UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Interfață grafică pentru AIM-RL

propusă de

Răzvan-Petru Leonte

Sesiunea: Iulie, 2024

Coordonator științific

Lect. Dr. Pistol Ionuț

UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

Interfață grafică pentru AIM-RL

Răzvan-Petru Leonte

Sesiunea: Iulie, 2024

Coordonator științific

Lect. Dr. Pistol Ionuț

Avizat,

Îndrumător lucrare de licență,

Lect. Dr. Pistol Ionuț.

Data: 24.06.2024 Semnătura: .....

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Leonte Răzvan-Petru** domiciliat în **România, jud. Iași, mun. Iași, Aleea Basarabi, nr. 8, bl. U3bis, sc. B, et. 4, ap. 17**, născut la data de **24 august 2002**, identificat prin CNP **5020824226749**, absolvent al Universității "Alexandru-Ioan Cuza" din Iași, **Facultatea de informatică** specializarea **informatică**, promoția 2024, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Interfață grafică pentru AIM-RL** elaborată sub îndrumarea domnului **Lect. Dr. Pistol Ionuț**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data: 24.06.2024 Semnătura: .....

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Interfață grafică pentru AIM-RL**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test, etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea ”Alexandru-Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Răzvan-Petru Leonte**

Data: 24.06.2024 Semnătura: .....

Cuprins

Introducere

În peisajul digital contemporan care se află în continuă evoluție, inteligența artificială și învățarea automată au devenit instrumente esențiale în diverse domenii, de la finanțe și medicină, până la literatură și artă. Aceste tehnologii nu doar că metamorfozează modurile de funcționare ale multor industrii, ci și creează noi oportunități atât pentru dezvoltatori, cât și pentru consumatori, schimbând treptat experiențele umane.

Un subdomeniu important derivat din simbioza tehnologiilor menționate anterior (inteligența artificială și învățarea automată) este ”Reinforcement Learning”-ul, care se concentrează pe antrenarea unor agenți, aceștia încercând la fiecare pas diferite acțiuni și comportamente. Agenții învață din rezultatele obținute și își adaptează următorii pași în funcție de ceea ce învață, cu scopul de a ajunge într-un punct extrem sau de a maximiza o recompensă cumulativă pe termen lung.

Atenția sporită asupra ”Reinforcement Learning”-ului este motivată de potențialul său de a aduce inovații semnificative în diferite domenii. ”Reinforcement Learning”-ul nu este doar un concept teoretic, ci are aplicabilități distince în practica demonstrată. În domeniul jocurilor video, agenții specifici au învățat nu numai să joace, ci chiar să depășească performanțele umane, demonstrând capacitatea lor de a lua decizii strategice complexe. În robotică, ”Reinforcement Learning”-ul este utilizat pentru a învăța roboții să se adapteze și să navigheze în medii necunoscute. Acești roboți pot fi antrenați să efectueze chiar și sarcini complexe, cum ar fi manipularea obiectelor delicate, fără a le pune în pericol integritatea.

Un exemplu important de framework utilizat în ”Reinforcement Learning” este ”OpenAI Gym” (rebotezat ”Gymnasium”), o platformă populară pentru dezvoltarea și evaluarea algoritmilor de ”Reinforcement Learning”. ”Gymnasium” oferă un set standardizat de medii de lucru, unde agenții pot fi antrenați și testați, permițând dezvoltatorilor să compare performanțele algoritmilor într-un mod consistent și replicabil. Popularitatea acestui framework a crescut exponențial în ultimii ani, devenind un in-

strument important în domeniul ”Reinforcement Learning”.

Statistici recente arată că interesul pentru ”Reinforcement Learning”, dar și utilizarea framework-urilor precum ”Gymnasium” au crescut semnificativ. Numărul de articole științifice orientate pe utilizarea ”Reinforcement Learning”-ului a crescut de peste patru ori în ultimii cinci ani, iar platformele de dezvoltare, precum Gymnasium, au întâmpinat o creștere substanțială a numărului de utilizatori activi. Aceste statistici demonstrează nu doar relevanța academică, ci și aplicabilitatea practică a ”Reinforcement Learning”-ului în rezolvarea de probleme complexe din domenii diverse.

Toate aceste realizări accentuează potențialul enorm al ”Reinforcement Learning”-ului în dezvoltarea de soluții care să balanseze perfect inovarea și eficiența, acestea fiind capabile să transforme majoritatea domeniilor, având fezabilitatea să creeze produse și să presteze servicii mai avansate pentru utilizatori. Interesul personal pentru acest domeniu este, astfel, bazat pe dorința de a contribui la această revoluție tehnologică, explorând și dezvoltând noi metode care să îmbunătățească viața de cotidiană.

Motivație

În ciuda avansurilor tehnologice semnificative realizate în ultimii ani, resursele disponibile sunt adesea complexe și dificil de utilizat, în special pentru începători. Astfel, dezvoltarea unei platforme interactive și ușor de folosit răspunde unor nevoi educaționale și de cercetare esențiale.

Proiectul GIRL (Graphical Interface for Reinforcement Learning) își propune să ofere utilizatorilor o platformă intuitivă și accesibilă pentru a realiza diferite experimente, satisfăcând astfel diferite curiozități științifice și ajutând utilizatorul să își consolideze cunoștințele în domeniu.

AIM-RL, framework-ul utilizat în cadrul proiectului GIRL, oferă o abordare modulară și flexibilă pentru proiectarea și testarea algoritmilor de ”Reinforcement Learning”. Acesta permite utilizatorilor să își creeze propriile medii de învățare și să ajusteze parametrii experimentelor, oferind astfel un spațiu vast pentru inovație și explorare.

În contextul actual, unde ”Reinforcement Learning”-ul devine tot mai integrat în soluțiile tehnologice de zi cu zi, capacitatea de a înțelege și de a aplica aceste tehnologii reprezintă un avantaj competitiv semnificativ. Proiectul GIRL nu doar că inițiază și educă utilizatorii în folosirea tehnicilor de bază ale ”Reinforcement Learning”-ului, dar îi și încurajează să exploreze noi direcții de cercetare și să contribuie la dezvoltarea continuă a acestui domeniu.

Astfel, prin intermediul aplicației GIRL, utilizatorii nu numai că își pot dezvolta competențele tehnice și teoretice în ”Reinforcement Learning”, dar pot aduce și contribuții valoroase comunității științifice și tehnologice, eliminând necesitatea utilizatorului să aibă cunoștințe de dezvoltare folosind limbaje de programare.

Capitolul 1

Resurse și tehnologii utilizate

Metodologia adoptată pentru dezvoltarea unei aplicații destinate experimentelor de ”Reinforcement Learning” trebuie să fie bine aleasă și perfect aplicată. Având în vedere complexitatea și cerințele tehnice ale proiectului, au fost utilizate tehnologii de ultimă generație pentru a asigura o implementare modernă și cât mai scalabilă. Procesul de dezvoltare a fost structurat în mod clar și sistematic încă de la început, pornind de la planificarea abordării și continuând cu execuția detaliată a fiecărui element din strategia inițială, astfel încât să obținem un produs final de înaltă calitate.

1.1 Tehnici folosite

Crearea unui proiect dedicat experimentelor de ”Reinforcement Learning” implică o planificare riguroasă și implementarea unor tehnologii moderne și eficiente. Proiectul va folosi framework-ul Angular 17 pentru dezvoltarea părții de frontend, Java Spring Boot pentru realizarea părții de backend și H2 pentru baza de date.

Înainte de începerea efectivă a dezvoltării proiectului, a fost realizată o diagramă UML utilizând instrumentul online ”draw.io” pentru a planifica arhitectura și fluxurile de lucru ale întregii aplicații. Diagrama UML a fost esențială pentru a vizualiza structura și relațiile dintre diferitele componente ale sistemului. Această planificare detaliată a permis o înțelegere clară a funcționalităților și a interdependențelor dintre elementele frontend, backend, database și AIM-RL, asigurând astfel o integrare coerentă și eficientă a tuturor componentelor.

Taskurile pentru dezvoltarea proiectului au fost create pe platforma Trello, având la bază elementele din diagrama UML. Fiecare componentă și relație descrisă în dia-

gramă a fost transformată într-un task specific, detaliat și atribuit unei etape de dezvoltare. Taskurile au fost organizate în liste corespunzătoare fiecărei faze de dezvoltare: To Do, In Progress, Testing și Done. Această abordare metodică a permis gestionarea eficientă a timpului și resurselor, oferind o vizibilitate clară asupra progresului proiectului.

Angular 17 este o alegere excelentă pentru dezvoltarea frontend-ului datorită flexibilității și performanțelor sale. Acest framework permite crearea unor interfețe de utilizator dinamice și responsive, esențiale pentru vizualizarea datelor complexe generate de experimentele de ”Reinforcement Learning”. Utilizatorii vor putea accesa rezultatele experimentelor într-un mod intuitiv și interactiv, facilitând astfel analiza și înțelegerea în adâncime a rezultatelor.

