

**LUCRARE DE LICENȚĂ**

**Absolvent:** Bolba-Mateescu Andrei-Ioan

**Coordonator:** Vasilescu Anca

Brașov

Iunie-Iulie 2023

Cuprins

[1. Introducere 2](#_Toc136603788)

[1.1. Scurtă descriere a temei 2](#_Toc136603789)

[1.2. Motivația alegere lucrării 3](#_Toc136603790)

[1.3. Contribuții personale 3](#_Toc136603791)

[1.4. Structura lucrării 6](#_Toc136603792)

[2. Abordarea temei 6](#_Toc136603793)

[2.1. Scopul aplicatiei 6](#_Toc136603794)

[2.2. Alte abordări în domeniu 10](#_Toc136603795)

[3. Managementul entităților 10](#_Toc136603796)

[3.1. Modelarea entităților 10](#_Toc136603797)

[3.2. Faza de implementare a bazei de date 10](#_Toc136603798)

[3.3. Managementul identităților si al utilizatorilor 10](#_Toc136603799)

[4. Arhitectura aplicației 10](#_Toc136603800)

[4.1. Separarea preocupărilor 10](#_Toc136603801)

[4.2. Arhitectura REST pentru designul unui API 10](#_Toc136603802)

[4.3. Modelul “Clean Arhitecture” definit de Uncle Bob 10](#_Toc136603803)

[4.4. CRUD folosind verbe HTTP 10](#_Toc136603804)

[4.5. Securitate 10](#_Toc136603805)

[5. Noțiuni teoretice și tehnologii folosite 10](#_Toc136603806)

[5.1. .NET 7 11](#_Toc136603807)

[5.2. Angular 14](#_Toc136603808)

[5.3. Microsoft SQL 14](#_Toc136603809)

[5.4. Securitatea datelor 14](#_Toc136603810)

[6. Prezentarea aplicației 14](#_Toc136603811)

[6.1. Introducere 14](#_Toc136603812)

[6.2. Stuctura aplicației 14](#_Toc136603813)

[6.3. Persistența datelor 14](#_Toc136603814)

[7. Concluzii și perspective de viitor 15](#_Toc136603815)

[7.1. Concluzii 15](#_Toc136603816)

[7.2. Posibilități de extindere 15](#_Toc136603817)

[8. Bibligrafie 15](#_Toc136603818)

# Introducere

## Scurtă descriere a temei

Primul pas pe care un student îl face pentru a începe o carieră în domeniul IT este, de cele mai multe ori, un internship. Și asta se poate observa foarte ușor pe piața job-urilor în domeniul specificat mai sus. Fie ca vorbim de site-uri precum eJobs, BestJobs sau chiar LinkedIn, se pot observa multe astfel de oferte de internship de la companii care caută studenți pe care să îi învețe ce au nevoie pentru a putea lucra in compania respectiva. Evident nu toți studenții ce reușesc să înceapă un internship reușesc să și obțină un post în acea companie. Acest lucru este cauzat de procesul de urmărire a evoluției studenților, ce este efectuat de un mentor ce le este atribuit in prima zi de lucru la noua companie.

Astfel tema aleasă și anume InternHub vine în ajutorul mentorilor si al managerilor pentru o mai buna urmărire a evoluției si pentru a ușura alegerea când vine vorba de oferirea unei oferte in cadrul companiei. InternHub este o aplicație unde mentori au grupuri cu studenți asignați lor si pot comunica mai ușor, pot vedea evoluția lor si le pot răspunde la întrebări.

Lucrarea își propune ușurarea procesul de urmărire si totodată înțelegere a bitumului de însușire de noi informații pentru fiecare student in parte prin oferirea de task-uri periodice, la care se adaugă provocările zilnice pe care le oferă mentori studenților pe care ii au sub observare. Sistemul de provocări zilnice are la baza idea de motivare prin procesul de gamification. Acest proces utilizează principii precum competiția, recompensele, punctele, clasamentele, provocările și elementele narative pentru a crește angajamentul, motivația și participarea într-o anumită activitate sau context. Astfel fiecare provocare dusă la bun sfârșit reprezintă un punct. Astfel la finalul luni se face un clasament pentru a vedea cine a fost mai activ. Astfel studenții se dezvolta, iar mentori vad rezultatele, le interpretează si pot lua o decizie mai ușor la finalul programului de internship.

In plus, aplicația dispune de un sistem de forum, asemănător lui StackOverflow, unde atât studenți, mentorii cat si administratori pot pune întrebări, nelămuriri sau chiar curiozități, iar ceilalți utilizatori, indiferent de rol pot adaugă răspunsuri. Pentru a știi care răspunsuri sunt bune si care nu, toate postările si răspunsurile pot fii votate in consecința cu upvote, daca se considera o întrebare sau un răspuns relevant, respectiv downvote daca nu este relevant. Astfel postarea are un punctaj care se calculează in felul următor: upvotes – downvotes, iar astfel ne putem da seama de calitatea postării.

Tot in intermediul aplicației se pot planifica ședințe, atât pentru grupuri, cat si individual. Platforma de susținere a acestor ședințe este la alegerea companiei care va utiliza in viitor acesta aplicație, sistemul din cadrul acestui soft are nevoie doar de un link către platforma aleasa, fie ea Google Meet, Zoom sau Teams. Toate au in comun acest avantaj al link ului pentru a intra in conferința.

