului de gestionare a sarcinilor în fundal sau în coadă (queue), un exemplu este de trimitere a e-mail sau a unui mesaj.
- Viteza de execuție a codului JavaScript: Datorită motorului V8 JavaScript, Node.js oferă o performanță excepțională în execuția codului JavaScript. Acest lucru poate fi benefic în cadrul aplicației Laravel pentru sarcini care necesită o procesare rapidă și eficientă a cererilor.
- Suport pentru programare asincronă și non-blocantă. Node.js este construit pe un model de programare asincronă și non-blocantă, ceea ce înseamnă că poate gestiona un număr mare de conexiuni simultane fără a bloca execuția codului. Acest lucru este util în cadrul aplicațiilor web care trebuie să gestioneze cereri concurente și interacțiuni în timp real.

- Actualizare și mentenanță. Tehnologia este utilizată în diferite domenii, de aceea are o comunitate largă, mentenanță și suport, actualizare de tehnologie tabelul 3.2, tabelul dat demonstrează faptul că vor fi și alte versiuni mai performante.

Prin integrarea acestei tehnologii în aplicația de mesagerie ne permite o flexibilitate mai largă și un număr larg de conexiuni concomitente.

Defapt Node.js poate fi utilizat în diferite aplicații cu diferit nivel de greutate, având diferite soluții pentru orice problemă

Composer este un manager de dependențe pentru limbajul de programare PHP, utilizat pentru gestionarea bibliotecilor și a pachetelor necesare în cadrul proiectelor PHP. În contextul dezvoltării unei aplicații de chat în Laravel, Composer poate fi folosit pentru a instala și gestiona diverse pachete și biblioteci necesare pentru funcționarea aplicației.

În cadrul Laravel composer are mai multe oportunități și beneficii pentru dezvoltarea aplicațiilor PHP cum ar fi

- Gestionarea eficientă a dependențelor: Composer simplifică gestionarea dependențelor externe ale proiectului tău PHP. Poți instala și actualiza rapid biblioteci și pachete externe necesare în cadrul proiectului tău, folosind o singură comandă.
- Reutilizarea codului: Prin intermediul Composer, poți accesa o gamă largă de pachete și biblioteci dezvoltate de comunitatea PHP. Aceste pachete oferă funcționalități diverse, de la abstrageri de baze de date până la integrarea cu servicii terțe sau framework-uri de dezvoltare.
- Creșterea eficienței dezvoltării: Utilizarea Composer elimină necesitatea de a scrie și de a menține manual cod pentru funcționalități comune, cum ar fi autoloading-ul claselor și gestionarea dependențelor. Aceasta permite dezvoltatorilor să se concentreze mai mult pe dezvoltarea funcționalităților unice ale aplicației și să economisească timp și efort.
- Flexibilitate în dezvoltare: Composer oferă flexibilitate în alegerea și utilizarea bibliotecilor și pachetelor externe în cadrul proiectului tău. Poți selecta cele mai potrivite pachete pentru cerințele tale specifice și poți personaliza configurația lor în funcție de nevoile aplicației.
- Gestionarea versiunilor și actualizările: Composer gestionează automat versiunile și actualizările bibliotecilor și pachetelor instalate în cadrul proiectului tău. Astfel, poți fi

sigur că folosești întotdeauna cele mai recente versiuni și că beneficiezi de corecții de erori și de actualizări de securitate.

Când s-a creat composer de bază a fost luat ideile de la Node.js, adică composer are aceleași idei ca și Node.js.

Pusher este un serviciu care permite comunicarea în timp real între server și clienți în aplicațiile web și mobile. Este utilizat pentru a implementa funcționalități de notificare în timp real și actualizări dinamice ale paginilor.

Folosind WebSocket-uri, Pusher facilitează comunicarea bidirecțională între server și client, permițând serverului să trimită actualizări către clienți în timp real, fără a necesita o reîncărcare a paginii sau o interacțiune suplimentară din partea utilizatorilor.