Pentru backend, Java Spring Boot oferă un mediu robust și scalabil, ideal pentru dezvoltarea de API-uri și gestionarea logicii aplicației. Integrarea cu baza de date H2, o bază de date în memorie ușor de utilizat, permite stocarea și accesarea rapidă a datelor experimentelor, dar și cele a utilizatorilor. Această combinație asigură o performanță optimă și o dezvoltare rapidă, fiind perfectă pentru un proiect de cercetare și dezvoltare.

GitHub este ales pentru controlul versiunilor și colaborarea între viitori dezvoltatori. Prin utilizarea Actions, GitHub permite automatizarea testelor și verificărilor de calitate a codului, asigurând astfel stabilitatea și fiabilitatea proiectului. Contribuitorii vor beneficia de un proces clar și eficient pentru propunerea de modificări și adăugarea de noi funcționalități.

Proiectul va fi structurat astfel încât să fie accesibil și ușor de utilizat de către toți participanții. Documentația detaliată și un sistem de feedback vor încuraja implicarea activă a comunității și vor contribui la succesul și scalabilitatea pe termen lung al proiectului.

1.2 Medii adoptate

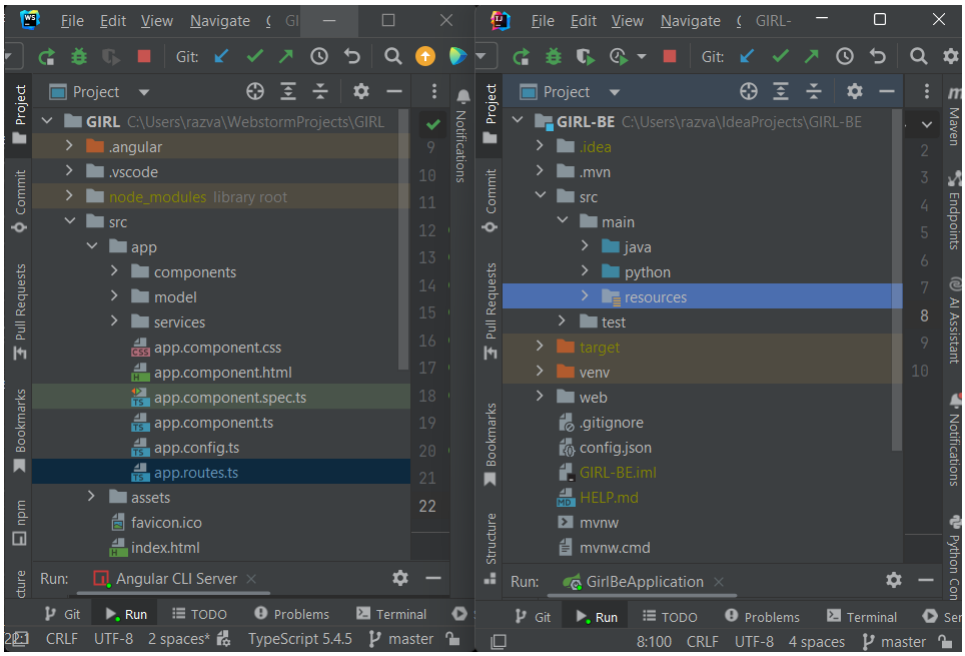
Setarea mediului de lucru pentru dezvoltarea aplicației a fost realizată cu atenție pentru a optimiza productivitatea și a maximiza timpul economisit în timpul dezvoltării platformei. S-a urmărit obținerea celor mai moderne medii de lucru, produsele celor de la ”JetBrains” fiind cele mai lăudate din industrie.

Pentru partea de frontend, am ales să folosesc ”WebStorm”, un mediu de lucru

renumit pentru suportul său avansat în dezvoltarea aplicațiilor de tip ”web”. ”WebStorm” oferă suport excelent pentru dezvoltare în HTML, CSS, și TypeScript, esențial pentru proiectele care utilizează framewrok-ul Angular 17. Funcționalitățile sale de auto-completare și ”IntelliSense” fac codarea mai rapidă și mai precisă, reducând semnificativ timpul de dezvoltare și numărul de potențiale erori apărute. De asemenea, integrarea sa cu diverse framework-uri populare, precum Angular, React sau Vue, asigură o dezvoltare și mai eficientă.

Pentru partea de backend, am optat pentru ”IntelliJ”, un mediu de lucru extrem de puternic și versatil. ”IntelliJ” oferă suport pentru Java, dar și pentru alte limbaje de programare, cum ar fi Python, facilitând astfel dezvoltarea aplicațiilor complexe de tip ”server”. Caracteristicile avansate de debugging, refactoring și integrare cu diverse unelte de version control si de build, cum ar fi Maven, Gradle și Git, contribuie la menținerea unui flux de lucru eficient și rapid.

Prin utilizarea ”WebStorm” și ”IntelliJ”, am creat un mediu de lucru modern și prietenos pentru un dezvoltator, productivitatea fiind cu mult sporită. Această configurare permite abordarea cu succes a cerințele complexe întâlnite pe parcursul proiectului și mi-a permis facilitarea unei soluții de înaltă calitate într-un timp optim.



1.3 Platforma AIM-RL

AIM-RL reprezintă un framework nou creat, având scopul de a ușura implementarea de noi probleme și modele asociate, dar și pentru a folosi ”Reinforcement Learning”-ul în găsirea de soluții pentru aceste probleme. AIM-RL este conceput pentru a fi cât mai clar și flexibil posibil, optimizând și simplificând implementarea principală a ”Reinforcement Learning”-ului, astfel fiindu-i permis utilizatorului să descrie probleme, să furnizeze modele și să personalizeze execuția.

AIM-RL este un pachet Python care poate fi compilat și instalat prin intermediul comenzii ”pip”. Scopul principal este de a oferi o implementare parametrică pentru ”Reinforcement Learning”, ușor de utilizat pentru diverse probleme și modele. Componenta principală este modulul ”qlearning”, care include funcția qlearning(instance, noofepochs, epsilon, alpha, discount, decay, limit, verbose=False, discountoptimisation=True), ea returnând un tuplu (Q, results, solutions, rate).

Funcția ”qlearning” poate fi apelată cu parametrii necesari specificați manual sau definiți în cadrul unui fișier JSON. Parametrii acceptați includ epsilon, decay, alpha, discount, epochs, limit, runs, model, generategraph și instances, modulul qlearning fiind capabil să ofere și o funcție pentru generarea de grafice (folosind biblioteca ”matplotlib”).

Rezultatele experimentelor efectuate cu AIM-RL sunt salvate într-un fișier CSV unic care include diverse valori, precum și graficul generat, modelul utilizat, instanța folosită, parametrii RL, identificatorul rundei, durata rundei, procentajul de epoci de succes și dimensiunea tabelului Q.

Un element central al proiectului ”GIRL” este utilizarea framework-ului AIM-RL, realizat în limbajul de dezvoltare Python, pentru realizarea efectivă a experimentelor de ”Reinforcement Learning”. AIM-RL primește parametrii experimentali de la utilizator prin intermediul interfeței Angular și al server-ului Java Spring Boot. După efec-

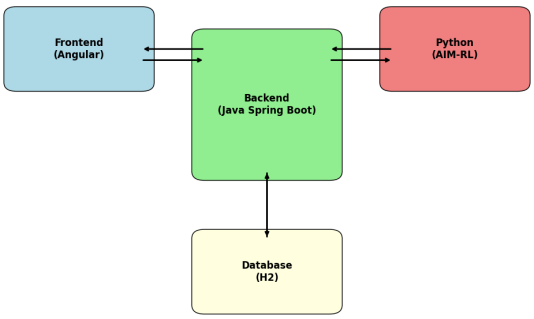
tuarea experimentului, AIM-RL trimite rezultatele înapoi către server-ul Java Spring Boot, care le transmite la rândul lui mai departe către Angular pentru vizualizare. Acest flux asigură o integrare perfectă între părțile de frontend, backend, dar și procesul de calcul propriu-zis, oferind utilizatorilor o experiență coerentă și eficientă.

Capitolul 2

Graphical Interface for Reinforcement Learning

2.1 Funcționalități principale

”GIRL” este o platformă dezvoltată în Angular pentru frontend și în Java Spring Boot pentru backend. Serverul interacționează cu un program Python ce implementează framework-ul AIM-RL de ”Reinforcement Learning” și utilizează o bază de date de tip H2. Scopul principal al aplicației este punerea la dispoziție pentru utilizatori a posibilității de a crea și manipula diferite experimente de ”Reinforcement Learning”.



Frontend-ul Angular comunică cu backendul Spring Boot prin API-uri pentru autentificare, gestionarea utilizatorilor, și manipularea experimentelor. Backendul Spring Boot integrează și coordonează execuția algoritmilor de ”Reinforcement Learning” din

framework-ul AIM-RL. Backendul Spring Boot gestionează interacțiunile cu baza de date H2 pentru manipularea datelor utilizatorilor, dar și a experimentelor acestora.