Un alt feature cheie al aplicației este sistemul de chat, care facilitează comunicarea intre utilizatori, putând sa fie făcute si grupuri. Comunicarea este un aspect important in cadrul dezvoltării tinerilor aflați la început de drum in acest domeniu, astfel InternHub vine in ajutorul lor prin aceste sistem ce comunicare in timp real intre utilizatori.

Toate aceste caracteristici fac din InternHub nu doar o simplă aplicație, ci una care are un scop bine stabilit si care dispune de uneltele necesare pentru a duce la împlinirea scopului propus in timp alocat programului de internship din cadrul companiei ce utilizează acest soft.

## Motivația alegere lucrării

Ideea lucrarii a plecat de la stagiul meu de practica, unde ca in multe locuri nu este inca implementat un sistem de genul acesta, deoarece acest lucru tine fiecare companie in parte. Lispa unui astfel de sistem s-a simtit deoarece feedback-ul nu vine asa de usor cum ne-am imagina, iar existaente unui portal intern de genul lui InternHub ar fii facut o diferenta.

Astfel motivatia alegeri acestei teme este un motiv personal, deoarece simt ca m-ar fii ajutat la cand eram in practica. In plus o aplicatie de acest tip are o aplicabilitate reala in toate companiile de profil IT, care au genul acesta de programe de internship, fie ca sunt pentru stagiile de practica fie ca sunt programe de internship cu scopul pregatirii noilor angajati sa lucreze cu uneltele companiei pe proiecte reale folosind modul de lucru din cadrul companiei.

Utilitatea reala in cadrul unei companii este unul din motivele pentru care am ales sa gandesc, dezvolt si sa implementez o astfel de aplicatie. In plus este o aplicatie destul de unica, dar despre asta o sa vorbim mai multe in subcapitolul 2.2, unde o sa vedem ce alte abordari mai sunt pe piata si daca acestea exista, fiind o aplicatie care este nisata sa rezolve o anumita problema.

## Contribuții personale

Aplicatia are o structura de tip full-stack formata din Angular, ca si front-end si .NET ca si back-end. Legat de stocarea datelor am folosit Microsoft SQL Server, dar mai multe despre structura aplicatiei vom discuta in capitolul 6, la subcapitolul 6.2.

Atat pe partea de Angular, .NET cat si Microsft SQL Server eu am gandit, proiectat si dezvoltat componentele neceasre unei aplicatii de acest tip. Puteam folosi Firebase, dar nu avem acceasi libertate cand vine vorba de creearea ahritecturii back-end-ului, si manipularii requestu-urilor pentru fiecare componenta importanta.

Acum voi trece prin fiecare dintre cele 3 puncte pentru a prezenta si explica contributiile personale pentru acest proiect si pentru a intelege de ce am optat pentru o varianta in detrimentul altei variante.

* **Front-end (Angular)**

În cadrul dezvoltării aplicației InternHub, am jucat un rol important în elaborarea componentei de front-end. Acesta a implicat mai multe responsabilități și a necesitat utilizarea unei varietăți de abilități și cunoștințe.

Am proiectat și dezvoltat interfața utilizator, asigurându-mă că este ușor de utilizat și estetic plăcută. În acest proces, am folosit si dezvoltat cunoștințele mele de UX și UI pentru a crea o interfață care să fie intuitivă și eficientă pentru utilizatori.

Un aspect central al contribuției mele a fost crearea și gestionarea componentelor Angular si comunicarea lor, reusind in multe cazuri sa evit apelarea de requesturi care ar fii ingreunat aplicatia. Am reusit acest lucru prin manipularea DOM-ului (Document Object Model) prin pasarea datelor intre copmmonente cu unelte speicifce Angular. Am folosit acest cadru de lucru pentru a structura aplicația într-o serie de componente reutilizabile, care au îmbunătățit claritatea și manevrabilitatea codului.

Am implementat funcționalități ale utilizatorului, asigurându-mă că acestea sunt funcționale și eficiente. Acest lucru a inclus implementarea sistemelor de autentificare și autorizare, a formularelor și a altor funcționalități esențiale. Legat de partea de autorizare si autentificare vom discuta mai multe la capitolul 5, subcapitolul 5.4 Securitatea Datelor. Am incercat pe cat posibil sa evit duplicarea codului prin utilizarea unei componente pentru doua sau mai multe sarcini asemenatoare, precum adaugarea si editarea unei entitati in baza de date.

Pentru partea de front-end contributiile au fost notabile deoarece este o componenta complexa din punct de vedere atat al structurii cat si al design-ului. Legat de comunicarea cu API-ul s-a bazat in totitalitate pe pe verbe HTTP, mai multe in capitolul 4 Arhitectura aplicatiei, la subcapirolul 4.4 CRUD folosind verbe HPTTP.

* **API (Framework .NET)**

Pentru partea de back-end, am proiectat si dezvoltat un API de la 0 folosindu-ma de design pattern-ul MVC (Model-View-Controller), impreuna cu alte design pattern-uri care au avut ca si rezultat un cod curat, usor de inteles si de parcurs.

