Monitorizarea si controlul proiectelor software

Adrian Dinu Urse

Coordonator stiintific:

Prof. dr.ing. Alexandru BOICEA

Proiectul "Monitorizarea si controlul proiectelor software" reprezinta o platforma web care vine in ajutorul directorilor de companii sa gestioneze eficient activitatea din organizatiile lor.

Aplicatia isi propune sa ofere o solutie de gestiune a dezvoltarii proiectelor software, prin unelte si functionalitati care sa ajute la monitorizarea si controlul proiectelor software in timp real, astfel incat managerii de proiect sa poata lua decizii eficiente si mai rapide.

Prin intermediul acestei platforme web, managerii de companii pot reduce riscul de a intampina probleme in timpul derularii proiectelor si asigura finalizarea lor cu succes.

Cuvintele cheie care descriu tematica lucrarii: monitorizarea proiectelor software, controlul proiectelor software, platforma web, gestionarea proiectelor si a resurselor, stadiul proiectelor in timp real, planificare si organizare, urmarirea si controlul costurilor.

**1. Inroducere**

Întrucât tehnologia și cerințele clienților evoluează constant, proiectele software au devenit din ce în ce mai complexe, iar gestiunea lor devine din ce in ce mai dificila. Multe proiecte se confrunta cu probleme de management care de multe ori ajung sa nu fie finalizate, sa nu respecte cerintele clientilor, sau sa nu fie livrate la timp.

De aceea este necesar un sistem de monitorizare si control al proiectelor care să permită managerilor de proiect să urmărească stadiul proiectelor în timp real și să ia măsuri imediate in cazul problemelor care apar. O astfel de aplicatie web ofera management pentru activitatile din cadrul unei companii IT, cu scopul de a imbunatati procesul de dezvoltare interna. Aceasta ofera o platforma centralizata pentru gestionarea tuturor activitatilor si resurselor implicate in dezvoltarea proiectelor, cum ar fi urmarirea task-urilor, planificarea si organizarea activitatilor, precum si urmarirea si controlul costurilor. Aceste functionalitati ajuta la optimizarea fluxului de munca, reducerea timpului si costurilor implicate in dezvoltarea proiectelor.

**2. Problema analizată**

Tema propusa analizeaza problema ineficientei si a nerespectarii termenelor de livrare in dezvoltarea proiectelor si ofera o solutie software pentru a facilita managementul proiectelor, al organizatiei si al performantei intr-un mod placut si eficient.

**3. Obiectivele lucrarii**

**3.1 Obiective principale**

* Obiectivul principal al proiectului consta in dezvoltarea unei platforme web care sa imbunatateasca gestiunea dezvoltarii proiectelor software prin unelte si functionalitati destinate sa ajute la monitorizarea si controlul intr-un mod eficient al acestora.
* Implementarea unei solutii care sa permita managerilor de proiecte sa reduca posibilele riscuri in timpul dezvoltarii proiectelor si sa asigure finalizarea lor cu succes.
* Testarea si validarea aplicatiei

**3.2 Obiective secundare**

* Crearea unei aplicatii cu o interfata simpla, intuitiva, prietenoasa cu utilizatorul
* Implementarea unei aplicatii sigure, care nu permite accesul utilizatorilor neaturoziti la baza de date si care sa restrictioneze accesul utilizatorilor in functie de rolul acestora.
* Furnizarea de date in timp real despre stadiul proiectelor

**3. Contextul științific relevant subiectului**

În prezent, aplicațiile web reprezintă o componentă cheie a lumii digitale. Odată cu creșterea accesibilității la internet și a utilizării dispozitivelor mobile, aplicațiile web au devenit tot mai populare și mai frecvent utilizate de către utilizatorii din întreaga lume.

Aceste aplicații pot fi utilizate într-o varietate de contexte și domenii. Ele oferă utilizatorilor o serie de avantaje, precum accesul ușor la informații și servicii, interacțiunea rapida cu alte persoane, precum și o experiență personalizată și adaptată nevoilor și preferințelor lor.

În ultimii ani, dezvoltarea tehnologiei și a industriei IT a dus la o creștere exponențială a numărului de proiecte software în cadrul companiilor și organizațiilor din diverse domenii. Aceste proiecte sunt deosebit de complexe și necesită o coordonare și o gestionare eficientă pentru a putea fi finalizate cu succes.

În acest context, cercetările și dezvoltarea soluțiilor de monitorizare și control al proiectelor software au devenit un subiect de mare interes pentru comunitatea științifică și pentru industria IT.