În primă fază, utilizatorii au posibilitatea de a se autentifica sau de a-și crea un cont nou. Odată intrat în cont, un utilizator poate accesa secțiunea ”Home” pentru a citi informații despre dezvoltator sau secțiunea ”Documentation”, utilizatorii găsind aici detalii despre framework-ul AIM-RL. Secțiunea dedicată profilul utilizatorului oferă informațiile esențiale contului, inclusiv rangul de profesionalism, numărul de experimente completate sau în curs de desfășurare. Utilizatorii au posibilitatea de a-și edita informațiile, chiar și poza de profil, dar și de a se deloga sau a-și șterge contul. În secțiunea ”Dashboard”, utilizatorii pot crea noi experimente, dar și să le acceseze pe cele anterioare.

2.2 Frontend

TypeScript (TS) și Angular 17 sunt esențiale pentru construirea de aplicații web interactive și performante. TypeScript, o extensie a JavaScript-ului, adresează multe dintre limitările acestuia, oferind tipuri statice, acestea eficientizând depistarea erorilor. Această caracteristică face ca scrierea codului să fie mai precisă și mai sigură, reducând astfel timpul necesar pentru înțelegere, dezvoltare și testare.

Alegerea unui framework potrivit pentru dezvoltarea aplicației este crucială, deoarece fiecare framework vine cu avantaje și dezavantaje unice. Pentru acest proiect, am optat pentru Angular 17+ datorită stabilității sale și a ecosistemului bine dezvoltat. Angular este cunoscut pentru arhitectura sa robustă, care facilitează gestionarea aplicațiilor de mari dimensiuni. Un alt factor decisiv a fost popularitatea sa în comunitatea de dezvoltatori, ceea ce mărește mulțimea de idei și probleme deja rezolvate de către alți dezvoltatori.

Unul dintre principalele avantaje ale TypeScript este capacitatea sa de a preveni erorile comune prin verificarea tipurilor. Acest lucru este deosebit de important în proiectele mari, unde consistența și claritatea codului sunt esențiale. În plus, Angular 17+ oferă o serie de instrumente și funcționalități integrate. Aceste caracteristici permit dezvoltarea rapidă și eficientă a aplicațiilor, asigurând în același timp performanțe optime.

Prin alegerile TypeScript, respectiv Angular 17, acest proiect beneficiază de un mediu de dezvoltare modern și eficient, care nu doar că îmbunătățește productivitatea,

dar și asigură o bază solidă pentru scalabilitate și întreținere pe termen lung.

2.2.1 Componente

Partea de frontend a acestui proiect este dezvoltată utilizând framework-ul Angular 17 (specifică acestui framework este utilizarea de componente) și este structurat în mai multe componente specifice esențiale, fiecare având un rol în funcționarea generală a aplicației. Această arhitectură modulară permite o dezvoltare scalabilă, dar și o întreținere facilă.

Main

Această componentă este coloana vertebrală a întregii interfețe, oferind structura principală și fluxul de navigație al utilizatorului. Prin integrarea elementelor esențiale precum ”app-nav” și ”router-outlet”, această componentă facilitează o experiență de utilizare fluidă și intuitivă.

”App-nav” reprezintă bara de navigare vizibilă în partea laterală a interfeței. Aceasta conține meniul principal, dar și unica modalitate care permite utilizatorilor să acceseze rapid celelalte componente, dar și funcționalități ale aplicației. App-nav este proiectată pentru a oferi o utilizare clară și eficientă, ajutând clienții să se orienteze rapid în aplicație și să ajungă la componenta dorită.

În centrul acestei ”coloane vertebrale” se află ”router-outlet”, un element Angular 17+ fundamental care gestionează încărcarea dinamică a altor componente în funcție de ruta specificată de utilizator. Acesta funcționează ca o ”poartă” prin care sunt încărcate și afișate componentele corespunzătoare pentru fiecare pagină sau secțiune a aplicației. Prin utilizarea acestui ”router-outlet”, aplicația devine mult mai dinamică, permițând o navigare fluentă între diferitele părți ale platformei.

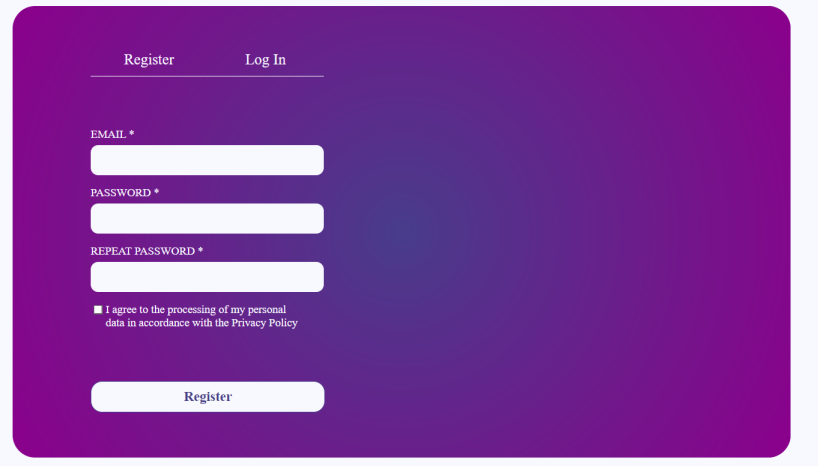
Autentificare

Componenta de autentificare reprezintă primul punct de contact al utilizatorilor cu aplicația, fiind esențială pentru gestionarea autentificării și securității în cadrul aplicației. Aceasta include două funcționalități principale: crearea unui cont și accesarea unui cont deja existent.

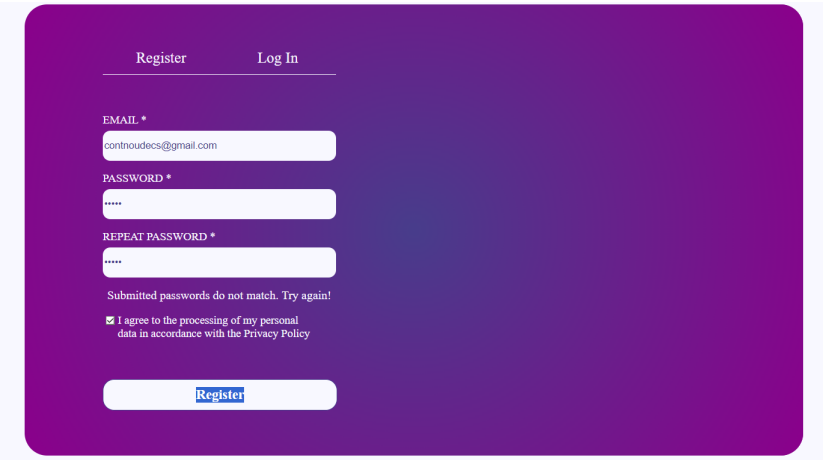
Un aspect important al acestei componente este alegerea modalității de autentificare dorite printr-un simplu ”click” pe optiunea dorită. Această funcționalitate oferă

utilizatorilor flexibilitatea de a naviga rapid între crearea unui cont nou și accesarea contului existent.

Pentru crearea unui cont, utilizatorii completează un formular care solicită introducerea unui email unic și a unei parole, care trebuie apoi repetată pentru a fi confirmată. Acest proces este crucial pentru securitatea aplicației, asigurându-se că fiecare cont este asociat cu un email valid și cu o parolă corect confirmată.

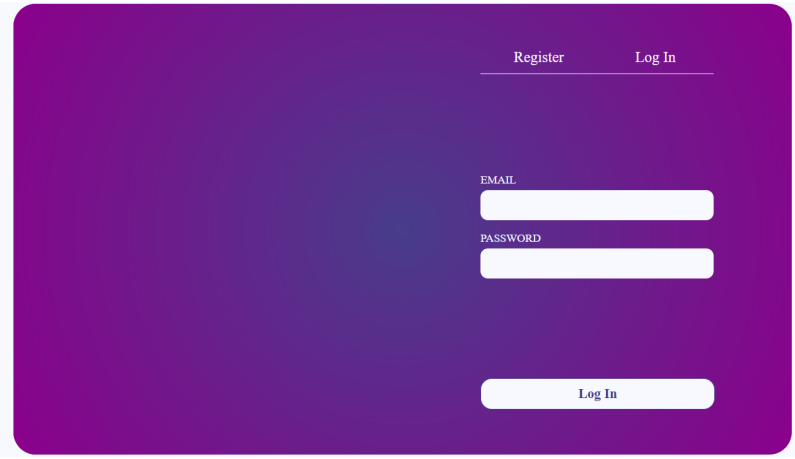


În cazul în care email-ul este deja înregistrat sau parola nu este confirmată corect, utilizatorii vor primi mesaje de eroare distincte, care le indică ce trebuie corectat pentru a reuși să își creeze contul.



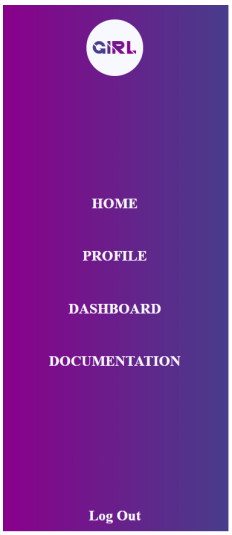
Pe de altă parte, funcția de accesare a unui cont deja existent permite utilizatorilor să se autentifice prin introducerea email-ul și a parolei corespondente contului care

este dorit să fie accesat. Aceste informații sunt verificate în mod securizat prin comunicarea cu backend-ul aplicației, care gestionează autentificarea și autorizarea accesului utilizatorului la contul dorit.