Legat de acest pattern, si anume MVC, vom discuta mai mutle in capitolul X, subcapitolul X.X, deoarece are un rol esential in proiectarea unei arhitecturi back-end eficient, usor de inteles urmand principile de Clean Code.

Unul dintre aspectele cheie ale contribuției mele a fost proiectarea și dezvoltarea API-urilor în .NET. Am utilizat aceste API-uri pentru a crea punți de comunicare între front-end și back-end, permițând transmiterea datelor între aceste două componente într-un mod eficient și securizat.

Securitatea este o parte cruciala in crearea unui API si de aceea am acordat o atentie marita acestui luri prin implementarea diverselor strategii de securitate pentru a ma asigura că datele utilizatorilor sunt protejate și că numai utilizatorii autorizați au acces la informații sensibile. Acest lucru este foarte relevant in cadrul aplicatiei InternHub, deoarece exista mai multe tipuri de utilizatori cu drepturi de access diferite si cu drepturi de scriere diferita. Acest lucru a inclus utilizarea de token-uri JWT (JSON WebToken), care permite doar utilizatorilor aplicatiei sa aiba access la resursele oferite de catre back-end, și implementarea de politici CORS. Legat de dreputurile de acees acestea se verifica in momentul in care reuest-ul ajunge in back-end si in fuctie de rang-ul utilizatorul de executa cererea sau se intoarce codul de erorare 401 Unauthorized.

Un alt aspect important de care s-a tinut cont in crearea back-end-ul a fost gerstionarea logicii de business si comunicarea cu baza de date. Acesta a implicat crearea și gestionarea entităților și a relațiilor între ele, precum și implementarea de operațiuni CRUD (Create, Read, Update, Delete) pentru manipularea datelor. Acest lucru s-a realizat prin implementarea design-pattern-ului repository. Un repository reprezintă o colecție de entități (de exemplu, obiecte) și oferă metode pentru accesarea și manipularea acestor entități. De exemplu, un repository poate oferi metode precum getAllElements(), getElementById(), createElement(), editElement(), deleteElement(), etc. O sa discutam mai multe despre acest pattern in caipolul X, subacpitulul X.X.

Pentru proiectarea bazei de date s-a folosit Entity Framewok, dupa modelul code-first, ceea ce a dus ca modele din cadrul API-ului sa fie voluminoase, avand infomatii care nu era necesare in totalitate. Pentre a rezolva acest lucru am ales sa folosesc obiecte de DTO (Data Transfer Object) care au in componenta doar informatiile necesare in absolut toate reqest-urile, avand cate un DTO pentru fie care model. Evident avand acest tip de obiecte este necesare si o mapare, care datorile tool-ului numit AutoMapper din ecosistemul .NET s-a realizat cu usurinta fiind necesare doar declarea relatiilor intre obiectele mapate.

Crearea unui API de la zero in detrimentul utilizarii back-end-ului de la FireBase are mai multe avantaje. Prin acestea se numara flexibilitatea modelarii entitatilor si modelarea request-urilor. Modelarea entitatilor este un punct important in cadrul unui back-end, deorece putem modela obiectul trimis catrea front-end dupa nevoile specifica fiecarui request. In plus un API creat de la zero faciliteaza testarea eventualelor bug-uri prin procesul de debugging, care nu este la fel de facil pentru un API de tipul FireBase.

* **Baze de date (Microsoft SQL Server)**

Asa cum am specificat mai sus pentru baza de date s-a utilizat Entity Framework impreuna cu pattern-ul code-first. Acest lucru ma ajuta sa am un management mai bun al bazei de date si faciliteaza schimbarile din baza de date, daca acestea sunt necesare, fie pentru adaugarea unui camp nou, fie pentru adaugarea unui feature nou, ce necesita o table sau mai multe in plus.

Asa cum am afirmat si mai sus am utilizat modelul de creare a bazei de date code-first, ceea ce a însemnat că schema bazei de date a fost generată în cod folosind migrari pentru a crea baza de date in Microsoft SQL Server. Acest lucru a necesitat evident si gandirea relatiilor intrea tabele prin chei straine.

De asemenea, am implementat diverse tehnici pentru a asigura securitatea datelor. Acest lucru a inclus criptarea datelor sensibile, implementarea de controale de acces la nivel de date și utilizarea de tranzacții pentru a asigura integritatea datelor. Un exemplu concret este stocarea datelor sensibile precum parola utilizatorului care este stocata sub forma de 2 valori: valoarea rezultate in urma criptari si cheia dupa care s-a realizat criptarea. Mai multe despre acest subiect o sa discutam in capitolul 5, la subcapitolul 5.4 Securitatea Datelor.

Contributia personala in crearea bazei de date a fost resprectarea pasilor de crearea a bazei de date, fiecare pas avand cerinte si nevoie speciale. O sa trecem pe scurt prin pasii parcursi in crearea bazei de date folosite in aplicatie InternHub:

Primul pas a fost indetificarea nevoilor care se refera la ce tipuri de date sunt necesare pentru fiecare entitate si am ales cea mai buna varianta pentru fiecare tip. Putin mai problematica a fost stocarea de imagini dar pana la urma am gasit o solutie buna si pentru acesta problema.