Tema propusa este foarte relevanta in contextul stiintific actual, cercetările și dezvoltarea soluțiilor de monitorizare și control al proiectelor software reprezintă o direcție importantă în dezvoltarea tehnologiilor și a industriei IT.

**4. Studiu de piata/Abordari existente**

Într-un mediu de afaceri tot mai complex și competitiv, companiile care activează în domeniul dezvoltării software trebuie să fie capabile să livreze produse de înaltă calitate, în termeni rapizi și cu costuri cât mai reduse. Monitorizarea și controlul proiectelor software reprezintă elemente esențiale pentru a atinge aceste obiective și pentru a asigura succesul într-un mediu dinamic și imprevizibil.

In prezent exista diverse solutii software concepute sa ajute companiile la gestiunea dezvoltarii proiectelor software. In continuare, voi analiza abordarile si solutiile existente pe aceasta tema.

**4.1 Jira Software**

Este o platforma de management a proiectelor dezvoltata de compania australiană Atlassian. A fost lansata in 2002, folosita initial doar de dezvoltatorii software pentru urmarirea problemelor tehnice din cadrul proiectelor. In timp, a evoluat intr-o unealta flexibila si complexa de gestionare a proiectelor, fiind adoptata de organizatii din diverse domenii de activitate.

Jira ofera o gama larga de functionalitati, precum gestionarea sarcinilor, urmarirea problemelor de dezvoltare, planificarea etapelor de dezvoltare din cadrul proiectelor, urmarirea timpului si a resurselor, integrare cu Bitbucket și GitHub.

**4.2 Trello**

Trello este o aplicatie gratuita de gestionare a proiectelor la scara mica si a sarcinilor, care permite utlizatorilor sa isi organizeze activitatile intr-un mod vizual, pe baza unui sistem de panouri si liste. Ofera o modalitate simpla si usor de utilizat de planificare a activitatilor.

Utilizatorii Trello pot crea tablouri pentru diferite proiecte sau sarcini, adaugand liste pentru fiecare etapă de dezvolare a proiectului. În plus, Trello oferă o serie largă de integrări și plugin-uri, care permit utilizatorilor să sincronizeze datele cu alte aplicații terțe, cum ar fi Google Drive, Slack sau Jira.

**4.3 Asana**

Serviciul Asana este o platforma web si pentru mobil de management, proiectata sa ajute echipele sa organizeze, sa urmareasca si sa gestioneze activitatile lor. A fost lansata comercial in aprilie 2012 de catre Dustin Moskovitz și Justin Rosenstein, foști angajați ai Facebook.

Caracteristicile Asana include crearea proiectelor, crearea sarcinilor, atribuirea sarcinilor membrilor echipei, urmărirea progresului sarcinilor și stabilirea termenelor limită. Ofera, de asemenea, posibilitatea de acomunica direct in aplicatie, comentarii și notificări automate.

Asana poate fi integrata cu diverse aplicatii, precum Google Drive, Slack si Dropbox

**4. Solutia propusa**

**4.1 Structura proiectului**

Aplicatia propusa este structura in două componente majore: Back-End si Front-End.

**4.1.1 Componenta Back-End**

Componenta Back-End este partea server a aplicatiei, responsabilă cu gestionarea si stocarea datelor prin intermediul sistemelor de baza de date, implementarea logicii de business a aplicatiei. Ea nu comunica in mod direct cu utilizatorul, primeste si proceseaza cererile trimise de componenta Front-End. Se ocupa de interactiunea cu baza de date, autentificarea și autorizarea utilizatorilor, gestionarea sesiunilor, securitatea datelor.

In dezvoltarea backend am folosit framework-ul Java Spring, care este o unealta populara, open-source, enterprise, conceputa pentru a crea aplicatii web de sine statoare care ruleaza pe Masina Virtuala Java(Java Virtual Machine – JVM).

In cadrul proiectului, am utilizat urmatoarele module Java Spring, care faciliteaza dezvoltarea aplicatiilor web.

1. Spring MVC ofera suport pentru crearea aplicatiilor web bazate pe arhitectura Model-View-Controller (MVC). Permite o separare clara intre date, prezentare si logica de control.

2. Spring Data este modulul care se ocupa de interactiunea cu baze de date relationale si nerelationale. Din acest modul am folosit Spring Data JPA ce furnizează o interfață simplificată pentru a lucra cu baze de date relaționale utilizând Java Persistence API (JPA) si Spring Data Java Database Connectivity(JDBC) care oferă suport pentru conectarea la diverse baze de date si executarea cererilor.