Navigația

Componenta de navigație este un element crucial al interfeței aplicației, fiind amplasată strategic în partea stângă a ecranului pentru a oferi utilizatorilor acces rapid și intuitiv la funcționalitățile principale ale aplicației. Acestea sunt formate din meniul platformei, care include link-uri către componentele esențiale, acestea fiind ”Home”, ”Documentation”, ”Profile” și ”Dashboard”.



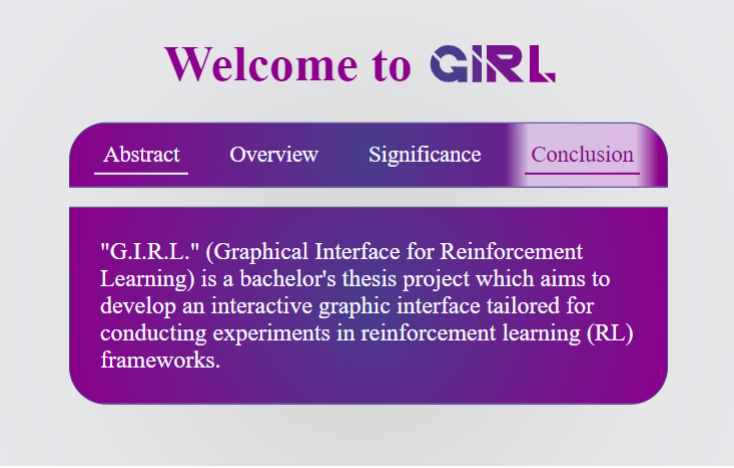
Meniul din componenta de navigație este proiectat pentru a fi vizibil și accesibil în mod constant pe parcursul navigării utilizatorului prin aplicație, oferind o navigare ușor de înțeles. Fiecare secțiune din meniu este configurată să redirecționeze utilizatorii către paginile corespunzătoare.

În plus față de navigarea între secțiuni, componenta de navigație include și un buton de delogare, acesta permițând utilizatorilor să se deconecteze rapid și în siguranță din contul lor. Această funcționalitate este esențială pentru securitatea și confidențialitatea datelor utilizatorilor, asigurând că informațiile personale sunt protejate.

Home

Componenta ”Home” constituie un punct de intrare strategic în aplicație, punând la dispoziție utilizatorilor detalii personale despre dezvoltatorul și motivația acestuia în dezvoltarea produsului. Utilizatorii sunt întâmpinați cu o prezentare a creatorului aplicației, care poate include informații despre experiența sa și principiile care stau la baza proiectului.

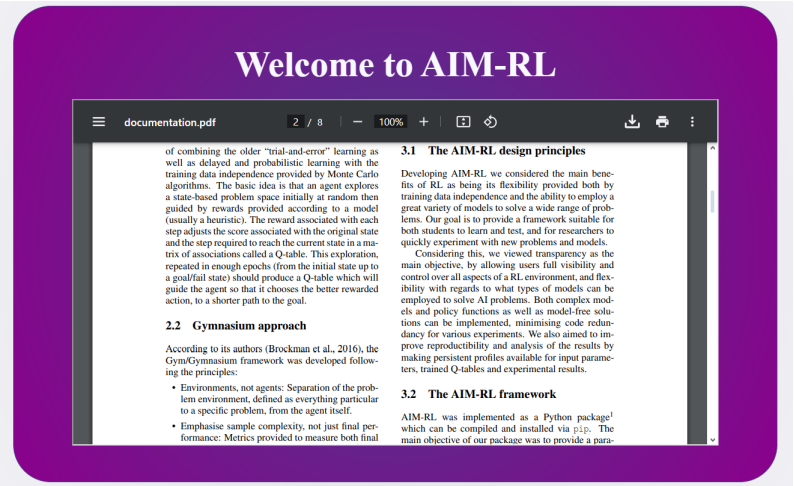
Pe lângă acest aspect, componenta ”Home” oferă utilizatorii și posibilitatea să exploreze secțiuni distincte, cum ar fi ”Abstract” (rezumatul proiectului), ”Overview” (informații generală), ”Significance” (importanța proiectului) și ”Conclusion” (concluziile).



Componenta ”Home” îmbogățește experiența utilizatorului, facilitând o înțelegere profundă a scopului, dar și a beneficiilor aplicației în sine. Utilizatorii sunt încurajați să exploreze mai în profunzime și să se angajeze activ în utilizarea și dezvoltarea aplicației.

Documentație

Componenta de documentație oferă detalii despre framework-ul utilizat pentru rezolvarea experimentelor. Aceasta include ghiduri, tutoriale și exemple de utilizare, ajutând utilizatorii să înțeleagă mai bine modul de funcționare al framework-ului AIM-RL, dar și îndrumându-i spre a înțelege cum pot realiza experimente de ”Reinforcement Learning”. Această componentă este esențială pentru utilizatorii care doresc să exploreze și să utilizeze pe deplin capabilitățile oferite de această platformă.



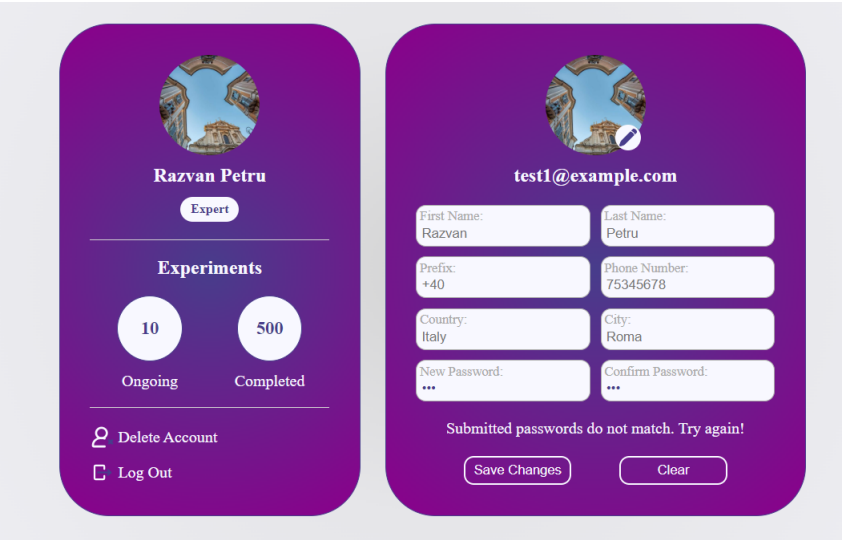
Profil

Această componentă este concepută pentru a oferi utilizatorului o privire de ansamblu asupra detaliilor contului propriu, dar și o modalitate ușoară de gestionare și editare a acestor informații ale contului. Componenta este divizată în două părți, fiecare cu propriul său scop unic: prezentarea generală a profilului utilizatorului și modalitatea principală de editare a informațiilor profilului.

Prezentarea generală Primul chenar (din partea stângă a componentei) servește drept prezentarea generală a profilului. Utilizatorul poate vizualiza rapid informațiile esențiale ale propriului cont și poate accesa funcționalități importante. În partea de sus este afișată poza de profil și numele complet al utilizatorului, urmat de rangul său. Rangul este determinat în funcție de numărul de experimente duse la bun sfârșit, un utilizator având ocazia să fie identificat drept ”Beginner”, ”Intermediate”, ”Advanced”,

”Expert” sau ”Master”. Această modalitate de gamificare urmărește încurajarea utilizatorilor să se implice cât mai mult pe platformă, pentru a debloca ranguri noi.

Prezentarea generală include, de asemenea, un rezumat al activității utilizatorului în cadrul platformei, arătând numărul de experimente în desfășurare, dar și numărul celor finalizate. Aceste statistici oferă utilizatorilor o imagine de ansamblu a progresului făcut, dar și a implicării lor în cadrul ”GIRL”. În partea de jos a prezentării generale, există două butoane: unul pentru deconectare de la contul curent și unul pentru ștergerea definitivă a contului. Aceste opțiuni sunt ușor accesibile, dar totuși separate de conținutul principal urmărind prevenirea accidentelor.



Editor de Profil Al doilea chenar (din partea dreaptă a componentei) este rezervată pentru editarea profilului. Această parte permite utilizatorilor să își actualizeze detaliile personale, având în partea de sus din nou poza de profil a utilizatorului, dar de această dată aceasta fiind interactivă. Utilizatorul poate să încarce o nouă imagine după ce face click pe poză, această funcționalitate îmbunătățind unicitatea profilului de utilizator.

Sub poza de profil, se află adresa de email a utilizatorului. Aceasta este nemodificabilă, asigurându-se astfel integritatea cheii principale de identificare a unui utilizator. După adresa de email, urmează diverse câmpuri pentru editarea diferitelor detalii personale: prenumele, numele de familie, prefixul telefonic, numărul de telefon, țara și orașul, fiecare dintre aceste câmpuri este populat cu informații care indică valorile

curente ale acestor detalii. În plus, există câmpuri pentru schimbarea parolei, unde utilizatorul trebuie să introducă noua parolă și să o confirme prin reintroducerea acesteia în câmpul următor. Dacă parolele nu se potrivesc, este afișat un mesaj de eroare, solicitând să se corecteze greșeala.