Apoi am ajuns la etapa de proeictare conceptuala a bazei de date. In aceasta este etapa am definit strcutura bazei de date, relatiile intre tabele si tipurile de relatii intre tabele. Aici am utlizat modele conceptuale, precum diagramele Entitate-Relație (ER), pentru a defini entitățile din baza de date și relațiile dintre acestea.

Dupa aceata etapa am ajuns la proiectarea bazei de date unde am transformat modelul conceptual intr-un model logic de date. Tot aici am definit structura tabelelor, a câmpurilor, a relațiilor și a restricțiilor.

Urmatorul pas a fost acele de implementarea bazei de date unde am creat migrarile specifica pentru ficare entitate dupa modelul Fluent Migrator. Schema bazei de date a fost implementată urmand sa fie populate cu date de inceput pentru a putea face etapa urmatoare, si anume faza de testare si optimizare a bazei de date. Aici am testat performanța, securitatea și funcționalitatea bazei de date.

Toate aceste etape reprezintă un ciclu continuu, întrucât baza de date a fost modifcata pentru a fii in conformitate cu nevoile aplicatiei, find nevoita să evolueze odată acestea. Pentru fiecare modificare adusa bazei de date s-au urmat aceste etape si de accea spunem ca pasii prezentati mai sus reprezinta un ciclu continuu.

Contributiile personale pentru acesta aplcatie sunt destul de mari si acest lucru se datoreaza complexitatii aplicatiei.

## Structura lucrării

Lucrarea este impartita in 8 capitole care contin in mare urmatoarele informatii:

* Capitolul 1: Introducere unde se prezinta in pe scurt tema aleasa, motivatie pentru tema aleasa urmata de contributiil perosonala si structura lucrarii.
* Capitolul 2: Abordarea temei, unde se prezinta scopul aplicatiei si se discuta eventuale alte implementari existente pe piata.
* Capitolul 3: Managementul entitatilor ce prezinta prezinta baza de date, fazele de impementare si managementul utilizatorilor.
* Capitolul 4: Arhitectura aplicatiei ce prezinte cum este impartita aplicatia prin separarea preocuparilor, tipurile de request-uri folosite si modelul Clean Arhitecture prezentat de Uncle Bob.
* Capitolul 5: Tehnologi folosite este probail una dintre cele mai interesante parti a lucrarii deoarece descrie partea practica.
* Capitolul 6: Prezentarea aplicatiei unde se prezinta mai in detaliu structura aplicatiei, comunicarea intre framework-uri si persistenta datelor.
* Capitolul 7: Concluzii si perspective de viitor unde se ajunge la o concluzie legata de acesta aplcaitie si se prezinta perspectivele de viitor ale acestei aplicatii.
* Capitolul 8: Bibliografie un se regasesc link-uri utile pe care le-am folosit in elaborarea lucrarii si a aplicatiei.

# Abordarea temei

## Scopul aplicatiei

Aplicatia are ca și scop principal facilitarea monitorizării evoluției studenților aflați în internship la o companie. Majoritatea își caută un internship pentru că reușesc să castige niste bani, sa prinda niste experinta in acest domeniu si totodata sa isi faca practica de specialitate, care este materie obligatorie in cadrul facultatilor. Astfel la finalizarea internship-ului, stundentul primeste foaia ce ii atesta ca a facut stagiul de practica si dupa o discutie cu mentorul aceste reuseste, daca a avut rezultate bune sa prinda un post de programator ajutor pana la finalizarea studiilor.

Dupa finalizarea programului de practica si obtinerea caietului de practica completat, stundentul merge cu acel caiet la profesor iar acest pune nota pentru acesta materie, fara sa stie exact daca rezultatele de pe acel caiet sunt reale sau nu. Astfel InternHub isi propune sa includa si profesorul coordonator al practicii de specialitate in acest proces prin accesul la evolutia studentului. Astfel el isi poate da seama daca studentul chiar a avut rezultate bune, sau mai putin bune.

In plus aplicatie isi propune nu doar monitorizarea acestor rezultate, ci si imbunatatirea studentilor de trec prin compania respectiva, prin feedback-ul oferit de mentor si prin ajutorul celorlalti din cadrul echipei de intern prin raspunsul proactiv la discutiile deschide pe forum de catre toate lumea.

Scopul forumul este acela de ajutor reciproc in cadrul companiei prin raspunsul oamenilor mai exprimentati la intrebarile celor aflati la inceput de drum. Astefel ei afla noi informatii pe care le pot citii toti internii din companie. Totodata ei pot da raspunsuri care vor fii apreciate de cei mai experimentati, sau care vor fii corectate pentru a nu continua cu informatii eronate.