3. Spring Security ofera suport pentru implementarea autentificarii si autorizarii utilizatorilor, protectie impotriva atacurilor comune prin diverse mecanisme de criptare. Pentru autorizare am folosit Java Web Token(JWT), care este un token de acces emis si validat de sistemul de autentificare. Acest token este pus in cererile ulterioare pentru a confirma identitatea utilizatorului și pentru a permite accesul la resursele protejate.

4. Spring Boot este un modul ce simplifica configurarea si dezvoltarea aplicatiilor Spring, oferind posibilitatea de a genera rapid un schelet de aplicatie si de a seta configuratiile necesare.

Am utilizat framework-ul Hibernate pentru a face maparea între obiectele Java și tabelele din baza de date, prin intermediul tehnologiei ORM (Object-Relational Mapping).

Pentru a gestiona si controla schimbarile de structura ale bazei de date, a mentine integritatea datelor am folosit Liquibase ce este o biblioteca open-source independenta de baza de date. Aceasta permite dezvoltatorilor să definească, să urmărească și să aplice schimbări la schema bazei de date într-un mod controlat și reversibil. In loc sa se foloseasca cod SQL, se folosesc fișiere XML sau YAML pentru a descrie schimbarile pe baza de date.

**4.1.2 Component Front-End**

Componenta Front-end, partea client a aplicatiei, reprezinta interfata grafica prin care utlizatorul poate sa acceseze datele stocate in baza de date si sa interactioneze cu acestea

Aceasta este formată dintr-un set de fișiere HTML, CSS și JavaScript, care sunt încărcate în browserul utilizatorului și procesate in browser-ul utilizatorului.

Pentru a dezvolta partea de front-end am folosit framework-ul Angular, ce este o platforma open-source dezvoltata de Google, utlizata pentru crearea aplicatiilor web dinamice si scalabile. Ea este bazata pe limbajul TypeScript o versiune stricta a limbajului JavaScript, si ofera o serie de componente si functionalitati utile, precum sistemul de rute, validare de date, si interconectarea cu API-uri externe.

**4.1.3 Comunicarea intre cele doua componente**

Cele 2 componente comunica prin intermediul API-urilor RESTful. Frontendul, care rulează în browserul utilizatorului, face cereri către backend pentru a obține date sau a executa anumite acțiuni. Cererile sunt făcute către anumite endpoint-uri, care reprezintă punctele de acces la resursele din backend.

Backend-ul, primește cererile, le procesează și returnează un răspuns către frontend. Răspunsul poate fi o pagină HTML generată dinamic, un fișier JSON, o imagine sau orice alt tip de date.

Interacțiunea între cele două componente este realizată prin intermediul protocoalelor HTTP și HTTPS. Cererile sunt de obicei făcute prin metodele HTTP GET, POST, PUT, DELETE sau PATCH, în funcție de acțiunea pe care trebuie să o execute backend-ul. Răspunsurile sunt, de obicei, trimise în format JSON, care este un format ușor de citit și procesat de către frontend.

**4.2. Arhitectura proiectului**

Aplicatia propusa are o arhitectura Three-tier, care este o arhitectura ierarhică închisă, în 3 niveluri, specifica sistemelor care include o baza de date. Ea este des folosită în sistemele bazate pe Web.

1. Nivelul de prezentare(frontend): Nivelul cu care utilizatorul final interactioneaza, se ocupa cu prezentarea datelor intr-un mod intuitiv si accesibil.

2. Nivelul aplicatie(backend – business logic tier): Este executat pe un server web si este responsabil cu procesarea si manipularea datelor. Implementeaza logica de business, validarea datelor si interactiunea cu baza de date.

3. Nivelul stocare: Este executat pe un server de baze de date si este responsabil cu stocarea si gestionarea datelor.

Avantajele arhitecturii:

* Cele 3 niveluri pot fi dezvoltate în mod independent, de echipe diferite.
* Fiecare dintre cele 3 niveluri poate fi îmbunatatit sau înlocuit independent, în cazul unei schimbari a cerintelor sau a unei schimbari tehnologice (hardware nou).
* Nivelul stocare, analog unui subsistem Repository, poate fi partajat de diferite aplicatii care utilizeaza aceeasi baza de date.
* 

  Architectural pattern for a three-tier application

  Separarea nivelului interfata de nivelul aplicatie permite existenta mai multor interfeṭe utilizator pentru subsistemul/subsistemele care implementeaza logica aplicatiei.