În ansamblu, componenta de profil este concepută pentru a fi intuitivă și ușor de utilizat, oferind o experiență perfectă pentru vizualizarea și actualizarea informațiilor personale. Prin utilizarea elementelor de tip ”placeholder”, a elementelor de tip ”clickable”, dar și a butoanelor de acțiune, se asigură gestionarea eficientă a profilului de către utilizator, contribuind la creșterea experienței pozitive pe platformă.

Dashboard

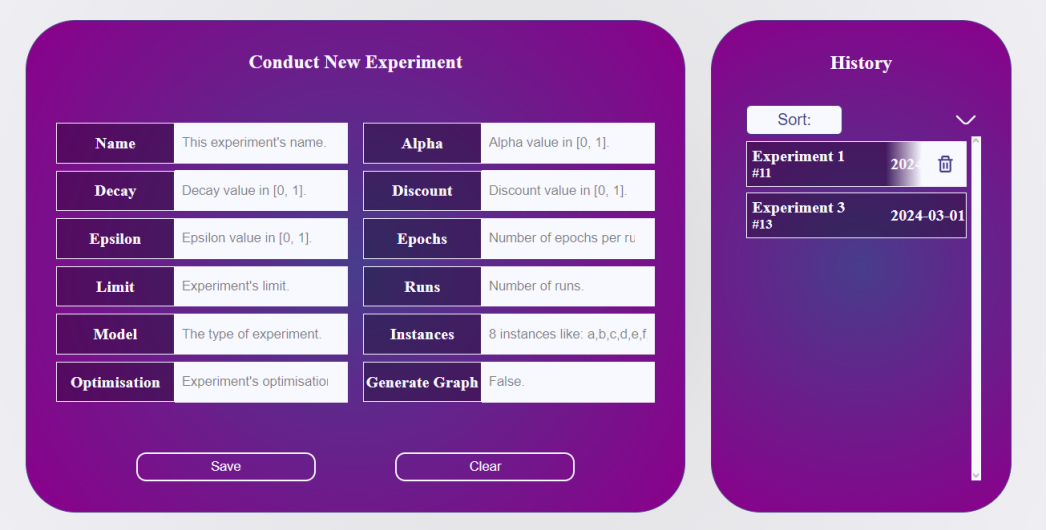
Componenta ”Dashboard” este concepută pentru a permite utilizatorului să gestioneze propriile experimente într-un mod eficient și organizat. Aceasta este împărțită în două secțiuni principale: una pentru crearea unui nou experiment și alta pentru gestionarea experimentelor anterioare.

Creare experimente Primul chenar (partea stângă a componentei) este destinat creării noilor experimente. Utilizatorul este nevoit să introducă parametrii necesari pentru configurarea și rularea unui experiment, fiecare parametru având câte un câmp dedicat în care se pot introduce doar valori specifice (acestea fiind comunicate folosind elemente de tip ”placeholder”). După completarea acestor câmpuri, utilizatorii au la dispoziție butoanele de „Save” și „Clear”.

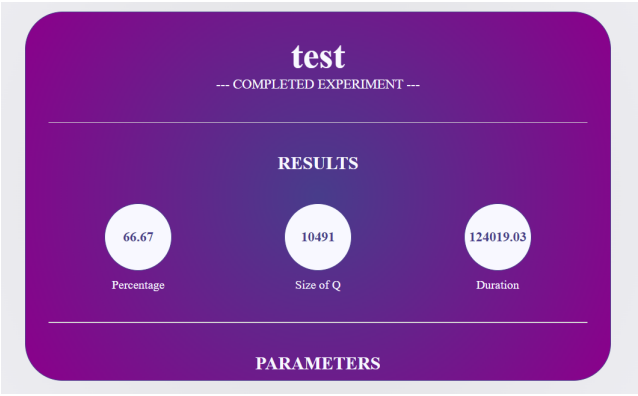
Istoric experimente Al doilea chenar (partea dreaptă a componentei) este destinat vizualizării istoricului de experimente. În această secțiune, utilizatorii au la dispoziție o listă cu experimentele rulate anterior. Lista poate fi sortată prin intermediul butonului „Sort” (element de tip ”hover”) fie în funcție de nume, fie după data creeri. Această funcționalitate permite utilizatorului să își organizeze istoricul în funcție de propriile preferințe.

Fiecare experiment din istoric este afișat împreună cu numele, numărul de identificare și data acestuia. Dacă utilizatorul plasează cursorul peste partea dreaptă a unui experiment, va apărea o iconiță sub formă de gunoi. Apăsarea acestei iconițe va duce la ștergerea definitivă a experimentului.

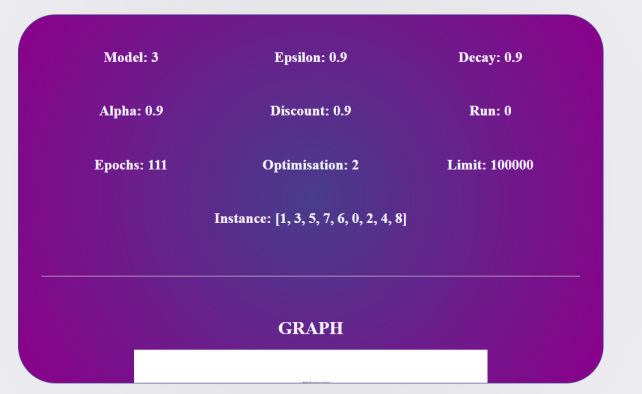
În plus, dacă utilizatorul apasă oriunde altundeva pe un experiment (iconița sub formă de gunoi nu este vizibilă), acesta va fi direcționat către pagina completă a acelui experiment, unde pot fi găsite toate detaliile și rezultatele.



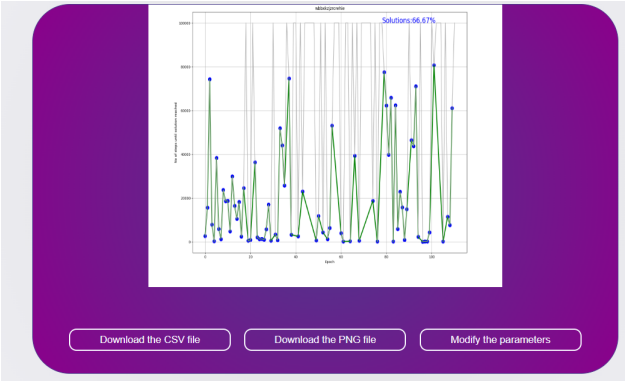
Vizualizarea unui experiment Pagina de vizualizare a unui experiment este structurată în așa fel încât să îi fie oferite utilizatorului toate informațiile într-o mod clar și lizibil. Sub titlul experimentului (aflat în partea de sus), sunt prezentate principalele rezultate ale experimentului. Acestea includ trei indicatori principali, cum ar fi procentajul de succes, dimensiunea setului de date utilizat și durata experimentului. Această secțiune oferă o privire rapidă asupra performanței și eficienței experimentului.



Imediat sub se află enumerați parametrii inițiali ai experimentului, aceștia fiind urmați de un grafic unic generat pentru acel experiment. Acesta ajută utilizatorul să înțeleagă mai bine dinamica experimentului și rezultatele obținute, însă această funcționalitate este disponibilă doar experimentelor pentru care parametrul inițial de generare a graficului este setat adevărat (”True”).



Utilizatorul are acces la trei butoane pentru acțiuni suplimentare. Primul permite descărcarea rezultatelor sub format CSV, al doilea permite descărcarea graficului în format PNG, iar al treilea buton oferă posibilitatea de a schimba parametrii inițiali pentru a rula din nou experimentul sub o altă formă. Aceste butoane au contribuția de a adaugă un nivel suplimentar de interactivitate și flexibilitate în gestionarea experimentelor.



2.2.2 Routing

”Routing”-ul în aplicația ”GIRL” este gestionat în principal prin intermediul framework-ului Angular 17+, care oferă un sistem de ”routing” flexibil, esențial pentru crearea unei aplicații web scalabile și ușor de folosit.

```
import { Routes } from '@angular/router';
import {HomeComponent} from "./components/home/home.component";
import {ProfileComponent} from "./components/profile/profile.component";
import {ExperimentsComponent} from "./components/experiments/experiments.component";
import {DocumentationComponent} from "./components/documentation/documentation.component";
import {AuthenticationComponent} from "./components/authentication/authentication.component";
import {MainComponent} from "./components/main/main.component";
import {ExperimentResultComponent} from "./components/experiment-result/experiment-result.component";


export const routes: Routes = [
  {path: '', component: MainComponent, children: [
      {path: 'home', component: HomeComponent},
      {path: 'profile', component: ProfileComponent},
      {path: 'experiments', component: ExperimentsComponent},
      {path: 'documentation', component: DocumentationComponent},
      {path: 'experiment/:id', component: ExperimentResultComponent},
      {path: '', redirectTo: 'home', pathMatch: 'full'}
    ]},
  {path: 'authentication', component: AuthenticationComponent}
];
```

Codul prezentat prezintă un set de combinații de rute pentru diferitele componente ale aplicației. La începutul fișierului, sunt importate toate componentele necesare, acestea reprezentând diverse părți ale aplicației.