Acesta oferă o platformă de comunicare și de partajare a cunoștințelor care poate îmbunătăți eficiența și productivitatea. Iată câteva motive specifice pentru care o companie ar putea dori să implementeze un astfel de forum:

* Partajarea cunoștințelor: Asa cum am spus anterior un sistem de forum ajuta la partajarea cunostintelor, ce raman scrise atata timp cat aplicatie ruleaza in cadrul companiei, si este Partajarea cunoștințelor: Un forum de acest gen permite angajaților să pună întrebări și să primească răspunsuri de la colegii lor. Acesta poate fi un mod eficient de a partaja cunoștințele și de a învăța de la ceilalți.
* Documentarea soluțiilor: Odată ce o întrebare a fost răspunsă pe forum, acea informație devine disponibilă pentru toți ceilalți angajați, indiferent de statulul lor in companie, fie ca sunt interni sau mentori. Aceasta poate ajuta la prevenirea duplicării eforturilor și la asigurarea că soluțiile la probleme comune sunt ușor accesibile.
* Implicarea angajaților: Forumul permit angajaților sa arate proactivitate cand vine vorba de cei aflati la inceput de drum, ce le pot devenii colegii dupa aceasta perioada. Interesul companiei este sa dezvolte acesti tineri si sa ramana la ei in companie. De aceea aceasta proactivitate este esentiala pentru asta, iar forumul faciliteaza acest pas. Acest lucru poate îmbunătăți satisfacția la locul de muncă și poate încuraja angajații să se dezvolte profesional.
* Creșterea eficienței: Prin furnizarea de răspunsuri la întrebări comune, forumul ajuta la economisirea timpului folosit pentru cautarea solutiilor la probleme comune si poate reduce stresul angajatilor.
* Îmbunătățirea comunicării: Forumul ajuta la îmbunătățirea comunicării în cadrul companiei in care este utilizat oferind un spațiu unde angajații pot discuta și pot colabora pe diverse subiecte.

Sistemul de chat este un sistem intalnit in aproape orice aplicatie de pe acest pamant, deoarece faciliteaza comunicarea intre oameni. Sistemul implement aplicația InternHub isi propune acesta eficienta in comunicare, avand posibilitatea comunicarii de tip 1-la-1 dar si pe grupuri de interes. Aceasta isi doreste sa fie un sistem utilizat nu doar pentru partea de munca din cadrul companiei dar si pentru parte de timp liber prin crearea unor grupuri cu interese comune, fie ca vorbim de sport, moda, arta culinara sau chiar partea auto.

In plus acest sistem ofera cateva benefici chiei ce il ajuta la indeplinirea scopului sau final si anume eficienta. Printre aceste puncte tari ale sistemului de chat se numara:

* Comunicare rapidă: Chat-ul permite angajaților să comunice rapid unul cu celălalt, fără a fi nevoie să trimită e-mailuri sau să inițieze apeluri telefonice. Aceasta poate îmbunătăți eficiența, deoarece permite angajaților să rezolve rapid problemele și să răspundă la întrebări.
* Colaborare: Chat-ul faciliteaza colaborarea între echipe sau între membrii echipei. Angajații discuta despre proiecte, împărtăsesc idei și primi input în timp real de la toti membri echipei. Acest lucru este foarte impotant pentru un stundet aflat pe pozitie de internship, deoarece are suport din partea unor oameni cu experienta. Astfel procesul de asumare al informatii devine mult mai facil.
* Documentarea conversațiilor: Spre deosebire de conversațiile telefonice sau față în față, chat-urile se pasteaza si se pot accesa ulterior. Acest lucru este util pentru urmarirea procesului de invatare al stundentului dar si pentru procesul de monitorizarea al unui proiect.
* Disponibilitate: Chat-ul permite angajaților să rămână conectați și să comunice eficient, indiferent de locație sau fus orar.Desigur acest lucru este posibil si telefon, dar acesta vine cu anumite costuri ce difera in functie de locul de unde este initiat apelul. Esita si varianta aplicatii lor de tip WhatsApp, dar oameni prefera sa separe viata profesionala de cea personala, astfel un sistem de chat poate ajuta la acest lucru. Acest lucru poate fi deosebit de util pentru companiile cu angajați sau echipe la distanță.
* Îmbunătățirea moralului angajaților: Chat-ul intern din cadrul companiei contribuie la îmbunătățirea moralului și a culturii companiei. El ofera un loc pentru angajați să socializeze și să creeze o atmosferă mai prietenoasă și mai colaborativă, ceea ce duce la un mediu de lucru sanatos ce ajuta in timp eficienta si calitatea proiectelor.
* Suport tehnic intern: Chat-ul poate fi utilizat pentru a oferi suport tehnic rapid pentru angajați, in special celor alfati in primele luni intr-un context de genul acesta. Fiind prima interactiune cu mediul de lucru intrebarile ce apar in minte celor aflati la inceput sunt inevitabile si au nevoie de raspuns rapid. Chat-ul reuseste sa faca acest lucru prin raspunul scris, care ramane acolo si poate fii o sursa de documentare pentru o alta problema. Acesta poate fi un canal eficient pentru rezolvarea problemelor tehnice sau pentru a răspunde la întrebări.

Sistemul de challenge-uri zilnice, avand un sistem de gamification in spate, in colaborarea cu cel de task-uri mai lungi duce la cresterea eficientei lucrului la proiecte si totodata duce la dorinta de acumulare de informatii noi prin finalizarea task-urilor.

Sistemul de challenge-uri zilnice se bazeaza pe ideea completarii lor de catre interni si oferirea de puncte de catre mentor. Aceste puncte se vor folosi la crearea unui clasament, ce se actulizeaza pe zi ce trece si care, la final, poate oferii un premiu primilor 3 angajati. Acest pas este la alegerea companiei, iar implementarea lui duce la motivarea stundetilor sa fie cat mai competitivi si sa completeze cat mai bine acest challenge pentru acumulare de puncte cat mai multe.