Fig. 1 - Diagrama arhitecturii Three-tier

**4.3. Baza de date**

Pentru gestionarea datelor, componenta Back-End foloseste sistem de baza de date PostgreSQL.

PostgreSQL este un sistem de gestiune a bazelor de date relaționale (RDBMS) open source, cunoscut și sub numele de Postgres. Initial a fost numit Postgres, referindu-se la originile sale ca un succesor al bazei de date Ingres. A fost dezvoltat de Universitatea din California, Berkeley, începând cu anii '80 și a fost lansat în mod oficial în 1996 cu numele PostgreSQL pentru a reflecta suportul pentru SQL.

PostgreSQL are multe functionalitati avansate, care îl fac un sistem de gestiune a bazelor de date foarte puternic și flexibil.

1. Tranzacții ACID: PostgreSQL implementează tranzacțiile ACID (Atomic, Consistent, Isolated, Durable), ceea ce înseamnă că orice modificare a datelor din baza de date este tratată ca o tranzacție și este gestionată astfel încât să fie atomică, consistentă, izolată și durabilă.

2. Chei străine și integritate referențială: PostgreSQL permite definirea cheilor străine pentru a asigura integritatea referențială a datelor și pentru a stabili relații între tabele.

3. Declanșatoare (triggere): PostgreSQL suportă declanșatoare care pot fi activate automat de evenimente, precum inserarea, actualizarea sau ștergerea de date, și care pot executa acțiuni personalizate în funcție de nevoile aplicației.

4. Tipuri definite de utilizator

5. Indexare avansată: PostgreSQL oferă o gamă largă de mecanisme de indexare, inclusiv indexare unică sau multiplă, indexare parțială, indexare pentru expresii și multe altele, pentru a creste performanța interogărilor

6. Replicare: PostgreSQL permite replicarea datelor între mai multe instanțe de PostgreSQL, ceea ce permite scalarea orizontală și distribuirea sarcinilor între mai multe servere.

In cadrul proiectului baza de date ruleaza intr-un container Docker, care ofera urmatoarele avantaje:

1. Portabilitate: Utilizarea unui container Docker pentru baza de date poate face ca proiectul să fie mai ușor de portat între diferite medii de dezvoltare și de producție. Containerul poate fi configurat și testat local, apoi poate fi încărcat și rulat pe un server de producție fără a fi necesare alte configurări.

2. Consistența mediului: Containerul Docker poate asigura consistența mediului în care rulează baza de date, asigurând că toate dependențele și setările necesare sunt disponibile și corespund exact cu cele utilizate în timpul dezvoltării.

3. Izolare: Rularea bazei de date in Docker asigura o izolare mai mare între diferitele componente ale proiectului, precum aplicația și baza de date, ajutand la prevenirea unor eventuale probleme de securitate sau de performanță între aceste componente.

4. Scalabilitate: Utilizarea unui container Docker pentru baza de date poate face ca scalarea să fie mai rapida și mai eficientă, întrucât containerul poate fi clonat sau redimensionat odata cu cresterea complexitatii aplicatiei.

5. Automatizare: Datorita container-ului Docker, procesele de compilare, implementare si testare devin mai rapide.

Sistemul foloseste 12 entitati.

Entitatile User si Authority sunt folosite pentru securitatea aplicatiei, in special pentru autentificarea si autorizarea utilizatorilor in aplicatie. Acestea asigura accesul doar utilizatorilor cu permisiunile necasare la resursele din baza de date.

Tabela People este intr-o relatie one-to-one cu tabela Users si reprezinta entitatea folosita pentru a gestiona informatiile despre angajatii companiei. Aceasta permite extinderea funcționalității de gestionare a utilizatorilor, adăugând informații specifice despre angajat.

Entitatile Departments, Roles sunt legate prin intermediul unei relatii one-to-many cu entitatea People. Departments este folosita pentru a organiza structura companiei în diferite departamente, iar Roles reprezinta pozitia/functia pe care o are fiecare angajat in cadrul companiei.

Entitatea Companies este folosita pentru a gestiona informatiile despre o companie, fiind intr-o relatie one-to-many cu Departments.

Entitatile Projects, Teams, Phases, Sprints, Risks, Tasks sunt folosite pentru monitorizarea si controlul dezvoltarii proiectelor software.