Principala structură de rutare este definită într-o constantă de tip ”Routes”. Ruta de bază (cu path-ul ”) este asociată cu componenta ”Main”, care servește drept ”coloană vertebrală” a aplicației și conține alte rute de tip ”copil”. Acestea sunt definite într-un tablou unidimensional astfel încât fiecare componentă specifică să fie încărcată în funcție de ruta specificată. De asemenea, există o rută care redirecționează toate rutele neidentificate către componenta ”Home”, folosind redirectTo: ’home’ și pathMatch: ’full’. Aceasta asigură redirecționarea utilizatorului către pagina principală în cazul introducerii unei rute neexistente.

În afara structurii principale de rutare, există o rută separată specială pentru încărcarea componentei de autentificare. Aceasta gestionează procesele de autentificare a utilizatorului, inclusiv crearea noilor conturi, dar și autentificarea în conturi deja existente.

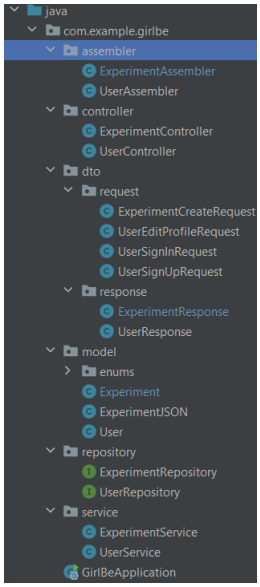
Fiecare rută definită în componenta ”Main” va fi combinată cu cea ”Nav”, ceea ce înseamnă că navigația principală a aplicației va fi prezentă împreună cu toate com-

ponentele care fac parte din aceste rute. Prin utilizarea router-outlet, componenta ”Main” permite încărcarea dinamică a componentelor copil în funcție de ruta specificată, menținând în același timp navigația principală (componenta ”Nav”) pe fiecare pagină.

În concluzie, sistemul de rutare implementat în această aplicație Angular 17 oferă o structură clară și eficientă pentru navigarea între diferitele componente. Acesta contribuie la o experiență de utilizare intuitivă, permițând utilizatorului să acceseze rapid funcționalitățile dorite, în timp ce componenta de navigație rămâne constant prezentă, asigurând acces rapid la principalele secțiuni ale aplicației.

2.3 Backend

2.3.1 Java Spring Boot



Crearea server-ului

Pentru componenta principală de backend a aplicației ”GIRL” a fost utilizat framework-ul ”Java Spring Boot”. Serverul creat cu ajutorul acestui framework conține multiple endpoint-uri destinate gestionării variatelor funcționalități legate de conturile utilizatorilor, precum și de experimentele acestora. Structura serverului este organizată în

diverse pachete, fiecare fiind responsabil pentru aspecte distincte ale funcționalității aplicației. Această organizare clară a rolurilor contribuie la o separare eficientă a responsabilităților, facilitând astfel mentenanța și dezvoltarea ulterioară a aplicației.

Model

Acest pachet cuprinde obiectele de bază și enumerările folosite în cadrul aplicației. Acest pachet include clasa Experiment, care encapsulează toate atributele și comportamentele asociate cu un experiment, clasa ExperimentJSON care reprezintă un format specific al obiectului Experiment în JSON, utilizat pentru procesele de serializare și deserializare și clasa User care reprezintă obiectul utilizatorului, cuprinzând atribute specifice utilizatorului, cum ar fi numele de utilizator, parola și informațiile de profil.

DTO

Pachetul DTO este organizat în 2 părți distincte: ”Request” și ”Response”, fiecare conținând DTO-uri pentru tratarea datelor de intrare și celor de ieșire. Aceste DTO-uri definesc structura datelor transmise dintre client și server.

Subpachetul ”Request” conține clasa ExperimentCreateRequest care encapsulează datele necesare creerii unui nou experiment, clasa UserEditProfileRequest care se ocupă de actualizările profilului utilizatorului, clasa UserSignInRequest care este folosită pentru autentificarea utilizatorului și UserSignUpRequest care se ocupă de înregistrarea inițială a utilizatorului.

Subpachetul ”Response” include clasele ExperimentResponse care furnizează clientului informații detaliate despre un experiment și UserResponse care pune la dispoziție date legate de utilizator într-un format structurat.

Assembler

Pachetul ”Assembler” joacă un rol esențial în transformarea obiectelor de domeniu în obiecte de tip DTO (Obiect Transfer de Date) și invers, asigurându-se astfel o separare clară între reprezentarea internă și cea externă a datelor.

Clasa ”ExperimentAssembler” este responsabilă pentru convertirea obiectelor de domeniu de tip Experiment în DTO-urile corespunzătoare. Această conversie asigură că structura datelor respectă formatul dorit pentru interacțiunile de rețea, realizând transferul de date fără între client și server.

Similar, clasa UserAssembler se ocupă de transformarea obiectelor de domeniu de tip User în DTO-uri, asigurând că datele legate de utilizatori sunt corect formate pentru comunicarea cu exteriorul, menținându-se astfel integritatea datelor.

Controller

Acest pachet este dedicat gestionării cererilor HTTP primite și definirii endpoint-urilor aplicației. "Controller"-ele acționează ca intermediari între client și pachetul de servicii. Acestea procesează cererile, invocă metodele serviciilor și returnează răspunsuri.

ExperimentController are sarcina de a mapa cererile HTTP legate de experimente la serviciile corespunzătoare, dar și de a gestiona operațiile CRUD pentru experimente, asigurându-se astfel că aceste cereri sunt procesate precis.

UserController se ocupă de cererile HTTP legate de operațiunile utilizatorului, cum ar fi: înregistrarea, autentificarea și editarea profilului. Comunicând cu serviciul UserService, UserController execută logica specificată.

Service

Pachetul "Service" encapsulează logica serviciilor aplicației, acestea fiind responsabile pentru executarea funcționalităților și regulilor esențiale, interacționând adesea cu Repository-urile pentru preluarea datelor.

Clasa ExperimentService se concentrează pe logica serviciilor legate de experimentele unui utilizator, coordonând împreună cu ExperimentRepository manipularea experimentelor, dar și asigurând respectarea regulilor de către toate operațiile întreprinse.

Clasa UserService se ocupă de logica serviciilor specializate pe controlarea utilizatorilor, interacționând cu UserRepository pentru a efectua sarcini, cum ar fi înregistrarea utilizatorilor, autentificarea lor, dar și actualizările de profil.

Repository

Pachetul "Repository" conține interfețe care facilitează operații de acces la date. Aceste interfețe abstractizează interacțiunea cu baza de date, oferind o abordare clară și testabilă pentru manipularea datelor.

Interfața ExperimentRepository definește operațiile CRUD pentru obiectele de tip Experiment, asigurând gestionarea datelor într-un mod eficient.

Similar, interfața UserRepository manipulează operațiile CRUD pentru obiectele de tip User, susținând funcționalități cum ar fi: găsirea utilizatorilor după numele de utilizator sau după email, simplificând astfel folosirea datelor utilizatorilor.

2.3.2 Comunicarea cu Frontend-ul

Serverul realizat cu ajutorul framework-ului ”Java Spring Boot” interacționează cu partea de frontend, dezvoltată folosind framework-ul Angular 17, prin intermediul endpoint-urilor API-ului de tip RESTful. Această interacțiune se bazează pe un model client-server, în care frontend-ul (clientul) trimite cereri de tip HTTP către endpoint-urile serverului (backend-ul), care procesează aceste cereri și returnează răspunsuri specifice.

Această arhitectură asigură o separare limpede a responsabilităților, unde frontend-ul se ocupă de interfața cu utilizatorul, dar și de experiența acestuia, în timp ce backend-ul manipulează logica serviciilor și accesul la date. Comunicarea prin endpoint-urile API-ului de tip RESTful și utilizarea formatului JSON în schimbul de informații contribuie la un design modular și scalabil, permițând actualizări și întrețineri facile ale ambelor părți ale platformei ”GIRL”. De asemenea, această abordare permite frontend-ului și backend-ului să fie dezvoltate, testate și implementate independent, oferind astfel flexibilitate și eficiență în procesul de dezvoltare a aplicației.

2.3.3 Comunicarea cu AIM-RL

În cadrul mediului de dezvoltare ”IntelliJ”, pe lângă directorul destinat codului sursă Java, există și un director dedicat pentru cod Python, acesta conținând framework-ul AIM-RL. Acest aranjament permite integrarea eficientă a funcționalităților oferite de AIM-RL în aplicația Java, dar și nivele sporite de accesibilitate și manipulare.