Asemanator lui Jira, sistem de task-uri isi propune motinorizarea evolutiei studentilor prin completarea treptata a lor. Progresul poate fii vazut atat de catre mentor cat si de catre profesul resposanibil de stagiul de practica din facultatea din care face parte studentul.

Cum Jira s-a dovedit a fii eficient de-a lungul timpului, sistemul de task-uri din cadrul aplicatiei InternHub isi propune cat de eficient se poate raportat la Jira. Acest sistem ofera numeroase beneficii printrea care:

* Urmărirea progresului: Sistemul de task-uri permite mentorilor si profesorului resposabil să urmărească progresul studentului in cadrul perioadei de practica. Astfel se pot sesiza anumite lacune pe care studentul le are si se pot remedia usor atat din partea mentorului din cadrul companiei cat si din partea profesorului.
* Transparența: Sistemul de task-uri oferă transparență asupra a ceea ce se lucrează, cine lucrează la ce, și care sunt termenele. Aceasta poate duce la o mai bună coordonare și la o mai bună înțelegere a sarcinilor de către toată echipa.
* Eficiența: Prin structurarea și organizarea muncii, un sistem de task-uri poate ajuta la creșterea eficienței și la reducerea riscului de a uita sau a neglija sarcini importante.
* Comunicarea și colaborarea: Sistemul de task-uri faciliteaza comunicarea și colaborarea în cadrul unei echipe. Membrii echipei pot adăuga comentarii la sarcini, pot împărtăși actualizări și pot colabora pentru a rezolva probleme.

Sistem de feedback este un sistem vital pentru evoluția oricarui angajat din cadru unei companii, deoacere prin feedback oamenii evolueaza si pot observa greseli pe care altfel nu le-ar fii sesizat la fel de usor.

Un sistem de feedback are numeroase avantaje atat in cadrul unei echipe de lucru la un proiect cat si in cadrul unui grup de oameni. Printre avantajele sistemului de feedback din cadrul aplcaitiei InternHub se enumera:

* Îmbunătățirea performanței: Feedback-ul regulat ajută oamenii să înțeleagă ce fac bine și unde au nevoie de îmbunătățire. Astfel un student care nu stie foarte multe despre acest domeniu fiind la inceput poate evolua usor prin implementarea feedback-ului primit de la oamenii mai experimentati. Acest lucru le permite să își îmbunătățească abilitățile și să crească în performanță.
* Dezvoltare profesională: Feedback-ul ghidaza dezvoltarea profesională a oamenilor, indicându-le zonele în care au nevoie de mai multă instruire sau experiență. Astfel mentori stiu unde trebuie sa puna accent, iar profesorul responsabil poate folosii aceste informatii pentru a ajuta studentul in dezvoltarea sa cat timp mai este in facultate.
* Rezolvarea problemelor: Prin feedback, problemele pot fi identificate și rezolvate rapid. Acest lucru poate duce la o îmbunătățire a eficienței și a productivității.
* Recunoașterea muncii bine făcute: Un sistem de feedback oferă oportunități pentru recunoașterea și aprecierea oamenilor pentru munca lor bine făcută, ceea ce poate duce la creșterea moralului și a angajamentului față de companie.
* Comunicare deschisă: Un sistem de feedback poate ajuta la crearea unei culturi de comunicare deschisă și onestă în cadrul companiei.

Astfel aplicatia nu are ca scop doar monitorizarea procesului de evolutie al unui student si usurarea alegerii daca se continua colaboarare sau nu, ci si dezvoltarea tuturor angajatilor ce utilizeaza aceasta aplicatie in cadrul companiei.

## Alte abordări în domeniu

# Managementul entităților

## Modelarea entităților

## Faza de implementare a bazei de date

## Managementul identităților si al utilizatorilor

# Arhitectura aplicației

## Separarea preocupărilor

## Arhitectura REST pentru designul unui API

## Modelul “Clean Arhitecture” definit de Uncle Bob

## CRUD folosind verbe HTTP

## Securitate

# Noțiuni teoretice și tehnologii folosite

Aplicația InternHub este o aplicație full-stack, din punct de vedere al dezvoltării, fiind formată din două parți: frontend și backend. Persistarea datelor a fost facută utilizând Microsoft SQ Server.

Pentru partea de backend a fost folosit framework-ul .NET cu C# ca si limbaj de programare. Motivarea pentru acesta alegere va fii în urmatoarele pagini, începând cu subcapitolul 5.1. pentru partea de backend.

Frontend-ul a fost realizat utilizând Angular și ca și limbaj de programare s-a folosit typescript. De asemenea motivarea pentru folosirea acestui framework o vom aborda la subcapitolul 5.2.

Pentru persistența datelor s-a folosit Microsoft SQ Server împreună cu Entity Framwrok pentru comunicarea backend-ului cu baza de date. Și pentru acesta avem motivarea alegerii la subcapitolul 5.3.