În Laravel, integrarea cu Pusher este simplificată prin intermediul Laravel Echo și a bibliotecii Pusher PHP. Laravel Echo este o librărie JavaScript care permite abonarea la evenimente în timp real și manipularea acestora în interfața utilizatorului, în timp ce biblioteca Pusher PHP este utilizată pentru a trimite evenimente către Pusher.

Prin integrarea Pusher în aplicația Laravel, creează funcționalitate interactivă și dinamică, precum ar fi notificările, statutul online sau off-line, când a fost ultima dată online în timp real.

Tot odată Pusher permite analizarea requesturilor cum ar fi numărul de utilizator conectați și numărul de trimiteri a mesajelor.

3.2 Interacționarea

Aplicația este concepută pentru a fi intuitivă și ușor de utilizat, astfel încât utilizatorii să se simtă confortabil în gestionarea și navigarea în interiorul ei.

Pagina principală a aplicației este pagina de autentificare și înregistrare. După autentificare, utilizatorii sunt redirecționați către pagina principală a aplicației de mesagerie.

În figura 3.3 este prezentată metoda de trimitere a unui mesaj între 2 persoane, iar mai jos este descrierea procesului.

- User 1. Utilizatorul accesează interfața aplicației și navighează către secțiunea de chat. În interfață, utilizatorul poate selecta un utilizator cu care dorește să comunice sau poate deschide o conversație nouă.

- Trimiterea cererii HTTP către server. Atunci când utilizatorul trimite un mesaj, clientul (browserul) va face o cerere HTTP către server pentru a trimite mesajul.
- Recepția cererii pe server (backend). Serverul Laravel primește cererea HTTP și o rutează către controlerul corespunzător pentru gestionarea mesajului.
- Validarea datelor de intrare. În controler, datele primite (cum ar fi conținutul mesajului, id-ul utilizatorului de la care provine și id-ul utilizatorului către care se trimite mesajul) sunt validate pentru a se asigura că sunt conforme cu cerințele (de exemplu, mesajul să nu fie gol).
- Procesarea și stocarea mesajului în baza de date. După validare, mesajul este procesat și stocat în baza de date. Se poate adăuga și un marcaj pentru a indica dacă mesajul a fost citit sau nu de către destinatar.
- Emiterea evenimentului de trimitere a mesajului către Pusher. După ce mesajul este stocat cu succes în baza de date, serverul Laravel emite un eveniment către Pusher, pentru a notifica clientul că a fost trimis un nou mesaj.
- Recepția evenimentului de către client (frontend). Clientul (browserul) care a trimis cererea primește evenimentul emis de server prin intermediul Pusher. JavaScript-ul din interfața aplicației ascultă evenimentul și reacționează în consecință, afișând mesajul nou în interfață.
- Actualizarea interfeței utilizatorului. Interfața utilizatorului este actualizată pentru a afișa noul mesaj în conversația corespunzătoare.
- User 2. Utilizatorul primește notificare în interfață că mesajul a fost trimis cu succes și că are un mesaj în așteptare.

După ce user 2 vizualizează mesajul, acesta va fi marcat ca fiind citit pentru user 1. Astfel, utilizatorii pot urmări starea și evoluția conversațiilor lor în timp real.

CONCLUZII

Crearea unei aplicații de mesagerie interactive este un proces complex și captivant, care implică multiple etape și decizii pe parcursul întregului proiect. În acest proces, am îmbinat cunoștințele teoretice cu abilitățile practice pentru a dezvolta o aplicație care să îndeplinească cerințele și să ofere o experiență plăcută utilizatorilor, am aflat câte ceva nou pentru mine iar ulterior voi utiliza experiența acumulată și în alte proiecte.

Prima etapă a fost planificarea și definirea cerințelor proiectului. Am identificat obiectivele și funcționalitățile principale ale aplicației, precum și tehnologiile și instrumentele potrivite pentru implementare. Am decis să folosim Laravel ca framework PHP principal, datorită flexibilității, securității și eficienței sale în dezvoltarea aplicațiilor web complexe.

Următoarea etapă a fost proiectarea și implementarea arhitecturii aplicației. Am definit structura bazei de date și modelele de date necesare pentru stocarea utilizatorilor, conversațiilor și mesajelor. Am folosit migrări și seminte în Laravel pentru a crea și popula baza de date în mod eficient.