Utilizatorul interacționează cu interfața frontend, specificând parametrii necesari pentru experimentul de ”Reinforcement Learning”, cererea fiind transmisă către serverul backend unde este prelucrată și unde se generează o comandă pentru terminal, care include apelul funcției qlearning cu parametrii primiți de la utilizator. Comanda este construită astfel încât să specifice corect locația scriptului Python și să includă toți parametrii necesari pentru funcționarea fără probleme a procesului.

După executarea comenzii din terminal, rularea funcției qlearning este declanșată în mediul Python. Aceasta începe experimentul propriu-zis, procesul de învățare prin

”Reinforcement Learning” conform specificațiilor utilizatorului fiind inițializat. După finalizarea experimentului, rezultatele sunt preluate și salvate într-un fișier CSV și unul PNG (pentru grafic), acestea fiind accesibile ulterior pentru analizare. Backendul transmite rezultatele experimentului înapoi către frontend, unde utilizatorul poate vizualiza și analiza performanțele experimentului concluzionat cu ajutorul modelului de ”Reinforcement Learning”.

```
public List<Experiment> createExperiment(ExperimentCreateRequest experimentCreateRequest, Long userId) {
    User user = userRepository.findById(userId).orElse( other: null);
    runScript(experimentCreateRequest);
    List<Experiment> experiments = ExperimentAssembler.createExperimentsFromCSV(experimentCreateRequest, user);
    experimentRepository.saveAll(experiments);
    return experiments;
}

private void runScript(ExperimentCreateRequest experimentCreateRequest) {
    ObjectMapper mapper = new ObjectMapper();
    File file = new File( pathname: "src/main/python/config.json");
    try {
        mapper.writeValue(file, ExperimentAssembler.createExperimentJSON(experimentCreateRequest));
        System.out.println("JSON file created successfully: " + file.getPath());

        Process process = Runtime.getRuntime().exec( command: "python ./src/main/python/main8-puzzle.py ./src/main/python/config.json");
        int exitCode = process.waitFor();
        if (exitCode == 0) {
            System.out.println("Python script executed successfully.");
        } else {
            System.out.println("Python script execution failed with exit code: " + exitCode);
        }

    } catch (IOException | InterruptedException e) {
        e.printStackTrace();
    }
}
```

Acest sistem permite o integrare fluidă între codul dezvoltat în Java și cel dezvoltat în Python, beneficiind de avantajele ambelor limbaje de programare și asigurând o flexibilitate ridicată în dezvoltarea și testarea serverului, dar și a aplicației în sine. În plus, această arhitectură modulară facilitează întreținerea și scalabilitatea aplicației în viitor, permițând dezvoltatorilor să adauge noi funcționalități și să optimizeze algoritmii de ”Reinforcement Learning” fără a pune în pericol celelalte componente ale aplicației. Se asigură astfel o adaptabilitate sporită la continua schimbare a cerințelor utilizatorului, dar și la evoluția rapidă a domeniului de ”Reinforcement Learning”.

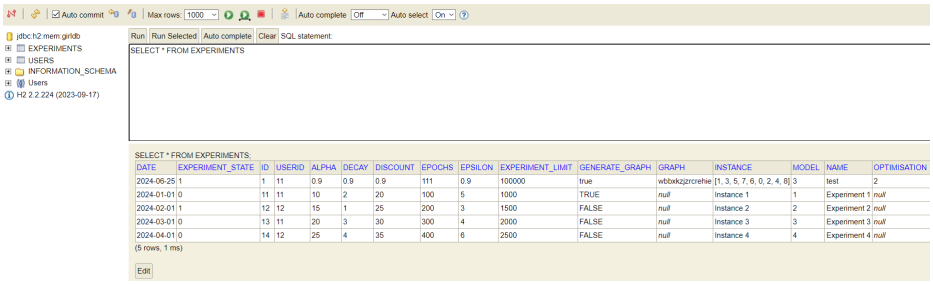
2.3.4 H2

Pentru baza de date a aplicației ”GIRL”, s-a optat pentru utilizarea H2, un motor de bază de date relațional realizat în Java. Alegerea H2 a fost influențată de mai mulți factori care se potrivesc nevoilor proiectului. Unul dintre avantajele majore ale acestei baze de date este ușurința de integrare cu aplicațiile realizate cu ajutorul frameworkului Java Spring Boot, datorită suportului său excelent pentru configurarea în memorie

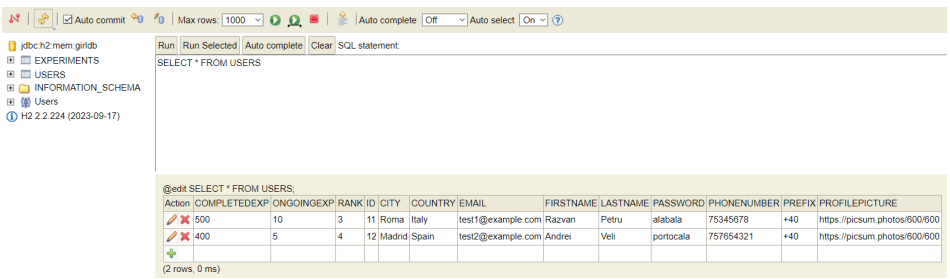
sau pe disc. Este garantată dezvoltarea și testarea rapidă, deoarece este permisă o configurare simplă a bazelor de date.

H2 oferă performanțe bune și în același timp o amprentă mică de memorie, ceea ce îl face ideal pentru aplicații care necesită un motor de bază de date ușor și rapid. În plus, H2 are o compatibilitate excelentă cu alte baze de date relaționale majore, cum ar fi PostgreSQL și MySQL, permițând potențiale tranziții către alte sisteme de baze de date, dacă vor fi necesare în viitor.

Cu toate acestea, există și anumite dezavantaje asociate cu utilizarea bazei de date H2. Unul dintre principalele minusuri este că, fiind un motor de baze de date în memorie, datele nu sunt persistente după închiderea aplicației, excepție fiind cazul în care se configurează explicit stocarea pe disc. De asemenea, H2 nu este recomandat pentru utilizarea în producție pe scară largă datorită limitărilor sale în manipularea unui volum mare de date, dar și a încărcării concurente ridicate.



În ceea ce privește structura bazei de date a aplicației ”GIRL”, tabelele principale sunt cele ale experimentelor și ale utilizatorilor. Tabelul experimentelor conține informații esențiale despre fiecare experiment în parte, fiind inclusă o cheie străină (id-ul user-ului), care face referință la utilizatorul căruia îi aparține experimentul respectiv.

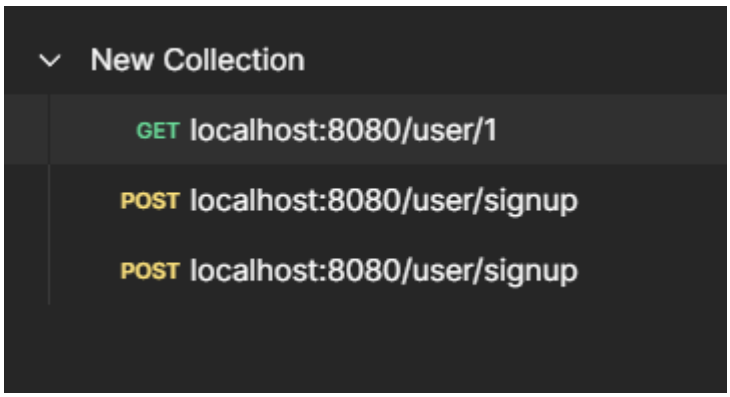


Tabelul user-ilor stochează detalii importante despre fiecare utilizator, câmpurile destinate stocării email-ului și a id-ului fiind unice. Această structură relațională asigură o organizare clară și stabilă a datelor, permițând operațiile CRUD, dar și interogările complexe necesare pentru anumite funcționalități ale aplicației.