## .NET 7

.NET 7 este successoul lui .NET 6 și se focusează să fie unificat, modern, simple și de asemenea rapid. .NET 7 o să aibă suport tehnic pentru o perioadă de 18 luni, termen ce este standard in cadrul ecosistemului .NET. Lansat pe data de 8 noiembrie 2022, acesta va avea parte de suport tehnic până la data de 14 mai 2024.

Printre noile caracteristici aduce de la .NET 6 la .NET 7 se enumeră performanța, serializarea JSON, matematica generică, expresii regulate, noi librării .NET, sdk .NET și de asemenea actulizări ce rezolva unele probleme ale subsistemelor ce folosesc ecosistemul .NET.

Pentru a întelege mai în detaliu ce înseamnă aceste noi caracteristici pentru .NET 7, o să le dezvoltăm puțin pe unele dintre cele prezentare mai sus:

* **Performața**: Performanța este una dintre caracteristicile cheie ale .NET 7, iar toate caracteristicile sale sunt concepute ținând cont de performanță. În plus, .NET 7 include îmbunătățiri destinate exclusiv performanței pentru următoarele îmbunătățiri: înlocuirea pe stivă (On-stack replacement), optimizare ghidată pe profil (Profile-guided optimization), imbunatățirea generării de cod pentru Arm64 și performanța îmbunatățită pentru timpul de rulare mono
* **Serializarea JSON:** .NET 7 include îmbunătățiri ale serializării System.Text.Json în următoarele domenii: serializarea polimorfică și suport pentru membrii necesari
* **Matematica generică**: .NET 7 și C# 11 includ inovații care vă permit să efectuați operații matematice în mod generic.Scrierea unei funții care are ca scop efectuarea unor operații matematice pentru două numere se simplifică, deoarece nu mai este nevoie supraîncărcarea metodei pentru fiecare tip dorit. Acum este posibilă scrierea unei metode generice în care parametrul tip este constrâns să fie un tip asemănător unui număr.
* **Expresii regulate**: Biblioteca de expresii regulate a .NET a înregistrat îmbunătățiri semnificative funcționale și de performanță în .NET 7 [1]

Ecosistemul .Net este un ecosistem dezvoltat si menținut de către Microsoft care rulează în principal pe Microsoft Windows. Dezvoltarea acestiu ecosistem a început la finalul anilor 1990, sub alt nume față de cel cunoscut astăzi. Inițial numit Next Generation Windows Services, era parte a strategiei .NET. Odată cu începutul anului 2000, primele versiuni beta ale lui .NET 1.0 au fost lansate, urmând ca in februarie 2002 să fie anunțată și totodată lansată prima versiune a .NET, și anume .NET 1.0. În prezent framework-ul se alfă la versiunea .NET 8, urmând ca în noiembrie 2023 să fie lansata versiunea .NET 8.

Cel mai mare avantaj al framework-ului .NET este că acceptă platforma Windows. Aproape toată lumea lucrează cu mașini Windows. [2]

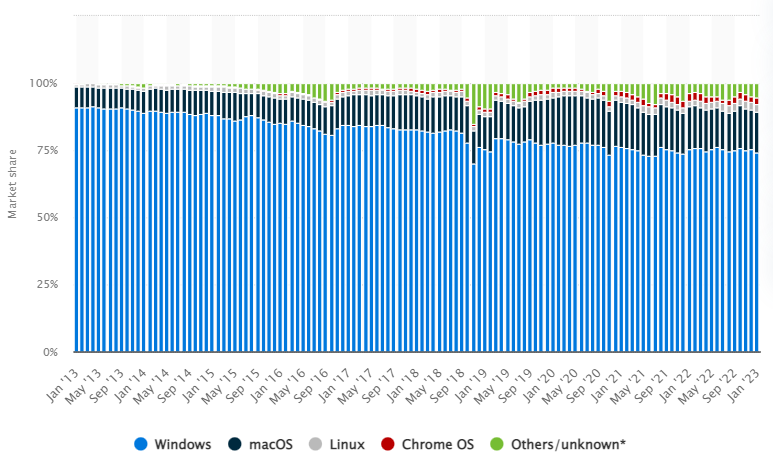


Figura 5.1 - Cota de piață globală deținută de sistemele de operare

Conform figurii 5.1 se poate observa că sistemul de operare preferat în întreaga lume este windows cu un procetaj constant mai mare de 74% în ultimii 10 ani. [3]

Un alt argument în favoarea ecosistemului .NET este constanța lansării de noi versiuni ale acestui ecosistem, și totodată menținerea suportului pentru versiunile lansate anterior versiunii curente, pentru a face mai ușoară trecerea de la o versiune la alta. La momentul actual acest timp de suport tehnic este de 6 luni după lansarea ultimei versiuni.

Pentru lucrarea în cauză am ales să lucrez cu .NET 7, deoarece este cea mai nouă și care are avantaje considerabile în fața altor opțiuni. Un alt motiv pentru .NET 7 este și incheierea suportului tehnic pentru .NET 3.1 în decembrie 2022, la puțin timp după lansarea lui .NET 7, iar tendința generală în cadrul companiilor este de a migra codul de la .NET 3.1 la .NET 6 sau .NET 7. Motivul pentru .NET 6 este că, asemenea lui .NET 3.1 este o versiune LTS (Long Term Suport), ceea ce îi oferă suport până în noiembrie 2024. [4]

Motivul pentru alegerea verisunii curent in detrimentul unei versunii LTS este, pe lângă cele prezentate la începutul acestui subcapitol este faptul că am dorit sa experimentez ultima versiune.