Odată cu fundația pusă, am trecut la dezvoltarea funcționalităților cheie ale aplicației. Am implementat sistemul de autentificare și înregistrare pentru utilizatori, folosind funcționalitățile native ale framework-ului Laravel. Am dezvoltat și interfața de utilizator pentru pagina principală a aplicației, care include lista de conversații și interacțiunea cu mesajele.

Un aspect crucial al aplicației a fost implementarea comunicării în timp real. Am integrat Pusher în aplicație pentru a permite trimiterea și recepționarea mesajelor în timp real între utilizatori. Folosind Laravel Echo și JavaScript, am reușit să sincronizăm conversațiile în timp real și să notificăm utilizatorii despre mesajele noi.

Pe măsură ce dezvoltarea avansa, am continuat să testăm și să optimizăm aplicația pentru performanță și securitate. Am efectuat teste unitare și de integrare pentru a ne asigura că aplicația funcționează corect și că datele utilizatorilor sunt protejate.

În final, am ajuns la etapa de implementare și lansare a aplicației. Am configurat serverul și am pus în aplicare măsurile de securitate necesare pentru a asigura o experiență sigură și fiabilă pentru utilizatori. Apoi, am lansat aplicația și am început să colectăm feedback de la utilizatori pentru a identifica posibile îmbunătățiri și actualizări viitoare.

În concluzie, procesul de creare a acestei aplicații de mesagerie interactive a fost unul provocator și încântător. Am reușit să combinăm cunoștințele noastre teoretice cu abilitățile practice pentru

a dezvolta o aplicație care să îndeplinească cerințele și să ofere o experiență plăcută și utilă utilizatorilor. Cu fiecare etapă a procesului, am învățat noi tehnologii și tehnici, consolidându-ne cunoștințele și îmbunătățindu-ne abilitățile în dezvoltarea de aplicații web.

REFERINȚE BIBLIOGRAFICE

https://www.egov.md/sites/default/files/document/attachments/strategia_de_transformare_digitala_2023-2030.pdf