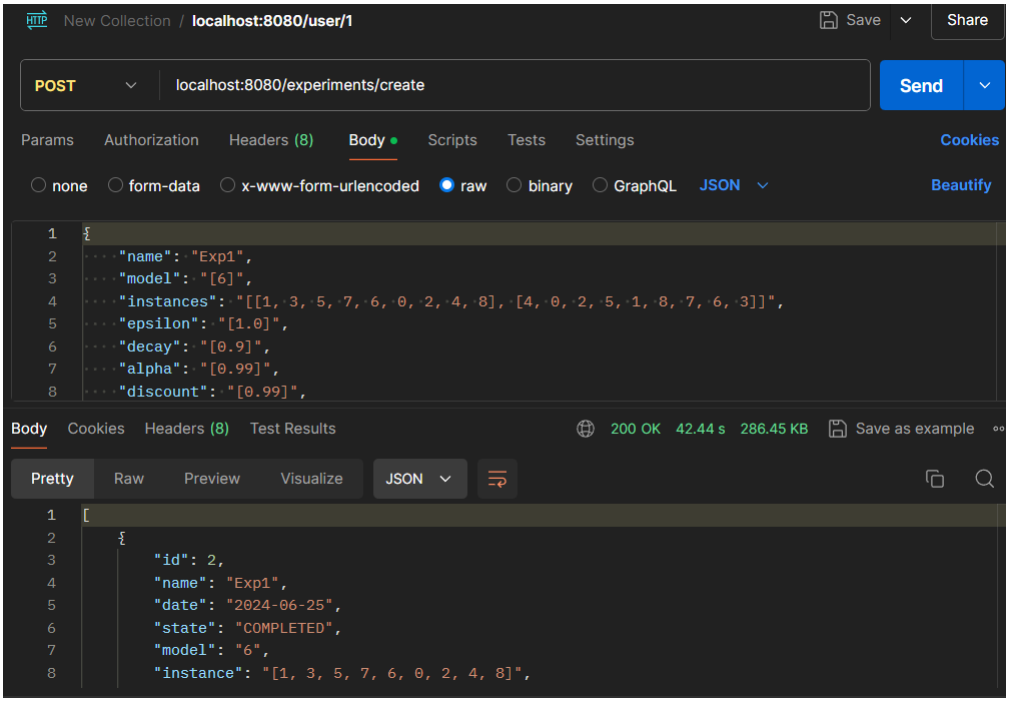
2.3.5 Postman

Postman este un instrument pentru dezvoltarea și testarea endpoint-urilor unui API, acesta facilitând verificarea și înțelegerea acestora. Postman oferă o interfață prietenoasă și ușor de folosit pentru crearea de cereri de tip HTTP, vizualizarea răspunsurilor, dar și automatizarea scripturilor de testare. Prin suportul său pentru diverse metode HTTP (cum ar fi GET, POST, PUT și DELETE), Postman oferă posibilitatea testării în detaliu a tuturor funcționalităților unui API, făcându-l astfel indispensabil în practicile moderne de dezvoltare a aplicațiilor.

Decizia de a folosi Postman pentru testarea funcționalității serverului aplicației "GIRL" a fost motivată de mai mulți factori distincți. În primul rând, interfața intuitivă a instrumentului permite configurarea și execuția rapidă a cererilor de testare, reducând drastic curba de învățare pentru noii utilizatori. Capacitatea sa de a salva și organiza cererile în colecții sporește eficiența, oferind posibilitatea unei testări sistematice, dar și a reutilizării unor cazuri vechi de testare. Aceste capabilități sunt esențiale pentru asigurarea integrității endpoint-urilor unui API, dar și pentru asigurarea unei interacțiuni fără probleme între componentele backend și frontend în cadrul unei aplicații.



Pentru a testa serverul construit cu Java Spring Boot, Postman a fost utilizat extensiv pe parcursul ciclului de dezvoltare. Instrumentul a fost folosit pentru a trimite cereri HTTP către diversele endpoint-uri API definite în controlerele serverului. Spre exemplu, pentru testarea funcționalității de creere a unui experiment, a fost trimisă o cerere POST (conținând un JSON cu datele experimentului și id-ul utilizatorului) către endpoint-ul specific. Postman a permis inspectarea statusului răspunsului, a headerelor, dar și a corpului răspunsului, verificând astfel dacă serverul a procesat corect cererea de autentificare și a oferit un răspuns adecvat, asigurându-se astfel că operațiile CRUD au fost interpretate în mod corect de către server.



Utilizarea Postman pentru testarea funcționalităților și endpoint-urilor serverului a adus numeroase beneficii. Acesta a asigurat validarea formatelor de cerere și răspuns, asigurându-se că datele schimbate între client și server respectă structurile specifice.

În general, Postman s-a dovedit a fi un instrument valoros în dezvoltarea și testarea serverului aplicației ”GIRL”. Prin utilizarea sa, s-a reușit identificarea și rezolvarea rapidă a problemelor în ciclul de dezvoltare a platformei, făcând astfel aplicația ”GIRL” mai stabilă, dar și mai performantă.

Capitolul 3

Concluzii

În concluzie, dezvoltarea unei platforme web dedicate experimentelor de ”Reinforcement Learning” a fost un proces complex și provocator, care a implicat o planificare riguroasă, utilizarea unor tehnologii moderne și eficiente, dar și înțelegere in profunzime a conceptelor specifice acestui domeniu și a modului în care acestea pot fi implementate într-un mediu web.

3.1 Deznodământ

Alegerea framework-urilor Angular 17 pentru frontend și Java Spring Boot pentru backend, împreună cu baza de date H2, au oferit o implementare armonioasă și potențial scalabilă. Integrarea noului framework AIM-RL, dezvoltat în Python, a permis realizarea efectivă a experimentelor de ”Reinforcement Learning” într-un mod eficient și ușor de înțeles. Utilizarea unor instrumente moderne, precum Postman, WebStorm și IntelliJ, a grăbit procesul de dezvoltare și a sporit exponențial productivitatea.

Organizarea sarcinilor prin intermediul platformei Trello a fost de foarte mare ajutor pentru folosirea cât mai eficientă a timpului avut, asigurând în același timp o imagine clară asupra progresului făcut și permițând adaptarea instantă la potențiale schimbări. Deși proiectul a fost dus la bun sfârșit de o singură persoană, abordarea metodică și utilizarea uneltelor de management (al timpului) au contribuit semnificativ la succesul avut.

Produsul final oferă utilizatorului șansa de a face experimente de ”Reinforcement Learning” într-un mod ușor de înțeles și interactiv, punându-se accent pe analiza și înțelegerea rezultatelor. Acest proiect personal demonstrează faptul că o planificare

detaliată și utilizarea unor tehnologii de ultimă generație pot conduce la crearea unui produs inovativ și performant.

3.2 Posibile îmbunătățiri

Deoarece aplicația a fost dezvoltată având o atenție sporită îndreptată spre mărirea scalabilității, există câteva direcții de îmbunătățire inițiale, acestea putând fi explorate de viitorii dezvoltatori pentru a spori funcționalitatea și performanța platformei.

O schimbare posibilă ar fi integrarea unei baze de date relaționale mai avansate (cum ar fi PostgreSQL sau MySQL) în locul celei actuale (H2). Aceasta ar permite controlarea unui volum mai mare de date, avansând nivelele de securitate și scalabilitate.

Un alt aspect ce ar putea fi îmbunătățit este reprezentat de adăugarea unei modalități avansate de vizualizare a rezultatelor unui experiment. Integrarea unor biblioteci de grafică (precum D3.js sau Chart.js) ar permite utilizatorilor să observe concluzia experimentelor făcute într-un mod mult mai dinamic, oferind o analiză mai amplă a performanțelor algoritmilor de ”Reinforcement Learning”.

O altă posibilă îmbunătățire ar fi adăugarea de suport pentru execuția experimentelor de ”Reinforcement Learning”. Prin utilizarea unor tehnologii mai puternice (precum Kubernetes sau Docker Swarm) aplicația ar putea susține execuția paralelă a experimentelor pe mai multe noduri simultan, reducând astfel timpul așteptat pentru obținerea rezultatelor și sporind capacitatea de calcul a server-ului în sine.

Încă o îmbunătățire ar fi cea de introducere unui sistem de prietenie, care să permită utilizatorilor să ceară prietenia altor utilizatori, prietenii având posibilitatea să poarte conversații între ei sau chiar să își trimită experimente personale, colaborarea și schimbul de informații fiind plusuri importante într-un domeniu științific. Pe baza rangului fiecărui utilizator, s-ar putea realiza și un clasament între prieteni pe baza numărului de experimente duse la bun sfârșit, putând fi astfel promovată competiția sănătoasă, dar și învățarea colaborativă, utilizatorii fiind motivați să își îmbunătățească abilitățile și să obțină rezultate mai bune.

Prin explorarea și implementarea acestor idei, aplicația ar putea evolua într-un instrument și mai puternic pentru consumul de experimente de ”Reinforcement Learning”, fiind astfel oferită utilizatorilor o experiență mai bogată și mai familiară.

Bibliografie

- Jonas Schmedtman, *"The Complete JavaScript Course"*, Udemy, 2022
- Mosh Hamedani, *"The Complete Angular Course"*, Udemy, 2023
- ForrestKnight, *"Everything You NEED to Know About WEB APP Architecture"*, Youtube, 2023
- Beau Carnes, *"Master API Testing with Postman"*, freeCodeCamp.org, 2023
- Segun Ajibola, *"How to Use Git and GitHub – Introduction for Beginners"*, freeCodeCamp.org, 2023
- Bro Code, *"Java Full Course for free"*, Youtube, 2023
- Bro Code, *"Python Full Course for free"*, Youtube, 2023
- Bro Code, *"Learn Data Structures and Algorithms for free"*, Youtube, 2023
- Telusko, *"Spring Boot Tutorials — Full Course"*, Youtube, 2023
- Devtiro, *"The ULTIMATE Guide to Spring Boot"*, Youtube, 2023
- Kindson The Tech Pro, *"H2 In-Memory Database with Spring"*, Youtube, 2024
- Code With Valan, *"H2 Database"*, Youtube, 2024
- Arxiv Insights, *"An introduction to Reinforcement Learning"*, Youtube, 2024
- Ionuț-Cristian Pistol & Andrei Arusoaie, *"AIM-RL: A new Framework Supporting Reinforcement Learning Experiments"*, PDF, 2024
- geeksforgeeks.org, 2021-2024
- stackoverflow.com, 2022-2024
- reddit.com, 2022-2024