C# este un limbaj de programare care acceptă mai multe paradigme.Inițial lansat în 2000, acesta a ajuns deja la versiunea 11 lansată odată cu .NET 7 în noiembrie 2022. Asemănător lui .NET, C# primește contant actulizări și lansări de noi versiuni, astfel fiind o alegere optimă pentru partea de backend și totodată pentru frontend. [5]

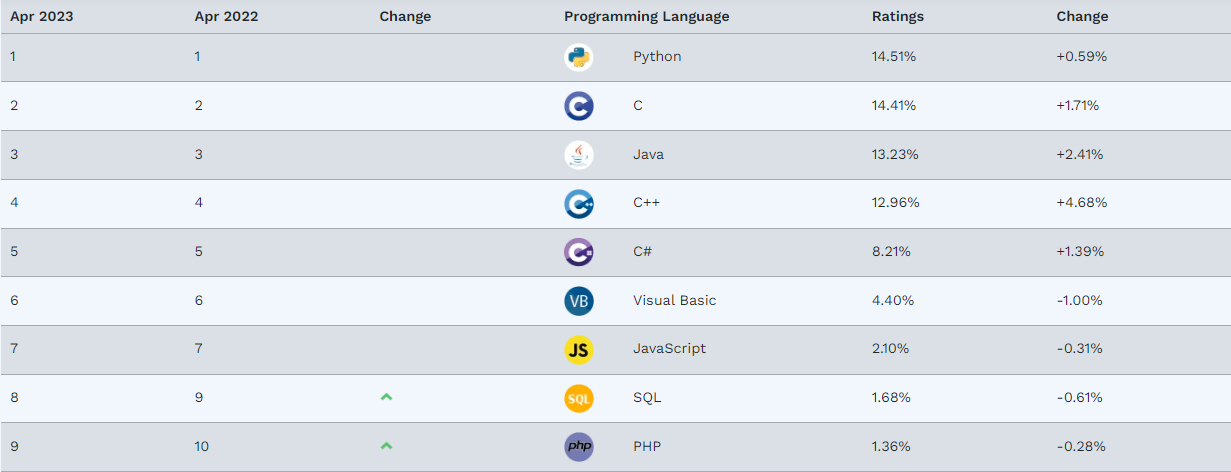


Figura 5.2 – TIOBE Programming Community Index

Conform figurii 5.2 se poate observa că deși C# nu este cel mai popular limbaj de programare este în continuă creștere fiin al 5-lea cel mai utilizat limbaj de programare. În iunie 2001, la puțin timp după lansare avea un raiting de 0,84%, iar este la 8,21% cu o creștere de 1.39% în ultimul an. [6]

Legat de API, pentru acesta s-a folosit ASP.NET Core Web API. Conform stasticilor făcute de cei la stack overflow din 2022 legată de framewrok-uri de dezvoltare pentru aplicatii web ASP.NET Core și ASP.NET sunt destul de populare fiind imediat după Angular și Vue.js în rândul tuturor celor intervievați, în număr de 58 743, așa cum se poate observa în figura 5.3. În rândul programatoril experimentați ASP.NET Core urcă o poziție deasupra lui Vue.js, iar ASP.NET rămâne pe acceți pozitie. Numărul programatoril experimentați intervievași de către cei de la Stackoverflow este puțin mai mic, dar rămâne un numâr consderabil de persoane intervievate, în număr de 45 297 , așa cum se poate observa în figura 5.4. În schimb cei ce sunt la început de drum aleg mai puțin ASP.NET și ASP.NET Core. Numărul celor intervievați ce se află la început de carieră este de 4932, așa cum se poate observa în figura 5.3. Cu toate acestea primele 2 opțini sunt la fel, fie ca vorbim de utilizator cu experința fie că vorbim de cei la început de drum. [7]

|  |  |
| --- | --- |
| Figura 5.3 – Toate răpunsurile la sondaj | Figura 5.4 – Răspunsurile programatorilor experimentați la sondaj |
| Figura 5.4 - Răspunsurile programatorilor aflați la început de carieră la sondaj |  |

## Angular

## Microsoft SQL

## Securitatea datelor

# Prezentarea aplicației

## Introducere

## Structura aplicației

## Persistența datelor

# Concluzii și perspective de viitor

## Concluzii

## Posibilități de extindere

# Bibligrafie

[1] What's new in .NET 7, <https://github.com/dotnet/docs/blob/main/docs/core/whats-new/dotnet-7.md>

[2] .NET, https://en.wikipedia.org/wiki/.NET

[3] Global market share held by operating systems for desktop PCs, from January 2013 to January 2023, <https://www.statista.com/statistics/218089/global-market-share-of-windows-7/>

[4] .NET and .NET Core Support Policy, <https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core>

[5] C Sharp (programming language), <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>

[6] C# in 2023: The MOST POPULAR Programming Language?, <https://www.bytehide.com/blog/c-wants-to-become-the-most-popular-programming-language-in-2022>

[7] 2022 developer survey, <https://survey.stackoverflow.co/2022/#most-popular-technologies-webframe>