<https://en.wikipedia.org/wiki/Laravel>

<https://ro.wikipedia.org/wiki/MySQL>

<https://en.wikipedia.org/wiki/Node.js>

<https://getcomposer.org/doc/00-intro.md>

ANEXE

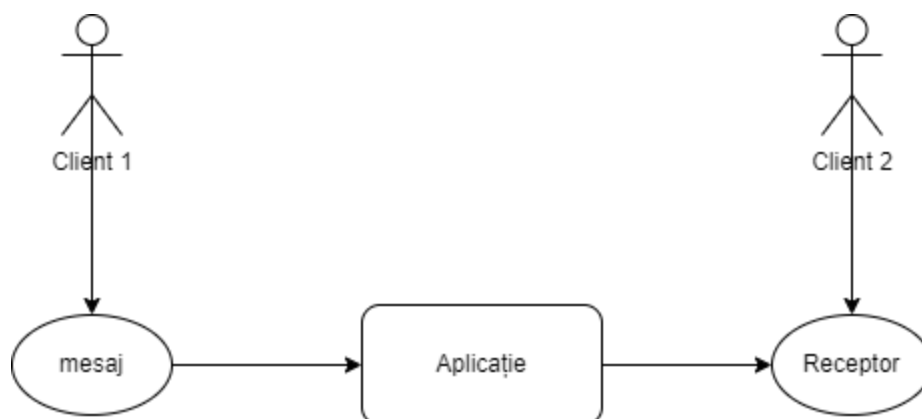


Figura 2.1 Metoda simplificată de trimitere a mesajului

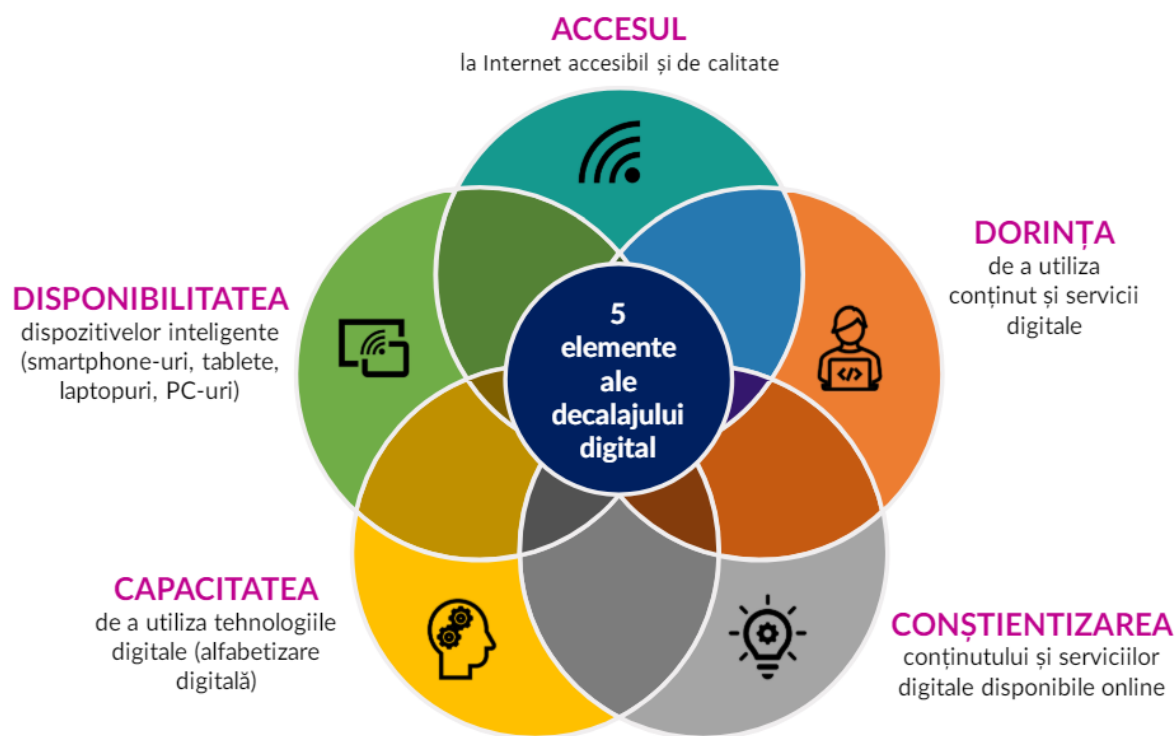


Figura 2.2 5 elemente ale decalajului digital

Tabelul 3.1 Actualizările Laravel

Versiune	Data de lansare	versiunea PHP
1.0	Iunie 11	
2.0	sept 11	
3.0	22 feb 12	
3.1	27 mar 12	
3.2	22 mai 12	
4.0	28 mai 13	$\geq 5,3,0$
4.1	12 dec 13	$\geq 5,3,0$
4.2	01 iun 14	$\geq 5,4,0$
5.0	04 feb 15	$\geq 5,4,0$
5.1 LTS	09 iun 15	$\geq 5,5,9$
5.2	21 dec 15	$\geq 5,5,9$
5.3	23 aug 16	$\geq 5,6,4$
5.4	24 ian 17	$\geq 5,6,4$
5.5 LTS	30 aug 17	$\geq 7,0,0$
5.6	07 feb 18	$\geq 7.1.3$
5.7	04 sept 18	$\geq 7.1.3$
5.8	26 feb 19	$\geq 7.1.3$
6 LTS	03 sept 19	7,2 – 8,0
7	3 martie 20	7,2 – 8,0
8	08 sept 20	7,3 – 8,1
9	08 feb 20	8,0 – 8,2
10	14 feb 23	8,1 - 8,3
11	12 mar 24	$\geq 8,2$
12	T1 2025	

■ Versiune veche
 ■ Versiune mai veche, încă păstrată
 ■ **Ultima versiune**
■ Cea mai recentă versiune de previzualizare
 ■ Lansare viitoare