Diagram, schematic

Description automatically generated

Fig. 2 - Diagrama bazei de date

**4.4 Stadiul curent al dezvoltarii**

Am implementat principalele endpoint-uri, care permit efectuarea operatiilor CRUD creare, citire, actualizare și ștergere pe fiecare entitate din proiect. Acestea sunt expuse de componenta Back-End a aplicatiei si accesate de componenta Front-End.

Pentru a asigura protectia si integritatea datelor stocate in aplicatie, am realizat sistemulul de autentificare si autorizare a utilizatorilor folosind framework-ul Java Spring Security JWT. Acest sistem de securitate utilizează token-uri JWT pentru a autentifica utilizatorii și pentru a-i autoriza să acceseze anumite resurse ale aplicației. Token-urile JWT sunt generate la momentul autentificării și sunt utilizate pentru a verifica identitatea și autorizațiile utilizatorilor în momentul accesului la date.

Am configurat un serviciu de mail, care trimite mail-uri bazate pe un template pentru diferitele activitati din aplicatie, precum activare a contului, creare a contului, resetare a parolei.

Pe partea de frontend, am creat o pagina de autentificare, ca punct de inceput in aplicatie, unde utilizatorii isi pot introduce credentialele pentru intra in contul lor. Am realizat o pagina de inregistrare, care permite utilizatorilor sa isi creeze un cont nou.

Am implementat inregistrarea si configurarea initiala a companiei, in cazul in care utilizatorul autentificat este directorul companiei, prin intermediul unei pagini care permite crearea unei companii noi si definirea departamentelor din cadrul acesteia.

Am creat un utilizator cu drepturi de administrator care are acces la toate functionalitatile sistemului, inclusiv pagini de gestionare a utilizatorilor si a entitatilor din proiect, o interfata pentru a urmari metricile din sistem, o interfata pentru a urmari starea aplicatiei si a bazei de date, o interfata pentru a monitorizarea tuturor cererilor care sunt trimise in aplicatie, o pagina pentru testarea tuturor endpoint-urile expuse de Back-End.

Pentru a testa corectitudinea și funcționalitatea aplicației, am implementat trei tipuri de teste: teste unitare, care verifică comportamentul și funcționarea individuală a fiecărei funcționalități, teste de integrare, care verifică modul în care diferitele componente ale aplicației interacționează între ele, și teste de performanță, care verifică cum funcționează aplicația la diferite sarcini.