Tabelul 3.2 Actualizările Node.Js

Release	Status	Release date	Maintenance end
0.10.x	End-of-Life	2013-03-11	2016-10-31
0.12.x	End-of-Life	2015-02-06	2016-12-31
4.x	End-of-Life	2015-09-08	2018-04-30
5.x	End-of-Life	2015-10-29	2016-06-30
6.x	End-of-Life	2016-04-26	2019-04-30
7.x	End-of-Life	2016-10-25	2017-06-30
8.x	End-of-Life	2017-05-30	2019-12-31
9.x	End-of-Life	2017-10-01	2018-06-30
10.x	End-of-Life	2018-04-24	2021-04-30
11.x	End-of-Life	2018-10-23	2019-06-01
12.x	End-of-Life	2019-04-23	2022-04-30
13.x	End-of-Life	2019-10-22	2020-06-01
14.x	End-of-Life	2020-04-21	2023-04-30
15.x	End-of-Life	2020-10-20	2021-06-01
16.x	End-of-Life	2021-04-20	2023-09-11
17.x	End-of-Life	2021-10-19	2022-06-01
18.x	Maintenance LTS	2022-04-19	2025-04-30
19.x	End-of-Life	2022-10-18	2023-06-01
20.x	Active LTS	2023-04-18	2026-04-30
21.x	Maintenance	2023-10-17	2024-06-01
22.x	Current	2024-04-24	2027-04-30
23.x	Planned	2024-10-14	2025-06-01
24.x	Planned	2025-04-22	2028-04-30

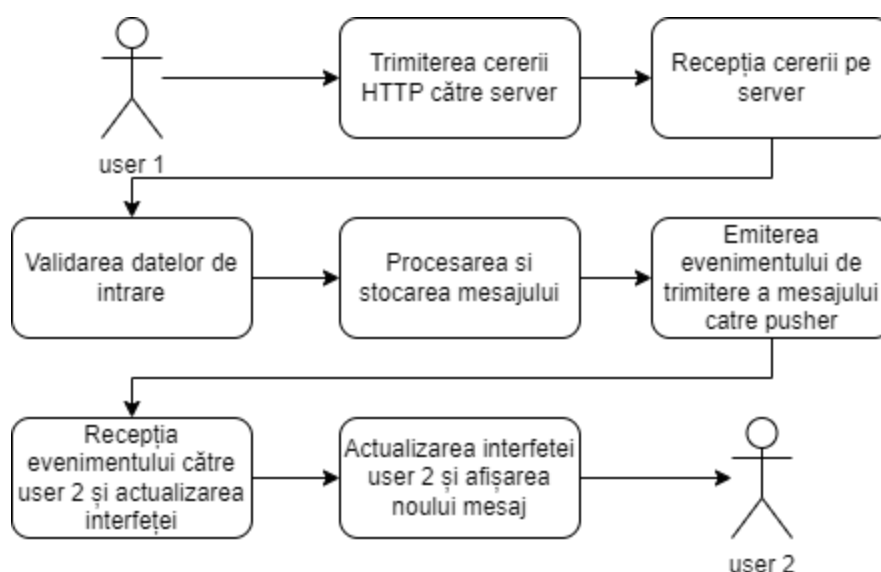


Figura 3.3 Metoda de trimitere a mesajelor

Tabela 3.4 Tabelele bazei de date

Tabelele
ch_favorites
ch_messages
failed_jobs
migrations
password_reset_tokens
personal_access_tokens
sessions
users

Tabelul 3.5 descrierea tabelelor de baza

ch_favorites

Field	Type	Null	Key	Default	Extra
id	Char(36)	No	Primary	Null	
user_id	Bigint	No		Null	
favorite_id	Bigint	No		Null	
created_at	Timestamp	Yes		Null	
updated_at	timestamp	Yes		Null	

ch_messages

Field	Type	Null	Key	Default	Extra
id	Char(36)	No	Primary	Null	
from_id	Bigint	No		Null	
to_id	Bigint	No		Null	
body	Varchar(5000)	Yes		Null	
attachment	Varchar(255)	Yes		Null	
seen	Tinyint(1)	No		0	
created_at	Timestamp	Yes		Null	
updated_at	timestamp	Yes		Null	