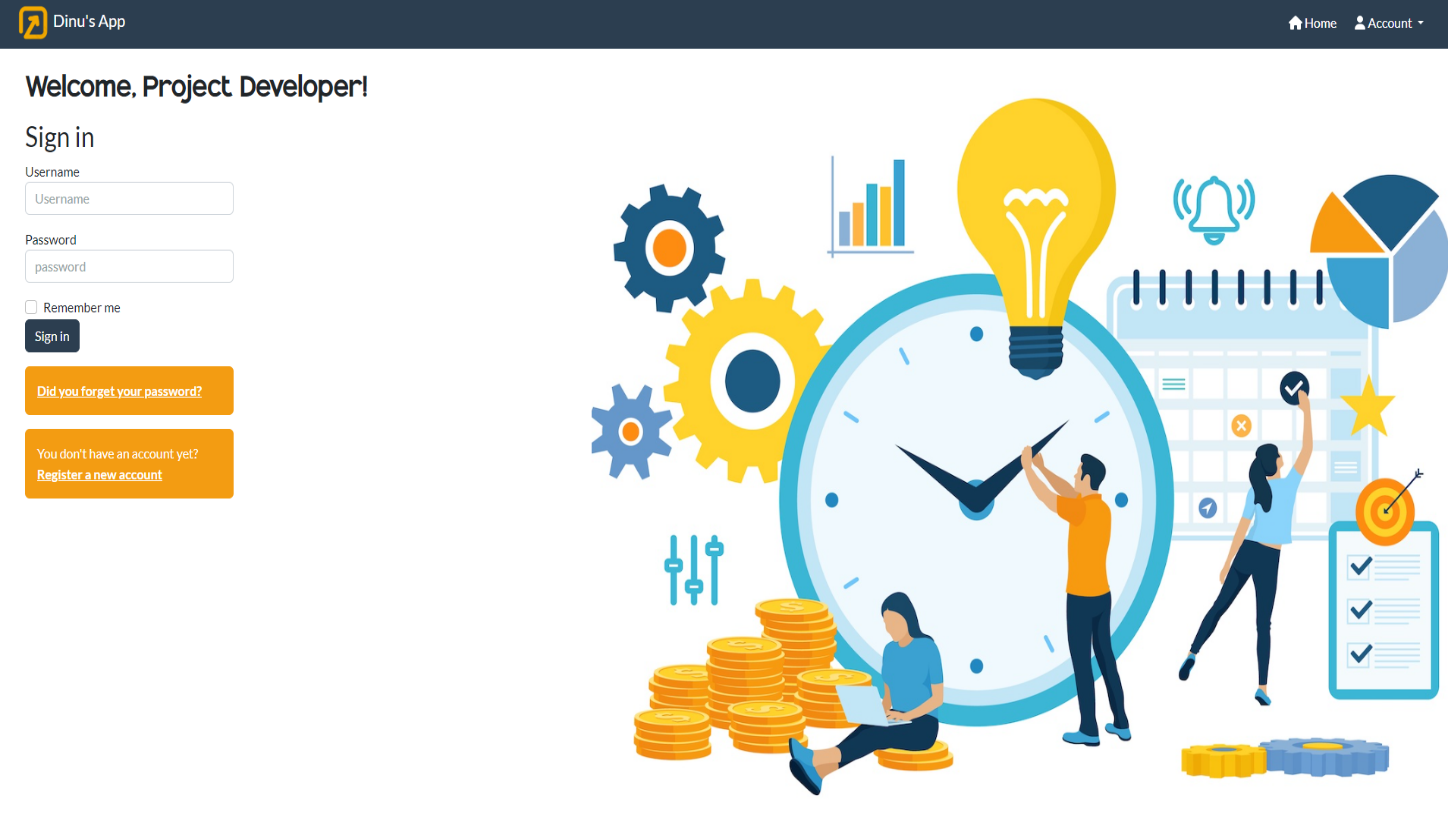


Fig 3 - Pagina de inceput

Graphical user interface, application

Description automatically generated

Fig 4 – Pagina pentru inregistrarea utilizatorilor

Graphical user interface, text, application, chat or text message

Description automatically generated

Fig 5 – Pagina pentru gestionarea utilizatorilor din aplicatie

Graphical user interface, text, application

Description automatically generated

Fig 6 – Pagina pentru gestionarea entitatii People

Graphical user interface, application

Description automatically generated

Fig 7 – Pagina pentru testarea endpoint-urilor expuse de Back-End

**5. Concluzii**

Proiectul "Monitorizarea si controlul proiectelor software" isi propune sa ofere o solutie eficienta pentru gestionarea dezvoltarii proiectelor software, prin intermediul unei platforme web. Astfel sistemul pune la dispozitie posibilitatea de a monitoriza si controla proiectele in timp real,de a gestiona organizarea companiei, de a planifica etapele dezvoltarii, de a urmari si controla costurile si stadiul proiectelor.

In concluzie, aplicatia propusa fi un instrument util pentru orice companie IT care doreste sa optimizeze procesele de dezvoltare a proiectelor software si sa asigure finalizarea lor cu succes.

**6. Bibliografie**

[1] Curs Inginerie Software – 2021-2022

[2] "PostgreSQL," Hosterion, 2021 <https://hosterion.ro/enciclopedie/gazduire/postgresql>

[3] PostgreSQL. (2022, April 28) Wikipedia <https://en.wikipedia.org/wiki/PostgreSQL>

[4] "What is Jira Software," Education Wiki, 2021. <https://ro.education-wiki.com/2603260-what-is-jira-software>

[5] Jira Software. (2022, April 17). Wikipedia <https://en.wikipedia.org/wiki/Jira_(software)>

[6] Asana, Inc. (2022, April 29). Wikipedia <https://en.wikipedia.org/wiki/Asana,_Inc>.

[7] "Java Spring Boot," IBM, 2022. <https://www.ibm.com/topics/java-spring-boot>

[8] Liquibase. (2022, April 21). Wikipedia <https://en.wikipedia.org/wiki/Liquibase>

[9] Angular. (2022, April 29).Wikipedia <https://ro.wikipedia.org/wiki/Angular#:~:text=Angular%20(numit%20%C8%99i%20%22Angular%202,de%20utilizatori%20individuali%20%C8%99i%20companii>

[10] PostgreSQL 2022, April 28 Wikipedia <https://en.wikipedia.org/wiki/PostgreSQL>